Convolutional Spatial Attention Model for Reading Comprehension with Multiple-Choice Questions

Zhipeng Chen†, Yiming Cui†‡∗, Wentao Ma†, Shijin Wang†, Guoping Hu†
†Joint Laboratory of HIT and iFLYTEK (HFL), iFLYTEK Research, Beijing, China
‡Research Center for Social Computing and Information Retrieval (SCIR), Harbin Institute of Technology, Harbin, China
∗{zpchen,ymcui,wtma,sjwang3,gphu}@iflytek.com

Abstract

Machine Reading Comprehension (MRC) with multiple-choice questions requires the machine to read given passage and select the correct answer among several candidates. In this paper, we propose a novel approach called Convolutional Spatial Attention (CSA) model which can better handle the MRC with multiple-choice questions. The proposed model could fully extract the mutual information among the passage, question, and the candidates, to form the enriched representations. Furthermore, to merge various attention results, we propose to use convolutional operation to dynamically summarize the attention values within the different size of regions. Experimental results show that the proposed model could give substantial improvements over various state-of-the-art systems on both RACE and SemEval-2018 Task11 datasets.

Introduction

Owing to the rapid release of various large-scale datasets, Machine Reading Comprehension (MRC) has become enormously popular in Natural Language Processing. For example, cloze-style MRC (such as CNN/DailyMail (Hermann et al. 2015), Children’s Book Test (CBT) (Hill et al. 2015)), span-extraction MRC (such as SQuAD (Rajpurkar et al. 2016)), and multiple-choice MRC (such as MCTest (Richardson, Burges, and Renshaw 2013), RACE (Lai et al. 2017), SemEval-2018 Task11 (Ostermann et al. 2018)).

In this paper, we mainly focus on solving the reading comprehension with multiple-choice questions. At the beginning of the reading comprehension study, this type of reading comprehension task was not that popular because there is no large-scale dataset available and thus we cannot apply neural network approaches to solve them. To bring more challenges to reading comprehension task and mitigate the absence of large-scale multi-choice reading comprehension dataset, Lai et al.(2017) propose a new dataset called RACE. Compared to the earlier MCTest (Richardson, Burges, and Renshaw 2013), the RACE dataset is made from the English examinations for Chinese middle and high school students, consisting near 100,000 questions generated by human experts, and is far more challenging than the MCTest.

In this paper, we propose a novel model called Convolutional Spatial Attention (CSA) to fully utilize the hierarchical attention information for reading comprehension with multiple-choice questions. The proposed model first encode the passage, question, and candidates into word representations which are enhanced by the additional POS-tag and matching features. Then we concentrate on enriching the representation of the candidates by incorporating the passage, question information, and further calculate the attentions between the passage, question, and candidates, forming the spatial attentions. To further extract the representative features in the spatial attentions, we propose to use convolutional neural network to dynamically conclude adjacent regions with different window size. We mainly test our CSA model on two multiple-choice reading comprehension datasets: RACE and SemEval-2018 Task11, and our model achieves state-of-the-art performances on both of them. The examples of each dataset are given in Figure 1. The main contributions of our paper can be summarized as follows.

- We focus on modeling different semantic aspects of the candidates, by integrating the passage and question information, forming the 3D spatial attention among the passage, question, and candidates.
- We propose a Convolutional Spatial Attention (CSA) mechanism to dynamically extract representative features from the spatial attentions.
- The proposed model gives substantial improvements over various state-of-the-art systems on both RACE and SemEval-2018 Task11 datasets, showing its generalization and extensibility to other NLP tasks.

Related Works

Massive progress has been made on machine reading comprehension field in recent years. The booming of the MRC can trace back to the release of the large-scale datasets, such as CNN/DailyMail (Hermann et al. 2015) and CBT (Hill et al. 2015)). After the release of these datasets, various neural network approaches (Chen, Bolton, and Manning 2016; Kadlec et al. 2016; Cui et al. 2017; Dhingra et al. 2017) have been proposed and become fundamental components in the future studies. Another representative dataset is SQuAD (Rajpurkar et al. 2016), which was difficult than the cloze-style reading comprehension and requires the machine to
generate a span in the passage to answer the questions. With rapid progress on designing effective neural network models (Xiong, Zhong, and Socher 2016; Seo et al. 2016; Wang et al. 2017; Hu, Peng, and Qiu 2017), recent works on these datasets have surpassed the average human performance, such as QANet (Yu et al. 2018) etc.

However, current machine reading comprehension models are still struggling with solving the questions that need reasoning over multiple sentences or even passage. To solve the reading comprehension with multiple-choice questions, various approaches have been proposed, and most of them are focusing on designing effective attentions or persuading enriched representations for prediction. When releasing the RACE dataset, Lai et al.(2017) also adopted and modified two models of the cloze-style reading comprehension: Gated Attention Reader (Dhingra et al. 2017) and Stanford Attention Reader (Chen, Bolton, and Manning 2016). However, experimental results show that these models are not capable of this task. Parikh et al.(2018) introduced ElimiNet which use a combination of elimination and selection to get refined representation of the candidates. Xu et al.(2017) proposed the Dynamic Fusion Networks (DFN), which uses multi-hop reasoning mechanism for this task. Zhu et al.(2018) proposed the Hierarchical Attention Flow model, which leverage candidate options to model the interactions among passage, questions, and candidates.

Though various efforts have been made, we believe there is still a large room for designing effective neural networks for better characterizing the relations between the passage, questions, and candidates. To this end, we summarize the main differences between our model and existing models for this task into three aspects. First, we calculate various representations of the candidate to better characterize it for prediction. Second, when calculating attention, we apply additional trainable weights to dynamically adjust the attention values, which is more flexible. Third, unlike the previous works only utilize the final-level hierarchical attentions, we propose to use every attention in each hierarchy, and use convolutional neural network to capture features for predicting the answer.

## Convolutional Spatial Attention Model

### Task Definition

The RACE dataset (Lai et al. 2017) for reading comprehension with multiple-choice questions was proposed, which consists of 28,000+ passages and near 100,000 questions generated by human experts for the English examinations of Chinese middle and high school students. Different from the earlier MCtest dataset (Richardson, Burges, and Renshaw 2013), the RACE dataset is significantly larger, and thus we can apply deep learning approaches for this task. As all the questions and choices are generated by human experts, RACE dataset provides more comprehensive and realistic evaluation on machine reading comprehension than the other popular datasets such as CNN/DailyMail, SQuAD datasets, whose answer should appear in context. Also according to the analysis by Lai et al.(2017), a large portion of the questions in RACE need reasoning over various clues, which makes it more challenging and suitable to evaluate the ability of the reading comprehension systems. SemEval-2018 Task11 is closely the same with the RACE dataset but with two candidates and small size. The examples of each dataset are given in Figure 1.

### The Model

In this section, we will give a detailed description on the proposed model. The main neural architecture of our model is depicted in Figure 2. Throughout this section, we will use $P$ for representing the passage, $Q$ for the question, $C$ for the candidates. Note that, as the operations on each candidate are the same, for simplicity, we only take one of the candidates for illustration.

### Word Representation

We transform each word in the passage, question, and candidates into continuous representations. In this paper, there are three components in the embedding layer, which can be listed as follows.

- **Word Embedding** $E_{word}$: We use traditional pre-trained GloVe embedding for initialization (Pennington, Socher,
and Manning 2014) and keep fixed during the training process.

- **ELMo Embedding** $E_{elmo}$: For this part, we use pre-trained ELMo (Peters et al. 2018) embedding.

- **Feature Embedding** $E_{feat}$: We also utilize three additional features to enhance the word representations.

  1. **POS-tag Embedding** $E_{pos}$: We use NLTK (Bird and Loper 2004) for part-of-speech tagging for each word. Similar to traditional word embeddings, we assign different trainable vectors for each part-of-speech tag.

  2. **Word Matching** $F_{match}$: Take the text as an example, if the word in text also appears in question or candidate, we set the value as one, otherwise set it as zero. In this way, we can also add this feature to the question and candidate.

  3. **Fuzzy Word Matching** $F_{fuzzy}$: Similar to the word matching feature, but we loosen the matching criteria as partial matching. For example, we regard ‘teacher’ and ‘teach’ as fuzzy matching, because the string ‘teacher’ is partially matched by ‘teach’.

We concatenate three embedding components to form the final word representations for the text $E_P \in \mathbb{R}^{|P| \times e}$, question $E_Q \in \mathbb{R}^{|Q| \times e}$, and candidates $E_C \in \mathbb{R}^{|C| \times e}$, where $|P|, |Q|, |C|$ are the length of the passage, question, and candidates, $e$ is the final embedding size (including all three components).

$$E = [E_{word}; E_{elmo}; E_{feat}] \quad (1)$$

$$E_{feat} = [E_{pos}; F_{match}; F_{fuzzy}] \quad (2)$$

After obtaining embedding representations, we further feed each word embedding into a shared highway network (Srivastava, Greff, and Schmidhuber 2015) with $\tanh$ output activation (denoted as $\sigma$). In this paper, we apply two consecutive highway networks with shared weights. Then we use Bi-Directional LSTM (Graves and Schmidhuber 2005) to model the contextual information, forming $H_P \in \mathbb{R}^{|P| \times h}$, $H_Q \in \mathbb{R}^{|Q| \times h}$, and $H_C \in \mathbb{R}^{|C| \times h}$ ($h$ is hidden size of Bi-LSTM). Note that, we use different Bi-LSTMs for the passage, question, and candidates.

$$\tilde{H} = \sigma(2\text{-Highway}(E)) \quad (3)$$

$$H = \text{Bi-LSTM}(\tilde{H}) \quad (4)$$

### Enriched Representation

Calculating attention and generating enriched representation play very important roles in machine reading comprehension. In our model, we will calculate various types of enriched representations for better characterizing the candidate and question, which are the essential components in this task. The procedure for generating enriched representation is illustrated in Algorithm 1.

For example, we wish to embed the passage information into the candidate representation to better aware the relevant part in the passage and obtain the passage-aware candidate representation $R_{CP}$. According to Algorithm 1, $R_{CP}$ can be generated as follows.

- **[Line 1]** Given the Bi-LSTM representations of passage $H_P$ and candidate $H_C$, we first calculate the attention matrix where each element indicate the matching information between them. In this paper, we adopt the attention
Algorithm 1 Enriched Representation.

Input:
- Time-Distributed representation $X_1$
- Time-Distributed representation $X_2$

Initialize:
- Random weight matrix $W_1 \in \mathbb{R}^{h \times h_{att}}$
- Random weight matrix $W_2 \in \mathbb{R}^{h \times h_{att}}$
- Diagonal weight matrix $D \in \mathbb{R}^{h_{att} \times h_{att}}$
- All-one weight matrix $W \in \mathbb{R}^{\{X_1 \times |X_2|\}}$

Output: $X_2$-aware $X_1$ representation $Y$

1. Calculate attention matrix $M' \in \mathbb{R}^{\{X_1 \times |X_2|\}}$:
   \[
   M' = f(W_1 X_1^T) \cdot D \cdot f(W_2 X_2^T)
   \]
2. Apply element-wise weight: $M = M' \odot W$
3. Apply softmax function to the last dimension of $M$:
   \[
   M_{att} = \text{softmax}(M)
   \]
4. Calculate raw representation $Y' \in \mathbb{R}^{X_2 \times h}$:
   \[
   Y' = M_{att}^T \cdot X_1
   \]
5. Concatenate raw representation $Y'$ and raw input $X_1$,
   then apply Bi-LSTM:
   \[
   Y = \text{Bi-LSTM}([X_1; Y'])
   \]
6. return $Y$

Convolutional Spatial Attention

With previously generated representations, we can calculate the matching matrix to measure the similarity between them. In this paper, we adopt simple dot product to obtain the matching matrix. As the candidate information is important to answer the question, firstly we calculate the matching matrix using various candidate representations to the self-attended question representation $R_{self-Q}$. The motivation is to use the question information as the key to extracting candidate information in different levels. We use question-aware candidate representation $R_{CQ}$, passage-aware candidate representation $R_{CP}$, and candidate Bi-LSTM representation $H_C$, as shown below.

\[
\begin{align*}
M_{11} &= R_{CQ} \cdot R_{self-Q} \\
M_{12} &= R_{CP} \cdot R_{self-Q} \\
M_{13} &= H_C \cdot R_{self-Q}
\end{align*}
\]

In a similar way, we can also replace the self-attended question representation $R_{self-Q}$ with passage-aware question representation $R_{QP}$ in Equation 9, 10, 11, to obtain $M_{21}, M_{22}, M_{23}$. Then we concatenate all matrices on the channel dimension to form a Spatial Attention Cube $M \in \mathbb{R}^{6 \times |C| \times |Q|}$, which is similar to an ‘image’ with 6 channels.

\[
M = [M_{11}; M_{12}; M_{13}; M_{21}; M_{22}; M_{23}]
\]

In order to extract high-level features inside the spatial attention cube, we use CNN-MaxPooling operation to dynamically conclude adjacent attention information, which is similar to the traditional operation on the image. Formally, we first apply convolutional operation on the $M$ to summarize different length of adjacent elements. We adopt three convolutional kernels: 5, 10, and 15, along with the dimension of the question length.

After convolution operation, we could get three features maps w.r.t. different convolutional kernels. The procedure can be illustrated as the following equations.

\[
\begin{align*}
O_1 &= \text{Max-Pooling}_{5 \times 3}\{\text{CNN}_{1 \times 5}(M)\} \\
O_2 &= \text{Max-Pooling}_{1 \times 2}\{\text{CNN}_{1 \times 10}(M)\} \\
O_3 &= \text{Max-Pooling}_{1 \times 1}\{\text{CNN}_{1 \times 15}(M)\}
\end{align*}
\]

In this way, we used CNN-MaxPooling operation to obtain three feature vectors $O_1, O_2, O_3$ by using different convolutional kernels and max-pooling intervals.

Final Prediction

After obtaining three feature vectors, we flatten, concatenate, and feed them into a fully-connected layer to get a scalar value denoting the possibility of being the correct answer. Recall that, we have several candidates for a given question, so we will get $N$ candidate scores in this stage. We apply softmax function to these scores to obtain the final probability distributions over the candidates.

\[
s_i = w^T \cdot [O_1; O_2; O_3]
\]

\[
Pr(A|P, Q, C) = \text{softmax}([s_1; \ldots; s_N])
\]

To train our model, we use traditional cross entropy loss to minimize the gap between the prediction and the ground truth.
Experiments

Experimental Setups

To evaluate our system, we carried out experiments on the following two public datasets.

- **RACE**: English examinations for Chinese middle and high school students. The questions are generated by human experts, which has four candidates for each question. The test set consists of 4,934 instances, where RACE-M (middle school) has 1,436 instances, and RACE-H (high school) for 3,498 instances.

- **SemEval-2018 Task11**: The dataset provided by the SemEval-2018 Task11 organizer, which mainly focus on solving commonsense reading comprehension. Each question has two candidates to choose from.

The data are tokenized and lower-cased by using Natural Language Toolkit (NLTK) (Bird and Loper 2004), and all punctuations are removed. The main hyper-parameters of our model are listed in Table 2. Note that, except for the candidate numbers, all hyper-parameters are identical among two datasets. The word embeddings are initialized by the pre-trained GloVe word vectors (Common Crawl, 6B tokens, 100-dimension) (Pennington, Socher, and Manning 2014), and keep fixed during training. The words that do not appear in the pre-trained word vectors are set to the unk token and initialized accordingly. We use Adam (Kingma and Ba 2014) for weight optimizations with an initial learning rate of 0.001. In order to prevent overfitting, we apply dropout of 0.35 to all the representation layers. The models are built on Keras platform (Chollet and others 2015) with Tensorflow backend (Abadi et al. 2016).

Overall Results

**RACE.** The experimental results are shown in Table 1. As we can see that, our CSA model shows significant improvements over various state-of-the-art systems by a large margin. We also compared our model to the recent unpublished work DFN, while our model gives an absolute gain of 1.0% in the overall test set, and further gains can be obtained by incorporating ELMo, demonstrating the effectiveness of our proposed model. Also, when compared to the Hierarchical Attention Flow model (Zhu et al. 2018), our CSA model show substantial improvements, indicating that utilizing the attentions in each hierarchy and using convolutional neural network to extract the most representative features from the spatial attentions are useful in this task.

When it comes to the ensemble results, though we only use 7 models in ensemble with majority voting approach, the overall results show an absolute gain of 2% over the previous state-of-the-art result by DFN. Also we observed that our model shows slightly worse result in RACE-M but significantly better in RACE-H regardless of single model or ensemble. These results indicate that our model is more capable of solving difficult question (high school). While on the contrary, it may hurt the performance on the relatively easier question (middle school). We will give a detailed analysis for illustrating this phenomenon in the next section.

**SemEval-2018 Task11.** The distributions of question type is quite different between the RACE and SemEval-2018 Task11 datasets. To test if our model could generalize to

Table 1: Experimental results on RACE. The best previous results are in italics, and overall best results are in bold face.

<table>
<thead>
<tr>
<th>Model</th>
<th>RACE-M</th>
<th>RACE-H</th>
<th>RACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sliding Window (Lai et al. 2017)</td>
<td>37.3</td>
<td>30.4</td>
<td>32.2</td>
</tr>
<tr>
<td>Stanford AR (Lai et al. 2017)</td>
<td>44.2</td>
<td>43.0</td>
<td>43.3</td>
</tr>
<tr>
<td>GA Reader (Lai et al. 2017)</td>
<td>43.7</td>
<td>44.2</td>
<td>44.1</td>
</tr>
<tr>
<td>ElimiNet (Parikh et al. 2018)</td>
<td>N/A</td>
<td>N/A</td>
<td>44.5</td>
</tr>
<tr>
<td>Hierarchical Attention Flow (Zhu et al. 2018)</td>
<td>45.0</td>
<td>46.4</td>
<td>46.0</td>
</tr>
<tr>
<td>Dynamic Fusion Network (Xu et al. 2017)</td>
<td>51.5</td>
<td>45.7</td>
<td>47.4</td>
</tr>
<tr>
<td>CSA Model (single model)</td>
<td>51.0</td>
<td>47.3</td>
<td>48.4</td>
</tr>
<tr>
<td>CSA Model + ELMo (single model)</td>
<td>52.2</td>
<td>50.3</td>
<td>50.9</td>
</tr>
<tr>
<td>GA Reader (6-ensemble)</td>
<td>-</td>
<td>-</td>
<td>45.9</td>
</tr>
<tr>
<td>ElimiNet (6-ensemble)</td>
<td>-</td>
<td>-</td>
<td>46.5</td>
</tr>
<tr>
<td>GA + ElimiNet (12-ensemble)</td>
<td>-</td>
<td>-</td>
<td>47.2</td>
</tr>
<tr>
<td>Dynamic Fusion Network (9-ensemble)</td>
<td>55.6</td>
<td>49.4</td>
<td>51.2</td>
</tr>
<tr>
<td>CSA Model (7-ensemble)</td>
<td>55.2</td>
<td>52.4</td>
<td>53.2</td>
</tr>
<tr>
<td>CSA Model + ELMo (9-ensemble)</td>
<td>56.8</td>
<td>54.8</td>
<td>55.0</td>
</tr>
</tbody>
</table>

Table 2: Hyper-parameter settings.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Descriptions</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Passage max length</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>Question max length</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Candidate max length</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Word embedding</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>Bi-LSTM hidden size</td>
<td>250</td>
</tr>
<tr>
<td></td>
<td>Attention hidden size</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>ELMo embedding size</td>
<td>1024</td>
</tr>
<tr>
<td></td>
<td>POS-tag embedding</td>
<td>16</td>
</tr>
</tbody>
</table>
Table 3: Experimental results on SemEval-2018 Task11. The two top-ranked systems are listed as baselines.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMA (Chen et al. 2018)</td>
<td>84.48</td>
<td>80.94</td>
</tr>
<tr>
<td>TriAN (Wang 2018)</td>
<td>83.84</td>
<td>81.94</td>
</tr>
<tr>
<td>CSA Model (single model)</td>
<td>83.63</td>
<td>82.20</td>
</tr>
<tr>
<td>CSA Model + ELMo (single model)</td>
<td>83.84</td>
<td>83.27</td>
</tr>
<tr>
<td>CSA Model (ensemble)</td>
<td>85.27</td>
<td>83.95</td>
</tr>
<tr>
<td>HMA (ensemble)</td>
<td>86.46</td>
<td>84.13</td>
</tr>
<tr>
<td>CSA Model (ensemble)</td>
<td>84.05</td>
<td>84.34</td>
</tr>
<tr>
<td>CSA Model + ELMo (ensemble)</td>
<td>85.05</td>
<td>85.23</td>
</tr>
</tbody>
</table>

Table 4: Ablations on several model components.

<table>
<thead>
<tr>
<th>Model</th>
<th>RACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSA Model</td>
<td>48.52</td>
</tr>
<tr>
<td>w/o attention weight</td>
<td>48.18</td>
</tr>
<tr>
<td>w/o enriched representation</td>
<td>47.52</td>
</tr>
<tr>
<td>w/o convolutional spatial attention</td>
<td>47.30</td>
</tr>
<tr>
<td>CSA Model + ELMo</td>
<td>50.89</td>
</tr>
<tr>
<td>w/o attention weight</td>
<td>49.49</td>
</tr>
<tr>
<td>w/o enriched representation</td>
<td>49.78</td>
</tr>
<tr>
<td>w/o convolutional spatial attention</td>
<td>48.47</td>
</tr>
</tbody>
</table>

We also removed our convolutional spatial attention mechanism in the model. The results show a significant drop in performance by 1.22%, indicating that the proposed convolutional spatial attention is effective in extracting most representative values among the various attention matrices. The same results drop is shown by incorporating ELMo. Further more, whether we use ELMo or not, our convolutional spatial attention has improved significantly. That also proved that our model can get father improvement after some big data trained toolkit like ELMo or some other transformer.

The details of the ablations are shown below.

- **w/o attention weight**: We change \( M = M' \odot W (\text{in Alg 1 line 2}) \) to \( M = M' \), where \( W \) is the attention weight.
- **w/o enriched representation**: We remove interaction of candidate-to-question (\( R_{CQ} \) in Figure 2) and candidate-to-passage (\( R_{CP} \) in Figure 2) and only use candidate's LSTM output \( R_C \).
- **w/o convolutional spatial attention**: We use two fully-connected layer to transform the matching matrix to a score. The first fully-connected layer squeeze the matching matrix to a vector along the question length dimension. The second fully-connected layer squeeze the vector to a scalar score. As every candidate has six matching matrices, so we can get six scores. Finally, we add the six scores as the final prediction.

### Analysis and Discussion

#### Quantitative Analysis

In CSA model, we adopt various enriched representation for better characterizing candidate information. We compare the performance between the original CSA model and discard enriched candidate representation, as shown in Figure 3.

As we can see that our model yields significant improvements on the question type ‘how’ and ‘why’, which are relatively difficult than the other type of questions requiring high-level reasoning within the context, demonstrating that our CSA model performs better on the relatively sophisticated questions. On the contrary, we find that our CSA model shows relatively inferior performance on the question type ‘who’, ‘where’, and ‘when’, which are often answered by a single word or entity. This phenomenon suggests that further efforts should be made on balancing the word-level attention and highly abstracted attention. These results also explain why our CSA model shows significant improvements on RACE-H subset while giving a relatively inferior performance on RACE-M subset. As the sample number of RACE-H subset is three times the size of RACE-M, the overall performance of our model still shows significant improvements over various state-of-the-art systems.

#### Case Study

We also randomly sampled one hundred question from RACE test set, and classified them into three categories according to its difficulty.

- **The first level is the question that can be answered by matching a few words in the passage. If the model finds**
the right place in the text, the answer is easy to find. With the attention mechanism, the neural model is especially good at solving this kind of question.

- The second level is the question that can be answered by using a few sentences without reasoning, which is relatively difficult than the first level.
- The third level is the question that needed to comprehensive reasoning via multiple clues in the passage. This kind of question is more complicated. To answer this question, the intention of the questioner must be captured.

The results are shown in Table 5. As we can see that the current model shows relatively good performance on the first and second level questions than the third one, indicating that more investigation should be made on solving those questions that need reasoning.

<table>
<thead>
<tr>
<th>State</th>
<th>Total #</th>
<th>Right #</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>100</td>
<td>56</td>
<td>56.0%</td>
</tr>
<tr>
<td>First level</td>
<td>30</td>
<td>18</td>
<td>60%</td>
</tr>
<tr>
<td>Second level</td>
<td>30</td>
<td>20</td>
<td>66.7%</td>
</tr>
<tr>
<td>Third level</td>
<td>40</td>
<td>18</td>
<td>45.0%</td>
</tr>
</tbody>
</table>

Table 5: Case analysis on RACE dataset.

Error Analysis

In this part, a interesting example is shown in Figure 4. Our model choose the right answer “C”. How can our model choose the right answer? Because the candidate “C” has a bigest semantic overlap with passage. To improve that, we change candidate “C” to “ellipses in talking”, but the model can still choose the right answer. Even we change candidate “C” to “ellipses in talking”. From this experiment, we can see that our model can really handle part of high level semantic matching. Further more, we change the question to “The conversation is mainly about _.”, the model still choose the candidate “C” with a high probability. This reflects the problem that the deep learning model is not sensitive to some special important information in passage and question. That phenomenon more or less approve the weakness of neural model. So we need to do more in-depth research to design some special component like CNN in dealing with picture to deal with text in reading comprehension.

RACE Dataset

Passage

As is known to all, in daily conversation people often use simple words and simple sentences, especially elliptical sentences. Here is an interesting conversation between Mr Green and his good friend Mr Smith, a fisherman. Do you know what they are talking about?

Question

The text is mainly about _.

Candidates

A how to catch fish.  B how to spend a Sunday  C ellipses in conversations  D joy in fishing

Figure 4: Example of RACE dataset. The correct answer is depicted in bold face.

Conclusion

We propose the Convolutional Spatial Attention (CSA) model to tackle the machine reading comprehension with multiple-choice questions. The proposed model could fully extract the mutual information among the passage, question, and candidates, to form the enriched representations using the modified attention mechanism that has trainable weights. To summarize attention matrices from various sources, we propose to use convolutional operation to dynamically summarize the attention values within the different size of regions, which is beneficial to capture diverse features. Experimental results show that the proposed CSA model could give substantial improvements over various state-of-the-art systems on both RACE and SemEval 2018 Task11 datasets. Also, the ablation studies verify the effectiveness of several proposed components in our model, and analysis also shows that the CSA model is superior in solving relatively sophisticated questions (such as ‘why’ or ‘how’ questions).

Acknowledgments

We would like to thank all three anonymous reviewers for their thorough reviewing and providing constructive comments to improve our paper. This work was supported by the National Key R&D Program of China via grant No. 2016YFC0800806.
References


