Learning and Knowledge Transfer with Memory Networks for Machine Comprehension

Mohit Yadav. Lovekesh Vig. Gautam Shroff

TCS Research New-Delhi

Presented by Kyo Kim

April 24, 2018
Overview

1. Motivation
2. Background
3. Proposed Method
4. Dataset and Experiment Results
5. Summary
Motivation
Problem

- Obtaining high performance in "machine comprehension" requires abundant human annotated dataset.
  - Measured by question answering performance.
- In a real-world dataset with small amount of data, wider range of vocabulary can be observed and the grammar structure is often complex.
High-level Overview of Proposed Method

1. Curriculum based training procedure.
2. Knowledge transfer to increase the performance in dataset with less abundant labeled data.
3. Pre-trained memory network on small dataset.
Background
End-to-end Memory Networks

1. Vectorize the problem tuple.
2. Retrieve the corresponding memory attention vector.
3. Use the retrieved memory to answer the question.
Vectorize the problem tuple
- **Problem tuple:** \((q, C, S, s)\)
  - \(q\): question
  - \(C\): context text
  - \(S\): set of answer choices
  - \(s\): correct answer \((s \in S)\)
- **Question and context embedding matrix** \(A \in \mathbb{R}^{p \cdot d}\)
  - **Query vector:** \(\vec{q} = A\Phi(q)\)
    - \(\Phi\): Bag of words
  - **Memory vector:** \(\vec{m}_i = A\Phi(c_i)\) for \(i = 1, \cdots, n\) where \(n = |C|\) and \(c_i \in C\)
Retrieve the corresponding memory attention vector

- Attention distribution: \( a_i = \text{softmax}(\tilde{m}_i^\top \tilde{q}) \).
- Second memory vector: \( \tilde{r}_i = B\Phi(c_i) \) where \( B \) is another embedding matrix similar to \( A \).
- Aggregated vector: \( \tilde{r}_o = \sum_{i=1}^{n} a_i \tilde{r}_i \)
- Prediction vector: \( \hat{a}_i = \text{softmax}((\tilde{r}_o + \tilde{q})^\top U\Phi(s_i)) \)
  - \( U \) is the embedding matrix for the answers
Answer the question

- Pick $s_i$ that corresponds to the highest $\hat{a}_i$.

Cross-entropy Loss

$$L(P, D) = \frac{1}{N_D} \sum_{n=1}^{N_D} \left[ a_n \cdot \log(\hat{a}_n(P, D)) + (1 - a_n) \cdot \log(1 - \hat{a}_n(P, D)) \right]$$
First proposed by Bengio et al. (2009)

Introduce samples with increasing "difficulty".

Better local minima even under non-convex loss.
Pre-training

- Have a pre-trained model to initially guide the training process in a similar domain.

Joint-training

- Exploit the similarity between two different domains by training the model in two different domains simultaneously.
Proposed Method
Curriculum Inspired Training (CIT)

Difficulty Measurement

\[ SF(q, S, C, s) = \frac{\sum_{word \in \{q \cup S \cup C\}} \log(Freq(word))}{\#\{q \cup S \cup C\}} \]

- Partition the dataset into a fixed number *chapter_size* with increasing difficulty.
- Each chapter consists of \( \bigcup_{i=1}^{current\_chapter} partition[i] \).
- The model is trained with fixed number of epochs per chapter.
Loss Function

\[ L(P, D, en) = \frac{1}{N_D} \sum_{n=1}^{N_D} \left[ (a_n \cdot \log(\hat{a}_n(P, D))) \right. \]

\[ + (1 - a_n) \cdot \log(1 - \hat{a}_n(P, D)) \cdot 1_{en \geq c(n) \cdot epc} \]

- \( en \) : Current epoch
- \( c(n) \) : Chapter number that the example \( n \) is assigned to
- \( epc \) : Epochs per chapter
Joint-Training

General Joint Loss Function

\[
\hat{L}(P, TD, SD) = 2\gamma \cdot L(P, TD) + 2(1 - \gamma) \cdot L(P, SD) \cdot F(N_{TD}, N_{SD})
\]

- \(TD\): Target dataset
- \(SD\): Source dataset
- \(N_D\): Number of examples in the dataset \(D\)
- \(\gamma\): Tunable weight parameter
Loss Functions

Joint-training

\[ \gamma = \frac{1}{2} \text{ and } F(N_{TD}, N_{SD}) = 1 \]
\[ \hat{L}(P, TD, SD) = L(P, TD) + L(P, SD) \]

Weighted joint-training

\[ \gamma \in (0, 1) \text{ and } F(N_{TD}, N_{SD}) = \frac{N_{TD}}{N_{SD}} \]
\[ \hat{L}(P, TD, SD) = 2\gamma \cdot L(P, TD) + 2(1 - \gamma) \cdot L(P, SD) \cdot \frac{N_{TD}}{N_{SD}} \]
Loss Functions Cont.

Curriculum joint-training

\[ \gamma = \frac{1}{2} \text{ and } F(N_{TD}, N_{SD}) = 1 \]

\[ \hat{L}(P, TD, SD) = L(P, TD, \text{en}) + L(P, SD, \text{en}) \]

Weighted Curriculum joint-training

\[ \gamma \in (0, 1) \text{ and } F(N_{TD}, N_{SD}) = \frac{N_{TD}}{N_{SD}}. \]

\[ \hat{L}(P, TD, SD) = 2\gamma \cdot L(P, TD, \text{en}) + 2(1 - \gamma)L(P, SD, \text{en}) \cdot \frac{N_{TD}}{N_{SD}} \]
\[ \gamma = 0 \text{ and } c \in \mathbb{R}^+ \]

\[ \hat{L}(P, TD, SD) = c \cdot L(P, SD) \]
Dataset and Experiment Results
### Dataset

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td># Train</td>
<td>280</td>
<td>1400</td>
<td>11,000</td>
<td>22,000</td>
<td>55,000</td>
<td>55,000</td>
</tr>
<tr>
<td># Validation</td>
<td>120</td>
<td>200</td>
<td>3,924</td>
<td>3,924</td>
<td>3,924</td>
<td>2,500</td>
</tr>
<tr>
<td># Test</td>
<td>200</td>
<td>400</td>
<td>3,198</td>
<td>3,198</td>
<td>3,198</td>
<td>2,000</td>
</tr>
<tr>
<td># Vocabulary</td>
<td>2856</td>
<td>4279</td>
<td>26,550</td>
<td>31,932</td>
<td>40,833</td>
<td>42,311</td>
</tr>
<tr>
<td># Words ≠ Dailymail-55K</td>
<td>—</td>
<td>—</td>
<td>1,981</td>
<td>2,734</td>
<td>6,468</td>
<td>—</td>
</tr>
</tbody>
</table>

**Figure:** Dataset used for experiments.
### Experiment Results

**Figure:** The table has two major rows. The upper row are models that only used the target dataset. The lower rows are models that used both the target and source dataset.

<table>
<thead>
<tr>
<th>Model + Training Methods</th>
<th>CNN-11 K</th>
<th>CNN-22 K</th>
<th>CNN-55 K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train</td>
<td>Valid</td>
<td>Test</td>
</tr>
<tr>
<td>SW+W2V §</td>
<td>43.90</td>
<td>43.01</td>
<td>42.60</td>
</tr>
<tr>
<td>MemNN §</td>
<td>98.98</td>
<td>45.96</td>
<td>46.08</td>
</tr>
<tr>
<td>MemNN+CIT §</td>
<td>96.44</td>
<td>47.17</td>
<td><strong>49.04</strong></td>
</tr>
<tr>
<td>SW+Dailymail ‡</td>
<td>30.19</td>
<td>31.21</td>
<td>30.60</td>
</tr>
<tr>
<td>MemNN+W2V ‡</td>
<td>86.57</td>
<td>43.78</td>
<td>45.99</td>
</tr>
<tr>
<td>MemNN+SrcOnly ‡</td>
<td>25.12</td>
<td>26.78</td>
<td>27.08</td>
</tr>
<tr>
<td>MemNN+Pre-train ‡</td>
<td>92.82</td>
<td>52.87</td>
<td>52.06</td>
</tr>
<tr>
<td>MemNN+Jo-train ‡</td>
<td>65.78</td>
<td>53.85</td>
<td>55.06</td>
</tr>
<tr>
<td>MemNN+CIT+Jo-train ‡</td>
<td>77.74</td>
<td>55.93</td>
<td>55.74</td>
</tr>
<tr>
<td>MemNN+W+Jo-train ‡</td>
<td>71.72</td>
<td>54.30</td>
<td>55.70</td>
</tr>
<tr>
<td>MemNN+W+CIT+Jo-train ‡</td>
<td>80.14</td>
<td>56.91</td>
<td><strong>57.02</strong></td>
</tr>
</tbody>
</table>

Mohit Yadav. Lovekesh Vig. Gautam Shroff (TCS Research New-Delhi)
**Figure**: Categorical performance measurement in CNN-11 K. The table has two major rows. The upper row are models that only used the target dataset. The lower rows are models that used both the target and source dataset.
## Experiment Results

### Figure: Knowledge transfer performance result.

<table>
<thead>
<tr>
<th>Training Methods</th>
<th>MCTest-160</th>
<th></th>
<th>MCTest-500</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>One</td>
<td>Multi.</td>
<td>All</td>
<td>One</td>
</tr>
<tr>
<td>SW</td>
<td>66.07</td>
<td>53.12</td>
<td>59.16</td>
<td>54.77</td>
</tr>
<tr>
<td>SW+D</td>
<td>75.89</td>
<td>60.15</td>
<td>67.50</td>
<td>63.23</td>
</tr>
<tr>
<td>SW+D+W2V</td>
<td>79.46</td>
<td>59.37</td>
<td>68.75</td>
<td>65.07</td>
</tr>
<tr>
<td>SW+D+CNN-11K</td>
<td>79.78</td>
<td>59.37</td>
<td><strong>67.67</strong></td>
<td>64.33</td>
</tr>
<tr>
<td>SW+D+CNN-22K</td>
<td>76.78</td>
<td>60.93</td>
<td><strong>68.33</strong></td>
<td>64.70</td>
</tr>
<tr>
<td>SW+D+CNN-55K</td>
<td>78.57</td>
<td>59.37</td>
<td><strong>68.33</strong></td>
<td>65.07</td>
</tr>
<tr>
<td>SW+D+CNN-11K+W2V</td>
<td>77.67</td>
<td>59.41</td>
<td>68.69</td>
<td>65.07</td>
</tr>
<tr>
<td>SW+D+CNN-22K+W2V</td>
<td>78.57</td>
<td>60.16</td>
<td>69.51</td>
<td>66.91</td>
</tr>
<tr>
<td>SW+D+CNN-55K+W2V</td>
<td>79.78</td>
<td>60.93</td>
<td><strong>70.51</strong></td>
<td>66.91</td>
</tr>
</tbody>
</table>

Mohit Yadav. Lovekesh Vig. Gautam Shroff (TCS Research New-Delhi)  
Presented by Kyo Kim  
April 24, 2018  
24 / 27
**Figure:** Loss convergence comparison between model trained with CIT and without CIT.
MemNN is often used in QA.
Ordering the samples lead to better local minima.
Joint-training is useful in obtaining better performance on small target dataset.
Using pre-trained model improves performance.
The End