End-to-End Memory Networks with Knowledge Carryover for Multi-Turn Spoken Language Understanding

SEP. 12th, 2016 @ San Francisco
Outline

Introduction

- Spoken Dialogue System
- Spoken/Natural Language Understanding (SLU/NLU)

Contextual Spoken Language Understanding

- Model Architecture
- End-to-End Training

Experiments

Conclusion & Future Work
Introduction

Spoken Dialogue System

Spoken/Natural Language Understanding (SLU/NLU)

Contextual Spoken Language Understanding

Model Architecture

End-to-End Training

Experiments

Conclusion & Future Work
Spoken Dialogue System (SDS)

- **Spoken dialogue systems** are intelligent agents that are able to help users finish tasks more efficiently via spoken interactions.

- **Spoken dialogue systems** are being incorporated into various devices (smart-phones, smart TVs, in-car navigating system, etc).

Good intelligent assistants help users to organize and access information conveniently
Dialogue System Pipeline

Speech Signal → ASR → Hypothesis

Language Understanding (LU)
- User Intent Detection
- Slot Filling

Semantic Frame (Intents, Slots)
- request_movie
- genre=action
- date=this weekend

Dialogue Management (DM)
- Dialogue State Tracking
- Policy Decision

Output Generation

Text Input
- Are there any action movies to see this weekend?

Screen Display
- location?

Text response
- Where are you located?
LU Importance

Learning Curve of System Performance

- Upper Bound
- DQN - 0.00
- DQN - 0.05
- Rule - 0.00
- Rule - 0.05

RL Agent w/o LU errors
Rule Agent w/o LU errors
LU Importance

The system performance is sensitive to LU errors, for both rule-based and reinforcement learning agents.

Learning Curve of System Performance

- Upper Bound
- DQN - 0.00
- DQN - 0.05
- Rule - 0.00
- Rule - 0.05

RL Agent w/o LU errors
RL Agent w/ 5% LU errors
>5% performance drop

Rule Agent w/o LU errors
Rule Agent w/ 5% LU errors
SLU usually focuses on understanding single-turn utterances. The understanding result is usually influenced by 1) local observations 2) global knowledge.
Spoken Language Understanding

Domain Identification → Intent Prediction → Slot Filling

\( D \) communication

\( I \) send_email

\( U \) just sent email to bob about fishing this weekend

\( S \) just sent email to bob about fishing this weekend

\( \rightarrow send_email(contact_name="bob", subject="fishing this weekend") \)

\( U_1 \) send email to bob

\( S_1 \) B-contact_name

\( \rightarrow send_email(contact_name="bob") \)

\( U_2 \) are we going to fish this weekend

\( S_2 \) B-message I-message I-message I-message I-message

\( \rightarrow send_email(message="are we going to fish this weekend") \)
Outline

Introduction

Spoken Dialogue System

Spoken/Natural Language Understanding (SLU/NLU)

Contextual Spoken Language Understanding

Model Architecture

End-to-End Training

Experiments

Conclusion & Future Work
**MODEL ARCHITECTURE**

1. **Sentence Encoding**
   
   \[ m_i = \text{RNN}_{\text{mem}}(x_i) \]
   
   \[ u = \text{RNN}_{\text{in}}(c) \]

2. **Knowledge Attention**
   
   \[ p_i = \text{softmax}(u^T m_i) \]

3. **Knowledge Encoding**
   
   \[ h = \sum_i p_i m_i \]
   
   \[ o = W_{kg}(h + u) \]

---

**Idea:** additionally incorporating contextual knowledge during slot tagging

---

MODEL ARCHITECTURE

1. Sentence Encoding
   \[ m_i = \text{RNN}_{\text{mem}}(x_i) \]
   \[ u = \text{RNN}_{\text{in}}(c) \]

2. Knowledge Attention
   \[ p_i = \text{softmax}(u^T m_i) \]

3. Knowledge Encoding
   \[ h = \sum_i p_i m_i \quad o = W_{kg}(h + u) \]

**Idea:** additionally incorporating contextual knowledge during slot tagging

---

END-TO-END TRAINING

- Tagging Objective

\[ \mathbf{y} = \text{RNN}(\mathbf{o}, \mathbf{c}) \]

slot tag sequence \hspace{1cm} contextual utterances & current utterance

\[ p(\mathbf{y} | \mathbf{c}) = p(\mathbf{y} | w_1, ..., w_T) = \prod_i p(y_i | w_1, ..., w_i). \]

- RNN Tagger

\[ h_t = \phi(Mo + Ww_t + Uh_{t-1}) \]

\[ \hat{y}_t = \text{softmax}(Vh_t) \]

Automatically figure out the attention distribution without explicit supervision
Outline

Introduction
- Spoken Dialogue System
- Spoken/Natural Language Understanding (SLU/NLU)
- Contextual Spoken Language Understanding
- Model Architecture
- End-to-End Training

Experiments

Conclusion & Future Work
EXPERIMENTS

- Dataset: Cortana communication session data
  - GRU for all RNN
  - adam optimizer
  - embedding dim=150
  - hidden unit=100
  - dropout=0.5

<table>
<thead>
<tr>
<th>Model</th>
<th>Training Set</th>
<th>Knowledge Encoding</th>
<th>Sentence Encoder</th>
<th>First Turn</th>
<th>Other</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN Tagger</td>
<td>single-turn</td>
<td>x</td>
<td>x</td>
<td>60.6</td>
<td>16.2</td>
<td>25.5</td>
</tr>
</tbody>
</table>

The model trained on single-turn data performs worse for non-first turns due to mismatched training data.
EXPERIMENTS

- Dataset: Cortana communication session data
  - GRU for all RNN
  - adam optimizer
  - embedding dim=150
  - hidden unit=100
  - dropout=0.5

<table>
<thead>
<tr>
<th>Model</th>
<th>Training Set</th>
<th>Knowledge Encoding</th>
<th>Sentence Encoder</th>
<th>First Turn</th>
<th>Other</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN Tagger</td>
<td>single-turn</td>
<td>x</td>
<td>x</td>
<td>60.6</td>
<td>16.2</td>
<td>25.5</td>
</tr>
<tr>
<td></td>
<td>multi-turn</td>
<td>x</td>
<td>x</td>
<td>55.9</td>
<td>45.7</td>
<td>47.4</td>
</tr>
</tbody>
</table>

Treating multi-turn data as single-turn for training performs reasonable
EXPERIMENTS

- Dataset: Cortana communication session data
  - GRU for all RNN
  - adam optimizer
  - embedding dim=150
  - hidden unit=100
  - dropout=0.5

<table>
<thead>
<tr>
<th>Model</th>
<th>Training Set</th>
<th>Knowledge Encoding</th>
<th>Sentence Encoder</th>
<th>First Turn</th>
<th>Other</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN Tagger</td>
<td>single-turn</td>
<td>x</td>
<td>x</td>
<td>60.6</td>
<td>16.2</td>
<td>25.5</td>
</tr>
<tr>
<td></td>
<td>multi-turn</td>
<td>x</td>
<td>x</td>
<td>55.9</td>
<td>45.7</td>
<td>47.4</td>
</tr>
<tr>
<td>Encoder-Tagger</td>
<td>multi-turn</td>
<td>current utt (c)</td>
<td>RNN</td>
<td>57.6</td>
<td>56.0</td>
<td>56.3</td>
</tr>
<tr>
<td></td>
<td>multi-turn</td>
<td>history + current (x, c)</td>
<td>RNN</td>
<td>69.9</td>
<td>60.8</td>
<td>62.5</td>
</tr>
</tbody>
</table>

Encoding current and history utterances improves the performance but increases the training time
EXPERIMENTS

- Dataset: Cortana communication session data
  - GRU for all RNN
  - adam optimizer
  - embedding dim=150
  - hidden unit=100
  - dropout=0.5

<table>
<thead>
<tr>
<th>Model</th>
<th>Training Set</th>
<th>Knowledge Encoding</th>
<th>Sentence Encoder</th>
<th>First Turn</th>
<th>Other</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN Tagger</td>
<td>single-turn</td>
<td>x</td>
<td>x</td>
<td>60.6</td>
<td>16.2</td>
<td>25.5</td>
</tr>
<tr>
<td></td>
<td>multi-turn</td>
<td>x</td>
<td>x</td>
<td>55.9</td>
<td>45.7</td>
<td>47.4</td>
</tr>
<tr>
<td>Encoder-Tagger</td>
<td>multi-turn</td>
<td>current utt (c)</td>
<td>RNN</td>
<td>57.6</td>
<td>56.0</td>
<td>56.3</td>
</tr>
<tr>
<td></td>
<td>multi-turn</td>
<td>history + current (x, c)</td>
<td>RNN</td>
<td>69.9</td>
<td>60.8</td>
<td>62.5</td>
</tr>
<tr>
<td>Proposed</td>
<td>multi-turn</td>
<td>history + current (x, c)</td>
<td>RNN</td>
<td>73.2</td>
<td>65.7</td>
<td>67.1</td>
</tr>
</tbody>
</table>

Applying memory networks significantly outperforms all approaches with much less training time
EXPERIMENTS

- Dataset: Cortana communication session data
  - GRU for all RNN
  - adam optimizer
  - embedding dim=150
  - hidden unit=100
  - dropout=0.5

<table>
<thead>
<tr>
<th>Model</th>
<th>Training Set</th>
<th>Knowledge Encoding</th>
<th>Sentence Encoder</th>
<th>First Turn</th>
<th>Other</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN Tagger</td>
<td>single-turn</td>
<td>x</td>
<td>x</td>
<td>60.6</td>
<td>16.2</td>
<td>25.5</td>
</tr>
<tr>
<td></td>
<td>multi-turn</td>
<td>x</td>
<td>x</td>
<td>55.9</td>
<td>45.7</td>
<td>47.4</td>
</tr>
<tr>
<td>Encoder-Tagger</td>
<td>multi-turn</td>
<td>current utt (c)</td>
<td>RNN</td>
<td>57.6</td>
<td>56.0</td>
<td>56.3</td>
</tr>
<tr>
<td></td>
<td>multi-turn</td>
<td>history + current (x, c)</td>
<td>RNN</td>
<td>69.9</td>
<td>60.8</td>
<td>62.5</td>
</tr>
<tr>
<td>Proposed</td>
<td>multi-turn</td>
<td>history + current (x, c)</td>
<td>RNN</td>
<td>73.2</td>
<td>65.7</td>
<td>67.1</td>
</tr>
<tr>
<td></td>
<td>multi-turn</td>
<td>history + current (x, c)</td>
<td>CNN</td>
<td>73.8</td>
<td>66.5</td>
<td>68.0</td>
</tr>
</tbody>
</table>

NEW! NOT IN THE PAPER!

CNN produces comparable results for sentence encoding with shorter training time
Outline

Introduction
- Spoken Dialogue System
- Spoken/Natural Language Understanding (SLU/NLU)

Contextual Spoken Language Understanding
- Model Architecture
- End-to-End Training

Experiments

Conclusion & Future Work
Conclusion

- The proposed end-to-end memory networks store contextual knowledge, which can be exploited dynamically based on an attention model for manipulating knowledge carryover for multi-turn understanding.
- The end-to-end model performs the tagging task instead of classification.
- The experiments show the feasibility and robustness of modeling knowledge carryover through memory networks.
Future Work

• Leveraging not only local observation but also global knowledge for better language understanding
  – Syntax or semantics can serve as global knowledge to guide the understanding model
THANKS FOR YOUR ATTENTION!

The code will be available at
https://github.com/yvchen/ContextualSLU