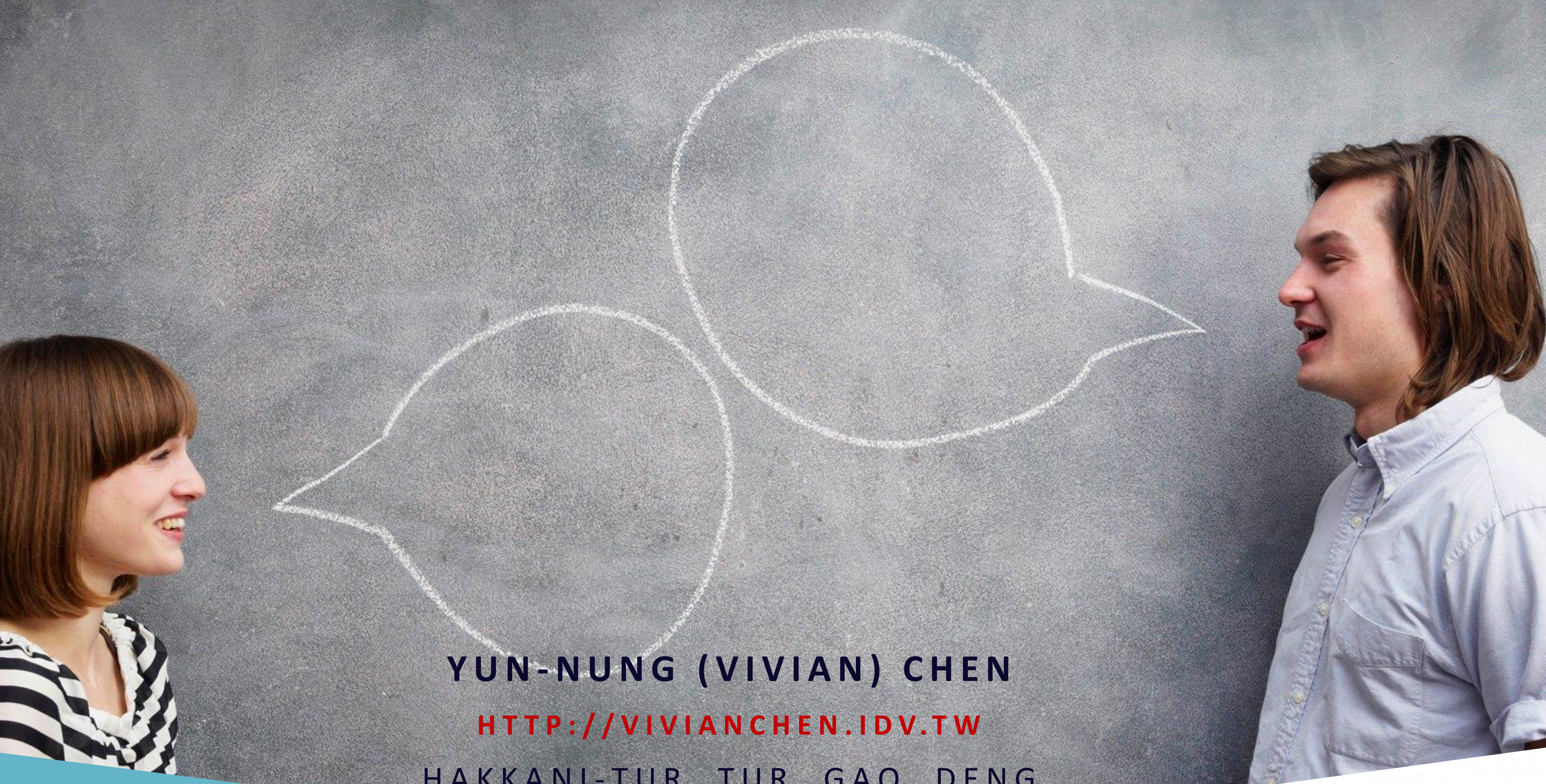


YUN-NUNG (VIVIAN) CHEN

HTTP://VIVIANCHEN.IDV.TW

HAKKANI-TUR, TUR, GAO, DENG

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

Microsoft Research

INTERSPEECH 2016

National Taiwan University

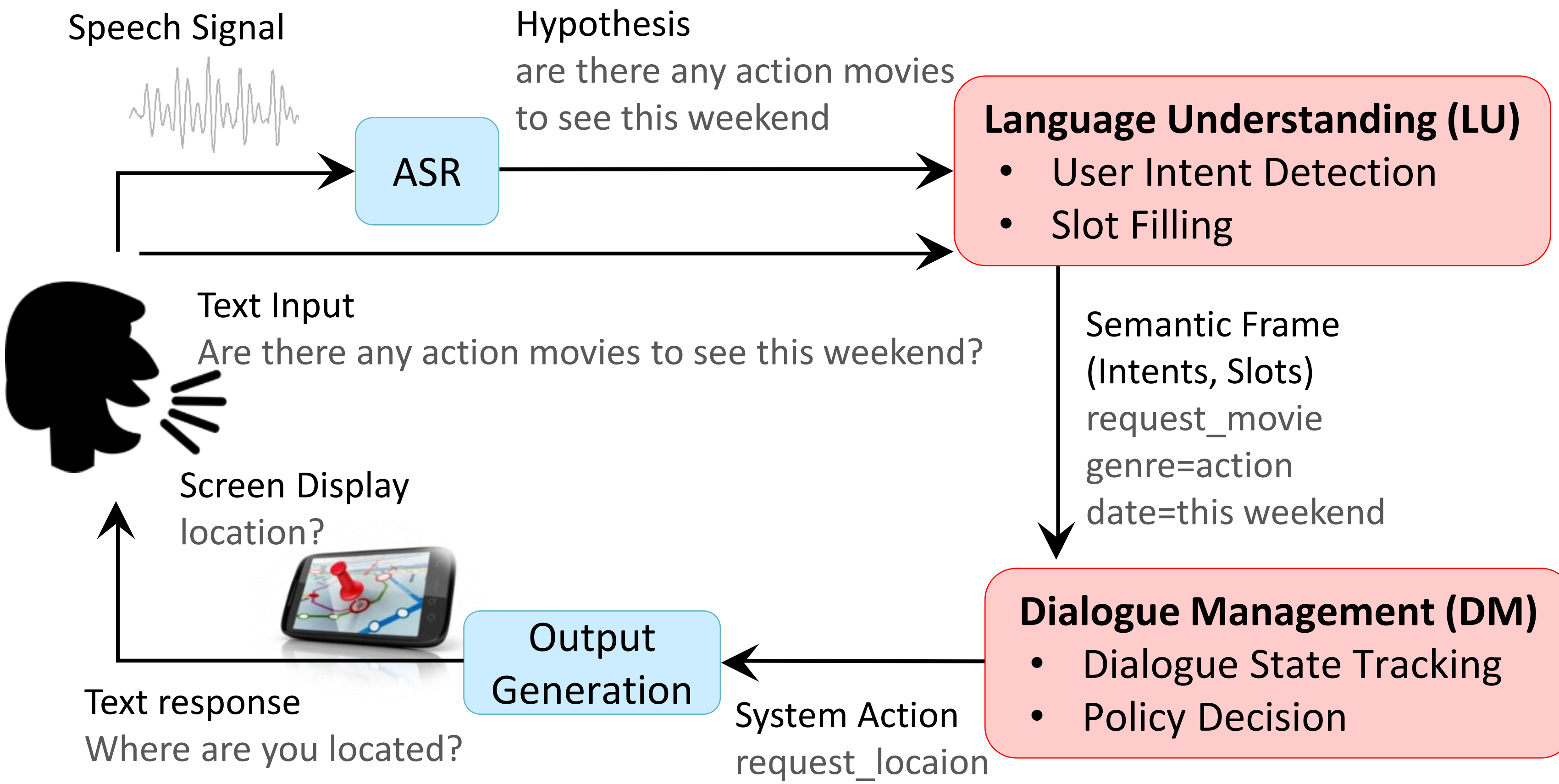
Outline

- Introduction
  - Spoken Dialogue System
  - Spoken/Natural Language Understanding (SLU/NLU)
- Contextual Spoken Language Understanding
  - Model Architecture
  - End-to-End Training
- Experiments
- Conclusion & Future Work

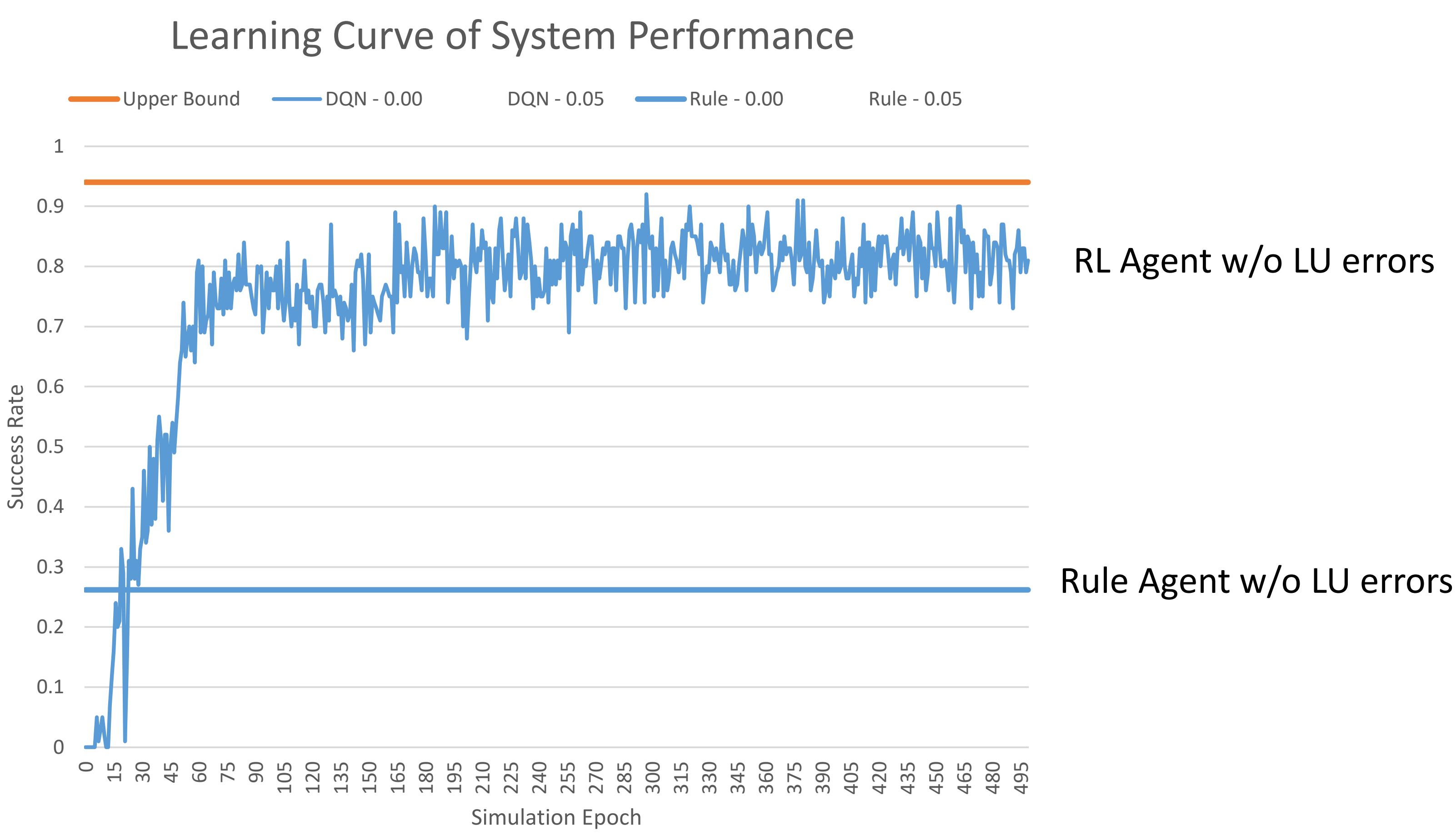
Outline

- Introduction
  - Spoken Dialogue System
  - Spoken/Natural Language Understanding (SLU/NLU)
- Contextual Spoken Language Understanding
  - Model Architecture
  - End-to-End Training
- Experiments
- Conclusion & Future Work

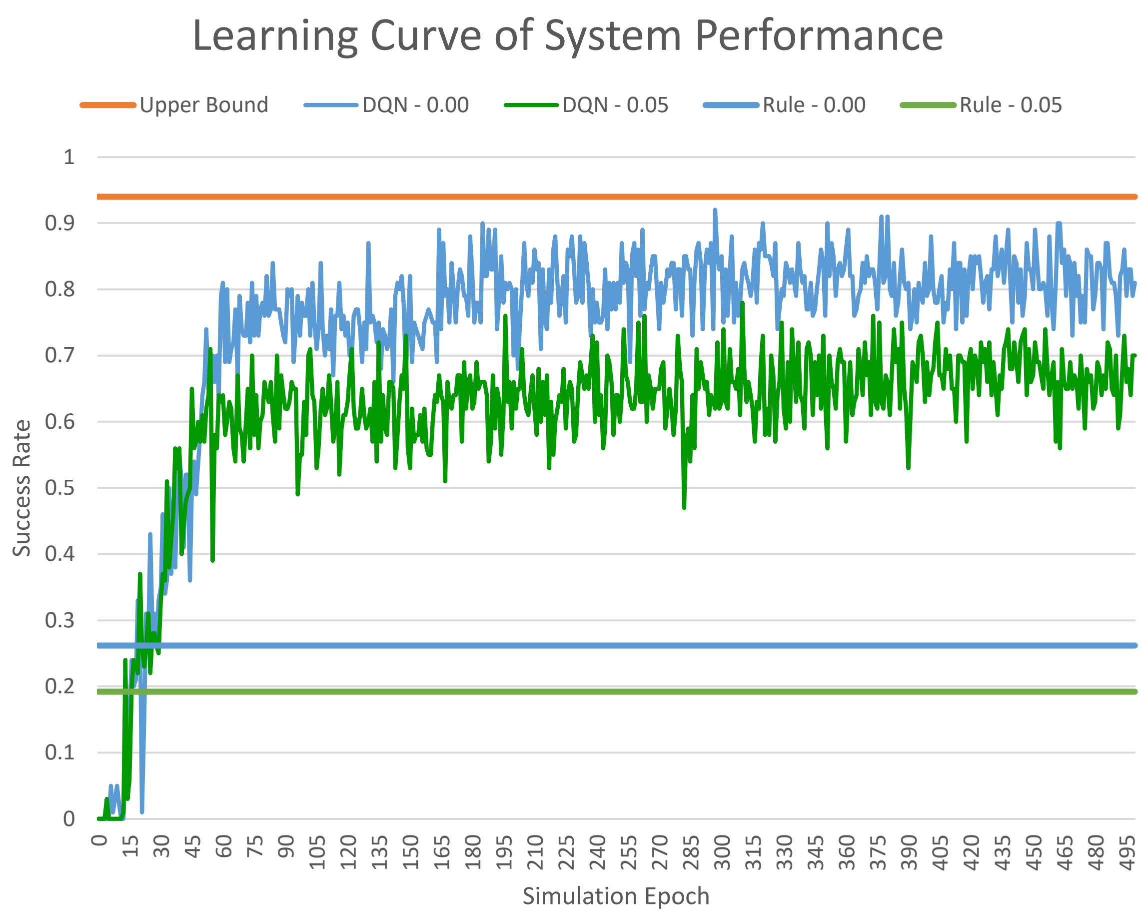
Dialogue System Pipeline



LU Importance



LU Importance



RL Agent w/o LU errors

RL Agent w/ 5% LU errors

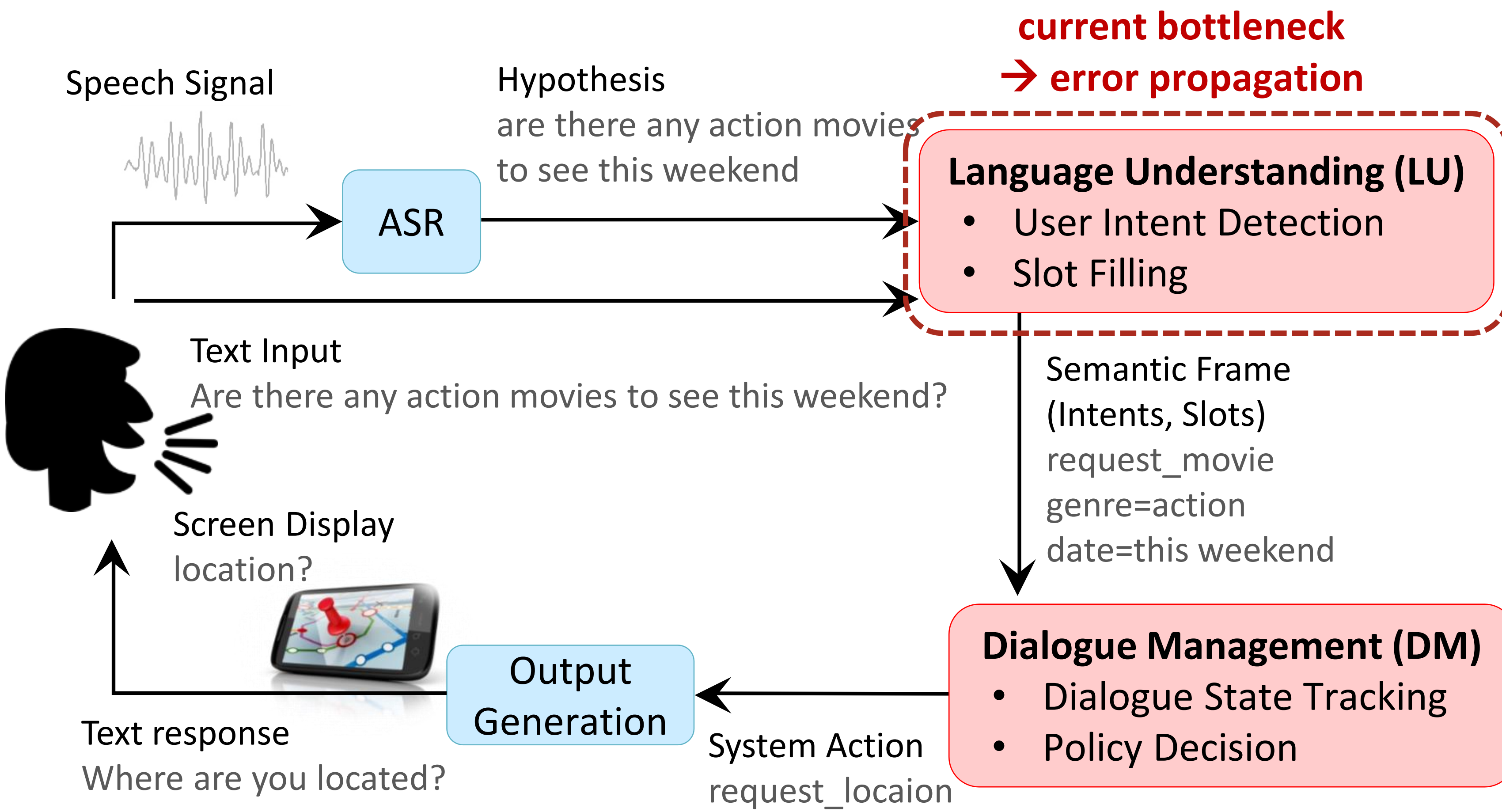
>5% performance drop

Rule Agent w/o LU errors

Rule Agent w/ 5% LU errors

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

Dialogue System Pipeline



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

Single Turn

U	just	sent	email	to	bob	about	fishing	this	weekend
S	O	O	O	O	B-contact_name	O	B-subject	I-subject	I-subject

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

Multi-Turn

U_1	send	email	to	bob
S_1				B-contact_name

→ 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	I-message	I-message

→ 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

$$m_i = \mathrm{RNN}_{\mathrm{mem}}(x_i)$$

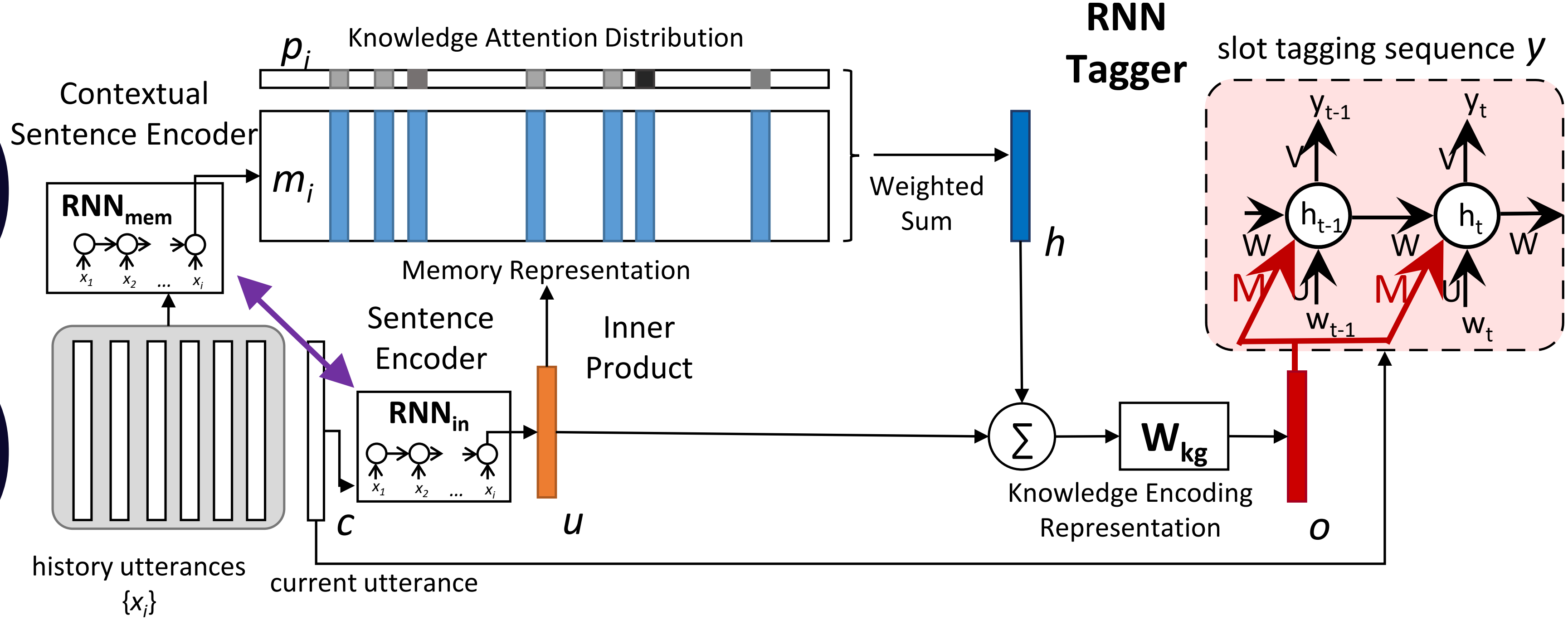
$$u = \mathrm{RNN}_{\mathrm{in}}(c)$$

2. Knowledge Attention

$$p_i = \mathrm{softmax}(u^T m_i)$$

3. Knowledge Encoding

$$h = \sum_i p_i m_i \quad o = W_{\mathrm{kg}}(h + u)$$



Idea: additionally incorporating contextual knowledge during slot tagging

MODEL ARCHITECTURE

1. Sentence Encoding

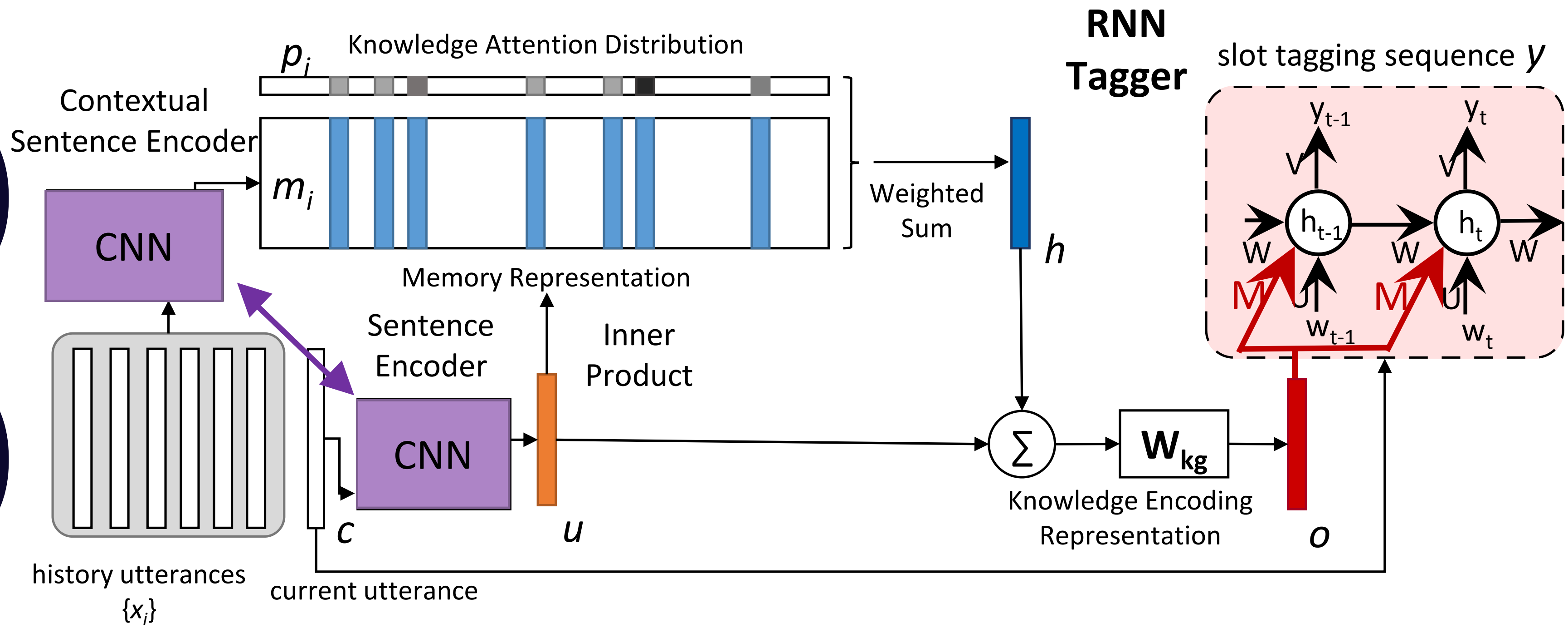
$$m_i = \mathrm{RNN}_{\mathrm{mem}}(x_i)$$
$$u = \mathrm{RNN}_{\mathrm{in}}(c)$$

2. Knowledge Attention

$$p_i = \mathrm{softmax}(u^T m_i)$$

3. Knowledge Encoding

$$h = \sum_i p_i m_i \quad o = W_{\mathrm{kg}}(h + u)$$



Idea: additionally incorporating contextual knowledge during slot tagging

Chen, et al., "End-to-End Memory Networks with Knowledge Carryover for Multi-Turn Spoken Language Understanding," in Interspeech, 2016.

END-TO-END TRAINING

- Tagging Objective

$$\boldsymbol{y} = \mathrm{RNN}(o, \boldsymbol{c})$$

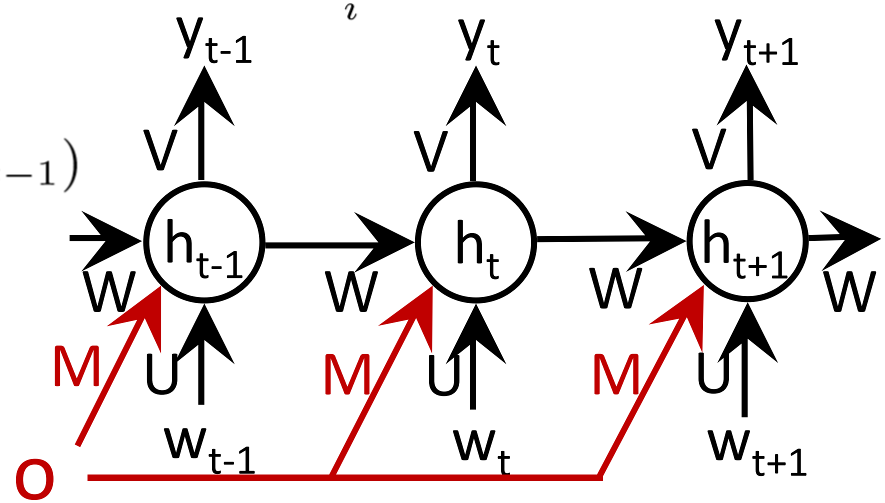
slot tag sequence contextual utterances & current utterance

$$p(\boldsymbol{y} \mid \boldsymbol{c}) = p(\boldsymbol{y} \mid w_1, ..., w_T) = \prod_i p(y_i \mid w_1, ..., w_i).$$

- RNN Tagger

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

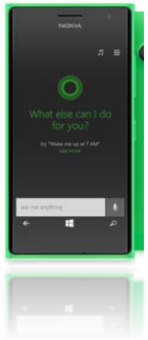
$$\hat{y}_t = \mathrm{softmax}(Vh_t)$$



Automatically figure out the attention distribution without explicit supervision

Outline

EXPERIMENTS

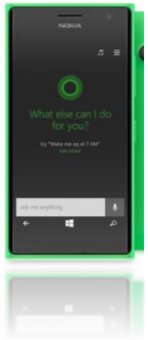


- Dataset: Cortana communication session data
  - GRU for all RNN
  - adam optimizer
  - embedding dim=150
  - hidden unit=100
  - dropout=0.5

Model	Training Set	Knowledge Encoding	Sentence Encoder	First Turn	Other	Overall
RNN Tagger	single-turn	x	x	**60.6**	16.2	25.5

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

EXPERIMENTS

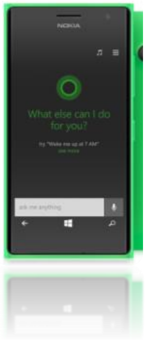


- Dataset: Cortana communication session data
  - GRU for all RNN
  - adam optimizer
  - embedding dim=150
  - hidden unit=100
  - dropout=0.5

Model	Training Set	Knowledge Encoding	Sentence Encoder	First Turn	Other	Overall
RNN Tagger	single-turn	x	x	**60.6**	16.2	25.5
	multi-turn	x	x	55.9	**45.7**	**47.4**

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

EXPERIMENTS

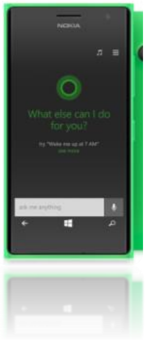


- Dataset: Cortana communication session data
  - GRU for all RNN
  - adam optimizer
  - embedding dim=150
  - hidden unit=100
  - dropout=0.5

Model	Training Set	Knowledge Encoding	Sentence Encoder	First Turn	Other	Overall
RNN Tagger	single-turn	x	x	60.6	16.2	25.5
	multi-turn	x	x	55.9	45.7	47.4
Encoder-Tagger	multi-turn	current utt (c)	RNN	57.6	56.0	56.3
	multi-turn	history + current (x, c)	RNN	**69.9**	**60.8**	**62.5**

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

EXPERIMENTS

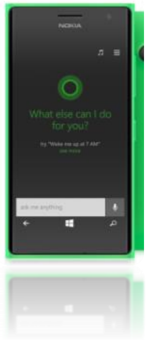


- Dataset: Cortana communication session data
  - GRU for all RNN
  - adam optimizer
  - embedding dim=150
  - hidden unit=100
  - dropout=0.5

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

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

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

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
  - "Knowledge as a Teacher: Knowledge-Guided Structural Attention Networks," arXiv preprint arXiv: 1609.03286

Q & A

THANKS FOR YOUR ATTENTION!

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