TOSHIBA

Leading Innovation

Dialog State Tracking for Unseen Values using an Extended Attention Mechanism

T. Yoshida, K. Iwata, H. Fujimura and M. Akamine
Research & Development center,
Toshiba Corporation

Introduction

Purpose:
Easily extendable dialog state tracking

Problem: Task may vary after model trained

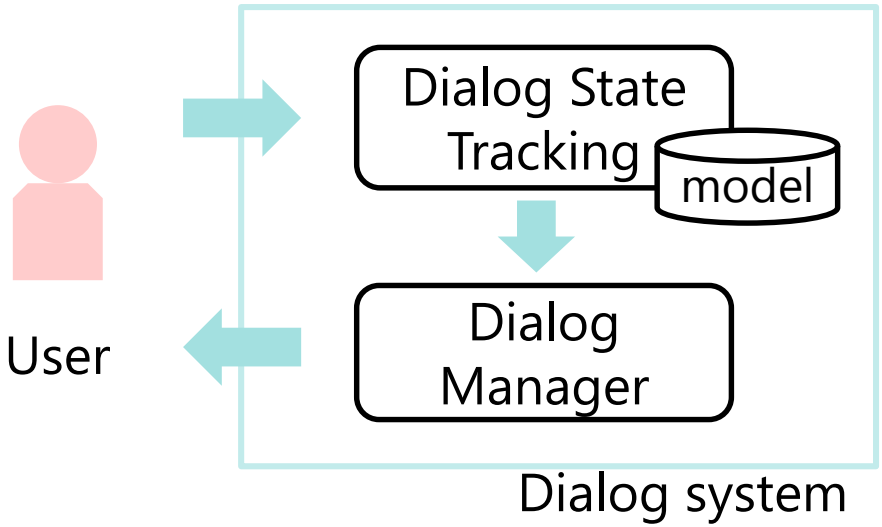
- Track new values in testing

"Extendable":

- Unseen values in testing

"Easily":

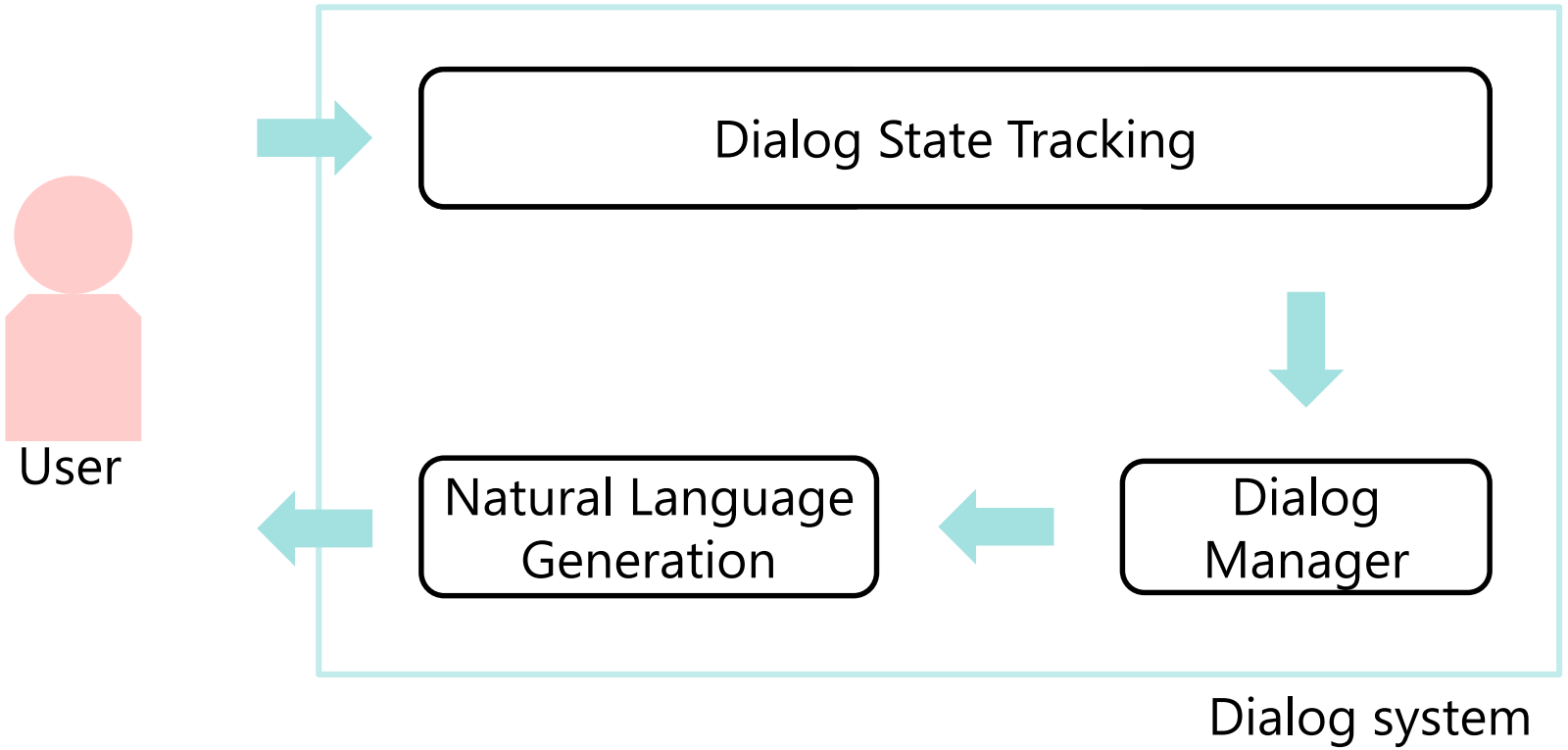
- No hand-crafted rules
- No retraining



Outline

- **Background**
- **Proposed method**
- **Experiments**
- **Conclusion**

Task-oriented spoken dialog system

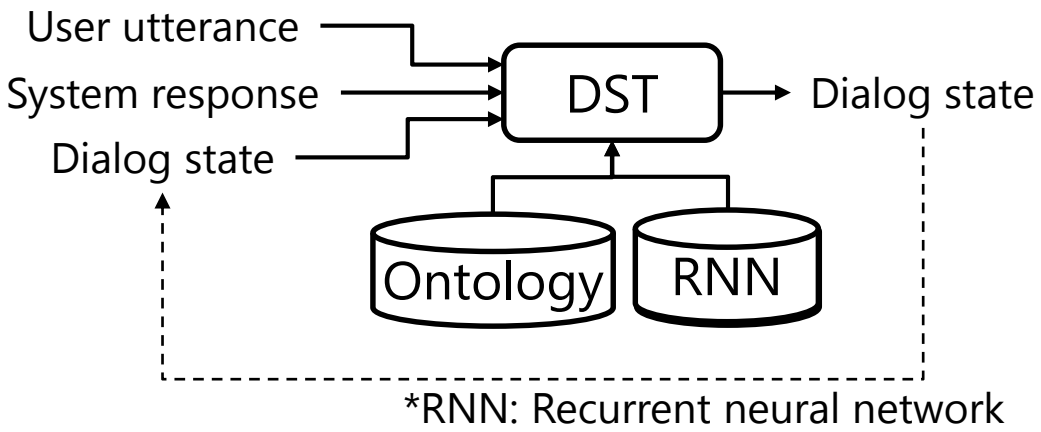


Dialog state tracking (DST)

- **Example:**

System response	User utterance	inquiry
Hello. May I help you?	Tell me an Italian restaurant	italian
Do you prefer Tokyo?	Yes	italian & tokyo
How about xxx restaurant	Show me another one.	italian & tokyo

- **Input:**
  - User utterance
  - System response
  - Previous dialog state
- **Output:**
  - Current dialog state



*RNN: Recurrent neural network

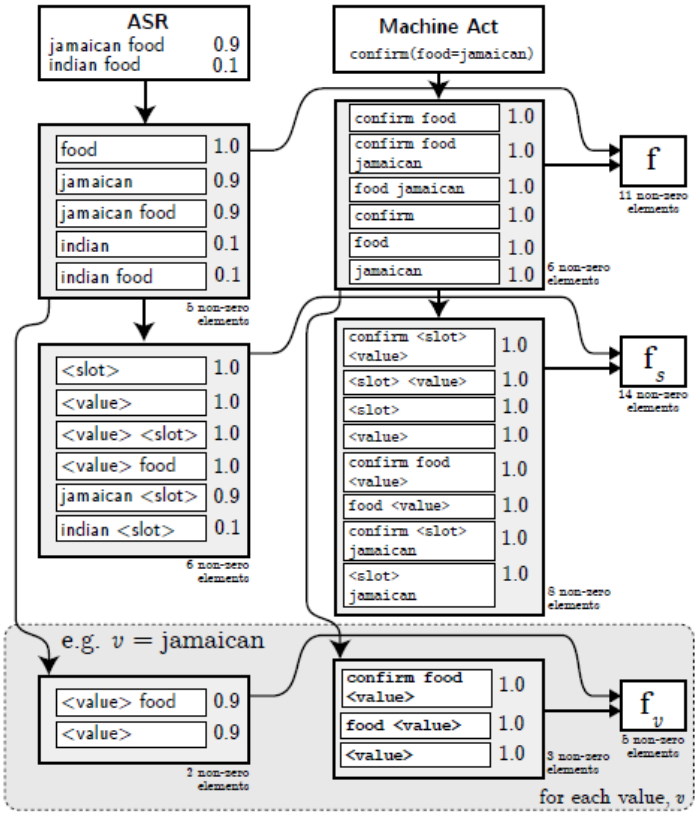
RNN with delexicalization [Henderson 14]

Generalized features and a value independent RNN

- **Delexicalization**
  - Replace keywords with labels
    "Jamaican food"➡"<value> <slot>"

- **Decoder**
  - An RNN calculates a reliability score for each value

☺ detect unseen values without retraining
☹ require rules and synonym lists



[Henderson 14]

Issue and approach

Purpose: easily extendable DST

Issue: RNN-based DST has difficulty in handling unseen values

- Retraining or crafting rules is necessary

Approach:

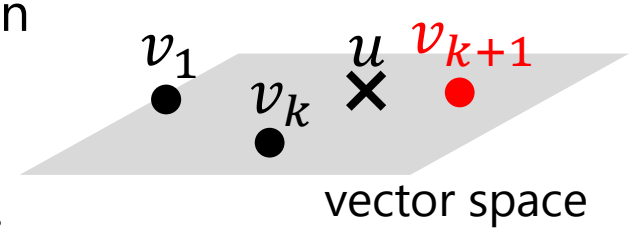
A) Relevant keyword extraction

B) Fully data-driven tracking

Approach

A) Relevant keyword extraction

– New $k+1$ th value can be tracked based on the semantic similarity



Attention-based keyword extraction with cosine similarity-based decoder [Jang 16]

A) Relevant keyword extraction

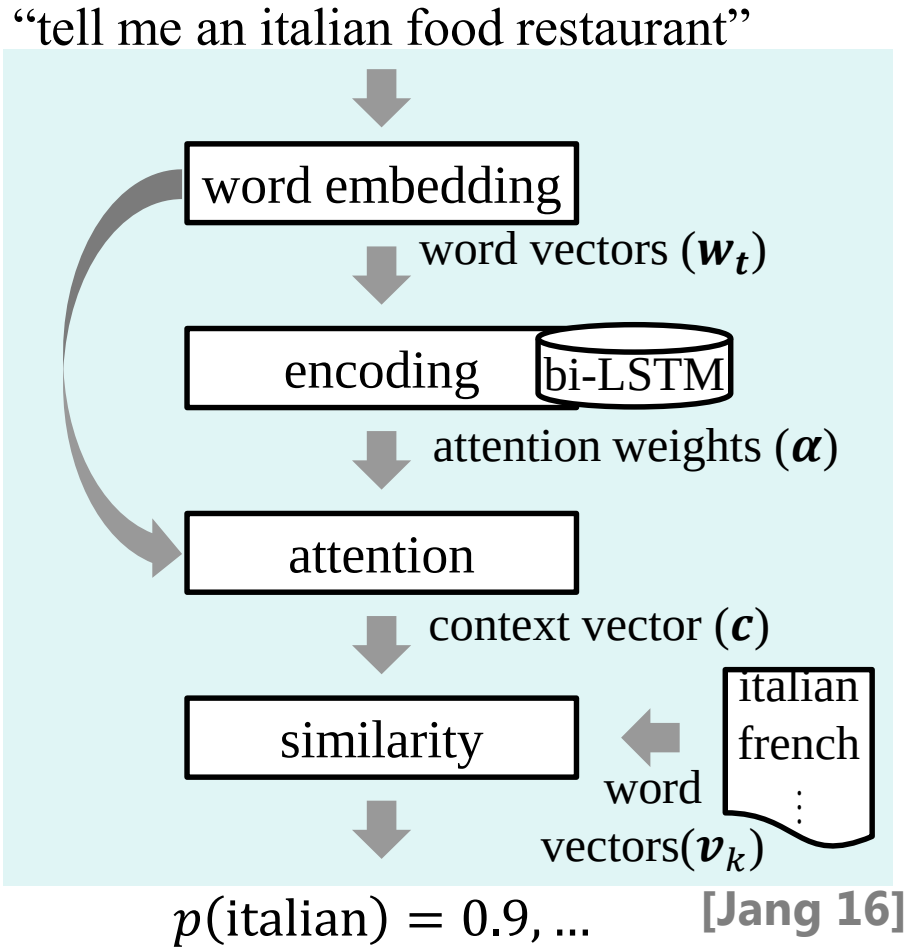
- **Extract relevant keyword**
  - Attention weights represent relevance of each word

$$\boldsymbol{c} = \sum \alpha_t \boldsymbol{w_t}$$

- **Detect unseen values**
  - Output a probability based on semantic similarity

$$y_k = \boldsymbol{c} \cdot \boldsymbol{v}_k / |\boldsymbol{c}||\boldsymbol{v}_k|$$

☺ Detect unseen values without retraining and crafting rules
☹ Can't exploit system response and previous dialog state

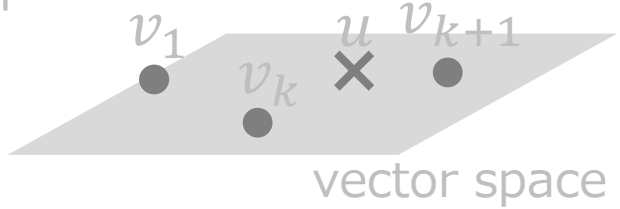


[Jang 16]

Approach

A) Relevant keyword extraction

– New $k+1$ th value can be tracked based on the semantic similarity



➡ **Attention-based keyword extraction with cosine similarity-based decoder[Jang 16]**

B) Fully data-driven tracking

– Integrate previous dialog state, system response, and user utterance

➡ **Sentinel mixture model [Merity 16]**

System response	User utterance	inquiry
Hello. May I help you?	Tell me an italian restaurant	italian
What area do you prefer? Tokyo?	Yes	italian & tokyo
How about xxx restaurant	what is phone number?	italian & tokyo

B) Fully data-driven tracking

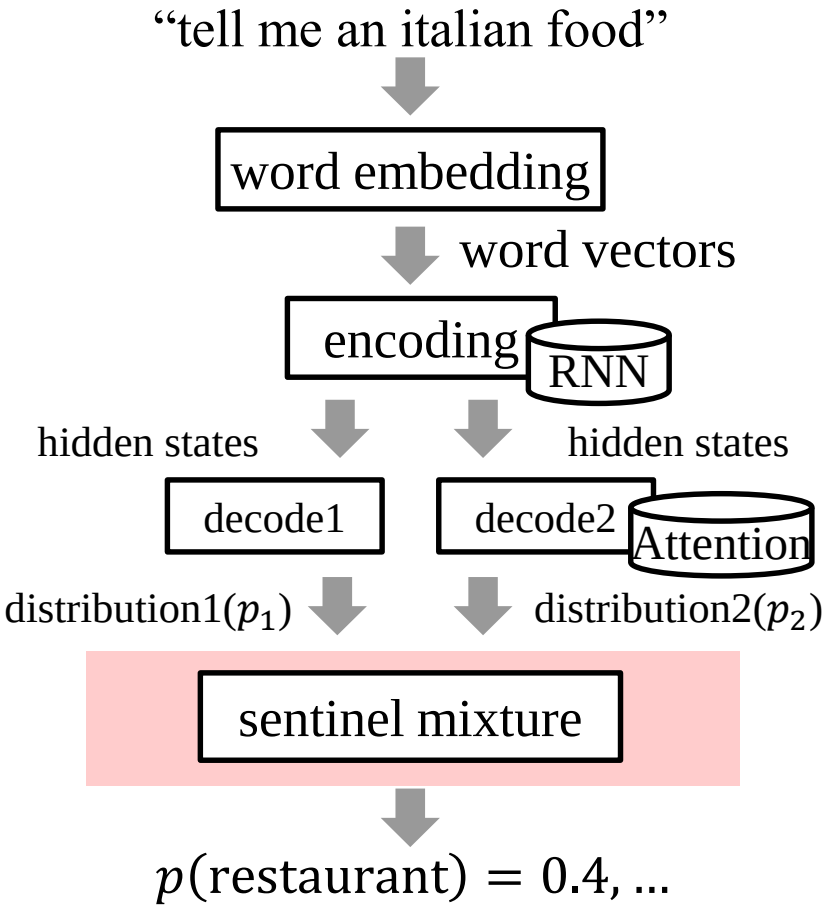
Integrate different cues with their importance

- **Sentinel mixture**
  - Estimate two kinds of distribution

    p_1, p_2

  - Integrate them using a weight parameter g

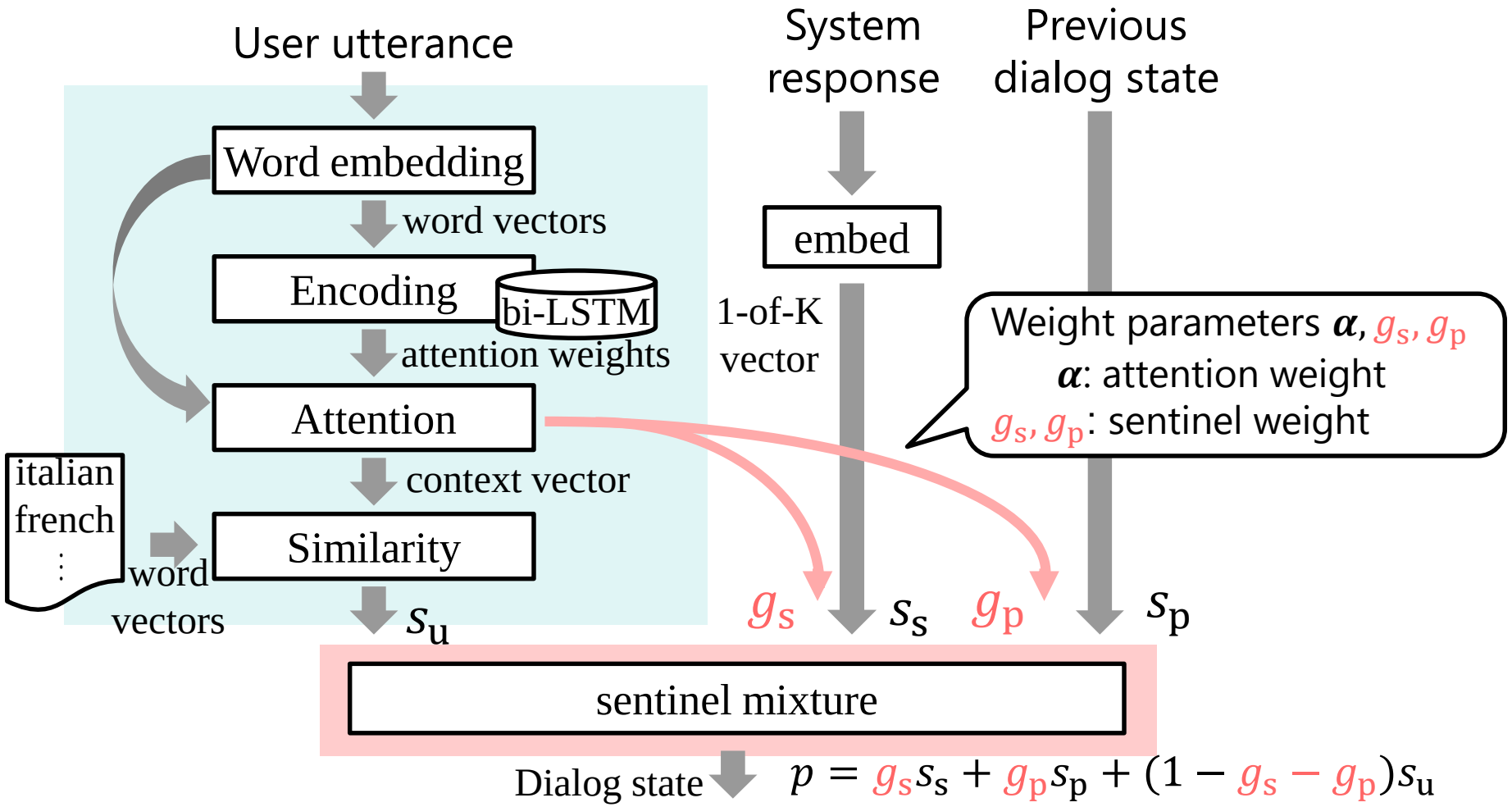
    $$p = gp_1 + (1 - g)p_2$$



☺ Integrates two kinds of distributions with its importance

[Merity 16]

Proposed method



☺ Exploit cues from system response and previous output
☺ Detect unseen values without retraining and rules

Implementation

Models implemented using Chainer [Tokui 15]

Input:

- User utterance: 1-best ASR result
- System response: action tag

Output:

- Probability of each value in a slot

Parameter settings:

- Word vector model: GloVe (300 dims) [Pennington 14]
- Bidirectional LSTM of 32 nodes
- Fully connected layers of 32 nodes
- Adam, Learning rate = 0.001
- Max iteration = 200, mini-batch=32
- Dropout, word dropout

Experiment

Metric: accuracy

$$\text{accuracy} = \frac{\#\text{ of true positive}}{\#\text{ of samples}}$$

Data: DSTC 2 and DSTC 3 datasets

	Training	DSTC 2 Test set		DSTC 3 Test set		
	Values	Seen	Unseen	Seen	Unseen	Examples of unseen
area	7	7	0	3	14	girton, arbury, ...
food	92	92	0	20	10	american, cafe food, ...
price range	5	5	0	6	1	free

Only seen values

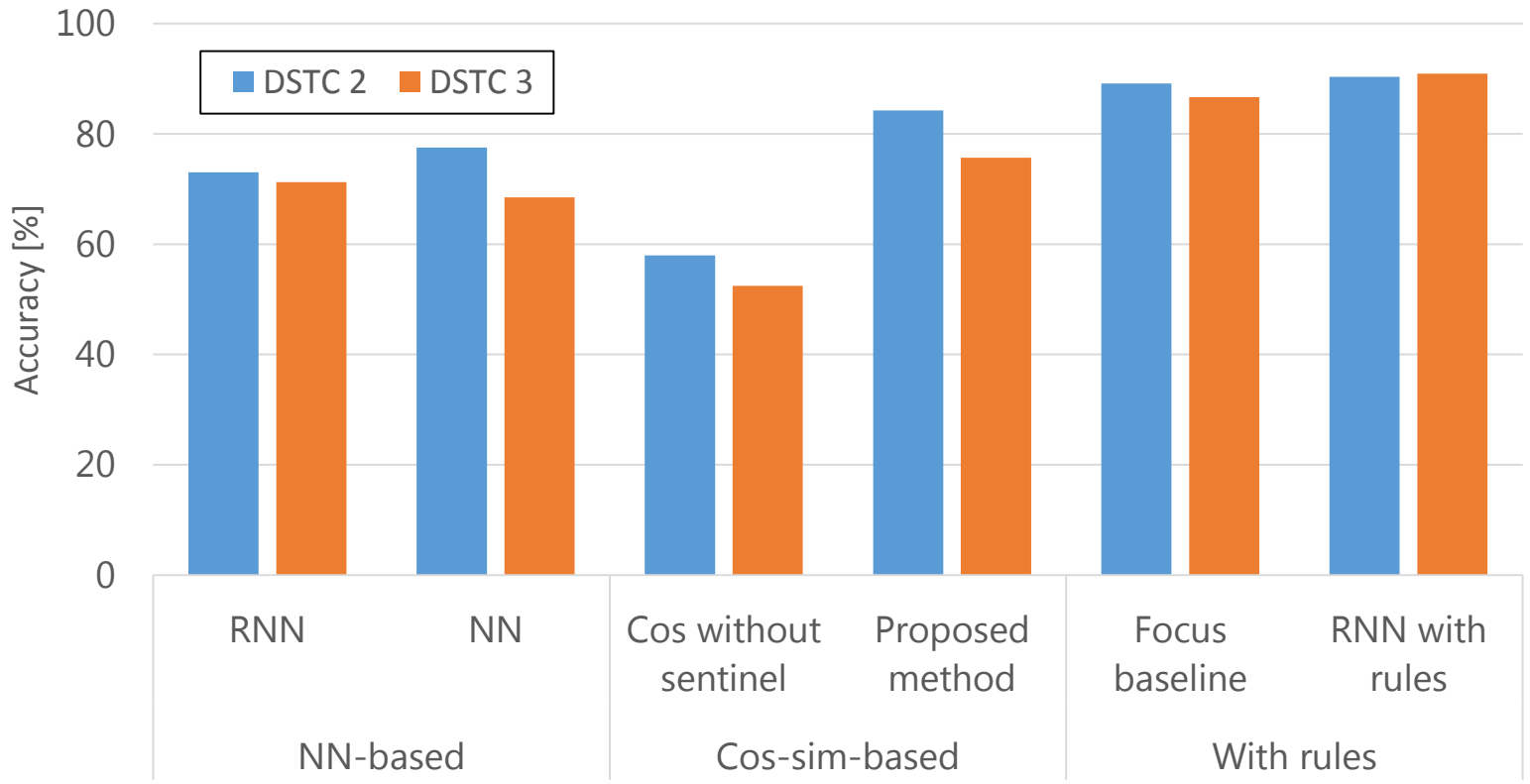
seen & unseen

Baseline methods in comparison

- ***cos without sentinel*: cos-dec and rules**
- ***RNN*: LSTM**
- ***NN*: fully-connected NN**

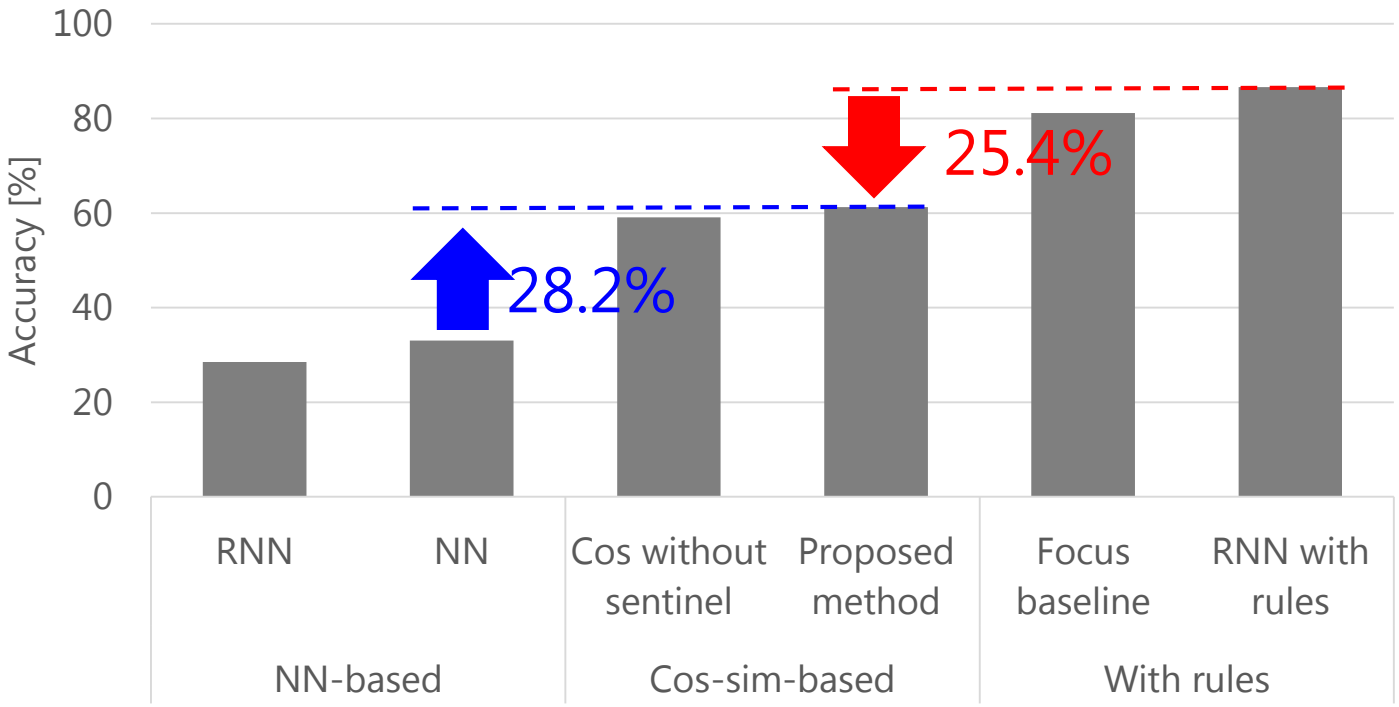
- **DSTC 2 & 3 participants**
  - ***focus baseline*** [Henderson 14]
  - ***RNN with rules*** [Henderson 14]

Accuracy for DSTC 2 & DSTC 3



- **Proposed method > NN, RNN > Cos without sentinel**
- **With rules > NN-based, Cos-sim-based**

Accuracy for unseen values

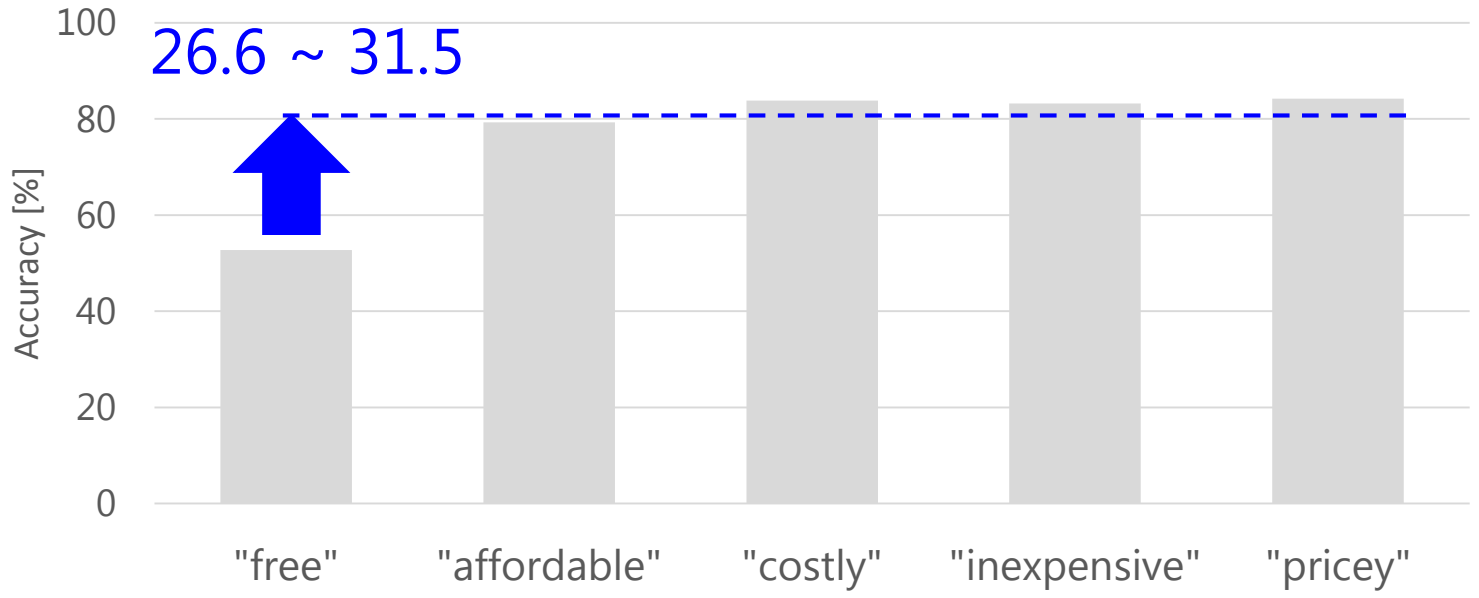


- **Cos sim-based > NN-based**
  - RNN/NN has difficulty in generalizing to unseen values
- **With rules > Cos sim-based**

Accuracy of the price range slot

- **Tracking “free” is difficult**

  ← The reason may be quality of the word vector model
  (5-top most similar words: “Free”, “download”, “downloads”, “downloadable”, “online”.)



Discussion

☺ Neither additional retraining nor crafting rules are required

- After adding new values to the ontology, the DST method can detect them

  ➡ **Easily extendable**

☹ A "good" word vector of each value is required

- Difficult to detect "free"
- Difficult to detect proper noun, telephone number, etc

☹ Can't distinguish values including the same words

- E.g. "chinese" and "chinese takeaway"

Conclusion

- **Proposed easily extendable DST**
  - ⬅ Detect unseen values **without retraining** and **hand-crafted rules**
  - Relevant keyword extraction using attention mechanism
  - Fully data-driven tracking using sentinel mixture architecture
  - Experimental results show the effectiveness of the proposed method

Future work

- Improve accuracy further
- Track values of proper nouns, numbers, etc.
- Track unseen slots

Thank you