Task-oriented dialogue & Ontology based state tracking

By: Divye & Victor

Task-based Spoken dialogue systems (SDS)

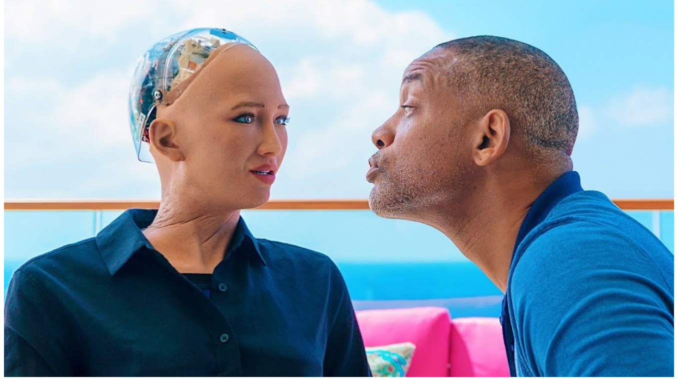
- Spoken dialogue systems (SDS) allow users to interact with computer applications through conversation
- Task-based systems help users achieve goals
  - For Example: Finding a restaurant

Human : Restaurant in any area that serves Vietnamese food

```
Machine : thanh binh serves vietnamese food
type , is there anything else i can help
you with ?
```

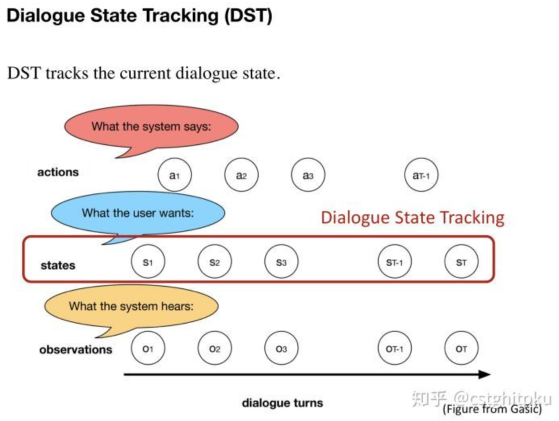
Human : What is the phone number and postcode?

```
Machine : the number is 01223 362456 and
the area code is c.b 3, 0 a.f . is there
anything else i can do for you ?
```



Dialogue State Tracking (DST)

The dialogue state tracking (DST) component of an Spoken dialogue systems (SDS) serves to interpret user input and update the belief state, which is the system’s internal representation of the state of the conversation.



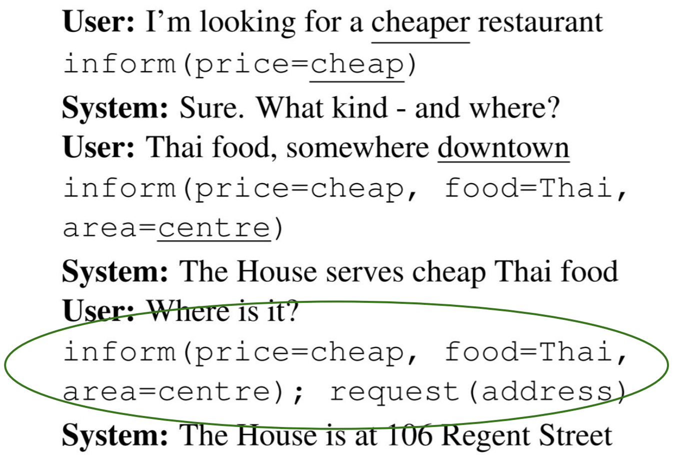
Dialogue State Tracking (DST)

DST tracks the current dialogue state.

(Figure from Gašić)

What is a Ontology-based State?

- State: Representation of the user’s belief
    - Collection of slot/value pairs
        - Slot: Food
        - Value: Thai
- 2 types of slots
    - Informational slots: Search constraints
        - Example: Inform(Price=Cheap, Food=Thai)
    - Requestable slots: Questions by the users
        - Example: Request(address)
- Need to define these slot/values
    - Ontology



User: I’m looking for a cheaper restaurant

`inform(price=cheap)`

System: Sure. What kind - and where?

User: Thai food, somewhere downtown

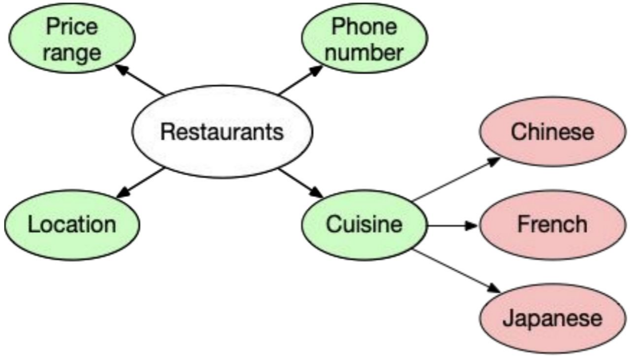
`inform(price=cheap, food=Thai, area=centre)`

System: The House serves cheap Thai food

User: Where is it?

`inform(price=cheap, food=Thai, area=centre); request(address)`

System: The House is at 106 Regent Street



Problem

- DST: critical component
- Scalability/Dealing with complex dialogue domains.
    - SLU models: large amounts of annotated training data.
    - Hand-crafted lexicons
        - Wen et al., 2016

Discussion question: Why do you think?

Older Approaches

- Separate SLU and DST
  - SLU: Detect slot-value pairs from ASR
    - Binary classification/Sequence model
  - DST: Update the belief state
  - Discussion question: Thoughts? Drawbacks?

- Joint SLU/DST
  - Generate belief states from ASR predictions
  - Relies on delexilization
    - Replace text with generic labels in the dialogue
  - Discussion question: Drawbacks?

Neural Belief Tracker: Data-Driven Dialogue State Tracking

Nikola Mrkšić[1], Diarmuid Ó Séaghdha[2]

Tsung-Hsien Wen[1], Blaise Thomson[2], Steve Young[1]

[1] University of Cambridge

[2] Apple Inc.

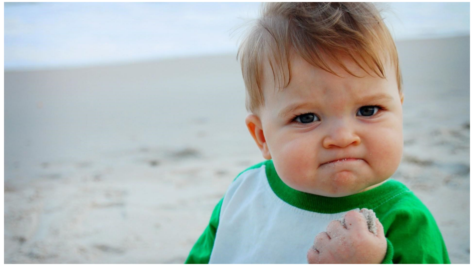
`{nm480, thw28, sjy}@cam.ac.uk`

`{doseaghdha, blaisethom}@apple.com`

Neural Belief Tracking (NBT)

- NBT: Leverages representation learning.
- NBT uses pre-trained word vectors
    - Represent user utterances and dialogue context.
- Match performance of SOTA models which rely on hand-crafted semantic lexicons.

DST shifts towards pre-trained word embeddings



Next Topics

Dataset

Models

Experiments

Dataset

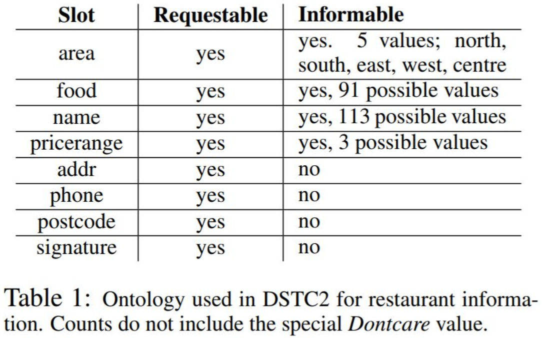
Used 2 datasets

- Dialog State Tracking Challenge 2 (DSTC2):
  - The domain is restaurant search
  - 2207 training dialogues and the test set consists of 1117 dialogues
  - Henderson et al., 2014

- Wizard of Oz 2.0 (WoZ2.0):
  - The domain is restaurant search
  - 1200 dialogues. 600 training, 200 validation and 400 test set
  - Wen et al., 2017

Dialog State Tracking Challenge 2 (DSTC2)

- Paid Amazon Mechanical Turkers were assigned tasks and asked to call the dialog systems.
  - DM-HC: Maintains single top dialog state + handcrafted policy - Train/Dev
  - DM-POMDPHC: Dynamic Bayesian network + handcrafted policy - Train/Dev
  - DM-POMDP: DM-POMDPHC + POMDP reinforcement learning - Test
  - Results from these trackers were manually corrected
- Callers were asked to find restaurants that matched particular constraints on the slots area, price range and food.



Slot	**Requestable**	**Informable**
area	yes	yes. 5 values; north, south, east, west, centre
food	yes	yes, 91 possible values
name	yes	yes, 113 possible values
pricerange	yes	yes, 3 possible values
addr	yes	no
phone	yes	no
postcode	yes	no
signature	yes	no

Table 1: Ontology used in DSTC2 for restaurant information. Counts do not include the special *Dontcare* value.

DSTC2 Data Analysis

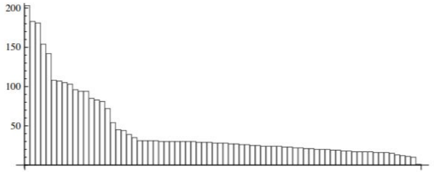
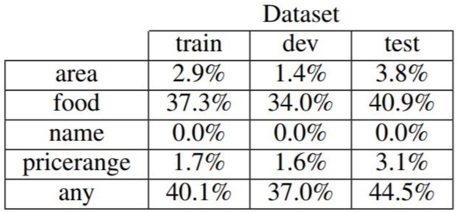


Figure 1: Histogram of values for the food constraint (excluding dontcare) in all data. The most frequent values are Indian, Chinese, Italian and European.



	Dataset		
	train	dev	test
area	2.9%	1.4%	3.8%
food	37.3%	34.0%	40.9%
name	0.0%	0.0%	0.0%
pricerange	1.7%	1.6%	3.1%
any	40.1%	37.0%	44.5%

Table 2: Percentage of dialogs which included a change in the goal constraint for each informable (and any slot). Barely any users asked for restaurants by name.

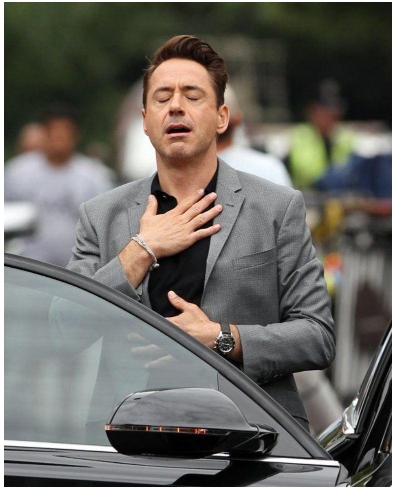
High WER

- 26.4% in train
- 31.9% on dev
- 28.7% test

Furthermore, the DSTC2 data had to be cleaned: Various spelling errors

Wizard-of-Oz 2.0 (WoZ2.0)

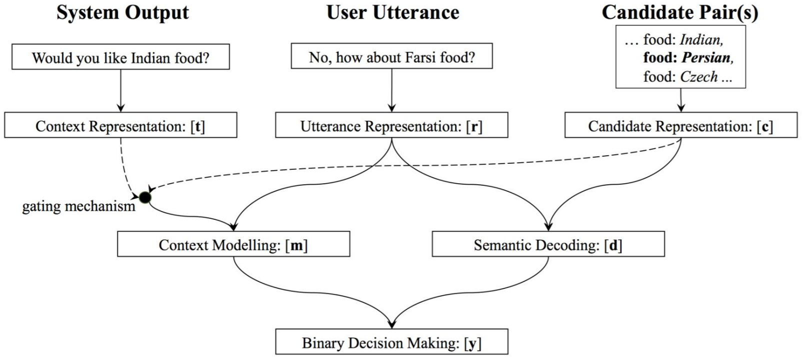
- Users: Task similar to DSTC2
    - Type in natural language sentences to fulfil the task
- Amazon Mechanical Turk: The dialogue system
    - Used DSTC2 ontology.
- WOZ focuses on semantic understanding
- Not testing robustness to ASR errors.



The model

Input:

- System dialogue acts preceding the user input
- User utterance
- A single candidate slot-value pair
    - Is FOOD=ITALIAN expressed in ‘*I’m looking for good pizza*’.


System Output
Would you like Indian food?
Context Representation: [t]
User Utterance
No, how about Farsi food?
Utterance Representation: [r]
Candidate Pair(s)
… food: Indian,
food: Persian,
food: Czech ...
Candidate Representation: [c]
gating mechanism
Context Modelling: [m]
Semantic Decoding: [d]
Binary Decision Making: [y]

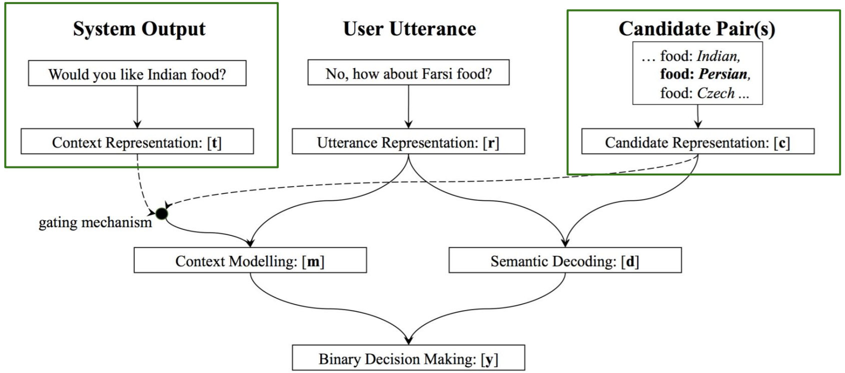
Output

- Is the slot-value pair is part of the state or not

The model (continued)

- Candidate Pair
    - Representation: Word embeddings of the slot/value names
    - In case of multiple words: add embeddings

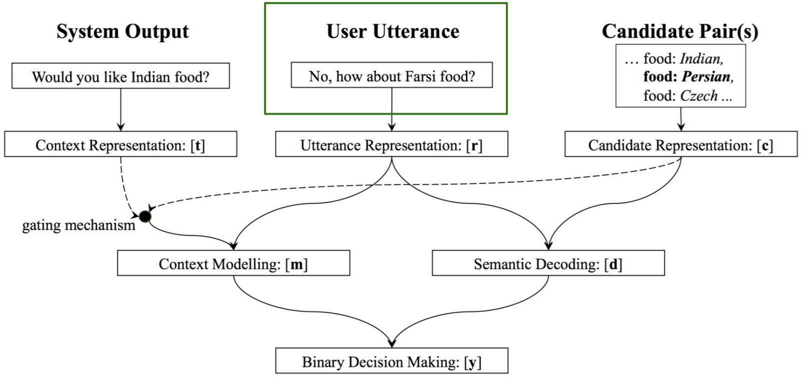
- The system dialogue acts (tq, ts, tv)
    - System Request: tq
    - System Confirm: (ts, tv)
    - Representation: Word embeddings of the slot/value names


System Output
Would you like Indian food?
Context Representation: [t]
User Utterance
No, how about Farsi food?
Utterance Representation: [r]
Candidate Pair(s)
… food: Indian,
food: Persian,
food: Czech …
Candidate Representation: [c]
gating mechanism
Context Modelling: [m]
Semantic Decoding: [d]
Binary Decision Making: [y]

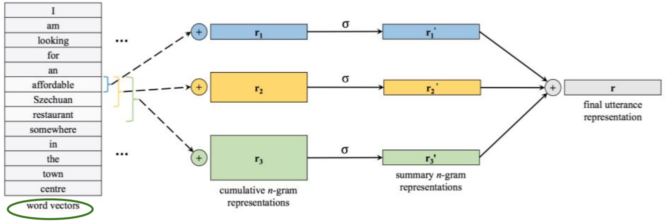
User Utterance Representation

User utterance encoder

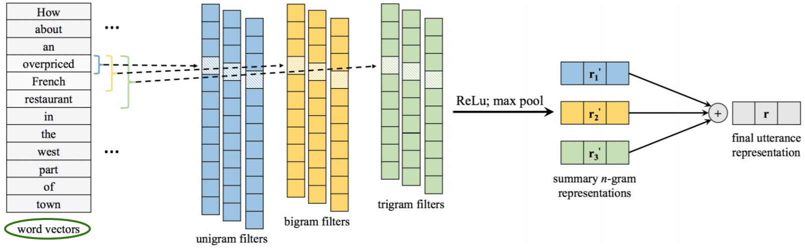
They consider 2 models


System Output
Would you like Indian food?
Context Representation: [t]
User Utterance
No, how about Farsi food?
Utterance Representation: [r]
Candidate Pair(s)
… food: Indian,
food: Persian,
food: Czech …
Candidate Representation: [c]
gating mechanism
Context Modelling: [m]
Semantic Decoding: [d]
Binary Decision Making: [y]

- NBT-DNN



- NBT-CNN



Paragram-SL99

Why?

Semantic Decoding

- Intent for the current candidate pair
  - Example: ‘I want Thai food’ with food=Thai

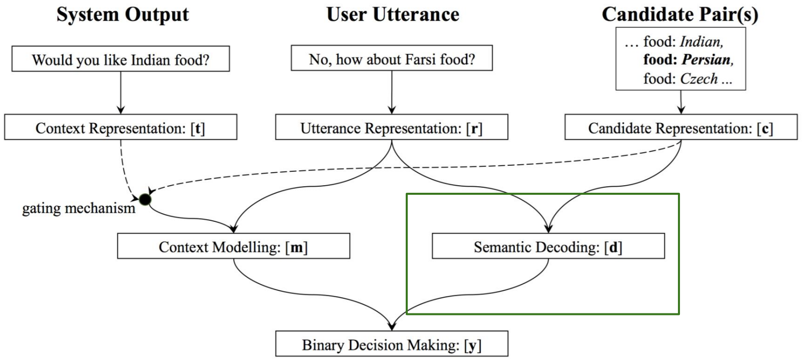
$\mathbf{c_s}$: Vector space representations of slot name

$\mathbf{c_v}$: Vector space representations of slot value

$$\mathbf{c} \;= \sigma\big(W_c^s(\mathbf{c_s} + \mathbf{c_v}) + b_c^s\big)$$

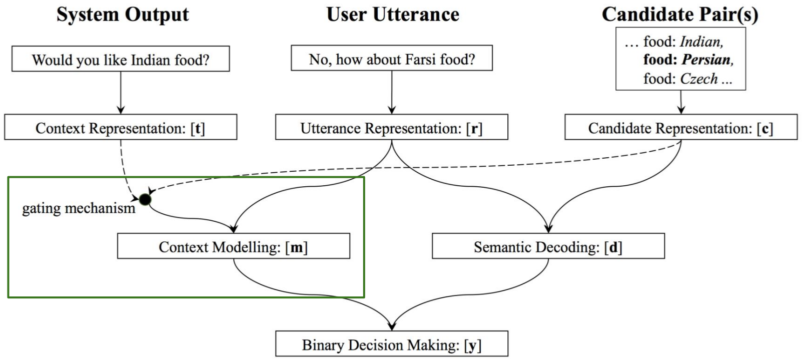
$$\mathbf{d} = \mathbf{r} \otimes \mathbf{c}$$

$\otimes$: Element-wise vector multiplication


System Output
User Utterance
Candidate Pair(s)
Would you like Indian food?
No, how about Farsi food?
… food: Indian,
food: Persian,
food: Czech …
Context Representation: [t]
Utterance Representation: [r]
Candidate Representation: [c]
gating mechanism
Context Modelling: [m]
Semantic Decoding: [d]
Binary Decision Making: [y]

Context Modelling

Encode context of the System


System Output
User Utterance
Candidate Pair(s)
Would you like Indian food?
No, how about Farsi food?
... food: *Indian*,
food: *Persian*,
food: *Czech* ...
Context Representation: [**t**]
Utterance Representation: [**r**]
Candidate Representation: [**c**]
gating mechanism
Context Modelling: [**m**]
Semantic Decoding: [**d**]
Binary Decision Making: [**y**]

$\mathbf{c_s}$: Vector space representations of slot name

$\mathbf{c_v}$: Vector space representations of slot value

$\mathbf{t_q}$: Vector space representations system request

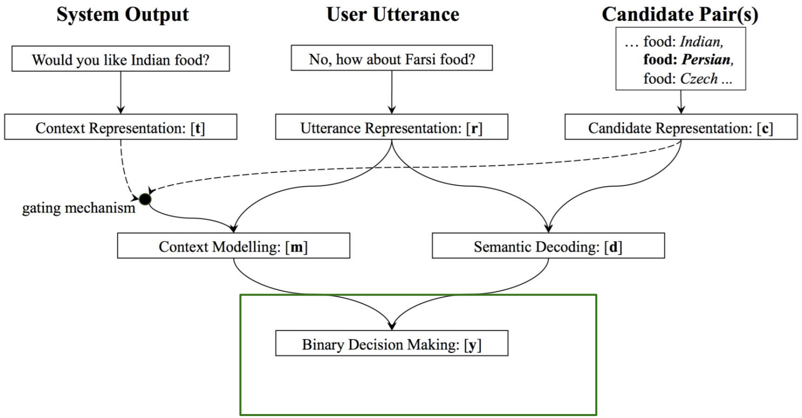
$\mathbf{t_s}$: Vector space representations of the slot name for system confirm

$\mathbf{t_v}$: Vector space representations of the slot value for system confirm

$$\mathbf{m_r} = (\mathbf{c_s} \cdot \mathbf{t_q})\mathbf{r}$$

$$\mathbf{m_c} = (\mathbf{c_s} \cdot \mathbf{t_s})(\mathbf{c_v} \cdot \mathbf{t_v})\mathbf{r}$$

Binary Decision making


System Output
User Utterance
Candidate Pair(s)
Would you like Indian food?
No, how about Farsi food?
... food: *Indian*,
food: *Persian*,
food: *Czech* ...
Context Representation: [**t**]
Utterance Representation: [**r**]
Candidate Representation: [**c**]
gating mechanism
Context Modelling: [**m**]
Semantic Decoding: [**d**]
Binary Decision Making: [**y**]

$\mathbf{c_s}$: Vector space representations of slot name

$\mathbf{c_v}$: Vector space representations of slot value

$\mathbf{t_q}$: Vector space representations system request

$\mathbf{t_s}$: Vector space representations of the slot name for system confirm

$\mathbf{t_v}$: Vector space representations of the slot value for system confirm

$$\mathbf{m_r} = (\mathbf{c_s} \cdot \mathbf{t_q})\mathbf{r}$$

$$\mathbf{m_c} = (\mathbf{c_s} \cdot \mathbf{t_s})(\mathbf{c_v} \cdot \mathbf{t_v})\mathbf{r}$$

$\phi_n(x)$: Maps input vector x to dimension n

$\phi_n(x) = \sigma(Wx + b)$

$$\mathbf{y} = \phi_2\big(\phi_{100}(\mathbf{d}) + \phi_{100}(\mathbf{m_r}) + \phi_{100}(\mathbf{m_c})\big)$$

Belief State Update Mechanism

Multiple ASR hypothesis (multiple user utterances): noisy speech recognition

This paper proposes a simple way to deal with that:

For dialogue turn t

h^t : List of N ASR hypothesis

p_i^t : Posterior probability for h_t^i

sys^{t-1} : preceding system output

s : slot

v : slot value, $v \in V_s$

The NBT model gives us: $\mathbb{P}(s, v \mid h_i^t, sys^{t-1})$

Turn level probability that (s, v) was expressed in the given hypothesis

The predictions for N such hypotheses are then combined as:

$$\mathbb{P}(s, v \mid h^t, sys^{t-1}) = \sum_{i=1}^{N} p_i^t \, \mathbb{P}\left(s, v \mid h_i^t, sys^t\right)$$

Belief State Update Mechanism

For dialogue turn t

h^t : List of N ASR hypothesis

p_i^t : Posterior probability for h_t^i

sys^{t-1} : preceding system output

s : slot

v : slot value, $v \in V_s$

$\mathbb{P}(s, v \mid h_i^t, sys^{t-1})$: NBT Model

$$\mathbb{P}(s, v \mid h^t, sys^{t-1}) = \sum_{i=1}^{N} p_i^t \, \mathbb{P}\left(s, v \mid h_i^t, sys^t\right)$$

For the final update: need to combined with the (cumulative) belief state up to time (t − 1)

$$\begin{aligned}\mathbb{P}(s, v \mid h^{1:t}, sys^{1:t-1}) \; &= \; \lambda \, \mathbb{P}\left(s, v \mid h^t, sys^{t-1}\right) \\ &+ (1-\lambda) \, \mathbb{P}\left(s, v \mid h^{1:t-1}, sys^{1:t-2}\right)\end{aligned}$$

λ : constant coefficient: 0.55, tuned on DSCT2 dev set

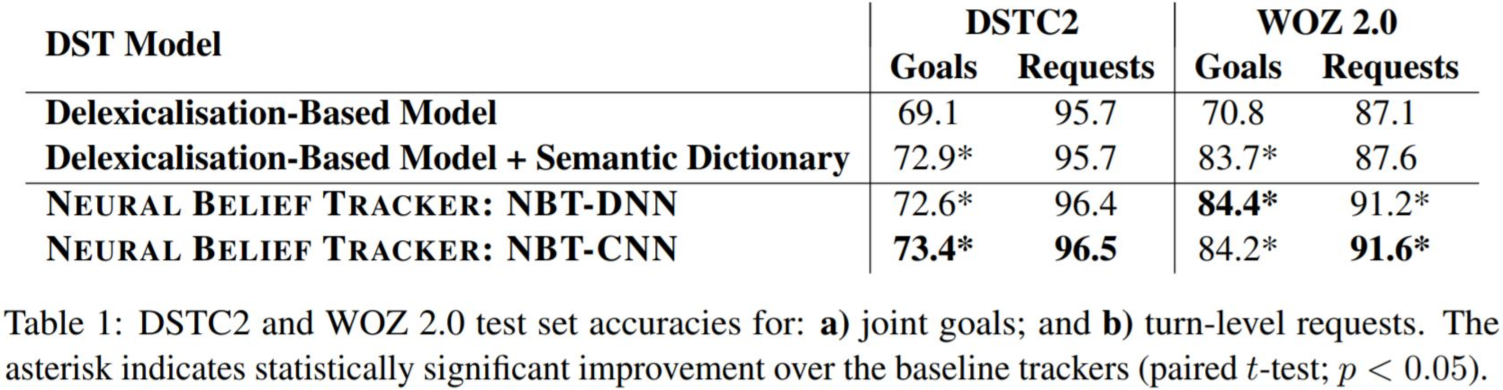
Slot Value Detection

$$V_s^t = \{v \in V_s \;|\; \mathbb{P}\left(s, v \mid h^{1:t}, sys^{1:t-1}\right) \geq 0.5\}$$

- For requests,
  - all slots in V_{req}^t are deemed to have been requested.
  - Requestable slots: single-turn user queries

- For informable (i.e. goal-tracking) slots,
  - The value with highest probability in V_s^t

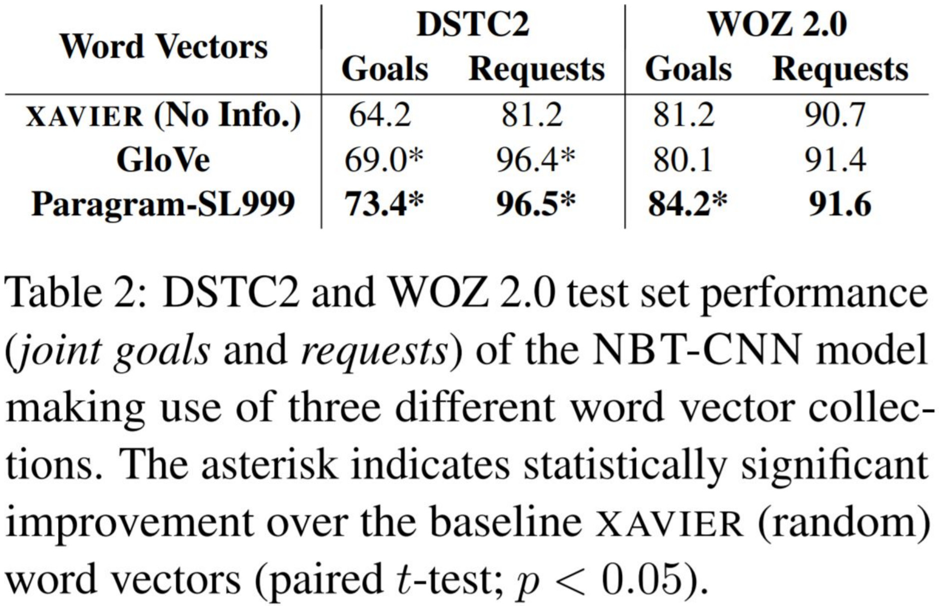
Experiments



DST Model	**DSTC2**		**WOZ 2.0**	
	Goals	**Requests**	**Goals**	**Requests**
Delexicalisation-Based Model	69.1	95.7	70.8	87.1
Delexicalisation-Based Model + Semantic Dictionary	72.9*	95.7	83.7*	87.6
Neural Belief Tracker: NBT-DNN	72.6*	96.4	**84.4***	91.2*
Neural Belief Tracker: NBT-CNN	**73.4***	**96.5**	84.2*	**91.6***

Table 1: DSTC2 and WOZ 2.0 test set accuracies for: **a)** joint goals; and **b)** turn-level requests. The asterisk indicates statistically significant improvement over the baseline trackers (paired t-test; $p < 0.05$).

Experiments (continued)



Word Vectors	DSTC2		WOZ 2.0	
	Goals	**Requests**	**Goals**	**Requests**
XAVIER (No Info.)	64.2	81.2	81.2	90.7
GloVe	69.0*	96.4*	80.1	91.4
Paragram-SL999	**73.4***	**96.5***	**84.2***	**91.6**

Table 2: DSTC2 and WOZ 2.0 test set performance (*joint goals* and *requests*) of the NBT-CNN model making use of three different word vector collections. The asterisk indicates statistically significant improvement over the baseline XAVIER (random) word vectors (paired t-test; $p < 0.05$).

Discussion

What do you think about the results?

Why does GloVe perform that bad?

Thoughts on the belief update mechanism?

NBT-CNN vs NBT-DNN?

WoZ2.0 vs DSTC2?