

Learning End-to-End Goal-Oriented Dialog

Antoine Bordes, Y-Lan Boureau, Jason Weston

¹Facebook

ICLR 2017

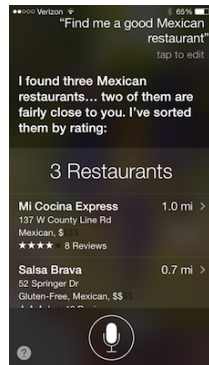
Presenter: Jack Lanchantin

Outline

- 1 Goal Oriented Dialog Tasks
- 2 Models
 - Rule-based Systems
 - Classic IR Methods
 - Supervised Embedding Models
 - Memory Networks
- 3 Experiments
- 4 Conclusions

Goal Oriented Dialog

- **Goal oriented dialog**: understand a user request and complete a task
- **“Slot filling”**: predefine the structure of a dialog state as a set of slots to be filled
 - e.g. restaurant reservation system slots: location, price, cuisine type
 - Doesn't scale to new domains



End-to-end Dialog Systems

- All components are trained on past dialog. Usually done in **non-goal oriented “chit-chat”** settings
- **Goal oriented dialog goes beyond language modeling.**
 - e.g. asking questions to clearly define user requests, querying KBs
- **This paper:** can end-to-end models be competitive with traditional slot-filling models?

Restaurant Reservation Simulation

- **Goal**: book a table at a restaurant
- **Data**: simulation based on a KB whose facts contain restaurants and four types of properties:
 - 1 **Type**: 10 choices (French, Thai, ...)
 - 2 **Location**: 10 choices (London, Tokyo, ...)
 - 3 **Price Range**: 3 choices (cheap, moderate, expensive)
 - 4 **Rating**: 8 choices (1,2,...,8)
- **Approach**: evaluate models on 4 sub-tasks of restaurant reservation

Task 1: Issuing API Calls

Hi!

Hello, what can I help you with today?

I'd like to book a table for six people in an expensive price range with British food.

I'm on it!

Where should it be?

In London.

Ok, let me look into some options for you.

```
api call(British, London, Six, Expensive)
```

- Bot must ask questions for filling the missing fields and eventually generate the correct corresponding API call.

Task 2: Updating API Calls

```
api_call(British, London, Six, Expensive)
```

Actually i would prefer for four.

Sure. Is there anything else to update?

No.

Ok let me look into some options for you.

```
api_call(British, London, Four, Expensive)
```

- Users ask to update their requests between 1 and 4 times (sampled uniformly). Bot must correct for changes.

Task 3: Displaying Options

```
* The_Place R_phone The_Place_phone
* The_Place R_cuisine british
* The_Place R_address The_Place_address
* The_Place R_location london
* The_Place R_availability four
* The_Place R_price expensive
* The_Place R_rating 7
* The_Fancy_Pub R_phone The_Fancy_Pub_phone
* The_Fancy_Pub R_cuisine british
* The_Fancy_Pub R_address The_Fancy_Pub_address
* The_Fancy_Pub R_location london
* The_Fancy_Pub R_availability four
* The_Fancy_Pub R_price expensive
* The_Fancy_Pub R_rating 8
```

What do you think of this option: The_Fancy_Pub

No, I don't like that.

Sure, let me find an other option for you.

What do you think of this option: The_Place

Let's do it!

Great let me do the reservation.

- Given a user request, **query the KB using API call** and add the facts from the call to dialog history.
- **Propose options to users** by listing restaurant names sorted by rating

Task 4: Providing Extra Info

Can you provide me the address?

Here it is: The_Place_address.

You rock!

Is there anything else I can help you with?

No thanks.

You're welcome.

- Users ask for the **phone number** of the restaurant, **address**, or **both**
- Bot must learn to use the KB facts correctly to answer

Task 5: Full Dialog



Task 5 Conducting full dialogs

Datasets

	Tasks	T1	T2	T3	T4	T5	T6	Concierge
DIALOGS <i>Average statistics</i>	Number of utterances:	12	17	43	15	55	54	8
	- user utterances	5	7	7	4	13	6	4
	- bot utterances	7	10	10	4	18	8	4
	- outputs from API calls	0	0	23	7	24	40	0
DATASETS <i>Tasks 1-5 share the same data source</i>	Vocabulary size	3,747					1,229	8,629
	Candidate set size	4,212					2,406	11,482
	Training dialogs	1,000					1,618	3,249
	Validation dialogs	1,000					500	403
	Test dialogs	1,000 ^(*)					1,117	402

Datasets

	Tasks	T1	T2	T3	T4	T5	T6	Concierge
DIALOGS <i>Average statistics</i>	Number of utterances:	12	17	43	15	55	54	8
	- user utterances	5	7	7	4	13	6	4
	- bot utterances	7	10	10	4	18	8	4
	- outputs from API calls	0	0	23	7	24	40	0
DATASETS <i>Tasks 1-5 share the same data source</i>	Vocabulary size	3,747					1,229	8,629
	Candidate set size	4,212					2,406	11,482
	Training dialogs	1,000					1,618	3,249
	Validation dialogs	1,000					500	403
	Test dialogs	1,000 ^(*)					1,117	402

- At each turn of the dialog, test if model can **predict** bot utterances and API calls **by selecting a candidate**
- Candidates are ranked from a set of all bot utterances and API calls

Outline

- 1 Goal Oriented Dialog Tasks
- 2 Models
 - Rule-based Systems
 - Classic IR Methods
 - Supervised Embedding Models
 - Memory Networks
- 3 Experiments
- 4 Conclusions

Rule-based Systems

- For tasks T1-T5, it is possible to hand-code a rule based system that achieves 100%.
- Tasks T6 and Concierge are not simulated, so they require more complex rules (which is hopefully where machine learning can help).

Outline

1 Goal Oriented Dialog Tasks

2 Models

- Rule-based Systems
- **Classic IR Methods**
- Supervised Embedding Models
- Memory Networks

3 Experiments

4 Conclusions

- **TF-IDF Match**

- Rank candidate responses by **matching score between the input and the response**

- **Nearest Neighbor**

- Using the input, **find the most similar conversation in training set**, and output the response from that conversation

Outline

- 1 Goal Oriented Dialog Tasks
- 2 **Models**
 - Rule-based Systems
 - Classic IR Methods
 - **Supervised Embedding Models**
 - Memory Networks
- 3 Experiments
- 4 Conclusions

Supervised Embedding Models

- Candidate response y scored against input x : $f(x, y) = (Ax)^T By$
 - A and B are $d \times V$ word embedding matrices
- Embeddings trained with margin ranking loss: $f(x, y) > m + f(x, \bar{y})$

Outline

1 Goal Oriented Dialog Tasks

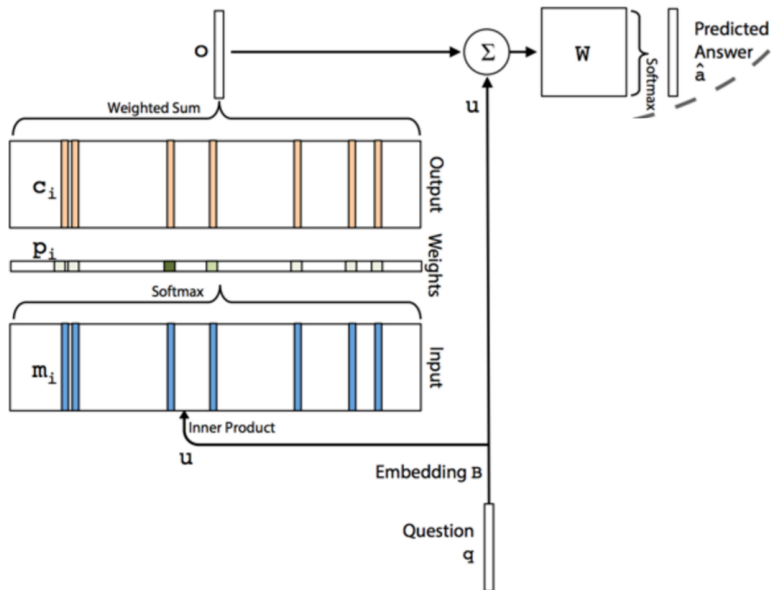
2 Models

- Rule-based Systems
- Classic IR Methods
- Supervised Embedding Models
- **Memory Networks**

3 Experiments

4 Conclusions

Memory Networks



Match Type Features

- Augment the vocab with 7 special words, one for each of the KB entity types (cuisine type, location, price range, party size, rating, phone number and address).

Results

Task	Rule-based Systems	TF-IDF Match		Nearest Neighbor	Supervised Embeddings	Memory Networks	
		no type	+ type			no match type	+ match type
T1: Issuing API calls	100 (100)	5.6 (0)	22.4 (0)	55.1 (0)	100 (100)	99.9 (99.6)	100 (100)
T2: Updating API calls	100 (100)	3.4 (0)	16.4 (0)	68.3 (0)	68.4 (0)	100 (100)	98.3 (83.9)
T3: Displaying options	100 (100)	8.0 (0)	8.0 (0)	58.8 (0)	64.9 (0)	74.9 (2.0)	74.9 (0)
T4: Providing information	100 (100)	9.5 (0)	17.8 (0)	28.6 (0)	57.2 (0)	59.5 (3.0)	100 (100)
T5: Full dialogs	100 (100)	4.6 (0)	8.1 (0)	57.1 (0)	75.4 (0)	96.1 (49.4)	93.4 (19.7)
T1(OOV): Issuing API calls	100 (100)	5.8 (0)	22.4 (0)	44.1 (0)	60.0 (0)	72.3 (0)	96.5 (82.7)
T2(OOV): Updating API calls	100 (100)	3.5 (0)	16.8 (0)	68.3 (0)	68.3 (0)	78.9 (0)	94.5 (48.4)
T3(OOV): Displaying options	100 (100)	8.3 (0)	8.3 (0)	58.8 (0)	65.0 (0)	74.4 (0)	75.2 (0)
T4(OOV): Providing inform.	100 (100)	9.8 (0)	17.2 (0)	28.6 (0)	57.0 (0)	57.6 (0)	100 (100)
T5(OOV): Full dialogs	100 (100)	4.6 (0)	9.0 (0)	48.4 (0)	58.2 (0)	65.5 (0)	77.7 (0)
T6: Dialog state tracking 2	33.3 (0)	1.6 (0)	1.6 (0)	21.9 (0)	22.6 (0)	41.1 (0)	41.0 (0)
Concierge ^(*)	n/a	1.1 (0.2)	n/a	13.4 (0.5)	14.6 (0.5)	16.7 (1.2)	n/a ^(†)

- Metrics: Per-response accuracy and (Per-dialog accuracy)

Conclusions

- Open dataset and task set for evaluating end-to-end goal-oriented dialog learning methods in a systematic and controlled way.
- The breakdown in tasks will help focus research and development to improve the learning methods
- Illustrated how to use the testbed using Memory Networks, which prove an effective model on these tasks relative to other baselines, but are still lacking in some key areas.