Learning End-to-End Goal-Oriented Dialog

Antoine Bordes, Y-Lan Boureau, Jason Weston

\(^1\)Facebook

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
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?
**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

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

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)
Task 2: Updating API Calls

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 another 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

Users ask for the **phone number** of the restaurant, **address**, or both.

Bot must learn to use the KB facts correctly to answer.

---

**Chat Example:**

User: Can you provide me the address?

Bot: Here it is: The Place address.

User: You rock!

Bot: Is there anything else I can help you with?

User: No thanks.

Bot: You're welcome.
Task 5: Full Dialog

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)

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)

* 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.

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.
Datasets

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

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.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
<th>Concierge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of utterances:</td>
<td>12</td>
<td>17</td>
<td>43</td>
<td>15</td>
<td>55</td>
<td>54</td>
<td>8</td>
</tr>
<tr>
<td>- user utterances</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>13</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>- bot utterances</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>4</td>
<td>18</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>- outputs from API calls</td>
<td>0</td>
<td>0</td>
<td>23</td>
<td>7</td>
<td>24</td>
<td>40</td>
<td>0</td>
</tr>
</tbody>
</table>

Datasets

<table>
<thead>
<tr>
<th>DATASETS</th>
<th>Tasks 1-5 share the same data source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vocabulary size</td>
<td>3,747</td>
</tr>
<tr>
<td>Candidate set size</td>
<td>4,212</td>
</tr>
<tr>
<td>Training dialogs</td>
<td>1,000</td>
</tr>
<tr>
<td>Validation dialogs</td>
<td>1,000</td>
</tr>
<tr>
<td>Test dialogs</td>
<td>1,000(*)</td>
</tr>
</tbody>
</table>
Outline

1 Goal Oriented Dialog Tasks

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

3 Experiments

4 Conclusions
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).
1. Goal Oriented Dialog Tasks

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

3. Experiments

4. Conclusions
Classic IR Methods

- **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
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

![Diagram of Memory Networks](image)
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

<table>
<thead>
<tr>
<th>Task</th>
<th>Rule-based Systems</th>
<th>TF-IDF Match</th>
<th>Nearest Neighbor</th>
<th>Supervised Embeddings</th>
<th>Memory Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>no type</td>
<td>+ type</td>
<td>no match type</td>
<td>+ match type</td>
</tr>
<tr>
<td>T1: Issuing API calls</td>
<td>100 (100)</td>
<td>5.6 (0)</td>
<td>22.4 (0)</td>
<td>55.1 (0)</td>
<td>100 (100)</td>
</tr>
<tr>
<td>T2: Updating API calls</td>
<td>100 (100)</td>
<td>3.4 (0)</td>
<td>16.4 (0)</td>
<td>68.3 (0)</td>
<td>68.4 (0)</td>
</tr>
<tr>
<td>T3: Displaying options</td>
<td>100 (100)</td>
<td>8.0 (0)</td>
<td>8.0 (0)</td>
<td>58.8 (0)</td>
<td>64.9 (0)</td>
</tr>
<tr>
<td>T4: Providing information</td>
<td>100 (100)</td>
<td>9.5 (0)</td>
<td>17.8 (0)</td>
<td>28.6 (0)</td>
<td>57.2 (0)</td>
</tr>
<tr>
<td>T5: Full dialogs</td>
<td>100 (100)</td>
<td>4.6 (0)</td>
<td>8.1 (0)</td>
<td>57.1 (0)</td>
<td>75.4 (0)</td>
</tr>
<tr>
<td>T1(OOV): Issuing API calls</td>
<td>100 (100)</td>
<td>5.8 (0)</td>
<td>22.4 (0)</td>
<td>44.1 (0)</td>
<td>60.0 (0)</td>
</tr>
<tr>
<td>T2(OOV): Updating API calls</td>
<td>100 (100)</td>
<td>3.5 (0)</td>
<td>16.8 (0)</td>
<td>68.3 (0)</td>
<td>68.3 (0)</td>
</tr>
<tr>
<td>T3(OOV): Displaying options</td>
<td>100 (100)</td>
<td>8.3 (0)</td>
<td>8.3 (0)</td>
<td>58.8 (0)</td>
<td>65.0 (0)</td>
</tr>
<tr>
<td>T4(OOV): Providing inform.</td>
<td>100 (100)</td>
<td>9.8 (0)</td>
<td>17.2 (0)</td>
<td>28.6 (0)</td>
<td>57.0 (0)</td>
</tr>
<tr>
<td>T5(OOV): Full dialogs</td>
<td>100 (100)</td>
<td>4.6 (0)</td>
<td>9.0 (0)</td>
<td>48.4 (0)</td>
<td>58.2 (0)</td>
</tr>
<tr>
<td>T6: Dialog state tracking 2</td>
<td>33.3 (0)</td>
<td>1.6 (0)</td>
<td>1.6 (0)</td>
<td>21.9 (0)</td>
<td>22.6 (0)</td>
</tr>
</tbody>
</table>

**Concierge(*)**

<table>
<thead>
<tr>
<th></th>
<th>Rule-based Systems</th>
<th>TF-IDF Match</th>
<th>Nearest Neighbor</th>
<th>Supervised Embeddings</th>
<th>Memory Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>n/a</td>
<td>1.1 (0.2)</td>
<td>n/a</td>
<td>13.4 (0.5)</td>
<td>14.6 (0.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16.7 (1.2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>n/a (†)</td>
</tr>
</tbody>
</table>

- **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.