Lecture 8

Training
Conversational Virtual Assistants

By Monica Lam and Giovanni Campagna
Outline

• Introduction to traditional data acquisition approaches
• Wizard-of-Oz (WOZ) conversations: manual annotation
• Machine-to-Machine (M2M): synthesis then manual paraphrase
• Genie: Few-shot + synthesis for WOZ conversations
Quiz

HOW TO ACQUIRE DATA?
Where do we get training data?
1. Real-Life Recordings of Human Agents

- Human user, human agent
- From real conversations (phone marketing, customer support, ...)
  - “This call may be monitored for quality and training purposes”
  - Confidentiality: Outsource providers often cannot see a client’s data
- Not available to academia
- Cannot see the mistakes of an automated agent
2. Real-Life Recordings of an Agent

- Human user, computer agent
- Chicken and egg (aka *bootstrapping*): we don’t have a real agent until we train the model
- Same issues
  - Confidentiality:
    - Outsource providers often cannot see a client’s data
  - Not available to academia
3. Wizard-of-Oz (WOZ)

• Human user, human agent
• Paid workers (crowdworkers) on both sides
  • One worker pretends to be an automated agent
  • Another worker pretends to be a user, following a prompt
• The worker is given specific tasks
  • Does not reflect real life
    • changing their minds, exploring options
    • errors made by the agent
How to Annotate Human 2 Human Traces?

- Human agent uses GUI to access the database
- Record interaction with that UI (clicking, typing, etc.) each turn
- Convert UI interaction trace to slots

- Simple, cheap (no extra effort on top of call center worker)
  - Are we done?
Outline

• Introduction to traditional data acquisition approaches
• Wizard-of-Oz conversations: manual annotation
• Machine-to-Machine: Paraphrased Synthesis
• Genie: Few-shot + synthesis for WOZ conversations
Case Study: MultiWOZ [Budzianowski et al 2018]

• A large dataset of dialogues
  • Annotated with domain + slots at every turn
  • Application: concierge service in Cambridge, UK
  • 7 domains → 5 domains
    • restaurants, hotels, attractions, train, taxi
    • police, hospital dropped due to bad quality
  • Number of slots: 30
  • Closed ontology: 1,829 values
  • Representation: slot-value pairs
MultiWOZ Dataset

• **Users are asked to book**
  • One of various places (restaurant, hotel, train, taxi) and ask about attractions
  • Many dialogues cross multiple domains

• **Collected using Wizard-of-OZ conversations by crowdworkers**
  • ~56k turns train
  • ~7k turns test and ~7k turns dev
  • ~7 turns/dialogue on average
Example of a MultiWOZ dialogue

- U: I am looking for a restaurant. The restaurant should serve world food and should be in the centre.
- A: Unfortunately, I could not find a restaurant that serves that kind of food in that area.
- U: What expensive restaurants are there?
- A: The Cambridge Chop House is expensive and it's British food. Is that okay?
- U: No. Try looking for a restaurant that serves Korean food.
- A: I found the Little Seoul restaurant. Would you like me to make a reservation?
- U: Yes, a table for 2.
- A: I would be happy to reserve for you, what day and time are you looking for?
- U: Thank you, I need a reservation for 2 people at 13:15 on Monday.
- A: gg3dcufd is your confirmation code, may I help you with anything else?
- U: No that was everything. Thanks
DST of MultiWOZ

• Dialogue state
  • Cumulative information of all the slots in all the domains
• Given a dialogue with alternating user-agent turns in natural language: $u_1, a_1, u_2, a_2, \ldots$
  For each turn $i$,
    Predict dialogue state of $(u_1, a_1, \ldots, u_i)$
    Metric: Joint Accuracy: Accurate only if all slots are correct.
• Errors accumulate over time
  • The first turn is the easiest
Quiz

• If turn $i-1$ is wrong, is turn $i$ guaranteed to be wrong?
MultiWOZ 2.0

- Annotated by recording the UI of the agent worker
- Joint accuracy: 54.9

https://paperswithcode.com/sota/multi-domain-dialogue-state-tracking-on
Errors in Annotation

• UI traces do not capture exact DST at every turn
  • Agents delay entering search criteria by one-two turns
    • “What cuisine?” “Italian” “What price?” “cheap”
      → type cheap & Italian at once

• Agents resolve the answer in their mind only
  • “What cuisine?” “Italian” “What price?” “cheap”
    → type Italian, top result is cheap → reply to user
MultiWOZ 2.1 [Eric et al]

- 2.1: complete reannotation
  - Move slots to the right turn, add slots that the agent ignored
  - Fixed 32% of dialogue state annotations across 40% of the dialogue turns
- Joint accuracy
  - TripPyTripPy+SCoRe: 60.5%
  - TripPy+CoCoAug: 60.5%
  - TripPy+SaCLog: 60.6%
- TripPy:
  - Constructs a label map to handle value variants.
  - Three-way loss to be robust to errors
    - Copy from sentence, context, or ontology
- SCoRe:
  - Schema-aware curriculum learning

https://paperswithcode.com/sota/multi-domain-dialogue-state-tracking-on-1
Re-Annotations of MultiWOZ

- Reannotated 4 times manually to fix annotation errors
  - 2.1: Move slots to the right turn, add slots that the agent ignored
  - 2.2: Normalize types: numbers, times, enums
  - 2.3: Fix errors introduced in 2.1
  - 2.4: Enforced convention on when to include a slot or not
- State of the art on version 2.4 [Ye et al 2021]:
  - STAR [Ye et al 2021]: 73.6%
    - STAR achieves 56% on 2.1
    - Annotations make a big difference!
    - Very specialized architectures, closed terminology, not general
<table>
<thead>
<tr>
<th>Error Type</th>
<th>Conversation</th>
<th>MultiWOZ 2.1</th>
<th>MultiWOZ 2.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context Mismatch</td>
<td><strong>Usr</strong>: Hello, I would like to book a taxi from restaurant 2 two to the museum of classical archaeology.</td>
<td>taxi-destination=museum of archaeology and anthropology</td>
<td>taxi-destination=museum of classical archaeology</td>
</tr>
<tr>
<td>Mis-Annotation</td>
<td><strong>Usr</strong>: I need a place to dine in the centre of town.</td>
<td>rest.-area=none</td>
<td>rest.-area=centre</td>
</tr>
<tr>
<td>Not Mentioned</td>
<td><strong>Usr</strong>: I am planning a trip in Cambridge.</td>
<td>hotel-internet=dontcare</td>
<td>hotel-internet=none</td>
</tr>
<tr>
<td>Multiple Values</td>
<td><strong>Usr</strong>: Something classy nearby for dinner, preferably Italian or Indian cuisine?</td>
<td>rest.-food=Indian</td>
<td>rest.-food=Indian</td>
</tr>
<tr>
<td>Typo</td>
<td><strong>Usr</strong>: I am looking for a restaurant that serves Portuguese food.</td>
<td>rest.-food=Portuguese</td>
<td>rest.-food=Portuguese</td>
</tr>
<tr>
<td>Implicit Time Processing</td>
<td><strong>Usr</strong>: I need a train leaving after 10:00.</td>
<td>train-leaveat=10:15</td>
<td>train-leaveat=10:00</td>
</tr>
<tr>
<td>Slot Mismatch</td>
<td><strong>Usr</strong>: Can you please help me find a place to go in town in the same area as the hotel? Preferably a college.</td>
<td>attraction-name=college</td>
<td>attraction-name=none</td>
</tr>
<tr>
<td></td>
<td><strong>Sys</strong>: I recommend Charlie Chan. Would you like a table?</td>
<td>attraction-type=none</td>
<td>attraction-type=college</td>
</tr>
<tr>
<td>Incomplete Value</td>
<td><strong>Usr</strong>: Yes. Monday, 8 people, 10:30.</td>
<td>rest.-name=Charlie</td>
<td>rest.-name=Charlie Chan</td>
</tr>
<tr>
<td>Delayed Annotation</td>
<td><strong>Usr</strong>: Please recommend one and book it for 6 people.</td>
<td>hotel-book people=none</td>
<td>hotel-book people=6</td>
</tr>
<tr>
<td></td>
<td><strong>Sys</strong>: I would recommend express by holiday inn Cambridge.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>From what day should I book?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Usr</strong>: Starting Saturday. I need 5 nights for 6 people.</td>
<td>hotel-book people=6</td>
<td>hotel-book people=6</td>
</tr>
<tr>
<td>Unnecessary Annotation</td>
<td><strong>Usr</strong>: I am looking for a museum.</td>
<td>attraction-area=centre</td>
<td>attraction-area=none</td>
</tr>
<tr>
<td></td>
<td><strong>Sys</strong>: The Broughton house gallery is a museum in the centre.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Usr</strong>: That sounds good. Could I get their phone number?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Sentences That Cannot Be Represented As Slots

“I was hoping you could **recommend** something”.

“Are there any churches **or** museums on the east side?”

“I would like the **latest** train leaving that will arrive by 9:15 please”.

The agent cannot possibly return the result needed!
Outline

- Introduction to traditional data acquisition approaches
- Wizard-of-Oz conversations: manual annotation
- Machine-to-Machine: Paraphrased Synthesis
- Genie: Few-shot + synthesis for WOZ conversations
Schema-Guided Dialogue (SGD) Dataset

- Meaning representation: slot-value
- Extend approach of Sempre (Overnight paper) to conversation
  - Paraphrase with crowdsource workers
  - Train with paraphrase, test with paraphrase
- Conversation: Model user & system agents as:
  - (Probabilistic) domain independent rule-based systems
  - 11 user dialogue acts, 10 system dialogue acts

Data Acquisition

a. Generated scenario with (Probabilistic) domain independent rule-based systems
b. Vary the expression of the value
c. Generation with 1 template
d. Crowdsourced paraphrase

Relies heavily on manual paraphrases!
Synthesis $\rightarrow$ Paraphrase

(b) Distribution of dialogue acts in training set.
## SGD Dataset

<table>
<thead>
<tr>
<th>Metric</th>
<th>DSTC2</th>
<th>WOZ2.0</th>
<th>FRAMES</th>
<th>M2M</th>
<th>MultiWOZ</th>
<th>SGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of domains</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>7</td>
<td>16</td>
</tr>
<tr>
<td>No. of dialogues</td>
<td>1,612</td>
<td>600</td>
<td>1,369</td>
<td>1,500</td>
<td>8,438</td>
<td>16,142</td>
</tr>
<tr>
<td>Total no. of turns</td>
<td>23,354</td>
<td>4,472</td>
<td>19,986</td>
<td>14,796</td>
<td>113,556</td>
<td>329,964</td>
</tr>
<tr>
<td>Avg. turns per dialogue</td>
<td>14.49</td>
<td>7.45</td>
<td>14.60</td>
<td>9.86</td>
<td>13.46</td>
<td>20.44</td>
</tr>
<tr>
<td>Avg. tokens per turn</td>
<td>8.54</td>
<td>11.24</td>
<td>12.60</td>
<td>8.24</td>
<td><strong>13.13</strong></td>
<td>9.75</td>
</tr>
<tr>
<td>Total unique tokens</td>
<td>986</td>
<td>2,142</td>
<td>12,043</td>
<td>1,008</td>
<td>23,689</td>
<td><strong>30,352</strong></td>
</tr>
<tr>
<td>No. of slots</td>
<td>8</td>
<td>4</td>
<td>61</td>
<td>13</td>
<td>24</td>
<td>214</td>
</tr>
<tr>
<td>No. of slot values</td>
<td>212</td>
<td>99</td>
<td>3,871</td>
<td>138</td>
<td>4,510</td>
<td><strong>14,139</strong></td>
</tr>
</tbody>
</table>

Synthesis/Paraphrase: many more dialogues than MultiWOZ

Cost of paraphrases (including verification) is still a limiting factor.
Evaluation

- Evaluation set: synthesized then paraphrased
  - >= 95% accuracy intent classification
  - >= 95% F1 score slot tagging

Quiz: Great results! Are we done?
## Two Worlds

<table>
<thead>
<tr>
<th></th>
<th>MultiWOZ</th>
<th>Machine-2-Machine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semantics</td>
<td>Slot-values</td>
<td>Slot-values</td>
</tr>
<tr>
<td>Training Data Cost</td>
<td><strong>High cost</strong></td>
<td><strong>Moderate cost</strong></td>
</tr>
<tr>
<td>Test</td>
<td>WOZ</td>
<td>Paraphrases of synthesized conversations from a probabilistic state machine</td>
</tr>
<tr>
<td>Results</td>
<td>73% slot accuracy</td>
<td>95% intent 95% F1</td>
</tr>
</tbody>
</table>

We want to get an even lower annotation cost than M2M that can handle the realistic data in WOZ
# Best of Both Worlds: Genie

<table>
<thead>
<tr>
<th></th>
<th>MultiWOZ</th>
<th>Machine-2-Machine</th>
<th>Genie (on MultiWOZ)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Semantics</strong></td>
<td>Slot-values</td>
<td>Slot-values</td>
<td>ThingTalk</td>
</tr>
<tr>
<td><strong>Training Data Cost</strong></td>
<td><strong>High cost</strong> Error-prone</td>
<td><strong>Moderate cost</strong> Synthesis + manual paraphrases</td>
<td><strong>Low cost</strong> synthesis + few shot + self-training</td>
</tr>
<tr>
<td><strong>Test</strong></td>
<td>WOZ</td>
<td>Paraphrases of synthesized conversations from a probabilistic state machine</td>
<td>WOZ</td>
</tr>
<tr>
<td><strong>Results</strong></td>
<td>73% slot accuracy</td>
<td>95% intent 95% F1</td>
<td>87.5% slot accuracy 79.2% exact match</td>
</tr>
</tbody>
</table>
Outline

• Introduction to traditional data acquisition approaches
• Wizard-of-Oz conversations: manual annotation
• Machine-to-Machine: Paraphrased Synthesis
• Genie: Few-shot + synthesis for WOZ conversations
Agent Architecture

1. **Contextual Semantic Parser (NLU):**
   User utterance + context $\rightarrow$ user state

2. **ThingTalk Execution:**
   ThingTalk $\rightarrow$ result state

3. **Agent policy:** code to decide the response
   User context + result $\rightarrow$ agent state

4. **Natural Language Generator (NLG):**
   agent state $\rightarrow$ agent utterance

4. **Output Context:**
   Summarizes the dialogue to date
   Becomes input context for next turn
Genie’s Approach

1. Conversational data synthesis
   • Extend ThingTalk for conversations
2. A full agent implementation: not just dialogue tracking!
3. The gap between synthesis and WOZ
4. Contextual semantic parsing
5. Experimental results
Extend ThingTalk for Conversations

- ThingTalk:
  - Can represent every possible query
  - Add user and agent dialogue acts for conversations
  - Synthesis: schema-guided dialogues
Restaurant

Dialog Plan

Find a restaurant ➔ Book a restaurant

Skills

Restaurant [“restaurant”, “food place”]

id: Entity(Restaurant)

go: Location

price: Enum(cheap, moderate, expensive)

makeReservation [“reserve #”, “book #”]

restaurant: Entity(Restaurant)

book_people: Number [min=1]

book_day: Date [“for #”, “for # people”]

book_time: Time [“at #”, “for #”]

A Restaurant Assistant

Request
I’d like to find a moderately priced restaurant

ProposeOne
I have Terun. It’s a moderately priced restaurant that serves pizza.

ProposeN
I found Terun and Coconuts. Both are moderately priced.

AskAction
I like that. Can you help me book it? I need it for 3 people.

SearchRefine
I don’t like pizza. Do you have something Caribbean?

InfoQuestion
Can you tell me the address of Terun?

Large Language Models

Automatically uses large language models
to generate domain-specific training data & train
I don't like to be near the trains. Do you have something by the lake?

I'd like to find a moderately priced hotel.

I have Best Western. It's a moderately priced hotel that is by the train station.

I found Best Western and Holiday Inn. Both are moderately priced.

I don't like to be near the trains. Do you have something by the lake?

Can you tell me the address of Best Western?
Do you have something more recent?

What are the hits by Taylor Swift?

I have Shake it Off from 2014.

I found Shake it off and You Belong With Me.

I like that. Play the song please.

Do you have something more recent?

Can you tell me the year of release?
Data Synthesis

• Extend the template system used in AutoQA
• Generates a wide variety of user utterances in ThingTalk
  • Using the 900 templates, with automatic annotation and paraphrases, and data augmentation (with values)
• Generates a small number of alternative agent policies
  • Just use 1 template to generate agent text (we do not need to parse them)
Synthesize Variety in Training Data

1. Per-field annotations
   - Attributes
   - Genie Grammar Templates

2. Sentence compositions
   - Pre-trained Model

3. Domain-based paraphrases
   - Pre-trained Model
   - Genie Dialogue Models

4. Dialogues

Database + API Schemas

restaurant {  
cuisine : String,  
rating : Number,  
... }

restaurant

restaurant{
  cuisine: String,
  rating: Number,
  ...}

food/dishes$value food/cuisine/dishes

adj: $value

rating

passive verb: Rated $value stars
noun: $value stars
adj: $value-star

cuisine

cuisine: String

restaurant

restaurant{
  cuisine: String,
  rating: Number,
  ...}
Expressiveness of ThingTalk (MultiWOZ Examples)

“I was hoping you could recommend something”.

AskRecommend;

“Are there any churches or museums on the east side?”

Execute: @Attraction(), type == “church” || type == “museum”;

“I would like the latest train leaving that will arrive by 9:15 please”.

Execute: sort(leave_at desc of @Train(), arrive_by <= 9:15)[1];
### Comparison With Previous Representations

<table>
<thead>
<tr>
<th>Feature</th>
<th>MultiWOZ Slots</th>
<th>TreeDST</th>
<th>ThingTalk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executable</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Dialogue Acts</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Multi-domain turns</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Slot constraints</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Comparisons</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Logical And</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Logical Or, Not</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Projection</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ranking</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

TreeDST representation paper, Cheng et al; Apple; EMNLP 2020
Genie’s Approach

1. Conversational data synthesis
   • Extend ThingTalk for conversations
2. A full agent implementation: not just dialogue tracking!
3. The gap between synthesis and WOZ
4. Contextual semantic parsing
5. Experimental results
## Synthesis vs Execution

<table>
<thead>
<tr>
<th></th>
<th>Synthesis</th>
<th>Executing the Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User input</strong></td>
<td>Generate from template</td>
<td>Receive real input</td>
</tr>
<tr>
<td><strong>Execution</strong></td>
<td>Provide mock values (for efficiency)</td>
<td>Invoke the real database</td>
</tr>
<tr>
<td><strong>Agent Policy</strong></td>
<td>Probabilistic choose from alternatives to simulate different human agents</td>
<td>Decide on just one policy for each input.</td>
</tr>
</tbody>
</table>

### Diagram:

1. **Contextual Semantic Parser**
   - User utterance: "I want restaurants that offer Indian food"
   - Input user context:
     - RecommendMany;
     - Restaurant, price $\Rightarrow$ cheap
     - { name = "Pizza Hut City Centre", area = centre, ... }
     - { name = "The Missing Sock", area = east, ... }

2. **Execution**
   - User state:
     - Exec: Restaurant, food $\Rightarrow$ "indian"
     - & & price $\Rightarrow$ cheap
   - Agent context:
     - Exec:
       - Restaurant, food $\Rightarrow$ "indian" & & price $\Rightarrow$ cheap
       - { name = "Kohinoor", area = centre, ... }
       - { name = "Royal Spice", area = north, ... }

3. **Agent Policy**
   - Agent utterance: "Do you have a specific part of town in mind?"
   - Agent state:
     - SearchQuestion: area;
   - Output user context:
     - SearchQuestion: area;
     - Restaurant, food $\Rightarrow$ "indian" & & price $\Rightarrow$ cheap
     - { name = "Kohinoor", area = centre, ... }
     - { name = "Royal Spice", area = north, ... }
Genie’s Approach

1. Conversational data synthesis
   • Extend ThingTalk for conversations
2. A full agent implementation: not just dialogue tracking!
3. The gap between synthesis and WOZ
4. Contextual semantic parsing
5. Experimental results
MultiWOZ Dataset in ThingTalk (Few-Shot)

• Recall MultiWOZ dataset: 5 domains
  • ~56k turns train
  • ~7k turns test and ~7k turns dev
  • ~7 turns/dialogue on average

• MultiWOZ (ThingTalk):
  • A few-shot reannotation in ThingTalk Dialogue Representation
    • By experts!
    • ~1K turns of dev set: 1/63 of the original!
    • Full test set (7K turns)
Representation Power of ThingTalk

- **97.7% of the test set is representable**
  - 99.8% of the user utterances
  - 97.6% of the agent utterances
- What’s in MultiWOZ that cannot be represented?
  - User: (Treated as errors in DST experiment)
    - Questions that can’t be answered using the given database
    - Out-of-domain questions
    - Note: random chitchat is fine
  - Agent: (Marked as ‘invalid’ in DST experiment)
    - Asking the user to wait
Quiz

• Does it mean that our synthesized dataset resembles the validation dataset?
Genie’s Approach

1. Conversational data synthesis
   - Extend ThingTalk for conversations
2. A full agent implementation: not just dialogue tracking!
3. The gap between synthesis and WOZ
4. Contextual semantic parsing
5. Experimental results
An Example from MultiWOZ

- U: Please book a table for 5 at 14:30 on Wednesday at Royal Spice. I also need to find a place to stay

- A: I was able to book your table successfully. Your reference number is kqmxil0z. Now, what type of accommodations are you looking for today?

Quiz: Can we represent this in ThingTalk?

Quiz: Can we synthesize this example?
Synthesis vs WOZ

- Multi-domain turns
- Domain switches
- Abandoned transactions

Quiz: Should we add these transitions to the abstract state machine?
We can’t synthesize all possible states exhaustively!
Quantitative Differences
between the WOZ validation dataset and synthesized data

- 68% Trained
- 14% Not Synthesizable
- 15% Not Representable
- 2% Not Trained

Validation data set of MultiWoz

Quiz: What’s the difference between WOZ and real life?
Quiz

• The semantic parser may correctly parse something that cannot be synthesized.
• What happens to the agent?
# Synthesis vs Execution

<table>
<thead>
<tr>
<th>Synthesis</th>
<th>Executing the Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User input</strong></td>
<td><strong>Generate from template</strong></td>
</tr>
<tr>
<td><strong>Execution</strong></td>
<td><strong>Provide mock values (for efficiency)</strong></td>
</tr>
<tr>
<td><strong>Agent Policy</strong></td>
<td><strong>Probabilistic choose from alternatives to simulate different human agents</strong></td>
</tr>
</tbody>
</table>

1. **Contextual Semantic Parser**
   - User utterance: “I want restaurants that offer Indian food”
   - User state: RecommendMany; Restaurant, food = “indian” && price = cheap
     - { name = "Pizza Hut City Centre", area = centre, ... }
     - { name = "The Missing Sock", area = east, ... }

2. **Execution**
   - Agent context
     - Exec; Restaurant, food = “indian” && price = cheap
       - { name = "Kohinoor", area = centre, ... }
       - { name = "Royal Spice", area = north, ... }
   - Output user context
     - SearchQuestion: area;
       - Restaurant, food = “indian” && price = cheap
         - { name = "Kohinoor", area = centre, ... }
         - { name = "Royal Spice", area = north, ... }

3. **Agent Policy**
   - "Do you have a specific part of town in mind?"
Quiz

• How to improve the semantic parser for unsynthesizable ThingTalk sentences?
Training Strategy for WOZ

• Add Few-shot manually annotated WOZ data
  • As realistic as possible
  • Encodes unexpected user behavior
  • Improves variability of natural language
  • Only need a small amount
Self-Training

• **Goal**: take advantage of *unannotated* WOZ dialogue data
• **Self-training**: Use the best parser so far to annotate new data
  • Train a semantic parser with synthesis + few shot
  • Use that to generate ThingTalk for unannotated data
• **Note**: WOZ has both user and agent sentences
  • Also need to train semantic parser to parse *agent* utterances
• **Annotation is not perfect**
  • Cheap way to use unannotated data
  • Empirically self-training helps
Genie’s Approach

1. Conversational data synthesis
   • Extend ThingTalk for conversations
2. A full agent implementation: not just dialogue tracking!
3. The gap between synthesis and WOZ
4. Contextual semantic parsing
5. Experimental results
Contextual Semantic Parsing with BART

- Concatenate context and utterance into a single input
- BART output is directly the desired user state

SearchQuestion(food): @Restaurant(),
  price == moderate #[results=…]

I’m looking for an Italian restaurant

Execute: @Restaurant(),
  price == enum moderate && food == “Italian”
Fine-tuning BART

- Fine-tune BART on triples of (context, utterance, user state)
- Treat ThingTalk as text (input and output)
  - Tokens are transformed to make them look like natural language ("price_range" → "price range")

- No architecture changes
  - No new parameters
- Fine-tune all parameters
Synthesis + Few Shot

Domain Information

Data Synthesis

State Machine: Policy Function + Templates

Value Datasets

Augmentation

Automatic Paraphrasing

Fine-tuned BART

Few-Shot Annotated Data

2nd Fine-tuned Model
Synthesis + Few Shot + Self-Training

Domain Information

State Machine: Policy Function + Templates

Value Datasets

Data Synthesis

Augmentation

Automatic Paraphrasing

Fine-tuned BART

Few-Shot Annotated Data

2nd Fine-tuned Model

Self-training: Automatically Annotated Data

3rd Fine-tuned Model
Genie’s Approach

1. Conversational data synthesis
   • Extend ThingTalk for conversations
2. A full agent implementation: not just dialogue tracking!
3. The gap between synthesis and WOZ
4. Contextual semantic parsing
5. Experimental results
Experiment: Applying ThingTalk to MultiWOZ

- MultiWOZ: reannotation of (partial) dev, test set with ThingTalk
- Abstract state machine
  - 20 agent transitions, 43 user transitions
- Training set:
  - 831k synthesized
  - ~1k turns few-shot (2% of original training)
  - 56k self-trained
  - We use the semantic parser to annotate the training data
Test Accuracy

- Annotate test set (7k turns), partial dev set (1K turns) in ThingTalk

<table>
<thead>
<tr>
<th>Training Set</th>
<th>Exact match</th>
<th>Slots only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Few-shot only</td>
<td>73.7%</td>
<td>81.6%</td>
</tr>
<tr>
<td>Genie</td>
<td>79.2%</td>
<td>87.5%</td>
</tr>
</tbody>
</table>
Performance of Genie on MultiWOZ

Dev Accuracy
Annotate test set (7k turns), partial dev set (1K turns) in ThingTalk

<table>
<thead>
<tr>
<th>Training Set</th>
<th>Exact match</th>
<th>Slots only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Few-shot only</td>
<td>75.6%</td>
<td>81.7%</td>
</tr>
<tr>
<td>Synthetic only</td>
<td>61.8%</td>
<td>73.1%</td>
</tr>
<tr>
<td>Genie</td>
<td><strong>81.4%</strong></td>
<td><strong>88.7%</strong></td>
</tr>
</tbody>
</table>
MultiWOZ 5.0 (ThingTalk)

Overall turn-by-turn exact match accuracy = 79%
with few-shot annotations (2% of original)
Conclusion

• **Difference between Machine-2-Machine and WOZ**
  • It is easy to get good results on Machine-2-machine / manual paraphrase
  • WOZ conversations: do not follow the state machine, have greater variety in NL
  • WOZ much closer to real life, but still not real life
• **ThingTalk with dialogue acts** is expressive enough to represent WOZ
  • Agent needs to handle all of ThingTalk, not just synthesizable ones.
• **Sample-efficient training**
  • Synthesis, automatic paraphrase
  • Few shot (for NL for out-of-simulation states)
  • Self-trained (if unannotated date are available)
• **Contextual semantic parser**
  • Formal context eliminates reanalyzing history of dialogue
  • 79% state-by-state exact match accuracy on MultiWOZ
• **Limitations:**
  • State machine is derived from MultiWOZ (multi-domains)
  • Good for transactions