USER SIMULATION FOR DIALOGUE SYSTEMS

Hsien-Chin Lin, 22 Nov 2019
Why do we need a simulated user (SU)?

Dialogue system

- Natural Language Understanding
- Belief Tracking
- Natural Language Generation
- Policy Agent

environment

agent

❤❤❤❤❤ reward
Why do we need a simulated user (SU)?

For training

- RL need lots of interaction to learn the policy
- Learning from real user
  - costly
  - time-consuming
- Learning from data
  - collecting interactable data is not easy
- Learning from SU
Why do we need a simulated user (SU)?

For evaluation

- Human evaluation
  - costly and time-consuming
  - hard to reproduce
- Automatic evaluation
  - success rate, rewards, ...
  - NLG metrics: not consistent with human evaluation
- Evaluating by SU is easy to reproduce, cross-model comparison
Different kinds of user simulation

Summarize SU in different aspects

- **Granularity**
  - Semantic level
  - Natural Language level
    - template, retrieval, generation

- **Methodology**
  - n-gram: Bi-gram, graph model, bayesian model, HMM, ...
  - rule-based: agenda-based
  - data driven: Seq2Seq, inverse RL, adversarial model, ...
Previous studies

non-DL approaches

- N-gram
- Graph based
- Agenda based
Previous studies

N-grams SU (Eckert et al. 1997)

- Bi-gram model $P(a_u | a_m)$
  - only looks on the latest system action
  - cannot produce coherent user behaviour
  - the SU may produce illogical behaviour if the user goal changes
- Look longer history
- incorporate user goal into user state
- HMM (Cuayáhuitl et al. 2005), Baysian model (Pietquin and Dutoit 2009)…
Previous studies

Graph-based SU (Scheffler and Young, 2000)

- All possible paths in a network
- Need extensive domain knowledge
- Not practicable for complex domain

Figure 3: Partial structure for utterance construction in the banking application.
Rule-based SU

Agenda-based approach (Schatzmann et al. 2007)

- user state $S$ is described as an agenda $A$ and a goal $G$

- Example:

  $C_0 = \begin{bmatrix}
  \text{type} = \text{bar} \\
  \text{drinks} = \text{beer} \\
  \text{area} = \text{central}
  \end{bmatrix}

  \quad R_0 = \begin{bmatrix}
  \text{name} = \\
  \text{addr} = \\
  \text{phone} =
  \end{bmatrix}

  $A_1 = \begin{bmatrix}
  \text{inform(type = bar)} \\
  \text{inform(drinks = beer)} \\
  \text{inform(area = central)} \\
  \text{request(name)} \\
  \text{request(addr)} \\
  \text{request(phone)} \\
  \text{bye()}
  \end{bmatrix}

  \quad \text{Sys 0} \quad \text{Hello, how may I help you?}

  \quad \text{Sys 1} \quad \text{Ok, a wine bar. What price range?}

  \quad A_2 = \begin{bmatrix}
  \text{negate(drinks = beer)} \\
  \text{inform(pricerange = cheap)} \\
  \text{inform(area = central)} \\
  \text{request(name)} \\
  \text{request(addr)} \\
  \text{request(phone)} \\
  \text{bye()}
  \end{bmatrix}

  \quad \text{Usr 1} \quad \text{I'm looking for a nice bar serving beer.}

  \quad \text{Usr 2} \quad \text{No, beer please!}

- The probabilities can be learned from corpus or set manually
Summary of these models

These models suffer from...

- Inability to take dialogue history
- Rigid structure to ensure coherent user behavior
- Need lots of labor effort for designing rules
- Domain dependent
Seq2Seq models

- Semantic to Semantic
- Combined agenda-base with seq2seq
- Semantic to Utterence
- Hierarchical seq2seq
- comparison of different settings
uniform select a goal $G = (C, R)$
- $C$: constraints, food-type, price range, ...
- $R$: requests, name, address, ...

context $c_t$ concatenated with
- $a_{m,t}$: recent machine acts
- $inconsist_t$: inconsistency
- $const_t$: constraints status
- $req_t$: requests status

semantic level (El Asri et al., 2016)
### Example of the context vector

<table>
<thead>
<tr>
<th>Machine output / User answer</th>
<th>Machine acts</th>
<th>Inconsistency vector</th>
<th>Constraints status</th>
<th>Requests status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome! How may I help you?</td>
<td>000000000100</td>
<td>000000</td>
<td>001</td>
<td>10110111</td>
</tr>
<tr>
<td></td>
<td>greet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is there a cheap restaurant downtown?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A cheese restaurant. What is your budget?</td>
<td>0000010001</td>
<td>000010</td>
<td>011</td>
<td>10110111</td>
</tr>
<tr>
<td></td>
<td>implicit-confirm, request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No, I said a cheap restaurant.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Panda express is a cheap restaurant downtown.</td>
<td>01000000100</td>
<td>000000</td>
<td>111</td>
<td>10110111</td>
</tr>
<tr>
<td></td>
<td>offer, inform</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>What is the address of this place?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Panda express is located at 108 Queen street.</td>
<td>01000000100</td>
<td>000000</td>
<td>111</td>
<td>10111111</td>
</tr>
<tr>
<td></td>
<td>offer, inform</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Examples of contexts in a dialogue with a restaurant-seeking system. The user goal has two constraints (cheap and downtown) and one request (address).
**Seq2Seq SU**

**Experiment**

- **Dataset:** DSTC2, DSTC3
- **Baseline**
  - Bi-gram, agenda-based
  - Sequence-to-one: outputs a probability distribution over a predefined set of compound acts (size: 54)
- **Measurement**
  - F-score, i.e. $\text{precision} = \frac{\text{# of correctly predicted dialog acts}}{\text{# of predicted dialog acts}}$
Result

- **Average F-score on 50 runs**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Bigram</th>
<th>Agenda-based</th>
<th>Sequence-to-one</th>
<th>Sequence-to-sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSTC2 Validation</td>
<td>0.20</td>
<td>0.24</td>
<td>0.37</td>
<td>0.34</td>
</tr>
<tr>
<td>DSTC2 Test</td>
<td>0.09</td>
<td>0.18</td>
<td>0.29</td>
<td>0.27</td>
</tr>
<tr>
<td>DSTC3 Test</td>
<td>—</td>
<td>0.13</td>
<td>0.19</td>
<td>0.18</td>
</tr>
</tbody>
</table>

- The Seq2One is slightly better than Seq2Seq because it’s an easier task
- The Seq2Seq has better scalability (the number of possible acts might grow)
- The recall is relatively low on larger actions space (54 in DSTC2, 94 in DSTC3)
Use the agenda-based model for planning
If the dialog act can be found in templates then use templates
Else use Seq2Seq model for NLG
Semantic to Utterance (Kreyssig et al. 2018)

- System structure
  - The setting of Goal Generator and Feature Extractor is like (El Asri et al., 2016)
  - The input sequence is Feature History
  - The output sequence is User Utterance
**Seq2Seq SU**

**Generate non-deterministic result**

- Beam-search is often used to generate a sequence by RNNs
- Taking $n$ beams with the highest probability $P(w_t w_{t-1} \ldots w_0 | p)$

- Sample $n$ words per beam from the probability distribution
The policy trained with NUS can perform well on both SUs

Overfitting: the policy performing best on the NUS was not the one on the ABUS

<table>
<thead>
<tr>
<th></th>
<th>Train. Sim.</th>
<th>Eval. Sim.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NUS</td>
<td>ABUS</td>
</tr>
<tr>
<td>NUS-best</td>
<td>13.0</td>
<td>98.0 (^{N_1})</td>
</tr>
<tr>
<td>ABUS-best</td>
<td>1.53</td>
<td>71.5 (^{A_1})</td>
</tr>
<tr>
<td>NUS-avg</td>
<td>12.4</td>
<td>96.6</td>
</tr>
<tr>
<td>ABUS-avg</td>
<td>-7.6</td>
<td>45.5</td>
</tr>
</tbody>
</table>

Table 2: Results for policies trained for 4000 dialogues on NUS and ABUS when tested on both USs for 1000 dialogues. Five policies with different initialisations were trained for each US. Both average and best results are shown.
Experiments – Cross-Model Evaluation

- In five seeds for NUS, the performance is all better with less data
- This behavior was not observed for the policies trained with the ABUS

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NUS</td>
<td>ABUS</td>
<td></td>
</tr>
<tr>
<td>NUS-best</td>
<td>13.0</td>
<td>98.0(^N_1)</td>
<td>13.3</td>
</tr>
<tr>
<td>ABUS-best</td>
<td>1.53</td>
<td>71.5(^A_1)</td>
<td>13.8</td>
</tr>
<tr>
<td>NUS-avg</td>
<td>12.4</td>
<td>96.6</td>
<td>11.2</td>
</tr>
<tr>
<td>ABUS-avg</td>
<td>-7.6</td>
<td>45.5</td>
<td>13.5</td>
</tr>
</tbody>
</table>

Table 2: Results for policies trained for 4000 dialogues.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NUS</td>
<td>ABUS</td>
<td></td>
</tr>
<tr>
<td>NUS-best</td>
<td>12.2</td>
<td>95.9</td>
<td>13.9</td>
</tr>
<tr>
<td>ABUS-best</td>
<td>-4.0</td>
<td>54.8</td>
<td>13.2</td>
</tr>
<tr>
<td>NUS-avg</td>
<td>12.0</td>
<td>95.4</td>
<td>12.2</td>
</tr>
<tr>
<td>ABUS-avg</td>
<td>-9.48</td>
<td>42.3</td>
<td>12.8</td>
</tr>
</tbody>
</table>

Table 3: As Table 2 but trained for 1000 dialogues.
Experiments – human Evaluation

- The NUS performs better
- The overfitting is also observed, the best performing policy was the policy that performed best on the other US

<table>
<thead>
<tr>
<th>Training Simulator</th>
<th>Human Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rew.</td>
</tr>
<tr>
<td>NUS - $\mathcal{N}_1$</td>
<td>13.4</td>
</tr>
<tr>
<td>NUS - $\mathcal{N}_2$</td>
<td>13.8</td>
</tr>
<tr>
<td>ABUS - $\mathcal{A}_1$</td>
<td>13.3</td>
</tr>
<tr>
<td>ABUS - $\mathcal{A}_2$</td>
<td>13.1</td>
</tr>
</tbody>
</table>

Table 4: Real User Evaluation. Results over 250 dialogues with human users. $\mathcal{N}_1$ and $\mathcal{A}_1$ performed best on the NUS. $\mathcal{N}_2$ and $\mathcal{A}_2$ performed best on the ABUS. Rewards are not comparable to Table 2 and 3 since all user goals were achievable.
Seq2Seq SU

Discussion

- Less labelling for generate natural language compared with semantic response
- NUS excelled on both evaluation tasks
Hierarchical User Simulator (HUS) (Gür et al. 2018)

- An end-to-end hierarchichal seq2seq approach
- Without any feature extraction and external state tracking annotations
- Encode user goal: $h^C = Enc(e^C; \theta_C)$
- Encode system turn: $h^S_i = Enc(e^{S_i}; \theta_S)$
- Encode dialogue history
  
  $h^D_0 = h^C$
  $h^D_i = Enc(\{h^S_i\}_{i=1}^T; \theta_D)$
- $L_{crossent}$: cross-entropy error between candidate and correct user sequence

Fig. 2: HUS model: Boxes are RNN cells, colors indicate parameter sharing.
The output of HUS is deterministic

Add a Gaussian distribution generator

Sample $z_x \sim N(z | \mu_x, \Sigma_x)$

$$\mu_x = W_\mu h^D_{t-1} + b_\mu$$

$$\Sigma_x = W_\Sigma h^D_{t-1} + b_\Sigma$$

The decoder will be initialized with $\hat{h}_t^D = FC([h_t^D ; z_x])$

KL divergence between prior and posterior distribution

$$L_{var} = \alpha KL(N(z | \mu_x, \Sigma_x) | N(z | \mu_y, \Sigma_y))$$

in order to make sure the behavior will be consistent
Goal Regularization (VHUSReg)

- Generating long dialogues when user turns diverge from the initial user goal
- Initialize the history encoder with zero, then \( \hat{h}_t^D = FC([h_t^D; h^c]) \)
- Minimize the divergence between user goal and user turn token

\[
L_{reg} = ||b_t^u - BOW(C)|| + ||b_t^D - BOW(U_t)|| + ||b_t^S - BOW(S_t)||
\]
Seq2Seq SU

Experiment results

- SL
  - Supervised end-to-end policy
  - Map user utterence to system actions

- RL policy outperformed SL
  - Especially on EM, the SL may stuck in local minima and cannot recover some of the slot-value pairs

- RL is more robust, even with weaker SU

<table>
<thead>
<tr>
<th></th>
<th>SL</th>
<th>RL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact Match (%)</td>
<td>75.67</td>
<td>94.69</td>
</tr>
<tr>
<td></td>
<td>94.69</td>
<td>98.27</td>
</tr>
<tr>
<td>Partial Match (%)</td>
<td>94.3</td>
<td>96.15</td>
</tr>
<tr>
<td></td>
<td>96.51</td>
<td>9.615</td>
</tr>
<tr>
<td>Dialogue Length</td>
<td>12.03</td>
<td>7.45</td>
</tr>
<tr>
<td></td>
<td>7.076</td>
<td></td>
</tr>
</tbody>
</table>

+ dialogue length

<table>
<thead>
<tr>
<th></th>
<th>SL</th>
<th>RL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact Match (%)</td>
<td>86.1</td>
<td>94.33</td>
</tr>
<tr>
<td></td>
<td>94.33</td>
<td>98.2</td>
</tr>
<tr>
<td>Partial Match (%)</td>
<td>95.69</td>
<td>98.2</td>
</tr>
<tr>
<td></td>
<td>96.51</td>
<td>7.076</td>
</tr>
<tr>
<td>Dialogue Length</td>
<td>11.8005</td>
<td>7.803</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>SL</th>
<th>RL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact Match (%)</td>
<td>82.52</td>
<td>95.53</td>
</tr>
<tr>
<td></td>
<td>95.53</td>
<td>98.43</td>
</tr>
<tr>
<td>Partial Match (%)</td>
<td>98.2</td>
<td>7.92</td>
</tr>
<tr>
<td></td>
<td>97.08</td>
<td></td>
</tr>
<tr>
<td>Dialogue Length</td>
<td>6.878</td>
<td>7.92</td>
</tr>
</tbody>
</table>

HUSReg

<table>
<thead>
<tr>
<th></th>
<th>SL</th>
<th>RL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact Match (%)</td>
<td>96.19</td>
<td>98.8</td>
</tr>
<tr>
<td></td>
<td>96.19</td>
<td>98.56</td>
</tr>
<tr>
<td>Partial Match (%)</td>
<td>98.56</td>
<td>98.8</td>
</tr>
<tr>
<td></td>
<td>97.08</td>
<td>6.878</td>
</tr>
<tr>
<td>Dialogue Length</td>
<td>8.0555</td>
<td>6.905</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>SL</th>
<th>RL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact Match (%)</td>
<td>91.90</td>
<td>95.98</td>
</tr>
<tr>
<td></td>
<td>91.90</td>
<td>98.52</td>
</tr>
<tr>
<td>Partial Match (%)</td>
<td>97.67</td>
<td>98.52</td>
</tr>
<tr>
<td></td>
<td>97.08</td>
<td>6.905</td>
</tr>
</tbody>
</table>
Seq2Seq SU

Human evaluation

- The dialogue is transferred to natural language by template
- All SUs get better score and less standard deviation

<table>
<thead>
<tr>
<th>Model</th>
<th>Average Score (Standard Deviation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agenda-based</td>
<td>4.56 (0.859)</td>
</tr>
<tr>
<td>HUS+dialogue length</td>
<td>4.86 (0.545)</td>
</tr>
<tr>
<td>VHUS</td>
<td>4.88 (0.472)</td>
</tr>
<tr>
<td>HUSReg</td>
<td>4.88 (0.452)</td>
</tr>
<tr>
<td>VHUSReg</td>
<td>4.83 (0.594)</td>
</tr>
</tbody>
</table>
Comparison between different settings (Shi et al. 2019)

- Compare different settings
  - Policy: agenda-based and model-based
  - NLG: template, retrieval, and generation
  - Evaluation: direct and indirect
Seq2Seq SU

Automatic direct evaluation

- Use perplexity, vocabulary size and utterance length to measure NLG quality
- Retrieval-based models have the largest Vocab
- Retrieval-based model can generate the longest sentences, but End-to-End model is also doing good
- Although the PPL is the largest for retrieval-based models, it also has the biggest Vocab and longest utterence length

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Agenda-Template (AgenT)</td>
<td>SL</td>
<td>Agenda</td>
<td>Template</td>
<td>10.32</td>
<td>180</td>
<td>9.65</td>
<td>4.07</td>
<td>4.56</td>
<td>4.88</td>
<td>2.4</td>
<td>4.50</td>
</tr>
<tr>
<td>Agenda-Retrieval (AgenR)</td>
<td>SL</td>
<td>Agenda</td>
<td>Retrieval</td>
<td>33.90</td>
<td>383</td>
<td>11.61</td>
<td>3.50</td>
<td>4.22</td>
<td>4.58</td>
<td>3.9</td>
<td>3.74</td>
</tr>
<tr>
<td>Agenda-Generation (AgenG)</td>
<td>SL</td>
<td>Agenda</td>
<td>Generation</td>
<td>7.49</td>
<td>159</td>
<td>8.07</td>
<td>3.32</td>
<td>3.92</td>
<td>4.64</td>
<td>2.5</td>
<td>3.36</td>
</tr>
<tr>
<td>SL-Template (SLT)</td>
<td>SL</td>
<td>Template</td>
<td></td>
<td>9.32</td>
<td>192</td>
<td>9.83</td>
<td>4.80</td>
<td>4.80</td>
<td>4.98</td>
<td>2.6</td>
<td>4.74</td>
</tr>
<tr>
<td>SL-Retrieval (SLR)</td>
<td>SL</td>
<td>Retrieval</td>
<td></td>
<td>29.36</td>
<td>346</td>
<td>11.06</td>
<td>4.40</td>
<td>3.99</td>
<td>4.88</td>
<td>4.3</td>
<td>4.01</td>
</tr>
<tr>
<td>SL-End2End (SLE)</td>
<td>End-to-End</td>
<td></td>
<td></td>
<td>13.47</td>
<td>205</td>
<td>10.95</td>
<td>3.32</td>
<td>2.62</td>
<td>3.18</td>
<td>2.7</td>
<td>2.64</td>
</tr>
</tbody>
</table>
Human direct evaluation

- Fluency: Templates. They are written by human
- Coherence: Agenda-based in general better than model-based
- Goal adherence: Infusing the goal is more difficult for End2End.
- Diversity: Retrieval-based is good at diversity but is not as good in fluency
  Template-based outperformed on fluency but suffer from diversity
  Generation-based suffer from generic responses
Automatic indirect evaluation

- Model-based converge faster. Capture the major path instead of exploring all the possible paths.
- Retrieval-based converged slower because of larger vocabulary size.
The system can handle more language variations will do better on Solved ratio

The efficiency doesn’t always correlated to the dialog length (AgenG and SLE)

The satisfaction is not only related to solved ration but also efficiency and latency

Naturalness is related to solved ratio (overall performance)
Cross model evaluation

- Agenda-based with retrieval-based NLG has the best performance. This result agrees with the human evaluation.
- More type of SU will give better quality of evaluation. User SLT prefers SLT (0.975) than AgenG (0.965), but in overall AgenG is better.
- The diagonal is usually the highest. RL policy is not general over all kind of users.

<table>
<thead>
<tr>
<th>Usr\Sys</th>
<th>Sys-AgenT</th>
<th>Sys-AgenR</th>
<th>Sys-AgenG</th>
<th>Sys-SLT</th>
<th>Sys-SLR</th>
<th>Sys-SLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AgenT</td>
<td>0.975</td>
<td>0.960</td>
<td>0.790</td>
<td>0.305</td>
<td>0.300</td>
<td>0.200</td>
</tr>
<tr>
<td>AgenR</td>
<td>0.540</td>
<td>0.900</td>
<td>0.785</td>
<td>0.230</td>
<td>0.230</td>
<td>0.235</td>
</tr>
<tr>
<td>AgenG</td>
<td>0.725</td>
<td>0.975</td>
<td>0.950</td>
<td>0.355</td>
<td>0.300</td>
<td>0.20</td>
</tr>
<tr>
<td>SLT</td>
<td>0.985</td>
<td>0.985</td>
<td>0.985</td>
<td>0.990</td>
<td>0.965</td>
<td>0.730</td>
</tr>
<tr>
<td>SLR</td>
<td>0.925</td>
<td>0.975</td>
<td>0.965</td>
<td>0.975</td>
<td>0.935</td>
<td>0.630</td>
</tr>
<tr>
<td>SLE</td>
<td>0.770</td>
<td>0.820</td>
<td>0.815</td>
<td>0.840</td>
<td>0.705</td>
<td>0.770</td>
</tr>
<tr>
<td>Average</td>
<td>0.820</td>
<td><strong>0.935</strong></td>
<td><strong>0.882</strong></td>
<td><strong>0.616</strong></td>
<td>0.573</td>
<td>0.461</td>
</tr>
</tbody>
</table>
Discussion

- Model-based perform relatively worse
- Model-based doesn’t explore all possible paths (Act6)

Figure 5: Dialog act distribution comparison. Act1 to Act7 corresponds to the seven user dialog acts, “inform restaurant type”, “inform restaurant type change”, “ask info”, “make reservation”, “make reservation change time”, “anything else”, and “goodbye”
Summary

- The generating model may suffer from generating generic results
- We can get better policy with more diverse output SU
- The policy of SU need to explore all possibilities
Inverse RL

Inverse RL (Chardramohan et al., 2011)

- The SU can be view as an MDP \( \{S, A, P, \gamma\}/R \)
- Reward function \( R_\theta(s, a) = \theta^T \phi(s, a) = \sum_{i=1}^{k} \theta_i \phi_i(s, a) \)
- Q-function \( Q^\pi(s, a) = E\left[ \sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, a_0 = a \right] \)
- \( Q^\pi(s, a) = E\left[ \sum_{i=0}^{\infty} \gamma^i \theta^T \phi(s, a) | s_0 = s, a_0 = a \right] = \theta^T \mu^\pi(s, a) \)
- \( \mu^\pi(s, a) \) feature expectation can be model as the discounted measure of features according to system visitation frequency, given \( m \) trajectories (\( H^i \) is the length of the \( i^{th} \) trajectory), \( \mu^\pi(s, a) \) can be modeled as:

\[
\mu^\pi(s, a) = \frac{1}{m} \sum_{i=0}^{m} \sum_{t=0}^{H_i} \gamma^i \phi(s_t^i, a_t^i)
\]
Algorithm 1 User simulation using imitation learning

1: Compute $\mu_{\text{expert}}$ from dialogue corpus
2: Initiate $\Pi$ with random policy $\pi_{\text{predict}} = \pi_0$ and compute $\mu_{\text{predict}}$
3: Compute $t$ and $\theta$ such that

$$t = \max_\theta \{ \min_{\pi_{\text{predict}} \in \Pi} \theta^T (\mu_{\text{expert}} - \mu_{\text{predict}}) \} \text{ s.t. } ||\theta||^2 \leq 1$$

4: **if** $t \leq \xi$ **then** Terminate
5: **end if**
6: Train a new policy $\pi_{\text{predict}}$ for userMDP optimizing $R = \theta^T \phi(s, a)$ with RL (Lspi).
7: Compute $\mu_{\text{predict}}$ for $\pi_{\text{predict}}$; $\Pi \leftarrow \pi_{\text{predict}}$
   Goto to step 3.
Summary

- We can train a MDP SU from a fix corpus
- In the paper, they only conducted a simple experiment
- The cost of computing is a lot. (RL in the inner-loop)
Collaboration-based (Didericksen et al. 2017)

- Collaboration-based SU utilizes the similarity between different users to predict the user’s next action
- Label propagation: train a simple classification model on a part of the data to label the entire dataset
- Easy to incorporate external knowledge, e.g. user profile to pre-filter the act candidates
- Can be run very fast
Build a Conversational Agent Overnight (Shah et al. 2018)

- Build a dialogue system by M2M and crowdsourcing
- Collect data by Wizard-of-Oz setup may suffer from
  - Not cover all the interactions
  - Unfitting dialogues (too simplistic or too convoluted)
  - Need more efforts to filter errors
Generating outline via self-play

- Outlines are easier to generate
- Don’t need to generate complex and diverse language
Conclusion

The rule-based methods

- More controllable
- Generate all possible paths
  - Domain-dependent
  - Not scalable
  - Labor-consuming

The model-based methods

- Learn user behaviour from corpus
- Less labor effort
- Adapt to new domain easier
  - Focus on main paths, not all
  - Incoherence goal
What’s next?

- Generate more various outputs and more human-like behaviour
- Persona for SU
- Error models: ASR, ambiguity, ...
- How to use IRL, adversarial training for SU?
- Self-training via Machine-to-machine interaction
Reference

- User modeling for spoken dialogue system evaluation
  Eckert, Wieland, Esther Levin, and Roberto Pieraccini, 1997

- HUMAN-COMPUTER DIALOGUE SIMULATION USING HIDDEN MARKOV MODELS
  Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon and Hiroshi Shimodaira. 2005

- Training Bayesian networks for realistic man-machine spoken dialogue simulation
  Olivier Pietquin, Stéphane Rossignol, and Michel Ianotto, 2009

- Probabilistic simulation of human-machine dialogues
  Scheffler, Konrad, and Steve Young, 2000

- Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System
  Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye and Steve Young, 2007

- A Sequence-to-Sequence Model for User Simulation in Spoken Dialogue Systems
  Layla El Asri, Jing He, Kaheer Suleman, 2016
Reference

- **A User Simulator for Task-Completion Dialogues**
  Xiujun Li, Zachary C. Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, Yun-Nung Chen, 2017

- **Neural User Simulation for Corpus-based Policy Optimisation for Spoken Dialogue Systems**
  Kreyssig F, Casanueva I, Budzianowski P, Gašić M, 2018

- **USER MODELING FOR TASK ORIENTED DIALOGUES**
  Izzeddin Gur, Dilek Hakkani-Tur, Gokhan Tur, Pararth Shah, 2018

- **How to Build User Simulators to Train RL-based Dialog Systems**
  Weiyan Shi, Kun Qian, Xuewei Wang, Zhou Yu, 2019

- **User Simulation in Dialogue Systems using Inverse Reinforcement Learning**
  Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, Olivier Pietquin, 2011

- **Collaboration-based User Simulation for Goal-oriented Dialog Systems**
  Devin Didericksen, Oleg Rokhlenko, Kevin Small, Li Zhou, Jared Kramer, 2017
Building a Conversational Agent Overnight with Dialogue Self-Play
Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, Larry Heck, 2018