Learning Dialogue Strategies with a Simulated User

Jost Schatzmann and Steve Young

Cambridge University Engineering Department
Trumpington Street, Cambridge CB21PZ, UK
{js532, sjy}@eng.cam.ac.uk

Dialog on Dialogs Meeting
Carnegie Mellon University, 19 August 2005
User Simulation-Based Learning

- Learn dialogue strategies through trial-and-error interaction with a simulated user.

Diagram:
- Simulated User
- Dialogue Manager
- Strategy
- Reinforcement Learning
- Supervised Learning
- Dialogue Corpus
Agenda

- Work on Evaluation: Experiments and Results
- Agenda-based User Modelling
Research Questions

- How good are the currently available simulation techniques? Can they...
  - produce human-like behaviour?
  - cover the variety of real user behaviour?

- What is the effect of the user model on the learned strategy?
  - Influence on strategy performance?
  - Influence on strategy characteristics?
  - Are the strategies merely fitted to a particular UM?
  - Can we find UM-independent forms of strategy evaluation?

SIGdial paper

ASRU paper
User Modelling Techniques

State of the art in intention-level modelling:

- **Bigram model:** $p(a_u|a_s)$
- **Levin model:** $p(\text{yes_answer}|\text{expl_conf})$
- **Pietquin model:** $p(\text{yes_answer}|\text{expl_conf}, \text{goal})$

UMs typically not trained on real data

Standard evaluation practice is to test learned strategy on the user model used for learning
Experiments

- **Training**
  - Corpus
  - Bigram Model
  - Levin Model
  - Pietquin Model

- **Learning**
  - DM
  - Bigram Model
    - Bigram Strategy
  - Levin Model
    - Levin Strategy
  - Pietquin Model
    - Pietquin Strategy

- **Testing**
  - Bigram Model
  - Levin Model
  - Pietquin Model
Comparative Evaluation

- Performance of the learned strategy depends on the quality of the UM.

![Bar chart showing performance comparison between real dialogue data and simulated dialogues]

- Real dialogue data
- Simulated dialogues
Strategies learned with a poor UM can fail when tested on a better UM.
Learned strategies exploit weaknesses in UMs
UM-independent Evaluation

- Techniques for evaluating new strategies on real dialogue data would be helpful

![Graph showing similarity to learned strategy against dialogue reward](image)
Agenda

- Work on Evaluation: Experiments and Results
- Agenda-based User Modelling
Motivation

- Currently have drastically different levels of sophistication for DM and UM

- Fail to model context which extends beyond the previous dialogue turn

User: I want to go from Boston to London.
System: Going from Austin to London. And when do you want to fly?
User: No, from Boston to London.
System: From Boston to London, is that correct?
User: Yes. And I’m flying on March 15th.
**Agenda-based User Model**

- **Idea**: MDP User Model with agenda-based state representation

  - Combines user state and user goal representation
  - Naturally encodes dialogue history
  - Allows delayed user responses (priority of actions)
Agenda-based User Model

- Assume cooperative user behaviour to label dialogues
- Learn output probabilities to model user behaviour

- Potential scope for modelling uncertainty about true state of user agenda (‘Hidden Agendas’)
Summary

- Current lack of solid user models and reliable evaluation standards is a major roadblock to simulation-based strategy learning.

- Work on agenda-based user models may help to enhance our model of the user state and improve simulation quality.
Thank you!

js532@cam.ac.uk
http://mi.eng.cam.ac.uk/~js532/


The rest of this slide deck is only a backup for further questions.
Need to deviate from known strategies to explore new and potentially better ones

System: Where are you flying from, where are you flying to, on what date are you flying, when is your preferred time, do you have a preferred airline and would you like a window-seat?

Sim. User: Flying from Boston to London on March 15 at 9am with Delta Airlines. Window seat please.

Real User: ??????
Strategy Confidence Scores (2/3)

- **Idea:** System designer needs a confidence measure indicating how reliable the learned strategy is

- Define strategy confidence as function of the likelihood of the user response in the given context

\[
\text{conf}(\pi) = \frac{1}{N} \sum_{i=0}^{N} \frac{1}{N_i} \sum_{t=0}^{N_i} \text{conf}(a_{u,t,i}, a_{s,t,i}, s_{t,i})
\]

\[
\text{conf}(a_u, a_s, s) = p(a_u | a_s) p(a_s | s)
\]
Reliability score can be integrated into the learning process by weighting the reward. Good strategy with respect to performance and reliability.

Spectrum of strategies with increasing reliability.
**Idea:** Produce acoustic-level output and optimize strategy for system-specific error conditions

- Combine high-level user model with statistical NLG and TTS for strategy learning
- No need to simulate recognition and understanding errors
User Studies (1/1)

- Evaluate performance of new user models using real users
  - Test simulation quality using listening tests
  - Test strategy performance using questionnaires
  - Test usefulness of reliability scores
Summary

- Work on Evaluation (January to July 2005)
  - Experiments and Results

- Project Proposals (Summer 2005 to Summer 2007)
  - Introduction of strategy confidence scores
  - Agenda-based User Models
  - Strategy learning under system-specific error conditions
  - User studies
Experiments

- Implemented a handcrafted DM and trained three different UMs
  - Bigram model: \( p(a_u|a_s) \)
  - Levin model: \( p(\text{yes_answer}|\text{expl_conf}) \)
  - Pietquin model: \( p(\text{yes_answer}|\text{expl_conf}, \text{goal}) \)

- Implemented Q-Learning DM, learned strategies with each UM and compared performance and characteristics
- Cross-model evaluation of strategies
- Investigated user-model independent techniques for testing learned strategies
Phase II: New User Models (2/3)

- **Idea 2**: Use clustering to construct networks of user behaviour.
- **Motivation**: Networks are well-suited for encoding dialogue context, but their manual construction is expensive.
- Represent each dialogue as follows:

```
SD1 -> ASI -> SU -> AU -> SD2
```

- We want to cluster user states, but the user state can never be fully observed or captured.
- However, we can cluster user actions and assume that similar actions imply similar contexts.

```
SD1 -> ASI -> C1 -> AU -> SD2
```
**Phase II: New User Models (3/3)**

- **Idea 2, contd.:** Overlay all dialogue sequences to obtain a network
- **Use frequency counts to obtain transition probabilities**

---

**Clustered User States** represent common nodes in the network – similar to “choice points” used by Scheffler and Young.

**Transition probabilities can be derived from number of dialogues that contain each path**

**System dialogue states are also clustered to pool training data**

---

Diagram:

- $s_{d1}$
- $s_{d2}$
- $c_1$
- $s_{d3}$
- $s_{d4}$

Transitions:

- $a_{si}$
- $a_{sj}$
- $a_{sh}$
- $a_u$
Backup Slides for Sigdial paper
Evaluation must cover two aspects

- Can the model produce human-like behaviour?
  - Does it produce user responses that a real user might have given in the same dialogue context?

1. Need to compare real and simulated user responses!

- Can the model reproduce the variety of human behaviour?
  - Does it represent the whole user population?

2. Need to compare real and simulated dialogue corpora!
- Split the corpus into training and testing data

- Evaluate how well the model can predict the user responses in the test data
  - Feed in all information about dialogue history and user goal
  - Compare simulated user turn and real user turn
  - Use Precision and Recall to measure how closely the predicted turn matches the real user turn
Use of Precision and Recall

Evaluate turn by turn:

Dialogue in the test set:

Sys: greeting
    instructions
    request_info orig_city

Usr: unknown
    provide_info orig_city london

Sys: implicit_conf orig_city london
    request_info dest_city

Usr: no_answer
    provide_info orig_city boston

Simulated user responses:

P=100%, R=50%

Usr: provide_info orig_city london

P=0%, R=0%

Usr: yes_answer
    provide_info dest_city paris

- P = Correctly predicted actions / All predicted actions
- R = Correctly predicted actions / All actions in real response
Results: Precision and Recall

- Precision and Recall

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigram</td>
<td>17.83</td>
<td>21.66</td>
</tr>
<tr>
<td>Levin</td>
<td>37.98</td>
<td>31.57</td>
</tr>
<tr>
<td>Pietquin</td>
<td>40.16</td>
<td>33.38</td>
</tr>
</tbody>
</table>

- What do the results mean?
- Is this analysis sufficient?
Simulated vs. real corpora

- We need to evaluate if the model can reproduce the variety of user behaviour in the training data
  - Generate a whole corpus through interaction between the sim. user and the DM
  - Use statistical metrics to compare the simulated corpus to the real one
Statistical metrics

- High-level dialogue features
  - Dialogue length (in number of turns)
  - Turn length (in number of actions)
  - Proportion of user vs system talk

- Dialogue Style and Cooperativeness
  - Frequency of different user and system speech acts (average number of occurrences per dialogue)
  - Proportion of goal-directed actions vs. Grounding actions vs dialogue formalities vs. Unrecognised actions
  - Number of times information is requested, provided, re-requested, re-provided

- Dialogue Success and Efficiency
  - Average goal / subgoal achievement rate
  - Goal completion time
Results: Goal completion rates / times

- Goal completion rates and times
Project overview

- **Phase I**
  - Evaluation of the current state of the art
  - Re-assessment of standard evaluation practices
  - Introduction of strategy confidence scores

- **Phase II**
  - Development of new user models
  - Separation of user and error model

- **Phase III**
  - Acoustic-level simulation
  - Strategy learning under system-specific error conditions

- **Phase IV**
  - User studies

Work completed
Simple statistical metrics can distinguish simulated from real dialogue data.
Motivation

- Lack of a solid user model is currently a major roadblock to automatic DM design

- Lack of rigorous evaluation standards has led to uncertainty about the validity of simulation–based learning

- Goal is to develop user and error modelling techniques that enable us to learn strategies which outperform competing handcrafted strategies when tested on human users
Backup slides for simulation techniques
- Bigram model: $p(a_u|a_s)$
- Levin model: $p(\text{yes\_answer}|\text{expl\_conf})$
- Pietquin model: $p(\text{yes\_answer}|\text{expl\_conf, goal})$
Overview of simulation techniques

- User simulation for strategy learning is a young field of research:
  - Lin and Lee (2000)

- Closely related work on user simulation for SDS evaluation:
  - Lopez-Cozar et al. (2003)
  - Araki et al. (1997, 1998)

- Simulation on intention- rather than word- or acoustic level

- N-gram model for predicting the next user intention
  \[ \hat{u}_t = \text{arg max} \ P(u_t|s_t) \]

- Simulated user responses often unrealistic and inconsistent

  System: What is your departure city?
  User: New York
  System: What is your destination?
  User: New York
Different approaches to user simulation

- Probabilistic:
  - Araki et al. (1997, 1998)
  - Lin and Lee (2000, 2001)
  - Lopez-Cozar et al. (2003)

- Deterministic:
  - PR
  - L00
  - L97

- Hand-crafted:
  - A
  - LL

- Data-driven:
  - SY
  - LC
  - LL
Levin, Pieraccini, Eckert (2000)

- Attempt to account for weaknesses of the n-gram model
- Assume a simple dialogue model and hand-select appropriate probabilities for predicting user responses

![Diagram showing the flow of dialogue and probabilities](image)

- $P(n|\text{greeting})$
- $P(\text{provide DATE} \mid \text{constrain DESTINATION})$
- $P(\text{yes} \mid \text{relax AIRLINE})$

User responses still not goal-consistent!
Scheffler and Young (1999 - 2002)

- User model includes user goal and user’s beliefs on current system status

<table>
<thead>
<tr>
<th>Goal field</th>
<th>Value</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>GET_FILM_LIST</td>
<td>Specified</td>
</tr>
<tr>
<td>Film</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Cinema</td>
<td>ARTS_PICT_HOUSE</td>
<td>Pending</td>
</tr>
<tr>
<td>Day</td>
<td>TODAY</td>
<td>Pending</td>
</tr>
</tbody>
</table>

- User acts according to the given goal until it is completed
- Frequencies of different goals are estimated from corpus
Utterance generation lattices, obtained by analysing possible dialogue path in existing prototype system.

Heavily task-dependent approach!

- Pietquin combines ideas from Scheffler’s and Levin’s work
- Probabilities are conditioned on user’s goal and memory

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Priority</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESSOR</td>
<td>Pentium</td>
<td>High</td>
<td>0</td>
</tr>
<tr>
<td>SPEED</td>
<td>800</td>
<td>High</td>
<td>0</td>
</tr>
<tr>
<td>RAM</td>
<td>256</td>
<td>Low</td>
<td>0</td>
</tr>
<tr>
<td>HDD</td>
<td>60</td>
<td>Low</td>
<td>0</td>
</tr>
</tbody>
</table>

- $P(n|\text{greeting,goal})$
- $P(\text{provide RAM | constrain HDD, goal, memory})$
- $P(\text{yes | relax RAM, goal})$
- $P(\text{close | asked for SPEED, goal, memory})$
Graph-based DM
I want to fly to London!

And when do you want to go?

prov_info
dest_city
london

requ_info
depart_date
Dialogue as a Markov Decision Process

- Describe dialogue in terms of states and actions
- View DM strategy as a mapping from states to actions

Dialogue State:
- orig_city confirmed
- dest_city known
- depart_date unknown
- depart_time unknown

System Action:
- <impl_conf, dest_city, london>
- <requ_info, depart_date>
Reinforcement Learning (1/2)

- Learning DM explores its environment through trial-and-error and receives a reward $r_t$ at each time $t$.

- Aim is to maximise to the cumulative discounted reward over time $r_{(t)}$

$$ r_{(t)} = \gamma^{t+1} r_{t+2} + \gamma^{t+2} r_{t+3} + \ldots $$
Reinforcement Learning (2/2)

- Estimate value of taking action \( a \) in state \( s \)

\[
Q^\pi(s, a) = E_\pi(r(t) \mid s_t = s, a_t = a)
\]

- Define the optimal policy \( \pi^* \)

\[
\pi^*(s) = \arg \max_a Q^*(s, a)
\]
The Q-learning update rule

\[ Q(s, a) := (1 - \alpha)Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a')) \]