Structured Probabilistic Modelling for Dialogue Management

Pierre Lison
Language Technology Group
University of Oslo

ESSLLI 2015, Barcelona
• The dialogue management task

• A hybrid logical/probabilistic approach
  • Probabilistic rules
  • Parameter estimation
  • Experiments

• Demonstration of the OpenDial toolkit
Outline for this talk

- The dialogue management task
- A hybrid logical/probabilistic approach
  - Probabilistic rules
  - Parameter estimation
- Experiments
- Demonstration of the OpenDial toolkit
What is dialogue management?

• A component in (spoken) dialogue systems

• In charge of "managing" the interaction
  • Maintain a representation of the current state of the dialogue
  • Select the next system actions based on this state
  • Predict how the interaction is going to unfold

• Difficult problem!
  • Dialogue is complex (many contextual factors to capture)
  • Dialogue is uncertain (ambiguities, unexpected events, etc.)
Typical dialogue architecture

Extra-linguistic environment

- Language understanding
- Recognition hypotheses
- Speech recognition

Interpreted dialogue acts

- State update
- Action selection

System responses

Generation

- Utterances to synthesise
- Speech synthesis

User

Dialogue state

Dialogue management

input speech signal (user utterance)

output speech signal (machine utterance)
Existing DM techniques

<table>
<thead>
<tr>
<th>Logical approaches</th>
<th>Statistical approaches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fine-grained control of conversation</td>
<td>Robust, data-driven models of dialogue</td>
</tr>
<tr>
<td>Limited account for uncertainties</td>
<td>Need large quantities of training data</td>
</tr>
</tbody>
</table>

A new hybrid modelling framework based on *probabilistic rules*
Outline for this talk

- What is dialogue management?

- A hybrid logical/probabilistic approach
  - Probabilistic rules
  - Parameter estimation
  - Experiments

- Demonstration of the OpenDial toolkit
The key idea

• We start with the core ideas behind probabilistic dialogue modelling:
  
  • Dialogue state represented as a Bayesian Network
  
  • Each variable captures a *relevant aspect of the interaction* (dialogue history, user intentions, external context, etc.)
  
  • The dialogue state is regularly updated with *new observations* (spoken inputs, new events), according to domain-specific probabilistic models
  
  • ... and used to determine the *next actions* to execute, according to domain-specific utility models
The key idea

• instead of expressing the domain models using traditional formats (e.g. probability tables)...

• ... we adopt a high-level representation based on probabilistic rules.

• The probabilistic rules provide an *abstraction layer* on top of probabilistic (graphical) models

But:

• Less parameters to estimate (=easier to learn from small amounts of data)

• Can express expert knowledge in human-readable form
Rule structure

• Basic skeleton: if... then...else construction:

∀ x, if (condition₁ holds) then ...
... else if (condition₂ holds) then ...
... else ...

• Mapping between conditions and (probabilistic) effects

• Can use logical operators and universal quantifiers

• Two types of rules: probability and utility rules
Two types of rules

**Probability rules**

*Conditional probability distributions* between state variables

**General structure:**

\[
\text{if (condition}_1\text{) then} \\
\text{P(effect}_1\text{)} = \theta_1, \\
\text{P(effect}_2\text{)} = \theta_2, \ldots \\
\text{else if (condition}_2\text{) then} \\
\text{P(effect}_3\text{)} = \theta_3, \ldots \\
\ldots
\]

**Utility rules**

*Utility functions for system actions given state variables*

**General structure:**

\[
\text{if (condition}_1\text{) then} \\
\text{U(action}_1\text{)} = \theta_1, \\
\text{U(action}_2\text{)} = \theta_2, \ldots \\
\text{else if (condition}_2\text{) then} \\
\text{U(action}_3\text{)} = \theta_3, \ldots \\
\ldots
\]
Examples of probabilistic rules

∀ x,

\[ \text{if } (\text{last-user-act} = x \land \text{system-action} = \text{AskRepeat}) \text{ then} \]
\[ P(\text{next-user-act} = x) = 0.9 \]

“If the system asks the user to repeat his last dialogue act \( x \), the user is predicted to comply and repeat \( x \) with probability 0.9”

∀ x,

\[ \text{if } (\text{last-user-act} = \text{Request}(x) \land x \in \text{perceived-objects}) \text{ then} \]
\[ U(\text{system-action} = \text{PickUp}(x)) = +5 \]

“If the user asks the system to pick up a given object \( x \) and \( x \) is perceived by the system, then the utility of picking up \( x \) is 5”
Rule instantiation

• At runtime, the rules are "executed" by instantiating them in the dialogue state:
  • The rules can be seen as "high-level templates" for the generation of a classical probabilistic model
  • Inference (for state update and action selection) is then performed on this grounded representation

• The use of logical abstractions allows us to capture complex relations between variables in a compact, human-readable form
∀ x, if (last-user-act = x ∧ system-action = AskRepeat) then P(next-user-act = x) = 0.9
∀ x,

if \((\text{last-user-act}=\text{Request}(x) \land x \in \text{perceived-objects})\) then

\(U(\text{system-action}=\text{PickUp}(x)) = +5\)
Processing workflow

- **Information state architecture**, with the dialogue state encoded as a Bayesian Network
- External modules (e.g. ASR, vision) add new observations
- Probability rules employed to update the dialogue state (following the new observations)
- Utility rules employed to determine the system actions
Domain representation

- Dialogue domains are represented in OpenDial in an XML format containing:
  - The initial dialogue state for the interaction
  - A collection of domain models (see below)
  - A collection of external modules & their configuration

- The domain models are simply a collection of (probability or utility) rules that are triggered by a common update event
  - Each model represents a distinct "processing step"
Outline for this talk

• What is dialogue management?
• A hybrid logical/probabilistic approach
  • Probabilistic rules
  • Parameter estimation
• Experiments
• Demonstration of the OpenDial toolkit
Parameter estimation

- Probabilistic rules may include parameters (unknown probabilities or utilities)

- **Bayesian learning** approach:
  - Start with initial prior over possible parameter values
  - Refine the distribution given the observed data $\mathcal{D}$

$$P(\theta | \mathcal{D}) = \eta P(\mathcal{D}; \theta) P(\theta)$$

- Posterior distribution
- Normalisation factor
- Likelihood of the data
- Prior distribution
∀ x,
if \text{(last-user-act = x ∧ system-action = AskRepeat)} \text{ then }
\Pr(next-user-act = x) = \theta

\begin{figure}
\centering
\includegraphics[width=\textwidth]{parameter_estimation.png}
\end{figure}
Learning paradigms

• Different types of training data:
  
  • *Supervised learning*: Wizard-of-Oz interactions
    
    **Goal**: find the parameter values that best “imitate” the Wizard’s conversational behaviour
  
  • *Reinforcement learning*: real or simulated interactions
    
    **Goal**: find the parameter values that provide the best fit for the collected observations

[P. Lison. Model-based Bayesian Reinforcement Learning for Dialogue Management (Interspeech 2013)]
Outline for this talk

• What is dialogue management?

• A hybrid logical/probabilistic approach
  • Probabilistic rules
  • Parameter estimation

• Experiments

• Demonstration of the OpenDial toolkit
User evaluation

• **Task**: instruct the robot to move across the table, pick one cylinder and release it on the landmark

• **Comparison of three modelling approaches:**
  1. A handcrafted finite-state automaton
  2. A factored statistical model
  3. A model structured with probabilistic rules
Experimental procedure

• **Step 1:** collect Wizard-of-Oz interaction data

• **Step 2:** Estimate the internal parameters for the 3 models with the collected data

• **Step 3:** Conduct user trials for the 3 approaches

• **Step 4:** Compare them on dialogue quality metrics

**Dialogue domain:**
- 26 user actions
- 41 system actions
- State size: $35 \times 10^6$ (10 variables)

**Parameter estimation:**
- 10 recorded WoZ interactions
- 3 parameters in handcrafted automaton (thresholds)
- 433 parameters in factored statistical model
- 28 parameters in model encoded with probabilistic rules
Learning curve

Training: 9 Wizard-of-Oz interactions (770 system turns)
Testing: 1 Wizard-of-Oz interaction (71 system turns)
User trials

- 37 participants (16 M / 21 F)
- Average age: 30.6

Interacting with Lenny through spoken dialogue

Pierre Lison
University of Oslo

- Average duration: 5:06 mins
- All captured on videos
User trials

• Each participant in the trial repeated the task three times

• One interaction for each modelling approach (in randomised order)

• Evaluation metrics:
  
  • **Objective metrics:** list of 9 measures extracted from the interaction logs
  
  • **Subjective metrics:** survey of 6 questions filled by the participants after each interaction
### Empirical results

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Finite-state automaton</th>
<th>Factored statistical model</th>
<th>Rule-structured model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average number of repetition requests</td>
<td>18.68</td>
<td>12.24</td>
<td>0*</td>
</tr>
<tr>
<td>Average number of confirmation requests</td>
<td>9.16</td>
<td>10.32</td>
<td>5.78*</td>
</tr>
<tr>
<td>Average number of repeated instructions</td>
<td>3.73</td>
<td>7.97</td>
<td>2.78</td>
</tr>
<tr>
<td>Average number of user rejections</td>
<td>2.16</td>
<td>2.59</td>
<td>2.59</td>
</tr>
<tr>
<td>Average number of physical movements</td>
<td>26.68</td>
<td>29.89</td>
<td>27.08</td>
</tr>
<tr>
<td>Average number of turns between moves</td>
<td>3.63</td>
<td>3.1</td>
<td>2.54*</td>
</tr>
<tr>
<td>Average number of user turns</td>
<td>78.95</td>
<td>77.3</td>
<td>69.14</td>
</tr>
<tr>
<td>Average number of system turns</td>
<td>57.27</td>
<td>54.59</td>
<td>35.11*</td>
</tr>
<tr>
<td>Average duration (in minutes)</td>
<td>6:18</td>
<td>7:13</td>
<td>5:24*</td>
</tr>
</tbody>
</table>

**Objective**

**Subjective**

<table>
<thead>
<tr>
<th><strong>“Did you feel that…”</strong></th>
<th>Finite-state automaton</th>
<th>Factored statistical model</th>
<th>Rule-structured model</th>
</tr>
</thead>
<tbody>
<tr>
<td>... the robot correctly understood what you said?”</td>
<td>3.32</td>
<td>2.92</td>
<td>3.68</td>
</tr>
<tr>
<td>... the robot reacted appropriately to your instructions?”</td>
<td>3.70</td>
<td>3.32</td>
<td>3.86</td>
</tr>
<tr>
<td>... the robot asked you to repeat/confirm your instructions?”</td>
<td>2.16</td>
<td>2.19</td>
<td>3.3*</td>
</tr>
<tr>
<td>... the robot sometimes ignored when you were speaking?”</td>
<td>3.24</td>
<td>2.76</td>
<td>3.43</td>
</tr>
<tr>
<td>... the robot thought you were talking when you were not?”</td>
<td>3.43</td>
<td>3.14</td>
<td>4.41*</td>
</tr>
<tr>
<td>... the interaction flowed in a pleasant and natural manner?”</td>
<td>2.97</td>
<td>2.46</td>
<td>3.32</td>
</tr>
</tbody>
</table>

Scale from 1 (worse) to 5 (best)
Outline for this talk

• The dialogue management task
• A hybrid logical/probabilistic approach
  • Probabilistic rules
  • Parameter estimation
  • Experiments

• Demonstration of the OpenDial toolkit
Conclusion

• Development of a new modelling framework for dialogue management, based on **probabilistic rules**

  • *Hybrid* approach at the crossroads between logical and statistical methods

  • Rule parameters can be learned from data

• Experimental studies demonstrate the benefits of the approach

• Concrete implementation in the OpenDial software toolkit