UiO : University of Oslo

Dialogue Management with Probabilistic Rules

Pierre Lison
Language Technology Group
University of Oslo

November 27, 2012
LORIA

Introduction

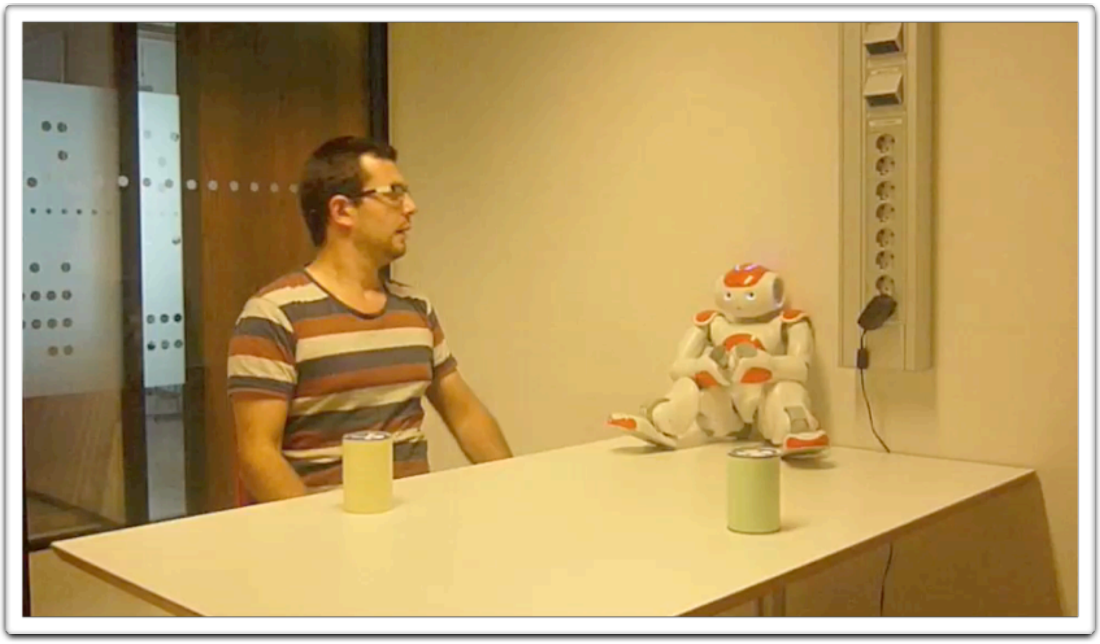
- *Statistical models* is getting increasingly popular in spoken dialogue systems

Advantages	**Challenges**
Explicit account of *uncertainties*, increased *robustness* to errors	Good domain data is *scarce* and expensive to acquire!
Better domain- and user-*adaptivity*, more *natural* and *flexible* conversational behaviours	*Scalability* to complex domains (state space grows exponentially with the problem size)

Introduction

- **Scalability** remains a challenge for many domains
  - Examples: Human-robot interaction, tutoring systems, cognitive assistants & companions
  - Must model a rich, dynamic context (users, tasks, situated environment)
  - State more complex than a list of slots to fill (rich *relational structure*)

Introduction



Outline

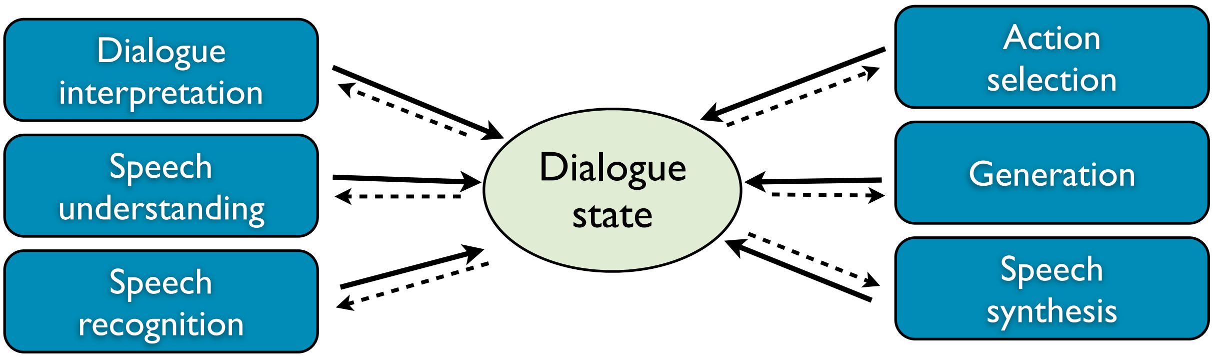
- Generalities
- Probabilistic rules
- Parameter learning
- Conclusions

Outline

- **Generalities**
- Probabilistic rules
- Parameter learning
- Conclusions

General architecture

- *Information-state* based approach to dialogue management (and dialogue systems):
  - The *dialogue state* represents all the information available to the agent (and relevant for decision-making)
  - Various processes are attached to this state and read/write to it

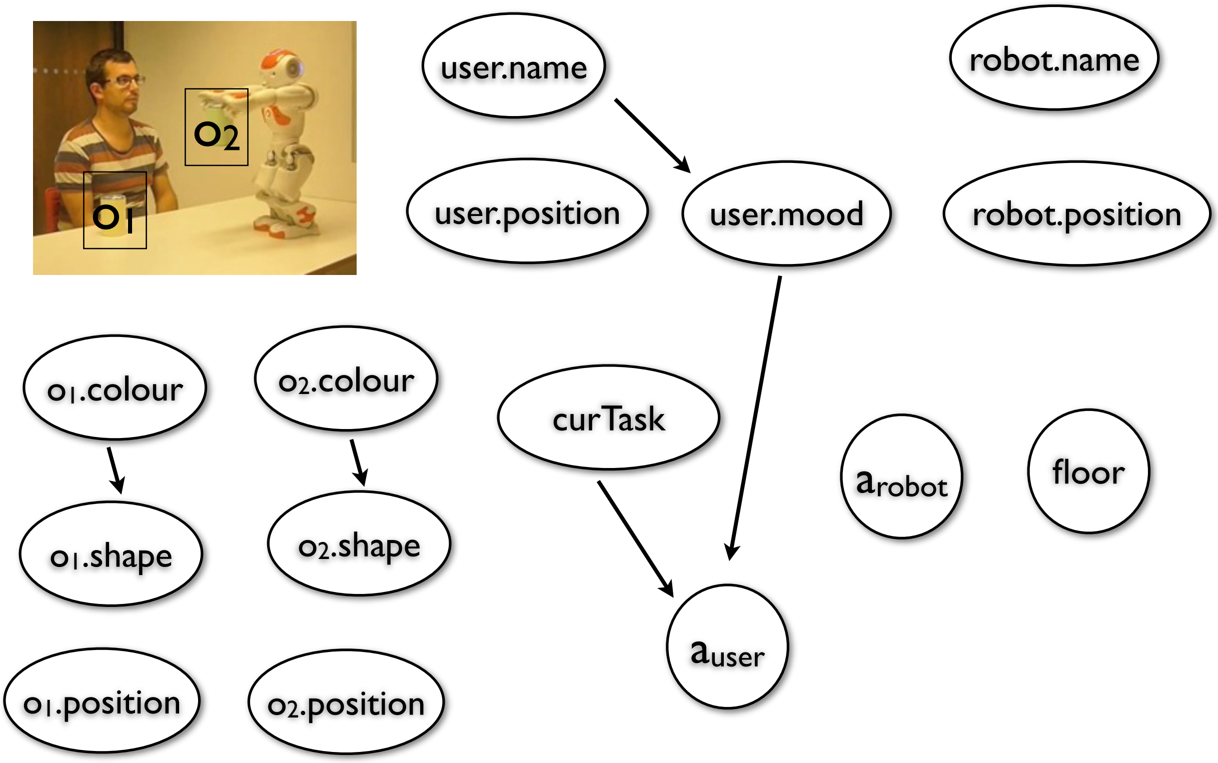


General architecture

- How do we represent the dialogue state?
- Requirements:
  - Must be able to factor the state into distinct *variables*
  - The content of some variables might be *uncertain*
  - Possible *probabilistic dependencies* between variables

Dialogue state encoded as a **Bayesian Network** (i.e. a directed graphical model)

Dialogue state: example



Research problem

- Dialogue management is responsible for a wide range of processing operations:
  - interpretation of the user dialogue acts
  - selection of the next system action to perform
  - prediction of the next steps in the interaction

Complex modelling problem (many interacting variables)

Pervasive uncertainty (ASR errors, ambiguity, unpredictable user behaviour)

Data for parameter estimation is scarce and domain-specific

Research goal

- We would like to construct probabilistic models of dialogue that:
  - can operate on rich state representations
  - can incorporate prior domain knowledge
  - can be estimated from limited amounts of data
- This is basically the central question I'm trying to address for my PhD

UiO : University of Oslo

Research goal

- Many approaches in A.I. and machine learning have tried to tackle related problems
- Solutions typically involve the use of more *expressive representations* (hierarchical or relational abstractions)
  - Can yield more *compact* models that generalise better

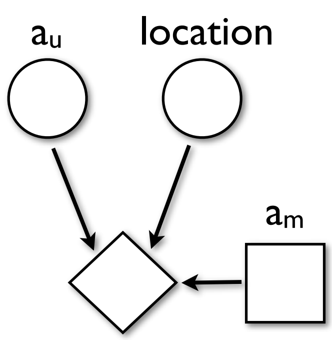
→ I'm developing such a formalism for dialogue management: **probabilistic rules**

Key idea

- Observation: dialogue models exhibit a fair amount of *internal structure*:
  - Probability and utility distributions can often be *factored*
  - Even if the full distribution has many dependencies, the probability (or utility) of a *specific outcome* often depends on a much smaller subset
  - Finally, the values of the dependent variables can often be grouped into *partitions*

Example of partitioning

- Consider a dialogue where the user asks a robot yes/no questions about his location
- The state contains the following variables :
  - Last user dialogue act, e.g. $\mathrm{a_u} = \mathrm{AreYouIn(corridor)}$
  - The robot location, e.g. $\mathrm{location} = \mathrm{kitchen}$
- You want to learn the utility of $\mathrm{a_m} = \mathrm{SayYes}$



- The combination of the two variables can take many values, but they can be partitioned in two sets:

$$\mathrm{a_u} = \mathrm{AreYouIn(x)} \land \mathrm{location} = \mathrm{x} \longrightarrow \text{positive utility}$$
$$\mathrm{a_u} = \mathrm{AreYouIn(x)} \land \mathrm{location} \neq \mathrm{x} \longrightarrow \text{negative utility}$$

Outline

- Generalities
- **Probabilistic rules**
- Parameter learning
- Conclusions

Probabilistic rules

- **Probabilistic rules** attempt to capture such kind of structure
- *High-level templates* for a classical graphical model (in our case, a Bayesian Network)
- Advantages:
  - (Exponentially) fewer parameters to estimate
  - Easier to incorporate prior domain knowledge

Probabilistic rules

- The rules take the form of structured **if...then...else** cases
- Mapping from *conditions* to (probabilistic) *effects*:

if (condition$_1$ holds) **then**
  P(effect$_1$)= θ_1, P(effect$_2$)= θ_2, …
else if (condition$_2$ holds) **then**
  P(effect$_3$) = θ_3, …
…

Probabilistic rules

- *Conditions* are (arbitrarily complex) logical formulae on state variables
- *Effects* are value assignments on state variables
- Effect probabilities are *parameters* that can be estimated from data

Example:

if (a_m = AskRepeat) **then**
  $P(a_u' = a_u) = 0.9$
  $P(a_u' \neq a_u) = 0.1$

Utility rules

- The formalism can also describe *utility models*
- In this case, the rule maps each condition to an assignment of *utility values* for particular actions:

if (condition$_1$ holds) **then**
  Q(actions$_1$)= θ_1, Q(actions$_2$)= θ_2, ...
else if (condition$_2$ holds) **then**
  Q(actions$_3$) = θ_3, ...
...
else
  Q(actions$_n$) = θ_n, ...

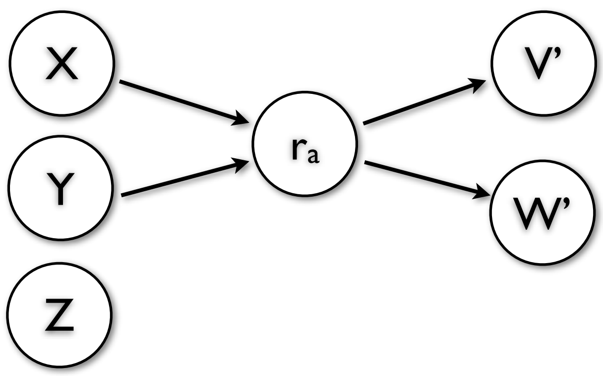
Rule instantiation

- How are the rules applied to the dialogue state?
- The rules are *instantiated* in the Bayesian Network, expanding it with new nodes and dependencies

r_a:

if ($X = ... \vee Y \neq ...$) **then**
  $P(V = ... \wedge W = ...) = 0.6$

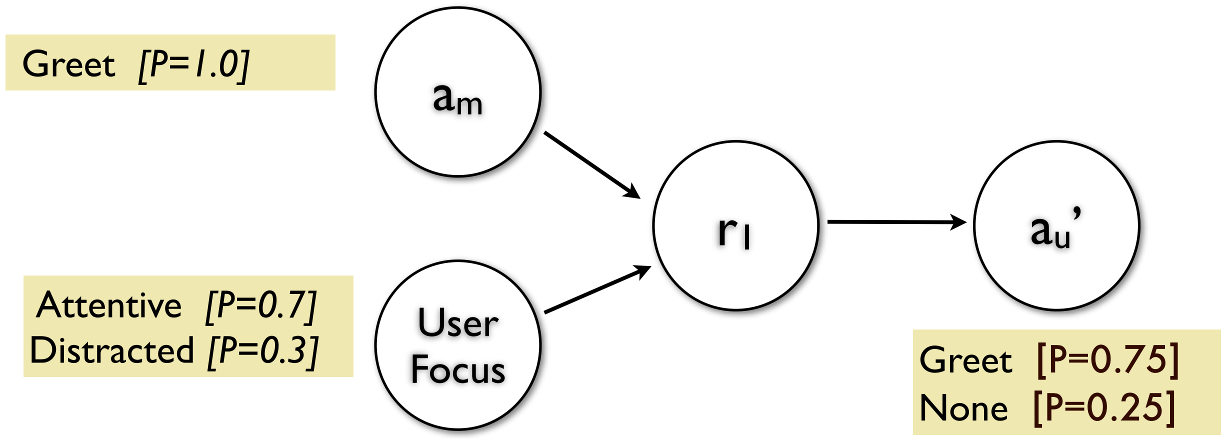
(The ... dots in r_1 should be replaced by concrete values)



Rule instantiation

Example r_1:

if (a_m = Greet ∧ UserFocus = Attentive) **then**
  $P(a_u'$ = Greet) = 0.9
else if (a_m = Greet ∧ UserFocus = Distracted) **then**
  $P(a_u'$ = Greet) = 0.4

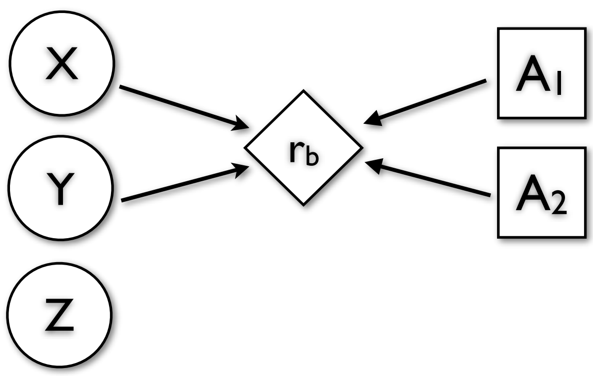


Rule instantiation

- The instantiation procedure is similar for utility rules, although one must employ utility and decision nodes:

r_b:

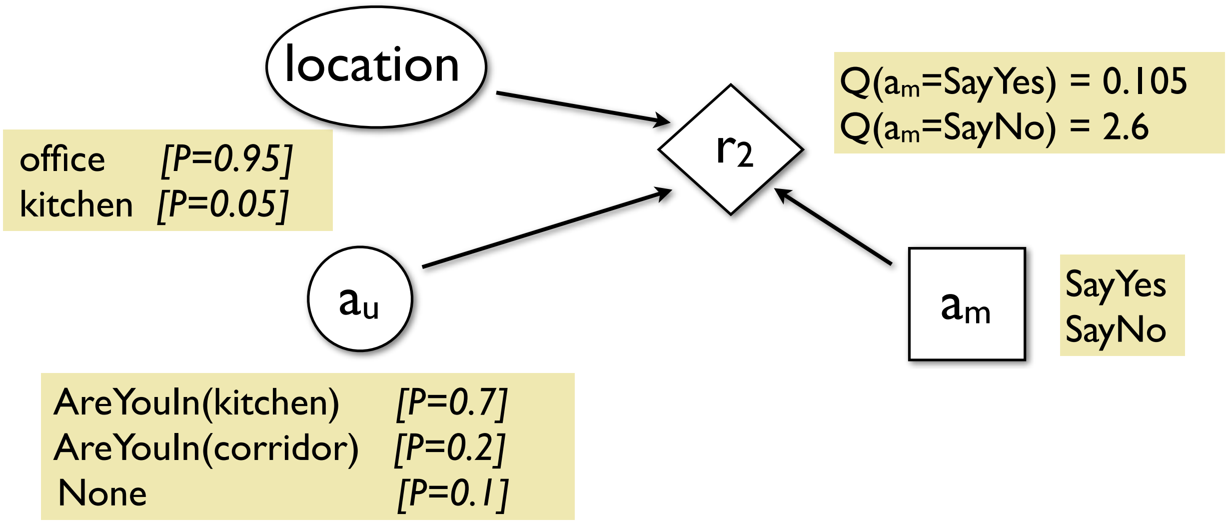
if (X = ... ∨ Y ≠ ...) **then**
  $Q(A_1$ = ... ∧ A_2 = ...) = 3



Rule instantiation

Example r2:

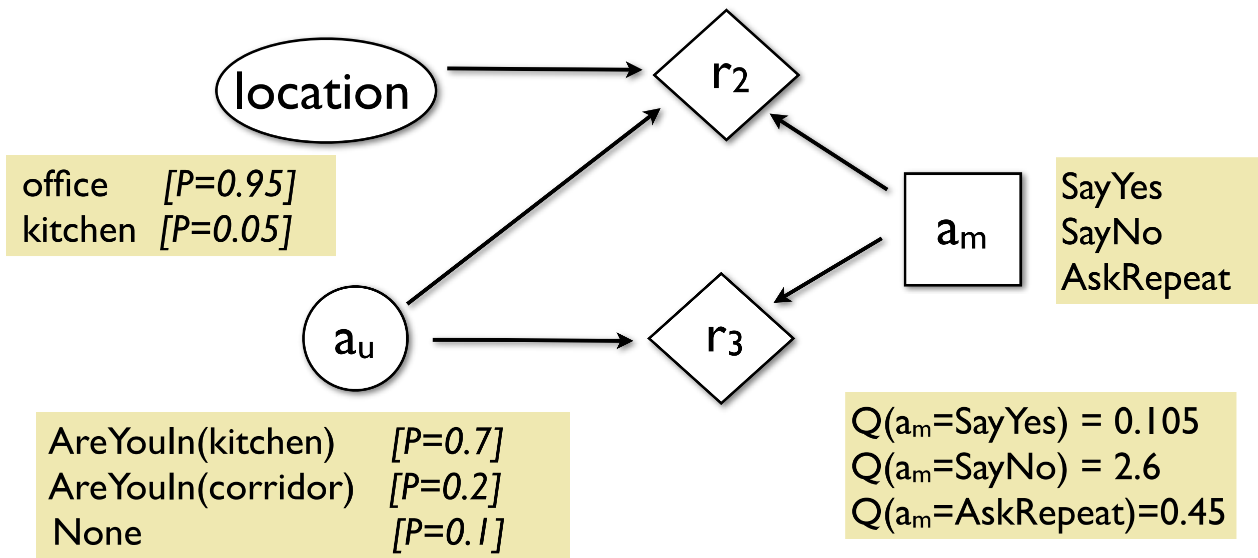
$$\begin{aligned}&\textbf{if } (a_u = \text{AreYouIn}(x) \wedge \text{location} = x) \textbf{ then}\\ &\quad \{Q(a_m = \text{SayYes}) = 3.0\}\\ &\textbf{else if } (a_u = \text{AreYouIn}(x) \wedge \text{location} \neq x) \textbf{ then}\\ &\quad \{Q(a_m = \text{SayNo}) = 3.0\}\end{aligned}$$



Rule instantiation

Example r3:

$$\begin{aligned}&\textbf{if } (a_u \neq \text{None}) \textbf{ then}\\ &\quad \{Q(a_m = \text{AskRepeat}) = 0.5\}\end{aligned}$$

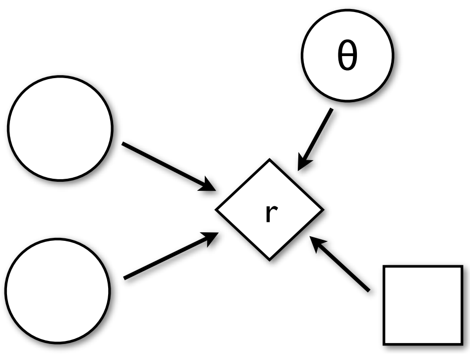


Outline

- Introduction
- Probabilistic rules
- **Parameter learning**
- Conclusions

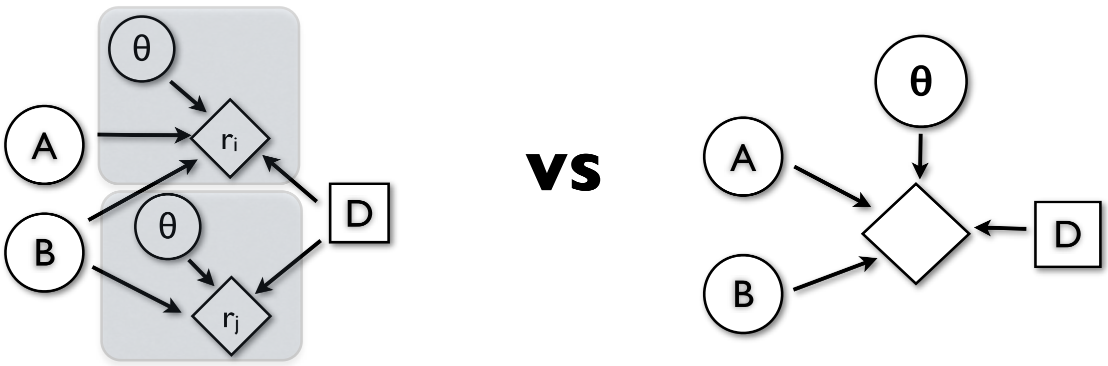
Parameter learning

- The rule parameters (probabilities or utilities) must be estimated from empirical data
- We adopted a Bayesian approach, where the parameters are themselves defined as variables
- The parameter distributions will then be modified given the evidence from the training data



Evaluation

- Policy learning task in a human-robot interaction scenario, based on Wizard-of-Oz training data
- Objective: estimate the utilities of possible system actions
- Baselines: «rolled-out» versions of the model
  - «plain» probabilistic models with identical input and output variables, but without the condition and effect nodes as intermediary structures



Experimental setup

- Interaction scenario: users instructed to teach the robot a sequence of basic movements (e.g. a small dance)
- Dialogue system comprising ASR and TTS modules, shallow components for understanding and generation, and libraries for robot control
- The Wizard had access to the dialogue state and took decisions based on it (among a set of 14 alternatives)
- 20 interactions with 7 users, for a total of 1020 turns

Each sample *d* in the data set is a pair (b_d, t_d):

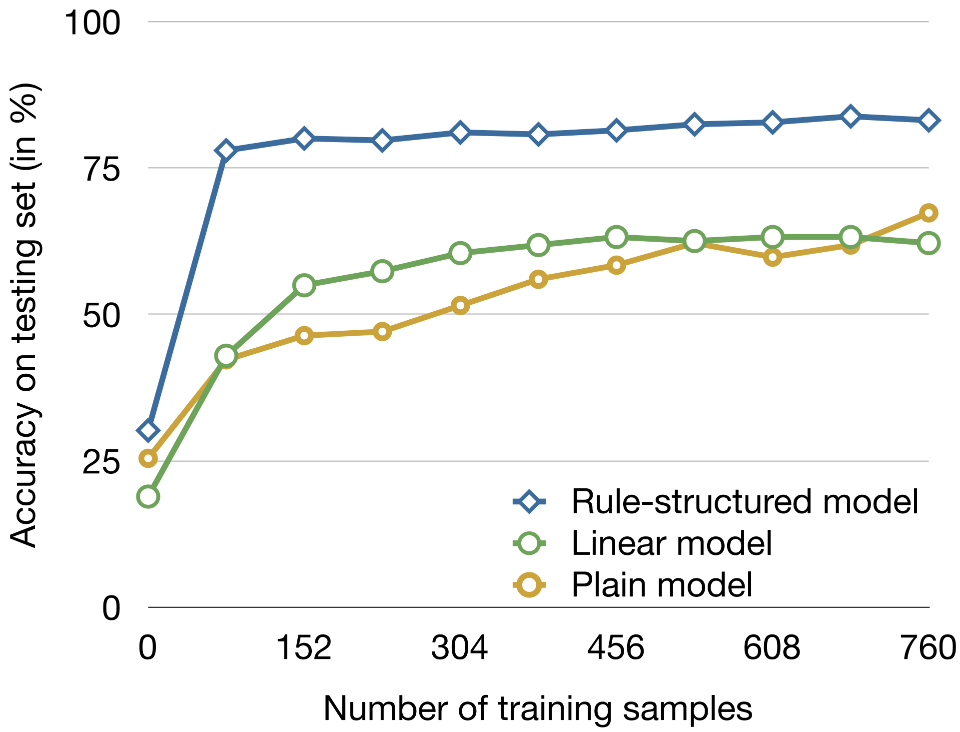
- b_d is a recorded dialogue state
- t_d is the «gold standard» system action selected by the Wizard at the state b_d

Empirical results

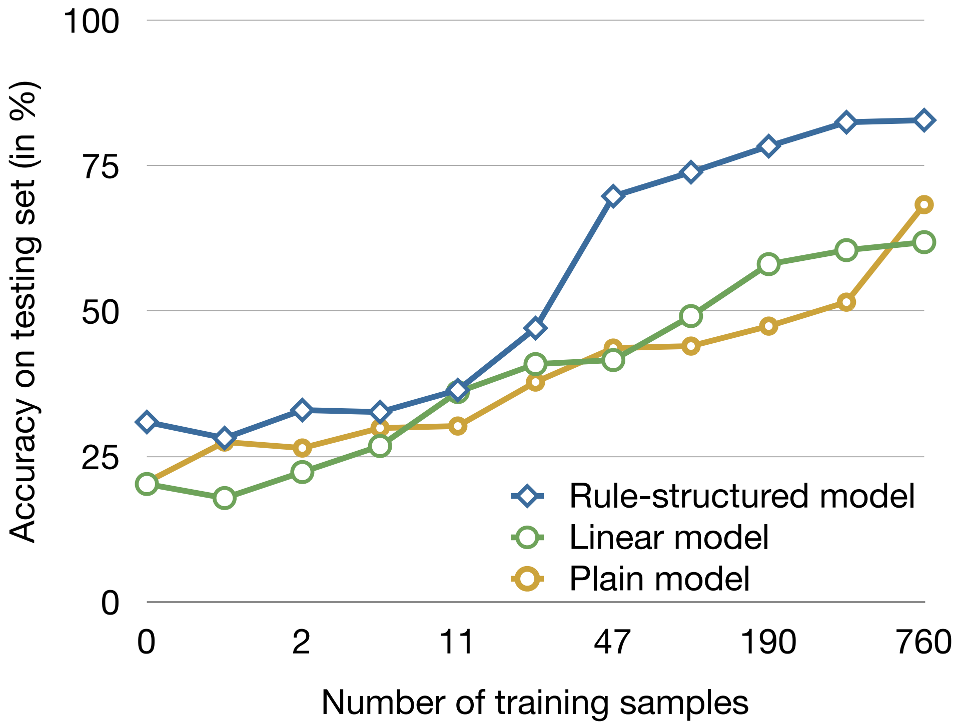
- Data set split into training (75%) and testing (25%)
- Accuracy measure: percentage of actions corresponding to the ones selected by the Wizard
  - But Wizard sometimes inconsistent / unpredictable
- The rule-structured model outperformed the two baselines in accuracy and convergence speed

Type of model	Accuracy (in %)
Plain model	67.35
Linear model	61.85
Rule-structured model	**82.82**

Learning curve (linear scale)



Learning curve (log-2 scale)



Outline

- Introduction
- Probabilistic rules
- Parameter learning
- **Conclusions**

Conclusions

- **Probabilistic rules** used to capture the underlying structure of dialogue models
- Allow developers to exploit powerful generalisations and domain knowledge
- ... without sacrificing the probabilistic nature of the model

Current & future work

- *Model-based reinforcement learning*: instead of relying on annotated data, learn the parameters from (real or simulated) interactions
- Apply the probability and utility rules to perform *online planning*
- Perform *joint optimisations* of several dialogue models, all encoded with probabilistic rules
- Development of a software toolkit (openDial) and evaluation in a human-robot interaction domain

Questions, comments?

- Still a work in progress - comments, suggestions are *most* welcome!