Towards Dialogue Management in Relational Domains

Pierre Lison

Department of Informatics
University of Oslo, Norway
plison@ifi.uio.no

Abstract

Traditional approaches to dialogue management rely on a fixed, predefined set of state variables. For many application domains, the dialogue state is however best described in terms of a collection of varying number of *entities* and *relations* holding between them. These entities might correspond to objects, places or persons in the context of the interaction, or represent a set of tasks to perform. Such formalization of the state space is well-suited for many domains, but presents some challenges for the standard probabilistic models used in dialogue management, since these models are propositional in nature and thus unable to directly operate on such state representation. To address this issue, we present an alternative approach based on the use of expressive *probabilistic rules* that allow for limited forms of universal quantification. These rules take the form of structured mappings between input and output variables, and function as high-level templates for the probability and utility models integrated in the dialogue manager. We present in this abstract the general formalisation of this approach, focusing on the use of universal quantifiers to capture the relational structure of the domain.

1. Introduction

Slot-filling applications are sometimes perceived as the "prototypical" domain for spoken dialogue systems. Their dialogue state is usually represented as a collection of slots representing the user intention (such as booking a flight ticket or getting information about bus routes), and the purpose of the interaction is to fill these slots as accurately and efficiently as possible. The state space for these applications can be large, but is strictly limited by the number of slots and the number of values allowed for each slot.

Many dialogue domains cannot however be easily described in such a way. This is especially the case for situated domains such as human-robot interaction (Iwahashi, 2006; Kruijff et al., 2010), , cognitive assistants and companions (Cavazza et al., 2010), and tutoring systems (Eskenazi, 2009). These domains are not limited to the completion of a single task with predefined features but have to represent a varying number of tasks, complex user models and – last but not least – a rich, dynamic environment. This external environment is often best described in terms of a varying number of *entities* related to one another. Examples of such relational structure of entities include:

- Collections of physical objects in a visual scene, each described by specific features (colour, shape) and relations with other objects (e.g. spatial relations);
- Indoor environments topologically structured in various rooms and spaces in which to navigate;
- Stacks of tasks to complete by the agent, each task being possibly connected to other tasks via precedence or inclusion relationships.

This formalisation of the state space in terms of entities related to one another is well-suited for many domains, but is not easy to encode using classical probabilistic models. The expressive power of such models is indeed limited to propositional logic and thus unable to directly capture the relational semantics of such representation. In this abstract, we present an alternative approach we are currently developing in our research work, based on *probabilistic rules* augmented with a limited form of universal quantification. These rules provide a compact encoding for the various probability and utility models integrated in the dialogue manager. This abstract focuses on the integration of universal quantifiers to the rule structure, enabling us to capture the relational structure of the domain.

2. Approach

We base our approach on the concept of *probabilistic rules*, as already outlined in (Lison, 2012b; Lison, 2012a). We'll first briefly review this description formalism, and then explain how to extend it to handle relational domains.

2.1 Probabilistic rules

A rule is defined as a condition-effect mapping, where each condition is mapped to a set of alternative effects. Each effect is associated with a probability value. The list of conditions is ordered and takes the form of a "**if** ... **then** ... **else**" case expressing the distribution of the output variables depending on the inputs.

Formally, a probability rule r is defined as an ordered list of cases, where each case is associated with a condition c_i as well as a distribution over stochastic effects $\{(e_i^1, p_i^1), ..., (e_i^k, p_i^k)\}$. For each stochastic effect e_i^j, we have that $p_i^j = P(e_i^j | c_i)$, where $p_i^{1 \cdots m}$ satisfy the usual probability axioms. The rule reads as such:

if (c_1) **then**
$$\{P(e_1^1) = p_1^1, \; ... \; P(e_1^k) = p_1^k\}$$
else if (c_2) **then**
$$\{P(e_2^1) = p_2^1, \; ... \; P(e_2^l) = p_2^l\}$$
...
else if (c_n) **then**
$$\{P(e_n^1) = p_n^1, \; ... \; P(e_n^m) = p_n^m\}$$

The conditions c_i are expressed as (propositional) logical formulae grounded in the input variables. They can be arbitrarily complex formulae connected by conjunctive, disjunctive and negation operators. The effects are defined in the same way and encode specific value assignments for a set of output variables.

One can also use the framework to encode *utility* distributions instead of probability distributions. In such case, the above structure remains essentially the same, with the probability distributions $P(\cdot)$ over output variables replaced by utility distributions $Q(\cdot)$ over specific action(s) that can be executed by the agent.

At runtime, these rules are then instantiated by extending the Bayesian network representing the dialogue state with new nodes and dependencies capturing the semantics of the rules, as illustrated in Figure 1. Rules can trigger one another in complex chain of updates. The application of these rules effectively updates the dialogue state as new information becomes available, and also determines the set of actions that can be executed at any given point in the interaction.

This system design is directly inspired by information-state update approaches to dialogue management (Larsson and Traum, 2000), with the notable difference that the rules are in our case probabilistic rather than deterministic, and include parameters that can be estimated from data. The estimation of these rule parameters is done based on a Bayesian approach, by integrating additional variables in the dialogue state representing the model uncertainty. Given some training data collected via Wizard-of-Oz experiments or real interactions, it becomes possible to update the probability distributions of these variables and narrow down their spreads to the values providing the best fit for this data. The interested reader is invited to consult (Lison, 2012b; Lison, 2012a) for more details.

2.2 Quantification

We now describe how to extend this framework to handle relational domains. We would like to express that a rule holds for *any* state variable satisfying some criteria, something which requires some form of universal quantification. The full expressivity of first-order logic seems however overkill for our more modest modelling needs. Our proposed solution is therefore to extend the formalism with a limited form of quantification, and express the rules in the following form:

$$\begin{aligned}
&\forall \mathbf{x} = x_1, ...x_k : \\
&\textbf{if } (c_1(\mathbf{x})) \textbf{ then} \\
&\quad \{P(e_1^1(\mathbf{x})) = p_1^1,\ ...\ P(e_1^k(\mathbf{x})) = p_1^k\} \\
&... \\
&\textbf{else if } (c_n(\mathbf{x})) \textbf{ then} \\
&\quad \{P(e_n^1(\mathbf{x})) = p_n^1,\ ...\ P(e_n^m(\mathbf{x})) = p_n^m\}
\end{aligned}$$

In other words, the above formalisation allows for certain variables $x_1, ...x_k$ in the conditions and effects to be *underspecified*. In such a case, the rule will be instantiated for every possible assignment of the underspecified variables.

It has been previously shown that the probability and utility parameters of such rules can be efficiently learned from data (Lison, 2012b). One of the key advantages of such representation is it allows for powerful forms of *parameter sharing* – for instance, the effect probabilities in the above rule are now made independent of the actual instantiation of the variables $x_1, ...x_k$. Quantification mechanisms are thus more than mere "syntactic shorthands" for grounded rules, as they allow the learning algorithm to generalise over the observed data and therefore greatly reduce the number of parameters to estimate.

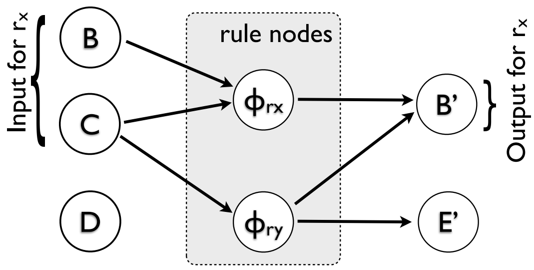

Figure 1: Example of application of two rules r_x and r_y on a dialogue state including two variables B and C, leading to the updated variables B' and E'.

2.3 Examples

We present here two simple examples of rules employing the above quantification mechanism. Assume a dialogue state containing a set of objects, each with a specific colour. Rule r_1 below states that, if the user requests the robot to pick up the object, the robot should do so with utility 5.

$$\begin{aligned}
&r_1 : \forall o : \\
&\quad \textbf{if } (a_u = \text{RequestPickUp}(o)) \textbf{ then} \\
&\quad\quad \{Q(a'_m = \text{PickUp}(o)) = 5\}
\end{aligned}$$

In the above rule, a_u denotes the user dialogue act, a'_m denotes the (next) system dialogue act, and o is a reference to an arbitrary physical object. Rule r_1 is a rule specifying the *utility* of the PickUp(o) action – the dialogue system will then select the action a'_m with the highest utility. Thanks to the universal quantifier, we are able to express that the utility of the PickUp(o) action does not depend on the actual identity of the o object being referred to.

Rule r_2 is another example of rule, which predicts the next dialogue act from the user given the current dialogue state. Rule r_2 states that, if the robot utters a question such as "What colour is the object", the user is likely to utter a dialogue act such as "the object is X" at the next turn, where X is the actual colour of the object.

$$\begin{aligned}
&r_2 : \forall o, c : \\
&\quad \textbf{if } (a_m = \text{WhatIsColour}(o) \wedge o.colour = \text{c}) \textbf{ then} \\
&\quad\quad \{P(a'_u = \text{Assert}(\text{Is}(o, \text{c}))) = 0.9\}
\end{aligned}$$

If the dialogue state contains e.g. one single object o_1 with $P(o_1.colour = blue) = 0.8$, the probability of the user uttering "the object is blue" is therefore 0.72. This prediction can be used to e.g. prime the ASR's language model or rerank the output of the dialogue act classifier.

3. Conclusions and Future Work

Interactions in situated environments are both complex and uncertain – that is, they exhibit both high degrees of structural complexity and high levels of uncertainty. Spoken dialogue systems operating in such conditions must therefore address these two challenges. In our view, one promising strategy is to rely on *expressive probabilistic models* able to exploit powerful generalisations and domain knowledge without sacrificing the model's stochasticity.

We have presented in this abstract a simple approach for specifying the probability and utility models of a dialogue manager when the dialogue state is expressed as a relational structure. The key idea is to integrate a limited form of universal quantification in the rule specification, and instantiate the rules for every possible valid assignment of the unbound variables. The approach is currently in development as part of an open source dialogue toolkit called openDial, which is entirely based on probabilistic rules.

The advantages of explicitly encoding relational structures in the state space has long been recognised in A.I. and machine learning (Getoor and Taskar, 2007; Otterlo, 2012). The main challenge is to formalise the probabilistic models operating on this state space to exploit this relational structure. This is often realised via abstraction mechanisms such as first-order logic (Richardson and Domingos, 2006). In this paper, we adopted a somewhat simpler approach based on a restricted form of quantification.

The key question that we are currently investigating is how to keep the formalism tractable for dialogue management, which needs to operate in real-time. Since each rule has to be instantiated for every possible assignment of the underspecified variables, we need to perform some form of *aggressive pruning* to quickly discard irrelevant instantiations of the rules – for instance, instantiations of the rule for which no condition applies. We are currently looking into the best way to define such pruning heuristics.

The empirical evaluation of our approach will be realised in a human-robot interactions scenario, involving a *Nao* robot (from Aldebaran Robotics) operating in a tabletop setting containing a few visual objects that can be picked up and moved according to the instructions of a human user.

Figure 2: Example of interaction with the Nao robot in a shared visual scene containing simple objects.

4. References

M. Cavazza, R. Santos de la Camara, M. Turunen, J. Relaño-Gil, J. Hakulinen, N. Crook, and D. Field. 2010. How was your day? an affective companion ECA prototype. In *Proceedings of the 11th SIGDIAL Meeting on Discourse and Dialogue*, pages 277–280.

M. Eskenazi. 2009. An overview of spoken language technology for education. *Speech Commununications*, 51:832–844.

L. Getoor and B. Taskar. 2007. *Introduction to Statistical Relational Learning*. The MIT Press.

N. Iwahashi. 2006. Robots that learn language: Developmental approach to human-machine conversations. In *Symbol Grounding and Beyond, Third International EELC Workshop*, pages 143–167. Springer.

G.-J. M. Kruijff, P. Lison, T. Benjamin, H. Jacobsson, Hendrik Zender, and Ivana Kruijff-Korbayová, 2010. *Situated Dialogue Processing for Human-Robot Interaction*, chapter 8. Springer Verlag, Heidelberg, Germany.

S. Larsson and D. R. Traum. 2000. Information state and dialogue management in the trindi dialogue move engine toolkit. *Natural Language Engineering*, 6:323–340.

P. Lison. 2012a. Declarative design of spoken dialogue systems using probabilistic rules. In *Proceedings of the 16th SemDial Workshop*, Paris, France.

P. Lison. 2012b. Probabilistic dialogue models with prior domain knowledge. In *Proceedings of the 13th SIGDIAL Meeting on Discourse and Dialogue*, pages 179–188, Seoul, South Korea.

M. Otterlo. 2012. Solving relational and first-order logical markov decision processes: A survey. In *Reinforcement Learning*, volume 12 of *Adaptation, Learning, and Optimization*, pages 253–292. Springer Berlin Heidelberg.

M. Richardson and P. Domingos. 2006. Markov logic networks. *Machine Learning*, 62:107–136.