Structured Probabilistic Modelling for Dialogue Management

Doctoral Dissertation by

Pierre Lison

Department of Informatics
Faculty of Mathematics and Natural Sciences
University of Oslo

Submitted for the degree of Philosophiae Doctor

April 15, 2013

Abstract

TODO

Acknowledgements

TODO

don't forget to mention the HPC infrastructure

Contents

Chapter 1

Introduction

Spoken language is one of the most powerful system of communication at our disposal. A large part of our waking hours is spent in social interactions mediated through natural language. The pivotal role of spoken language in our daily lives is largely due to its remarkable proficiency at conveying elaborate thoughts in a robust, flexible and efficient manner.

Is it possible to exploit this simple observation to develop more human-friendly technologies? Most of our everyday activities are now relying on "smart" electronic devices of various kinds, from mobile phones to personal computers and in-car navigation systems. As these technologies gain in autonomy and sophistication, it becomes increasingly important to design user interfaces that can combine expressiveness, adaptivity and ease of use. In this context, it seems judicious to endow these devices with a capacity to understand, even in a limited manner, the communication medium that is most natural to us, namely spoken language.

The ongoing research on *spoken dialogue systems* (SDS) is precisely trying to achieve this objective. A spoken dialogue system is a computer agent that is able to converse with humans through everyday spoken language in order to perform its task(s). Such systems are expected to play an ever-increasing role in our daily interactions with technology. They have a wide range of applications, ranging from phone-based systems for information access and service delivery to voice-enabled software for hand-held devices, navigation assistants, interactive tutoring systems, and (in a not-too-distant future) service robots assisting us in our everyday environments.

Figure 1.1 illustrates an example of interaction between a human user and a spoken dialogue system. When the user starts talking, the system extracts the corresponding speech signal through a microphone. The speech signal is then processed to analyse its content. Once this operation is completed, the system must then decide how to react. In our case, the system decides to greet back the user and selects the words to express it ("*good morning, sir*"). The final step is then to synthesise these words through an artificial voice, which closes the loop[1].

1.1 Motivation

Although the deployment of spoken dialogue systems is attractive for many reasons, their practical development can be a demanding enterprise. Speech is indeed much more complex than other

[1] Needless to say, the schema hides a great deal of internal complexity. In particular, it omits the existence of non-verbal inputs and outputs (e.g. additional modalities, external actions) which are present in most applications. The next chapter will describe in more details the software architectures used to design spoken dialogue systems.

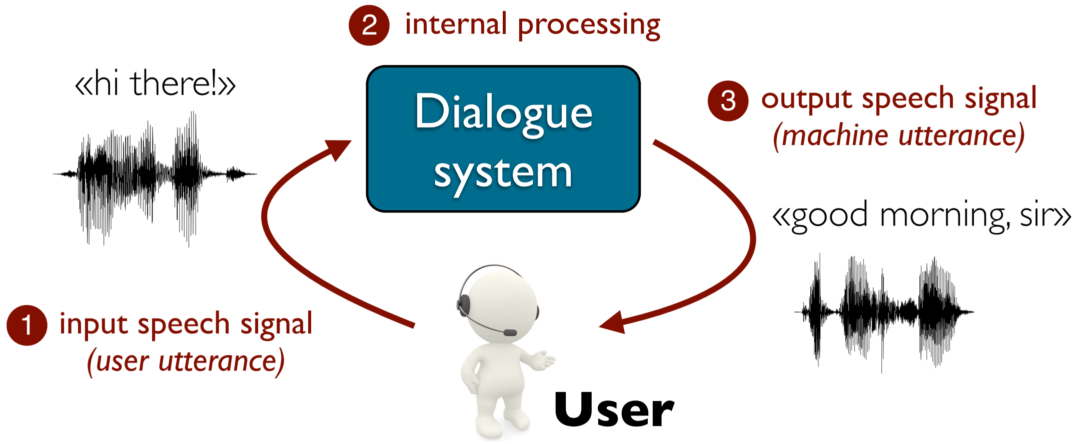

Figure 1.1: Schematic view of a spoken dialogue system

modalities for user interaction such as keyboards or touch screens.

The present thesis concentrates on the problem of *dialogue management*. Dialogue management is a central function in spoken dialogue systems. It serves a double role. Its first task is to maintain a representation of the current dialogue state. This representation might include any information that is relevant for the system, and often include features related to the dialogue history, the external context, and the current tasks to perform. This dialogue state is regularly updated with new information, which comes either in the form of new user utterances or perceived changes in the context. The second task of dialogue management is to make decisions. Based on the current state of the interaction, dialogue management must decide which actions to undertake. These actions are often communicative in nature (e.g. uttering a sentence), but can also pertain to physical actions to execute (e.g. grasping an object).

Dialogue management is therefore responsible for controlling the flow of the interaction, by (1) deciding how to interpret the user inputs in their context and (2) selecting which actions to perform next. In the example from Figure 1.1, this step corresponds to the decision of responding to the user utterance "*hi there!*" with another greeting action, "*good morning, sir*".

Along with speech recognition, dialogue management has proven be be one of the most difficult processing step in spoken dialogue systems. To understand why this is the case, it is useful to have a closer look at two defining features of verbal interactions:

1. Verbal interactions are highly *structured*. Their analysis reveals the presence of multiple relations straddling the linguistic and extra-linguistic levels of interpretation. Spontaneous conversations are notably rife with partial utterances and referring expressions that can only be uncovered based on their context. Furthermore, verbal interactions are at their core a fundamentally collaborative activity where the contributions of the different speakers are tightly coupled (**??**). A dialogue move is therefore only intelligible within the larger pragmatic context that gave rise to it.

2. Verbal interactions are also crippled with *uncertainties*. In order to make sense of a given utterance, a conversational agent must face numerous sources of uncertainty, including error-prone speech recognition, lexical, syntactic and referential ambiguities, partially observable environments, and unpredictable interaction dynamics.

The combination of these two properties forms an explosive mix. In order to make sense of the interaction and act appropriately, the dialogue system must be able to perform complex reasoning in order to interpret the user intentions and plan the best course of action. And it must do so under high levels of noise and uncertainty, where many pieces of information can contain errors or be missing, ambiguous, or fragmentary. This problem is known in Artificial Intelligence as *sequential decision-making under uncertainty* (**??**), and it remains to this day a difficult computational task, especially for complex domains such as dialogue.

Research on dialogue management can be divided into two main lines of investigation that reflect their focus on either of the two linguistic properties we just mentioned.

On the one hand, structural complexity is often dealt with conceptual tools borrowed from formal logic. These approaches provide principled methods for the interpretation and generation of dialogue moves through logical reasoning on the basis of a formal representation of the mental states of the dialogue participants (including their shared knowledge). This representation might incorporate the beliefs, desires and intentions of each agent (**??**), social obligations (**?**), or open questions raised during the interaction (**??**). These approaches can provide detailed analyses of various dialogue behaviours, but they generally assume complete observability of the dialogue context and provide only a very limited account (if any) of errors and uncertainties. In addition, they require the knowledge base on which the inference is grounded to be completely specified in advance by domain experts. Their deployment in practical applications is therefore non trivial.

On the other hand, the problem of uncertainty is usually addressed by probabilistic modelling techniques (**???**). The state of the dialogue is here represented as a probability distribution over possible worlds. This distribution represents the system's current knowledge of the interaction and is regularly updated as new observations are collected. These probabilistic models provide an explicit account for the various uncertainties that can arise during the interaction. They also enable the dialogue behaviour to be automatically optimised in a data-driven manner instead of relying on hand-crafted mechanisms. Dialogue strategies can therefore be adapted to new environments or users without having to be reprogrammed. However, these models typically depend on large amounts of training data to estimate their parameters – a requirement that is hard to satisfy for most dialogue domains. In addition, the probabilistic models are usually limited to a handful of state variables and are difficult to scale to domains featuring rich conversational contexts.

The work described in this thesis aims at reconciling these two strands of research via a new, hybrid computational framework for dialogue modelling.

1.2 Contributions

The present thesis details an original approach to dialogue management based on *structured probabilistic modelling*. Its objective is to provide probabilistic models of dialogue that are (1) more scalable to rich domains, (2) easier to estimate from small amounts of training data.

There is an extensive body of work in the machine learning and decision-theoretic planning literature which shows how to address this issue by relying on more expressive representations, able to capture relevant aspects of the problem *structure* in a compact manner. By taking advantage of hierarchical or relational abstractions, system developers can leverage their domain knowledge to yield probabilistic models which are both easier to learn (due to a reduced number of parameters) and more efficient to use (since the structure can be exploited by the inference algorithm).

This thesis demonstrates how to translate these insights into dialogue modelling. We present a new framework for describing probabilistic models of dialogue, based on the concept of *probabilistic rules*. These rules express the distributions in terms of structured mappings associating specific conditions on a set of input variables to probabilistic effects defined on a set of output variables.

The presented modelling framework offers two major benefits. Most importantly, the reliance on more expressive representations allows us to drastically reduce the number of parameters associated with the models. Instead of being encoded through traditional probability tables, the conditional distributions between variables are expressed through high-level rules that capture the dependencies between variables via a small set of parameters (one for each possible effect). As a consequence, these models are much easier to learn and generalise to unseen data.

In addition, the framework enables expert knowledge to be directly incorporated into the probabilistic models. System developers are thus free to exploit powerful abstractions to encode their prior knowledge of the dialogue domain in the form of pragmatic rules, generic background knowledge, or task-specific constraints. While there exists previous work on the integration of expert knowledge using finite-state policies or ad-hoc constraints (**??**), these approaches essentially use this information source as an external filter to a classical model. By contrast, our approach seeks to incorporate this expert knowledge in the very structure of the statistical model.

At runtime, these probabilistic rules are instantiated on the variables that compose the dialogue state. This instantiation is realised by converting the rules into the nodes of a Bayesian Network (a.k.a. a directed graphical model). The probabilistic rules can therefore be seen as providing high-level templates for the construction of a classical probabilistic model. After grounding the rules in the Bayesian Network, various inference operations can be triggered to e.g. update the model with new observations or search for the action yielding the highest utility. Standard algorithms for probabilistic inference such variable elimination (**?**) and likelihood weighted sampling (**?**) are applied for this purpose.

We conducted several experiments to assess the validity of our approach in different learning scenarios:

1. The first experiment, detailed in Section 4.3, focussed on the problem of estimating the utilities of various system actions given a small data set collected from Wizard-of-Oz interactions.[2]. Based on dialogue models encoded with probabilistic rules, the utilities of the different actions were learned through the systematic application of Bayesian inference in a supervised learning setting. We were then able to show that the rule structure enabled the learning algorithm to converge faster and with better generalisation performance than unstructured models. This work was originally presented in (**?**).

2. The second experiment, described in Section 5.3, extended the above approach to reinforcement learning. The goal of this study was to estimate the transition model of the domain from interactions with a user simulator. We compared the relative learning performance of two modelling approaches: one relying on unstructured distributions, and one based on

[2]A Wizard-of-Oz interaction is an experimental procedure borrowed from Human-Computer Interaction (HCI) studies (**?**). In a Wizard-of-Oz experiment, the subjects are asked to interact with a computer system which has all the appearances of reality, but is actually remotely controlled by an (unseen) human agent operating behind the curtains. Wizard-of-Oz studies are often conducted to provide the system designers with interaction data from real users before the system is fully implemented.

probabilistic rules. The empirical results demonstrated once more the benefits of capturing the domain structure with probabilistic rules. The results were first published in XXX

3. Finally, the third experiment was designed to evaluate the approach through live interactions with real users. to be completed

An additional contribution of our thesis is a software toolkit that implements all the representations and algorithms presented in this work. The toolkit is dubbed `openDial` and is freely available under an open source licence.[3] It enables system developers to design, evaluate and deploy dialogue systems based on probabilistic rules. The toolkit is fully generic since all all domain-specific knowledge is declaratively specified in the rules for the domain. This design choice effectively simplifies the system architecture to a small set of core algorithms for accessing and updating the dialogue state (?). The `openDial` toolkit comes with a user interface allowing developers to interactively test their system and visualise how the internal dialogue state is evolving over time. Its implementation is described in Appendix C.

We carried out all the experiments described in this thesis in a *human–robot interaction* (HRI) domain. The selection of this specific domain as a test bed for our framework was motivated by two factors. First of all, HRI domains often embody a rich mix of contextual features extracted from the situated environment and the tasks to complete by the agent. Moreover, HRI domains must frequently significant levels of uncertainty due to imperfect sensors, unreliable motors, and failure-prone speech recognition.

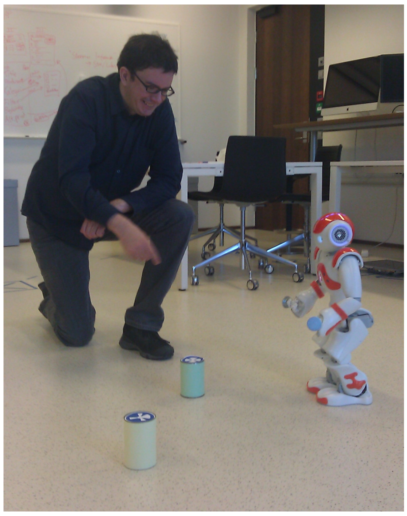

Figure 1.2: Human user interacting with the Nao robot.

The Nao robot from Aldebaran Robotics was used as a platform for all our experiments [4]. An example of interaction with the robot is shown in Figure 1.2. Most of our experiments involved the Nao robot interacting with a human user in a shared visual environment featuring a few basic objects that can be automatically perceived by the robot. The user were instructed to command the robot to execute various tasks such as grasping objects and moving them from one place to the other. The robot was also able to answer questions related to his own perception (e.g. "*do you see the red cylinder?*"). A detailed description of the experimental setup will be provided in the upcoming chapters.

[3]The toolkit can be downloaded at `http://opendial.googlecode.com`.
[4]cf. `http://www.aldebaran-robotics.com`.

1.3 Outline of the Thesis

We provide here a brief outline of the thesis structure, chapter by chapter.

Chapter 2: Background

This chapter introduces the fundamental concepts and methods used throughout this thesis. We start with an overview of some of the core linguistic properties of dialogue and describe key notions such as turn-taking, dialogue acts and grounding. We then describe the software architectures used to design spoken dialogue systems and the role of each component within them. We also mention a range of important applications for spoken dialogue systems. Finally, we survey the various approaches that have been put forward in the research literature to address the dialogue management problem. In particular, we review both hand-crafted and statistical approaches to the design of dialogue strategies.

Chapter 3: Probabilistic Rules

This chapter lays down the theoretical foundations of our approach. We start by reviewing the core notions of graphical models, since they constitute the formal basis for our framework. We then define what probabilistic rules are and how they are internally structured through conditions and effects. We describe two main types of rules, used to respectively encode probability and utility models. Following this, we explain how the rules are practically instantiated in the Bayesian Network representing the dialogue state. The chapter also addresses some advanced modelling questions, and concludes by discussing related work that also aimed at reducing the dimensionality problem when learning dialogue strategies.

Chapter 4: Learning from Wizard-of-Oz data

This chapter shows how the parameters attached to probabilistic rules can be automatically learned from training data, in a supervised learning fashion. The algorithm to estimate these parameters is grounded in Bayesian inference. To validate our approach, we detail an experiment showing how to learn the utilities of a set of actions from Wizard-of-Oz data collected in a human–robot interaction domain. The experiment illustrates in particular the benefits of applying probabilistic rules.

Chapter 5: Learning from Interactions

This chapter builds upon the previous chapter and extends it to a reinforcement learning context. We show that is is possible to efficiently learn the parameters of dialogue models from observations collected during the interaction itself, without having access to any gold standard annotations. The learning procedure follows a model-based Bayesian reinforcement learning approach. Finally, we report the results of an experiment carried out with a user simulator. The experiment concentrated on the estimation of the transition model in a HRI domain, and evaluated the relative performance of a model structured with probabilistic rules compared to a plain probabilistic model.

Chapter 6: User Evaluation

This chapter presents a user evaluation of our approach in a HRI domain. XXX

Chapter 7: Concluding Remarks

The final chapter concludes this dissertation with a summary of the presented research contributions, followed by an outline of future work.

Chapter 2

Background

We introduce in this chapter the most importance concepts, methods and tools employed in the field of spoken dialogue systems – with special emphasis on dialogue management. We start by laying down the linguistic foundations of our work and some of the key properties of dialogue. A proper understanding of these aspects is indeed a prerequisite for the design of conversationally competent dialogue systems.

2.1 What is spoken dialogue?

We communicate in order to fulfil a wide array of social functions, such as exchanging ideas, recollecting experiences, sustaining relationships, or collaborating with others to accomplish shared goals. These communication skills are developed in early childhood, and our cognitive abilities are in many ways shaped and amplified by this disposition for verbal interaction.

2.1.1 Turn-taking

Turn-taking is one of the most basic (yet often neglected) aspect of spoken dialogue. The physical constraints of the communication channel impose that participants take turns in order to speak. Turn-taking is essentially a resource allocation problem. In this case, the resource to allocate is the conversational floor, and social conventions dictate how the dialogue participants are to take and release their turns.

The field of *conversation analysis* studies what these conventions are, how they vary across cultures and languages, and how they combine to shape conversational behaviours. Human conversations are indeed remarkably efficient at turn-taking. Empirical cross-linguistic studies have shown that the average transition time between turns revolves around 250 ms (**?**).[1] In addition, most of the utterances do not overlap: **?**) argues that less than 5 % of the speech stream contains some form of overlap in spontaneous conversations.

A wide variety of cues are used to detect turn boundaries, such as silence, hesitation markers, syntax (complete grammatical unit), intonation (rising or falling pitch) and body language (**?**).

[1] Interestingly, this duration is shorter than the time required for a human speaker to plan the motor routines associated with the physical act of speaking. This means that the next speaker must start planning his utterance before the current turn is complete, and predict when a potential turn boundary is likely to appear.

These cues can occur jointly or in isolation. Upon reaching a turn boundary, a set of social conventions govern who is allowed to take the turn. The current speaker can explicitly select the next person to take the turn, for instance when greeting someone or asking a directed question. This selection can also occur via other mechanisms such as gaze (?). When no such selection is indicated, other participants are allowed to take the turn. Alternatively, the current speaker can continue to hold the floor until the next boundary.

mixed initiative multi-party dialogue

ping-pong type of interactions Current systems rely on an overly simplistic model of the interaction, where each speaker take discrete turns with noticeable gaps in between, more resembling a walkie-talkie dialogue than a natural conversation.

Bohus paper

2.1.2 Dialogue acts

called communicative acts, speech acts, dialogue move, etc.

Searle taxonomy, inspired by Austin mention non-sentential utterances or ellipsis and references to context (deictics, anaphora, etc.)

prosodic layer

2.1.3 Grounding

talk about feedbacks, common ground, and alignment

dialogue as collaborative activity

interpret each other's utterance cooperatively

reducing the communicative effort

[2]

[3]

2.2 Spoken Dialogue Systems

2.2.1 Architectures

beyond the boundaries of the isolated utterance

mention the question of adaptivity

[2] An elliptical (also called non-sentential) utterance is a linguistic construction that lacks an overt predicate, such as "*where?*", "*at 8 o'clock*", "*a bit less, thanks*" and "*brilliant!*". Their interpretation generally requires access to the dialogue history to recover their intended meaning (?).

[3] A deictic marker is a linguistic reference to an entity that is determined by the context of enunciation. Examples of such markers are "*here*" (spatial reference), "*yesterday*" (temporal reference), "*this mug*" (demonstrative) or "*you*" (reference to a dialogue participant).

2.2.2 Components

2.2.3 Applications

2.3 Dialogue Management

2.3.1 Hand-crafted approaches

Some topics investigated in this paradigm are the semantic and pragmatic interpretation of dialogue moves (**??**), the rhetorical structure of dialogue (**?**), or the use of plan-based reasoning to infer the user intentions (**??**). These approaches are able to provide detailed analyses of various dialogue behaviours, but they generally assume complete observability of the dialogue context and provide only a very limited account (if any) of uncertainties.

Finite-state automata, frame-based, logical reasoning, etc.

2.3.2 Statistical approaches

This is typically done by representing the dialogue domain as a Markov Decision Process (MDP) or Partially Observable Markov Decision Process (POMDP) and subsequently estimating the parameters of these models from data (**?**).

related to partial observability (noisy spoken inputs, unknown user intentions) and stochastic action effects (the user behaviour can be hard to predict).

Probabilistic modelling techniques must however face two important challenges. The most pressing issue is the paucity of appropriate data sets. Statistical models often require large amounts of training data to estimate their parameters. Unfortunately, real interaction data is scarce, expensive to acquire, and difficult to transfer from one domain to another. Moreover, many domains exhibit a rich internal structure with multiple tasks to perform, sophisticated user models, and a complex, dynamic context. In such settings, the dialogue system might need to track a large number of variables in the course of the interaction, which quickly leads to a combinatorial explosion of the state space.

MDP and POMDPs

2.4 Summary

Chapter 3

Probabilistic Rules

3.1 Graphical models

3.1.1 Representation

3.1.2 Inference

3.2 Definitions

3.2.1 Conditions

3.2.2 Effects

3.2.3 Parameters

3.3 Types of rules

3.3.1 Probability rules

3.3.2 Utility rules

3.3.3 Examples

3.4 Rule instantiation

3.4.1 Dialogue state

3.4.2 Instantiation algorithm

3.4.3 Pruning mechanisms

3.5 Advanced modelling

3.5.1 Strings, numbers and collections

3.5.2 Quantifiers

3.6 Related work

3.7 Conclusion

Chapter 4

Learning from Wizard-of-Oz data

4.1 Bayesian parameter estimation

4.1.1 Key idea

4.1.2 Parameter priors

talk about simplifying assumptions: we are learning from partial data

4.1.3 Approximate inference

4.2 Estimation of action utilities from Wizard-of-Oz data

4.2.1 Data representation

4.2.2 Integrating the evidence

4.3 Experiments

4.3.1 Wizard-of-Oz data collection

4.3.2 Experimental setup

4.3.3 Empirical results

4.3.4 Analysis

4.4 Conclusion

Chapter 5

Learning from interactions

5.1 Bayesian Reinforcement Learning

5.1.1 Model-free methods

5.1.2 Model-based methods

5.2 Online planning

5.3 Experiments

5.3.1 Wizard-of-Oz data collection

5.3.2 User simulator

5.3.3 Experimental setup

5.3.4 Empirical results

5.3.5 Analysis

5.4 Conclusion

Chapter 6

User evaluation

Chapter 7

Concluding remarks

7.1 Summary of contributions

7.2 Future work

Appendix A

Relevant probability distributions

Uniform distribution

Multinomial distribution

Normal distribution

Dirichlet distribution

Kernel distribution

Should we include the last one?

Appendix B

Domain specification for HRI scenario

Appendix C

The openDial toolkit

Index