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

June 13, 2013

Abstract

TODO

Acknowledgements

Contents

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Contributions . . . 3
1.3 Outline of the Thesis . . . 6

2 Background **9**
2.1 What is spoken dialogue? . . . 9
2.1.1 Turn-taking . . . 10
2.1.2 Dialogue acts . . . 11
2.1.3 Interpretation of dialogue acts . . . 12
2.1.4 Grounding . . . 13
2.2 Spoken Dialogue Systems . . . 15
2.2.1 Architectures . . . 15
2.2.2 Components . . . 18
2.2.3 Applications . . . 21
2.3 Dialogue Management . . . 22
2.3.1 Hand-crafted approaches . . . 22
2.3.2 Statistical approaches . . . 24
2.4 Summary . . . 31

3 Probabilistic Modelling of Dialogue **35**
3.1 Graphical Models . . . 35
3.1.1 Bayesian Networks . . . 35
3.1.2 Reasoning over time . . . 38
3.1.3 Decision problems . . . 39
3.1.4 Inference . . . 41
3.1.5 Learning . . . 44
3.2 Modelling of dialogue domains . . . 47
3.2.1 Dialogue state tracking . . . 47
3.2.2 Action selection . . . 50
3.2.3 User and error simulation . . . 51
3.3 Summary . . . 51

4 Probabilistic Rules **53**
4.1 Definitions . . . 53
4.1.1 Conditions . . . 54
4.1.2 Effects . . . 54
4.1.3 Parameters . . . 54
4.1.4 Rule types . . . 54
4.1.5 Examples . . . 54
4.2 Rule instantiation . . . 54
4.2.1 Dialogue state . . . 54
4.2.2 Instantiation algorithm . . . 54
4.2.3 Pruning mechanisms . . . 54
4.3 Advanced modelling . . . 54
4.3.1 Strings, numbers and collections . . . 54
4.3.2 Quantifiers . . . 54
4.4 Related work . . . 54
4.5 Conclusion . . . 54

5 Learning from Wizard-of-Oz data **55**
5.1 Bayesian parameter estimation . . . 55
5.1.1 Key idea . . . 55
5.1.2 Parameter priors . . . 55
5.1.3 Approximate inference . . . 55
5.2 Estimation of action utilities from Wizard-of-Oz data . . . 55
5.2.1 Data representation . . . 55
5.2.2 Integrating the evidence . . . 55
5.3 Experiments . . . 55
5.3.1 Wizard-of-Oz data collection . . . 55
5.3.2 Experimental setup . . . 55
5.3.3 Empirical results . . . 55
5.3.4 Analysis . . . 55
5.4 Conclusion . . . 55

6 Learning from interactions **57**
6.1 Bayesian Reinforcement Learning . . . 57
6.1.1 Model-free methods . . . 57
6.1.2 Model-based methods . . . 57
6.2 Online planning . . . 57
6.3 Experiments . . . 57
6.3.1 Wizard-of-Oz data collection . . . 57
6.3.2 User simulator . . . 57
6.3.3 Experimental setup . . . 57
6.3.4 Empirical results . . . 57
6.3.5 Analysis . . . 57
6.4 Conclusion . . . 57

7 **User evaluation** 59

8 **Concluding remarks** 61
8.1 Summary of contributions 61
8.2 Future work 61

A **Relevant probability distributions** 63

B **Additional Proofs** 65

C **Domain specification for user trials** 67

D **The openDial toolkit** 69

Bibliography 71

Index 87

Mathematical notations

Probabilistic models:

X	Random variable
$Val(X)$	Range of values for the variable X
$P(X)$	Probability distribution for the random variable X
$P(X_1, ...X_n)$	Joint probability distribution for X_1, ... X_n
$P(X_1, ...X_n \mid Y_1, ...Y_n)$	Conditional probability distribution for X_1, ... X_n given Y_1, ... Y_n
$E(X)$	Expectation of the random variable X

(Partially observable) Markov Decision Processes:

s	Current state
$\mathcal{S}$	Set of possible states
s_t	State at time t
a	System action
$\mathcal{A}$	Set of possible actions
$R(s, a)$	Immediate reward of action a in state s
γ	Discount factor
h	Planning horizon
$V(s)$	Value function for state s (= expected return)
$Q(s, a)$	Action–value function for action a in state s
$\pi(s)$	MDP dialogue policy, defined as a function $\pi : \mathcal{S} \rightarrow \mathcal{A}$
o	Observation
$\mathcal{O}$	Set of possible observations
b	Belief state $b(s) = P(s)$
$\mathcal{B}$	Belief state space (($\|\mathcal{S}\| - 1$)-dimensional simplex)
$V(b)$	Value function for belief state b
$Q(b, a)$	Action–value function for action a in belief state b
$\pi(b)$	POMDP dialogue policy, defined as a function $\pi : \mathcal{B} \rightarrow \mathcal{A}$

Graphical Models:

$(\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z})$	Conditional independence of the variables $\mathbf{X}$ and $\mathbf{Y}$ given $\mathbf{Z}$
$Y \rightarrow X$	Directed edge from the variable Y to the variable X
$parents(X)$	Parents of the variable X such that $\forall Y \in parents(X), Y \rightarrow X$
$P(\mathbf{Q} \mid \mathbf{E} = \mathbf{e})$	Probability query on variables $\mathbf{Q}$ given evidence $\mathbf{E} = \mathbf{e}$

$U(\mathbf{Q} \mid \mathbf{E}=\mathbf{e})$	Utility query on variables $\mathbf{Q}$ given evidence $\mathbf{E}=\mathbf{e}$

Dialogue-specific variables:

u_u	User utterance
$\tilde{u}_u$	ASR recognition hypotheses for user utterance
a_u	User dialogue act
$\tilde{a}_u$	NLU Interpretation hypotheses for the user dialogue act
i_u	User intention
c	Interaction context
a_m	System dialogue act
u_m	System utterance

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 and efficient manner.

Is it possible to exploit this basic fact to develop more user-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, user interaction design becomes increasingly important. User interfaces should offer rich communication capabilities that can unlock the full potential of their applications, yet remain easy to understand and control. One natural way to achieve this goal is to endow computers with a capacity to understand, even in a limited manner, the communication medium that is most intuitive to human beings, namely spoken language.

The ongoing research on *spoken dialogue systems* (SDS) is precisely trying to implement this objective. A spoken dialogue system is a computer system able to converse with humans via everyday spoken language. Such systems are expected to play an ever-increasing role in our interactions with technology. They have a wide range of applications, ranging from voice-enabled mobile applications to navigation assistants, smart home environments, tutoring systems, and (in a not-too-distant future) service robots assisting us in our daily chores.

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 analysis is completed, the system must then decide how to react. In this 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 an appealing strategy to enhance the user interaction experience in many of today's technologies, their practical development can be a de-

[1] Needless to say, the schema hides a great deal of internal complexity. The next chapter describes in more detail the software architectures used to design practical spoken dialogue systems.

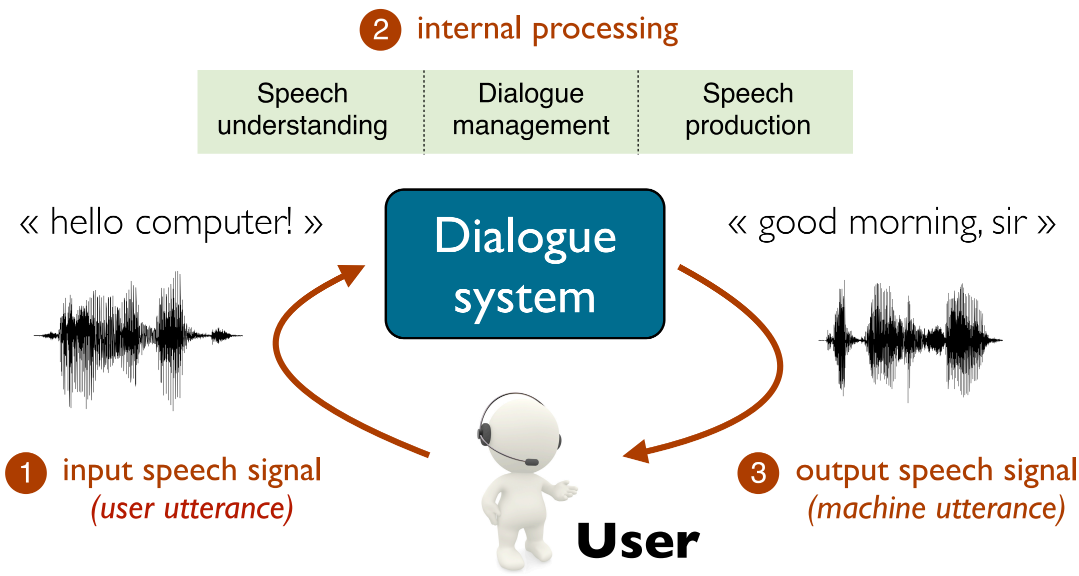

Figure 1.1: Schematic view of a spoken dialogue system.

manding enterprise. Speech is indeed vastly more complex than other 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, at the intersection between speech understanding and production. 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 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) interpreting 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 "*hello computer!*" with another greeting action, "*good morning, sir*".

Along with speech recognition, dialogue management is arguably one of the most difficult processing task in spoken dialogue systems. This difficulty stems from two defining features of verbal interactions:

1. Verbal interactions are *complex*. Taking part in a dialogue requires tracking a multitude of factors, such as the interaction history, the hypothesised goals and preferences of the dialogue participants, and the external situation. These factors depend on one another through multiple relations straddling the linguistic and extra-linguistic boundaries. Everyday utterances are notably rife with elliptical constructions, references and implied content that are only intelligible within the larger conversational context in which they appear. Dialogues are also expected to follow a number of social conventions. Selecting the action that is most appropriate in a particular situation is thus a non-trivial problem.

2. Verbal interactions are also crippled with *uncertainties*. In order to make sense of a given dialogue, 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 resort to sophisticated reasoning in order to interpret the user intentions in their context 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 be erroneous, missing, ambiguous, or fragmentary. This task is known in Artificial Intelligence as *sequential decision-making under uncertainty*, and is a particularly difficult (and often intractable) computational problem, 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 challenges 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 (Cohen and Perrault, 1979; Allen and Perrault, 1980), social obligations (Traum and Allen, 1994), or open questions raised during the interaction (Larsson, 2002; Ginzburg, 2012). These approaches can yield detailed analyses of various conversational behaviours, but they generally assume complete observability of the dialogue state 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 (Roy et al., 2000; Frampton and Lemon, 2009; Young et al., 2010). 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 through a new, hybrid framework for computational dialogue modelling.

1.2 Contributions

The present thesis details an original approach to dialogue management based on *structured probabilistic modelling*. The overarching objective of this work is to design probabilistic models of dialogue that are scalable to rich conversational domains, yet only require limited amounts of training data to estimate their parameters.

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 in dialogue modelling.

The theoretical underpinnings of the thesis are grounded in *probabilistic graphical models* (Koller and Friedman, 2009). Graphical models provide a generic, principled framework for representing and reasoning over complex probabilistic problems. They also come with well-defined data structures and efficient general-purpose algorithms for model estimation and inference. As shown by e.g. Thomson and Young (2010), the dialogue state can be elegantly represented as a Bayesian Network (a well-known type of directed graphical model) factored in a set of state variables describing various aspects of the conversational situation. The complete dialogue state is graphically depicted as a directed acyclic graph where the nodes correspond to particular variables and the edges are conditional dependencies between variables. To exploit such representation for decision-making purposes, the dialogue state can also be extended with action and utility nodes that describe the utility for the agent of performing particular actions in a given situation.

The statistical estimation of such complex probabilistic structures is however a non-trivial task owing to the large number of variables and dependencies involved. The main novelty of our approach is the idea of representing the model distributions in a structured manner through the use of *probabilistic rules*. These rules encode the conditional distributions between variables in terms of structured mappings associating particular conditions defined on a set of input variables to probabilistic effects defined on a set of output variables. The relations between variables are expressed by instantiating the probabilistic rules in the graphical model.

The resulting modelling framework offers two major benefits. Most importantly, the reliance on more expressive representations can drastically reduce the number of parameters associated with the models. Instead of being encoded through traditional probability tables, the conditional distributions between states variables are expressed through high-level rules that capture conditional dependences with a compact 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 integrated in the probabilistic dialogue models. System developers can therefore exploit powerful abstractions to encode their prior knowledge of the dialogue domain in the form of pragmatic rules or task-specific constraints. While the usefulness of such expert information has long been recognised, its exploitation has most often been reduced to a mere external filter to a classical model (Heeman, 2007; Williams, 2008b). By contrast, our approach incorporates such knowledge in the very structure of the statistical model.

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

1. The first experiment, detailed in Section 5.3, focused on the problem of estimating the utili-

ties 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 imitation learning. We were 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 (Lison, 2012b). reread this part afterwards?

2. The second experiment, described in Section 6.3, 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 probabilistic rules. The empirical results demonstrated the benefits of capturing the domain structure with probabilistic rules. The results were first published in (Lison, 2013).

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

An additional contribution of this thesis is a software toolkit that implements all the representations and algorithms presented in this work. The toolkit is called `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. All domain-specific knowledge is declaratively specified in the rules for the domain. The system architecture is therefore reduced to a small set of core algorithms for accessing and updating the dialogue state (Lison, 2012a). This architectural design makes the toolkit fully generic and domain-independent. 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 D.

[2] A Wizard-of-Oz interaction is an experimental procedure borrowed from the field of human-computer interaction (Dahlbäck et al., 1993). 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.

[3] The toolkit can be downloaded at http://opendial.googlecode.com.

We carried out all the experiments described in this thesis in a *human–robot interaction* (HRI) domain. The selection of this application 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 experience significant levels of uncertainty due to imperfect sensors, unreliable motors, and failure-prone speech recognition.[4]

The Nao robot from Aldebaran Robotics was used as a platform for all our experiments.[5] 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. A detailed description of the evaluation setups used in the experiments is provided in the Chapters 4–6.

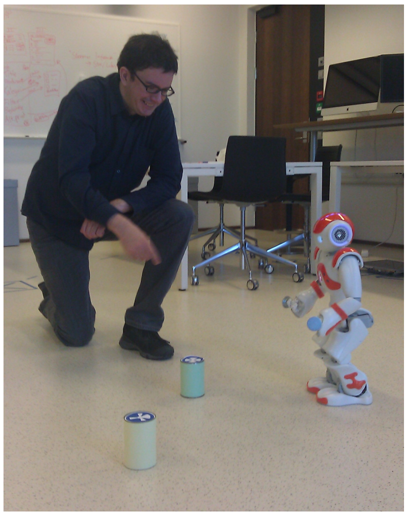

Figure 1.2: Human user interacting with the Nao robot.

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 Modelling of Dialogue

We start by reviewing the core notions of directed graphical models, since they constitute the formal basis for our framework. write this

[4] For practical reasons, the microphones are often placed on the robot itself, at a significant distance from the speaker. This distance between source and receiver is a major degradation factor in speech recognition (Wölfel and McDonough, 2009). Moreover, the microphones are also adjacent to a number of mechanical motors which may disturb the sound signal and lead to spurious detections.

[5] cf. http://www.aldebaran-robotics.com.

Chapter 4: Probabilistic Rules

This chapter lays down the theoretical foundations of our approach. We 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 5: 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 demonstrating 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 6: 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 7: User Evaluation

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

Chapter 8: 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 important concepts and methods employed in the field of spoken dialogue systems, with special emphasis on dialogue management. We start by reviewing some key linguistic concepts that are particularly relevant for our work: turn-taking, dialogue acts and grounding. A proper understanding of these aspects is indeed a prerequisite for the design of conversationally competent dialogue systems. After this linguistic overview, we move to a more technical discussion of the software architectures used to implement practical dialogue systems. These architectures typically comprise multiple processing components, from speech recognition to understanding, dialogue management, output generation and speech synthesis. We briefly describe the role of each component and their positions in the global processing pipeline.

Last but not least, the final section of this background chapter delves into the diverse set of approaches that have been put forward to formalise the dialogue management problem. We first present hand-crafted approaches, starting with finite-state policies and pursuing with more sophisticated methods based on logic- or plan-based reasoning. Finally, we detail the more recently developed statistical approaches to dialogue management that seek to automatically extract dialogue strategies from data, based on supervised and reinforcement learning methods.

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.

One of the most important property of dialogue is that it is fundamentally a *collaborative activity* (emphasis on both terms). It is, first of all, an *activity*, which means that is is (1) motivated by the desire to achieve specific (practical or social) goals; (2) subject to costs to minimise (the communication effort); and (3) composed of a temporal sequence of basic actions. Furthermore, if we abstract from so-called "internal dialogues" with oneself, dialogue involves per definition at least two participants that must act together to keep the dialogue on track. As shown by a wealth of studies in psychology and linguistics (Clark and Schaefer, 1989; Allwood et al., 1992; Clark, 1996; Garrod and Pickering, 2004; Tomasello et al., 2005), human conversations are characterised by a high degree of *collaboration* between interlocutors. The individuals participating in a dialogue routinely collaborate in order to coordinate their contributions and ensure mutual understanding, thereby making the interaction more efficient. This collaboration is done mostly unconsciously and is part and parcel of the conversational skills we develop as speakers of a given language.

We describe in the next sections four major aspects of this collaborative activity:

1. The dialogue participants take *turns* in a conversation.
2. These turns are structured into basic communicative units called *dialogue acts*.
3. The interpretation of these dialogue acts is subordinated to the *conversational context* in which they are uttered.
4. The participants continuously provide *grounding signals* to each other in order to indicate how they understand (or fail to understand) each other's contributions.

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 called 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 and how they combine to shape conversational behaviours in various languages and cultures. 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. (Stivers et al., 2009).[1] In addition, most of the utterances do not overlap: Levinson (1983) indicates 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, as described by Duncan (1972). 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 (Sacks et al., 1974). 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.

Turn-taking is closely related to the notion of *initiative* in human–computer interaction. The vast majority of dialogue systems currently deployed are either system-initiated or user-initiated. In a system-initiated dialogue, the dialogue system has full control on how the interaction is unfolding – i.e. the system is asking all the questions and waiting for the user responses. A user-initiated dialogue is the exact opposite: in such settings, the user is assumed to lead the interaction and request information from the system. The most complex – but also most natural – interaction style is the mixed-initiative, where both the user and the dialogue system are allowed to take the initiative at any time and decide to either provide or solicit information whenever they see fit (Horvitz, 1999).

[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.

The turn-taking behaviour of most current-day dialogue systems remains quite rudimentary. The most common method to detect the end of a user turn is to wait for a silence longer than a manually fixed threshold, typically ranging between ½ and 1.0 second. Some system architectures also include routines for handling barge-ins – that is, user interruptions – (Ström and Seneff, 2000), while others simply ignore them altogether. Turn-taking has recently become a focus of research in its own right in the dialogue system literature (Raux and Eskenazi, 2009; Gravano and Hirschberg, 2011), in an effort to break away from the ping-pong interaction style that characterises most current dialogue interfaces.

2.1.2 Dialogue acts

Each turn is constituted of one or more utterances. As argued by Austin (1962) and Searle (1969), utterances are nearly always purposeful: they have specific goals and are intended to provoke a specific psychological effect on the listener(s). They should therefore best be described as actions rather than abstract statements about the world. The notion of dialogue act embodies precisely this idea.[2] Bunt (1996) defines a dialogue act as a "functional unit of a dialogue used by the speaker to change the context".

In his seminal work on the philosophy of language, Searle (1979) established a taxonomy of speech acts divided in five central categories:

Assertives: Committing the speaker to the truth of a proposition.
Examples: "*I swear I saw him on the crime scene.*", "*I bought more coffee.*"

Directives: Attempts by the speaker to get the addressee to do something.
Examples: "*Clean your room!*", "*Could you post this for me?*"

Commissives: Committing the speaker to some future course of action.
Examples: "*I will deliver this review before Monday.*", "*I promise to work on this.*"

Expressives: Expressing the psychological state of the speaker about a state of affairs.
Examples: "*I am so happy for you!*", "*Apologies for being late.*"

Declaratives: Bringing about a different state of the world by the utterance.
Examples: "*You're fired.*", "*We decided to let you pass this exam.*"

Modern taxonomies of dialogue acts are significantly more detailed than the one introduced by Searle. They also provide detailed accounts of various dialogue-level phenomena such as grounding (cf. next section) that were absent from Searle's analysis. The most well-known annotation scheme is DAMSL (Dialogue Act Markup in Several Layers) and was initially formalised by Core and Allen (1997). DAMSL defines a rich, multi-layered annotation scheme for dialogue acts that is both domain- and task- independent. A modified version of this scheme was applied to annotate

[2] Dialogue acts have gone through multiple names over time, owing to the diverse range of research fields that have studied them, from philosophy to descriptive and computational linguistics. As listed in McTear (2004), alternative denominations include speech acts (Searle, 1969), communicative acts (Allwood, 1976), conversation acts (Traum and Hinkelman, 1992), conversational moves (Sinclair and Coulthard, 1975), and dialogue moves (Larsson et al., 1999).

the Switchboard corpus[3] based on a set of 42 distinct dialogue acts (Jurafsky et al., 1997), including greeting and closing actions, acknowledgements, clarification requests, self-talk, responses, and many more. An interesting aspect of DAMSL is the use of two complementary dimensions in the markup: the *forward-looking functions*, which are the traditional speech acts in Searle's sense (assertions, directives, information requests, etc.) and the *backward-looking functions* that respond back to a previous dialogue act and can signal agreement, understanding, or provide answers. Both backward- and forward-looking functions can be present in the same utterance.

Determining the dialogue act corresponding to a given utterance is a non-trivial operation. The type of utterance only gives a partial indication of the underlying dialogue act – a question can for instance express a directive ("*Could you post this for me?*"). In order to accurately classify a dialogue act, a variety of linguistic factors must be taken into account, such as prosody, lexical, syntactic and semantic features, and the preceding dialogue history (Jurafsky et al., 1998; Shriberg et al., 1998; Stolcke et al., 2000; Keizer and op den Akker, 2007).

2.1.3 Interpretation of dialogue acts

Dialogue acts are strongly contextual in nature: their precise meaning can often only be comprehended within the particular conversational context that gave rise to them. The successful interpretation of dialogue acts must therefore venture beyond the boundaries of the isolated utterance. We briefly review here three striking aspects of this dependence on context.

Implicatures

As shown by Grice (1989), an important part of the semantics of dialogue acts is not explicitly stated but rather implied from the context. Consider the following constructed example:

A: Is William working today?
B: He has a cold.

In order to retrieve the "suggested" meaning behind B's utterance – namely, that William is probably not working –, one needs to assume that B is cooperative and that his response is therefore relevant to A's question. If an utterance initially seems to deliberately violate this principle, the listener must search for additional hypotheses required to make sense of the dialogue act. Grice (1989) formalised these ideas in terms of a cooperative principle composed of four conversational maxims that are assumed to hold in a natural conversation: the maxim of quality ("be truthful"), the maxim of quantity ("be exactly as informative as required"), the maxim of relation ("be relevant") , and the maxim of manner ("be relevant"). These notions have been further developed by various theorists such as Wilson and Sperber (2002) and Horn and Ward (2008). A computational account of these implicatures (and application to dialogue systems) is provided by Benotti (2010).

Non-sentential utterances

Non-sentential (also called elliptical) utterances are linguistic constructions that lack an overt predicate. They include expressions such as "*where?*", "*at 8 o'clock*", "*a bit less, thanks*" and "*brilliant!*".

[3]The Switchboard corpus is a corpus of spontaneous telephone conversations collected in the early 1990's. It includes about 2430 conversations averaging 6 minutes in length; totalling over 240 hours of recorded speech with native speakers of American English (Godfrey et al., 1992).

Their interpretation generally requires access to the recent dialogue history to recover their intended meaning. This can lead to ambiguities in the resolution, as illustrated in these examples modified from Fernández et al. (2007):

- A: "When do they open the new station?" $\rightarrow$ B: "Tomorrow" (*short answer*)
- A: "They open the station today" $\rightarrow$ B: "Tomorrow" (*correction*)
- A: "They open the station tomorrow" $\rightarrow$ B: "Tomorrow" (*acknowledgement*)

Various accounts of non-sentential utterances have been proposed, based on e.g. discourse coherence (Schlangen and Lascarides, 2003) or interaction-oriented semantics (Fernández, 2006; Ginzburg, 2012). Machine learning approaches have also been developed (Schlangen, 2005; Fernández et al., 2007).

Referring expressions

Finally, dialogue acts are replete with linguistic expressions that refer to some aspect of the conversational context. These references can be either deictic or anaphoric.

A deictic marker is a 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), "*you*" (reference to a person), or even pointing gestures. By their very definitions, deictic markers refer to different realities depending on the situation in which they are used: a "*here*" uttered in a classroom differs from a "*here*" uttered in the countryside.

In addition, dialogue can also include anaphoric expressions – that is, expressions that refer to an element that has been previously mentioned through the history of the dialogue. An simple example of such anaphoric expression can be seen in the question-answer pair "*Is William working today?*" $\rightarrow$ "*He has a cold*", where the pronoun "*he*" must be resolved to "*William*".

The appropriate processing of deictic and anaphoric expressions is an important question in dialogue systems, and pertains both to the interpretation and production process. Multiple approaches have been pursued, relying on symbolic (Eckert and Strube, 2000) or statistical techniques (Strube and Müller, 2003; Stent and Bangalore, 2010). Researchers have also investigated the integration of salience measures (Kelleher and Van Genabith, 2004), multimodal cues (Frampton et al., 2009; Chen et al., 2011), the processing of spatial referring expressions (Zender et al., 2009) and the incrementality of the resolution process (Schlangen et al., 2009; Poesio and Rieser, 2011).

2.1.4 Grounding

Dialogue acts are executed as part of a larger collaborative activity that requires the active coordination of all conversational partners, i.e. speaker(s) as well as hearer(s). This coordination takes place at various levels. The first and most visible level is the content of the conversational activity. The partners must ensure mutual understanding of each other's contribution, to control that they remain "on the same page". In addition, they also coordinate the process by which the conversational activity moves forward – by signalling that they are attending to the person who currently holds the conversational floor and acknowledging her/his contributions to the dialogue.

As an illustration, consider this short excerpt from a conversation transcribed in the British National Corpus (Burnard, 2000) :

KATHLEEN : How come they can take time off yet you can't?
STEVE : He's been there longer than me.
KATHLEEN : Oh.
STEVE : I can, I might have two holidays now, two days' holiday. ...
KATHLEEN : Well ... I don't get that, me.
STEVE : What?
KATHLEEN : All these two days' holiday and this, you've had Christmas.
STEVE : You get two point summat[4] days per month worked
KATHLEEN : Oh so you should've got them for January? ...
STEVE : right?
KATHLEEN : Yeah.
STEVE : And I worked three month before Christmas so I got six point summat days
KATHLEEN : For Christmas.
STEVE : so then I had all Christmas off.
KATHLEEN : Oh!
Yeah I get it now.
... I thought you got Christmas off like we got Christmas off.
STEVE : No.
You gotta earn them. ...

(http://www.phon.ox.ac.uk/SpokenBNCdata/KCX.html)

We can observe in this short dialogue that the interlocutors constantly rely on the *common ground* of the interaction to move their discussion forward. They regularly check what pieces of information are mutually known and understood (e.g. "*right?*"). They also make use of a variety of signals to indicate when things are properly grounded ("*oh*", "*yeah*", "*I get it*") and when they are not ("*I don't get that*", "*what?*"). This common ground grows as the dialogue unfolds – for instance, the system of holiday entitlement is not initially part of the shared knowledge for both speakers at the onset of the conversation, but becomes so towards the end.

The common ground is defined as the collection of shared knowledge, beliefs and assumptions that is established during an interaction.[5] Each dialogue act is built upon the current common ground and participates in its gradual expansion and refinement. This process is called *grounding*. A variety of feedback mechanisms can be used to this effect. As described by Clark and Schaefer (1989), positive evidence of understanding can be expressed via cues such as:

Continued attention: The hearer shows that he/she continues to attend to the speaker.

Relevant next contribution: The hearer produces a relevant follow-up, as in the answer "*He's been there longer than me*" following the question that precedes it.

[4]"Summat" is slang for "something" in the Yorkshire region.

[5]An information that is part of the common ground for a given group is more than simply known by every member of the group. All group members must also be aware that the information is shared and known by the other members. Formally speaking, a proposition p is part of the common knowledge for a group of agents G when all the agents in G know p, and they also all know that they all know p, and they all know that they all know that they all know p, *ad infinitum*. This definition can be rigorously formalised using the mathematical apparatus of set theory or epistemic logic (Meyer and Van Der Hoek, 2004).

Acknowledgement: The hearer nods or utters a backchannel such as "*mm*", "*uh-uh*", "*yeah*", or an assessment such as "*I see*", "*great*", "*I get it now*".

Demonstration: The hearer demonstrates evidence of understanding by reformulating or completing the speaker utterance.

Display: The hearer reuses part of the previous utterance.

Communication problems can also occur, owing to e.g. misheard or misunderstood utterances. The hearer should in this case provide negative feedback to signal trouble in understanding. A large panel of clarification and repair strategies are available to recover from these communicative failures. These strategies include backchannels ("*mm?*"), confirmations ("*Do you mean that...?*"), requests for disambiguations, invitations to repeat, and tentative corrections.

All in all, these positive and negative signals enable the dialogue participants to dynamically synchronise what the speaker intends to express and what the hearers actually understand. This grounding process operates mostly automatically, without deliberate effort. It is closely related to the concept of interactive alignment that has recently been articulated by Garrod and Pickering (2004, 2009). Humans show a clear tendency to (unconsciously) imitate their conversational partners. In particular, they automatically align their choice of words, a phenomenon called lexical entrainment (Brennan and Clark, 1996). But alignment also occurs on several other levels such as grammatical constructions (Branigan et al., 2000), pronunciation (Pardo, 2006), accents and speech rate (Giles et al., 1991), and even gestures and facial expressions (Bavelas et al., 1986).

A proper treatment of grounding is critical for the development of conversational interfaces. As already mentioned in the introductory chapter, comprehension errors are indeed ubiquitous in spoken dialogue systems. The potential sources of misunderstandings are abundant, from error-prone speech recognition to out-of-domain utterances, unresolved ambiguities, and unexpected user behaviour. Appropriate grounding strategies are crucial to address these pitfalls. Grounding for dialogue systems is an active area of research and important advances have been made regarding the formalisation of rich computational models of grounding (Traum, 1994; Matheson et al., 2000), the generation of clarification requests (Purver, 2004; Rieser and Moore, 2005), the design of human-inspired error handling strategies (Skantze, 2007), the integration of non-verbal cues such as gaze, head nods and attentional focus (Nakano et al., 2003) and the development of incremental grounding mechanisms (Visser et al., 2012).

2.2 Spoken Dialogue Systems

After reviewing some of the core properties of human dialogues, we now discuss how to develop practical computer systems that aim to emulate such type of conversational behaviour. In the previous chapter, Figure 1.1 represented a dialogue system as a black box taking speech inputs from the user and generating spoken responses. Real systems have however a complex internal structure, as we detail in the next pages.

2.2.1 Architectures

Spoken Dialogue Systems (SDS) often take the form of complex software architectures that encompass a wide range of interconnected components. These components are dedicated to various tasks related to speech processing, understanding, reasoning and decision-making. These tasks can be grouped into five major components:

1. *Speech recognition*, in charge of mapping the raw speech signal to a set of recognition hypotheses for the user utterance(s).

2. *Natural language understanding*, in charge of mapping the recognition hypotheses to high-level semantic representations of the dialogue act performed by the user.

3. *Dialogue management*, in charge of interpreting the purpose of the dialogue act in the larger dialogue context and deciding what communicative action to perform (if any).

4. *Natural language generation*, in charge of finding the best linguistic (and extra-linguistic) realization for the selected communicative action.

5. And finally, *text-to-speech synthesis*, in charge of synthesizing an audio signal out of the generated utterance.

Figure 2.1 shows the flow of information for a prototypical spoken dialogue system. It should be noted that many systems rely on additional middleware to act as a "software glue" between the components and handle the information exchange and scheduling of modules (Turunen, 2004; Herzog et al., 2004; Bohus and Rudnicky, 2009; Schlangen et al., 2010).

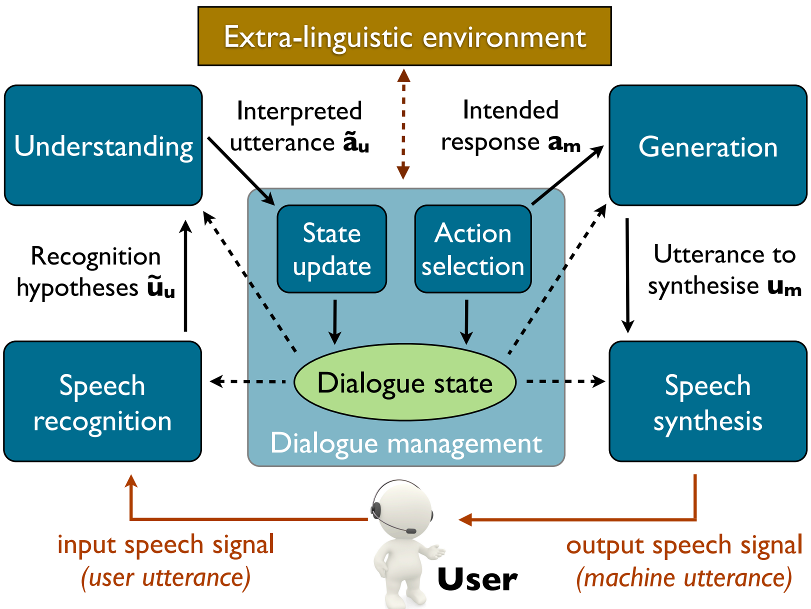

Figure 2.1: Information flow for a typical spoken dialogue system. The solid lines denote necessary input and outputs while the dotted lines represent optional contextual information.

Spoken dialogue systems can rely on other modalities than speech. In particular, additional communication channels such as touch, gestures, gaze, and other body movements can be fruitfully exploited. As shown by eg. Wahlster (2006), multiple modalities can be employed to enrich communication in both directions (understanding and generation). In particular, the system can refine its understanding of the actual user intentions by fusing information perceived through multiple information channels such as gestures (Stiefelhagen et al., 2004) or gaze (Koller et al., 2012). Non-verbal modalities can also be exploited to enhance how information is presented back to the user and convey additional grounding signals, through e.g. facial expressions and gestures. The use of multiple modalities can notably reduce understanding errors and cognitive load (Oviatt et al., 2004) as well as improve the overall user experience (Jokinen and Hurtig, 2006). For all their advantages, multimodal architectures pose however a number of additional challenges related to timing, synchronisation (Salem et al., 2013) and increased system complexity.

In addition to these non-verbal modalities, many dialogue domains are also grounded in an external context that must also be accounted for. This external context might be a physical environment for human-robot interaction (Goodrich and Schultz, 2007), a virtual world for embodied virtual agents (Kopp et al., 2003), a spatial location for in-car navigation systems, or simply a database of factual knowledge for information systems. Contextual factors of relevance for the application must be continuously monitored by the dialogue system (and updated whenever necessary), as many components depend on the availability of such context model for their internal processing. Furthermore, the agent can often actively influence this context through external actions (for instance, a grasping action will modify the location of the gripped object). This contextual awareness necessitates the integration of additional functionalities for perception and actuation. In human–robot interaction domains, these extra-linguistic modules can notably include subsystems for object and scene recognition, spatial navigation, and various motor routines for locomotion and manipulation (Fritsch et al., 2005; Hawes et al., 2007).

Several types of architectures have been proposed to assemble these components in a unified framework. The simplest approach is to arrange the components sequentially in a pipeline starting from speech recognition and ending with speech synthesis. This approach, although relatively straightforward to develop, has a number of shortcomings, amongst which the rigidity of the information flow and the difficulty of inserting feedback loops between components. Pipelines also offer poor turn-taking capabilities, since the system is unable to react before the pipeline has been fully traversed (Raux and Eskenazi, 2009). More advanced architectures – including the one put forward in this thesis – are based on the notion of *information state* (Larsson and Traum, 2000b; Bos et al., 2003). These approaches are essentially blackboard architectures revolving around a central dialogue state that is read and written by various modules connected to it. These modules monitor the state for relevant changes, in which case they trigger their processing routines and update the state with the result. The main advantages of such architectures are (1) a more flexible information flow, since the modules are allowed to process and update information in any order, and (2) the possibility to define modules that take full advantage of the contextual information encoded in the dialogue state. Figure 2.2 provides a graphical illustration of the difference between pipeline and information-state-based architectures.

Finally, a last aspect of dialogue system architectures that has been subject to recent research pertains to *incremental processing*. Many dialogue architectures must wait for an utterance to be fully pronounced to start its interpretation and decide on subsequent actions. This workflow usually leads to poor reactivity and unnatural conversational behaviours. To address this shortcoming, new architectures have been proposed to allow for incremental processing at various stages of interpretation and decision-making (Schlangen and Skantze, 2009).

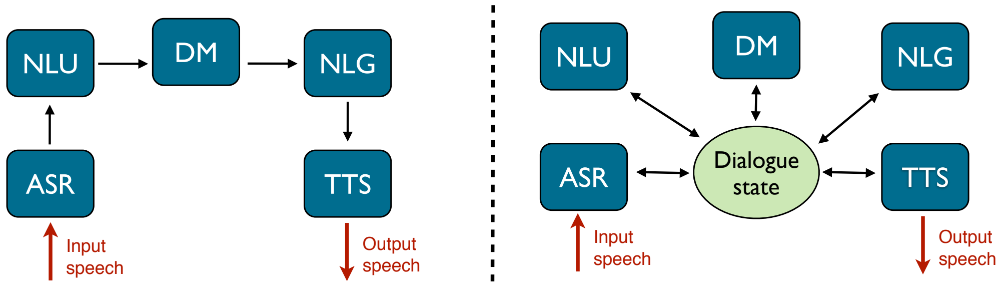

Figure 2.2: Comparison between pipeline (left) and ISU (right) system architectures. Abbreviations: ASR = *Automatic Speech Recognition*, NLU = *Natural Language Understanding*, DM = *Dialogue Management*, NLG = *Natural Language Generation*, and TTS = *Text-to-Speech Synthesis*.

2.2.2 Components

As explained in the previous section, the components of a dialogue systems can typically be grouped in five major steps. We briefly describe here the role of these components and define their respective inputs and outputs.

Speech recognition

Upon detection of a new speech signal emanating from the user, the first task is to recognise the corresponding utterance. Speech recognition is responsible for converting the raw speech signal from the microphone(s) into a set of hypotheses $\tilde{u}_u$ representing the words uttered by the user. To this end, the speech signal is first converted into a digital format and split into short frames (usually 10 ms). A set of acoustic features is then extracted for each frame using signal processing techniques. Once these acoustic features are extracted, two statistical models are combined to estimate the most likely recognition hypotheses: the *acoustic model* and the *language model*.

The acoustic model defines the observation likelihood of particular acoustic features for a given phone[6], while the language model defines the probability of a given sequence of words. This formalisation rests on the representation of the speech recognition task as a *Hidden Markov Model* (HMM), where the states represent the sequence of phones, and the observations are the acoustic features.

For the practical development of spoken dialogue systems, the most important element of a speech recogniser is the language model. The language model effectively represents the set of utterances that can be accepted as inputs to the system (and their relative probabilities). The model can be encoded either in the form of a hand-crafted recognition grammar, or via statistical modelling based on a particular corpus of reference. In the latter case, the language model typically takes the form of an N-gram model, often a bi- or tri-gram corrected with appropriate smoothing and back-off techniques (Jelinek, 1997; Chen and Goodman, 1999). It is also often beneficial to dynamically modify the language model at runtime to reflect the changing context and dialogue state. This real-time model adaptation can notably be realised by priming the words or expressions that are most contextually relevant (Gruenstein et al., 2005; Lison, 2010).

[6] A phone is an individual sound unit of speech. Technically speaking, acoustic models are not defined over entire phones but over sub-segments, typically decomposed into three parts: beginning, middle and end.

The output of the speech recogniser is typically a N-best list (or recognition lattice) representing a set of possible hypotheses for the utterance, together with their relative confidence score or probabilities. Thus, the output of the speech recogniser is a list expressed as:

$$\tilde{u}_u = \langle(\tilde{u}_u^{(1)}, p^{(1)}), (\tilde{u}_u^{(2)}, p^{(2)}), ...(\tilde{u}_u^{(n)}, p^{(n)})\rangle$$

where $\tilde{u}_u^{(i)}$ represents a specific recognition hypothesis and $p^{(i)}$ its corresponding probability.[7]

Natural language understanding

Once the recognition hypotheses for the utterance have been generated by the speech recogniser, the next task is to extract its semantic content. The goal of natural language understanding (NLU) is to build a representation of the meaning(s) expressed by the form of a given utterance. This task is a notoriously difficult endeavour, due to the combination of various factors. The first difficulty lies in speech recognition errors, with WER (Word Error Rates) often revolving around 20 % for many dialogue applications. The syntactic and semantic analysis of dialogue utterances is likewise complicated by the occurrence of sentential fragments and disfluencies of various sorts (e.g. filled pauses, repetitions, corrections), not to mention the ambiguities that must be resolved at various linguistic levels.

Natural language understanding can be decomposed in a number of steps. Parsing corresponds to the task of extracting the syntactic structure of the utterance and mapping it to a semantic representation. Spoken language parsing can be realised through various techniques, from keyword or concept spotting (Komatani et al., 2001; Zhang et al., 2007) to shallow semantic parsing (Coppola et al., 2009), grammar-based parsing (Van Noord et al., 1999) and statistical parsing (He and Young, 2005). It has been shown useful to apply upstream preprocessing techniques to correct speech recognition errors (Ringger and Allen, 1996) and filter out disfluencies (Johnson and Charniak, 2004). In addition, referring expressions might also need to be resolved (Funakoshi et al., 2012). Finally, the dialogue act associated with the utterance must be determined (Stolcke et al., 2000; Keizer and op den Akker, 2007). De Mori et al. (2008) provides a survey of the various models and techniques used in the field of spoken language understanding.

Given speech recognition hypotheses $\tilde{u}_u$ given as inputs, and possibly a representation of the dialogue history and external context, the task of natural language understanding is to extract a corresponding N-best list of dialogue act hypotheses $\tilde{a}_u$ defined as:

$$\tilde{a}_u = \langle(\tilde{a}_u^{(1)}, p^{(1)}), (\tilde{a}_u^{(2)}, p^{(2)}), ...(\tilde{a}_u^{(n)}, p^{(n)})\rangle$$

where $\tilde{a}_u^{(i)}$ represents a dialogue act hypothesis, usually represented in a logical form with various predicates and arguments, and $p^{(i)}$ its corresponding probability.

[7] In order to be proper probabilities, the usual axioms $0 \leq p^{(i)} \leq 1$ for all $p^{(i)}$ and $\sum_{i=1}^{n} p^{(i)} = 1$ must be satisfied. It should also be noted that in practice, many speech recogniser only provide raw confidence scores for their hypotheses. Estimating the exact correspondence between these scores and meaningful probabilities is a non-trivial task that has been investigated by e.g. Williams (2008a).

Dialogue management

Dialogue management occupies a central stage in spoken dialogue systems. As already mentioned in the introductory chapter, dialogue management serves a double role. The first task of the dialogue manager is to maintain a representation of the current dialogue state and update it as new information becomes available. This dialogue state should encode every information that is of general relevance for the dialogue system. The dialogue state can therefore also include parts of the dialogue history (encoded as a temporally ordered sequence of dialogue acts performed by the conversational partners), the current conversational floor, the status of the task(s) to fulfil, and various features describing the context of the interaction. Furthermore, the dialogue state can also include information that is indirectly inferred from the individual observations provided by the other modules. In particular, many dialogue systems include a variable that explicitly encode the hypothesised user intention. This user intention, although never directly observed, can often be derived from the user inputs through a sequence of reasoning steps. Similarly, the dialogue state can also define features that attempt to characterise the user and her/his preferences. Depending on the theoretical premises chosen by the system designer, the dialogue state can either be encoded as a fully observable data structure, or can be extended to explicitly represent partial observability through the definition of probability distributions on the values of the state variables.

The second task of dialogue management is to make decisions based on this dialogue state. This task is often called *action selection*. The decision pertains to the next action to perform by the system, and can be a communicative action (e.g. a piece of information to communicate, a question to task, a grounding signal to convey), an external action (e.g. a physical movement for a robot or a database manipulation for a booking system), a combination of the two, or no action at all. The action selection process can take many forms, from the application of rules to the use of offline or online planning techniques.

The dialogue management step leads to two distinct outcomes: (1) an updated dialogue state s' that reflects the observations received as inputs (user dialogue acts, contextual changes etc.), and (2) a selected system action denoted as a_m (the m subscript standing for "machine" to distinguish it from the user act a_u). Similarly to the user dialogue act a_u, the system action a_m is often encoded in a logical form with predicates and arguments.

Section 2.3 describes in more detail the various approaches and techniques that have been proposed in the literature to tackle the dialogue management problem.

Generation

Assuming the selected system action a_m relates to a communicative action, the following step is to find the best linguistic realisation for the abstract goal defined in a_m. As for natural language understanding, a variety of techniques can be adopted, from shallow generation strategies based on canned sentences or templates to more more sophisticated approaches based on sentence planning and surface realisation (Stone et al., 2003; Koller and Stone, 2007). More recently, statistical methods have also been pursued in enhance the robustness and user-adaptivity of the generation algorithms (Rieser and Lemon, 2010a; Dethlefs and Cuayáhuitl, 2011).

The inputs of the generation module are the selected system action a_m and optionally the features defined in the dialogue state s (e.g. the user model and the external context). Given this information, the generation module will produce a corresponding user utterance denoted u_m. In the case of multimodal systems, the module might also deliver realisations for other modalities than the speech channel, such as gestures or facial expressions.

Speech synthesis

The final step of the processing cycle is to synthesise the utterance in a speech waveform – a process called *text-to-speech synthesis*. This mapping is performed in two consecutive stages. The utterance is first converted into a phonemic representation. This conversion involves various operations related to text normalisation, phonetic and prosodic analysis. Once this conversion is completed, the resulting phonemic representation is fed into a synthesiser in charge of producing the actual waveform. This synthesis can either be performed by gluing together pre-recorded units of speech from a speech database (concatenative synthesis) or by generating sounds using explicit acoustic models of the vocal tract (formant and articulatory synthesis). Most current dialogue systems rely on concatenative synthesis, and in particular unit selection (Hunt and Black, 1996).

2.2.3 Applications

Spoken dialogue systems have a wide variety of applications, ranging from academic research prototypes to mature commercial products. The first applications can be found in telephone-based systems for information access and service delivery. A large variety of systems have been developed in this area, for applications as diverse as automated call-routing (Gorin et al., 1997), travel planning (Walker et al., 2001), weather updates (Zue et al., 2000), bus schedule retrieval (Raux et al., 2005) or tourist information (Lemon et al., 2006). The recent emergence of smartphones also led to the development of new voice interfaces for multimodal local search (Ehlen and Johnston, 2013), cross-lingual communication (Xu et al., 2012) and even pedestrian exploration (Janarthanam et al., 2012). Many of these ideas have found their way into commercial products, as evidenced by the success of applications such as Apple's Siri, Nuance's Dragon Go! and Google Now.

Spoken dialogue systems can also be applied in domains where the use of touch interfaces and screens should be avoided because it is impractical or dangerous. This is notably the case for in-car navigation systems where voice interfaces are to be preferred for safety reasons (Hansen et al., 2005; Castronovo et al., 2010). The recent trends towards ubiquitous computing and "ambient" intelligence for smart home environments also offer promising applications of dialogue system technology (Vipperla et al., 2009; López-Cózar and Callejas, 2010).

Spoken dialogue systems are applied to increasingly complex and open-ended interaction domains, where the artificial agent is no longer a mere executor of user commands, but becomes more of a collaborator or intelligent assistant. Conversational interfaces have notably developed in the healthcare sector to monitor – and hopefully improve – the health condition and fitness of patients through interactive dialogues (Bickmore and Giorgino, 2006; Ståhl et al., 2009; Morbini et al., 2012). Substantial research has also been devoted into the development of interactive tutoring assistants in various learning settings (Chi et al., 2011; Dzikovska et al., 2011; Jan et al., 2011; Traum et al., 2012).

Finally, dialogue systems form an integral part of many robotic systems. Robots are deployed in increasingly social environments, such as homes, offices, schools and hospitals. There is therefore a growing need for robots endowed with communicative abilities, amongst which spoken dialogue understanding. Human-robot interaction is an active area of research and has focused on aspects such as situated dialogue processing (Cantrell et al., 2010; Kruijff et al., 2010), adaptivity (Doshi and Roy, 2008), symbol grounding (Roy, 2005; Lemaignan et al., 2012) and multimodal interaction (Stiefelhagen et al., 2004; Salem et al., 2012; Mirnig et al., 2013).

2.3 Dialogue Management

Various approaches have been proposed to formalise the dialogue management problem. Common to virtually all approaches to dialogue management is (1) the representation of the agent's knowledge of the current situation in a data structure called the *dialogue state* and (2) the use of a decision mechanism to select the action to perform in each dialogue state. A wide range of strategies have been proposed to represent, update and act upon this dialogue state. We first describe hand-crafted approaches and then move on to the more recently developed statistical methods.

2.3.1 Hand-crafted approaches

Finite-state automata

The simplest approach to dialogue management relies on finite-state automata (FSA). A finite state automaton is defined by a collection of states and directed edges between them. Decision-making is made possible by associating each state with a specific action to execute at that state. Each edge in the automata is labelled with a condition on the user input that, if satisfied, will update the current state from the edge source to its target. Figure 2.3 illustrates an example of FSA for a simple, system-initiated interaction that takes user directions. If the user response is different from the five expected inputs, the system will ask the user to repeat until a legal input is provided. The system will continue to request directions until the "*stop*" command is uttered.

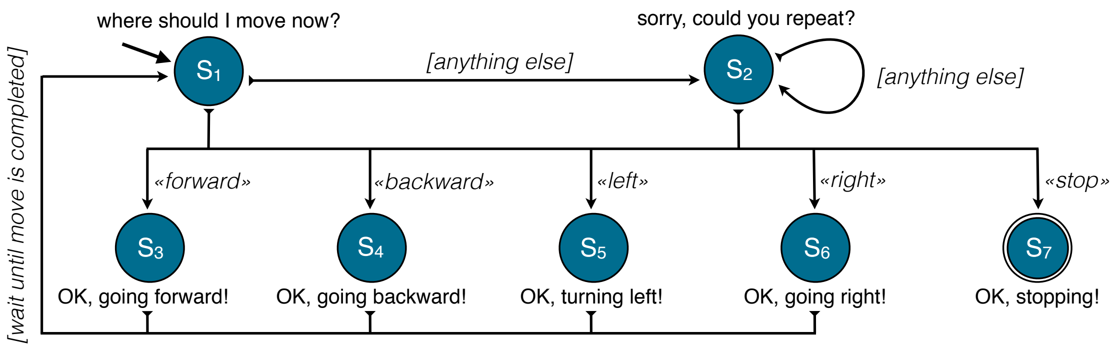

Figure 2.3: Example of finite-state automaton (FSA) for dialogue management, with 7 possible states. The starting state for this FSA is s_1 and the (unique) ending state is s_7. The edges $s_3, \ldots s_7 \rightarrow s_1$ are traversed once the movement is completed, and the two edges $s_1, s_2 \rightarrow s_2$ are traversed for any other user input than the five specified directions.

A finite-state automaton is formally defined as a tuple $\langle \mathcal{S}, \Sigma, \delta, s_0, \mathcal{F} \rangle$, where $\mathcal{S}$ is the set of possible states, Σ the set of inputs that the system can accept (in this case, the user dialogue acts, and possibly external events), $\delta : \mathcal{S} \times \Sigma \rightarrow \mathcal{S}$ the transition function mapping every (state,input) pair to its successor state, s_0 the start state, and $\mathcal{F}$ the set of final states.

Finite-state automata provides a simple and efficient way to design a dialogue manager. Their expressive power is however limited, as the dialogue state of a FSA is represented as a single, atomic symbol, and the possible user moves by a finite enumeration of possible transitions. Finite-state automata are therefore difficult to scale to larger domains where the dialogue state might need to track multiple variables and allow for a large number of user dialogue acts.

Logic- and plan-based approaches

To overcome the rigidity of finite-state automata, richer representations of the dialogue state are required. A popular solution is apply frame-based representations that encode the dialogue state as a frame constituted of a set of slot-value pairs (Seneff and Polifroni, 2000) . Frame-based systems start with an empty frame that is gradually filled by the user inputs. After each user move, a set of production rules define what actions to take – typically, a request to elicit a value for a particular slot – based on the current frame. The process continues until all slots are filled, which mark the completion of the dialogue.

Due to their greater expressivity, frame-based systems offer a number of advantages in terms of domain modelling and dialogue control. They remain however difficult to extend to other domains than classical slot-filling applications (such as flight booking). The *information state* approach (Larsson and Traum, 2000a) is an attempt to provide a more solid theoretical foundation for dialogue management in rich conversational domains. As already mentioned in Section 2.2.1, information state approaches rely on a blackboard architecture where various modules are attached to a central workspace called the information state. This information state is therefore continuously monitored by the modules integrated the dialogue system, and represent the full contextual knowledge available to the agent. In addition to the usual variables describing the dialogue history and the application task, the information state can also incorporate "mentalistic" entities such as the private and shared beliefs of the conversational agents. The information state can exhibit a rich internal structure encoded as attribute-value matrices (AVMs) or typed records (Cooper, 2012).

Upon reception of a relevant input, the dialogue manager modifies this information state using a collection of update rules. In addition to state-internal operations that modify particular variables of the information state, the update rules are also used to derive the actions to execute by the agent. Given a collection of rules and a generic strategy to apply them, the dialogue manager can both update its state and select the next action to perform by way of logical inference. This action selection can notably be grounded on the set of open questions raised and not yet answered during the interaction (Larsson, 2002; Ginzburg, 2012).

Plan-based approaches such as the ones developed by Freedman (2000) and Allen et al. (2001) take one step further. These approaches also rely on complex representations of the dialogue state that notably encompass the belief, desires and intentions (BDI) of each agent (Cohen and Perrault, 1979; Allen and Perrault, 1980). But instead of update rules, classical planning is used to update the state and select the next action. In such settings, both the user and the system are assumed to act in pursuit of their long term goals. The interpretation of the user actions is thus cast as a *plan recognition* problem, where the system seeks to derive the belief, desires and intentions that best explain the observed conversational behaviour of the speaker. Similarly, the selection of system actions is derived from the (task-specific) long term objectives of the system. This search for the best action is an instance of a classical planning task, which can be solved using off-the-shelf planning algorithms. These algorithms require the declaration of a planning domain that specifies the preconditions and effects of every action.

Benefits and limitations of hand-crafted approaches

The hand-crafted approaches to dialogue management we have described are attractive due to their ability to capture rich conversational phenomena. They have also laid the foundations for substantial advances in the semantic and pragmatic interpretation of dialogue moves (Thomason and Stone, 2006; Ginzburg, 2012), the formalisation of social obligations (Traum and Allen, 1994), the rhetorical structure of dialogue (Asher and Lascarides, 2005), or the use of plan-based reasoning to infer the user intentions (Allen and Perrault, 1980; Litman and Allen, 1987). They nevertheless suffer from two notable shortcomings:

1. They assume complete observability of the dialogue context and provide only a very limited account (if any) of uncertainties. This assumption is unfortunately difficult to reconcile with the technological limitations of spoken dialogue systems, where speech recognition errors and out-of-domain utterances are abundant.

2. They require the dialogue domain to be specified by hand, either through the definition of an finite-state automaton, a collection of update rules or a set of action schemas for planning. This requirement is hard to satisfy for many domains, since the behaviour of real users is often challenging to anticipate (unsurprisingly, human behaviour can be difficult to predict) and can deviate significantly from the expectations of the system developers.

Statistical approaches, to which we now turn, have been specifically developed to address these two issues.

2.3.2 Statistical approaches

Common to all statistical approaches to dialogue management is the idea of automatically optimising the dialogue policy (that is, a function associating each possible dialogue state to a system action) from interaction data. Starting from this shared premise, statistical approaches vary along multiple dimensions such as the category of learning algorithm, the representation of the dialogue state and policy, and the type of data on which to estimate the models.

Supervised learning

Many statistical approaches to dialogue management require the collection of so-called "Wizard-of-Oz" data. As already mentioned in the introduction chapter, a Wizard-of-Oz experiment is an interaction in which a human user is asked to interact with a system that is remotely operated by a human agent (without the user being made aware of this control). A hidden wizard is often preferred to a visible human interlocutor, as people tend to behave differently when they talk to a machine or a human person (Jönsson and Dahlbäck, 1988). One can collect multiple interactions of this type and record the wizard decisions at each point, along with their context. Formally, if the gathered interactions are composed of n wizard-selected actions, we can encode the resulting data set as a sequence $\{\langle s_i, a_i \rangle : 1 \leq i \leq n\}$ of state-action pairs, where s_i denotes the dialogue state at time i and a_i the corresponding action selected by the wizard.

The resulting data set can be fed to a supervised learning algorithm in order to construct a dialogue policy that attempts to imitate the conversational behaviour of the wizard. Learning the dialogue policy is thus seen as a classification problem with the state space $\mathcal{S}$ as possible inputs and the action space $\mathcal{A}$ as possible outputs. The goal of the learning algorithm is to construct a classifier $C : \mathcal{S} \rightarrow \mathcal{A}$ that optimises the classification accuracy for the Wizard-of-Oz data set, considering the wizard actions as "gold standards". Various classifiers can be used such as decision trees, Naive Bayes (Williams and Young, 2003), logistic regression (Rieser and Lemon, 2006; Passonneau et al., 2012) or direct maximum likelihood estimation coupled with a distance measure between states (Hurtado et al., 2005).

In a supervised setting, action selection is essentially viewed as a sequence of isolated decision problems. As argued by Levin et al. (2000), this formalisation ignores some important characteristics of conversational behaviour. Dialogue is fundamentally a dynamic process where the state and action at time t have a direct influence on the resulting state at time $t + 1$. This temporal relation between states is typically lost with classical supervised learning approaches. Furthermore, the state space grows exponentially with the number of state variables, and can therefore reach very large sizes. The training data available from a fixed Wizard-of-Oz corpus will therefore only cover a fraction of the state space for the domain. As a consequence, many states encountered at run-time will have no appropriate training examples on which to ground the action selection. Function approximations and abstraction techniques can however be used to mitigate this problem of data sparsity.

Reinforcement learning (with MDPs)

Reinforcement learning (RL) presents an attractive solution to the problem of dialogue policy optimisation. A reinforcement learning problem typically revolves around an *agent* interacting with its environment, typically to perform some practical task. Through its actions, the agent is able to change the state of its environment. After each action, the agent can observe both the new environment state resulting from its actions, as well as a numerical reward encoding the immediate value (positive or negative) of the executed action in relation to the agent's goal.[8] The goal of the learning agent is to find the best action to execute in any given state via a process of trial-and-error – the best action being the one that maximise the agent's expected long-term reward. We provide here a brief introduction to the contral concepts in reinforcement learning, and refer the interested reader to Sutton and Barto (1998) for more details.

Reinforcement learning tasks are based on the definition of a *Markov Decision Process* (MDP), which is a tuple $\langle \mathcal{S}, \mathcal{A}, T, R \rangle$ where:

- $\mathcal{S}$ is the state space of the domain and represents the set of all (mutually exclusive) states.
- $\mathcal{A}$ is the action space and represents the possible actions that can be executed by the agent.
- $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the transition function and encodes the probability $P(s'|s, a)$ of reaching state s' after executing action a in state s.
- $R : \mathcal{S} \times \mathcal{A} \rightarrow \Re$ is the reward value associated with the execution of action a in state s.

[8] Depending on the learning domain, the reward function can either be defined on the state or on the state-action pair. The latter formalisation is most common for dialogue management.

As we can see from the graphical illustration in Figure 2.4, the state at time $t + 1$ is dependent both on the previous state at time t and the action a_t performed by the system. After each action, the system received a reward $r_t = R(s_t, a_t)$ that depends both on the state and selected action.

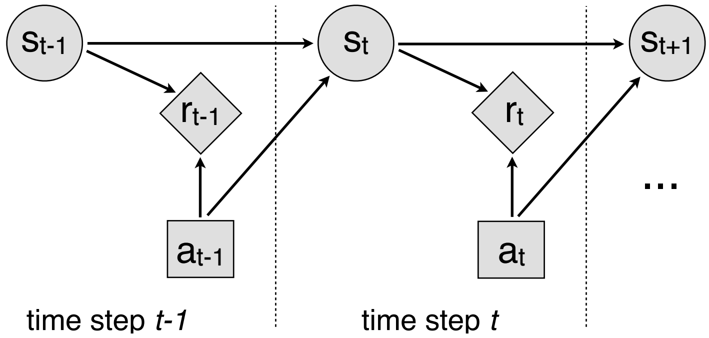

Figure 2.4: Graphical illustration of a Markov Decision Process (MDP) unfolded on a few time steps. By convention, the chance variables (here, the state) are represented with circles, the action variables with squares, and the value variables with diamonds. Greyed entities indicate observed variables (in the MDP case, all variables are observed). Directed edges reflect conditional dependencies between variables. Section 3.1 describes in more detail the mathematical foundations of such graph structure.

Given a particular MDP problem, the goal of the learning agent is to find an optimal policy $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$ that maps each possible state to the best action to execute at that state. The best action is the action that maximises the *expected return* for that state, which is the discounted sum of rewards, starting from the current state up to a potentially infinite horizon. In this sum, a geometric discount factor γ indicates the relative worth of future rewards in regard to present ones, with $0 < \gamma \leq 1$. For a given policy π, the expected return for an arbitrary state s in $\mathcal{S}$ is expressed through the value function $V^\pi(s)$:

$$V^\pi(s) = E\left\{r_0 + \gamma r_1 + \gamma^2 r_2 + ... \;\middle|\; s_0 = s, \pi \text{ is followed}\right\} \tag{2.1}$$

$$= E\left\{\sum_{i=0}^{\infty} \gamma^i r_i \;\middle|\; s_0 = s, \pi \text{ is followed}\right\} \tag{2.2}$$

where $r_i = R(s_i, \pi(s_i))$ is the reward received at time i after performing the action specified by the policy π in state s_i. Equation (2.2) can be rewritten in a recursive form, leading to the well-known Bellman equation (Bellman, 1957):

$$V^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, \pi(s)) V^\pi(s') \tag{2.3}$$

In other words, the expected return in state s equals its immediate reward plus the expected return of its successor state. The recursive definition offered by the Bellman equation is crucial for many reinforcement learning methods, since it allows the value function to be estimated by an iterative process in which the value function is gradually refined until convergence.

Another useful concept is the action-value function $Q^\pi(s, a)$ that expresses the return expected after performing action a in state s and following the policy π afterwards:

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V^{\pi}(s') \tag{2.4}$$

The value and action-value functions are closely related, as $V^{\pi}(s) = \max_a Q^{\pi}(s,a)$.

The objective of the agent is to find a policy π^* such that its expected return $V^{\pi^*}(s) \geq V^{\pi}(s)$ for any state s and policy π. For any given MDP definition, there is at least one policy that satisfies this constraint. To find this optimal policy, many methods do not perform a direct search in policy space but rather seek to estimate the optimal value and action-value functions (respectively denoted V^* and Q^*) via a sequence of updates. Once the iterations have converged to their final values, the optimal policy becomes straightforward to derive:

$$\pi^*(s) = \operatorname*{argmax}_a Q^*(s,a) \tag{2.5}$$

Reinforcement learning methods can be classified in two families. *Model-based* approaches seek to estimate an explicit model of the MDP (in particular the transition probabilities) from collected data and subsequently optimise a policy based on this model. This policy optimisation can be performed with classical dynamic programming techniques (Bertsekas and Tsitsiklis, 1996) or with more advanced Bayesian methods (Dearden et al., 1999). Alternatively, one may adopt a *model-free* approach and skip the estimation of the underlying MDP model in order to directly learn the V^* or Q^* functions from the agent's interaction experience, using techniques such as Q-learning (Watkins and Dayan, 1992), SARSA (Rummery, 1995) or gradient descent (Sutton et al., 2009).

In order to apply reinforcement learning to the problem of optimising dialogue policies, dialogue management must first be cast as a Markov Decision Process. This formalisation can be realised in the following manner:

- The state space $\mathcal{S}$ corresponds the set of possible dialogue states that capture relevant information about the conversational context, such as the local dialogue history and the current status of the task(s) to fulfil.
- The actions space $\mathcal{A}$ corresponds the set of actions that can be executed by the dialogue system.
- The transition function T captures the "dynamics" of the conversation, and indicates how the dialogue state is supposed to change as a result of the system actions (and in particular how the user is expected to respond).
- The reward function R expresses the objectives and costs of the application. A common reward function is to assign a high positive value for completing the task successfully, a high negative value for failing to accomplish the task, and a small negative value for soliciting the user to repeat or clarify her/his intention.

For most dialogue domains, the reward is fixed in advance by the system designer. The transition probabilities are however typically unknown. While it is possible to follow a model-based strategy and learn explicit distributions for the transition probabilities (Walker, 2000; Singh et al., 2002; Tetreault and Litman, 2006), the majority of approaches have adopted model-free techniques. Due to the significant amounts of data necessary to reach convergence, it is often impossible to directly learn the value function from interactions with real users for most practical domains.[9] Instead, most recent approaches have relied on the construction of a user simulator able to generate unlimited numbers of interactions on the basis of which the dialogue system can optimise its policy (Levin et al., 2000; Pietquin, 2008; Frampton and Lemon, 2009). The user simulator can either be designed by hand based on assumptions about the user conversational behaviour (Pietquin and Dutoit, 2006), "bootstrapped" from existing datasets or Wizard-of-Oz studies (Georgila et al., 2006; Rieser and Lemon, 2010b), or generated by modelling the user as a second reinforcement learning agent that optimises its own reward function (English and Heeman, 2005). The reliance on a user simulator for policy optimisation has the major advantage of allowing the learning agent to explore millions of dialogue trajectories on a scale that would be impossible to achieve with real users. As pointed out by several authors, simulated interactions run however the risk of deviating from real user behaviours (Paek, 2006). Finally, it is worth noting that several researchers have attempted to combine the benefits of supervised and reinforcement learning methods, either by initialising a RL algorithm with a policy estimated via supervised methods (Williams and Young, 2003; Rieser and Lemon, 2006), or by expressing the action-value function with a mixture of estimates from supervised and reinforcement learning (Henderson et al., 2008). Hierarchical extensions of the standard MDP model have also been explored (Cuayáhuitl et al., 2010).

Reinforcement learning (with POMDPs)

A limitation faced by MDP approaches is the assumption that the dialogue state is fully observable. As we have frequently noted in this chapter, this assumption does not hold for most dialogue systems, owing to the presence of multiple sources of uncertainty, in particular from speech recognition. An elegant solution to this problem is to extend the MDP framework by allowing the state to be a hidden variable that is indirectly inferred from observations. Such extension gives rise to a *Partially Observable Markov Decision Process* (POMDP). POMDPs are formally defined as tuples $\langle \mathcal{S}, \mathcal{A}, T, R, \mathcal{O}, Z \rangle$. As in a classical MDP, $\mathcal{S}$ represents the state space, $\mathcal{A}$ the action space, T the transition probability $P(s'|s, a)$ between states, and R the reward function $R(s, a)$. However, the actual state is not directly observable anymore. Instead, the process is associated with an observation space $\mathcal{O}$ that expresses the set of possible observations that can be perceived by the system. The function Z then defines the probability $P(o|s)$ of observing o in the current state s. Figure 2.5 provides a graphical illustration of the POMDP framework.

The agent knowledge at a given time is represented by the *belief state* b, which is a probability distribution $P(s)$ over possible states. After each system action a and subsequent observation o, the belief state b is updated to incorporate the new information. This belief update is a simple application of Bayes' theorem:

$$b'(s) = P(s'|b, a, o) = \frac{P(o|s')\ P(s'|b, a)}{P(o|b, a)} \tag{2.6}$$

$$= \alpha\ P(o|s) \sum_s P(s'|s, a)\ b(s) \tag{2.7}$$

[9] This issue can however be partially alleviated through the use of more structured representations for the dialogue models, as we shall see in Chapter 6.

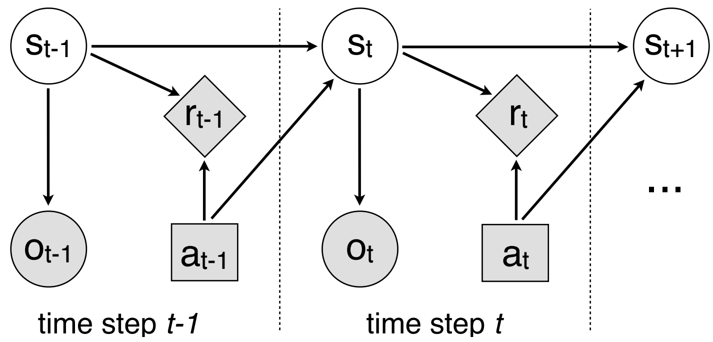

Figure 2.5: Graphical illustration of a Partially Observable Markov Decision Process (POMDP) unfolded on a few time steps. Compared to Figure 2.4, we notice that the state is not directly accessible anymore, but must be inferred from the observations.

where $\alpha = P(o|b, a)$ serves as a normalisation constant and is usually never calculated explicitly.

In the POMDP setting, a policy is a function $\pi = \mathcal{B} \rightarrow \mathcal{A}$ mapping each possible belief state to its optimal action. Mathematically, the belief state space $\mathcal{B}$ is a $(|\mathcal{S}|-1)$-dimensional simplex, which is a continuous, high-dimensional space. The optimisation of the dialogue policy is therefore significantly more complex than for MDPs. The value function V^π for a policy π is the fixed point of Bellman's equation:

$$V^\pi(b) = \sum_{s \in S} R(s, a) b(s) + \gamma \sum_{o \in \mathcal{O}} P(o|b, \pi(b))\ V^\pi(b') \tag{2.8}$$

where b' is the updated belief state following the execution of action $\pi(b)$ and the observation of o, as in Equation (2.7). The optimal value function V^* for finite-horizon problems is known to be piecewise linear and convex in belief space, as proved by Sondik (1971). The value function can therefore be represented by a finite set of vectors, called $\boldsymbol{\alpha}$-vectors. Each vector $\boldsymbol{\alpha}_i$ is associated with a specific action $a(i) \in \mathcal{A}$.[10] The vectors are of size $|\mathcal{S}|$ and $\boldsymbol{\alpha}_i(s)$ is a scalar value representing the value of action $a(i)$ in state s. Given these vectors, the value function simplifies to:

$$V^*(b) = \max_i \boldsymbol{\alpha}_i \cdot b \tag{2.9}$$

And the policy π^* can be rewritten as:

$$\pi^*(b) = a\left(\operatorname*{argmax}_i(\boldsymbol{\alpha}_i \cdot b)\right) \tag{2.10}$$

Extracting the $\boldsymbol{\alpha}$-vectors associated with a POMDP problem is however a computationally difficult task, and exact solutions are intractable beyond toy domains. Efficient approximate solutions have however been recently developed, such as point-based algorithms (Pineau et al., 2003; Kurniawati et al., 2008) and Monte Carlo planning (Silver and Veness, 2010). Deriving the $\boldsymbol{\alpha}$-vectors for a given POMDP ("solving" the POMDP) is known to be a PSPACE-complete problem (Papadimitriou and Tsitsiklis, 1987), which means that the best known algorithms will take time $2^{poly(n,h)}$ to solve a problem with n states and a planning horizon h.

[10]Note that the reverse is not true: each action can be associated with an arbitrary number of vectors.

Modelling a dialogue domain as a POMDP is similar in most respects to the MDP formalisation. The observations in $\mathcal{O}$ typically correspond to the possible N-best lists that can be generated by the speech recogniser and NLU modules, and can also include observations perceived via the other modalities. To reduce the complexity of parameter estimation – and ensure the belief update operation remains tractable – the state s is often factored in distinct variables, as shall be explained in the next chapter.

To extract the dialogue policy from a given POMDP definition, one may rely on POMDP solution methods, as shown by Williams and Young (2007); Williams et al. (2008). Such strategy is however only suitable for relatively small action-state spaces and requires the specification of an explicit observation model.[11] Most recent POMDP approaches have instead focused on the derivation of a dialogue policy from interactions with a user simulator via reinforcement learning. For tractability reasons, many of these approaches have involved the reduction of the full belief state to a simpler representation such as the "summary state" described by Williams and Young (2005). In such setting, the optimal (action-)value function is estimated from direct interactions with the user simulator via techniques such as grid-based discretisations (Young et al., 2010) or function approximation (Thomson and Young, 2010; Daubigney et al., 2012). Finally, non-parametric methods based on Gaussian Processes have recently been proposed (Gašić et al., 2011).

Benefits and limitations of statistical approaches

As stated in the previous sections, one key benefit of statistical approaches is the improved robustness towards errors and unexpected events. This robustness stems primarily from the use of probabilistic reasoning techniques that explicitly account for the uncertainty inherent in spoken dialogue. The second benefit is the possibility to optimise dialogue policies in a principled, data-driven manner based on a generic specification of the system objectives expressed in the reward function. This specification allows the system designer to explicitly encode the various goals and costs of the system. This possibility to represent trade-offs between multiple, sometimes conflicting objectives is one important advantage of reinforcement learning approaches. Empirical studies have shown that automatically optimised policies can outperform hand-crafted strategies in both simulated environments and real user trials, based on objectives and subjective metrics of dialogue success (Lemon and Pietquin, 2007; Young et al., 2013).

Statistical modelling techniques come however with a number of challenges of their own. 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. User simulators can partly alleviate this problem, but they must themselves be bootstrapped from data, and offer no guarantee of producing conversational behaviours that reflect those of real users. The computational complexity of the learning algorithm can also be problematic. Statistical approaches – and especially POMDP-based systems – must often carefully engineer their state and action variables to limit the size of the search space and ensure the learning process remains tractable. Albeit several dimensionality reduction techniques have been proposed in the literature to address this issue (Williams and Young, 2005; Young et al., 2010; Cuayáhuitl et al., 2010; Crook and Lemon, 2011),

[11] Such explicit observation model is often difficult to elicit for dialogue domains since there is an infinite set of possible N-best lists that can be generated.

most work has so far concentrated on slot-filling applications. Domains such as tutoring systems, cognitive assistants and human-robot interaction must however deal with even richer state-action spaces, 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. How to define appropriate statistical models for these open-ended dialogue domains remains an open question, to which the present thesis aims to offer preliminary answers.

Finally, many practical dialogue applications have generic constraints on the dialogue flow. Such constraints may for instance correspond to business rules specific to the particular application. Due to the automatic optimisation mechanism, incorporating such general constraints in a statistically learned dialogue policy can be a complex procedure. As noted by Paek and Pieraccini (2008), this lack of direct control on the final policy is one of the main reasons for the slow adoption of RL approaches in industrial systems. Although some researchers have worked on the integration of expert knowledge into dialogue policy learning (Williams, 2008b; Henderson et al., 2008), much work remains to be done to bring about a unified approach to dialogue management that combines the robustness of data-driven approaches with the expressivity and flexibility of hand-crafted strategies.

Table 2.1 present a comparison of the most important hand-crafted and statistical methods to dialogue management in terms of state representation, account of uncertainty, type of state update and action selection mechanism. The last row also describes how the approach developed in this thesis stands in comparison to these methods.

2.4 Summary

We have presented in this chapter the most important concepts and methods in the area of dialogue management. Starting with a linguistic analysis of the most important dialogue phenomena, we discussed several key aspects of verbal interactions, such as their articulation in sequences of turns and dialogue acts. We also stressed the importance of contextual knowledge in the interpretation and production of dialogue acts, and the role of grounding signals to maintain mutual understanding among the conversational partners.

Section 2.2 described how spoken dialogue systems are practically designed. As we have explained, dialogue systems are often instantiated in complex software architectures that comprise numerous interconnected components for tasks such as speech recognition, understanding, dialogue management, natural language generation and speech synthesis. Dialogue systems can also be extended to handle (i.e. both perceive and act upon) extra-linguistic modalities and environmental factors. The range of possible applications of dialogue system technology is particularly broad and includes domains as varied as mobile applications for information access and service delivery, in-car navigation systems, smart home environments, cognitive assistants, tutoring systems, and social robots.

The last section presented an overview of the dialogue management task. A key concept shared by virtually all approaches to dialogue management is the *dialogue state*: a data structure whose role is to encode the system knowledge of the current conversational situation. This dialogue state can vary greatly in complexity depending on the chosen framework – from the atomic symbols used in finite-state approaches to the rich nested feature structures found in information state formalisms. Based on this dialogue state, an action selection mechanism is then responsible for the selection of

Approach	State representation	State uncertainty	State update mechanism	Action selection mechanism
Finite State Automata	Atomic state	no	Traversal of matching edge	Action associated with node
Frame-based systems (e.g. Seneff and Polifroni, 2000)	Slot/value pairs	no	Slot-filling given user inputs	Production rules
Information state update (e.g. Larsson and Traum, 2000a)	Information state with rich feature structures	no	Update rules	Decision rules
Plan-based systems (e.g. Freedman, 2000; Allen et al., 2001)	Belief-Desire-Intentions [BDI] model	no	Plan recognition and update of BDI model	Classical planning
Supervised approaches (e.g. Hurtado et al., 2005)	Atomic or factored state	no	Extraction of state variables from history and task status	Classifier estimated from Wizard-of-Oz data by supervised learning
MDP-based systems (e.g. Walker, 2000; Levin et al., 2000)	Atomic or factored state	no	Extraction of state variables from history and task status	Policy optimised via reinforcement learning (based on dynamic programming or user simulation)
POMDP-based systems (e.g. Roy et al., 2000; Young et al., 2010)	Atomic or factored state	yes	Belief state update based on Eq. (2.7)	Policy optimised via reinforcement learning or using POMDP solvers
Approach presented in this thesis	Factored state	yes	Structured belief state update (with probabilistic rules)	Policy optimised via (Bayesian) supervised or reinforcement learning

Table 2.1: Comparison of dialogue management approaches.

the next action to execute. In hand-crafted approaches, this mechanism is manually specified by the application developer, either via direct mappings from state to actions, or indirectly through the use of planning techniques. Statistical approaches, on the other hand, seek to automatically optimise dialogue policies from (real or simulated) interaction data. This optimisation can be performed using various learning techniques, from supervised learning on a Wizard-of-Oz data set to reinforcement learning with a user simulator and a general reward function. Reinforcement learning techniques can themselves be divided into MDP approaches, where action effects are stochastic but the dialogue state itself is assumed to be known, and POMDP approaches, which incorporate both stochastic action effects and state uncertainty.

The last section concluded its review of dialogue management approaches by noting that both hand-crafted and statistical methods have significant challenges to address. This is especially striking for open-ended domains such as human-robot interaction, which exhibit both high levels of noise and uncertainty and a rich dialogue context. One of the central claims of this thesis is that these domains are best addressed with a hybrid approach to dialogue management that combines probabilistic modelling with expert knowledge about the domain structure. The next chapters demonstrate how to formalise such modelling approach.

Chapter 3

Probabilistic Modelling of Dialogue

The previous chapter presented a number of hand-crafted and statistical approaches to dialogue management. We noted that statistical approaches necessitate the specification of various probability and utility models in order to track the dialogue state and determine the optimal action to perform at each time. For example, MDPs are defined through a transition function and a reward function, and POMDPs also include an additional observation function.

We have so far ignored the question of how these probability and utility models are internally represented. Given the large state spaces that are common in dialogue management, plain tabular approaches are only tractable for small domains. Fortunately, it is often possible to factor these models into smaller distributions that are easier to estimate and manipulate. *Graphical models* provide a mathematically principled approach to this task. They provide powerful methods for representing, estimating and reasoning over complex probabilistic problems, and are particularly well-suited to the design of efficient, general-purpose algorithms that can exploit their graph-theoretic structure for inference purposes.

The main part of this chapter describes the core properties of (directed) graphical models,[1] their formal representation, and their use in learning and inference tasks. The chapter starts with the most fundamental type of directed graphical model: Bayesian networks. We then show how to extend the formalism to (1) capture temporal sequence and (2) express decision-theoretic problems through actions and utilities. We also briefly describe the most important algorithms for inference and model estimation that have been developed for such graphical models.

The last section of this chapter describes how these representations can be practically applied to dialogue management. We outline the most common factorisations of the dialogue state and review previous work on the use of graphical models for dialogue modelling.

3.1 Graphical Models

3.1.1 Bayesian Networks

Let $\mathbf{X} = X_1 ... X_n$ denote a ordered set of random variables, where each variable X_i is associated with a range of mutually exclusive values. This range can be either discrete (finite or infinite) or continuous. For dialogue models, the range of a variable X_i is typically discrete and can be

[1] There also exist undirected graphical models such as Markov networks, but they will not be discussed nor employed in this thesis.

explicitly enumerated. The enumeration of values for the variable X_i can be written $Val(X_i) = \{x_i^1, ..., x_i^m\}$.

In the general case, the variables $\mathbf{X}$ can contain complex probabilistic dependencies. These dependencies can be expressed through the joint probability distribution $P(X_1...X_n)$. The size of this joint distribution is however exponential in the number n of variables, and is therefore difficult to manipulate (let alone estimate and reason over) directly.

It is fortunately possible to exploit conditional independence properties to reduce the complexity of the joint probability distribution. For three disjoint sets of random variables $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$, we say that $\mathbf{X}$ and $\mathbf{Y}$ are conditionally independent given $\mathbf{Z}$ iff $P(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z}) = P(\mathbf{X} \mid \mathbf{Z}) P(\mathbf{Y} \mid \mathbf{Z})$ for all possible combination of values for $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$. This conditional independence is denoted $(\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z})$.

Conditional independence allows a joint probability distribution to be decomposed into smaller distributions that are much easier to work with. For a variable X_i in $X_1...X_n$, we can define the set $parents(X_i)$ as the minimal set of predecessors of X_i such that the other predecessors of X_i are conditionally independent of X_i given $parents(X_i)$. Note that this set can be empty if the variable X_i is independent of all other variables. This definition enables us to decompose the joint distribution based on the chain rule:

$$
\begin{aligned}
P(X_1, ...X_n) &= \prod_{i=1}^{n} P(X_i \mid X_1, ...X_{i-1}) && (3.1) \\
&= \prod_{i=1}^{n} P(X_i \mid parents(X_i)) && (3.2)
\end{aligned}
$$

This decomposition can be graphically represented in a *Bayesian network*. A Bayesian network is a directed acyclic graph (DAG) where each random variable is represented by a distinct node. These nodes are connected via directed edges that reflect conditional dependencies. In other words, an edge $X_m \rightarrow X_n$ indicates that $X_m \in parents(X_n)$. Each variable X_i in the Bayesian network must be associated with a specific conditional probability distribution $P(X_i \mid parents(X_i))$. Together with the directed graph, the conditional probability distributions (CPDs) fully determine the joint probability distribution of the Bayesian network.

Given such definition, the Bayesian network can be directly used for inference by querying the distribution of a subset of variables, often given some additional evidence. Two operations are especially useful when manipulating probability distributions:

- Marginalisation (also called "summing out"), which derives the probability of the variables X given its conditional distribution $P(X \mid Y)$ and the distribution $P(Y)$:

$$
P(X) = \sum_{y \in Val(Y)} P(X, Y) = \sum_{y \in Val(Y)} P(X \mid Y) P(Y) \tag{3.3}
$$

- Bayes' rule, which reverses the order of a conditional distribution between two variables X and Y (possibly with some background evidence $\mathbf{e}$):

$$
P(X \mid Y, \mathbf{e}) = \frac{P(Y \mid X, \mathbf{e}) P(X \mid \mathbf{e})}{P(Y \mid \mathbf{e})} \tag{3.4}
$$

As an illustration, Figure 3.1 provides an example of Bayesian network that models the probability of occurrence of a fire at a given time. The probability of this event is dependent on the current weather. In addition, two monitoring systems are used to detect possible fires; one on the ground (e.g. via a lookout tower), and one via satellite.

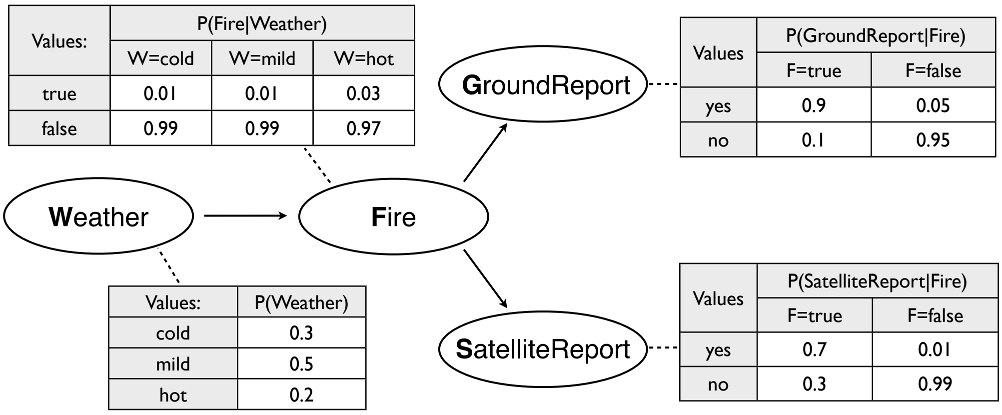

Figure 3.1: Example of Bayesian network with four random variables.

Example 1 Based on the network in Figure 3.1, we can calculate the probability of a fire given that a fire has been reported on the ground but not via satellite, and that the weather is mild.[2] Applying Bayes' rule shown in Equation 3.4, we can derive:

$$P(F \mid W{=}m, G{=}y, S{=}n) = \frac{P(G{=}y, S{=}n \mid F, W{=}m)\, P(F \mid W{=}m)}{P(G{=}y, S{=}n \mid W{=}m)} \quad (3.5)$$

Since $(T \perp A \mid F)$, we can further simplify:

$$= \frac{P(G{=}y \mid F)\, P(S{=}n \mid F)\, P(F \mid W{=}m)}{P(G{=}y, S{=}n \mid W{=}m)} \quad (3.6)$$

As the only function of the denominator is to normalise the final probabilities, it can be replaced by a normalisation constant α:

$$= \alpha\, P(G{=}y \mid F)\, P(S{=}n \mid F)\, P(F \mid W{=}m) \quad (3.7)$$

Given Equation 3.7, the probability of a fire becomes $\alpha \times 0.9 \times 0.3 \times 0.01$, while the probability of the absence of fire is $\alpha \times 0.05 \times 0.99 \times 0.99$. After normalisation, the final probability of a fire is then estimated to be ≈ 0.052. □

[2]Variable names and values are abbreviated to their first letter for better readability.

The distributions shown in Figure 3.1 are called *categorical* distributions.[3] There is one distinct categorical distribution for every combination of values in $parents(X_i)$ – in other words, the model includes a total of 8 distributions. These categorical distributions can be encoded with simple look-up tables that map every possible value in $Val(X_i)$ to a particular probability. Many other representations for discrete CPDs are however conceivable, such as deterministic distributions and distributions based on independence of causal influence (Díez and Druzdzel, 2006).

A Bayesian network can also contain continuous distributions. These distributions are usually encoded with *density functions* represented in a parametric form. A well-known example of parametric distribution is the normal distribution $\mathcal{N}(\mu, \sigma^2)$, which is defined by its two parameters μ and σ^2. Continuous distributions can also expressed with non-parametric methods such as Kernel Density Estimation (KDE). Finally, hybrid models involving both discrete and continuous variables can be defined. The reader is invited to refer to Bishop (2006) and Koller and Friedman (2009) for more details on parametric and non-parametric distributions. Appendix A enumerates the most important discrete and continuous probability distributions used in this thesis.

3.1.2 Reasoning over time

In order to apply Bayesian networks to tasks such as dialogue management, two additional elements are necessary. The first extension, which we cover in this section, is to allow variables to evolve as a function of time. Such temporal dependencies are indeed necessary to account for the dynamic nature of dialogue (the dialogue state is not a static entity and is expected to change over time). Two assumptions are usually made to structure such temporal dependencies:

1. The first assumption, called the Markov assumption, is that the variable values at time t only depend on the previous time slice $t-1$. Formally, let $\mathbf{X}$ be an arbitrary collection of variables. We denote by $\mathbf{X}_t$ the random variables that express their values at time t. The Markov assumption states that $(\mathbf{X}_t \perp \mathbf{X}_{0:(t-2)} \mid \mathbf{X}_{t-1})$.

2. The second assumption is that the process is stationary[4] – that is, that the probability $P(\mathbf{X}_t \mid \mathbf{X}_{t-1})$ is the same for all values of t.

Given these two assumptions, we can define a stochastic process with a probability distribution $P(\mathbf{X}_t \mid \mathbf{X}_{t-1})$ that specifies the distribution of the variables $\mathbf{X}$ at time t given their values at time $t-1$. Such model is called a *dynamic Bayesian network* (DBN). The distribution $P(\mathbf{X}_t \mid \mathbf{X}_{t-1})$ can be internally factored and include dependencies both between the time slices $t-1$ and t and within the slice t. Figure 3.2 shows a concrete example of dynamic Bayesian network. The DBN provides a factored representation of the distribution $P(R_t, F_t, G_t \mid R_{t-1}, F_{t-1})$.

Given the specification of the distribution $P(\mathbf{X}_t \mid \mathbf{X}_{t-1})$ and an initial distribution $P(\mathbf{X}_0)$, a dynamic Bayesian network can be "unrolled" onto multiple time slices. This unrolled model corresponds to a classical Bayesian network.

[3]The categorical distribution is often conflated with the *multinomial* distribution, which specifies the number of times an exclusive event will occur in a repeated independent multinomial trial. A categorical distribution is equivalent to a multinomial distribution for a single observation.

[4]A *stationary* process must be distinguished from a *static* process: a static process is a stochastic process that remains constant for all time steps. In contrast, a stationary process can change over time, but the transition model that describes the dynamics of this process remains constant.

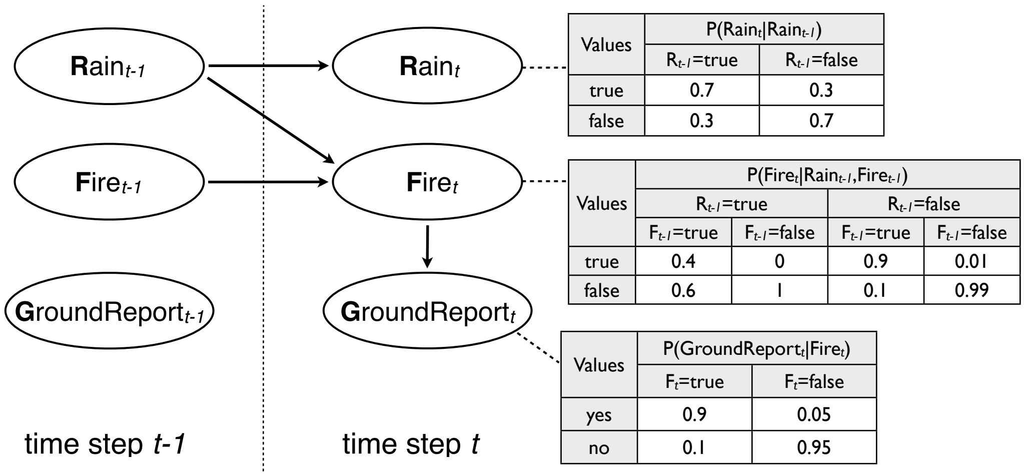

Figure 3.2: Example of dynamic Bayesian network.

Example 2 We can calculate the probability of a fire at time t, given that we know that the probability of a fire at time $(t{-}1 =$ is $P(F_{t-1}) = 0.05$ and no rain was recorded at that time. Using marginalisation, we can derive:

$$P(F_t) = \sum_{w \in Val(F_{t-1})} \sum_{r \in Val(R_{t-1})} P(F_t, F_{t-1}{=}w, R_{t-1}{=}r) \tag{3.8}$$

$$= \sum_{w \in Val(F_{t-1})} P(F_t \mid F_{t-1}{=}w, R_{t-1}{=}n) P(F_{t-1}{=}w) \tag{3.9}$$

Summing up the probabilities for the two possible values of F_{t-1}, Equation 3.9 returns the result $P(F_t = true) = 0.9 \times 0.05 + 0.01 \times 0.95 \approx 0.055$. □

3.1.3 Decision problems

Dynamic Bayesian networks are well-suited to represent temporal processes. However, in sequential decision tasks such as dialogue management, tracking the current state over time is only the first step of the reasoning process. The agent must also be able to subsequently calculate the relative utilities of the various actions that can be selected at that particular state. Graphical models for such decision tasks must therefore explicitly represent the relation between state variables, actions, and utilities.

Dynamic decision networks[5] (DDNs) extend the dynamic Bayesian networks described in the previous section with a representation of action variables and their corresponding utilities. Dynamic decision networks may include three classes of nodes:

1. *Chance nodes* correspond to the classical random variables that have been discussed so far. As for the previous types of graphical models, chance nodes are associated with CPDs that define the relative probabilities of the node values given the values in the parent nodes.

[5]Decision networks are also called influence diagrams.

2. *Decision nodes* (sometimes also called action nodes) correspond to variables that are under the control of the system. The values of these nodes reflect an active choice made by the system to execute particular actions.

3. *Utility nodes* express the utilities (from the system's point of view) associated with particular situations expressed in the node parents. Typically, these parents combine both chance and decision variables. Utility nodes are coupled with utility distributions that associate each combination of values in the node parents with a specific (negative or positive) utility.

Figure 3.3 illustrates an example of dynamic decision network, where an decision variable $AirTanker$ specifies two actions that the system can execute (here, $DropWater$ or $Wait$). The utility variable U encodes the utility function associated with these two actions depending on the current state of the fire – in this case, a large positive value ($+5$) if the water is dropped when a fire occurs, a large negative value (-5) if no water is dropped when a fire occurs, and a small negative value (-1) if the water is dropped in the absence of a fire. It is also worth noting that $Fire_t$ depends on both the $Fire_{t-1}$ and the action variable $AirTanker_{t-1}$. The system is therefore able to actively influence the evolution of the state based on its own actions.

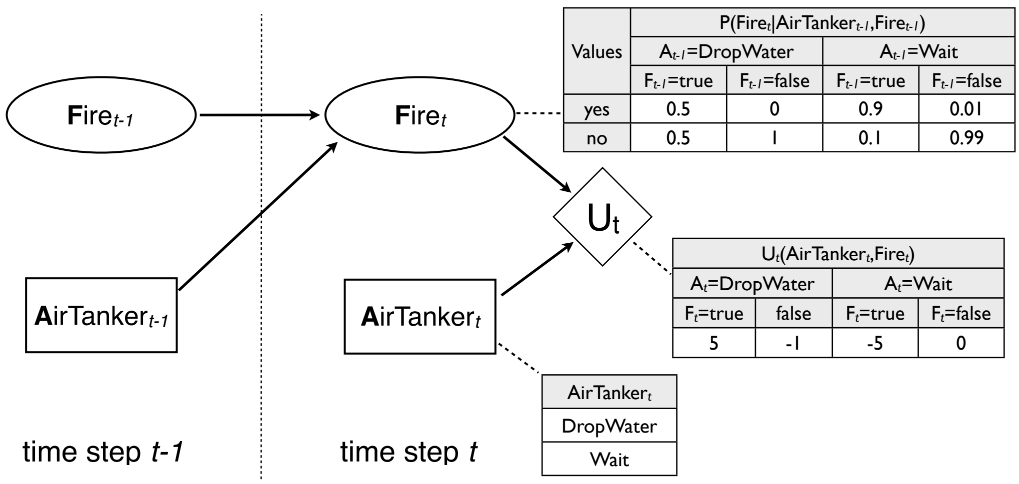

Figure 3.3: Example of dynamic decision network with 2 chance variables ($Fire_{t-1:t}$), two decision variables ($AirTanker_{t-1:t}$) and one utility variable (U_t).

Example 3 Assuming the probability of a fire at time t is $P(Fire_t) = 0.05$, we can calculate the utility of the two actions $DropWater$ and $Wait$ based on the network in Figure 3.3:

$$U_t(A_t) = \sum_{w \in Val(F_t)} U_t(A_t, F_t = w) \tag{3.10}$$

For $A_t = DropWater$, the utility is therefore $0.05 \times 5 - 0.95 = -0.7$, while the utility of $A_t = Wait$ is equal to $0.05 \times (-5) = -0.25$. A rational agent will therefore favour the action $A_t = Wait$ in this situation. □

The MDP (Markov Decision Process) and POMDP (Partially Observable Markov Decision Process) models presented in the previous chapter can be explicitly represented as dynamic decision networks. The key idea is to *factor* the state into distinct variables with possible conditional dependencies between one another. Similarly, action variables can also be split into distinct variables. For a MDP, the state will take the form of a set of variables $\mathbf{S}_t$, and the transition function $P(\mathbf{S}_t \,|\, \mathbf{S}_{t-1}, \mathbf{A}_{t-1})$ be represented as as a dynamic Bayesian network (Boutilier et al., 1999). The reward function can also be encoded as utility variables $\mathbf{R}_t$ connected to relevant sets of state and action variables.[6] The formalisation is similar for POMDPs, with the inclusion of observation variables $\mathbf{O}_t$ connected to the state variables $\mathbf{S}_t$ through conditional dependencies expressing $P(\mathbf{O}_t \,|\, \mathbf{S}_t)$ (Poupart, 2005). At time t, the observation variables $\mathbf{O}_t$ will be observed while the state variables $\mathbf{S}_t$ remain hidden. Figure 3.4 illustrates this factorisation.

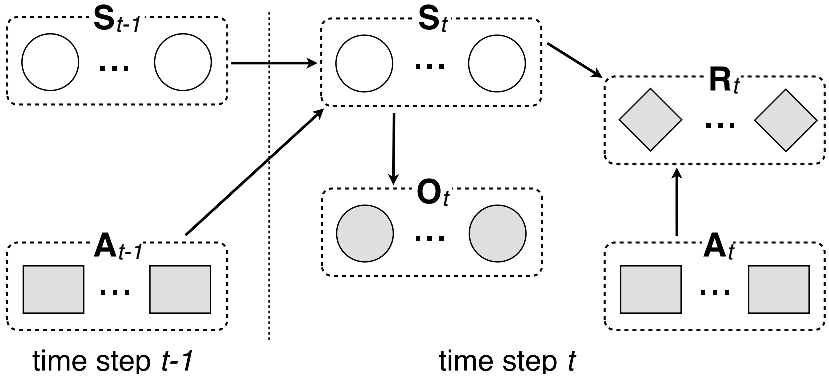

Figure 3.4: Representation of a POMDP as a dynamic decision network with state variables $\mathbf{S}$, action variables $\mathbf{A}$, observations variables $\mathbf{O}$, and reward variables $\mathbf{R}$. Greyed entities denote observed variables (up to time t).

3.1.4 Inference

Generalities

The main purpose of probabilistic graphical models is to evaluate *queries* – that is, calculate a posterior distribution over a subset of variables, given some evidence. Given a graphical model defining the joint probability distribution of a set of variables $\mathbf{X}$, a probability query is a posterior distribution of the form $P(\mathbf{Q} \,|\, \mathbf{E} = \mathbf{e})$, where $\mathbf{Q} \subset \mathbf{X}$ denotes the query variables and $\mathbf{E} \subset \mathbf{X}$ the evidence variables, and $\mathbf{e}$ a possible assignment of values for these variables. If the set of evidence variables is empty, the query is reduced to the calculation of a marginal distribution. The belief update operation mentioned in the previous chapter (Equation 2.7) is an example of such query, with the query variables corresponding to the dialogue state, and the evidence being the new observation. Graphical models augmented with decision and utility variables can also be used to answer utility queries of the form $U(\mathbf{Q} \,|\, \mathbf{E} = \mathbf{e})$. In this case, the query variables often correspond to decision nodes whose utility is to be estimated.

[6]When several utility variables are defined within the same graphical model, the total utility is typically defined as the sum of all utilities (based on the notion of *additive independence*).

A wide range of inference algorithms have been developed to efficiently evaluate these probability and utility queries. These algorithms can be either exact or approximate.

Exact algorithms calculate the precise posterior distribution corresponding to the query through a sequence of manipulation operations on the CPDs contained in the graphical model. One popular algorithm for exact inference is variable elimination (Zhang and Poole, 1996). Variable elimination relies on dynamic programming techniques to evaluate a query through a a sequence of matrix operations (summation and pointwise product). These operations are defined on so-called "factors" that represent CPDs in a matrix format. Variable elimination can be generalised to handle utility queries using joint factors (Koller and Friedman, 2009). Other algorithms for exact inference can be alternatively used, such as message passing on clique trees (Jensen et al., 1990).

Exact inference remains unfortunately difficult to scale to large, densely interconnected graphical models, and approximate techniques are often unavoidable in many practical domains. Algorithms for approximate inference in graphical models include approaches such as loopy belief propagation (Murphy et al., 1999), variational methods (Jordan et al., 1999), and a wide array of sampling techniques (MacKay, 1998), sometimes also called Monte Carlo methods. Popular sampling techniques include various flavours of importance sampling (Fung and Chang, 1989; Cheng and Druzdzel, 2000) and Markov Chain Monte Carlo (MCMC) approaches such as Gibbs sampling (Pearl, 1987; Gamerman and Lopes, 2006). In contrast to other approximation methods, sampling can be straightforwardly applied to arbitrary probabilistic models, and notably hybrid models that combine continuous and discrete variables. They are also guaranteed to converge to the correct results at the large sample limit. Their practical performance can however be difficult to predict for modest sample sizes.

Although the graph-theoretic structure of graphical models enables inference algorithms to be much more effective, it remains an intrinsically difficult computational task. In fact, inference on unconstrained Bayesian Networks is known to be #P-hard, which is a complexity class that is strictly harder than NP-complete problems. This holds both for exact inference (Cooper, 1990), and – perhaps more surprisingly – also for approximate inference (Dagum and Luby, 1993).

The `openDial` toolkit we have developed for this thesis includes two inference algorithms: generalised variable elimination and a specific type of importance sampling algorithm called likelihood weighting (see below). These algorithms are used to update the dialogue state after system actions and upon the reception of new observations, and to select system actions on the basis of this updated dialogue state. Appendix D provides more detail on the technical aspects of this implementation.

Likelihood weighting

To make our discussion of inference frameworks for graphical models more concrete, we describe below a simple but efficient sampling method called *likelihood weighting* (Fung and Chang, 1989), which we have used as inference algorithm for many of the experiments conducted in this thesis. We limit the present discussion to inference in Bayesian networks, but the sampling algorithm can be extended to decision networks in a straightforward manner.

The general intuition behind all sampling algorithms is to estimate the posterior distribution expressed in the query by collecting a large quantity of samples – that is, specific assignments of values to the variables in the model – drawn from the graphical model. Likelihood weighting proceeds by sampling the random variables in the graphical model one by one, in topological order (i.e. from parents to children).[7] For instance, sampling the network in Figure 3.1 will start with the variable $Weather$, then $Fire$ (based on the value drawn for the parent $Weather$), and finally $GroundReport$ and $SatelliteReport$ (based on the value drawn for $Fire$). In order to take into account the evidence $P(\mathbf{Q} \mid \mathbf{E} = \mathbf{e})$, every sample is associated with a specific *weight* that expresses the likelihood of the evidence given the assignment for all the other variables. The pseudocode in Algorithms 1 and 2 (modified from Russell and Norvig (2010); Koller and Friedman (2009)) outline the inference procedure.

Algorithm 1 LIKELIHOOD-WEIGHTING $(\mathcal{B}, \mathbf{Q}, \mathbf{E} = \mathbf{e}, N)$

Input: Bayesian network $\mathcal{B}$ over $\mathbf{X}$
Input: Set of query variables $\mathbf{Q}$
Input: Evidence $\mathbf{E} = \mathbf{e}$
Input: Number N of samples to draw
Output: Posterior distribution $P(\mathbf{Q} \mid \mathbf{E} = \mathbf{e})$ given $\mathcal{B}$

Let $\mathbf{W}$ be a vector of weighted counts for each possible value for $\mathbf{Q}$, initialised to zero
for $i = 1 \rightarrow N$ **do**
  $\mathbf{x}, w \leftarrow$ WEIGHTED-SAMPLE$((\mathcal{B}, \mathbf{E} = \mathbf{e})$
  $\mathbf{q} \leftarrow$ values for $\mathbf{Q}$ in $\mathbf{x}$
  $\mathbf{W}[\mathbf{q}] \leftarrow \mathbf{W}[\mathbf{q}] + w$
end for
Normalise the counts in $\mathbf{W}$
return $\mathbf{W}$

Algorithm 2 WEIGHTED-SAMPLE $(\mathcal{B}, \mathbf{E} = \mathbf{e})$

Let $X_1, ...X_n$ be a topological ordering for $\mathbf{X}$
Initialise sample $\mathbf{x} \leftarrow \langle \mathbf{e} \rangle$
Initialise weight $w \leftarrow 1$
for all $X_i \in X_1, ...X_n$ **do**
  $\mathbf{x}_{parents(X_i)} \leftarrow$ values for $parents(X_i)$ in $\mathbf{x}$
  if $X_i \in \mathbf{E}$ **then**
    $x_i \leftarrow$ value of X_i in $\mathbf{e}$
    $w \leftarrow w \times P(X_i = x_i \mid \mathbf{x}_{parents(X_i)})$
  else
    $x_i \leftarrow$ random sample drawn from $P(X_i \mid \mathbf{x}_{parents(X_i)})$
    $\mathbf{x} \leftarrow \mathbf{x} \cup \langle x_i \rangle$
  end if
end for
return $\mathbf{x}, w$

Example 4 Assume we want to estimate the distribution $P(Fire \mid GroundReport = yes, SatelliteReport = no)$ via likelihood weighting based on the network in Figure 3.1. The following procedure is followed to draw a particular sample:

[7] A partial order on the nodes in the graph can always be found since the network is a directed acyclic graph.

1. The sample is initialised with the provided evidence: $\mathbf{x} \leftarrow \langle \mathit{GroundReport} = \mathit{yes}, \mathit{SatelliteReport} = \mathit{no} \rangle$ and associated weight $w \leftarrow 1$.
2. A value is sampled from $P(\mathit{Weather})$ – for example, mild.
3. A value is sampled from $P(\mathit{Fire} \mid \mathit{Weather} = \mathit{mild})$ – for example, false.
4. The weight is updated to account for the evidence $\mathit{GroundReport} = \mathit{yes}$, leading to $w \leftarrow w \times P(\mathit{GroundReport} = \mathit{yes} \mid \mathit{Fire} = \mathit{false}) = 0.05$.
5. The weight is updated to account for the evidence $\mathit{SatelliteReport} = \mathit{no}$, leading to $w \leftarrow w \times P(\mathit{SatelliteReport} = \mathit{no} \mid \mathit{Fire} = \mathit{false}) = 0.05 \times 0.99 = 0.495$.

The generated sample is therefore $\langle \mathit{Weather} = \mathit{mild}, \mathit{Fire} = \mathit{false}, \mathit{GroundReport} = \mathit{yes}, \mathit{SatelliteReport} = \mathit{no} \rangle$ and is assigned a weight $w = 0.495$. After gathering a large number of such samples, the final distribution is derived by normalising the total weight accumulated for each value of Fire. □

3.1.5 Learning

We have so far pushed aside the question of how the probability distributions in the graphical model are exactly derived. Early approaches often relied on probability distributions elicited from human experts based on plausible or statistical associations they have observed. Although useful in domains where no data is available, hand-crafted models are unfortunately difficult to scale (only models with a limited number of probabilities can be elicited in such manner), and are vulnerable to human errors and inaccuracies. A more principled strategy is therefore to automatically estimate these distributions from experience – that is, via statistical estimation based on a collection of examples in a training set.

Two distinct types of learning tasks can be distinguished. The most common task is *parameter estimation*. Parameter estimation assumes the general structure of the graphical model (i.e. the dependencies between variables) is known, but not the parameters of the individual CPDs. Most discrete and continuous distributions are indeed "parametrised" – that is, they rely on the specification of particular parameters that define the exact shape of the distribution. A categorical distribution on k values has for instance k parameters that assign the relative probability of each outcome. Similarly, a normal distribution $\mathcal{N}(\mu, \sigma^2)$ is governed by its two parameters μ and σ^2.

The second possible learning task is *structure learning*. In structure learning, the agent is required to simultaneously learn both the structure (i.e. the directed edges) and the parameters of the graphical model, given only the list of variables and the training data. This task is significantly more complex than parameter estimation. For dialogue management, the graph structure of the dialogue models can often be designed by the application designer, and we shall therefore concentrate on the parameter estimation problem.

Maximum Likelihood Estimation

The most straightforward estimation method is maximum likelihood estimation (MLE). Maximum likelihood estimation searches for the parameters values that provide the best "fit" for the provided data set. In other words, the parameters will be set to the values that maximise the likelihood of the data set. Given a data set $\mathbf{d}$, a graphical model and a set of parameters $\boldsymbol{\theta}$ to estimate in this model, the MLE learning objective is to find the values $\boldsymbol{\theta}^*$ that maximise the likelihood $P(\mathbf{d} \mid \boldsymbol{\theta})$, often written in logarithmic form:

$$\boldsymbol{\theta}^* = \operatorname*{argmax}_{\boldsymbol{\theta}} P(\mathbf{d} \mid \boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} \log P(\mathbf{d} \mid \boldsymbol{\theta}) \tag{3.11}$$

If the data samples cover the complete set of variables in the model, this likelihood can be neatly decomposed in a set of local likelihoods, one for each CPD (see Koller and Friedman (2009) for a proof). Based on this decomposition, the $\boldsymbol{\theta}^*$ values can be derived in closed-form. For a categorical distribution, the MLE estimates will simply correspond to the relative counts of occurrences in the training data.

Example 5 We can estimate the probability distribution for the variable $Weather$ in Figure 3.1 from data. Assume we have collected 100 samples in which 25 are marked as cold, 47 as mild, and 28 as hot. The MLE method will directly derive from this dataset the probability distribution $P(Weather) = \langle 0.26, 0.47, 0.28 \rangle$. □

The learning problem becomes more complex for partially observed data in which the data samples contain hidden variables. For the Bayesian network in Figure 3.1, an example of partially observed sample is $\langle Weather = mild, GroundReport = yes, SatelliteReport = no \rangle$, where the occurrence of fire is not specified. In such cases, the likelihood function is no longer decomposable and the MLE estimate is not generally amenable to a closed-form solution. Optimisation methods are thus required to find the optimal parameter values. Two common methods for this optimisation are gradient ascent (Binder et al., 1997) and Expectation Maximisation (Green, 1990).

The main drawback of maximum likelihood estimation is its vulnerability to overfitting when learning with small data sets. For instance, if we only had collected one single data point $Weather = cold$ for the previous example, the MLE estimate for the distribution of $P(Weather)$ would be $\langle 1, 0, 0 \rangle$. In other words, maximum likelihood estimation does not take into account any prior knowledge about the relative probability of particular parameter hypotheses, which may lead to very unreasonable estimates for low frequency events.

Bayesian learning

An alternative to maximum likelihood estimation is *Bayesian learning*. The key idea of Bayesian approaches to parameter estimation is to view the CPD parameters as random variables and to derive their posterior distributions after observing the data. Bayesian learning starts with an initial prior over the range of parameter values and gradually refines this distribution through probabilistic inference based on the observation of the samples in the training data.

Each distribution $P(X_i \mid parents(X_i))$ with unknown parameters is therefore associated with a parent node $\theta_{X_i|parents(X_i)}$ that define its parameter distribution. This parameter distribution is often continuous and multivariate. Intuitively, we can think of the variable $\theta_{X_i|parents(X_i)}$ as defining a "distribution over possible distributions".

Based on this formalisation, parameter estimation can be elegantly reduced to a problem of probabilistic inference over the parameters. Given a prior $P(\boldsymbol{\theta})$ on the parameter values and a data set $\mathbf{d}$, the posterior distribution $P(\boldsymbol{\theta} \mid \mathbf{d})$ is given by Bayes' rule:

$$P(\boldsymbol{\theta} \mid \mathbf{d}) = \frac{P(\mathbf{d} \mid \boldsymbol{\theta})\, P(\boldsymbol{\theta})}{P(\mathbf{d})} \tag{3.12}$$

Note that the maximum likelihood estimators described in the previous section coincide with the most probable Bayesian estimator given a uniform prior distribution on the parameters.

The posterior distribution $P(\boldsymbol{\theta} \mid \mathbf{d})$ can be calculated with standard inference algorithms for graphical models as described in the previous section. It is often convenient to encode the distributions of the parameter variables as *conjugate priors* of their associated CPD distribution. In such case, the prior $P(\boldsymbol{\theta})$ and posterior $P(\boldsymbol{\theta} \mid d)$ after observing a data point d are ensured to remain within the same distribution family. In particular, if the distribution of interest is a categorical distribution (such as the variable $Weather$), its parameter distribution can be encoded with a *Dirichlet* distribution, which is known as the conjugate prior of categorical and multinomial distributions. A Dirichlet distribution is a continuous, multivariate distribution of dimension k (with k being the size of the multinomial) that is itself parametrised with so-called *concentration hyperparameters* denoted $\boldsymbol{\alpha} = [\alpha_1, ...\alpha_k]$. Additional details about the formal properties of Dirichlet distributions can be found in Appendix A.

Figure 3.5 illustrates this Bayesian learning approach to parameter estimation for the variable $Weather$. Given that the variable possesses three alternative values, the allowed values for the parameter $\theta_{Weather}$ are three-dimensional vectors $\langle\theta^1_{Weather}, \theta^2_{Weather}, \theta^3_{Weather}\rangle$, with the standard constraints on probability values: $\theta^i_{Weather} \geq 0$ for $i = \{1, 2, 3\}$ and $\theta^1_{Weather} + \theta^2_{Weather} + \theta^3_{Weather} = 1$. As we can observe in the figure, these constraints effectively limit the range of possible values to a 2-dimensional simplex. The $\boldsymbol{\alpha}$ hyperparameters can be intuitively interpreted as "virtual counts" of the number of observations in each category. In Figure 3.5, we can see that the hyperparameters $[5, 10, 5]$ lead to higher probability densities for parameters around the peak $\langle 0.25, 0.5, 0.25\rangle$. As the number of observations increases, the Dirichlet distribution will gradually concentrate on a particular region the parameter space until convergence.

In the case of completely observed data, Bayesian learning over several parameters can be decomposed into independent estimation problems (one for each parameter variable):

$$P(\mathbf{d} \mid \boldsymbol{\theta}) = \prod_{\theta_i \in \boldsymbol{\theta}} P(\mathbf{d} \mid \theta_i) \tag{3.13}$$

As in the maximum likelihood estimation case, the learning task becomes more complicated when dealing with partially observed data, as we can no longer represent the posterior distribution as a product of independent posteriors over each parameter. In this setting, the full posterior is often too complex (and sometimes even multimodal) to be amenable to an analytic solution. Sampling techniques based on importance sampling or MCMC can however be applied to offer reasonable approximations of this posterior. As we shall see in Chapter 5 and 6, the work presented in this thesis is directly grounded in such approximate Bayesian learning methods.

Table 3.1 briefly summarises the parameter estimation methods discussed in this section.

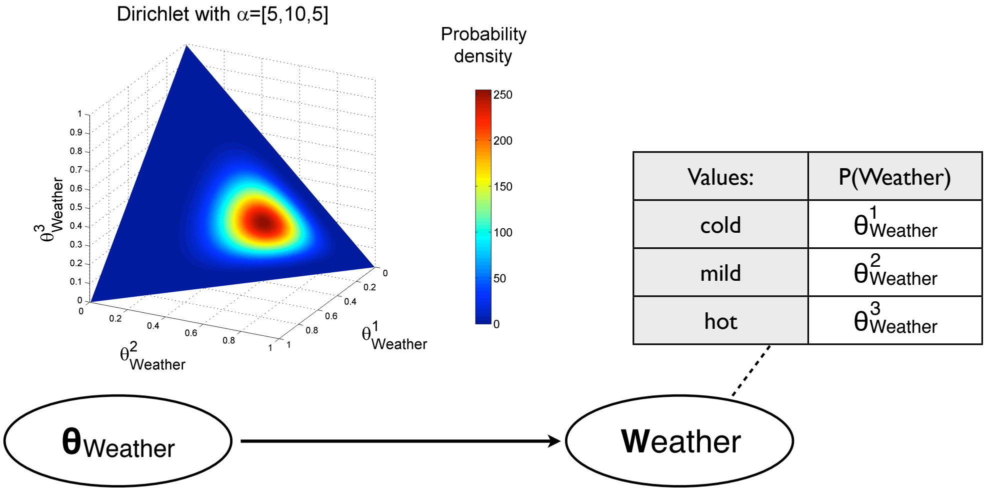

Figure 3.5: Bayesian network with variable $Weather$ and associated parameter $\theta_{Weather}$. As $Weather$ is a categorical distribution, the distribution $P(\theta_{Weather})$ is expressed as a Dirichlet with three dimensions that reflect the relative probabilities for the three values in $Val(Weather)$.

Training data	**Maximum Likelihood Estimation**	**Bayesian Learning**
Fully observed	Maximisation of local likelihood functions	Query on local posterior distribution over each parameter
Partially observed (hidden variables)	Iterative optimisation of global likelihood function	Query on full posterior over parameters via sampling

Table 3.1: Summary of parameter estimation approaches for directed graphical models.

3.2 Modelling of dialogue domains

We describe in this section the various modelling techniques that have been used to structure the state and action spaces in statistical approaches to dialogue management. Some of these techniques are directly grounded in the graphical models we have just detailed, while others rely on other types of internal structures. Section 3.2.1 discusses various formalisations of the dialogue state and (in frameworks that include state uncertainty) the probabilistic models used to update it. Section 3.2.2 details the various utility models that have been employed to represent action selection in dialogue. Finally, Section 3.2.3 takes a closer look at probability models for user and error simulation.

3.2.1 Dialogue state tracking

Dialogue states must often encompass multiple variables to represent various aspects of the conversational situation, such as the local dialogue history, the hypothesised user intention(s) and preferences, and various contextual factors. As already mentioned in the background chapter (Section 2.3.2), supervised and MDP-based approaches to dialogue management assume that this current dialogue state is fully observable, while POMDP-based approaches explicitly account for state uncertainty by viewing the dialogue state as partially observable. We start by reviewing state representations for the fully observable case and then discuss probabilistic modelling methods that account for partial observability.

Without state uncertainty

Supervised and MDP-based approaches to dialogue management encode the dialogue state as a list of feature-value pairs. The dialogue state is therefore factored into a number of independent variables (one for each feature). As the dialogue state is assumed to be fully observed, there is no need to define conditional dependencies between these variables.

Most early approaches adopted simple state representations including only essential information such as the the status of the slots to fill and the last user utterance (Levin et al., 2000; Singh et al., 2000; Scheffler and Young, 2002; Hurtado et al., 2005). The voice-enabled email client described in Walker (2000) include features that capture additional measures related to the overall task progress, history of previous system attempts, confidence thresholds and timing information. There has also been some work on the automatic identification of relevant state variables, using methods from structure learning in decision networks (Paek and Chickering, 2006) and feature selection Tetreault and Litman (2006).

In contract to the above approaches, Henderson et al. (2008) relies on a much larger state space based on rich representations of the conversational context. Inspired by information state approaches to dialogue management, their state space captures detailed information such as complete history of dialogue acts and fine-grained representations of the task status, amounting to a total of 10^386 possible states. Such rich state representations allows the dialogue manager to exploit much broader contextual knowledge in its decision-making. However, it also creates important challenges regarding action selection, as generalisation techniques are necessary to scale up the learning procedure to such large state spaces (cf. Section 3.2.2).

With state uncertainty

POMDP approaches explicitly express state uncertainty through the definition of a belief state b, which is a probability distribution $P(s)$ over possible states. After a system action a in belief state b followed by observation o, the belief state b is updated according to Equation (2.7), repeated here for convenience:

$$b'(s) = \alpha \; P(o|s) \sum_s P(s'|s,a) \; b(s) \tag{2.7}$$

Belief update thus requires the specification of two probabilistic models: the observation model $P(o|s)$ and the transition model $P(s'|s,a)$. In the first application of the POMDP framework to dialogue management, found in the seminal work of Roy et al. (2000), the state is represented by a single variable expressing the user intention, and hand-crafted models were used for the belief update. Zhang et al. (2001) extended the previous approach by introducing a factored state representation based on Bayesian Networks, where the state includes both the user intention and the system state. Williams et al. (2005); Young et al. (2010) further refined this factorisation by decomposing the dialogue state into three distinct variables that respectively represent the last user dialogue act, the user intention and the dialogue history. Bui et al. (2010) added to this factorisation a specific variable for the user's affective state. Finally, Thomson and Young (2010) relied on Bayesian Networks to encode fine-grained dependencies between the various slots expressed in the user intention.

Although these approaches differ in their target domains and internal representations, they often rely on a common factorisation scheme. Many current POMDP approaches to dialogue management factor the state s into (at least) three distinct variables $s = \langle a_u, i_u, c \rangle$, where a_u is the last user dialogue act, i_u the current user intention(s), and c the interaction context and dialogue history. Assuming that the observation o only depends on the last user act a_u, and that a_u depends on both the user intention i_u and the last system action a_m, Equation (2.7) is then rewritten as:

$$b'(a_u, i_u, c) = P(a'_u, i'_u, c' \mid b, a_m, o) \tag{3.14}$$

$$= \alpha \, P(o \mid a'_u) \; P(a'_u \mid i'_u, a_m) \sum_{i_u, c'} P(i'_u \mid i_u, a_m, c') P(c') \; b(i_u) \tag{3.15}$$

The transition model is decomposed in this factorisation into two distinct distributions:

1. The distribution $P(a'_u \mid i'_u, a_m)$ is called the *user action model* and defines the probability of a particular user action given her/his underlying intention and the last system act. It should ideally express the likelihood how the user action a'_u following the system action a_m and the compatibility of a'_u with the user intention i'_u (Young et al., 2010).

2. The distribution $P(i'_u \mid i_u, a_m, c)$ is the *user goal model* and captures how the user intention is likely to change as a result of the context and system actions. This user goal model is strongly domain-specific.

These two distributions are usually derived from collected interaction data with the parameter estimation techniques outlined in Section 3.1.5. A graphical illustration of this state factorisation is shown in Figure 3.6.

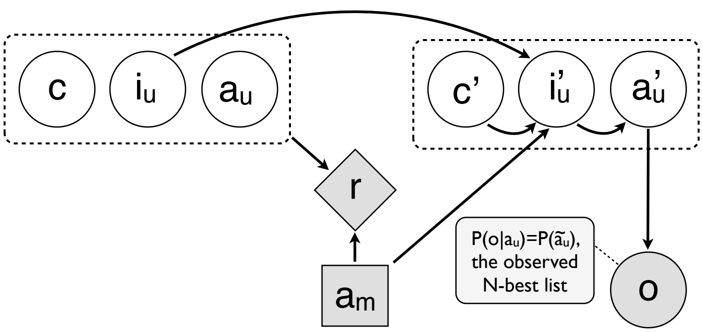

Figure 3.6: Common factoring of the state space for a POMDP-based dialogue system, where c represents the dialogue context, i_u the user intention(s), a_u the user dialogue act, a_m the system act, and o the observed user dialogue acts. The representation omits the conditional dependencies for the variable c', which are contingent on the particular context in place for the domain.

The observation model $P(o \mid a'_u)$ is often rewritten as $P(\tilde{a}_u)$, the dialogue act probability in the N-best list provided by the speech recognition and semantic parsing modules (cf. Section 2.2.2), based on the following approximation:

$$P(o \mid a'_u) = \frac{P(a'_u \mid o)\, P(o)}{P(a'_u)} \approx P(a'_u \mid o) \tag{3.16}$$

Since the probabilities $P(a'_u \mid o)$ are provided at runtime by the ASR and NLU modules, there is no explicitly specified observation set and observation model in such formalisation. This modelling approach has the major advantage of circumventing the statistical estimation of the observation model, a difficult problem since the number of possible N-best lists is theoretically infinite. However,it also means that traditional POMDP solution techniques are unable to apply, since they require an explicit observation set and observation model to extract the $\boldsymbol{\alpha}$-vectors corresponding to the optimal policy (Shani et al., 2013). Most POMDP approaches therefore employ reinforcement learning techniques based on user simulators to optimise dialogue policies. Some researchers have however attempted to estimate explicit observation models. Williams et al. (2008) investigated how to integrate observations that include (continuous-valued) ASR confidence scores into a classical POMDP framework and devised particular density functions for this purpose. Chinaei et al. (2012) showed how an observation model could be empirically estimated from interaction data, based on a simple bag-of-words approach.

Finally, substantial work has been devoted to the inclusion of non-verbal observations and environmental factors into the dialogue state. In the human-robot interaction domain, Prodanov and Drygajlo (2003); Hong et al. (2007) have notably applied Bayesian networks for inferring the underlying user intention based on observations arising from verbal and non-verbal sources.

3.2.2 Action selection

After describing the dialogue state and the different probabilistic models that can be used to update it, we now focus on the search for an optimal dialogue policy. This dialogue policy is typically defined as the one that maximises a given utility model. For reinforcement learning approaches, this utility model is the expected cumulative reward $Q(s, a)$ (or $Q(b, a)$ in the partially observable case). We now detail how this utility model is internally represented and estimated in both model-based and model-free approaches.

Model-based approaches

Paek and Chickering (2006) present an elegant approach to dialogue policy optimisation for MDPs based on dynamic decision networks. The dialogue domain used in their experiments was a command-and control, speech-enabled web browser. Their strategy was to explicitly represent the dialogue management task as a decision network and learn both the structure and parameters of this network from user simulations. The state space initially included all features that could be automatically logged from the interactions. Based on the simulated dialogues, the learning algorithm was able to automatically discover the subset of state variables that were relevant for decision-making as well as the transition probabilities between these variables. They also experimented with various Markov orders to analyse the impact of longer state histories on the system performance. After the estimation of the decision network, dynamic programming techniques are used to extract a dialogue policy that is optimal with respect to the learned models. Given the complexity of the optimisation, the dynamic programming solution was approximated via forward sampling (Kearns, 1999).

Most approaches to action selection assume that the reward model can be encoded in advance by the system designer. Boularias et al. (2010); Chinaei and Chaib-draa (2012) demonstrate an alternative approach based on inverse reinforcement learning (IRL) for POMDPs. Their main idea is to exploit Wizard-of-Oz data for a voice-enabled intelligent wheelchair to automatically infer a reward model. This task of inferring a reward model from expert demonstrations is a prototypical instance of IRL: the agent observes how an expert performs the task and must find the hidden reward model that best explains this behaviour. Inverse reinforcement learning in partially observable domains is however difficult to scale beyond small domains due to the complexity of the optimisation problem (Choi and Kim, 2011). The POMDP models described by Boularias et al. (2010); Chinaei and Chaib-draa (2012) accordingly only include a handful of states and observations and rely on simple probabilistic models without internal structure.

Model-free approaches

linear approximations (Thompson), ways to represent the reward and utilities

3.2.3 User and error simulation

Pietquin and Dutoit (2006) apply Bayesian networks to the statistical estimation of user simulators.

User simulation via inverse reinforcement learning: (Chandramohan et al., 2011).

blabla

As we have discussed in Section 2.3.2, reinforcement learning approaches can be divided into two groups: (1) model-based approaches, which seek to estimate an explicit representation of the (PO)MDP model and subsequently derive a policy based on this model, and (2) model-free approaches, which directly learn the action-value action Q from interaction experience.

This extension allows us to express not only the dialogue state but also the utility of various actions relative to this state. Based on this model, the optimal action to perform can be retrieved by way of generic inference algorithms.

3.3 Summary

Chapter 4

Probabilistic Rules

This chapter spells out the dialogue modelling approach developed in this thesis. As we have seen in the previous chapter, graphical models can help reduce the complexity of probability and utility models by exploiting independence properties between variables. Based on this framework, the state of the dialogue system can be efficiently encoded as a network of interconnected variables. These variables are dynamically updated as a function of the system actions and observations. Graphical models can also easily represent decision-theoretic problems through the inclusion of decision and utility variables. We argued that this generic representation offered a number of theoretical and practical advantages for various learning and inference tasks.

Despite these attractive properties, graphical models do also unfortunately suffer from scalability problems when faced with complex dialogue domains. Conditional dependencies between variables can lead to an rapid increase in the number of distributions included in the model. Alas, only limited amounts of training data are available for most dialogue domains. Estimating the model distributions in such setting is therefore particularly challenging. To address this issue, we introduce in this chapter the notion of *probabilistic rules*, which are structured mappings between conditions and (parametrised) effects. These rules function as *high-level templates* for the construction of the Dynamic Decision Network. The key advantage of such structured modelling approach is the drastic reduction of the number of parameters compared to traditional representations. We also argue that these expressive representations are particularly well suited to encode the probability and utility models used in dialogue management, where substantial amounts of expert knowledge can be exploited to structure the relationships between variables.

The chapter is divided in five sections: Section 4.1 describes in detail how probabilistic rules are defined in terms of conditions and effects and provide some concrete examples of rules for dialogue management. Section 4.2 connects these definitions to the graphical models described in the previous chapter by showing how probabilistic rules are practically instantiated into a Dynamic Decision Network. Finally, Section 4.3 addresses some advanced modelling issues and Section 4.4 relates the approach to previous work.

4.1 Definitions

the rules define a conditional Bayesian network

4.1.1 Conditions

4.1.2 Effects

4.1.3 Parameters

4.1.4 Rule types

4.1.5 Examples

4.2 Rule instantiation

4.2.1 Dialogue state

Our approach is based on information state

4.2.2 Instantiation algorithm

4.2.3 Pruning mechanisms

4.3 Advanced modelling

4.3.1 Strings, numbers and collections

4.3.2 Quantifiers

4.4 Related work

Heriberto's relational state: Cuayáhuitl (2011)

4.5 Conclusion

Chapter 5

Learning from Wizard-of-Oz data

5.1 Bayesian parameter estimation

5.1.1 Key idea

5.1.2 Parameter priors

talk about simplifying assumptions: we are learning from partial data

5.1.3 Approximate inference

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

Sell it as some type of imitation learning?

5.2.1 Data representation

5.2.2 Integrating the evidence

5.3 Experiments

5.3.1 Wizard-of-Oz data collection

5.3.2 Experimental setup

5.3.3 Empirical results

5.3.4 Analysis

5.4 Conclusion

Chapter 6

Learning from interactions

6.1 Bayesian Reinforcement Learning

6.1.1 Model-free methods

6.1.2 Model-based methods

6.2 Online planning

6.3 Experiments

6.3.1 Wizard-of-Oz data collection

6.3.2 User simulator

6.3.3 Experimental setup

6.3.4 Empirical results

6.3.5 Analysis

6.4 Conclusion

Chapter 7

User evaluation

ANOVA?

Chapter 8

Concluding remarks

8.1 Summary of contributions

8.2 Future work

Formally characterise the expressivity of the rules and extend them to handle Ginzburg style update rules?

Try to learn a policy in a fully online fashion with real users, without simulator

do online reinforcement learning with real users and combine imitation+reinforcement learning

Appendix A

Relevant probability distributions

Uniform distribution

Multinomial distribution

Normal distribution

Dirichlet distribution

Kernel distribution

Should we include the last one?

Appendix B

Additional Proofs

chap:proofs

derivation of the MLE estimate for a categorical distribution by deriving the likelihood function. Use Lagrange?

show that the posterior distribution after seeing an example is not a dirichlet anymore, but is still in the exponential family. –> really useful?we are only dealing with partially observed data anyway

Appendix C

Domain specification for user trials

put here a summary of the probabilistic rules applied in the last experiment (user evaluation)

Appendix D

The openDial toolkit

Similarity to Olympus, Jaspis, Ariadne dialogue architectures?

Bibliography

J. Allen, G. Ferguson, and A. Stent. An architecture for more realistic conversational systems. In *Proceedings of the 6th international conference on Intelligent user interfaces (IUI 2001)*, pages 1–8, New York, NY, USA, 2001. ACM.

J. F. Allen and C. R. Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178, 1980.

J. Allwood, J. Nivre, and E. Ahlsén. On the semantics and pragmatics of linguistic feedback. *Journal of Semantics*, 9:1–26, 1992.

J. S Allwood. *Linguistic communication as action and cooperation : a study in pragmatics*. PhD thesis, Dept. of Linguistics, University of Göteborg, 1976.

N. Asher and A. Lascarides. *Logics of Conversation*. Cambridge University Press, Cambridge, 2005.

J. L. Austin. *How to do things with words*. Harvard University Press, Cambridge, Mass., 1962.

J. B Bavelas, A. Black, C. R. Lemery, and J. Mullett. “I show how you feel”: Motor mimicry as a communicative act. *Journal of Personality and Social Psychology*, 50(2):322–329, 1986.

R.E. Bellman. *Dynamic programming*. Princeton University Press, Princeton, NY, 1957.

L. Benotti. *Implicature as an Interactive Process*. PhD thesis, Université Henri Poincaré, Nancy, 2010.

D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1st edition, 1996.

T. W. Bickmore and T. Giorgino. Health dialog systems for patients and consumers. *Journal of Biomedical Informatics*, pages 556–571, 2006.

J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29(2-3):213–244, 1997.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

D. Bohus and A. I. Rudnicky. The RavenClaw dialog management framework: Architecture and systems. *Computer Speech & Language*, 23(3):332–361, July 2009.

J. Bos, E. Klein, O. Lemon, and T. Oka. DIPPER: Description and formalisation of an information-state update dialogue system architecture. In *4th SIGdial Workshop on Discourse and Dialogue*, pages 115–124, 2003.

A. Boularias, H. R. Chinaei, and B. Chaib-draa. Learning the reward model of dialogue pomdps. In *Proceedings of the NIPS Workshop on Machine Learning for Assistive Technology (MLAT 2010)*, 2010.

C. Boutilier, T. Dean, and S. Hanks. Decision-Theoretic Planning: Structural Assumptions and Computational Leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.32.4547.

H. P. Branigan, M. J. Pickering, and A. A. Cleland. Syntactic co-ordination in dialogue. *Cognition*, 75(2):B13–B25, 2000.

S. E. Brennan and H. H. Clark. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology-Learning Memory and Cognition*, 22(6):1482–1493, 1996.

T. H. Bui, J. Zwiers, M. Poel, and A. Nijholt. Affective dialogue management using factored POMDPs. In R. Babuska and F.C.A. Groen, editors, *Interactive Collaborative Information Systems*, volume 281 of *Studies in Computational Intelligence*, pages 209–238. Springer Verlag, March 2010.

H. C. Bunt. Dynamic Interpretation and Dialogue Theory. In M. M. Taylor, F. Néel, and D. G. Bouwhuis, editors, *The Structure of Multimodal Dialogue, Volume 2*. John Benjamins, 1996.

L. Burnard. User reference guide for the british national corpus. Technical report, Oxford University, 2000.

R. Cantrell, M. Scheutz, P. W. Schermerhorn, and X. Wu. Robust spoken instruction understanding for HRI. In *HRI*, pages 275–282. ACM, 2010.

S. Castronovo, A. Mahr, M. Pentcheva, and C. A. Müller. Multimodal dialog in the car: combining speech and turn-and-push dial to control comfort functions. In Takao Kobayashi, Keikichi Hirose, and Satoshi Nakamura, editors, *Proceeding of Interspeech*, pages 510–513. ISCA, 2010.

S. Chandramohan, M. Geist, F. Lefèvre, and O. Pietquin. User simulation in dialogue systems using inverse reinforcement learning. In ISCA (2011), pages 1025–1028.

L. Chen, A. Wang, and B. Di Eugenio. Improving pronominal and deictic co-reference resolution with multi-modal features. In *Proceedings of the SIGDIAL 2011 Conference*, pages 307–311. Association for Computational Linguistics, 2011.

S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393, 1999.

J. Cheng and M. J. Druzdzel. Ais-bn: An adaptive importance sampling algorithm for evidential reasoning in large bayesian networks. *Journal of Artificial Intelligence Research*, 13(1):155–188, 2000.

M. Chi, K. Van Lehn, D. J. Litman, and P. W. Jordan. An evaluation of pedagogical tutorial tactics for a natural language tutoring system: A reinforcement learning approach. *International Journal of Artificial Intelligence in Education*, 21(1-2):83–113, 2011.

H. R. Chinaei and B. Chaib-draa. An inverse reinforcement learning algorithm for partially observable domains with application on healthcare dialogue management. In *ICMLA (1)*, pages 144–149. IEEE, 2012. ISBN 978-1-4673-4651-1.

H. R. Chinaei, B. Chaib-draa, and L. Lamontagne. Learning observation models for dialogue POMDPs. In L. Kosseim and D. Inkpen, editors, *Advances in Artificial Intelligence*, volume 7310 of *Lecture Notes in Computer Science*, pages 280–286. Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-30353-1_24. URL http://dx.doi.org/10.1007/978-3-642-30353-1_24.

J. Choi and K.-E. Kim. Inverse reinforcement learning in partially observable environments. *Journal of Maching Learning Research*, 999999:691–730, July 2011.

H. H. Clark. *Using Language*. Cambridge: Cambridge University Press, 1996.

H. H. Clark and E. F. Schaefer. Contributing to discourse. *Cognitive Science*, 13(2):259–294, 1989.

P. R. Cohen and C. R. Perrault. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3(3):177–212, 1979.

G. F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42(2-3):393–405, 1990.

R. Cooper. Type theory and semantics in flux. In Tim Fernando Ruth Kempson and Nicholas Asher, editors, *Handbook of the Philosophy of Science, Volume 14: Philosophy of Linguistics*. Elsevier, 2012.

B. Coppola, A. Moschitti, and G. Riccardi. Shallow semantic parsing for spoken language understanding. In *Proceedings of Human Language Technologies: The 10th meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2010)*, pages 85–88. Association for Computational Linguistics, 2009.

J. G. Core and J. F. Allen. Coding dialogs with the DAMSL annotation scheme. In *Proceedings of the Working Notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*, Cambridge, MA, November 1997.

P. A. Crook and O. Lemon. Lossless value directed compression of complex user goal states for statistical spoken dialogue systems. In ISCA (2011), pages 1029–1032.

H. Cuayáhuitl. Learning dialogue agents with bayesian relational state representations. In *Proceedings of the IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems (IJCAI-KRPDS)*, pages 9–15. IJCAI, 2011.

H. Cuayáhuitl, S. Renals, O. Lemon, and H. Shimodaira. Evaluation of a hierarchical reinforcement learning spoken dialogue system. *Computer Speech & Language*, 24:395–429, 2010.

P. Dagum and M. Luby. Approximating probabilistic inference in bayesian belief networks is np-hard. *Artificial Intelligence*, 60(1):141 – 153, 1993.

N. Dahlbäck, A. Jönsson, and Lars Ahrenberg. Wizard of oz studies: why and how. In *Proceedings of the 1st International Conference on Intelligent User Interfaces*, pages 193–200. ACM, 1993.

L. Daubigney, M. Geist, and O. Pietquin. Off-policy learning in large-scale POMDP-based dialogue systems. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4989 –4992, 2012.

R. De Mori, F. Bechet, D. Hakkani-Tur, M. McTear, G. Riccardi, and G. Tur. Spoken language understanding. *Signal Processing Magazine, IEEE*, 25(3):50–58, 2008.

R. Dearden, N. Friedman, and D. Andre. Model-based Bayesian Exploration. In K. B. Laskey and H. Prade, editors, *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI 1999)*, pages 150–159. Morgan Kaufmann, 1999.

N. Dethlefs and H. Cuayáhuitl. Hierarchical reinforcement learning and hidden markov models for task-oriented natural language generation. In *ACL (Short Papers)*, pages 654–659. The Association for Computer Linguistics, 2011.

F. J. Díez and M. J. Druzdzel. Canonical probabilistic models for knowledge engineering. Technical Report CISIAD-06-01, UNED, Madrid, Spain, 2006.

F. Doshi and N. Roy. Spoken language interaction with model uncertainty: an adaptive human-robot interaction system. *Connection Science*, 20(4):299–318, 2008.

S. Duncan. Some signals and rules for taking speaking turns in conversations. *Journal of Personality and Social Psychology*, 23:283–292, 1972.

M. O. Dzikovska, A. Isard, P. Bell, J. D. Moore, N. B. Steinhauser, G. E. Campbell, L. S. Taylor, S. Caine, and C. Scott. Adaptive intelligent tutorial dialogue in the BEETLE II system. In *Proceedings of the 15th international conference on Artificial intelligence in education*, AIED'11, pages 621–621, Berlin, Heidelberg, 2011. Springer-Verlag.

M. Eckert and M. Strube. Dialogue acts, synchronizing units, and anaphora resolution. *Journal of Semantics*, 17(1):51–89, 2000.

P. Ehlen and M. Johnston. A multimodal dialogue interface for mobile local search. In J. Kim, J. Nichols, and P. A. Szekely, editors, *IUI Companion*, pages 63–64. ACM, 2013.

M. English and P. Heeman. Learning mixed initiative dialog strategies by using reinforcement learning on both conversants. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 1011–1018, Vancouver, British Columbia, Canada, 2005. Association for Computational Linguistics.

R. Fernández. *Non-Sentential Utterances in Dialogue: Classification, Resolution and Use*. PhD thesis, Ph. D. thesis, King?s College, London. 935, 2006.

R. Fernández, J. Ginzburg, and S. Lappin. Classifying non-sentential utterances in dialogue: A machine learning approach. *Computational Linguistics*, 33(3):397–427, September 2007.

M. Frampton and O. Lemon. Recent research advances in reinforcement learning in spoken dialogue systems. *Knowledge Engineering Review*, 24(4):375–408, 2009.

M. Frampton, R. Fernández, P. Ehlen, M. Christoudias, T. Darrell, and S. Peters. Who is "you"?: combining linguistic and gaze features to resolve second-person references in dialogue. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 273–281. Association for Computational Linguistics, 2009.

R. Freedman. Plan-based dialogue management in a physics tutor. In *Proceedings of the sixth conference on Applied natural language processing*, ANLC '00, pages 52–59. Association for Computational Linguistics, 2000.

J. Fritsch, M. Kleinehagenbrock, A. Haasch, S. Wrede, and G. Sagerer. A flexible infrastructure for the development of a robot companion with extensible hri-capabilities. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, pages 3408–3414, 2005.

K. Funakoshi, M. Nakano, T. Tokunaga, and R. Iida. A unified probabilistic approach to referring expressions. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '12, pages 237–246. Association for Computational Linguistics, 2012.

R. M. Fung and K.-C. Chang. Weighing and integrating evidence for stochastic simulation in bayesian networks. In *Proceedings of the 5th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 209–220, 1989.

D. Gamerman and H. F. Lopes. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*, volume 68. Chapman & Hall/CRC, 2006.

S. Garrod and M. J. Pickering. Joint action, interactive alignment, and dialog. *Topics in Cognitive Science*, 1(2):292–304, 2009.

S. Garrod and M.J. Pickering. Why is conversation so easy? *Trends in Cognitive Sciences*, 8:8–11, 2004.

M. Gašić, F. Jurčíček, B. Thomson, Kai Yu, and S. Young. On-line policy optimisation of spoken dialogue systems via live interaction with human subjects. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 312–317, 2011.

K. Georgila, J. Henderson, and O. Lemon. User simulation for spoken dialogue systems: learning and evaluation. In *Proceedings of the Ninth International Conference on Spoken Language Processing (INTERSPEECH-ICSLP 2006)*, 2006.

H. Giles, N. Coupland, and J. Coupland. 1. accommodation theory: Communication, context, and consequence. *Contexts of accommodation: Developments in applied sociolinguistics*, page 1, 1991.

J. Ginzburg. *The Interactive Stance*. Oxford University Press, New York, 2012.

J. J. Godfrey, E. C. Holliman, and J. McDaniel. SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings of ICASSP*, volume 1, pages 517–520 vol.1, 1992.

M. A. Goodrich and A. C. Schultz. Human-robot interaction: a survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.

A. L. Gorin, G. Riccardi, and J. H. Wright. How may i help you? *Speech Communication*, 23(1-2): 113–127, 1997.

A. Gravano and J. Hirschberg. Turn-taking cues in task-oriented dialogue. *Computer Speech & Language*, 25(3):601 – 634, 2011.

P. J. Green. On use of the EM algorithm for penalized likelihood estimation. *Journal of the Royal Statistical Society*, 52(3):443–452, 1990.

H.P. Grice. *Studies in the Way of Words*. Harvard University Press, 1989.

A. Gruenstein, C. Wang, and S. Seneff. Context-sensitive statistical language modeling. In *Proceedings of 9th European Conference on Speech Communication and Technology (Interspeech 2005)*, pages 17–20, 2005.

J. H. L. Hansen, X. Zhang, M. Akbacak, U.H. Yapanel, B. Pellom, W. Ward, and P. Angkititrakul. CU-move: Advanced in-vehicle speech systems for route navigation. In H. Abut, J. H . L. Hansen, and K. Takeda, editors, *DSP for In-Vehicle and Mobile Systems*, pages 19–45. Springer, 2005.

N. Hawes, A. Sloman, J. L. Wyatt, M. Zillich, H. Jacobsson, G.-J. M. Kruijff, M. Brenner, G. Berginc, and D. Skocaj. Towards an integrated robot with multiple cognitive functions. In *AAAI*, pages 1548–1553. AAAI Press, 2007.

Y. He and S. Young. Semantic processing using the hidden vector state model. *Computer Speech & Language*, 19(1):85 – 106, 2005.

P. Heeman. Combining reinforcement learning with Information-State update rules. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 268–275, 2007.

J. Henderson, O. Lemon, and K. Georgila. Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Computational Linguistics*, 34:487–511, 2008.

G. Herzog, A. Ndiaye, S. Merten, H. Kirchmann, T. Becker, and P. Poller. Large-scale software integration for spoken language and multimodal dialog systems. *Natural Language Engineering*, 10(3-4):283–305, September 2004.

J.-H. Hong, Y.-S. Song, and S.-B. Cho. Mixed-initiative human–robot interaction using hierarchical bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 37(6):1158–1164, November 2007.

L. Horn and G. Ward. *Handbook of pragmatics*, volume 26. Wiley-Blackwell, 2008.

E. Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, CHI '99, pages 159–166, New York, NY, USA, 1999. ACM.

A. J. Hunt and A. W. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *Proceedings of the 21st International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1996)*, volume 1, pages 373–376 vol. 1, 1996.

L. F. Hurtado, D. Griol, E. Sanchis, and E. Segarra. A stochastic approach to dialog management. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2005)*, pages 226–231. IEEE, 2005.

ISCA, editor. *Proceedings of the 12th Annual Conference of the International Speech Communication Association (Interspeech 2011)*, 2011.

D. Jan, E. Chance, D. Rajpurohit, D. DeVault, A. Leuski, J. Morie, and D. Traum. Checkpoint exercise: Training with virtual actors in virtual worlds. In *Intelligent Virtual Agents*, volume 6895 of *Lecture Notes in Computer Science*, pages 453–454. Springer, 2011.

S. Janarthanam, O. Lemon, X. Liu, P. Bartie, W. Mackaness, T. Dalmas, and J. Goetze. Integrating location, visibility, and question-answering in a spoken dialogue system for pedestrian city exploration. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 134–136. Association for Computational Linguistics, 2012.

F. Jelinek. *Statistical methods for speech recognition*. MIT Press, Cambridge, MA, USA, 1997.

F. V. Jensen, K. G. Olesen, and S. K. Andersen. An algebra of bayesian belief universes for knowledge-based systems. *Networks*, 20(5):637–659, 1990.

M. Johnson and E. Charniak. A tag-based noisy channel model of speech repairs. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, ACL '04. Association for Computational Linguistics, 2004.

K. Jokinen and T. Hurtig. User expectations and real experience on a multimodal interactive system. In *INTERSPEECH*. ISCA, 2006.

A. Jönsson and N. Dahlbäck. Talking to a computer is not like talking to your best friend. In *Proceedings of the First Scandinavian Conference on Artificial Intelligence*, pages 53–68, 1988.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, November 1999.

D. Jurafsky, L. Shriberg, and D. Biasca. Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13. Technical report, University of Colorado at Boulder Technical Report 97-02, 1997.

D. Jurafsky, E. Shriberg, B. Fox, and T. Curl. Lexical, prosodic, and syntactic cues for dialog acts. In *Proceedings of ACL/COLING-98 Workshop on Discourse Relations and Discourse Markers*, pages 114–120, 1998.

M. Kearns. A sparse sampling algorithm for near-optimal planning in large markov decision processes. In *Machine Learning*, pages 1324–1331, 1999.

S. Keizer and R. op den Akker. Dialogue act recognition under uncertainty using Bayesian networks. *Natural Language Engineering*, 13:287–316, 11 2007.

J. Kelleher and J. Van Genabith. Visual salience and reference resolution in simulated 3-d environments. *Artificial Intelligence Review*, 21(3-4):253–267, June 2004.

A. Koller and M. Stone. Sentence generation as a planning problem. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 336–343, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

A. Koller, K. Garoufi, M. Staudte, and M. W. Crocker. Enhancing referential success by tracking hearer gaze. In *SIGDIAL Conference*, pages 30–39, 2012.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

K. Komatani, K. Tanaka, H. Kashima, and T. Kawahara. Domain-independent spoken dialogue platform using key-phrase spotting based on combined language model. In P. Dalsgaard, B. Lindberg, H. Benner, and Z.-Hua Tan, editors, *INTERSPEECH*, pages 1319–1322. ISCA, 2001.

S. Kopp, B. Jung, N. Lessmann, and I. Wachsmuth. Max - a multimodal assistant in virtual reality construction. *Künstliche Intelligenz*, 17(4):11, 2003.

G.-J. M. Kruijff, P. Lison, T. Benjamin, H. Jacobsson, H. Zender, I. Kruijff-Korbayovà, and N. Hawes. Situated dialogue processing for human-robot interaction. In H. Christensen, G.-J. M.. Kruijff, and J. L. Wyatt, editors, *Cognitive Systems*, pages 311–364. Springer Verlag, 2010.

H. Kurniawati, D. Hsu, and W.S. Lee. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proc. Robotics: Science and Systems*, 2008.

S. Larsson. *Issue-based Dialogue Management*. PhD thesis, Gothenburg University, 2002.

S. Larsson and D. R. Traum. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natuarl Language Engineering*, 6(3-4):323–340, September 2000a. ISSN 1351-3249.

S. Larsson and D. R. Traum. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural language engineering*, 6(3 & 4):323–340, 2000b.

S. Larsson, R. Cooper, E. Engdahl, and P. Ljunglöf. Information states and dialogue move engines. *Electronic Transactions on Artificial Intelligence*, 3(D):53–71, 1999.

S. Lemaignan, R. Ros, E. A. Sisbot, R. Alami, and M. Beetz. Grounding the interaction: Anchoring situated discourse in everyday human-robot interaction. *International Journal of Social Robotics*, 4(2):181–199, 2012.

O. Lemon and O. Pietquin. Machine Learning for Spoken Dialogue Systems. In *Proceedings of the 10th European Conference on Speech Communication and Technologies (Interspeech'07)*, pages 2685–2688, 2007.

O. Lemon, K. Georgila, and J. Henderson. Evaluating effectiveness and portability of reinforcement learned dialogue strategies with real users: the TALK TownInfo evaluation. In *Spoken Language Technology Workshop, 2006. IEEE*, pages 178–181, 2006.

E. Levin, R. Pieraccini, and W. Eckert. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing*, 8(1):11–23, 2000. ISSN 1063-6676.

S. C. Levinson. *Pragmatics*. Cambridge University Press, 1983.

P. Lison. A salience-driven approach to speech recognition for human-robot interaction. In *Interfaces: Explorations in Logic, Language and Computation*, pages 102–113. Springer Verlag, 2010.

P. Lison. Declarative design of spoken dialogue systems with probabilistic rules. In *Proceedings of the 16th Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2012)*, pages 97–106, 2012a.

P. Lison. Probabilistic dialogue models with prior domain knowledge. In *Proceedings of the SIGDIAL 2012 Conference*, pages 179–188, 2012b.

P. Lison. Model-based bayesian reinforcement learning for dialogue management. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association (Interspeech)*, 2013. (accepted).

D. J. Litman and J. F. Allen. A plan recognition model for subdialogues in conversations. *Cognitive Science*, 11(2):163–200, 1987.

R. López-Cózar and Z. Callejas. Multimodal dialogue for ambient intelligence and smart environments. In H. Nakashima, H. Aghajan, and J.-C. Augusto, editors, *Handbook of Ambient Intelligence and Smart Environments*, pages 559–579. Springer US, 2010.

D. J.C. MacKay. Introduction to monte carlo methods. In *Learning in graphical models*, pages 175–204. Springer, 1998.

C. Matheson, M. Poesio, and D. R. Traum. Modelling grounding and discourse obligations using update rules. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference (NAACL 2000)*, pages 1–8, 2000.

M. F. McTear. *Spoken dialogue technology - toward the conversational user interface*. Springer, 2004.

J.-J. Ch. Meyer and W. Van Der Hoek. *Epistemic logic for AI and computer science*, volume 41. Cambridge University Press, 2004.

N. Mirnig, A. Weiss, G. Skantze, S. Al Moubayed, J. Gustafson, J. Beskow, B. Granström, and M. Tscheligi. Face-to-face with a robot: What do we actually talk about? *International Journal of Humanoid Robotics*, 10(1), 2013.

F. Morbini, E. Forbell, D. DeVault, K. Sagae, D. R. Traum, and A. A. Rizzo. A mixed-initiative conversational dialogue system for healthcare. In *SIGDIAL Conference*, pages 137–139. The Association for Computer Linguistics, 2012.

K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: an empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence (UAI 1999)*, pages 467–475, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1-55860-614-9.

Y. I. Nakano, G. Reinstein, T. Stocky, and J. Cassell. Towards a model of face-to-face grounding. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, pages 553–561. Association for Computational Linguistics, 2003.

S. Oviatt, R. Coulston, and R. Lunsford. When do we interact multimodally?: cognitive load and multimodal communication patterns. In *Proceedings of the 6th international conference on Multimodal interfaces*, pages 129–136. ACM, 2004.

T. Paek. Reinforcement learning for spoken dialogue systems: Comparing strengths and weaknesses for practical deployment. In *Proceedings of the Interspeech Workshop "Dialogue on Dialogues - Multidisciplinary Evaluation of Advanced Speech-based Interacive Systems"*, 2006.

T. Paek and D. M. Chickering. Evaluating the markov assumption in markov decision processes for spoken dialogue management. *Language Resources and Evaluation*, 40(1):47–66, 2006.

T. Paek and R. Pieraccini. Automating spoken dialogue management design using machine learning: An industry perspective. *Speech Communications*, 50(8-9):716–729, August 2008.

C. Papadimitriou and J. N. Tsitsiklis. The complexity of markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, August 1987.

J. S. Pardo. On phonetic convergence during conversational interaction. *The Journal of the Acoustical Society of America*, 119:2382, 2006.

R. J. Passonneau, S. L. Epstein, and T. Ligorio. Naturalistic dialogue management for noisy speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 6(8):928–942, 2012.

J. Pearl. Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence*, 32 (2):245–257, 1987.

O. Pietquin. Optimising spoken dialogue strategies within the reinforcement learning paradigm. In *Reinforcement Learning, Theory and Applications*, pages 239–256. I-Tech Education and Publishing, 2008.

O. Pietquin and T. Dutoit. A probabilistic framework for dialog simulation and optimal strategy learning. *IEEE Transactions on Audio, Speech and Language Processing*, 14(2):589–599, December 2006.

J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 1025 – 1032, 2003.

M. Poesio and H. Rieser. An incremental model of anaphora and reference resolution based on resource situations. *Dialogue & Discourse*, 2(1):235–277, 2011.

P. Poupart. *Exploiting structure to efficiently solve large scale partially observable markov decision processes*. PhD thesis, University of Toronto, Toronto, Canada, 2005.

P. Prodanov and A. Drygajlo. Bayesian networks for spoken dialogue management in multimodal systems of tour-guide robots. In *Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech)*, pages 1057–1060, 2003.

M. Purver. *The Theory and Use of Clarification Requests in Dialogue*. PhD thesis, King's College, University of London, August 2004.

A. Raux and M. Eskenazi. A finite-state turn-taking model for spoken dialog systems. In *HLT-NAACL*, pages 629–637. The Association for Computational Linguistics, 2009.

A. Raux, B. Langner, D. Bohus, A. W. Black, and M. Eskenazi. Let's go public! taking a spoken dialog system to the real world. In *INTERSPEECH*, pages 885–888. ISCA, 2005.

V. Rieser and O. Lemon. Using logistic regression to initilise reinforcement-learning-based systems. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT 2006)*, pages 190–193, 2006.

V. Rieser and O. Lemon. Natural language generation as planning under uncertainty for spoken dialogue systems. In Emiel Krahmer and Mariët Theune, editors, *Empirical methods in natural language generation*, pages 105–120. Springer-Verlag, Berlin, Heidelberg, 2010a.

V. Rieser and O. Lemon. Learning human multimodal dialogue strategies. *Natural Language Engineering*, 16:3–23, 2010b.

V. Rieser and J. D. Moore. Implications for generating clarification requests in task-oriented dialogues. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 239–246. Association for Computational Linguistics, 2005.

E. K. Ringger and J. F. Allen. Error correction via a post-processor for continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1996)*, pages 427–430, Washington, DC, USA, 1996. IEEE Computer Society.

D. Roy. Semiotic schemas: A framework for grounding language in action and perception. *Artificial Intelligence*, 167(1-2):170–205, 2005.

N. Roy, J. Pineau, and S. Thrun. Spoken dialogue management using probabilistic reasoning. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 93–100. Association for Computational Linguistics, 2000.

G. A. Rummery. *Problem Solving with Reinforcement Learning*. PhD thesis, Cambridge University, 1995.

S. J. Russell and P. Norvig. *Artificial Intelligence - A Modern Approach*. Pearson Education, 2010.

H. Sacks, E. A. Schegloff, and G. Jefferson. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735, dec 1974.

M. Salem, S. Kopp, I. Wachsmuth, K. Rohlfing, and F. Joublin. Generation and evaluation of communicative robot gesture. *International Journal of Social Robotics, Special Issue on Expectations, Intentions, and Actions*, 2012.

M. Salem, S. Kopp, and F. Joublin. Generating finely synchronized gesture and speech for humanoid robots: a closed-loop approach. In *HRI*, pages 219–220. IEEE/ACM, 2013.

K. Scheffler and S. Young. Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning. In *Proceedings of the second international conference on Human Language Technology (HLT 2002)*, pages 12–19, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

D. Schlangen. Towards finding and fixing fragments: using ml to identify non-sentential utterances and their antecedents in multi-party dialogue. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 247–254. Association for Computational Linguistics, 2005.

D. Schlangen and A. Lascarides. The interpretation of non-sentential utterances in dialogue. In *Proceedings of the 4th SIGDIAL Workshop on Discourse and Dialogue*, 2003.

D. Schlangen and G. Skantze. A general, abstract model of incremental dialogue processing. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*, pages 710–718. Association for Computational Linguistics, 2009.

D. Schlangen, T. Baumann, and M. Atterer. Incremental reference resolution: The task, metrics for evaluation, and a Bayesian filtering model that is sensitive to disfluencies. In *Proceedings of the 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2009)*, pages 30–37. Association for Computational Linguistics, 2009.

D. Schlangen, T. Baumann, H. Buschmeier, S. Kopp, G. Skantze, and R. Yaghoubzadeh. Middleware for incremental processing in conversational agents. In *Proceedings of the 11th Annual Meeting of the Special Interest Group in Discourse and Dialogue (SIGDIAL 2010)*, pages 51–54. Association for Computational Linguistics, 2010.

J. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969.

J. Searle. *Expression and Meaning. Studies in the Theory of Speech acts*. Cambridge University Press, 1979.

S. Seneff and J. Polifroni. Dialogue Management in the Mercury Flight Reservation System. In *ANLP-NAACL Workshop on Conversational Systems*, Seattle, 2000.

G. Shani, J. Pineau, and R. Kaplow. A survey of point-based POMDP solvers. *Autonomous Agents and Multi-Agent Systems*, 27(1):1–51, 2013. ISSN 1387-2532.

E. Shriberg, A. Stolcke, D. Jurafsky, N. Coccaro, M. Meteer, R. Bates, P. Taylor, K. Ries, R. Martin, and C. Van Ess-Dykema. Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and speech*, 41(3-4):443–492, 1998.

D. Silver and J. Veness. Monte-carlo planning in large POMDPs. In *Advances in Neural Information Processing Systems (NIPS 2010)*, pages 2164–2172, 2010.

J. Sinclair and M. Coulthard. *Towards an Analysis of Discourse*. Oxford University Press, 1975.

S. P. Singh, D. J. Litman, M. J. Kearns, and M. A. Walker. Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *Journal of Artificial Intelligence Research*, 16:105–133, 2002.

Satinder P. Singh, Michael J. Kearns, D. J. Litman, and Marilyn A. Walker. Empirical evaluation of a reinforcement learning spoken dialogue system. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 645–651. AAAI Press, 2000. ISBN 0-262-51112-6.

G. Skantze. *Error Handling in Spoken Dialogue Systems: Managing Uncertainty, Grounding and Miscommunication*. PhD thesis, Royal Institute of Technology (KTH), Stockholm, 2007.

E. J. Sondik. *The Optimal Control of Partially Observable Markov Processes*. PhD thesis, Stanford University, 1971.

O. Ståhl, B. Gambäck, M. Turunen, and J. Hakulinen. A mobile health and fitness companion demonstrator. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*, pages 65–68. Association for Computational Linguistics, 2009.

A.J. Stent and S. Bangalore. Interaction between dialog structure and coreference resolution. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT 2010)*, pages 342–347, 2010.

R. Stiefelhagen, C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel. Natural human-robot interaction using speech, head pose and gestures. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, volume 3, pages 2422–2427. IEEE, 2004.

T. Stivers, N. J. Enfield, P Brown, C. Englert, M. Hayashi, T. Heinemann, G. Hoymann, F. Rossano, J. P de Ruiter, K.-E. Yoon, and S. C. Levinson. Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences*, 106(26):10587–10592, 2009.

A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema, and M. Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373, 2000.

M. Stone, C. Doran, B. Webber, T. Bleam, and M. Palmer. Microplanning with communicative intentions: The SPUD system. *Computational Intelligence*, 19(4):311–381, 2003.

M. Strube and C. Müller. A machine learning approach to pronoun resolution in spoken dialogue. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, pages 168–175. Association for Computational Linguistics, 2003.

N. Ström and S. Seneff. Intelligent barge-in in conversational systems. In *Proceedings of the 6th International Conference of Spoken Language Processing (ICSLP/Interspeech 2000)*, pages 652–655. ISCA, 2000.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 1998.

R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML 2009)*, pages 993–1000. ACM, 2009.

J. Tetreault and D. Litman. Using reinforcement learning to build a better model of dialogue state. In *Proceedings of the 11th Conference of the European Association for Computational Linguistics (EACL 2006)*, 2006.

R. H. Thomason and M. Stone. Enlightened update: A computational architecture for presupposition and other pragmatic phenomena. In *Presupposition Accommodation*. Ohio State Pragmatics Initiative, 2006.

V. Thomson and S. Young. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech & Language*, 24:562–588, October 2010.

M. Tomasello, M. Carpenter, J. Call, T. Behne, and H. Moll. Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and Brain Sciences*, 28:675–691, 9 2005.

D. R. Traum. *A Computational Theory of Grounding in Natural Language Conversation*. PhD thesis, University of Rochester, Rochester, NY, USA, 1994.

D. R. Traum and J. F. Allen. Discourse obligations in dialogue processing. In *Proceedings of the 32nd annual meeting of the Association for Computational Linguistics*, pages 1–8. Association for Computational Linguistics, 1994.

D. R. Traum and E. A. Hinkelman. Conversation acts in task-oriented spoken dialogue. *Computational Intelligence*, 8:575–599, 1992.

D. R. Traum, P. Aggarwal, R. Artstein, S. Foutz, J. Gerten, A. Katsamanis, A. Leuski, D. Noren, and W. R. Swartout. Ada and grace: Direct interaction with museum visitors. In *The 12th International Conference on Intelligent Virtual Agents (IVA 2012)*, volume 7502 of *Lecture Notes in Computer Science*, pages 245–251. Springer, 2012.

M. Turunen. *Jaspis–A Spoken Dialogue Architecture and Its Applications*. PhD thesis, University of Tampere, Department of Computer Sciences, Finland, 2004.

G.-J. Van Noord, G. Bouma, R. Koeling, and M.-J. Nederhof. Robust grammatical analysis for spoken dialogue systems. *Natural Language Engineering*, 5:45–93, 2 1999.

R. C. Vipperla, M. Wolters, K. Georgila, and S. Renals. Speech input from older users in smart environments: Challenges and perspectives. In *Proceedings of HCI International: Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*, number 5615 in Lecture Notes in Computer Science. Springer, 2009.

T. Visser, D. Traum, D. DeVault, and R. op den Akker. Toward a model for incremental grounding in spoken dialogue systems. In *The 12th International Conference on Intelligent Virtual Agents (IVA 2012)*, Santa Cruz, CA, September 2012.

W. E. Wahlster. *SmartKom: Foundations of Multimodal Dialogue Systems*. Cognitive Technologies. Springer Verlag, 2006.

M. Walker, B. Pellom, J. Polifroni, A. Potamianos, P. Prabhu, A. Rudnicky, S. Seneff, and D. Stallard. DARPA Communicator dialog travel planning systems: The June 2000 data collection. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Interspeech 2001)*, pages 1371–1374, 2001.

M. A. Walker. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *Journal of Artificial Intelligence Research*, 12(1):387–416, 2000.

C. Watkins and P. Dayan. Technical note: Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

J. D. Williams. Exploiting the ASR n-best by tracking multiple dialog state hypotheses. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association (Interspeech 2008)*, pages 191–194. ISCA, 2008a.

J. D. Williams. The best of both worlds: Unifying conventional dialog systems and POMDPs. In *International Conference on Speech and Language Processing (ICSLP 2008)*, Brisbane, Australia, 2008b.

J. D. Williams and S. Young. Using Wizard-of-Oz simulations to bootstrap Reinforcement- Learning based dialog management systems. In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue*, 2003.

J. D. Williams and S. Young. Scaling up POMDPs for dialog management: The "summary pomdp" method. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2005)*, pages 177–182, 2005.

J. D. Williams and S. Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21:393–422, 2007. doi: 10.1016/j.csl.2006.06.008.

J. D. Williams, P. Poupart, and S. Young. Factored Partially Observable Markov Decision Processes for Dialogue Management. In *Proceedings of the 4th Workshop on Knowledge and Reasoning in Practical Dialog Systems*, 2005.

J. D. Williams, P. Poupart, and S. Young. Partially Observable Markov Decision Processes with continuous observations for dialogue management. In L. Dybkjær and W. Minker, editors, *Recent Trends in Discourse and Dialogue*, volume 39 of *Text, Speech and Language Technology*, pages 191–217. Springer Netherlands, 2008.

D. Wilson and D. Sperber. Relevance theory. *Handbook of pragmatics*, 2002.

M. Wölfel and J. McDonough. *Distant speech recognition*. Wiley, 2009.

F. Xu, S. Schmeier, R. Ai, and H. Uszkoreit. Yochina: Mobile multimedia and multimodal crosslingual dialog system. In *Proceedings of International Workshop On Spoken Dialogue Systems Technology (IWSDS 2012)*. Springer, 2012.

S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24:150–174, 2010.

S. Young, M. Gačić, B. Thomson, and J. D. Williams. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.

H. Zender, G.-J. M. Kruijff, and I. Kruijff-Korbayová. Situated resolution and generation of spatial referring expressions for robotic assistants. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 1604–1609, Pasadena, CA, USA, July 2009.

B. Zhang, Q. Cai, J. Mao, E. Chang, and B. Guo. Spoken Dialogue Management as Planning and Acting under Uncertainty. In *Proceedings of 7th European Conference on Speech Communication and Technology*, pages 2169–2172, 2001.

N. Lianwen Zhang and D. Poole. Exploiting causal independence in bayesian network inference. *Journal of Artificial Intelligence Research (JAIR)*, 5:301–328, 1996.

P. Zhang, Q. Zhao, and Y. Yan. A spoken dialogue system based on keyword spotting technology. In J. A. Jacko, editor, *Proceedings of the 12th international conference on Human-Computer Interaction (HCI 2007)*, volume 4552 of *Lecture Notes in Computer Science*, pages 253–261. Springer, 2007.

V. Zue, S. Seneff, J.R. Glass, J. Polifroni, C. Pao, T.J. Hazen, and L. Hetherington. JUPITER: a telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing*, 8(1):85–96, 2000.

Index

acoustic model, 18
action selection, 20
alignment, 15
ambiguities, 19
Artificial Intelligence, 3

backchannel, 15
backward-looking function, 12
barge-in, 11
Bayesian inference, 4
Bayesian Network, 4
Belief-Desire-Intention model, 3

clarification strategy, 15
common ground, 14
contextal awareness, 17
contextual modelling, 17
conversation analysis, 10
conversational floor, 10

DAMSL, 11
declarative design, 5
deictic marker, 13
dialogue act, 11
dialogue management, 2, 16, 20
dialogue modelling, 4
dialogue state, 2, 20
dialogue state update, 20
dialogue system architectures, 15
dialogue understanding, 16

error handling, 15
expert knowledge, 4

formal logic, 3
forward-looking function, 12

grounding, 14

Hidden Markov Model, 18
human–robot interaction, 5, 17

implicature, 12
incremental processing, 18
information state, 17
initiative, 10

language model, 18
lexical entrainment, 15
likelihood weighted sampling, 4

mixed initiative, 10
multi-modal dialogue system, 17
multi-modality, 17

N-best list, 19
Nao robot, 5
natural language generation, 16, 21
negative feedback, 15
non-sentential utterance, 12

openDial, 5

pipeline architecture, 17
positive feedback, 14
prior knowledge, 4
probabilistic modelling, 3
probabilistic rules, 4

reinforcement learning, 5
relational structure, 2
repair strategy, 15

sequential decision-making under uncertainty, 3
speech act, 11
speech recognition, 16, 18
speech synthesis, 21

spoken dialogue dystems, 1
spoken language, 1
supervised learning, 4

text-to-speech synthesis, 16, 21
turn-taking, 10

uncertainty, 2
user evaluation, 5
user interfaces, 1
user model, 20

variable elimination, 4

Wizard-of-Oz interaction, 4
Word Error Rate, 19