The Knowledge Engineering Review, Vol. 00:0, 1–24. © 2006, Cambridge University Press
DOI: 10.1017/S000000000000000 Printed in the United Kingdom

A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies

JOST SCHATZMANN, KARL WEILHAMMER, MATT STUTTLE, STEVE YOUNG

Cambridge University Engineering Department, Trumpington Street, Cambridge CB21PZ, UK
E-mail: {js532, kw278, mns25, sjy}@eng.cam.ac.uk

Abstract

Within the broad field of spoken dialogue systems, the application of machine-learning approaches to dialogue management strategy design is a rapidly growing research area. The main motivation is the hope of building systems that learn through trial-and-error interaction what constitutes a good dialogue strategy. Training of such systems could in theory be done using human users or using corpora of human-computer dialogue, but in practice the typically vast space of possible dialogue states and strategies cannot be explored without the use of automatic user simulation tools.

This requirement for training statistical dialogue models has created an interesting new application area for predictive statistical user modelling and a variety of different techniques for simulating user behaviour have been presented in the literature ranging from simple Markov Models to Bayesian Networks. The development of reliable user simulation tools is critical to further progress on automatic dialogue management design but it holds many challenges, some of which have been encountered in other areas of current research on statistical user modelling, such as the problem of "concept drift", the problem of combining content-based and collaboration-based modelling techniques, and user model evaluation. The latter topic is of particular interest, because simulation-based learning is currently one of the few applications of statistical user modelling which employs both direct "accuracy-based" and indirect "utility-based" evaluation techniques.

In this paper, we briefly summarize the role of the dialogue manager in a spoken dialogue system, give a short introduction to reinforcement-learning of dialogue management strategies and review the literature on user modelling for simulation-based strategy learning. We further describe recent work on user model evaluation and discuss some of the current research issues in simulation-based learning from a user modelling perspective.

1 Introduction and Overview

Research on user modelling has a long history in the field of Spoken Dialogue Systems (SDS) and Natural Language Processing (NLP). Traditionally, the most commonly cited motivation for this work has been to enable dialogue systems to adapt to individual user needs and preferences by consulting a model of their human dialogue partner, similar to the way humans intuitively use mental models of their interlocutor when engaging in a conversation. The majority of research on user modelling in SDS and NLP has always had a strong linguistically motivated focus on non-statistical methods and the modelling techniques employed have often been highly complex in nature and frequently based on manually constructed rules or various forms of logic (Zukerman and Litman, 2001).

In recent years, however, the application of machine-learning approaches to dialogue system design has created a need for simple statistical user models which are probabilistic in nature and trainable on existing human-computer dialogue data. The purpose of user modelling in this new area of SDS research is not to model the precise characteristics of an individual user during the live operation of a system, but to provide the basis for user simulation tools which can be used during the development phase of a new dialogue system (Levin et al., 2000; Young, 2002a).

Current work on predictive, statistical user modelling in SDS is of particular importance to the design of dialogue management strategies. The task of the Dialogue Manager (DM) in a spoken dialogue system is to control the flow of the dialogue and to decide which actions to take in response to user input. In order to complete these tasks, the DM needs a *dialogue strategy* that defines when to take the initiative in a dialogue, when to confirm receipt of a piece of information, how to identify and recover from recognition and understanding errors, and so forth. The manual development of a good dialogue strategy is a major design challenge and has been described as "an art, rather than a science" (Jones and Galliers, 1996) for many years.

Machine-learning approaches to dialogue management attempt to learn optimal strategies from corpora of real human-computer dialogue data using automated "trial-and-error" methods instead of relying on empirical design principles. However, the size of currently available annotated dialogue corpora is far too small to sufficiently explore the vast space of possible dialogue states and strategies. In addition, no guarantee can be given that the truly optimal strategy is indeed present in a given training corpus and it may thus be argued that an optimal strategy cannot be learned from a fixed corpus, regardless of its size. An interesting solution to this problem is to train a statistical, predictive user model for simulating user responses which can then be used to learn dialogue strategies through trial-and-error interaction between the dialogue manager and the simulated user (Levin et al., 2000; Scheffler, 2002; Pietquin, 2004; Henderson et al., 2005; Filisko and Seneff, 2005).

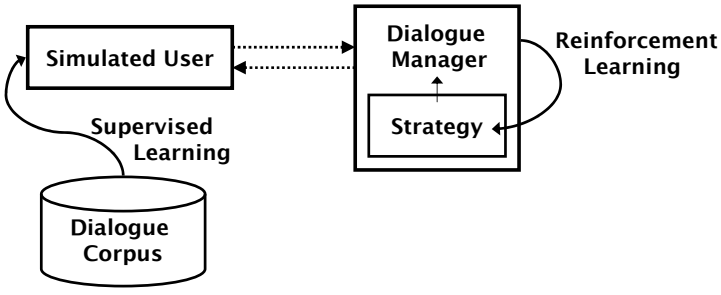

Figure 1 Learning Dialogue Strategies with a Simulated User: A statistical model for simulating user behaviour is first trained on a corpus of human-computer dialogues. Using reinforcement-learning, an optimal dialogue strategy can then be found through trial-and-error interaction between the simulated user and the learning dialogue manager.

More specifically, the user simulation-based setup for learning dialogue strategies consists of two phases, as illustrated in Figure 1. First, the user model is trained on a dialogue corpus to learn what

responses a real user would provide in a given dialogue context[1]. A form of supervised learning, i.e. "learning by example" is commonly used for this step. In the second phase the trained user model is used to predict responses to system actions. The learning system interacts with the simulated user and optimises its dialogue strategy based on the feedback given by the simulated user. This form of *Reinforcement-Learning* with an automatic user simulation tool allows any number of training dialogues to be generated and it also enables dialogue strategies that are not present in the existing corpus of human computer dialogues to be explored. It enables the DM to deviate from the known strategies and learn new and potentially better ones.

User simulation-based learning of dialogue strategies presents a very interesting application of predictive statistical user modelling. The work is critical to further progress on machine-learning approaches to dialogue management but it is also highly challenging. The user model needs to be capable of producing responses that a real user might have given in a certain dialogue situation, taking into account observable aspects such as the dialogue history as well as unobservable aspects such the user's goal, memory and preferences. The user model must also cover a broad variety of natural behaviour and reflect the diversity of a real user population to ensure that the learned strategies work well for all kinds of human user behavior. Within a specific dialogue however, the responses should be consistent and reflect individual user characteristics such as expertise, cooperativeness and patience. Finally, the user model must achieve a trade-off between adequately covering the complexity of human behaviour and being simple enough for its parameters to be trainable on a limited amount of corpus data.

To date, strategy learning with a simulated user has been addressed almost entirely by research groups with a background in dialogue systems design. In this paper, we survey the existing literature and try to show that much could be gained by approaching the problem from a user modelling viewpoint. We argue that many problems encountered in building user models for simulation-based learning are problems also encountered in other areas of statistical user modelling. Our aim is to show that an understanding of user modelling concepts such as *concept-drift* and the use of user modelling terminology (eg. *content- vs. collaboration-based approaches*) may be very helpful for future work on dialogue strategy learning.

Research on simulation-based learning is also interesting with regards to user model evaluation. In recent work, the topic has been approached using direct and indirect forms of evaluation to assess both the quality of the currently available user models as well as the quality of the dialogue strategies learned with these user models. The results are summarized in this paper and are likely to be of interest to research groups working on statistical user modelling inside and outside of spoken dialogue systems design.

The paper is structured as follows. We first provide a brief introduction to the role of the dialogue manager in a spoken dialogue system (Section 2) and explain how reinforcement learning techniques can be used for finding optimal dialogue management strategies through trial and error interaction with a simulated user (Section 3). We then summarize the background literature on user simulation-based learning and discuss current research issues in the field from a user modelling perspective (Section 4). Finally, we report on recent work on user model evaluation (Section 5) and conclude with a brief summary (Section 6).

[1] The difficulties of finding an adequate Markovian state representation of dialogue context and building a user model that generalises well to unseen dialogue situations will be discussed in this paper.

2 Dialogue Management in Spoken Dialogue Systems

The field of spoken dialogue systems has seen rapid developments over the past decade, both in the academic (Jurafsky et al., 1994; Zue, 1997; Seneff et al., 1998; Glass, 1999; Young, 2002a) as well as the commercial domain (Aust et al., 1995; Gorin et al., 1997; Gurston et al., 2001; Pieraccini et al., 2004). Speech recognition is turning into a mature technology and voice-driven interfaces are becoming a common part of everyday life through mass-market applications such as portable devices, in-car navigation systems or telephone-based information systems. Overall, the more recently deployed commercial applications are gradually changing the public perception that speech interfaces are notoriously unreliable, badly designed and error-prone.

The aim of this section is to explain the function of a dialogue management strategy in a dialogue system and to motivate the use of statistical approaches to strategy design. We first provide a very brief overview of the general architecture of a spoken dialogue system and outline the role of the dialogue manager in Section 2.1. Section 2.2 introduces the concept of a dialogue strategy and investigates two of the key issues in dialogue management strategy design. Section 2.3 concludes by contrasting empirical and statistical approaches to strategy development.

2.1 The Role of the Dialogue Manager

The general architecture of a spoken dialogue system (SDS) is best visualised using a block diagram depicting the flow of information between the user and the major system components. As shown in Figure 2, the dialogue manager (DM) is the central component of a dialogue system and interfaces with the input- as well as the output-processing side of the system. It receives the semantically parsed representation of the user input and generates an appropriate high-level representation of the next system action. The task of the DM is to control the flow of the dialogue and to decide which actions to take in response to user input. The DM has to interpret the incoming semantic representation of the user input in the context of the dialogue, resolve ellipsis and anaphora, evaluate the relevance and completeness of user requests, identify and recover from recognition and understanding errors, retrieve information from peripheral databases and format appropriate system replies.

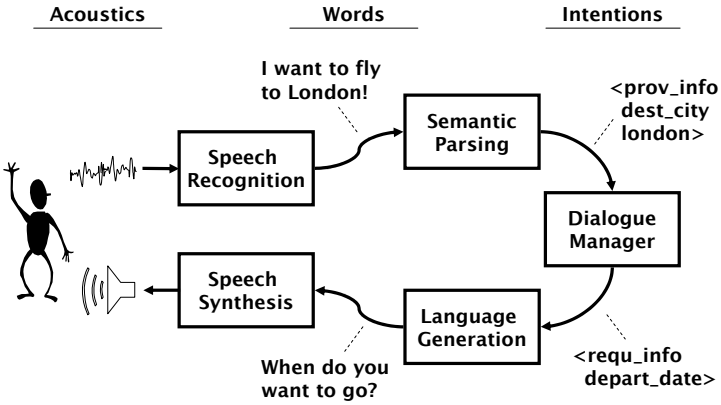

Figure 2 The main components of a Spoken Dialogue System: The speech recognizer receives the acoustic signal emitted by the user, translates it into a feature based representation and outputs the most likely sequence of words, or alternatively a list of the most likely hypotheses. The text-based output of the speech recognizer is then semantically parsed by a language understanding component and associated with a meaning representation. On the output generation side, the reverse process is followed: the language generation component translates the high-level representation of a system action into a sequence of words which is then synthesised using a text-to-speech component.

2.2 Dialogue Management Strategies

In order to complete the tasks described above and to decide "what to say", the dialogue manager needs to track the *dialogue history* and update some representation of the current state of the dialogue. In addition, the DM needs a *dialogue strategy* that defines the conversational behaviour of the system, for example when to take the initiative in a dialogue or when to establish common ground.

The performance of a spoken dialogue system is highly dependent on the quality of its DM strategy. Unfortunately, the design of a good strategy is far from trivial since there is no clear definition as to what constitutes a good strategy. User populations are often diverse, thus making it difficult to foresee which form of system behaviour will lead to quick and successful completion of the dialogue. In addition, the omnipresence of recognition errors leads to constant uncertainty about the true intention of the user. As applications reach greater levels of complexity, the process of designing robust dialogue strategies becomes very time-consuming and expensive.

To further illustrate the complexity involved in dialogue strategy design, the following subsections briefly discuss two frequently arising design issues: the *degree of initiative* a system can take in a dialogue and the choice of *confirmation strategy*. Our goal here is not to provide a complete overview or to argue in favour of any specific design choices, but to indicate the potential benefits of automatic machine-learning techniques in the context of dialogue strategy design. Two issues which have recently received much attention but are not covered here include *error handling and recovery* (Bohus, 2004; Bohus and Rudnicky, 2005a) and accounting for *uncertainty* introduced by the noisy communication channel (Bohus and Rudnicky, 2005b; Williams, 2003; Williams et al., 2005a,b).

2.2.1 Degrees of Initiative

Dialogue management strategies are often classified according to the degree of initiative they take in a dialogue. Systems following a *system-initiative* strategy take the lead role in a dialogue by asking precise, direct questions such as "Where are you flying from? What is your destination? What is your preferred departure date?". The user is sequentially prompted for every necessary piece of information and his or her response is expected to contain only the requested piece of information. This type of approach is often chosen in commercial applications, because it usually results in short user responses, less out-of-domain utterances and higher recognition performance (Kamm, 1995).

Mixed-initiative systems on the other hand, share control over the dialogue flow with the user. Systems of this type often start with an open question such as "How can I help you?". They can usually process more complex linguistic constructs and allow the user to specify several pieces of information at the same time. Mixed-initiative systems place less constraints on the information flow between the user and the system and thus potentially allow for more efficient dialogues. An expert user who is familiar with the system for instance, is not forced to step through a long series of prompts. Instead, he or she is able to enter all the necessary information without having to wait for the system to request it (Smith and Hipp, 1994; Webber and Joshi, 1982).

Given adequate recognition performance, mixed-initiative strategies lead to a more powerful form of dialogue control because they exploit the ability of human language to simultaneously convey multiple pieces of information in a compact representation. Indeed, user frustration with current system-initiative SDS often arises because this advantage is not exploited. A series of yes/no questions for a telephony application, for instance, may just as well be implemented using touch-tone input. Users are often reluctant to use a voice interface if the same functionality can be delivered using a more noise-robust and less error-prone communication channel. With a mixed-initiative strategy, the benefits of using a natural language interface should be more evident to the user.

Studies such as (Potjer et al., 1996), however, show that mixed-initiative strategies may not always be preferable given the limited recognition and understanding capabilities of current systems. A system initiative strategy places stronger constraints on the user input and enables the search space of the recognizer to be highly constrained. As a consequence, the recognition performance and goal achievement rate under high-noise conditions is often better when a system-initiative strategy is used. The reduced

number of misunderstandings can lead to greater user-satisfaction, despite the strong constraints placed on the dialogue.

In choosing between system-initiative and mixed-initiative, one also needs to consider that users draw on previous experiences with spoken dialogue systems and tend to develop a mental model of the system behaviour. Users who are used to traditional forms of system-initiative "slot-filling" dialogue, may expect the SDS to fill one slot after the other. An open question such as "How can I help you?" may appear irritating, since it gives no clear indication as to what response the system expects. Even expert-users often prefer a clearly structured system-initiative dialogue over a mixed-initiative one (Walker et al., 1997). The perceived dialogue progress is also very important. A system-initiative strategy may be the less efficient form of communication, but it provides an assuring feeling of constant progress.

Research has suggested the use of hybrid approaches (Rosset et al., 1999; Rudnicky et al., 1999; Fleming and Cohen, 2001) that allow a mixed-initiative system to back off to system-initiative when dialogue problems are detected. This way, the system can regain control when the dialogue seems to deviate from the correct path or when the recognition performance suffers.

2.2.2 Confirmation Strategies

A second key-issue in designing a dialogue manager is the choice of confirmation strategy. Since human-computer interaction always involves a noisy communication channel, the system needs to confirm its recognition results using grounding actions. The frequency of confirmations and the choice of confirmation style however, leaves plenty of scope for design decisions. In a flight booking system, for instance, the system can collect all the necessary pieces of information and then attempt to confirm all of them in a single turn. Alternatively, it can confirm each piece of information as it goes along.

The system can also choose between *implicit* and *explicit* confirmations. The latter type of confirmation requires the user to verify the recognition result in the next dialogue turn, as shown in the dialogue below:

```
System: Where do you want to go?
User:   To New York.
System: Your destination is New York - is that correct?
User:   Yes.
System: Ok. And on what date do you want to go?
```

By using implicit confirmations, the system can combine the confirmation of one "slot" with a request for a different slot, as shown below:

```
System: Where do you want to go?
User:   To New York.
System: And on what date do you want to go to New York?
```

The performance of different confirmation strategies has been investigated in a number of studies (Smith, 1998; Shin et al., 2002; Niimi and Nishimoto, 1999). The results show that implicit confirmations can increase the efficiency of the dialogue but that their use also involves a higher risk. If the implicitly confirmed piece of information is incorrect, it can be difficult and time-consuming to rectify the misunderstanding and repair the dialogue. A possible way of combining explicit and implicit confirmations is to choose the style of confirmation dynamically, depending on the recognition confidence score produced by the speech recognizer. If the recognition result has a high confidence, an implicit confirmation is chosen. If the confidence is low, the system resorts to an explicit confirmation. If the confidence is very low, the system may even decide to reject the user utterance completely and re-request the piece of information.

2.3 Empirical and Statistical Approaches to Dialogue Strategy Design

Over the last decades, numerous research papers have attempted to develop recommendations and guidelines for dialogue management. Examples of empirically obtained design principles can be found

in (Bernsen et al., 1996, 1998; Rosset et al., 1999) and many further references are listed in (Walker et al., 1998; Walker, 2000). Yet, the process of designing dialogue management strategies is still often driven by "trial and error" (Levin et al., 2000) and remains an "art rather than a science" (Jones and Galliers, 1996).

Typically, a small number of strategies are hand-crafted by a group of expert designers. A spoken dialogue system is then built that explores each of these alternative strategies. The system is then tested on real users and conclusions are drawn as to which strategy performs best for the given task (Gorin et al., 1996; Polifroni et al., 1992; Simpson and Fraser, 1993; Litman and Pan, 1999). A large number of subjects is usually needed to obtain results of statistical significance, leading to high development costs and slow cycles of iterative refinements. Interpretation of the results is also often difficult since user satisfaction or dissatisfaction with a dialogue system or strategy may be due to many different factors. The results can be biased through subtle influences such as the order in which the strategies are presented to the subjects (Devillers and Bonneau-Maynard, 1998) and often do not necessarily provide a clear guide to further improvements.

As the process of designing, implementing and testing dialogue management strategies is becoming increasingly complex, the focus of scientific research is increasingly turning from empirical methods to data-driven techniques. *Statistical approaches* which model processes probabilistically and estimate parameters from corpora of real human-computer dialogues are gaining popularity and have affected all areas of SDS research over the past decade (Young, 2002b,a).

The motivation for training models on real data rather than relying on traditional linguistically-motivated techniques is clear. The latter approach of observing and analysing user phenomena and then manually constructing rules to cover them incurs significant development and maintenance costs. Moreover, the approach is inherently suboptimal because no guarantee can be given, that the hand-crafted set of rules does in fact optimally cover all possible aspects of user behaviour. Especially as systems become more complex, it is nearly impossible to test whether the system will perform well in all possible dialogue scenarios and whether the given set of rules is indeed optimal.

Statistical models, in contrast, can be trained on corpora of real human-computer dialogue and they explicitly model the variance in user behaviour that hand-written rules cannot cover. While model construction and parameterization is of course still dependent on expert knowledge, the hope is to build systems which offer more robust performance, improved portability, better scalability and greater scope for adaptation. The success of statistical approaches, however, is of course naturally tied to the quality of the models and the quality of the data.

3 Reinforcement-Learning of Dialogue Strategies with a Simulated User

This section summarizes the relevant background knowledge on machine-learning approaches to dialogue strategy design. It introduces the Markov Decision Process (MDP)-model of human-computer dialogue (Subsection 3.2) and explains how Reinforcement-Learning techniques can be used to find optimal dialogue strategies within the MDP framework (3.3). It further contrasts so-called *model-based* and *simulation-based* approaches to strategy learning and discusses the advantages and disadvantages of both approaches (3.4).

The overview given in this section on machine-learning and strategy optimisation is limited to work directly relevant to dialogue strategy design and it is by no means complete. More comprehensive reviews of reinforcement learning can be found in (Sutton and Barto, 1998) and (Kaelbling et al., 1996).

3.1 Introduction to Strategy Learning

The machine learning community distinguishes different types of learning approaches (Ghahramani, 2004). *Supervised* learning follows a "learn-by-example" approach. During the training phase, example problems are presented to the learning agent, together with the correct solutions. For a classification task such as speech recognition, for instance, the data samples are labelled with their orthographic representations, i.e. their corresponding classes. From these examples, probability estimates are obtained for a parametric model. If this model generalises well, then it will also be capable of classifying unseen data into the correct classes.

Unsupervised learning is distinguished from supervised learning by the fact that there is no *a priori* knowledge of the target outputs. The observations are not labelled with their corresponding classes and it is left to the learning algorithm to construct a representation of the input that is helpful for, say, decision making, predicting future inputs or outlier detection. In short, unsupervised learning can be described as "finding patterns in the data above and beyond what would be considered pure unstructured noise" (Ghahramani, 2004).

Reinforcement-Learning (RL) is a form of machine learning which may be viewed as "learning through trial-and-error". It is sometimes introduced as a form of unsupervised learning, although it cannot be regarded as "learning without a teacher". The learning *agent* interacts with its dynamic *environment*, receives feedback in the form of positive or negative rewards according to some reward function and tries to optimise its actions so as to maximize the overall reward. By assigning small negative rewards for every action and a large positive reward for successful completion, an agent can be trained to act in a goal-oriented and efficient manner without providing explicit examples of ideal behaviour or specifying how a given task is to be completed (Sutton and Barto, 1998; Kaelbling et al., 1996).

The nature of the dialogue strategy learning problem is such that we cannot present correct input/output pairs or examples of ideal DM behaviour. In a corpus of human-computer dialogues, we may have annotations indicating successful and unsuccessful dialogues, but we cannot know which system action is best in any given dialogue state in the "long term interest" of the DM. Dialogue strategy optimization is thus not a suitable candidate for supervised learning (Walker, 1993). For a given dialogue corpus, supervised learning can only be used to find the strategy taken most frequently by the system, but it cannot find the optimal one.

However, it is usually possible to define what constitutes the successful completion of a dialogue. In a travel booking scenario, for instance, a positive reward can be associated with the completion of a flight booking. Using RL, the learning dialogue manager can be trained to complete this goal as often and as efficiently as possible but without specifying in advance how the goal should be reached. As we will explain in the remainder of this section, it is up to the DM to automatically explore the vast space of possible dialogue states and actions in order to find an optimal mapping from states to actions.

3.2 Dialogue as a Markov Decision Process

The Markov-Decision-Process (MDP) model serves as a formal representation of human-machine dialogue and provides the basis for formulating strategy learning problems (Biermann and Long, 1996;

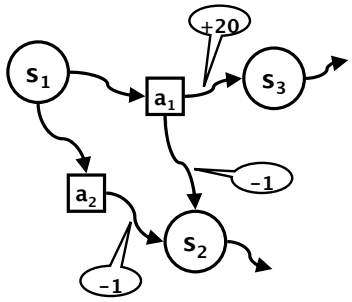

Figure 3 Dialogue as a Markov Decision Process (MDP): The Dialogue Manager acts as a learning agent travelling through a network of interconnected dialogue states. At each time t, the DM is in state s_t, takes action a_t, transitions into state s_{t+1} and receives a reward r_{t+1}. A dialogue strategy $\pi(s) \rightarrow a$ can be viewed as a deterministic mapping from states to actions.

Levin and Pieraccini, 1997; Singh et al., 1999; Levin et al., 2000). Every MDP is formally described by a finite state space S, a finite action set A, a set of transition probabilities T and a reward function R. At each time step t, the dialogue manager is in a particular state $s_t \in S$. It executes the discrete action $a_t \in A$, transitions into the next state s_{t+1} according to the transition probability $p(s_{t+1}|s_t, a_t)$ and receives a reward r_{t+1}. The Markov Property ensures that the state and reward at time $t+1$ only depend on the state and action at time t.

$$P(s_{t+1}, r_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, r_{t-1}, ..., s_0, a_0) = P(s_{t+1}, r_{t+1}|s_t, a_t) \quad (1)$$

Using the MDP model of dialogue, we can visualise the DM as an agent travelling through a network of interconnected dialogue states (see Figure 3). Starting in some initial state, the DM transitions from state to state by taking actions and collecting rewards as it goes along. Since the user response to a system action is uncertain, the transitions are non-deterministic, meaning that the choice of some action a in some state s at time t does not always lead to the same next state and reward at time $t+1$.

The MDP model allows a dialogue management strategy (or policy) $\pi(s) \rightarrow a$ to be viewed as a deterministic[2] mapping from states to actions: For every state s, the policy selects the next system action a. It also enables us to formalize dialogue management as a mathematical optimization problem. The optimal policy π^* is the policy that maximizes the cumulative reward over time, i.e. the policy that selects those actions that yield the highest reward over the course of the dialogue.

3.2.1 State-Space and Action Set

In order to map human-computer dialogue onto the MDP model, a well defined representation of the dialogue state-space S and the system action set A is required. In theory, the action set may cover a vast variety of system utterances and the state may include all the currently available information regarding internal and external processes controlled by the dialogue system. However, while it is desirable to capture as many aspects of the dialogue state and the action set as possible, the choice of granularity is constrained by limits of computational tractability. Finding a compact state representation which satisfies the Markov property is often particularly difficult.

For slot-filling dialogues, a common approach is to base the number of state variables on the number of slots that need to be filled and grounded. For a cinema booking system, for instance, this might cover the name of the film, the desired day and the desired screening time. The total number of possible states is then determined by the number of states each individual slot can be in. If all possible values of each slot are considered as separate states, the total number of states easily becomes unmanageable. It is

[2]Stochastic dialogue behaviour can be achieved using a probabilistic mapping from states to actions but a predictable, deterministic strategy is usually preferred.

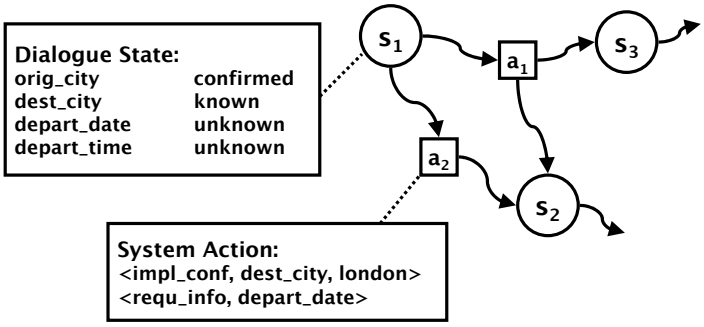

Figure 4 The Markov-Decision-Process model requires an exact specifi cation of the system action set and the state space. In a simple flight booking dialogue system, for example, the dialogue state may encode the status of the slots *origin city*, *destination city*, *departure date* and *departure time*. System actions can be represented as tuples consisting of a *dialogue act*, a *slot name* and an optional *slot value*.

therefore standard practice to use slot-status categories such as “unknown”, “known” and “confirmed”. (See (Williams, 2003) for an overview of state space representations used by different research groups.)

Similarly, the system action set is often limited to a small number of actions such as “Request all”, “Request n slots”, “Verify n slots”, “Verify All”, or “Quit”. A slightly more flexible alternative is to represent each action as a tuple consisting of a speech act, a slot name and a slot value, as shown in Figure 4. Note that the number of conceptually different system prompts depends on the choice of action set, but the wording of individual system prompts is not affected by the high-level representation of the action set.

3.3 Reinforcement Learning

We can define R to be the total discounted return the learning dialogue manager receives when starting at time t. A discount factor γ, with $0 \leq \gamma \leq 1$, can be used to weight immediate rewards more strongly than distant future rewards.

$$R = \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + ... = \sum_{k=1}^{\infty} \gamma^k r_{t+k} \tag{2}$$

Following the Maximum Expected Utility Principle (Keeney and Raiffa, 1976; Russell and Norvig, 1995), our goal is to find the optimal policy π^* that maximizes the expected value of R. Reinforcement learning offers a variety of algorithms for finding π^* within the MDP framework through trial-and-error interaction between the learning agent and its dynamic environment. Such algorithms are commonly based on the definition of a value-function $V^\pi(s)$ that estimates the expected return of being in a given state s given a policy π

$$V^\pi(s) = E_\pi(R|s_t = s). \tag{3}$$

Alternatively, we can define a value function $Q^\pi(s, a)$ that estimates the expected return of taking action a in a given state s and following π thereafter.

$$Q^\pi(s, a) = E_\pi(R|s_t = s, a_t = a) \tag{4}$$

Using the following short-hand notation for the transition probabilities and the expected rewards

$$\mathcal{T}(s', a, s) = P(s_{t+1} = s'|a_t = a, s_t = s) \tag{5}$$

$$\mathcal{R}(s', a, s) = E(r_{t+1}|s_{t+1} = s', a_t = a, s_t = s) \tag{6}$$

we can rewrite $V^\pi(s)$ and $Q^\pi(s, a)$ as follows:

$$V^{\pi}(s) = \sum_{s' \in S} \mathcal{T}(s', \pi(s), s)(\mathcal{R}(s', \pi(s), s) + \gamma V^{\pi}(s')) \tag{7}$$

$$Q^{\pi}(s, a) = \sum_{s' \in S} \mathcal{T}(s', a, s)(\mathcal{R}(s', a, s) + \gamma V^{\pi}(s')) \tag{8}$$

If the transition probabilities $\mathcal{T}(s', a, s)$ and the reward function $\mathcal{R}(s', a, s)$ are known in advance, Dynamic Programming techniques, Value Iteration, or Policy Iteration can be used to find an optimal policy π^* *off-line*, i.e. without any interaction of the DM with its environment. If $\mathcal{T}(s', a, s)$ and $\mathcal{R}(s', a, s)$ are unknown, we cannot systematically account for every possible combination of state and action and instead the agent needs to interact with its environment to learn these probabilities.

The Q-Learning algorithm (Watkins and Dayan, 1992b) is one of the simplest forms of *on-line* reinforcement-learning. It works by maintaining Quality-values (Q-values) for every pair (s, a) of state s and action a. These values estimate the expected return of taking action a in state s and following π thereafter, as expressed in Equation 4.

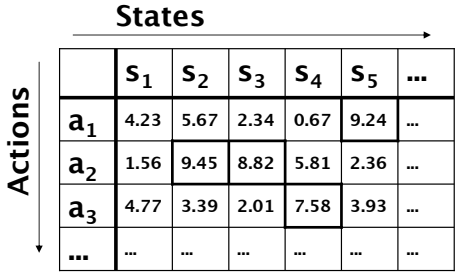

Figure 5 The optimal dialogue strategy selects the system action a with the highest expected value in each state s.

One may visualise the strategy learning process with the help of a matrix of states versus actions, as shown in Figure 5. The process is started by arbitrarily initializing the Q-values for all state-action pairs. As the learning dialogue manager interacts with the (simulated) user, the Q-values are iteratively updated to become better estimates of the expected return of the state-action pairs. After each system action a in state s, the user response results in a transition to state s' and a reward r and the Q-value is accordingly recalculated using

$$Q(s, a) := (1 - \alpha)Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a')) \tag{9}$$

where α represents a learning rate parameter that decays from 1 to 0. Once the Q-values for each state-action pair have been estimated[3], the optimal policy can be found using

$$\pi^*(s) = \arg\max_{a} Q^*(s, a). \tag{10}$$

To ensure convergence of the Q-values (Watkins and Dayan, 1992a), the values of α at each cycle i must conform to

$$\sum_{i=1}^{\infty} \alpha_i = \infty \qquad \text{and} \qquad \sum_{\mathrm{i}=1}^{\infty} \alpha_{\mathrm{i}}^2 < \infty. \tag{11}$$

During the learning process, it is necessary to achieve a compromise between exploration (trying out random actions) and exploitation (taking actions which have already been shown to lead to high rewards). The most straightforward solution is an ϵ-greedy control policy. At each step, a random number $0 < \beta < 1$ is generated and the next action is selected randomly if $\beta < \epsilon$. If $\beta \geq \epsilon$, the best action is taken. ϵ is usually selected so as to decrease with the number of training cycles from 1 to 0.1.

[3]In theory, all state-action combinations must be visited an infi nite number of times to guarantee that the policy is optimal.

3.4 Model-based and Simulation-based Learning

Dialogue strategy learning has been approached using two different forms of reinforcement learning which are referred to as *model-based* and *simulation-based* learning.

The model-based approach uses the training data to build a complete model of the environment, i.e. it estimates the transition probabilities $\mathcal{T}$ from a corpus in which the state transitions have been logged. Parameter estimation can be done using simple Maximum Likelihood Estimation based on the relative frequency of occurrence of each transition

$$\mathcal{T}(s', a, s) = \frac{count(s', a, s)}{count(s, a)}. \tag{12}$$

The model-based approach neglects the actual user responses to system actions - it only considers the system-state transitions and the rewards associated with them. Once the transition probabilities have been estimated and the reward function has been set by the system designer, the optimal strategy can be found *off-line* (i.e. without any interaction between the DM and its environment) as described in the preceding section. This approach has been heavily used by research groups between 1998 and 2002, most notably by M. Walker, D. J. Litman and S. Singh (Walker et al., 1998; Singh et al., 1999; Litman et al., 2000; Singh et al., 2000; Walker, 2000; Singh et al., 2002) and also in very early work by Levin et al. (Levin and Pieraccini, 1997).

The model-based approach is a simple form of reinforcement-learning and does not require a user simulation tool. However, it has not been used in more recent research due to a number of significant deficiencies:

1. Currently available corpora are not large enough to reliably estimate transition probabilities for practical systems. Even for trivial problems, the number of possible combinations of states and actions easily exceeds $100,000$[4] and the provision of training data large enough to explore this state space and obtain reliable estimates is simply not tractable. To cope with data sparsity, state-aggregation and back-off schemes for parameter estimation have been proposed by (Denecke et al., 2004; Goddeau and Pineau, 2000), but these are essentially smoothing methods and they offer no guarantee that the learned strategy is indeed robust.
2. When learning from a fixed corpus, the DM can only use state-action combinations that were explored at the time of the corpus data collection. It cannot try out entirely new strategies, since no transition probabilities can be computed for unseen state-action combinations. A deviation in user behaviour during the live operation of the system is likely to move the system into an unexplored portion of the state space, possibly causing the system to fail. Furthermore, the optimal strategy is not guaranteed to be present in the training corpus.
3. The model-based approach has the principal drawback, that the dialogue model (i.e. the state-space and action set representation) needs to be known in advance so that the corpus can be annotated correspondingly for estimating the state transition probabilities. Preferably, the state space and action set should be fixed even before the data collection in order to be able to log state transitions during the collection. However, if the chosen state space representation or action set turns out to be problematic, it is very difficult to change the dataset or its transcriptions.

The simulation-based approach, pictured in Figure 1 on page 2, is an alternative to *off-line* model-based learning and involves two phases: First a user simulation tool is trained on the given corpus using supervised learning. The dialogue strategy is then learned *on-line* through interaction between the simulated user and the dialogue manager using an algorithm such as Q-Learning as explained in the previous subsection. The simulation-based approach requires a reliable user model which generalises to unseen dialogue situations. As such it is the more complex approach but it offers the following advantages:

1. The simulated user allows any number of training episodes to be generated, so that the learning dialogue manager can exhaustively explore the space of possible strategies.

[4]Henderson et al. (Henderson et al., 2005), for example are working with dialogue state spaces on the order of 10^{87}. Currently available corpora in comparison typically contain on the order of 10^3 annotated dialogues.

2. It enables strategies to be explored which are not in the training data. The learning DM can deviate from the known strategies and try out new and potentially better strategies.
3. The system state space and action set do not need to be fixed in advance, because the system is not trained on corpus data. If the given representation turns out to be problematic it can be changed and the system retrained using the simulated user[5].

Simulation-based learning has been the preferred form of learning dialogue strategies over recent years and can be found in (Eckert et al., 1997; Levin et al., 2000; Scheffler, 2002; Scheffler and Young, 1999, 2000, 2001, 2002; Pietquin, 2004; Georgila et al., 2005a; Schatzmann et al., 2005a).

[5] A change in the system action set, however, may still require modifications to the dialogue manager's interface with the simulated user.

4 Statistical User Modelling Techniques for Simulation-Based Learning

4.1 User Modelling in Spoken Dialogue Systems and Natural Language Processing

As mentioned in the introduction to this paper, research on User Modelling has a long history in the field of Natural Language Processing (NLP) and Spoken Dialogue Systems (SDS), particularly in areas concerned with the design of adaptive and adaptable systems. Traditionally, the majority of work on User Modelling in NLP and SDS is linguistically motivated and of a non-statistical nature. The aim of research in this area is typically to construct a representative model of the individual user in order to describe his or her state during a course of interaction with the system. In a spoken dialogue environment this may include information about the user's goal or knowledge regarding previously exchanged pieces of information. Depending on the application domain, the model can also include the user's preferences, knowledge about the subject matter, beliefs, capabilities, level of satisfaction with the system behaviour and so on.

The inspiration for building user models for use in spoken dialogue systems originally stems from research on human-human interaction. People intuitively use models of their dialogue partners when they engage in a conversation. The style of language and level of formality, for instance, depends on the mental model of the interlocutor. Our mental model can also help us to understand the dialogue partner's intentions, even if they have not been well articulated or worded in a precise manner. In a similar fashion a dialogue system can benefit from consulting a user model.

A recent review on the "Synergies and Limitations" of User Modelling and NLP by Zukerman and Litman (Zukerman and Litman, 2001) shows that research has investigated the use of user models in almost all subfields of spoken dialogue systems. Particularly active areas are Natural Language Generation (NLG) and Natural Language Understanding (NLU).

In NLG, content generators can consult user models to produce output which is tailored to specific attributes of the individual user. Examples of work in this area demonstrate how systems can attempt to match characteristics such as expertise and interests, or domain-specific information such as medical records, investment portfolios, etc. (Wallis and Shortliffe, 1985; Stock, 1993; Binsted et al., 1995; Bontcheva, 2005; Harvey et al., 2005). In NLU, user modelling often takes the form of plan recognition. The aim here is to make inferences about a user's preferences and goals in order to (for example) respond better to ill-formed queries (Carberry, 2001) or to detect expressions of doubt (Carberry and Lambert, 1999).

According to Zukerman and Litman, user modelling is less frequently used in dialogue management. Very early examples of research in this area are dominated by knowledge-based formalisms and various types of logic aimed at modelling the complex beliefs and intentions of agents (Moore, 1977; Hustadt, 1994; Dols and van der Sloot, 1992; Bretier and Sadek, 1996). In more recent years, dialogue systems have tended to focus on cooperative, task-oriented rather than conversational forms of dialogue so that user models are now typically less complex. User models in dialogue management now often take the form of simple rules or attribute-value pairs (Komatani et al., 2003). As in NLG and NLU, the general aim is usually to tailor the behaviour to specific user attributes. These can be either fixed attributes, such as "a preference for afternoon meetings" (Elzer et al., 1994) or dynamic attributes, such as the cognitive load on a user during a conversation (Berthold and Jameson, 1999).

4.2 Statistical User Models for Simulating Dialogue

The recent work on user modelling for strategy learning with a simulated user differs in many respects from traditional work on user modelling in SDS and NLP. The purpose of the user model in this new context is not to capture the individual characteristics of a specific user, but to form the basis of a user simulation tool which is representative of a diverse user population and which takes the role of a synthetic dialogue partner for the learning dialogue manager during the development phase of a new system.

As such, the user model must be able to predict user responses that a real user might give in the same dialogue context and it must also represent the statistical distribution over all types of user behaviour and user responses. These requirements clearly favour a statistical approach to user modelling. Most

importantly, the probabilistic nature of statistical models allows for a degree of "lifelike" randomness in user behaviour. Very early research on dialogue system design and evaluation has explored the use of deterministic user models for simulating dialogues (Smith, 1998; Guinn, 1998; Ishizaki et al., 1999; Lin and Lee, 2001), but these cannot naturally account for the variability in real human user behaviour.

The second advantage of statistical models is that the model parameters can be estimated using real human-computer dialogue data. Hand-coded models require a very careful choice of parameter values, which is difficult because human user behaviour is poorly understood and system-specific factors or user population-dependent aspects are hard to estimate using "common sense". The statistical, data-driven approach of learning appropriate parameter values from training data rather than relying on heuristics or empirical principles is less likely to introduce a bias in user behaviour. The user model is also likely to be more robust, more scalable and easier to port to a different domain or dataset.

Statistical predictive user models in the context of dialogue strategy learning have been explored by a number of research groups over the last decade. It is possible to classify the different approaches with regard to the level of abstraction at which they model dialogue. This can be at either the acoustic level, the word level or the intention-level. The latter is a particularly useful compressed representation of human-computer interaction. Intentions cannot be observed, but they can be described using speech-act and dialogue-act theory (Searle, 1969; Bunt, 1981; Traum, 1999).

In recent years, simulation on the intention-level has been most popular. This approach was first taken by (Eckert et al., 1997) and has been adopted in later work on user simulation by most other research groups (Levin et al., 2000; Scheffler, 2002; Pietquin, 2004; Georgila et al., 2005a; Cuayahuitl et al., 2005). Modelling interaction on the intention-level avoids the need to reproduce the enormous variety of human language on the level of speech signals or word sequences. Examples of user simulation on the word-level (Araki et al., 1997; Watanabe et al., 1998) or acoustic level (Lopez-Cozar et al., 2003; Filisko and Seneff, 2005) are rare and their robustness, portability and scalability is somewhat limited.

Several different statistical user modelling techniques have been applied to the problem of simulation-based learning. All of them exploit the Markov assumption that the next user action can be predicted based on some representation of the current state of the dialogue. The state representation typically includes the previous system action, but may also cover longer dialogue histories and other variables representing the user goal, user memory, and so forth.

In general, Markov Models are based on the assumption that the likelihood of occurrence of an event depends only on a fixed number of previous events and hence the relevant history can always be encoded in a finite state space. Given a sequence of events $x_n, x_{n-1}, ..., x_{n-m+1}$, a probability distribution can be estimated over all events following this sequence in a set of training data. The likelihood that a certain event x_{n+1} follows the given sequence can then be computed using the conditional probability $p(x_{n+1}|x_n, x_{n-1}, ..., x_{n-m+1})$. Outside the field of spoken dialogue systems research, Markov Models are a commonly used statistical user modelling technique. A typical application is the prediction of documents a user is most likely to request next when browsing web-pages (Bestavros, 1996; Horvitz et al., 1998). In the field of speech recognition and spoken dialogue systems, *n-grams* are a popular form of Markov Model and well-known for their use in predicting the likelihood of different word sequences in statistical language modelling (Jelinek, 1976; Rabiner and Juang, 1993).

4.2.1 N-grams

In the context of dialogue strategy evaluation and learning, statistical predictive user models were first suggested by Eckert, Levin and Pieraccini (Eckert et al., 1997, 1998). Their pioneering work introduces a simple n-gram model for predicting the user action $\hat{a}_{u,t}$ at time t that is most probable given the dialogue history of system and user actions. In practice, data sparsity may prohibit the use of long dialogue histories. Eckert et al. approximate the full history with a bigram model, using $a_{s,t}$ to denote the system action at time t:

$$\hat{a}_{u,t} = \arg\max_{a_{u,t}} P(a_{u,t}|a_{s,t-1}, a_{u,t-1}, a_{s,t-2}, ..., a_{u,0}, a_{s,0}) \quad (13)$$

$$\approx \arg\max_{a_{u,t}} P(a_{u,t}|a_{s,t-1}) \quad (14)$$

The bigram model suggested by Eckert et al. has the advantage of being purely probabilistic and fully domain-independent. Its weakness is that it does not place any constraints on the simulated user behaviour. Any user action may follow any system action, regardless of the rest of the dialogue history. The generated responses may correspond well to the previous system action, but they often do not make sense in the larger context of the dialogue since all state information is neglected. Dialogues simulated with a bigram user model often continue for a long time, with the user continuously changing his goal, repeating information or violating logical constraints.

```
System:     Where are you flying from?
User model: I am flying from New York.
System:     Where do you want to go?
User model: I'm going to New York.
```

In later work by the same group of authors, Levin, Eckert and Pieraccini (Eckert et al., 1997, 1998; Levin et al., 2000) describe how the pure Bigram model can be modified to account for a more realistic degree of conventional structure in dialogues. Instead of allowing any user response $a_{u,t}$ to follow a given system action $a_{s,t-1}$, only the probabilities for "sensible" pairs of user response and system action are estimated, while all others are assumed to be zero. For a system request for a slot S_i, for instance, only the probability is estimated that the user actually provides a value for slot S_i and the probability that he or she provides a value for a different slot S_k.

$$P(provide\_info\ S_i|request\_info\ S_i) \quad (15)$$

$$P(provide\_info\ S_k|request\_info\ S_i) \quad (16)$$

The set of probabilistic user model parameters selected by Levin et al. implicitly characterizes the level of cooperativeness and the degree of initiative taken by the simulated user. The Levin model places stronger constraints on the user actions than the pure Bigram model, but it also makes assumptions concerning the format of the dialogue. If the dialogue manager deviates from the anticipated simple mixed-initiative slot-filling dialogue format then a new set of parameters is needed.

Like the Bigram model, the Levin model does not ensure consistency between different user actions over the course of a dialogue because of the flawed assumption that every user response depends only on the previous system turn. As in the Bigram Model, the user actions can violate logical constraints and the synthetic dialogues often continue for a long time, with the user continuously changing its goal or repeating information.

4.2.2 Graph-based Models

Scheffler and Young (Scheffler, 2002; Scheffler and Young, 1999, 2000, 2001) propose the use of a graph-based model to overcome the lack of goal consistency that the Levin model suffers from while maintaining variability in user behaviour. Their work combines deterministic rules for goal-dependent actions and probabilistic modelling to cover the conversational behaviour of the simulated user.

In Scheffler and Young's model, all of the possible "paths" that a user may take during a dialogue need to be mapped out in advance in the form of a network. The arcs of the network represent actions and the nodes represent "choice points". Some of the latter are identified as probabilistic choice points, representing a "random decision" by the simulated user. Suitable probability values for these choice points are estimated from data. The remaining nodes in the network are deterministic choice points. The route taken at these points depends on the user goal, which remains fixed over the course of the dialogue and

is represented as a table consisting of slot-name, slot-value pairs with associated status variables. The explicit representation of the user goal ensures that the user always acts in accordance with his goal. Figure 6 illustrates the concept of choice points using an example from the cinema-ticket booking domain.

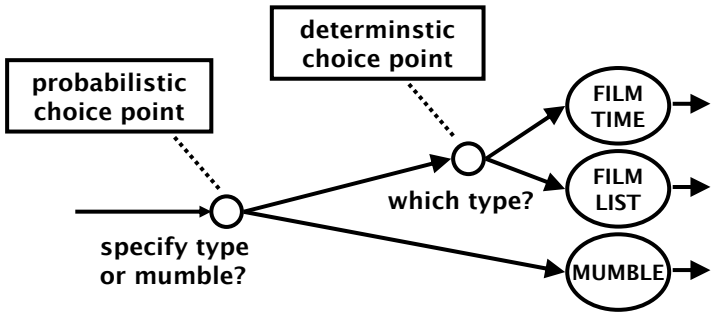

Figure 6 In Scheffler and Young's model each *choice point* represents a (possibly unconscious) user decision that influences the further direction of the dialogue. The diagram shows an example from the cinema-ticket booking domain. At the left (probabilistic) choice point, a "random" decision is made whether the user utters something incomprehensible (a "mumble" intention) or moves to the next choice point. In the latter case, the user reaches a deterministic choice point, at which he/she can either request a listing of films or ask for showtimes. The choice made at this point depends on the user goal.

The major disadvantage of Scheffler and Young's work is the high dependence on domain-specific knowledge. The exhaustive specification of all possible dialogue path is the critical issue. The authors show that this task can be partly automated if a prototype dialogue system is available and the range of acceptable user actions has been well defined for each state of the prototype dialogue manager. Nevertheless, the identification of probabilistic and deterministic choice points still requires manual effort.

4.2.3 Bayesian Networks

Pietquin (Pietquin, 2004, July 2005; Pietquin and Beaufort, 2005; Pietquin and Dutoit, 2005) combines features from Scheffler and Young's work with the Levin Model in an attempt to avoid the manual effort involved in building networks of choice points while ensuring that actions are consistent with the user's goal. The core idea of his work is to condition the set of probabilities selected by Levin et al. on an explicit representation of the user goal and memory.

$$P(provide\_info\ S_i | request\_info\ S_i,\ goal,\ memory) \tag{17}$$

As in Scheffler and Young's work, the user goal is simply a table of <slot name, slot value> pairs with associated status variables. Pietquin uses the status variables to rank the user's priority for each slot value and to track how often a certain piece of information has been mentioned by the user during the conversation with the system. Although these models of the user goal and memory are rather crude, they do acknowledge that a better understanding of the user *state* and its effect on conversational behaviour is needed to make accurate predictions of user behaviour.

In his work, Pietquin hand-selects all model parameters using empirical principles and common sense. Suitable probability values (eg. Equation 17) are set by hand. Pietquin suggests that in future work the conditional dependencies in the user model be graphically visualised and implemented as a Bayesian network, as illustrated in Figure 7. According to Pietquin, the network can be easily extended to include factors such as the level of cooperativeness, the degree of initiative, etc. It is however not intuitively clear which parameters are potential candidates for modelling these user attributes and how they can be trained on human-computer dialogue data.

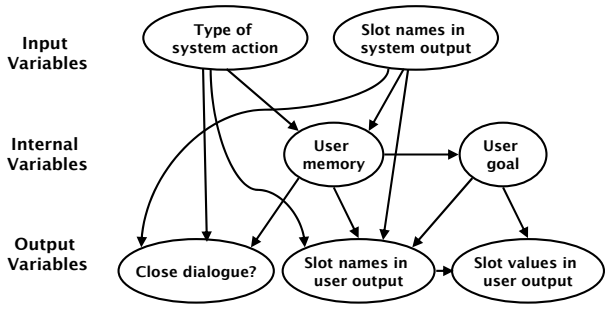

Figure 7 Pietquin suggests a Bayesian network approach to user modelling. The input variables to the network are the type of system action (eg. "greeting", "request info", etc.) and the slot names mentioned by the system. The output variables are the slot names and values mentioned by the user and a boolean variable indicating whether the user will hang up or not. The user goal and the user memory function as internal variables of the network.

4.2.4 Machine-Learning techniques

Recent work by Georgila, Henderson and Lemon (Georgila et al., 2005a) returns to Markov Models and explores the use of richer state descriptions, longer dialogue histories and machine-learning techniques. In Georgila et al's work, dialogue is viewed as a sequence of *Information States* (Bos et al., 2003), each of which is represented as a large feature vector describing the current state of the dialogue, the previous dialogue history and any ongoing actions. The rich state description helps to compensate for the Markov assumption, but it also requires a large amount of training data to reliably estimate model parameters.

Georgila et al. investigate two different methods for predicting the next user action given a history of information states. The first method reuses the n-gram model suggested by Eckert et al., but with n ranging from 2 to 5 in order to cover longer spans of dialogue history. Georgila et al. find that best results can be obtained with a 4-gram, ie. by using a history of 4 information states to predict the next user action. The authors do not comment on data sparsity problems, but given that the state space used in the experiments has on the order of 10^{87} states, it must be presumed that many state sequences never occur in the training data.

The second method examined by Georgila et al. uses linear feature combination to map from a state s to a vector of real-valued features $f(s)$. Most of these features are binary indicating the presence or absence of a piece of information (e.g. destination place known, departure time confirmed, etc) and the remainder are continuous real values (e.g. estimated word error rate). In total, there are around 290 such features (Georgila et al., 2005b,a). Supervised learning is used to estimate a set of weights w_a for each action a which describe how "useful" each vector element of $f(s)$ is for predicting a. Once the weights are estimated, a probability distribution $\hat{P}(a|s)$ can be computed over all user actions by applying a normalised exponential function to the inner product of $f(s)$ and w_a. ($f(s)^T$ denotes the transpose of $f(s)$.)

$$\hat{P}(a|s) = \frac{\exp(f(s)^T w_a)}{\sum_a \exp(f(s)^T w_a)} \tag{18}$$

The linear feature combination approach is attractive, because it allows the relevance of each state feature for predicting the next user action to be learned automatically from data. Since the Information State includes not only the current state of the dialogue but also information about the previous dialogue history, any aspect that possibly influences user behaviour can in principle contribute to the prediction of the next action.

4.2.5 Hidden Markov Models

Very recent work by Cuayahuitl, Renals, Lemon, and Shimodaira (Cuayahuitl et al., 2005) presents a method for dialogue simulation based on Hidden Markov Models (HMMs). Their method generates

both system and user actions, the objective being to extend an existing small corpus of human-computer dialogue data with simulated dialogues.

Cuayahuitl et al. experiment with different variations of HMMs. The most advanced one is an Input-Output-HMM (IOHMM), as shown in Figure 8 (adapted from (Cuayahuitl et al., 2005)). The model is characterised by a set of (visible) states $S = \{S_1, S_2, ..., S_N\}$ and a set of observations $V = \{V_1, V_2, ..., V_M\}$. The states S represent system turns, and the observations V correspond to the system action set. The state at time t is denoted using q_t and $a_{s,t}$ is the system action at time t. The user responses are represented using a set of user intentions $H = \{H_1, H_2, ..., H_L\}$ and the user action at time t is denoted using $a_{u,t}$. The behaviour of the model is governed by a set of transition probabilities $P(q_{t+1}|q_t, a_{s,t})$ and a set of output probabilities $P(a_{s,t}|q_t, a_{u,t-1})$. User responses are predicted using a set of user model probabilities $P(a_{u,t}|q_t, a_{s,t})$.

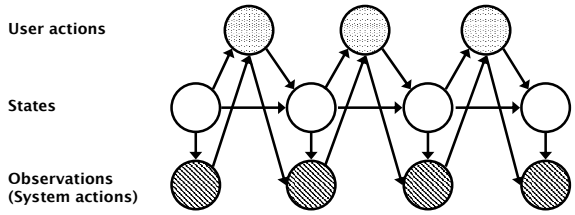

Figure 8 Cuayahuitl et al. suggest a dialogue simulation method based on Input-Output-HMMs, an extension of standard Hidden Markov Models. The figure shows the states representing system turns (white circles), the observations representing system actions (dark shaded circles) and the user actions (light shaded circles). The arcs denote conditional dependencies.

Instead of training a single giant IOHMM for simulating complete dialogues, all conversations in the dialogue corpus are divided into subgoals and a separate model is trained for each subgoal. A full dialogue is viewed as a concatenation of subgoals and a bigram model $P(g_n|g_{n-1})$ is used to predict the sequence of subgoals.

4.3 User Simulation-Based Learning from a Statistical User Modelling Perspective

Quite remarkably, almost all of the research groups investigating the topic of user simulation-based strategy learning have a background in dialogue systems design rather than user modelling. To date, the user modelling community has taken fairly little notice of the ongoing work on simulation-based reinforcement-learning in SDS, although many of the modelling techniques used in strategy learning are in fact similar to standard statistical user modelling techniques as we have pointed out in the previous section. Vice versa, the SDS community is also largely unaware of related research in the field of user modelling and references to user modelling research in papers on simulation-based learning are almost non-existent. We argue that much could be gained by casting the user simulation problem in SDS as a statistical user modelling problem and we provide support for this claim in the following two subsections.

4.3.1 Content and Collaboration-based Modelling

Research on predictive statistical user modelling commonly distinguishes between *content-based* and *collaboration-based* forms of modelling (Zukerman and Albrecht, 2001). While content-based approaches use the user's history of actions to predict future actions, collaboration-based approaches predict the user's next action based on the actions taken by other users in the same state. Content-based techniques are typically employed when a sufficient amount of data describing the user's past behaviour is available and it appears to be a reliable indicator of his future behaviour. "Persistence of interest" is important for content-based systems. In contrast, collaboration-based approaches are attractive when large amounts of data are available from other previous users and when user behaviour is sufficiently homogeneous for actions taken by users in the same state to be likely actions of the new user. Collaboration-based systems are less influenced by long-term information and exploit better context-dependent data (Burke, 2002; Resnick and Varian, 1997).

Despite being a common classification for statistical user modelling techniques, the concepts of content-based and collaboration-based learning have so far not been used in the context of user modelling for simulation-based strategy learning. Nevertheless, it is interesting to consider the problem of modelling user dialogue behaviour in the light of the above classification. Indeed, all of the modelling techniques which are currently available for simulation-based learning are primarily collaborative approaches. This is not surprising given that we typically have a fair amount of dialogue data from many different previous users but only a small amount of data per user.

However, many characteristics of individual user behaviour (such as expertise and cooperativeness) are not covered well by the currently available techniques. As we will further discuss in the section on user model evaluation, the current user models generate a variety of user behaviour which reflects the statistical distribution in the training data, but they do not acknowledge that some aspects of user behaviour are likely to remain fixed over the course of a dialogue while others are likely to be variable.

It is hence desirable to combine the collaborative models currently available with content-based modelling components to ensure that user actions are taken in accordance with the previous user dialogue history and that they reflect individual user characteristics. The models presented by Scheffler and Young and by Pietquin (reviewed above) both introduce the concept of a user goal in their user models to achieve goal-consistency over the course of a dialogue. Pietquin further introduces a simple model of user memory and suggests further work on modelling user satisfaction. As we have discussed in the previous section, the use of such content-based forms of modelling in otherwise collaborative models is clearly beneficial to the realism of the simulated user behaviour. We argue that user dialogue behaviour is in fact a prime candidate for combining content-based and collaboration-based approaches.

It is interesting to note that it is well known in the user modelling community that collaboration-based approaches naturally tend to model an average user unless careful steps are taken to model individual users or user groups (such as experts and novices, for instance) (Alspector et al., 1997; Horvitz et al., 1998). It is only in very recent evaluation efforts (Schatzmann et al., 2005a) that this pitfall has been highlighted in research on strategy learning. As Zukerman and Albrecht (Zukerman and Albrecht, 2001) formulate: “the fundamental inability of the collaborative approach to represent the idiosyncrasies of individual users calls for a solution which combines both (content-based and collaboration based) types of modelling approaches.” We believe this to be equally true for user modelling in the context of dialogue strategy learning.

4.3.2 Concept Drift

A second concept from the realm of machine-learning for statistical user modelling which may be of use for future research on simulation-based strategy learning is *Concept Drift* (Widmer and Kubat, 1996; Webb et al., 2001). Concept drift describes the problem that user attributes change over the course of interaction with a system. Traditionally, this has been of particular relevance for user modelling in the areas of information retrieval and personalized news recommender systems. Systems of this type typically attempt to recommend documents to the user based on previously requested documents. The underlying assumption of “persistence of interest” (Lieberman, 1995), however, is often flawed because a user’s interests, preferences and behaviour can change. A person who has read several news articles concerning a certain current event, for example, may be dissatisfied if the system recommends yet another article on the same topic a week later.

A similar problem applies to simulation-based learning. Some of the user models currently available for modelling user dialogue behaviour include the notion of a user goal, but they all assume that the user goal stays fixed over the course of a dialogue. This assumption is of course simplistic. In reality, a user might start a conversation (say with a flight booking system) with a vague idea of booking “a low-cost flight from London to Madrid for the Christmas Holidays”. It is only as the dialogue evolves that the user goal becomes more concrete, for example as details regarding the exact departure date and time are added. The user may not have a specific airline in mind at the start of the conversation but may still express a preference when reminded about a frequent-flyer programme. The user goal may also change as the user

becomes aware of system limitations: For example, the user may have a priority for sitting at a window-seat, but may change his mind when he realizes that the system cannot accommodate his request. He may also decide to drop this priority if only aisle seats are available on a significantly cheaper flight option.

In research on statistical user modelling outside of spoken dialogue systems, suggestions have been made to address the problem of concept drift. One approach is to weight recent observations more strongly than old observations (Webb and Kuzmycz, 1996), another idea is to use windowing techniques to adjust the span of training data which is used to make a prediction (Klinkenberg and Renz, 1998). A common problem is that the amount of relevant recent training data is small, so that the use of back-off models has been suggested in instances when no reliable prediction can be made using the recent data (Chiu and Webb, 1998; Billsus and Pazzani, 1995).

However, all of the above suggestions apply to purely content-based forms of user modelling and are likely to require significant modifications to be useful for the primarily collaborative modelling techniques used for simulating user dialogue behaviour. One may also argue that the challenge of modelling changes in the user goal over the course of a dialogue is potentially even more complex than modelling the lack of persistence in user interests in other user modelling applications because it is not just a temporal problem. The key issue is that we require a better understanding and a better model of how the system's behaviour causes changes in the user goal. This is especially an issue in spoken dialogue systems where recognition performance can vary significantly over time thereby having a significant impact on the dialogue.

5 Evaluation Techniques

The evaluation of statistical user models is an ongoing area of research and a variety of techniques can be found in the literature. As described in (Zukerman and Albrecht, 2001), these are typically methods adapted from other fields of research such as Information Retrieval or Machine Learning. A distinction can be made between *direct* methods of evaluation using metrics such as Accuracy or Precision and Recall and *indirect* methods which employ metrics such as Utility.

Direct evaluation methods assess the user model by testing the quality of its predictions. Precision and Recall for instance can be used to test a Recommender system by measuring what proportion of recommended items is relevant and what proportion of the relevant items is recommended (Raskutti et al., 1997). Similarly, Accuracy measures what percentage of predicted events actually occur. Accuracy is a well-known measure in Spoken Dialogue Systems for measuring the performance of speech recognizers and it is also widely used in User Modelling for measuring the performance of predictive models (Gervasio et al., 1998).

Indirect evaluation metrics such as Utility attempt to measure the quality of a user model by evaluating the effect of the model on the system performance. The goal is to obtain a measure of the benefit or penalty involved in making a "good" or "bad" prediction. Indirect methods can be said to be closer to studies with real users, because they take the user model's impact on the user into account (Zukerman and Albrecht, 2001).

Simulation-based strategy learning is an interesting application from a statistical user modelling viewpoint, because it has been approached using both direct as well as indirect methods of evaluation. Direct methods have been used to evaluate the quality of the simulated user output, and indirect methods have been used to evaluate the performance of the strategies that we can learn with the available user models. The latter part is particularly interesting, because it is one of the applications of statistical user modelling where the utility of some user model A has been evaluated by testing the system performance on a competing user model B.

5.1 Evaluation of Simulation Quality

The quality of the simulated user actions generated by the user model has been assessed using a variety of different direct evaluation methods, including Perplexity and Precision and Recall. One may argue that it is ideal to use these measures complementarily rather than alternatively, because each of them answers a slightly different question.

5.1.1 Precision and Recall

A possible starting point for evaluating a statistical model of user dialogue responses is to assess whether the model can generate human-like output. In other words, does it produce responses that a real user might have given in the same dialogue context? If human-computer dialogue data from the same domain is available, this can be tested by comparing real and simulated user responses using precision and recall as suggested in (Schatzmann et al., 2005a).

A corpus of real human-computer dialogues is split into a training and a test set and each dialogue is annotated as a sequence of turns, with each turn consisting of a variable number of actions, as shown in Figure 9. Evaluation is then carried out on a turn by turn basis. Each of the system turns in the test set is separately fed into the simulation, together with the corresponding dialogue history and the current user goal. The response turn generated by the simulated user is then compared to the real response given by the user in the test set.

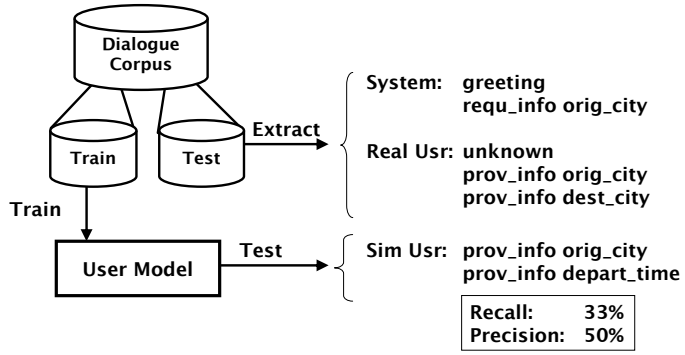

Figure 9 To evaluate how well the user model predicts the actions of a real user, dialogue scenarios are extracted from the unseen test set and presented to the simulated user. The generated user responses are compared to the real user responses and the accuracy of the predictions is evaluated using Precision and Recall.

Precision and Recall are used to quantify how closely the synthetic turn resembles the real user turn. These metrics are a common measure of goodness in user modelling outside SDS (Zukerman and Albrecht, 2001). Recall (R) measures how many of the actions in the real response are predicted correctly. Precision (P) measures the proportion of correct actions among all the predicted actions. A simulated action is considered correct if it matches at least one of the actions in the real user response.

$$P = 100 * \frac{Correctly\ predicted\ actions}{All\ actions\ in\ simulated\ user\ response\ turn} \tag{19}$$

$$R = 100 * \frac{Correctly\ predicted\ actions}{All\ actions\ in\ real\ user\ response\ turn} \tag{20}$$

To rank approaches using a single measure, the balanced F-measure (van Rijsbergen, 1979) can be used which represents the harmonic mean of P and R:

$$F = \frac{2PR}{P+R} \tag{21}$$

As reported in (Schatzmann et al., 2005a), some of the better currently available user models achieve Precision and Recall scores of around 35%, while the Bigram baseline achieves a score of 20%. It is of course not possible to specify what levels of Precision and Recall need to be reached in order to claim that a simulated user is realistic, but the metrics provide an intuitive feel for the accuracy of the predicted responses and they can be easily used to rank competing user models. A valid criticism of Precision and Recall, however, is that they place a high penalty on unseen user responses, even if the generated responses are in fact acceptable and believable.

5.1.2 Statistical metrics for comparing corpora

Precision and Recall deliver a rough indication of how realistic the *best* response is that the simulated user can generate, i.e. the response with the highest likelihood. On its own however, this form of evaluation is not sufficient because it does not evaluate whether the simulated user covers a variety of user behaviour. A dialogue strategy must perform well for all kinds of possible user responses, not just the one with the highest probability. It is thus necessary to produce a large number of dialogues through interaction between the dialogue manager and the simulated user in order to assess if the synthetic dataset has the same statistical properties as the training data set. It is of course important to use a dialogue manager with the same strategy as the one used to record the real dialogues. The process is illustrated in Figure 10.

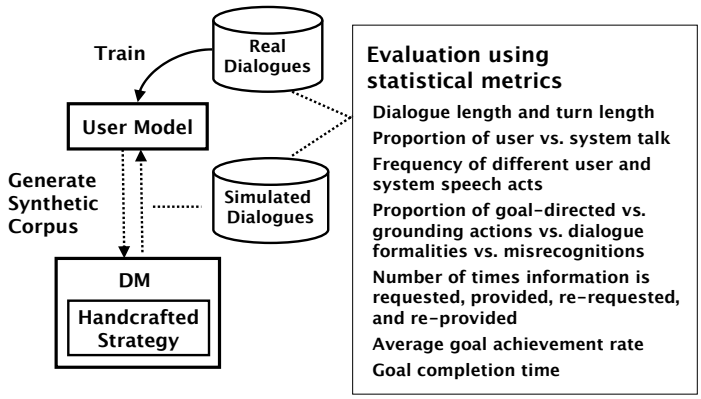

Figure 10 To assess how well the user model reproduces the variety in human user behaviour, a synthetic corpus is generated through interaction between the simulated user and the dialogue manager. Metrics such as the average length of a dialogue can then be used to evaluate how well the simulated dialogues match the statistical distribution of the real dialogue corpus.

A number of different statistical metrics have been proposed for this form of evaluation (Scheffler and Young, 2001; Schatzmann et al., 2005a). These may be divided into three groups:

1. *High-level features of the dialogue*: Average length of the dialogue (in number of dialogue turns), average number of actions per dialogue turn, proportion of user vs. system talk (ratio of user vs. system actions), etc.
2. *Style of the dialogue*: Frequency of different speech acts, ratio of goal-directed actions vs. grounding actions vs. dialogue formalities vs. misunderstandings, user cooperativeness (proportion of slot values provided when requested), etc.
3. *Success rate and efficiency of the dialogues*: Goal achievement rates, goal completion times, etc.

Again, it must be noted that any set of statistical metrics can only be a rough indicator of how good a simulation technique is. It is not possible to specify what range of values a synthetic corpus needs to satisfy in order to be sufficiently realistic. Moreover, no guarantee can be given that a simulated dialogue is realistic even if all of its properties are identical to the training data. Yet, an extensive set of evaluation measures forms a helpful toolkit for comparing simulation techniques and identifying possible weaknesses. The metrics described above, for instance, cover important features such as dialogue length, style and efficiency. By using a sufficiently large variety of measures we can ensure that a user model cannot be easily trained so as to achieve perfect scores on each of them.

5.1.3 Perplexity

Perplexity is a commonly used evaluation metric in the field of statistical language modelling for testing how well a given model predicts the sequence of words in a given test dataset (Young et al., 2002). It has been suggested as a means of user model evaluation by (Georgila et al., 2005a) and it is a useful

metric for determining whether the simulated dialogues contain similar action sequences (or dialogue state sequences) as real human-computer dialogue data. The definition of Perplexity (PP) is based on the per-action (per state) entropy H representing the amount of non-redundant information provided by each new action (state) on average.

$$PP = 2^{\hat{H}} \tag{22}$$

The latter can be approximated (Young et al., 2002) as

$$\hat{H} = -\frac{1}{m} \log_2 P(a_1, a_2, \ldots, a_m) \tag{23}$$

where $P(a_1, a_2, \ldots, a_m)$ is the probability estimate assigned to the action sequence $a_1, a_2, \ldots, a_m$ by a user model.

However, Perplexity is not necessarily a good indicator for how likely the user model is to predict a realistic response in the context of an unseen dialogue situation. In simulation-based learning, our goal is to apply new strategies to the user model, and no guarantee can be given that a user model with a low perplexity would indeed produce reasonable user responses in such a situation.

5.1.4 HMM Similarity

Another interesting method for measuring the similarity between real and simulated dialogues has recently been presented by Cuayahuitl et al. (Cuayahuitl et al., 2005). As described in Section 4.2.5 of this paper, Cuayahuitl et al. use Hidden Markov Models trained on human-computer dialogues to generate simulated dialogues. To evaluate the realism of a generated corpus with respect to a real corpus, the authors propose to train HMMs on both corpora and to compute the similarity between the two corpora in terms of the distance between the two HMMs. The assumption is of course that the smaller the distance between the two HMMs, the greater the similarity between the two corpora, and hence the greater the realism of the simulated dialogues. To compute the distance $D(\omega; \lambda_r, \omega; \lambda_s)$ between the HMMs λ_r (trained on the real dialogues) and λ_s (trained on the simulated dialogues), Cuayahuitl et al. use the symmetrised Kullback-Leibler divergence which is defined as

$$D(P, Q) = \frac{D_{KL}(P \parallel Q) + D_{KL}(Q \parallel P)}{2} \tag{24}$$

where D_{KL} is the distance between the probability distributions P and Q.

The variable ω can be either the state q, the system action a_s or the user action a_u (using the notation from Section 4.2.5). Cuayahuitl et al. measure the distance using each of the above and average the result over all three variables.

5.2 Evaluation of Learned Strategies

Indirect methods of evaluating statistical user models assess the *Utility* of a user model in terms of its effect on the performance of the system as a whole. In the context of strategy learning this involves measuring the performance of the learned strategies. A strategy on its own, however, is difficult to evaluate - it is preferable to evaluate the quality of dialogues carried out with the given strategy[6]. Ideally, it would be preferable to do this by recording dialogues between real users and a system using the learned strategy. Unfortunately, user studies are usually time-consuming and expensive and they require a working system which is often not available during the development phase of a project.

The standard evaluation practice used by research groups in strategy learning is thus to test the learned strategy by running it against a user model and measuring the "quality" or "success" of the generated dialogues. Some metric such as the number of filled and grounded slots is typically used, with penalties for excessively long dialogues. The score obtained with the learned strategy is then compared to the score

[6]It is possible to compare learned strategies based on the Q-value of the initial state, but this does not allow for comparisons with dialogue data recorded using handcrafted strategies, because the Q-value of the handcrafted strategy is not known.

that a handcrafted strategy achieves when tested against the user model (Levin et al., 2000; Pietquin, 2004; Henderson et al., 2005). Interestingly, the current standard practice in the field is to test the learned strategy on the same user model that was used during the learning process. The user model used for testing is typically trained on a held-out portion of the dialogue corpus which was not used for training the user model used for learning the strategy. Nevertheless this appears to be a somewhat questionable method of evaluation since it cannot detect when a strategy is merely fitted to a particular type of user model.

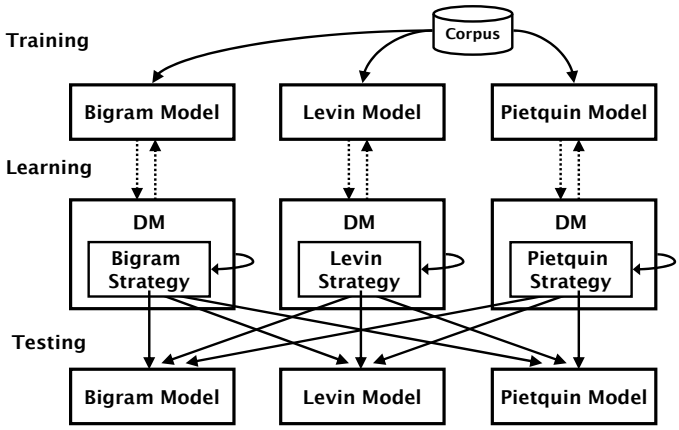

Figure 11 In the cross-model evaluation described in (Schatzmann et al., 2005b), competing user models are trained on an identical corpus of human-computer dialogues. Strategies are then learned with each user model using identical learning setups and the results are tested by running the learned strategies against each of the user models.

In recent work by (Schatzmann et al., 2005b), this practice has been re-assessed by performing comparative studies between different modelling techniques and by testing learned strategies on user models not used during learning. As shown in Figure 11, three prominent user models are trained on the same dialogue corpus, and strategies are then learned with all three user models using the same learning setup. The learned strategies are then tested on the user models used during learning to reproduce the results of previous evaluation studies but also on other user models to obtain a cross-comparison.

The results of the "traditional" form of evaluation appear very promising. All of the learned strategies appear to outperform handcrafted strategies, even those learned with relatively unsophisticated user models. When performing the cross-model comparison, however, the results (see Figure 12) show that strategies learned with a poor user model may appear to perform well when tested on the same user model, but fail when tested on good user models.[7] One may conclude that the quality of a learned strategy heavily depends on the quality of the user model and that the current standard evaluation practice of learning and testing strategies with the same user model can produce highly misleading results.

Moreover, even the good results obtained with strategies learned with user models of better quality must be challenged. Further analysis of the learned strategies in (Schatzmann et al., 2005b) indicates that strategies often perform well because they exploit deficiencies in the user model. Indeed, all of the learned strategies attempt to fill all of the unfilled slots in a single turn because none of the user simulations include a good model of user cooperativeness. The models currently do not acknowledge that users are less likely to respond positively to a high number of simultaneous requests. They also do not model well the fact that the number of misunderstandings is likely to increase with the number of simultaneous requests. This can lead to rather unnatural dialogues, as illustrated by the snippet below:

```
System: Where would you like to fly from and where would you like to
fly to? On what date would you like to fly and at what time? What is your
```

[7] See (Schatzmann et al., 2005b) for further details on the performance metric used to evaluate the performance of the strategies and note that the results shown here are computed on a more recent extended dataset.

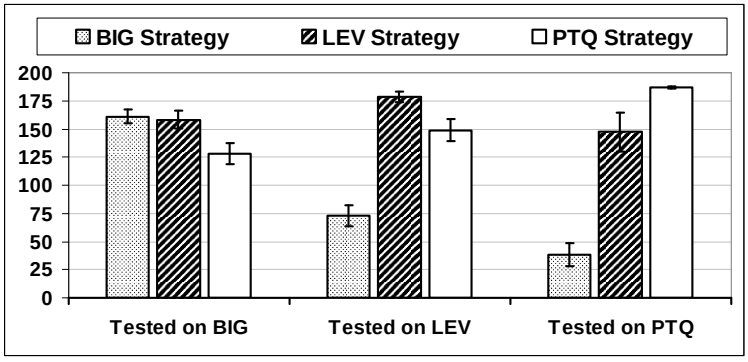

Figure 12 A cross-model evaluation of user models shows that a strategy learned with a poor user model may appear to perform well when tested on the user model used during learning but fails when tested on a different user model. The height of the shaded bars shows the performance of the learned strategies and the thin lines indicate 95% confi dence intervals.

```
preferred airline and would you like to sit at a window or an aisle seat?

  User model: I would like to fly from Boston to London. I would like to fly
on November 14 at 2pm. My preferred airline is BA and I would like to sit at
a window seat.
```

When tested on human users, the strategies are of course likely to perform much worse than the evaluation results indicate. Significant improvements are certainly needed in modelling user cooperativeness, recognition errors and dialogue misunderstandings but also in modelling user goals and user expertise as explained in (Schatzmann et al., 2005a,b).

6 Summary

This paper has provided a short summary of the role of the dialogue manager in a spoken dialogue system (SDS) and explained how machine-learning techniques can be used to automatically learn dialogue management strategies from dialogue data with the help of a simulated user. We have surveyed the recent literature on predictive statistical user modelling in SDS, explained the motivation for work in this area, and described the modelling techniques applied. The paper has also briefly discussed some of the current research issues in simulation-based learning from a user modelling perspective and provided an overview of the methods used and results obtained in recent evaluation studies.

Our aim has been to show that much could be gained by closer collaboration between researchers in the two fields of Dialogue Strategy Learning and Statistical User Modelling. To date, the problem of statistical user modelling for simulating user behaviour has been approached almost entirely by research groups with a background in SDS design and references to related work in the User Modelling domain are extremely rare. Vice versa, the User Modelling community has taken little notice of the work done in Spoken Dialogue Systems although considerable overlap between the two areas of research clearly exists.

We have argued that an understanding of User Modelling concepts such as concept-drift will be needed for future work on user simulation for strategy learning, for instance when modelling changes in the user goal during a course of interaction with the system. We also argue that predicting user dialogue behaviour is a prime example of an application that could benefit from combining collaboration-based and content-based modelling approaches - a research problem that is also of current interest in other areas of Statistical User Modelling.

Simulation-based Learning of Dialogue Strategies is still a nascent field of research and as our summary of recent evaluation studies has shown many problems still persist despite good progress in recent years. The lack of solid statistical user models for simulating dialogue behaviour currently remains a major roadblock for future progress on automatic design of dialogue management components, but the evident

benefits of learning optimal strategies from human-computer dialogue data warrant further research in this area.

Acknowledgements

The research reported in this paper was supported by the Gates Cambridge Trust and the EU FP6 TALK Project. We would like to thank J. D. Williams, H. Ye, O. Lemon, K. Georgila and J. Henderson for many helpful discussions and suggestions.

References

J. Alspector, A. Koicz, and N. Karunanithi. Feature based and clique based user models for movie selection: A comparative study. *User Modelling and User Adapted Interaction*, 7(4):279–304, 1997.

M. Araki, T. Watanabe, and S. Doshita. Evaluating dialogue strategies for recovering from misunderstandings. In *In Proc. IJCAI Workshop on Collaboration Cooperation and Conflict in Dialogue Systems*, pages 13–18, 1997.

H. Aust, M. Oerder, F. Seide, and V. Steinbiss. The philips automatic train timetable information system. *Speech Communication*, 17:249–262, 1995.

N. O. Bernsen, H. Dybkjar, and L. Dybkjar. Principles for the design of cooperative spoken human-machine dialogue. In *Proc. of ICSLP*, pages 729–732, 1996.

N. O. Bernsen, H. Dybkjar, and L. Dybkjar. *Designing Interactive Speech Systems: From First Ideas to User Testing*. Springer-Verlag, Germany, 1998.

A. Berthold and A. Jameson. Interpreting symptoms of cognitive load in speech input. In *In Proceedings of the Seventh International Conference on User Modeling*, pages 235–244, 1999.

A. Bestavros. Speculative data dissemination and service to reduce server load, network traffic and service time for distributed information systems. In *Proceedings of the 1996 International Conference on Data Engineering (ICDE-96)*, 1996. New Orleans, Louisiana.

W. Biermann and P. M. Long. The composition of messages in speech-graphics interactive systems. In *Proc. of the International Symposium on Spoken Dialogue*, pages 97–100, 1996.

D. Billsus and M. Pazzani. A hybrid user model for news story classification. In *User Modeling, Proceedings of the Seventh International Conference*, pages 99–108, 1995. Banff, Canada.

K. Binsted, A. Cawsey, and R. Jones. Generating personalised patient information using the medical record. In *Proceedings of the Fifth Conference on AI in Medicine*, pages 29–41, 1995.

D. Bohus. Error awareness and recovery in task-oriented spoken dialogue systems. Technical report, Ph.D. Thesis Proposal CMU, 2004.

D. Bohus and A. Rudnicky. Sorry, i didn't catch that! - an investigation of non-understanding errors and recovery strategies. In *Proceedings of SIGdial*, 2005a. Lisbon, Portugal.

D. Bohus and A. Rudnicky. Constructing accurate beliefs in spoken dialog systems. In *Proceedings of ASRU*, 2005b. San Juan, Perto Rico.

K. Bontcheva. Tailoring the content of dynamically generated explanations. In *M. Bauer, P.J. Gmytrasiewicz, J. Vassileva (eds). User Modelling 2001: 8th International Conference, Lecture Notes in Artificial Intelligence 2109,*, 2005. Springer Verlag.

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*, 2003. Sapporo, Japan.

P. Bretier and M. D. Sadek. A rational agent as the kernel of a cooperative spoken dialogue system: Implementing a logical theory of interaction. In *Proceedings of ATAL*, pages 189–203, 1996. Sapporo, Japan.

H. C. Bunt. Rules for the interpretation, evaluation and generation of dialogue acts. In *In IPO annual progress report 16*, pages 99–107, 1981. Technische Universiteit Eindhoven.

R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User Adapted Interaction*, 12(4):331–370, 2002.

S. Carberry. Techniques for plan recognition. *Journal of User-Modeling and User-Adapted Interaction*, 11(1-2):31–48, 2001.

S. Carberry and L. Lambert. A process model for recognizing communicatove acts and modelling negotiation subdialogues. *Computational Linguistics*, 25(1):1–53, 1999.

P. Chiu and G. Webb. Using decision trees for agent modeling: improving prediction performance. *User Modelling and User Adapted Interaction*, 8:131–152, 1998.

H. Cuayahuitl, S. Renals, O. Lemon, and H. Shimodaira. Human-computer dialogue simulation using hidden markov models. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2005.

M. Denecke, K. Dohsaka, and M. Nakano. Learning dialogue policies using state aggregation in reinforcement learning. In *In Proceedings of INTERSPEECH 2004*, pages 325–328, 2004.

L. Devillers and H. Bonneau-Maynard. Evaluation of dialog strategies for a tourist information retrieval system. In *Proc. of ICSLP 98*, pages 1187–1190, 1998. Sydney, Australia.

F. J. Dols and K. van der Sloot. Modelling mutual effects in belief-based interactive systems. In *In Proceedings of the Third International Workshop on User Modeling*, pages 3–19, 1992. Dagstuhl, Germany.

W. Eckert, E. Levin, and R. Pieraccini. User modelling for spoken dialogue system evaluation. In *Proc. of ASRU*, pages 80–87, 1997.

W. Eckert, E. Levin, and R. Pieraccini. Automatic evaluation of spoken dialogue systems. Technical Report TR98.9.1, ATT Labs Research, 1998.

S. Elzer, J. Chu-Carroll, and S. Carberry. Recognizing and utilizing user preferences in collaborative consultation dialogues. In *In Proceedings of the Fourth International Conference on User Modeling*, pages 19–24, 1994. Hyannis, MA, USA.

E. Filisko and S. Seneff. Developing city name acquisition strategies in spoken dialogue systems via user simulation. *Proc. of 6th SIGDial Workshop on Discourse and Dialogue*, 2005. Lisbon, Portugal.

M. Fleming and R. Cohen. A user modeling approach to determining system initiative in mixed-initiative artificial intelligence systems. In *In Proceedings of the 8th International Conference on User Modeling (UM2001)*, 2001.

K. Georgila, J. Henderson, and O. Lemon. Learning user simulations for information state update dialogue systems. *Proc. of 9th European Conference on Speech Communication and Technology (Eurospeech)*, 2005a. Lisbon, Portugal.

K. Georgila, O. Lemon, and J. Henderson. Automatic annotation of COMMUNICATOR dialogue data for learning dialogue strategies and user simulations. In *Proc. of DIALOR*, 2005b.

M. T. Gervasio, W. Iba, and P. Langley. Learning to predict user operations for adaptive scheduling. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 721–726, 1998. Madison, WI.

Z. Ghahramani. Unsupervised learning. In *Bousquet, O., Raetsch, G., and von Luxburg, U. (eds.), Advanced Lectures on Machine Learning*, 2004. Springer Verlag.

J. Glass. Challenges for spoken dialogue systems. In *Proceedings of the 1999 IEEE ASRU Workshop*, 1999.

D. Goddeau and J. Pineau. Fast reinforcement learning of dialog strategies. In *IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2000. Istanbul, Turkey.

A. Gorin, G. Riccardi, and J. Wright. How may i help you? *Speech Communication*, 23:113–127, 1997.

A. L. Gorin, B. A. Parker, R. M. Sachs, and J. G. Wilpon. How may i help you. In *In Proceedingsof International Symposium on Spoken Dialogue*, page 5760, 1996.

C. I. Guinn. An analysis of initiative selection in collaborative task-oriented discourse. *User Modeling and User-adapted Interaction*, 8(3-4):255–314, 1998.

P. J. Gurston, M. Farrell, D. Attwater, J. Allen, H. Kuo, M. Afify, E. Fosler-Lussier, and C. Lee. Oasis natural language call steering trial. In *Proceedings of Eurospeech*, 2001.

T. Harvey, S. Carberry, and K. Decker. Tailored responses for decision support. In *Proceedings of the 10th International Conference on User Modeling (UM-05)*, 2005.

J. Henderson, O. Lemon, and K. Georgila. Hybrid reinforcement/supervised learning for dialogue policies from communicator data. In *Proc. of IJCAI Workshop on KRPDS*, 2005. Edinburgh, Scotland.

E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 256–265, 1998.

U. Hustadt. A multi-model logic for stereotyping. In *In Proceedings of the Fourth International Conference on User Modeling*, pages 87–92, 1994. Hyannis, MA, USA.

M. Ishizaki, M. Crocker, and C. Mellish. Exploring mixed-initiative dialogue using computer dialogue simulation. *Journal of User-Modeling and User-Adapted Interaction*, 9(1-2):79–91, 1999.

F. Jelinek. Continuous speech recognition by statistical methods. *Proc. IEEE*, 64:532–556, 1976.

K. S. Jones and J. Galliers. *Evaluating Natural Language Processing Systems: An Analysis and Review*. Springer Verlag, 1996.

D. Jurafsky, C. Wooters, G. Tajchman, J. Segal, A. Stolcke, E. Fosler, and N. Morgan. The berkeley restaurant project. In *Proc. of ICSLP*, pages 2139–2142, 1994.

L. P. Kaelbling, M. L. Littman, and A. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

C. Kamm. *User Interfaces for Voice Applications In D. Roe and J. Wilpon (Eds), Voice Communication between Humans and Machines*. National Academy Press, 1995.

R. Keeney and H. Raiffa. *Decisions with Multiple Objectives*. John Wiley and Sons, 1976.

R. Klinkenberg and I. Renz. Adaptive information filtering: learning in the presence of concept drift. In *AAAI/ICML-98 Workshop on Learning for Text Categorization, Technical Report WS-98-05*, 1998. Madison, Wisconsin.

K. Komatani, S. Ueno, T. Kawahara, and H. G. Okuno. Flexible guidance generation using user model in spoken dialogue systems. In *Proceedings of the 41st Annual Meeting of the ACL*, 2003. Sapporo, Japan.

E. Levin and R. Pieraccini. A stochastic model of computer-human interaction for learning dialogue strategies. In *Proc. of Eurospeech 97*, pages 1883–1886, 1997. Rhodes, Greece.

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

H. Lieberman. Letizia: An agent that assists web browsing. In *IJCAI 95, Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 924–929, 1995. Montreal, Canada.

B.-S. Lin and L.-S. Lee. Computer-aided analysis and design for spoken dialogue systems based on quantitative simulations. *IEEE Transactions on Speech and Audio Processing*, 9(5):534–548, 2001.

D. J. Litman and S. Pan. Empirically evaluating an adaptable spoken dialogue system. In *In Pro-ceedings of the 7th International Conference on User Modeling*, 1999.

D. J. Litman, M. S. Kearns, S. Singh, and M. A. Walker. Automatic optimization of dialogue management. In *In Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, 2000. Saarbrucken, Germany.

R. Lopez-Cozar, A. de la Torre, J. Segura, and A. Rubio. Assessment of dialogue systems by means of a new simulation technique. *Speech Communication*, 40:387–407, 2003.

R. C. Moore. Reasoning about knowledge and action. In *IJCAI-77*, pages 223–227, 1977.

Y. Niimi and T. Nishimoto. Mathematical analysis of dialogue control strategies. In *Proc. Eurospeech 99*, pages 1535–1538, 1999.

R. Pieraccini, K. Dayandhi, J. Bloom, J. Dahan, M. Phillips, B. R. Goodman, and K. V. Prasad. Multimodal conversational systems for automobiles. *Communications of the ACM*, 47(1):47–49, 2004.

O. Pietquin. *A Framework for Unsupervised Learning of Dialogue Strategies*. PhD thesis, Faculte Polytechnique de Mons, 2004.

O. Pietquin. A probabilistic description of man-machine spoken communication. In *Proceedings of the 5th IEEE International Conference on Multimedia and Expo (ICME 2005)*, July 2005. Amsterdam, The Netherlands.

O. Pietquin and R. Beaufort. Comparing asr modeling methods for spoken dialogue simulation and optimal strategy learning. In *Proceedings of the 9th European Conference on Speech Communication and Technology (Interspeech/Eurospeech 2005)*, 2005. Lisbon, Portugal.

O. Pietquin and T. Dutoit. A probabilistic framework for dialog simulation and optimal strategy learning. *IEEE Transactions on Speech and Audio Processing, Special Issue on Data Mining of Speech, Audio and Dialog (to appear)*, 2005.

J. Polifroni, L. Hirschmann, and S. Seneff. Experiments in evaluating interactive spoken language systems. In *Proc. of the DARPA Speech and Natural Language Workshop*, pages 28–33, 1992. Harriman, NY.

J. Potjer, A. Russel, and L. Boves. Subjective and objective evaluation of two types of dialogues in a call assistance service. In *In IEEE Third Workshop on Interactive Voice Technology for Telecommunications Applications, IVTTA*, pages 89–92, 1996.

L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, NJ, 1993.

B. Raskutti, A. Neitz, and B. Ward. A feature-based approach to recommending selections based on past preferences. *User Modelling and User Adapted Interaction*, 7(3):179–218, 1997.

P. Resnick and H. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

S. Rosset, S. Bennacef, and L. Lamel. Design strategies for spoken language dialogue systems. In *Proceedings Eurospeech 99*, pages 1535–1538, 1999.

A. Rudnicky, E. Thayer, and P. Constantinides. Creating natural dialogs in the carnegie mellon communicator system. In *Proc. Eurospeech*, pages 1531–1534, 1999.

S. Russell and P. Norvig. *Artificial Intelligence: A modern approach*. Prentice Hall, 1995.

J. Schatzmann, K. Georgila, and S. Young. Quantitative evaluation of user simulation techniques for spoken dialogue systems. *Proc. of 6th SIGDial Workshop on Discourse and Dialogue*, 2005a. Lisbon, Portugal.

J. Schatzmann, M. Stuttle, K. Weilhammer, and S. Young. Effects of the user model on simulation-based learning of dialogue strategies. *Proc. of ASRU 2005*, 2005b. San Juan, Puerto Rico.

K. Scheffler. *Automatic design of spoken dialogue systems*. PhD thesis, Cambridge University, 2002.

K. Scheffler and S. J. Young. Probabilistic simulation of human-machine dialogues. In *Proc. ICASSP*, pages 1217–1220, 2000. Istanbul.

K. Scheffler and S. J. Young. Corpus-based dialogue simulation for automatic strategy learning and evaluation. In *Proc. NAACL Workshop on Adaptation in Dialogue Systems*, pages 64–70, 2001.

K. Scheffler and S. J. Young. Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning. In *Proc. Human Language Technology*, pages 12–18, 2002. San Diego.

K. Scheffler and S. J. Young. Simulation of human-machine dialogues. Technical Report CUED/F-INFENG/TR 355, Cambridge University Engineering Dept., 1999.

J. R. Searle. *Speech acts. An essay on the philosophy of language*. Cambridge University Press, 1969.

S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, and V. Zue. Galaxy ii: A reference architecture for conversational system development. In *Proc. of ICSLP*, pages 931–934, 1998.

J. Shin, S. Narayanan, L. Gerber, A. Kazemzadeh, and D. Byrd. Analysis of user behaviour under error conditions in spoken dialogue. In *Proceedings of ICSLP*, 2002.

A. Simpson and N. A. Fraser. Black box and glass box evaluation of the sundial system. In *Proceedings of the Third European Conference on Speech Communication and Technology*, pages 1423–1426, 1993.

S. Singh, M. S. Kearns, D. J. Litman, and M. A. Walker. Reinforcement learning for spoken dialogue systems. In *Proc. of NIPS*, 1999. Denver, Colorado.

S. Singh, M. S. Kearns, D. J. Litman, and M. A. Walker. Empirical evaluation of a reinforcement learning spoken dialogue system. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 645–651, 2000. Austin, TX.

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

R. W. Smith. An evaluation of strategies for selectively verifying utterance meanings in spoken natural language dialog. *International Journal of Human-Computer Studies*, 48:627– 647, 1998.

R. W. Smith and D. R. Hipp. *Spoken Natural Language Dialog Systems: A practical approach*. Oxford University Press, 1994.

O. Stock. *Alfresco: Enjoying the combination of Natural Language Processing and Hypermedia for Information Exploration In M. T. Maybury (Ed.) Intelligent Multimedia Interfaces*. AAAI Press / MIT Press, 1993.

R. S. Sutton and A. G. Barto. *Reinforcement Learning*. MIT Press, 1998.

D. Traum. Speech acts for dialogue agents. In *M. Wooldridge and A. Rao (eds.), Foundations of Rational Agency, Dordrecht, Kluwer*, pages 169–201, 1999.

C. J. van Rijsbergen. *Information Retrieval, Second Edition*. Butterworths, London, 1979.

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

M. Walker. *Informational Redundancy and Resource Bounds in Dialogue*. PhD thesis, University of Pennsylvania, 1993.

M. Walker, D. Hindle, and J. Fromer. Evaluating competing agent strategies for a voice email agent. In *Proc. of the 5th European European Conference on Speech Communication and Technology*, 1997. Rhodes, Greece.

M. Walker, J. C. Fromer, and S. Narayanan. Learning optimal dialogue strategies: A case study of a spoke dialogue agent for email. In *Proc. of the 36th Annual Meeting of the Association of Computational Linguists*, pages 1345–1352, 1998.

J. W. Wallis and E. H. Shortliffe. *Customized Explanations using Causal Knowledge. In B. C. Buchanan and E. H. Shoortliffe (Eds) Rule-based expert systems: The MYCIN Experiments of the Stanford heuristic Programming Project*. Addison Wesley, 1985.

T. Watanabe, M. Araki, and S. Doshita. Evaluating dialogue strategies under communication errors using computer-to-computer simulation. *Trans. of IEICE, Info Syst.*, E81-D(9):1025–1033, 1998.

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

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

G. Webb, M. Pazzani, and D. Billsus. Machine learning for user modeling. *User Modeling and User-Adapted Interaction*, 11:19–29, 2001.

G. I. Webb and M. Kuzmycz. Feature based modelling: A methodology for producing coherent, consistent, dynamically changing model of agent's competencies. *User Modelling and User Adapted Interaction*, 5(2):117–150, 1996.

B. Webber and A. Joshi. Taking the initiative in natural language database interaction: Justifying why. In *Proc. of COLING 82*, pages 413–419, 1982.

G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23:69–101, 1996.

J. Williams, P. Poupart, and S. Young. Partially observable markov decision processes with continuous observations for dialogue management. *Proc. of 6th SIGDial Workshop on Discourse and Dialogue*, 2005a. Lisbon, Portugal.

J. D. Williams. A probabilistic model of human/computer dialogue with application to a partially observable markov decision process. In *Ph D first year report. Department of Engineering, University of Cambridge*, 2003.

J. D. Williams, P. Poupart, and S. Young. Factored partially observable markov decision processes for dialogue management. In *4th Workshop on Knowledge and Reasoning in Practical Dialog Systems, International Joint Conference on Artificial Intelligence (IJCAI)*, 2005b. Edinburgh, Scotland.

S. Young. Talking to machines (statistically speaking). In *Proc. of ICSLP*, 2002a. Denver, Colorado.

S. Young. The statistical approach to the design of spoken dialogue systems. Technical Report CUED/F-INFENG/TR.433, Cambridge University Engineering Department, 2002b.

S. J. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book (for HTK Version 3.2)*. University of Cambridge, 2002.

V. Zue. Conversational interfaces: Advances and challenges. In *Proc. of Eurospeech*, 1997.

I. Zukerman and D. Albrecht. Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction*, 11:5–18, 2001.

I. Zukerman and D. Litman. Natural language processing and user modeling: Synergies and limitations. *User Modeling and User-Adapted Interaction*, 11:129–158, 2001.