Available online at www.sciencedirect.com
ScienceDirect
Speech Communication 50 (2008) 716–729
SPEECH COMMUNICATION
www.elsevier.com/locate/specom

Automating spoken dialogue management design using machine learning: An industry perspective

Tim Paek [a,*], Roberto Pieraccini [b]

[a] Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
[b] SpeechCycle, 26 Broadway, 11th Floor, New York, NY 10004, USA

Received 27 June 2007; received in revised form 6 January 2008; accepted 14 March 2008

Abstract

In designing a spoken dialogue system, developers need to specify the actions a system should take in response to user speech input and the state of the environment based on observed or inferred events, states, and beliefs. This is the fundamental task of dialogue management. Researchers have recently pursued methods for automating the design of spoken dialogue management using machine learning techniques such as reinforcement learning. In this paper, we discuss how dialogue management is handled in industry and critically evaluate to what extent current state-of-the-art machine learning methods can be of practical benefit to application developers who are deploying commercial production systems. In examining the strengths and weaknesses of these methods, we highlight what academic researchers need to know about commercial deployment if they are to influence the way industry designs and practices dialogue management.

Keywords: Dialogue management; Machine learning; Reinforcement learning; Industry

1. Introduction

Automated systems that interact with users using speech recognition as the primary modality go by various names, such as spoken dialogue systems, interactive voice response systems (IVR), voice user interfaces (VUI), or simply, speech applications. Whatever the name, all of these systems rely on the fundamental task of *dialogue management*, which concerns what action or response a system should take in response to user input. The action taken may depend on a myriad of factors, from features of the speech recognizer (e.g., the confidence score of an utterance), to features of the dialogue interaction (e.g., the number of repairs taken so far), to features of the application domain (e.g., company guidelines for customer service), and to the response and status of external backends, devices, and data repositories. In many system architectures, a distinct *dialogue manager* exists to oversee and control the entire conversational interaction and execute any number of functions (e.g., Allen et al., 1998; Polifroni and Seneff, 2000). Some of these functions include updating new user input, resolving ambiguities, managing speech recognition grammars, communicating with external backends, and so forth. Ultimately, the dialogue manager prescribes the *next* action for each turn of an interaction. Because actions taken by the system directly impact users, the dialogue manager is largely responsible for how well the system performs and is perceived to perform by users—i.e. the user experience.

Given the importance of dialogue management, both as a research problem as well as a commercial enabler of advanced applications, researchers have recently been turning to machine learning methods to formalize and optimize the action selection process. In particular, one approach that has been gaining momentum is reinforcement learning (Sutton and Barto, 1998). In this approach, the dynamic interaction of a spoken dialogue is represented as a fully

* Corresponding author. Tel.: +1 425 703 8647; fax: +1 425 706 7329.
E-mail address: timpaek@microsoft.com (T. Paek).

0167-6393/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.specom.2008.03.010

or partially observable Markov decision process (MDP) and an optimal policy is derived which prescribes what actions the system should take in various states of the dialogue so as to maximize a reward function. The idea of having a system that can learn interactively from the rewards it receives is appealing given the alternative: crafting system responses to all possible user input, which, unfortunately, characterizes the status quo. In commercial settings, application developers, together with VUI designers, typically hand-craft dialogue management strategies using rules and heuristics. At best, these rules are based on the accumulated knowledge and trial-and-error experience of industry practitioners. At worst, they are based on intuition and limited experience. Either way, because it is extremely challenging to anticipate every possible user input, hand-crafting dialogue management strategies is an error-prone process that needs to be iteratively refined and tuned, which of course requires much time and effort. The reinforcement learning approach promises to give application developers a tool for automating the design of dialogue management strategies by having the system learn these strategies from feedback data. This casts dialogue management as an optimization problem, which, once solved, could remove the "art" from the process and facilitate rapid application development.

In this paper, we present an industry perspective on machine learning for spoken dialogue management in general, and on reinforcement learning in particular. We offer this perspective in light of the commercial success of speech applications. What began in academic institutions and industry laboratories more than 50 years ago has recently blossomed into a thriving business (Pieraccini and Lubensky, 2005). Hundreds of commercial systems are being deployed each year, adhering to industry-wide standards and protocols, such as VoiceXML, CCXML, MRCP, etc. Unfortunately, as researchers have started to point out, dialogue systems in industry have been evolving on a parallel path with those in academic research. Unless a "synergistic convergence" of architectures, abstractions and methods is reached from both communities, ideas and technologies in academic research run the risk of being overlooked by industry practitioners (Pieraccini and Huerta, 2005). This paper endeavors to bridge the gap between the two communities so that more research on spoken dialogue management can benefit commercial applications, which in turn would allow the research to impact thousands, if not millions, of customers on a daily basis.

This paper divides into three sections. In the first section, we describe how commercial applications are built and delineate the kinds of requirements, specification, and tuning that is typically used for dialogue management. In particular, we highlight the important role of VUI design and specification, drawing examples from real applications. In the second section, we investigate the pursuit of automated spoken dialogue management and attempt to distinguish between the hype and reality of automatic learning. In particular, we review current state-of-the-art reinforcement learning methods and consider to what extent dialogue management can be automated by these methods and whether these methods can be of practical benefit to application developers. Finally, in the last section, we draw upon the issues raised in the previous sections to discuss how researchers may be able to effectively influence the design and practice of dialogue management in industry, and in so doing, allow their research to touch the lives of multitudes of customers who interact with deployed commercial systems.

2. Commercial development and deployment

The development of a commercial spoken dialogue system (SDS) is a complex process. Dialogue management is only a part of it, though arguably the most expensive. Development starts with the collection of *requirements*, which describe what the system will and will not do. Before starting any development activity, requirements need to be properly and thoroughly defined to contain the scope of the final product. For example, for a banking application, the requirements will define which type of transactions the system is going to perform, such as account balances, fund transfer, and other inquiries. Moreover, customers often have strict business rules that need to define the way certain operations are performed. For instance, certain transactions may not be performed in the absence of proper user identification or validation, or the sequence of certain operations may be constrained. Only after the requirements have been defined and jointly agreed upon by the customer and the vendor who is going to develop the application, can a proper development phase begin.

Development processes are different from vendor to vendor, but they all include a functional specification and a detailed design of the interaction with the voice user interface (VUI). The functional *specification* is a high level definition of the different phases of the interaction, and is typically defined by a workflow where each block encapsulates atomic logical components defined in a hierarchical structure. For instance, in the simple banking application mentioned above, a functional module would be defined for account balance and one for fund transfer. Each high level module would be expanded into smaller components, such as requesting the source account, the destination account, and the amount of money to transfer.

The VUI design phase is mostly concerned with user experience. As such, it has to deal with the way the system speaks and interprets user speech input. Commercial systems are built by logically sequencing predetermined system utterances—called *prompts*—and the interpretation of the user responses based on context-specific grammars or statistical language models. The actual interaction logic is determined by a graph—called the *call-flow*—where the arcs are associated with conditional statements based on user speech input and other variables, and the nodes are associated with system actions.

Along with the VUI design, there has to be a parallel activity devoted to the development and tuning of the