Chapter 1
Spoken Language Understanding for Natural Interaction: The Siri Experience

Jerome R. Bellegarda

Abstract Recent advances in software integration and efforts toward more personalization and context awareness have brought closer the long-standing vision of the ubiquitous intelligent personal assistant. This has become particularly salient in the context of smartphones and electronic tablets, where natural language interaction has the potential to considerably enhance mobile experience. Far beyond merely offering more options in terms of user interface, this trend may well usher in a genuine paradigm shift in man-machine communication. This contribution reviews the two major semantic interpretation frameworks underpinning natural language interaction, along with their respective advantages and drawbacks. It then discusses the choices made in Siri, Apple's personal assistant on the iOS platform, and speculates on how the current implementation might evolve in the near future to best mitigate any downside.

1.1 Introduction

In recent years, smartphones and other mobile devices, such as electronic tablets and more generally a wide variety of handheld media appliances, have brought about an unprecedented level of ubiquity in computing and communications. At the same time, voice-driven human-computer interaction has benefited from steady improvements in the underlying speech technologies (largely from a greater quantity of labeled speech data leading to better models), as well as the relative decrease in the cost of computing power necessary to implement comparatively more sophisticated solutions. This has sparked interest in a more pervasive spoken language interface, in its most inclusive definition encompassing speech recognition, speech synthesis, natural language understanding, and dialog management.

J.R. Bellegarda (✉)
Apple Inc., One Infinite Loop, Cupertino, CA 95014, USA
e-mail: jerome@apple.com

J. Mariani et al. (eds.), *Natural Interaction with Robots, Knowbots and Smartphones: Putting Spoken Dialog Systems into Practice*, DOI 10.1007/978-1-4614-8280-2_1,

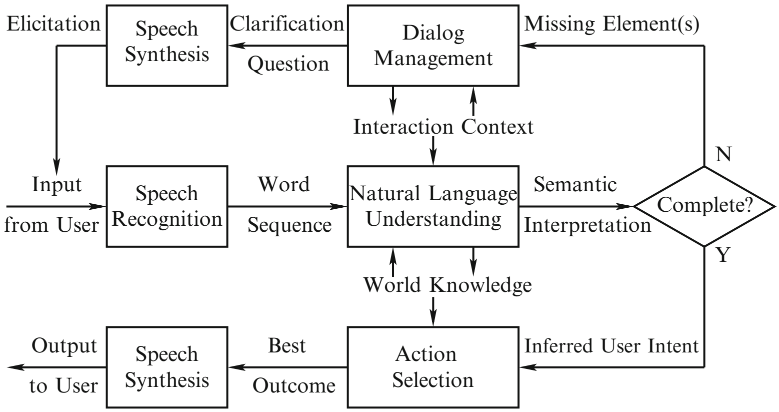


Fig. 1.1 Overview of "intelligent personal assistant" interaction model

To wit, multiple voice-driven initiatives have now reached commercial deployment, with products like Apple's Siri [1], Google's Voice Actions [8], Microsoft's Bing Voice Search [13], Nuance's Dragon Go! [15], and Vlingo [21]. The well-publicized release of Siri in Apple's iPhone 4S, in particular, may have heralded an irreversible shift toward the "intelligent personal assistant" paradigm: just say what you want, and the system will automatically figure out what the best course of action is. For example, to create a new entry on his/her calendar, the user may start the interaction with an input like:

Schedule a meeting with John Monday at 2pm (1.1)

The system then has to recognize that the user's intent is to create a new entry and deal with any ambiguities about the attributes of the entry, such as who will be invited (*John Smith* rather than *John Monday*) and when the meeting will take place (*this coming Monday* rather than *last Monday*).

An overview of the underlying interaction model is given in Fig. 1.1. The speech utterance is first transcribed into a word sequence on which to perform natural language understanding, leading to a semantic interpretation of the input. In case any element is missing, dialog management relies on interaction context to elicit the relevant information from the user. Once the semantic interpretation is complete, task knowledge guides the selection of the best action for the situation at hand. Finally, the selected outcome is conveyed to the user. Success in this realm is measured in subjective terms: *how well* does the system fulfill the needs of the user relative to his/her intent and expectations? Depending on the task, "well" may variously translate into "efficiently" (with minimal interruption),"thoroughly" (so the task is truly complete), and/or "pleasantly" (as might have occurred with a human assistant).

Of course, many of the core building blocks shown in Fig. 1.1 have already been deployed in one form or another, for example, in customer service applications with automatic call handling. Wildfire, a personal telephone assistant, has similarly been available since the mid-1990s [22]. Yet in most consumers' perception, at best the resulting interaction has not been significantly more satisfying than pressing touch-tone keys. So how to explain the growing acceptance of Siri and similar systems? While the interaction model of Fig. 1.1 has not suddenly become flawless, it has clearly matured enough to offer greater perceived flexibility. Perhaps a key element of this perception is that the new systems strive to provide a direct answer whenever possible, rather than possibly heterogeneous information that may contain the answer, as in the classical search paradigm.

Arguably, the most important ingredient of this new perspective is the accurate inference of user intent and correct resolution of any ambiguity in associated attributes. While speech input and output modules clearly influence the outcome by introducing uncertainty into the observed word sequence, the correct delineation of the task and thus its successful completion heavily hinges on the appropriate semantic interpretation of this sequence. This contribution accordingly focuses on the two major frameworks that have been proposed to perform this interpretation and reflects on how they each contribute to the personal assistant model.

The material is organized as follows. The next section describes the statistical framework characteristic of data-driven systems, while Sect. 1.3 does the same for the rule-based framework underpinning expert systems and similar ontology-based efforts. In Sect. 1.4, we focus on Siri as an example and discuss in particular how the choices adopted proved critical to a successful deployment. Finally, the article concludes with some prognostications regarding the next natural stage in the evolution of the user interface.

1.2 Statistical Framework

1.2.1 Background

Fundamentally, the statistical approach to semantic interpretation is aligned with the data-driven school of thought, which posits that empirical observation is the best way to capture regularities in a process (like natural language) for which no complete *a priori* model exists. This strand of work originated in speech recognition, where in the 1980s probabilistic models such as hidden Markov models were showing promise for reconstructing words from a noisy speech signal [16]. Applying similar probabilistic methods to natural language understanding involved the integration of data-driven evidence gathered on suitable training data in order to infer the user's intent.

The theoretical underpinnings for this kind of reasoning were first developed in the context of a partially observable Markov decision process (POMDP) [17]. The key features of the POMDP approach are (1) the maintenance of a system of beliefs, continually updated using Bayesian inference, and (2) the use of a policy whose performance can be quantified by a system of associated rewards and optimized using reinforcement learning via Bellman's optimality principle [10]. Note that Bayesian belief tracking and reward-based reinforcement learning are mechanisms that humans themselves appear to use for planning under uncertainty [6]. For example, experimental data shows that humans can implicitly assimilate Bayesian statistics and use Bayesian inference to solve sensorimotor problems [11].

This in turn motivated the application of the POMDP framework to spoken dialog systems, to similarly learn statistical distributions by observation and use Bayes' rule to infer posteriors from these distributions [24]. However, this proved challenging in practice for several reasons. First, the internal state is a complex combination of the user's goal, the user's input, and the dialog history, with significant uncertainty in the user's utterances (due to speech recognition errors) propagating uncertainty into the other entities as well. In addition, the system action space must cover every possible system response, so policies must map from complex and uncertain dialog states into a large space of possible actions.

1.2.2 Current State of the Art

Making the POMDP framework tractable for real-world tasks typically involves a number of approximations. First, state values can be ranked and pruned to eliminate those not worth maintaining. Second, joint distributions can be factored by invoking some independence assumptions that can be variously justified from domain knowledge. Third, the original state space can be mapped into a more compact summary space small enough to conduct effective policy optimization therein. Fourth, in a similar way, a compact action set can be defined in summary space and then mapped back into the original master space [23].

As an example, Fig. 1.2 shows a possible POMDP implementation for the meeting scheduling task associated with (1.1). It illustrates one time step of a (partial) dynamic Bayesian network, in which the (hidden) system state and (observed) event are represented by open and shaded circles, respectively, while the (observed) command executed by the system is denoted by a shaded rectangle. The state is decomposed into slots representing features such as *person* (indexed by p), *date* (indexed by d), *location*, and *topic* (not shown). Each slot comprises information related to user goal, user input, and dialog history so far. In this simple example, the only dependence modeled between slots is related to the person information. This configuration, known as a "Bayesian update of dialog state" (BUDS) system [20], retains the ability to properly represent system dynamics and to use fully parametric models, at the cost of ignoring much of the conditional dependency inherent in real-world domains.

Because the state of the system (encapsulating the intent of the user) is a hidden variable, its value can only be inferred from knowledge of the transition probabilities between two successive time instants and the observation probabilities associated with the observed event. This leads to a belief update equation of the form:

$$b_{t+1} = K \cdot O(o_{t+1}) \cdot T(c_t) \cdot b_t \,, \tag{1.2}$$

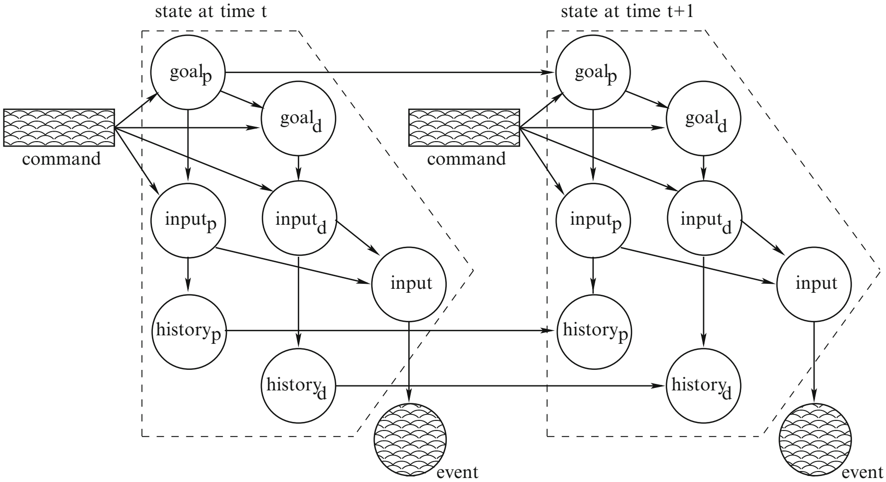


Fig. 1.2 (Partial) dynamic Bayesian network for meeting scheduling task

where the N-dimensional vector $b = [b(s_1) \dots b(s_N)]^T$ is the belief distribution over N possible system states s_i, $O(o)$ is a diagonal matrix of observation probabilities $P(o|s_i)$, and $T(c)$ is the $N \times N$ transition matrix for command c. Given some assumed initial value b_0, (1.2) allows the belief state to be updated as each user input is observed. Since the actual state is unknown, the action taken at each turn must be based on the belief state rather than the underlying hidden state.

This mapping from belief state to action is determined by a policy $\pi : b \longrightarrow c$. The quality of any particular policy is quantified by assigning rewards $r(s,c)$ to each possible state-command pair. The choice of specific rewards is a design decision typically dependent on the application. Different rewards will result in different policies and most likely divergent user experiences. However, once the rewards have been fixed, policy optimization is equivalent to maximizing the expected total reward over the course of the user interaction. Since the process is assumed to be Markovian, the total reward expected in traversing from any belief state b to the end of the interaction following policy π is independent of all preceding states. Using Bellman's optimality principle, it is possible to compute the optimal value of this value function iteratively. As mentioned earlier, this iterative optimization is an example of reinforcement learning [18].

1.2.3 Trade-Offs

From a theoretical perspective, the POMDP approach has many attractive properties: by integrating Bayesian belief monitoring and reward-based reinforcement learning, it provides a robust interpretation of imprecise and ambiguous human interactions, promotes the ability to plan interactions so as to maximize concrete objective functions, and offers a seamless way to encompass short-term adaptation and long-term learning from experience within a single statistical framework. Still, it is potentially fragile when it comes to assigning rewards, as encouraging (respectively discouraging) the correct (respectively wrong) state-command pair can be a delicate exercise in the face of a huge space of possible such pairs.

In addition, as is clear from (1.2), the computational complexity of a single inference operation is $\mathscr{O}(N^2)$, where N is the number of possible system states. Thus, for even moderately large values of N exact computation becomes intractable, which makes it challenging to apply to real-world problems. The necessary approximations all have drawbacks, be it in terms of search errors, spurious independence assumptions, quantization loss from master to summary space, or imperfect user simulation to generate reinforcement data [7].

1.3 Rule-Based Framework

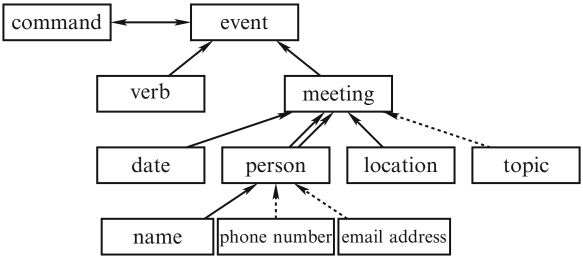
1.3.1 Background

In contrast with the systems just mentioned, the rule-based framework does not attempt to leverage data in a statistical way. At its core, it draws its inspiration from early expert systems such as MYCIN [4]. These systems, relying on an inference engine operating on a knowledge base of production rules, were firmly rooted in the artificial intelligence (AI) tradition [12]. Their original purpose was to create specialized agents aimed at assisting humans in specific domains (cf., e.g., [14]). Agent frameworks were later developed to create personal intelligent assistants for information retrieval. In this context, the open agent architecture (OAA) introduced the powerful concept of delegated computing [5]. This was later extended to multi-agent scenarios where distributed intelligent systems can model independent reactive behavior (cf., e.g., [19]).

In the early to mid-2000s, DARPA's PAL (perceptive assistant that learns) program attempted to channel the above efforts into a learning-based intelligent assistant comprising natural language user interaction components layered on top of core AI technologies such as reasoning, constraint solving, truth maintenance, reactive planning, and machine learning [3]. The outcome, dubbed CALO for the Cognitive Assistant that Learns and Organizes, met the requirements for which it was designed, but because of its heterogeneity and complexity, it proved difficult for nonexperts to leverage its architecture and capabilities across multiple domains. This sparked interest in a more streamlined design where user interaction, language processing, and core reasoning are more deeply integrated within a single unified framework [9].

An example of such framework is the "Active" platform, which eschews some of the sophisticated AI core processing in favor of a lighter-weight, developer-friendly version easier to implement and deploy [9]. An application based on this

Fig. 1.3 Active ontology for meeting scheduling task



framework consists of a set of loosely coupled services interfacing with specialized task representations crafted by a human expert. Using loosely coupled services eases integration of sensors (cf. speech recognition, but also vision systems, mobile or remote user interfaces, etc.), effectors (cf. speech synthesis, but also touch user interfaces, robotics, etc.), and processing services (such as remote data sources and other processing components).

1.3.2 Current State of the Art

In the "Active" framework, every task is associated with a specific "active ontology." Whereas a conventional ontology is a static data structure, defined as a formal representation for domain knowledge, with distinct classes, attributes, and relations among classes, an active ontology is a dynamic processing formalism where distinct processing elements are arranged according to ontology notions. An active ontology thus consists of a relational network of concepts, where concepts serve to define both data structures in the domain (e.g., a meeting has a date and time, a location, a topic, and a list of attendees) and associated rule sets that perform actions within and among concepts (e.g., the date concept derives a canonical date object of the form: `date(DAY, MONTH, YEAR, HOURS, MINUTES)` from a word sequence such as *Monday at 2pm*).

Rule sets are collections of rules where each rule consists of a condition and an action. As user input is processed, data and events are inserted into a fact store responsible for managing the life cycle of facts. Optional information can be specified to define when the fact should actually be asserted and when it should be removed. As soon as the contents of the fact store changes, an execution cycle is triggered and conditions evaluated. When a rule condition is validated, the associated action is executed. The active ontology can therefore be viewed as an execution environment.

To fix ideas, Fig. 1.3 shows the active ontology for the meeting scheduling task associated with (1.1). The active ontology consists of a treelike structure defining the structure of a valid command for this task. The command operates on a complete event concept representing the action of scheduling a meeting. The meeting concept itself has a set of attributes comprising one or more persons, a topic, a location and a date. Structural relationships are denoted by arrows, which relate to a “has a” ontological notion. For instance, topic, date, location, and person concepts are members of a meeting.

Structural relationships also carry cardinality information and record whether children nodes are optional, mandatory, unique, or multiple. For instance, the relationship between person and meeting is multiple and mandatory, which is denoted by a double solid arrow. On the other hand, the relationship between topic and meeting is unique and optional, which is denoted by a single dashed arrow. This structure is used to provide the user with contextual information. In the example of (1.1), as the location node is linked as mandatory, the user will be asked to provide a location. Through this mechanism, the active ontology not only generates a structured command but also builds dynamic information to interactively assist the user.

As alluded to earlier, concepts incorporate various instantiations of canonical objects. For example, *Monday at 2pm* and *tomorrow morning* are two instances of date objects in the date concept. These objects relate to a “is a” ontological notion. To the extent that rule sets can be specified to sense and rate incoming words about their possible relevance to various concepts, this makes the domain model portable across languages. In addition, it has the desirable side effect of making the approach insensitive to the order of component phrases.

1.3.3 Trade-Offs

Pervasive in the above discussion is the implicit assumption that language can be satisfactorily modeled as a finite state process. Strictly speaking, this can only be justified in limited circumstances, since, in general, the level of complexity of human languages goes far beyond that of context-free languages. Thus, rule-based systems may be intrinsically less expressive than data-driven systems.

In addition, an obvious bottleneck in their development is the specification of active ontologies relevant to the domain at hand. For the system to be successful, each ontology must be 100 % complete: if an attribute is overlooked or a relationship between classes is missing, some (possibly rare) user input will not be handled correctly. In practice, this requires the task domain to be sufficiently well-specified that a human expert from the relevant field is able to distill it into the rule base. This so-called knowledge engineering is typically hard to “get right” with tasks that are highly variable or subject to a lot of noise.

On the plus side, once the ontology correctly captures the whole domain structure, deployment across multiple languages is relatively straightforward. Since a near-exhaustive list of relevant word patterns is already included inside each concept and word order is otherwise largely ignored, only individual surface forms have to be translated. This makes this approach paradoxically similar in spirit to (data-driven) bag-of-words techniques such as latent semantic mapping [2].

1.4 The Siri Experience

1.4.1 Scope and Behavior

Siri was originally formed as a startup company to leverage the results of the CALO project within a much tighter effort with a commercial focus. Its architecture adopted the "Active" platform described earlier as the intermediate layer between mobile I/O and web services. Initial efforts centered on creating tools to make it easier to develop the necessary domain modules and associated active ontologies. The aim was to alleviate many of the issues linked to the ontology bottleneck. With such tools in place, along with the necessary software engineering work required to support them, it became less tedious for human experts to create active ontologies for a given task domain. Upon acquisition by Apple, the Siri team further concentrated on integration and scalability issues.

One important aspect of the Siri experience is the system's response to user requests that fall outside of the well-defined domains it knows about. A typical example is a query such as "Siri, do you love me?", for which a factual response is besides the point. When faced with such out-of-domain queries, most systems typically exhibit a rather "clinical" behavior, with responses like: "*Sorry, I don't understand what you mean.*" While technically adequate, such behavior lacks humanness. In contrast, Siri tries to provide somewhat more entertaining and/or whimsical responses. Not only does this policy inject a bit of sassiness into the system, it also makes the out-of-domain fall-back more palatable to the user.

To illustrate, Table 1.1 gives some examples of such requests under three different scenarios, along with some of the answers provided. Note that giving the same input three times in a row results in three different answers, as the same response would likely be annoying and otherwise destroy the illusion of anthropomorphism. Imbuing the assistant with such socially adept behavior substantially contributed to giving Siri its unique personality. We believe that this strategy proved critical to a successful deployment.

Table 1.1 Examples of out-of-domain requests

User:	Should I go to bed, Siri?
Siri:	*I think you should sleep on it*
User:	I love you!
Siri:	*That's nice. Can we get back to work now?*
User:	I love you!!
Siri:	*Oh, I bet you say that to all your Apple products*
User:	I love you!!!
Siri:	*All you need is love. And your iPhone*
User:	You are useless...
Siri:	*I don't really like these arbitrary categories*

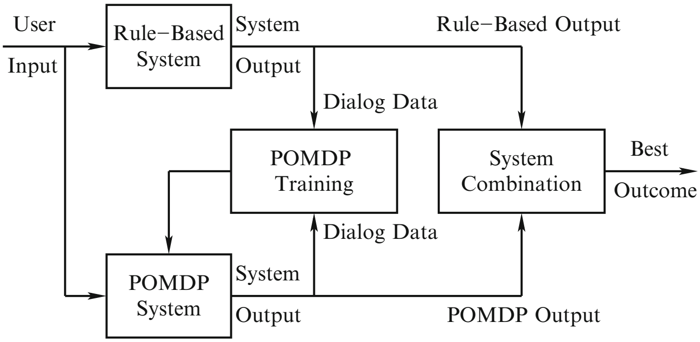


Fig. 1.4 Toward the convergence of rule-based and statistical frameworks

1.4.2 Likely Evolution

Like all rule-based systems, Siri suffers from some of the pitfalls mentioned in Sect. 3.3. It is legitimate to ask whether mitigating this downside might involve integrating the associated top-down outlook with the bottom-up outlook adopted by the statistical framework. This possibility unfolds naturally from the inherent complementarity in the respective advantages and drawbacks of the two approaches. Whereas ontology specification requires upfront labor-intensive human expertise, data-driven systems can be run in completely automated fashion. On the other hand, rule-based systems can be deployed right away, while the statistical framework calls for a large amount of suitable training data to be collected beforehand. On the flip side, the former is much more amenable to leveraging know-how across languages, thus enabling rapid deployment in multiple languages, while in the latter every language essentially involves the same amount of effort.

Complementarity between the frameworks, moreover, goes beyond a mere data-vs-knowledge distinction. Whereas rule-based systems are generally sensitive to noise, in principle the POMDP approach can cope with various sources of uncertainty. Yet its elegant optimization foundation assumes specification of suitable rewards, which are probably best informed by empirical observation, and thus rules derived therefrom. In addition, POMDP systems typically involve deleterious approximations to reduce the computational complexity inherent to the sophisticated mathematical machinery involved. In contrast, the AI framework may be intrinsically less expressive but tends to exhibit a more predictable behavior.

Such complementarity bodes well for an eventual convergence between the two approaches, perhaps by way of the virtuous cycle illustrated in Fig. 1.4. First, the deployment of a rule-based system such as Siri provides some real-world dialog data that can be used advantageously for POMDP training, without the difficulties inherent to data collection via user simulation. This in turn enables the deployment of a statistical system like BUDS, which further provides real-world data to refine POMDP models. Such large-scale data collection potentially removes one of the big limiting factors in properly handling uncertainty. It thus becomes possible to combine the rule-based and statistical outputs to come up with the best outcome, based on respective confidence measures for both systems (which may vary over time). By enabling more robust reasoning and adaptation, this strategy should considerably strengthen the cognitive aspects of natural language understanding.

1.5 Conclusion

In this contribution, we have examined the emerging deployment of the "intelligent personal assistant" style of interaction. Under this model it is critical to accurately infer user intent, which in turn hinges on the appropriate semantic interpretation of the words uttered. We have reviewed the two major frameworks within which to perform this interpretation, along with their most salient advantages and drawbacks. Ontology-based systems, such as Siri, are better suited for initial deployment in well-defined domains across multiple languages, but must be carefully tuned for optimal performance. Data-driven systems based on POMDP have the potential to be more robust, as long as they are trained on enough quality data.

The inherent complementarity between these two frameworks sets the stage for the two to converge toward a more cognitive mainstream user interface, which will take intelligent delegation to the next level across many more usage scenarios. Under that hypothesis, the personal assistant model ushers in the next natural stage in the evolution of the user interface: as depicted in Fig. 1.5, the desktop, browser, and search metaphors of past decades thus lead to a new solve metaphor focused on context and tasks. The underlying assumption is that the user will increasingly get used to expressing a general need and letting the system fulfill it in a stochastically consistent manner. This development will likely be a key stepping stone toward an ever more tangible vision of ubiquitous intelligence.

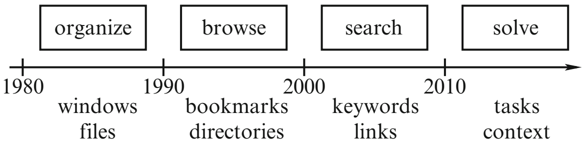


Fig. 1.5 Natural stages in the evolution of the user interface

References

1. Apple Inc. http://www.apple.com/iphone/features/siri.html. Accessed Oct 2011
2. Bellegarda, J.R.: Latent semantic mapping. In: Deng, L., Wang, K., Chou, W. (eds.) Signal Processing Magazine, Special Issue on Speech Technology and Systems in Human-Machine Communication, vol. 22(5), pp. 70–80, Sep 2005
3. Berry, P., Myers, K., Uribe, T., Yorke-Smith, N.: Constraint solving experience with the CALO project. In: Proceedings of Workshop on Constraint Solving Under Change and Uncertainty, pp. 4–8 (2005)
4. Buchanan, B.G., Shortliffe, E.H.: Rule–Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison–Wesley, Reading (1984)
5. Cheyer, A., Martin, D.: The open agent architecture. J. Auton. Agents Multi-Agent Syst. **4**(1), 143–148 (2001)
6. Fu, W.-T., Anderson, J.: From recurrent choice to skill learning: a reinforcement-learning model. J. Exp. Psychol. Gen. **135**(2), 184–206 (2006)
7. Gasic, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., Young, S.: Training and evaluation of the HIS POMDP dialogue system in noise. In: Proceedings of 9th SIGdial Workshop Discourse Dialog, Columbus, OH (2008)
8. Google Mobile. http://www.google.com/mobile/voice-actions (2008)
9. Guzzoni, D., Baur, C., Cheyer, A.: Active: a unified platform for building intelligent web interaction assistants. In: Proceedings of 2006 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, IEEE Computer Society, 2006
10. Kaelbling, J.L., Littman, M., Cassandra, A.: Planning and acting in partially observable stochastic domains. Artif. Intell. **101**, 99–134 (1998)
11. Kording, J.K., Wolpert, D.: Bayesian integration in sensorimotor learning. Nature **427**, 224–227 (2004)
12. Laird, J.E., Newell, A., Rosenbloom, P.S.: SOAR: an architecture for general intelligence. Artif. Intell. **33**(1), 1–64 (1987)
13. Microsoft Tellme. http://www.microsoft.com/en-us/Tellme/consumers/default.aspx (2008)
14. Morris, J., Ree, P., Maes, P.: SARDINE: dynamic seller strategies in an auction marketplace. In: Proceedings of ACM Conference on Electronic Commerce, pp. 128–134 (2000)
15. Nuance Dragon Go! http://www.nuance.com/products/dragon-go-in-action/index.htm (2011)
16. Rabiner, L.R., Juang, B.H., Lee, C.-H.: An overview of automatic speech recognition, Chapter 1. In: Lee, C.-H., Soong, F.K., Paliwal, K.K. (eds.) Automatic Speech and Speaker Recognition: Advanced Topics, pp. 1–30. Kluwer Academic Publishers, Boston (1996)
17. Sondik, E.: The optimal control of partially observable markov decision processes. Ph.D. Dissertation, Stanford University, Palo Alto, CA (1971)
18. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. Adaptive Computation and Machine Learning. MIT Press, Cambridge (1998)
19. Sycara, K., Paolucci, M., van Velsen, M., Giampapa, J.: The RETSINA MAS Infrastructure. Technical Report CMU- RI-TR-01-05, Robotics Institute Technical Report, Carnegie Mellon University, 2001
20. Thomson, B., Schatzmann, J., Young, S.: Bayesian update of dialogue state for robust dialogue systems. In: Proceedings of International Conference on Acoustics Speech Signal Processing, Las Vegas, NV (2008)
21. Vlingo Mobile Voice User Interface. http://www.vlingo.com/ (2008)
22. Wildfire Virtual Assistant Service, Virtuosity Corp. http://www.wildfirevirtualassistant.com (1995)
23. Williams, J., Young, S.: Scaling POMDPs for spoken dialog management. IEEE Trans. Audio, Speech Lang. Process. **15**(7), 2116–2129 (2007)
24. Williams, J., Poupart, P., Young, S.: Factored partially observable Markov decision processes for dialogue management. In: Proceedings of 4th Workshop Knowledge Reasoning in Practical Dialogue Systems, Edinburgh, UK (2005)