Large–Scale Personal Assistant Technology Deployment: the Siri Experience

Jerome R. Bellegarda

Apple Inc.
Cupertino, California 95014, USA
jerome @ apple.com

Abstract

Natural language interaction has the potential to considerably enhance user experience, especially in mobile devices like smartphones and electronic tablets. 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. Multiple voice-driven initiatives, such as Apple's Siri, have now reached commercial deployment. Bringing this technology into the real world raises a number of issues that ordinarily are not brought to the fore by the research practioner. Yet paying close attention to such aspects is critical to the success of the associated product. This paper discusses some of the attendant choices made in Siri, and speculates on their likely evolution going forward.

Index Terms: Voice-driven human-computer interaction, spoken language interpretation, dialog system integration.

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.

Multiple voice-driven initiatives have now reached commercial deployment, among others Apple's Siri [1], Google's Voice Actions [2] and Google Now [3], Microsoft's Bing Voice Search [4], Nuance's Dragon Go! [5] and Nina [6], and many startup efforts like Speaktoit [7]. 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 2 (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, in this case who will be invited (*John Smith* rather than *John Monday*) and when the meeting will take place (*this coming Monday in the afternoon* rather than, say, *last Monday in the middle of the night...*).

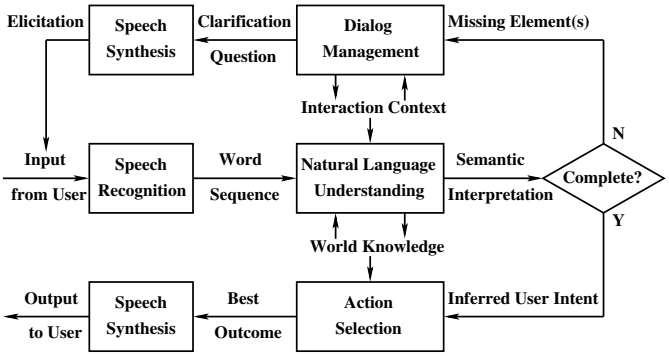

Figure 1: *Overview of "Intelligent Personal Assistant" Interaction Model.*

An overview of the underlying interaction model is given in Fig. 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 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 1990's [8]. 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 sudden enthusiasm for Siri and similar systems? While the interaction model of Fig. 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 alternative outlook 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. Crucial to this process is the unraveling of a tangle of issues that research practioners normally do not face, involving intertwined elements such as system integration, scalability, latency, and robustness. This contribution accordingly examines some of the choices made in the deployment of Siri, as well as the benefits that ensued.

The material is organized as follows. The next section briefly reviews the two standard frameworks for semantic interpretation, in order to illuminate their respective advantages and drawbacks. In Section 3, we focus on the choices made in Siri, and discuss in particular how essential they proved to a successful deployment. Finally, the paper concludes with some prognostications regarding the likely evolution of the personal assistant model.

2. Semantic Interpretation

Two major approaches to semantic interpretation are: (i) the rule-based framework underpinning expert systems and similar ontology-based efforts, and (ii) the statistical framework characteristic of data-driven systems.

At its core, the former draws its inspiration from early systems, such as MYCIN [9], firmly rooted in the artificial intelligence (AI) tradition [10]. These systems rely on an inference engine operating on a knowledge base of production rules. Their original purpose was to create specialized agents aimed at assisting humans in specific domains (cf., e.g., [11]). 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 [12]. This was later extended to multi-agent scenarios where distributed intelligent systems can model independent reactive behavior (cf., e.g., [13]). In the early to mid-2000's, DARPA's PAL (Perceptive Assistant that Learns) program channeled 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 [14].

In contrast, the statistical approach to semantic interpretation 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. Applying probabilistic methods to natural language understanding involves 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) [15]. The key features of the POMDP approach are (i) the maintenance of a system of beliefs, continually updated using Bayesian inference, and (ii) 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 [16]. Note that Bayesian belief tracking and reward-based reinforcement learning are mechanisms that humans themselves appear to use for planning under uncertainty [17]. For example, experimental data shows that humans can implicitly assimilate Bayesian statistics and use Bayesian inference to solve sensorimotor problems [18]. 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 [19].

2.1. Major Trade-Offs for Rule-Based Framework

The DARPA's PAL program culminated into a prototype dubbed CALO, for the Cognitive Assistant that Learns and Organizes. By most accounts, CALO met the requirements for which it was designed, but because of its heterogeneity and complexity, it proved difficult for non-experts 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 [20].

An example of such design 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 [20]. An application based on this framework consists of a set of loosely coupled services interfacing with specialized task representations crafted by a human expert. This strategy 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).

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) as well as associated rule sets that perform actions within and among concepts (e.g., the date concept derives a canonical date object from a word sequence such as *Monday at 2*). The active ontology can therefore be viewed as an execution environment.

A major downside of this approach 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.

Another obvious bottleneck 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 normally 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 [21].

2.2. Major Trade-Offs for Statistical Framework

The widespread deployment of POMDP-based spoken dialog systems proved equally challenging 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. Finally, the quality of any particular policy is quantified by assigning rewards 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.

Making the POMDP framework tractable for real-world tasks therefore 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 [22]. 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]. And finally, the reward function must be carefully tuned to suitably encapsulate the complexities of the domain, while remaining simple enough to support iterative optimization [24].

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, the computational complexity of a single inference operation is $\mathcal{O}(N^2)$, where N is the number of possible system states. Thus, for even moderately large values of N, exact computation becomes intractable. The necessary approximations all have drawbacks, be it in terms of search errors, spurious independence assumptions, or quantization loss from master to summary space [25]. But perhaps the most significant impediment to pervasive deployment across multiple domains is the need for a massive amount of dialog data to train the large number of parameters involved. Since gathering that much data beforehand is intrinsically unworkable, user simulation is often resorted to in order to generate synthetic reinforcement data. These techniques, however, are still far from perfect [26].

3. The Siri Experience

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 mentioned earlier as the intermediate layer between mobile I/O and web services.

The choice of the rule-based rather statistical framework was driven by practical considerations. First, even though it may well be intrinsically more expressive in the long run, a POMDP system would require a large amount of data which simply was not available in the type of domains considered. Second, user simulation was deemed to suffer from too many caveats to provide ersatz training material robust enough for the general user envisioned.

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 system integration, scalability and latency, and robustness issues.

3.1. System Integration

Arguably, system integration is the cornerstone of a successful personal assistant deployment: the technology must be pervasive in order to reliably support users in their day-to-day activities. Since the Siri architecture mediates mobile I/O with web services through domain modules, this assumes progress on three fronts: (i) a deep integration within the operating system, so as to leverage user-level information such as contact, calendar, email, and other personal data; (ii) an integration of assistant capabilities across domains, to enable potential coreference resolution and leverage any other temporary context state; and (iii) a broad integration with a variety of web services, to offer an appealing palette of utility and support successful transactions in the selected domains.

From a realistic point of view, at best a startup company would only ever be able to address (ii). Running the personal assistant as a separate application inherently limits the level of integration that can be achieved with the operating system. And at the other end of the spectrum, negotiating with established web services almost always requires marketing muscle that startups, by definition, are lacking. Note how oblique such considerations are to the technical aspects of semantic interpretation. They "merely" involve low-level SPI access for data acquisition and manipulation, yet this ability is obviously vital to support higher-level inference algorithms.

Incidentally, an ancillary benefit emerges from having access to all layers of the operating system: know-how to make speech and language algorithms more resource-efficient. On mobile devices, low power consumption and low memory use are critical, and an otherwise useful feature which places unreasonable demands on either is doomed to fail.

3.2. Scalability and Latency

Once integration issues are resolved, the next step is to scale the system up to the expected load level, in order to ensure an acceptable response time in all situations. Again, this is not something ordinarily brought to the fore by the research practioner, but it is an integral part of system deployment. There are two distinct aspects to keep in mind: (i) data bandwidth, which must be large enough to accommodate the peak number of service requests, and (ii) computing power, which must be sufficient to avoid perceived latency in the associated transactions. This has repercussions in terms of the basic architecture of the system: it must be modular enough that it can take advantage of

Table 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.*

any available new server to meaningfully grow capacity.

Two other aspects of scalability may be somewhat more familiar: deployment in multiple languages, and large-scale analytics for the purpose of model adaptation. Nowadays, and particularly when it comes to mobile devices, it is not really an option to release any feature in one language only. Since every language has its idiosyncrasies, this inevitably complicates the process, not just in terms of engineering resources but also in terms of quality control. In addition, adaptation must be factored in at the outset, with proper feedback loops in place to suitably leverage any adaptation data.

3.3. Robustness

Once integration and scalability issues are under control, robustness issues need to be addressed. Robustness is often cast in terms of acoustic noise from unpredictable environments, speaker variability due to dialects and accents, and perhaps emotional shifts and associated linguistic and prosodic changes. All such issues indeed must be considered when training the underlying models.

But another important, yet often overlooked, aspect of robustness 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 usually 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 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 pivotal to a successful deployment.

4. Discussion

We have examined some of the challenges to be overcome in the 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 touched upon the inherent complementarity between the two major frameworks within which to perform this interpretation. Ontology-based systems, such as Siri, adopt a top-down outlook which requires upfront labor-intensive human expertise and must be carefully tuned for optimal performance, but are better suited for initial deployment in well-defined domains across multiple languages. In contrast, because of their bottom-up outlook, data-driven systems based on POMDP can be run in completely automated fashion and have the potential to be more robust, as long as they are trained on enough quality data that has to be gathered beforehand.

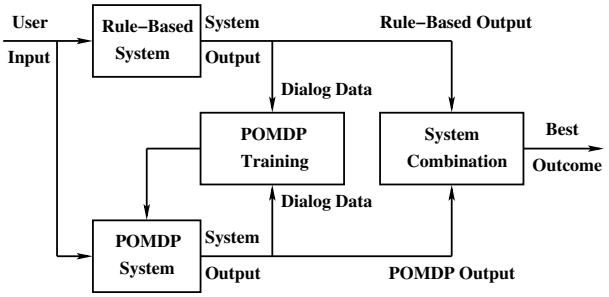

Figure 2: *Toward the Convergence of Rule-based and Statistical Frameworks.*

Such complementarity bodes well for an eventual convergence between the two approaches, perhaps by way of the virtuous cycle illustrated in Fig. 2. First, the deployment of a rule-based system provides some real-world dialog data that can be used advantageously for POMDP training, without the drawbacks inherent to data synthesis via user simulation. This in turn enables the deployment of a statistical system, 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.

This in turn sets the stage for taking 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. 3, 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.

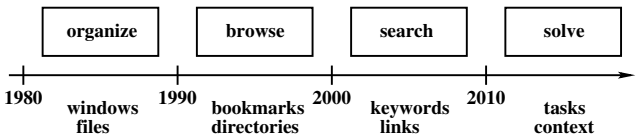

Figure 3: *Natural Stages in the Evolution of the User Interface.*

5. References

[1] Apple Siri, http://www.apple.com/ios/siri/, October 2011.

[2] Google Mobile, http://www.google.com/mobile/voice-actions, 2008.

[3] Google Now, http://www.google.com/landing/now, 2012.

[4] Microsoft Tellme, http://www.microsoft.com/en-us/Tellme/consumers/default.aspx, 2008.

[5] Nuance Dragon Go!, http://www.nuance.com/products/dragon-go-in-action/index.htm, 2011.

[6] Nuance Nina, http://www.nuance.com/for-business/by-solution/customer-service-solutions/solutions-services/mobile-customer-service/nina/index.htm, 2012.

[7] Speaktoit Assistant, http://www.speaktoit.com/index.htm, 2012.

[8] Wildfire Virtual Assistant Service, Virtuosity Corp., http://www.wildfirevirtualassistant.com, 1995.

[9] B.G. Buchanan and E.H. Shortliffe, *Rule–Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Reading, MA: Addison–Wesley, 1984.

[10] J.E. Laird, A. Newell, and P.S. Rosenbloom, "SOAR: An Architecture for General Intelligence", *Artif. Intell.*, Vol. 33, No. 1, pp 1–64, 1987.

[11] J. Morris, P. Ree, and P. Maes, SARDINE: "Dynamic Seller Strategies in an Auction Marketplace," in *Proc. ACM Conf. Electronic Commerce*, pp. 128–134, 2000.

[12] A. Cheyer and D. Martin, "The Open Agent Architecture," *J. Autonomous Agents and Multi-Agent Systems*, Vol. 4, No. 1, pp 143–148, 2001.

[13] K. Sycara, M. Paolucci, M. van Velsen, and J. Giampapa, "The RETSINA MAS Infrastructure," Technical Report CMU-RI-TR-01-05, Robotics Institute Technical Report, Carnegie Mellon, 2001.

[14] P. Berry, K. Myers, T. Uribe, and N. Yorke-Smith, "Constraint Solving Experience with the CALO Project," in *Proc. Workshop Constraint Solving under Change and Uncertainty*, pp. 4–8, 2005.

[15] E. Sondik, "The Optimal Control of Partially Observable Markov Decision Processes," Ph.D. Dissertation, Stanford Univ., Palo Alto, CA, 1971.

[16] J.L. Kaelbling, M. Littman, and A. Cassandra, "Planning and Acting in Partially Observable Stochastic Domains," *Artif. Intell.*, Vol. 101, pp. 99–134, 1998.

[17] W.-T. Fu and J. Anderson, "From Recurrent Choice to Skill Learning: A Reinforcement-Learning Model," *J. Exp. Psychol. Gen.*, Vol. 135, No. 2, pp. 184–206, 2006.

[18] J K. Kording and D. Wolpert, "Bayesian Integration in Sensorimotor Learning," *Nature*, Vol. 427, pp. 224–227, 2004.

[19] J. Williams, P. Poupart, and S. Young, "Factored Partially Observable Markov Decision Processes for Dialogue Management," in *Proc. 4th Workshop Knowledge Reasoning in Practical Dialogue Systems*, Edinburgh, UK, 2005.

[20] D. Guzzoni, C. Baur, and A. Cheyer, "Active: A Unified Platform for Building Intelligent Web Interaction Assistants," in *Proc. 2006 IEEE/WIC/ACM Int. Conf. Web Intelligence and Intelligent Agent Technology*, 2006.

[21] J.R. Bellegarda, "Latent Semantic Mapping," *Signal Proc. Magazine, Special Issue Speech Technol. Syst. Human–Machine Communication*, L. Deng, K. Wang, and W. Chou, Eds., Vol. 22, No. 5, pp. 70–80, September 2005.

[22] B. Thomson, J. Schatzmann, and S. Young, "Bayesian Update of Dialogue State for Robust Dialogue Systems," in *Proc. Int. Conf. Acoustics Speech Signal Processing*, Las Vegas, NV, 2008.

[23] J. Williams and S. Young, "Scaling POMDPs for Spoken Dialog Management," *IEEE Trans. Audio, Speech Lang. Processing*, Vol. 15, No. 7, pp. 2116–2129, 2007.

[24] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, Series on Adaptive Computation and Machine Learning, Cambridge, MA: MIT Press, 1998.

[25] M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and S. Young, "Training and Evaluation of the HIS POMDP Dialogue System in Noise," in *Proc. 9th SIGdial Workshop Discourse Dialog*, Columbus, OH, 2008.

[26] H. Ai, J. Tetreault and D. Litman, "Comparing User Simulation Models for Dialog Strategy Learning," in *Proc. NAACL-HLT*, Rochester, NY, 2007.