Natural Language Engineering **10** (2): 1–4. © 2004 Cambridge University Press
DOI: 10.1017/S1351324904003390 Printed in the United Kingdom

Industry watch

ROBERT DALE
Centre for Language Technology, Macquarie University, Sydney, Australia

(*Received 18 March 2004*)

Here's something I read the other day:

> IVR [interactive voice response] technology is at a point now where consumers almost cannot tell the difference between talking to a person and talking to a computer.

So said Richard Feinberg, director of the Purdue University Center for Customer Driven Quality, quoted in the online CRM Daily, February 18th, 2004.

I suspect Richard lives on a different planet from the one I live on. Here's an interaction with my local automated taxi cab booking service:

System:	To book a taxi, say yes. To check a booking, say check. For anything else, say no.
Me:	Yes.
System:	Are you travelling from 14A Spears Avenue Balmain?
Me:	Yes.
System:	And what suburb are you going to?
Me:	Milson's Point.
System:	How many passengers are travelling?
Me:	One.
System:	Are you ready now?
Me:	Yes.
System:	Confirming your booking for a taxi from 14A Spears Avenue Balmain to Milson's Point for one person leaving now. Is that correct?
Me:	Yes.
System:	Your booking for the next available taxi is confirmed. Your reference number is . . .

I was talking to a machine, and it was pretty obvious that I was doing so. This conversation is as natural as the conversation you might have when being interrogated by a policeman. Yes, ScanSoft and Nuance will treat you to demos of applications that are a little more naturalistic than this, but they're still a long way away from being like talking to a person. I've yet to see a deployed application that comes close to the naturalism of, for example, the Communicator system put together by Alex Rudnicky's team at CMU (Rudnicky et al 1999), and I think few people would be fooled into thinking even that was a real person. The reality is that talking to a machine is a very different thing from talking to a person, and is

likely to remain so for quite some time; most successful deployed speech recognition systems are like the taxi booking application above.

Which is not to say that systems which engage in unnatural conversations are a bad thing. These systems work, and they work because they make use of clever tricks. Again taking the taxi booking application as an example:

- The system uses reverse lookup on the caller's phone number to get address information, so there's no need to ask for that information — which is a very good thing, since the recognition accuracy on street addresses would be likely to be rather poor.
- Other than identifying the suburb (a recognition task handled by a grammar that covers on the order of 800 names), the destination address isn't requested; the taxi driver who takes the job only cares about the destination suburb, and can get the precise address from the passenger when they are in the cab.
- Most people who call taxis want them as soon as possible; so asking whether the caller is ready to go now is a far better dialogue management strategy than asking something like 'When do you want the taxi?'. This particular system can engage in a conversation to determine a specific requested time if it has to, but it avoids doing this if it can.

Even with all these ways of avoiding tricky situations, it is typical for systems of this kind to only automate around 70–75% of the calls they receive. With the right business model, that's enough to turn a profit for someone, and good enough not to frustrate too many callers to the extent that they use a competitor instead.

But, as noted above, the conversation is not natural. It's not like any of those conversations with machines you see in science fiction movies, nor is it like the conversations with machines we see in corporate 'here's what the future will be like' videos. What I find interesting is the dramatic difference between, on the one hand, our visions of potential dialog systems — exemplified by those fictional portrayals as well as the kinds of systems we see being developed in research labs — and on the other hand the kinds of systems that are being deployed today. Why is there this difference?

There might be any number of reasons for this.

- Perhaps developers of existing deployed systems just aren't aware of what is possible; they just don't have the right skills in-house. If only those industry people would come to our conferences! They should be hiring our PhD students so that they have access to all the wonderful ideas that are discussed in our research papers, and begin to realise the full potential of the technology.
- Perhaps they're using the wrong tools. Over-reliance on dialog development frameworks like VoiceXML imposes unnecessary restrictions on the kinds of systems that can be built. For heaven's sake, VoiceXML doesn't even incorporate a notion of dialog history, far less any of the machinery you'd need to reason about a user's intentions!

Anyone who has tried to deploy a real dialog system will recognize that neither of those possibilities are very plausible. The reality is that it's remarkably hard to

get even a simple system to work in a real working environment. It's one thing to set up a demo where self-selecting users can patiently work through an interaction with a system in a non-noisy environment; it's another thing altogether to put in place a system that will work for a random caller, whose voice characteristics may be far from what was envisaged when the system's acoustic models were developed, who might already be impatient to get the service or information they need, who may have limited experience of interacting with this kind of technology, who may have all manner of inappropriate expectations of the technology (perhaps they've been watching too many science fiction movies or even — heaven forbid — those 'here's what the future will be like' videos), and who, to cap it all, may be calling on a restricted-bandwidth mobile phone outside on a busy street. Given these circumstances of use, it doesn't make a lot of sense to try to develop really smart systems. The smarter you try to make a system, the greater the risk of failure when misrecognitions occur; and misrecognitions *will* occur.

Well, you might say, this is a temporary state of affairs. Once our speech recognition capabilities get better, we'll be able to deploy the kinds of systems we've always dreamed off: systems which are indistinguishable from real human agents.

But is that really what we want? Perhaps we focus too much on trying to make our systems 'natural'. We don't have anything like a theory of human-machine communication, and so in thinking of how we might extend these technologies, we fall back on theories of human-human communication. Why should we expect the principles of human-human communication to be applicable to interactions with machines? It's plausible that there might be some commonalities; but surely any such hypothesis should not be accepted without careful questioning. When I call up the system introduced earlier to book a taxi, I have a very focussed purpose, there's no scope for chatting about the weather or other niceties, there's no politeness; I'm just interacting with a machine to get the job done, and it's more like using a touch-tone telephone to navigate a bank's menu-based system than it is like talking to a human teller in a bank. Why should I expect this to be like talking to a person? Talking to a machine is fundamentally different from talking to a machine, just as typing query terms into a search engine is very different from talking to a librarian. For those who can remember HAL, perhaps we have been too seduced by science fiction.

* * *

Most people will by now have had some exposure to deployed speech recognition applications. Here in Sydney, I'm aware of systems that will let you book taxis, as in the above example, or buy and sell shares, check flight times and determine your air miles status, get numbers from directory assistance, and place bets on horse races. If you've managed to avoid these kinds of systems so far, get in touch with a local speech application developer and ask them for a demo CD, or the numbers of demo lines you can call.

Deployed systems that use VoiceXML are only now beginning to appear; until relatively recently, designing a dialog meant pasting together chunks of C++ code.

You can find out more about VoiceXML at www.voicexml.org. Unfortunately, the demise of the once relatively unrestricted Nuance Developer Program means that it's now quite hard to get a free VoiceXML development environment that you can use on your desktop; short-sighted, surely, if what we really want to see is lots of skilled-up VoiceXML developers, and a perfect opening for the Microsoft-sponsored SALT. The VoiceXML vs SALT debate will be covered in a future *Industry Watch.*

Each year, when I teach my students about spoken language dialog systems, I try to find a live demonstration of a current working airline travel reservation system. I call up the system, and I try to work through the sample dialog presented in the early GUS paper (Bobrow et al 1977)— impressive back then, even though input at the time was from the keyboard because there wasn't a decent working speech recogniser around at the time. Every system I tried got hopelessly lost until I tried the CMU Communicator system, which performed remarkably well. You can try out CMU's system yourself: dial-up instructions are at www.speech.cs.cmu.edu/Communicator.

For portrayals of speech and language technology in the movies, see www.ics.mq.edu.au/~rdale/resources/nlpinthemovies/. For down-to-earth alternatives to natural conversational interfaces, see CMU's Universal Speech Interface at www-2.cs.cmu.edu/~usi/, or read about the European Telecommunications Standards Institute's generic spoken command vocabulary for ICT devices and services in report ESTI ES 202 076, available from http://pda.etsi.org/pda/AQuery.asp.

If you have views on any of the above, drop a note to `rdale@acm.org`, and I'll follow them up in a future column.

References

Bobrow, D., Kaplan, R., Kay, M., Norman, D., Thompson, H. and Winograd, T. (1977) GUS, a frame-driven dialog system. *Artificial Intelligence*, **8**(2): 155–173.

Rudnicky, A., Thayer, E., Constantinides, P., Tchou, C., Shern, R., Lenzo, K., Xu W., and Oh, A. (1999) Creating natural dialogs in the Carnegie Mellon Communicator system. *Proceedings of Eurospeech 1999*, **4**, 1531–1534.