The Future of Automatic Speech Recognition

Ellis Cave – 3/14/2012

The Foundation

The invention of the telephone revolutionized the way humans communicated. The ability to immediately converse with another person thousands of miles away changed civilization in a fundamental way, and that change is still going on.

One of the more significant changes wrought by the introduction of the telephone was the way that people conducted business. Before the telephone, most businesses were primarily local entities, serving customers within an area that was limited by the distance that a customer could travel in a reasonable time. With a telephone, business customers didn't have to physically go to the business location to conduct business. Customers could call the company to get product information and place orders, saving time and money.

Businesses quickly realized that they could significantly expand their clientele if they could support business operations over the phone. Customer service telephone centers were born, and forward-looking companies were able to expand their business nationally and even globally. However, companies soon discovered that supporting a large customer service staff to man the telephones was a major expense for the business. Businesses began to look for ways to reduce the number of people required to support their customers, while still providing the services the customers wanted.

Various schemes were hatched to reduce call center costs, including automatic call distributors to queue callers for agents, and voice annunciators to automatically speak basic repetitive information to callers. Unfortunately, none of these schemes seemed to help much in slowing the spiraling costs of customer services. The basic fact was, the more customers a business had, the more people the business needed to support its' customers over the phone.

It wasn't until the early 1960s that a technology which could really help solve the customer support cost problem was developed, and this breakthrough occurred surprisingly by accident. Even then, it took nearly 20 years after its introduction, to become ubiquitous enough that it could affect the bottom line of businesses' support costs.

The Beginnings of Automation

The idea of automated self-service in call centers was started, most likely unintentionally, by the Bell Telephone company in 1963. The Bell System, a monopoly in the US back then, introduced the touch-tone phone at the 1963 Seattle World's Fair. Bell touted the new key-dialing scheme as a much faster way to dial phone numbers when compared with the older rotary-dial system.

Touch-tone phones didn't reach the general public until the early 1970's. However, once the phones became readily available, it wasn't long before it was discovered that the tones produced by these new touch-tone phones could be used to control interactions with a remote device during a call, and the world hasn't been the same since.

In the mid 1980's most phones (in the US at least) were equipped with a touch-tone keypad. Detecting which touch-tone key the caller pressed was a highly reliable process over the phone network, so engineers and developers, who were being pushed to come up with ways to automate business services and lower costs, knew they could use that fact to automate conversations. They came up with the idea to present callers with a spoken menu of choices, after which the caller could make their selection, using the dialing keys on the caller's phone.

Businesses loved the idea. Customer service costs could be dramatically reduced. A touch-tone/voice response system (called an Interactive Voice Response systems or IVR) could pay for itself in six months or less. Some companies (one was where I worked) built their business around selling other businesses systems that would provide this single function– automating customer service tasks to reduce costs. It didn't take long before "press one for this, press two for that" became the mantra for the customer service department of any modern, cost-sensitive business.

However, as these new telephone-keypad-automated applications were rolled out and became more complex, businesses found that customers started complaining about the inability to talk with a live person. This was somewhat perplexing to the business owner, as all of the same service options were available from the automated system as from the live agents. Businesses heard the complaints, but the siren call of lower costs tended to drown out the complaining customers. After all, the customers were still getting the information they called about (at least most of the time). It just required the customer to learn a simple new technique to get that information. Businesses wrote off the complaints as just customer reluctance to embrace the new technology.

However, some customer-sensitive businesses started investigating their customer complaints, and it didn't take too long to pinpoint what the problem was. The touch-tone menu technique that the automated systems used to let users select their tasks was frustrating the callers. Instead of allowing a caller to simply tell the customer service agent what they wanted, a caller was asked to listen to a list of menu choices, and then press buttons to select their choice. To the caller, this was not the most efficient way for them to communicate their issues. It was like the game twenty questions, except the caller didn't want to play the game. To make matters worse, the menu choices that were presented may or may not provide a selection for the issue that they were calling about.

The customer service department of most businesses handles a wide range of issues. A bank customer may want to know about account balances, transferring money, verifying a deposit, or if a check cleared. Callers to an airline may ask about purchasing tickets, flight schedules, gate numbers, cancelled flights, meals and seats, frequent flyer miles. The list of possible issues a caller might have could be quite long, and a spoken menu listing all the possible choices could take several minutes to present. This is a huge problem for a menu-based voice system. Developers tried many schemes to help mitigate the long menu issue, but unfortunately, nothing seemed to do much to improve the basic problem.

Given the state of the art in the 1980's, developers didn't have much choice about implementing menus for task selection. When the only user input method is a 12-button keypad on the telephone, the only practical option to determine a users' intention is to use menus. So, the tyranny of the menu was perpetuated. Business knew that their customers would not necessarily be pleased with a move to automate their customer service calls, but the economics of automation tended to win out nearly every time.

The Speech Solution

High-performance speech recognition systems from vendors such as Speechworks and Nuance became available in the mid-1990s, which gave developers hope that the menu issue would finally go away. Instead of going through interminable menus, speech vendors claimed that speech input would allow the system to just ask "How may I help you?", and then let the user say what they want to do. Speech recognition technology vendors even branded their products with catchy phrases like "Say Anything" and "How May I Help You" to make their point. True conversational systems would be built. Life would be good.

If this was true, then the whole tradeoff between automation and customer satisfaction could be avoided. No more frustrating menus, just say what you want. Automation would be embraced by customers. True conversational systems would be available to everyone.

Unfortunately, the reality was pretty far from the speech recognition vendor's hype. Yes, speech systems could be built that would allow callers to say what they wanted. What the speech vendors failed to mention, is that in order to build such a system, one needed to jump through a daunting set of hoops. Speech recognition engines (often called Automatic Speech Recognizers or ASRs) don't just automatically transcribe arbitrary speech into text, they need to be trained. Furthermore, just because you have the text of what someone said or typed, that doesn't mean that the system will understand what the user is asking. Converting speech to text, and then understanding that text are both hard problems, yet it takes solving both of those problems to correctly handle a caller's answer to an open-ended question like "How may I help you"

The Reality of Speech Technology

Let's take a brief look at the process of getting a speech engine to handle open-ended questions. For the best accuracy converting speech to text, speech engines need to be pre-trained on all of the words, phrases and sentences that a caller might be expected to say to it. If a caller says a word or phrase that the ASR has not been trained on, it will try to morph that utterance into something it HAS heard before, which usually doesn't work out too well. Worse yet, if one tries to train the ASR system on EVERYTHING that might be said, error rates go through the roof, as there will often be too many words and phrases that sound alike to the ASR engine, and it gets them all mixed up.

The best approach is to train the ASR engine on as narrow a domain of utterances as possible, where all of the training utterances are specifically in the application domain. One shouldn't train the ASR to recognize the phrase "What is the price of a pepperoni pizza?" if the application is about banking. In fact, the words "pepperoni" and "pizza" should probably both be left out of the ASR's training vocabulary, just to keep the possible word choices minimized. Then it was up to the system designer to somehow prevent the caller from saying things that were out-of-vocabulary for the ASR, which is a non-trivial task. That was the state of the art in ASR systems then, and it still is today.

Even with a narrowed vocabulary, the basic problem remains that no customer will say the same thing to a "how may I help you" prompt. So how

can we train an ASR system with the expected responses, if there ARE NO standard expected responses from the user?

How to Make Speech Work

If you ask a speech scientist about this problem, they will all say pretty much the same thing: Collect lots of utterance data. However, you must collect utterance data that will exactly match the type of utterances that you expect to hear in the final application. Collect the real answers when a caller is asked "How may I help you?", in your specific environment. Make sure that the utterance collection is being done in EXACTLY the same environment as the final automated system will see – the same prompts, the same type of callers, etc.

The more recordings that are collected and analyzed in your specific application domain, the fewer mistakes the ASR engine will ultimately make. However, the speech vendors didn't say too much about HOW one should come up with this large corpus of specific utterances that must be captured as responses to specific prompting questions.

If one is able to somehow collect these utterances, all of those collected utterances will need to be transcribed and categorized, so the ASR engine can be trained to accurately convert spoken phrases to text. That same transcribed & categorized utterance data can be also used to train the categorization process that analyzes the text from the ASR to determine what the caller wants to do. The goal is to get a large corpus of utterance data that represents the statistical majority of the ways that callers will respond to the "How may I help you?" prompt.

I won't go into further into all the technical steps of the process to get an ASR engine to correctly analyze a caller's utterances, but suffice it to say that the process can take many months, require the collection and transcription of hundreds of thousands of caller utterances, and cost many thousands of dollars. There are shortcuts that can be taken which significantly reduce the cost and effort, but performance usually suffers as well. Because of the extensive work required, true natural language understanding of even a single prompt-response dialog turn is usually relegated to very-high-volume applications with well-defined domain-specific vocabularies, which can spread the up-front expense over millions of calls.

Even after successfully jumping through all of the technical hoops training an ASR, a developer may get the ASR engine to correctly categorize only between 60% and 80% of the customer's responses. And, the more

selection categories you have, the harder it is for the ASR to identify the correct one.

Back to the Future

The difficulties with open-ended user responses have caused automated dialog application developers to fall back on a tried-and true scheme to determine the user's request: menus. Modern automated-dialog systems simply replaced the touch-tone menus of the 80's with the speech-selected menus of the 90's.

Today, instead of "Press one for your account balance, or press two to transfer money", you get the prompt "Do you want your account balance, or to transfer money?" This social engineering scheme for prompts considerably narrows the scope of what a typical user will respond to the prompt, which helped tremendously in raising ASR accuracy. A caller's typical response to the aforementioned prompt is typically either "account balance" or "transfer money", which drastically reduces the complexity of the ASR's decision tree, and thus increases accuracy.

Socially-engineered prompts essentially gave the caller a strong hint about how they were supposed to respond, which made the ASRs' job much easier. This allowed developers to define the expected user responses to the ASR in a fairly small set of data called a "grammar" instead of the large statistical training sets that would be needed with a "how may I help you" prompt. ASR accuracy with these "directed dialogs" as they were called, was much improved, and speech applications could be developed cost-effectively.

The directed dialog approach made speech applications much more reliable and practical, but the initial phases of a dialog process was back to being a menu. Speech-selected menus suffer from exactly the same malaise as the touch-tone menus of the past – long prompts, mazes of hierarchical menus, and frustrated users. It seemed that no matter how we tried, we couldn't get away from the tyranny of the menu.

Other ASR Problems

Speech input systems have another problem that touch-tone systems don't have – errors. Touch tone input, though it may be cumbersome, is highly reliable. When a user presses "one" on their phone, the chances are very high that the IVR in the network will correctly determine that fact. When a user speaks the selection "account balance", there are many factors that considerably lower the probability that the IVR system will get the user's selection right. The caller may have a heavy accent. There may be lots of

background noise. The caller may just speak their request in a strange way. In each case, the ASR either can't determine the users' request, or worse, picks the wrong selection for the user. This inaccuracy of ASR engines has many bad side effects.

While long menus typically cause the most user frustration, the "I'm sorry but I didn't understand you" prompt runs a close second in frustrating callers. It is the nature of ASR that, if it doesn't understand your utterance the first time, chances are it won't understand on the second or third try, either. Several "don't understand" prompts in a row will invariably cause callers to hang up in frustration. Developers have tried all kinds of schemes to try and soften the repetitive "didn't understand" issue, but nothing seems to work very well.

If the ASR engine picks the wrong task, the user's frustration is even worse. Trying to recover from a wrong selection is a daunting problem in dialog design, so dialog designers tend to validate the user's choice by reiterating the choice – e.g. "Did you say account balance?" While this reiteration is a conservative approach, it also tends to frustrate callers, who get tired of the validation prompts after each selection.

The Failed Promise of Speech

For all of the above reasons, speech input has not been the panacea for conversational systems that we originally expected it to be. Don't get me wrong, speech has helped greatly in automating certain customer service tasks that are cumbersome with key input. It is much easier to answer the question "What city are you flying to?" with the spoken word "Philadelphia" than spelling the word out on a 12-button keypad. Unfortunately, the set of tasks that truly benefit from this type of speech input are only a small portion of most customer service tasks.

Speech recognition technology has allowed certain common customer service tasks to be automated. It just hasn't solved the problem of letting the user SELECT those tasks very well. Once a user's requirements have been determined (called the task routing step) conversational systems have been very successful in automating the actual customer task interactions.

So, speech hasn't really solved our basic problem. Voice automation in the call center can definitely reduce costs, but still at the cost of user dissatisfaction

Where We Are Today

Humans learned to talk long before they learned to type - that is if they ever learned to type at all. I effectively slept through my high school typing class, never suspecting that in the future I would be using a typewriter keyboard every day. The human speech interface has evolved over millions of years to be the most efficient way for humans to communicate. We are most comfortable conversing with another human using speech to communicate our issues.

Customers like to interact with businesses in a personal way. They want to believe that a business knows their name, and knows about their history with the company issues wants their business. Callers believe that they will get the best service if they could just talk to a knowledgeable, well-trained CSR who knows them and knows there issues and preferences.

The problem we face is a technological limitation. One can automate customer service using speech today, but the speech interface with its menu structures is not user-friendly enough to make customers want to use it. 15 years ago, when the internet wasn't widely available, the customer had no choice. Struggle through the menus, or hope for a live service agent. Today, customers usually have a choice – struggle through IVR menus, or browse the companies' website for their information. The limitations of current ASR systems are causing customers to more and more choose the web.

The good news is that ASR systems are becoming increasingly more accurate in the process of transcribing spoken utterances into text. Error rates for ASR engines have been dropping steadily for years. Software algorithms now can adapt to speakers' accents or dialects, and adaptation times are shrinking, reducing training time. Automatic dictation/transcription technology, which converts speech into text, is moving into the mainstream. Commercial products, such as Nuance's "Naturally Speaking" software, are regularly used to automatically transcribe speech, when manual methods are not practical.

However, all of these improvements only solve part of the problem – the conversion of speech into text. The second part of the automated conversational problem that still must be solved is to "understand" what the transcribed text means.

In narrow domains, keyword spotting can do a fairly good job of detecting a caller's intention. If the words "account balance" appear somewhere in a banking customers' transcribed utterance, it is likely that they want to know how much money they have in their account. However, this assumption can be dangerous. The caller could have said something like "I don't want to

know my account balance just yet" – and then the machine blithely proceeds to tell the customer their account balance.

Today we can transcribe speech fairly accurately, and text is already in a form that is readily processed by a computer. The largest barrier still keeping us from a fully conversational system is getting a machine to understand the text output of the ASR – to truly "understand" what the caller is saying. When we reach this goal, we will then be able to do away with menu structures in automated dialog systems.

The Future of Conversational Systems

The science of understanding human language is called linguistics. The science of machine understanding of human language is called computational linguistics. Often considered an obscure backwater of computer science, computational linguistics is rapidly developing the tools and knowledgebase that will make it possible for machines to truly understand human language in its natural form. Computational linguistics is building the path that will lead us to one of the holy grails of automation: natural language understanding (NLU) by machines.

Computational linguistics (I will abbreviate from here on as CL) is on the verge of significant breakthroughs in understanding human language, in its natural form. The key to any automated conversational system is the capability to understand a human's spoken or written comments in their natural form – as if spoken or written to another human.

I predict that within the next five years, we will have automated conversational systems that will overcome the menu problem. These systems will be able to carry on dialogs where it will be difficult to determine whether the entity talking is human or automation. This is not to say that a human user would not be able to ask questions that will confuse the AI, but if the conversation stays in the AIs area of expertise, the conversation will be very natural.

It will probably be some time before you will be able to buy a handy housekeeping robot from the local appliance store. However, the capability to converse with a machine, using natural language, is coming very soon. Initially however, the conversational domains if these conversational AIs will be fairly narrow.

For example, there will be a banking AI system that can perform the duties of a traditional bank teller, at least over the phone or on the web. As long as the discussion is all about banking – your bank account, interest rates,

banking hours, etc. it will be hard to tell that you aren't interacting with a person. However, if your conversation deviates from the banking AI's intended domain, things go bad fairly rapidly. If you ask the banking AI for the price of a pizza from the local store, it will quickly become obvious that this customer service agent is a machine.

Five years from now, conversational AIs will probably not pass the Turing Test, at least as defined by Alan Turing in his 1950 paper *Computing Machinery and Intelligence*. It will be fairly easy for a human to detect a conversational AI, by simply leading the conversation away from that AIs area of expertise. However, in their specialty area, the AI will eventually become more capable than most humans, in that specific task.

A bank teller AI can be extensively trained in all aspects of the bank teller's job (which when you get right down to it, isn't all that complicated). As the AI performs its' task, learning algorithms will train that AI to continually adapt and improve its bank teller skills. At some point, the bank teller AI will become skilled enough at its job that it will out-perform human tellers in most situations. At this point it won't take long for human teller jobs to disappear.

Even if the bank teller AI is not quite as good as a well trained human teller, the economics of automation – on the job 24-7, never complains, low salary, no medical benefits, etc. - will drive live tellers out of a job. And don't forget, once you have one expert teller AI, you can create as many clone tellers as you like.

The first true AIs will be idiot savants – able to do one thing amazingly well, and nothing else well at all. Any job that primarily relies on the communication and exchange of known information to humans in a narrow domain, will be a prime candidate to be replaced by specialist AIs. Most information service jobs – customer service, technical support, sales, and other similar jobs will become automated, as soon as a domain-specific AI can be trained to handle the issues. AI companies will spring up, selling specialist AIs in various flavors- bank tellers, order takers, technical support specialists, receptionists, the list will be endless. The advent of the idiot savant AI will likely cause a major upheaval in the job market, and perhaps a major instability in society.

CL technology will also provide the underpinnings of another significant technology advance –real-time speech translation. Within the next 10 years, Translational AIs – another aspect of the idiot savant characteristic – will convert any language into another. One will be able to call up a friend in Japan, and converse with them naturally, even if you only speak English,

and they only speak Japanese. - I predict that this single advance will have a more significant effect on the melding of cultures and lowering of national barriers, than internet has had on these issues so far.

In the next few years, CL technology and specialist AIs will create a revolution in labor markets, forcing many workers to learn new skills to stay employed. And, it isn't clear that there will be new jobs available to employ all those who are put out of work by the AI revolution. Translation AIs will break down cultural, political, and national boundaries. If the Singularity is near, then CL with its idiot savant AIs will be leading the wave of tumultuous change - likely the first disruptive technology of the coming Singularity.