Voice User Interface Design for Automated Directory Assistance

Esther Levin and Amir M. Mané.

City College of New York, USA; Voice Advantage, USA
esther@cs.ccny.cuny.edu, amir@voice-advantage.com

Abstract

This paper focuses on the challenges that one encounters when building for commercial deployment an automated system for Directory Assistance (DA.) The design for an automated DA system needs to take into account constraints and requirements that arise from three distinct aspects of the application, namely, the business drivers, the user needs, and the strengths and weaknesses of voice technologies.

1. Introduction

Directory Assistance is the first, and probably still the most common 'information transaction' in which a caller is paying a fee for a morsel of information. On the face of it, this is a rather simple exchange: the user provides the names of the locality and the listing, and the system delivers a phone number. In reality, many obstacles must be overcome in order to deliver high quality service. As we will describe in this paper, any feasible design for an automated DA system needs to take into account constraints and requirements that arise from three distinct aspects of this application, namely, the business drivers for automation and quality customer care, the needs of the users as they are expressed in the voice user interface (VUI) and the strengths and weaknesses of the voice technologies. Given the large volume of DA calls (the total number of DA queries in 2004 was expected to be around 6.5 billion calls in the US alone and the revenues to reach nearly $6.8B[1])it is not surprising that automation of DA is regarded as 'the holy grail' for Voice Technologies. Voice Recognition and Speech Synthesis have the potential to significantly reduce the biggest contributors to the expense associated with delivering this service, namely the Operators and the Information Technology investment that is needed to support their work. At the same time, the service providers are sensitive to customers' perception of poor quality of service. After all, DA is an unusual service in that the customer is most often *paying* an additional fee above and beyond the cost of the call itself. Customer complaints, or even the fear of getting them, are increasing the challenge of VUI design: The goal is not merely to achieve a sufficient level of automation, but to do so while maintaining a high level of customer satisfaction [2].

This paper focuses on the set of specific challenges and questions that one must address in designing an automated system for DA. Note however, that we do not attempt to prescribe a solution to the different challenges; rather, we try to map the issues the designer has to consider and describe some alternative solutions and their implications.

2. Dialog Design for DA

The top-level design of the dialog for the automated system follows closely the call flow of the traditional DA service. In this section we will traverse the typical call-flow and discuss the specific manifestation of the DA challenges and possible ways to address them.

2.1. Finding the Locality

There are several good reasons for starting the dialog with the locality question. First of all, most callers have learned to expect this to be the first question, based on the pre-automation methodology that the service providers put in place. Second, the knowledge of the locality can help later in the dialog when listing name needs to be determined allowing to deploy locality-based listing language models rather than generic one, and/or in the database search.

There are several issues to watch for:

- Some localities may have a 'local pronunciation', that is, a way that people who live in the area refer to the town. Absecon, (a:b 'si: ken) New Jersey and Des Plaines ('des 'pleinz), Illinois are examples of town names that are pronounced by locals in a manner that would be unexpected to someone who is not familiar with them. Using the right local pronunciation is important not only for Voice Recognition, but also for Speech Synthesis, since unless you incorporate the appropriate pronunciation into the readout, the user may reject the locality confirmation even when the locality was recognized correctly.
- When asked for a 'city and state' the callers often respond with the name of the city only. They assume that the system is local and therefore knows the state, or that it has common geographical knowledge, and therefore it knows that New York City is in New York while Nashville is in Tennessee.

2.2. Listing Type

To perform a search in the directory database, the DA Operator needs to have three data elements:

- What is the locality of the desired listing?
- Whether the search should be conducted on the Residential or the Business & Government database?
- What is the name of the listing?

In practice, in the absence of Voice Automation, the caller is asked the second question only when the Operator cannot infer from the name of the listing whether the request is for business or residence. It is easy for the Operator to perform a simple 'meta analysis' of the listing name. If the caller says: "McDonalds restaurant" or "Jeff McDonalds" the operator will know that in the first case the request is for a business listing and in the second for residential. However, for an automated system it would be advantageous to determine the listing type before it needs to recognize the listing name. With the knowledge of the listing type the system can deploy substantially different dialog strategies for the different cases. For example, for residential listings, the system can take advantage of the structure of the listing name and ask separately for last name and, if necessary for first name. Narrowing the search early on can also improve the speech

recognition accuracy, and prevent a system from sounding 'stupid' in the case of misrecognition when a caller is asked to confirm a listing of a wrong type. In the rest of this section we assume that a question like "Are you looking for a business, government or residential listing?" was inserted into the call flow and the system can deploy different strategies for recognition and search for business or government (biz/gov) listings and for residential listings.

2.3. Handling Business and Government Requests

Business listings constitute less than 20 % of the total number of listings in the directory, but they account for the vast majority (roughly 80%) of the DA requests. Thus, the automation of business listings is essential for the success of a DA application. Several attributes of the task contribute to the challenges of its automation:

- The number of listings in the DA database is enormous: there are over 15 million biz/gov listings. Even when broken by localities, the number of biz/gov listings for a single large locality may exceed 400,000.
- The distribution of calls per listing is flat. Unlike the related task of toll-free directory assistance, where a small number of most frequently requested listings account for a big portion of the calls, the distribution of calls per listing in the white-pages regular DA is quite flat with a huge tail. That means that high automation rates can be achieved in this task only if all or almost all of the listings in the database are automated.
- The automated system needs to be updated on a daily basis, since each day a substantial portion of the listings is changed.
- Usability studies have indicated that often users refer to the business listing in a fashion that is different from the way that the business described itself in the directory. The variations include omission, addition and substitution of parts of the business name. For example, they ask for "Antonio's Pizzeria" when the business is listed as "Antonio's Brick Oven Pizza."; for 'Fuji Restaurant' when the listing name is 'Fuji Japanese steakhouse', or for 'Danny's Nails', when a listing name is " Creative Nails by Danny'.
- The entries in the directory are often ill suited for a spoken dialogue because of the formatting of the entry, abbreviations, typographical errors and more.

2.3.1. Elements for Solution:

Due to the large directory volume, the established methodology based on manually created grammars and recorded prompts for listing names that is currently deployed [3-5] for systems that automate small to medium size directories (e.g. enterprise call routing systems and toll free directory assistance) is not applicable for the automation of DA: first, for such directory volume we need to rely on automated methods both for grammar and prompt generation, and second, the use of simple–structured grammars that contain a unique branch for every listing results in prohibitively large grammars.

Below we describe three major tasks that need to be addressed in order to automate a large directory: 1) generation of *nominal* listing name to be used by speech synthesis for confirmation and readout; 2) generation of language models that enable recognition of different variants of biz/gov listings and the related directory search that retrieves the potential listing form the directory given the results of the ASR; and 3) the listings disambiguation strategies.

1. Nominal Listing Name Generation

The following are some of the categories of idiosyncrasies of the data that need to be addressed in order to create meaningful listing name to be used for prompt generation:

Business Listings are Optimized for Visual Presentation. Entries in the directory are often maintained in a format that is optimal for visual scanning but not for prompt generation; for example the listing could be for Parker, Jeffrey H. MD, while a better name to read out for such listing would be Doctor Jeffrey Parker. Similarly, 'The Henry Hudson Company' sounds better than 'HUDSON HENRY COM THE', the name this business is listed under.

Business Listings Contain Many Abbreviations. Some of the abbreviations in the directory are 'universal', such as RN for Registered Nurse, LLC for Limited Liability Company or CPA for Certified Public Accountant. Others are locality or context specific. For example, most medical doctors have MD following their names, but MD stands for Maryland in many listings in this state.

Business Listings are Entered With Different Names for the Same Entity. Entries in the directory vary in the way that they refer to the same entity. For example, a popular restaurant chain may be listed as 'McDonalds,' 'McDonalds Restaurant,' 'McDonald Family Restaurant,' 'McDonald Restaurant number one thousand seven hundred and fifty three' etc. To the caller who is asking for McDonald Restaurant, these are differences without distinction.

Business and Government Listings are Oftentimes Nested. Known in the industry as a 'caption set' numerous entities have multiple listing associated with them, each appearing as a 'subheading' in the data structure. These subheadings can go fairly deep, and it is not self-evident what piece of information is most relevant to the user request. It would not be unusual to have the entry for The United States Government, and then in the subheading include more specific information such as US Army, Recruiting Station. Clearly, it is not sufficient to take the top-level entry and use it for automation.

Businesses Use Invented Words. Companies are rather creative in coming up with names. While some (Xerox, refrigerator) make their way into the dictionary as verbs and nouns, many, such as Verizon, Accenture, Kwik Kopy or FedEx, remain without a phonetic representation in the standard dictionary.

The Solution: Automated Data Cleaning. Some of the issues above can be addressed by a data preprocessor [6] that can employ a set of 'cleaning rules' that will determine when and how to rearrange data elements in the listing? Which abbreviations should be dropped off, and which ones need to be expanded and how? What are the common misspellings for a given word? What is the most appropriate way to represent a business that has multiple names; and what is the best way to support a caption set? Important issue in the design of the pre-processor is its efficiency, since the task of data preprocessing needs to be repeated each time that new listings are entered into the directory.

2. Separation of ASR and Directory Search.

In the traditional approach the grammar contains a unique branch for each listing that compiles all the linguistic representations of the corresponding listing, and has a reference to the listing id in the directory. Therefore, during recognition, full listing information can be found trivially, without a separate directory search. However, for the

directory size of US biz/gov directory, this simple approach would result in prohibitively large grammars. An alternative approach is to separate speech recognition from the directory search, using a compact stochastic language model to cover listing names. However, since in a compact language model the association between the recognized utterance and a listing ID is lost, a separate search component is needed that takes the results obtained from ASR and outputs the listings that have high similarity with the recognized utterance[7]. These two related tasks are described below.

Generation of Language Models. The accepted methodology for stochastic language models generation is to collect a transcribed corpus of spoken utterances that represents most of the ways people can ask for a listing. However, due to the size of the listing directory, this traditional approach is clearly unfeasible. An alternative is to deploy a *variation model* that for each listing in the directory outputs a set of possible ways the user may refer to it and thus creating a 'pseudo-corpus' of automatically generated linguistic representations of the listings. This variation model can be based on hand-crafted or data-driven rules, provided a corpus of nominal listing names and the way people refer to them. Formally, variation model can be described by a conditional probability distribution *P(W|L)* that quantifies the probability that a user will refer to listing *L* will by a sequence of words *W*. In this task of creation of pseudo corpus variation model is used in its generative mode, i.e, for each listing *L* is generates a finite set of most probable linguistic representations *W* of the listing. After the pseudo-corpus had been generated classical methods can be applied to estimate the parameters of stochastic language model such as N-gram, with one modification: In the pseudo corpus, each listing variation *W* for each listing *L* has a probability attributed to it, *P(W,L) = P(W|L) P(L)*, where *P(W|L)* is the probability estimated by the variation model and *P(L)* is the prior probability of the listing. To incorporate this probability into language model estimation each n-gram in the pseudo corpus is given a weight that is proportional to *P(W,L)* and the its frequency is estimated by summing the weights of all such n-grams in the pseudo-corpus.

Directory Search. The directory search component of the system takes the speech recognition results, either in the form of N-Best or a lattice) and outputs an ordered list of closely matching listing. Ultimately, directory search task can be formalized as ranking each listing in the database according to *P(L|O)* where *O* are the observations representing the acoustic signal of the users utterance describing the listing name. Practically we will approximate this similarity measure as *P(L|R)*, where *R* is the result of speech recognition, comprising a finite set of strings W with their corresponding probability *P(W|O)*, P

$$P(L \mid R) = \sum_{w \in R} P(L \mid W) P(W \mid O)$$, where *P(L|W)* is related to the variation model *P(W|L)* by *P(L|W)=P(W|L)P(L)/P(W)*. This relationship is not a coincidence, since in generative mode variation model performs a function that is dual to that of the search: variation model outputs the most probable linguistic realization of a listing, while the search outputs the most probably listing for a given linguistic realization.

Another important issue to watch for in the design of the search is the efficiency of the search algorithms. The directory to be searched is large, and the search happens on-line, as the caller is waiting for system response, making linear algorithms whose performance scales with the number of listings a poor choice. A possible way to address this problem is to index the directory, use the index to create a list of possibly matching listings, and then use the similarity measure conditional *P(L|R)* to re-rank the candidate listings.

3. Business Listings Disambiguation

Essentially, the primary role of the Operator, in most instances, is to serve as a relay between the caller and the system. However when the system returns multiple candidate matches to the user's request, the Operator plays a more active role in matching one of those candidates to the caller's request. Of course, when automation is introduced this intelligence has to be built into the automated system. In the case of disambiguation of business listings the following instances are of particular interest.

Business listings often have multiple locations. Some business entities have multiple locations within a locality. Thus finding out that the user is looking for Starbucks may be only an intermediate step toward identifying the desired telephone number. A good dialogue design needs to consider the number of locations that are associated with this listing.

- If there are only two branches, the dialogue may let the user choose between them, or even present both of them and save a turn in the dialogue.
- If there are more than two, but still a manageable number, the system may present the various addresses to the caller and let them choose the one that they want.
- If there are so many branches that presenting a list is no longer feasible, the system may wish to ask the caller for the address that the business is on.

Sometimes business listings have similar names. Many businesses sound a lot alike. The word American is used very frequently as well as the word First. So it is not impossible that a locality would have the First American Bank and the First American Trust Company. The user may say First American, and now the onus is on the system to advance the dialogue in a way that would help the user get to the right listing. It is relatively simple when two names are confusable. It is a lot harder if the number of candidates is higher.

2.4. Handling Residential Listings.

In some respects, structuring a dialogue for residential request is easier. There are more options that make sense in this context. For example, asking to say and spell the person last name (Smyth, s m y t h) [8] is a reasonable request in this context, while asking to spell the first word in a business name may be less logical (e.g. the Boston globe, t h e.) Yet the reality is that automating residential requests is fraught with its own set of problems

The first decision the application designer is facing is in the context of residential request is whether to prompt for a listing name or for a last name and first name separately.

1. Prompting for a listing name.

A general prompt like "what listing" in the context of residential listing request may elicit a variety of caller responses, including first name and last name "Bob Schwartz" or "Robert Schwartz", last name only "Schwartz", or "Doctor Schwartz", or more that one name as in "John and Emily Schwartz. The problem is that the information in the directory for residential listing may be quite different from

the caller's request. For example, the caller may be looking for Bob Schwartz, but Bob's number is listed under his spouse name, Emily, as Robert Schwartz, or under R. and L. Schwartz. Clearly, first name information elicited from the caller is not very reliable for listing identification. Therefore, if we compile a grammar for full listing name, we cannot expect the recognition to be very accurate. Moreover, building a grammar for full listing name will result in unfeasibly large grammars: there are a total of 100, 000,000 listings, and the size of the grammar in this case will scale with the number of listings.

2. Prompting for a last name and first name separately.

A possible solution to the problem of unreliable first name information is to base the search on last name only, using the first name for disambiguation if necessary. This strategy also helps to cap the size of the grammars needed for good coverage. The distribution of last names is generally quite peaked. Usually, a few tens of thousands most frequent last names are sufficient to provide a good coverage of listings even for the largest cities. The issues to consider when prompting for last name are:

There are Many Unique Names. The ethnical diversity of the US is evident in the number of unique names. There are 1.1 million unique last names in the US[1]. Many of them, such as Xoumphayvient, appear rather infrequently (once) and are hard to pronounce. Thus, for a good portion of the requests there will be no entry in the standard dictionary. The system will not be able to recognize the name, and if by some chance it recognizes it, the system will have a hard time pronouncing the name in a fashion that would allow the caller to understand the name.

Some names are extremely popular. If the caller is asking for John Smith in a community that is relatively large, the search is likely to return too many candidates for a practical search. The name Smith accounts for 1% of the US population, and the top 11 last names cover 5% of the US population. While a question about the street address may reduce the set, it stands to reason that if they don't know the phone number of the person, people would not know what street the person lives on.

Some names are homonyms. One of the difficulties in using last name as the search string in a data search is that there are many last names that are pronounced the same but are spelled differently (e.g. Lee and Li, Manheim and Mannheim). A possible strategy for dealing with the situation is to confirm one of the alternative pronunciations, but then continue the search with the assumption that ALL alternate spelling have been confirmed, and trying to match additional information such as first name, to select among all of them.

Some names are near-homonyms. The case of near-homonyms is far more challenging. Names like Barker and Parker, or Andersen and Anderson are easily confusable. If the system is to ask the user to confirm "that's Anderson, right?" it is most likely that the answer will be yes even when the request was for Andersen. Now, the search would continue without a glimmer of a hope of getting to the right listing. It is possible to skip the name confirmation and ask for first name in the hope that this would provide the clue, to keep all near-homonyms alternative viable while continuing the dialogue, or even to engage the user in disambiguation by providing the spelling of the name. However no solution is able to fully address this problem.

Many names have more than one pronunciation. The US being a melting pot is reflected in the names of its residents. Some people adhere to the pronunciation that is used in their country of origin; others abandon it in favor of some Americanized version; either way, other people may further distort the 'foreign' name Thus, many names have multiple pronunciations, with none of them being more 'correct' than the others. For example, Chen is both a Hebrew and a Chinese last name, with very different pronunciation for the two variations. It is easier to compensate for this during the recognition task (by including both pronunciations) than it is to ensure that the system is delivering the right pronunciation during confirmation or disambiguation. It would not be unusual for a caller not to recognize that the system has actually captured the right name when the system reads the name back to the caller.

3. Summary

In this paper we illustrated how a careful consideration of business drivers, user needs and technology constraints along with domain knowledge serve as a guide in the design of a Voice User Interface for Directory Assistance. It is our belief that to be successful the system designer must balance these considerations.

4. References

[1] Kathleen Pierz. " Consumers & Internet-based Telephone Number Lookups: A National Consumer Research Study", Pierz Group Report, October 6, 2004,

[2] Mane, A. M., Maintaining and Increasing Customer Satisfaction Through Voice Automation of Directory Assistance, *Speech Technology magazine,* May 2003.

[3] Olasson, D., Ju, Y.C., Bhatia, S., Herron, D., Liu, J., "MS Connect: a fully featured auto-attendant. System Design, Implementation and Performance", Proc ICSLP 2004, in *Proc. Int. Conf. on Spoken Language Processing*. Jeju, South Korea, Oct, 2004.

[4] A. Abella et al, "VPQ: A Spoken Language Interface to Large Scale Directory Information," in *ICSLP 98*, (Sydney, Australia),pp. 2863–2867, 1998, November 30 - December 4, 1998.

[5] Seide, F. and Kellner, A. "Towards an automated directory information system." *Proc. EuroSpeech '97*, 1327-1330, 1997.

[6] Spiegel, M. F. and Machi, M. J, 1990. Synthesis of names by a demi-syllable-based speech synthesizer (Orator). *Journal of the American Voice Input/Output Society* 7:1--10.

[7] Natarajan, P., Prasad, R., Schwartz, R., Makhoul, J. "A Scalable Architecture for Directory Assistance Automation", *ICASSP 2002*, Orlando, Florida, May 2002.

[8] Michael Meyer and Hermann Hild. Recognition of spoken and spelled proper names. *EUROSPEECH, vol. 3, pages 1579-1582*, Rhodes, Greece, September 1997.

[9] H. Schramm, B. Rueber, and A. Kellner. "Strategies for name recognition in automatic directory assistance systems." *Speech Communication*, 31: pp 329-338, 2000

[10] Spiegel, M.F. "The Difficulties with Names; Overcoming Barriers to Personal Voice Services" Speech Technology Magazine, May / June 2003