© IMAGESTATE

[Ye-Yi Wang, Dong Yu, Yun-Cheng Ju, and Alex Acero]

An Introduction to Voice Search

[A look at the technology, the technological challenges, and the solutions]

Voice search is the technology underlying many spoken dialog systems (SDSs) that provide users with the information they request with a spoken query. The information normally exists in a large database, and the query has to be compared with a field in the database to obtain the relevant information. The contents of the field, such as business or product names, are often unstructured text. For example, directory assistance (DA) [1] is one of the most popular voice search applications, in which users issue a spoken query and an automated system returns the phone number and address information of a business or an individual. Other voice search applications include music/video management [2], business and product reviews [3], stock price quotes, and conference information systems [4], [5].

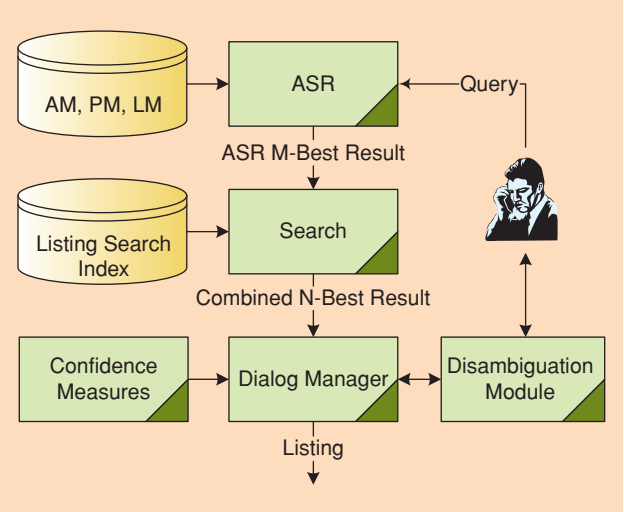
Figure 1 shows the typical architecture of a voice search system, where a user's utterance is first recognized with an automatic speech recognizer (ASR) that utilizes an acoustic model (AM), pronunciation model (PM), and language model (LM). The m-best results from the ASR are passed to a search component to obtain the n-best semantic interpretations; i.e., a list of up to n entries in the database. The interpretations are passed to a dialog manager (DM) subsequently. The DM utilizes confidence measures, which indicate the certainty of the interpretations, to decide how to present the n-best results. If the system has high confidence on a few entries, it directly presents them to the user. Otherwise, a disambiguation module is exploited to interact with the user to understand what he actually needs.

VOICE SEARCH AND OTHER SPOKEN DIALOG TECHNOLOGIES

SDSs are often chronologically categorized into three generations: informational, transactional, and problem solving [6], [7] (earlier command and control speech applications in the 1980s are not considered as SDSs in this categorization). The first-generation SDSs focus on providing users with the information they request, such as flight status and weather information. The second-generation SDSs conduct transactions automatically with users; e.g., to book air flight tickets or perform bank balance transfers. The third-generation SDSs are often used in customer support by interacting with callers to diagnose the problems they are experiencing with a device or a service.

Digital Object Identifier 10.1109/MSP.2008.918411

1053-5888/08/$25.00©2008IEEE



[FIG1] **Voice search system architecture.**

The chronological (functional) categorization of SDSs does not necessarily imply the level of technological difficulties. Some of the problems in informational SDSs remain the most challenging topics in spoken dialog research. To better understand different technological challenges, SDSs can be categorized technologically into three categories: form filling, call routing, and voice search. Form filling is the most commonly used technology deployed in the first- and the second-generation SDSs, where directed or mixed-initiative dialog systems are used to gather the attribute values of an entity that users are interested in (e.g., the originating and destination cities of a flight). In such systems, users often have to use canned expressions within a small domain. In a directed-dialog system, users' utterances may contain only what the system has prompted for, which is often a single piece of semantic information; while in a mixed-initiative system, users may volunteer more semantic information in a single utterance—we call this type of semantic understanding *high resolution* in the sense that multiple semantic constituents (commonly called "slots") need to be identified. The call-routing applications remove the constraints on what users can say, so users can speak naturally. This is accomplished at the expense of limiting the target semantic space: the understanding of natural language inputs is often achieved with statistical classifiers, which map users' inputs to a list of possible destination classes (intents). The classifiers can hardly perform high-resolution understanding with many slots, or scale up with a huge number (e.g., thousands to millions) of destination classes. Voice search applications differ from the form-filling applications in their lack of detailed, high-resolution semantic analysis. They are similar to call-routing applications with respect to the naturalness of user inputs and the huge input space. However, they differ from call-routing applications in the sense that their semantic space, or in the terminology of call-routing systems, the inventory of the "destination classes" is enormous—sometimes in the range of millions of entries. Data are seldom sufficient to train a statistical classifier. Table 1 compares the three types of technologies.

The form-filling and call-routing spoken dialog technologies have been discussed in great detail in [8] and [9]. This article reviews the voice search technology—we will only focus on the search from a field of unstructured text items. The issues related to the search of other media (e.g., audio and video search), including recognition and indexing, is beyond the scope of this article. We will describe the history of the voice search technology, discuss the technological challenges, and survey the solutions to these challenges.

HISTORY

Early work on voice search focused on DA. Institutions on both sides of the Atlantic deployed experimental systems during mid to late 1990s. The early studies focused mainly on residential DA [10]–[12], and speech recognition was the major topic of research—as long as personal names get correctly recognized, the search can be a simple database lookup. As a result, the dialog strategies centered on limiting the scope (hence perplexity) of the target listing space for ASR and the confidence measures mostly relied on features from the ASR. Related work includes enterprise-level auto-attendant (also known as name dialing) services from Phonetic Systems (acquired by ScanSoft, then merged with Nuance), AT&T [13], IBM [14], and Microsoft [15]. While automating residential DA is important in reducing the operational cost, it is only a small portion (19%) of the total received calls compared to the 61% of business DA calls [12]. Therefore, there have been increasing interests in business DA recently, with the commercial deployments from Tellme (acquired by Microsoft), Jingle Networks, AT&T, Google, Verizon, and Cingular (merged with AT&T Wireless), and an experimental system from Microsoft [1]. Because the level of linguistic variance is much higher in business DA queries, spoken language understanding (SLU)/search aiming at correctly interpreting a user's intent becomes an important research topic. The linguistic variance increases the ambiguity and uncertainty in the interpretation of a user's intent. As a result, dialog research focuses on the disambiguation strategy as well as the confidence measures that look into features from different system components to accurately predict the end-to-end performance of interpreting a spoken query.

[TABLE 1] COMPARING FORM-FILLING, CALL-ROUTING, AND VOICE SEARCH TECHNOLOGIES.

	USER INPUT UTTERANCES		TARGET SEMANTIC REPRESENTATION	
	NATURALNESS	INPUT SPACE	RESOLUTION	SEMANTIC SPACE
FORM FILLING/DIRECTED DIALOG	LOW	SMALL	LOW	SMALL
FORM FILLING/MIXED-INITIATIVE	LOW-MEDIUM	SMALL	HIGH	SMALL
CALL ROUTING	HIGH	LARGE	LOW	SMALL
VOICE SEARCH	MEDIUM-HIGH	LARGE	LOW	MEDIUM-LARGE

Other voice search applications include the stock quote system from Tellme and a product/business rating system from Microsoft [3]. Separate efforts have been made on conference information systems by Carnegie Mellon University [4] and by the collaboration among AT&T, ICSI, Edinburgh University, and Speech Village [5], where users can request information about thousands of papers published through conferences. In entertainment, Daimler is investigating digital music management in automobiles [2]. Like the business DA applications, all these new voice search applications call for research activities in search/SLU and dialog management in addition to speech recognition.

With the broad adoption of mobile devices and the availability of wireless access to the Internet, many companies are actively engaged in the space of voice search on mobile or in-car devices [2], [16]. New research challenges include multimodal [graphical user interface (GUI) with touch screen and speech] user interfaces [2], [16] and efficient and scalable client-server architectures.

TECHNOLOGICAL CHALLENGES

Voice search poses new challenges to the spoken dialog technology in the following areas.

- *Speech Recognition:* The state-of-the-art ASR systems have high error rates on voice search tasks. The vocabulary size of a voice search system can be much larger than a typical form-filling or a call-routing application, sometimes reaching millions of lexical entries. Many lexical entries in international individual/business names are out of vocabulary and lack reliable pronunciation information. Calls are often made from different noisy environments. In addition, the constraints from language models are often weaker than other ASR tasks—the perplexity of a language model is often high (e.g., 400–500 bits for business DA) for voice search.
- *Spoken Language Understanding/Search:* One big problem in SLU is the enormous semantic space—a DA system can easily contain hundreds of thousands (if not millions) of listings in a city. There is also a high level of linguistic variance in the input space. For example, users might not use the official name of a business in a DA or business rating system. They would typically say, for instance, Sears instead of the listed official name, Sears Roebuck & Co. In addition, the SLU/search component must be robust to ASR errors.
- *Dialog Management:* The difficulties in ASR and SLU cause much confusion and uncertainty. The dialog manager has to effectively narrow down the scope of what a user may say to reduce the confusability and uncertainty. Search results often contain multiple entries. Disambiguation strategy is crucial in obtaining sufficient information for the correct understanding of users' intents with as few dialog turns as possible. Confidence measures are important for the dialog manager to take an appropriate action with each of the hypothesized interpretations, such that the dialog can recover gracefully from ASR and SLU errors.
- *Feedback Loop:* No systems can be perfectly built at the initial deployment. Dialog system tuning is often performed painstakingly by spoken dialog experts, starting from error analysis from the logged interaction data to find the flaws in dialog and prompt design, grammar coverage, system implementation, etc. An interesting research topic is the automatic or semi-automatic discovery and remedy of design/implementation flaws.

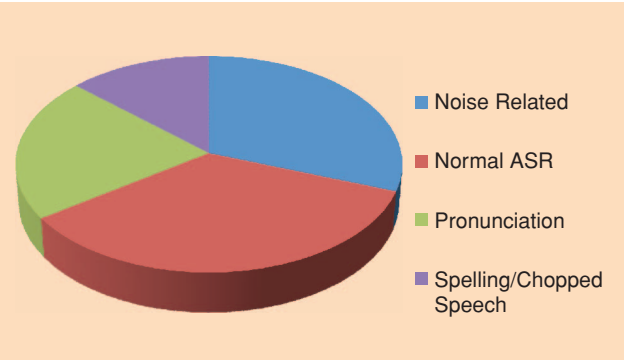
The grand challenge in voice search application is robustness. The CSELT's study on Telecom Italia's DA system [12] showed that even though the automation rate was 92% in a laboratory study, the actual field trial automation rate was only 30% due to unexpected behavior of novice SDS users and environment noise.

TECHNOLOGY REVIEW

This section reviews the technology that addresses the challenges to voice search applications. Not surprisingly, much of the technology is developed with DA systems because they are the most popular voice search applications so far. However, the technology is often applicable to other applications as well. For example, the product/business rating systems [3] directly used the technology developed in a DA application [1].

SPEECH RECOGNITION

A detailed error analysis for proper name recognition was reported in an auto-attendant system [14]. Figure 2 shows the distribution of different causes of errors. Besides 35% of normal recognition errors, 31% were noise related and 22% were pronunciation related. Many of the calls were made in a noisy environment over different noise channels. Therefore, noise robustness is crucial to improve the ASR accuracy. On the other hand, there were many foreign names that are difficult to pronounce in an auto-attendant/DA system. In fact, pronunciation is a pervasive problem that poses challenge in many other voice search applications too. For example, users may specify *Petit Bonheur by Salvatore Adamo* in a music search. Hence, pronunciation modeling is another important topic in ASR for voice search. In addition, better acoustic and language models are always important to reduce the ASR error rate.



[FIG2] ASR error analysis for a voice search application.

ACOUSTIC MODELING

IBM's auto-attendant system applied speaker clustering in its acoustic model [14]. Simple human Markov models (HMMs) that have one Gaussian per context-independent phone state were trained first for each speaker. Then the vectors of the means of these models were clustered with the k-means algorithm. For each test utterance, the cluster model that yielded the highest likelihood was selected. In doing so, different channel and noise conditions can be more precisely modeled by different cluster models, so noise-related problems are alleviated. In addition to speaker clustering, speaker adaptation is effective to bring the performance of a speaker-independent system closer to that of a speaker-dependent system. Unlike normal speaker adaptation, the adaptation in [14] was massive in the sense that the adaptation data were obtained from a pool of recent callers rather than a single speaker. The massive adaptation is helpful due to the fact that a caller often calls the same set of individuals, and that a caller may try a name repeatedly when a recognition error occurs. While massive adaptation is helpful to bring down the error rate for frequent callers, unsupervised utterance adaptation aims at improving the accuracy from an unknown speaker. In this adaptation scheme, the test utterance itself was used for adaptation with a two-pass decoding. In the first pass, a speaker-independent system or the system after massive adaptation was used to obtain the automatic transcript. Then a forward-backward algorithm was applied to obtain the adaptation statistics. After adapting the acoustic models using the collected statistics, the caller's utterance was decoded in a second pass with the adapted model—this second pass may adversely increase the latency of a voice search system. Overall, with all these acoustic model enhancements and an unsupervised derivation of pronunciations (to be described below), a 28% error reduction was observed.

SPOKEN DIALOG SYSTEMS ARE OFTEN CHRONOLOGICALLY CATEGORIZED INTO THREE GENERATIONS: INFORMATIONAL, TRANSACTIONAL, AND PROBLEM SOLVING.

PRONUNCIATION MODELING

One approach to an improved pronunciation model is via augmenting the dictionary with pronunciation variants. Data-driven algorithms are commonly applied, which typically include four steps: generating phonetic transcriptions with a recognizer, aligning the auto transcriptions with manually created canonical pronunciations, deriving rules mapping from canonical pronunciations to the variants, and pruning the rules. One limitation of this approach is that the canonical reference pronunciations must be available.

The IBM auto-attendant system [14] adopted an acoustics-only-based pronunciation generation approach [17]. The advantage of this approach is that no canonical pronunciation is required. This makes it more practical in voice search applications since many words do not exist in a pronunciation dictionary. With this approach, a trellis of subphone units was constructed from an utterance. The transition probabilities in the trellis were derived by weighting the transition probabilities of all the context-dependent realizations of the subphone units in an HMM acoustic model. A Viterbi search was performed to obtain the best subphone sequences from the trellis and a pronunciation was subsequently derived from the sequence. Experiments in [14] showed a 17% relative error reduction when the test set and training set had overlapping unseen words.

Trade-offs often have to be made in adding pronunciation variants to a dictionary. The additional pronunciations, on the one hand, make the word models match the actual acoustic signal more precisely; on the other hand, they give rise to a large number of highly confusable word models. Instead of augmenting an existing pronunciation dictionary with variants, a pronunciation distortion model was introduced in [18] to rescore the n-best hypotheses generated from a first recognition pass. The distortion model incorporates the "knowledge source" about the common distortions observed in a specific spoken language. For example, only insertions were considered in the distortion model for French in [18] because it is frequently observed that silence segments are often inserted between certain pairs of consonants like [m][n], and a schwa is often inserted after a consonant at the end of an utterance. Formally, let A and W denote the acoustic signal and text of a caller's utterance, and τ_w a phone sequence that may be distorted from the canonical pronunciation of W. Then a hypothesis $\widehat{W}$ can be selected from the first-pass n-best recognitions according to the following decision rule:

$$\begin{aligned}\widehat{W} &= \operatorname*{argmax}_{W} p(W|A) \\ &= \operatorname*{argmax}_{W} \sum_{\tau_w} p(W, \tau_w|A) \\ &\approx \operatorname*{argmax}_{W,\tau_w} p(W, \tau_w|A) \\ &\approx \operatorname*{argmax}_{W,\tau_w} p(\tau_w)p(A|\tau_w)p(W|\tau_w).\end{aligned}$$

The last approximation in the equation includes an application of the Bayes' rule and an assumption of independency between A and W given τ_w. The prior of a distorted phone sequence, τ_w, can be written in terms of the canonical pronunciation of W, η_w, and δ_w, the difference between τ_w and η_w: $p(\tau_w) = p(\eta_w\,\delta_w) = p(\delta_w|\eta_w)p(\eta_w)$ and δ_w. In theory, $p(\delta_w|\eta_w)$ can be estimated from data. In [18], a uniform distribution over all plausible insertions was used instead for $p(\delta_w|\eta_w)$ due to the lack of data. $p(A|\tau_w)$ in the decision rule can be obtained from the acoustic model with all possible alignments between A and τ_w. Since only insertion is considered in [18], $p(W|\tau_w)$ was obtained by multiplying the probabilities of all successful insertions. Experiment results showed that the rescoring had improved the one-best accuracy from 50% to 59.8%.

LANGUAGE MODELING

Early DA systems compiled directory entries into a finite-state grammar as the language model for ASR. This rule-based language model does not scale up well with directory size due to increased perplexity. It was found that the ASR accuracy decreases linearly with logarithmic increases in directory size [10]. On the other hand, it was noticed that the distribution of the requested listings followed the Zipf's law: 10% (20%) of call volumes were covered by only 245 (870) listings. So in [10], [19], a semi-automated DA system was built that only covered the frequently requested listings and relayed the remaining requests to human operators.

THE VOCABULARY SIZE OF A VOICE SEARCH SYSTEM CAN BE MUCH LARGER THAN A TYPICAL FORM-FILLING OR A CALL-ROUTING APPLICATION, SOMETIMES REACHING MILLIONS OF LEXICAL ENTRIES.

One problem of the rule-based LMs constructed from database listings is their poor coverage. Callers seldom say a business name exactly as it appears in the database—just consider the earlier example of *Sears Roebuck & Co.* versus *Sears*. It was mentioned in [20] that variant expressions for business names could be semi-automatically derived from data. Although it did not report how this was achieved. A straightforward method would compare a caller's utterance (e.g., *Kung-Ho Chinese Restaurant*) with the actual listing released to the caller (e.g., *Kung-Ho Cuisine of China*) by operators and learn that "Chinese restaurant" is a synonym of "cuisine of China." This synonym rule-based approach is usually expensive; the rule coverage is highly restricted by the data available, and the rules may be over-generalized without careful crafting.

The problem was tackled without using the data from callers in [21]. A method was proposed to automatically construct a finite-state signature LM from a business directory database alone, which would accept different query variants. Here a signature is a subsequence of the words in a listing that uniquely identifies the listing. For example, with the listings "3-L Hair World on North 3rd Street" and "Suzie's Hair World on Main Street," "3-L," "Hair 3rd," and "Hair Main" are signatures because they occur in only one listing. On the contrary, the subsequences "Hair World" and "World on" are not signatures because they appear in both listings. Based on the signatures, a finite state transducer can be constructed as follows (the example is taken from [21]):

$<S>$:= *3-L Hair World? On? North? 3rd ? Street? : 1* |
3-L Hair? World? On? North 3rd ? Street? : 1 |
3-L Hair? World on? North? 3rd ? Street? : 1 |
3-L? Hair World? On? North 3rd ? Street? : 1 |
Suzie's? Hair World? On? Main Street? : 2 |
Suzie's Hair World? On? Main? Street? :2 |
Suzie's Hair? World on? Main? Street? :2 |
Suzie's? Hair? World on? Main Street? :2

where each entry in the grammar corresponds to a signature. The terms in a signature are obligatory whereas the terms in a listing but not in the signature are optional (marked by "?"). The numbers after ":" is the semantic output from the transducer that represents the ID of a listing in the database. In doing so, every utterance matched by a rule can be uniquely associated with a listing. Because the nonessential words are optional, this makes the grammar more robust to utterances that omit these words. When the directory becomes larger, an entry may bear no signature because each of its subsequences can be a subsequence of another entry. This problem was handled with confusion sets in [21].

The rationale behind the signature grammar is that any term in an entry is droppable as long as the drop does not cause the confusion with another entry. While this is very practical in reducing the search ambiguity, it may be risky in modeling human language—speakers are very likely to drop terms that would lead to ambiguity. For example, they often say *Calabria* instead of *Calabria Restaurant* even though the former may cause confusion with "Calabria Electric" and "Calabria Jack J Do."

Another approach to improved robustness is via statistical n-gram models [1], [19]. An n-gram model is more robust because it does not require a user's utterance to match a rule exactly, because it provides a statistical framework for fair comparison between different hypotheses, and because it has well-studied smoothing algorithms to estimate the likelihood of unseen events more accurately. Ideally, a statistical n-gram model should be built from the transcripts of real calls, which demonstrate not only the different ways callers refer to businesses but also the probability of each such ways. Unfortunately, it is not realistic to collect enough calls to provide a good coverage for a large listing set, especially during the early stage of development. An interpolated LM was proposed to estimate the n-gram probability in [1]: $p(w) = \lambda p_\tau(w) + (1-\lambda)p_l(w)$, where $p_\tau(w)$ is the LM built using the transcripts of real calls, $p_\tau(w)$ is the LM built using a listing database, and λ is the interpolation weight, which was tuned with a cross-validation set collected under real usage scenario. Here $p_\tau(w)$ can be constructed from data straightforwardly. Building $p_l(w)$, on the other hand, takes more considerations because the database entries may not reflect the actual ways that callers refer to them. A statistical variation model was introduced to account for the common differences between database listings and the actual callers' queries. The model was based on the rationale similar to that of the signature model; namely, callers are more likely to say the words that distinguish one listing from others. However, instead of making risky binary decisions, it modeled the importance of a word statistically according to its discriminative capability and its positions in a listing (based on the observations that callers are more likely to say the initial words in a listing). Here, the discriminative capability of a word was determined by its inverse document frequency, and a position importance weight $w_l^i (0 < w_l^i \leq 1)$ was associated to each word position. A word was droppable with a probability inversely proportional to its importance. In addition, the model took into account the business

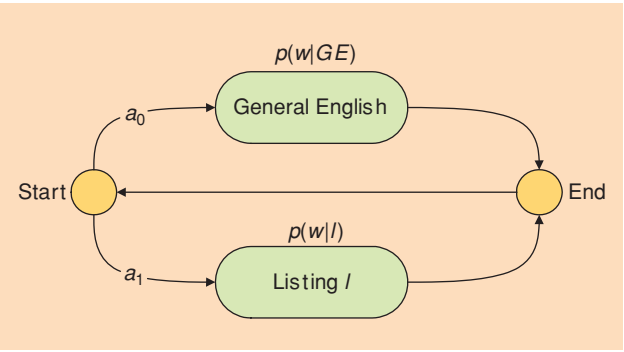
category information for smoothing—each word had a probability to transition to category words (e.g., "restaurant"). The transition probability correlated to the importance and the category-indication capability (a mutual-information-based measure) of a word. Furthermore, an efficient interpolation with a large vocabulary background LM [22] had provided additional robustness. The internal investigation in Microsoft has revealed that the statistical language model, together with the vector space model for listing search, has greatly outperformed the signature-based approach—at the same precision level, the recall has been almost doubled.

SPOKEN LANGUAGE UNDERSTANDING/SEARCH

The task of SLU is to map a user's utterance to the corresponding semantics. In voice search, the semantics is the intended entry in a database. Hence, the SLU becomes a search problem.

In early voice search applications like residential DA, SLU is not an issue, since there is not much expressional variance in saying a person's name. Search is basically a database lookup, with careful considerations of initials, titles, and homophones. If a finite-state-based LM is used for ASR, each rule is uniquely associated with a listing or a confusion set. There is no need of a separate search component either. However, due to the deficiency of the finite-state-based LMs in modeling the actual human language, n-gram

DISAMBIGUATION STRATEGY IS CRUCIAL IN OBTAINING SUFFICIENT INFORMATION FOR THE CORRECT UNDERSTANDING OF USERS' INTENTS WITH AS FEW DIALOG TURNS AS POSSIBLE.



[FIG3] HMMs for listing search.

LMs are often adopted in advanced voice search applications. In such cases recognitions are no longer associated with a specific database listing. Hence, a separate search step is necessary. Here robustness is again a crucial issue—the search algorithm should be robust to not only linguistic variance but also recognition errors. Statistical models were proposed for solutions.

BBN adopted a channel model for listing search [19]. Given a locality C (DA dialogs often start by asking users for the city and state information; see the "Dialog Management" section for details) and a query Q recognized from a user's utterance, it looks for the listing $\hat{L}$ according to the following decision rule:

$$\begin{aligned}\hat{L} &= \operatorname*{argmax}_{l} p(L|C,Q) \\ &= \operatorname*{argmax}_{l} p(C,Q|L)p(L) \\ &\approx \operatorname*{argmax}_{l} p(C|L)(Q|L)p(L).\end{aligned}$$

In [19], the prior distribution $p(L)$ and the locality distribution $p(C|L)$ were estimated from training data. The training data were the transcripts of real users' utterances augmented with database listings. The query distributions $p(Q|L)$ were modeled with a two-state HMM illustrated by Figure 3. In this model, a word w in Q is generated from either the general English (GE) state or the state corresponding to a listing l, which is a value of the random variable L. With this model, $p(Q|L) = \prod_{w\in Q}(a_0 p(w|GE) + a_1 p(w|L))$. Here the transition weights a_0 and a_1 were tied across the HMMs for all values of L. The transition and emission probabilities were estimated from training data. This model is robust due to the inclusion of the GE state, which captures filler phrases like *I need the number of* or ASR errors. The combination of real user data and the database listings facilitates high accuracy on frequently requested listings and simultaneously enables broad coverage of less frequently requested listings.

Microsoft Research applied a term frequency-inverse document frequency (TF-IDF) weighted vector space model (VSM) for business listing and product name search. The VSM is widely used for informational retrieval (IR). The standard VSM has been enhanced for voice search in [1]. The first enhancement regards the duplicate words in listings and queries. In traditional IR, documents and queries are generally long. The term frequency resembles the true distribution underlying a document/query. Listings and queries in voice search, on the other hand, are short in general, so the surface term frequency may not be a reliable estimate of the true underlying distribution. A small noise is more likely to bring different search results. For example, the query *Big 5*, intended for "Big 5 Sporting Goods," results in the listing "5 star 5"—the additional "5" in the listing brings it closer to the query. Since the term frequency is not reliable for search among short listings, each term gets a unit count in voice search. A duplicate word is treated as a different term; e.g., by replacing the second "5" in the example with 5_2nd. This effectively adds another dimension to the vector space. Since the IDF of this new dimension is much higher, it plays a more important role in query matching. A query without duplicate words like "Big 5" will have a larger angle from a listing with duplicate words. The angle will be significantly reduced if the query does contain the second term. So "5 5" will match "5 star 5" better. The second enhancement to

the standard VSM is about the use of category information. Callers often voluntarily provide category information (like restaurant, hospital, etc.) in their queries. These category words can be identified according to the mutual information between them and the categories in a database. If category information is detected in a user's query, the category information about a listing in the database can be appended to the listing's vector so it can be compared with the query's category. Or the category information of a query and a listing can be compared in a separate step. With this enhancement, the VSM would rank the listing "Calabria Ristorante Italiano" higher than "Calabria Electric" for the query *Calabria Restaurant*. The third enhancement aims at the robustness to ASR errors. Instead of using word unigrams or bigrams as terms, character n-gram unigrams or bigrams were used as terms to construct the vectors. The rationale is that the acoustically confusable words may have shared subword units. For example, the listing "Lime Wire" is rewritten as a sequence of character 4-grams—$Lim Lime ime_ me_W e_Wi _Wir Wire ire$, where "$" indicates the start and the end of the listing and "_" indicates separation of words. If a caller's query *Lime Wire* is incorrectly recognized as *Dime Wired*, there is no word overlapping but still much character n-gram overlapping between the ASR output and the intended listing.

THE RATIONALE BEHIND THE SIGNATURE GRAMMAR IS THAT ANY TERM IN AN ENTRY IS DROPPABLE AS LONG AS THE DROP DOES NOT CAUSE THE CONFUSION WITH ANOTHER ENTRY.

DIALOG MANAGEMENT

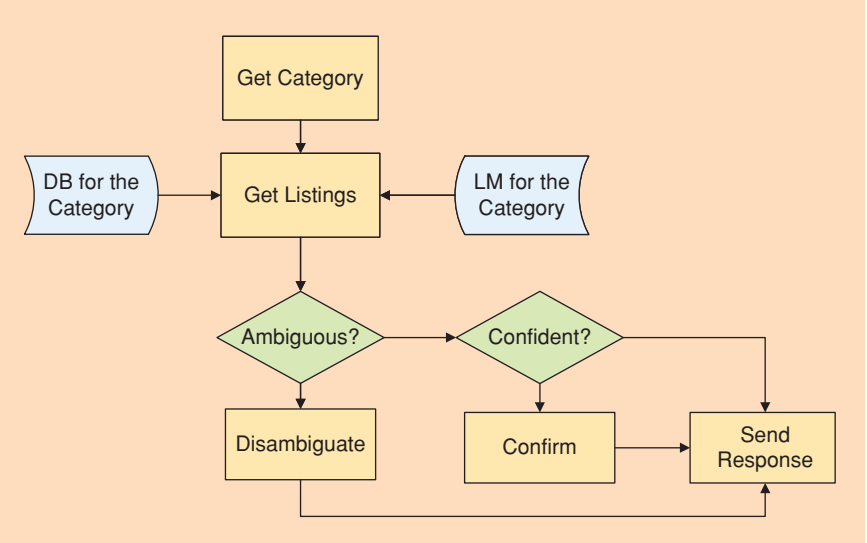
Figure 4 shows the common dialog strategy in voice search applications. The dialog starts with prompting a user for the category information about the item they are looking for to narrow down the downstream LM and search spaces. The category can be the city/state information in a DA system [1]; the business/product separation (national business, local business, or product) in a voice rating system [3], or a "search-by" attribute of the music metadata (e.g., title, album, genre, artist, etc.) in a music search dialog system [2]. A category-specific LM is subsequently used to recognize the user's query containing the listing information, and the search component looks for the listing in a category-specific database. If multiple listings are found, a disambiguation subdialog is engaged; otherwise the dialog system either directly sends the user the listing information or asks for user confirmation if the confidence score is low.

Many voice search applications use their own task-specific dialog strategies. For example, based on the finding that the accuracy on spelled names is much higher than that on spoken names, the residential DA in [23] exploited a multistage dialog strategy to improve the accuracy of proper name recognition. A listing was identified by first recognizing its spelling from a caller. The spelling word graph greatly reduced the listing space for subsequent recognition of names and addresses.

DISAMBIGUATION

Most voice search dialog systems adopt an application-specific disambiguation strategy. In residential DA, people with the same name are disambiguated with their addresses [23]. In business DA, business categories are used for disambiguation [1]: from the set of businesses returned by the search component, a list of possible categories is compiled. For example, the query *Calabria* results in multiple search results, "Calabria Ristorante Italiano" in category Restaurants, "Calabria Jack J Do" in Doctors and Clinics, and "Calabria Electric" in Electric Contractors. These categories are read to the user for selection. All the matching business names in the selected category are subsequently read to the user until one is selected or the list is exhausted. Similar disambiguation strategy is used in a multimodal voice search application [2], where multiple music titles are displayed in a GUI for users' selection when they belong to the same category, or the different categories are displayed for disambiguation first. The GUI allows users to scan the information visually, which makes the multimodal interaction more effective.

One problem of the hard-wired disambiguation strategy is its inefficiency with long category/entry lists in a speech-only interface. It has been suggested that spoken dialog strategies such as summaries are a verbal equivalent of the visual



[FIG4] Common voice search dialog strategy.

scanning behavior that makes GUIs effective [24]. Hence, summarization can be used when the search component returns a large ambiguous set. Figure 5 shows an exemplar dialog taken from [25]. Here the ambiguous listings are summarized along common attributes such as price ranges and cuisines, which guide users to provide the most effective information for disambiguation. In contrast to the hard-wired disambiguation strategy, the attributes were selected automatically by using a decision-theoretic user model and using the association rules derived from database subset in the dialog focus [25].

IT HAS BEEN SUGGESTED THAT SPOKEN DIALOG STRATEGIES SUCH AS SUMMARIES ARE A VERBAL EQUIVALENT OF THE VISUAL SCANNING BEHAVIOR THAT MAKES GUIS EFFECTIVE.

CONFIDENCE MEASURE

Confidence measures are used to determine what to do with the search results for a spoken query. The results will be played to callers if the confidences are high, otherwise a confirmation/disambiguation subdialog will be invoked. Confidence measures are also used to determine when to elevate an automated service conversation to a live agent in an early dialog stage if the confidence on the key information (e.g., an individual's last name in a residential DA system) is too low [23].

ASR-only confidence measures were used in many early residential DA systems because search was not a significant source of uncertainty. A well-studied confidence measure is the word or sentence posterior probability that can be calculated from an ASR lattice, which was shown to be more effective than some other heuristics [26]. A sentence posterior probability obtained from an n-best list was used in [23] for DA. Another confidence measure originally proposed for utterance verification [27] was applied in [20]. It is based on hypothesis testing that leads to a measure of likelihood ratio.

In late voice search applications where statistical search is applied for robustness, confidence measures that take into account uncertainties from different system components are more adequate. BBN's DA system applied a generalized linear model classifier to compute confidence score from a set of features extracted from spoken queries and listings [19]. The feature set included word confidences, ASR n-best frequency, etc. Among them, the two most important features were the required and allowable word sets. Much like IBM's signatures, the required word set for a listing is a set of word tuples, at least one of which must be present in a recognized query in order to associate the listing with the query. The allowable word set is a list of words that are allowable in a query to be associated with the listing.

A confidence model based on a maximum entropy classifier was introduced for the Microsoft Research's experimental business DA system [28]. Unlike the required and allowable set features in [19], it takes into consideration the importance of words in a listing with features based on the automatically acquired IDF statistics of the word. The classifier takes multiple features drawn from the ASR, search component and dialog manager, and the combined features extracted from multiple components. For example, the search-related features for a hypothesized listing L and a recognized query Q include the VSM similarity between L and Q; the ratio between the maximum IDF value among the words existing in both L and Q and the maximum IDF value among all the words in L. The combined features attempt to model the dependency among features across different components of voice search. One such feature is the ASR confidence on the word that also exists in L and has the highest IDF value; i.e., the ASR confidence on the word that contributes the most to the search result. The effectiveness of the features were studied with statistical significance tests, which gave rise to several application-independent features for confidence measures in the general voice search framework [28].

User: Tell me about restaurants in London.
System: I know of 596 restaurants in London. All price ranges are represented. Some of the cuisine options are Italian, British, European, and French.
User: I'm interested in Chinese food.
System: I know of 27 restaurants in London that serve Chinese cuisine. All price ranges are represented. Some are near the Leicester Square tube station.
User: How about a cheap one?
System: I know of 14 inexpensive restaurants that serve Chinese cuisine. Some are near the Leicester Square tube station. Some are in Soho.

[FIG5] Search results summarization for disambiguation.

CLOSING THE FEEDBACK LOOP

Every SDS needs to be tuned, often through multiple iterations, for improved performance. This involves a painstaking process of error analysis from logged data. An automatic or semi-automatic tuning tool is one of the most wanted items by many dialog experts. Due to the extreme difficulty of the problem, little work has been seen on automatic remedy for design/implementation flaws in the feedback loop. Most research work focused on automatic flaw discovery from logged data.

In [29], an unsupervised learning algorithm was proposed to obtain the linguistic variants of listings that were not modeled in the Telecom Italia's DA system. A phone-looped model was exploited to obtain the phonetic transcriptions for the utterances that failed the automated service and were routed to the operators. The phonetic transcriptions were clustered with a furthest

neighbor hierarchical clustering algorithm, where two clusters with the shortest distance were merged in iterative steps. The distance between two clusters was defined as the furthest distance between two instance phonetic transcriptions in the clusters, and the distance between two phonetic transcriptions was obtained with the Viterbi alignment using the log-probability for phone insertion, deletion, and substitution, where the probabilities were trained using a set of field data by aligning each decoded phonetic sequence with its corresponding manual transcription. A cluster in the hierarchy was selected according to the following criteria: the number of instances in the cluster must exceed a threshold and the dispersion of the cluster must be smaller than another threshold. The central element of a selected cluster was presented to a spoken dialog expert as a candidate variant of a business listing.

CONFIDENCE MEASURES ARE USED TO DETERMINE WHEN TO ELEVATE AN AUTOMATED SERVICE CONVERSATION TO A LIVE AGENT IN AN EARLY DIALOG STAGE IF THE CONFIDENCE ON THE KEY INFORMATION IS TOO LOW.

A similar algorithm was proposed in [30] to discover the semantic intents that were not covered by an auto-attendant SDS in Microsoft [15]. The system was originally designed to connect a caller to a Microsoft employee with name dialing. It was later found that in addition to name dialing an employee, callers often ask for connections to an office, such as "security" or "shuttle service." To discover these uncovered intents, a LM-based acoustic clustering algorithm was proposed. Unlike the algorithm in [29] that clusters the 1-best phonetic transcriptions, it treats the word transcription and the cluster they belong to as hidden variables and optimizes the parameters associated with them with respect to an objective function. Specifically, given a fixed number of clusters, it builds a cluster-specific language model $p(w|c)$ and a cluster prior model $p(c)$ to maximize $p(x) = \sum_{c,w} p(x, w, c) = \sum_{c,w} p(x|w)\, p(w|c)\, p(c)$, the likelihood of the observed acoustic signal x. In practice, recognition was decoupled from cluster training: a task-independent large vocabulary ASR was used to obtain the hypotheses w and their posterior probabilities. Since w and c are hidden variables, the expectation maximization (EM) algorithm was used to estimate the probability $p(c)$ and $p(w|c)$ by maximizing the objective function. Here the EM algorithm took as input the hypotheses w and $p(w|x)$ obtained from the task-independent ASR. In [30], unigram language models were used for $p(w|c)$. With these cluster-specific distributions, a Kullback-Leibler (KL)-divergence-based distance measure was used in hierarchical clustering. The EM algorithm was subsequently applied for several iterations to re-estimate the model parameters after merging two clusters. The cluster priors obtained from the EM algorithm was used to rank the clusters for presentation to spoken dialog experts.

SUMMARY

This article categorized spoken dialog technology into form filling, call routing, and voice search, and reviewed the voice search technology. The categorization was made from the technological perspective. It is important to note that a single SDS may apply the technology from multiple categories. Robustness is the central issue in voice search. The technology in acoustic modeling aims at improved robustness to environment noise, different channel conditions, and speaker variance; the pronunciation research addresses the problem of unseen word pronunciation and pronunciation variance; the language model research focuses on linguistic variance; the studies in search give rise to improved robustness to linguistic variance and ASR errors; the dialog management research enables graceful recovery from confusions and understanding errors; and the learning in the feedback loop speeds up system tuning for more robust performance.

While tremendous achievements have been accomplished in the past decade on voice search, large challenges remain. Many voice search dialog systems have automation rates around or below 50% in field trials. This provides a fertile ground and great opportunities for future research.

AUTHORS

Ye-Yi Wang (yeyiwang@microsoft.com) received a B.S. in 1985 and an M.S. in 1988, both in in computer science from Shanghai Jiao Tong University, and an M.S. in computational linguistics in 1992 and a Ph.D. in human language technology in 1998, both from Carnegie Mellon University. He joined Microsoft Research in 1998. His research interests include spoken dialog systems, natural language processing, language modeling, statistical machine translation, and machine learning. He served on the editorial board of the *Chinese Contemporary Linguistic Theory* series. He is a coauthor of *Introduction to Computational Linguistics* (China Social Sciences Publishing House, 1997), and he has published over 40 journal and conference papers. He is a Senior Member of IEEE.

Dong Yu (dongyu@microsoft.com) joined Microsoft in 1998 and Microsoft Speech Research Group in 2002, where he is currently a researcher. He holds a Ph.D. in computer science from University of Idaho, an M.S. in computer science from Indiana University, Bloomington, an M.S. in electrical engineering from the Chinese Academy of Sciences, and a B.S. (with honors) in electrical engineering from Zhejiang University (China). His current research interests include speech processing, robust speech recognition, discriminative training, spoken dialog systems, voice search technology, machine learning, and pattern recognition. He has published more than 40 papers and applied for more than 30 patents in these areas and has served as program committee member, organization committee member, and reviewer for many conferences.

Yun-Cheng Ju (yuncj@microsoft.com) received a B.S. in electrical engineering from National Taiwan University in 1984 and a master's and Ph. D. in computer science from the University of Illinois at Urbana-Champaign in 1990 and 1992, respectively. He joined Microsoft in 1994. His research interests include spoken dialog systems, natural language processing, language modeling, and voice search. Prior to joining Microsoft, he worked at Bell Labs for two years. He has published and co-published over 20 journal and conference papers and filed over 30 U.S. and international patents.

Alex Acero (alexac@microsoft.com) received an M.S. from the Polytechnic University of Madrid in 1985, an M.S. from Rice University in 1987, and a Ph.D. from Carnegie Mellon University in 1990, all in electrical engineering. He was with Apple Computer and Telefonica I+D. In 1994 he joined Microsoft Research, Redmond, Washington, where, since 2005, he has been a research area manager. He is an affiliate professor of electrical engineering at the University of Washington, Seattle. He is author of the books *Acoustical and Environmental Robustness in Automatic Speech Recognition* (Kluwer, 1993) and *Spoken Language Processing* (Prentice Hall, 2001), and he has written invited chapters in four edited books and over 150 technical papers. He holds 35 U.S. patents. His research interests include speech and audio processing, natural language processing, image understanding, multimedia signal processing, and multimodal human-computer interaction. He is a Fellow of IEEE. He is the IEEE Signal Processing Society vice president-technical directions and an editorial board member for I*EEE Journal of Selected Topics In Signal Processing* and *IEEE Signal Processing Magazine*. He has also served the Society as a distinguished lecturer, a member of the Board of Governors, and an associate editor for *IEEE Signal Processing Letters* and *IEEE Transactions of Audio, Speech and Language Processing*. He has also been a member of many committees, including chair of the Speech Technical Committee of the IEEE Signal Processing Society and publications chair of ICASSP'98. He is member of the editorial board of *Computer Speech and Language*.

REFERENCES

[1] D. Yu, Y.-C. Ju, Y.-Y. Wang, G. Zweig, and A. Acero, "Automated directory assistance system: From theory to practice," in *Proc. Interspeech*, Antwerp, Belgium, 2007, pp. 2709–2712.

[2] S. Mann, A. Berton, and U. Ehrlich, "How to access audio files of large data bases using in-car speech dialogue systems," in *Proc. Interspeech*, Antwerp, Belgium, 2007, pp. 138–141.

[3] G. Zweig, P. Nguyen, Y.-C. Ju, Y.-Y. Wang, D. Yu, and A. Acero, et al., "The voicerate dialog system for consumer ratings," in *Proc. Interspeech*, Antwerp, Belgium, 2007, pp. 2713–2716.

[4] D. Bohus, S.G. Puerto, D. Huggins-Daines, V. Keri, G. Krishna, R. Kumar, A. Raux, and S. Tomkoohus, "ConQuest: An open-source dialog system for conferences," in *Proc. Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2007, pp. 9–12.

[5] G. Andreani, G. Di Fabbrizio, M. Gilbert, D. Gillick, D. Hakkani-Tür, and O. Lemon, "Let's DiSCoH: Collecting an annotated open corpus with dialogue acts and reward signals for natural language helpdesk," in *Proc. IEEE/ACL Workshop on Spoken Language Technology*, Palm Beach, Aruba, 2006, pp. 218–221.

[6] R. Pieraccini and J. Huerta, "Where do we go from here? Research and commercial spoken dialog systems," in *Proc. 6th SIGdial Workshop on Discourse and Dialogue*, Lisbon, Portugal, 2005, pp. 1–10.

[7] D. Nahamoo, "Speech technology opportunities and challenges (keynote speech)," presented at IEEE/ACL Workshop on Spoken Language Technology, Palm Beach, Aruba, 2006.

[8] Y.-Y. Wang, L. Deng, and A. Acero, "Spoken language understanding: An introduction to the statistical framework," *IEEE Signal Processing Mag.*, vol. 22, no. 5, pp. 16–31, Sept. 2005.

[9] M. Gilbert, J.G. Wilpon, B. Stern, and G. Di Fabbrizio, "Intelligent virtual agents for contact center automation," *IEEE Signal Processing Mag.*, vol. 22, no. 5, pp. 32–41, Sept. 2005.

[10] C.A. Kamm, C.R. Shamieh, and S. Singhal, "Speech recognition issues for directory assistance applications," *Speech Comm.*, vol. 17, no. 3-4, pp. 303–311, 1995.

[11] M. Lennig, G. Bielby, and J. Massicotte, "Directory assistance automation in Bell Canada: Trial results," in *Proc. 2nd IEEE Workshop Interactive Voice Technology for Telecommunications Applications (IVTTA94)*, 1994, pp. 9–13.

[12] R. Billi, F. Cainavesio, and C. Rullent, "Automation of Telecom Italia directory assistance service: Fieldtrial results," in *Proc. IEEE 4th Workshop Interactive Voice Technology for Telecommunications Applications*, Torino, Italy, 1998, pp. 11–16.

[13] B. Buntschuh, C. Kamm, G.D. Fabbrizio, M.M.A. Abella, S. Narayanan, I. Zeljkovic, R.D. Sharp, J. Wright, S. Marcus, J. Shaffer, R. Duncan, and J.G. Wilponntschuh, "VPQ: A spoken language interface to large scale directory information," in *Proc. International Conference on Spoken Language Processing*, Sydney, Australia, 1998, pp. 2863–2866.

[14] Y. Gao, B. Ramabhadran, J. Chen, H. Erdogan, and M. Picheny, "Innovative approaches for large vocabulary name recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, 2001, pp. 53–56.

[15] D. Ollason, Y.-C. Ju, S. Bhatia, D. Herron, and J. Liuason, "MS Connect: A fully featured auto-attendant: System design, implementation and performance," in *Proc. Int. Conf. Spoken Language Processing*, Jeju Island, Korea, 2004, pp. 2845–2848.

[16] A. Acero, R. Chambers, Y.-C. Ju, X. Li, J. Odell, P. Nguyen, O. Scholz, and G. Zweigo, "Live search for mobile: Web services by voice on the cellphone," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Las Vegas, Nevada, 2008.

[17] B. Ramabhadran, L.R. Bahl, P.V. deSouza, and M. Padmanabhan, "Acoustics-only based automatic phonetic baseform generation," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Seattle, WA, 1998, pp. 309–312.

[18] F. Béchet, R. De Mori, and G. Subsol, "Dynamic generation of proper name pronunciations for directory assistance," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Orlando, FL, 2002, pp. I-745–I-748.

[19] P. Natarajan, R. Prasad, R.M. Schwartz, and J. Makhoul, "A scalable architecture for directory assistance automation," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Orlando, FL, 2002, pp. I-21–I-24.

[20] F. Béchet, E. d. Os, L. Boves, and J. Sienel, "Introduction to the IST-HLT project speech driven multimodal automatic directory assistance (SMADA)," in *Proc. Int. Conf. Spoken Language Processing*, Beijing, China, 2000, pp. 731–734.

[21] E.E. Jan, B.ı. Maison, L. Mangu, and G. Zweigan, "Automatic construction of unique signatures and confusable sets for natural language directory assistance applications," in *Proc. Eurospeech*, Geneva, Switzerland, 2003, pp. 1249–1252.

[22] D. Yu, Y.-C. Ju, Y.-Y. Wang, and A. Acero, "N-gram based filler model for robust grammar authoring," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Toulouse, France, 2006, pp. I–1.

[23] H. Schramm, B. Rueber, and A. Kellner, "Strategies for name recognition in automatic directory assistance systems," *Speech Commun.*, vol. 31, no. 4, pp. 329–338, 2000.

[24] C.A. Kamm, M. Walker, and L.R. Rabiner, "The role of speech processing in human-computer intelligent communication," *Speech Commun.*, vol. 23, pp. 263–278, no. 4, 1997.

[25] J. Polifroni and M. Walker, "An analysis of automatic content selection algorithms for spoken dialogue system summaries," in *Proc. IEEE/ACL Workshop on Spoken Language Technology*, Palm Beach, Aruba, 2006, pp. 186–189.

[26] F. Wessel, R. Schlüter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 9, no. 3, pp. 288–298, Mar. 2001.

[27] C.-H. Lee, "A unified statistical hypothesis testing approach to speaker verification and verbal information verification," in *Proc. COST 250*, Rhodes, Greece, 1997, pp. 63–72.

[28] Y.-Y. Wang, D. Yu, Y.-C. Ju, G. Zweig, and A. Acero, "Confidence measures for voice search applications," in *Proc. Interspeech*, Antwerp, Belguim, 2007, pp. 2721–2724.

[29] C. Popovici, M. Andorno, P. Laface, L. Fissore, M. Nigra, and C. Vair, "Learning new user formulations in automatic directory assistance," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Orlando, FL, 2002, pp. I-17–I-20.

[30] X. Li, A. Gunawardana, and A. Acero, "Unsupervised semantic intent discovery from call log acoustics," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Philadelphia, PA, 2005, pp. 45–48.

SP