[Michael L. Seltzer, Yun-Cheng Ju, Ivan Tashev, Ye-Yi Wang, and Dong Yu]

In-Car Media Search

[Performance, challenges, and recent advancements]

© INGRAM PUBLISHING & PHOTODISC

Mobility in Media Search

Over the last decade, our ability to access, store, and consume huge amount of media and information on mobile devices has skyrocketed. While this has allowed people who are on the go to be more entertained, informed, and connected, the small-form factor of mobile devices makes managing all of this content a difficult task. This difficulty is significantly amplified when we consider how many people are using these devices while driving in automobiles and the high risk of driver distraction such devices present. A recent government study concluded that drivers performing complex secondary tasks such as operating or viewing a mobile device or personal digital assistant (PDA) were between 1.7 and 5.5 times more likely to be involved in a crash or near crash [1].

Recognizing the risk posed by the use of mobile devices by drivers, most major car manufacturers have begun selling systems for operating these devices using voice-driven interfaces. Because driving occupies both the user's hands and eyes, voice control has long been proposed as an ideal means of performing in-car tasks. Early in-car systems used voice commands to control many dashboard functions such as the radio, compact disc (CD) player, and climate control. To limit the number of commands active at a time, hierarchical menus were introduced, which put a significant burden on the user to maintain a mental model of the system's menu structure. To alleviate these

Digital Object Identifier 10.1109/MSP.2011.941065
Date of publication: 15 June 2011

1053-5888/11/$26.00©2011IEEE

problems, the research community began investigating new approaches to voice input and multimodal interaction that are robust to the natural ways in which users speak and more tolerant of the common mistakes they make, e.g., [2] and [3]. Such an approach is critical for today's applications where the user may be choosing from one of 50,000 songs on their media player or looking for one of 100,000 local businesses on their navigation device.

A block diagram of a state-of-the-art in-car infotainment system is shown in Figure 1. In a typical scenario, the user issues a query or command by voice, which is then processed by a speech recognizer. The recognizer output is then processed by a search engine that looks through the relevant database index for the best match to the user's query. The results are passed to a dialog manager and returned to the user via speech synthesis and/or a graphical display. Throughout the interaction, the dialog manager keeps track of the state of the system and loads the proper language model (LM) and search index. As this processing chain indicates, implementing this system requires many technologies. While all of the components shown in the figure are important, in this article, we focus on the three technologies that we believe have the biggest impact on the performance of these systems: speech recognition, information retrieval (IR), and multimodal interaction.

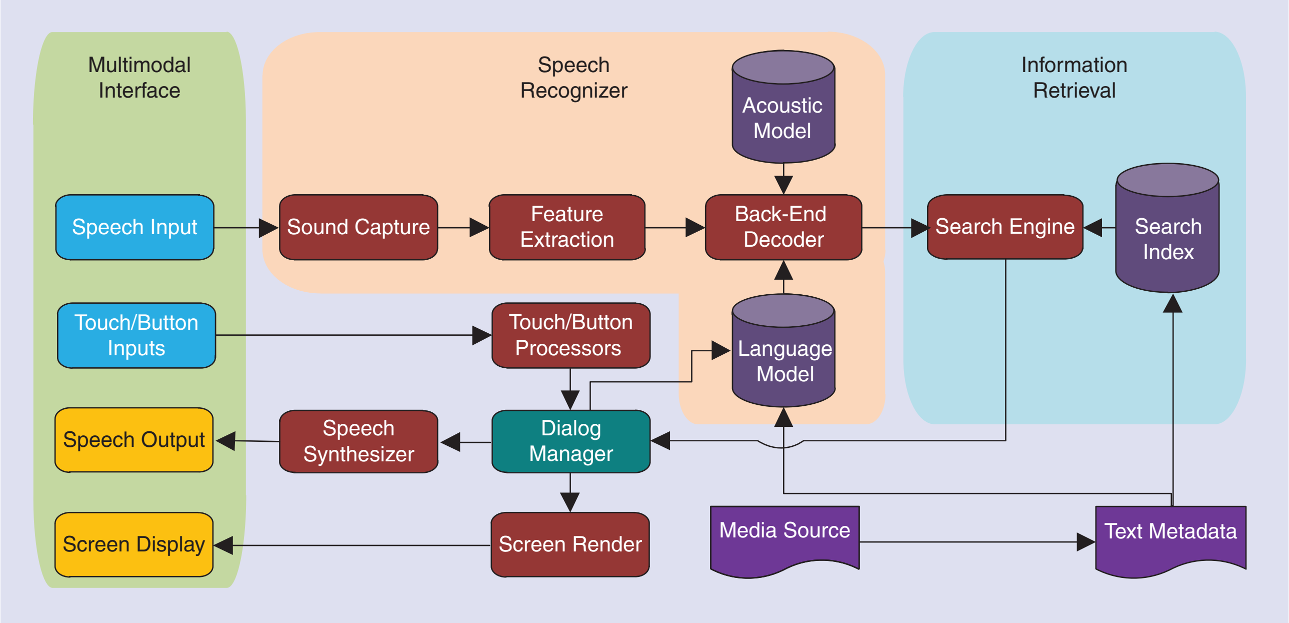
While each of these three fields has been studied for many years, there are unique challenges that arise when they are integrated into a single system for use by drivers in automobiles. For example, because the driver's primary focus is always on the road, only limited cognitive resources and attention can be devoted to the system. As a result, there is a high likelihood that the user will not remember the proper command syntax or the correct name of the item requested. In addition, traditional IR algorithms used in Web search are designed assuming that every word in the query is meaningful. When the queries are generated from noisy speech-recognition output, this may not be true. These challenges strongly influence the design of such systems, and to address them successfully, it is critical to understand the interactions between each of the system components. By taking a holistic view of the end-to-end user experience, the performance of the overall system can exceed the sum of its parts.

In this article, we first present an introduction to speech recognition, IR, and multimodal interaction. In each of these areas, recent algorithmic advances are discussed with particular emphasis on aspects that are important for the in-car media search scenario. We also describe how in-car systems are evaluated in terms of both system performance and driver distraction, including the use of driving simulator studies to assess driver distraction and cognitive load. We then show how all of these technologies have been used to create a prototype in-car infotainment system called Commute UX. We describe the functionality of this system and take a detailed look at two applications, music search and voice reply to text messages. Performance evaluation of these applications with real users is discussed. Finally, we discuss an ongoing research in this area and assess the outlook for the future.

SPEECH RECOGNITION IN AUTOMOTIVE AND MOBILE ENVIRONMENTS

An automatic speech recognition (ASR) system operates by finding the most likely word sequence W^* for a given observed series of acoustic speech events A. This can be done via Bayes' rule, as

$$W^* = \text{argmax}_W P(W|A) = \text{argmax}_W P(A|W)P(W).$$



[FIG1] A block diagram of an in-car multimodal system for media search and information access.

Here the acoustic speech events A are represented with acoustic features obtained through the sound capture and feature extraction processes. These features are then processed using an acoustic model and an LM. The acoustic model $P(A|\mathrm{W})$ describes the probabilities of observing acoustic features given a sequence of words or phonemes, while the LM $P(W)$ captures the prior probabilities of seeing certain words or word sequences. More details about the material in this section can be found in [4].

SOUND CAPTURE

Because the automobile is a noisy environment, the ability to capture the driver's speech as cleanly as possible is critical for success. The signal is first captured by one or more microphones typically located in the headliner, the rearview mirror, or the dashboard. The microphone signals are then processed by a series of signal processing blocks, including acoustic echo cancellation, beamforming noise suppression, and automatic gain control. While the blocks in this processing chain work to improve the overall quality of the captured speech signal, too much noise suppression can cause distortions in the output, which can be detrimental to the speech recognizer. As a result, it is important to adjust the operation of these algorithms for best end-to-end speech-recognition performance instead of highest signal-to-noise ratio (SNR) or other signal processing criteria.

FEATURE EXTRACTION

Once the audio signal is captured, it is processed to extract features for recognition that are discriminative and robust. The features are typically derived from the short-time power spectrum of each frame. The most common features are Mel-frequency cepstral coefficients, though alternate features such as perceptual linear prediction coefficients are also frequently used. It is advantageous to remove noise from the speech-recognition features directly in addition to or instead of the noise suppression applied during sound capture. Widely deployed noise-suppression techniques include spectral subtraction and vector Taylor series feature enhancement [5]. To further reduce the variability in the features, normalization techniques such as cepstral mean normalization are typically applied to the features. The final augmented features used in the ASR consist of the normalized features and their first and second derivatives.

User's Query	User's Intent
All Rise from Blues	*Track*: All Rise *Genre*: Blues
Sarah, In the Arms of an Angel	*Track*: Angel *Artist*: Sarah McLaughlin
Legally Blonde Soundtrack	*Album*: Legally Blonde *Genre*: Soundtrack
Boyz II Men, Hard to Say Goodbye	*Track*: It's So Hard to Say Goodbye to Yesterday *Artist*: Boyz II Men

[FIG2] Examples of spoken queries made by users for items in a personal music collection. There is significant mismatch between the query and the metadata.

ACOUSTIC MODELING

State-of-the-art speech recognizers use acoustic models based on hidden Markov models (HMMs). For large vocabulary speech recognition, it is impractical to have an HMM for every word in the vocabulary, so words are broken down into a set of phonetic units. Each of these units is then modeled by an HMM. It is very helpful to model a phoneme in the context of its neighboring phonemes, so context-dependent triphones are typically used. For example, the HMM that models central "eh" sound in "get" would be modeled separately from the HMM that models the same "eh" sound in the word "beg."

The parameters of an HMM speech recognizer are trained using a corpus of training data consisting of the speech audio and the corresponding transcripts. Because speech-recognition systems are statistical pattern classifiers, they perform best when the data used to train the system match the data seen in deployment. Of course, this is impractical to achieve because of the sheer variety in environmental conditions and sensor configurations. As a result, most models are built using a paradigm called multistyle training, in which the systems are trained using data collected from all acoustic conditions expected to be seen [6]. Further gains can be achieved by noise adaptive training, in which the training data are processed through the same audio pipeline to be used in deployment [5]. This approach has been shown to be superior to either multistyle training or feature enhancement alone.

One of the major challenges in acoustic modeling for an automotive task is collecting speech training data. It is a time-consuming and expensive process. An alternative solution is to use clean high-quality speech utterances, filters that represent the transfer function between the user and the in-car microphone, and samples of in-car noise to create synthetic in-car training data [7].

LANGUAGE MODELING

In addition to a properly trained acoustic model, the performance of a speech-recognition system is critically dependent on the quality of its LM. An LM assigns prior probabilities to utterances or word sequences and significantly constrains the decoding search space. The simplest LM is a probabilistic context-free grammar (PCFG) that lists a set of possible items, each with a corresponding probability of being spoken. PCFGs can be nested to create more complicated grammar structures. PCFGs are easy for the application developers to create and implement, and as a result, this type of grammar is used in many voice platforms on the market today. However, a PCFG-based LM has several limitations. First, it does not scale up well with the number of entries. In fact, the ASR accuracy decreases linearly with logarithmic increases in the number of entries [8]. More importantly, it is not robust to natural spoken queries because of its poor coverage. Utterances spoken by the users that do not exactly match the items in the grammar have a very high chance of being misrecognized. For simple commands, this may not be an issue, but for tasks like music or business search, this is problematic. This is illustrated by a study we performed in which users were asked to make spoken queries for items in their music collections. Example queries are shown in Figure 2, and the overall percentages of queries that did not match the exact song title, album, artist, or genre exactly are shown in Figure 3.

[WHILE SPEECH RECOGNITION AND IR ARE CRITICAL UNDER THE HOOD TECHNOLOGIES, THE USER INTERFACE IS THE DIRECT LINK BETWEEN THE SYSTEM AND THE DRIVER.]

To account for the frequent mismatch between user queries and the actual items they are asking for, a more robust solution is required. Statistical n-gram models offer a flexible and principled approach to improving the robustness of the LM since such models do not require a user's query to match an entry in the list exactly. Using the statistical the n-gram, the probability of a word sequence $p(w_1, \ldots, w_K)$ is approximated as $\prod_{i=1}^{K} p(w_i \mid w_{i-1}, \ldots, w_{i-n-1})$, and the probability of unseen word sequences can be reliably estimated using well-studied smoothing algorithms. Thus, the flexibility of the n-gram over the PCFG lies in the fact that it models the probabilities of short sequences of words (typically one to three words) rather than entire utterances.

A statistical n-gram model performs best when it is built from the transcripts of real utterances. Unfortunately, collecting enough real utterances can be difficult, especially in the early stages of system development. A feasible and effective compromise is to interpolate multiple LMs trained from different data sources [9]. For example, an LM $p_t(w)$ built using the transcripts of real queries can be combined with an LM $p_d(w)$ built from database entries as $p(w) = \lambda p_t(w) + (1-\lambda) p_d(w)$, where the interpolation weight λ can be tuned using a cross-validation set collected from real usage [10]. Building $p_t(w)$ is straightforward if transcripts are available. Otherwise, a semisupervised approach may be taken to extract information from query logs [11]. However, careful measures have to be taken in building $p_d(w)$, as the database entries may not reflect the actual ways users refer to them as we have just explained. One approach that has been shown to be effective is to introduce a statistical variation model to derive the actual users' queries from the database entries [10]. No matter how the LM is built, additional robustness can be obtained by incorporating a large vocabulary background LM since users may utter carrier phrases or there might be side conversations captured in the utterance.

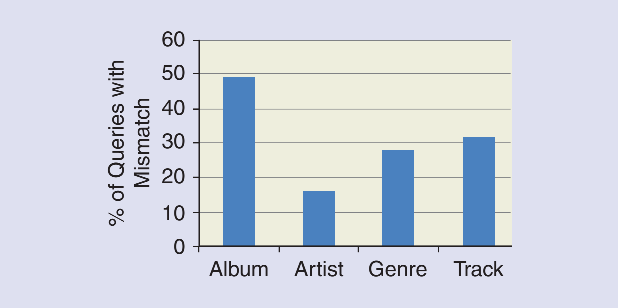
IR USING SPOKEN QUERIES

Once the ASR system hypothesizes what the user said, this output is used to search for the desired item. This process, called spoken query information retrieval (SQIR), is an important technology that can be broadly applied to a variety of in-car applications. This section first introduces traditional text-based IR, which is the technological foundation for SQIR and then addresses the specific challenge of SQIR, particularly how to make IR more robust to ASR errors.

OVERVIEW OF IR

Given a collection of documents D and a query q, the task of the IR is to find a subset of documents from D that are relevant to q and rank them in its decreasing order of relevance. To perform IR, two major problems need to be addressed: 1) how to quickly find the relevant documents from the keywords in the query and 2) how to sort the IR results appropriately. The first problem is addressed by using an inverted index that operates much like the index at the end of a book that allows the readers to quickly find the context of a keyword [12]. The second problem, called relevance ranking, defines the metrics to determine the relevance of returned documents so that they can be sorted. There are three major classes of relevance metrics. They include the TF*IDF-weighted vector space model (VSM), the BM25 classical probabilistic model, and the language model approach to IR (LMIR).

The VSM represents a query or document as a vector. The relevance of the document vector d to the query vector q is measured as the cosine of the angle between the two vectors. Each element in the vector is a weight that represents the importance of a term (e.g., a word) to a query/document. Intuitively, the importance should increase as the term appears more frequently in the query/document, and it should decrease if it appears in many documents, as it is less discriminative. The term frequency (TF) $tf_t(d)$ is the number of occurrences of term t in d, and the inverse document frequency (IDF) often takes the form of the logarithm of the total number of documents divided by the number of documents containing term t: $idf_t = \log(|\mathrm{D}|/|\{d: t \in d\}|)|)$. The weight for term t in the vector is the product of its TF and IDF scores. There are many other ways to weight the vector elements based on the TF and IDF scores, which are discussed in [13].



[FIG3] Percentage of spoken queries that were mismatched to the corresponding text in the metadata.

An example of the VSM with TF-IDF weights is shown in Figure 4 for a small music collection of four actual song titles. The cosine distance between the query and each of the songs is computed and used to rank the search results.

THE PRINCIPAL GOAL OF AN IN-CAR USER INTERFACE IS TO CREATE AN INTUITIVE USER EXPERIENCE WHILE MINIMIZING DRIVER DISTRACTION.

BM25 is a classic probabilistic IR model, which is based on the assumption that the eliteness of a term can be modeled by two Poisson distributions [14]. With some approximations, the TF described earlier can be modified to the following weighting scheme:

$$\frac{tf_t}{k\left((1-b)+b\frac{L(d)}{L_{\text{avg}}}\right)+tf_t},$$

where $L(d)$ is the length of document d and L_{avg} is the average document length of the collection. The constant k makes the weight a nonlinear function of the TF, such that the effect of increasing an already large TF score is minimal, i.e., the score should saturate at some value. The constant b controls the degree of normalization with respect to the document length to adjust the TF.

The LMIR [15] assumes that a query was generated from a relevant document, and the level of relevance can be modeled by the posterior probability according to the following channel model:

$$P(d|q) = P(q|d) * \frac{P(d)}{P(q)} \propto P(q|d) * P(d) \approx \prod_{i=1}^{n} P(q_i|d)P(d) \approx \prod_{i=1}^{n} P(q_i|d). \qquad (2)$$

Here the document unigram LM $P(q_i/d)$is used, which can be replaced by high-order n-grams. The last approximation in the above expression assumes that d, which is a random variable representing different documents, follows a uniform distribution a priori. Alternatively, a nonuniform $P(d)$ can be used.

IR FROM STRUCTURED METADATA

In many cases, text documents are structured, consisting of multiple fields. Improved IR performance can be obtained by algorithms that exploit this structure. In BM25F, the term frequencies are linearly combined before the BM25 nonlinear TF saturation function is applied [16]. While BM25F is an extension to BM25, the HMM approach to structured document IR (HMMIR) can be viewed as an extension to LMIR [17]. In HMMIR, a query term is assumed to be generated from a particular field of a document represented by a hidden variable. The hidden fields are further assumed to follow a Markov dependency. Using f to represent the sequence of fields aligned to the query words in q,

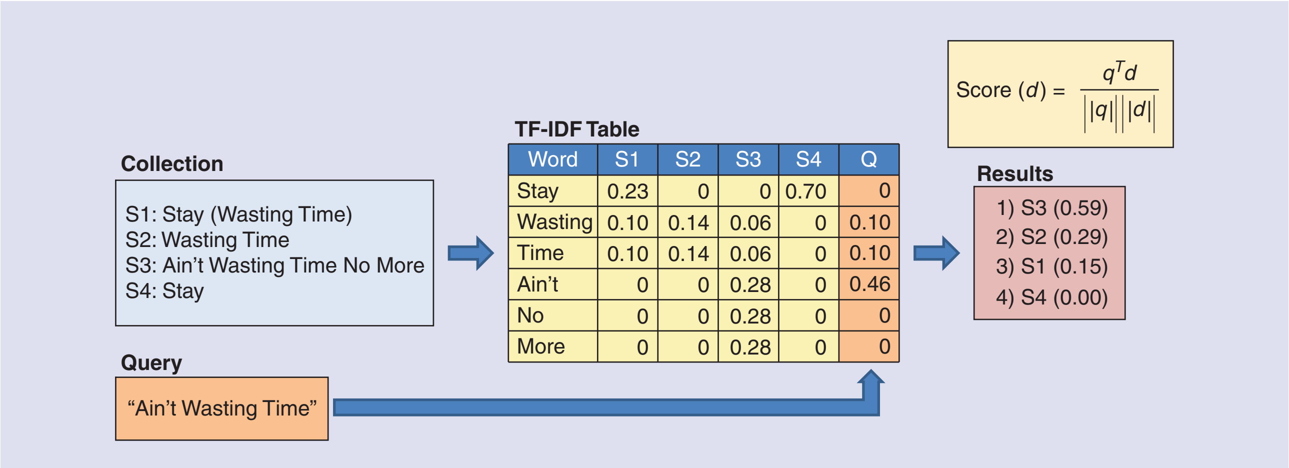
$$P(d|q) = \sum_{f} P(q|f,d)P(f|d) \approx \sum_{f} \prod_{i=1}^{n} P(q_i|f_i,d)P(f_i|f_{i-1},d). \qquad (3)$$

The details for estimating the emission probabilities $P(q_i|f_i,d)$ and the transition probabilities $P(f_i|f_{i-1},d)$ can be found in [17].

FROM TEXT QUERIES TO SPOKEN QUERIES—ROBUST SQIR

In text search, there is an implicit assumption that every word in the query typed by the user is intended and meaningful. In fact, commercial search engines have been engineered to return only those documents that contain all query terms except for a fixed set of stop words. In SQIR, the query is generated by a speech-recognition system and as a result may contain erroneous words. Making the search perform well in the presence of errors in the query is the central challenge of SQIR.

To improve the robustness to ASR errors, two main approaches have been proposed. In the first approach, the word sequence hypothesized by the recognizer is decomposed into smaller units under the assumption that acoustically confusable words will have many units in common at the subword level. In [10], character n-grams are used instead of words as the terms



Collection

S1: Stay (Wasting Time)
S2: Wasting Time
S3: Ain't Wasting Time No More
S4: Stay

Query

"Ain't Wasting Time"

TF-IDF Table

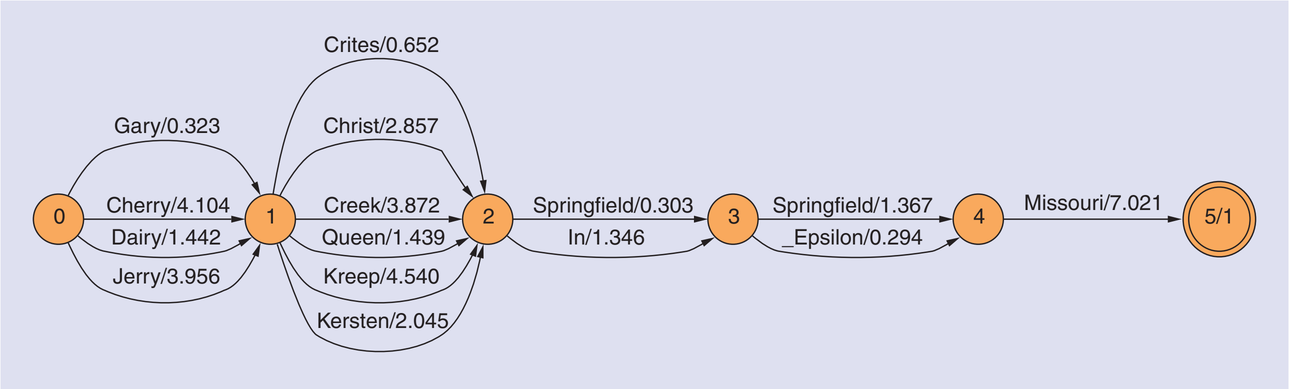
Word	S1	S2	S3	S4	Q
Stay	0.23	0	0	0.70	0
Wasting	0.10	0.14	0.06	0	0.10
Time	0.10	0.14	0.06	0	0.10
Ain't	0	0	0.28	0	0.46
No	0	0	0.28	0	0
More	0	0	0.28	0	0

$$\text{Score }(d) = \frac{q^T d}{|q|\,|d|}$$

Results

1) S3 (0.59)
2) S2 (0.29)
3) S1 (0.15)
4) S4 (0.00)

[FIG4] An example of a VSM with TF-IDF for song titles.



[FIG5] A word-confusion network. Even though the user's query "Dairy Queen Springfield Missouri" was not the top hypothesis, it is present in the network.

in the VSM. For example, the entity "LimeWire" is rewritten as a sequence of character four-grams – $Lim Lime ime_ me_W e_Wi _Wir Wire ire$, where "$" indicates the start and the end of the listing and "_" indicates the separation of words. If a user's query "Lime Wire" is misrecognized as "Dime Wired," there is no word overlap but still many character *n*-grams are common between the ASR output and the intended entity. Similarly, the word sequence can also be broken up into phonetic units using a pronunciation dictionary, and the search can be performed in the space of *n*-grams of phonemes. This approach was applied to a destination entry task in [18], where many acoustically confusable street names are present, e.g., thirtieth versus thirty-eighth versus thirteenth.

The second approach to improving the robustness of SQIR uses the recognizer to generate multiple candidate hypotheses from the recognizer rather than just one. The candidate hypotheses are output in the form of an *n*-best list, a word confusion network, or a lattice. Figure 5 shows an example of a WCN for the query "Dairy Queen Springfield Missouri." In the network, each position in an utterance is associated with a set of confusable words and their negative log posterior probabilities obtained from the recognizer. Although the one-best path from the network misrecognizes the query as "Gary Crites Springfield Missouri," the correct entity name "Dairy Queen" is present in the word-confusion network. The knowledge of the entity in the semantic space, as reflected in the IR relevance metrics, can help recover the correct recognition [19]. Simply put, "Dairy Queen" is more likely to be a valid listing than "Gary Crites."

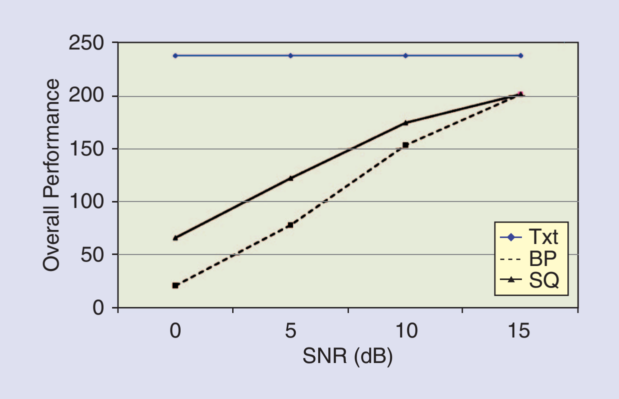
In the speech-in list-out system proposed in [20], SQIR is performed using a query constructed from all terms in the word lattice, each weighted by its posterior probability. Figure 6 shows the performance of this approach on a task to retrieve technical reports using spoken queries as a function of the SNR of speech. As the figure shows, the use of the lattices provides a significant gain in performance in noisier conditions.

MULTIMODAL INTERACTION

While speech recognition and IR are critical under the hood technologies, the user interface is the direct link between the system and the driver. It is therefore perhaps the most crucial component in the success of any in-car media search system. The principal goal of an in-car user interface is to create an intuitive user experience while minimizing driver distraction.

One of the principal ways to reduce driver distraction is to exploit multiple input and output modalities during interaction with the system. Experiments in which the users must perform two tasks simultaneously have shown that modality allocations that take this into account result in less interference than if both tasks have to be performed using the same interaction channels [21]. In the car, the typical available input modalities are speech, buttons on the steering wheel or console, and a touch screen. Output modalities typically include visual displays, earcons (short audio sounds, the audio equivalent of an icon), and voice prompts but may also include tactile displays and force feedback. Augmented reality displays projected onto the windshield have also been prototyped in research laboratories. Within these categories of input and output modalities, many variations are possible.

While there are many aspects of multimodal system design that are critical for good system performance, one of the most important is to leverage the strength of each input



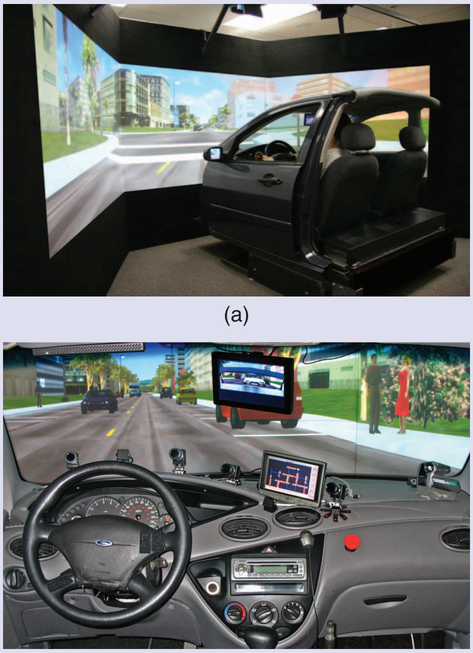
[FIG6] A comparison of the retrieval performance using text (Txt), the top recognition hypothesis (BP) or the word lattice (SQ) as the query as a function of SNR. Using the word lattice is beneficial at low SNRs.

and output modality properly. For example, using touch or buttons to scroll through thousands of audio tracks is time consuming, and in a car, this has been shown to be as distracting as operating a phone [22]. On the other hand, using speech input is a very efficient way to search through such a collection. Even if the query is ambiguous due to speech-recognition errors or the query terms themselves, it is very efficient to return a short list of candidate items to the user. Given a short list of candidate options, speech is no longer necessarily the most efficient modality for either input or output. For example, having a text-to-speech synthesis voice read four song titles aloud may be more distracting and annoying to a driver than simply glancing at a list of four choices. Using buttons or a touch screen to select the desired item from the list is faster and less error prone than using voice input. In short, speech is a good modality for choosing or narrowing down from a large collection of items, whereas a touch screen or buttons are more suitable for smaller lists of only a few items [23].

[ONE OF THE PRINCIPAL WAYS TO REDUCE DRIVER DISTRACTION IS TO EXPLOIT MULTIPLE INPUT AND OUTPUT MODALITIES DURING INTERACTION WITH THE SYSTEM.]

Other key aspects about multimodal in-car user interface design include the following:

- *Discoverability:* At any point during system use, the user should be able to easily discern what the allowable commands and action are. For speech interfaces, ways to do this include the "say what you see" principle in which users can say any command that is visible on a display [24] or the "What can I say?" command where the users can always ask the system for valid commands.
- *Graceful failure with alternate modalities*: No matter how good the speech recognizer is, there will be cases when it fails, due to a speaker's accent or a noisy environment. In such cases, the user should still be able to complete the task using an alternate input modality [25].
- *Design of voice prompts*: Properly designed voice prompts spoken by the system can reduce dialog turns, unnecessary confirmations, and the chance the user will say something unexpected to the system, which, in turn, reduces task completion time and minimize distraction [26].


(a)

(b)

[FIG7] (a) Exterior and (b) interior views of a high-fidelity driving simulator. The car body sits on a computer-controlled motion platform. The car cabin is equipped with video cameras and a gaze tracker. (Photos courtesy of Andrew L. Kun.)

One of the difficulties in designing multimodal user interfaces is that their evaluation is highly subjective and requires extensive usability testing. One way to objectively evaluate in-car interfaces is to conduct driving simulator studies to quantify the impact of various design choices on driving performance. Driving studies can be performed using simple setups such as a video gaming steering wheel and pedals connected to a personal computer (PC) or elaborate immersive simulators with large-screen projection, 180° field of view, and a car body on a computer-controlled motion platform, as shown in Figure 7.

A typical simulator study consists of two simultaneous tasks: a primary driving task and a secondary task that involves one or more user interfaces under test. Typical driving tasks include the lane-change task, where the road signs indicate various lane changes to be performed, and the car-following task, where the driver is instructed to maintain a constant distance to a lead car. Performance for both the driving task and secondary tasks are recorded. For the driving task, the system can measure the speed, lane position, and/or following distance as well as any additional traffic violations. In addition, video cameras and gaze trackers can be used to evaluate the user's attention. For the secondary task, task duration and task completion rate can be measured as well. All of this data can be used to inform decisions about user preference, system performance, and driver distraction.

Several studies that have been recently published have begun to assess the effect of speech interfaces on driving performance. For example, a lane-change task study was performed in [27] to compare speech input to traditional device input for music selection, phone dialing, and destination entry under the assumption of perfect speech recognition. It was concluded that speech interfaces can potentially be far less distracting than manual device input, as measured by mean lane deviation, and frequency and duration of gazes away from the road. The impact of speech-recognition accuracy on driving performance was assessed in [28]. Study participants performed a car-following task while performing a simple dialog task. The study showed that when the recognition accuracy was low, drivers exhibited a greater average variance in steering wheel angle, indicating a detrimental effect on driving performance.

ONE OF THE DIFFICULTIES IN DESIGNING MULTIMODAL USER INTERFACES IS THAT THEIR EVALUATION IS HIGHLY SUBJECTIVE AND REQUIRES EXTENSIVE USABILITY TESTING.

EXAMPLE APPLICATIONS

The technologies described for speech recognition, IR, and multimodal interaction can be combined to create a useful system for media search and information access. For example, the Commute UX multimodal infotainment system [29] is a research prototype that runs on an embedded platform with a speech-recognition engine designed for automotive applications. It has a 7-in touch-screen display and is connected to a cluster of five buttons on the steering wheel for push-to-talk, cancel, up, down, and select. The system can use input from speech, touch, or button, and output can be spoken and/or displayed on the screen. The system enables the driver to make phone calls, play music from a collection, reply to incoming text messages, and search the car owner's manual (while parked).

Regardless of the algorithms used in each component, the overall user experience is consistent across applications. At any time, the user makes a query by pressing the push-to-talk button and then speaking. Speech recognition and SQIR are performed, and the top result or results are returned. When only a single result is returned or the top result has a much higher score than the other choices, the system does not prompt the user for confirmation. If there are several comparable choices, then the top four choices are displayed on the screen. Auxiliary information is provided to prevent ambiguity in the list of items. For example, in response to a request for "yellow submarine," the list may contain "Yellow Submarine [track]" and "Yellow Submarine [album]." The user can select the desired item using steering wheel buttons, touch, or speech. If none of the choices matches the user's desired intent, the user respeaks the query. We believe this is more efficient and less cognitively demanding than initiating a dialog with the user to correct the error.

We now describe how the music search and text message reply applications in Commute UX operate. We chose these two as examples of tasks with structured and unstructured data. The same technology is used for the other applications in the system.

MUSIC SEARCH AND PLAYBACK

Probably the most common task for in-car media search is the ability to play music from a collection stored on a portable media device or memory card. To build the speech-recognition LMs and the index for the IR engine, the music collection is crawled to extract the metadata from each of the audio files. Each item in the metadata that can be queried is considered a structured document composed of one or more fields. For example, a document for the song "It's So Hard to Say Goodbye to Yesterday" could be represented as

Artist:	Boys II Men
Track:	It's So Hard to Say Goodbye to Yesterday
Album:	Legacy—The Greatest Hits Collection
Genre:	R&B/Soul

The documents do not need to contain all fields. Removing the track field in this example would result in a new entity that represents the album rather than the song. To exploit the structure in the music metadata, IR was performed using the HMMIR approach described earlier. Recall from (3) that the HMMIR retrieval engine needs to compute $P(q_i|f_{i,}d)$, the probability of that query word came from a given field in a given document. This is done via a set of LMs trained using maximum likelihood estimation and smoothed with a document-specific model and a global LM trained from all the metadata

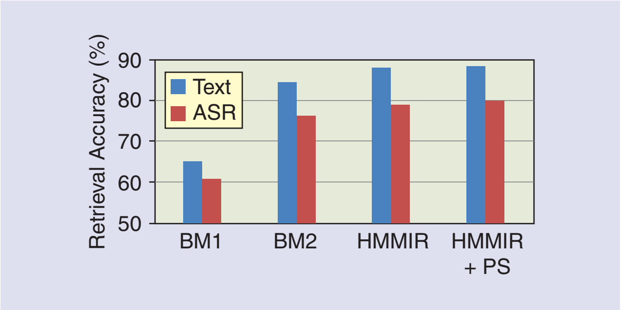
$$P(q_i|f_{i,}d) = \lambda_f P(q_i|f_i, d)+\lambda_d P(q_i|d)+\lambda_g P(q_i), \qquad (4)$$

where $\lambda_f+\lambda_d+\lambda_g = 1$.

To obtain further robustness against speech-recognition errors, a model of phonetic confusability was incorporated into the HMMIR generative model. To do so, the LM $P(q_i|f_i, d)$ was reestimated as the interpolation of the scores of four models, the three shown in (4) and a new term $P_r(q_i|f_i, d)$ that computes the phonetic similarity between the query word and the terms in the field f_i. This new term is computed via dynamic programming and a phonetic confusability matrix.

To validate this approach to music search, a series of experiments was performed using 425 spoken queries from 29 different users. The users were asked to speak a series of music queries and then later were asked to identify the music item they were looking for. The metadata of the desired items was added to a preexisting music collection, resulting in a database of 11,000 songs, each with associated metadata. Example queries are shown in Figure 2. Notice that information from more than one field is often specified in a query. In fact, more than half of the queries contained information from multiple fields.

We compared the performance of the HMMIR-based approach with two baseline models. The first model (BM1) uses LMIR for retrieval with an LM for each field and assumes that each query will only contain information about a single field. This system mimics the behavior of many commercially available systems. The second baseline model (BM2) also uses LMIR but collapses the structure in each item and treats all words in all fields equally. This is a simple way to handle multifield



[FIG8] **Retrieval accuracy for a music search task for text and speech input.**

queries. The results are shown in Figure 8 for both text queries (transcriptions of the spoken queries) and ASR output. The word error rate (WER) of the ASR output is 25.3%. As the figure indicates, the HMMIR approach improves the performance over both baseline models. Using the phonetic similarity model provides a small additional improvement. Comparing the text and speech-recognition performance, it is interesting to note that, while the word accuracy has degraded by 25% from the text transcriptions, the degradation in IR is less than 10%, showing the IR system's robustness to ASR errors.

REPLYING TO TEXT MESSAGES

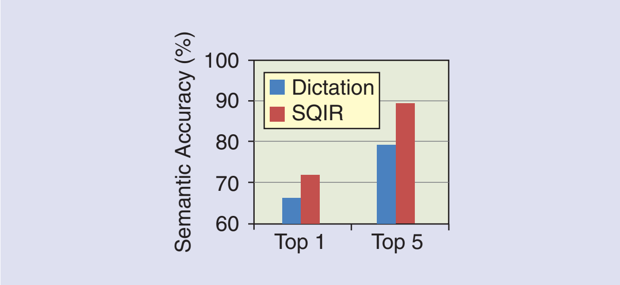
While the combination of speech recognition and IR seems like a natural fit for performing a search of a media collection by voice, it can also be applied in a less traditional manner to other tasks. One of the biggest sources of driver distraction is text messaging while driving. Current in-car systems on the market today allow the users to send one of a small set of common replies (typically 20) using steering wheel buttons or speech to scroll through the set and access the desired message. Using buttons can be tedious and time consuming, while using speech is quite difficult as it requires the user to commit the exact set of text messages to memory. Another approach would be to treat text messaging as a large vocabulary dictation task much like an e-mail or document dictation. However, this approach is problematic because of the high error rates of large vocabulary dictation expected in a car environment. Correcting dictation errors is not feasible for drivers who otherwise need to pay attention to the road.

Alternatively, a SQIR approach can be applied in a manner that takes advantage of the simplicity of canned responses and the flexibility and naturalness of dictation [30]. In this approach, a large set of short message service (SMS) responses is collected to form a training corpus. This corpus is used to train an n-gram dictation-style LM and to build a VSM-based index in which each SMS message is considered a document. Generalization from the messages in the training corpus is achieved by converting the messages into templates using slots. For example, the message "I'll see you at 2:30 pm" has a corresponding template "I'll see you at <T>." In particular, references in the training data to numbers, times, names, and places were replaced with slots. At run time, the recognized utterance and the syntactic parsing tree from the recognition result are obtained. For example, for the utterance "five minutes I'll see you at two thirty pm," the parse tree contains "<N> = 5(five)," and "<T> = 2:30 PM (two thirty pm)." This gives us enough information to construct the search query "<N> minutes I'll see you at <T>" as well as values to instantiate the slots of any retrieved template. Because of the limited coverage of the templates present in the IR engine, what is returned may not be an exact match for what the user said but is often a semantically equivalent paraphrase, as shown in the following example interaction:

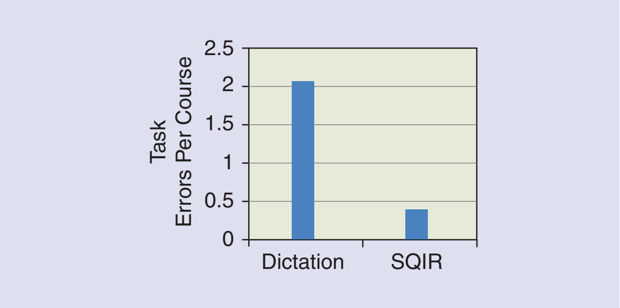
Incoming Message:	Lunch together?
User's Response:	No, I can't have lunch today how about next week.
ASR Hypothesis:	No, I can get lunch today out of next week.
SQIR Hypothesis:	Not today. Next week?

Of course, this approach will only work if the user's utterance closely matches the training data. For that reason, the system is solely intended for replying to time-sensitive messages that typically involve time and confirmation.

To evaluate this approach to spoken SMS replies, experiments were performed to compare a traditional dictation approach to the SQIR approach. Both systems used the same n-gram LM trained from a corpus of 14,000 text messages from 350 participants. In the SQIR approach, the training corpus was also used to construct a VSM-based search engine. Both approaches were evaluated using a separate corpus of about 1,100 spoken SMS reply messages collected from 14 different participants.



[FIG9] **Accuracy of spoken SMS replies using two different approaches: an *n*-gram dictation LM and an SQIR-based approach using a template library.**



[FIG10] **Average number of user confirmation errors for the two approaches to replying to SMS messages by voice.**

Speech data were recognized by the dictation and SQIR systems. In the dictation system, the recognizer generated the top five candidate hypotheses. In the SQIR system, a candidate list of SMS reply templates was retrieved for each recognition hypothesis in the n-best list and the set of all candidate templates was ranked according to TF-IDF score. In both cases, the semantic accuracy of the items in the final n-best list was evaluated by an independent rater. Figure 9 shows the semantic accuracy of the two different approaches for the top one and top five candidate hypotheses. As the figure shows, the SQIR approach using template messages provides an improvement in semantic accuracy over a more traditional dictation approach.

Although this shows that the SQIR approach to replying to SMS messages is more accurate than the dictation approach, the users may have trouble verifying whether the SMS response templates proposed match their intended meaning, especially while driving. To investigate this, a driving simulator study was performed comparing the dictation and SQIR approaches. In this study, participants had to drive a high fidelity driving simulator while obeying all the rules of the road. As a secondary task, participants listened to an incoming SMS message and a formulated reply, and then repeated the reply to the system. A list of four candidate responses was then presented, and the users were asked to pick the correct reply or reject all choices. In this experiment, no actual recognition was performed, and the results presented to the user were obtained from the log files of the previous experiment. The responses were selected so that the correct response was always in the candidate list. There were 16 participants, and each performed ten SMS reply tasks on each driving course, half with the dictation approach and half with the SQIR approach, and the order was randomized.

Interestingly, the study showed no significant differences in driving performance between the two methods. However, as shown in Figure 10, the average number of user errors per driving course was approximately six times higher for the dictation approach than the SQIR approach. That is, drivers were far more likely to correctly locate the response in the list that best matched what they said when using the SQIR system. In the post hoc analysis, it was determined that the dictation system often presented several phonetically confusable options in the candidate list, which made choosing the correct response more difficult for the users.

CONCLUSIONS

THE WIDESPREAD ADOPTION OF SMARTPHONES AND PORTABLE MEDIA PLAYERS HAS CREATED A SIGNIFICANT CHALLENGE FOR THE AUTOMOTIVE COMMUNITY.

The widespread adoption of smartphones and portable media players has created a significant challenge for the automotive community. Studies have shown that operating these types of devices while driving causes levels of distraction that match or exceed those caused by conventional mobile phones. In this article, we have attempted to describe how in-car systems for media search and IR have the potential to significantly reduce driver distraction while enabling users to perform their desired tasks. By combining speech recognition with IR, we can create applications for media search and information access that are robust to speech-recognition errors and the user's natural language input.

While we have described the algorithms and methods used to perform speech recognition, SQIR, and multimodal interaction, we would also like to stress that there is ample room for improvement in each of these areas. Despite progress in front-end processing and acoustic modeling, the adverse effects of environmental noise remain an issue for speech recognizers. Nonstationary noise and background talkers are especially problematic. In SQIR, much of the technology has been borrowed from well-known methods in Web search. Yet, clearly, a Web page and a song in a collection are significantly different, and as such, alternate algorithms for relevance ranking of media could be possible. Multimedia interaction today is limited to speech, buttons, and touch. However, the use of video, gestures, augmented reality, and alternate sensors are being explored in the automotive research community and could profoundly impact the in-car user experience in the future.

AUTHORS

Michael L. Seltzer (mseltzer@microsoft.com) received his B. Sc. degree from Brown University in 1996 and his M.S. and Ph.D. degrees from Carnegie Mellon University in 2000 and 2003, respectively, all in electrical and computer engineering. Since 2003, he has been a researcher in the Speech Technology Group at Microsoft Research. From 2005 to 2008, he was a member of the IEEE SPS Speech and Language Technical Committee and editor-in-chief of its electronic newsletter. He is a Senior Member of the IEEE and is currently an associate editor of *IEEE Transactions on Audio, Speech, and Language Processing*. His research interests include speech recognition in adverse environments, acoustic modeling, speech signal processing, and machine learning for speech and audio.

Yun-Cheng Ju (yuncj@microsoft.com) received his B.S. degree in electrical engineering from National Taiwan University in 1984 and his master's and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign in 1990 and 1992, respectively. He worked at Bell Labs for two years and joined Microsoft in 1994. He has published and copublished more than 30 journal and conference papers and filed more than 50 U.S. and international patents. His research interests include spoken dialog systems, natural language processing, language modeling, and voice search.

Ivan Tashev (ivantash@microsoft.com) received his diploma engineer in electronics and a Ph.D. degree in computer science from the Technical University of Sofia, Bulgaria, where he was an assistant professor until he joined Microsoft in 1998. He is a member of the Speech Technology Group at Microsoft Research. He works in the area of audio signal processing and has created numerous technologies transferred to Microsoft Windows, Round Table, Microsoft Auto, and Kinect. He is a Senior Member of the IEEE and a member of the Audio and Acoustic Signal Processing Technical Committee. He has published more than 70 papers, four books, and has 18 U.S. patents granted.

Ye-Yi Wang (yeyiwang@microsoft.com) received his B.S. and M.S. degrees in computer science in 1985 and 1988, respectively, from Shanghai Jiao Tong University. He received his M.S. degree in computational linguistics and Ph.D. degree in human language technology in 1992 and 1998, respectively, both from Carnegie Mellon University. He joined Microsoft Research in 1998, and he is currently a principal lead researcher in the Online Service Division (Bing Search), Microsoft Corporation. He is currently an associate editor of *ACM Transactions on Asian Language Information Processing.* He is a coauthor of *Introduction to Computational Linguistics*, and he has published more than 50 journal and conference papers and book chapters. He is a member of ACM and ACL and a Senior Member of the IEEE. His research interests include IR, user intent understanding, spoken dialog systems, and natural language processing.

Dong Yu (dongyu@microsoft.com) received his Ph.D. and M.S. degrees in computer science and M.S. and B.S. degrees in electrical engineering. He is a researcher at the Microsoft Speech Research Group. He has published more than 70 papers and filed more than 40 patents. He is a Senior Member of the IEEE and is an associate editor of *IEEE Signal Processing Magazine*, the lead guest editor of *IEEE Transactions on Audio, Speech, and Language Processing* (Special Issue on Deep Learning for Speech and Language Processing), and a guest professor at the University of Science and Technology of China. His current research interests include speech processing and machine learning.

REFERENCES

[1] S. G. Klauer, T. A. Dingus, V. L. Neale, J. D. Sudweeks, and D. J. Ramsey, "The impact of driver inattention on near-crash/crash risk: An analysis using the 100-car naturalistic driving study data," United States Government National Highway Traffic Safety Administration, Office of Human-Vehicle Performance Research, 2006.

[2] R. Pieraccini, K. Dayanidh, J. Bloom, J. G. Dahan, M. Phillips, B. R. Goodman, and K. V. Prasad, "A multimodal conversational interface for a concept vehicle," in *Proc. Eurospeech*, Geneva, Switzerland, 2003, pp. 2233–2236.

[3] D. Buhler and W. Minker, "The SmartKom mobile car prototype system for flexible human-machine communication," in *Spoken Multimodal Human-Computer Dialogue in Mobile Environments* (Text Speech and Language Technology). New York: Springer-Verlag, vol. 28, no. 2, pp. 185–202, 2005.

[4] X. D. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing*. Englewoodcliffs, NJ: Prentice-Hall, 2001.

[5] J. Droppo and A. Acero, "Environmental robustness," in *Springer Handbook on Speech Processing and Speech Communication*. New York: Springer-Verlag, 2008, pp. 653–679.

[6] E. Lippmann, A. Martin, and D. B. Paul, "Multi-style training for robust isolated-word speech recognition," in *Proc. ICASSP*, Dallas, TX, 1987, pp. 705–708.

[7] D. Giuliani, M. Matassoni, M. Omologo, and P. Svaizer, "Training of HMM with filtered speech material for hands-free recognition," in *Proc. ICASSP*, Phoenix, AZ, Mar. 1999, pp. 449–452.

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

[9] F. Jelinek and R. Mercer,"Interpolated estimation of Markov source parameters from sparse data," in *Proc. Workshop on Pattern Recognition in Practice*, 1980, pp. 381–397.

[10] Y.-Y. Wang, D.Yu, Y.-C. Ju, and A. Acero, "An introduction to voice search," *IEEE Signal Processing Mag. (Special Issue on Spoken Language Technology)*, vol. 25, no. 3, pp. 29–38, May 2008.

[11] X. Li, P. Nguyen, G. Zweig, and D. Bohus, "Leveraging multiple query logs to improve language models for spoken query recognition," in *Proc. ICASSP*, 2009, pp. 3713–3716.

[12] C. D. Manning, P. Raghaven, and H. Schutze, *Introduction to Information Retrieval*. Cambridge, UK: Cambridge Univ. Press, 2008.

[13] E. Greengrass, "Information retrieval: A survey," Technical Rep., Univ. Maryland, Baltimore, 2000.

[14] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inform. Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[15] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *Proc. ACM SIGIR Conf. Research and Development in Information Retrieval*, Melbourne, Australia, 1998, pp. 275–281.

[16] S. Robertson, H. Zaragoza, and M. Taylor. "Simple BM25 extension to multiple weighted fields," in *Proc. ACM Conf. Information Knowledge Management* (CIKM), 2004, pp. 42–49.

[17] Y.-I. Song, Y.-Y. Wang, Y.-C. Ju, M. Seltzer, I. Tashev, and A. Acero, "Voice search of structured media data," in *Proc. ICASSP*, Taipei, Taiwan, 2009, pp. 3941–3944.

[18] M. L. Seltzer, Y. C. Ju, I. Tashev, and A. Acero, "Robust location understanding in spoken dialog systems using intersections," in *Proc. Interspeech*, Antwerp, Belgium, 2007, pp. 2813–2816.

[19] J. Feng, S. Banglore, and M. Gilbert, "Role of natural language understanding in voice local search," in *Proc. Interspeech*, Brighton, U.K., 2009, pp. 1859–1862.

[20] C. Forlines, B. Schmidt-Nielsen, B. Raj, K.Wittenburg, and P. Wolf, "A comparison between spoken queries and menu-based interfaces for in-car digital music selection," in *Proc. INTERACT*, Sept. 2005, pp. 536–549.

[21] C. D. Wickens, M. Vidulich, and D. Sandry-Garza, "Principles of S-R-C compatibility with spatial and verbal tasks: The role of display-control interfacing," *Hum. Factors*, vol. 26, pp. 533–534, 1984.

[22] D. Salvucci, D. Markley, M. Zuber, and D. Brumby, "iPod distraction: Effects of portable music-player use on driver performance," in *Proc. CHI*, San Jose, CA, 2007, pp. 243–250.

[23] P. Cohen, "The role of natural language in a multimodal interface," in *Proc. UIST*, Monterey, CA, 1992, pp. 143–149.

[24] N. Yankelovich, "How do users know what to say?," *ACM Interact.*, vol. 3, no. 6, pp. 32–43, 1996.

[25] S. Oviatt and R. Van Gent, "Error resolution during multimodal human computer interaction," in *Proc. ICSLP*, Philadelphia, PA, 1996, vol. 1, pp. 204–207.

[26] R. Harris, *Voice Interaction Design*. San Francisco, CA: Elsevier, 2005.

[27] J. Maciej and M. Vollrath, "Comparison of manual vs. speech-based interaction with in-vehicle information systems," *Accident Anal. Prevention*, vol. 41, no. 5, pp. 924–930, Sept. 2009.

[28] A. Kun, T. Paek, and Z. Medenica, "The effect of speech interface accuracy on driving performance," in *Proc. Interspeech*, 2007, pp. 1326–1329.

[29] I. Tashev, M. Seltzer, Y.-C. Ju, Y.-Y. Wang, and A. Acero, "Commute UX: Voice enabled in-car infotainment system," in *Proc. Mobile HCI'09: Workshop on Speech in Mobile and Pervasive Environments (SiMPE)*, Association for Computing Machinery, Inc., Bonn, Germany, 2009, pp. 22–28.

[30] Y.-C. Ju and T. Paek, "A voice search approach to replying to SMS messages in automobiles," in *Proc. Interspeech,* 2009, pp. 987–990.