A Voice Search Approach to Replying to SMS Messages in Automobiles

Yun-Cheng Ju, Tim Paek

Microsoft Research, Redmond, Washington, USA

{yuncj|timpaek}@microsoft.com

Abstract

Automotive infotainment systems now provide drivers the ability to hear incoming Short Message Service (SMS) text messages using text-to-speech. However, the question of how best to allow users to respond to these messages using speech recognition remains unsettled. In this paper, we propose a robust voice search approach to replying to SMS messages based on template matching. The templates are empirically derived from a large SMS corpus and matches are accurately retrieved using a vector space model. In evaluating SMS replies within the acoustically challenging environment of automobiles, the voice search approach consistently outperformed using just the recognition results of a statistical language model or a probabilistic context-free grammar. For SMS replies covered by our templates, the approach achieved as high as 89.7% task completion when evaluating the top five reply candidates.

Index Terms: SMS, information retrieval, voice UI, voice search

1. Introduction

Many mobile device users want to remain connected wherever they are, whatever they may happen to be doing, even driving. Because of the rapid growth of Short Message Service (SMS) text messaging, with market research predicting that more than 3 trillion messages will be sent in 2009 alone [1], automotive infotainment systems such as the *Ford Sync* [2] now provide drivers the ability to hear incoming messages using text-to-speech (TTS). However, the question of how best to allow users to respond to these messages using automatic speech recognition (ASR) remains unsettled. Given the acoustically challenging environment of automobiles and the potentially hazardous effects of poor ASR on driving performance [3], it is critical that users receive intelligible reply candidates which can be quickly selected by voice or touch (e.g., via a multimodal interface). In this paper, we consider several approaches to replying to SMS messages in automobiles and advocate a robust voice search approach based on template matching. The templates are empirically derived from a large SMS corpus which is also used for language modeling, and matches are accurately retrieved using a vector space model.

This paper is organized as follows: In Section 2, we assess possible approaches to replying to SMS messages and explain our motivation for casting SMS replies as a voice search problem. In Section 3, we delve into implementation details and describe how we derived both a hierarchical statistical language model (SLM) and our response templates from our SMS corpus. We also explicate how we perform template matching. In Section 4, we evaluate our approach within the acoustic setting of automobiles by comparing it against using just the recognition results of either the SLM or a probabilistic context-free grammar (PCFG). Finally, we conclude in Section 5 with a discussion of possible extensions, and opportunities for future research.

2. Possible Approaches

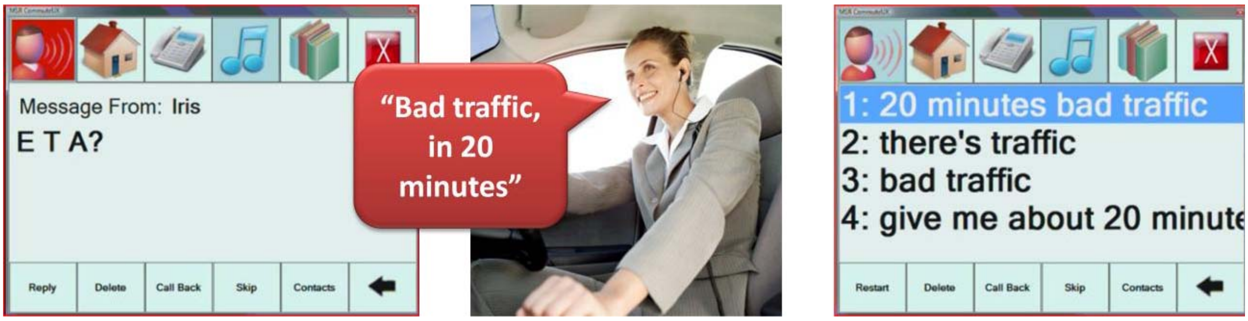
Safe driving requires the constant attention of drivers with their eyes predominantly on the road and their hands on the steering wheel. If a driver should receive a SMS message on the road, it would be better to respond using ASR than typing because voice affords “hands-free, eyes-free” [4] input. This is true assuming that utterances are correctly recognized. Unfortunately, under the noisy conditions of the road, misrecognitions abound, necessitating the visual inspection of recognition candidates before a SMS reply is dispatched. In fact, even state-of-the-art, server-based recognition of mobile voicemail [5] still faces word error rates (WER) of up to 30%-50%. Assuming that ASR in automobiles is likely to be error-prone, we seek an approach to replying to SMS messages that generates reply candidates which can be quickly accepted as correct or “good enough”, thereby minimizing driver distraction. We now consider possible approaches.

2.1. Canned response approach

The approach currently taken by Ford Sync is to allow users to reply to SMS messages using about a dozen canned responses. These responses are specified in a CFG for grammar-based recognition. Although ASR accuracy is generally very high (given the low perplexity), it comes at the expense of naturalness [4]. Users are constrained to use only pre-defined phrases, which they must commit to memory. While learning a dozen responses may not be too burdensome, asking users to learn more than that may deter and even repel potential users. Furthermore, as we demonstrate in Section 4, grammar-based recognition of canned responses does not scale well in comparison to other approaches.

2.2. SMS dictation approach

Another possible approach is to treat SMS replies as a dictation task and leverage n-gram SLMs [6] for recognition. Instead of recognizing conversational speech in general, the SLM can be trained and tuned on SMS messages. An example of this approach is Promptu System’s *ShoutOut* [7] for the iPhone, which hails itself as the first voice-to-SMS application. Unlike the previous approach (Section 2.1), dictating SMS messages frees users from having to remember canned responses and allows them to use natural, unconstrained speech. The SLM also provides robustness to out-of-grammar utterances [4].



Message From: Iris

E T A?

1: 20 minutes bad traffic

2: there's traffic

3: bad traffic

4: give me about 20 minute

Figure 1. Screenshots of a multimodal automotive infotainment system for SMS replies which utilizes voice search to match empirically derived response templates.

The downside of this approach is the correction experience. No matter how sophisticated the correction technique, such as multimodal error recovery using touch [8], handwriting gestures [9], or even the ubiquitous drop-down list of word alternates [9], correction interfaces invariably entail visual identification of where the error occurred, and cognitive effort to decide how best to edit it. Fixing dictation errors can therefore be demanding on drivers who otherwise need to pay attention to the road. And dictation errors are to be expected given the acoustically challenging environment of automobiles. Furthermore, researchers have found that when users encounter recognition errors with in-car speech interfaces, they tend to drive worse, presumably because they are trying to figure out why their utterances are failing [3]. Any frustration drivers may be feeling will be compounded by TTS articulation of misrecognized utterances. For example, suppose a user replies to the SMS message "*how about lunch?*" with "*can't right now running errands*", which then gets misrecognized as "*can right now fun in errands*". Not only would it be difficult to comprehend the TTS rendering of this recognized output, but it would also be burdensome to engage in multimodal correction while driving.

2.3. Voice search approach

Motivated by the shortcomings of the two previous approaches, we sought to combine the simplicity of having canned responses with the naturalness and robustness of SMS dictation. We contend that a voice search approach to SMS replies achieves both.

Voice search (see [10] for introduction) treats utterances as spoken queries to a large index, such as business listings [11][13] or music library [12]. It is formulated as both a recognition and information retrieval (IR) task, where an utterance is first converted into text and then used as a search query for IR [11]. Automated directory assistance (ADA) exemplifies the challenges and advantages of voice search. Not only are there millions of possible business listings (e.g., 18 million in the US alone), but users frequently do not know, remember, or say the exact business names as listed in the directory [10]. As such, voice search leverages n-gram SLMs to generalize the various ways of referring to listings, and to compress the language model [13]. In addition, voice search facilitates natural user expressions without resorting to semantic analysis or classification, such as call-routing [14], by leveraging robust and easy-to-train vector space models for IR, such as term frequency-inverse document frequency (TFIDF) [15]. These models do not need additional training data other than the index entries themselves, and the index can comfortably scale to millions of entries (as evidenced by commercially deployed ADA applications such as [16]).

Framing SMS replies as voice search enables us to reap its benefits. Instead of requiring users to memorize a dozen canned responses, we can maintain an index of thousands of responses. Responses can also be generalized as templates for greater applicability. Because vector space models are robust to variations and invariant to word order, users can also speak in an unconstrained fashion. For example, suppose a user again receives the SMS message "*how about lunch?*" Whether the user responds with "*can't right now running errands*", "*running errands can't right now*" or "*can't running errands right now*" does not matter; voice search will retrieve the same response "*can't right now running errands*". In fact, even if the SLM misrecognizes the utterance as "*can right now fun in errands*", because of low frequency terms like "errands", the vector space model is likely to retrieve relevant responses. The following is an example of how the voice search system we describe in Section 3 can recover from misrecognitions:

1. System: "*Message from Iris. how about lunch? Say 'Reply', 'Delete', 'Call back' or 'Skip'*"
2. User: "*Reply. No I can't have lunch today. How about next week?*"
3. System [recognized output]: **no I** can get **lunch today** out of **next week**
4. System [earcon]: "*'not today next week?' say 'Yes' or a number on the list.*"
5. User: "*Yes*"
6. System: "*Got it. Message sent.*"

In the example above, even when the recognized output has 40% WER, the system still retrieves relevant reply candidates. Furthermore, it is important to note that because response templates are all corrected for typos and spelling mistakes (as we describe in the next section), users always receive relevant and intelligible reply candidates. They never see the intermediate recognized output of the SLM, which can sometimes produce "word soup", such as line (3). This kind of incorrect recognized output would certainly confuse and distract drivers.

3. Implementation

We implemented a voice search approach to SMS replies for a prototype multimodal automotive infotainment system we are developing at Microsoft Research. The system utilizes a mobile speech engine specifically geared for the low memory footprint requirement of Windows CE 7. Figure 1 displays screenshots of a user interacting with the system. We also refer readers to our accompanying video demonstration.

While voice search provides a framework for natural and robust recognition of SMS replies, the performance of our approach directly depends on the coverage and quality of the index of SMS replies. Without adequate coverage, we forfeit all the benefits of voice search. On the other hand, a balance has to be reached between quantity and quality. Ill-formed SMS replies in the index not only degrade performance, but also produce non-sensible candidates, which can potentially distract drivers. In this section, we describe our SMS data collection, how we generalized SMS replies into templates, and how we perform template matching with recognition results.

3.1. SMS data collection

Due to the asynchronous nature of SMS communication, it is perfectly appropriate not to reply to some text messages right away, especially if the driver is otherwise occupied. We conducted an informal questionnaire to determine what types of SMS messages might provoke a response. In general, we found that messages that invite specification of time and duration, confirmation and acknowledgement, and yes/no answers engendered the most responses. We selected 60 SMS messages such as "*When are we supposed to meet?*", "*I can't go*", "*Can you pick up the kids?*", and "*Are you done?*" We then conducted an online web study sent to the Microsoft employees in which we asked participants to type in text message responses to those 60 SMS messages. From 350 participants, we collected around 14,000 messages in total, comprising 7,500 distinct responses, with around 900 of them occurring two or more times. We manually fixed typos and spelling errors, and also text-normalized any SMS acronyms and slang. In previous studies, we found that proper text normalization can significantly improve voice search performance [17].

3.2. Generalization into templates

In order to boost coverage and improve search relevance, we generalized the SMS responses in the data collection into templates with slots. For example, by abstracting a time slot for the responses "*See you in 10 minutes*" and "*Call you in 5 minutes*", we can generate the response "*See you in 5 minutes*" even though the original time specification was 10 minutes. We abstracted four common types of ***slots***, including **number <D>** (e.g., 15), **time <T>** (e.g., 9:00 PM), **name <N>** (e.g., Michelle), and **place <P>** (e.g., Starbucks). Each of the slots could support rich variations: for example, "at seven o'clock", and "at five thirty P M" are all valid fillers for the time slot.

We used a hierarchical SLM toolkit described in [18] to accommodate slots as regular words in the n-gram building process. Each slot is described in a separate CFG grammar which is linked and expanded by the recognizer on-the-fly. The rules for the number and time slots are referenced from a base grammar library that is available as part of the Microsoft Speech SDK. As for the name and place slots, our prototype maintained limited lists of entries. We plan to populate the lists with personalized information from the user's contact list, GPS points of interest, and favorites in the map and calendar applications. Because the contents of the slots are stored in CFGs, we do not necessarily need to dynamically rebuild the SLM on the device. We are also looking into the possibility of including the messages the user sent (which are stored in the "Sent message" folder) to augment the response templates as a further step toward personalization.

3.3. Template matching

At run time, we obtain both the recognized utterance and the syntactic parsing tree from the recognition result. For example, for the utterance "*five minutes I'll see you at two thirty pm*", the parse tree contains "<**N**>=5(five)", and "<**T**>=2:30PM (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. Note that we discard templates with slots which do not occur in the recognition result. Finally, to obtain a list of candidate replies, we retrieve templates for each recognition result in an n-best list of phrase alternates and then merge and rank the union of the templates according to relevancy, in particular, the TFIDF score.

4. Evaluation

In order to evaluate our voice search approach against both the canned response and the SMS dictation approaches discussed in Section 2, we conducted another study to accumulate a fresh test set of recorded SMS replies. In particular, we recruited 14 participants from the Seattle metropolitan area of diverse occupational backgrounds who all claimed to use SMS text messaging at least twice a week and up to 20 times a day. Participants were compensated with Microsoft software.

In the study, participants received the SMS messages we used in the informal questionnaire described in Section 3.1, including the 60 SMS messages we chose for our web study. They were asked to first type a response to the SMS messages (as they might on their phones), and then speak their typed response into a close-talk microphone. We collected 1200 utterances using this protocol.

4.1. Test set

Because we are primarily interested in allowing users to respond to SMS messages with common, useful replies, we discarded atypical, whimsical responses. For example, for the SMS message "*Are you upset about something*", a participant responded "*Oprah didn't like my book*". We ended up with 1141 replies in our test set.

In order to analyze the coverage of our response templates on the fresh test set, we retrieved matches using our vector space model and the typed responses as queries. We then recruited an independent rater who was unaware of the purpose of our study to judge the relevancy of the top match into the 4 categories listed in Table 1 below.

Category	Examples	Counts
Perfect	"*Sorry, I already ate*" for "*I already ate, sorry*"	766
Good	"*Sure, now is OK*" for "*Sure, now*"	195
OK	"*Don't think so, maybe at 5:00*" for "*5:00, I think*"	85
Miss	"*I'm free for 5 minutes*" for "*free in 5 minutes*"	95

Table 1: Relevancy of response templates

Considering any reply that is judged good and above as "Good", we found that our response templates covered 84% (961/1141) of the test set.

	All SMS Replies (1141 utterances)			SMS Replies Covered by Templates (961)		
	Canned (PCFG)	Dictation (SLM)	Voice Search	Canned (PCFG)	Dictation (SLM)	Voice Search
Top 1	35.0%	59.9%	**61.6%**	41.5%	66.7%	**72.0%**
Top 3	37.0%	69.7%	**73.3%**	43.9%	76.3%	**84.7%**
Top 5	37.2%	72.7%	**78.0%**	44.1%	79.2%	**89.7%**

Table 2: Task completion rates for the canned response, SMS dictation, and voice search approaches for both the entire set of SMS replies and those that are covered by the response templates.

Finally, given that we are developing SMS reply functionality for an automotive infotainment system, we desired to model in-vehicle recognition. As such, using the process described in [19], we convolved the clean replies with a random car impulse response, adjusted the recorded speech to exhibit the Lombard effect, and then mixed in the noise of city-street driving with air conditioning on a low setting.

4.2. Experiment & Results

As our evaluation experiment, we submitted all of the convolved automotive SMS replies to the three approaches described in Section 2. For the canned response approach, we added all of our response templates to a PCFG using the counts of the templates as their rule weights. For the SMS dictation approach, we treated the recognition result of our hierarchical SLM as the final SMS reply (i.e., we left out the IR step). As our performance measure, we had another independent rater judge the reply candidates produced by the three approaches into the categories specified in Table 1. The rater was unaware of how those SMS replies were generated. We again considered any reply judged as good or above as “acceptable” or sufficient for task completion.

Table 2 shows the results of our evaluation experiment. Overall, the voice search approach outperforms the other two approaches with respect to the top 1, top 3, and top 5 reply candidates. Even though the canned response approach has the same template coverage as the voice search approach, it performs the worst because of variations in referring to the template (as discussed in Section 2.3) and because it is less robust to misrecognitions. As predicted, SMS dictation approach did not fare well in producing intelligible reply candidates. Indeed, the fact that the voice search approach consistently beat SMS dictation with respect to all top reply candidates confirms the value of leveraging IR techniques to improve robustness.

Note that the performance of our voice search approach is limited by both coverage and ASR accuracy, especially for the top result which is only 59.9% on the overall test set. However, if we look at just the SMS replies covered by our templates (i.e., the 84% of the data mentioned in Section 4.1) the voice search approach fairs much better, achieving as high as 89.7% task completion when evaluating the top five reply candidates. This evinces the potential of the voice search approach since it is possible to always collect more templates to increase coverage.

5. Conclusions and Future Directions

In this paper, we examined three possible approaches to replying to SMS messages in automobiles and advocated a voice search approach based on template matching. The templates are empirically derived from a SMS corpus and matches are accurately retrieved using a vector space model. Furthermore, because the templates are all corrected for typos and spelling mistakes, users always receive intelligible reply candidates. In our evaluation experiment, the voice search approach consistently outperformed the canned response and SMS dictation approaches, and for SMS replies covered by our templates, it achieved 89.7% task completion with respect to the top five reply candidates.

As future research, we plan to explore whether the voice search approach can also be applied to draft SMS messages, not just SMS replies. We also plan to investigate the value of personalizing the templates.

6. Acknowledgements

We thank Ivan Tashev for synthesizing in-car speech for our evaluation and Ye-Yi Wang, Xiao Li, Bruce Buntschuh and Peter Monaco for many useful discussions.

7. References

[1] R. Stross, “What carriers aren’t eager to tell you about texting”, New York Times, December 26, 2008. Retrieved April 15: http://www.nytimes.com/2008/12/28/business/28digi.html?_r=3

[2] Ford Sync: http://www.syncmyride.com/

[3] A. Kun, T. Paek & Z. Medenica, “The effect of speech interface accuracy on driving performance”, *Proc. of Interspeech*, 2007.

[4] R. Rosenfeld, D. Olsen, & A. Rudnicky, “Universal speech interfaces”, *Interactions* 8(6): 34-44, 2001.

[5] F. Seide, P. Yu, & Y. Shi, “Towards spoken-document retrieval for the enterprise: Approximate word-lattice indexing with text indexers,” *Proc. of ASRU*, 2007.

[6] R. Rosenfeld, “Two decades of statistical language modeling: Where do we go from here?”, *Proc. of the IEEE*, 2000.

[7] Promptu Systems ShoutOut: http://www.promptu.com

[8] T. Paek, B. Thiesson, Y. Ju & B. Lee. “Search Vox: Leveraging multimodal refinement and partial knowledge for mobile voice search”, *Proc. of UIST*, 2008.

[9] B. Suhm, B. Myers, & A. Waibel. “Multimodal error correction for speech user interfaces”, *ACM TOCHI*, 8(1), 60-98, 2001.

[10] Y. Wang, D. Yu, Y.C. Ju, & A. Acero “An introduction to voice search”, *IEEE Signal Processing Magazine*, 2008.

[11] P. Natarajan, R. Prasad, R. Schwartz, & J. Makhoul, “A Scalable Architecture for Directory Assistance Automation”, *Proc. ICASSP*, 21-24, 2002.

[12] S. Mann, A. Berton, & U. Ehrlich, “How to access audio files of large databases using in-car speech dialogue systems,” *Proc. of Interspeech*, 2007.

[13] D. Yu, Y.C. Ju, Y.-Y. Wang, G. Zweig, & A. Acero, “Automated directory assistance system: From theory to practice,” *Proc. of Interspeech*, 2007.

[14] M. Gilbert, J Wilson, B. Stern, & G. Di Fabbrizio, “Intelligent virtual agents for contact center automation,” *IEEE Signal Processing Magazine*, 2005

[15] G. Salton & M. McGill, eds. *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc. 1983

[16] Live Search Mobile: http://livesearchmobile.come

[17] Y.C. Ju, & J. Odell, “A language-modeling approach to inverse text normalization and data cleanup for multimodal voice search applications,” *Proc. of Interspeech*, 2008.

[18] Y.C. Ju, Y. Wang, & A. Acero, “Call Analysis with Classification Using Speech and Non-Speech Features”, *Proc. of ICSLP*, 2006.

[19] I. Tashev, A. Lovitt, & A. Acero. “Unified Framework for Single Channel Speech Enhancement”. *Proc. of 2009 IEEE Pacific Rim*