Speech Technology and Localisation

Roger C.F. Tucker, Marianne Hickey, Nick Haddock[1],
Client and Media Systems Laboratory
HP Laboratories Bristol
HPL-2002-43
March 18th , 2002*

E-mail: roger_tucker@hp.com, marianne_hickey@hp.com, nick@nickhaddock.com

speech-as-data, speech recognition, speech compression, voice dialogue systems

This paper addresses two aspects of speech technology. The first is the use of speech rather than text as the media for information representation in a computer - since speech needs no localization this approach reduces localization barriers and is also simpler for people with low levels of literacy - but to help organize the speech, tools are needed which allow quick and easy access to the speech-as-data records. The second aspect is the voice dialogue system, which allows information access by telephone - for this localization is necessary, and approaches to this are briefly discussed.

Abstract

This paper addresses two aspects of speech technology. The first is the use of speech rather than text as the media for information representation in a computer – since speech needs no localization this approach reduces localization barriers and is also simpler for people with low levels of literacy – but to help organize the speech, tools are needed which allow quick and easy access to the speech-as-data records. The second aspect is the voice dialogue system, which allows information access by telephone – for this localization is necessary, and approaches to this are briefly discussed.

Keywords: Speech-as-data, speech recognition, speech compression, voice dialogue systems

1. Speech-as-Data

Speech is font and keyboard independent. It is independent of literacy or indeed any kind of education or language skills and is the communication medium most used by humans, yet computers make little use of it.

The main reason for this is that internally computers process text, and conversion of speech to and from text is a difficult and error-prone task. Only in the last few years has automatic speech recognition (ASR) technology become usable enough, but even now there are limitations – either the user must train the system (and in effect train themselves), or recognition must be domain-specific and a limited vocabulary employed. Text-to-speech (TTS) synthesis has been available for over 20 years, but only recently has become at all natural sounding. And of course, localisation for such systems is a big task.

Where the use of the speech interface is primarily for entering & obtaining information rather than for control, an alternative to ASR and TTS is for the information to be retained in speech form, so the speech *is* the data. Because there is no transformation between input and output a lot of expressive detail is retained that would be lost in text. There are big drawbacks though. One is that a greater amount of storage is needed for speech compared to text. But the major problem is that browsing and accessing stored speech would appear to be tedious and time-consuming, as anyone who has used today's telephone-based voice messaging systems knows. In fact, this is more to do with the legacy of tape and other continuous media than anything intrinsic in the data type. With appropriate organisation and visual tools, information stored as speech can actually be accessed quickly and easily.

Early speech-as-data work concentrated on audio-only interfaces [1][2] – however, our focus is on standard computing devices with displays. In particular we are interested in low-cost and robust hand-held devices such as the WinCE-based PDAs or the Simputer [3]. These devices are starting to prove popular because they allow internet connectivity anywhere at lower initial cost than a desktop PC. We have a twofold reason to be particularly interested in them as speech-as-data devices – firstly because their small size is far more suited to speech input than keyboard input, and secondly as they may well become the pervasive computing device in the developing world. For people with low literacy, speech-as-data applications whether on a PDA or a desktop PC will enable computer use that otherwise they would be barred from.

2. Speech-as-Data Tools

We are most interested in communications-related applications - e-mail and the associated contacts database – since communication is the driving force behind connectivity. (The techniques we describe can also be applied very usefully to audio content but that is not our focus.) This means of course that the recipient also needs to be able to deal with speech-as-data. They may well be using a conventional PC, so both desktop and hand-held versions of all the tools are needed. As storage capabilities on the desktop are far greater, tools for organisation are all the more important.

2.1 Display/Editing

Figure 1 shows the main display & editing window of a speech-as-data system, based on the interface described in [4]. In this particular case, a 100 second recording is displayed as a series of lines about 8 seconds long.
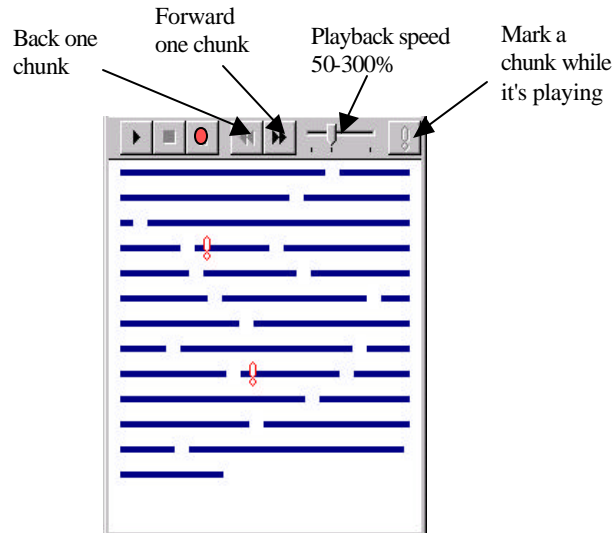

Figure (1): Display Window for Speech-as-Data Records

The key features of the display are:

1. Chunking: the speech record is divided into chunks based on the pauses in the recording. Whilst ideally the chunks would match phrases in the speech, it is equally important that they are a suitable length for skipping and marking. The algorithm creating the chunks [5] has the concept of a target length (about 5 seconds), which it tries to achieve within the constraints of not breaking up a continuous phrase.

2. Random Access: any part of the speech record can be accessed immediately by clicking on that part of the display.

3. Scanning: what is so quick and easy with text is much harder with audio. Two options are available for this, skipping and fast-play. Skipping allows instant movement to the start of the next or previous chunk. Fast play speeds up the playback by up to 300%.

4. Tagging: the ability to mark a place in the speech is very important and, used properly, ensures that most of the speech need only be listened to once. Although the figure shows only one marker, a number of culturally-appropriate markers could be made available for different categories. We have used icons for phone numbers and dates/times to good effect.

The chunk is the basic editing element. Facilities are provided for joining adjacent chunks and splitting a chunk into two, the latter being done on the basis of the longest pause in the chunk. A finer grain editing could be provided, but at the expense of complicating the interface and tools.

2.2 Speech in Applications

Applications supporting speech-as-data need to be able to accept speech in any field as an alternative to text. This may be out of necessity (because of script or literacy problems), convenience (because of the awkwardness of textual input), or choice (because speech is more appropriate than text for that kind of information).

To illustrate how a speech-enabled application might work, we have speech-enabled a contacts manager, see Figure 2. In this example, all the names are entered as text, though they could in principle be voice.
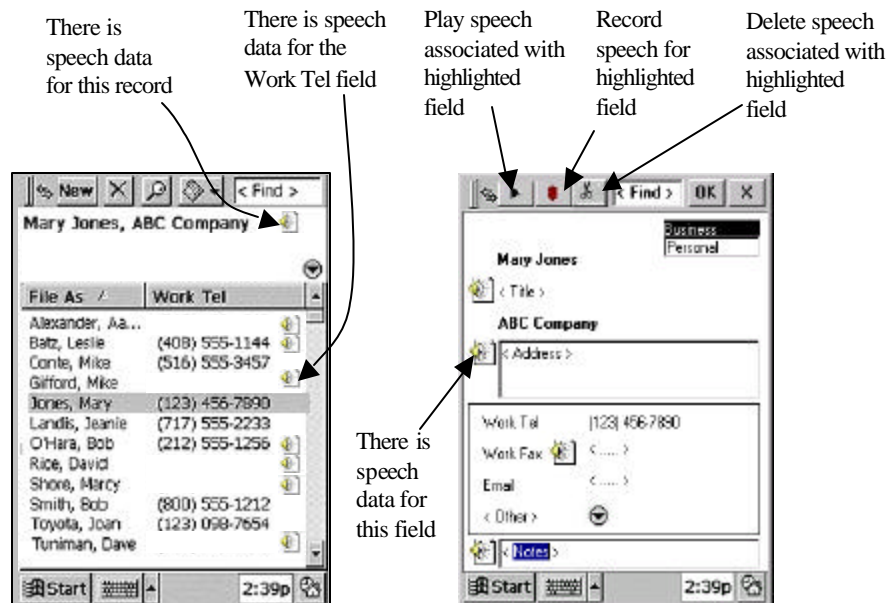

Figure (2): Speech-enabled Contacts Application

Although the name field works much better as text, everything else can quite reasonably be entered and retained as speech. There are four criteria for choosing between speech and text:

1. Do I need this information for visually browsing the records? If so, it is well worth the effort of entering text.

2. Is this information better expressed in speech or text? Personal feelings are more easily conveyed using speech.

3. Is this information that would take a long time to enter by hand e.g. an address or note? Speech saves time now at the expense of possibly more time later.

4. Is this information that I may not need but would like to keep just in case? Fax numbers, addresses and titles are fields in this category. In this case it is easy to read them in and then they are there if ever needed.

2.3 Searching by Voice

2.3.1 Field Recognition

The File As field in Figure 2 has names as text because it would be tedious to have to browse through a lot of speech to find the name of the person you are looking for. However, it is not difficult to provide a facility that will search a field of an application for a particular piece of audio, as long as you assume that the user can exactly specify the contents of the field in the search request. The technology required is speaker-dependent speech recognition, which requires no localisation and is what mobile phones use for their dial-by-voice feature.

2.3.2 General searching

Where a user cannot exactly specify the contents of a field, or they just want to search for a phrase embedded in any field, wordspotting technology can be used. Standard algorithms are not suitable both because the search needs to be specified as speech not text, and because it has to run orders of magnitude faster than real-time. Section 4 of [6] goes into details of ways this can be done. Unfortunately, the

technology needs acoustic models for the language used, and providing these for a local language rather negates the otherwise localisation-free properties of speech-as-data.

When an ASR system *is* available with suitable large vocabulary language models, its output can be used without the need for correction by applying summarisation techniques based on confidence score and inverse word frequency [7]. The text becomes a quick way of locating relevant spoken information and need never be corrected unless the information is needed in textual form.

2.4 Compression of Speech Data

2.4.1 Requirements

For speech-as-data to be viable on a small device, the speech must not take up a lot of memory. There are numerous speech coding schemes, many being published standards, optimised for telephony. These offer a high perceptual quality (i.e. speech must sound natural and like the talker), but can afford to be a little unintelligible as the hearer can always request a repeat. For this reason a restricted 3.4kHz bandwidth has been used for telephony from the early days of frequency-division multiplexing until now.

For speech-as-data the perceptual quality need not be high, but the speech must be as intelligibile as possible. Phone numbers and alphanumeric details must be understood without error otherwise the entire concept fails. At the same time a high degree of compression is needed. The military solution to a similar requirement (high intelligibility, low bit rate) is the LPC10 vocoder [8]. This is a speech coder that relies completely on encoding the parameters of a model of speech production. Advances in parametric coding [9] have substantially improved the quality (perceptual and intelligibility) of the LPC vocoder, and it is now close to the quality needed for speech-as-data whilst keeping a very low bit rate.

2.4.2 A Wideband Low-Bit-Rate Speech Coder

In [10], a speech coder is described based on LPC vocoding that improves intelligibility by extending the bandwidth of the encoded speech to 8kHz. The input signal is split into two bands, and the 0-4kHz band is encoded with a 10th order LPC model, and the 4-8kHz band is encoded with a 2nd order LPC model. The parameters are put through predictors and the residual is encoded using variable-rate Rice coding. Variable-rate coding is a luxury that stored speech can benefit from, which transmitted speech cannot. For extra intelligibility, the frame length is reduced from 22.5ms to 16ms. This reduction in frame length does not increase the bit rate proportionately as the smaller frame size is more predictable. At 2.4kbit/sec, the high-band parameters take up about 500 bit/sec. The interesting result reported in [10] is that when comparing the wideband and narrowband versions with both optimised for 2.4kbit/sec operation, the wideband coder performs 2.4 points on the DRT (Dynamic Rhyme Test [11]) scale better than the narrowband coder. This demonstrates that even at this very low bit rate, bandwidth is more useful than finer quantisation.

2.4.3 Recognition of Compressed Speech

Various studies have shown that recognition performance degrades when recognising from speech compressed below 16kbit/sec [12,13]. For very low bit rate coders the problem is quite serious. The problem can be bypassed by computing and encoding the acoustic features used for recognition separately. These require at least 4kbit/sec storage [14,15,16] and when stored alongside the speech, more than double the storage requirements. In fact the two representations (coded speech and acoustic features) are only slightly different forms of the same information. In [17] a way of deriving the acoustic features *from* the coded speech is described, but as the resulting features are slightly different from those derived directly from the speech, it is important that the recogniser is trained on these features as well.

3. Voice Dialogue Systems

Voice dialogue systems allow access to information and carrying out of transactions using an ordinary phone, an attractive proposition where cost, connectivity or font/keyboard problems prohibit the use of a PC or PDA. A basic capability has been available for many years using recorded voice for output and DTMF for input. A good voice dialogue system is potentially quicker and easier to use, as it uses speech

input instead of DTMF and allows more flexibility in the dialogue through the use of text to speech (TTS) synthesis for the voice output.

Although the basic automatic speech recognition (ASR) technology to achieve these systems has been available for almost 20 years, it is only recently that sufficient robustness has been achieved for successful widespread deployment. This has been achieved as much through collecting large amounts of real data from field trials, as any great improvements in the algorithms. Such is the success that the W3C has defined the W3C Speech Interface Framework [18], a suite of markup languages including VoiceXML 2.0, which allows anyone to design and specify a voice dialogue. The resulting voice application can then be loaded into a voice portal and interfaced at the back-end to any web-based database or service.

3.1 Styles of Dialogue

Dialogues can be system initiated, where the user simply has to answer questions posed by the system, or mixed initiative, where the system can pose much more open questions (eg how can I help you?) and the user can control the direction of the dialogue as well as the system. An example of a mixed initiative system is the Jupiter weather information system from MIT [19]. System initiated dialogues work well when the questions by their nature have a predictable set of answers - otherwise the user has to be told the possible answers, which is little better than a DTMF system. Mixed initiative dialogues by their nature invoke a wide range of responses from the user, and need refining over a long period of time because of this. Both require high performance recognition, except perhaps when the set of possible responses in a system initiated system is very small.

A less demanding use of voice dialogue is in multimodal systems. These use other modalities, usually a Graphical User Interface (GUI) of some sort, as well as the voice dialogue. The two can operate concurrently, allowing hands-free use or voice guidance for naive users, or at different times but with the data and context shared between the two modalities. For instance in the UK it is possible to use a GUI to specify your regular usage of trains for commuting, and then to phone into the system to find out when the next train home is leaving. Very little user input is needed during the phone call, making it implementable with just DTMF. Work is beginning on standards for multimodality and requirements have been published [20].

3.2 Localisation of Speech Engines

Localisation of speech recognition technology is a very big issue in the deployment of voice dialogue systems. Since it is the large quantities of field data that give the robustness to speech recognition a lot of effort is needed to support a new language or even a new dialect. There is quite a focus around the world on reducing this effort through the use of multilingual modelling - this pools data already collected in various languages to provide a strong starting point for any new language. The August 2001 issue of Speech Communication is devoted to this topic [21].

For text to speech, the problem is much less as only one speaker is needed per language, but the design, recording and manual labelling or correction of the speaker database still takes quite a few months. A viable alternative is to use the acoustic database for an existing language, and map the phonemes - the MBROLA project [22] has made this very easy by collating a database of different languages, including Telegu and Hindi.

Common to both ASR and TTS is the need to convert from written words into their constituent phonemes. To do this a combination of phonetic dictionary and letter to sound rules are needed. Whereas for ASR a certain amount of ambiguity can be left in the acoustic models (i.e. two alternative pronunciations can both be modelled), for TTS the correct sounds must be specified if at all possible. So far there is little progress towards speeding up the localisation of this process for TTS. The ability to borrow from existing languages would save time and effort, but TTS systems would need to be compatible and made available for others to use. The open source publication of Edinburgh/CMUs Festival TTS system [23] is a step towards this, as are the EULER project [24] and the Multilingual Toolkit Project [25].

4. Conclusion

In this paper we have described tools that enable speech to be used effectively as a data type by providing means for browsing, organisation, compression and searching of the speech. Making speech a major part of the user interface reduces the burden on localisation - text input/output becomes a convenience rather than a necessity.

For general browsing and editing, we describe a 2D random-access interface, which gives text-like properties to the speech. These properties are phrasing, easy skimming and replay, and the ability to mark important sections with a visual icon for future reference. If the speech can be broken down into short segments only a few seconds long, and associated with the fields of an application, it is easy to find the information and quick to listen to it – such organisation can be achieved by allowing the user to enter speech into any field where there might normally have been text. To avoid using up memory with a lot of stored speech, a low bit-rate compression scheme is described with high intelligibility to allow names, numbers and addresses to be retrieved from the speech and which also works well with speech recognisers. This is achieved with wideband (8kHz) parametric encoding, which requires only 500 bit/sec extra and even at rates as low as 2.4 kbit/sec gives greater intelligibility than narrowband encoding.

We also describe some different approaches to voice dialogue systems – system initiated, mixed initiative and multimodal. The ASR and TTS engines for these do need localisation, and we give some pointers to current work to simplify this process.

5. References

1. Stifleman L J, Arons B, Schmandt C et al. VoiceNotes: A Speech Interface for a Hand-Held Voice Notetaker. In: Proceedings of INTERCHI, April 1993, ACM Press, 1993, pp 179-186
2. Arons B. Interactively Skimming Recorded Speech. PhD Thesis, M.I.T., 1994
3. http://www.simputer.org/
4. Loughran S, Haddock N, Tucker R. Voicemail Visualisation from Prototype to Plug In: Technology Transfer through Component Software, HP Labs Technical Report HPL-96-145 1996. Available from http://www.hpl.hp.com/techreports/index.html
5. Tucker R C F, Collins M J. Speech Segmentation. U.S. Patent 6,055,495. April 2000.
6. Tucker R C F, Hickey M, Haddock N. Speech-as-Data Technologies for Personal Information Devices, HP Labs Technical Report 2002. Available from http://www.hpl.hp.com/techreports/index.html
7. Valenza R, Robinson A.J, Hickey M et al. Summarisation of Spoken Audio through Information Extraction. In: Proceedings of ESCA ETRW Workshop on Accessing Information in Spoken Audio, April 99, pp 111-116
8. Tremain TE. The Government Standard linear predictive Coding Algorithm: LPC10. Speech Technology 1982, pp 40-49
9. McCree AV, Barnwell III TP. A Mixed Excitation LPC Vocoder Model for Low Bit Rate Speech Coding. IEEE Trans. Speech and Audio processing, July 1995, 3:242-250
10. Seymour CW, Robinson AJ. A Low-Bit-Rate Speech Coder using Adaptive Line Spectral Frequency Prediction. In: Proceedings of Eurospeech '97, 1997
11. Voiers WD. Diagnostic Evaluation of Speech Intelligibility. In: Speech Intelligibility and Speaker Recognition, Mones E Hawley, Ed., Dowden, Hutchinson & Ross, Inc., 1977, pp 374-387
12. Euler S, Zinke J. The Influence of Speech Coding Algorithms on Automatic Speech Recognition. In: Proceedings ICASSP'94, Adelaide, IEEE, 1994, I:621-624
13. Lilly BT, Paliwal KK. Effect of Speech Coders on Speech Recognition Performance. In: Proceedings ICSLP'96, pp 2344-2347
14. Ramaswany GN, Gopalakrishnan PS. Compression of Acoustic Features for Speech Recognition in Network Environments. In: Proceedings of ICASSP'98, IEEE, 1998 2:977-980
15. Digalakis V, Neumeyer LG, Perakakis M. Quantization of Cepstral Parameters for Speech recognition over the WWW. In: Proceedings of ICASSP'98, IEEE, 1998 2:989-992
16. Speech Processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms. ETSI ES 201 108 version 1.1.2. April 2000.
17. Tucker R, Robinson AJ, Christie J et al. Recognition-Compatible Speech Compression for Stored Speech. In: Proceedings of ESCA ETRW Workshop on Accessing Information in Spoken Audio, April 1999, pp 69-72
18. http://www.w3.org/Voice/
19. V. Zue, et al., "JUPITER: A Telephone-Based Conversational Interface for Weather Information," IEEE Transactions on Speech and Audio Processing, Vol. 8, No. 1, January 2000.
20. http://www.w3.org/TR/multimodal-reqs
21. Special issue on Multi-lingual Interoperability in Speech Technology, Speech Communication, Vol 35, Nos. 1-2 August 2001
22. http://tcts.fpms.ac.be/synthesis/mbrola.html
23. http://tcts.fpms.ac.be/synthesis/euler/

24. http://festvox.org/
25. http://www.media.mit.edu/cogmac/mltk/multilingual_toolkit.htm