SPEECH PROCESSING

J.R. de Pijper
IPO, Eindhoven, The Netherlands

E. Klabbers
Center for Spoken Language Understanding, OGI School of Science and Engineering, Beaverton, USA

Keywords: Speech synthesis, speech recognition

Contents

Summary

We discuss how speech can be synthesized and recognized. Moreover, we look at several application areas and a complete dialogue system.

1. Introduction

If a human wants to access a system or application, usually a two-way interaction is necessary: The user needs to be able to give information to the system (input) and the system must be able to give information to the user (output). This user-system interaction is achieved through what is usually called the “user interface”. In the early

days of computing, input to a system was done by means of punch cards and output was printed on paper. This is a very indirect way of interacting, with potentially long delays between input and output. Nowadays, we have graphical user interfaces: the most frequently used input devices are the mouse and the keyboard, and the most frequently used output device is the computer screen, even though printed output remains important. Human-system interaction is much more direct and, if enough artificial intelligence is programmed into the system, something very much resembling a true dialog between user and system can take place.

It seems reasonable to maintain that human-system interaction can be made even more direct, and more natural, if *speech* is used for input and output, rather than mouse, keyboard, screen and printer. After all, speech is man's most natural medium of communication and does not need the acquiring of skills such as typing or mouse-screen coordination. Indeed, the use of speech in human-system interaction is becoming more common. For instance, office suites are being equipped with speech capabilities, applications that will read out your e-mail to you are flooding the shareware circuit, telephones are beginning to allow voice dialling, and standards for programmers who want to incorporate the use of speech in their applications are emerging. This trend appears to be solid, and we can expect to see a steady increase of applications, systems and services that use speech in the years to come.

There are at least two reasons for the increasing popularity of speech in the interaction between human and machine. One is the fact that speech technology is improving, the other is that computers are getting more powerful every year, so that this speech technology can be made available to the general public.

1.1 Speech in the user interface

In the interaction between human and system, speech can be used for both input and output. We talk of speech input when the user talks to a system and the system recognizes what the user said. The system's task here is *speech recognition*. Speech output is when the system talks to the user. The system's task here is *speech synthesis*. Later in this article, the techniques underlying speech recognition and speech synthesis will be explored in some detail.

Although it is true to say that speech input and output can make user-system interaction more direct and more natural, it is important to realize that speech is not always the best or most efficient way of achieving the goal. One reason for this is that neither speech synthesis nor speech recognition are as yet perfect: Synthetic speech is still less understandable than natural speech and speech recognition has not yet reached the point where recognition is perfect. This means that misunderstandings between user and system will inevitably occur and that the system will have to take this possibility into account. As a result, any dialog between human and system will suffer. Another reason is that sometimes speech is not the most effective medium to present information. For instance, some types of information are much more efficiently presented as a table, graph or picture.

Most systems that use speech for input and/or output are in fact *multimodal* systems, in which speech is just one modality and is combined with more conventional modalities such as keyboard and mouse input and screen output. The system may decide whether to use speech or some other modality, or it may offer the use of speech as an option to the user.

Sometimes, speech is the only available option. This is especially the case when the user contacts a system, e.g., an information retrieval system, by telephone: the only way in which the system can provide output to the user is by means of speech, though input to the system can be done through both speech and the telephone keys.

In some situations, hands and/or eyes are needed for other tasks, such as driving a car or operating a machine. In this case, speech is preferred for reasons of safety. For instance, it is safer for a car driver to dial a telephone number by speaking it (or a name representing it) into a microphone than by actually pushing buttons, which would require the driver to look at the telephone keypad and take one hand off the steering wheel.

People with certain kinds of physical disabilities can often benefit enormously from the use of speech. The blind are perhaps the most obvious example: they cannot read printed or graphical output and heavily depend on spoken output. On the input side, some people have disabilities that prevent them from using the keyboard or the mouse. For these, it is very useful if a system has speech input functionality built into it.

On the other hand, sometimes the use of speech in human-system interaction is counterproductive, in the sense that a task may be completed more efficiently using another modality. A proficient typist will probably finish a letter more quickly by simply typing it than by using a dictation system, given the current state of speech recognition technology. In the case of speech output, it is important to realize that speech is transient in nature. Anything that is output to the screen stays there and can be inspected at leisure, but what the system says must be remembered by the user. Given the limited capabilities of man's short-term memory, this implies that speech is not suitable to present even moderately large amounts of data.

This article will first discuss the technical aspects of speech synthesis and speech recognition, the two sides of the speech technology coin. After that, several application areas will be discussed where speech can be used to advantage. Finally, speech input and output will be placed in the context of a complete dialogue system.

2. Speech synthesis

2.1 Introduction

Speech can be generated by a machine in a variety of ways. If the number of messages to be spoken by a system is truly limited, it can be as simple as playing back prerecorded utterances at the appropriate times. This results in perfectly natural speech, but requires a relatively large amount of memory. This, however, is not really synthesis, since the system does not generate the utterances, but just plays them back.

We can therefore define speech synthesis as the process whereby a machine can generate spoken utterances that were not stored in memory as such. Nearly all systems that support speech output must be able to generate more messages than can be stored in memory as prerecorded utterances, either because of memory limitations, or because it is not known beforehand what utterances need to be generated. In this case, some form of speech synthesis is called for.

Very roughly, two types of speech synthesis systems can be distinguished: parameterized systems and concatenative systems. Parameterized systems generate speech completely from scratch, while concatenative systems work by concatenating prerecorded bits of speech, large or small. Parameterized speech is the more flexible, since every speech parameter can be controlled. The best known synthesis system of this type is DecTalk. However, it is extremely difficult to design a parameterized speech synthesis system whose speech quality approaches that of natural speech. With concatenative systems, a better speech quality can be achieved, at the expense of some flexibility. For this reason, most speech-capable applications employ some form of concatenative synthesis, and this article will concentrate on that.

We make a distinction here between three types of concatenative speech synthesis: diphone concatenation, phrase concatenation and unit concatenation. These are discussed in the following sections.

2.2 Diphone concatenation

In a language such as English, about 50 different basic speech sounds, so-called “phonemes”, can be distinguished. All words, phrases and utterances in a language are formed by different sequences of these phonemes, much as in written language all possible sentences are formed by different sequences of only a small number of letters. For instance, the word “seen” is made up of 3 phonemes and can be phonetically represented as / s i n /. It differs from the word “soon” only in the second phoneme: / s u n /.

It is a natural thought that it should be possible to synthesize speech by prerecording instances of every phoneme in a language and then concatenating them as needed to generate an utterance. The word “seen” could then be synthesized simply by retrieving the recordings for the three phonemes /s/, /i/ and /n/ and playing them back. This is also an attractive idea because the number of phonemes in a language is small and a phoneme instance typically has a duration of only 50 to 250 ms. Thus, storage and retrieval on a computer system do not present any problems.

Unfortunately, it does not work this way: Speech generated by phoneme concatenation is at best highly unnatural and at worst completely unintelligible. The reason for this is that speech is a dynamic phenomenon: the articulatory organs like mouth, tongue and vocal chords do not “snap” from one phoneme position to the next, but rather change shape continuously from one target position to another: they are in a continual state of transition. In consequence, the /s/ phoneme in “seen” is clearly different acoustically from the /s/ phoneme in “soon”, because during the articulation of /s/ the mouth is already preparing for the articulation of the following phoneme.

This has given rise to the notion of the "diphone": A diphone is a unit of speech that consists of the second half of one phoneme and the first half of the next. In terms of diphones, the word "seen" can be represented as / #-s s-i i-n n-# /, where "#" indicates silence. In other words, the word "seen" can be built up of 4 diphones: silence to /s/, /s/ to /i/, /i/ to /n/ and /n/ to silence. The technique of recording diphones for all possible combinations of phonemes and then concatenating them as required to synthesize utterances is known as "diphone synthesis". With this technique it is possible to generate synthetic speech of relatively high quality. Diphone speech is highly intelligible, although still clearly distinguishable from natural speech.

The reason why the diphone as a concatenation unit is more successful than the phoneme is that, since a diphone contains the transition between two phonemes, much of the dynamics of speech is coded in the diphones themselves. When two diphones are concatenated, the "weld" between the two is positioned in an acoustically relatively stable part of the speech wave. In the "seen" versus "soon" example mentioned above, the differences between the two instances of the /s/ phoneme are recorded in the diphones themselves: The word "seen" uses the /s-i/ diphone, whereas "soon" uses the /s-u/ diphone.

On the downside, to record all possible transitions between 50 phonemes, roughly 50 * 50 = 2500 diphones are needed. However, since the size of a diphone is the same as the size of a phoneme, storage requirements are still quite modest: A high-quality diphone database will typically require less than 10 MB of disk space; if a lower quality is acceptable, only 5 MB are needed.

To generate diphone speech, it is not sufficient to just retrieve and concatenate diphones. In the resulting string of diphones each phoneme would have an arbitrary duration, giving rise to a haphazard rhythm, and the resulting speech would have no intonation, it would be monotone. Such speech is barely understandable and completely unnatural. Therefore, a set of rules has to applied that will calculate correct durations for each phoneme and an appropriate pitch contour for the utterance as a whole.

This complex of rhythmic and intonational features is called the "prosody" of an utterance. Prosody serves numerous functions in human speech. A single word, e.g., the word "yes", can be pronounced in many different ways, to convey determination, surprise, doubt, or any of a whole range of emotions. This is the use of prosody to convey "mood" or "emotion" or the "attitude" of the speaker; it may add a whole new layer of meaning to the literal meaning of an utterance. Prosody is also used to highlight the *structure* of utterances. Important syntactic boundaries may be accompanied by a steeply rising pitch movement and a pause, to give just one example.

This is not the place to go into form and function of prosody in speech. The point is that a synthetic speech utterance which is rendered perfectly on the segmental level will still sound very unnatural if the prosodic realization is faulty.

-
-
-

Bibliography

F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, ISBN: 0-262-10066-5, 1998. [Reference for hidden markov models]

J. Kessens, M. Wester and H. Strik, Improving the performance of a Dutch CSR by modeling within-word and cross-word pronunciation variation, *Special issue of Speech Communication on 'Modeling Pronunciation Variation for Automatic Speech Recognition'*, Vol. 29, No. 2-4, pp. 193-207, 1999. [Describes the OVIS system]

Biographical Sketches

Esther Klabbers received the M.A. degree in language and computer science in 1995 from the Department of Language and Speech, University of Nijmegen, the Netherlands. In 2000, she received the Ph.D. degree from the Eindhoven University of Technology, Eindhoven, the Netherlands, where her dissertation was "Segmental and Prosodic Improvements to Speech Generation". She then continued as a Postdoctoral Researcher with the Spoken Language Interface Programme, IPO, Center for User-System Interaction, Eindhoven, the Netherlands. She is currently a Postdoctoral Researcher at the Center for Spoken Language Understanding of the Oregon Graduate Institute of Science and Technology in Beaverton, Oregon, the United States. Her expertise involves many aspects of speech synthesis, such as prosodic modeling, corpus design, quality evaluation, and applications.

Jan Roelof de Pijper studied English Language and Literature at Utrecht University, The Netherlands, where I received my Master's degree in 1976. After teaching English for a year at a secondary school, he did a PhD project at the IPO, then still Institute for Perception Research, from 1976-1980. From 1980-1984 he again worked as a secondary-school teacher. In 1983, he received a PhD degree from Utrecht University on a thesis entitled *Modelling British English Intonation: An analysis-by-resynthesis of British English intonation*. In 1984, he accepted a position as assistant professor, again at the IPO, now IPO, Center for User-System Interaction, where he still works today. The larger part of his scientific career has been devoted to the study of prosodic phenomena in speech, with an emphasis on models of intonation and on phrasing, i.e., how prosody is used to mark boundaries in speech. Later, the focus shifted to implementing and improving the speech output component of applications, in particular automatic information providing systems. In this context, he has worked with word concatenation in combination with prosodic postprocessing, phrase concatenation, and diphone-based speech synthesis.