ADAPTING A TTS SYSTEM TO A READING MACHINE FOR THE BLIND

Thomas Portele
Jürgen Krämer

Institut für Kommunikationsforschung und Phonetik
Universität Bonn
email: tpo@ikp.uni-bonn.de

ABSTRACT

Synthesis systems that convert orthographic text into speech usually make assumptions about the input that are no longer valid when used in combination with a scanner and OCR software.This paper describes our experience of adapting our TTS system for use use in such a reading machine.

As synthesis systems move from the laboratory to applications, some surprises may be in store. This paper is intended to share our experiences with other developers in order to prevent them from repeating our mistakes.

1. MOTIVATION

Speech synthesis systems have reached a certain level of quality; their intelligibility is close to human voices but synthesized utterances lack the liveliness found in natural speech. However, for some applications intelligibility is the crucial factor. One of the earliest and most important applications of speech synthesis systems is in a reading machine for blind people (Kurzweil, 1976; Fellbaum, 1996).

Such a machine usually consists of a scanner, text recognition software, and a speech synthesizer. The piece of paper to be read is scanned, transformed into text, and the text is synthesized. One major problem is caused by the speech synthesis being at the end of a sequence of processes and, therefore, it must be able to deal with erroneous input.

Other demands by blind users include moving through the text, changing global prosodic parameters, spelling out words, reading punctuation marks, and changing lexica and abbreviation lists.

2. DEALING WITH OBSCURE INPUT

The combination of a scanner and OCR software might sometimes lead to obscure input for the synthesis system; this is especially the case when scanning faxes. In our experience nearly everything is possible. Figure 1 shows some typical input; the original was a telephone bill.

Speech synthesis systems, on the other hand, are usually developed in a laboratory under the assumtion of optimal input. This not only excludes misrecognized characters but also typing errors, grammatically malformed sentences etc. Thus, the system might not be robust enough for real-world applications.

Our system (Portele et al., 1994) crashed when it encountered words that were 200 characters long and sentences with more than 300 words. In order to avoid such effects, exceptionally long words are split into shorter words, and exceptionally long sentences are split into shorter sentences. The basic assumption is that in these cases an erroneous input will not be made worse by such a treatment.

```
 r.
E05b8095011 bis
 E0568095121 für Titel 51310
Pos. 4 Institut für Kom nikationsforschun Q
und' Poppelsdorfer A ee 47
Abrechnungsmonat:01l95 bi5
12/95 Kommunikationsforschung und Phonetik
Abrechnungsdatum:29,04.96
Für e Zeitraum von Ol/95 bis 12/95 wuren Ihnen
die nachstehend
auf9efuhrten privaten Telefonre hnungen
ubersandt, 8is zum
29.04.9b sind hierzu fognde Zahlungen
eingegangen. Rechnungs-Nr
vom Rechnungsbetraq t DM) einezahlter Betra Q
t DM) E058095011
10.fl2.95 13.57 13.57 E0568095021 1.5.03.95
13.57 i3.57 ED568095031
11.04.95 2.76 0.00 E0568095fl41 10.05.95 1.38
0.00 E58095051
03.07.95 1.84 0.00 E0568095061 1807.5 1.84
0.00 E05b8095fl71
22.08.95 3.22 E0568095fl81 12.09.95 O.9 O.OD
E0568095091 18.10.95
2.07 0.00 E5809511 16.11.95 3.68 D.00 E
5b8fl95111 11.12.95 1.84
fl.Dfl EOSbBO95121 1601.96 l.bl 0,00
Somit steht z.Zt, noch ein etraq von 20.93DM
offen.Sie werden
qebeten,diesen Btra innerhalb von 7 Ta en bei
der Univrsitatskasse
einzuzahlen.Bei ▪berwelsungen unbedint ie
Rechnungsnummer(n) angeben.
```

Figure 1: Typical input for the synthesis system generated by the OCR software. The original was a telephone bill.

The phonemic transcription and the synthesis algorithms must also deal with such an input. This can be done by default phonemes for each grapheme. In concatenative systems, the unit selection algorithm must be able to synthesize every phoneme combination regardless of linguistic limitations (Portele et al., 1994a).

To test the system we used input files with randomly generated ASCII characters. First, the system was evaluated automatically to check its robustness; after successfully passing this test it was checked by listening to its output in order to verify that every readable input was synthesized appropriately. This test was very extensive and revealed a lot of built-in hidden assumptions about the input that had to be eliminated in order to guarantee the system's robustness.

3. PROSODIC PARAMETERS

Most blind users want to change global prosodic parameters according to their needs. While some users browse through the text using a very high speaking rate, others want to listen carefully to the synthetic speech at a rather slow speaking rate. The system must provide speaking rates ranging from less than half the standard rate to at least three times the normal rate (measured in syllables/second).

These changes must not result in unintellegible speech. Nonlinear modifications are necessary: steady states are changed by a larger extent than transitions in order to guarantee intellegibility for fast speech. Plosive bursts must be preserved. Therefore the dynamic and static parts of each synthesis unit must be indicated.

We use a parameter "dynamics" that is computed by comparing a frame (which in voiced segments is one pitch period, and in unvoiced segments is 15 ms) with adjacent frames. The euclidean distance between two mel-cepstrum representations serves as a parameter value; it is transformed to fit into the range between 0 (maximum change) and 2 (no difference to both neighbours). This value is multiplied with the original timing factor generated by the system's prosody control for each frame.

Global parameters of the intonation contour must be adapted to the speaking rate. For faster speaking rates the F_0 range must be decreased in order to to avoid too rapid changes.

Additionally, German health insurance companies demand the ability to change the "liveliness" of the synthetic speech. In our system this is achieved solely by changing the F_0 range. Using a superpositional F_0 control might lead to some difficulties because several parameters have to be adjusted (e.g. in Möbius' (Möbius, 1995) interpretation of Fujisaki's model (Fujisaki, 1988) phrase amplitude and accent amplitude have to be changed); although if the parameters are changed in a consistent way the resulting speech quality might be superior to our solution. In our system a quasilinear approach (Heuft & Portele, 1996) is used.

4. WORDS IN AN UNTTERANCE

A common feature in reading machines is spelling; a keystroke from the user causes the machine to spell out the word just spoken. This implies that the system can react to keystrokes, can determine which word was being spoken at the time, and has access to its orthography in order to produce the correct spelling.

In multitasking systems the first problem, reacting on a keystroke with minimal delay, is easy to achieve; in systems like MS-Windows the synthesis system must periodically check whether a keystroke has happened, and must react. In our system the address of a user-defined function is passed to the synthesis module. The synthesis module calls this function during the synthesis process; the function has to check for keystrokes and has to generate an appropriate response. Among these responses must be the termination of the synthesis process, and this must be handled by the synthesis module.

Most high-quality systems work as utterance synthesizers. Utterances are synthesized as a whole and not as single words in order to simulate human speech as an overlap of articulatory gestures extending over word boundaries. While utterances with pauses between the words are usually better recognized by automatic speech recognition systems, human listeners have more difficulties.

In our system an utterance is synthesized whilst the audio device is playing the audio file from the previous utterance. The durational structure of the previous utterance must be retained in order to allow the determination of the word just played should a key be pressed. This implies that the sound duration module has information of the position of the word boundaries in the utterance.

As soon as an appropriate keystroke is encountered, the current position in the audio file is determined; from this information the current word is computed using the information from the sound duration module.

Two operations can now be performed: the synthesis of the word before the word played at the time of the keystroke, or the spelling of the word. The synthesis of single words from a given utterance is done with a reduced speaking rate. There is the problem of whether to choose the intonation as if the word is still embedded in an utterance, or with a continuation rise, or with a final fall. For simplicity we synthesize single words as one-word utterances but this choice is not based on any empirical data.

While only the phonetic description is needed for the repetition of one word, the spelling of a word implies knowledge of its orthography. In order to avoid reprocessing, the orthographic and phonemic description of all previous words must be kept (especially if the repetition of whole sentences and paragraphs is required). Although this is manageable with a certain overhead, the system will be tied to one particular application. In our latest version we decided to leave all the handling to the user interface in order to maintain portability and interface simplicity (Table 1). In our experience, the benefits of avoiding repeated computation do not outweigh the problem of increased storage overhead. If, however, the symbolic analysis takes a large part of the computing time it might be necessary to avoid recomputation.

5. READING PUNCTUATION MARKS

To serve as a reading machine, the synthesis system must be able to optionally read out punctuation marks as "colon" or "question mark", for example. This might be problematic if a reanalysis is to be avoided because the option might be changed by the user between the first analysis and the repeat. Our first solution avoided reanalysis whenever possible, and we were forced to transform punctuation marks into text and to mark these text parts appropriately in order to leave them out in case the option is turned off.

If reanalysis is allowed the problem can easily be handled by the user interface, e.g. by a flag in the call to the synthesis module. This was another reason for allowing reanalysis and reprocessing in order to gain simplicity.

However, when reading out punctuation marks, it is not clear how to integrate them into the intonation pattern. One can either treat them as utterances on their own, as prosodic phrases on their own, or as words just added at the end of the utterance. The last solution is ineffective because no separation between content information and meta-information is provided. The first solution does not work with

commas, and, additionally, does not reflect the close relationship between the utterance and its punctuation marks.

Open(Voice,Abbreviationlist)
Synthesize(InputText,ResultSoundFile,ReadPunctuaionMarks)
Spell(InputText,ResultingSoundFile)
AddAbbreviation(Abbreviation,FullForm)
ChangeAbbreviationList(Abbreviationlist)
AddLexicalItem(Item)
ChangeSpeechRate(Value)
ChangeF0Range(Value)
Close()

Table 1: Interface functions supported by our system for the application as a reading machine.

Punctuation marks, therefore, were integrated into the utterance but treated as prosodic phrases on their own. This works with commas, because because no utterance-final intonation pattern is involved. It also works with punctuation marks implying a terminal intonation pattern, like exclamation marks.

Qquestion marks are more difficult. The words “question mark” constitute the very last prosodic phrase of an utterance; they are supposed to carry the intonation appropriate for a question. But that sounds “as if the computer is not sure about what it says” as one user stated. If the utterance is terminated like a statement and the phrase before the words "question mark" is intonated as non-final, not much indication about sentence modality can be gathered from the intonation contour. If the question intonation is shifted to the phrase before the words “question mark”, the phrase sounds unnatural. The best solution to this dilemma has shown to be a non-final intonation pattern (continuation rise) for the final phrase of the question, and a terminal intonation pattern for the phrase “question mark”.

6. CHANGING THE LEXICON

In many applications standard solutions for lexical items, symbols and abbreviations are not sufficient; this is especially true for reading machines for the visually handicapped because of the large variety of users. Every user lives in his own social environment, with its own abbreviations, lexical items (e.g. names) and symbol interpretations. And every user faces many different situations. Instead of implementing a “one size fits all” solution it is usually better to let the user choose the way symbols are handled (the ‘-’ sign ought to be ommited in a phone book whereas its ommission in a bank statement is a severe mistake), abbreviations are interpreted (in a medical document “TBC” might mean something different than in an announcement) and words are pronounced.

A user, therefore, demands the ability to choose between different “styles”. The system must change its internal tables accordingly. An appropriate interface must be present.

Additionally, users might want to add or change some lexical items. This must be handled by the synthesis system. In our case the change or the addition of an abbreviation is rather easy because no implicit order is assumed in these tables. The addition of a lexical item, however, is more difficult because our lexicon is ordered alphabetically, and the search strategies assume such an ordering (Portele & Krämer, 1995). Therefore, a user-definable lexicon was implemented. A new lexical item is entered syllable by syllable and transcribed by our rule-based transcription system; then it is synthesized and the user can refine the transcription. This is an interactive process with trial and error because many users are not familiar with the phonetic alphabet. This must be done by the synthesis system because the phonetic representation is an integral part of the system and not of the user interface.

7. CONCLUSIONS

Laboratory systems developed with laboratory material under laboratory conditions will have problems when applied to “real” data because of erroneous input and user demands not met by the usual laboratory features.

Speech synthesis systems are developed with the main focus on speech quality (which, of course, is the most important feature). But ease of use, flexibility and robustness are as important for many users as high-quality speech.

In our experience a clear separation between the user interface and the synthesis module is necessary in order to maintain flexibility and robustness. If the extra cost of reanalysis and reprocessing is comparably low it should be allowed in the interest of simplicity. If the synthesis system is changed to handle spelling, word-by-word synthesis, and similar features, a certain overhead is imposed. Tables must be retained and additional information about word boundaries must be present.

The system must be adapted to support the following features:

1. very high and very low speaking rates
2. information about word positions in the resulting speech file
3. reading punctuation
4. choosing and changing abbreviation lists and lexical items

Other features, like different voices, are desirable but not crucial for the application in a reading machine.

Acknowledgements: Thanks to Simon King for valuable comments on content and form, and to all the visually hadicapped users who suffered under our first versions.

8. REFERENCES

Fellbaum, K. “Einsatz der Sprachsynthese im Behindertenbereich”, (to appear in: *Fortschritte der Akustik - DAGA 96*, Bonn, 1996)

Fujisaki, H. (1988) “A Note on the Physiological and Physical Basis for the Phrase and Accent Components in the Voice Fundamental

Frequency Contour", in *Vocal Physiology: Voice Production, Mechanisms and Functions*, ed. by O. Fujimura, Raven, New York, 347-355, 1988

Heuft, B.; Portele, T. (1996) "Synthesizing Prosody- a Prominence-Based Approach" (this volume)

Kurzweil, R. (1976) "The Kurzweil Reading Machine: A Technical Overview", in *Science, Technology and the Handicapped*, ed. by M.R. Redden and W. Schwandt, 3-11, 1976

Möbius, B. (1995) "Components of a Quantitative Model of German Intonation", *Proc. ICPhS 95*, Stockholm, 108-115, 1995

Portele, T.; Heuft, B.; Höfer, F.; Meyer, H.; Hess, W. (1994) "A New High Quality Speech Synthesis System for German", in *Progress and Prospects of Speech Research and Technology - CRIM/ FORWISS Workshop*, ed. By H. Niemann, R. de Mori, G. Hanrieder, Munich, 274-277, 1994

Portele, T.; Höfer, F.; Hess, W. (1994a) "Structure and Representation of an Inventory for German Speech Synthesis", *Proc. ICSLP'94*, Yokohama, 1759-1762, 1994

Portele, T.; Krämer, J. (1995) "Symbolverarbeitung im Sprach-synthesesystem HADIFIX", *Elektronische Sprachsignal-verarbeitung VI*, Wolfenbüttel, 97-104, 1995