ISCA Archive
http://www.isca-speech.org/archive

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

ARTIC: A NEW CZECH TEXT-TO-SPEECH SYSTEM USING STATISTICAL APPROACH TO SPEECH SEGMENT DATABASE CONSTRUCTION

Jindřich Matoušek and Josef Psutka

University of West Bohemia, Department of Cybernetics,
Univerzitní 22, 306 14 Plzeň, Czech Republic,
jmatouse@kky.zcu.cz, psutka@kky.zcu.cz

ABSTRACT

This paper presents ARTIC[1], a brand-new Czech text-to-speech (TTS) system. ARTIC (ARtificial Talker In Czech) is a concatenation-based system that consists of three main, relatively independent, components: speech segment database, text analyzer and speech synthesizer. A statistical approach to speech segment database construction is used: Hidden Markov models are employed to model triphones on the basis of the large speech corpus and to segment the corpus into triphone-based speech units – basic speech units used by the synthesizer. A speech segment selection algorithm is described to choose the representative instance of each speech unit from the segmented speech corpus. A text processing module converts the written text at the input of TTS system to the sequence of phones – basic phonetic units needed to describe the pronunciation of the input text – and prosodic marks. Finally, speech processing is performed using two versions of a PSOLA algorithm.

1. INTRODUCTION

In this paper, we describe ARTIC, a brand-new Czech TTS system. The system is concatenation-based (see Figure 1), so speech segment database (SSD) must be constructed before the synthesis itself. In contrast to many working systems today, we use triphone-based speech units. A statistical approach is employed to find these units from natural speech corpus and to select the appropriate representatives to be used during speech synthesis.

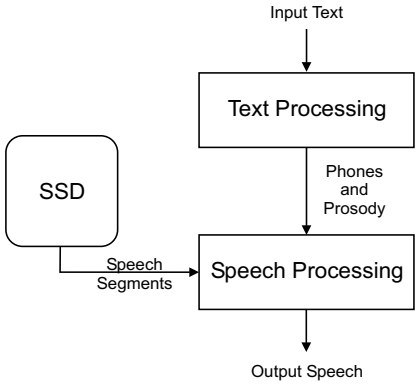

Figure 1: Simplified scheme of our concatenation-based TTS system.

[1] This work was supported by the Grant Agency of Czech Republic no. 102/96/K087 and by the Ministry of Education of Czech Republic FRVŠ project no. F451.

The aim of a TTS system is to generate speech at the output of the system from the arbitrary input text. The first task is then to perform the analysis of the input text. This process is called text processing.

The last but very important task of a TTS system is to generate speech. Since concatenation-based approach to speech synthesis is used in our system, PSOLA-like methods how to concatenate units smoothly are proposed and described.

The paper is organized as follows. Section 2 describes the process of speech segment database construction from more detailed point of view. In Section 3 text processing is depicted. Section 4 then concerns speech processing. Finally, Section 5 contains the conclusion and outlines our future work.

2. SPEECH SEGMENT DATABASE CONSTRUCTION

Hidden Markov models (HMMs) are used as a statistic tool both for speech unit modeling and for an automatic segmentation of a speech corpus [1]. The main advantage of this approach is that labor-intensive and time-consuming manual work is reduced to minimum. Moreover, comparing to traditionally used diphones, more precise speech units – triphones – can be employed. Triphone HMMs are then trained on the basis of the speech corpus. The process of a SSD construction can be described in several steps (see Figure 2) and will be discussed in the next subsections.

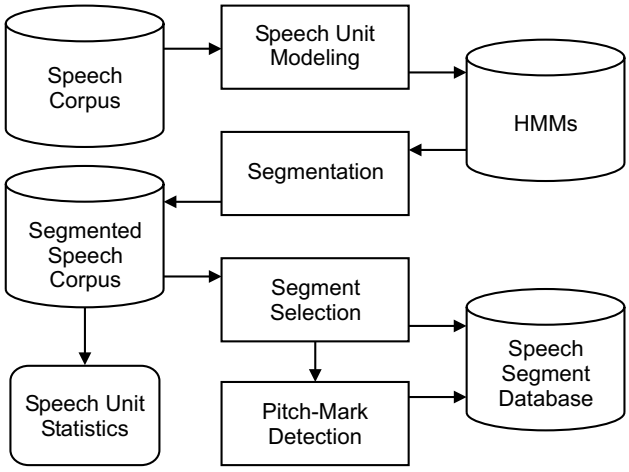

Figure 2: Scheme of speech segment database construction process.

2.1 Speech Corpus Construction

In our system we use a large single-speaker continuous-read-speech corpus (SC). The SC comprises about 90 minutes of naturally read speech recorded from four radio tales. The speech was sampled with 44 kHz and quantized into 16 bits. Then it was listened to, divided into utterances and annotated (i.e. transcribed at a word level). For the purposes of HMM training (see Section 2.2) each utterance was resampled with 16 kHz, Hamming windowed with 25 ms and pitch-asynchronously parameterized using 39 mel frequency cepstral coefficients (including energy, delta and delta-delta acceleration coefficients). As the last step, each utterance was transcribed phonetically using Czech phonetic transcription rules (see Section 3.2).

2.2 Speech Unit Modeling

The modeling of speech units was performed in a similar way as in [2]. Three state left-to-right HMMs with no skips and with 6 ms frame rate were used to model triphones (context-dependent phones). Starting from 45 Czech phones, 8.540 triphone models (which were seen in the SC), including crossword ones, were trained using the HTK system [6]. These models consist of 25.617 generally different states. Ideally, all possible Czech triphones should occur often enough in the SC to be modeled reasonably by HMMs. Taking into account a great number of triphones (91.125 theoretically possible for selected Czech Phonetic Alphabet [8]), it is practically impossible to create such a SC. Hence, the SC described above (which is thought to be sufficiently representative) is used, and an efficient algorithm is employed to cluster acoustically similar sub-phone units (so called clustered states [1] – i.e. units that correspond to states of a tied triphone HMM) using binary decision trees [4]. Thanks to this technique, triphone HMMs are more robust and moreover, triphones not present in the SC can be made up of appropriate clustered states. After clustering, the number of triphones increased to 16.176. These triphones (so called tied triphones) cover all possible speech contexts (including those not seen in the SC) and share only 7.742 clustered states (see Table 1). The clustered states represent the basic speech units of our TTS system.

Units / Clustering	Before clustering	After clustering
Triphones	8,540	16,176
States	25,617	7,742

Table 1: Number of triphones and states before and after tree-based state clustering.

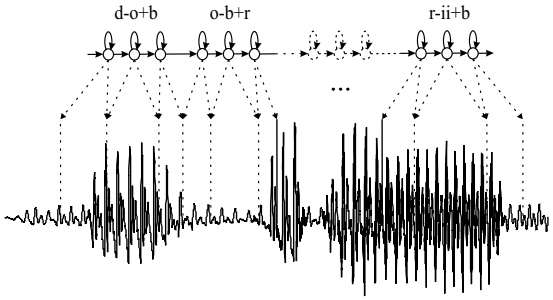

Figure 3: Example of automatic speech segmentation into clustered states.

2.3 Automatic Segmentation of a Speech Corpus

In this step Viterbi algorithm [7] is employed to automatically segment SC into clustered states. Figure 3 shows the example of this segmentation. As a result, time labels describing the boundaries of each speech unit in an utterance are added to the SC.

2.4 Speech Segment Selection

After segmenting, there are multiple instances of each speech unit in the SC. In our system, each speech unit is represented by a single speech segment. A simple segment selection algorithm (similar to [1]) is adopted to choose suitable segments. For the later use in a PSOLA algorithm both too short and low-energy instances are undesirable, because they can cause a degradation of the synthetic speech. Hence, statistics about duration and energy of each speech unit were computed over all instances of given speech unit in the SC. Segment selection runs in three steps:

1. Segments with duration lower than corresponding median value (multiplied with some threshold) are excluded from the selection.

2. Similarly, segments with energy lower than corresponding median value (multiplied with some threshold) are not selected.

3. From resulting instances, segment with maximum HMM score imposed by the Viterbi search is chosen as a representative and is stored to the SSD.

Median values were used rather than mean values, because using mean values in some cases (when only few instances were available for particular speech unit in the SC and one of them has duration or energy much more different from the others) led to selecting an incorrect segment and to synthesizing the speech unit with incorrect duration or energy (see Section 4).

To be able to perform a PSOLA algorithm, each speech segment, selected as the representative of a speech unit, was also a subject of a pitch-mark detection algorithm, which computed the positions of pitch-marks (i.e. moments of glottal closures) in the speech signal. The information about pitch-mark positions was added to the SSD.

3. TEXT PROCESSING

A text is the only input of the TTS system. Hence, an arbitrary text should be processed thoroughly to obtain important knowledge about what to synthesize. A text analyzer is a significant part of a TTS system, because it converts the written form of speech to the phonetic form (i.e. gathers the information about *what* should be synthesized) and generates prosodic features of the resulted speech (i.e. described *how* should be the output speech synthesized).

Firstly, an input text should be pre-processed, i.e. some operations must be performed such as sentence end detection and transcription of digits, abbreviations, etc. This is not the

case of our system in the time of writing this paper, as we suppose that only "clean" text can appear at the input of our TTS system. Then ideally, the "clean" text should be subject of syntactic, semantic, morphological, and contextual analysis to get important phonetic and prosodic information. Since our system generates monotonous speech so far, no prosody analysis is required except of simple pause detection analysis (see Section 3.1). So the key task of the text analyzer consists of converting the input text to the sequence of phones – the basic phonetic units needed to catch the pronunciation of the synthesized utterance. This process is called phonetic transcription (letter-to-phone conversion – see Section 3.2 for more details).

3.1 Pause Detection

Although the resulted speech has been thought to be monotonous so far, the intelligibility and naturalness of the synthetic speech will be enhanced if pauses are included. A very simple pause detection algorithm has been incorporated into the system up to now. It searches for the punctuation marks in the input text and replaces them with the symbol of pause [P] (see Example 1).

Example 1: Pause detection.

Text :	▫▫▫▫ě▫▫▫ rádi vzpomínají na doby, kdy byli ▫▫▫▫ě▫▫ě▫▫▫▫
Pauses:	[P] ▫▫▫▫ě▫▫▫ rádi vzpomínají na doby [P] kdy byli ▫▫▫▫ě▫▫ě▫▫▫▫▫▫▫

Obviously, this is a very simplified approach to a pause detection problem. In practice, not all punctuation marks are associated with pauses. Moreover, pauses can appear between words where no punctuation mark is written.

3.2 Automatic Phonetic Transcription

Phonetic transcription was traditionally performed by human experts. In contrast, we use automatic approach to the phonetic transcription problem in our system. Since Czech is a "phonetic language" both with a high degree of inflection (there are many forms for a single word, e.g. noun) and with a high degree of derivation (usage of prefixes and suffixes), it is useful to apply phonetic transcription rules in the form (1) to generate a sequence of phones from a sequence of letters [7]. Letter sequence `A` with both left context `C` and right context `D` is transcribed as phone sequence `B`.

$$\mathtt{A} \rightarrow \mathtt{B} \;/\; \mathtt{C\_D} \qquad (1)$$

The next rule (2) is an example of a typical Czech phonetic rule, which transcribes letter `d` to phone `dj`. Right context is formed by letters ě, `i`, í, while left context is ignored.

$$\mathtt{d} \rightarrow \mathtt{dj} \;/\; \_\langle \check{\mathtt{e}}, \mathtt{i}, \acute{\mathtt{i}} \rangle \qquad (2)$$

In our system we currently use a set of about 50 transcription rules. For less inflectional languages (e.g. English) different, dictionary-based approach to phonetic transcription is almost always used. In this conception "phonetic dictionary", which contains phonetic transcription of all words, is employed. In our system, this approach is also used to transcribe non-Czech words (so called exceptions) like foreign names, etc. Our dictionary of exceptions currently consists of about 620 stem-like forms.

Example 2: Phonetic transcription using Czech Phonetic Alphabet [8].

Letters :	[P] ▫▫▫▫ě▫▫▫ rádi vzpomínají na doby [P] kdy byli ▫▫▫▫ě▫▫ě▫▫▫▫▫▫▫
Phones:	[P] d o s p j e l ii r aa dj i f s p o m ii n a j ii n a d o b i [P] g d i b i l i j e sh tj e dj e t m i [P]

4. SPEECH PROCESSING

A speech synthesizer takes care of speech processing and forms the core of the TTS system. A concatenation-based synthesizer uses pre-recorded speech segments of basic speech units (stored in the SSD), modifies their prosodic features and concatenates them smoothly into the resulted speech.

Phones are not suitable to use as the basic speech units because they do not contain co-articulation phenomenon. Hence, phonetic input of the synthesizer (i.e. sequence of phones and prosodic marks delivered by the text-processing module) must be employed to derive sequence of speech units suitable for the use in a concatenation-based synthesizer. In our system we use speech segments that correspond to a clustered state of a triphone HMM (see Section 2.2 for more details).

Example 3: Speech unit derivation process.

Text:	▫▫▫▫ě▫▫▫ rádi vzpomínají na doby, kdy byli ▫▫▫▫ě▫▫ě▫▫▫▫
Pauses:	[P] ▫▫▫▫ě▫▫▫ rádi vzpomínají na doby [P] kdy byli ▫▫▫▫ě▫ ▫ě▫▫▫ [P]
Phones:	[P] d o s p j e l ii r aa dj i f s p o m ii n a j ii n a d o b i [P] g d i b i l i j e sh tj e dj e t m i [P]
Triphones:	sil sil-d+o d-o+s o-s+p s-p+j p-j+e j-e+l e-l+ii l-ii+ch … i-dj+e dj-e+t e-t+m t-m+i m-i+sil sil
Tied Triphones:	sil sil-d+o d-o+s o-s+p s-p+ng p-j+ee j-e+l e-l+ii aw-ii+ch … i-dj+oo dj-e+t oo-t+m c-m+i mg-i+sil sil
Clustered States:	sil_2_1 sil_4_1 d_2_81 d_3_51 d_4_46 o_2_134 o_3_206 o_4_79 s_2_67 s_3_142 s_4_61 … t_2_63 t_3_42 t_4_18 m_3_7 m_4_66 i_2_52 i_3_171 i_4_23 sil_2_1 sil_4_1

A PSOLA technique was selected to perform the concatenation and to modify both fundamental frequency (F_0) and duration of speech units to desired values. Time-domain version of this algorithm (i.e. TD-PSOLA) was firstly incorporated into the system [5], because it is easy to implement and gives high-quality speech, so it is very suitable to use in the first version of our system. Lately, some experiments were made with a parametric-domain variant of a PSOLA method – RELP-

PSOLA (Residual Excited Linear Predictive PSOLA). This technique has one more advantage over the standard TD-PSOLA – it enables spectral properties of speech to be under control and hence gives speech of a higher quality.

The positions of pitch-marks play the key role in the PSOLA algorithm. As mentioned in Section 2.4, pitch-mark detection algorithm was applied to determine the positions of pitch-marks in voiced parts of speech units. In unvoiced parts of speech, there is no activity of vocal cords and hence no pitch-mark positions are found. To be able to change the duration of unvoiced parts of speech units, "unvoiced pitch-marks" are uniformly distributed in unvoiced speech regions. Each pitch-mark defines a short-term signal, which is centered on the current pitch-mark position and Hanning windowed. The distance between the current and either the previous or the next pitch-mark position defines the size of the short-term signal. Fundamental frequency of to-be-concatenated speech units is then modified by changing the distance between short-term signals associated with adjacent pitch-mark positions. Monotonous F_0 is used for the present. Duration of speech units is changed by inserting or deleting short-term signals, preserving the desired F_0. Increasing duration of a speech unit by a high factor leads to a number of repetitions of the same short-term signal in the synthetic speech segment and causes the distortion of the synthetic speech. To avoid this problem, short instances of speech units were not selected as the representatives and were not stored in SSD (see Section 2.4).

In the time domain version PSOLA algorithm is applied directly to the speech signal of speech units. In RELP-PSOLA residual signal is employed instead and a moreover traditional LPC filter is used to generate speech. To do that, each speech unit must be a subject of both pitch-synchronous LP filtering to get LP coefficients (representing the properties of a vocal tract) and pitch-synchronous inverse LP filtering to get residual signals (representing the excitation source).

The requested synthetic duration of every speech unit is set using the statistics computed over all instances of the speech unit in SC [1]:

$$d_i = m_i + k \cdot s_i \,, \tag{3}$$

where m is median and s is standard deviation of duration of the speech unit i and k is so-called scaling factor. This factor can increase duration of the speech unit (i.e. slows down the speech rate) so that the synthetic speech is better to understand.

Moreover, energy of each speech unit is scaled to have its median value. This energy normalization prevents abrupt changes of energy from unit to unit and so smoothes the distribution of energy in the synthetic speech. Here, it is evident why low-energy instances of speech units were not selected in the segment selection process (see Section 2.4): if they were selected, they would be scaled up with a high factor now, introducing the artifacts into the synthetic speech.

5. CONCLUSION AND FUTURE WORK

Described TTS system aims to generate intelligibly and naturally sounding speech. To achieve this, a very precise, automatically built triphone-based Czech SSD and the PSOLA technique are used. Although the very first fully functional version of our system has been implemented so far, the first output samples are very hopeful and promising (listen to [SOUND 00444_01.WAV] and [SOUND 00444_02.WAV] for TD-PSOLA or [SOUND 00444_03.WAV] and [SOUND 00444_04.WAV] for RELP-PSOLA implementation).

There are plenty of things to improve in all three major parts of the system, of course.

- As for the SSD, we will focus to pitch-synchronous HMM training, which should enable more precise speech corpus segmentation, and will make some experiments with model topology.
- Text processing should be also enhanced to pre-process text and to detect prosodic features (i.e. pauses, fundamental frequency contour and duration distribution).
- Of course, speech processing needs to be improved too. An effective smoothing algorithm should be proposed for RELP-PSOLA to smooth spectral discontinuities at the boundaries of speech units in the synthetic speech. A more general cepstral approach to speech production can be also taken into account and combined with PSOLA algorithm [3].

6. REFERENCES

1. Donovan, R.E., and Woodland, P.C. "A Hidden Markov-Model-Based Trainable Speech Synthesizer," *Computer Speech and Language* 13: 223–241, 1999.
2. Matoušek, J. "Speech Synthesis Using HMM-Based Acoustic Unit Inventory," *Proceedings of Eurospeech'99*, Budapest, 1999, pp. 2323–2326.
3. Matoušek, J., Müller, L., and Psutka, J. "Text-To-Speech Synthesis Using HMM-Based Triphones," *Proceedings of ICSPAT'99*, Orlando, 1999.
4. Young, S., Odell, J.J., and Woodland, P.C. "Tree-Based State Tying for High Accuracy Acoustic Modelling," *Proceedings of the ARPA Workshop on Human Language Technology*, Plainsboro, New Jersey, 1994, pp. 307–312.
5. Moulines, E., and Charpentier, F. "Pitch-Synchronous Waveform Processing Techniques for Text-to-Speech Synthesis Using Diphones," *Speech Communication*, 9, 1990, pp. 453–467.
6. Young, S., et al. "The HTK Book," Entropic Inc., 1999.
7. Psutka, J. "Communication With Computer by Speech," (in Czech), Academia, Prague, 1995.
8. Nouza, J., Psutka, J., and Uhlíř, J. "Phonetic Alphabet for Speech Recognition of Czech," *Radioengineering,* 6, 1997, pp. 16–20.