Building a New Czech Text-to-Speech System Using Triphone-Based Speech Units*

Jindřich Matoušek

University of West Bohemia, Department of Cybernetics,
Univerzitní 8, 306 14 Plzeň, Czech Republic
tel.: +420 19 7491 150, fax: +420 19 279050
jmatouse@kky.zcu.cz

Abstract. This paper presents ARTIC, 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 (SSD), text analyzer and speech synthesizer. A statistical approach to speech segment database construction is used. Text processing module includes the phonetic transcription of written text and the conversion to synthetic units. Speech processing is performed using a PSOLA algorithm.

1 Introduction

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

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 process the input text and to convert it to the sequence of units which are then used in the speech synthesis stage. This process is called text processing.

The last task of a TTS system is to generate speech. Since concatenation-based approach to speech synthesis is used in our system, a method how to concatenate units smoothly is proposed and described.

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

* This work was supported by the project No. VS97159 and No. MSM23520004 of the Ministry of Education of Czech Republic.

2 Speech Segment Database

Hidden Markov Models (HMMs) are used as a statistic tool for an automatic speech corpus segmentation [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.

2.1 Speech Corpus

In our system we use a large single-speaker continuous-read-speech corpus (SC). 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. 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 acceleration coefficients). To be able to perform a PSOLA algorithm, each utterance 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. As the last step, each utterance was transcribed automatically using Czech phonetic transcription rules (see section 3.1).

2.2 Automatic Segmentation of a Speech Corpus

The automatic segmentation of the speech corpus was performed in a similar way as in [2]. Three state left-to-right HMMs with no skips and with 6 ms frame

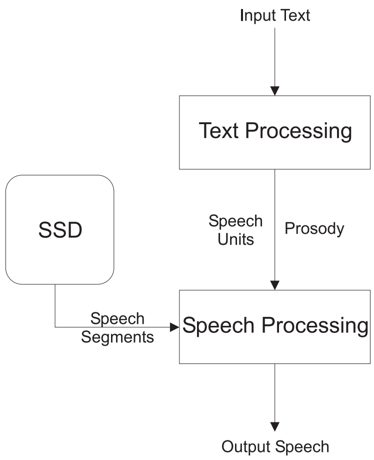

Fig. 1. Simplified scheme of our concatenation-based TTS system

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

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

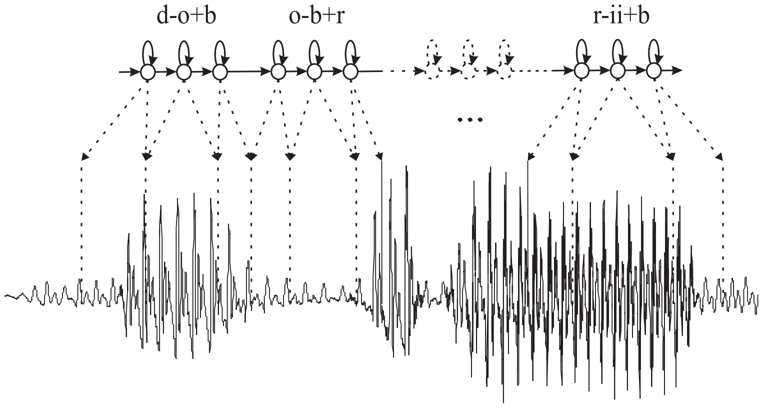

Fig. 2. Example of automatic speech segmentation into clustered states.

rate were used to model triphones. Starting from 45 Czech phones, 8.540 cross-word triphone models (which were seen in the SC) 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 SC to be modeled reasonably by HMMs. Taking into account a great number of triphones (91.125 theoretically possible), it is practically impossible to create such a SC. Hence, described SC, 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 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 context and shared only 7.742 clustered states (see Table 1). The clustered states represent the basic speech units of our TTS system. Finally, triphone HMMs are used to automatically segment the SC into these units (see Fig. 2). As a result, time labels of each speech unit are added to the SC.

2.3 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. Simple segment selection algorithm (similar to [1]) is adopted to choose suitable segments. For the later use both too short and low-energy instances are undesirable. Hence, statistics about each speech unit duration and energy were computed over all instances of given speech unit. Then, instances with duration or energy lesser than corresponding median values (and some threshold) are not selected. 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 present for particular speech unit in the SC and one of them has duration or energy much more greater than the others) led to selecting an incorrect segment and to synthesizing the speech unit with incorrect duration or energy (see section 4).

3 Text Processing

Text is the only input of the TTS system. Hence, text should be processed thoroughly to obtain important knowledge about what to synthesize. Text analyzer is an important part of a TTS system, because it tells the synthesizer which speech units to use and how to modify them. Firstly, input text should be pre-processed, i.e. some operations must be performed to transcribe 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. So the key task of text analyzer consists of converting input text to the sequence of clustered states – the basic speech units needed by the synthesizer. This process requires phonetic transcription (letter-to-phone conversion) to be done (see section 3.1 for more details). Finally, it is no problem to convert the sequence of phones to a sequence of triphones and then to a sequence of clustered states using the decision trees. Example of text processing is shown in Table 2.

3.1 Automatic Phonetic Transcription

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.

$$A \rightarrow B / C\_\, D \;. \tag{1}$$

The next rule is an example of a typical Czech phonetic rule, which transcribes nasal letter n to phone [ň]. Right context is formed by letters *ě*, *i*, or *í*, left

Table 2. Example of text-processing.

Text:	Myslím, tedy jsem.
Pauses:	Myslím [P] tedy jsem
Phones:	pau mislím pau tedi sem pau
Triphones:	pau pau-m+i m-i+s i-s+l s-l+í l-í+m í-m+pau pau pau-t+e t-e+d e-d+i d-i+s i-s+e s-e+m e-m+pau pau
Tied Triphones:	pau pau-m+i m-i+s i-s+l s-l+í l-í+m í-m+pau pau pau-t+e t-e+d ee-d+ii d-i+s i-s+e s-e+m e-m+pau pau
Clustered states:	pau_2_1 pau_4_1 m_2_4 m_3_3 m_4_66 i_2_54 i_3_168 i_4_89 s_2_107 s_3_112 s_4_42 l_2_43 l_3_104 l_4_94 í_2_35 í_3_113 í_4_108 m_2_80 m_3_113 m_4_5 pau_2_1 pau_4_1 t_2_18 t_3_10 t_4_39 e_2_71 e_3_122 e_4_112 d_2_44 d_3_34 d_4_36 i_2_77 i_3_107 i_4_82 s_2_109 s_3_89 s_4_70 e_2_113 e_3_35 e_4_173 m_2_68 m_3_1 15 m_4_6 pau_2_1 pau_4_1

context is ignored.

$$n \rightarrow \check{n} / \_ \ \langle \check{e}, i, \acute{\imath} \rangle \ . \quad (2)$$

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 like foreign names, etc. Our dictionary of exceptions currently consists of about 620 stem-like forms.

4 Speech Processing

Speech synthesizer takes care of speech processing and forms the core of the TTS system. Concatenation-based synthesizer prosodically modifies speech segments stored in the SSD and then concatenates them according to sequence of speech units delivered by the text analyzer. A PSOLA technique was selected to perform the concatenation and to change both the fundamental frequency (F_0) and the duration of speech units to desired values. Time-domain version of this algorithm (i.e. TD-PSOLA) has been incorporated into the system so far [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.

The positions of pitch-marks play the key role in PSOLA algorithm. As mentioned in section 2.1, pitch-mark detection algorithm was used to determine the positions in voiced speech. In unvoiced speech, there is no vocal cords activity and hence no pitch-mark positions are found. To be able to change the duration of unvoiced parts of speech, "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 size of the short-term signal is defined either by the previous or the next pitch-mark position. Fundamental frequency of speech units to concatenate is then modified by changing the distance between short-term signals centered on 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. The requested duration of every speech unit is set to its median value computed over all instances of the speech unit in SC (see section 2.3). Moreover, energy of each speech unit is scaled to have its median value.

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 a 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.

There is a lot of things to improve in all three major parts of the system, of course. As for SSD, we will focus to pitch-synchronous HMM training 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. F_0 contour and duration distribution). Of course, speech processing needs to be improved too. Our aim is to implement a PSOLA algorithm in parametric-domain to enable controlling spectral properties of resulting speech (as proposed e.g. in [3]).

References

1. Donovan R.E., Woodland P.C.: A Hidden Markov-Model-Based Trainable Speech Synthesizer. Computer Speech and Language, 13. (1999) 223–241.
2. Matoušek J.: Speech Synthesis Using HMM-Based Acoustic Unit Inventory. Proceedings of Eurospeech'99. Budapest (1999) 2323–2326.
3. Matoušek J., Psutka J., and Tychlt Z.: Statistical Approach to the Automatic Synthesis of Czech Speech. Proceedings of TSD'99. Mariánské Lázně (1999) 376–379.
4. Young S.: Tree-Based State Tying for High Accuracy Acoustic Modelling. Proceedings of the ARPA Workshop on Human Language Technology. Plainsboro, New Jersey (1994) 307–312.
5. Moulines E., Verhelst W.: Time-Domain and Frequency-Domain Techniques for Prosodic Modification of Speech; In: Speech Coding and Synthesis, edited by Kleijn W.B., Paliwal K.K. Elsevier (1995) 519–555.
6. Young S.: et al.: The HTK Book. Entropic Inc. (1999).
7. Psutka J.: Communication With Computer by Speech (in Czech). Academia, Prague (1995).