An Overview of the AlfaNum Text-to-Speech Synthesis System

Milan Sečujski, Vlado Delić*, Darko Pekar*[&], *Radovan Obradović*, Dragan Knežević*[&]
*Faculty of Engineering, University of Novi Sad, Serbia
[&]AlfaNum Ltd., Novi Sad, Serbia

Abstract

The paper gives a brief review of the development of the first widely applied text-to-speech synthesis system in the Serbian language The system was developed at the Faculty of Engineering, University of Novi Sad, Serbia. At the signal processing level, the system can switch between two techniques (TD-PSOLA and hybrid H/N synthesis). The problems related to prosody and naturalness of the synthetic speech were handled using a large accentuation-morphologic dictionary as well as rule-based syntax analysis.

1. Introduction

Since speech is the most natural means of communication between humans, people have been trying to develop system that would enable them to extend this interface to communication with machines as well. Beside automatic speech recognition (ASR), that enables machines to understand human commands, the technology of text-to-speech synthesis (TTS) enables them to address humans, providing them with various information in a way more natural and in some cases more efficient than the standard visual computer output. Another important application of text-to-speech synthesis is providing independence in computer access to the physically impaired (particularly the visually impaired).

However, text-to-speech is a language dependent technology and in some regions it can hardly ever be imported from abroad as most other technologies can. It has to be developed for each language separately, especially in case of languages such as Serbian, having in mind all its peculiarities as a Slavic language. It would have been unrealistic to expect some of the world's biggest companies dealing with speech technologies to decide to develop quality TTS for such a small and closed market in the near future.

The AlfaNum group at the Faculty of Engineering, University of Novi Sad, Serbia, has been dedicated to the development of speech synthesis for seven years, and the most important result of their effort is the first text-to-speech system in Serbian, taking into account linguistic information and thus greatly enhancing intelligibility and naturalness of synthesised speech. The system is also adapted for text-to-speech in Croatian and Macedonian. The system is the basis of a number of applications, such as an audio library and a speech-enabled web site for the visually impaired, and plays an important role in many commercial applications including interactive voice response systems and call centres.

2. System Overview

During the development of the first high-quality TTS for Serbian language, AlfaNum team has encountered many problems linked to bridging the gap between plain text and synthesised speech with all its typical features such as intelligibility and naturalness. There is no explicit information in a plain text concerning phone durations, pitch contours nor energy variations. These factors also depend on the meaning of the sentence, emotions and speaker characteristics, which further aggravates the task of attaining high naturalness of synthesised speech [1]. The Serbian language belongs to a group of tonal languages, having high-low pitch patterns permanently associated with words [2], thus a dictionary-based strategy for lexical stress assignment had to be employed in this case. The accentuation dictionary used for this is described in [3]. The Croatian and Macedonian version of the system require separate accentuation dictionaries. Language specific issues of the AlfaNum speech synthesiser are discussed in section 3.

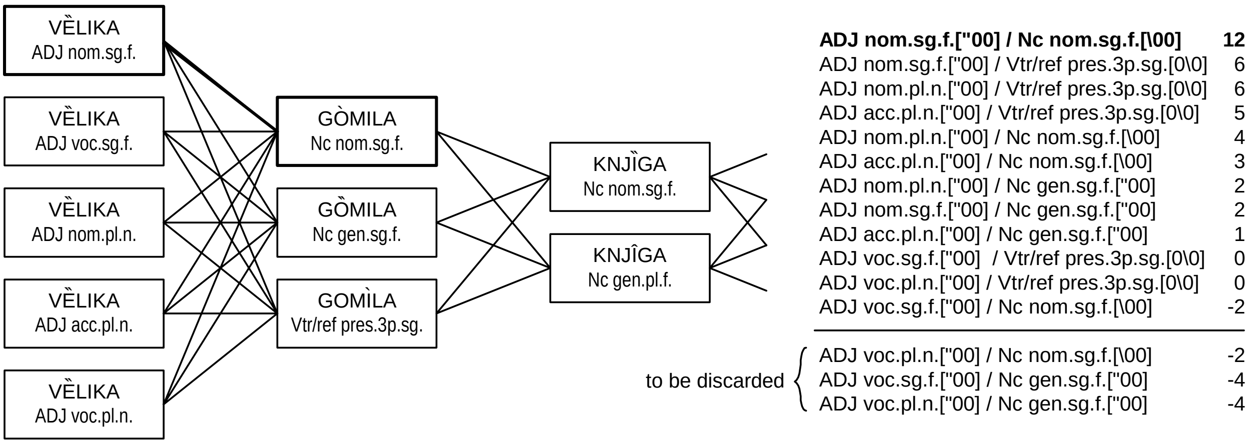
TTS systems in general have two main functions. Production of sound that is supposed to simulate human speech is referred to as *low-level synthesis*. *High-level synthesis*, on the other hand, is related to the conversion of written text into an appropriate representation of the desired acoustic signal (phoneme identity, duration, f_0 contour and energy), suitable for driving a low-level component of the synthesis system. In order to achieve satisfactory quality of the synthesised text, both component must be designed with utmost care.

2.1. High-Level Synthesis

High-level synthesis module includes processing of text and its conversion into a suitable data structure describing speech signal to be produced. The necessary steps include expanding numbers, abbreviations and other non-orthographic expressions, as well as resolution of morphological and syntactic ambiguities. The latter is based on a comprehensive accentuation dictionary as well as rule-based syntax analysis. A correct resolution of morphological and syntactic ambiguities is important because errors may easily lead to errors in accentuation, impairing naturalness of the synthesised speech. Naturalness of synthetic speech is not merely a question of aesthetics, because incorrect accentuation patterns can mislead listeners or force them to temporarily focus their attention to lexical segmentation (identification of individual words in the input speech stream) instead of the actual meaning of the text. This problem has especially serious potential consequences in tonal languages such as Serbian. In an intelligibility test described in detail in [4], a group of listeners was provided with synthesised sentences with prosody features based on accentuation – accurate, inaccurate (misleading) and neutral. The experiment showed that the opinion score was consistently the highest for sentences with prosody features based on accurate accentuation, and furthermore, that innacurately accentuated synthesised speech was harder to understand than speech with completely neutral prosodic features.

In the current commercially available version of the system, lexical stress assignment, as one of the most important factors influencing the shape of the pitch contour, is based on the algorithm described in [5]. This approach has proved to produce a reasonably correct stress pattern, with word error rate as low as 2,8% for Serbian language.

After the initial tokenisation of the input text, the words are looked up in the dictionary and a list of all possible part-of-speech (POS) and morphologic category values that correspond to given inflectional forms is created. In languages with poor inflection, tags usually contain only POS information, whereas in highly inflective languages tags usually contain much more information. The next step consists of context analysis, which considers a word in its context and seeks to determine its tag given the possible tags of neighbouring words. The input data for context analysis consist of a list of possible tags of all words contained in the sentence. As it would be impossible to consider all tag combinations separately, an algorithm similar to dynamic programming is used, thus keeping the number of partial hypotheses under control. A partial hypothesis contains a string of N tagged words at N initial positions in the sentence. Each of the partial hypotheses is scored (estimated as more or less likely) based on rules defined according to the statistics of different parts-of-speech in Serbian language, as well as most regular dependencies among them. Partial hypotheses with low scores are discarded before being expanded by adding the N+1-th word, thus the total number of partial hypotheses cannot exceed the product of stack size limit and maximum number of possible tags per word. Figure 1. represents an example of a step in the disambiguation algorithm for the sentence "*Velika gomila knjiga stoji na stolu*" (*There is a large heap of books on the table.*) The diagram shows the situation after all the hypotheses of length two are considered, and three of them with lowest scores are to be discarded (in the example the stack size limit is equal to 12).



ADJ nom.sg.f.["00] / Nc nom.sg.f.[\00]	**12**
ADJ nom.sg.f.["00] / Vtr/ref pres.3p.sg.[0\0]	6
ADJ nom.pl.n.["00] / Vtr/ref pres.3p.sg.[0\0]	6
ADJ acc.pl.n.["00] / Vtr/ref pres.3p.sg.[0\0]	5
ADJ nom.pl.n.["00] / Nc nom.sg.f.[\00]	4
ADJ acc.pl.n.["00] / Nc nom.sg.f.[\00]	3
ADJ nom.pl.n.["00] / Nc gen.sg.f.["00]	2
ADJ nom.sg.f.["00] / Nc gen.sg.f.["00]	2
ADJ acc.pl.n.["00] / Nc gen.sg.f.["00]	1
ADJ voc.sg.f.["00] / Vtr/ref pres.3p.sg.[0\0]	0
ADJ voc.pl.n.["00] / Vtr/ref pres.3p.sg.[0\0]	0
ADJ voc.sg.f.["00] / Nc nom.sg.f.[\00]	-2
ADJ voc.pl.n.["00] / Nc nom.sg.f.[\00]	-2
ADJ voc.sg.f.["00] / Nc gen.sg.f.["00]	-4
ADJ voc.pl.n.["00] / Nc gen.sg.f.["00]	-4

Fig. 1. An example of a step in the disambiguation algorithm

The result of context analysis is a list of words with their corresponding tags, as well as accentuation pattern, which is the most important feature from the point of view of speech synthesis. This pattern is subsequently converted into an f_0 curve which is to be applied during speech signal generation, by assigning each accented word an initial f_0 curve, concatenating such f_0 curves and smoothing the result. The synthesised speech is highly intelligible and reasonably natural-sounding, much more than any other attempts at speech synthesis in Serbian so far.

Some of the recent improvements of the high-level synthesis module include improved context analysis using transformation-based part of speech tagging, with several language-specific modifications related to data sparseness issues in languages with high inflection. The basic idea of transformation-based part-of-speech tagging is described in [6].

2.2. Low-Level Synthesis

The term low-level synthesis refers to the actual process of producing a sound that is supposed to imitate human speech as closely as possible, according to a representation of the sentence provided by the high-level synthesis module, including at least the information related to phoneme identities and durations, as well as temporal changes of f_0 and energy.

In all of the available versions of the system, the concatenative approach has been selected as the most promising. The AlfaNum R&D team has recorded a large speech database and labeled it using visual software tools specially designed for that purpose. By keeping score of every phone in the database and its relevant characteristics, use of phones in less than appropriate contexts was avoided, which further contributed to overall synthesised speech quality. This synthesiser is not diphone-based as almost all other speech synthesisers developed for related languages are. The TTS engine can use larger speech segments from the database, according to both phonetic and prosodic requirements, and select them at runtime in order to produce the most intelligible and natural-sounding utterance for a given plain text [7].

According to differences between the existing and the required values of parameters previously defined, each speech segment which can be extracted and used for synthesis is assigned *target cost*, and according to differences at the boundaries between two segments, each pair of segments which can be concatenated is assigned *concatenation cost*. Target cost is the measure of dissimilarity between existing and required prosodic features of segments, including duration, f_0, energy and spectral mismatch. Concatenation cost is the measure of mismatch of the same features across unit boundaries. The degree of impairment of phones is also taken into account when selecting segments, as explained previously. The task of the synthesiser is to find a best path through a trellis which represents the sentence, that is, the path along which the least overall cost is accumulated.

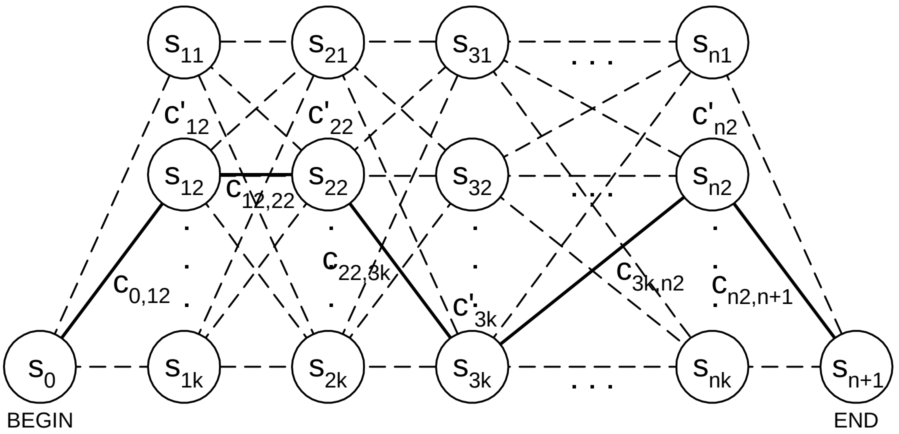


Fig. 2. Finding the best path through a trellis representing a sentence.

The chosen path determines which segments are to be used for concatenation, as shown in Figure 2. Segment modifications related to smoothing and prosody manipulation are made using the TD-PSOLA algorithm.

In the version developed recently, an alternative to the TD-PSOLA low-level synthesis algorithm has been introduced – hybrid harmonic/noise (H/N) synthesis [1]. The segmental intelligibility tests have still to be carried out, yet the first results seem to be encouraging.

3. Issues Related to Language Dependency

For the development of a multilingual TTS, separate speech databases are generally required for each language, although the Serbian database is at the moment used for Macedonian. This leads to a quite insignificant decrease in speech quality, due to the fundamental similarity between phonetic inventories of these two languages. The Macedonian speech database is expected to be recorded soon. If phonetic inventories are similar enough, as is the case for Serbian, Croatian and Macedonian language, it is appropriate accentuation and appropriate pitch contours that will make synthesised speech sound naturally, almost regardless of the original language of the database. Although the most obvious example of language dependence of TTS systems is the necessity for separate speech databases, it would be a serious mistake to assume that it would be sufficient to record a speech database in a new language for realisation of a high-quality TTS system in that language, even in case of very similar languages.

The task of a high-level synthesis module is to analyse the input text and to convert it into a representation suitable for driving the low-level synthesis component of the system. As described above, this task includes accentuation of each word in the sentence as well as its syntax analysis. All the resources necessary for this are language dependent. Namely, for each language a different accentuation dictionary had to be used, as well as different rules for scoring partial hypotheses in the algorithm described in section 2.1. The straightforwardness of the accentuation system in the Macedonian language allowed certain simplifications of the accentuation procedure in Macedonian, including the use of a dictionary of exceptions rather than a full accentuation dictionary.

Improvements of the high-level synthesis module related to context analysis using transformation-based part of speech tagging, described in section in 2.1., are possible only for Serbian at the moment, because a large corpus of previously morphologically annotated text is required for training. Such a corpus has been designed within the AlfaNum project in Serbian language, but there are still no similar corpora available for Croatian or Macedonian. All the tools designed for transformation-based POS tagging are language independent and the improvement will be easily extended to Croatian and Macedonian once an appropriate corpus is available.

Other issues related to language dependency are discussed in more detail in [8].

3. Conclusion

The system described in this paper is the first fully functional text-to-speech synthesiser in Serbian language, adapted to Croatian and Macedonian as well. It is constantly being improved by introducing novel techniques both at high and low synthesis level.

The most significant application of this system so far is anReader, a speech synthesiser for the visually impaired that, combined with software known as screen-readers, offers them complete independence in computer access. The number of the visually impaired computer users in Serbia has increased significantly since anReader was presented for the first time, and its popularity in Croatia and FYR Macedonia is also on the rise. Other applications of the system include the Audio-library, a client-server system enabling the visually impaired to access a large database of books via local network or the Internet, as well as a wide array of interactive voice response (IVR) systems supporting speech communication with the caller.

4. Acknowledgment

This work was supported in part by the Ministry of Science and Environment Protection of Serbia within the Project “Development of speech technologies in Serbian and their application in ‘Telekom Srbija’” (TR-6144A).

References

1. *Thierry Dutoit*, An Introduction to Text-to-Speech Synthesis, Kluwer Academic Publishers, Dordrecht/Boston/London, 1997.

2. *Ilse Lehiste, Pavle Ivić,* Word and Sentence Prosody in Serbocroatian, MIT Press, Cambridge, MA, 1986.

3. *Milan Sečujski,* DOGS 2002: An accentuation dictionary of the Serbian language intended for text-to-speech synthesis, Bečej, Serbia, 2002.

4. *Milan Sečujski, Radovan Obradović, Darko Pekar, Ljubomir Jovanov, Vlado Delić,* Text, Speech and Dialogue (TSD): AlfaNum System for Speech Synthesis in Serbian Language, Brno, Czech Republic, 2002.

5. *Milan Sečujski*, IEEE EUROCON: Obtaining Prosodic Information from Text in Serbian Language, Belgrade, Serbia, 2005.

6. *Eric Brill,* Computational Linguistics 21(4): Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging, MIT Press, Cambridge, MA, 1995.

7. *Mark Beutnagel, Mehryar Mohri, Michael Riley*, EUROSPEECH, Rapid unit selection from a large speech corpus for concatenative speech synthesis, Budapest, Hungary, 1999.

8. *Milan Sečujski,* Die Unterschiede zwischen dem Bosnischen/Bosniakischen, Kroatischen und Serbischen: The difference between Serbian and Croatian from the point of view of speech technologies, Graz, Austria, 2007.