A Romanian Syllable-Based Text-To-Speech System

OVIDIU BUZA, GAVRIL TODEREAN
Department of Telecommunications
Technical University of Cluj-Napoca
26 – 28 G. Baritiu Str., 400027, Cluj-Napoca
ROMANIA

Abstract: - In this article we present the way we have built a syllable-based TTS system for Romanian. The system contains: a text analyser capable to separate syllables from input text and detect accentuation, a vocal database with recorded syllables, a unit matching module and a synthesizer. The analyser was built using a LEX generator by mean of two sets of phonetic rules. Vocal database was generated through an automated wave segmentation procedure.

Key-Words: - syllable text-to-speech system, rule-driven text analyser, automatic segmentation

1 Introduction

Concatenation of waveforms represents a method more and more used in our days because of high level of naturalness in produced speech. Corpus-based methods are among best approaches, but they need great efforts for database maintaining.

Syllable-base methods can be an alternative, as they need a limited units database. Using of syllables in synthesis also leads to a good level of speech naturalness and low concatenation error rate because of small number of concatenation points inside the synthesized text.

This article presents an original approach for constructing a syllable-based TTS system for Romanian. The syllable approach is very appropriate in our case, because Romanian spoken language contains a big number of opened vowels that gives a constant rhythm of speech and similar manner of accentuating words.

Also, Romanian language contains a relative small number of syllables, so we have obtained a reduced size of vocal database.

Our text-to-speech system consists of ([7]):
- a text analysis module that brings input text and produces basic units, that in our approach are syllables, and prosody data, which mean the information about how words are accentuated;
- a unit matching module that generates acoustic units sequence according to the linguistic units detected from the input and prosody data;
- a speech synthesis module that generates speech based on the acoustic units sequences.

The particular aspects of our work are:
- using of linguistic and phonetic rules based of which we have done text analysis and obtained appropriate units and prosody data;
- automatic generation of database from recorded sequences.

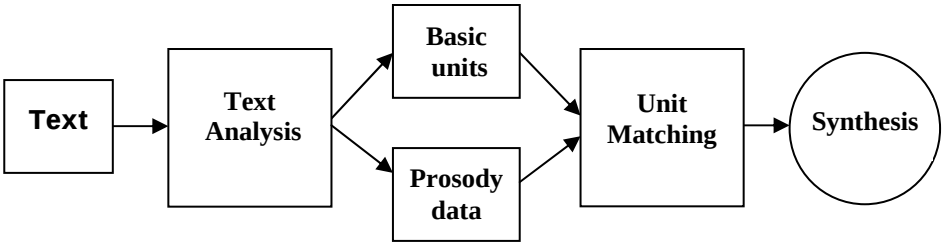

Fig.1. Main functionalities of our Text-to-Speech system

First of all we have built a linguistic analyzer module that is capable to split the input text into syllables. Next step was to determine accentuation by mean of a phonetic analyzer. Then we have automatically produced a database with PCM coded syllables of Romanian language. Synthesis was done by concatenating acoustic units from database and giving appropriate commands to the computer' sound blaster for voice generation.

2 Text Analysis

First stage in text analysis is the detection of linguistic units: sentences, words and segmental units, that in our approach are the word syllables.

Detection of sentences and words is done based on punctuation and literal separators. For detection of syllables we had to design a set of linguistic rules for splitting words into syllables, inspired from Romanian syntax rules ([2], [3]).

The principle used in detecting linguistic units is illustrated in figure no. 2. Here we can see the structure of text analyser that corresponds to four modules designed for detection of units, prosody information and unit processing.

These modules are:

- lexical analysis module for detection of basic units;
- phonetic analysis module for generating prosody information;
- high level analysis module for detection of high-level units;
- processing shell for unit processing.

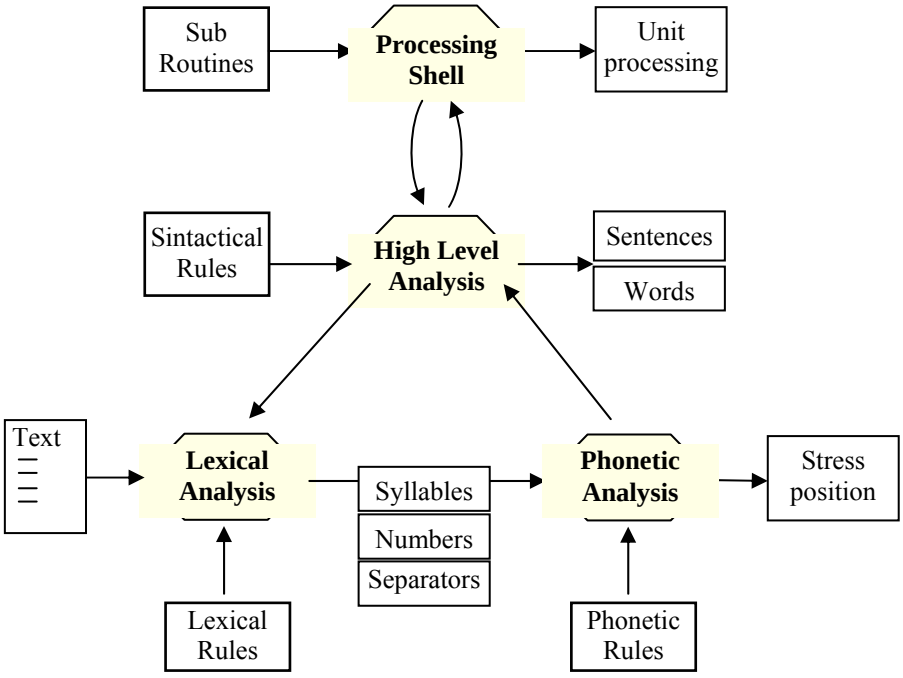

Fig.2. Text analyser for syllable detection

Processing shell accomplishes the unit processing task and controls the subsequent modules. The shell calls high-level analyser for returning main syntactic units. High-level analyser calls the lexical analyser for input text parsing and detection of basic units. Then phonetic analysis module is called for generating stress information.

Lexical analyser extracts text characters and clusters them into basic units. We refer to the detection of alphabetical characters, numerical characters, special characters and punctuation marks. Using special lexical rules (that have been presented in [8]), alphabetical characters are clustered as syllables, digits are clustered as numbers and special characters and punctuation marks are used in determining of word and sentence boundaries.

Phonetic analyser gets the syllables between two breaking characters and detects stress position, i.e. the accentuated syllable from corresponding word.

Then, high-level analyser takes the syllables, special characters and numbers provided by the lexical analyser, and also prosodic information, and constructs high-level units: words and sentences. Also basic sentence verification is done here.

Processing shell finally takes linguistic units provided from the previous levels and, based on some computing subroutines, classifies and stores them in appropriate structures. From here synthesis module will construct the acoustic waves and will synthesize the text.

3 Lexical Analysis for Syllable Detection

Lexical analyzer is called by the higher level modules for detection of basic lexical units: syllables, breaking characters and numbers. The lexical analyzer is made by using LEX scanner generator [4]. LEX generates a lexical scanner starting from an input grammar that describes the parsing rules. Grammar is written in BNF standard form and specifies character sequences that can be recognized from the input. These sequences refer to syllables, special characters, separators and numbers. Also BNF grammar specifies the actions to be taken in the response of input matching, actions that will be accomplished by the processing shell subroutines.

The whole process realized by the lexical analyzer is illustrated in figure no. 3. As we can see, input text is interpreted as a character string. At the beginning, current character is classified in following categories: digit, special character or separator, and alphanumeric character. Taking into account left and right context, current character and the characters already parsed are grouped to form a lexical unit: a syllable, a number or a separator. Specific production rules for each category indicate the mode each lexical unit is formed and classified, and also realize a subclasification of units (for numbers if they are integer or real numbers, and for separators – the type: word or sentence separator, affirmative, interrogative, imperative or special separator).

Once the unit type and subtype is identified, corresponding character sequence is stored and transmitted to the high-level analyzer by mean of specific actions, as they will be described in next paragraph (***Process syllable, Process number, Process separator***).

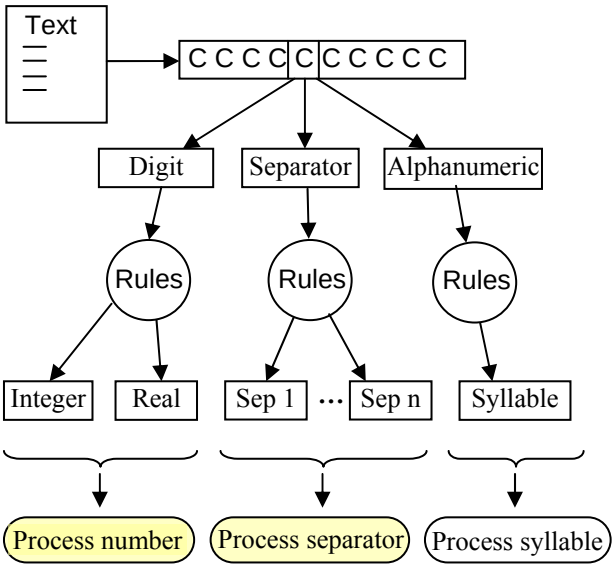

Fig.3. Lexical analyser for syllable detection

3.1 Specific actions of lexical analyser

Specific actions inform high-level module about matching of syllables, numbers and breaking characters. Inside lexical parser three types of input response actions are defined as follows:

A. Process syllable – this is the action to be taken when a syllable is matched in specific location of one word.

Special attention is taken when a syllable is matched at the beginning of a word. In Romanian, different word decomposition rules apply when a character sequence occurs at the beginning or in the middle or the final part of a word.

B. Process number – is the action to be taken when a number is matched from the input. The number is identified as INTEGER or REAL type. In future stage, numbers will be translated in orthographic alphabetical form.

C. Process separator - is the action corresponding to a breaking character matching from the input. Breaking characters and punctuation marks are used for detecting word and sentence boundaries.

3.2 Syllable rules matching

Regarding syllable rules matching process inside lexical analyser, two types of rule sets were made: a basic set consisting of three general rules, and a large set of exception rules which statues the exceptions from the basic set.

(A) The basic set shows the general decomposition rules for Romanian. First rule is that a syllable consists of a sequence of consonants followed by a vowel:

```
syllable ={CONS}*{VOC}                    (R1)
```

Second rule statues that a syllable can be finished by a consonant if the beginning of the next syllable is also a consonant:

```
syllable={CONS}*{VOC}{CONS}/{CONS}(R2)
```

Third rule says that one or more consonants can be placed at the final part of a syllable if this is the last syllable of a word :

```
syllable={CONS}*{VOC}{CONS}*/{SEP}(R3)
```

(B) The exception set is made up from the rules that are exceptions from the three rules of above. These exceptions are situated in the front of basic rules. If no rule from the exception set is matched, then the syllable is treated by the basic rules. At this time, the exception set is made up by more then 100 rules. Rules are grouped in subsets that refer to resembling character sequences. All these rules are explained in [7], [8].

4 Syllable Accentuation

The principle for determining syllable accentuation is shown in the following diagram:

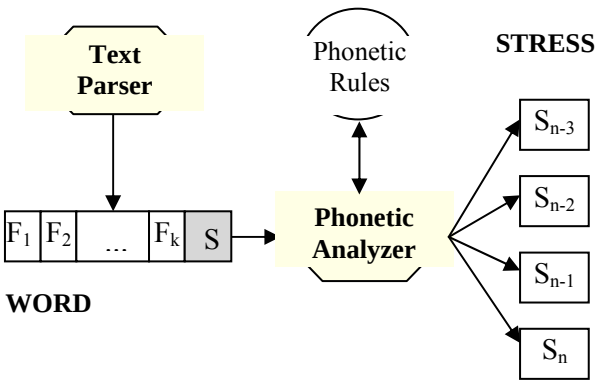

Fig.4. The principle of detecting syllable accentuation

The parser returns current word from input stream. The word consists of series of phonemes $F_1, F_2, ..., F_k$ and is delimited by a separator S. Phonetic analyser reads this word and detects syllable accentuation based on phonetic rules.

In Romanian stressed syllable can be one of last four syllables of the word: S_n, S_{n-1}, S_{n-2} or S_{n-3}, (S_n is the last syllable). Most often, stress is placed at last but one position.

The rules set consists of this general rule (S_{n-1} syllable is stressed):

```
{LIT}+/{SEP}        { return(SN_1);}
```

and a consistent set of exceptions, organized in classes of words having the same termination. In [7] one can find the complete set of rules.

5 Vocal Database Construction

Vocal database includes a subset of Romanian language syllables. Acoustic units were separated from male speech and normalized in pitch and amplitude.

Segmentation was done through an automated procedure which can detect silence/speech and voiced/unvoiced signal. Our approach uses time domain analysis of signal. After a low-pass filtering of the signal, zero-cross (Z_i) wave samples were detected. Minimum (m_i) and maximum (M_i) points between two zeros were also computed.

Separation between silence and speech is done using an amplitude threshold T_s . In silence segments all MIN and MAX points have to be smaller than T_s:

$$\begin{cases} |M_i| < T_s \\ |m_i| < T_s \end{cases}, i = s \ldots s+n \qquad (1)$$

In (1) s is the segment index and n is the number of samples in that segment.

For speech segments distance between two adjacent zero-cross points ($D_i = d(Z_i, Z_{i+1})$) is computed. Decision of voiced segment is assumed if distance is greater than a threshold distance V:

$$D_i > V \ , \quad i = s, \ldots , s+n \qquad (2)$$

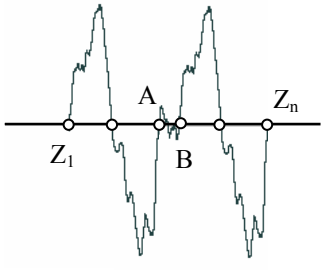

Fig.5. A voiced segment of speech

For the zero points between A and B from figure 5 to be included in the voiced segment, a look-ahead technique has been applied. A number of maximum N_k zero points between Z_i and Z_{i+k} can be inserted in voiced region if $D_{i-1}>V$ and $D_{i+k}>V$:

$$\begin{cases} D_j < V & , \ j=i..k; \quad k<=N_k \\ D_{i-1} > V & , \ i = s,\dots , s+n \\ D_{i+k} > V & \end{cases} \qquad (3)$$

Finally, if conditions (2) and (3) are not accomplished, current segment is assumed unvoiced.

After segmentation, voiced and unvoiced segments are coupled according to the syllable chain that is used in vocal database construction process. Acoustic units are labelled and stored in database. Each region boundary can be viewed with a special application and, if necessary, can be adjusted.

Vocal database with recorded syllables has a tree data structure. Each node in the tree corresponds with a syllable characteristic, and a leaf represents appropriate syllable.

Units have been inserted in database following the classification:

- after length of syllables : we have two, three or four character syllables (denoted S2, S3 and S4) and also singular phonemes (S1);
- after position inside the word: initial or median (M) and final syllables (F);
- after accentuation: stressed or accentuated (A) or normal (N) syllables.

This classification offers the advantage of reducing time for matching process between phonetic and acoustic units.

6 Unit Matching Process

The matching process is done according to the three–layer classification of units: number of characters in the syllable, accentuation and the place of syllable inside the word.

If one syllable is not founded in vocal database, this will be constructed from other syllables and separate phonemes that are also recorded. Following situations may appear:

(a) Syllable is matched in appropriate accentuated form. In this case acoustic unit will be directly used for concatenation.

(b) Syllable is matched but not the accentuation. In this case, unit is reconstructed from other syllables and phonemes which abide by the necessary accentuation.

(c) Syllable is not matched at all, so it will be constructed from separate phonemes.

7 Implementation

The purpose of our work was to build a speech synthesis system based on concatenation of syllables. The system includes a syllable database in which we have recorded 386 two-character syllables: 283 middle-word syllables and 103 ending-word syllables, 139 most frequent three-character syllables and 37 four-character syllables. Syllables that are not included in database are synthesized from existing syllables and separate phonemes that are also recorded.

The speech synthesis system first invokes text analyzer for syllable detection, then phonetic analyser for determining the accentuation. Appropriate unit (stressed or unstressed) is matched from vocal database, and speech synthesis is accomplished by syllable concatenation.

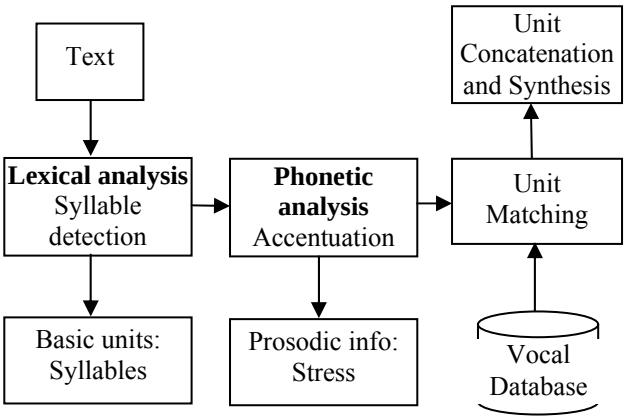

Fig.6. The principle of our syllable-based speech synthesis system

8 Conclusions and Results

We have presented in this article a complete method for construction of a syllable-based TTS system. Special efforts have been done to accomplish the text processing stage. After serious researches in linguistic field, we have designed one set of rules for detecting word syllables and a second set for determining which syllable is accentuated in each word. Even these sets are not complete, they cover yet a good

majority of cases. The lexical analyzer is entirely based on rules that assure more than 85% correct syllable detection at this moment, since accentuation analyser provides about 75% correct detection rate.

The advantages of detecting syllables through a rules-driven analyser are: separation between syllables detection and system code (different from [9], where syllables detection algorithm is integrated in source code); from here we have easy readability and accessibility of rules. Other authors ([1]) have used LEX only for pre-processing stage of text analysis, and not for units detection process itself. Some methods support only a restricted domain ([6]), since our method supports all Romanian vocabulary. The rules-driven method also needs less resources than dictionary-based methods (like [5]).

Also our automated segmentation method assures less error in concatenation points: waves begin and stop at zero-points and contain integer numbers of periods.

About speech synthesis outcome, first results are encouraging, and after a post-recording stage of syllable normalization we have obtained a good quality of text synthesis. In future implementations, F0 adaptive correction in concatenation points will improve this performance.

References:

[1] Burileanu D., et al., A Parser-Based Text Preprocessor for Romanian Language TTS Synthesis, *Proceedings of EUROSPEECH'99*, Budapest, Hungary, vol.5, pp. 2063-2066, Sep. 1999.

[2] Constantinescu-Dobridor G., *Sintaxa limbii române*, Editura Ştiinţifică, Bucureşti, 1994

[3] Ciompec G. et al., *Limba română contempo- rană. Fonetică, fonologie, morfologie*, Editura Didactică şi Pedagogică, Bucureşti, 1985.

[4] Free Software Foundation, Flex - a scanner generator, http://www.gnu.org /software/flex/manual, October 2005.

[5] Hunt A., Black A., Unit selection in a concatenative speech synthesis system using a large speech database, *Proc. ICASSP '96*, Atlanta, GA, May 1996, pp. 373–376.

[6] Lewis E., Tatham M., Word And Syllable Concatenation In Text-To-Speech Synthesis**,** *Sixth European Conference on Speech Communications and Technology*, pp. 615—618, ESCA, Sep. 1999.

[7] Buza O., *Vocal interractive systems*, doctoral paper, Electronics and Tele-communications Faculty, Technical University of Cluj-Napoca, 2005

[8] Buza O., Toderean G., Syllable detection for Romanian text-to-speech synthesis, *Sixth International Conference on Communications* COMM'06 Bucarest, June 2006, pp. 135-138.

[9] Burileanu C. et al., *Text-to-Speech Synthesis for Romanian Language: Present and Future Trends,*http://www.racai.ro/books/awde/burileanu.html