PROPHON - AN INTERACTIVE ENVIRONMENT FOR TEXT-TO-SPEECH CONVERSION

Marina Bäckström*    Ken Ceder    Bertil Lyberg

Swedish Telecom,
Technology Department,
S-123 86 Farsta, Sweden.

ABSTRACT

Prophon is an interactive environment for developing applications and conducting research in text-to-speech conversion. The environment, which is written in the programming language Prolog, can be considered as a shell for developing knowledge based systems. It is created specifically for the linguistic domain and encourages an incremental development of text-to-speech systems. Prophon offers a rich set of tools for creating rules at different levels.

INTRODUCTION

In this paper an interactive environment for text-to-speech conversion is presented. The system is called Prophon and is constructed specifically for the linguistic domain. It offers a rich set of development tools e.g., editing, explanation and tracing facilities which in an interactive way enables the user to explicitly express linguistic knowledge at different levels. The environment which is implemented in the programming language Prolog, can be considered as a shell for development of knowledge based systems. This implies that the linguistic knowledge is totally separated from the rest of the program. In order to express this linguistic knowledge, a high level notation formalism is used. Such an approach has been used before [1,2,3]. The advantage of using a notation formalism that is known by linguists and phoneticians is that the user need not have any experience in computer programming. Due to the declarative and interpretative nature of Prolog the environment facilitates a rapid construction of a prototype that the user easily can augment and modify incrementally. The interaction between the environment and the user employs a notation that, as closely as possible, corresponds to phonetic and linguistic terminology. It is thus possible to create systems with a structure that mirrors both linguistic intuition and models of the mental processes of the speaker.

DESCRIPTION OF THE SYSTEM

Prophon integrates different levels of linguistic analyses such as syntactic, phonological and prosodic analysis as well as acoustic parameter levels. In text-to-speech conversion it is often necessary to refer to different levels of the linguistic analyses in order to generate an accurate phonetic specification and a proper pronunciation including prosody of the synthetic speech.

The different parts of the text-to-speech development system are shown in fig. 1.

The system described is being used for the development of a text-to-speech conversion system for Swedish.

Syntactic analysis

The input text is first parsed, generating a syntactic structure. The analysis is carried out by means of a phrase-structure grammar implemented in DCG (Definite Clause Grammar) [4] that is supported in Prolog. The parser uses a top down, left to right parsing strategy. The rules in the grammar have, in this notation, a form that is easily comprehensible for linguists. A rule in the grammar can have the following form.

\[
\text{sentence}(s(NP,VP)) \rightarrow \\
\text{noun_phrase}(NP), \\
\text{verb_phrase}(VP).
\]

Syntactic information is necessary for generating proper pronunciation at both word and phrase level. The word "syntes" in Swedish is pronounced [sýntes] if it is a noun but [*sýntas] if it is a verb. The syntactic structure for the Swedish sentence "Pojken, som hörde syntes, syntes." (The boy who heard synthesis was seen.) is shown in fig. 2. At the phrase level, syntactic information is important for the generation of the suprasegmental features such as intonation and duration.

Lexicon

The lexicon contains approximately 60 000 basic entries that are expanded by inflection rules to
about 500,000 words. The unexpanded version was developed at Chalmers University of Technology, Department of Information Theory.

The expanded lexicon contains information about the phonetic transcription at word level and information about syntactic categories. The lexical information about the word "syntes" is shown in fig. 3. Semantic information will be included in the near future.

**Rules for grapheme to phonetic transcription**

If a word is not included in the lexicon, the phonetic transcription is derived from the orthography by the grapheme-to-phonetic transcription module. The transcription module consists of an ordered set of grapheme-to-phoneme and phonological rules that can be either context sensitive or context free. These rules generate the phonetic transcription at the word level.

The formalism corresponds as closely as possible to the one used by linguists and phoneticians [5]. A similar formalism has been used in other text-to-speech systems [1,2,3].

The orthography (string of graphemes) is converted to a string of phonemes. Each phoneme is then associated with a feature matrix consisting of a set of binary features, which uniquely defines the phoneme. The user is free to specify any number of such binary features. The phoneme /d/ is represented:

\[ d = [\text{\text{-voc}}, \text{+cons}, \text{+ant}, \text{+cor}, \text{-cont}, \text{-nas}, \text{+voice}] \]

By using a feature specification it is possible to refer to groups of phonemes. The feature specification \([\text{+voc}, \text{-cons}]\) will accordingly refer to all vowels.

The rules have the following general form:

**Condition \rightarrow Action / Context _ Context**

The condition specifies the string that is to be transformed and the left and right contexts indicate under which circumstances the actual transformation takes place. The condition, action and contexts may consist of a string of graphemes, phonemes and/or a set of feature matrices. They can also be empty.

Below is an example of a rule that transforms the
Some extensions of this formalism are allowed:

1. The ability to use variables as feature coefficient e.g., \([Avoc, Bcons]\), where the variables A and B can be either + or -.

2. The ability to express an arbitrary number of segments e.g., \([+seg] (m,n)\) i.e. no less than m and no more than n segments.

3. The ability to express "exclusive or" e.g., \([-seg v +seg]\) i.e. either the feature \([-seg]\) or the feature \([+seg]\).

The inferences are made in a forward chaining manner, where the rules are applied from top to bottom, and the input string is scanned from left to right.

**Rules at phrase level**

In order to generate proper pronunciation at phrase level it is necessary to have the possibility to associate the different linguistic levels and the acoustic parameter level with each other. The acoustic correlates of the suprasegmental features such as fundamental frequency and duration have to be manipulated with reference to e.g., phrase structure and phrase position in order to produce high quality synthetic speech.

**Syntactic Structure**

![Syntactic Structure Diagram]

**Transcription**

\[
\text{pojken som hörde syntes syntes.}
\]

Fig. 2 The syntactic structure for the Swedish sentence "Pojken, som hörde syntes, syntes." (The boy who heard synthesis was seen).

**Fig. 3 Information about the two instances of the word "syntes" in the expanded lexicon.**
Rules for parameter generation

Each allophone in the phonetic transcription is associated with a parameter default matrix that consists of parameter values. The phonetic transcription and the string of parameter matrices are then input to the parameter rules. The user can specify the different parameters needed for the parallel/cascade synthesizer. Each parameter can be specified at an arbitrary number of points in the time domain where each point is associated with a parameter value and a function that specifies the interpolation between the value in question and the parameter value in the following point.

The general form of the parameter specifications is:

\[
\text{allophone} = \left[ \text{parameter1 time value (function)} \right. \\
\text{time value (function) ..........}
\]

\[
\text{parameter2 time value (function).........} \right]
\]

Example:

\[
a = \left[ \ldots \text{F1} \ .4t \ 604 \ (\text{lin}) \ .6t \ 608 \ (\cos) \ldots \right]
\]

The parameter rules have a form similar to the rules at the linguistic levels. The condition and the context consist of a string of allophones and/or a set of binary features, and the action part specifies the manipulations of the parameters in the time dimension. It is also possible to refer to different points in time.

\[
\text{Condition} \rightarrow [\text{parameter1}: \text{point time value (function)} ] / \text{Context} \_ \text{Context}
\]

Example:

\[
a \rightarrow [ \text{F1:2} \ 0.7t +100 \ (\text{lin}) ] / _ \_ [+\text{cons}]
\]

At the acoustic parameter levels, it is possible to handle synthesis units based on both allophones and larger units, e.g., demisyllables. It is therefore possible to combine different synthesis strategies in order to generate the best attainable speech quality.

The reference maps for larger units are created from real speech by means of synthesis by analysis, in order to get the parameter values for the formant synthesizer. Each map is then divided into segments. It is therefore possible to change a single segment in the reference map. This mechanism is especially valuable for the generation of proper duration values at the word and phrase levels.

Cascade-parallel formant synthesizer

A parallel/cascade synthesizer developed by Klatt [6] is used for the generation of speech from the acoustic parameter level. The code is written in Lisp and interacts with the SPIRE speech analysis package [7]. It is therefore easy to compare the synthetic version with natural speech references during the rule development.

ACKNOWLEDGEMENT

Our sincere thanks are due to Richard Schulman, Stockholm University for valuable discussions and suggestions during the development of the research tool.

REFERENCES


