ABSTRACT

Automatic synthesis of speech from arbitrary text requires two basic operations: linguistic analysis of input text and speech waveform generation. The achieved quality of the second stage very much depends on the reliability and richness of information generated in the first stage.
In this paper we discuss possibilities and problems of text analysis for multilingual speech synthesis. The language independent approach requires the separation of all the language specific information into the language specific inventory, which is composed of different lexica, various dictionaries and lists. The remaining core represents the language independent text-processing engine.

Keywords: multilingual text processing, linguistic analysis, Unicode character set.

1. INTRODUCTION

First stage in any text to speech conversion is the handling and analysis of the text to be processed. According to the general, unrestricted nature of the text those algorithms have to be highly flexible and robust.

To cope with this complex task most of the currently available state of the art systems employ sophisticated and elaborated grammars, morphological analysis and language specifically tuned rule sets [6]. The generation or adaptation of those routines for new languages requires human mother tongue experts and consumes substantial time and resources.

In this paper a first approach for a completely multilingual and data driven text analysis module for the PAPAGENO TTS system is presented. A language independent core engine uses a set of exchangeable basic and derived knowledge sources specific to each language. See figure 1 for illustration of this concept of separating core engine and knowledge sources. This set can be subdivided in two main classes:

- Basic knowledge sources such as phonetic or linguistic dictionaries commercially available and simple lists (see section 4).
- Data driven generalisation routines trained on those basic knowledge sources.

In section 2 the system architecture consisting of an integrated queuing mechanism as well as a token normalisation routine are briefly introduced.

Section 3 illustrates a multistage concept for end of sentence detection.

Section 4 highlights as an example a set of knowledge sources for German. Results in application...
together with discussion and future conclude the paper.

2. SYSTEM ARCHITECTURE

2.1 Integrated queuing mechanism

In the first processing step the unstructured stream of characters is grouped into tokens by a lexical scanner using a simple list of possible delimiters (see section 3 for possible examples) [2].

All generated tokens are then organised as a queue data structure. This queuing mechanism provides an integrated workbench for all operations like splitting, recombination or replacement of tokens at the subsequent processing stages.

This implicitly avoids shortcomings and disadvantages of other e.g. hierarchically architectures. Figure 2 shows the sequential arranged but intertwined processing steps.

To allow fast and easy transfer to many languages the Unicode character set [1] is supported for all orthographic representation in input text and dictionaries.

2.2 Token normalisation

For syntactical analysis at later processing stages the linguistic properties of the words in the input text have to be determined.

The tokens separated by the lexical scanner are not necessarily identical with the entries in the linguistic dictionary. The reasons for not being able to match a token to the appropriate lexicon entry are:

- The lexicon entry is a combination of two or more tokens. A typical example is *New York*. For both tokens *New* and *York* separate lexicon entries may exist, while only their combination is assigned the correct properties to.
- The token is a combination of two or more lexicon entries like in *text-to-speech*.
- Even in either of the above mentioned alteration text denoted by a token may not be contained in the dictionary. Typical examples are loanwords and proper names.

To complete the general linguistic dictionary additional specialised dictionaries for abbreviations or acronyms may be used if available for this language.

In an offline preparation process neural networks are trained for generalisation of the linguistic dictionary to handle out of vocabulary (OOV) words. The linguistic categories of those words are then determined by their orthography.

![Figure 2](image-url)

*Figure 2: Queuing mechanism for subsequent but intertwined processing stages in the text analysis engine.*

3. MULTISTAGE CONCEPT FOR END OF SENTENCE DETECTION

As the acoustical realisation of the demanded phone sequences are computed sentence by sentence the correct determination of sentence boundaries is crucial for the overall listening impression [3]. Those sentence boundaries cannot be derived straightforward by the analysis of punctuation marks due to the wide spread use of those symbols in more complex linguistic constructs.

The most prominent example of ambiguous punctuation marks is the point in European languages. It often appears as parts of abbreviations, acronyms, ordinal numbers or as an initial in names. It can represent a full stop, a part of an abbreviation, or both.

This complexity is reflected by a multistage procedure for end of sentence detection:

- The first step right after the separation of the token by the lexical scanner all punctuation marks that potentially can be a sentence delimiter are marked as a possible end of sentence token (PEOS).
- Those PEOS tokens can be deleted or merged with surrounding tokens if found to be constituent of a complex structure as e.g. float numbers.
• PEOS tokens still existent after the token normalisation and the number handling are left as candidates for sentence delimiters to a final decision in the tagging module. At this processing stage a maximum of information on the surrounding linguistic constructs is available and can be basis of an embedded and integrated decision [5].

Figure 3: Linguistic analysis components in interaction with the text-processing engine presented in this paper.

4. EXAMPLE OF KNOWLEDGE SOURCES FOR GERMAN

As an example of a possible set language specific knowledge sources the following dictionaries and lists were employed in our experiments:

Linguistic dictionary:
This linguistic dictionary was designed to cover the text of daily newspaper “Sueddeutsche Zeitung” for the last some months and contains more than 1,000,000 entries.

To every orthographic representation a possibly ambiguous list of linguistic categories is assigned. For example the German word “schon” can have the categories particle and adverb.

Abbreviation dictionary:
In this dictionary more than 8,000 common German abbreviations are listed.

Acronym dictionary:
This dictionary contains about 13,000 entries.

Digits dictionary:
As a preliminary workaround the number handling, routine is based on a finite state grammar for German language, together with a dictionary of 120 basic elements together with their full forms.

0
*znull *znull.ADJC
0.
*znullst
1
*zein *zein.DET2:neNz:aeNz:neMz
1.
*zerst

Ongoing research is going to replace this simple routine by a data driven approach generalising underlying structures from a small number sample set.

Token delimiter list:
The initial token delimiter list used in the lexical scanner comprises some twenty entries such as blank, hyphen and punctuation marks.

PEOS list:
The possible end of sentence positions to be marked as PEOS tokens are denoted by one of the six symbols in the PEOS list:

. ! ? ; EOF

5. DISCUSSION AND CONCLUSION

This paper presents the separation of a text analysis module into a language independent core engine and language dependent knowledge sources. As an implementation example German language was used. Ongoing work for Slovenian and English show promising intermediate results.

A favourable structure of a queuing mechanism for subsequent but intertwined processing stages was
presented. Together with a final state machine coding [4], and state of the art hash tables this architecture results in a processing speed of about 1,300 tokens per second on a fast PC.

Crucial for the effectiveness of this approach for the fast and easy integration of new languages is the performance of data driven methods for the generation or completion of the language specific knowledge sources. The availability of big dictionaries and corpora together with more sophisticated modelling techniques enable a broader coverage and generalisation capabilities.

6. REFERENCES


