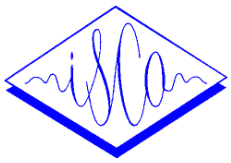

3rd International Conference on
Spoken Language Processing (ICSLP 94)
Yokohama, Japan
September 18-22, 1994

TEXT PROCESSING *WITHIN* A SPEECH SYNTHESIS SYSTEM

Anders Lindström *Mats Ljungqvist*

Telia Promotor Infovox AB, Box 2069, S–171 02 Solna, Sweden
E-mail: `firstname.lastname@infovox.se`

ABSTRACT

The first stage of text-to-speech (TTS) conversion involves analyzing the text to determine the correct pronunciation of the individual words or word groups.

Building upon a previously devised modular architecture for text processing within the context of a TTS system, we describe recent development in each of the areas "Word Pronunciation" and "Text Analysis", and describe the different modules involved. We describe how the tokeniser interacts with the lexicon in handling word groups (collocations), abbreviations and typing case. We also describe and give further references to on-going work, where the output of a probabilistic part-of-speech tagger will be used as input to a prosodic phrasing algorithm.

BACKGROUND

The aim of this paper is to describe recent development in the field of text processing for text-to-speech (TTS) conversion of Swedish. The work is carried out in the general framework of multi-lingual, high-quality speech synthesis [12].

The problems that arise within the first part of TTS conversion are indeed complex, and involve knowledge from several domains, such as lexicography, morphology, syntax, semantics, as well as less formalised knowledge regarding typing conventions etc. This was discussed in a previous paper [11].

Traditionally, the second part of TTS conversion, which involves the actual sound generation process, has always been the bottleneck, due to its immediate effect on the segmental intelligibility and naturalness of the synthetic speech. Accordingly, most effort has gone into enhancing that part of TTS systems, but as qualities such as naturalness and intelligibility of synthetic speech increase for isolated syllables, words or phrases, there is also an increasing need to improve the various components in the first stage, in order to achieve improvements when synthesizing longer stretches of text, particularily when the text is in principle unrestricted in domain. The improvements to be made and features to be desired can be divided into the following two categories:

Word Pronunciation: Fewer errors in word pronunciation. This includes dealing with neologisms, proper names, numerical expressions, as well as proper treatment of orthographic conventions, such as abbreviations, typing case etc.

Text Analysis: Robust strategies for determining word accents, assigning prominence and deriving grouping and phrasing information.

Recently, these areas have been treated by Liberman and Church [10] and by Riley and Sproat [14].

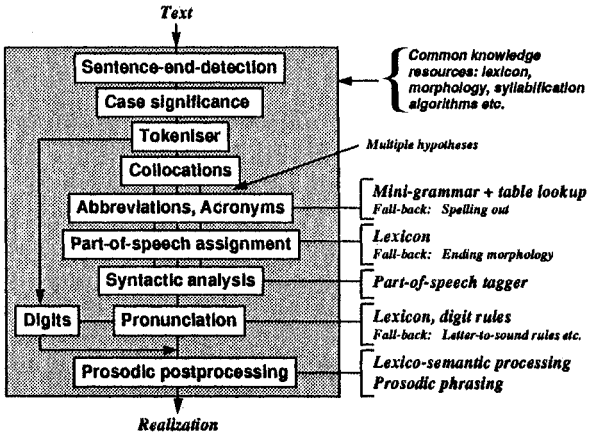

Figure 1: *System architecture for the text processing in text-to-speech conversion.*

A MODULAR APPROACH

Our approach has the following properties: It is *modular*, which means that special care has been taken to separate the knowledge into different *knowledge sources* (KS:s). One of the KS:s is a large, lemmatized lexicon for Swedish [7], which features inflectional and derivational morphology including compound word formation, case sensitivity and multi-word expressions (collocations). The lexicon, as well as other KS:s is furthermore capable of providing *multiple hypotheses*. Other KS:s detect end-of-sentence, handle abbreviations, acronyms, numerical expressions and typing case etc. One further feature of the approach is that the KS:s are all *within* the TTS system, and are allowed to share knowledge with each other. An outline of the system architecture is shown in Fig. 1.

The suggested approach offers several advantages, one being that the modularity makes knowledge development

	SUC subcorpus	
	Occurrences	Per cent
Surface forms in the corpus	282,138	
Non-lowercase words (NL)	23,365	of corpus: 8.3%
Lexical lookup of NL: number of word hypotheses when *ignoring case information*	40,676	
Lexical lookup of NL: number of matches (no hypotheses or compound generation)	22,896	of NL: 98%
- of which found in original case	8,000	of NL: 34%
- of which found in complementary case	14,896	of NL: 64%

Table 1: *Corpus-weighted statistics on typing case*

and knowledge maintenance easier. Another main benefit of the proposed scheme is that ambiguities of a variety of kinds can get a satisfactory resolution. This is the indirect result of the combination of using an approach handling multiple hypotheses and allowing the individual modules to incrementally add to the solution. This way, decisions do not have to be made until sufficient evidence has emerged on which to base the decisions. Another property of the system design is that different modes (e.g. different reading modes) can be implemented by simply changing the knowledge content in single modules *or* by changing the control structure. In a similar way, adaptation to different domains can be made through exchange of entire modules.

In the present paper, we describe recent development in each of the areas "Word Pronunciation" and "Text Analysis", and describe the KS:s involved. We describe how the tokeniser interacts with the lexicon and give examples of how imposing simple, common-sense constraints regarding typing case can considerably reduce the number of hypotheses, and thereby ease the burden on the parser, by producing none but the *feasible* word (or word-sequence) hypotheses. We also describe and give further references to on-going work, where the output of a probabilistic part-of-speech tagger will be used as input to a prosodic phrasing algorithm.

WORD PRONUNCIATION

Following the line of argument given in [11], we have opted for an approach which relies primarily on a large lexicon, and only to a smaller extent on rule-based methods. Using a large lexicon is essential in order to obtain high pronunciation accuracy, but the sheer size of the lexicon makes the problem of choosing between multiple hypotheses much more difficult. Therefore, an effort has been made to draw upon the various typing conventions that appear in ordinary texts, such as the use of the upper-/lowercase distinction and the (different) uses of punctuation marks in abbreviations.

Case Significance

Most text, written in latin script, follows some conventions regarding typing case. For instance, proper names, as well as the first word of a sentence are normally capitalized. In German, nouns are distinguished from other word-classes by being capitalized. Uppercase is usually used for acronyms, but can also be used for emphasis, or (as in this paper) for headings. A TTS system can obviously benefit from the information that typing case conventions supply.

i fall *conjunction* /i'fall/ (in case)
i går *adverb* /i'gå:r/ (yesterday)
av och till *adverb* /'a:vå'till/ (occasionally)
här om natten *adverb* /hä:råmn'atten/ (the other night)
till sjöss *adverb* /ti$'ös/ (to sea)
Motala ström *place name* /mo:tala'strömm/

Figure 2: *Examples of collocations included in the lexicon*

To estimate the size of the problem and what can be gained, the following calculations were made: In running text, taken from the SUC corpus [4], 8.3 % of the words contain capital letters (the vast majority are capitalized), and can be said to be "non-lowercase", NL. When looking up those words using the lexicon's built-in compounding and derivational morphology and *ignoring case information* (i.e., the words were mapped to upper-case, lower-case and capitalized before look-up) 40,676 word hypotheses were generated. The reason for this number being larger than the original amount of NL words, is that the lexicon finds all homographs. When the built-in compounding morphology was turned off, and the influence of homographs was eliminated, there were 22,896 matches, corresponding to a coverage of about 98 % of the lexicon. 8,000 of these matches were found in their original case while 14,896 were found in their complementary case, the figures are shown in Table 1.

Many first generation TTS systems did not take typing case into account at all, but used a lexicon where this information was "normalized away". One can conclude, from the figures in Table 1, that the method of disregarding case information breaks down when using a large lexicon, on the other hand, looking up every word exactly as written is too restrictive. Instead, *case significance* (CS) should first be determined. Typing case is typically *not* significant if all the words within a particular text window are written using one and the same case. The consequences of CS are then taken when the words are looked up in the lexicon.

Collocations and Tokenization

The identification of collocations, i.e. recurrent combinations of words as they appear in context, is something which could considerably increase the naturalness of synthetic speech. In human speech, collocations act as prosodic units and are subject to a higher degree of reduction and internal coarticulation than they would be, had they been ordinary, separate words. In accordance with Hedelin and Huber [7], a primarily lexical approach is chosen for handling collocations. Currently, there are several hundred common collocations in the lexicon, not counting particle verbs or reflexive verbs, which amount to several thousand. Some of these are shown in Fig. 2.

It is very common that TTS systems treat "space" as a word delimiter. This is obviously essential to avoid when dealing with collocations (and also in some other cases, which are mentioned below). Therefore, we've given the tokeniser the ability to tokenize "across space" in order

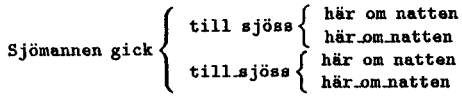

Figure 3: *A "sentence tree" for the Swedish sentence* The sailor put out to sea the other night

to form multi-word expressions. The tokeniser relies on the lexicon for information on possible collocations, and builds a "sentence tree", as that shown in Fig. 3, containing *all* possible tokenizations of the sentence. One reason for dealing with this problem at this stage, rather than when tagging, is the following: It should make a difference for the syntactic analysis component to treat a collocation as *one* constituent, rather than as the sum of its components. For instance, if to_and_fro is seen as an adverbial, rather than as a sequence of three tags[1], the tagger can cluster this adverbial with others it has "seen".

The treatment of abbreviations and acronyms follows what was reported in [11]: Only one generic form of the abbreviation is stored in the lexicon, e.g. "t_ex_" ("for example"). A mini-grammar of different typing conventions that are commonly used is fitted between the lexicon and the tokeniser, so that the generic form matches all the possible ways of typing it: "t ex", "t. ex.", or "t.ex.".

Lexicon

The Swedish lexicon [7] contains approximately 116,000 entries (including abbreviations, acronyms, common compounds). With the addition of around 30,000 proper names [2] it expands into approximately 900,000 surface forms. The lexicon is lemmatized, i.e., each head-word in the lexicon is associated with an inflexion rule from a set of approximately 500 rules that is capable of generating all possible inflexions of the word. Each word is listed in the lexicon with its part-of-speech label, inflexion code, phonetic transcription(s), information regarding its morphological make-up, as well as derivational and compound information. The lexicon handles compound words, either by explicit listing or by algorithmic generation. This is a necessity in languages like Swedish and German where compounds are written without indication of internal word boundaries.

Words that are not listed in the lexicon are assigned sets of plausible part-of-speech labels based upon their suffixes. These are later resolved by the part-of-speech tagger. The phonetic transcriptions of unknown words are generated by grapheme-phoneme rewrite rules. Similarly, phonetic transcriptions of cardinal, ordinal and decimal numbers are generated algorithmically.

The performance of the lexicon was investigated using a corpus of 100,000 words (10,000 unique surface forms) extracted from some of the Swedish texts of the SUC [4]. The coverage varied between 96 % and 99 % on these texts. About 20 % of the look-ups generated multiple hypotheses.

Name Pronunciation The issue of name pronunciation has received quite some intererest in the past few years, mostly due to the range of possible applications involving synthesis of names. Some 30,000 Swedish proper names (including 22,000 surnames and 6,000 first names) are available from previous work on proper names [2], and have been integrated in the main lexicon. The European LRE project *Onomastica* will produce quality-controlled transcriptions of proper names, street names and geographical names derived from the telephone directories of the 11 telecom partners [5].

TEXT ANALYSIS

Segmenting the text

A modern TTS system should be capable of producing appropriate macroprosodic realizations to indicate the structure of a text, e.g. its division into sections and paragraphs. This is particularly important when synthesizing longer stretches of text. If available, information regarding the underlying semantic and pragmatic structure of the text, for instance in terms of topic boundaries, could be used, as suggested by recent work in information retrieval [6].

Lexical semantics Another related and important aspect of text analysis involves keeping track of already mentioned items or concepts, so that they can be deaccented when re-encountered in the text. This has been explored in restricted domains by Horne *et al.* [9].

Even when the domain is less restricted, it is of vital perceptual importance that the TTS system should at least be capable of dividing the text into sentences, paragraphs, and perhaps sections, so that this information can be conveyed to the listener by means of prosody.

End-of-Sentence Detection The sentence is one of the textual units that it is important to be able to detect with some accuracy. We've used a pattern-matching approach, described in [11], which yields high accuracy.

Syntactic Analysis

For a TTS system, syntactic analysis serves two purposes: One task is to achieve homograph disambiguation, particularily in the case of heterophone homographs, but also in the case of differing word senses. The other task is to provide information that is useful for the later realization stages, particularily for prosodic realizations.

Tagging for Prosody It is debatable what type of parsing is actually needed in TTS conversion, and in particular for prosodic purposes. Bachenko and Fitzpatrick showed that, at least for English, it is possible to obtain improved prosodic phrasing by applying grouping algorithms to the output of a probabilistic part-of-speech tagger [1]. The overall idea is to use the stream of word/tag pairs, output by the tagger, and apply algorithms that try to build up prosodic phrases, and assign salience values to the boundaries between phrases.

[1]The third "word" can probably not be found in any lexicon, other than as part of this collocation.

A part-of-speech tagger based on hidden Markov modelling [3] is currently being ported to Swedish, using the SUC corpus and tag set [4]. When trained on 302,000 words of text from the genre-weighted SUC corpus, using a reduced tag set of 26 tags, 94 % of the words were tagged correctly (85 % of the ambiguous words). The output from this type of tagger can be used by prosodically motivated grouping and phrasing algorithms, as indicated in [8].

DISCUSSION

In this communication, we have described recent development in each of the areas "Word Pronunciation" and "Text Analysis", and described some of the *knowledge sources* (KS:s) involved. We have implemented these KS:s in the framework of a modular architecture, centered around a large, lemmatized lexicon. In this and previous communications, we have pointed out that this scheme simplifies domain adaptabilitity and allows for easier knowledge build-up and maintenance. We have furthermoref described how the tokeniser interacts with the lexicon when dealing with collocations and abbreviations. Frequent multi-word expressions, as well as multi-token abbreviations, are stored in the lexicon, while the tokeniser is given the ability to tokenize "across space", and give *all* sentence hypotheses, for later disambiguation. We have also given examples of how imposing simple, common-sense constraints regarding typing case can considerably reduce the number of hypotheses, and thereby ease the burden on the syntactic analysis component. On-going work was also described, where the output of a probabilistic part-of-speech tagger is used as input to a prosodic phrasing algorithm. The tagger has been adapted to Swedish using a genre-weighted corpus material, and is capable of high-accuracy tag assignment.

As regards word pronunciation, there is further work to be done in the field of collocations. The current strategy only caters for sequential collocations, whereas dependencies over longer stretches can't be captured. Some of the latter could be treated in the prosodic post-processing module.

Within the field of text analysis, statistical approaches for end-of-sentence detection could possibly be considered for easier adaptation to other domains or languages [13].

It should also be investigated to what extent topic boundary location can be used in conjunction with the prosodic phrasing algorithms described here. Another issue for further study is what demands the tasks "homograph disambiguation" and "extract prosodically relevant information" put on the choice of tag set for the part-of-speech tagger, to what extent those demands are conflicting, and how they affect tagging accuracy.

References

[1] J. Bachenko and E. Fitzpatrick. A computational grammar of discourse-neutral prosodic phrasing in English. *Computational Linguistics*, 16:155–167, Sept. 1990.

[2] R. Carlson, B. Granström, and A. Lindström. Predicting name pronunciation for a reverse directory service. In *Proc. of the European Conference on Speech Technology*, volume 1, pages 113–116, Paris, 1989.

[3] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. A practical part-of-speech tagger. In *Proc. of the 3 rd Conference on Applied Natural Language Processing*, pages 133–140, Trento, Italy, 1992.

[4] E. Ejerhed, G. Källgren, O. Wennstedt, and M. Åström. The linguistic annotation system of the Stockholm–Umeå corpus project. Publications from the dept. of general linguistics, Umeå, Dept. of General Linguistics, Umeå University, December 1992.

[5] J. Gustafson. ONOMASTICA—creating a multi-lingual dictionary of European names. In G. Bruce, D. House, and P. Touati, editors, *Papers from the Eighth Swedish Phonetics Conference*, volume 43 of *Working Papers*, pages 66–69, Lund, Backagården, May 1994. Lund University.

[6] M. A. Hearst. Multi-paragraph segmentation of expository text. In *Proc. of the 32 nd ACL*, Las Cruces, NM, June 1994. ACL.

[7] P. Hedelin and D. Huber. A new dictionary of Swedish pronunciation. In K. Morland and K. Sørstrømmen, editors, *Proc. of the Scandinavian Conference in Computational Linguistics*, pages 105–117. Norwegian Computing Centre for the Humanities, Bergen, Norway, 1991.

[8] M. Horne and M. Filipsson. Generating prosodic structure for Swedish text-to-speech. In *Proc. of the Third Intl. Conf. on Spoken Language Processing*, Yokohama, 1994.

[9] M. Horne, M. Filipsson, M. Ljungqvist, and A. Lindström. Referent tracking in restricted texts using a lemmatized lexicon: Implications for generation of intonation. In *Proc. of the European Conference on Speech Technology*, volume 3, pages 2011–2014, Berlin, 1993. ESCA.

[10] M. Y. Liberman and K. W. Church. Text analysis and word pronunciation in text-to-speech synthesis. In *Proc. of the DARPA Workshop on Speech and Natural Language Processing*, 1989.

[11] A. Lindström, M. Ljungqvist, and K. Gustafson. A modular architecture supporting multiple hypotheses for conversion of text to phonetic and linguistic entities. In *Proc. of the European Conference on Speech Technology*, volume 2, pages 1463–1466, Berlin, Sept. 1993. ESCA.

[12] M. Ljungqvist, A. Lindström, and K. Gustafson. A new system for text-to-speech conversion, and its application to Swedish. In *Proc. of the Third Intl. Conf. on Spoken Language Processing*, Yokohama, 1994.

[13] D. D. Palmer and M. A. Hearst. Adaptive sentence boundary disambiguation. Technical Report UCB/CSD-94-797, Computer Science Division, UC Berkeley, Feb. 1994.

[14] M. Riley and R. Sproat. Text analysis tools in spoken language processing. In *Proc. of the 32 nd ACL*, Las Cruces, NM, June 1994. ACL.