THE STRUCTURAL DESIGN OF THE CSTR TEXT TO SPEECH SYSTEM

Mike McAllister*

ABSTRACT

As part of the Text-to-Speech research at Edinburgh University's Centre for Speech Technology Research (EU_CSTR), a modular, linguistic knowledge based text-to-phoneme system has been implemented in Prolog. Its design considerations, the structure and coverage of its rule bases and typical output from the system are described in the body of this paper.

INTRODUCTION

The speech output research at EU_CSTR is centred around the four-year Text to Phoneme Project funded by NEC of Japan. This project has now completed three years of its four year span, the remit of which period is the design of a knowledge-based linguistic processing system capable of producing an annotated phoneme sequence from unlimited but well-formed text. This accelerated concentration on the higher-order aspects of speech generation is an ideal complement to the long standing tradition of speech synthesis research in the University's Linguistics Department.

As a development and evaluation tool a software system for text to phoneme translation has been implemented in Prolog. It is not necessary here to describe those features of the Prolog programming language which lead to its choice for this purpose, but for readers unfamiliar with it, it should be sufficient for the remainder of this discussion to categorise Prolog as a declarative language with powerful pattern-matching abilities and whose data structures are list-based rather than record-based.

MODULARITY

The system operates as a sequence of processing modules the rules and rule application techniques of each of which are independent of the others in terms of format but must be coordinated in terms of functional demarkation and dependence.

To illustrate, the grapheme to phoneme conversion module has an internal rule representation which is unlike any other in the system. This information and this representation is of no consequence to any other module of the system so long as it is only used internally. The morphological decomposition module passes undecomposable, unknown (to the lexicon) word stems to the grapheme to phoneme module and is returned a transcription of these stems. Both the stems and the transcriptions are in data formats agreed in advance of the implementation of each module and are all the information that each module needs about the other in terms of form.

However, the coverage of each of these two modules is directly interrelated. The decomposition module attempts to recognise, validate and stem-adjust for all of the affixes in its rule base. This means that the grapheme to phoneme rules do not attempt to model the pronunciation variations of common affixes, for example the pronunciation of the -ed suffix in the words 'fated', 'famed' and 'faced' is /ih d/, /d/ and /t/** respectively in the three cases. These pronunciation variations are dealt with by another module in the system. This functional coordination is, therefore, tripartite between the morphological decomposition, grapheme to phoneme conversion and morphophonology modules.

*Affiliation:
Centre for Speech Technology Research, Edinburgh University, 80 South Bridge, Edinburgh, EH1 1HN, Scotland.

** All transcriptions use the Edinburgh University Machine Readable Phonetic Alphabet (EU_MRPA)
SYSTEM MODULES

The full list and functional responsibilities of the modules of the system is as follows.

The Anomaly Processor reads text from the keyboard or from a file one sentence at a time according to its rules on valid sentence terminating punctuation. It recognises sentence structure punctuation and separates it from the strings to which it is appended. The treatment of abbreviations, proper names and other anomalies is based on the typographic form of the anomaly, i.e. the absence or presence of capitals, numbers, valid intra-word punctuation characters (hyphen, slash and apostrophe) and other punctuation characters. The dictionaries serving this module are also typographically organised.

The Morphological Decomposition module recognises and validates affixes in its rule base. It also deals with clitics and single word compounds whether marked by hyphenation or the slash character or unmarked. Much of the validation of potential affixes is based on the permissible word classes that a stem might be in order to attach to such an affix. Each affix is therefore marked with information concerning the word classes it may attach to and the word classes in may cause a word to become. The adjustment of word stems is also performed during the affix stripping process in order that stems may be found in the lexicon or underlying affixes stripped in their turn.

The Morph Dictionary should be described at this point to illustrate its central role in the functional coordination of the linguistic processing modules. The dictionary is accessed via the head-word of the entry. The entry is a list of items, some of whose positions are fixed for speed of access, where their existence is guaranteed, the remainder of which may be potentially unlimited in number and unrestricted in order.

A morph dictionary entry will always contain a head-word, a phonemic transcription and a word class list. The remainder of the entry is a list containing items which may be flags to later system mechanisms or might be described as attribute/value pairs in another formalism. For example, language of origin is an attribute which is only marked in a small minority of entries which are fairly recent borrowings into the language. Morphophonological adjustments which are associated with the co-occurrence of particular stems and certain affixes are flagged, to be detected further on in the system, e.g. the pronunciation alteration in the stem 'divine' when it occurs with the suffix "-ity" in 'divinity.

The strategic advantage of a list with a few fixed position items is as follows. Those items which are used in every word processed can be accessed by canonical order as quickly as if a record structure had been used. However, complete freedom is afforded in the tail of the entry to add new system flags as often as is desired without altering any of the access mechanisms which examine data held in the dictionary entries.

The Word Class List Construction module is a system integration data normalisation mechanism which coordinates information about word classes associated with affixes and those marked in the dictionary. This provides the word class disambiguation module with input data where each word in a sentence is represented by a single word class or a single list of word class candidates. Because of the coverage of closed classes by the dictionary, any form about which there is no information can be assumed to be a content rather than function word and is therefore labelled as a possible noun, verb, adjective and adverb.

The Word Class Disambiguation module is a strategic combination of two different parsing/disambiguation techniques.

The ubiquitous Prolog re-write grammar rule formalism is used in the first stage, where the rule translator and application mechanism have been adapted to seek confirmation of word class hypotheses from membership of the word class candidate list representing each word. When such confirmation is found, the word class for that word is fixed by that process. This grammar mechanism, although well understood and facilitating an extremely clear and concise rule formalism contains no strategy for near success or 'best fit' (or more particularly 'least poor fit') of parsing/disambiguation attempts. The second word class disambiguation engine remedies this situation, but is only called upon in the small minority
of cases when the re-write rule grammar has failed.

The second parsing strategy is a window-based disambiguation of word class candidates which operates by matching word class candidate sequences against distributional patterns corresponding to high frequency noun phrase and verb phrase structures. The distributional patterns for this engine are expressed in a rule format which is based on a central target item to be disambiguated and contains information about what classes may be in the candidate list both for the central item and its surrounding item both optional and compulsory.

As such, this distributional window engine is specifically focussed on disambiguation between alternative classifications of each ambiguous word and is not a purely bottom-up technique, whereas the re-write rule grammar can be seen as a traditional top-down approach.

The **Affix Disambiguation** module serves to choose between alternative accentual properties of various affixes. In doing so, it also chooses between pronunciation variants for the affixes concerned, and as such it must operate before the first of the pronunciation transformation modules (Morpho-phonology). This module applies its rules with reference to the free/bound status of word stems, word class and various attributes which have or have not been flagged in the dictionary entries.

The **Morpho-Phonology** engine applies a set of morph boundary phonological adjustment rules to poly-morphemic words. These rules may be described as using a 'before-and-after' format, where the salient phonemes on either side of the morph boundary are listed in the rule both in the target and result state. An embedded test facility allows stems to be checked for various markers which have been extracted from the dictionary or provided by the system at a later stage.

The **Lexical Stress Assignment** module assigns primary stress to the phonemic transcription of words on the basis of the accentual properties of its affixes, previously identified by the morphological decomposition module, or on the basis of the word's syllabic structure. Its algorithm is an enhancement of the systematisation first devised by Erik Fudge (see Bibliography). Secondary stress is assigned rhythmically to words in which the primary stress is on the third or later syllable. Pseudo-compounds of the "telegraph" and "telemetry" sort are dealt with here, again by an enhancement of the algorithm which Fudge suggests.

The **Vowel Reduction** module was initially implemented as a stand alone module, but the intricacies of the pseudo-compounds algorithm in the lexical stress assignment module have caused it to be called as a subordinate routine. Its operation remains autonomous, however, testing and reducing syllables in a list on the basis of their relative position to the ends of the word, the stressed syllable and their phonemic composition. Both the stress assignment and vowel reduction modules use a maximal onset syllabification strategy.

The **Sentence Level Phonology** engine is a duplicate of the morpho-phonology rule application engine except that it operates at word and not morph boundaries. It adjusts word initial and word final pronunciation to deal with such phenomena as 'linking-r' and the strong and weak forms of 'a' and 'the'.

The **Intonation** module is described in greater detail elsewhere in this volume. Briefly, it introduces into the phoneme string, that now represents the pronunciation of the input sentence, target frequencies for F0 to be interpolated between. To perform this task it is provided with a syntactic analysis of the sentence identifying the major constituents and a syllabified stressed transcription of each word in the sentence. The coordination of these two lists is by canonical order, as used by the affix disambiguation and lexical stress assignment modules.
AN INFORMATION PROCESSING VIEW

The text-to-phoneme system described can be viewed in terms of information generation, refinement and transformation.

In broad brush terms the first four modules (Anomaly Processing, Morphological Decomposition, Grapheme to Phoneme Conversion and Word Class List Construction) are all generating information from the raw input string. This information is refined by the two central filtering modules: Word Class Disambiguation and Affix Disambiguation. From then on various transformations on the phonemic representation are performed by the Morpho-phonology, Lexical Stress Assignment, Vowel Reduction and Sentence Level Phonology modules.

The Intonation module once again generates information, but this is symptomatic of its pivotal position in the process of Speech Generation, about which the swing from phonemic to phonetic processing takes place. This is illustrated by the fact that it generates one of the essential parameters for the synthesis of the individual allophones, based on information about the syntactic structure of the sentence and the accentual structure of the words which comprise it.

BIBLIOGRAPHY