TEXT-TO-SPEECH RULE AND DICTIONARY DEVELOPMENT

R.Mannell and J.E.Clark

Speech, Hearing and Language Research Centre
Macquarie University

ABSTRACT - The development and evaluation of a set of letter-to-sound rules and their associated lexicon and suffix stripping rules is outlined. These rule/databases form part of a complete text-to-speech system being developed at the Speech, Hearing and Language Research Centre.

INTRODUCTION

A general outline of the SHLRC text-to-speech system is shown in figure 1. This paper focuses on the development and evaluation of the letter-to-sound module. This module consists of three major routines. The first is a lexicon or dictionary which deals with the most common words and the most important exceptions to the letter-to-sound rules. This is augmented by a suffix stripper routine which identifies and temporarily removes suffixes from the input word and searches for the resulting root word in the lexicon. This means that the lexicon can now cover not only the words in the lexicon but also those derivatives of these words which can be derived from them by adding one (or more) of the most common suffixes. Those words that are not covered by the suffixes and the lexicon are then submitted to a third routine, a letter-to-sound rule routine, which produces broad phonetic script using rules supplied in the rule base. The remainder of this paper describes the development of these last three routines (ie. the rules, lexicon and suffix stripper).

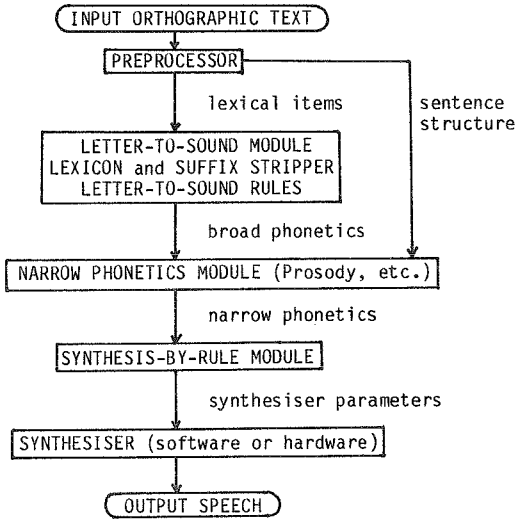

Figure 1. An overview of the SHLRC text-to-speech system

LETTER-TO-SOUND RULES

Laver and Clark (1982) converted a set of letter-to-sound rules originally developed by the Naval Research Laboratories (Elovitz et al, 1976) into a system capable of producing Australian English. These rules have been undergoing continuous refinement at SHLRC ever since.

The rules are written in an easily accessible rule language formatted in a way which is quite intelligible to most linguists. In order to link the rules to the TTS system they must first be passed through a pre-compiler which converts them into FORTRAN subroutines. The rules are accessed sequentially until a rule which satisfies the current part of the input string is found. This procedure defined by the rule is then performed, and a flag is incremented to point at the next unprocessed part of the input string, and so on until the whole string has been converted.

Some typical rules are as follows :-

```
#(CH)[C<1>]    = /k/          eg. Christmas
[C<1>]E(CH)    = /k/          eg. technical
(CH)           = /tʃ/         default "CH" rule
(C)[V<f>]      = /s/          eg. ceiling
(COM)[SUFFIX]  = /kʌm/        eg. coming
(CC)[V<1-n>]   = /ks/         eg. accept
(C)            = /k/          default "C" rule
```

where :- () contain the string to be modified
All other characters to the left of the "=" represent the orthographic context of the characters in the ()
[] surround codes, not orthographic characters
eg. C = consonant
V = vowel
SUFFIX = one of a list of suffixes
<> modifies the code in []
eg. [C<1>] = one consonant
[V<1-n>] = any number of vowels
[V<f>] = a front vowel
Letters not surrounded by any of the above are other orthographic characters which define the context.
= word break

The development and evaluation procedure described below has caused the deletion of existing rules, the insertion of new rules, the modification of existng rules, and the re-ordering of the rules to obtain better performance.

LEXICON

The lexicon described below was originally designed (1) with the intention of capturing as large a percentage of continuous running text as possible. This was achieved by examining two English language word count corpora (the Lancaster-Oslo-Bergen (LOB) British English corpus and the Brown American English corpus). It turned out that about 4500 words accounted for about 90% of all words used in continuous text, and so these words were added to the lexicon. This was done to reduce the use of the rather time consuming letter-to-sound rule system. Further, the most common exceptions to the rules were also added, especially if it turned out that the word could only be made to come out right by creating a rule for it alone.

Lexicon format and content has undergone considerable modification since it was first developed, however its content is still guided by the above considerations.

A typical lexicon entry is as follows:-

" a \P eɪ \G adj nnn art"

In this format, the headword field is fixed (15 characters) to facilitate indexing, whilst the remaining fields are variable. Further, all part-of-speech labels have been standardised to a set of several three-letter labels.

Words which are pronounced differently for different parts of speech, have each grammatically-meaningful pronunciation in the "\P" field. If a part-of-speech is not followed by a number in a bracket it takes the first part-of-speech. If it is followed by a number in a bracket it takes the correspondingly numbered pronunciation.

eg." abstract \P ˈæbstrækt əbˈstrækt \G adj nnn vbt(2)"

where the noun and adjective take the first pronunciation /ˈæbstrækt/ and the transitive verb (tagged with a (2)) takes the second pronunciation /əbˈstrækt/.
Certain other features were also included, but these are more appropriately discussed with the suffix stripper.

SUFFIX STRIPPER

A two-part suffix stripper rule-base and database has been designed to augment the lexicon and to increase its capture range. The first part of the rule/database contains words which frequently participate in compound word formation whilst the second part contains the majority of the normal suffixes. A typical suffix database entry is as follows :-

" -est ,[a,ɜ,ə,ɔ]=/Rəst/,*=/əst/"

which means
"when the suffix "-est" is to be attached to a word whose phonetic representation ends with one of the phonemes /a/,/ɜ/,/ə/,or/ɔ/ then this suffix is pronounced /Rəst/, otherwise, it is pronounced /əst/." ("R" is a linking /r/)

The evaluation procedure (below) helped to focus on some existing shortcomings in the database. For example, the original rule set did not allow for a linking /r/ for the suffix "-est' when it follows a word ending in /...ə/, but the new rule now includes the phoneme /ə/. This essential rule condition was missing from the rules of 26 suffixes.
A further type of rule was not even considered in the original database. That is a rule which states that when a certain suffix is added to a word ending in a certain phoneme, that phoneme is changed into another phoneme.

eg. " -ion ,[z]=/ʒən/,*=/ʃən/,{s>},{st>st},{t>},{z>}"

which means
"when the suffix "-ion" is to be attached to a word whose phonetic representation ends with the phoneme /z/, then this suffix is pronounced /ʒən/, otherwise, it is pronounced /ʃən/. However, a terminal /s/, /t/ (unless

that /t/ is preceeded by an /s/) or /z/ is first deleted."
eg. administration - /...eɪt/ + "-ion" becomes /...eɪʃən/

In other words, the rules in {} describe the fate of terminal phonemes in the root word, before the addition of the suffix. The first rule to be accessed which matches the root's terminal phoneme(s) is the rule used. Subsequent rules are then ignored. Therefore if terminal /t/ is to be deleted, but not terminal /st/ then the rules are ordered "{st>st},{t>}".

In the original suffix stripper algorithm, when a suffix deletion resulted in a modified version of the root word

eg. running ⟶ runn + -ing
happiness ⟶ happi + -ness
saved ⟶ sav + -ed

the modified root was searched for in the lexicon. This was possible because the headwords were entered into the lexicon as follows : -

\H run{double}
\H happ{y,i}
\H sav{e, }

Such a representation would have made indexing difficult, so instead certain rules were added to the lexicon subroutine. These rules state that if the word in the search buffer is not found and the suffix strip flag is set then :-

IF the word ends in two identical consonants then delete one and search for the result in the lexicon.
IF the word ends in an "i", then replace it with a "y" and search for the result in the lexicon.
IF the word ends in a consonant, then add an "e" and search for the result in the lexicon.

The suffix stripping algorithm runs into trouble when the root remaining after removing the suffix is identical to another word in the lexicon.

eg. barred ⟶ barr + -ed --> bar + -ed is correct, but
bared ⟶ bar + -ed is not correct.

To overcome this problem, any words so affected have had an extra field added to their lexicon entry. This "\H" field refers the lexicon routine to the appropriate headword if the suffix strip flag is set and if the suffix belongs to the second part of the suffix database.

eg1. bared ⟶ bar + -ed
In the lexicon the following entry is initially found.
" bar \H bare \P ba \G nnn prp vbt"
Since the suffix flag is set, and the suffix "-ed" is in the second part of the rule/database then the headword in the "\H" field is searched for,
ie. " bare \P bɛə \G adj vbt"
and the pronunciation /bɛəd/ is derived.

eg2. barman ⟶ bar + -man
In this case the suffix flag is set but the suffix "-man" is in the first part of the rule/database and so the pronunciation for "bar" is selected, giving the final pronunciation /bamən/.

The above alterations to the suffix stripping algorithm and its associated databases allowed quite a few, now redunant, entries to be removed from the lexicon leaving room for more high frequency exceptions to the TTS rules.

On the other hand, some exceptions to the new suffix rules needed to be added to the lexicon. The number of words removed far exceeded the number of words that needed adding.

EVALUATION AND DEVELOPMENT

Considerable improvement in the efficiency and accuracy of the rules has been achieved by producing an input word list which represents all the words in both the LOB and the Brown corpora, and which are also in the Macquarie dictionary database. This gave a total list of 18788 words. Each entry included the corpus word frequency count from both corpora and all pronunciations given in the Macquarie dictionary. A test version of the TTS program was then developed which allowed the entry of each word in this list to the top end of the TTS program (ie. the lexicon, suffix stripper and TTS rules) and then compared the output phonetic string with the acceptable pronunciations derived from the Macquarie dictionary. This allowed each word to be scored correct, almost correct (only schwa/vowel mismatches) and incorrect. Two sets of scores were derived as follows :-

i) Raw scores (ie. a score of one for each word)
ii) Corpus weighted scores (ie. a score for each word equal to the sum of the LOB and the Brown count for that word)

Further, it was deduced which rules were used and whether any rule was used correctly, almost correctly, or incorrectly, and a raw and a corpus weighted score was calculated.

All incorrect words were output to an errors file with their corpus scores and then this file was ordered in corpus count order with the most frequently occurring words at the top. Since it can be reasonably assumed that high corpus count words occur more frequently in normal running text than low corpus count words it was decided to determine the cause of errors in high frequency words first and to make adjustments to the TTS rules or suffix rules if possible, or otherwise to add the word to the lexicon. As a further aid to TTS rule enhancement, the rules with the highest corpus corrected error scores were examined closely, and in this way many rules were removed, repositioned, split into two or more rules of greater specificity, or replaced with a correct rule. In all, about 1/3 of the present TTS rules did not exist either at all or in their current form before this process was attempted.

Previous to adopting this methodology, about 3 months were spent improving the raw word score (ie. for the total system of lexicon plus suffixes plus TTS rules) from about 50% correct to 69% correct using a trial and error approach. Further, increased effort was eventually met with drastically diminishing results. Upon adopting this methodology, the raw score for words correct was easily improved by another 8% (to 77%) in just 7 working days, before diminishing returns again became a problem.
These raw results do not, however, properly reflect the accuracy of the system because many of the words would be rarely met in running text. At the beginning of the above 7 day period, the corpus weighted results showed 97.13% words correct. This improved to 99.14% by the end of that period. In other words, in an average text of 10000 words (including of course many repititions of the more common words) 9713 words originally, and 9914 words now, would be expected to be correct.

Figure 2 shows the percentage capture rate for words correctly output by the lexicon, lexicon plus suffixes, and the TTS rules for both raw and corrected scores. Clearly, the lexicon captures the largest number of individual tokens in average running text, whilst the TTS rules capture the largest number of unique words.

The suffix stripper is now accurate 88.00% of the time for unique words (ie. raw score) or 98.46% of the time for individual tokens in running text (ie. corpus-weighted score) for words not already captured by the lexicon. The TTS rules now have a corpus-weighted score of 85% correct (of words not already captured by the lexicon and suffix-stripper), which is about a 10% improvement over their previous accuracy. It seems likely that with more sophisticated error analysis these results can be improved to > 90%.

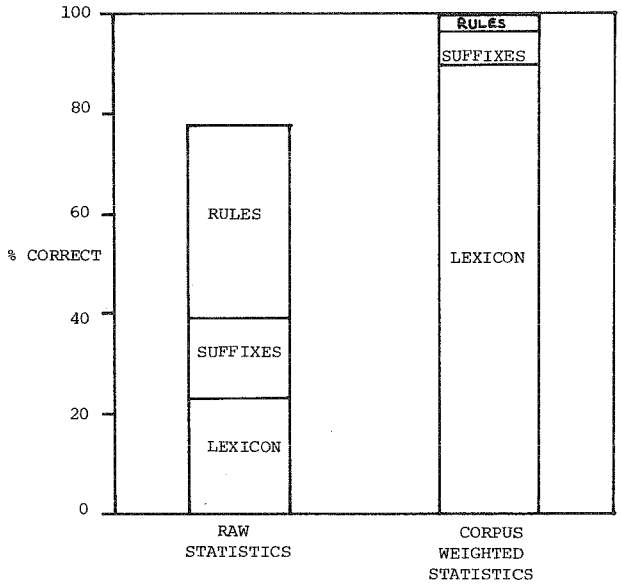

Figure 2. Raw and corpus weighted percentage capture rate of the rules, lexicon and suffix stripper.

NOTES

(1) The lexicon and suffix stripper were developed in their initial form by Clive Summerfield (at that time at SHLRC) and John Clark.

REFERENCES

LAVER, J. and CLARK, J.E. (1982), Australian English letter-to-sound rules based on the NRL rules, unpublished, pers.comm.

ELOVITZ, H., JOHNSON, R., McHUGH, A. and SHORE, J. (1976), Letter-to-sound rules for automatic translation of English text to phonetics, IEEE.Trans.Acoust.Speech and Sig.Proc., ASSP-24, 446-459.