Analysis of Unknown Words through Morphological Decomposition

Alan W Black
Dept of Artificial Intelligence,
University of Edinburgh
80 South Bridge,
Edinburgh EH1 1HN
Scotland, UK.
awb@ed.ac.uk

Joke van de Plassche
NICI,
University of Nijmegen,
Montessorilaan 3,
6525 HR Nijmegen
The Netherlands
PLASSCHE@kunpv1.psych.kun.nl

Briony Williams
Centre for Speech Technology
University of Edinburgh
80 South Bridge,
Edinburgh EH1 1HN
Scotland, UK.
briony@cstr.ed.ac.uk

Abstract

This paper describes a method of analysing words through morphological decomposition when the lexicon is incomplete. The method is used within a text-to-speech system to help generate pronunciations of unknown words. The method is achieved within a general morphological analyser system using Koskenniemi two-level rules.

Keywords: Morphology, incomplete lexicon, text-to-speech systems

Background

When a text-to-speech synthesis system is used, it is likely that the text being processed will contain a few words which do not appear in the lexicon as entries in their own right. If the lexicon consists only of whole-word entries, then the method for producing a pronunciation for such "unknown" words is simply to pass them through a set of letter-to-sound rules followed by word stress assignment rules and vowel reduction rules. The resulting pronunciation may well be inaccurate, particularly in English (which often shows a poor relationship between spelling and pronunciation). In addition, the default set of word classes assigned to the word (noun, verb, adjective) will be too general to be of much help to the syntactic parsing module. However, if the lexicon contains individual morphemes (both "bound" and "free"), an unknown word can be analysed into its constituent morphemes. Stress assignment rules will then be more likely to yield the correct pronunciation, and any characteristic suffix that may be present will allow for the assignment of a more accurate word class or classes (eg. `+ness` denotes a noun, `+ly` an adverb). Morphological analysis of words will therefore allow a significantly larger number of "unknown" words to be handled. Novel forms such as `hamperance`, and `thatcherisation` would probably not exist in a whole-word dictionary, but could be handled by morphological analysis using existing morphological entries. Also, the ability to deal with compound words would allow for significantly higher accuracy in pronunciation assignment.

A problem arises, however, if one or more of the word's constituent morphemes are not present in the morphological dictionary. In this case, the morphological analysis will fail, and the entire word will be passed to the letter-to-sound rules, with concomitant probable loss of accuracy in pronunciation assignment and word class assignment. It is far more likely that the missing morpheme will be a root morpheme rather than an affix, since the latter morphemes form a closed class which may be exhaustively listed, whereas the former form an open class which may be added to as the language evolves (eg. `ninja`, `Chunnel`, `kluge`, `yomp`). Therefore, it would be preferable if any closed-class morphemes in a (putatively) polymorphemic unknown word could be recognised and separated from the remaining material, which would then be assumed to be a new root morpheme. Letter-to-sound rules would then be applied to this putative new root morpheme (the pronunciation of the known material would be derived from the lexicon).

The advantages of this method are that the pronunciation and word stress assignment are more likely to be accurate, and also that, if there is a suitable suffix, the correct word class may be assigned (eg. in `yomping`, from `yomp` (unknown root) and `+ing` (known verb or noun suffix), which will be characterised as a verb or noun). Thus, in the case of `preamble`, the stripping of the prefix `pre-` will allow for the correct pronunciation `/p r ii a m b @ l/`: if the entire word had been passed to the letter-to-sound rules, the incorrect pronunciation `/p r ii m b @ l/` would have resulted. In addition to affixes, known root morphemes could also be stripped to leave the remaining unknown material. For example, without morphological analysis, `penthouse` may be wrongly pronounced as `/p e n th au s/`, with a voiceless dental fricative.

It is known that letter-to-sound rules are more accurate if they are not allowed to apply across morpheme boundaries (see [1, Ch. 6]), and this method takes advantage of that fact. Thus greater accuracy is obtained, for polymorphemic unknown words, if known morphs can be stripped before the application of letter-to-sound rules. It is this task that the work described below attempts to carry out.

The Alvey Natural Language Tools Morphological System ([5],[6]), already provides a comprehensive morphological analyser system. This system allows morphological analysis of words into morphemes based on user-defined rules. The basic system does not offer analysis of words containing unknown morphemes, nor does it provide a rank ordering of the output analyses. Both these latter features have been added in the work described below.

The system consists of a two tier process: first a morphological analysis, based on Koskenniemi's two-level morphology ([3]); secondly the statement of morphosyntactic constraints (not available in Koskenniemi's system) based on a GPSG-like feature grammar.

The morphographemic rules are specified as a set of high level rules (rather than directly as finite state transducers) which describe the relationship between a surface tape (the word) and a lexical tape (the normalised lexical form). These rules specify contexts for pairs of lexical and surface characters. For example a rule

```
+:e <=>
  { < s:s h:h > s:s x:x z:z y:i }
         --- s:s
```

specifies that a surface character `e` must match with a lexical character `+` when preceded by one of `sh`, `s`, `x`, `z` or the pair `y:i` (as in `skies` to `sky+s`), and succeeded by `s`. The "`---`" denotes where the rule pair fits into the context. For example the above rule would admit the following match

```
lexical tape:  b o x + s
surface tape:  b o x e s
```

The exact syntax and interpretation is more fully described in [5, Sect. 3] and [6, Ch. 2].

In addition to segmentation each lexical entry is associated with a syntactic category (represented as a feature structure). Grammar rules can be written to specify which conjunctions of morphemes are valid. Thus valid analyses require a valid segmentation *and* a valid morphosyntax. In the larger descriptions developed in the system a "categorial grammar"-like approach has been used in the specification of affixes. An affix itself will specify what category it can attach ("apply") to and what its resulting category will be.

In the work described here, the basic morphology system has been modified to analyse words containing morphemes that are not in the lexicon. The analysis method offers segmentation and morphological analysis (based on the word grammar), which results in a list of possible analyses. An ordering on these possible analyses has been defined, giving a most likely analysis, for which the spelling of the unknown morpheme can then be reconstructed using the system's original morphographemic rules. Finally, the pronunciation of the unknown morpheme can be assigned, using letter-to-sound rules encoded as two-level rules.

Analysis Method

The method used to analyse words containing unknown substrings proceeds as follows. First, four new morphemes are added to the lexicon, one for each major morphologically productive category (noun, verb, adjective and adverb). Each has a citation form of **. The intention is that the unknown part of a word will match these entries. Thus we get two-level segmentation as follows

```
lexical tape:  * 0 0 0 * + i n g + s
surface tape:  0 p a r 0 0 i n g 0 s
```

The special character 0 represents the null symbol (i.e. the surface form would be parings – without the nulls). This matching is achieved by adding two two-level morphological rules. The first rule allows any character in the surface alphabet to match null on the lexical tape, but only in the context where the lexical nulls are flanked by lexical asterisks matching with surface nulls.

The second rule deals with constraining the *:0 pairs themselves. It deals with two specific points. First, it ensures that there is only one occurrence of ** in an analysis (i.e only one unknown section). Second, it constrains the unknown section. This is done in two ways. Rather than simply allowing the unknown part to be any arbitrary collection of letters, it is restricted to ensure that if it starts with any of {h j l m n q r v x y z}, then it is also followed by a vowel. This (rightly) excludes the possibility of an unknown section starting with an unpronounceable consonant cluster e.g. computer could not be analysed as co- mput +er). Second, it ensures that the unknown section is at least two characters long and contains a vowel. This excludes the analysis of resting as re- st +ing.

These restrictions on the unknown section are weak and more comprehensive restrictions would help. They are attempts at characterising English morphemes in terms of the minimal English syllable. A more complex characterization, defining valid consonant clusters, vowels, etc. would be possible in this formalism, and the phonotactic constraints of English syllables are well known. However, the resulting rules would be clumsy and slow, and it was felt that, at this stage, any small gain in accuracy would be offset by a speed penalty.

The rules make use of sets of characters. Anything is a set consisting of all surface characters, BCDFGKPSTW and HJLMNQRVXYZ are sets consisting of those letters, V is the set of vowels and C the consonants. The character $ is used to mark word boundaries.

```
0:Anything <=>
   { *:0 < *:0 (0:Anything)1+ > }
   ---
     { *:0 < (0:Anything)1+ *:0 > }

*:0 <=>
   { 0:$ < 0:$ (=:=)1+ > } ---
       { < { 0:BCDFGKPSTW 0:V }
           0:Anything >
         < 0:HJLMNQRVXYZ 0:V > }
     or { < 0:C (0:V)1+ >
          < 0:V (0:C)1+ > } ---
       { < (=:=)1+ 0:$ > 0:$ }
```

The above rules are somewhat clumsily formulated. This is partly due to the particular implementation used, which allows only one rule for each surface:lexical pair[1] and partly due to the complexity of the phenomena being described.

Word Grammar

Using the above two rules and adding the four new lexical entries to a larger description, it is now possible to segment words with one unknown substring. Because the system encodes constraints for affixes via feature specifications, only morphosyntactically valid analyses will be permitted. That is, although ** is ambiguous in its category, if it is followed by +ed only the analysis involving the verb will succeed. For example, although the segmentation process could segment bipeds → ** +ed +s the word grammar section would exclude this analysis, since the +s suffix only follows uninflected verbs or nouns.

However, there are a number of possible mistakes that can occur. When an unknown section exists it may spuriously contain other morphemes, leading to an incorrect analysis. For example

```
colour -> co- **
readable -> re- ** +able
cartoons -> car ** +s (compound noun)
```

In actual fact, when words are analysed by this technique a large number of analyses is usually found. The reasons for the large number are as follows. Firstly, the assumed size of the unknown part can vary for the same word, as in the following:

[1] Ritchie ([4]) shows that this is not a restriction on the formal power of the rules.

```
entitled ->  **
entitled ->  ** +ed
entitled ->  en- ** +ed
entitled ->  en- **
```

Secondly, because `**` is four ways ambiguous, there can be multiple analyses for the same surface form. For example, a word ending in `s` could be either a plural noun or a third person singular verb.

These points can multiply together and often produce a large number of possible analyses. Out of the test set of 200 words, based on a lexicon consisting of around 3500 morphemes (including the `**` entries), the average number of analyses found was 9, with a maximum number of 71 (for `functional`).

Choosing an Analysis

In order to use these results in a text-to-speech system, it is necessary to choose one possible analysis, since a TTS system is deterministic. To do this, the analyses are rank ordered. A number of factors are exploited in the rank ordering:

- length of unknown root
- structural ordering rules ([1, Ch. 3])
- frequency of affix

Each of these factors will be described in turn. When analysing a word containing an unknown part, the best results are usually obtained by using the analysis with the shortest unknown part (see [1, Ch. 6]). Thus the analysis of `walkers` would be ordered as follows (most likely first):

```
** +er +s  >  ** +s  >  **
```

This heuristic will occasionally fail, as in `beers` where the shortest unknown analysis is `** +er +s`. But the correct result will be obtained in most cases.

The second ordering constraint is based on the ordering rules used in [1]. Some words can be segmented in many different ways (this is true even if all parts are known). For example

```
scarcity ->  scar city
scarcity ->  scarce +ity
scarcity ->  scar cite +y
```

A simple rule notation has been defined for assigning order to analyses in terms of their morphological parse tree. These rules can be summarised as

```
prefixing > suffixing >
       inflection > compounding
```

The third method used for ordering is affix frequency. The frequencies are based on suffix-as-tag (word class) frequencies in the LOB corpus of written English, given in [2]. Thus the suffix `+er` forming a noun from a verb (as in `walker`) was marked in the lexicon as being more likely than the adjectival comparative `+er`.

These constraints are applied simultaneously. Each rule has an appropriate weighting, such that the length of the unknown part is a more significant factor than morphological structure, which in turn is more significant than affix frequency.

Results

The method was subjected to a test procedure. The test used a basic lexicon of around 3500 morphemes, of which around 150 were affixes. From a randomly selected AI magazine article, the first 200 words were used which could not be analysed by the basic morphological system (i.e. without the unknown root section). When these 200 words were analysed using the method described in the previous sections, 133 words (67%) were analysed correctly, 48 words (24%) were wrong due to segmentation error, and 19 (9%) were wrong due to word class error. An analysis was deemed to be correct when the most preferred analysis had both the correct morphological structure and the correct word class.

Segmentation errors were due mainly to spurious words in sub-parts of unknown sections, e.g. `illustrate` → `ill ** ate`. Such errors will increase as the lexicon grows. To prevent this type of error, it may be necessary to place restrictions on compounding, such that those words which can form part of compounds should be marked as such (though this is a major research problem in itself). Word class errors occurred where the correct segmentation was found but an incorrect morphological structure was assigned.

The definition of error used here may be over-restrictive, as it may still be the case that erroneous segmentation and structure errors still provide analyses with the correct pronunciation. But at this time the remainder of the text-to-speech system is not advanced enough for this to be adequately tested.

Generating the Spelling of Unknown Morphemes

A method has been described for handling a word which cannot be analysed by the conventional morphological analysis process. This method may generate a number of analyses, so an ordering of the results is defined. However, in a text-to-speech system (or even an interactive spelling corrector), it may be desirable to add the unknown root to a user lexicon for future reference. In such a case, it will be necessary to reconstruct the underlying spelling of the unknown morpheme.

This can be done in a very similar way to that in which the system normally generates surface forms from lexical forms. The problem is the following: given a surface form and a set of spelling rules (not including the two special rules described above), define the set of possible lexical forms which can match to the surface form. This, of course, would over-generate lexical forms, but if the permitted lexical form is further constrained so as to match the one given from the analysis containing the `**` a more satisfactory result will be obtained.

For example, the surface form `remoned` would be analysed as `re-**+ed`. A matching is carried out character by character between the lexical and surface forms, checking each match with respect to the spelling rules (and hypothesizing nulls where appropriate). On encountering the `**` section of the lexical form, the process attempts to match all possible lexical characters with the surface form. This is of course still constrained by the spelling rules, so only a few characters will match. What is significant is that the minor orthographic changes that the spelling rules describe will be respected. Thus in this case the `**` matches `mone` (rather than simply `mon` without an `e`), as the spelling rules require there to be an `e` inserted before the `+ed` in this case.

Similarly, given the surface string `mogged`, analysed as `**+ed`, the root form `mog` is generated. However, the alternative forms `mogg` and `mogge` are also generated. This is not incorrect, as in similar cases such analyses are correct (eg. `egged` and `silhouetted` respectively). As yet, the method has no means of selecting between these possibilities.

After the generation of possible orthographic forms, the letter-to-sound rules are applied. As regards the format of these rules, what is required is something very similar to Koskenniemi two-level rules, relating graphemes to phonemes in particular contexts. A small set of grapheme to phoneme rules was written using this notation. However, there were problems in writing these rules, as the fuller set of rules from which they were taken used the concept of rule ordering, while the Koskenniemi rule interpretation interprets all rules in parallel. The result was that the rewritten rules were more difficult both to read and to write. Although it is possible (and even desirable) to use finite state transducers in the run-time system, the current Koskenniemi format may not be the best format for letter-to-sound rules. Some other notation which could compile to the same form would make it easier to extend the ruleset.

Problems

The technique described above largely depends on the existence of an appropriate lexicon and morphological analyser. The starting-point was a fairly large lexicon (over 3000 morphemes) and an analyser description, and the expectation was that only minor additions would be needed to the system. However, it seems that significantly better results will require more significant changes.

Firstly, as the description used had a rich morpho-syntax, words could be analysed in many ways giving different syntactic markings (eg. different number and person markings for verbs) which were not relevant for the rest of the system. Changes were made to reduce the number of phonetically similar (though syntactically different) analyses. The end result now states only the major category of the analysis. (Naturally, if the system were to be used within a more complex syntactic parser, the other analyses may be needed).

Secondly, the number of "stem" entries in the lexicon is significant. It must be large enough to analyse most words, though not so large that it gives too many erroneous analyses of unknown words. Also, while it has been assumed that the lexicon contains productive affixes, perhaps it should also contain certain derivational affixes which are not normally productive, such as `tele-`, `+ology`, `+phobia`, `+vorous`. These would be very useful when analysing unknown words. The implication is that there should be a special lexicon used for analysing unknown words. This lexicon would have a large number of affixes, together with constraints on compounds, that would not normally be used when analysing words.

Another problem is that unknown words are often place-names, proper names, loanwords etc. The technique described here would probably not deal adequately with such words.

So far, this technique has been described only in terms of English. When considering other languages, especially those where compounding is common (eg. Dutch and German), the method would be even more advantageous. In novel compounds, large sections of the word could still be analysed. In the above description, only one unknown part is allowed in each word. This seems to be reasonable for English, where there will rarely be compounds of the form `** +suf ** +suf`. However, in other languages (especially those with a more fully-developed system of inflection) such structures do exist. An example is the Dutch word *bejaardentehuizen* (old peoples homes), which has the structure *noun* `+en` *noun* `+en`. Thus it is possible for words to contain two (or more) non-contiguous unknown sections. The method described here could probably cope with such cases in principle, but the current implementation does not do so. Instead, it would find one unknown part from the start of the first unknown morpheme to the end of the final unknown morpheme.

Summary

A system has been described which will analyse any word and assign a pronunciation. The system first tries to analyse an input word using the standard analysis procedure. If this fails, the modified lexicon and spelling rule set are used. The output analyses are then ordered. For each unknown section, the underlying orthographic form is constructed, and letter-to-sound rules are applied. The end result is a string of phonemic forms, one form for each morpheme in the original word. These phonemic forms are then processed by morphophonological rules, followed by rules for word stress assignment and vowel reduction.

Acknowledgements

Alan Black is currently funded by an SERC studentship (number 89313458). During this project Joke van de Plassche was funded by the SED and Stichting Nijmeegs Universiteits Fonds. Briony Williams is employed on the ESPRIT "POLYGLOT" project. We should also like to acknowledge help and ideas from Gerard Kempen, Franziska Maier, Helen Pain, Graeme Ritchie and Alex Zbyslaw.

References

[1] J. Allen, M. Hunnicut, and K. Klatt. *Text-to-speech: The MITalk system.* Cambridge University Press, Cambridge, UK., 1987.

[2] S. Johansson and M. Jahr. Grammatical tagging of the LOB corpus: predicting word class from word endings. In S. Johansson, editor, *Computer corpora in English language research*, Norwegian Computing Centre for the Humanities, Bergen, 1982.

[3] K. Koskenniemi. A general computational model for word-form recognition and production. In *Proceedings of the 10th International Conference on Computational Linguistics*, pages 178–181, Stanford University, California, 1984.

[4] G. Ritchie. *Languages Generated by Two-level Morphological Rules.* Research Paper 496, Dept of AI, University of Edinburgh, 1991.

[5] G. Ritchie, S. Pulman, A. Black, and G. Russell. A computational framework for lexical description. *Computational Linguistics*, 13(3-4):290–307, 1987.

[6] G. Ritchie, G. Russell, A. Black, and S. Pulman. *Computational Morphology.* MIT Press, Cambrdige, Mass., forthcomming.