Word Boundary Identification from Phoneme Sequence Constraints in Automatic Continuous Speech Recognition

Jonathan HARRINGTON **Gordon WATSON** **Maggie COOPER**

The Centre for Speech Technology Research, University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, Scotland.

Abstract

This paper explores the extent to which phoneme sequence constraints can be used to identify word boundaries in continuous speech recognition. The input consists of phonemic transcriptions (without word boundaries indicated) of 145 utterances produced by 1 RP speaker. The constraints are derived by matching the complete set of 3 phoneme sequences that can occur across word boundaries to entries in large lexicons containing both citation and reduced form pronunciations. Phonemic assimilatory adjustments across word boundaries are also taken into account. The results show that around 37% of all word boundaries can be correctly identified from a knowledge of such phoneme sequence contraints alone, and that this figure rises to 45% when a knowledge of one- and two-phoneme words and all legal, word-initial and word-final, two-phoneme sequences are taken into account. The possibility of including such constraints in the architecture of a continuous speech recogniser is discussed.

1. Introduction

The identification of word boundaries from continuous speech by human listeners depends, in part, on an interaction between prosodic, syntactic and semantic processing. Since, however, this interaction is difficult to model in machines and since some prosodic variables, such as sentence stress patterns, are difficult to extract automatically from the acoustic waveform, the identification of word boundaries must often be accomplished by different kinds of processing in continuous speech recognisers: one possibility, discussed in Lamel & Zue (1984) and explored in this paper, depends on the incorporation of a knowledge of *phoneme sequence constraints*. Phoneme sequence constraints are based on a knowledge of phoneme sequences which do not occur word-internally: for example, since there are no words which end in /m g/[1] and since /m g l/ does not occur word-internally, a word boundary must occur after /m/ (Lamel & Zue, 1984). Harrington, Johnson & Cooper (1987) showed that word boundary CVC sequences are often excluded word-internally in monomorphemic words if the pre- and post-vocalic consonants are similar: thus, /s N V N/ (N = nasal), /C l V l/, /f V p/, /g V k/, /z V z/, /sh V sh/ are all excluded, or are at least extremely rare, word-internally in British English Received Pronunciation (RP). In the study discussed below, we extend the investigations of Lamel & Zue (1984) and Harrington et al. (1987) by developing an algorithm for the automatic identification of word boundaries from such sequences in a continuous speech recogniser.

In the Alvey Demonstrator continuous speech recogniser being developed at the Centre for Speech Technology Research (CSTR), Edinburgh University (Figure 1), the identification of word boundaries from a string of phonemes is accomplished by a chart-parsing strategy which matches the lexicon from left-to-right against a string of phonemic symbols that are themselves derived from the phonetic processing of the acoustic-waveform. In this system, only complete parsings of the phonemic units are passed to higher levels for syntactic and semantic processing. The only possible parsing, therefore, of the phonemic string /t ii ch i ng w i l/ is *teaching + will*, since there are no other paths which parse the entire string of phonemes.

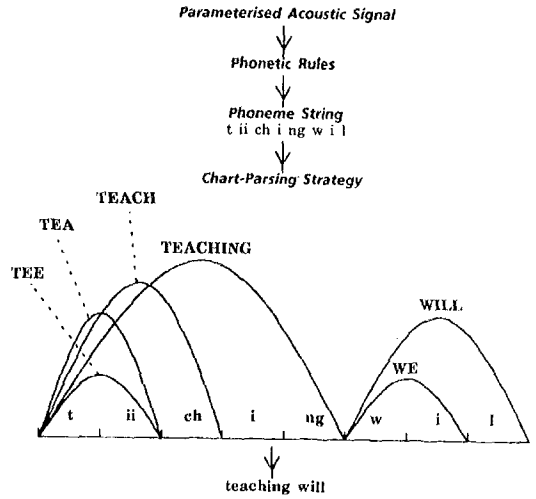

Fig. 1: A schematic outline of the components between acoustic waveform and lexical access in one of the continuous speech recognisers at Edinburgh University.

The relationship between the identification of word boundaries from phoneme sequence constraints and the chart-parsing strategy outlined above can be clarified with respect to Figure 1: at all points where the arcs do not overlap, it should be possible to insert a word boundary from a knowledge of phoneme sequence constraints. Since, therefore, the only point at which the arcs are non-overlapping is between /ng/ and /w/, phoneme sequence constraints should apply to insert a word boundary at that point (there being no monomorphemic words in the English language that contain a medial /ng w/). At the same time, however, Figure 1 would seem to suggest that the prior implementation of phoneme sequence constraints is superfluous, since all word boundaries can be found from the chart-parsing strategy. However, the application of phoneme sequence constraints may enable recovery when the chart-parsing strategy is unable to parse the phonemic string because of the incorrect derivation of a particular phoneme. Suppose, for example, that the acoustic-phonetic component incorrectly derives /oi ng/ from the parameterised acoustic waveform instead of /i ng/ (Figure 2).

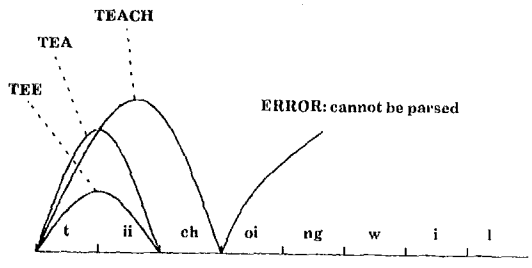

Fig. 2: The incorrect substitution of /oi/ for /i/ makes the above sequence unparsable since /ch oi ng/ occurs neither word-internally nor across word boundaries.

In this case, a left-to-right chart-parsing strategy would break off at /ch/ because /ch oi ng/ is unparsable: there are no words that end in /ch oi/ or begin with /oi ng/ and /oi/ is not usually considered to be a word (aside from an exclamation) in the English language. Since the strategy works from left-to-right, the phonemes which lie to the right of this error would also remain unparsed: thus *will* would not be derived from /w i l/, unless the chart-parsing strategy were modified in some way to be able to cope with this kind of error. If, on the other hand, phoneme sequence constraints had been applied, a word boundary would have been inserted between /ng/ and /w/. This would enable immediate recovery from the kind of error described above: in this case, if the chart-parsing strategy is unable to continue parsing phonemes at a particular point (from /ch/ to /oi/ to /ng/) it can continue parsing from the following word boundary (between /ng/ and /w/) that had been automatically inserted by phoneme sequence constraints. The prior application of phoneme sequence constraints, therefore, breaks up a single string of phonemes into smaller units, which, from the point of view of the left-to-right chart-parsing strategy, are independent of each other. A by-product of the prior insertion of word boundaries in this way is that the chart-parsing strategy could parse each of these units in parallel (Figure 3).

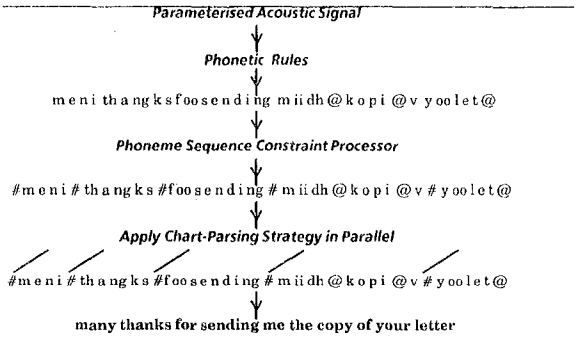

Fig. 3: The prior application of phoneme sequence constraints would enable the chart-parsing strategy to apply in parallel from all the pre-identified word boundaries.

Such a parallel strategy may be computationally faster than one which parses the string strictly from left-to-right.

As in Harrington & Johnstone (1988), sentences transcribed by a trained phonetician are used as the input data. The experiment does not take account, therefore, of any errors which may arise as a result of inaccuracies in the automatic extraction of the phonemes from the acoustic signal by the phonetic rule component of a continuous speech recogniser.

2 Method I

2.1 Word boundary sequences

In order to identify phoneme sequences which are excluded word-internally (and which therefore signal the presence of a word boundary), it is necessary to determine *a priori* the complete set of three phoneme sequences which can occur across word boundaries. For this purpose, a 'Word-lexicon' of the 23,000 most frequent words (including many derivational and inflectional morphological variants and compounds) in part of the Lancaster-Oslo-Bergen corpus (Johannson, Leech & Goodluck, 1978) was used with each word keyed to one citation form and zero or more reduced form pronunciations. The citation form entry, which is often identical to the one given in Gimson (1984), corresponds to a phonemicisation of an isolated production of the word at a moderately slow tempo. The reduced forms include variant phonemicisations of the same words which might occur in faster speech productions. In general, three different kinds of reduction rules are included: alternation rules in which segments are in free variation (e.g. /oo k sh @ n/, /o k sh @ n/, *auction*); deletion rules in which single segments may be deleted (/o k sh n/ from /o k sh @ n/, *auction*); and word-internal assimilation rules (/g u b b ai/ from /g u d b ai/, *good-bye*). The rules do not take into account phonological assimilation across word boundaries (see Harrington, Laver & Cutting (1986) for further details of the reduction rules). The reduced forms were derived from the citation forms by rule using a software package running on Xerox-1100 workstations in Interlisp-D (Cutting & Harrington, 1986). After the application of the reduction rules on the 23,000 word lexicon, around 70,000 reduced forms were derived (on average, therefore, each word is associated with 4 different pronunciations).

In order to derive the complete set of possible three phoneme sequences that occur across word boundaries, all final two phonemes (PP#) were paired with all initial phonemes (#P) of all citation and reduced forms, thus deriving the complete set of PP#P sequences (where P is any phoneme); and all final phonemes (P#) were paired with the first two phonemes (#PP) of all citation and reduced forms thus deriving the complete set of P#PP sequences. This pairing operation produced a total of 62,670 different three-phoneme sequences.

Subsequently, it was necessary to take into account some of the modifications to word boundary sequences which occur as a result of assimilatory processes since, as stated above, these were not included in the reduction rules. In order to take into account the realisation of /r/ in phrases such as /dh e@ r aa m e n i/ (*there are many*) and 'intrusive /r/' (/dh ii ai d i @ r i z/, *the idea is*), the sequences in (1) were paired with all word-initial vowel phonemes that occurred in the Word-lexicon:

(1) /u@ r, e@ r, i@ r, @ r, @@ r, oo r, aa r/

thus deriving, for example, /@ r# i/ (*measure is*), /aa r # au/ (*far out*) etc. In addition, /r/ was paired with all #VP sequences in the Word-lexicon where V is any word-initial vowel and P is any phoneme. This pairing operation results in sequences such as /r # i z/ (*measure is*), /r # au t/ (*far out*) etc.

In order to account for the assimilation of alveolars to bilabials preceding bilabials, all PP_t# sequences (where P is any phoneme and P_t is one of /t,d,n/) were extracted from the Word-lexicon. Final /t/, /d/, /n/ were then changed to /p/, /b/ and /m/ respectively (thus the PP_t# sequences /ii t #/, /ii d #/, /ii n #/ were changed to /ii p #/, /ii b #/, /ii m #/). The changed sequences were then paired with the labial consonants /p,b,m,f,v,w/. This pairing operation produces sequences such as /ii p # b/ (*eat by*), /ou m # f/ (*shown few*), /@@ m # w/ (*burn wood*).

A similar procedure was used to take account of the instability of some of the alveolars before palatals and velars as shown in Table 1 below.

/s/ to /sh/:	oo sh # sh	sh # sh uu	(*horse shoe*)
/z/ to /zh/:	i zh # sh	zh # sh u@	(*is sure*)
/t/ to /ch/:	a ch # y	ch # y oo	(*at your*)
/d/ to /jh/:	i jh # y	jh # y uu	(*did you*)
/t/ to /k/:	ai k # k	k # k uh	(*might come*)
/d/ to /g/:	ii g # k	g # k l	(*need cleaning*)
/n/ to /ng/:	e ng # k	ng # k a	(*when can*)

Table I: Some of the word boundary assimilation cases considered in the derivation of word boundary sequences.

Consideration was given to some deletion rules across word boundaries such as the deletion of the alveolar stop in /f aa s # s p ii ch/, (*fast speech*). In this case, a complete list of three-phoneme sequences occurring word-finally was made from the Word-lexicon where the penultimate consonant was a fricative and the final consonant an alveolar stop. The final alveolar stop was deleted and

the resulting two-phoneme sequence was paired with all members of #P (thus /aa s t #/ (*fast*) ⇒ /aa s #/ (*fast*) ⇒ /aa s # s/, *fast speech*). All word boundary sequences which resulted from the inclusion of these assimilation rules were added to the previously derived P#PP and PP#P sequences, thus producing a total of 69,819 word boundary sequences.

2.2 Word boundary sequences excluded word-internally.

We now wished to determine which word boundary sequences do not occur word-internally (since these enable the automatic detection of a word boundary). However, it is clear from the phonology literature (Fudge, 1969; Clements & Keyser, 1983) that sequential constraints on phonemes are not upheld across many morpheme boundaries. For example, it is well documented (Rockey, 1973) that only alveolars and palato-alveolars may follow /au/ (*town, howl, couch*). But such a constraint is not upheld word-internally across the morpheme boundary in a compound such as *cowboy*, /k au b oi/. Similarly, /uu au t/ does not occur morpheme-internally, but does occur in compounds such as *throughout*. Since the Word-lexicon includes compounds, sequences such as /uu au t/ would be considered to occur word-internally and would therefore be excluded from the list of phoneme sequence constraints that enable the automatic detection of a word boundary from a string of phonemes. But this has the unfortunate effect that a word boundary would not be inserted in the sequence *through outer*, /th r uu au t @/. Since in fact we prefer word boundaries to be inserted wherever possible, all compounds were removed from the Word-lexicon, as a result of which /uu au t/ would be included as a possible phoneme sequence constraint. Consequently, we would expect a word boundary to be inserted in both *through outer* and *throughout*. This implies either that *throughout* must be stored as /th r uu # au t/ in the lexicon which the chart-parsing strategy matches against the phonemic string, or else that morphological rules must apply after the phoneme sequence constraint processor to remove the medial # in *throughout*.

A similar argument applies to inflectional morpheme boundaries. For example, /n th s/ is excluded morpheme internally but does occur across stem/inflectional suffix boundaries (*months*). For the reasons outlined above, morphological variants with regular inflections (plurals, present and past tense suffixes) were removed from the Word-lexicon. Excluding these inflectional morphological variants has the (undesirable) effect that a boundary will be inserted between /th/ and /s/ in *three months time*, /th r ii m uh n th # s t ai m/. However, some inflectional morphological rules, which apply after the phoneme sequence constraint processor, are designed to convert these boundaries into morpheme (M) boundaries (see section 4 below).

Finally, it is also the case that many sequences that are excluded monomorphemically (e.g. /m ei sh/) can occur word-internally in derived morphological variants (/k o n f @ m ei sh @ n/, *confirmation*). A similar case could be made for removing derivational variants from the Word-lexicon and applying morphological rules to remove the # boundary from sequences such as /k o n f @ m # ei sh @ n/ which would result after the application of the phoneme sequence constraint processor. However, derivational variants were not removed, in part due to the complexity of the interaction between the inflectional and derivational morphological rules that would have to apply after word boundaries had been inserted automatically.

Only compounds and regular morphologically inflected variants were removed from the Word-lexicon; henceforth, the resulting lexicon with such entries removed will be referred to as the *Morpheme-lexicon*. The Morpheme-lexicon contained around 12,000 lexical entries after these morphological variants had been removed from the 23,000 Word-lexicon.

All word boundary sequences, including those which account for the assimilatory processes described in 2.1, were placed in one file and the medial word boundary symbol was removed. After all duplicate entries had been removed, the resulting file was matched against the Morpheme-lexicon in order to determine which boundary sequences do not occur 'morpheme'-internally. The matching algorithm for this purpose was a UNIX shell script running on a 12 mB Masscomp: it outputs the frequency with which the word boundary sequences occur word-internally in a given lexicon.

2.3 The word boundary identification algorithm

All word boundary sequences which did not occur 'morpheme'-internally were compiled into a discrimination tree in which, working from left to right, common phonemes share identical branches. At the end of each branch, an instruction is included for where the boundary should be inserted if the sequence is found in an input phonemic string (Figure 4).

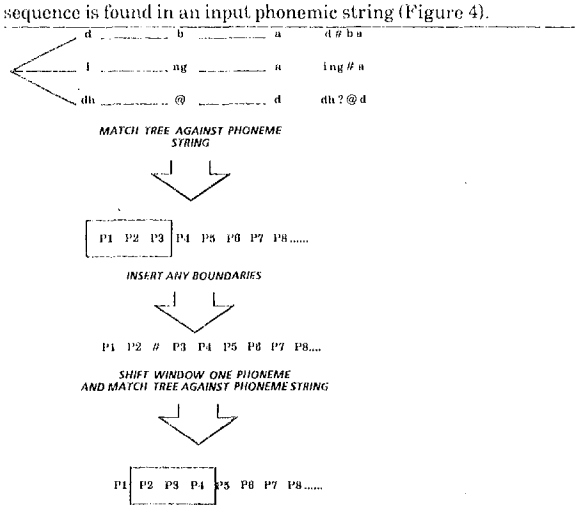

Figure 4: The process by which the tree containing the phoneme sequence constraints is matched against a phonemic input.

In the case of /d b a/, for example, the boundary must be inserted after /d/, since there are no entries in the Morpheme-lexicon with final /d b/. However, since there are entries that both end in /dh @/ and begin with /@ d/, /dh @ d/ cannot be unambiguously parsed: in this case a '?' is inserted after the first phoneme of the word boundary sequence. /dh ? @ d/ means, therefore, that a word boundary occurs either after /dh/, or after /@/.

For any given input phonemic string, the algorithm matches three phonemes at a time against the tree (Figure 4) from left-to-right through the string. If they match, a boundary is inserted at the appropriate place. Subsequently, the fixed window of three phonemes shifts one phoneme to the right and the new sequence is matched in the same way. Thus, the matching algorithm steps through the input string one phoneme at a time with a window-width of three phonemes until the end of the string is reached.

Phonemic transcriptions (excluding stress or boundary symbols) were made by a trained phonetician of 145 sentences produced by one RP speaker. The average numbers of words per utterance and phonemes per word were 10.73 and 4.04 respectively. The sentences were taken from a 'phonemically balanced' passage constructed for the speech recognition project at Edinburgh University; sentences from Section H of the Lancaster-Oslo-Bergen corpus (Johannson, Leech and Goodluck, 1978); and sentences from a corpus of business dictation collected at CSTR. The transcribed sentences, which clearly do not contain any errors that could have arisen as a result of phonetic processing of the acoustic waveform by a speech recogniser, were input to the algorithm schematically outlined in Figure 4.

3. Results I

The statistics on the automatically inserted # boundaries are shown in Table II.

Target number of word-boundaries	1411
Total number of inserted # boundaries	592
# correctly inserted	523
Remainder	69
Reduced forms not accounted for	14
Lexical items not accounted for	7
Corresponding to morpheme boundaries	44

Table II: Word boundaries automatically inserted in the 145 phonemically transcribed utterances.

The results show that 523/1411 (37%) of the target word boundaries were correctly detected. However, there were 69 automatically inserted # boundaries which did not correspond to word boundaries in the original utterances. Of these, 14 were incorrectly inserted because of the presence of reduced phonological forms in the utterances (e.g. /w @ dh/ for *with*) which we had failed to generate by rule; and 7 were inserted because some words occurred in the utterances that had not been included in the Word-lexicon (most of these were proper names). 44 # boundaries were inserted at morpheme boundaries, both in compounds (/h au # e v @/ for *however*) and preceding inflectional suffixes (/s i m # z/ for *seems*). In the next section, some morphology rules are described which attempt to convert the # at stem/suffix boundaries in cases such as /s i m # z/ into morpheme boundaries. Finally, 244 '?' were inserted at appropriate points (i.e. for each /P?QR/, where /PQR/ are phonemes, either /P#QR/ or /PQ#R/ occurred in the original utterances). The next section also describes rules for converting some of these '?' boundaries into definite # boundaries.

4. Method II

4.1 Morphology rules

The phonemic strings with the word boundaries inserted by the matching algorithm in Figure 4 are input to a second stage of processing which uses four additional sources of knowledge: PHON1 and PHON2 (a list of all one and two phoneme words in the Morphology-lexicon) and #PP and PP# (a list of all legal word-initial and word-final two phoneme sequences). Since these data are extracted from the Morphology-lexicon, they take account of phonologically reduced variants, but not the morphological variants that were excluded from the Word-lexicon.

The *morphology rules* test whether the two phonemes that occur to the right of an automatically inserted # are legal with respect to PHON1, PHON2, #PP andPP#. If they are not, the assumption is made that the # occurs across a stem/inflectional morpheme boundary. Morphological rules are then applied to shift the # to the correct place, if possible. Consider for example, the phrase *boys and girls in...* which, after the application of the first stage of processing, was analysed as:

(2) b oi # z a n ? g @@ l # z i n

The insertion of the word boundaries at this first stage of processing is attributable to the fact that neither /b oi z/ nor /g..@@..l..z/ occurred in the Morphology-lexicon. Furthermore, since there are no words that begin with /oi z/ nor /l z/, the relevant sequences would be stored as /b oi # z/ and /@@ l # z/ in the tree in Figure 4. The following test is now performed on the two phonemes to the right of the first # in (2):

(3) If /z a/ is not in #PP rewrite /oi # z a/ as /oi M z # a/
else rewrite /oi # z a/ as /oi M? z a/.

Informally, (3) states that if /z a/ cannot begin words (according to the Morphology-lexicon), /z/ must be an inflectional suffix of the previous word: therefore place an 'M' (morpheme boundary) before /z/ and shift the # symbol to the right of /z/. Alternatively, if /z a/ does begin words in the Morphology-lexicon, it is impossible to determine whether /z/ is a plural suffix or the first phoneme of a following word. In this case, M? is used to denote these two possibilities: it is an abbreviation for either /oi M z # a/ or /oi # z a/. In fact, since there are no words that begin with /z a/, (2) is analysed as /M z # a/. A solution with M? would occur if *boys are* were analysed at the first stage of processing as:

(4) b oi # z aa

since in this case /z/ can also be the first phoneme of a word (*Csar*).

A test is often performed with respect to PHON1 and/or PHON2 rather than #PP. This occurs in the following example, in which two # symbols have been automatically inserted in close proximity at the first stage of processing:

(5) b i g i n # z @ # t ai p # (*begins a type*)

In this case, a test is made to determine whether /z @/ occurs in PHON2 (i.e. whether it is is a two phoneme word). Since it is not, (5) is reanalysed as /b i g i n M z # @ # t ai p #/.

The test in (3) above is only made if the structural description of phonemes to the left and right of the # is met by certain conditions. Specifically, the test is performed in contexts such as those given in Table III.

PAST TENSE	
{p, k, f, th, s, sh}# t	(*tapped, missed, wished*)
voiced phonemes excluding /d/ #d	(*paved, seemed*)
PLURALS/PRESENT TENSE	
{p, t, k, f, th} # s	(*mats, picks, meets*)
voiced phonemes excluding /z, zh, jh/ # z	(*tabs,sings*)

Table III: Some of the contexts in which the morphology rules apply.

4.2 Resolving Ambiguities

The four sets of data PHON1, PHON2, #PP and PP# are also used to convert some '?' symbols into definite (#) word boundaries. In order to resolve the hypothetical ambiguity /ABC?DEF/, for example, it is first expanded into the two possible cases it represents in (7) and (8) below:

(7) ABC#DEF
(8) ABCD#EF

An attempt is then made to prove that either (7) or (8) is illegal (on the basis that, if (7) is illegal, ABC?DEF must correspond to the representation in (8) and *vice-versa*). (7) can be proved illegal if (9) is true:

(9) Either C is not in PHON1 and BC is not in PP#
Or D is not in PHON1 and DE is not in #PP

An informal interpretation of (9) is the following. If C is not a one-phoneme word, test whether BC is a legal two-phoneme sequence that can end words; if C is not a one-phoneme word and BC cannot end words, then (7) must be illegal. Otherwise, if (7) cannot be shown to be illegal on the basis of the phonemes that precede #, the phonemes that follow # are considered. In this case if D is not a one-phoneme word and if DE cannot begin a word, (7) must be illegal. Otherwise, (7) cannot be shown to be illegal and so the following (similar) test is applied to (8):

(10) (8) is illegal if:
Either D is not in PHON1 and CD is not in PP#
Or E is not in PHON1 and EF is not in #PP.

If neither (7) nor (8) can be proved illegal, the '?' cannot be resolved into #.

When two '?' symbols occur in close proximity, an expansion is made into four alternatives. If three of the alternatives can be proved illegal, both '?' symbols can be resolved as definite # symbols. For example, after the first stage of processing, *measuring the gun* was analysed as:

(11) /m e zh r i ng # dh ? @ ? g uh n /

This expands into the following alternatives:

(12) /m e zh r i ng # dh # @ # g uh n /.
(13) /m e zh r i ng # dh # @ g # uh n /.
(14) /m e zh r i ng # dh @ ## g uh n /.
(15) /m e zh r i ng # dh @ # g # uh n /.

(12) and (13) must be illegal since /dh/ is not a one-phoneme word ((13) is additionally illegal since /@ g/ is not a possible two-phoneme word). (15) is illegal since /g/ is not a one-phoneme word. Therefore (14) is the only possible analysis of (11).

This type of expansion into four possibilities is only made when 3 phonemes, or fewer, occur between the two '?' symbols: if more than three phonemes intervene, the result of resolving both ? symbols together is the same as if each ? symbol were considered separately.

Finally, the example with two '?' symbols in (11) is extended to the general case in which *n* '?' symbols occur in close proximity to one another (i.e. a series of *n* '?' symbols with 3, or fewer, phonemes between successive '?' symbols). These expand into 2^n alternatives. As in the example above, if $2^n - 1$ alternatives can be proved illegal, all *n* '?' symbols can be converted to # symbols.

4.3 Order of rules

After the application of the first stage of the word boundary insertion rules, *expansion rules* apply in which each '?' symbol is expanded into two alternatives. The *morphology rules* apply to each of these expanded alternatives and at all other points in the utterance at which their structural description is met. Only after the morphology rules have applied can any of the alternates be eliminated. The morphology rules must apply before eliminating alternatives, otherwise some alternatives might be incorrectly eliminated. This can be illustrated with the example *boys and girls* which, after the first stage of processing, was analysed as /b oi # z a n ? g @@ l # z/. This expands into:

(16) b oi # z a n # g @@ l # z
(17) b oi # z a n g # @@ l # z

If the elimination rules applied prior to morphological rules, both (16) and (17) would be eliminated, since /z a/ is not in #PP (and (17) is illegal since /n g/ is not in PP#). If, on the other hand, the morphology rules apply first, (18) and (19) would be derived from (16) and (17) respectively:

(18) b oi M z # a n # g @@ l M z #
(19) b oi M z # a n g # @@ l M z #

Only (19) would be eliminated, on the grounds that /n g/ is not a legal two-phoneme sequence occurring word-finally.

A further illustration of the interaction between the expansion rules, morphological rules and elimination of alternatives is shown in (20 - 33) below. After the first stage of processing, *months tie* (from a sentence in a gardening manual, 'after a few months, tie in more growth') was analysed as /m uh n th ? s t ? ai/. This expands to four alternatives:

(20) m uh n th # s t # ai i n
(21) m uh n th # s t ai # i n
(22) m uh n th s # t # ai i n
(23) m uh n th s # t ai # i n

Morphology rules are applied to the four alternatives:

(24) m uh n th M s # t # ai i n
(25) m uh n th M? s t ai # i n
(26) m uh n th s M t # ai i n
(27) m uh n th s M? t ai # i n

(25) and (27) are further expanded into the two alternatives they represent. This given a total of 6 alternatives:

(28) m uh n th M s # t # ai i n (from (24))
(29) m uh n th M s # t ai # i n (from (25))
(30) m uh n th # s t ai # i n (from (25))
(31) m uh n th s M t # ai i n (from (26))
(32) m uh n th s M t # ai # i n (from (27))
(33) m uh n th s # t ai # i n (from (27))

In eliminating the alternatives, a slight modification has to be made to the rules: rather than referring to two segments to the left and right of #, they refer to the two segments to the left of an M symbol (if present) and to two segments to the right of #. But the segments that intervene between an M and # are ignored. The following test would therefore be made to test the legality of (29):

(34) (29) is illegal if:
Either /th/ is not in PHON1 and /n th/ is not in PP#
Or /t ai/ is not in PHON2

It is possible to eliminate (28) since /t/ is not in PHON1. (31), (32) and (33) can be eliminated since /th s/ does not occur in PP# (final /th s/ occurring only across a stem/inflectional suffix boundary). (29) and (30) remain, and are collapsed into one representation in (35) using the M? notation:

(35) m uh n th M? s t ai # i n

The analysis shows therefore that /m uh n th ? s t ? ai/ corresponds to either *months tie in* or *month sty in*.

5. Results II

The statistics on the automatically inserted # boundaries are shown in Table IV.

Target number of word-boundaries	**1411**
Total number of inserted # boundaries	**690**
# correctly inserted	**645**
Remainder	**45**
Reduced forms not accounted for	14
Lexical items not accounted for	10
Corresponding to morpheme boundaries	21

Table IV: Word boundaries automatically inserted after the application of the morphology, expansion and elimination rules.

The results show that 645/1411 (45.7%) of the target word boundaries were correctly detected. This is an increase of around 9% compared with the result obtained prior to the application of the rules described in the preceding section. 24 # boundaries were inserted at inappropriate points, either because of the presence of

because of the presence of reduced forms in the utterances that we had not derived by rule, or because of lexical items that had not been included in the word-lexicon. All 21 inserted # symbols that corresponded to morpheme boundaries were inserted medially in compounds (e.g. *how#ever*, *there#fore*), while all automatically inserted # symbols that had occurred at stem/inflectional suffix boundaries (/s..i..m..#..z/ for *seems*) were converted to M or M? symbols using the morphology rules described above.

An approximate measure of the probability of a word boundary being incorrectly inserted can be made as follows. Firstly, since it was our intention that the algorithm should insert # symbols not only between words but also within compounds, the target number of boundaries to be identified can be considered to be 1411 (the number of word boundaries in the utterances) plus 78 (the number of boundaries occurring within compounds), i.e. 1489. Of these (see Table IV), 645 + 21 = 666 (44.7%) were correctly inserted. The probability of a word boundary being incorrectly inserted, either as a result of a reduced form which was not derived by rule, or because of the omission of a word from the Word-lexicon, is given by:

(36) $(24/(666 + 24) \times 100)\ \% \quad = \quad 3.5\%.$

6. Discussion

This study has shown that around 45% of all word boundaries can be correctly identified from a knowledge of three-phoneme sequences that occur across word boundaries but which do not occur word-internally together with a knowledge of one- and two-phoneme words and all two-phoneme sequences that can begin and end words. The result is based on hand-transcriptions which can be considered analogous to the phonemic strings that would be extracted automatically from the acoustic speech signal if the recogniser made no errors in this derivation.

A current area of investigation is to identify the set of phoneme sequences which occur neither across a word boundary nor word-internally. Such phoneme sequences can be easily obtained from the data sets discussed in this paper and they would enable errors to be detected in the acoustic-phonetic stage of processing in a continuous speech recogniser. Some examples of these sequences are given in (37):

(37) /l z ng/, /aa dh l/, /e w n/

For example, /e w n/ must be illegal since it does not occur word-internally and because it does not occur across word boundaries (both /e # w n/ and /e w # n/ must be ruled out on the grounds that there are no words which end in /e/ or /e w/). The incorporation of this kind of knowledge would enable an error to be detected if such a sequence were derived automatically after the acoustic-phonetic stage of processing.

7. References

Clements G.N. & Keyser S.J. (1983) *CV Phonology. A Generative Theory of the Syllable.* MIT Press: Cambridge Mass.

Cutting D. & Harrington J.M. (1986) Phongram: a phonological rule interpreter. In (Lawrence R. ed.) *Proceedings of the Institute of Acoustics*, 8, 461-469. Institute of Acoustic: Edinburgh.

Fudge E.C. (1969) Syllables. *Journal of Linguistics* 5, 253-286.

Gimson A.C. (1984) *English Pronouncing Dictionary* (Revised edition, originally compiled by D. Jones). Dent: London.

Harrington J.M. & Johnstone A. (1988, in press) The effects of equivalence classes on parsing phonemes into words in continuous speech recognition. *Computer Speech & Language.*

Harrington J.M., Johnson I. & Cooper M. (1987) The application of phoneme sequence constraints to word boundary identification in automatic, continuous speech recognition. In (Laver J. & Jack M. eds.) *European Conference on Speech Technology*, Vol. 1, 163-166.

Harrington J.M., Laver J. & Cutting D. (1986) Word-structure reduction rules in automatic, continuous speech recognition. In *Proceedings of the Institute of Acoustics* (R. Lawrence ed.) 8, 451-460. Institute of Acoustics: Edinburgh.

Johannson S., Leech G.N. & Goodluck H. (1978) *The Lancaster-Oslo/Bergen Corpus of British English.* Department of English, Oslo University.

Lamel L. & Zue V.W. (1984) Properties of consonant sequences within words and across word boundaries. *Proceeedings ICASSP* 42.3.1 - 42.3.4.

Rockey D. (1973) *Phonetic Lexicon.* Heyden: Oxford.

8. Notes

1 The CSTR Machine Readable Phonemic Alphabet for RP used in this paper is shown below:

/p/	*pea*	/f/	*fan*	/l/	*lee*
/b/	*bead*	/v/	*van*	/r/	*road*
/t/	*tea*	/th/	*think*	/w/	*win*
/d/	*day*	/dh/	*then*	/y/	*you*
/k/	*key*	/s/	*sing*	/m/	*man*
/g/	*guy*	/z/	*zoo*	/n/	*name*
/ch/	*chew*	/sh/	*shoe*	/ng/	*sing*
/jh/	*judge*	/zh/	*measure*	/h/	*hat*
/ii/	*we*	/o/	*hot*	/ei/	*stay*
/i/	*hit*	/oo/	*saw*	/ai/	*sigh*
/e/	*head*	/u/	*could*	/oi/	*toy*
/a/	*had*	/uu/	*who*	/au/	*now*
/aa/	*hard*	/@/	*the*	/ou/	*go*
/i@/	*here*	/u@/	*sure*	/e@/	*there*
/@@/	*first*				

This research was supported by SERC grant number GR/D29628 and is part of an Alvey funded project in continuous speech recognition. Our thanks to John Laver and Briony Williams for many helpful comments in the preparation of this manuscript.