Computer Speech and Language (1991) **5**, 379–392

Word boundary hypothesization in Hindi speech

G. V. Ramana Rao and B. Yegnanarayana

Department of Computer Science and Engineering, Indian Institute of Technology, Madras 600 036, India

Abstract

This paper proposes a method for hypothesizing word boundaries in Hindi speech. The method is based on the observation that function words such as case markers, pronouns and conjunctions occur frequently in Hindi text and spotting of these frequently occurring patterns is proposed as a means for hypothesizing word boundaries in a speech-to-text conversion system for Hindi. Initially, the idea was tested on a correct text with all word boundaries (except sentence boundaries) removed; the results showed that nearly 67% of the word boundaries were correctly hypothesized. Later, experiments with input containing errors simulated to represent speech environment showed that the proposed method is effective even at error levels as high as 50%. The implications of these results in the development of a speech-to-text conversion system for Hindi are discussed.

1. Introduction

Continuous speech does not offer many clues for hypothesizing the word boundaries. In order to develop a speech-to-text conversion system for any language, one has to use as many clues of speech and language as possible to determine the locations of word boundaries; a text without word boundaries is difficult to read. It is also difficult to determine word boundaries using a lexical search of the string of characters with the help of a dictionary. First, the complexity of such a search is nearly exponential; second, dictionary matching becomes more complex if some of the characters in the input string are ambiguous. It is interesting to note that even with some word boundaries, the readability of the transformed text improves significantly. One could identify and use clues based on speech and language characteristics. Prosodic features such as pause, duration and pitch can be used as speech clues to hypothesize some word boundaries, while language features such as phoneme sequence constraints and syntactic markers can also be exploited for placing the word boundaries. This paper focuses on the use of language clues for word boundary hypothesization.

Any speech recognition system involves several stages of processing. Initially, the input analog signal is digitized and stored in the computer. The digitized data is then converted into a sequence of symbols representing the input; this is called acoustic–phonetic analysis. From the sequence of symbols, a sequence of words is generated using a dictionary, which is known as lexical analysis. On the word sequence, higher level

0885–2308/91/040379+14 $03.00/0

knowledge sources such as language and task constraints are applied to generate the text output. In this paper, we present our approach for developing a speech recognition system, and in particular, we focus on the subproblem of word boundary hypothesization.

The problem of word boundary hypothesization can be stated as follows: given a sequence of symbols representing an utterance, how to place the boundaries so as to produce a sequence of words. This problem is important in the context of continuous speech recognition systems. In most of the speech recognition systems, the word boundaries are hypothesized as part of the word hypothesization process in the lexical analyser (Klatt, 1980). However, *a priori* hypothesization of word boundaries offers several advantages which are explained below.

In a speech recognition system, the input speech is converted into a sequence of symbols which are then matched against a prestored lexicon to hypothesize words; even when the symbols are correctly given, several word sequences may match the input. In addition, due to the vagaries of speech, the symbol corresponding to a segment of speech may not be correctly identified, which means that the output of the signal-to-symbol conversion stage will only be an approximate representation of the utterance. Hence, approximate string matching (ASM) techniques have to be used for matching the input with the lexicon, which further increases the complexity of the lexical matching. Studies of English (Harrington & Johnstone, 1987) showed that for some utterances, the number of possible word strings matching at broad class level, may exceed 10 million. By hypothesizing some word boundaries *a priori*, one would divide an input sentence into several smaller subsentences and since the lexical match is now limited to these subsentences, the match complexity is reduced to manageable proportions. Moreover, the matching process can now be performed in parallel over the various subsentences which speeds it up further.

Word boundary hypothesization simplifies the development of a speech-to-text conversion system. In a speech-to-text conversion system, the aim is to produce a transcription of the input speech which may be used later by a human. As the end-user is a human, the main problem in the development of such systems is to produce a symbolic representation of speech with word boundaries. Even if a few errors are present in the transcription, they can be corrected by the user using his knowledge of the syntax and semantics of the language and the knowledge specific to the task. Hence, it is the acoustic–phonetic and the lexical analysers that are important. The major task here is the development of a high-performance acoustic–phonetic analyser, referred to as the "phonetic engine" in literature (Mangione, 1986), which produces a phonetic transcription of the utterance. From this phonetic transcription, one can obtain a character sequence representing the utterance by using a pronunciation dictionary. If one can develop a method to place the word boundaries in this character sequence, a speech-to-text conversion system which does not use the higher level knowledge such as syntax and semantics can be developed.

Another advantage with *a priori* word boundary hypothesization is in handling unknown words. Usually, every text contains some proper nouns such as names of persons and places which would not be prestored in the dictionary. In lexical analysers performing word hypothesization only, such words cause the lexical analyser to return a "no match" condition. To recover, the lexical analyser has to skip each character of the unknown word and try matching with the lexicon until the next known word is reached. However, if some word boundaries were already hypothesized, one could then use the

simple recovery strategy of skipping the input until the next hypothesized word boundary. This strategy would work well if many word boundaries could be hypothesized.

In the following, we summarize the advantages offered by word boundary hypothesization:

1. The complexity of the lexical matching in large vocabulary speech recognition could be reduced.
2. If most of the word boundaries could be hypothesized, an inexpensive speech-to-text conversion system could be developed, using a phonetic engine, a symbol-to-character converter and a word boundary hypothesizer.
3. Unknown words can be handled easily.

We are attempting to develop a speech-to-text conversion system for the Indian language Hindi. The system is similar in function to a dictation machine; the system consists of three blocks organized in a hierarchical fashion as shown in Fig. 1. The first block is the acoustic–phonetic analyser, which produces a character transcription of the input speech. The second block is the lexical analyser, which produces a word sequence from the output of the acoustic–phonetic analyser. The third block is the syntacto-semantic analyser, which corrects the word sequence and produces a text output. Details of our speech-to-text conversion system are given in Yegnanarayana *et al.* (1989).

In the design of the speech-to-text conversion system, we tried to exploit the features of Indian languages. The most important feature is the phonetic nature of the languages. In many Indian languages, including Hindi, there is a close correspondence between phonemes and graphemes; hence, we use the characters of Hindi as the symbols, which means that the output of the acoustic–phonetic analyser will be a string of Hindi characters, thus eliminating the need for a pronunciation dictionary, and simplifying the lexical analysis which is normally not possible for English since English letters do not have a unique pronunciation. Moreover, Eswar (1990) showed that the errors in signal-to-symbol conversion caused by vagaries of pronunciation are less for Indian languages.

The word boundary hypothesizer is the first block in the lexical analyser module; it accepts the character sequence produced by the acoustic–phonetic analyser and hypothesizes word boundaries in it. The first step in the development is to identify the clues useful for word boundary hypothesization; both prosodic and language clues can be exploited in this regard. Earlier studies for English identified some clues for word boundary hypothesization. Lea (1980) discusses the application of prosodic features such as pause, duration and intonation for word boundary hypothesization. While these studies established the usefulness of prosodic clues in hypothesizing word boundaries, they are only applicable for English. Similar studies are yet to be carried out for Indian

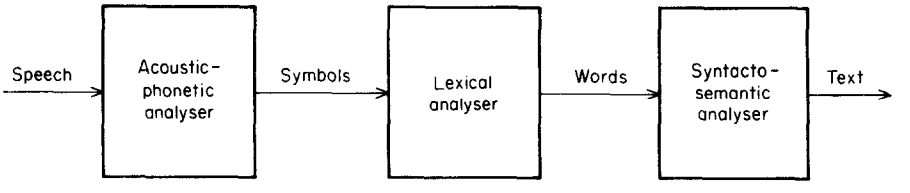

Figure 1. Block diagram of speech-to-text conversion system.

languages. More recently, Harrington, Watson and Cooper (1989) examined the use of phoneme sequence constraints to hypothesize word boundaries in English, but in the presence of ambiguities in the phonemes, they found that these clues are of limited use. However, language as a source of knowledge for word boundary hypothesization is unexplored. Our work reported here concentrates on identifying the language clues useful for word boundary hypothesization and applying them in the context of Hindi speech recognition.

2. Clues for word boundary hypothesization

In this section we describe the clues proposed by us for identifying word boundaries. Earlier work by Bhatia (1970) and Ohala (1983) identified some conditions to be satisfied at the beginning and ending of words. As these conditions are based on the properties of the lexicon, we named them "lexical clues" (see Fig. 2), but these lexical clues are not suited to hypothesize word boundaries; hence, we proposed clues based on the word frequency analysis of the Hindi language. The idea is to spot the patterns corresponding to the most frequently occurring words. If these patterns occur more frequently as words than as substrings of other words, then one can hypothesize word boundaries around the spotted patterns. We henceforth refer to these non-lexical clues as "pattern clues". The pattern clues also include some other frequently occurring patterns (not necessarily words). *Word boundary hypothesization using these pattern clues is equivalent to the spotting of frequently occurring patterns in the input speech.*

Two factors were primarily considered in selecting the pattern clues: (1) they should occur frequently; and (2) they should be quite general, i.e. they should occur in all types of text. Under these criteria, case markers and certain other word classes like pronouns and conjunctions qualify as pattern clues. In addition, certain verb endings are also included in the pattern clues; these patterns are small in number and they also serve important syntactic functions. The case markers can be treated as markers of noun phrases; they occur roughly in proportion to the noun phrases and, hence, are quite frequent. Similarly, verb endings and conjunctions serve as syntactic markers; they also occur frequently and in all types of texts. Initially, these patterns were chosen based on our knowledge of the language; some of the pattern clues are shown in Fig. 2.

A preliminary study was conducted to establish the usefulness of these clues in word boundary hypothesization. For the purpose of this study, only the most frequently occurring patterns such as case markers, conjunctions, some pronouns and a few verb endings are used. In total, four lexical clues and 25 pattern clues are used (see Fig. 2).

Two measures, frequency and correctness, were used to evaluate the clues. The frequency measure is used to indicate how useful a clue is in terms of the number of word boundary hypotheses it produces; it is defined as the ratio of the number of times a word boundary is correctly hypothesized using the clue to the number of word boundaries actually present in the input. The correctness measure indicates the confidence in the hypotheses generated using the clue; it is defined as the ratio of the number of times a word boundary is correctly hypothesized using the clue to the number of hypotheses generated using the clue.

The clues were evaluated using a 400-sentence text containing nearly 5000 word boundaries, collected from several sources to study the applicability of the clues in different contexts. Some of the sources were a children's story book, a graduate level text, a collection of short stories and a popular magazine. All the word boundaries

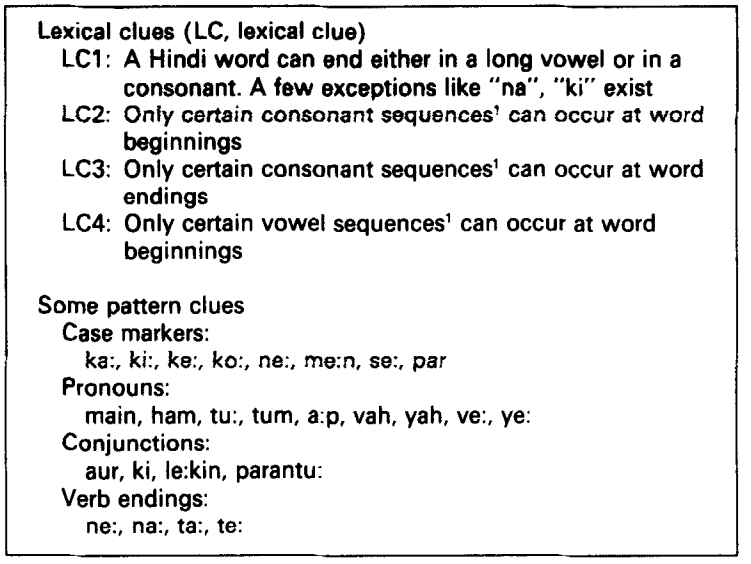

Figure 2. Lexical clues and some pattern clues. LC, lexical clue.
[1] For details refer to Bhatia (1970).

(except sentence boundaries) were removed from this text, then word boundaries were hypothesized using each clue. From these hypotheses, the frequency and correctness measures were calculated for each of the clues (Table I). The results for the pattern clues are shown in four groups, namely case markers, verb endings, pronouns and conjunctions. More details are given in Ramana Rao, Prakash & Yegnanarayana (1989).

Table I(a) shows the results for lexical clues. As the lexical clues specify only the conditions to be satisfied at word boundaries, they are not useful for hypothesizing word boundaries, and the frequency and correctness measures are not applicable for these clues. Hence, the correctness for lexical clues is redefined as the ratio of the number of word boundaries at which the clue is satisfied to the number of word boundaries at which the clue is applicable. Thus, the correctness for lexical clues indicates the applicability of the clue for verifying word boundaries. Table I(a) shows high values of correctness for the four lexical clues indicating their usefulness in verifying word boundary hypotheses; for example, the clue LC1 states that a Hindi word can end either in a long vowel or in a consonant. While this clue in itself is not useful to hypothesize any word boundaries, it can be used to verify the word boundary hypotheses generated by other clues. Pattern clues, especially case markers, pronouns and conjunctions, have reasonably good frequency and correctness values [see Table I(b)], indicating that a significant number of the word boundaries could be correctly located by them.

Based on the above, the word boundary hypothesizer is organized as follows: the pattern clues are used to hypothesize word boundaries, which, in practice, is a simple pattern matching. The patterns corresponding to the pattern clues are matched against the input sentence. On spotting a pattern in the input text that corresponds to a pattern clue, word boundaries are hypothesized around the pattern. At these hypothesized boundaries, the lexical clues are applied as verification rules. The corresponding results for word boundary hypothesization are shown in Table II.

Table I. The frequency and correctness values for the lexical and pattern clues obtained in the preliminary study. (a) Correctness for Lexical clues. LC, lexical clue. (b) Frequency and correctness for Pattern clues

(a)

Clue	Correctness
LC1	0·95
LC2	0·92
LC3	0·90
LC4	0·96

(b)

Clue	Frequency	Correctness
Case markers	0·26	0·88
Verb endings	0·04	0·83
Pronouns and conjunctions	0·14	0·82

TABLE II. Results of word boundary hypothesization obtained in the preliminary study

Type of clue (No. of clues)	Word boundaries located (per cent of total boundaries)	Correct boundary hypotheses (per cent of total hypotheses)
Case markers (8)	25	89
Verb endings (4)	3·6	90
Pronouns and conjunctions (13)	13·4	88
All clues together (25)	40	88·5

The first column in the table indicates the type of the clue and the number of clues used. The second column indicates the number of word boundaries found in the input text using that particular clue. For example, the table indicates that using casemarkers alone, 25% of the total word boundaries present in the input were located. The third column indicates the percentage of hypotheses that were correct for that clue, which is a measure of the confidence one can attach to the clue. For example, from the table one could say that out of every 100 word boundary hypotheses produced using casemarkers, 89 are likely to correspond to the actual boundaries.

3. Word boundary hypothesization through pattern spotting

The results of the preliminary study have established that: (1) pattern clues are useful in generating word boundary hypotheses; and (2) lexical clues can be used for verifying the

hypotheses generated by the pattern clues. Due to the small number of the pattern clues used, the word boundaries located are limited. To increase the number of word boundary hypotheses, it is necessary to add more pattern clues representing several other pronouns and some commonly used adjectives, adverbs and auxiliary verbs.

One problem, especially with the pronouns, is that many pronouns have morphological variants similar to the pronoun itself. For example, the pronoun "un" has morphological variants "unhe:" and "unho:n". If one uses only "un", then every occurrence of "unhe:" in the input text results in the hypothesization of an erroneous word boundary between "un" and "he:". To eliminate such errors, all the morphological variants of the patterns must also be included in the pattern clues. Hence, in our word boundary hypothesizer, all the morphological variants of the pronouns were also included in the pattern clues, resulting in a large number of pattern clues numbering around 120.

Another problem noticed with some of the pattern clues is that they also occur as substrings of other pattern clues. For example, the case marker "ne:" occurs as the suffix of many verbs in their verbal noun form. If one hypothesizes word boundaries on both sides of "ne:", several errors occur corresponding to the cases where "ne:" is part of a verb such as "karne:". In our word boundary hypothesizer, these are taken care of by hypothesizing only the boundary occurring after the pattern. Problems with patterns which are prefixes or substrings of other patterns are also accounted for in a similar fashion.

3.1. Word boundary hypothesization in correct input

A word boundary hypothesizer was developed using the new patterns. The input is a Hindi text containing nearly 12 000 words, the sentences collected from several different sources. All the word boundaries were removed from this text (but sentence boundaries were preserved). The results of word boundary hypothesization using the above patterns are shown in Table III. Note that the results are shown in groups only for clarity.

The results indicate that a significant number of the word boundaries, nearly 67%, were hypothesized correctly. Also, the confidence in the hypotheses generated was high, as indicated by the high correctness value ($>80\%$). Results corresponding to the case where the lexical clues are used to verify the hypotheses generated by the pattern clues were also shown. Note that the percentage of the word boundaries located is lower for the case when lexical clues were used for verification. This is because, in some cases, the pattern clues occurred as prefixes or suffixes of other words such as "ka:" in the word "ad^{h}ya:pika:". On applying the lexical clues for verification, both the boundaries hypothesized around the pattern clue "ka:" will be removed, though one of them is correct. However, the confidence in the hypothesized boundaries is higher when lexical clues are used for verification.

3.2. Word boundary hypothesization in erroneous input

Once the usefulness of the pattern clues in word boundary hypothesization is established for correct text input, the next step is to study its applicability for input containing errors. In a typical speech recognition system, the output of the acoustic–phonetic analyser (which is the input to the word boundary hypothesizer) will be a sequence of symbols approximately representing the input speech. As the acoustic–phonetic analyser

TABLE III. Results of word boundary hypothesization for correct text input. (a) Results of word boundary hypothesization without using lexical clues for verification. (b) Results of word boundary hypothesization after using lexical clues for verification

(a)

Type of clue (No. of clues)	Word boundaries located (per cent of total boundaries)	Correct boundary hypotheses (per cent of total hypotheses)
Case markers (8)	38·2	86·2
Verb endings (14)	4·6	80·8
Conjunctions (8)	11·4	80·0
Pronouns (77)	15·4	74·8
Adjectives, adverbs and aux. verbs (17)	9·0	80·4
All clues together (124)	67·1	83·0

(b)

Type of clue (No. of clues)	Word boundaries located (per cent of total boundaries)	Correct boundary hypotheses (per cent of total hypotheses)
Case markers (8)	35·2	92·2
Verb endings (14)	4·5	89·4
Conjunctions (8)	10·4	89·0
Pronouns (77)	14·6	80·2
Adjectives, adverbs and aux. verbs (17)	8·2	84·5
All clues together (124)	62·2	89·0

module of our speech-to-text conversion system is not complete, we have simulated the input data for the word boundary hypothesizer.

The simulated input data was obtained by introducing errors that are likely to occur in a speech recognition system. These errors are of three types: (1) substitution errors which are due to the acoustic–phonetic analyser hypothesizing a different phoneme in place of the uttered one; (2) deletion errors which are due to the acoustic–phonetic analyser missing out some phonemes; and (3) insertion errors which are due to the acoustic–phonetic analyser hypothesizing more than one phoneme for a single phoneme.

Substitution errors are caused by similarities between the phonemes which cause a speech recognition system to confuse between them. To simulate these, we created a similarity matrix giving various alternatives for each phoneme and the probability that the given phoneme will be confused with that alternative. These values were obtained after studying a large number of utterances with the help of a linguist. The similarity matrix is shown in Fig. 3. The exact numerical values for probabilities are not given, but the similarity between the sounds is specified using three values: "high" (H), "medium"

a	a: (H), o: (M), e: (M)
a:	a (H), o: (M), e: (M)
i	i: (H), e: (M), u (L), u: (L)
i:	i (H), e: (M), u: (L), u (L)
u	u: (H), o: (M), i (L), i: (L)
u:	u (H), o: (H), i (L), i: (L)
e:	a (M), a: (M), i (M), i: (M), o: (L)
ai	e: (H), a (M), i (M), o: (L)
o:	u (H), u: (H), e: (L)
au	o: (H), a (M), u (M)
k	t (H), p (M), ṭ (M), k^h (M), t^h (L), $ṭ^h$ (L)
k^h	t^h (H), c (H), t (M), k (M), $ṭ^h$ (L), c^h (L)
g	d (H), b (H), ḍ (H), g^h (M), d^h (L)
g^h	$ḍ^h$ (H), d^h (H), b^h (M), g (M)
c	c^h (H), k^h (H), t^h (H), k (M), t (L), p (L)
c^h	c (H), $ṭ^h$ (M), t^h (M), k^h (L)
j	j^h (H), g (M), ḍ (M), d (M), b (L)
j^h	j (H), d^h (M), g (L)
ṭ	p (H), t (H), k (H), $ṭ^h$ (M), t^h (M)
$ṭ^h$	t^h (H), c^h (H), ṭ (H), p^h (M), k^h (M)
ḍ	g (H), d (H), b (H), $ḍ^h$ (M), j (M), d^h (L)
$ḍ^h$	ḍ (H), d^h (H), b^h (H), g^h (M)
N	n (H), m (M)
t	k (H), p (H), ṭ (M), t^h (M)
t^h	k^h (H), $ṭ^h$ (H), t (M), p^h (M), p (L), ṭ (L)
d	g (H), ḍ (H), b (H), d^h (M)
d^h	$ḍ^h$ (H), g^h (H), b^h (M), d (M)
n	N (H), m (M), d (L)
p	t (H), k (H), ṭ (M), p^h (M), t^h (L)
p^h	p (H), $ṭ^h$ (M), t^h (M)
b	d (H), g (H), ḍ (H), b^h (M), p (L)
b^h	d^h (H), $ḍ^h$ (H), g^h (M), b (M)
m	n (H), N (M), b (L)
y	v (H), l (M), r (L)
r	l (H), y (M), v (M)
l	r (H), y (M), v (M)
v	y (H), l (M), r (L)
ś	ṣ (H), s (M), c (L)
s	ṣ (H), ś (M), c (L)
ṣ	ś (H), s (M)
h	ś (H), s (H)

Figure 3. Similarity matrix for phonemes shown in a list form.

(M) and "low" (L). These values specify only the relative occurrences of various alternatives for a given phoneme. For example, for the phoneme "i", the alternatives are: "i:" with a "high" similarity, "e:" with a "medium" similarity and "u" and "u:" with "low" similarities. It means that if there are some substitution errors for "i", most of them will be substitutions by "i:", a few by "e:" and very rarely by "u" and "u:". The equivalent numerical probability values for these similarities vary depending on the phoneme and the number of alternatives. To simplify the implementation, the number of alternatives for a phoneme was limited to six.

The similarity matrix is used to produce an erroneous text from a correct input text for a specified average error. This average error represents the probability of substitution for any given phoneme, but, in real speech, this probability of substitution is not the same for all phonemes. Some phonemes are more prone to substitution errors than others. To take care of this, the following general rule was adopted: "the consonants, in particular stop consonants, are more prone to errors than vowels". Hence, for a specified average error value, the average error for vowels was kept lower (nearly 30% less) than the average error for the consonant sounds.

The above implementation took care of the substitution errors, but insertion and deletion errors are also possible. We have also simulated some of the common deletion and insertion errors, observed in the development of our acoustic–phonetic analyser—some of these rules are listed in Fig. 4.

Several erroneous texts representing different average error values were generated in our simulation. The results of word boundary hypothesization with these inputs are shown in Fig. 5. From Fig. 5(a), it can be observed that there is a gradual fall in the percentage of correct boundaries spotted as the errors in the input text are increased. This is expected since some of the patterns in the input text might have been corrupted and hence were not spotted. One can explain the fall roughly as follows: if the average error probability for a phoneme is p, and the length of a pattern is L, then the probability that a patttern is uncorrupted in the input is given by $(1-p)^L$. Assuming that a given pattern clue has a frequency value of f for text with no errors, its frequency for an error probability of p is $f(1-p)^L$. Hence, the frequency for all patterns is given by $\Sigma f(1-p)^L$. This expected frequency is shown in Fig. 5(b). The plot shows a good agreement with the experimental values for low error values. But, for higher error values it falls off much faster than the experimental result. This result could be due to the fact that many of the pattern clues are similar and hence one corrupted pattern might be located as another pattern clue. In such a case, word boundaries would still be hypothesized around the corrupted pattern clue. For example, the pattern "ka:" might become "ki:" due to

Phoneme deletion rules:

1. A long stop consonant may be replaced by a short one.
   Ex: kk→k
2. Any consonant sequence may be misrecognized as the last consonant in the sequence.
   Ex: kt→t
3. The trill "r" may not be recognized due to its short duration.
4. The semi vowels may not be recognized when they precede any vowel and the vowel may be replaced by its longer version.
   Ex: ya→a:

Phoneme insertion rules:

1. The diphthongs "ai" and "au" may be misrecognized as the vowel sequences "a" followed by "i" and "a" followed by "u", respectively.

Figure 4. Phoneme deletion and insertion rules.

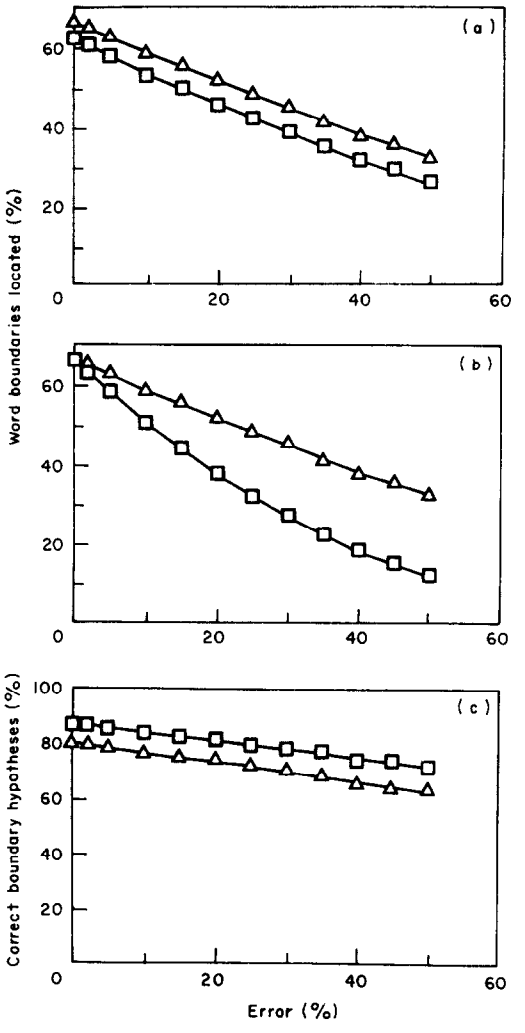

Figure 5. Results of word boundary hypothesization. (a) The plots of frequency vs. error for the cases when word boundaries are verified (—■—) and not verified (—▲—). (b) The plot of predicted (—■—) and observed (—▲—) frequencies. (c) The plot of correctness vs. error for the cases when word boundaries are verified (—■—) and not verified (—▲—).

errors, and since "ki:" is also a pattern clue, word boundaries are still hypothesized around it.

The plot of the percentage of correct word boundary hypotheses is shown in Fig. 5(c) which shows a slow rate of fall indicating that, even for high error values, one can place confidence in the pattern clues. This result could be explained as follows: an erroneous word boundary hypothesis is generated when some other input pattern is misrecognized as the pattern of a clue. There are two ways in which this happens: (1) when an input pattern which is part of another word (or words) is recognized as the pattern corresponding to a pattern clue; and (2) when some other input pattern is transformed

into the pattern corresponding to a pattern clue due to the errors in the input. Case (1) represents the erroneous hypotheses generated on error-free input and does not vary with errors in the input. However, case (2) represents the erroneous hypotheses due to errors in the input and it accounts for the drop in the correctness as errors are increased. However, only a few patterns are similar to the pattern clues and their frequencies of occurrence are low. Hence, the decrease in correctness due to case (2) above will be quite small and the correctness remains practically constant. This means that if the cost of an erroneous hypothesis is not very high, then one could use the pattern spotting technique for word boundary hypothesization even at high error levels.

3.3. Distribution of subsentences

Another important result is the distribution of the subsentences formed by the word boundary hypothesization. If, even after word boundary hypothesization, there are large subsentences, then the savings in the lexical analysis stage may be marginal. Though in our method two word boundaries are hypothesized around most of the pattern clues, the sentence is halved only as both the boundaries are close. Hence, even if many word boundaries are located by our word boundary hypothesizer, there may still be many large subsentences left, hence, a high value for frequency might not necessarily mean large savings in lexical search, hence, the distribution of the subsentences with respect to their size is important from the point of lexical analysis.

The distributions of the subsentences at five error levels (0, 10, 25, 40 and 50%), along with the original sentence distribution are shown in Fig. 6. They indicate a gradual shift in the distribution towards larger subsentences as the errors are increased. The plot of the average length of subsentences is shown in Fig. 7, which also indicates the increase in the length of the subsentences with increasing errors. However, if the errors are limited, the distribution is still biased towards short subsentences and, hence, significant savings in the lexical analysis could be obtained at low error levels.

4. Discussion of the results

The above results have shown that many word boundaries could be hypothesized by spotting frequently occurring patterns. However, not all pattern clues perform equally well at word boundary hypothesization, in fact, pronouns, with all their morphological variations, dominate in terms of numbers (nearly half of the total pattern clues) but they are not equally effective in producing word boundary hypotheses. Moreover, their correctness is also lower, indicating that this group produced more incorrect hypotheses. Obviously, one could remove them from the pattern clues and thereby gain in correctness and also reduce the number of patterns to be spotted, which would result in a drop in the number of word boundaries located. Hence, depending on the application, one could trade-off the system performance (in terms of the number of word boundaries spotted) against system simplicity and correctness. For a very simple system, one could even eliminate the lexical clues used for verification and gain in system speed.

The above studies clearly demonstrate the utility of the pattern spotting approach in hypothesizing word boundaries. The pattern clues consist of frequently occurring patterns like case markers, conjunctions and pronouns. For English, recognition of these function words in speech is more error-prone compared to spotting of other words; this is because many function words are distorted in English speech. Hence, our approach

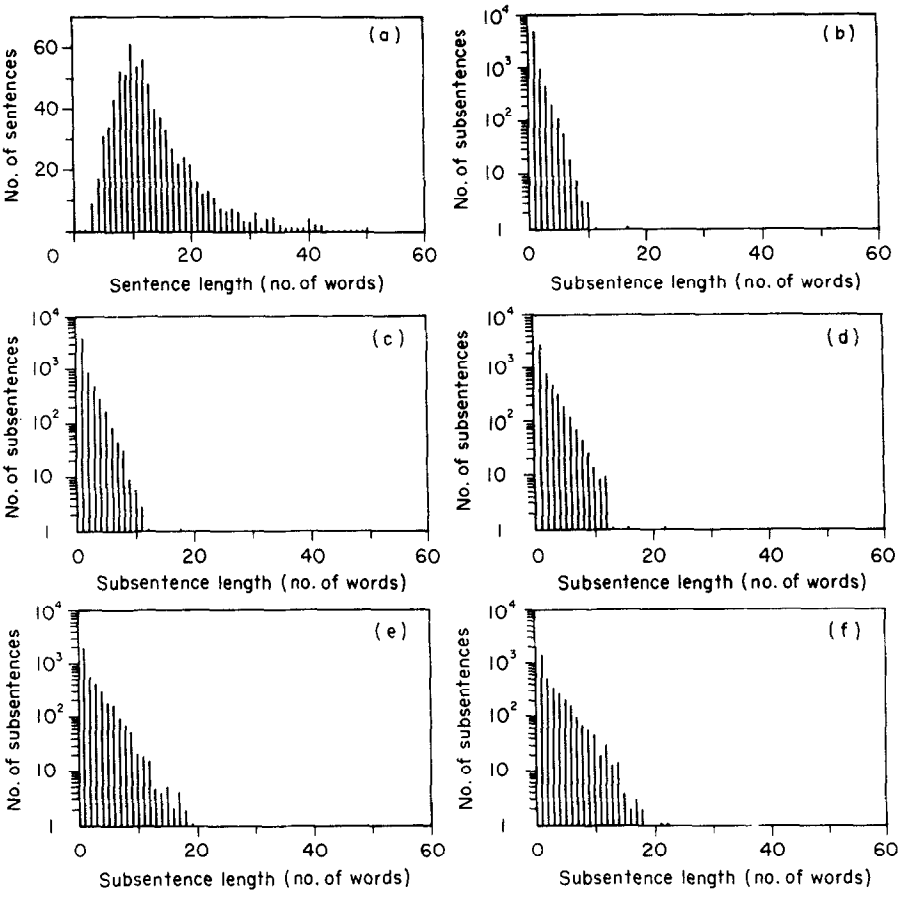

Figure 6. Distribution of the subsentences at various error levels. (a) The original distribution of the input sentences. (b)–(f) (plotted in log scale) The distributions of the subsentences after word boundary hypothesization for different error values, 0, 10, 25, 40 and 50%, respectively.

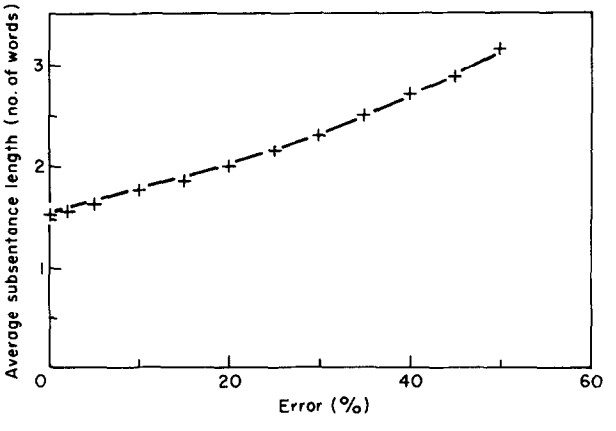

Figure 7. Plot of the average subsentence length vs. error.

may not work for word boundary hypothesization in English speech, but in Indian languages there are no significant differences between recognizing function words and other words. Hence, our method is well suited for tasks involving speech-to-text conversion in Indian languages.

We are grateful to M. Prakash who was associated in the earlier part of the work and helped in concretizing our ideas. Our thanks to C. Chandra Sekhar and S. Rajendran for reading the earlier drafts of the paper and suggesting many improvements.

References

Bhatia, Kailash Chandra (1970). *Hindi: bʰa:ʂa: me:n akʂar tatʰa: ʃabd ki: si:ma: (syllable and word boundaries in Hindi)*. Nagari Pracarini Sabha, Varanasi.

Eswar, P. (1990). A rule-based approach for spotting characters from continuous speech in Indian languages. Ph.D. Thesis, Indian Institute of Technology, Madras.

Harrington, J. & Johnstone, A. (1987). The effects of equivalence classes on parsing phonemes into words in continuous speech recognition. *Computer Speech and Language*, **2**, 273–288.

Harrington, J., Watson, G. & Cooper M. (1989). Word boundary detection in broad class and phoneme strings. *Computer Speech and Language*, **3**, 367–382.

Klatt, D. H. (1980). Overview of the ARPA speech understanding project. In *Trends in Speech Recognition* (W. A. Lea ed.), pp. 249–271. Prentice Hall, New Jersey.

Lea, W. A. (1980). Prosodic aids to speech recognition. In *Trends in Speech Recognition* (W. A. Lea, ed.), pp. 166–205. Prentice Hall, New Jersey.

Mangione, P. A. (1986). SSI's phonetic engine. *Speech Technology*, **3**(2), 84–86.

Ohala, M. (1983). *Aspects of Hindi Phonology*. Motilal Banarasidass, New Delhi.

Ramana Rao, G. V., Prakash M. & Yegnanarayana, B. (1989). Word boundary hypothesisation in Hindi speech. In *Proceedings of the European Conference on Speech Communication and Technology*, Vol. 2. Paris, pp. 360–363.

Yegnanarayana, B., Chandra Sekhar, C., Ramana Rao, G. V., Eswar, P. & Prakash, M. (1989). A continuous speech recognition system for Indian languages. In *Proceedings of the Regional Workshop on Computer Processing of Asian Languages*. Bangkok, pp. 347–356.