Dr. B. YEGNANARAYANA
Professor
Department of
Computer Science and Engineering
I.I.T., Madras-600 036, India

WORD BOUNDARY HYPOTHESISATION IN HINDI SPEECH

A THESIS

Submitted for the award of the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE AND ENGINEERING

by

GADDE VENKATA RAMANA RAO

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY
MADRAS **600 036,** INDIA.

FEBRUARY 1994

THESIS CERTIFICATE

This is to certify that the thesis entitled WORD BOUNDARY HYPOTHESISATION IN HINDI SPEECH submitted by GADDE VENKATA RAMANA RAO to the Indian Institute of Technology, Madras for the award of the degree of Doctor of Philosophy is a bonafide record of research work carried out by him under my supervision. The contents of this thesis have not been submitted and will not be submitted to any other Institute or University for the award of any degree or diploma.

Madras 600 036
Date: 18.2.94

(B. Yegnanarayana)
Professor
Dept. of Computer Sci.&Engg.
IIT, Madras.

ACKNOWLEDGEMENTS

I thank my research guide Prof. B.Yegnanarayana for the encouragement and support he had given me throughout the course of this research work. I had several discussions with him which greatly helped me in doing the research work as well as in writing the thesis. I gratefully acknowledge the time and effort that my research guide has spent on me.

I thank Prof. Kamala Krithivasan, Head, Department of Computer Science and Engineering, IIT, Madras for providing me the facilities to carry out my research work and also for the advice she had given me at critical times. I also thank the faculty and the staff of the Department of Computer Science and Engineering for the support they had given me.

This work could not have been completed but for the help given to me by my friends Dr. P. Eswar and M. Prakash, and colleagues in the Department of Computer Science and Engineering, Dr. Hema A. Murthy and C. Chandra Sekhar. I gratefully acknowledge their contributions, in matters technical and personal. The support they gave me, in the form of technical discussions and also in the form of pep talks, have greatly helped me.I also acknowledge the support of my colleagues, R. Sundar and N. Alwar and my friends Dr. K.V. Madhu Murthy and Dr. C.P. Mariadassou, who helped me at various stages of my research.

I acknowledge the cooperation I received from the members of Speech and Vision Laboratory, in particular from S. Rajendran and R. Ramaseshan. I am grateful to N. Naveen Kumar, who assisted me in collecting the speech data for my research.

CONTENTS

Chapter 3

Chapter 4

Chapter 9

ABSTRACT

This thesis addresses the problem of word baundary hypothesisation, which occurs in the context of a speech-to-text conversion system for Hindi. Normal speech is continuous sequence of sounds with no specific pauses to indicate word boundaries. Hence, to convert speech into the corresponding text, it is necessary to identify the boundaries between the words in the speech. In speech recognition systems, this is usually done by matching a symbolic representation of the input speech against a lexicon to obtain a string of words. However, this lexical analysis is computationally expensive. On the other hand, if some word boundaries can be identified even before lexical analysis, the complexity of the lexical analyser will be significantly reduced. This thesis focuses on the issues in the identification of word boundary clues, and on the effectiveness of the identified clues.

First, the importance of even partial identification of the word boundaries in reducing the computational complexity of lexical analysis is demonstrated through simulation studies. Later, studies are described for identification of clues for hypothesising word boundaries, which are based on the four knowledge sources, language, lexicon, prosody and acoustic-phonetics. The effectiveness of these clues is reported in terms of the percentages of the correct and incorrect word boundary hypotheses produced by the clues.

Studies in this thesis clearly demonstrate the following:

(i) Reduction in the complexity of lexical analysis even with a partial knowledge of word boundaries in a text,

(ii) Existence of language and lexical clues that can be exploited for placing a significant number of word boundaries correctly, and

(iii) Existence of speech-related clues such as prosodic and acoustic-phonetic clues, which can be used to identify many word boundaries in speech.

Chapter I

INTRODUCTION TO WORD BOUNDARY HYPOTHESISATION

1.1 Word boundary hypothesisation problem

The problem of word boundary hypothesisation(WBH) arises in the context of human communication with machines. The problem can be stated as follows: Given a string of symbols representing a sentence, word boundaries are to be placed in the symbol string to convert it into a string of words. This problem is relevant especially in the context of speech input to a machine, where the speaker does not consciously indicate the word boundaries while speaking. The work reported in this thesis is on the identification of clues to perform word boundary hypothesisation in a speech-to-text conversion system for the Indian language Hindi.

The role of the word boundary hypothesisation problem in speech recognition can be understood if one examines the way humans speak. Normal speech is a sequence of sounds with very few pauses to indicate word boundaries. To convert speech into the corresponding text, one needs to identify the positions of the missing word boundaries. In speech recognition systems, the word boundaries may be obtained by matching a symbolic representation of the speech, produced by a speech signal-to-symbol converter, against a lexicon. However, this process, called lexical analysis, produces a large number of alternate word strings, when the input symbol sequence contains errors. Moreover, it also involves a large number of computations. Since a significant percentage of the speech recognition time is spent on lexical analysis [Wolf and Woods 1980], one needs to simplify the lexical analysis to speed up the speech recognition. If some word boundaries can be identified before performing lexical analysis, the performance of the lexical analyser, both in number of computations and in accuracy, can be significantly improved.

Consider the operation of a lexical analyser when some word boundaries are

known. The lexical analyser can now match each substring between word boundaries against the lexicon and produce word alternatives. Note that even if all the word boundaries are known, the lexical analyser still has to match the substrings between the boundaries against the lexicon, due to possibility of errors in the input symbols. From the word alternatives produced by the lexical match, sentence alternatives can be formed by constructing strings of the word alternatives. Since the word start and end points are known, many sentence alternatives that contain words spanning across word boundaries, which would have been produced in the absence of word boundaries, are eliminated. This results in a reduction in the number of alternatives for a sentence. Also, the time needed for lexical matching will be reduced. It can be reduced further, if the lexical matches for words can be done in parallel.

Word boundary hypothesisation also simplifies the handling of unknown words, i.e., words which are not listed in the lexicon, such as names of persons and places. In the absence of word boundaries, on detecting an unknown word, the lexical analyser will have to search several alternative positions to detect the start of the next word. If word boundaries are known, it can start the lexical match from the next word boundary and leave the interpretation of the unknown word to later stages or to the user.

The development of a word boundary hypothesiser also simplifies the design of a speech-to-text conversion system. The main objective of such a system is to generate a text corresponding to the input speech. Typically, a speech-to-text conversion system consists of a speech signal-to-symbol conversion system and a symbol-to-text conversion system. Assuming that the symbols correspond to the orthographic characters of the language, the symbol string produced by the signal-to-symbol conversion differs from the desired text mainly in the missing word boundaries. A symbol string without word boundaries is difficult to read even for humans. The main

purpose of the symbol-to-text conversion system is to make the symbol string readable, by providing the missing word boundaries. The text can be further corrected, if necessary, using the higher level knowledge sources such as syntax and semantics to make it more meaningful.

In summary, the following are the advantages of the word boundary hypothesisation:

1. The complexity of lexical matching involved in large vocabulary speech recognition can be significantly reduced.

2. Unknown words can be handled.

3. If most of the word boundaries can be hypothesised, a useful speech-to-text conversion system can be developed, with only a speech signal-to-symbol converter and a word boundary hypothesiser.

It is interesting to note that a meaningful text with word boundaries can be read easily, even with some errors in characters and in word boundaries (see Fig.1.1 for illustration). Thus word boundary hypothesisation plays a crucial role in producing a readable output from a speech-to-text conversion system. But continuous speech does not contain any direct clues, such as pauses, to word boundaries. However, it is interesting to note that there are several language features which can be exploited for hypothesising word boundaries. Since the original input is speech signal, one can also exploit speech related clues for word boundary hypothesisation.

The objective of this thesis is to establish the significance of word boundary hypothesisation in speech recognition and to demonstrate that language and speech related clues do exist, which can be effectively used to hypothesise word boundaries. It is interesting to note that even a partial success in word boundary hypothesisation using these clues would generate a text which is significantly better than a text without word boundaries, from a readability point of view. Moreover, such a text with a few word

alounddhelongtableheagsnoddedenunisonyeswereedthuculturemustshangepropessors shoultperewardedeccordingtotheerdeachinguffectivenessnotportheirfmeanrezearchin sempatheticyibrationmyowlheadnoddeduponddownjustlikeatherwiseheedsthenodging continoedigglincedfultivelyatmynaighbourendcaughhementheactobmenitoringmaown degreeoffinceritynotsomuchaseripplearashedowanhisexpressiongetrayedalylackof convictionbhiswasseriousstuff

(a)

alound dhe long table heags nodded en unison yes we abreed thu culture must shange propessors shoult pe rewarded eccording to theer deaching uffectiveness not por their fame an rezearch in sempathetic yibration my owl head nodded up ond down just like ather wise heeds the nodging continoed i glinced fultively at my naighbour end caught hem en the act ob menitoring ma own degree of fincerity not so much as e ripple ar a shedow an his expression getrayed aly lack of conviction bhis was serious stuff

(b)

Around the long table heads nodded in unison. Yes we agreed the culture must change. Professors should be rewarded according to their teaching effectiveness not for their fame in research. In sympathetic vibration my own head bobbed up and down just like other wise heads. The nodding continued. I glanced furtively at my neighbour and caught him in the act of monitoring my own degree of sincerity. Not so much as a ripple or a shadow in his expression betrayed any lack of conviction. This was serious stuff.

(c)

Fig 1.1 An illustration of the improvement in the readability of a text due to word boundaries. In 1.1(a), a text with nearly 50% of the words in error(but only 10% of letters in error)is shown without any word boundaries. The same text is shown in 1.1(b) with word boundaries. It can be seen that the text in (b) is easier to read compared to the text in (a). The actual text is also shown in 1.1(c).

boundaries, will also reduce the complexity in the processing of later stages, such as lexical, syntactic and semantic analyses.

1.2 Issues in word boundary hypothesisation

In the previous section, it was argued that word boundaries can reduce the number of alternatives produced by a lexical analyser, and they can also reduce the computation involved in lexical analysis. However, one can also obtain a quantitative assessment of the effect of word boundaries on lexical analysis and thereby establish the importance of the word boundary hypothesisation in lexical analysis. This is the first task in our work.

The main issue to be addressed in such a study is the improvement in the performance of the lexical analyser due to word boundary hypothesisation. This involves two studies: (1) a study of the effect of errors in the input symbols on the performance of the lexical analyser when the input sentences contain no word boundaries, and (2) a study on the performance of the lexical analyser for the same input sentences when they contain word boundaries. In these studies, the performance of the lexical analyser is measured using two terms: (i) the number of alternate word strings (henceforth referred as alternatives) matching the input sentence, and (ii) the time spent on the lexical analysis. These measures reflect the different ways in which lexical analysis influences the overall speech recognition process. The first measure is an estimate of the computational load imposed on the later stages of processing, such as syntactic and semantic analyses, whereas the second measure is an estimate of the computation involved in the lexical analysis.

Once the importance of the word boundary hypothesisation is established, the next task is to identify some clues to perform word boundary hypothesisation. Since word boundary hypothesisation can be viewed as converting a string of symbols without

word boundaries into a text with word boundaries, one can exploit some language features to perform this task. In particular, one can utilise the constraints on the formation of words and their occurrence within a sentence. Thus the lexical and higher level linguistic knowledge sources such as syntax and semantics, which wili be referred henceforth as language knowledge, are needed to identify the clues for word boundary hypothesisation. In this context, we have identified two studies, one on the use of language clues to perform word boundary hypothesisation and another on the use of lexical clues.

In the context of speech recognition, the symbol string produced by the speech signal-to-symbol converter, usually contains errors. It is possible that in the presence of errors, the lexical and the language clues may not be as effective as in the case of error free input. But there are several speech related knowledge sources such as prosody and acoustic-phonetics which can be examined to identify additional clues. Thus we have identified additional studies on the identification of prosodic and acoustic-phonetic clues for word boundary hypothesisation.

From the above discussion, it is clear that clues based on the four knowledge sources, acoustic-phonetic, prosodic, lexical and language are useful for word boundary hypothesisation. However, it may not be possible to integrate all of these clues into a single module. This is due to the fact that the acoustic-phonetic and prosodic clues can be applied on the speech signal only, whereas the lexical and language clues are applicable on the symbol string output by the speech signal-to-symbol converter. Hence to perform word boundary hypothesisation, one needs to obtain the various clues from the appropriate knowledge sources and use them, rather than build a single module for word boundary hypothesisation. The present work focuses on the identification of clues to perform word boundary hypothesisation, and no attempt is made to address issues concerned with the application of these clues in an integrated fashion.

The following issues are addressed in this thesis:

1. Effect of wordboundary hypothesisation on lexical analysis,
2. Identification of language clues for word boundary hypothesisation,
3. Identification of lexical clues for word boundary hypothesisation,
4. Identification of prosodic clues for word boundary hypothesisation, and,
5. Identification of acoustic-phonetic clues for word boundary hypothesisation,

1.3 Studies on word boundary hypothesisation for Hindi

The first study in our work is on the effect of word boundaries on lexical analysis. Experiments were carried out to compare the time spent on lexical matching for the two cases of word boundaries known and unknown. The results proved conclusively that the presence of word boundaries in the input sentences can reduce the time spent on lexical analysis significantly. For example, lexical analysis time for the sentence ***pa:t^{h}sa:la: me: e:k nai: laḍaki: ne: prave:ṣ liya: hai*** was reduced by a factor of 1000 when the word boundaries were known, compared to the case when the word boundaries were unknown, when the sentence had 10% errors in phonemes.

Studies were also carried out to estimate the increase in the number of sentence alternatives for the input sentences. The studies showed that the presence of word boundaries, reduced the number of alternate word strings matching the input sentence. For example, for the sentence, ***pa:t^{h}sa:la: me: e:k nai: laḍaki: ne: prave:s liya: hai*** produced nearly twice the number of alternate word strings when the word boundaries were unknown, compared to the case when the word boundaries were known, at an input error rate of 10%. Thus the results of the study established that word boundary hypothesisation improves the performance of the lexical analyser significantly.

The next study relates to the identification of language clues for word boundary hypothesisation. The language clues proposed correspond to the frequently occurring

words. The idea is to spot symbol sequences corresponding to these frequently occurring words in the input sentences and hypothesise word boundaries around them. Depending on the frequency of occurrence of these words, one can hypothesise a large number of word boundaries.

The above idea was tested on a Hindi text. A number of clues corresponding to the frequently occurring words such as case markers, pronouns and other function words were used. The results show that about 70% of the word boundaries were detected correctly with errors around 20%.

However, in the context of speech recognition, the symbol string generated by the signal-to-symbol conversion contains errors. In our study, these errors were simulated in the input sentences and the performance of the language clues was estimated at various error rates. The results show that the language clues are useful for hypothesising word boundaries even at high error rates. For example, even at an input error rate of 50%, the language clues detected nearly 35% of the word boundaries with less than 35% incorrect hypotheses.

The third study is on the identification of lexical clues. In this, lexical constraints such as the constraints on the sequences of phonemes, were proposed as clues to hypothesise word boundaries. The idea is to identify all phoneme sequences which do not occur word-internally, and then hypothesise word boundaries within such sequences occurring in a sentence. For example, in English, the phoneme sequence *mgl* does not occur word-internally. Hence if it is observed in a text (as in some *glass*), one can hypothesise a word boundary within the sequence.

A number of phoneme sequences were used, ranging from simple sequences of vowels(V^+) and consonants(C^+) to much longer sequences of both vowels and consonants (CV^+C and VC^+V). Note that the superscript $^+$ is used to indicate a sequence of one or more symbols. Using these sequences, word boundaries were

detected in a Hindi text. The results show that longer sequences detect more word boundaries but they are also more error prone. Moreover, use of long sequences also increases the uncertainty in the actual location of the word boundary. Hence, in practice, this may not result in any significant improvement in lexical analysis, when compared to the shorter sequences.

In the speech context, the lexical clues are to be applied on a symbol string which may contain errors. Tests were conducted using sentences in which speech-like errors were simulated, and the performance of the lexical clues was estimated. The results show that the lexical clues are useful even for large percentage errors in the input symbol string.

In addition to the language and lexical clues, one can also exploit prosodic and acoustic-phonetic knowledge sources. In the study on the identification of prosodic clues for word boundary hypothesisation, four prosodic features of pause, duration, amplitude and pitch were considered. Of these, pauses are the simplest and most reliable clues to word boundaries. In our study, it was found that pauses can be used to detect only a few word boundaries, less than 20% of the total boundaries. It was also found that a simple strategy of hypothesising a word boundary after every long vowel results in the detection of many word boundaries (more than 70%) with errors less than 25%. Pitch changes were also found useful to hypothesise word boundaries. In our study, it was found that a drop in FO can be used to hypothesise a word boundary. This detected more than 70% of the word boundaries with errors less than 30%. A combination of pause, duration and pitch resulted in the detection of more than 75% of the word boundaries with errors less than 15%.

In the study on the identification of acoustic-phonetic clues, two simple clues, based on changes in the first formant were considered. The first clue uses changes in

F1 position at a vowel-consonant boundary to hypothesise word boundaries. This detected nearly 50% of the word-final vowels with errors around 25%. The second clue uses changes in F1 energy as a clue to word-final vowels. This detected about 30% of the word-final vowels with errors around 25%.

The above studies demonstrated the significance of word boundary hypothesisation in speech recognition and identified several word boundary clues based on the language, lexical, prosodic, and acoustic-phonetic knowledge sources. A detailed description of the studies is given in later chapters. The organisation of the thesis is given in the following section.

1.4 Organisation of the Thesis

In the earlier sections, the problem of word boundary hypothesisation was introduced and its need in the context of a speech recognition system was discussed. In the next chapter (chapter 2), a review of the studies on word boundaries is presented. To establish the importance of word boundaries in reducing the complexity of lexical analysis, a study was conducted in which the effect of word boundaries on lexical analysis was examined. This is described in chapter 3. This is followed by the studies on the identification of various clues for word boundary hypothesisation. Four different studies were made, each study concentrating on clues based on a particular knowledge source. In chapters 4, 5, 6 and 7, clues for each of these studies are described. In chapter 4, a study on the use of some language clues to hypothesise word boundaries is described. In chapter 5, studies on the use of some lexical clues, namely, phoneme sequence constraints, for word boundary hypothesisation are presented. In chapter 6, studies on the use of the prosodic features of pitch, duration and amplitude, in hypothesising word boundaries are described. In chapter 7, studies on the use of word boundary clues based on acoustic-phonetic knowledge are presented. The performance of the above four types of clues, namely, language, lexical, prosodic and acoustic-

phonetic clues, in reducing the lexical analysis time is presented in chapter 8. The work is summarised in chapter 9 and some issues for further investigation are also indicated in it.

Chapter 2

A REVIEW OF THE STUDIES ON,WORD BOUNDARY HYPOTHESISATION

2.1 Introduction

Several studies have been reported in literature in which a number of word boundary hypothesisation techniques were described. These studies can be broadly divided into two categories:(i) studies which addressed the role of word boundaries in improving the performance of a lexical analyser, and (ii) studies which proposed some clues for word boundary hypothesisation. Most of the reported studies were for English, though a few other studies also addressed the problem of word boundary hypothesisation for other languages like German and Japanese. All of these studies were reviewed in this chapter.

The review is also organised in two sections, with each section containing a review of the studies of the above two categories. Initially, in section 2.2, the studies which addressed the role of word boundaries in lexical analysis were reported. The studies on word boundary hypothesisation techniques were reported in the later section (section 2.3), which is further divided into four subsections, with each section reviewing the role of each of the four knowledge sources, language knowledge, lexical knowledge, prosodic knowledge, and acoustic-phonetic knowledge. In subsection 2.3.1, application of language knowledge for word boundary hypothesisation is reviewed. In subsection 2.3.2, word boundary hypothesisation techniques based primarily on the lexical knowledge are reviewed. In subsection 2.3.3, studies which examined the role of prosodic knowledge in word boundary hypothesisation are reviewed, and in subsection 2.3.4 word boundary hypothesisation techniques based on the acoustic-phonetic knowledge are described. This organisation of the studies is based on the primary clues used, although, in some studies, clues based on other knowledge sources were also used. For example, the Metrical Segmentation Strategy(MSS), described in subsection

2.3.3, uses both prosodic and language features and it was placed under the studies on prosodic clues because it primarily uses prosodic features.

2.2 Role of word boundaries in improving lexical analysis

The main aim in these studies is to explore the improvement in the performance of a lexical analyser and thereby establish the need for word boundary hypothesisation. This can be divided into two separate studies, (i) a study in which the performance of the lexical analyser on an input without word boundaries is estimated, and (ii) a study in which the performance of the lexical analyser on an input with word boundaries is estimated. The results are compared with those of (i) to establish the necessity of word boundary hypothesisation.

Two studies were reported in literature on the effect of word boundary ambiguity on lexical analysis. However, these studies, as mentioned later, have addressed the issues only partially. The first study was conducted at the Centre for Speech Technology Research(CSTR), Edinburgh [Harrington and Johnstone 1987]. In this study, the performance of a lexical analyser in the absence of any word boundaries in its input was studied. The performance of the lexical analyser was measured in terms of the number of alternate word strings matching the input sentence. The input to the lexical analyser was represented in two representations, phoneme and midclass [Dalby, Laver and Hiller 1986; Harrington and Johnstone 1987], and the number of alternate word strings matching a sentence (without any word boundaries) was estimated for both the representations. A dictionary containing the 4000 most frequent English words was used. The dictionary was represented internally as a tree-structured dictionary and a left to right matching strategy similar to the one described in [Cole and Jakimik 1980] was used. A total of 50 sentences were used in the matching. The results of the study are in the following:

1. When a midclass representation was used for the input, 32 of the 50 sentences had more than 10 million alternate word strings matching them. The largest number of alternate word strings was 3.25×10^{18} and the average number was 8.47×10^{16}.
2. When the input was represented in a phonemic form, only 3 of the 50 sentences matched 10,000 or more alternate word strings. The largest number of alternate word strings was 66,528 and the average number was 2,491.

The results of this study showed clearly that in the absence of word boundaries the higher level analysers (syntax and semantic) will have to select the correct sentence from a large number of alternatives. It can also be seen that even when the input utterance is correctly converted into phonemes, the absence of word boundary information leads to many word string alternatives. This problem is further compounded by the inaccuracies in the speech signal-to-symbol conversion as seen from the results for the midclass representation. However this study did not address the following issues:

1. The effect of word boundaries in reducing the number of alternate word sequences matching an utterance, and
2. The effect of word boundaries in reducing the computation time involved in lexical analysis.

The second study [Briscoe 1989] addressed the first issue to some extent. In this study four different lexical match strategies were compared in terms of the number of alternate word strings produced for a sentence. The strategies considered in this study are given below:

1. Lexical match was performed at each phoneme. In effect, this means that a word boundary was assumed after every phoneme. Moreover, knowledge of the previously successful matches was not used in performing the next match. This assumption is very unrealistic and practically no speech recognition system uses this.

2. Lexical match was performed at each syllable, which means that a word boundary was assumed after every syllable.

3. Lexical match was initiated at the beginning of the sentence and subsequently at the conclusion of each successful lexical match. This strategy was used in most speech recognition systems [Cole and Jakimik 1980].

4. Lexical match was initiated at each strong syllable. This strategy was based on the Metrical Segmentation Strategy(MSS) suggested for English [Cutler and Carter 1987]. Thus, in this strategy, information regarding word boundaries was used in matching.

The above four lexical match strategies were compared on three types of input representation of increasing complexity, namely, (a) A sentence represented as a sequence of phonemes, (b) a sentence transcribed such that the strong syllables were transcribed exactly whereas the weak syllables were transcribed into their broadclass categories, and (c) a sentence transcribed such that the strong syllables were transcribed into their midclass categories and the weak syllables into their broadclass categories. Thus the above three types of inputs were of increasing complexity. The results of the study were as follows:

1. For input type (a), Strategies 1 and 2 produced more matching word sequences than strategies 3 and 4, as expected.

2. For input types (b) and (c), strategy 1 again produced the highest number of matching word sequences. But unlike the earlier case, strategy 2 performed better than strategy 3 in that it produced less number of matching word sequences. Strategy 4 performed best by producing the least number of matching word sequences.

The above results established that knowledge of the word boundaries (even if it is only partial as in strategy 2) can improve the performance of the lexical match, especially when the input is only partially known (midclass or broadclass strings). Of

particular interest is the performance of strategy 2, in which the word boundaries were restricted to syllable boundaries (which is not very restrictive), and yet the strategy outperformed the strategy 3. Thus this study clearly established the need for word boundaries. However, this study compared the performances of the lexical analyser for three inputs of different complexity, but it did address the following issues:

1. Effect of errors in the input on the performance of a lexical analyser,

2. Effect of word boundaries on the performance of the lexical analyser when there are errors in the input.

Studies were conducted by us to address these issues and they are described later in chapter 3.

2.3 Techniques for word boundary hypothesisation

2.3.1 Word boundary hypothesisation techniques based on language knowledge

Not many studies were reported on the use of language knowledge sources such as syntax and semantics. However, studies reported in literature which primarily use some other knowledge source, also make use of language features. The feature used in many of these studies is the word frequency, i.e., the relative frequency with which a word (or a class of words) occurs in a text. The reason for this is simple. Whatever clues are observed at word level, they can be useful only if the words containing the clues occur frequently in a text. Otherwise such clues are applicable only occasionally. For example, consider the occurrence of the sound n in Hindi. It occurs mostly in word-final position. Thus the presence of n in a Hindi sentence can be used to hypothesise a word boundary. However, the words containing n occur rarely, and hence the utility of n in hypothesising word boundaries is practically nil.

In our studies, reported later in chapter 4, the word frequency information was used as a clue to hypothesise word boundaries.

2.3.2 Word boundary hypotheskation techniques based on lexical knowledge

For English, several studies were carried out, both by linguists and speech scientists, to identify lexical clues which can be used to detect word boundaries [Shipman and Zue 1982]. The clues used were basically constraints on sequences of phonemes, also known as phonotactic constraints. In one study [Lamel and Zue 1984], sequences of consonants of the form C^+ ($^+$ indicating a sequence of one or more consonants) were identified. These sequences were used to hypothesise some of the word boundaries in several texts. It was also suggested that such clues can also be used to detect word boundaries at a broadclass level. To identify the exact location of the word boundary in the consonant string (for example, a two consonant sequence C_1C_2 can contain a word boundary in three positions; before the sequence, within the sequence or after the sequence), it was suggested that additional knowledge such as acoustic-phonetics can be used. Though results on actual texts were not reported, the number of word boundaries that can be detected by these clues appear to be limited.

A more recent and exhaustive study on the use of the phoneme sequence constraints was done by Harrington [Harrington, Johnson and Cooper 1987] in which sequences of the types CV, VC and CVC were considered. In this study, all word-internal sequences of the given type were extracted from a dictionary. Also all possible sequences that can occur across word boundaries were found by considering all possible pairings of the words. From these word boundary sequences the word-internal sequences were removed. Thus the remaining sequences were sequences which occur only across word boundaries and these sequences were used to hypothesise word boundaries. It was found that nearly 45% of the word boundaries can be detected in an English text represented in a phonemic form with incorrect hypotheses less than 4%.

In a later study [Harrington, Watson and Cooper 1989], the above sequences were used to hypothesise word boundaries in strings represented using broadclasses. It

was found that at broadclass level, the applicability of the phoneme sequence constraints is limited. Results showed that the number of word boundaries detected were around *2%* of the word boundaries in the text with false alarms as high as 22%. Thus the utility of the phoneme sequence constraints to hypothesise word boundaries in texts containing speech signal-to-symbol conversion errors appears to be limited.

For Hindi, Ohala [Ohala 1983] gave a list of valid word-internal sequences of consonants and vowels(C^+ and V^+) which were called sequential constraints. However in these, sequences of the form CV, VC and CVC were not considered. But, Ohala does mention that some sequences such as *kyi* do not occur in word-internal position. As yet no study is reported on the use of these clues for detecting word boundaries in Hindi.

Our studies on the use of the phoneme sequence constraints in hypothesising word boundaries are reported in chapter 5.

2.3.3 *Word boundary hypothesisation techniques based on prosodic knowledge*

Most of the initial studies on word boundary detection focussed on the use of prosody. Many of these were studies on human perception and used listeners to determine the word boundaries in what was heard. In one study [Nakatani and Schaffer 1978] it was shown that stress patterns aid listeners in the detection of word boundaries. In this study, 'reiterant speech' was used to eliminate the phonetic information from a phrase. In reiterant speech, all syllables in a phrase are replaced by nonsense syllables like 'ma'. For example, 'Mary had a little lamb' became 'Mama ma ma mama ma'. The effect of syntax was minimised by using sentences with the same structure and by restricting the number of syllables to be replaced to three containing a noun phrase of an adjective and a noun. For example, the sentence 'The remote stream was perfect for fishing' became 'The mama ma was perfect for fishing'. Thus only prosodic cues were available to the listener. Using several such sentences, listeners

were asked to identify the boundaries between the words in the phrase. The results showed that the listeners were able to detect many word boundaries better than chance. An analysis of the results showed that stress and rhythm were the primary cues used by the listeners to detect worcl boundaries.

Similar studies on the use ad prosodic cues were performed for other languages also. In a study on Japanese speech [Nakagawa and Sakai 1979], utterances of one to three word sequences of city names and digits was synthesised. When the voiced segments in the synthesised speech were replaced by sinusoids and the unvoiced segments by white noise with both pitch and energy unaltered, 95% of the word boundaries in the sequences were still recognised by the listeners. When either energy or pitch is modified, the recognition rate dropped to 92%.

In a later study [Lea 1980], an algorithm was developed to detect major syntactic boundaries in English speech using fall/rise patterns in the pitch contour. The algorithm looks for substantial decreases in FO (7% or more) followed by an increase (7% or more). It then marks a boundary at the last of the lowest FO values in the valley. Application of this algorithm on a data of 230 sentences, resulted in locating nearly 90% of the major syntactic boundaries correctly with false alarm rate between 5 to 10%. In the same study it was suggested that some word boundaries can also be detected by using durational changes caused by 'prepausal lengthening'. It was also suggested that by detecting pauses in speech, which occur as long silences, one can detect a few word boundaries.

The relation between the prosody and the syntactic structure of a sentence was also examined in some recent studies. It was shown that boundary tones, which are distinctive changes in pitch, signal major boundaries in phrases and sentences [Beckman and Pierrehumbert 1986]. In a recent study, it was found that abrupt changes

in speaking rate indicate phrase boundaries in speech [Wightman and Ostendorf 1991]. A technique to detect phrase boundaries using pitch patterns was also reported [Simodaira and Kimura 1992]. In this study, a set of reference pitch patterns were obtained initially from a training data set. For the test data, the pitch pattern was obtained and it was matched against the stored patterns using dynamic programming. It was reported that this technique detected nearly 88% of the phrase boundaries correctly.

Recently, a word boundary detection technique called Metrical Segmentation Strategy(MSS) [Cutler and Norris 1988] was developed for English based on a strong/weak classification of the English syllables. It was based on an earlier observation that many English content words contain strong word initial syllables [Cutler and Carter 1987]. Thus, in the proposed strategy, a strong syllable was hypothesised as a word initial syllable. In one study [Harrington, Watson and Cooper 1989], it was found that this strategy results in the detection of nearly 47% of the word boundaries in a text with a false alarm rate of 33%. In another study using MSS [Cutler 1990], both the weak and strong vowels were used to hypothesise word boundaries. The weak vowels were used as clues for grammatical words while the strong words were used for content words. It was reported that the strategy correctly detects more than 80% of the content words in an utterance with less than 15% errors.

The use of MSS in human speech perception was also established in a recent study [Cutler and Butterfield 1991a]. In this study the speakers were informed that the listeners were having difficulty in finding word boundaries and were asked to produce deliberately clear speech to aid them. This speech was then analysed to identify the clues supplied by the speakers to indicate the word boundaries. The results showed that pause and lengthening of preboundary syllables were the clues produced by the speakers. Interestingly, these clues were stronger at word boundaries preceding weak

syllables than at word boundaries preceding strong syllables, indicating that the speakers are aware of the use of strong syllables to indicate word beginnings. Thus these findings confirmed the use of strong syllables to detect word boundaries by humans and validate the use of MSS.

The above studies established that prosodic knowledge can be used to hypothesise word boundaries in speech. The studies have also identified some prosodic features, such as pause, duration and pitch(F0), as possible clues to word boundaries. Our studies for Hindi using these prosodic features for word boundary hypothesisation are reported in chapter 6.

2.3.4 ***Word boundary hypothesisation techniques based on acoustic-phonetic knowledge***

Not many word boundary detection techniques were reported in literature in which the acoustic-phonetic knowledge was explicitly used. However, two techniques were reported which used spectral information to detect the word boundaries. Both operate directly on the speech signal and hypothesise word boundaries in it.

The first technique was developed for application in a connected word recognition task [Zelenski and Class 1983]. It used an algorithm which was based on estimation principles. In this the input speech signal was divided into a sequence of windows. The signal in the window was represented by a parameter vector x = $\{x_1, x_2, \ldots x_L\}$, where each of the x_i represent a speech parameter such as one of the outputs of a filter bank. The word boundary hypothesisation problem was posed as one of classifying a given window into one of the two classes: (i) class1 window, containing a word boundary, and (ii) class2 window, not containing a word boundary. Ideally the classifier should produce an output z, where $z = 1$ for window class 1, and $z = 0$ for window class 2.

The target value z can be approximated by an estimation d which is computed

by an estimator function E from the parameter vector x. Thus d = E(x). The estimator was optirnised for minimum mean squared estimation error. Depending on the ,type of the estimator function E, one would obtain a different performance in the classification. In the study, two types of estimator functions, linear and quadratic, were used.

The word boundary detector was first trained on a sample data to obtain the estimator function E. Then this estimated E was used to hypothesise word boundaries in other data. The system was tested on a connected digit task. It achieved very good recognition accuracy (> 90%) with incorrect hypotheses limited to less than 5%.

One important problem with this technique was that the system needs to be trained for all possible word pairs and hence it is useful only in the context of connected word recognition where the small vocabularies permit such training. Moreover, even for small vocabularies, the computation was large, being proportion21 to the vocabulary size.

The second technique [Ukita, Nitta and Watanabe 1986] tried to reduce the computations involved in training the classifier (or estimating E). In this, the problem of word boundary hypothesisation was posed as one of hypothesising a variable size window, that contains a word boundary, based on a measure called 'spectral change'. The spectral change for a frame i was defined as $SC(i) = | x_{i-1} - x_i | / | x_i |$ if $x_i \neq 0$, and, $SC(i) = 0$ otherwise, where x_i was a parameter vector of the i-th frame, where each element of the vector corresponds to the output power of one channel of a filter bank.

Depending on the spectral change computed, a window size was hypothesised. The hypothesisation was done such that for a small spectral change, the hypothesised window size was large and vice versa. The hypothesised window size indicates that a word boundary is likely to be present in a window of that size taken around the current

frame. This technique also achieved more than 90% recognition accuracy on a connected digit recognition task involving sequences of four Japanese digits.

Both the above techniques achieved a recognition accuracy of 90% or more and recognised many word boundaries. However both the techniques were task and vocabulary dependent and were tested only on small vocabularies (100 words or less). Extending them to task independent and large vocabulary continuous speech recognition will require large training and hence they are not suitable in the context of continuous speech recognition.

Whiie the above techniques were not applicable for word boundary hypothesisation in continuous speech, they have shown that spectral clues do exist for hypothesising word boundaries. In our studies, reported in chapter 6, some spectral features based on the acoustic-phonetic knowledge were used as clues for word boundary hypothesisation in continuous speech.

2.3.5 Word boundary hypothesisation techniques for Hindi

All of the earlier studies reviewed above were performed for English and in a few cases for German and Japanese languages. In the context of word boundary hypothesisation for continuous Hindi speech, studies were reported (other than the ones reported in this thesis) on the use of prosodic knowledge in the form of pitch variations to hypothesise word boundaries [Madhukumar 1993; Rajendran and Yegnanarayana 1994]. The technique is described later in chapter 6 where its performance is studied and several modifications are made by us to improve its performance.

2.4 Summary and Conclusions

In this chapter a review of the studies conducted on word boundary hypothesisation and its significance in speech recognition was presented. From these

studies, the following issues are identified for further study:

1. There is a necessity to perform studies on the significance of word boundaries in improving lexical analysis, especially in the presence of errors in the input text. In particular, one needs to perform studies to examine the effect of input errors on the performance of a lexical analyser, and how the presence of word boundaries in input sentences can improve the performance.

2. Almost all of the word boundary hypothesisation techniques developed are language specific. Hence, one needs to conduct studies addressing the problem of word boundary hypothesisation in Hindi speech.

In the succeeding chapters studies addressing these issues were reported.

Chapter 3

SIGNIFICANCE OF WORD BOUNDARIES IN LEXICAL ANALYSIS

3.1 Introduction

In the first chapter, arguments were presented to justify the need for performing word boundary hypothesisation before performing lexical analysis. Three reasons were given in this regard: (i) improvement in the lexical matching, (ii) easier handling of unknown words, and (iii) easy development of speech-to-text conversion systems. Reasons (ii) and (iii) are self evident. In this chapter, studies are reported establishing (i), i.e., improvement in lexical analysis due to word boundary hypothesisation.

There are two ways in which the word boundary hypothesisation can affect lexical analysis: (i) by reducing the number of alternate word strings for a given input sentence, and (ii) by reducing the computation involved in lexical matching. While these two are related, their implications in the context of speech-to-text conversion are different. A reduction in the number of alternate word strings means a reduction in the computations for the later stages of processing, such as syntactic and semantic analyses. On the other hand, a reduction in the lexical analysis time means a reduction in the total time for speech recognition. Since lexical analysis constitutes a major part of the speech recognition task, a reduction in the lexical analysis time can significantly speed up the overall recognition.

In chapter 2 (section 2.2), two earlier studies on the effect of word boundaries on lexical analysis were described. These studies established that the knowledge of word boundaries (even if it is only partial) can improve the performance of the lexical analyser, especially when the input is only partially known (midclass or broadclass strings) as likely in a speech recognition system. However these studies have not addressed the following issues in detail:

1. The effect of errors in input symbols on the performance of a lexical analyser.

2. The effect of word boundary information on the performance of a lexical analyser.

Studies were conducted to address these issues. A lexical analyser was developed to study its performance for two types of inputs: (i) input sentences without word boundaries, and (ii) input sentences with word boundaries. The performance of the lexical analyser was estimated using two measures: (i) the number of alternate word strings for a given input sentence, and (ii) the time spent on the lexical analysis, for varying input errors.

In the next section, the lexical analyser used in the studies is described. In section 3.3, the performance of the lexical analyser without word boundaries in input sentences is estimated for various input errors. In section 3.4, the performance of the lexical analyser is estimated for the input sentences with word boundaries. In section 3.5, the results of the above two studies are compared, and the effect of word boundaries on lexical analysis is discussed. In these studies it was assumed that all the word boundaries in the input are correct, which is not realistic. This constraint is removed in the next study, reported in section 3.6, in which the performance of the lexical analyser was estimated for varying number of word boundary errors. These results and their implications in speech recognition are discussed in section 3.7.

3.2 Lexical analyser

The lexical analyser matches the input sequence of phonemes representing a sentence against a prestored lexicon to produce alternate word strings matching the input. Since the input phoneme sequence may contain errors, approximate string matching [Hall and Dowling 1980] is used to produce the word strings. Hence one may obtain several alternate word strings matching the same input phoneme string but at different matching costs. The cost associated with a word sequence indicates the amount of its mismatch with the input phoneme sequence. Even for the same cost, one

usually obtains a large number of alternate word strings.

The lexical analyser used in our studies works as follows: It matches the input phoneme sequence starting from the leftmost unmatched phoneme against the lexicon and finds all matching words. Since the matching is approximarc, even for the same input phoneme sequence several words may be matched and the words matched may not exactly correspond to the input. **A** cost indicating the amount of mismatch between the input phoneme string and the word matched is associated with each of the hypothesised words. This cost is obtained from a prestored cost matrix. Thus several alternate word strings may be matched against the same input phoneme sequence with varying costs. For each of these alternate word strings, a record is created containing (i) the word sequence that matched the initial portion of the input sequence, (ii) a pointer to the input phoneme sequence indicating the next phoneme to be matched and (iii) the cost of the partial match. These records for the various alternate word strings are stored in the memory. From these, one record is selected for further matching. The record structure of an alternative and the lexical match algorithm are given below.

```
Alternative = RECORD
                matched_string:String;
                    {Contains the word sequence matching the
                      initial portion of the input)
                input_pointer:integer;
                    {Contains a pointer to the start of the
                      unmatched portion of the input)
                cost:integer;
                    {Contains the cost of matching the matched_string
                      with the initial portion of the input)
              END;
```

(i) Initially the entire sentence is unmatched. Create an alternative with a null string for matched_string, inputgointer set to one and zero cost.

(ii) select* an alternative from the memory. Set the unmatched string pointer to the inputgointer in the alternative.

* ***This selection can be done in many ways. A common choice is the least cost alternative. However, one may sometimes choose the highest cost alternative to reduce memory requirements.***

(iii) Starting from the phoneme pointed by the unmatched string pointer, match the input phoneme sequence with the 'lexicon until a word matching the input sequence is found. Form a new alternative by attaching the word matched to the matched_string, the input_pointer pointing to the position of the phoneme after the matched word and the cost set equal to the sum of the old cost and the cost of the current matching.

(iv) If the current alternative contains a word sequence matching the complete input sequence, output the word sequence, otherwise if the cost of the alternative is less than a threshold, store the alternative. Note that incomplete word sequences matching a sentence or alternatives whose costs exceed the threshold are discarded.

(v) Repeat (ii), (iii) and (iv) until no alternatives are left in the memory.

The lexicon used for matching was the Meenakshi Hindi-English dictionary[Mohan and Kapoor 1989]. It contained nearly 30,000 words. For verbs only the root form of the verb was given. To these dictionary words, the words taken from the sentences used in the studies were added. In total, the lexicon contained about 31,000 words. To facilitate the left-to-right matching strategy used, the dictionary was organised as a tree-structured dictionary [Wolf and Woods 1980; Hatazaki and Watanabe 1986].

A crucial component of the lexical analyser is the cost matrix used in estimating the cost of a mismatch. The validity of any simulation study on lexical analysis depends on how accurately the cost matrix reflects the speech signal-to-symbol conversion errors. The cost matrix used in the lexical analyser is shown in Fig.3.1. The costs were specified using three values, 'high(H)', 'mid(M)' and 'low(L)'. For example the figure shows that the phoneme /a/ has a low cost to /a:/ and a mid cost to /e:/, meaning that an acoustic-phonetic analyser would have misrecognised an uttered /a:/ as /a/ with a higher probability whereas it would have misrecognised an /e:/ as /a/ with a lower probability. These cost values were given based on the observations of the errors in the

a (a: L) (e: M) (o: M)#

a: (a L) (o: M) (e: M)#

i (i: L) (e: M) (u H) (u: H)#

i: (i L) (e: M) (u: H) (u H)#

u (u: L) (o: M) (i H) (i: H)#

u: (u L) (o: M) (i: H) (i H)#

e: (a M) (a: M) (i M) (i: M) (o: H)#

ai (e: L) (a: L) (a M) (i H) (o: H)#

o: (u: L) (u M) (a H)#

au (o: L) (a M) (u M)#

k (t L) (p M) (ṭ M) (k^h M) (t^h H) ($ṭ^h$ H)#

k^h (t^h L) (c L) (t M) (k M) ($ṭ^h$ H) (c^h H)#

g (b L) (d L) (ḍ L) (g^h M) (d^h H)#

g^h (d^h L) ($ḍ^h$ L) (b^h M) (g M)#

c (c^h L) (k^h L) (t^h L) (k M) (t H) (p H) #

c^h (c L) ($ṭ^h$ M) (t^h M) (k^h H)#

j (j^h L) (g M) (ḍ M) (d M) (b H)#

j^h (j L) (d^h M) (g H)#

ṭ (p L) (t L) (k L) ($ṭ^h$ M) (t^h M)#

$ṭ^h$ (t^h L) (c^h L) (t^h L) (p^h M) (k^h M)#

ḍ (g L) (d L) (b L) ($ḍ^h$ M) (j M) (d^h H)#

$ḍ^h$ (ḍ L) (d^h L) (b^h L) (g^h M)#

N (n L) (m M)#

t (k L) (p L) (ṭ M) (t^h M)#

t^h (k^h L) ($ṭ^h$ L) (p^h M) (t M) (p H) (ṭ H)#

d (g L) (ḍ L) (b L) (d^h M)#

d^h (g^h L) ($ḍ^h$ L) (b^h M) (d M)#

n (N L) (m M) (d H)#

p (t L) (k L) (ṭ M) (p^h M) (t^h H)#

p^h (f L) (p M) ($ṭ^h$ M) (t^h M)#

f (p^h L)#

b (d L) (g L) (ḍ L) (b^h M) (p H)#

b^h (d^h L) ($ḍ^h$ L) (g^h M) (b M)#

m (n L) (N M) (b H)#

y (v L) (l M) (r H)#

r (l L) (y M) (v M)#

l (r L) (y M) (v M)#

v (y L) (l M) (r H)#

s (z L) (S M) (c H)#

S (z L) (s M) (c H)#

z (s L) (S M)#

h (s L) (S L)#

Fig. 3.1 Cost matrix used in the lexical analysis.

speech signal-to-symbol conversion in the VOIS speech-to-text conversion system [Eswar 1990].

From the cost matrix the cost of substitution for phonemes was estimated in the following fashion: Let p be the observed phoneme and q be the matching phoneme. Then the cost of substituting p by q during the lexical analysis was given by

C(p,q) = 0, if p=q;

C(p,q) = 1, if the row for p in the cost matrix contains q with a 'low' cost;

C(p,q) = 2, if the cost matrix contains 'mid';

C(p,q) = 4, if the cost matrix contains 'high';

C(p,q) is infinite; if there is no entry for q in the row for p.

In Fig.3.2, a small sentence segment, *'me:ra: na:m ra:m* was matched against the lexicon and the various stages in matching are shown. As seen from the figure, initially two words *me:* and *me:ra:* were hypothesised as matching the initial portion of the input with zero cost and *me:l*, *me:la:* and *mai* with a cost of one. From these, the lowest cost alternative *me:ra:* was selected for further matching, producing the word strings *me:ra: na:*, *me:ra: na:m* with zero cost and *me:ra: na*, *me:ra: nam*, *me:ra: namr* and *me:ra: mu-* with a cost of one. These word strings were then stored back in the memory and match process was repeated. The figure shows all partial word sequences produced by the match whose costs were less than or equal to 1. It can be seen that many of the partial word sequences (for example, the word string *me:ra: na:*) were eliminated due to the threshold on the maximum cost (one substitution). Complete word strings matching with cost less than or equal to the maximum cost are shown highlighted in the figure along with their cost.

3 3 Lexical analysis without word boundaries

The first study on the lexical analyser is to estimate its performance when the input phoneme string contained no word boundaries. Two parameters were used to

me:
me:ra:
me:ra: na:
me:ra: na:m
me:ra: na:m ra:m 0.0
me: **ra:m**
me: **ra:m** a:
me: **ra:m** a:m
me: **ra:m a:mr**
me: **ra:m a:mr** a:
me: ra:m a:mr a:m 1.0
me: ra:m a:m ra:m 1.0
me: **ra:Na:**
me: ran
me: ran a:
me: ran a:m
me: ran **a:mr**
me: ran **a:mr** a:
me: ran a:mr a:m 1.0
me: ran a:m ra:m 1.0
me: **rEn**
me: **rEn** a:
me: **rEn** a:m
me: **rEn a:mr**
me: **rEn a:mr** a:
me: rEn a:mr a:m 1.0
me: rEn a:m ra:m 1.0
me: la:
me: la:n
me: **la:na:**
me: **la:n** a:
me: **la:n** a:m
me: **la:n a:mr**
me: **la:n a:mr** a:
me: la:n a:mr a:m 1.0
me: la:n a:m ra:m 1.0
me: la: na:
me: la: **na:m**
me: la: na:m ra:m 1.0
me:ra: na:m la:
me:ra: na:m la:m 1.0
me:ra: na
me:ra: nam
me:ra: namr
me:ra: namr a:
me:ra: namr a:m 1.0
me:ra: nam ra:m 1.0
me:ra: ma:
me:l
me:la:
me:la: na:
me:la: na:m
me:la: na:m ra:m 1.0
me:l a:
me:l a:n
me:l a:na:
me:l a:n a:
me:l a:n a:m
me:l a:n a:mr
me:l a:n **a:mr** a:
me:l a:n a:mr a:m 1.0
me:l a:n a:m ra:m 1.0
me:l a: na:
me:l a: **na:m**
me:l a: na:m ra:m 1.0
mai

Fig. 3.2 A sample output of the lexical analyser. The figure shows all the word strings which match some initial portion of the input phoneme string **'me:ra: na:m ra:m'**. Complete word strings with match cost less than or equal to 1 are shown highlighted in the figure.

express the performance, (i) the number of alternate word sequences produced for a given input phonemé sequence, and (ii) the time spent for matching. The study was done in two parts. The first part is for the case when the input phoneme sequence contained no errors. Hence only exact matching was used in the dictionary match. The second part of the study is on the performance of the lexical match when the input was assumed to contain errors likely in speech signal-to-symbol conversion. Hence approximate string matching was used in the lexical match.

3.3.1 Results of lexical analysis with exact matching

The lexical analyser program was run with an input text containing 100 sentences. The sentences were of varying lengths and on average contained 12 to 13 words. The results of the lexical match were ordered as per the number of alternative word strings and are shown in Table 3.1.

From the results it can be observed that a large number of sentences (64 out of 100) had less than 10 alternate word sequences. Only a small fraction (3 out of 100) had 1000 or more alternate word strings. The average number of alternate word strings for a sentence was about 120. Three sentences had only a single word string matching them. The highest number of alternate word strings matching any sentence were 2448.

These results were also used to study the effect of the length of the sentence (both in terms of number of words and number of phonemes) on the number of word sequences matching the sentence. Table 3.2(a) and Table 3.2(b) show the results of this study. It can be observed that in general longer sentences have more alternate word strings matching them. But length alone does not determine the number of alternatives since sentences of same length still had widely different number of matching word sequences. For example, in our data three sentences contained 24 words but the number of word sequences for them were 24,288 and 2488.

No. of alternatives	0-10	10-100	$100-10^3$	$>10^3$
No. of sentences	64	21	12	3

Table-3.1 Distribution of input sentences in terms of the number of alternate word strings matched. The results are obtained by performing exact matching of lexicon with input sentences. All the word boundaries are removed from the input sentences.

Length in words	No. of Sentences	Average no. of Alternatives
1-5	2	6
5-10	39	7
10-15	34	58
15-20	13	122
20-25	11	311
26-30	1	1536

(a)

Length in phonemes	No. of Sentences	Average no.: of Alternatives
16-30	21	5
31-45	32	6
46-60	22	60
61-75	12	138
76-90	12	380
> 90	1	850

(b)

Table_3.2 Variation in the Number of matching word sequences with sentence length. In 3.2(a) Sentence length is given in words and in 3.2(b) it is in phonemes.

3.3.2 Results of lexical analysis with approximate matching

The results of lexical analysis when exact matching was used, show that on the average 120 word sequences match a given sentence. However this number will increase if one allows errors in the input sentence as in a phoneme sequence produced by a speech signal-to-symbol conversion stage. The errors can be: (i) substitution errors, where an input phoneme was misrecognised as another phoneme, (ii) insertion errors, where a new phoneme was inserted in input sequence, and (iii) deletion errors, where a phoneme was missed out during recognition. In this study, only substitution errors are considered as these are more frequent and can be characterised easily. Such errors are handled in the lexical analysis by using approximate matching instead of exact matching with the lexicon. In the approximate matching, all word strings that match the input phoneme string within a specified mismatch cost are generated. If the mismatch cost is larger than the total errors in the input sentence, then the uttered sentence will also be generated by the lexical analyser. However, many other word strings may also be produced in this process.

The lexical analyser program was run again with an input text containing 10 sentences. The sentences were of varying lengths and contained between 25 to 45 phonemes. All word boundaries were removed from them. The lexical matching algorithm described earlier was used to match these sentences against the lexicon. The maximum mismatch between the input and the hypothesised word string was varied from 0 to 5. The results, in terms of the number of alternate word strings, are shown in Table 3.3.

Since a sentence on the average contained 35 phonemes, a variation in mismatch from 0 to 5 represented a variation in the input error from 0 to 12%. However in a speech recognition system, errors in the speech signal-to-symbol conversion may sometimes be as large as 50%. But the error variation in our studies

Match cost	Sentence number 1	2	3	4	5	6	7	8	9	10
0	16	4	8	6	6	8	6	7	16	24
1	496	66	108	116	910	254	77	235	232	726
2	8164	532	1004	1254	838	4088	698	4146	2112	11440
3	92402	3051	7466	9991	6025	45736	4999	51092	15666	126627
4	797820	14242	47570	65127	36837	404437	30246	491676	100389	1201124
5	_	57218	269140	365111	198673	3006759	159956	_	574250	_

Table-3.3 The no. of alternate word strings matching a sentence without word boundaries when approximate matching is used. Results are shown for 10 sentences at varying matching costs.

had to be restricted to the above range as the lexical analyser took several hours of computation, even for this small error.

The results show that the number of alternate word strings for an input sentence increase rapidly with increase in the mismatch. The number of alternate word strings was plotted against the mismatch cost for one sentence in Fig.3.3. It can be seen that the growth is nearly exponential: for example, an increase of one in the mismatch cost, the number of alternate word strings increased by a factor of 6.

The performance of the lexical analyser was also studied in terms of the time spent on the lexical analysis. These results are shown in Table_3.4. It can be seen that the time spent on lexical analysis also increases rapidly with increase in mismatch. A plot of the time against the mismatch cost for one sentence is shown in Fig.3.4. It can be seen that the time spent on lexical analysis also increases exponentially with increasing mismatch.

The above results show clearly that both the number of alternate word strings matching a sentence and the time spent on matching increase rapidly with increasing mismatch cost used for matching.

3.4 Lexical analysis with word boundaries

To estimate the improvement in the performance of the lexical analyser when the word boundaries are known in the input, the lexical matching algorithm was again applied on the sentences with all the word boundaries present. Since word boundaries are known, the lexical match was performed for each word separately to obtain alternate word strings for each word. The alternatives for the sentence were obtained by combining these word strings. The results of the lexical match in terms of the number of alternate word strings for the sentences are shown in Table_3.5.

The results show that the number of alternate word strings for a sentence still

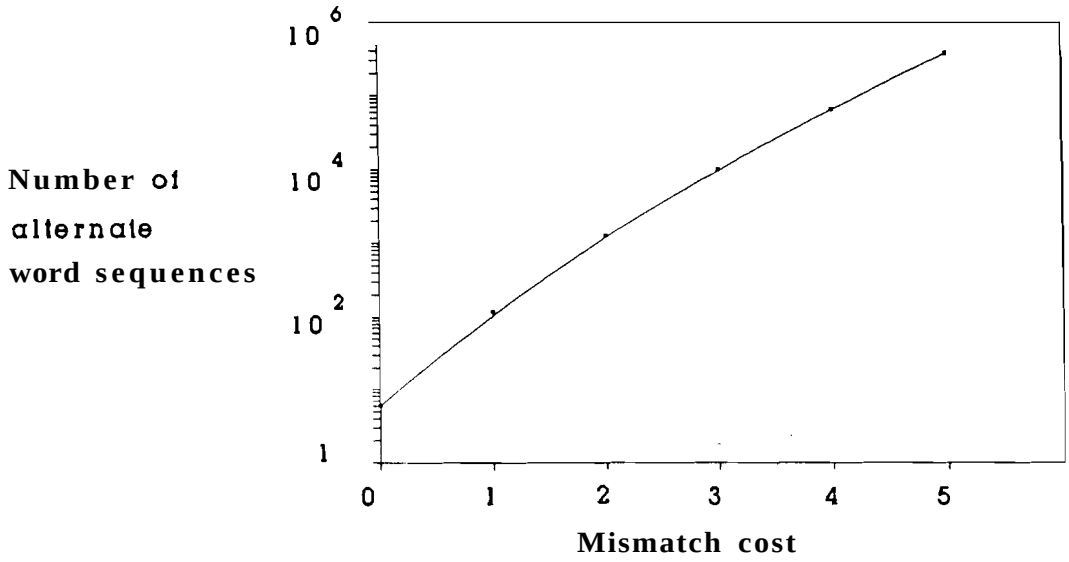


Fig.3.3 Results of lexical analysis on a sentence without word boundaries. The figure shows the number of alternate word sequences matching the sentence at various mismatch costs. It can be seen that the number grows exponentially(appears linear in log scale) with respect to the mismatch cost.

Match cost	Sentence number 1	2	3	4	5	6	7	8	9	10
0	1	0	1	0	0	0	0	0	1	2
1	18	3	7	4	3	5	3	11	13	47
2	238	20	62	36	27	64	16	149	108	674
3	2199	97	408	245	172	602	85	1503	712	6988
4	15936	397	2357	1532	996	4757	420	12261	3980	56712
5	_	1401	11918	7444	4902	31851	1868	_	19826	_

Table-3.4 The time spent(in seconds) on lexical analysis for a sentence without word boundaries. Results are shown for 10 sentences at varying matching costs.

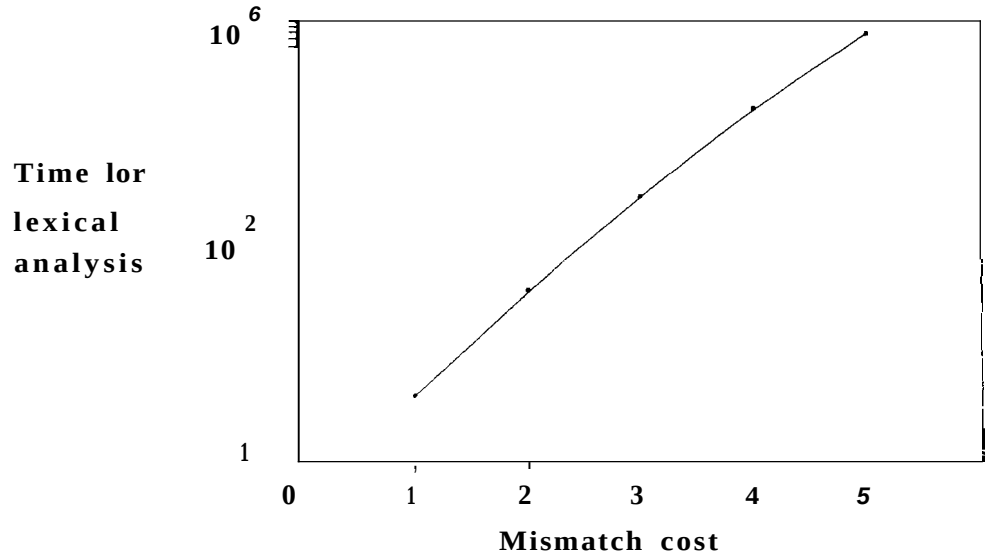


Fig. 3.4 Results of lexical analysis on a sentence without word boundaries. The figure shows the time taken (in sec.) for performing the lexical analysis at various mismatch costs. It can be seen that the lexical analysis time grows exponentially(appears linear in log scale) with respect to the mismatch cost.

Match cost	Sentence number 1	2	3	4	5	6	7	8	9	10
0	8	2	8	4	6	8	6	6	16	24
1	260	52	108	70	89	210	77	186	208	652
2	4236	376	958	688	790	2807	686	3020	1588	8988
3	46518	1924	6665	5035	5366	26457	4755	34221	10196	86278
4	385640	8012	39092	30293	31072	199992	27509	302713	57342	658932
5	2562330	28775	200327	157550	150542	1286321	136741	2217843	285020	4283796

Table_3.5 **The no. of alternate word strings matching a sentence with word boundaries when approximate matching is used. Results are shown for 10 sentences at varying matching costs.**

increase with increasing mismatch. A plot of the number of alternate word strings against the mismatch for a sentence is shown in Fig.3.5. It can be seen that the growth rate in the number of alternate word strings is still exponential.

The performance of the lexical analyser was also measured in terms of the time spent on the lexical analysis. These results are shown in Table 3.6. It can be seen that the time spent on lexical analysis is small when the input sentence contained word boundaries. A plot of the time against the mismatch is shown in Fig.3.6. It can be seen that the increase in the time spent is also small.

3.5 Comparison of the lexical analysis results

To estimate the effect of word boundaries on the performance of the lexical analyser, the results of the above two studies were compared. From the results of the previous sections, it can be seen that the number of word sequences matching an input sentence decreased when word boundaries were present in the input. This is illustrated in Fig.3.7, where the ratio between the number of alternate word strings for the two cases: (i) when the input sentences had no word boundaries, and (ii) when the input sentences had all the word boundaries, is plotted against the mismatch cost. It can be seen that the ratio, which represents the factor of reduction in the number of alternate word boundary sequences, increases with increasing mismatch. However, it can also be seen that the reduction is quite small (about 2.3 at a mismatch of 5). Moreover, the growth in the ratio is also small, indicating that for small errors, one cannot expect significant reduction in the number of alternate word strings produced by the lexical analyser due to word boundaries.

On the other hand, a comparison of the times spent on lexical analysis shows a significant improvement when the word boundaries were present in the input. To illustrate this, the ratio of the lexical analysis times for a sentence for the cases: (i) when the input sentences had no word boundaries, and (ii) when the input sentences

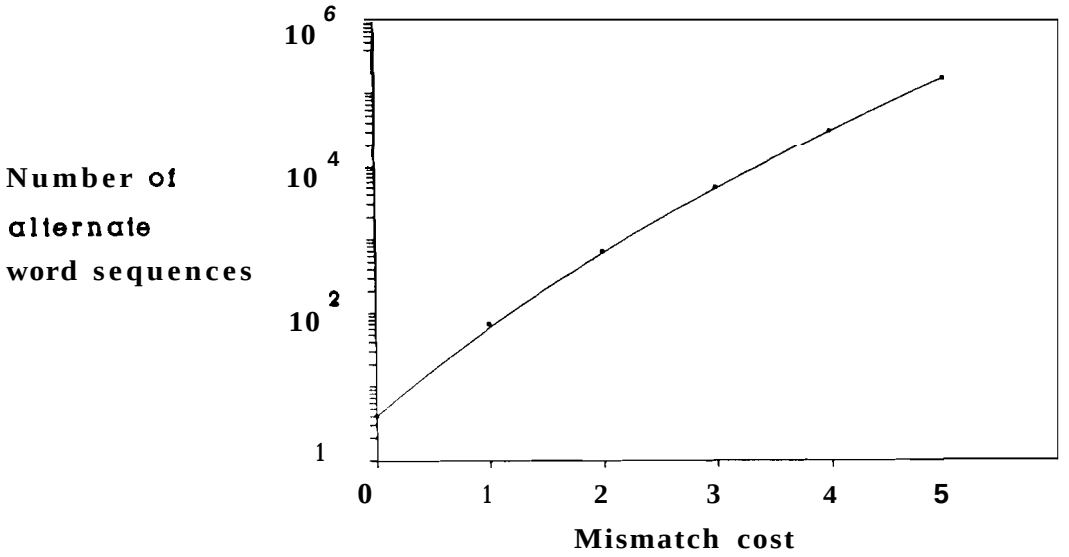


Fig.3.5 Results of lexical analysis on a sentence with word boundaries. The figure shows the number of alternate word sequences matching the sentence at various mismatch costs. It can be seen that the number grows exponentially(appears linear in log scale) with respect to the mismatch cost.

Match cost	Sentence number									
	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	1	0	1
2	1	0	1	1	1	2	1	2	1	2
3	2	1	2	2	2	5	2	4	3	4
4	3	1	3	4	3	11	3	7	6	9
5	4	1	5	8	5	22	4	11	9	15

Table 3.6 The time spent(in seconds) on lexical analysis for a-sentence with word boundaries. Results are shown for 10 sentences at varying matching costs.

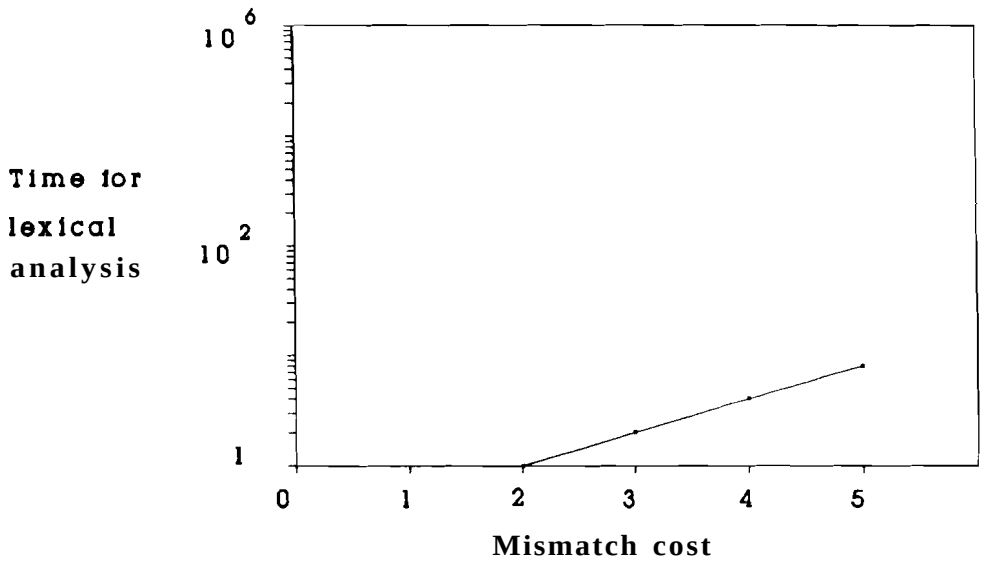


Fig.3.6 Results of lexical analysis on a sentence with word boundaries. The figure shows the time taken (in sec.) for performing the lexical analysis at various mismatch costs. It can be seen that the lexical analysis time grows exponentially (appears linear in log scale) with respect to the mismatch cost, though it grows much more slowly compared to that of the sentence when word boundaries are removed.

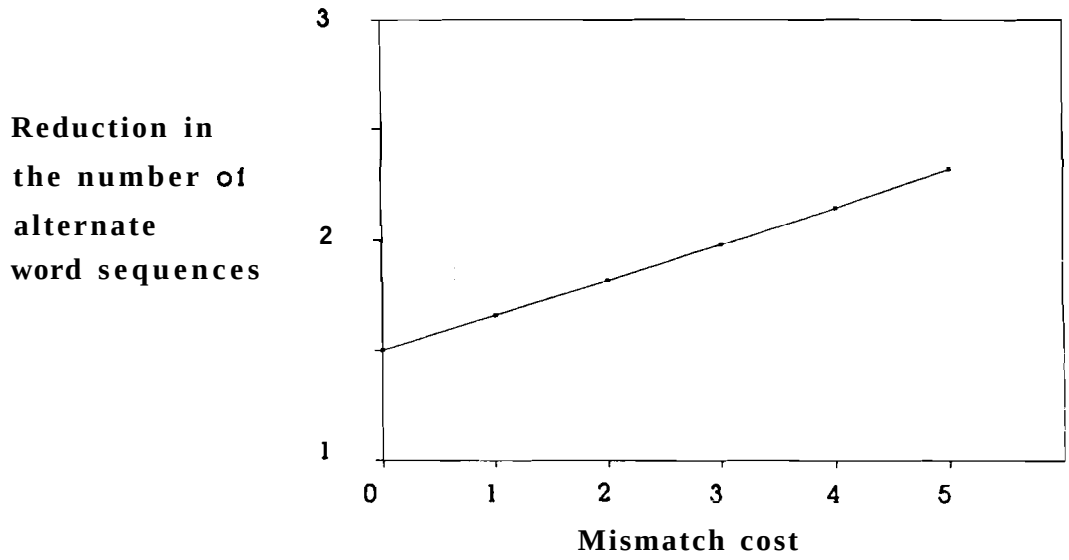


Fig.3.7 A comparison of the number of alternate word sequences produced for a sentence for the two cases, (i) sentence without word boundaries and (ii) sentence with word boundaries. In the figure the ratio of the number of alternate word sequences for a sentence without word boundaries to the number of alternate word sequences for the sentence with all word boundaries is plotted against the maximum mismatch cost between the sentence and an alternative. It can be seen that the ratio grows linearly with respect to the mismatch cost indicating that the presence of word boundaries in a sentence reduces the number of alternate word sequences, though the reduction may not be significant at small mismatches.

had all word boundaries, is plotted against the mismatch cost in Fig.3.8. It can be seen that the lexical analysis time is significantly reduced by the presence of the word boundaries in the input. Moreover, the reduction in the time is nearly exponential (appears linear in the log scale) indicating that the presence of word boundaries in the input greatly reduces the time spent on lexical analysis.

3.6 Lexical analysis with partial knowledge of word boundaries

The results reported in the previous sections established the necessity for word boundary hypothesisation for speech-to-text conversion. However, in these studies (section 3.4), it was assumed that all word boundaries in the input sentences were known. However, this is not a realistic assumption since any word boundary hypothesisation technique will miss a few word boundaries and also produce a few incorrect word boundary hypotheses. Hence one needs to study the performance of a lexical analyser when the input sentences contained varying number of word boundaries and also contained a few incorrect word boundaries.

Two studies were conducted to estimate the performance of the lexical analyser. In the first study, it was assumed that only a fraction of the total number of word boundaries in the input were present. However it was also assumed that all the word boundary hypotheses were correct. In the second study, even this constraint was removed and the performance of the lexical analyser was studied for varying percentages of correct and incorrect word boundary hypotheses.

The results of the first study are shown in Table 3.7. The results show the performance of the lexical analyser, in terms of the time spent on lexical analysis at various percentages of word boundaries in the input. It can be seen that increasing the number of word boundaries in input decreases the lexical analysis time. It can also be seen that even if a small fraction of the word boundaries are known, the lexical analysis

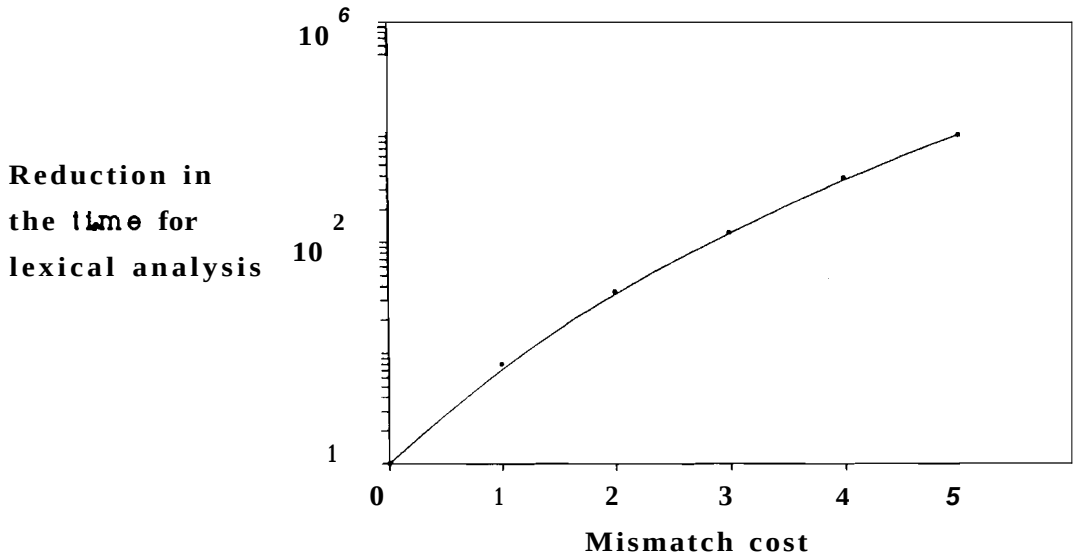


Fig.3.8 A comparison of the lexical analysis times for a sentence for the two cases, (i) sentence without word boundaries and (ii) sentence with word boundaries. In the figure the ratio of the lexical analysis time for a sentence without word boundaries to the lexical analysis time for the sentence with all word boundaries is plotted against the mismatch cost between the sentence and an alternative. It can be seen that the ratio grows exponentially (appears linear in log scale) with respect to the mismatch cost indicating that the presence of word boundaries in a sentence significantly reduces the time taken for lexical analysis, and the reduction increases rapidly with increasing mismatch.

Watch cost	Sentence number 1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	1	0	1	1	2
2	6	1	2	2	1	3	1	5	5	11
3	22	2	5	5	4	10	3	19	15	52
4	63	4	10	12	8	28	6	51	42	198
5	151	6	19	27	16	71	10	120	101	651

(a)

Match cost	Sentence number 1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0
1	3	1	1	1	1	2	1	2	2	5
2	24	3	8	6	4	13	5	10	14	40
3	137	9	32	26	18	65	16	52	68	245
4	640	23	109	99	63	290	49	217	283	1263
5	2406	56	315	342	199	1116	133	774	1041	5581

(b)

Table 3.7 The time spent(in seconds) on lexical analysis for a-sentence with (a) 50%, and (b) 25% word boundaries. Results are shown for 10 sentences at varying matching costs.

time reduces considerably from that of knowing no word boundaries. This is also illustrated in Fig.3.9, where the lexical analysis times for the two cases: (i) when the input sentence had no word boundaries and (ii) when the input sentence had some word boundaries, was plotted for a sentence against the mismatch cost. The plot shows the times for the three cases of all word boundaries known (100%), 50% of word boundaries known and 25% word boundaries known.

To study the effects of incorrect word boundary hypotheses on lexical analysis, the lexical analyser algorithm was modified. The modification was based on the following argument. Assume that a sentence has N word boundaries hypothesised in it of which M are incorrect. Since the lexical analyser has no knowledge of the incorrect hypotheses, it will have to try out all possible combinations of (N-M) word boundary hypotheses to ensure that the sentence with the correct word boundaries is also analysed. For example, consider a sentence w1#w2#w3#w4, which had four words w1, w2, w3, w4 separated by word boundaries (indicated by #). Now assume that a word boundary hypothesiser has produced a string w1#w21@w22w3#w4 with three word boundary hypotheses, where # indicates a correct word boundary and @ indicates an incorrect word boundary (note the division of w2 into w12 and w22). A lexical analyser can now produce the correct sentence w1#w2#w3#w4, only if its input is given as w1#w2w3#w4, or the incorrect word boundary was removed in the input. However, there is no way of identifying which of the three word boundary hypotheses is incorrect. Hence the only way to ensure that the lexical analyser receives the input w1#w2w3#w4 is to try out all possible input strings in which one word boundary is removed (in this example they correspond to the strings w1#w2w3#w4, w1w12@w22w3#w4, and w1#w12@w22w3w4). In general, if N word boundary hypotheses are produced of which M are incorrect, lexical analysis is to be performed on ${}^{N}C_{N-M}$ possible word strings, each containing (N-M) word boundary hypotheses.

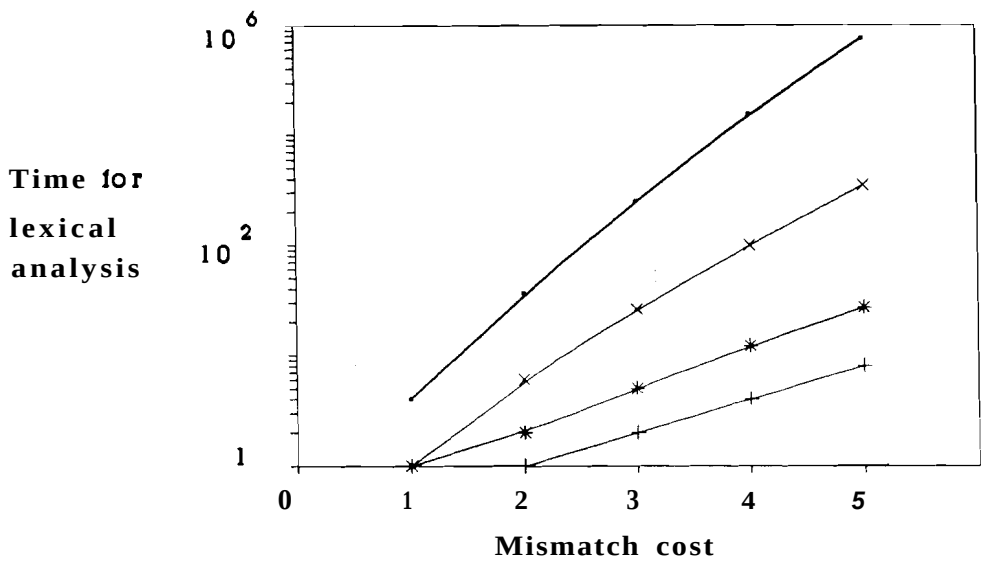


Fig.3.9 A comparison of the lexical analysis times for a sentence for varying percentages of word boundaries. In the figure, the lexical analysis times (in sec.) for a sentence are plotted against the mismatch cost, for four cases, (i) the sentence does not contain any word boundaries (indicated by . , (ii) the sentence contains 25% of total word boundaries (indicated by X), (iii) the sentence contains 50% of the total word boundaries (indicated by *), and (iv) the sentence contains all word boundaries (indicated by +). It can be seen that the lexical analysis times decrease with increasing percentage of word boundaries in the sentence, and, more importantly, the growth rate in the lexical analysis time also decreases with increasing word boundaries.

The above modified lexical analyser algorithm was applied on the 10 input sentences used in the earlier studies. In the studies, two cases of 25% and 50% errors in the word boundary hypotheses were used. These incorrect word boundary hypotheses were randomly placed in the input sentences. The results of the studies are presented in Table 3.8. The results are shown in terms of the lexical analysis time for four cases: (i) the word boundary hypotheses contain 25% incorrect hypotheses, and the correct hypotheses correspond to all word boundaries in the input, (ii) the word boundary hypotheses contain 50% incorrect hypotheses, and the correct hypotheses correspond to all word boundaries in the input, (iii) the word boundary hypotheses contain 25% incorrect hypotheses, and the correct hypotheses correspond to 50% of the word boundaries in the input, and (iv) the word boundary hypotheses contain 50% incorrect hypotheses, and the correct hypotheses correspond to 50% of the word boundaries in the input. It can be seen that even when 50% of the word boundary hypotheses are incorrect, the time spent on lexical analysis for a sentence with word boundaries is smaller than that of a sentence with no word boundaries hypothesised. This is illustrated in Fig.3.10, where the lexical analysis times for the above four cases were plotted against the mismatch cost. The plot also contains the lexical analysis time when no word boundaries are present. It can be seen that the time for lexical analysis when the input sentences contain word boundaries is significantly less than that for the case when the input has no word boundaries. Moreover the lexical analysis time grows more rapidly when the input has no word boundaries. Thus these results clearly show that the presence of word boundaries in the input, even if there are as many as 50% incorrect hypotheses, can significantly reduce the lexical analysis time.

3.7 Summary and Conclusions

In this chapter, we have described our studies on the effect of the word

Match cost	Sentence number 1	2	3	4	5	6	7	8	9	10
0	1	0	0	0	0	0	0	2	0	1
1	6.3	1.7	3.4	2.7	3.9	3.8	2.4	21	3.9	9.8
2	33	4.1	13	8.5	16	15	8.9	169	14	51
3	143	9.4	39	24	55	47	27	1062	42	223
4	508	20	105	60	179	119	69	5362	124	832
5	1432	37	254	145	530	263	160	22343	280	2706

(a)

Match cost	Sentence number 1	2	3	4	5	6	7	8	9	10
0	2	1	2	2	2	3	1	4	1	3
1	26	17	24	15	17	22	8.2	77	19	49
2	172	63	130	67	83	122	32	659	88	308
3	842	210	570	268	348	568	111	4256	360	1546
4	3355	643	2128	987	1347	2383	347	18781	1361	6318
5	12832	1807	7357	3023	5350	8972	977	_	3970	24071

(b)

Table 3.8 The lexical analysis times (in seconds) for 10 sentences with (a) 25%, and (b) 50% incorrect word boundary hypotheses. The correct hypotheses contain all word boundaries. The table is continued in the next page.

Hatch cost	Sentence number 1	2	3	4	5	6	7	8	9	10
0	1	0	0	0	0	0	0	1	0	1
1	4.2	1.7	2.2	2.8	1.6	3.1	1.2	8.9	2.9	5.1
2	27	5.3	8.9	13	6.2	15	4.5	76	11	22
3	140	15	31	58	24	60	15	500	36	81
4	594	41	87	276	74	189	50	2520	95	247
5	2014	95	214	917	223	538	132	10661	238	666

(c)

Match cost	Sentence number 1	2	3	4	5	6	7	8	9	10
0	1	0	0	0	0	0	0	2	0	1
1	21	6.6	25	14	5.9	14	5.4	20	15	48
2	166	24	164	69	30	88	24	154	74	398
3	960	75	838	293	135	449	95	905	287	2762
4	4428	214	3660	1109	668	2269	344	5062	1361	6318
5	18793	612	15220	3976	3589	10842	898	21240	4201	-

(d)

Table-3.8 The lexical analysis times (in seconds) for 10 sentences with (c) 25%, and (d) 50% incorrect word boundary hypotheses. The correct hypotheses contain 50% of total word boundaries.

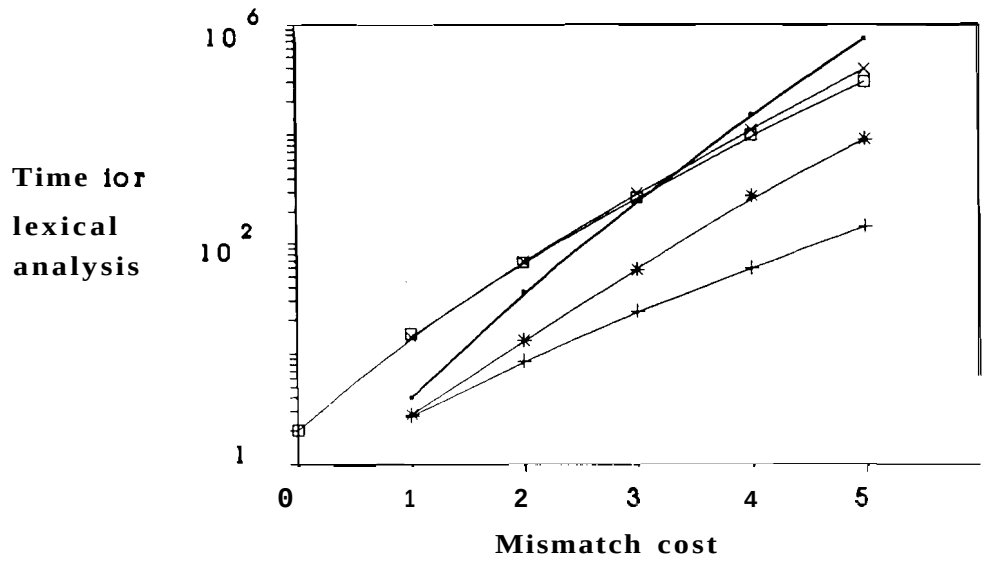


Fig.3.10 A comparison of the lexical analysis times for a sentence for varying percentages of word boundaries and varying percentage of incorrectly placed word boundaries. In the figure, the lexical analysis times (in sec.) for a sentence are plotted against the mismatch cost, for four cases, (i) the sentence contains 100% word boundaries and 25% incorrectly placed word boundaries(indicated by +), (ii) the sentence contains 100% word boundaries and 50% incorrectly placed word boundaries(indicated by *), (iii) the sentence contains 50% word boundaries and 25% incorrectly placed word boundaries(indicated by □), and, (iv) the sentence contains 50% word boundaries and 50% incorrectly placed word boundaries(indicated by X). The lexical analysis time for the sentence without any word boundaries is also shown in the figure(in bold). It can be seen that the presence of incorrectly placed word boundaries increases the lexical analysis time. It can also be seen that even when 50% of the word boundaries are incorrectly placed, the lexical analysis time is less than the lexical analysis time for a sentence without any word boundaries.

boundary information on the performance of a lexical analyser. In these studies the performance is estimated in terms of: (i) the number of alternate word sequences produced, and (ii) the time spent on lexical analysis. The results clearly establish the following:

1. Knowledge of the word boundaries can reduce the number of matching word sequences for an utterance.
2. Time spent on the lexical analysis depends critically on the word boundary information.
3. Incorrect placement of word boundaries can increase the lexical analysis time, but even when the 50% of the word boundaries placed are incorrect, there is a significant saving in the lexical analysis time compared to the case when the input has no word boundaries.

The results do not show as much an improvement in the number of word sequences as in the time spent on lexical analysis. But in an earlier study for English [Briscoe 1989], reported in section 2.2 of this thesis, it was found that the improvement in the number of alternatives is also large, which seems to be at variance with our results. The reason could be the low error rate used in our studies. In Briscoe's study, the lexical matching was done at broadclass level, which corresponds to a very large error rate in the input. Thus, it is possible that if our studies were done for large error rates, say 50% errors, they also may yield results comparable to the ones in Briscoe's study. However, at low error rates, the main advantage of word boundary hypothesisation seems to be in improving the speed of the lexical analyser.

These results show that the main role of the word boundary hypothesisation in speech recognition is in reducing the time for lexical analysis. As mentioned earlier in section 1.1, the time spent on lexical analysis forms a major part of the total time spent on speech recognition. Thus word boundary hypothesisation can greatly speed up the

speech recognition process. On the other hand, the number of alternate word strings output by the lexical analyser form the input to the higher level analysers such as syntax and semantic analysers. Since word boundary hypothesisation reduces the number of alternate word strings by a much smaller factor compared to the reduction in the lexical analysis time, at least at low input errors, the computational load of the higher level analysers may not be significantly reduced by the word boundary hypothesisation.

Chapter 4

WORD BOUNDARY CLUES BASED ON THE LANGUAGE KNOWLEDGE

4.1 Introduction

Most of the earlier techniques for detecting word boundaries in continuous speech were based on clues using the acoustic-phonetic, lexical and prosodic knowledge sources. Language knowledge, such as the syntactic and semantic knowledge was not explored for word boundary hypothesisation, though, as mentioned in chapter 2, it was used in an indirect way along with the other knowledge sources. The work reported in this chapter concentrates on identifying language clues useful for word boundary hypothesisation, and applying them in the context of speech recognition for Hindi. By language clues we refer to the various higher level linguistic features such as the syntactic and the semantic features of the language.

The chapter is organised as follows: In the next section (section 4.2), the language clues proposed for hypothesising word boundaries are described. In section 4.3, some issues relating to the application of these clues in a speech-to-text conversion system are discussed. The issues considered related to the type of the input on which the clues are to be applied and the measures used to estimate the performance of the clues. The results of word boundary hypothesisation using the language clues are presented in section 4.4. To reduce the incorrect hypotheses generated by the language clues, a few lexical constraints were used to verify the word boundary hypotheses. The results of word boundary hypothesisation using language clues and the lexical constraints are described in section 4.5. Finally, in section 4.6, a summary of these studies is given and their implications in the context of speech recognition are discussed.

4 3 Language clues for word boundary hypothesisation

In this section some clues for identifying word boundaries are described. The

proposed clues are based on the observation that in any language some words occur more frequently than others. Usually these words correspond to the function words of the language. If one can spot these frequently occurring words in a text (or, in the symbol string generated by the speech signal-to-symbol converter), then one can detect many word boundaries. Thus it is proposed to spot the symbol strings corresponding to the frequently occurring words and hypothesise them to be the words themselves. However, the symbol strings spotted may not always correspond to the words but may be part of some other word or words. If the symbol strings occur more frequently as words than as substrings of other words, then one can hypothesise word boundaries around the spotted strings with a large confidence. Thus the proposed language clues are nothing but the symbol strings corresponding to the frequently occurring words. Hence word boundary hypothesisation using the language clues is equivalent to the spotting of frequently occurring symbol strings in the input text . In practice, one may also use other frequently occurring symbol strings which may not necessarily be complete words (for example, verb endings). The algorithm for word boundary hypothesisation using language clues {Ramana Rao 1989; Ramana Rao and Yegnanarayana 1991] is given below.

Algorithm 4.1

1. Read the input text until a symbol string corresponding to one of the language clues is found, and,

2. Hypothesise word boundaries appropriately for that clue.

3. Repeat 1 and 2, till the end of input.

Two criteria were primarily considered in selecting the language clues: (1) they

should occur frequently, and (2) they should be quite general, i.e., they should occur in all types of texts. Under these criteria, case markers and certain other function word classes like pronouns and conjunctions qualify as language clues. In addition certain verb endings and a few frequently occurring auxiliary verbs can also be considereci as language clues. These clues are small in number, and they also have important syntactic and semantic functions. The case markers function as markers of noun phrases, indicating their role in the sentence. They occur roughly in proportion to the noun phrases and hence are quite frequent. Similarly, verb endings and conjunctions serve as syntactic markers and they also occur frequently and in all types of texts. Some of the Hindi language clues are given in Fig.4.1.

One problem with the language clues, 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* as a clue, then every occurrence of *unhe:* in the input text, results in the hypothesisation of an incorrect word boundary between *un* and he:. To eliminate such errors, all morphological variants of the language clues must also be included in the clues. Hence, in this study, all morphological variants of the pronouns were included in the language clues. This resulted in a large number of language clues numbering 124, of which the pronouns and their morphological variants numbered 77.

Another problem noticed with some of the language clues is that they also occur as substrings of other language clues. For example, the case marker *ne:* occurs as the suffix of many verbs in their verbal noun form (for example, *karne:*, *rahne:* etc.). If one hypothesises word boundaries on both sides of *ne:*, several errors will occur, corresponding to the cases where *ne:* is part of a verb such as *karne:*. In this study, these errors were taken care of by hypothesising only the boundary occurring after ne:. Problems with clues which are prefixes or substrings of other clues were also accounted

Case Markers:

ka:, ki:, ke:, ko:, ne:, me:n, se:, par

Pronouns:

main, ham, tu:, tum, a:p, vah, yah, ve:, ye:

Conjunctions:

aur, ki, le:kin, parantu:

Verb endings:

ne:, na:, ta:, te:

Fig.4.1 Some of the language clues used for word boundary hypothesisation.

for in a similar fashion.

4.3 Issues in the application of language clues

There are two important issues relating to the application of language clues. One is the input on which these clues are to be applied, and the second is the estimation of the performance of the clues. In practice, language clues are to be applied on the symbol string output by a speech signal-to-symbol converter. In absence of a speech signal-to-symbol converter, one needs to examine the performance of the clues using texts which contain errors similar to those that occur in the speech signal-to-symbol conversion. To do this, one needs to first identify the errors likely in the signal-to-symbol conversion and then simulate them in a text. Depending on how well the errors are identified, the performance estimates obtained will reflect the actual performance of the clues in a speech-to-text conversion system. The simulation of errors used in the studies is described in section 4.3.1.

The second issue relates to the estimation of the performance of the language clues in hypothesising word boundaries. In practice, one can express the performance in terms of the number of word boundaries correctly detected and the number of incorrect hypotheses generated. However, one would like to express the performance in terms of measures which reflect the utility of the clue in hypothesising word boundaries better. In this regard, three measures were used in our studies. These are described in section 4.3.2.

4.3.1 Input for the studies on the performance of language clues

As mentioned above, the input used in the studies on estimating the performance of the language clues in word boundary hypothesisation is a Hindi text in which speech-like errors are simulated. The text used in the studies contained 800 sentences with nearly 11,000 word boundaries. The text is represented in phonemic

form.

The above Hindi text was corrupted by introducing errors which are likely to occur during the speech signal-to-symbol conversion. These errors can be of three types: (i) substitution errors which are due to the signal-to-symbol converter hypothesising a different phoneme in place of the uttered one (for example, an uttered k may be misrecognised as a *t),* (ii) deletion errors which are due to the signal-to-symbol converter missing out some phonemes (for example, an uttered *r* may be missed out due to its short duration, especially if it occurs along with a vowel), and (iii) insertion errors which are due to the signal-to-symbol converter hypothesising more than one phoneme for a single phoneme (for example, an a may be misrecognised as the phoneme sequence a followed by i). Of these three types of errors, the insertion and deletion errors are difficult to characterise and hence in these studies only a few common ones were simulated.

The substitution errors are caused by the similarities between the phonemes which cause a speech signal-to-symbol converter to confuse between them. The common substitution errors were represented by a similarity matrix which listed out the various alternatives that may be hypothesised during the signal-to-symbol conversion for each phoneme along with the probability of such substitution. The alternatives for each phoneme and the corresponding probability values were obtained after studying a large number of utterances with the help of a linguist. The similarity matrix used in the studies is shown in Fig.4.2. In the figure, the exact numerical values for probabilities were not given, instead the similarity between the phonemes was specified using three values 'High'(H), 'Medium'(M), and 'Low'(L). These values are relative to a phoneme and specify only the relative occurrences of various alternatives to that 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

ऄ	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)
ऽ	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)
aऽ	o:(H), a(M), u(M)
k	t(H), ड(M), ṭ(M), k^h(ॅ), 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), ड(M), b(L)
j^h	j(H), d^h(M), g(L)
ṭ	p(H), t(H), ×(H), $ṭ^h$(M), t^h(M)
ḍ	g(H), ड(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	◌(H), ड(M)
t	×(H), ड(H), ṭ(M), t^h(M)
t^h	$×^h$(×), $ṭ^h$(×), t(M), p^h(M), p(L), ṭ(L)
ड	g(H), ḍ(H), b(H), d^h(M)
ड^h	$ḍ^h$(H), g^h(H), b^h(M), d(M)
n	N(H), m(M), d(L)
ड	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)
ड^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(×), s(M), c(L)
s	ṣ(×), ś(M), c(L)
ṣ	ś(×), s(M)
h	ś(H), s(H)

Fig.4.2 Similarity matrix used to simulate substitution errors in a Hindi text

that if there are some substitution errors for i, most of them will be substitutions by *i:*, a few by *e:* and only occasionally by *u* and *u:*. Thus the equivalent probability values for these similarities vary depending or, the phoneme and the number of alternatives. To simplify the implementation, the number of alternatives for a phoneme was limited to six.

As mentioned earlier, a few of the common insertion and deletion errors were also simulated. These errors were derived from our experience with the development of the speech signal-to-symbol converter module of the VOIS speech recognition system [Yegnanarayana etal. 1989; Chandra Sekhar etal. 1990; Eswar 1990]. These errors described as rules are shown in Fig.4.3.

Using the similarity matrix and the insertion and deletion rules, a program was developed which produced an incorrect text from a correct text input for a specified average error. This average error represented the probability of substitution for any given phoneme. However, in practice, the probability of substitution will not be 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 are more prone to errors than vowels'. Hence for a specified average error value, the average error for vowels was kept lower (nearly 20 percent less) than the average error for the consonant sounds. Using the above simulation program, several incorrect texts representing different average error values were generated. These were used as input to the word boundary hypothesisation algorithm based on the language clues (Algorithm 4.1).

4.3.2 *Measures for estimating the performance of language clues*

Three measures were used to express the performance of the language clues in hypothesising word boundaries. They were, Hit rate, Correctness, and Improvement.

Phoneme deletion rules:

(1) A long stop consonant may be replaced by a short one.

Ex: kk **-->** k

(2) Any consonant sequence may be misrecognised as the last consonant in the sequence.

Ex: kt --> t

(3) The trill **'r'** may not be recognised due **it's** short duration.

(4) The semi vowels may not be recognised when they precede any vowel and the vowel may be replaced by **it's** longer version.

Ex: ya **-->** a: **.**lml

Phoneme insertion rules:

(1) The dipthongs **'ai'** and **'au'** may be misrecognised as the vowel sequences **'a'** followed by **'i'** and **'a'** followed by **'u'** respectively.

Fig.4.3 Rules for deletion and insertion of phonemes used for simulating speech-like errors in a text.

These measures were defined as follows:

$$\text{Hit rate} = \frac{\text{Number of word boundaries found}}{\text{Total number of word boundaries}}$$

$$\text{Correctness} = \frac{\text{Number of incorrect hypotheses}}{\text{Total number of hypotheses}}$$

$$\text{Improvement} = \frac{\% \text{ word boundaries in the hypotheses}}{\% \text{ word boundaries in the input}}$$

Thus Hit rate indicates how well the word boundary hypothesiser detects word boundaries. Correctness indicates the probability that a given word boundary hypothesis is correct. In other words, it indicates the confidence one can place on the clues. Improvement is a comparison of the distributions of the word boundaries and word-internals in the input and output of the word boundary hypothesiser. If one imagines the word boundary hypothesiser as a sieve that selects word boundaries from a mixture of word boundaries and word-internals, the Improvement indicates the selectivity of the sieve. Thus a large factor of Improvement indicates that the technique used for word boundary hypothesisation works well.

As seen from the above definitions for the measures, the performance of a word boundary hypothesiser is not expressed by a single measure but by a combination of them. Ideally one needs a large Hit rate to ensure that most word boundaries are detected, a large Correctness to ensure that the number of incorrect hypotheses are few and a large Improvement to reflect the selectivity of the word boundary hypothesiser. Note that a large Hit rate and a large Correctness need not necessarily mean a large Improvement. For example, if a text contained 100 word boundaries and 25 word-internals, then hypothesising every position as a word boundary will result in a 100% Hit rate and 80% Correctness but in reality it is equivalent to knowing no word

boundaries. This is reflected by the Improvement which has a value of 1, or, in other words, there is no improvement.

4.4 Results of word boundary hypothesisation using language clues

The word boundary hypothesisation algorithm using language clues (Algorithm 4.1) was applied on the Hindi texts containing different percentages of errors in phonemes. All word boundaries were removed from them (but sentence boundaries were preserved) and then word boundaries were hypothesised using the language clues. The results of word boundary hypothesisation are described in the following sections. In section 4.4.1, the results of word boundary hypothesisation for correct input are described and in section 4.4.2, the results of word boundary hypothesisation for incorrect input are described.

4.4.1 Results of word boundary hypothesisation using language clues for correct input

The results of word boundary hypothesisation using language clues on a correct Hindi text are shown in Table_4.1. The results are shown in terms of the number of word boundaries detected and the number of correct and incorrect hypotheses. Note that the number of word boundaries detected and the number of correct hypotheses differ for some clues. This is because of the fact that some word boundaries were hypothesised by two clues. The results indicate that a large number of the word boundaries, nearly 67 percent, were hypothesised correctly. Moreover the number of incorrect hypotheses were also low compared to the number of correct hypotheses. To illustrate the performance better, the results were also shown using the three measures, Hit rate, Correctness, and Improvement. It can be seen that a large percentage of the word boundaries were detected as indicated by the Hit rate of 67% and the confidence in the hypotheses produced was high as indicated by the high Correctness (about 80 %). It can also be seen that there is a considerable improvement in the distribution of

	Case markers	Conjunc-tions	Pronouns	Verb endings	Aux. verbs	Adjectives & adverbs	All clues
WBs detected	3907	1114	1581	457	1724	792	7551
Correct hyp	4159	1131	1695	457	1817	896	9990
Incorrect hyp	732	286	708	162	769	292	2689
Hit rate	39%	11%	16%	4.6%	17%	8%	76%
Correctness	85%	80%	71%	74%	70%	75%	79%
Improvement	4.0	3.7	3.25	3.5	3.25	3.4	3.45

Table 4.1 Results of word boundary hypothesisation using language clues on correct input. The results are shown for each of the classes of clues seperately.

word boundaries and word-internals as shown by the Improvement value of 3.45. Note that the maximum Improvement possible for the text is 4.7 which corresponds to 100% Correctness in the word boundary hypotheses.

The results also show the relative performance of various groups of clues. It can be seen that the case markers performed best, with a Hit rate of 39% and a Correctness of 85%. Pronouns, which are the largest in number, detected only 17% of the word boundaries and their Correctness is also lower at 74%.

4.4.2 ***Results of word boundary hypothesisation using language clues for incorrect input***

Once the utility of the language clues in word boundary hypothesisation is established for correct text input, the next step is to study their utility for input texts containing errors. The word boundary hypothesisation algorithm was applied on texts in which speech-like errors were simulated as described in section 4.3.1. The results are shown in Table 4.2.

From the results, it can be seen that for all clues, the number of word boundaries detected decrease with increasing error. This is illustrated in Fig.4.4, where the number of word boundaries detected and the number of correct and incorrect word boundary hypotheses using the language clues are plotted against input error percentages. This fall in the number of word boundaries detected is expected since some of the phoneme strings in the input text corresponding to the clues might have been corrupted and hence were not spotted. This reduction in the number of word boundaries detected can be explained as follows. If the average probability of substitution (in other words, the average error percentage), for a phoneme is p, and the length of a clue (in terms of number of phoneme) is L, then the probability that a string corresponding to a clue is uncorrupted in the input is given by $(1-p)^L$. Assuming that a clue occurs N times in the text without errors, the number of times it will occur in an incorrect text with an average error probability of p is $N \times (1-p)^L$. Hence the total

	% Error in input text					
	0	10	20	30	40	50
WBs detected	7551	6764	6018	5298	4617	4124
Correct hyp	9990	8580	7416	6319	5347	4688
Incorrecthyp	2689	2765	2728	2667	2648	2769
Hit rate	76%	68%	61%	54%	47%	43%
Correctness	79%	76%	73%	70%	67%	63%
Improvement	3.45	3.3	3.20	3.1	3.0	2.8

(a)

	% Error in input text					
	0	10	20	30	40	50
WBs detected	3907	3470	2971	2670	2242	1980
Correct hyp	4159	3668	3140	2787	2339	2056
Incorrect hyp	732	766	802	845	859	930
Hit rate	39%	35%	30%	27%	23%	20%
Correctness	85%	83%	80%	77%	73%	69%
Improvement	4.0	3.85	3.7	3.6	3.4	3.2

(b)

	% Error in input text					
	0	10	20	30	40	50
WBs detected	1114	975	880	736	677	582
Correct hyp	1131	990	892	742	683	592
Incorrect hyp	286	275	250	254	229	237
Hit rate	11%	10%	9%	7.5%	7%	6%
Correctness	80%	78%	78%	74%	75%	71%
Improvement	3.7	3.65	3.65	3.45	3.5	3.3

(c)

	% Error in input text					
	0	10	20	30	40	50
WBs detected	1581	1328	1109	916	743	611
Correct hyp	1695	1401	1161	951	760	628
Incorrect hyp	708	713	712	632	622	621
Hit rate	16%	13.4%	11%	9.3%	7.6%	6.3%
Correctness	71%	66%	62%	60%	55%	50%
Improvement	3.25	3.05	2.9	2.8	2.55	2.35

(d)

Table-4.2 Results of word boundary hypothesisation using language clues on erroneous input. The above results are for (a) all the clues together, (b) case markers, (c) conjunctions, and (d) pronouns. The results are shown for various input error percentages(0,10,20,30,40 and 50%). The input text contained 10,737 word boundaries and 39,713 word internal positions.

	% Error in input text					
	0	10	20	30	40	50
WBs detected	457	442	432	425	400	400
Correct hyp	457	442	432	425	400	400
Incorrect hyp	162	172	185	185	207	245
Hit rate	4.6%	4.5%	4.4%	4.3%	4.1%	4.1%
Correctness	74%	72%	70%	70%	66%	62%
Improvement	3.5	3.4	3.3	3.3	3.1	2.9

(e)

	% Error in input text					
	0	10	20	30	40	50
WBs detected	1724	1413	1203	974	777	664
Correct hyp	1817	1478	1242	1002	797	668
Incorrect hyp	769	689	591	546	449	415
Hit rate	17%	14%	12%	10%	8%	6.9%
Correctness	70%	68%	68%	65%	64%	62%
Improvement	3.25	3.2	3.2	3.05	3.0	3.0

(f)

	% Error in input text					
	0	10	20	30	40	50
WBs detected	792	646	579	452	394	359
Correct hyp	896	720	636	493	421	383
Imorrecthyp	292	343	342	335	360	390
Hit rate	8%	6.5%	5.9%	4.6%	4%	3.7%
Correctness	75%	68%	65%	60%	54%	50%
Improvement	3.4	3.05	2.95	2.7	2.45	2.25

(g)

Table 4.2 Results of word boundary hypothesisation using language clues on erroneous input. The above results are for (e) verb endings, (f) auxilliary verbs and (g) adverbs and adjectives. The results are shown for various input error percentages(0,10,20,30,40 and 50%). The input text contained 10,737 word boundaries and 39,713 word internal positions.

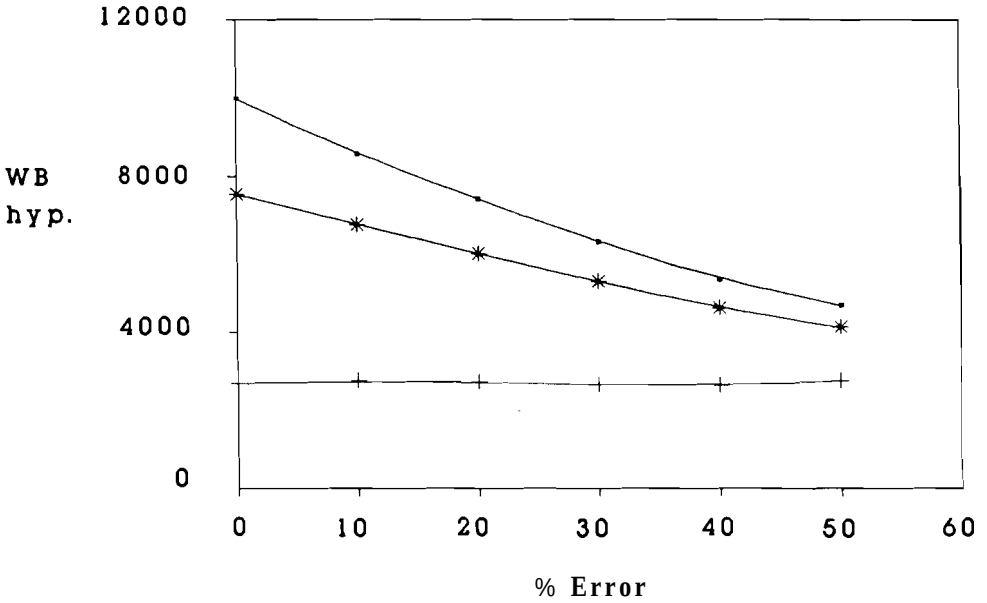


Fig.4.4 Results of word boundary hypothesisation using language clues. The clues were applied on a Hindi text containing 10,737 word boundaries and 39,713 word internal positions. In the figure, the number of correct word boundary hypotheses(indicated by .), the number of detected word boundaries(indicated by *), and the number of incorrect word boundary hypotheses(indicated by +) are shown at various input error percentages. Note that the number of correct WB hypotheses and the number of detected WBs differ, because some word boundaries were hypothesised by more than one clue.

number of correct word boundary hypotheses produced by all clues is given by $\sum_{I} N_i.(1-p)^{L_i}$. This expected number of word boundaries is compared with the observed number of correct word boundary hypotheses in Fig.4.5. The expected and observed number of word boundaries show a good agreement at low error percentages, upto 10%. But for higher errors, the number of predicted word boundaries falls off much faster than the experimental result. This may be due to the fact that many of the language clues are similar and hence one clue might be transformed to another clue due to the errors. In such a case, word boundaries will still be hypothesised around the corrupted clue. For example, the case marker *ka:*, which was one of the clues used, might become *ki:* due to errors. Since *ki:* is also a clue, word boundaries will still be hypothesised around it.

The number of incorrect word boundary hypotheses is also plotted against the input error in Fig.4.4. It shows a slow increase indicating that even at high error values, the number of incorrect hypotheses due to the language clues are limited. This can be explained as follows. An incorrect word boundary hypothesis is generated when some phoneme string in the input text is misrecognised as the phoneme string of a clue. There are two ways in which this can happen: (i) when a phoneme string in the text which is not a clue but is part of another word (or words), is recognised as the string corresponding to a clue, and (ii) when a word-internal phoneme string which was not a clue, is transformed into a string corresponding to a clue due to the errors in the input. An example of case (i) is the hypothesisation of word boundaries around the phoneme string *ka:* in the word *kala:ka:r*. An example of case (ii) is the hypothesisation of word boundaries around *ka:* in *paka:d* which was transformed from the word *pakad* due to errors. Case (i) represents the incorrect word boundary hypotheses generated on correct input and does not vary much with errors in the input, whereas case (ii) represents the incorrect hypotheses generated due to errors in the input and it accounts

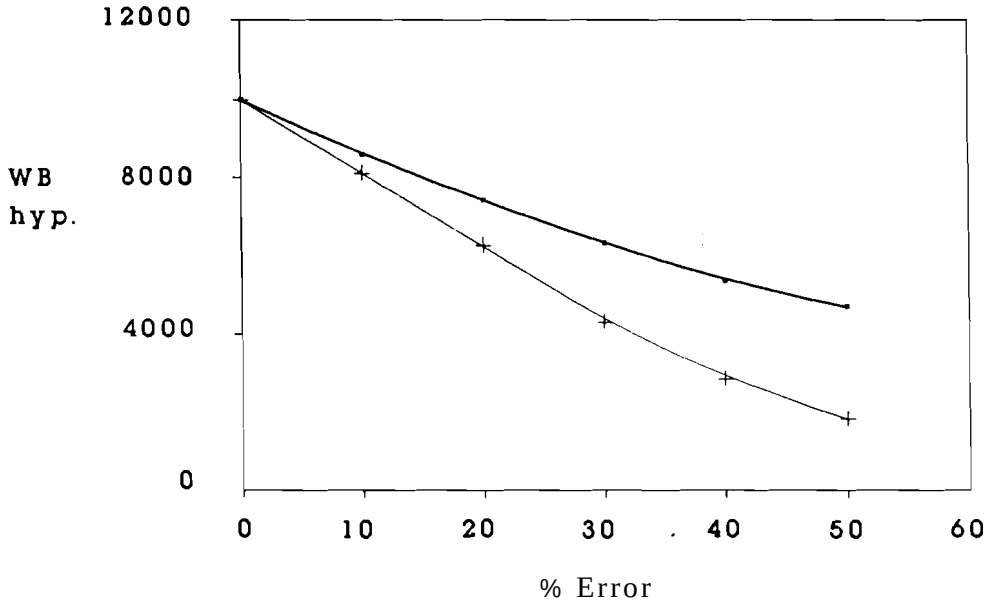


Fig.4.5 A comparison of the predicted and observed correct word boundary hypotheses. The observed correct word boundary hypotheses are shown by the thick line, whereas the predicted hypotheses are shown by the thin line.

for the increase in the number of incorrect hypotheses as input errors are increased. However, the number of phoneme strings similar to the clues and their frequencies of occurrence are low. Hence the increase in incorrect hypotheses, due to case (ii) above, is small and thus the number of incorrect hypotheses shows only a marginal increase with increase in input errors.

Results of the word boundary hypothesisation are shown in terms of the Hit rate, Correctness and Improvement in Fig.4.6. The figure shows that even at large errors, the Correctness of the word boundary hypotheses produced by the language clues, remains high indicating that one can use the language clues for word boundary hypothesisation in a speech-to-text conversion system.

4.4.3 *Distribution of subsentences in the hypotheses produced by the language clues*

Another important factor that needs to be studied is the distribution of the subsentences (strings of words with no intervening word boundaries) in the output produced by the word boundary hypothesisation algorithm. If even after word boundary hypothesisation, there are large subsentences, then the savings in the lexical analysis stage may be only marginal. For example, all the word boundaries hypothesised may occur in only one half of the sentence, leaving the other half of the sentence to be analysed by the lexical analyser. For example, in the word boundary hypothesisation algorithm using language clues, two word boundaries were hypothesised around most of the clues, but the sentence was only halved as both the boundaries were close. Hence even if many word boundaries were located using the language clues, there may still be many large subsentences left in the output text. Thus a high value for Hit rate might not necessarily mean correspondingly large savings in lexical search. Hence the distribution of the subsentences with respect to their size is important in improving lexical analysis.

The distributions of the subsentences at five error levels (0, 10, 25, 40 and 50%)

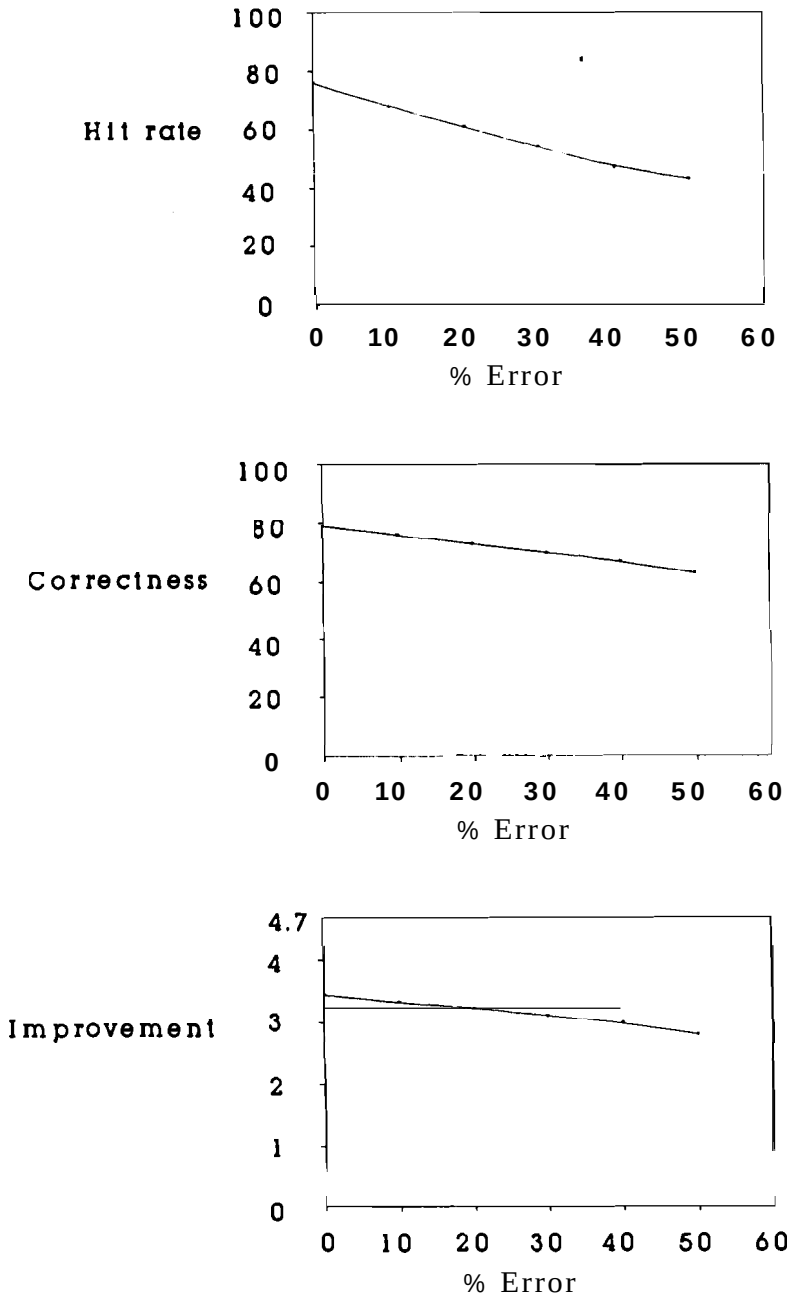


Fig.4.6 Results of word boundary hypothesisation using language clues. In the figure, the results are shown in terms of the Hit rate, Correctness and the Improvement. It can be seen that as the input error percentage increases, the performance of the clues deteriorates indicated by the falls in the Hit rate, Correctness and Improvement. However, it can also be seen that even when the input has 50% errors in phonemes, the Correctness is still more than 50%, indicating that the clues can be used on sentences with a large number of errors.

along with the original sentence distribution are shown in Fig.4.7. They indicate a gradual shift in the distribution towards larger Subsentences as the errors are increased. The plot of the average length of subsentences (in terms of number of words) against input error is shown in Fig.4.8. It also indicates the increase in the length of the subsentences with increasing errors. However if the errors are small, the distribution is still biased towards short subsentences and hence significant savings in the lexical analysis may be obtained at low error levels.

4.5 Use of lexical constraints to improve the performance of language clues

The above studies have showed the utility of language clues in detecting word boundaries. It was also seen that the performance of the language clues deteriorates as the errors in the input text increase. To improve the performance of the clues, one can use additional knowledge, such as lexical constraints, to verify the word boundary hypotheses produced by the language clues. An example of a lexical constraint is the rule that very few Hindi words end in short vowels. It was found that more than 90% of the words in a large lexicon (containing 30,000 words) have this property. This percentage is even higher for the words in a text, since most of the frequently occurring words, like case markers and conjunctions end in long vowels. Additional constraints relating to the valid word initial and final consonant sequences can also be used along with the language clues. The lexical constraints used in our study are obtained from [Bhatia 1970] and are shown in Fig.4.9.

The lexical constraints were used to eliminate some of the incorrect hypotheses produced by the language clues. A few of the correct word boundary hypotheses were also lost in this process. The results of word boundary hypothesisation when lexical constraints were used along with the language clues are given in Table_4.3. These results in terms of the Hit rate, Correctness and Improvement are also plotted in

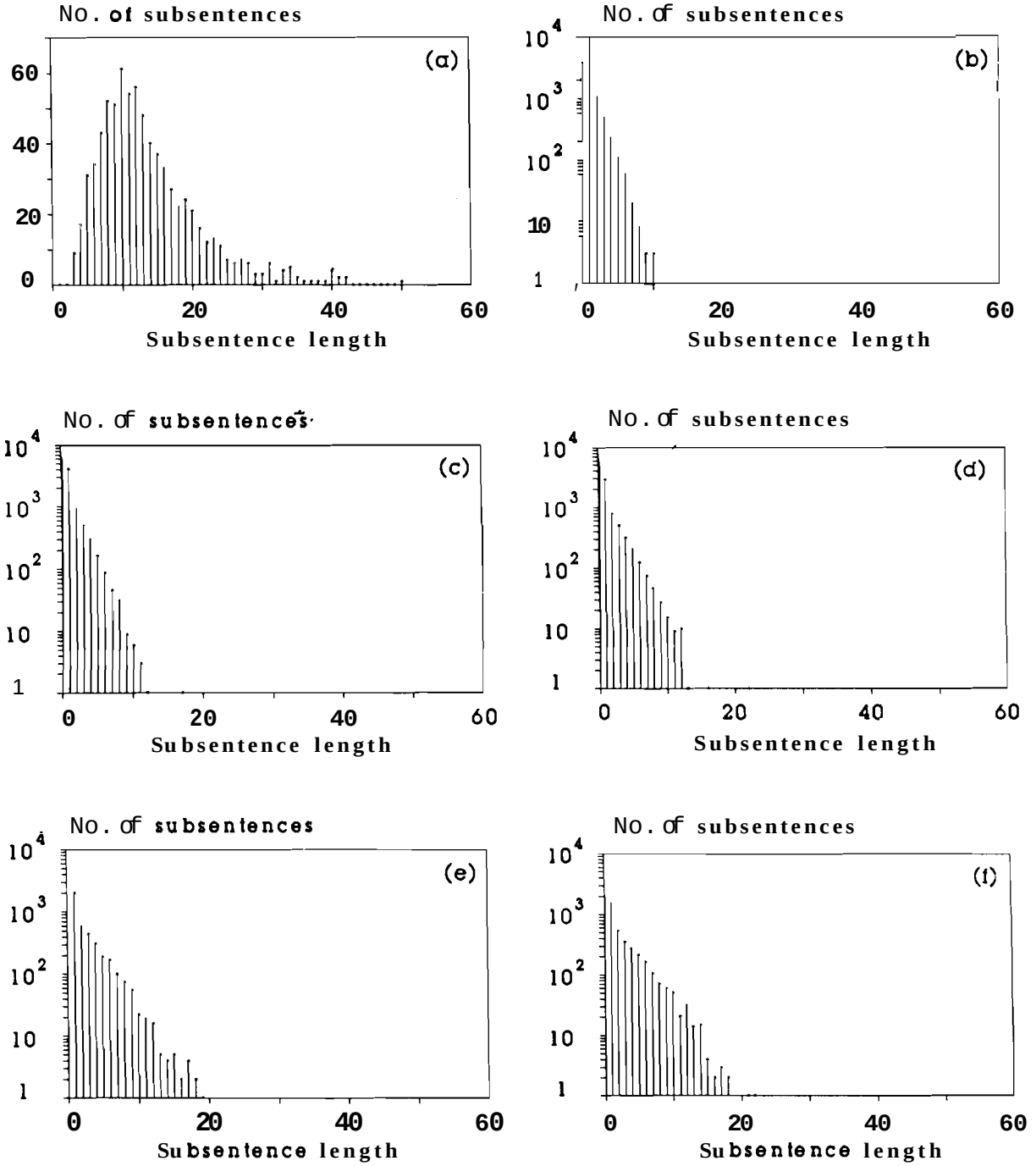


Fig.4.7 The distributions of the lengths(in words) of the subsentences produced by word boundary hypothesisation using language clues. The distributions are shown for various input error percentages of 0%, 10%, 20%, 30%, 40% and 50%, in the figures (a), (b), (c), (d), (e) and (f) respectively. It can be seen that as the percentage of input errors increases, the subsentences become longer.

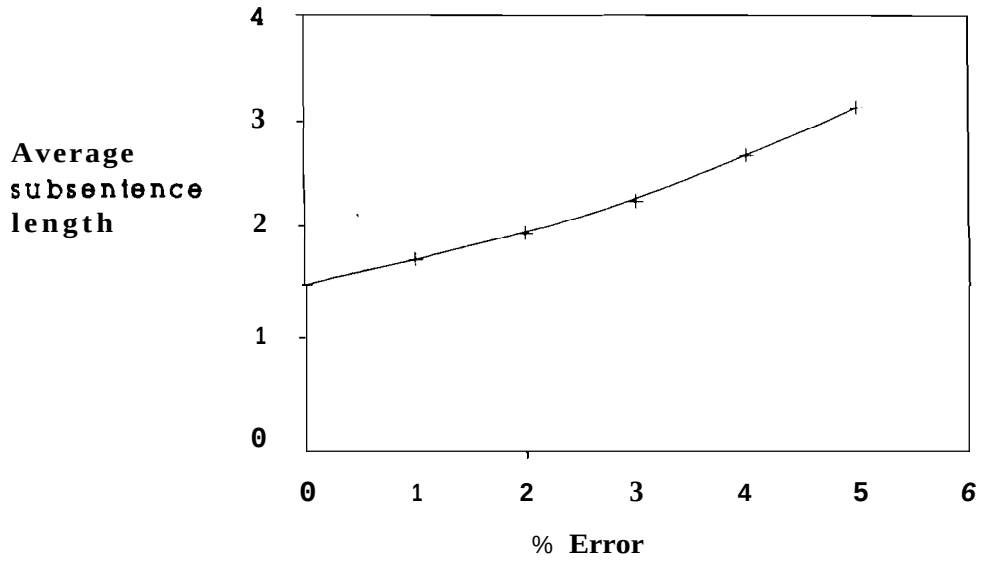


Fig.4.8 A plot of the average length(in words) of the subsentences produced by word boundary hypothesisation using language clues. It can be seen that as the percentage of input errors increases, the avearage subsentence length also increases, indicating a reduction in the performance of the clues.

LC1: A Hindi word can end either in a long vowel or in a consonant. A few exceptions like **'na', 'ki'** exist.

LC2: Only certain consonant sequences* can occur at word beginnings.

LC3: Only certain consonant sequences* can occur at word endings.

LC4: Only certain vowel sequences* can occur at word beginnings.

* for details refer [Bhatia, 1970].

Fig.4.9 Lexical constraints used to verify word boundary hypotheses produced by language clues.

	% Error in input text					
	0	10	20	30	40	50
WBs detected	7077	6210	5364	4616	3952	3452
Correct hyp	9240	7786	6533	5464	4532	3883
Incorrect hyp	1665	1655	1630	1619	1571	1623
Hit rate	71%	63%	54%	47%	40%	36%
Correctness	85%	82%	80%	77%	74%	71%
Improvement	3.8	3.7	3.6	3.5	3.35	3.2

Table 4.3 Results of word boundary hypothesisation using language clues on erroneous input. The above results are for all the clues together. Lexical constraints were used for verifying the word boundary hypotheses. eresults are shown for various input error percentages(0,10,20,30,40 and 50%). The input text contained 10,737 word boundaries and 39,713 word internal positions.

Fig.4.10 along with the results for the case when only language clues were used for word boundary hypothesisation. It can be seen that the use of lexical constraints along with language clues results in relatively higher Correctness and a lower Hit rate as compared to the results when the lexical constraints were not used. Thus the lexical constraints may be used in cases when one is willing to accept a lower Hit rate in return for a higher confidence in the hypotheses.

4.6 Summary and Conclusions

In this chapter, phoneme strings of the frequently occurring words, such as case markers, conjunctions and pronouns were proposed as clues for hypothesising word boundaries. The proposed clues were tested using texts in which all word boundaries were removed. Results showed that they perform well for correct input, detecting nearly 67% of the word boundaries with Correctness more than 80%. The clues were also tested using texts containing speech-like errors and it was shown that they work well even at high input error rates. The performance of the clues was also studied in terms of the word boundary distribution in the output text.

However not all language clues performed equally well at word boundary hypothesisation. It was observed that pronouns and their morphological variants dominated in terms of numbers (more than 50% of the clues correspond to various pronouns and their morphological variants) but they were not equally effective in producing word boundary hypotheses. Moreover, their Correctness was also lower compared to other groups, like case markers and conjunctions, indicating that the pronouns produced more incorrect hypotheses. Hence, one can remove the pronouns from the language clues and thereby increase the Correctness of the word boundary hypotheses and also reduce the number of clues to be spotted. However, this will result in a drop in the number of word boundaries detected. Hence, depending on the application, one can trade off the system performance (in terms of the number of word

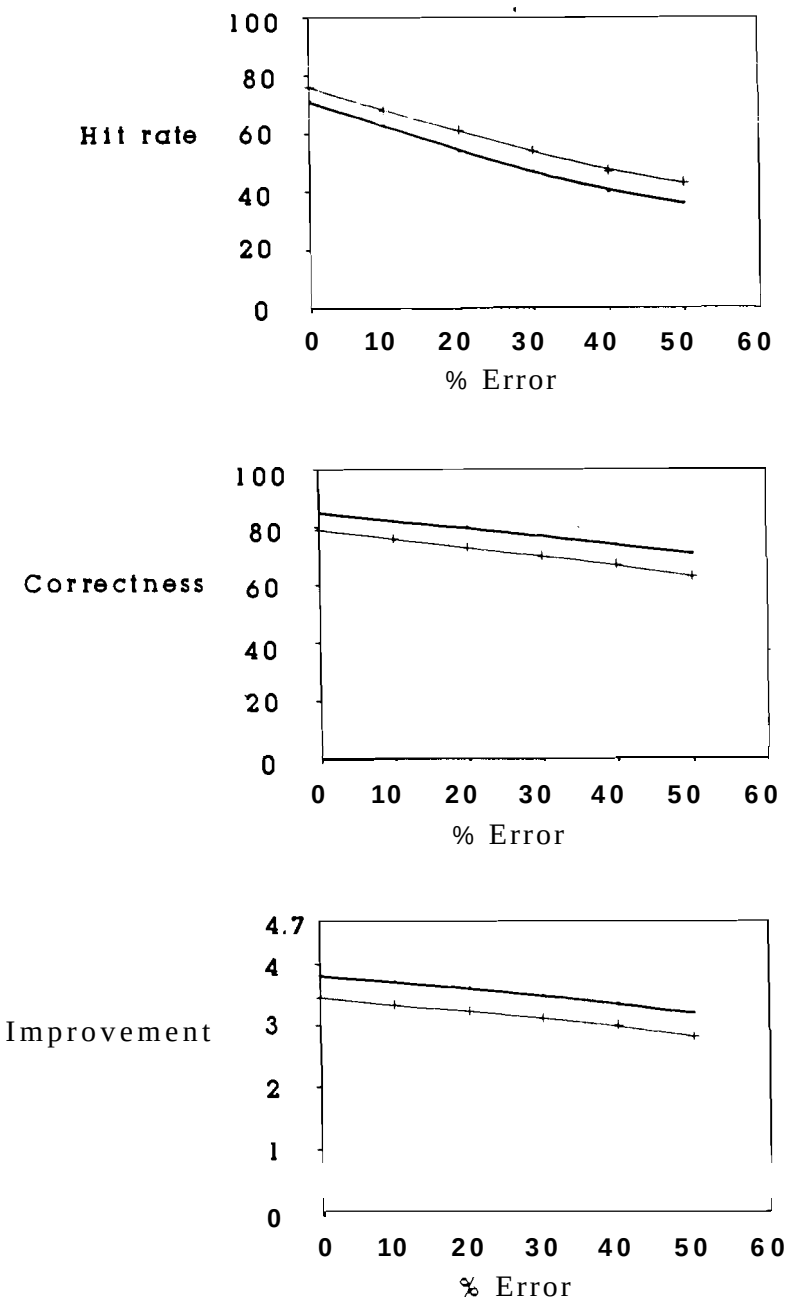


Fig.4.10 A comparison of the performances of the language clues for the cases, (i) when lexical constraints are used for verifying the word boundary hypotheses (indicated by thick line in the figure), and (ii) when no lexical constraints are used (indicated by thin line in the figure). The results are compared in terms of the Hit rate, Correctness and the Improvement. It can be seen that the use of lexical constraints increases the Correctness but reduces the Hit rate.

boundaries spotted) against system simplicity and the confidence in the hypotheses produced. For a very simple system, one can also eliminate the lexical clues used for verification and perform faster hypothesisation.

The idea of spotting of the frequently occurring words, like function words, can also be extended to other languages. However, for English, recognition of the function words in speech is more error prone compared to the recognition of other words. This is due to the fact that the function words are more distorted in English speech when compared to the content words. Hence the proposed approach may not work well for word boundary hypothesisation in English speech. But for Indian languages there are no significant differences between recognising function words and other words. Hence the proposed approach is well suited for tasks involving speech-to-text conversion in Indian languages.

The idea of spotting frequently occurring function words can also be extended to hypothesise some of the phrase and clause boundaries. This is because, some of the function words occur at the end of a phrase or a clause. Hence, if such a function word is spotted in input, then one can hypothesise a phrase/clause boundary after it. We are trying to exploit this idea in the syntax analyser module of the VOIS speech recognition system[Prakash, Rarnana Rao and Yegnanarayana 1989].

Chapter 5

WORD BOUNDARY CLUES BASED ON THE LEXICAL KNOWLEDGE

5.1 Introduction

In every language there exist some restrictions on the sequences of phonemes that can occur within words. These are commonly referred as 'Phonotactic constraints' or as 'Phoneme sequence constraints'. These constraints can often be used to hypothesise word boundaries in a sentence from which all word boundaries are removed. For example, the phoneme sequence *mgl* does not occur word-internally in English. Hence if a phoneme string representing an utterance contains such a sequence (for example *samglas* representing the word sequence, some glass), one can hypothesise a word boundary within the sequence as *mg#l* or as *m#gl* (# indicating the boundary). Further, one can locate the word boundary exactly, since the sequence *mg* also does not occur in a word and hence the boundary must lie between *m* and *g* as *m#gl*. Similar constraints on the word-internal phoneme sequences can also be identified for Hindi, and used for word boundary hypothesisation. For example, in Hindi, one can hypothesise a word boundary within the phoneme sequence *ktk* as it does not occur within a word. Thus word boundary hypothesisation using the phoneme sequence constraints involves spotting of phoneme sequences in the input text which are not word-internal sequences, and hypothesise word boundaries within them. Since these sequences act as clues to word boundaries and are obtained using a lexicon, they are referred as 'lexical clues' in our studies.

There are several issues to be considered before extracting the lexical clues from a dictionary. The first one is with the definition of word itself. The question to be answered is which sequences of phonemes are to be considered words. For example, in the case of a verb, several inflected forms corresponding to the various tenses exist though they may not be listed in the lexicon. Similarly some words may be derived

from other words, such as 'nationality', and 'nationalise' which are derived from the word 'nation'. Whether all such words are to be considered for the extraction of the lexical clues is an issue to be addressed. Another issue in extracting the lexical clues is with compound words. **A** compound word is formed from two or more words as the word 'afternoon' formed from the words 'after' and 'noon'. Whether a compound word is to be considered **as** a single word or **as** a sequence of several words is another issue to be decided. Problems may also arise due to assimilation of phonemes in speech. Thus the spoken word may not match the one actually stored in the lexicon which usually corresponds to its written form.

In our study, these issues were deferred to the lexicon builders. A commonly used dictionary of Hindi was selected and all entries listed in it were treated as words, be they simple words or compound words. Derived words which were listed in the dictionary were also considered in the study. But the various inflected forms of the words which were not listed in the dictionary were ignored. These included most inflected forms of verbs and the plural versions of many nouns. In Hindi, the spoken and the written forms of the words do not differ much as there is an almost one-to-one relationship between the Hindi letters and phonemes. One important difference arising from the 'a-deletion'[Ohala 19831 was taken care of in our representation of words.

Another issue to be addressed is the type of phoneme sequences to be used for word boundary hypothesisation. In most of the earlier studies for English, reported in section 2.3.2, the sequences of the form C^+, V^+, and CVC ($^+$ indicating sequence of one or more phonemes) are used. In our study, consonant sequences of type C^+, VC^+, C^+V and VC^+V, and vowel sequences of type V^+, CV^+, V^+C and CV^+C are used.

The chapter is organised **as** follows. In section 5.2, the extraction of the various types of lexical clues from the dictionary is described and the word boundary

hypothesisation algorithm using these clues is given. In section 5.3, the results of word boundary hypothesisation using lexical clues for correct input are presented. As in the case of language clues, the lexical clues are also to be applied on texts containing speech-like errors to estimate their performance in a speech-to-text conversion system. Hence lexical clues were applied on texts in which some errors were simulated (described in section 4.3). The results of this are presented in section 5.4. To increase the number of word boundaries detected by the clues, the clues were increased by adding infrequently occurring word-internal phoneme sequences. These results are presented in section 5.5. In section 5.6, the performance of the lexical clues extracted from a smaller dictionary is studied. Most of the word boundary hypotheses produced by the clues contain more than one possible location for the word boundary. Hence studies were done on the distribution of word boundaries within the lexical clues and the results are presented in section 5.7. Finally, a summary of the studies and their implications is given in section 5.8.

5.2 Lexical clues for word boundary hypothesisation

The lexical clues are the phoneme sequences which do not occur within a word. They were obtained by examining the words in a dictionary and listing the phoneme sequences that were present in the words. These word-internal phoneme sequences were then removed from the set of all possible phoneme sequences of that type, to obtain the clues. For example, to identify all lexical clues of type C^+, all consonant sequences present in the words of a dictionary, were removed from the set of all possible consonant sequences. The dictionary used for this purpose is the Meenakshi Hindi dictionary [Mohan and Kapoor 1989], which contained nearly 31000 words. It contained many compound words (nearly 3000) also. However it did not contain any of the inflected forms of verbs. In this study, some of the infrequent compound words were deleted and the rest were treated as single words. The resulting reduced

dictionary contained nearly 30000 words.

Once the lexical clues were identified, the following algorithm was used to hypothesise the word boundaries:

Algorithm 5.1 Word boundary hypothesisation using lexical clues

1. Read the input text until a phoneme sequence of the required type is observed.
2. Check whether the observed sequence is one of the lexical clues or not. If it is a clue, then hypothesise a word boundary within the sequence.
3. Repeat 1 and 2 till the end of input.

5.3 Results of word boundary hypothesisation using lexical clues for correct input

The above word boundary hypothesisation algorithm was used to hypothesise word boundaries in a Hindi text described in the section 4.3.1 of this thesis. The text did not contain any errors except that all word boundaries were removed from it (but sentence boundaries preserved). For each type of clue (C^+, V^+ etc.), the above word boundary hypothesisation algorithm was applied and the results are shown in Table 5.1. The results are shown in terms of the number of word boundaries detected and the number of correct and incorrect hypotheses and also in terms of the three measures, Hit rate, Correctness and Improvement.

One issue to be remembered with lexical clues is the uncertainty in the position in the word boundary. This is because of the fact that the clues result in the identification of a phoneme sequence within which a word boundary is hypothesised. However, if the sequence is long (for example a sequence of type CVVC), then the word boundary can be placed at a number of positions within the sequence. Thus the lexical clues result in only the approximate location of a word boundary. This is

illustrated in our results through the Improvement measure. In this context, the Improvement can be interpreted as the reduction in the uncertainty of word boundary position. For example, an Improvement of 2 means that when compared to the case of unknown word boundaries, the hypotheses produced by the clues arc twice more certain (or the probability that a hypothesised position corresponds to a word boundary is twice that of the probability that a randomly chosen position in the input corresponds to a word boundary).

5.3.1 *Lexical clues in the form of vowel sequences*

In this section, the results of word boundary hypothesisation using the lexical clues in the form of vowel sequences are presented. Four types of vowel sequences, V^+, CV^+, V^+C and CV^+C were considered in this. The results are presented first for simple vowel sequences of the form V^+, followed by progressively larger sequences.

(a) Simple vowel sequences of the form V^+

The results of word boundary hypothesisation using clues of the form V^+ are shown in Table_5.1. It can be seen that about 3.3% of the total word boundaries are hypothesised correctly. The Correctness is high at **93%** indicating that one can place a large confidence in the hypotheses.

Since the Hit rate is very low, the improvement in the lexical analysis due to the word boundary hypotheses may only be marginal. One way to improve the Hit rate is to use additional constraints such as the consonants preceding and succeeding the vowel sequences. The results for such sequences are given below.

(b) Vowel sequences of the form CV^+ and V^+C

The lexical clues of the form CV^+ and V^+C were used to hypothesise word boundaries and the results are also shown in Table_5.1. As seen from the table, the Hit rate increased more than two times for both the sequences when compared to the Hit rate of simple vowel sequences. However, the Correctness decreased for CV^+,

	Vowel sequences				Consonant sequences			
	V^+	CV^+	V^+C	CV^+C	C^+	VC^+	C^+V	VC^+V
WBs detected	369	771	900	2163	410	1099	1280	3482
Correct hyp	369	771	900	2163	410	1099	1280	3482
Incorrect hyp	34	108	35	184	26	73	258	628
Hit rate	3.4	7.2	8.4	20.1	3.8	10.2	11.9	32.4
Correctness	92	88	96	92	94	94	83	85
Improvement	4.2	2.2	2.5	2.0	2.9	2.1	1.9	1.66

Table-5.1 Results of word boundary hypothesisation using lexical clues on a correct input.

whereas for V^+C, it increased when compared to that of simple vowel sequences.

(c) Vowel sequences of the form CV^+C

Further improvement in Hit rate may be expected if both constraints of preceding and succeeding consonants were applied simultaneously. However the Correctness may decrease as longer sequences are more prone to errors.

The results of word boundary hypothesisation using CV^+C type of clues are also shown in Table 5.1. Nearly 20% of the word boundaries in a correct text input are detected. The Correctness is around 92%, which is greater than the value for CV^+ sequences.

5.3.2 *Lexical clues in the form of consonant sequences*

In this section the results of word boundary hypothesisation using the lexical clues in the form of consonant sequences are presented. The studies performed are similar to the ones on sequences of vowels. Four types of consonant sequences, C^+, VC^+, C^+V and VC^+V, were used to hypothesise word boundaries in a correct Hindi text. The results are presented for C^+ type of sequences first, followed by progressively larger consonant sequences involving the preceding and succeeding vowels.

(a) Simple sequences of consonants of the form C^+

The results are also shown in Table 5.1. It can be observed that nearly 3.9% of the word boundaries in the text were detected correctly while the Correctness was high at 93%. However, the Hit rate is low as in the case of simple vowel sequences, and to improve the Hit rate, one needs to utilise additional constraints like preceding and succeeding vowels.

(b) Consonant sequences of the form C^+V and VC^+

The number of word boundaries detected using simple sequences of consonants is low. Hence to detect more word boundaries, lexical clues of the form C^+V and

VC^+ were considered. It can be observed from Table 5.1 that the Hit rate more than doubled for both types of sequences as compared to the simple consonant sequences. The Correctness for sequences of type C^+V is however smaller than that of simple consonant sequences, whereas for sequences of type VC^+ it is nearly the same.

(c) Consonant sequences of the form VC^+V

The constraints of the preceding and succeeding vowels on consonant sequences resulted in an increase in the number of word boundaries detected. By applying both the constraints together in the form of sequences of type VC^+V, one can increase the Hit rate further. The results of word boundary hypothesisation for these clues are shown in Table 5.1. It can be seen that the Hit rate increased to 32%. The number of errors also increased though the Correctness at 85% is same as the value for the C^+V type of sequences.

In the above, it was observed that the sequences of types CV^+C and VC^+V detected the maximum number of word boundaries. One can further increase the number of word boundaries detected by applying these two types of clues together. The results for these clues together are also shown in Table 5.1. It can be seen that nearly 50% of the word boundaries were detected with a Correctness of 87%.

5.3.3 ***Errors in word boundary hypotheses produced by lexical clues***

(a) Vowel sequences (V^+, CV^+, V^+C and CV^+C)

Most of the errors in the word boundary hypotheses produced using constraints on vowel sequences were due to plural words. For example, the vowel sequence ***a:o:*** did not occur in any word in the dictionary, and hence it was used as a clue to hypothesise word boundaries. However this sequence occurs within the plural forms of many words such as the word ***lata:***. One of the plural forms of this word is ***lata:o:n*** containing the vowel sequence ***a:o:***. Since this is not a word-internal sequence derived from the dictionary, a word boundary will be hypothesised between ***a:*** and ***o:*** resulting

in an error. Similarly, the vowel sequence *a-o:* is itself a word, corresponding to one of the inflected forms of the verb *a:*. In this case also, a word boundary was wrongly hypothesised between *a* and *o:*. In these studies, it was observed that nearly 75% of the errors in the word boundary hypotheses from the lexical clues of the form V^+ were from this single sequence *a:o:*. Most of the remaining errors were due to compound words except for a few which were due to foreign words, mainly English words.

(b) Consonant sequences (C^+, VC^+, C^+V and VC^+V)

The errors in the word boundary hypotheses produced using constraints on the consonant sequences were dominated by the inflected words. Though a few other errors also occurred due to compound words, more than 60% of the errors were due to inflections. For example, verbs were represented in the dictionary using their root forms only. However, the input text contained many inflected forms of verbs which were produced from the root forms by appending appropriate sequences of consonants and vowels. For example, consider the verb *Samaj^{h}na:*. One of the inflected forms for this verb is *Samaj^{h}te:* containing a consonant sequence *j^{h}t* which did not occur in any word in the dictionary. Hence a word boundary was hypothesised between *j^h* and t in the word *Samaj^{h}te:* resulting in an error. Similarly the noun *udd^{h}andta:* is derived from the word *udd^{h}and* and it was not present in the dictionary. The sequence *ndt* did not occur in any word in the dictionary and hence a word boundary was wrongly hypothesised within the sequence *ndt* in *udd^{h}andta:*.

The above results show that most of the errors in the word boundary hypotheses produced by lexical clues are due to the derived words like plural forms, or verb inflections. Thus one can minimise these incorrect hypotheses by including all derived words in the dictionary. However, it is not easy to find all possible words which can be derived from the words in the dictionary. Moreover, as observed in our studies, the

percentage of incorrect hypotheses is small (as shown by the large Correctness values) and hence in our studies no derived words (other than the words which were present in the dictionary) were considered in extracting the lexical clues.

5.4 Performance of the lexical clues for incorrect input

The results of the previous section show that the lexical clues in the form of phoneme sequence constraints are useful to detect word boundaries in a correct input. However, in the context of speech recognition, the input to the word boundary hypothesiser produced by the speech signal-to-symbol conversion usually contains errors. Hence it is necessary to study the performance of the lexical clues in hypothesising word boundaries for an input text containing errors similar to those occurring in a speech-to-text conversion system.

The word boundary hypothesiser algorithm was applied on texts in which errors likely in speech to symbol conversion were simulated. The simulation of the errors was as described earlier in section 4.3.1. The results of the word boundary hypothesisation for the various types of sequences at different percentages of errors are shown in Table 5.2. The results show the number of word boundaries detected, the number of correct and incorrect hypotheses, the Hit rate, Correctness and Improvement for varying error rates.

The results show that the longer sequences (CV^+C and VC^+V) produced a larger number of incorrect hypotheses as compared to the shorter ones (V^+ and C^+). Also sequences containing a VC seem to be less error prone than sequences containing a CV. Thus sequences of type V^+C have a larger Correctness than sequences of type CV^+ for the same input error. Similarly VC^+ sequences have a larger Correctness compared to C^+V sequences though in this case the difference is much less.

Results are also shown in graphical form in Fig.5.1 and Fig.5.2.. In Fig.5.1, the number of word boundaries detected and the number of incorrect word boundary

	% Error in input text					
	0	10	20	30	40	50
WBs detected	369	380	379	395	413	410
Correct hyp	369	380	379	395	413	410
Incorrect hyp	34	66	96	130	166	201
Hit rate	3.4%	3.5%	3.5%	3.7%	3.8%	3.8%
Correctness	92%	85%	80%	75%	71%	67%
Improvement	4.2	3.8	3.4	3.1	2.9	2.7

(a)

	% Error in input text					
	0	10	20	30	40	50
WBs detected	900	929	957	984	1041	1074
Correct hyp	900	929	957	984	1041	1074
Incorrect hyp	35	64	87	115	133	189
Hit rate	8.4%	8.6%	8.9%	9.1%	9.7%	10%
Correctness	96%	94%	92%	90%	89%	85%
Improvement	2.5	2.4	2.3	2.2,	2.2	2.1

(b)

	% Error in input text					
	0	10	20	30	40	50
WBs detected	771	772	783	813	821	798
Correct hyp	771	772	783	813	821	798
Incorrect hyp	108	166	233	302	383	467
Hit rate	7.2%	7.2%	7.3%	7.6%	7.6%	7.4%
Correctness	88%	83%	77%	73%	68%	63%
Improvement	2.2	2.1	1.9	1.8	1.7	1.6

(c)

	% Error in input text					
	0	10	20	30	40	50
WBs detected	2163	2239	2307	2421	2508	2571
Correct hyp	2163	2239	2307	2421	2508	2571
Incorrect hyp	184	395	679	910	1140	1384
Hit rate	20.1%	20.8%	21.5%	22.5%	23.3%	23.9%
Correctness	92%	85%	77%	73%	69%	65%
Improvement	2.0	1.9	1.8	1.7	1.7	1.6

(d)

Table-5.2 Results of word boundary hypothesisation using lexical clues on erroneous input. The above results are for clues of types (a) V^+, (b) V^+C, (c) CV^+ and (d) CV^+C. The results are shown for various input error percentages (0,10,20,30,40 and 50%). The input text contained 10,737 word boundaries and 39,713 word internal positions.

	% Error in input text					
	0	10	20	30	40	50
WBs detected	410	445	460	487	518	562
Correct hyp	410	445	460	487	518	562
Incorrect hyp	26	72	119	146	184	202
Hit rate	3.8%	4.1%	4.3%	4.5%	4.8%	5.2%
Correctness	94%	86%	79%	77%	74%	73%
Improvement	2.9	2.7	2.6	2.6	2.5	2.5

(e)

	% Error in input text					
	0	10	20	30	40	50
WBs detected	1280	1326	1372	1379	1376	1423
Correct hyp	1280	1326	1372	1379	1376	1423
Incorrecthyp	258	366	454	520	586	661
Hit rate	11.9%	12.3%	12.8%	12.8%	12.8%	13.2%
Correctness	83%	78%	75%	73%	70%	68%
Improvement	1.9	1.8	1.8	1.7	1.7	1.65

(f)

	% Error in input text					
	0	10	20	30	40	50
WBs detected	1099	1194	1249	1303	1386	1400
Correct hyp	1099	1194	1249	1303	1386	1400
Incorrect hyp	73	178	284	407	468	534
Hit rate	10.2%	11.1%	11.6%	12.1%	12.9%	13.0%
Correctness	94%	87%	81%	76%	75%	72%
Improvement	2.1	2.0	1.9	1.8	1.8	1.7

(g)

	% Error in input text					
	0	10	20	30	40	50
WBs detected	3482	3483	3556	3552	3505	3485
Correct hyp	3482	3483	3556	3552	3505	3485
Incorrecthyp	628	852	995	1173	1306	1479
Hit rate	32.1%	32.3%	33.1%	33.0%	32.6%	32.4%
Correctness	85%	80%	78%	75%	73%	70%
Improvement	1.7	1.6	1.5	1.5	1.5	1.4

(h)

Table 5.2 Results of word boundary hypothesisation using lexical clues on erroneous input. The above results are for clues of types (e) C^+, (f) C^+V, (g) VC^+ and (h) VC^+V. The results are shown for various input error percentages (0,10,20,30,40 and 50%). The input text contained 10,737 word boundaries and 39,713 word internal positions.

	% Error in input text					
	0	10	20	30	40	50
WBs detected	5400	5486	5623	5739	5801	5845
Correct hyp	5400	5486	5623	5739	5801	5845
Incorrect hyp	812	1247	1664	2083	2446	2863
Hit rate	50%	51%	52%	53%	54%	54%
Correctness	87%	81%	77%	73%	70%	67%
Improvement	1.8	1.7	1.6	1.6	1.5	1.5

(i)

Table_5.2 Results of word boundary hypothesisation using lexical clues on erroneous input. The above results in (i) are for sequences of type $VC^{+}V$ and $CV^{+}C$ together. The results are shown for various input error percentages(0,10,20,30,40 and 50%). The input text contained 10,737 word boundaries and 39,713 word internal positions.

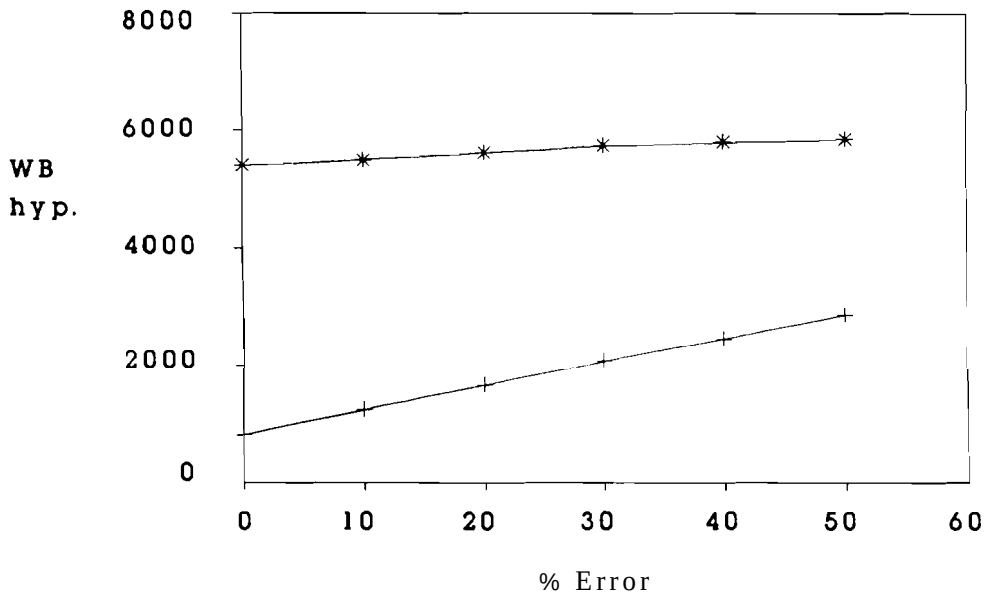


Fig.5.1 Results of word boundary hypothesisation using lexical clues. The clues were applied on a Hindi text containing 10,737 word boundaries and 39,713 word internal positions. In the figure, the number of detected word boundaries(indicated by *), and the number of incorrect word boundary hypotheses(indicated by +) are shown at various input error percentages. It can be seen that as the input errors increase, the number of incorrect word boundary hypotheses increases, whereas the number of detected word boundaries remains constant.

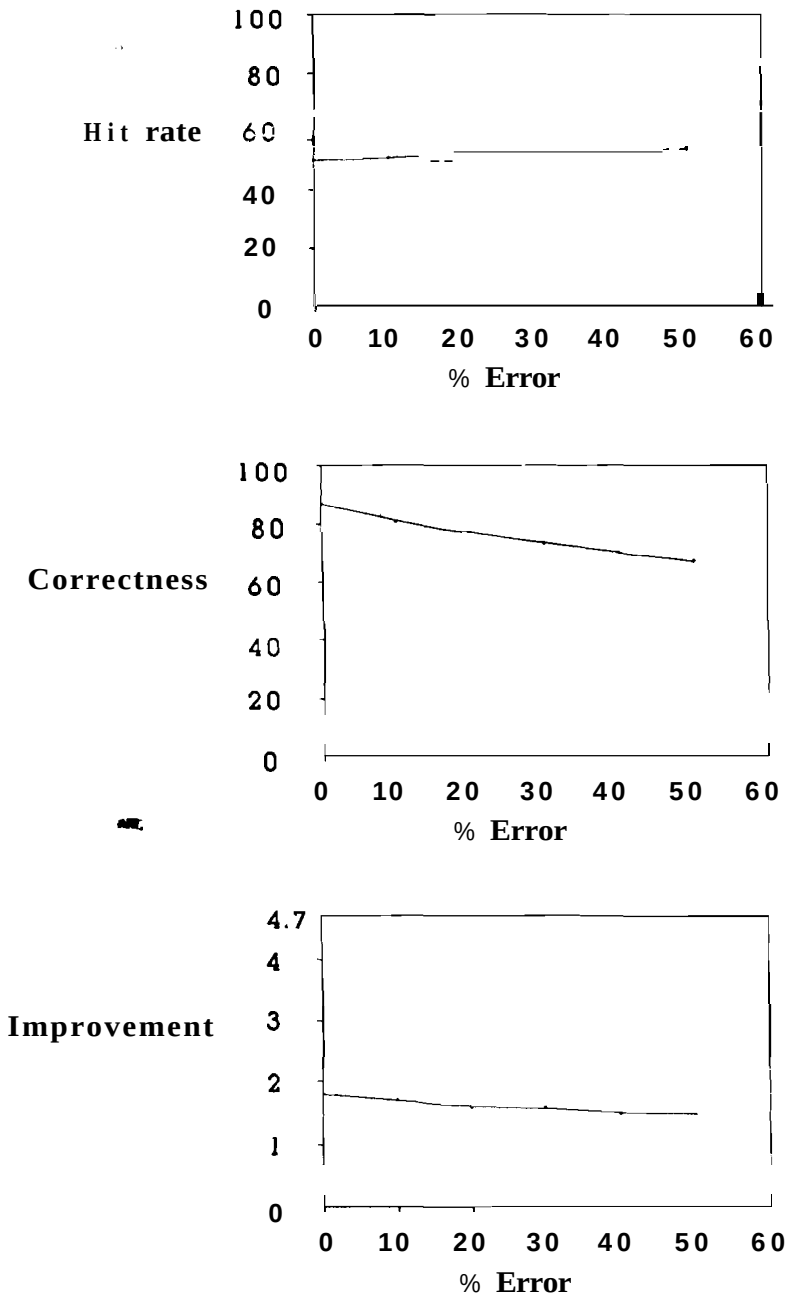


Fig.5.2 Results of word boundary hypothesisation using lexical clues. The results are shown in terms of the Hit rate, Correctness and the Improvement. It can be seen that as the input error percentage increases, the performance of the clues deteriorates indicated by the falls in the Hit rate, Correctness and Improvement. However, it can also be seen that even when the input has 50% errors in phonemes, the Correctness is still more than 50%, indicating that the clues can be used on sentences with a large number of errors.

hypotheses produced by the lexical clues of types CV^+C and VC^+V together, are plotted against the input error. It can be seen that the number of word boundaries detected remains practically constant, whereas the number of incorrect word boundary hypotheses increases steadily. In Fig.5.2, these results are shown in terms of the Hit rate, Correctness and Improvement. It can be seen that the Hit rate remains constant, whereas the Correctness and the Improvement decrease with increasing input error percentage. Similar results are observed for other types of clues also with minor variations. In general, it can be seen that for vowel sequences (V^+, CV^+, V^+C, and CV^+C) the Hit rate remains practically constant whereas for the consonant sequences (C^+, VC^+, C^+V and VC^+V) it shows a marginal increase.

This variation in the number of word boundary hypothesisation errors and the number of detected word boundaries with input error can be predicted as described in the following subsection:

5.4.1 ***Estimation of the number of word boundaries detected and the number of errors***

Let S be the set of word-internal phoneme sequences (of vowels say) and $\underline{S}$ be the complement of S, i.e., $\underline{S}$ is the set of the lexical clues or the phoneme sequences that occur only across a word boundary. Now consider an input text containing a word-internal phoneme sequence x, i.e., x is a member of S. Due to errors in the speech to symbol conversion (in our case due to the simulated errors) this sequence x may be misrecognised as another sequence y. In other words, in the input to the word boundary hypothesiser, y occurs in place of x. Now if y happens to be a member of $\underline{S}$, then a word boundary would be hypothesised within y resulting in an error. Hence one can estimate the word boundary hypothesisation errors for any input error percentage by estimating the probability that a word-internal sequence x is transformed into nonword-internal sequence y due to the errors in the speech signal-to-symbol

conversion .

Assume that p is the probability that a phoneme is misrecognised, i.e., p is the average phoneme error rate in the input text. Consider the sequence x to be of length L. Now the probability that x is unaffected by the input errors is given by $(1-p)^L$. Hence the probability that x is transformed into some other sequence y is given by $1-(1-p)^L$ (= P_c say) . Now the probability that the sequence x is transformed into a nonword-internal sequence y is given by P_c multiplied by the probability that the sequence y belongs to the set $\underline{S}$(= $P(y \in \underline{S})$). Hence if one can estimate $P(y \in \underline{S})$, then one can estimate the number of errors in the word boundary hypothesisation.

The estimation of $P(y \in \underline{S})$ is difficult and it usually depends on the sequence x. One can, however, make some simplifying assumptions and estimate it for those cases. One simple assumption made was that the transformed string y is equally likely to belong to S or $\underline{S}$, which holds good if the word-.internal sequences were randomly distributed. Thus $P(y \in \underline{S})$ is proportional to the size of $\underline{S}$ and is given by $|\underline{S}|/(|S|+|\underline{S}|)$ where $|S|$ stands for the size of S. Hence the probability that a sequence x is transformed into a nonword-internal sequence y is given by $(1-(1-p)^L) * |\underline{S}|/(|S|+|\underline{S}|)$. The number of word boundary hypothesisation errors is given by the number of word-internal sequences in the input text multiplied by the above probability.

One factor that was neglected in our estimation is the effect of the inherent errors, i.e., the incorrect hypotheses produced even with a correct input. These are nothing but the word-internal sequences in the input text which do not appear in any word in the dictionary. These are neglected in the above formula. However, if they are large in number, then their effect will also have to be considered. They will contribute to the incorrect hypotheses in two ways: (i) these sequences may remain unaffected by any input errors and thus continue to produce wrong hypotheses or (ii) they may get

transformed into other nonword-internal sequences and thus produce incorrect hypotheses. Suppose N_0 represents the word boundary hypothesisation errors for zero input error. Then the contribution of the first part is nothing but N_0 multiplied by the probability that the sequences did not change during the simulation, i.e., $N_0 * (1-p)^L$. The second part's contribution is $N_0 * (1-(1-p)^L) * |\underline{S}|/(|S| + |\underline{S}|)$. However for most forms of lexical clues, the number of inherent errors were small, especially for the cases of simple vowel and consonant sequences. Hence they can be neglected.

As seen from the formula, the number of incorrect word boundary hypotheses depends on the average error rate and the length of the sequence. Using the formula, the number of incorrect hypotheses for simple vowel and consonant sequences (V^+ and C^+) were estimated and are shown in Table 5.3(a), where they are compared with the observed number of incorrect hypotheses. The comparison of the predicted and the observed number of incorrect word boundary hypotheses for the simple sequences of vowels and consonants is also shown in graphical form in Fig.5.3.

It can be seen from the figure, that there is a good agreement between the predicted and observed numbers of incorrect hypotheses, for vowel sequences. However, the observed and predicted incorrect hypotheses for the consonant sequences differ significantly (almost by a factor of 3). The reason for this may be due to the similarities between the word-internal consonant sequences because such similarities cause many of the word-internal sequences to map again onto other word-internal sequences. Obviously in such a case, the factor $P(y \subset s)$ in the formula will be much smaller than what was assumed. Hence the observed number of incorrect word boundary hypotheses will be less than the predicted number. However the growth rate of the number of incorrect hypotheses with respect to errors in the input, should be quite similar. This is because the growth factor $(1-(1-p)^L)$ is independent of $P(y \in \underline{S})$

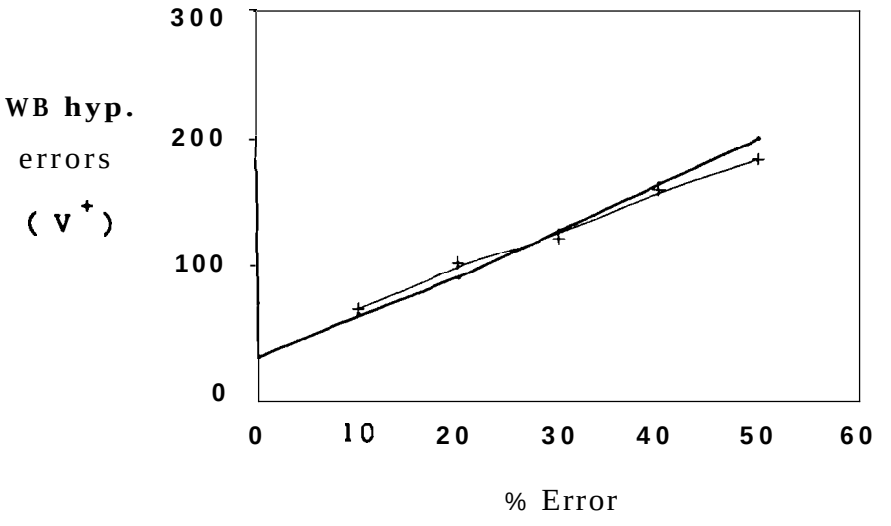
Sequence Type	% Error in input 0	10	2C	30	40	50
v^+	(34)	71 (66)	106 (94)	124 (130)	162 (166)	185 (201)
c^+	(26)	192 (72)	324 (119)	479 (146)	594 (184)	694 (202)

(a)

Sequence Type	<-- % Error in input --> 0	10	20	30	40	50
v^+ (3.04)		1.00 (1.00)	1.49 (1.42)	1.75 (1.97)	2.28 (2.51)	2.60
c^+ (2.81)		1.00 (1.00)	1.69 (1.65)	2.49 (2.03)	3.09 (2.56)	3.61

(b)

Table–5.3 A comparison of the Predicted and Observed word boundary errors for simple sequences of vowels and consonants (v^+ and c^+). 5.3 (a) shows the number of erroneous hypotheses at various input errors. 5.3 (b) shows the growth rates for the predicted and observed errors. The observed word boundary errors and their growth rate are shown in parentheses. The growth rate is defined as the ratio between the number of errors and the number of errors at an input error of 10%.



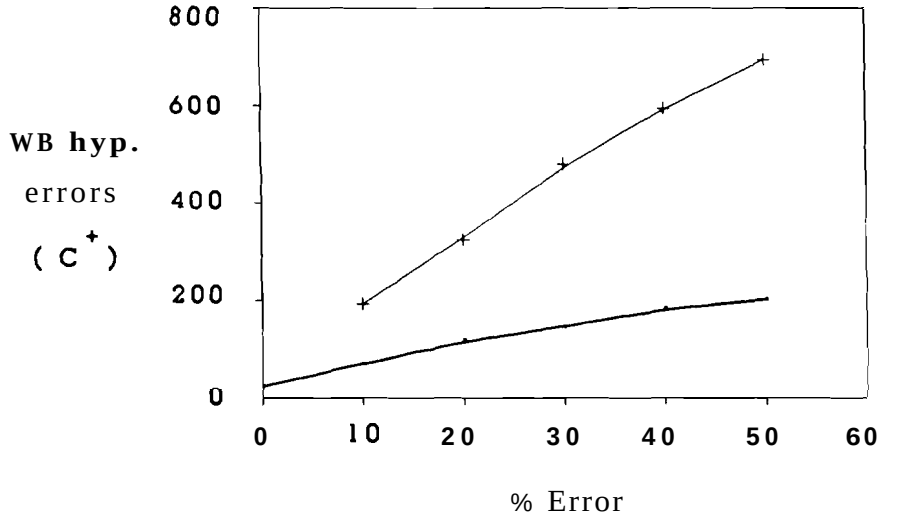


Fig.5.3 A comparison of the observed number of incorrect word boundary hypotheses(indicated by the thick line) and the predicted number of incorrect hypotheses(indicated by the thin line) at various input error percentages, for the two cases of simple vowel sequences(V^+) and simple consonant sequences(C^+). It can be seen that there is a good agreement for V^+ type of sequences whereas for C^+ type of sequences the agreement is poor.

(assuming that $P(y \in S)$ does not depend on p). Hence if one normalises the number of wrong hypotheses with respect to the number of wrong hypotheses at 10% error say, then the resulting values should be the same for the predicted and observed hypotheses. This is shown in Table 5.3(b) which shows a good agreement between the predicted and observed values.

The number of word boundaries detected by the lexical clues can also be estimated on similar lines. However, a qualitative assessment will serve the purpose in this case as the observed changes in the number of detected word boundaries for all sequences are small. The number of correctly detected word boundaries is affected in two ways by an increase in the input error:(i) The number will decrease due to some of the nonword-internal phoneme sequences containing word boundaries getting transformed to word-internal sequences, and, thereby become undetected, and (ii) The number will increase due to some of the sequences corresponding to the word-internal sequences and containing word boundaries getting transformed to nonword-internal sequences thereby getting detected. The expressions for both these factors are similar and depend roughly on the relative sizes of the sets of word-internal and nonword-internal phoneme sequences and also on the actual number of word-internal and nonword-internal sequences containing word boundaries present in the input. If these are of the same order, then the Hit rate will not change significantly.

5.5 Effect of adding infrequent word-internal phoneme sequences to the lexical clues

It was observed during the extraction of the word-internal phoneme sequences from the lexicon that many of the phoneme sequences occur only once or twice. Hence it was decided to include these infrequent word-internal phoneme sequences in the lexical clues and study the corresponding effect on the word boundary hypothesisation. It is likely that the number of word boundaries detected will increase since those word boundaries spanned by the newly added clues will now be detected. However the

number of incorrect hypotheses will also increase since these clues occurring in word-internal position will cause errors.

The results of word boundary hypothesisation using the new lexical clues (of types $CV^{+}C$ and $VC^{+}V$ together) are shown in Table 5.4 for various input error percentages. The lexical clues included all word boundary phoneme sequences which occurred in only one word in the dictionary. As predicted, the number of word boundaries detected and also the number of incorrect word boundary hypotheses increased compared to the earlier ones (shown in Table 5.2). An examination of the incorrect hypotheses showed that the increase was due to the addition of one or two sequences to the clues, i.e., those sequences which occur in one or two words in the dictionary but the words occur frequently in the text. For example, the sequence *ae:* occurred in only two words in the dictionary but the word *gae:* containing the sequence, occurred several times in the text and all these occurrences resulted in incorrect hypotheses.

The results are also shown in terms of the Hit rate, Correctness and Improvement. It can be seen that while the Hit rate increased, there is a small drop in the Correctness and Improvement, indicating that the overall performance of the clues deteriorated. This is illustrated in Fig.5.4, where the Hit rate, Correctness and the Improvement using the new lexical clues are compared against the Hit rate, Correctness and Improvement obtained using lexical clues which did not contain any word-internal sequences.

From the above, one can conclude that the number of word boundaries detected can be increased by adding the infrequent word-internal phoneme sequences to the lexical clues. However to control the number of incorrect hypotheses, their addition may have to be done selectively considering not only the frequency of

	% Error in input text					
	0	10	20	30	40	50
WBs detected	7678	7676	7749	7825	7883	7894
Correct hyp	7678	7676	7749	7825	7883	7894
Incorrect hyp	1914	2479	2950	3497	3861	4.323
Hit rate	72%	72%	72%	73%	73%	74%
Correctness	80%	76%	72%	69%	67%	65%
Improvement	1.7	1.65	1.6	1.55	1.5	1.4

Table 5.4 Results of word boundary hypothesisation using lexical clues containing some infreqent word internal phoneme sequences on erroneous input. The above results are for consonant sequences of type $vc^{+}v$ and $cv^{+}c$ together. The results are shown for various input error percentages(0,10,20,30,40 and 50%). The input text contained 10,737 word boundaries and 39,713 word internal positions.

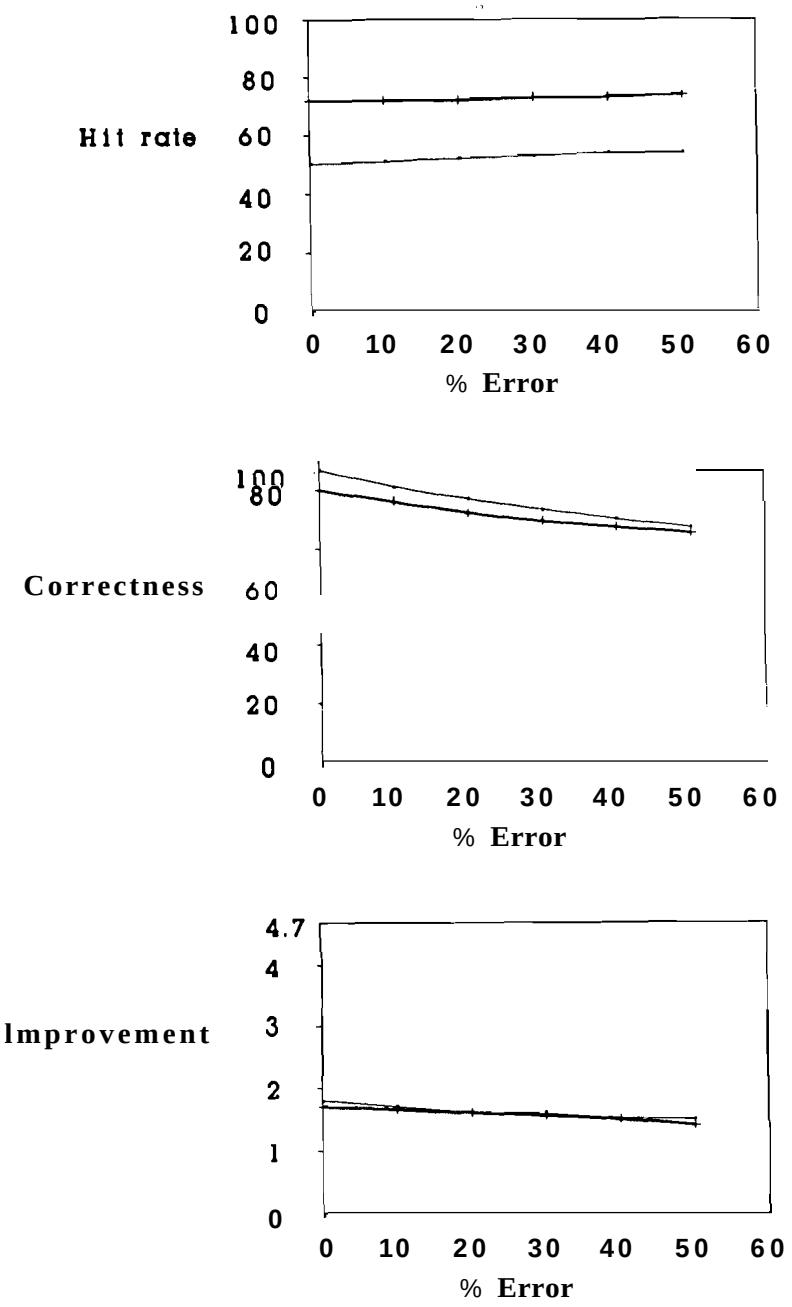


Fig.5.4 A comparison of the performances of the lexical clues for the cases, (i) when lexical clues contained no word internal sequences (indicated by thin line in the figure), and (ii) when lexical clues contained some infrequent word internal sequences (indicated by thick line in the figure). The results are compared in terms of the Hit rate, Correctness and the Improvement. It can be seen that the addition of infrequent word internal sequences increases the Hit rate significantly, with only a marginal reduction in the Correctness.

occurrence of the sequence but also the frequency of occurrence of the words containing the sequence.

5.6 Lexical clues from a small dictionary

The results of the previous section showed that Hit rate can be increased by increasing the number of lexical clues. However, to minimise the increase in the number of incorrect hypotheses, one needs to carefully choose the clues. One way of doing this, is to use a small dictionary which contains all frequently occurring words and extract lexical clues from it. Clues can be extracted in this fashion, especially if the speech recognition system containing the word boundary hypothesiser uses only a limited vocabulary (about 1000 words).

To study the performance of the lexical clues extracted from a small dictionary, a dictionary of about 2500 words containing frequently occurring Hindi words was used. The dictionary also included all the words that appeared in the input text including many inflected words. The lexical clues were extracted using the dictionary and they were used to hypothesise word boundaries. Results of the word boundary hypothesisation using clues of types CV^+C and VC^+V for various input error percentages are shown in Table 5.5.

The results show that the number of word boundaries detected increased as predicted. However, the number of errors also increased and the Correctness remained almost the same. The results are shown in terms of the Hit rate, Correctness and Improvement in Fig.5.5, where they are compared against the results obtained using lexical clues extracted from a large dictionary.

These results can be predicted based on the formula developed in the earlier section (section 5.4.1). The formula for the number of errors is given by $N * P(y \in S) * 1 - (1-p)^L)$, where N represents the total number of word-internal phoneme sequences in the text. When a small dictionary is used, the number of word-internal sequences

	% Error in input text					
	0	10	20	30	40	50
WBs detected	8399	8360	8358	8347	8331	8322
Correct hyp	8399	8360	8358	8347	8331	8322
Incorrect hyp	0	1713	3087	4162	5107	6029
Hit rate	78%	78%	78%	78%	78%	77%
Correctness	100%	83%	73%	66%	62%	58%
Improvement	4.7	1.8	1.6	1.6	1.5	1.45

Table 5.5 Results of word boundary hypothesisation using lexical clues extracted using a small dictionary of 2500 words on erroneous input. The above results are for consonant sequences of type $VC^{+}V$ and $CV^{+}C$ together. The results are shown for various input error percentages(0,10,20,30,40 and 50%). The input text contained 10,737 word boundaries and 39,713 word internal positions.

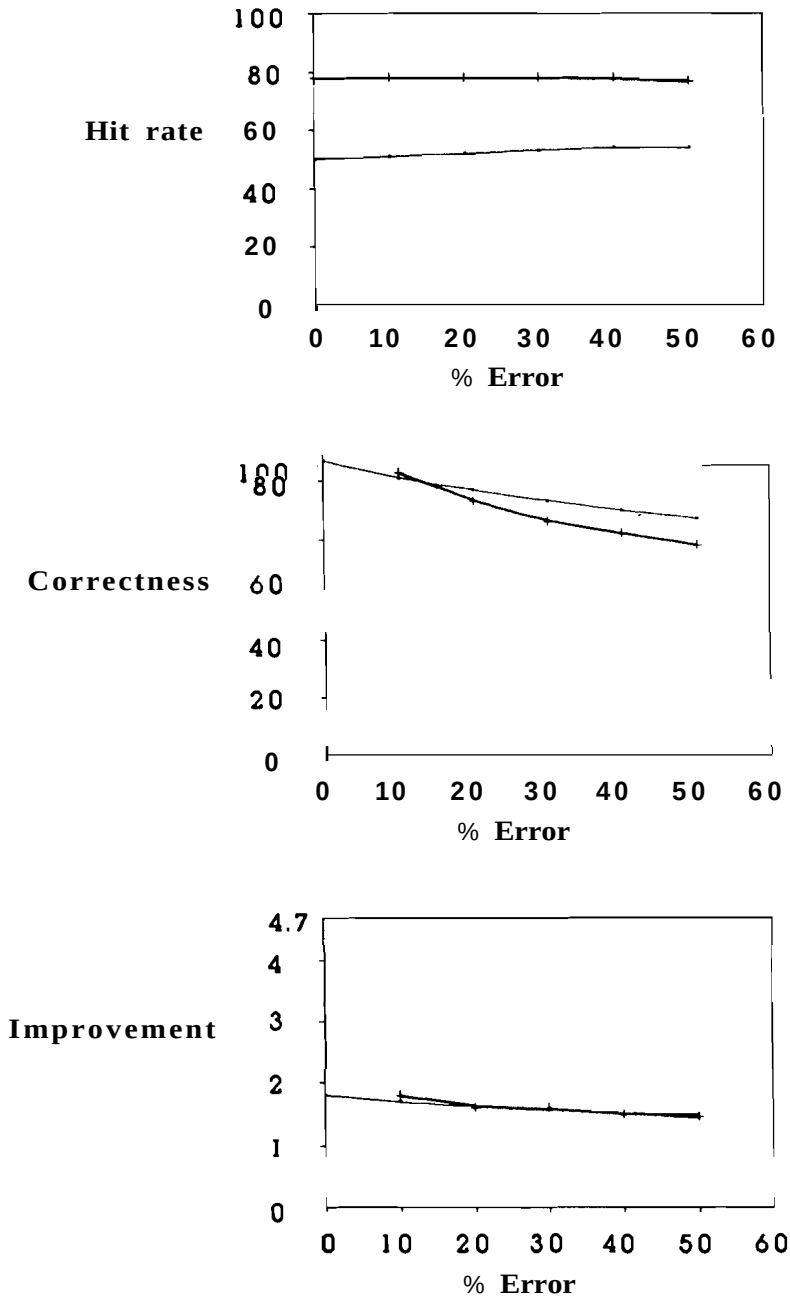


Fig.5.5 A comparison of the performances of the lexical clues for the cases, (i) when lexical clues are extracted from a large dictionary of 30,000 words (indicated by thin line in the figure), and (ii) when lexical clues are extracted using a small dictionary of 2,500 words (indicated by thick line in the figure). The results are compared in terms of the Hit rate, Correctness and the Improvement. It can be seen that the clues extracted using a small dictionary have a larger Hit rate but a smaller Correctness.

will be smaller and hence S and $P(y \in S)$ will be larger than the case for a large dictionary. Hence the number of errors in word boundary hypotheses will also be more though the growth may be the same.

5.7 **Locating word boundaries from the hypotheses produced by the lexical clues**

So far, results were presented to show that lexical clues in the form of phoneme sequence constraints can be used to hypothesise the presence of a word boundary within a sequence of phonemes. However, the exact location of the word boundary within the sequence was not identified. In some cases it may be possible to precisely locate the word boundary due to other constraints. For example, in the sequence ***ktk*** which was one of the lexical clues, one can place the word boundary as ***k#tk*** or as ***kt#k***, # indicating the word boundary. However ***tk*** is not a valid word initial consonant sequence and hence the word boundary can be exactly placed as ***kt#k***. However such constraints are available only in a few cases. For a majority of the clues, location of the word boundary will not be possible from the lexical constraints alone. However, it is possible that an examination of the distribution of the word boundaries within the hypothesised sequences may provide some clues to locate the word boundary. Hence it was decided to study the distribution of word boundaries within the hypothesised sequences.

(a) Simple sequences of vowels (V^+)

Most of the hypotheses produced using constraints on simple vowel sequences involve two vowels only. Hence in this case it was possible to locate the word boundary precisely. In about 10% of the hypotheses, vowel sequences containing three vowels occurred, and hence in these cases it was not be possible to locate the word boundary exactly.

(b) Simple consonant sequences (C^+)

Unlike the case of simple vowel sequences, most of the word boundary hypotheses produced by the constraints on sequences of consonants involve three or more consonants. Only about one-third of the hypotheses involve two consonants and only in these cases the word boundaries could be located precisely. The remaining hypotheses involve mainly three consonants. In such hypotheses the word boundary may lie between the first two consonants (as C#CC) or between the last two consonants (as CC#C). However we observed the latter to be more frequent than the former (85% of the cases). Hence in absence of other knowledge, one may place the boundary between the last two consonants in a three consonant sequence.

(c) Sequences of vowels of the types CV^+ and V^+C

The hypotheses produced by sequences of the types CV^+ and V^+C consist of mainly three phonemes, and are in the forms CVV and VVC, respectively. In these hypotheses the word boundary can be in two places: (i) between the vowels or, (ii) between the consonant and the vowel. In our studies, we observed that in the hypotheses produced by sequences of type CV^+, the word boundary was between the vowels in 99% of the cases. Hence one can safely place the word boundary between the vowels in the hypotheses produced by such sequences. However for the hypotheses produced by sequences of type V^+C the word boundary was between the consonant and the vowel in 26% of the cases and between the vowels in the remaining. Obviously it is not possible to precisely locate the word boundary in the hypotheses produced by these sequences, though a strategy of placing the word boundary between the vowels yields correct results in about 74% of the cases.

(d) Sequences of consonants of the type VC^+ and C^+V

Unlike the vowel sequences of the types CV^+ and V^+C, the hypotheses produced by the consonant sequences of the types VC^+ and C^+V contain a significant number of hypotheses with four or more phonemes. In such cases, even though the

word boundary is within the consonant sequence, it is not possible to locate it exactly. However nearly 75% of the hypotheses produced contain only three phonemes (either in the form **CCV** or **VCC),** and in some of these cases it may be possible to locate the boundary. It was observed that in the hypotheses produced by **CCV** type of sequences, the word boundary was between the consonants in 97% of the cases and between the second consonant and the vowel in only 2% of the cases. Thus for hypotheses produced by **CCV** sequences, the word boundary can be placed between the consonants. In the hypotheses produced by the **VCC** sequences, in **15%** of the cases, the boundary was between the vowel and the consonant and between the consonants in the remaining. Thus a strategy of placing the word boundary between the consonants seems appropriate for both **CCV** and **VCC** types of sequences.

(e) Sequences of type $\mathbf{CV^+C}$ and $\mathbf{VC^+V}$

In longer sequences of types CV^+C and VC^+V the word boundary may be located in several possible places. In the hypotheses produced by sequences of type CV^+C it was observed that nearly 66% of the word boundaries lie at the **VC** junction, i.e., between the last vowel and the last consonant. Another **30%** are located between the vowels and very few at the **CV** junction. In the hypotheses produced by the VC^+V sequences, 60% of the word boundaries occurred within the consonant sequence. Another 37% occurred at the **VC** junction, i.e., between the first vowel and the first consonant in the sequence. Very few occurred at the **CV** junction. Thus one may not be able to place the word boundary accurately in the hypotheses produced by these clues. However, one can narrow down the location by ignoring the **CV** junction for both types of clues.

The distributions of word boundaries in the word boundary hypotheses produced by the lexical clues were also examined by varying the input error

percentages. However, the results showed no significant changes.

From the above it can be seen that the word boundaries can be located accurately in many hypotheses produced by the constraints on sequences. However for long sequences of types CV^+C and VC^+V it was not possible to locate the word boundaries precisely. It is interesting to note that in all the above cases the VC junction contained many more word boundaries compared to the CV junction.

5.8 Summary and Conclusions

In this chapter, use of lexical knowledge in the form of phoneme sequence constraints in hypothesising word boundaries was examined. A number of studies were made using different types of sequences. The effect of input errors on the word boundary hypotheses was also examined and a formula was developed to predict the number of wrong hypotheses produced due to errors in the input. It was shown that reducing the dictionary size does not improve the Correctness of the clues. Studies were also performed to locate the word boundary position exactly in the hypotheses produced using the clues. The results show that for shorter sequences it is possible to place the word boundary accurately within the clue.

The following conclusions can be drawn based on our studies reported in this chapter:

1. Lexical clues in the form of phoneme sequence constraints can be used to hypothesise word boundaries in texts.
2. The lexical clues are also useful in hypothesising word boundaries especially when the input contains errors, as in the symbol sequence produced by a speech signal-to-symbol converter. However, the performance of the clues deteriorates gradually with increasing input errors.
3. Reducing the vocabulary (dictionary size) improves the performance of these clues.
4. It is possible to predict the location of the word boundary within the clue for most

clues except for clues of types $CV^{+}C$ and $VC^{+}V$.

The result. of the studies of this chapter clearly establish the utility of the clues based on the lexical knowledge. However, it is also to be noted that these clues are sensitive to errors in the input. The results of the earlier chapter also showed that clues based on the language knowledge such as syntax and semantics are also susceptible to input errors. These errors are caused by the inaccuracies in the speech signal-to-symbol conversion. Hence one needs to identify clues which can be applied before the signal-to-symbol conversion itself. In this context, one can explore other speech related knowledge sources such as prosody and acoustic-phonetics and identify word boundary clues based on them, which can be directly applied on the speech signal itself thereby avoiding the errors in the speech signal-to-symbol conversion. In the succeeding chapters, some prosodic and acoustic-phonetic clues for word boundary hypothesisation are described.

Chapter 6

WORD BOUNDARY CLUES BASED ON THE PROSODIC KNOWLEDGE

6.1 Introduction

Several studies have established the relation between the various prosodic features and the sentence structure in a language. For example, in languages like English, position of stress can change the type of a word, whether it is a noun or a verb. The pitch contour of a sentence indicates the type of the sentence, assertive or interrogative. Similarly boundaries between the major syntactic units (such as phrases or clauses) in a sentence are also indicated by pitch variations. Pauses or long silences in speech indicate some word boundaries, usually the boundaries between major syntactic units of the sentence. Prepausal lengthening of vowels and long interstress intervals can also be used to detect some word boundaries. In view of these results, one can expect the prosody to provide clues to detect word boundaries.

This chapter describes studies on the use of prosodic features as clues for hypothesising word boundaries in continuous speech. The prosodic features considered were pause, duration, and pitch. Three studies are reported, each focussing on the application of a particular prosodic feature to hypothesise word boundaries. The studies are described in the following sections. In section 6.2, the study on the use of pause for word boundary hypothesisation is described. In section 6.3, studies on the use of duration of a vowel as a clue to its position in a word are described. In section 6.4, the application of changes in pitch in detecting word boundaries is discussed. In section 6.5, a simple word boundary hypothesisation algorithm is described which combines the pause, duration and pitch clues. In section 6.6, location of the word boundary position in the hypotheses produced by the prosodic clues is discussed. The summary of the work is presented in section 6.7.

6.2 Word boundary hypothesisation using pause

Pauses in speech, though small in number, are the simplest and often the easiest clues to detect word boundaries. Pauses in continuous speech are detectdd by looking for long silence regions. Since some speech sounds such as unvoiced stops also contain silence regions, a duration threshold is used to discriminate between pauses and other silences. The threshold chosen should be sufficiently longer than the longest speech sound that contains silence. Usually a value around 250 msec. is used [Grosjean 1980]. The algorithm to detect pauses in speech is given below.

Algorithm 6.1:

1. Identify the silence regions in speech using energy and pitch.
2. Hypothesise a silence region as a pause, if its duration is longer than 250 msec.

Pause detection was performed on a speech data of 110 utterances, consisting of a text of 10 sentences uttered by 11 speakers. On this data, the above algorithm was applied and pauses were detected. Each detected pause was hypothesised as a word boundary. The results of the word boundary hypothesisation using pause are shown in Table 6.1. It can be seen that all the pauses detected correspond to word boundaries.

While the above results show that pause is a reliable clue to word boundaries, they also show that its utility is limited because of the small number of word boundaries detected (less than 2 word boundaries per sentence). Moreover, as the speaking rate increased, the number of pauses in the utterances reduced. This is illustrated in the table where the speakers are ordered by their speaking rate with the lowest corresponding to speaker 1 and the highest to speaker 11. Hence additional prosodic clues are to be used to increase the number of word boundaries.

6 3 Word-final vowel hypothesisation using duration

Speaker	Correct WBs	Incorrect WBs	Hit rate	Correctness	Improvement
1	29	0	19%	100%	4.7
2	27	0	18%	100%	4.7
3	23	0	15%	100%	4.7
4	21	0	14%	100%	4.7
5	21	0	14%	100%	4.7
6	18	0	12%	100%	4.7
7	19	0	13%	100%	4.7
8	19	0	13%	100%	4.7
9	18	0	12%	100%	4.7
10	16	0	11%	100%	4.7
11	17	0	11%	100%	4.7

Table-6.1 Results of word boundary hypothesisation using pause. A silence longer than 300 msec. is considered as a pause. Results are shown for 11 speakers.

Several studies on English speech showed that duration can be used to hypothesis: some of the word boundaries in speech. Lea [Lea, 1980] showed that interstress intervals can be used to detect some word boundaries. Lengthening of a vowel can also be used as a clue to word boundaries[Crystal and House 1988]. For Hindi, we have found that a simple algorithm which classifies all long vowels as word-final vowels can lead to a good word boundary detection.

The proposed algorithm is based on the following features of Hindi: (i) In Hindi, very few words end in a short vowel, and (ii) In any Hindi text, vowels occur twice as often as consonants in the position before a word boundary. The feature, (i) was verified by examining a large Hindi dictionary containing nearly 31,000 words. Of these, 13,479 words ended in vowels, in which 12,628 (or 94%) ended in long vowels. Similarly, (ii) was verified using a Hindi text of nearly 10,000 sentences which contained 143,578 words of which 118,016 (or 82%) ended in vowels. This is because, in any text case markers and other function words form nearly 40% of the total words and most of these end in vowels. For the remaining words, the vowels and consonants occur roughly in equal numbers before a word boundary. Thus, in the entire text, one can expect that about 70% of the word boundaries will be preceded by vowels and the rest 30% will be preceded by consonants. Since from (i) above, almost all vowels preceding word boundaries must be long vowels, one can expect that about 70% of the word boundaries will be preceded by long vowels. Thus a simple classification of vowels based on length will detect nearly 70% of the word boundaries. In addition, a recent study [Rajesh Kumar 1990] showed that vowels occurring in word-final position, i.e., last vowels in the word which may be succeeded by a consonant, are longer than the same vowels occurring in word-internal position. Hence some word-final vowels may also be detected by the proposed algorithm.

However long vowels can also occur in word-internal positions. Such

occurrences will lead to errors in the word boundary hypotheses. But it was observed that word-internilly short vowels occur more often than long vowels. Hence a long/short vowel classification will hypothesise more word boundaries than word-internal vowels.

An estimate of the performance of the above method can be obtained by studying the distribution of vowels in large texts. Using a text of 10,000 sentences containing 143,578 words which had 244,082 vowels, it was found that long vowels occurred 143,804 times in the text. Among these, long vowels in word-final position numbered 109,842. Hence detection of long vowels will hypothesise nearly 77% of the word boundaries correctly with incorrect hypotheses around 25%. Similar results can be expected from other texts.

The algorithm for word boundary hypothesisation [Ramana Rao 1992b] using duration is given below.

Algorithm 6.2 Word boundary hwothesisation using duration

1. Classify the given vowel into short/long vowel.
2. If it is classified as a long vowel, hypothesise a word boundary after the vowel.

This algorithm was used to hypothesise word boundaries in a speech data of 110 utterances consisting of 10 sentences uttered by 11 speakers. The speech data contained a total of 2,600 vowels. These vowels were segmented manually using visual and audio clues. On these vowel data, the above word boundary hypothesisation algorithm was applied. The results are shown in Table_6.2 for various durational thresholds. From these, one can see that the proposed clue performs well.

The results show that in addition to many word boundary vowels, a significant

Speaker (Mean dur)	WB hypotheses at various thresholds 45	55	65	75	85	95	105	115
1 (120ms)	104:134	104:133	104:124	104:109	103:85	98:73	**95:71**	78:56
2 (110ms)	104:128	104:116	102:104	96:88	87:74	**82:63**	72:53	60:48
3 (98ms)	104:123	104:112	104:92	102:74	**94:63**	85:50	73:43	56:29
4 (92ms)	102:110	99:93	91:71	**84:55**	70:43	59:34	44:24	35:13
5 (86ms)	101:109	95:88	89:63	**81:48**	65:40	52:26	40:17	33:9
6 (78ms)	98:111	93:83	**83:61**	69:45	56:32	45:23	31:12	23:8
7 (78ms)	100:98	95:84	**82:63**	63:45	54:34	40:24	29:14	22:10
8 (78ms)	103:98	91:80	**83:64**	73:48	53:23	43:24	32:15	20:8
9 (77ms)	100:92	96:75	**81:59**	69:42	53:26	39:23	29:14	22:10
10 (73ms)	92:87	**76:71**	63:53	54:35	41:26	37:14	29:8	25:5
11 (70ms)	93:88	**81:66**	67:53	55:32	39:23	29:12	21:8	17:1

(a)

Speaker	WB hypotheses (WB:WI)	Hit rate	Correctness	Improvement
1	95:71	63%	57%	2.7
2	82:63	55%	57%	2.7
3	94:63	63%	60%	2.8
4	84:55	56%	60%	2.8
5	81:48	54%	67%	3.2
6	83:61	55%	58%	2.7
7	82:63	55%	57%	2.7
8	83:64	55%	56%	2.6
9	81:59	54%	58%	2.7
10	76:71	51%	52%	2.4
11	81:66	54%	55%	2.6

(b)

Table–6.2 Results of word boundary hypothesisation using duration. In the Table, (a) shows the results in terms of the correct and incorrect word boundary hypotheses produced for various duration thresholds for 11 speakers. In (b), the results are shown in terms of Hit rate, Correctness and Improvement for a particular duration threshold. Note that duration is measured in milli seconds.

number of vowels which are not succeeded by a word boundary were also hypothesised. An examination of the errors showed that many of tbese incorrect hypotheses correspond to word-final vowels, i.e., vowels which are the last vowels in a word but which are succeeded by a consonant sequence, such as the vowel a in the word ***ko:mal***. As shown later in section 6.6, it is possible to locate the position of the word boundary from the word-final vowels. Hence, the word boundary hypothesisation was modified as word-final vowel hypothesisation algorithm given below.

Algorithm 6.3 Word-final vowel hypothesisation using duration

1. Classify the given vowel into short/long vowel.
2. If it is classified as a long vowel, hypothesise the vowel as a word-final vowel.

The word-final vowel hypothesisation algorithm was applied on a 110 sentence speech data and the results are shown in Table 6.3. The results are shown in terms of the number of word-final and word-internal vowels in the hypotheses, and also in terms of the Hit rate, Correctness and Improvement for various duration thresholds. It can be seen that a large number of word-final vowels were detected with significant Correctness (nearly 75%) and Improvement in the WF:WI distribution is also high.

It can be seen that the performance of the algorithm depends on the threshold used for short/long vowel classification. Using a smaller durational threshold led to the detection of a large number of word boundaries but with lower Correctness, whereas longer thresholds led to the detection of lesser number of word boundaries with higher Correctness. It can also be seen that longer thresholds lead to a larger Improvement. We have chosen to use a threshold for which the Correctness is greater than 75%.

It can be seen that this performance (shown in bold) occurs at different thresholds for different speakers. This is because the duration of vowels in continuous

Speaker (Mean dur)	WF hypotheses at various thresholds 45	55	65	75	85	95	105	115
1 (120ms)	150:88	150:87	149:79	147:65	142:45	136:35	**133:32**	109:24
2 (110ms)	150:82	149:71	147:59	136:47	119:41	**113:31**	100:23	84:23
3 (98ms)	150:77	149:67	147:49	137:38	**123:33**	108:26	94:22	71:13
4 (92ms)	142:70	136:56	122:40	**110:29**	90:23	77:16	58:10	43:5
5 (86ms)	145:65	133:50	119:33	**106:23**	86:19	70:8	54:3	39:3
6 (78ms)	139:70	127:49	**110:34**	93:21	75:13	60:8	40:3	30:1
7 (78ms)	138:60	131:48	**113:32**	85:23	71:17	53:11	38:5	29:3
8 (78ms)	142:59	128:43	**113:34**	99:22	69:17	55:12	41:6	26:2
9 (77ms)	137:55	128:43	**109:31**	90:21	67:12	50:12	36:7	26:6
10 (73ms)	130:49	**109:38**	89:27	77:12	58:9	47:4	35:2	30:0
11 (70ms)	128:55	**109:38**	93:27	71:16	49:13	36:5	27:2	18:0

(a)

Speaker	WB hypotheses (WB:WI)	Hit rate	Correctness	Improvement
1	133:32	89%	81%	3.3
2	113:31	75%	78%	3.2
3	123:33	82%	79%	3.2
4	110:29	73%	79%	3.2
5	106:23	71%	82%	3.3
6	110:34	73%	76%	3.1
7	113:32	75%	78%	3.2
8	113:34	75%	77%	3.1
9	109:31	73%	78%	3.2
10	109:38	73%	74%	3.0
11	109:38	73%	74%	3.0

(b)

Table 6.3 Results of word final vowel hypothesisation using duration. In (a), results are shown in terms of the correct and incorrect word boundary hypotheses produced for various duration thresholds for 11 speakers. In (b), the results are shown in terms of Hit rate, Correctness and Improvement for a particular duration threshold(shown in bold). Note that duration is measured in milli seconds.

speech depends strongly on the speaking rate. Hence a speaker independent method of selecting the threshold is needed. This threshold can be estimated based on the average vowel duration as follows:

Let us assume that the vowels can be divided into two classes, short and long, with average lengths L and 2L, respectively. Assuming a 40:60 distribution of short and long vowels (as observed in our text data), one would obtain the average length of a vowel as 1.6L. Hence the average length of a short vowel (L) is given by the average length of a vowel divided by 1.6. This itself can be used as a threshold for short/long vowel classification. However some of the short vowels will be longer than the computed average duration. In our studies, it was observed that a threshold value of 1.3L gives acceptable results. The results for this value of the threshold are shown in bold in the Table_6.2.

In the above, the duration of a vowel was measured in time units (in msec. etc.). One may also use the number of pitch peaks in the vowel as a measure of the duration of the vowel. Thus the duration of a vowel is the number of glottal pulses in that vowel. With this measure for duration, the word-final vowel hypothesisation algorithm 6.2 was again applied on the vowel data. The results are shown in Table_6.4.

A comparison of the results (Table_6.3 and Table_6.4) show that the two measures used for duration yield similar results. However the second measure seems to be slightly advantageous in that the variation in thresholds to be used seems to be less. However, for this measure of duration also, an estimate of the average duration of a vowel is needed to determine the threshold to be used.

6.4 Word-final vowel hypothesisation using pitch

The studies reported in the previous two sections showed that the prosodic features of pause and vowel duration can be used as clues to hypothesise word

Speaker (Mean dur)	WF hypotheses at various thresholds 8	9	10	11	12
1 (12)	141:44	135:30	127:23	**114:17**	98:14
2 (11)	132:37	118:32	**108:27**	95:22	73:19
3 (12)	142:36	136:39	128:33	**114:31**	107:26
4 (11)	134:52	122:41	**114:29**	102:22	97:19
5 (12)	135:58	127:44	115:33	**105:27**	99:20
6 (11)	131:50	120:40	**109:24**	99:18	88:14
7 (11)	133:50	126:41	**120:35**	111:23	101:19
8 (11)	130:43	120:32	**110:26**	98:20	81:15
9 (12)	124:51	118:44	114:39	**104:33**	92:26
10 (10)	114:37	**100:24**	85:18	77:12	67:7
11 (10)	115:37	**102:30**	89:21	76:16	67:10

(a)

Speaker	WF hypotheses (WF:WI)	Hit rate	Correctness	Improvement
1	114:17	76%	87%	3.5
2	108:27	72%	80%	3.1
3	114:31	77%	79%	3.1
4	114:29	77%	80%	3.1
5	105:27	70%	80%	3.1
6	109:24	73%	82%	3.3
7	120:35	81%	77%	3.0
8	110:26	74%	81%	3.2
9	104:33	70%	76%	3.0
10	100:24	67%	81%	3.2
11	102:30	68%	77%	3.0

(b)

Table-6.4 Results of word final vowel hypothesisation using duration. In (a), results are shown in terms of the correct and incorrect word boundary hypotheses produced for various duration thresholds for 11 speakers. In (b), the results are shown in terms of Hit rate, Correctness and Improvement for a particular duration threshold(shown in bold). Note that duration is measured in pitch cycles.

boundaries. In this section, studies on the use of the third prosodic feature of pitch frequency(F0) as a clue to word boundary hypothesisation are reported.

Recent studies on Hindi speech [Yegnanarayana, Rajendran, Rajesh Kumar, Ramachandran and Madhu Kumar 1992] suggested that every content word in continuous Hindi speech has a pitch pattern, namely the pitch frequency(F0) increases from left to right. Thus, in the sentence fragment ***narmada: nadi: ke: kina:re:*** the word ***narmada:*** will have FO increasing from left to right. Similarly the word ***nadi:*** will also have an increasing FO from left to right. This is illustrated in Fig.6.1, where the waveform and the pitch frequency are shown. Since FO in simple sentences falls from left to right, the fall will occur at the boundary between the two words. Thus by detecting such falls in FO one can detect word boundaries. On the other hand, if a function word occurs between the two content words, as in ***nadi: ke: kina:re:*** the FO peak will occur on ***i:*** in ***nadi:*** and the next valley on ***i*** in ***kina:re:***. Thus even in such cases one can detect word boundaries by detecting falls in FO. Strictly speaking, this detects only the word-final vowels. A word-final vowel hypothesisation algorithm based on the above [Madhukumar 1993; Rajendran and Yegnanarayana 1994] is given below:

Algorithm 6.4 Word-final vowel hypothesisation using FO

1. Compare the FO values of two successive vowels,
2. Hypothesise a vowel as a word-final vowel if the drop in FO from the current vowel to its next vowel is greater than a predetermined threshold.

This algorithm was used to detect word-final vowels in the 110 sentence speech data used in the earlier studies. The results are shown in Table 6.5. The results are shown in terms of the number of word-final and word-internal vowels in the hypotheses and also in terms of the Hit rate, Correctness and Improvement for various thresholds.

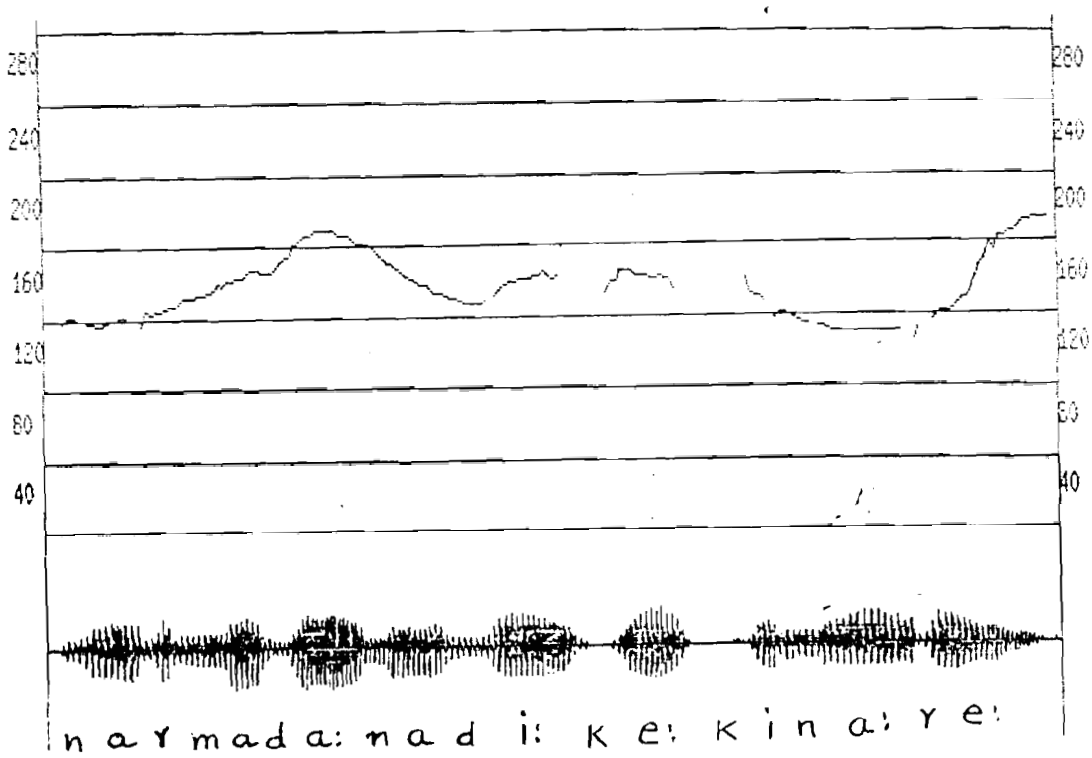


Fig.6.1 The plot of pitch frequency (FO) for the Hindi utterance **'narmada:** nadi: ke: **kina:re:'.** It can be seen that for all the content words the FO increases from left to right with maximum FO on the rightmost vowel in the word. For the function word **'ke:'** the FO lies between that of **'i:'** in **'nadi:'** and **'i'** in **'kina:re:.**

Speaker	WF hypotheses at various thresholds			
	0	2	4	6
1	111:20	109:17	98:13	85:11
2	104:22	92:19	89:16	82:13
3	102:46	94:31	79:21	69:15
4	110:40	97:28	87:21	70:12
5	108:32	99:24	94:20	84:16
6	114:19	98:15	91:13	74:11
7	115:24	102:13	98:11	88:8
8	110:23	92:12	89:11	77:10
9	107:34	87:18	72:15	58:8
10	103:35	89:28	81:25	76:23
11	107:28	88:19	80:16	71:11

(a)

Speaker	WF hypotheses (WF:WI)	Hit rate	Correctness	Improvement
1	111:20	74%	85%	3.4
2	104:22	69%	83%	3.3
3	102:46	68%	69%	2.7
4	110:40	73%	73%	2.8
5	108:32	72%	77%	3.0
6	114:19	76%	86%	3.4
7	115:24	77%	83%	3.3
8	110:23	73%	83%	3.3
9	107:34	71%	76%	3.0
10	103:35	68%	75%	2.9
11	107:28	71%	79%	3.1

(b)

Table–6.5 Results of word final vowel hypothesisation using pitch. In (a), results are shown in terms of the correct and incorrect word final vowel hypotheses (WF:WI) produced for various pitch thresholds for 11 speakers. In (b), the results are shown in terms of Hit rate, Correctness and Improvement for a pitch threshold of 0.

From the results, it can be seen that the algorithm performs well for many speakers. But for some speakers (speakers 3, 4, 5, 9 and 10), the performance is relatively poor, with the hypotheses containing a significant number of word-internal vowels. An examination of these errors revealed that in a majority of the cases, they correspond to short vowels. Thus one can improve the performance of the word-final vowel hypothesisation by using durational constraints along with the FO. This is described in the next section.

6.5 Word-final vowel hypothesisation using all prosodic clues

The previous sections showed that a significant number of word-final vowels can be detected by using the prosodic clues of pause, duration and pitch(F0). It is possible to improve the performance of the clues by applying them together. From the results of the previous section, it is clear that applying durational constraints to verify the word-final vowel hypotheses produced by the pitch clues will lead to a better performance. Hence, the word-final vowel hypothesisation algorithm using pitch was modified to include durational constraints [Ramana Rao 1992b] as given below.

Algorithm 6.5 Word-final vowel hypothesisation using FO and duration

1&2. Same as in algorithm 6.4,

3. From the word-final vowel hypotheses produced, remove all word-final vowel hypotheses whose duration is less than a duration threshold.

The above algorithm was applied on the 110 sentence speech data. A pitch threshold of 0 and a duration threshold roughly corresponding to the average short vowel duration were used. The results of the word-final vowel hypothesisation are shown in Table 6.6. It can be seen that there is a significant improvement for all the

Speaker	Hypotheses (WF:WI)	Hit rate	Correctness	Improvement
1	102:7	69%	94%	3.8
2	92:9	61%	91%	3.7
3	97:20	65%	83%	3.3
4	97:17	65%	85%	3.4
5	102:14	68%	88%	3.5
6	96:9	64%	91%	3.7
7	98:8	65%	92%	3.7
8	99:11	66%	90%	3.6
9	96:13	64%	88%	3.5
10	93:19	62%	83%	3.3
11	91:16	61%	85%	3.4

Table *6.6* Results of word final vowel hypothesisation using pitch and duration. The results are shown both in terms of WF:WI distribution and also in terms of Hit rate, Correctness and Improvement. A pitch threshold of 0 and a durational threshold of *0.66* x Avg. vowel duration are used.

speakers, with the incorrect hypotheses reduced by a factor of 2 or more, whereas the number of word boundaries detected dropped by less than 15%.

Detection of word boundaries through pause can be used to further improve the performance of the above word-final vowel hypothesisation algorithm based on pitch and duration (Algorithm 6.5). This is due to the fact that in long sentences containing several phrases/clauses, the FO contour gets reset within the sentence, mostly at major syntactic boundaries. This is usually explained by the need for the speaker to pause occasionally (possibly to take a breath), and, at such pauses, which coincide with major syntactic boundaries, the FO is reset to a high value. Thus long sentences often contain two or more smaller segments each of which shows a declining FO with FO reset to a high value after each segment. Due to this resetting of FO, the corresponding word boundaries will not be detected by our algorithm. However, these boundaries are usually followed by pauses, and, by detecting pauses one can detect them. Thus the algorithm for detecting word-final vowels using prosodic clues gets modified to the one given below.

Algorithm 6.6 Word-final vowel hypothesisation using F0, duration and pause

1,2&3. Same as in algorithm *6.5*

4. Hypothesise word boundaries using pause (this includes sentence boundaries also). Add these to the hypotheses produced FO and duration.

This algorithm was applied on our speech data and the results are shown in Table 6.7. It can be seen that more than 70% of the word boundaries are detected in all the three cases with Correctness more than 80%.

While the above modified algorithm for word boundary hypothesisation using pitch, duration and pause works well, it is difficult to explain the results, especially the

Speaker	Hypotheses (WF:WI)	Hit rate	Correctness	Improvement
1	131:7	88%	95%	3.9
2	119:9	79%	93%	3.8
3	120:20	80%	86%	3.4
4	118:17	79%	87%	3.5
5	123:14	82%	90%	3.6
6	114:9	76%	93%	3.8
7	117:8	78%	94%	3.8
8	118:11	79%	91%	3.7
9	114:13	76%	90%	3.6
10	109:19	73%	85%	3.4
11	108:16	72%	87%	3.5

Table-6.7 Results of word final vowel hypothesisation using pitch, duration and pause. The results are shown in terms of WF:WI distribution and also in terms of Hit rate, Correctness and Improvement. A pitch threshold of 0, a durational threshold of 0.66 x Avg. vowel duration and a 300msec. silence for pause are used.

durational effects. One possible explanation is to assume that the fall in F0 from a peak to a valley that occurs after a word boundary needs a minimum duration. Thus if the vowel after a word boundary is a short one, the fall will not completely occur in that vowel but will continue into the next vowel, and hence the short vowel will have a larger FO compared to its next vowel even if they are in the same word. This explanation is also supported by our observations that in most of these errors the difference in FO between the word initial short vowel and its succeeding vowel is quite small.

Another feature relates to the effect of speaking rate. it can be seen that the performance of the algorithm is poorer for speakers with high speaking rate. it is possible that at high speaking rates the region between a FO valley and the next peak may contain more than one content word, possibly a larger syntactic unit such as a phrase or a clause as in English speech [Lea 1980]. However this is only a conjecture and our results are not sufficient to support it.

6.6 Location of word boundaries from word-final vowels

In the studies described above it was found that the prosodic clues are useful to detect a number of word-final vowels. However, from these word-final vowel positions, one still needs to detect the word boundary location as there may be some consonants between the word-final vowel and the next vowel. For example, consider the sequence '...$V_1C_1C_2V_2$...', in which there are two consonants between the word-final vowel V_1 and the next vowel V_2. Thus the word boundary can be placed at any one of the three places: between V_1 and C_1, between C_1 and C_2 and between C_2 and V_2. However, the prosodic clues used in the above studies cannot select the correct word boundary location. Thus one needs additional clues to perform this. It is possible that language features may aid in this. Hence studies were performed to find the word boundary from

information of word-final vowel.

In our studies, various possible word boundary locations were considered in the phoneme sequence between a word-final vowel and the following vowel. Since the phoneme sequence between a word-final vowel and the next vowei is of the form $V_1C^*V_2$, the problem is to find the location of the word boundary within this sequence. There are mainly four possibilities: (i) there is no consonant between the vowels, i.e., of type V_1V_2, (ii) a single consonant between the vowels, i.e., of type $V_1C_1V_2$, (iii) two consonants between the vowels, i.e., of type $V_1C_1C_2V_2$, (iv) three consonants between the vowels, i.e., of type $V_1C_1C_2C_3V_2$. Sequences involving four or more consonants can be neglected, as they occur only occasionally.

For each of these four types of sequences, the number of times each possible word boundary location appears in a large text containing 143,578 word boundaries is obtained. These are shown below:

(i) Type of the sequence.= V_1V_2.

possible locations (no. of occurrences) are

$V_1\#V_2$ (11978)

(ii) Type of the sequence = $V_1C_1V_2$.

possible locations (no. of occurrences) are

$V_1\#C_1V_2$ (64826) and $V_1C_1\#V_2$ (5336).

(iii) Type of the sequence = $V_1C_1C_2V_2$.

possible locations (no. of occurrences) are

$V_1\#C_1C_2V_2$ (2835), $V_1C_1\#C_2V_2$ (46550) and $V_1C_1C_2\#V_2$ (522).

(iv) Type of the sequence = $V_1C_1C_2C_3V_2$.

possible locations (no. of occurrences) are

$V_1\#C_1C_2C_3V_2$ (27), $V_1C_1\#C_2C_3V_2$ (1230), $V_1C_1C_2\#C_3V_2$ (407) and $V_1C_1C_2C_3\#V_2$ (0).

From these results, it can be seen that the following simple strategy will detect many word boundaries.

Algorithm 6.7 Location of word boundaries from word-final vowels

1. If the sequence is of type V_1V_2, then place the word boundary between the vowels.
2. If there are some consonants between the vowels, then place the word boundary before the last consonant.

The above algorithm detects a total of 123,761 word boundaries correctly out of a total of 133,516 word boundaries (excluding the sentence boundaries) in the text. The number of wrong word boundary placements are 9644. Thus the algorithm correctly places the word boundary in 92% of the cases and produces only 7.5% errors.

The above word boundary location algorithm was applied on the word-final vowels hypothesised using the prosodic features of pitch, duration and pause (results of algorithm 6.6). The results in terms of the number of correct and incorrect word boundary hypotheses and also in terms of the Hit rate, Correctness and Improvement are shown in Table 6.8. It can be seen that more than two-thirds of the word boundaries were detected with Correctness more than 80%.

6.7 Summary and Conclusions

In this chapter, the use of the three prosodic features of pause, duration, and pitch were examined for word boundary hypothesisation. **A** word boundary hypothesisation algorithm was proposed using duration which performed quite well. In addition a modification was proposed to the word boundary hypothesisation algorithm using pitch by adding durational constraints, resulting in a nearly threefold reduction in the errors with only a small reduction in the word boundaries detected. Thus it was

Speaker	Hypotheses (WB: WB)	Hit rate	correctness	Improvement
1	128:10	85%	93%	4.4
2	115:13	77%	90%	4.2
3	116:24	77%	83%	3.9
4	115:20	77%	85%	4.0
5	119:18	80%	87%	4.1
6	111:12	74%	90%	4.2
7	114:11	76%	91%	4.3
8	114:15	76%	88%	4.1
9	112:15	75%	88%	4.1
10	106:22	71%	83%	3.9
11	104:20	69%	84%	3.9

Table_6.8 Results of word boundary detection using pitch, duration and pause. The results are shown in terms of WB:WI distribution and also in terms of Hit rate, Correctness and Improvement. A pitch threshold of 0, a durational threshold of 0.66 x Avg. vowel duration and a 300msec. silence for pause are used.

shown that the prosodic features of pitch and duration can significantly aid in detecting word boundaries. To detect major syntactic boundaries which were not detected by this algorithm, a further modification was made to it making use of pauses to detect such word boundaries. This algorithm combining pitch, duration and pause performed well in detecting most of the word-final vowels and with very few false alarms. However, these clues detected only word-final vowels and not the precise location of word boundaries. Hence a study was made to locate the position of the word boundary from the word-final vowel location. It was shown that by using a simple algorithm, nearly 92% of the word boundaries can be placed correctly, given the position of the word-final vowels. Using this algorithm on the word-final vowels hypothesised by the three prosodic clues together, resulted in the detection of 72% of the word boundaries with Correctness more than 80%.

From the above studies, the following conclusions can be drawn:

1. The prosodic features of duration and pitch can detect many word boundaries in continuous speech,

2. Additional boundaries corresponding to major syntactic boundaries, can be detected using pauses.

3. A combination of the three prosodic features detects a large number of word boundaries with a large Correctness.

In all the knowledge sources explored so far, prosody seems to perform best in word boundary hypothesisation. Results showed that nearly 70% of the word boundaries can be detected with incorrect hypotheses less than 15%. However, there are some limitations to the prosodic clues. A major limitation is with respect to the speech of nonnative speakers which is explained below.

Prosody varies from language to language, and hence the prosodic clues developed for word boundary hypothesisation in one language may not be applicable

for another. For nonnative speakers, the prosody used in their speech is that of their first language. Hence the prosodic clues developed using native speakers may not be applicable for them. For example, English is spoken in many countries across the world. For a large number of such speakers, it is only a second language, and for these the prosody differs from that of English. Thus while prosodic clues may work very well for a native speaker, for nonnative speakers they may fail.

So far three knowledge sources were examined and some clues were proposed for word boundary hypothesisation based on them. Of these, lexical and to a lesser extent language clues were found to be affected by the signal-to-symbol conversion errors. Prosodic clues are applicable only for native speakers. Thus all the clues examined till now are language specific. Hence there is a need to find clues which can be applied across languages. Such clues should make use of speech features alone to hypothesise word boundaries. Clues based on acoustic-phonetic knowledge satisfy this. Hence we discuss in the next chapter studies to identify some word boundary clues which are based on the acoustic-phonetic knowledge.

Chapter 7

WORD BOUNDARY CLUES BASED ON ,THE ACOUSTIC-PHONETIC KNOWLEDGE

7.1 Introduction

Studies reported in this chapter are on the application of spectral clues based on the acoustic-phonetic knowledge for hypothesising word boundaries in continuous speech. The proposed clues are based on the relationship between the speech production mechanism and the spectrum of the sound produced. The idea is to find the differences between the productions of sounds followed by a word boundary and sounds which are not. These differences in speech production are related to the speech spectrum using the acoustic-phonetic knowledge to obtain the spectral changes that occur at word boundaries. These spectral changes are then used to hypothesise word boundaries in continuous speech.

Two clues were examined in our studies. The first clue is based on the changes in the vocal tract configuration that occur at a vowel-consonant boundary and it uses the changes in the first formant(F1) frequency to hypothesise the word boundaries. The second clue for word boundary hypothesisation is based on a strong/weak vowel classification, and it uses the changes in the energy of F1. The clues aim at detecting the vowels preceding the word boundaries. The reason for limiting the clues for vowels only is that the properties of vowels change slowly and hence their spectra can be estimated reliably. Moreover, as mentioned in the earlier chapter, many words in a Hindi text end in vowels. Analysis of a text of nearly 10000 sentences containing 143,578 words showed that nearly 82% of the words in the text end in vowels. Thus if one detects all vowels preceding word boundaries then nearly three-fourths of the word boundaries will be found.

The chapter is organised as follows: In section 7.2, a study on the use of first

formant frequency(F1) change as a clue for detecting vowels preceding word boundaries, is described. In section 7.3, a study on the use of first formant energy change as a clue for detecting word-final vowels, is described. In section 7.4, these studies are summarised and some conclusions drawn from them are discussed.

72 Word boundary hypothesisation using first formant(F1) frequency

In this section, a technique for word boundary hypothesisation based on a spectral clue, namely, change in F1 position, is described. In the proposed technique, the spectral changes in a vowel preceding a word boundary are examined in terms of changes in the formants. In particular, the changes in the vocal tract configuration at a vowel-consonant(VC) juncture are expressed in terms of changes in the first formant(F1) frequency. Based on these, an algorithm to hypothesise word boundaries which makes use of changes in F1 position was developed. This is described in the following.

7.2.1 Algorithm for word boundary hypothesisation using changes in F1 position

The idea behind the proposed word boundary hypothesisation algorithm is based on the following argument: Consider a sequence of phonemes P_1P_2. Now if one can find a clue to differentiate between this sequence and the sequence $P_1\#P_2$ (# representing a word boundary), i.e., between the case when the phoneme sequence did not contain a word boundary and the case when the phoneme sequence contained a word boundary, then one can locate word boundaries. Obviously, the clue will depend on the types of the phonemes P_1 and P_2.

Now consider the various types of phoneme sequences that are possible across word boundaries. Considering only the classes of vowels(V) and consonants(C) four types are possible (P_1P_2 = VC, CV, VV, CC). However in our analysis of a large Hindi text, it was observed that in word initial position, more than 90% of the phonemes are

consonants. Thus one can consider only the cases where P_2 is a consonant and still detect 90% of the word boundaries in a text. Now P_1 can be either a vowel or a consonant. However, as mentioned earlier, in a Hindi text, nearly 80% of the word boundaries are preceded by vowels. Also it is easier to study the spectra of the vowels than those of the consonants. Hence the study was restricted to the case where P_1P_2 is of VC type. In other words, the aim is to find a technique to differentiate between the two cases of VC and V#C and use it to hypothesise word boundaries. Depending on the text, one can hope to detect around 70% of the word boundaries in the text.

Consider a VC sequence in continuous speech. Now, vowels are produced by keeping the vocal tract open for free flow of air and the consonants are produced by obstructing the flow of air. Hence in a VC juncture, the vocal tract will be changing its configuration from an open position (corresponding to the vowel) to a closed or a partially closed position (corresponding to the consonant). These changes in the vocal tract will correspondingly reflect as changes in the spectrum; in particular, as changes in the formant locations. Since the first formant F1 is proportional to the opening of the vocal tract, at the VC juncture F1 must decrease. Thus in any vowel preceding a consonant, the tail portion of the vowel would show a decreasing F1. However if there is a word boundary between the vowel and the consonant(V#C), depending on the influence of the consonant on the vowel across the word boundary, one may find the F1 to be constant or decrease by a smaller amount. Hence by selecting a proper threshold, one may be able to differentiate vowels preceding word boundaries from vowels which do not precede a word boundary . An algorithm for word boundary hypothesisation based on this idea is given below.

Algorithm 7.1 Word boundary hypothesisation using changes in F1 position

1. Select two frames of appropriate size in the tail portion of the vowel and obtain their

spectra.

2. Compare the F1 position in the two spectra. If F1 does not drop below a threshold, hypothesise a word boundary after the vowel.

7.2.2 Results of word boundary hypothesisation using changes in F1 position

The above algorithm was applied on the 2,600 vowel speech data spoken by 11 speakers which was described earlier. For each vowel in the data, two frames of size 256 samples (128 samples, for short vowels) were chosen from the tail portion. The spectra for these frames were computed using 16th order LP analysis and from these spectra, the first formant location was obtained. A word boundary is hypothesised after the vowel if the drop in F1 between the frames is less than a specified threshold. The results of word boundary hypothesisation are shown in Table 7.1. The results are shown in terms of the number of the vowels immediately preceding word boundaries(WB) and the number of vowels which do not precede a word boundary(WB) and also in terms of the Hit rate, Correctness and Improvement. Note that the set WB includes word-internal vowels and also word-final vowels which are followed by a consonant. Various thresholds were used for the drop in F1 and the corresponding results are shown in the table. In the table the threshold for the drop in F1 is shown in steps of approximately 20 Hz. For example, a threshold of 2 corresponds to a drop of about 40 Hz in F1.

From the results, one can observe that for all the speakers there is a considerable improvement in the word boundary(WB) to nonboundary(WB) vowel ratio. For example, a simple calculation shows that for a threshold of 1 (=20 Hz), for speaker 1, nearly 68% of the word boundary vowels in the input speech data are detected, but the number of nonboundary vowels hypothesised are only 31%. Similar

Speaker	WB hypotheses(WB:WB) at various thresholds				
	0	1	2	3	4
1	32:25	69:43	82:59	88:75	93:95
2	46:28	63:50	78:72	83:90	84:100
3	34:23	52:38	63:58	78:74	86:91
4	45:30	68:56	78:72	85:88	91:102
5	46:31	65:66	78:93	84:106	90:116
6	31:21	51:41	64:55	76:70	81:84
7	38:35	50:40	68:55	79:70	87:84
8	36:47	52:62	68:82	78:97	85:108
9	41:33	56:56	73:71	77:81	84:96
10	28:29	45:45	58:60	66:75	76:80
11	30:25	46:37	59:53	73:67	84:78

(a)

Speaker	WB hypotheses (WB:WB)	Hit rate	Correctness	Improvement
1	69:43	46%	62%	2.9
2	63:50	42%	56%	2.6
3	52:38	35%	58%	2.7
4	68:56	45%	55%	2.6
5	65:66	43%	50%	2.4
6	51:41	34%	55%	2.6
7	50:40	33%	56%	2.6
8	52:62	34%	46%	2.2
9	56:56	37%	50%	2.4
10	45:45	30%	50%	2.4
11	46:37	31%	55%	2.6

(b)

Table-7.1 Results of word boundary hypothesisation using changes in F1 position. In (a) results are shown in terms of vowels preceding word boundaries(WB) and vowels not preceding word boundaries(WB) which correspond to the correct and incorrect hypotheses. In (b) results are shown in terms of Hit rate, Correctness and Improvement for a F1 threshold of 1(=20Hz).

performances are observed for the other speakers also. Another observation is that as the threshold for the drop in F1 is increased, the number of word boundaries detected increases, but the number of nonboundary vowels increases faster. This is illustrated in the results by the changes in Hit rate, Correctness and Improvement. It can be seen that an increase in the threshold value leads to an increase in Hit rate and a decrease in Correctness. However the overall performance of the clue deteriorates as shown by the decrease in the Improvement factor.

Another interesting observation made relates to the incorrect hypotheses. An analysis of these showed that a large number of the incorrect hypotheses correspond to the word-final vowels, i.e., vowels which are the final vowels in a word and which are succeeded by a consonant sequence, as a in the word *ko:mal*. Hence one can view the word boundary hypothesisation algorithm as an algorithm that detects the word-final vowels (as shown below in Algorithm 7.2), and the earlier results can be represented in terms of the distribution of word-final vowels and word-internal vowels, as shown in Table 7.2. Note that in the earlier results (shown in Table 7.1), the class of nonboundary vowels(WB) included some of the word-final vowels and the word-internal vowels, whereas in Table 7.2 the word-final and word-internal vowels are shown separately.

Algorithm 7.2 Word-final vowel hypothesisation using changes in F1 position

1. Select two frames of appropriate size in the tail portion of the vowel and obtain their spectra.
2. Compare the F1 position in the two spectra. If F1 does not drop below a threshold, hypothesise the vowel as a word-final vowel.

From the results, it can be seen that the algorithm hypothesises more word-final

Speaker	WF hypotheses(WF:WI) at various thresholds				
	0	1	2	3	4
1	42:14	89:23	109:32	119:43	129:59
2	58:16	89:24	114:37	126:48	128:57
3	46:12	69:23	92:32	113:43	123:57
4	56:19	92:32	106:44	118:55	127:66
5	55:22	87:44	108:63	120:70	129:77
6	34:18	58:34	80:39	99:47	111:54
7	49:24	67:23	92:31	109:40	121:50
8	52:31	75:39	97:53	113:62	125:68
9	52:22	76:36	100:44	110:48	122:58
10	36:21	58:32	79:39	92:49	108:55
11	35:20	55:28	76:36	92:48	108:54

(a)

Speaker	WF hypotheses (WF:WI)	Hit rate	Correctness	Improvement
1	89:23	59%	79%	3.1
2	89:24	59%	79%	3.1
3	69:23	46%	75%	2.9
4	92:32	61%	74%	2.9
5	87:44	58%	66%	2.5
6	58:34	39%	63%	2.4
7	67:23	45%	74%	2.9
8	75:39	50%	66%	2.5
9	76:36	51%	68%	2.6
10	58:32	39%	64%	2.4
11	55:28	37%	66%	2.5

(b)

Table-7.2 Results of word final vowel hypothesisation using changes in F1 position. In (a) results are shown in terms of word final vowels(WF) and word internal vowels(WI) which correspond to the correct and incorrect hypotheses. In (b) results are shown in terms of Hit rate, Correctness and Improvement for a F1 threshold of 1(=20Hz).

vowels than word-internal vowels. Thus the algorithin can also be used to hypothesise word-final vowels: To illustrate this, the results are represented in terms of Hit rate, Correctness and Improvement in the table. It can be seen, for speaker 1, that the algorithm hypothesises (at a threshold of 1) nearly **60%** of the word-final vowels with Correctness around **80%.**

From the word-final vowels hypothesised, one can locate the word boundaries using Algorithm **6.6.** These word boundary hypothesisation results are shown in Table 7.3. **A** comparison with Table 7.1 shows that more word boundaries are detected with greater Correctness.

In the above technique, it was assumed that the vocal tract position would change from a relatively open position to a closed position at a VC juncture. However not all consonants are produced by the complete closure of the vocal tract. Consonants such as semivowels and fricatives are produced by a partial closure of the vocal tract. Thus the change in vocal tract opening at the VC junction will be small, if C is a semivowel or a fricative. Hence the change in F1 position will also be small, and it may not be possible to differentiate between the vowels preceding a word boundary and vowels preceding such sounds, i.e., between V#C and VC cases if C is a semivowel or a fricative. This will result in many incorrect word boundary hypotheses. However, if the consonant is a stop or a nasal, the vocal tract will be completely closed and hence one can differentiate better between the cases of vowels followed by a word boundary and the case of vowels not followed by a word boundary. In Table 7.4, the word boundary hypothesisation results shown in Table 7.1 are presented in terms of the succeeding consonant classes for three speakers (1, 2 and 3). It can be seen immediately that the best performance of the technique is for the class 'pauses' which consist of vowels followed by long silences. This observation validates our proposed clue, because in

Speaker	WB hypotheses(WB:$\underline{WB}$) at various thresholds				
	0	1	2	3	4
1	40:16	87:25	106:35	115:47	125:63
2	56:18	86:27	111:40	123:51	124:61
3	44:14	67:25	90:34	110:46	119:61
4	55:20	90:34	108:46	115:58	124:69
5	53:25	84:47	105:66	116:74	125:81
6	33:19	56:36	77:42	96:50	108:57
7	48:25	65:25	89:34	106:43	117:54
8	50:33	73:41	94:56	109:66	121:72
9	50:24	74:38	97:47	107:51	118:62
10	34:23	56:34	76:42	88:53	104:59
11	34:21	53:30	74:38	89:51	104:58

(a)

Speaker	WB hypotheses (WB:$\underline{WB}$)	Hit rate	Correctness	Improvement
1	87:25	58%	78%	3.7
2	86:27	57%	76%	3.6
3	67:25	45%	73%	3.4
4	90:34	60%	73%	3.4
5	84:47	56%	64%	3.0
6	56:36	37%	61%	2.9
7	65:25	43%	72%	3.4
8	73:41	49%	64%	3.0
9	74:38	49%	66%	3.1
10	56:34	37%	62%	2.9
11	53:30	35%	64%	3.0

(b)

Table 7.3 Results of word boundary hypothesisation using changes in **F1 position.** The results were obtained using **F1** position changes to hypothesise word final vowels and from these the word boundares were located using algorithm 6.7. In (a) results are shown in terms of correct word boundary hypotheses(WB) and incorrect word boundary hypotheses($\underline{WB}$). In (b) the results are shown in terms of Hit rate, Correctness and Improvement for a **F1** threshold of 1(= 20Hz).

Succeeding consonant class	Speaker 1		Speaker 2		Speaker 3	
	WB:WB input	WB:WB hyp.	WB:WB input	WB:WB hyp.	WB:WB input	WB:WB hyp.
vocal stops	39:48	22:7	39:48	22:15	41:50	20:13
nasal stops	7:25	6:7	8:23	5:10	7:27	3:4
semivowels	4:18	1:13	4:17	3:9	5:11	3:6
fricatives	14:13	13:7	14:12	9:5	7:11	2:6
trill	2:14	1:4	1:17	1:5	1:22	0:4
laterals	1:6	0:1	0:6	0:2	5:3	3:1
'h' sound	20:10	16:4	21:11	10:4	10:10	3:4
pauses etc.	17:0	12:0	17:0	13:0	23:0	18:0

Table_7.4 Distribution of the word boundary hypotheses in terms of succeeding consonant classes. The distributions are shown for three speakers(speakers 1, 2 and 3). The F1 threshold used is 1(=20Hz).

vowels followed by pauses, there will be no change in the vocal tract opening, and hence, the word boundaries following the vowels will be hypothesised correctly by our technique. As seen from the results, more than two-thirds of the word boundaries in this class were detected for all the speakers. For the remaining classes, one could immediately see that the discrimination between the vowels followed by a word boundary and the vowels not followed by a word boundary is maximum for the stops and nasals. This is expected since stops (vocal or nasal) are precisely the sounds for which the vocal tract closes completely and hence our algorithm should perform best for them. In fact for the other consonant classes, the results show that the algorithm does not distinguish between vowels preceding a word boundary and vowels which do not precede a word boundary.

From the above discussion one can conclude that the knowledge of the succeeding consonant can greatly aid in producing better word boundary hypotheses. However in the context of speech recognition, such information is not easily obtained though the broadclass to which the consonant belongs (stops, nasals, semivowels etc.) is known. By making use of such information one can improve upon the above technique. For example, one can place more confidence in the word boundary hypotheses if the vowel is succeeded by a stop and a lower confidence if the next sound is not a stop.

One important factor that can affect the performance of this word boundary hypothesisation technique is the speaking rate. This is because the proposed clue was based on the assumption that it is possible to differentiate vowels that are followed by a word boundary from the vowels that are word-internal. Changes in the spectrum (in particular, the F1 position) were used to perform this. However, as the speaking rate increases, the coarticulation effects across word boundaries will also increase, and the differences between vowels followed by a word boundary and vowels which are not, will decrease. Hence one would expect a degradation in the performance of the clue at

high speaking rates. This is also borne out by our results. In our speech data, the speakers were ordered by their speaking rate, with speaker 1 speaking at the slowest rate and speaker 11 speaking the fastest. It can be seen that the performance of the clue is best for speakers 1 and 2, and least for the speakers 10 and 11.

7.3 Word boundary hypothesisation from changes in first formant energy

In the above, a word boundary hypothesisation algorithm that hypothesises word boundaries from changes in the vocal tract configuration was presented. In the following, another word boundary hypothesisation algorithm that is based on the relationship between the changes in the excitation (or the source) and the spectra of word-final vowels is presented.

7.3.1 *Algoritlim for word-final vowel liypotliesisation using changes in F l energy*

The proposed technique for hypothesising word boundaries is based on measuring the spectral changes within a vowel. It is based on the differences in the spectra of a vowel when the effort put in its production is varied. For vowels uttered with more effort, the spectrum contains strong formants whereas for lesser effort the spectrum will show weak formants [Baken and Daniloff 1991]. This is illustrated in Fig.7.1. The idea for word boundary hypothesisation is based on the observation that in some vowels preceding a word boundary, the spectrum of the later half of the vowel is dominated by the peak corresponding to the first formant(F1). This was explained as follows: When a vowel is succeeded by a word boundary, the effort put in producing the vowel decreases and hence the formants weaken due to an increase in the glottal roll off. However the higher formants weaken faster than the first formant and hence the first formant F1 will become more prominent in the spectrum. Hence in a normalised spectrum (total energy set to unity) F1 will have a relatively stronger peak.

It was assumed that the reduction in the effort in producing vowels occurs in all

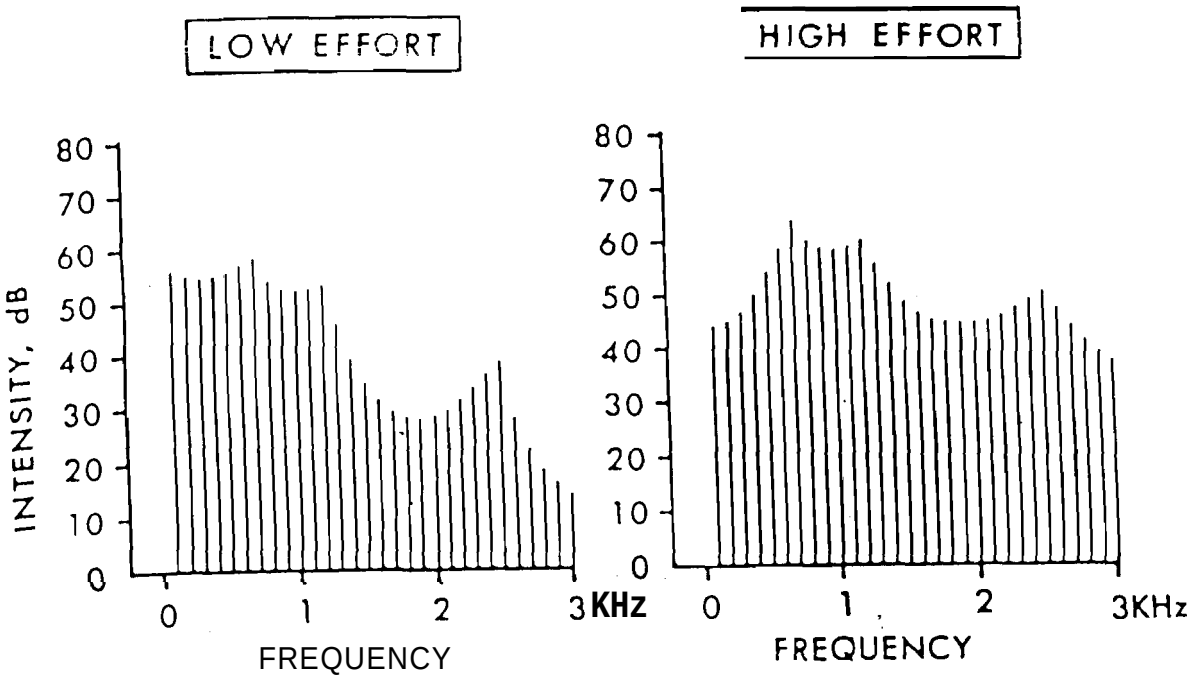


Fig.7.1 The spectra of the vowel 'a:' produced with (i) a low effort and (ii) a high effort. It can be seen that in the spectrum for low effort, the first formant apperas stronger compared to other formants.

vowels preceding word boundaries, i.e., in word-final vowels. If this assumption is valid, then word boundaries can be detected by looking at the changes in the normalised spectrum of a vowel. If the higher formant energies show a decrease, then one can hypothesise a word boundary after that vowel. However, in practice, the higher formants are difficult to detect reliably. Hence the detection procedure is modified to look for steady or increasing F1 energy because a decrease in the energies of the higher formants in a normalised spectrum implies a relative rise in the energy of F1 peak. The algorithm for word-final vowel hypothesisation [Ramana Rao 1992a] is given below.

Algorithm 7.3 Word-final vowel hypothesisation using changes in F1 energy

1. Divide the given vowel into three frames of equal size.
2. Compare the F1 energies in the normalised spectra of the three frames. If the F1 energy does not decrease from first frame to the third, hypothesise the vowel as a word-final vowel.

7.3.2 Results of word boundary hypothesisation using changes in F1 energy

The algorithm was applied on the speech data consisting of 2,600 vowels taken from a total of 110 utterances consisting of 10 Hindi sentences uttered by 11 speakers described earlier. The results of the word-final vowel hypothesisation are shown in Table 7.5.

The results show only a moderate improvement in the ratio of word-final to word-internal vowels. Also the number of word-final vowels detected by this technique is quite small, between 25 to 35% of the total word-final vowels. Obviously, these results show that our assumption that all word-final vowels are weak and hence show a rising F1 energy is not fully correct, even though the hypothesised vowels contain more

Speaker	WF hypotheses (WF:WI)	Hit rate	Correctness	Improvement:
1	50:17	33%	75%	2.9
2	48:15	32%	76%	3.0
3	36:10	24%	78%	3.1
4	47:18	31%	72%	2.8
5	68:22	45%	76%	3.0
6	48:23	32%	68%	2.6
7	51:19	34%	73%	2.8
8	53:23	35%	70%	2.7
9	54:20	36%	73%	2.8
10	38:19	25%	67%	2.6
11	42:19	28%	69%	2.6

Table 7.5 Results of word final vowel detection using F1 energy. The results are shown for 11 speakers.

word-final vowels than word-internal vowels.

7.4 Summary and Conclusions

In this chapter, two word boundary hypothesisation algorithms based on changes in the first formant were described. The first algorithm uses changes in the first formant position to hypothesise word boundaries. Tests with a speech data consisting of 2,600 vowels taken from the utterances of 11 speakers showed that nearly 60% of the word boundaries were detected with Correctness about 70%. When compared with the original distribution of the word boundary and the word-internal vowels an Improvement by a factor of 2 was observed.

However, this technique suffers from several problems, the important one being the effects of speaking rate. It was seen that an increase in the speaking rate decreases the Hit rate and also the Correctness. Thus one needs to find ways of normalising the technique with respect to the speaking rate. In the absence of this, one may not be able to use the technique independently, but may use it along with other clues. Thus one can conclude that while this technique shows promise, it may be advantageous to use it in combination with other clues which are more reliable.

The second word boundary hypothesisation algorithm in this chapter uses changes in the first formant energy to hypothesise word boundaries. This was also tested on the 2,600 vowel speech data. Results showed that the performance of this technique was moderate in that it detected about 30% of the word boundaries with about 75% Correctness.

From the results of the studies, one can clearly see that of the two the technique using changes in F1 position to hypothesise word boundaries is superior. The reasons for the poor performance of the second clue may be the following:

1. The assumption that the effort in the speech production drops on every word-final

vowel may not be fully correct. It is quite likely that this assumption is true only at some word boundaries and hence the word boundaries detected by this technique will be limited.

2. It is also possible for weak vowels to occur in **a** word-internal position and they may contribute to the errors.

From the results of these word boundary hypothesisation studies, we can conclude that techniques based on acoustic-phonetic clues are useful. Since the clues are based on the speech production mechanism, it is likely that these techniques are applicable for other languages as well. However, this is to be explored.

Chapter 8

PERFORMANCE OF WORD BOUNDARY CLUES IN IMPROVING LEXICAL ANALYSIS

8.1 Introduction

In the preceding chapters, several word boundary clues were identified and their performance was presented in terms of the correct and incorrect word boundary hypotheses and also in terms of Hit rate, Correctness and Improvement. Since the purpose of identifying word boundaries is to improve the lexical analysis, we conducted a study to estimate the improvement in the lexical analysis time for a sentence when word boundaries were hypothesised in it using the various word boundary clues. In the study a total of 10 sentences were used. All word boundaries (except sentence boundaries) were removed from these sentences, and then word boundaries were hypothesised using the clues. The resulting sentences with some correct and some incorrect word boundaries were used as input to the lexical analyser described in chapter **3,** and the improvement in the lexical analysis was examined. These studies are presented in the following sections.

8.2 Studies on the reduction in lexical analysis time due to word boundary hypothesisation

This section presents the results of studies made to estimate the reduction in lexical analysis time due to the word boundaries hypothesised using the word boundary clues. The reduction was estimated for each type of clues, namely, language clues, lexical clues, prosodic clues and acoustic-phonetic clues, separately.

8.2.1 Performance of language clues in improving lexical analysis

The language clues are to be applied on a symbol string generated by the speech signal-to-symbol conversion module of the speech recognition system. This symbol string usually contains some errors, and these errors affect the performance of

the language clues in word boundary hypothesisation. In chapter 4 , it was shown that the number of correct word boundaries detected and the number of incorrect word boundary hypotheses produced by the language clues vary with the percentage errors in the input sentences. Hence, to estimate the lexical analysis time for a sentence, which depends on the number of correct and incorrect word boundaries in the sentence, one needs to assume some percentage errors in the sentence. In our study, a maximum error percentage of 15% was assumed in the input sentence. Though such an error percentage may be smaller than that of many current signal-to-symbol conversion systems, its choice was justified by the following reasons:(1) Firstly, even if current signal-to-symbol conversion systems may produce more errors, it is possible to achieve less than 10% errors in signal-to-symbol conversion, as shown by the spectrogram reading experiments[Cole, Rudnicky, Zue and Reddy 1980], and, (2) more importantly, beyond this error percentage, the computation time for lexical analysis is going to be large and beyond our measurement limit of 1 day.

Word boundaries were hypothesised in each of the input sentences in which some of the phonemes were replaced. The resulting sentences were used as the input to the lexical analyser and the time for lexical analysis was measured for varying mismatch costs. Note that a mismatch cost of 1 corresponds to an input error percentage of 3%(the mismatch cost of 5 corresponds to input error of 15%). The results of the lexical analysis are shown in Table 8.1. The lexical analysis times of these sentences without word boundaries (given in Table 3.4) are also shown in parentheses.

From the results it can be seen that there is a significant reduction in the lexical analysis time of a sentence when word boundaries were hypothesised in it using the language clues. However, it can also be seen that the reduction is not uniform. For some sentences, the reduction is as large as 30 whereas for a few other sentences the

Match cost	Sentence number 1	2	3	4	5	6	7	8	9	10
0	0 (1)	0 (0)	0 (1)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (1)	0 (2)
1	6 (18)	1 (3)	3 (7)	2 (4)	6 (3)	21 (5)	5 (3)	1 (11)	0 (13)	13 (47)
2	35 (238)	4 (20)	15 (62)	5 (36)	25 (27)	124 (64)	26 (16)	2 (149)	1 (108)	83 (674)
3	153 (2199)	15 (97)	62 (408)	13 (245)	86 (172)	572 (602)	113 (85)	4 (1503)	3 (712)	426 (6988)
4	534 (15936)	50 (397)	246 (2357)	39 (1532)	265 (996)	2223 (4757)	457 (420)	8 (12261)	6 (3980)	1849 (56712)
5	1544 _	149 (1401)	907 (11918)	111 (7444)	723 (4902)	7553 (31851)	1720 (1868)	14 _	10 (19826)	6953 _

Table 8.1 The lexical analysis times (in seconds) for 10 sentences containing word boundaries hypothesised using language clues.

reduction is as low as 2. This is because in some sentences the language clues hypothesised all the word boundaries correctly without any errors, whereas in a few others only a few word boundaries were hypothesised correctly with many incorrect hypotheses. Tiius there is a significant variation in the percentages of correct and incorrect word boundary hypotheses across the sentences and this resulted in the variation in the reduction of lexical analysis times.

The reduction in the lexical analysis time due to the word boundaries hypothesised by the lexical clues for one sentence (same as the sentence used in Section 8.2.1) is plotted in Fig. 8.1. It can be seen that there is a significant reduction in the lexical analysis time due to the word boundaries hypothesised by the language clues. Moreover, the reduction increases exponentially with increasing mismatch cost between the sentence and an alternate word string.

8.2.2 Performance of lexical clues in improving lexical analysis

A study similar to the one above for language clues was carried out using lexical clues. In this study also, a maximum error percentage of 15% was assumed in the input sentences and the lexical clues were used to hypothesise word boundaries in the erroneous sentences. The resulting sentences were used as input to the lexical analyser and the results of the lexical analysis are shown in Table 8.2.

From the Table one can observe that there is a reduction in the time for lexical analysis due to the word boundaries hypothesised by the lexical clues. However, it can also be seen that the reduction in the lexical analysis time is much less when compared to that of the language clues. This is because the lexical clues were able to hypothesise lesser number of word boundaries when compared to the language clues. Also, as in the case of language clues, the percentages of correct and incorrect word boundary hypotheses varied across the sentences and hence there is a corresponding variation in the reduction in the lexical analysis times for the sentences.

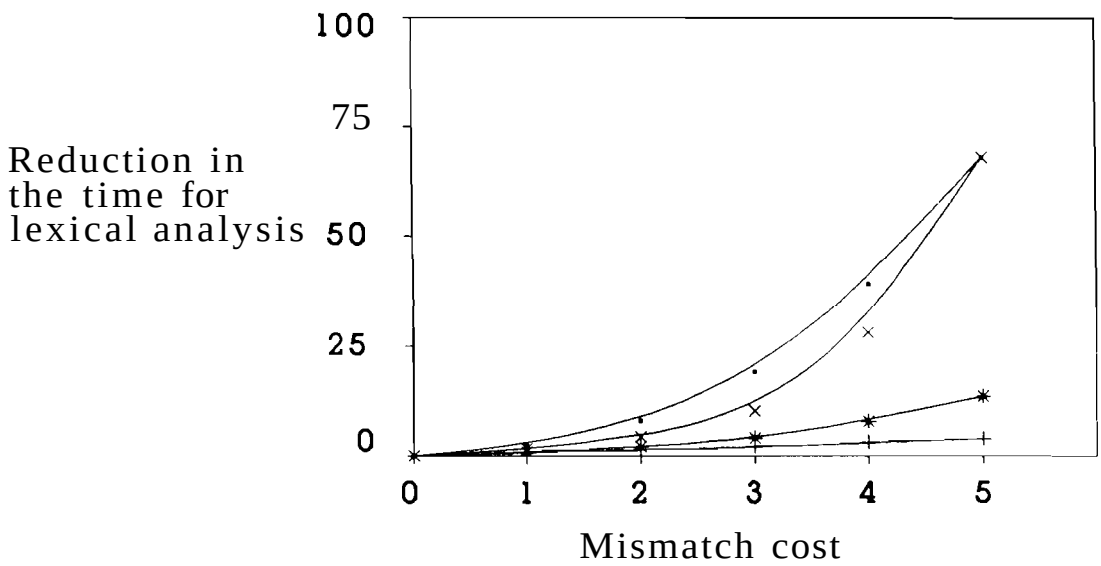


Fig.8.1 An illustration of the reduction in the lexical analysis complexity (time) due to the word boundaries hypothesised by word boundary clues. In the figure, the ratio of the lexical analysis time of a sentence without any word boundaries to the lexical analysis time of the sentence with word boundaries hypothesised by the word boundary clues is shown. The ratios are plotted for the four types of clues, namely, language clues (marked by .), lexical clues (marked by +), prosoidc clues (marked by X) and acoustic-phonetic clues (marked by *). It can be seen that the reduction in the lexical analysis time is maximum for prosoidc and language clues, whereas for lexical and acoustic-phonetic clues the reduction is much less.

Match cost	Sentence number 1	2	3	4	5	6	7	8	9	10
0	0 (1)	0 (0)	0 (1)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (1)	0 (2)
1	7 (18)	2 (3)	3 (7)	5 (4)	2 (3)	2 (5)	3 (3)	12 (11)	5 (13)	14 (47)
2	69 (238)	6 (20)	23 (62)	26 (36)	6 (27)	13 (64)	16 (16)	117 (149)	35 (108)	169 (674)
3	506 (2199)	23 (97)	117 (408)	120 (245)	20 (172)	69 (602)	86 (85)	893 (1503)	204 (712)	1501 (6988)
4	2840 (15936)	79 (397)	495 (2357)	503 (1532)	62 (996)	303 (4757)	420 (420)	5634 (12261)	1009 (3980)	10631 (56712)
5	13674 _	241 (1401)	1795 (11918)	1939 (7444)	174 (4902)	1166 (31851)	1901 (1868)	26022 _	4380 (19826)	_ _

Table-8.2 The lexical analysis times (in seconds) for 10 sentences containing word boundaries hypothesised using lexical clues.

The reduction in the lexical analysis time due to'the word boundaries hypothesised by the lexical clues for one sentence (same as the sentence used in Section 8.2.1) is also plotted in Fig. 8.1.

8.2.3 Performance of prosodic clues in improving lexical analysis

Unlike the language and lexical clues, the prosodic and the acoustic-phonetic clues can be applied directly on the speech signal(though some prior segmentation of sounds is needed). The prosodic clues of pause, duration and pitch were used together (Algorithm 6.6&6.7 together) to hypothesise word boundaries in the input utterances. The utterances were obtained by one native Hindi speaker reading the 10 sentence text used. Using the word boundary hypotheses the lexical analysis times were estimated for the input sentences at various input error rates. The results are shown in Table_8.3. It can be seen that there is a significant reduction in the lexical analysis times for all the sentences.

The reduction in the lexical analysis time due to the word boundaries hypothesised by the prosodic clues for one sentence (same as the sentence used in Section 8.2.1) is also plotted in Fig.8.1.

8.2.4 Performance of acoustic-phonetic clues in improving lexical analysis

Word boundaries were hypothesised in the input utterances using the acoustic-phonetic clues (both clues applied together). Using these word boundary hypotheses in the sentences the lexical analysis times were estimated. The results are shown in Table_8.4. From the table, it can be seen that there is a reduction in the lexical analysis times for the sentences due to the word boundaries hypothesised by the acoustic-phonetic clues.

The reduction in the lexical analysis time due to the word boundaries hypothesised by the acoustic-phonetic clues for one sentence (same as the sentence

Match cost	Sentence number 1	2	3	4	5	6	7	8	9	10
0	0 (1)	0 (0)	0 (1)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (1)	0 (2)
1	4 (18)	0 (3)	5 (7)	3 (4)	1 (3)	5 (5)	0 (3)	3 (11)	5 (13)	9 (47)
2	25 (238)	1 (20)	24 (62)	9 (36)	4 (27)	27 (64)	1 (16)	13 (149)	21 (108)	43 (674)
3	118 (2199)	2 (97)	92 (408)	24 (245)	12 (172)	118 (602)	3 (85)	49 (1503)	82 (712)	178 (6988)
4	438 (15936)	3 (397)	302 (2357)	55 (1532)	31 (996)	435 (4757)	8 (420)	163 (12261)	290 (3980)	615 (56712)
5	1329 _	5 (1401)	865 (11918)	109 (7444)	72 (4902)	1393 (31851)	17 (1868)	495 _	909 (19826)	1848 _

Table 8.3 The lexical analysis times (in seconds) for 10 sentences containing word boundaries hypothesised using prosodic clues.

Hatch cost	Sentence number 1	2	3	4	5	6	7	8	9	50
0	0 (1)	0 (0)	0 (1)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (1)	0 (2)
1	19 (18)	6 (3)	6 (7)	5 (4)	7 (3)	6 (5)	7 (3)	22 (11)	5 (13)	12 (47)
2	140 (238)	20 (20)	29 (62)	18 (36)	38 (27)	32 (64)	27 (16)	163 (149)	21 (108)	68 (674)
3	777 (2199)	61 (97)	115 (408)	63 (245)	167 (172)	134 (602)	96 (85)	898 (1503)	83 (712)	307 (6988)
4	3407 (15936)	161 (397)	372 (2357)	196 (1532)	664 (996)	496 (4757)	316 (420)	4001 (12261)	298 (3980)	1143 (56712)
5	12694 _	387 (1401)	1017 (11918)	550 (7444)	2366 (4902)	1676 (31851)	990 (1868)	15023 _	942 (19826)	3578 _

Table 8.4 The lexical analysis times (in seconds) for 10 sentences containing word boundaries hypothesised using acoustic-phonetic clues.

used in Section 8.2.1) is also plotted in Fig.8.1.

8.3 Summary and Conclusions

In this chapter, a study was reported in which the performance of the various word boundary clues, namely, the language, lexical , prosodic and acoustic-phonetic clues, in reducing the lexical analysis time was estimated. The clues were used to hypothesise word boundaries in a Hindi sentence and the resulting sentence with some word boundaries was used as input to the lexical analyser. It was shown that all the clues resulted in word boundary hypotheses which significantly reduced the lexical analysis time for the sentence. This is illustrated in Fig.8.1, where the reduction in lexical analysis time was plotted against the mismatch cost. Among the clues, prosodic and language clues performed best reducing the lexical analysis time by a factor of 68 for 15% input error. On the other hand, the performance of the lexical and acoustic-phonetic clues was less with reductions of 4 and 14 respectively.

The results of the study demonstrated the utility of the word boundary clues proposed in this thesis in improving the performance of a lexical analyser. It was also shown that among the word boundary clues, the language and the prosodic clues perform significantly better than the other clues, namely, lexical and acoustic-phonetic clues.

Chapter 9

SUMMARY AND CONCLUSIONS

In this thesis, several clues were identified for word boundary hypothesisation in continuous Hindi speech. The clues were based on various knowledge sources such as syntax and semantics (referred as language knowledge), lexicon, prosody and acoustic-phonetics. Using each of these knowledge sources, some clues were proposed for word boundary hypothesisation and their utility was verified in the context of speech recognition. Brief summaries of these studies are given below.

In the first study (described in chapter **3),** the significance of word boundary hypothesisation in the context of continuous speech recognition, and, in particular, in the lexical analysis stage, was established. In this, experiments were conducted to estimate the reduction in the number of alternate word sequences produced by a lexical analyser and also the reduction in the time taken for lexical analysis, due to the presence of word boundaries in the input. Results showed that both the number of alternate word sequences and the time for lexical analysis were reduced, but the reduction was larger in the time taken for lexical analysis. This implies that the main effect of word boundary hypothesisation is in reducing the time spent on lexical analysis. Since lexical analysis is expected to be the most time consuming stage in speech iecognition, the presence of word boundaries will significantly reduce the overall speech recognition time.

Studies were also conducted on the effect of word boundary errors on the time spent on lexical analysis. It was found that even when an input sentence contained 50% word boundary errors, the time for lexical analysis was significantly less than the time for lexical analysis of the sentence without word boundaries.

The above study conclusively established the importance of word boundary hypothesisation in improving the performance of a lexical analyser. The next four

studies (reported in chapters 4, 5, 6 and 7) are on the evaluation of some word boundary clues proposed by us. In each of the studies a particular feature of Hindi speech is examined and clues were proposed for hypothesising word boundaries based on these features. The performance of each of these clues in improving lexical analysis is discussed in chapter 8.

The first study on identifying word boundary clues examined the use of word frequency information. The clues proposed are the phoneme sequences corresponding to the frequently occurring function words. The idea was to spot these clues in a Hindi text and hypothesise word boundaries around them. It was found that the clues detected nearly 70% of the word boundaries with incorrect word boundary hypotheses less than 20% for correct texts. Even when the input text contained errors in 50% of the phonemes, the clues were able to detect about 35% of the word boundaries.

In the second study (described in chapter 5), the constraints on Hindi phoneme sequences were proposed as word boundary clues. The clues correspond to the phoneme sequences which do not occur in any Hindi word. By spotting such sequences in a correct text, nearly 50% of the word boundaries were detected with incorrect hypotheses less than 15%. When the input text contained errors in 50% of the phonemes, the clues detected 50% of the word boundaries correctly with incorrect hypotheses around 40%.

In the third study (described in Chapter 6), clues based on the prosodic features of Hindi, namely, pause, duration, and pitch were used as clues to hypothesise word boundaries. The performance of each of these clues was estimated using a speech data of 110 sentences produced by 11 speakers. A technique which combines these clues together was proposed and it was found that nearly three-fourths of the word boundaries were detected with incorrect hypotheses less than 20%.

In the last study (described in chapter 7), two word boundary clues were proposed which are based on the changes in the vocal tract configuration (as reflected in F1 position) and the excitation (as reflected in F1 energy). These clues were applied on a 110 sentence speech data uttered by 11 speakers. Results showed that the change in F1 position is useful in hypothesising word boundaries while the change in F1 energy seems to be relatively less useful.

The performance of the word boundary clues identified in the above four studies in reducing lexical analysis time is estimated in chapter 8. In this each type of word boundary clues is applied on a text of 10 sentences and the resulting sentences were input to the lexical analyser. The lexical analysis times for these sentences were compared to the lexical analysis times for the same sentences without word boundaries, and the reduction in the time due to the hypothesised word boundaries is estimated. It was found that the reduction is higher for prosodic and language clues whereas for lexical and acoustic-phonetic clues it is less.

From the studies reported in this thesis, the following conclusions are drawn:

1. It is possible to detect many word boundaries in Hindi speech. This is an important contribution of this thesis since in many of the earlier studies, only a few word boundaries were detected reliably. On the other hand, in our studies, the prosodic clues of pitch, duration and pause, together detected more than 70% of the word boundaries in the utterances with incorrect hypotheses less than 20%. Similarly, clues based on language knowledge and lexical knowledge also detected a large number of word boundaries.

2. The language knowledge and prosodic knowledge seem to be the most promising sources for clues to word boundaries. This is evident from our studies in which techniques based on these knowledge sources provided the best results. Moreover in most of the other techniques also the language features are exploited to detect word

boundaries. For example, the technique of detecting word boundaries using changes in F1 frequency, utilises the fact that in any Hindi text nearly 70% of the words end in vowels which is a language feature. Similarly in the technique based on duration also, it is the language features that are exploited. However this also means that the techniques developed may not work for other languages as different languages have different features and their prosody also differ. Hence one needs to develop different techniques for different languages keeping in mind the peculiarities of those languages. Another disadvantage, especially for clues based on prosodic knowledge, is that it depends on the speaker also, whether the speaker is a native or a nonnative speaker.

3. Acoustic-phonetic knowledge based clues can also be used to hypothesise word boundaries (for example, the clue of changes in F1 reported in this thesis). However, it appears that the applicability of these clues is limited to slow speech because in rapid speech the distinction between word-internal and word-final sounds decreases. But such clues may still be useful in conjunction with other clues possibly based on the other knowledge sources. One important advantage with these clues is that they are language independent and hence are applicable for a number of languages which have similar sounds whereas all the other clues are language and speaker dependent.

Suggestions for further work

The work reported in this thesis established the significance of word boundary hypothesisation in speech recognition and also identified several clues to hypothesise word boundaries. However, a number of issues are still to be resolved. These are discussed below.

One important issue which was not addressed in this thesis is the integration of the various word boundary clues to develop a word boundary hypothesiser. This is because, the clues identified are applicable on different types of inputs. For example,

the language and lexical clues are to be applied on a symbol sequence produced by a speech signal-to-symbol converter, whereas the prosodic and the acoustic-phonetic clues are to be applied on the speech signal or some representation of it. Hence, it may not be possible to develop a single word boundary hypothesiser module. Moreover, to apply all the clues together, one needs to assign relative confidences to each of the clues and then build a mechanism to combine these confidences. However, assigning the confidences is not an easy task. For example, from the results of the above studies, a simple suggestion may be to place more faith in the prosodic and higher level linguistic clues and give less weightage to the rest. However, such choices are very much affected by the task context. For example, if the task is small vocabulary connected word recognition, such as a telephone help facility, one may find more use to the acoustic-phonetic and lexical clues rather than the prosodic and the higher level linguistic clues since the sentences (or word sequences) do not have much structure. On the other hand, if the task is large vocabulary speech recognition, the reverse may be true. For conversational speech, it may be something in between these. Another problem with some of the clues, viz. prosodic clues, is that they are also speaker dependent to some extent. Hence a better view is to treat each of these clues as part of the respective knowledge sources and apply them in the overall speech recognition context rather than try to build a word boundary hypothesiser module separately.

One other issue not addressed in the thesis is the verification of the proposed clues by performing perception studies in which these clues can be verified using human subjects. The approach taken by us in this thesis is more of an engineering one in that we posed the problem of word boundary hypothesisation and tried to find clues which can aid in solving it. It is quite possible that the clues identified by us may or may not be used by humans when they recognise speech. Obviously studies on human perception need to be performed to establish this. Such perceptual studies will not only

establish the reliability of the clues but may also provide ideas on how the clues are to be applied.

Based on the above, we suggest the following studies on word boundary hypothesisation for further work.

1. Studies to find more clues, especially clues based on the acoustic-phonetic knowledge.

2. Studies on human perception of word boundaries, in line with studies done for English [Butterfield and Cutler 1990; Cutler and Butterfield 1990a, 1990b, 1991b], to establish whether the proposed clues are used by humans in their recognition of speech.

3. Once a sizable number of clues are identified, one can also investigate the integration of the clues to develop a word boundary hypothesiser for Hindi speech.

REFERENCES

Baken R.J. and Daniloff R.G. (1991), Readings in clinical spectrography of speech, Singular publishing group and KAY elemetrics, USA.

Beckman M. and Pierrehumbert J. (1986), Intonational structure in Japanese and English, Phonology yearbook 3, pp.255-309.

Bhatia Kailash Chandra (1970), *Hindi: b^ha-s a: me:n aks ar tat^ha: ś abd ki: si:ma:* (syllable and word boundaries in Hindi), Nagari Pracarini Sabha, Varanasi.

Briscoe E.J.(1989), Lexical access in connected speech recognition, Proceedings of the Twenty-seventh Congress of the Association for Computational Linguistics, Vancouver.

Butterfield S. and Cutler A. (1990), Intonational cues to word segmentation in clear speech?, Proc. Institute of Acoustics 12, part 10, pp.87-94.

Chandra Sekhar C., Ramana Rao G.V., Eswar P., Saikumar J., Ramasubramanian N., Sundar R. and Yegnanarayana B. (1990), Development of a speech-to-text system for Indian languages, Proc. Frontiers in Knowledge based Computing(KBCS 90), pp.457-466.

Cole R.A. and Jakimik J. (1980), A model of speech production, In Perception and Production of Fluent Speech (R.A. Cole ed.), Hillsdale, New Jersey.

Cole R.A., Rudnicky A.I., Zue V.W., and Reddy D.R (1980), Speech as patterns on

paper, In Perception and Production of Fluent Speech (R.A. Cole ed.), Hillsdale, New Jersey, pp.3-50.

Crystal T.H. and House A.S. (1988), Segmental durations in connected speech signals: Current results, JASA 83, pp.1553-1573.

Cutler A. and Carter D.M. (1987), The predominance of strong initial syllables in the English vocabulary, Computer Speech and Language 2, pp.133-142.

Cutler A. and Norris D. (1988), The role of strong syllables in segmentation for lexical access, Journal of Experimental Psychology: Human Perception and Performance 25, pp.385-400.

Cutler A. (1990), Exploiting Prosodic Probabilities in Speech Segmentation, In Cognitive Models of Speech Processing (T.M.ALtmann ed.), pp.105-121, MIT press.

Cutler A. and Butterfield S. (1990a), Durational cues to word boundaries in clear speech, Speech Communication 9, pp.485-495.

Cutler A. and Butterfield S. (1990b), Syllabic lengthening as a word boundary cue, Proc. Third Australian international conference on Speech Science and Technology, pp.324-328.

Cutler A. and Butterfield S. (1991a), Rhythmic cues to speech segmentation: Evidence from juncture misperception, Journal of Memory and Language 2, pp.133-142.

Cutler A. and Butterfield S. (1991b), ,Word boundary cues in clear speech: A supplementary report, Speech Communication, 10, pp.335-353.

Dalby J., Laver J. and Hiller S.M. (1986), Midclass phonetic analysis for a continuous speech recognition system, Proc. of The Institute of Acoustics, 8.7, pp.347-354.

Eswar P. (1990), A rule based approach for character spotting from continuous speech in Indian languages, Ph.D thesis, IIT Madras, India.

Grosjean F. (1980), Linguistic structures and performance structures: Studies in pause distribution, In Temporal Variables in Sueech (Dechert H.W. and Raupach M. ed.), Mouton, The Hague, pp.91-106.

Hall P.A.V. and Dowling G.R. (1980), Approximate string matching, ACM Computing Surveys 12, pp.381-402.

Harrington J. and 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., Johnson I. and Cooper M. (1987), The application of phoneme sequence constraints to word boundary identification in automatic, continuous speech recognition, Proceedings of European Conference on Speech Technology 1, pp.163-166.

Harrington J., Watson G. and Cooper M. (1989), Word boundary detection in broadclass and phoneme strings, Computer Speech and Language 3, 367-382.

Hatazaki K. and Watanabe T. (1987), Large vocabulary word detection by searching in a Tree-structured word dictionary, Proc. ICASSP, pp.848-851.

Lamel L. and Zue V. (1984), Properties of consonant sequences within words and across word boundaries, Proc. ICASSP, 42.3.1-42.3.4.

Lea W.A. (1980), Prosodic aids to speech recognition, In Trends in speech recognition (W.A.Lea ed.), 166-205, Prentice Hall, New Jersey.

Madhukumar A.S. (1993), Intonation knowledge for speech systems for Indian language, Ph.D thesis, IIT, Madras.

Mohan B. and Kapoor B.N. (1989), Meenakshi Hindi-English dictionary, Meenakshi Prakashan, Meerut, India.

Nakagawa S. and Sakai T. (1979), A recognition system of connected spoken words based on word boundary detection, Studia Phonologica, 13.

Nakatani L.H. and Schaffer J.A. (1978), Hearing 'words' without words: Prosodic cues for speech perception, JASA 63(1), pp.234-245.

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

Prakash M., Ramana Rao G.V., Chandra Sekhar C. and Yegnanarayana B. (1989), Parsing spoken utterances in an inflectional language, Proc. EUROSPEECH'89, pp.546-549.

Rajendran S. and Yegnanarayana B. (1994), Word boundary hypothesisation for continuous speech in Hindi based on FO patterns, Submitted to Speech communication.

Rajesh Kumar S.R. (1990), Significance of durational knowledge for a Text-to-Speech System in an Indian Language, M.S. thesis, IIT, Madras.

Ramana Rao G.V., Prakash M. and Yegnanarayana B. (1989), Word boundary hypothesisation in Hindi speech, Proc. EUROSPEECH'89, pp.546-549.

Ramana Rao G.V. and Yegnanarayana B. (1991), Word boundary hypothesisation in Hindi speech, Computer Speech and Language 5, pp.379-392.

Ramana Rao G.V. (1992a), Detection of word-final vowels in speech using first formant energy, Proc. Regional workshop on Computer Processing of Asian Languages(CPAL-2), Kanpur, pp.243-247.

Ramana Rao G.V. (1992b) , Detection of word boundaries in continuous speech using pitch and duration, Fourth Australian international conference on Speech Science and Technology(SST-92), Brisbane, Australia.

Simodaira H. and Kimura M. (1992), Accent phrase segmentation using pitch pattern clustering, Proc, ICASSP 1, pp.217-220.

Shipman D.W. and Zue V.W. (1982), Properties of large lexicons: Implications for advanced Isolated Word Recognition systems, Proc. ICASSP.

Ukita T., Nitta T. and Watanabe S. (1986), A speaker independent recognition algorithm for connected word using word boundary hypothesiser, Proc, ICASSP, pp.1077-1080.

Wightman C.W. and Ostendorf M. (1991), Automatic recognition of prosodic phrases, Proc. ICASSP 1, pp.321-324.

Wolf J J. and Woods W.A. (1980), The HWIM speech understanding system, in Trends in speech recognition (Lea W.A. ed.), Prentice Hall, New Jersey, pp.316-339.

Yegnanarayana B., Chandra Sekhar C., Ramana Rao G.V., Eswar P. and Prakash M. (1989), A continuous speech recognition system for Indian languages, Proc. Regional Workshop on Computer Processing of Asian Languages(CPAL-1), pp.347-356.

Yegnanarayana B., Rajendran S., Rajesh Kumar S.R., Ramachandran V. and Madhukumar A.S. (1992), Knowledge sources for a Text-to-Speech system in Hindi, Proceedings gf the Second Regional Workshop on Comvuter Processing of Asian Languages(CPAL-2), Kanpur, India, pp.233-242.

Zelenski R. and Class F. (1983), A segmentation algorithm for connected word recognition based on estimation principles, IEEE ASSP 31, pp.818-827.

LIST OF PUBLICATIONS

1. G.V.Ramana Rao, Detection of word boundaries in continuous speech using pitch and duration, Fourth Australian international conference on Speech Science and Technology(SST-92), Brisbane, Australia, Oct 1992.

2. G.V.Ramana Rao, Detection of word final vowels in speech using first formant energy, Proc. Regional workshop on Computer Processing of Asian Languages(CPAL-2), Kanpur, pp243-247, March 1992.

3. G.V.Ramana Rao and B.Yegnanarayana, Word boundary hypothesisation in Hindi speech, Computer Speech and Language, vol.5, no.4, pp379-392, Dec 1991.

4. G.V.Ramana Rao, M.Prakash and B.Yegnanarayana, Word boundary hypothesisation in Hindi speech, Proc. EUROSPEECH'89, Paris, vol.1, pp546-549, Sep 1989.

5. M.Prakash, G.V.Ramana Rao, C.Chandra Sekhar and B.Yegnanarayana, Parsing spoken utterances in an inflectional language, Proc. EUROSPEECH'89, Paris, vol.1, pp546-549, Sep 1989.

6, C.Chandra Sekhar, G.V.Ramana Rao, P.Eswar etal., Development of a speech-to-text system for Indian languages, Proc. Frontiers in Knowledge based Computing(KBCS 90), Pune, pp.467-476, Dec 1990.

7. B.Yegnanarayana, C.Chandra Sekhar, G.V.Ramana Rao, P.Eswar and M.Prakash, A continuous speech recognition system for Indian languages, Proc. Regional workshop on Computer of Asian Languages(CPAL-1), Bangkok, pp347-356, Sep 1989.