

MACHINE RECOGNITION OF TEXT USING
CONTEXTUAL POSTPROCESSING

Khalid J. Siddiqui

A Thesis
in
The Department
of
Computer Science

Presented in Partial Fulfillment of the Requirements
for the degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

April 1981

© Khalid Javed Siddiqui, 1981

ABSTRACT

MACHINE RECOGNITION OF TEXT USING CONTEXTUAL POSTPROCESSING

Khalid J. Siddiqui

A three stage text-recognition system, which uses contextual information is proposed in this thesis.

Three different passages were selected from English texts. A textual data set consisting of hand-printed characters was taken from Munson's multiauthor data. Machine-printed characters were chosen from OCR 'A' and OCR 'B' characters sets. The passages for text recognition experiments were set-up using the characters included in these data sets, and thus called machine-printed and hand-printed / texts. Some statistical tests were conducted for comparing three passages.

Preprocessing techniques which perform smoothing and size-normalization were included. A feature-extraction and training system based on statistical notions was adopted. Preprocessing and feature extraction constituted the first stage of the system. Second stage was a maximum selector and known as SCRS. Third stage was a contextual postprocessor. Recognition algorithms based on Markov assumptions were studied and used as contextual processor for evaluating their

performance on text. Modified Viterbi algorithm (MVA) based on this assumption was used extensively. Imbricate Block Decoding algorithm (IBDA) was also proposed. A dictionary method in which the search is a function of a constant parameter was implemented. A hybrid approach which includes dictionary method and MVA is proposed as well. These methods were extensively compared with each other by examining their performance on three passages.

Perfect character recognition rates were observed on machine-printed passages, while over 99% recognition was observed from some of the algorithms on hand-printed text. Among context aided algorithms studied, hybrid approach gave the best results and MVA was a reasonably efficient algorithm. As compared to previous block decoding algorithms, much better results and efficiency was obtained from IBDA.

Computational complexity of the algorithms presented, was also analysed. Subject to the performance objectives and the quality of data, this analysis will help in deciding the algorithm to be used.

To my fiance Ghazala,
mom and dad and little sisters.

ACKNOWLEDGEMENTS

The writer gratefully acknowledges the guidance and assistance of Prof. R. Shinghal and Prof. C. Y. Suen, Department of Computer Science, Concordia University, which enabled the successful completion of this study. Their support has seen through many difficult moments during his stay at Concordia.

The writer also wishes to thank his colleagues, Bilal in particular, in the Department of Computer Science at Concordia University for their many useful suggestions. Appreciate also the services and cooperation provided by staff of the department and Computer Center for this research. Special thanks to dear parents, for providing peace of mind throughout.

This work was supported by funds from the Department of Education of Quebec. The text of this thesis was formatted by the TYPESET word-processing package at Concordia University.

Last, but not least, thank you Ghazala, for your patience and understanding.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Thesis Organization1
1.2 Contextual Text Recognition System4
1.3 Mathematical Model for Recognition7
1.4 The Proposed Text Recognition System10

CHAPTER 2

HISTORICAL REVIEW

2.1 Preview of the Chapter14
2.2 Character Recognition14
2.3 The Use of Context in Text Recognition20
2.4 Contextual Algorithms: Past Approaches23
2.4.1 Dictionary Look-up Methods23
2.4.2 Probability Distribution methods using N-grams24
2.4.3 Hybrid Methods26

CHAPTER 3

PREPROCESSING, FEATURE EXTRACTION AND TRAINING

3.1 Preview of the Chapter27
3.2 Munson's data set of character patterns28
3.2.1 Noise Elimination28
3.2.2 Size Normalization38
3.2.3 Feature Extraction45
3.2.4 Training of the classifier51
3.3 OCR 'A' Patterns52
3.4 OCR 'B' Patterns57

CHAPTER 4

COMPILATION OF PASSAGES FOR TEXT RECOGNITION EXPERIMENTS

4.1	Preview of the Chapter59
4.2	Passages59
4.3	Comparison of Passages63
4.3.1	Test Hypothesis H_0165
4.3.2	Test Hypothesis H_0270
4.4	Setting-up Passages for Text Recognition Experiments73

CHAPTER 5

SINGLE CHARACTER RECOGNITION SUBSYSTEM

5.1	Preview of the Chapter76
5.2	Single Character Recognition Subsystem (SCRS)76
5.3	Confusion Matrices88

CHAPTER 6

CONTEXTUAL POSTPROCESSOR MODIFIED VITERBI ALGORITHM

6.1	Preview of the Chapter92
6.2	Contextual Postprocessor (CPPS)92
6.2.1	Estimation of Transition Probabilities95
6.3	Modified Viterbi Algorithm (MVA)96
6.3.1	Description of the Algorithm99
6.3.2	Implementation of the Algorithm	...104
6.4	Experiments with MVA/VA	...108

CHAPTER 7

CONTEXTUAL POSTPROCESSOR BLOCK DECODING AND HYBRID METHOD

7.1	Preview of the Chapter	...115
7.2	Block Decoding Algorithms	...115
7.2.1	Block Decoding Algorithm (BDA)	...116
7.2.2	Imbricate Block Decoding Algorithm (IBDA)	...129
7.3	Dictionary Method	...139
7.4	Hybrid Method	...148
7.5	Final Remarks	...151

CHAPTER 8

CONCLUSION AND SOME SUGGESTIONS FOR FURTHER RESEARCH

8.1	Preview of the Chapter	...153
8.2	Summary Review	...153
8.3	Suggested Improvements in the Present System	...161
8.4	Suggestions for Further Research in this area	...162
	Appendix	...165
	Bibliography	...177

LIST OF TABLES

3.1	Frequency distribution of characters in training and testing set of OCR 'A' data set.	...56
4.1	Unigram Bayesian Probabilities.	...60
4.2	Bigram Bayesian Probabilities.	...61
4.3	Frequency distribution of characters in 3 passages.	...64
4.4	Ho1: Passages represent commonly used English?	...68
4.5	Ho2: Passages are Mutually Similar?	...72
5.1-A	Recognition rates on 3 data sets obtained by using SCRS only.	...83
5.1-B	Recognition rates on Munson's data obtained by using SCRS only.	...84
5.2	Confusion Matrix for training set of Munson's data.	...89
5.3	Confusion Matrix for testing set of OCR 'B'.	...90
6.1	Transition Bayesian Probabilities.	...97
6.2-A	Recognition rates obtained by using Viterbi Algorithm.	...109
6.2-B	Percentage gain in recognition rates by using VA over not using context.	...111
6.3	Recognition rates obtained by using Modified Viterbi algorithm.	...112
7.1	Recognition rates obtained by using Block Decoding Algorithm.	...124
7.2	Recognition rates obtained by using Imbricate Block Decoding Algorithm.	...134
7.3	Dictionary Constituents.	...142
7.4	Recognition rates obtained by using Dictionary Look-up algorithm.	...144
7.5	Words in Pas-1 recognized at preliminary classification step of DLA.	...146
7.6	Recognition rates obtained by using Hybrid Method.	...149

8.1 Summary of results and comparison of recognition rates obtained by using different algorithms. ...157

8.2 Word Recognition rates obtained by using Some Selected Algorithms. ...159

LIST OF FIGURES

1.1	Basic Components of a Recognition System5
1.2	Outline of the Proposed Text Recognition System.12
2.1	Eleven Fonts used by Glucksman.	...18
3.1	Some typical Patterns from Munson's data base.29
3.2	S-window- points involved in smoothing operation.	...32
3.3	L-window-- points involved in Hole detection.	...32
3.4	Points of Interest in Smoothing operation.	...34
3.5	Noise-eliminated Pattern of the one given in FIGURE 3.4.	...39
3.6	Height-normalized Pattern of the one given in FIGURE 3.5.	...44
3.7	Size-normalized Pattern of the one given in FIGURE 3.5.	...46
3.8	Ordering of the 36 regions each region being 4 X 4 of Munson's patterns.	...48
3.9	A typical region: Count of black points in 16 squares make one feature element in the vector space.	...49
3.10-A	Some typical Patterns from OCR 'A' data set.	...53
3.10-B	Size-normalized Pattern of Character 'D' of FIGURE 3.10-A.	...54
3.11-A	A typical Pattern of character 'A' from OCR 'B' data set.	...58
3.11-B	Size-normalized pattern of character of FIGURE 3.11-A.	...58
4.1	A line of text from Pas-1 written by subject 28:	...75
5.1-A	Design of Single Character Recognition Subsystem (SCRS).	...78
5.1-B	3-Ordered choices for letter A; output from SCRS operating on testing set from Munson's data.	...79

6.1	Context Aided Text Recognition System (detailed design).	..94
6.2	Illustration of Modified Viterbi Algorithm.	..102

CHAPTER 1

INTRODUCTION

1.1 THESIS ORGANIZATION

This thesis comprises of eight Chapters, an Appendix and a Bibliography.

Chapter 1, which introduces the text recognition problem, starts by giving a note on organization of the thesis in this section. Section 1.2 describes the basic components of a typical character-recognition system and investigates how the need for the context was developed. The mathematical model for the problem is derived in section 1.3. The text-recognition system proposed, is outlined in section 1.4.

Later, each chapter opens with its preview, which gives an overall picture of the chapter. Therefore, we will not describe the first section of chapters 2 through 8 any further.

The second chapter reviews the field of character recognition. Section 2.2 describes the approaches and the techniques developed in the past for character-recognition. Section 2.3 reviews the previous research in the use of the contextual information as an aid to character-recognition systems. The contextual algorithms developed so far are described in Section 2.4.

Chapter 3 describes the characteristics of the data sets and methods used to transform character images into a form suitable for recognition system. Munson's data and preprocessing done on its patterns are described in section 3.2. Training of the recognizer using preprocessed patterns is also described in this section. Similar information about OCR 'A' and OCR 'B' data sets is provided in sections 3.3 and 3.4, respectively.

In chapter 4, the three passages and the tests conducted on them are described. The information about source of passages used, and some of their significant characteristics are given in section 4.2. Section 4.3 includes two statistical tests made on the three passages. The procedure, how the passages were set up for text recognition experiments, is described in section 4.4.

Chapter 5, describes the components and the functions of SCRS. In section 5.2, this stage of classification and the experiments conducted are further described. Confusion Matricies built from Munson's and OCR 'B' data sets are shown in section 5.3.

Chapter 6 describes the third stage of the proposed text recognition system and some of the algorithms used in the same. The detailed description of this stage is given in section 6.2. Modified Viterbi algorithm is mentioned in section 6.3. The experiments conducted using this algorithm

are described in section 6.4.

Some more contextual algorithms used as CPPS are described in Chapter 7. Section 7.2 describes Block Decoding Algorithm (BDA) and Imbricate Block Decoding Algorithm (IBDA). The experiments conducted using them and the results thus obtained are also described in the same section. A dictionary method and the experiment conducted using it, are described in section 7.3. Section 7.4 includes the description of a hybrid method. An experiment conducted using this method is also described in the same section. A discussion on the results obtained from the experiments is included in section 7.5.

Chapter 8, is the last chapter and reviews the experiments conducted in this thesis. Section 8.2 reviews the results of the experiments and derives conclusions from them. In section 8.3 several suggestions for the improvement of the present system are described. Some guidelines for further study and research are presented in section 8.4.

The appendix at the end contains the three passages that were used in text-recognition experiments. The research materials referenced in this thesis are listed in the bibliography.

1.2 CONTEXTUAL TEXT RECOGNITION SYSTEM

The last two decades were a period of a rapid growth for pattern recognition technology. During this period many papers appeared on pattern classification, training procedures, picture processing algorithms, cellular recognition machines and the application of recognition technology. Among these applications were optical character recognition, scene analysis, finger-print identification, analysis of bubble chamber tracks, analysis of blood cells, vector cardiograms, and the mapping of chromosomes and lunar landscapes [T02-74].

The field of character recognition has been a popular and challenging subject in which the researchers are actively interested. This includes the recognition of machine-printed and hand-printed characters.

A typical character recognition system is shown in FIGURE 1.1 It is made up of three major components, (1) the character transformation device, (2) the information selector, and (3) the recognition logic.

The character transformation device scans the character and converts it into a digital or logical image. Making measurements, the information selector extracts features from these images. The recognition logic examines the features to determine the identity of the character.

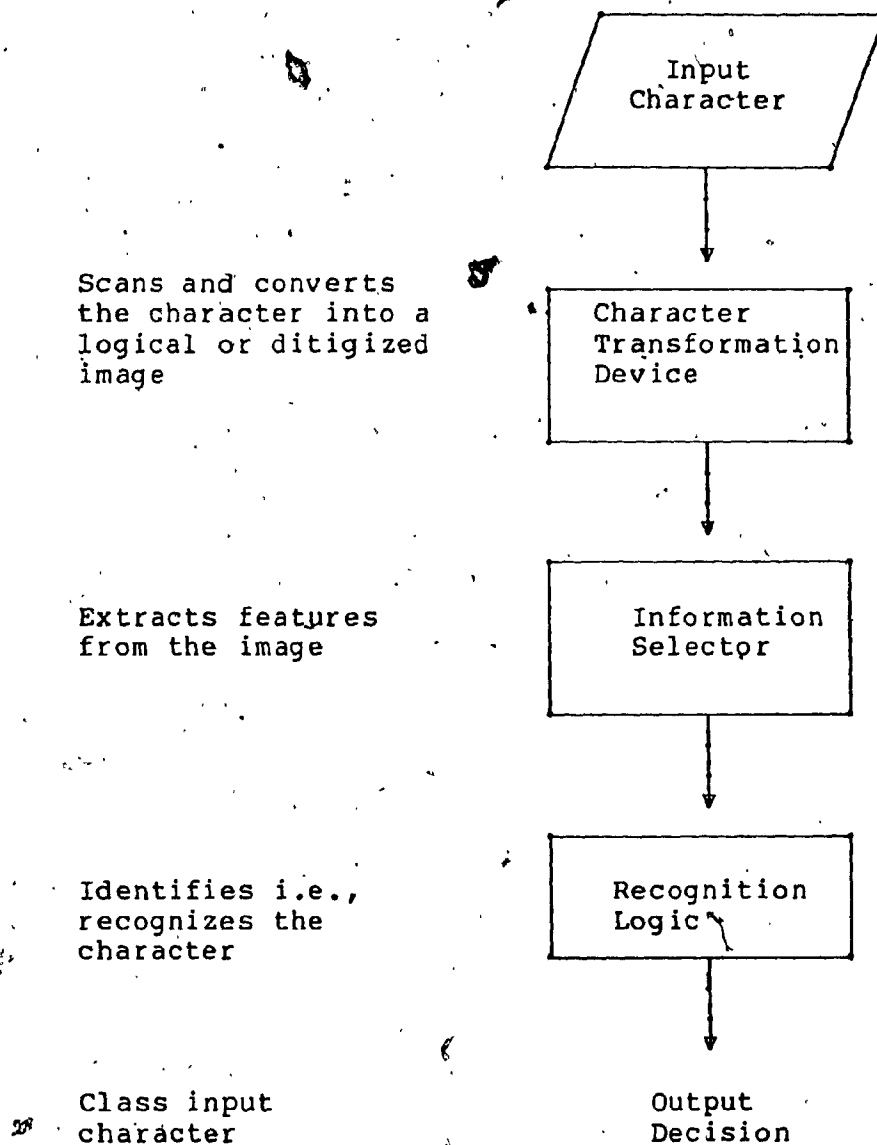


FIGURE 1.1: Basic Components of a Recognition System.

The research efforts in optical character recognition can be listed under two categories: (1) the recognition of machine-printed characters, and (2) the recognition of hand-written characters. Encouraging results in recognition of individual characters motivated some researchers to examine the character sequences in a text in natural language or a program in computer language. These efforts used contextual information as an aid to the recognition logics and proved to be very successful, and hence demand detailed investigation for practical applications. Some researches [CHU-75, SH1-77, T05-77] investigated the amount of contextual information required by a system as well.

The use of context (i.e., Contextual Information) to improve the recognition of characters in a text figures prominently in this thesis. The technique was briefly discussed by Bledsoe and Browning [BLE-59] but otherwise received scant attention in the past. Some studies have been carried out under simplifying assumptions of Markov dependence in character-pairs (bigrams) and character-triplets (trigrams); for example. An early progress report was given by Munson [MUN-68]

Munson [MUN-68] and Duda and Hart [DUD-68] had developed algorithms to read handprinted FORTRAN programs. Some studies to recognize character sequences in natural languages have been reported by Hanson, Riseman and Fisher [HAN-76], Shinghal and Tousaint [SH3-79, SH4-79], and Vossler and Branston

[VOS-64].

To be useful, a text recognition system must be general, simple, efficient, economical and should give the low error rate. Therefore, the following objectives were set to develop the text recognition system, described in this thesis.

1. To search for simple character recognition algorithms.
2. To make efficient use of information about letter combinations supplied by linguistic experts and computer scientists. That is to include the use of contextual information as an aid to recognition.
3. To examine the flexibility and effectiveness of different contextual algorithms.
4. To search for an optimal text recognition system which performs equally well on different texts and type-fonts.
5. To reduce the costs of these algorithms.

1.3 MATHEMATICAL MODEL FOR RECOGNITION

Researchers [DUD-73] have shown that Bayes decision rule is an optimal statistical approach to the problem of pattern recognition. Recognition of text can be considered as a problem of recognizing letter sequences constituting the text. One way of solving this problem is to recognize individual character by making a decision on a sequence of them. Thus if an individual character be recognized, so would the character sequence and hence the entire text.

Assume that $X_0, X_1, \dots, X_n, X_{n+1}$ be the feature-vectors belonging to patterns comprising a sequence. The recognition process associates each pattern X_i ($0 \leq i \leq n+1$) with one of the possible character classes of Z_i (say). If Z_i is identical to the name of the pattern-class of X_i , then X_i is recognized correctly, else it is misrecognized.

Let $P(X_0, \dots, X_{n+1}/Z_0, \dots, Z_{n+1})$ denote the probability of sequence X_0, \dots, X_{n+1} conditioned on Z_0, \dots, Z_{n+1} and $P(Z_0, \dots, Z_{n+1})$ denote the a priori probability of character sequence Z_0, \dots, Z_{n+1} .

To classify X_0, \dots, X_{n+1} , we need to maximize that character-sequence which maximizes posteriori probability $P(Z_0, \dots, Z_{n+1}/X_0, \dots, X_{n+1})$ or a monotonic function of it, e.g., its log.

Using Bayes theorem we obtain:

$$\begin{aligned} \log P(Z_0, \dots, Z_{n+1}/X_0, \dots, X_{n+1}) &= \\ \log P(X_0, \dots, X_{n+1}/Z_0, \dots, Z_{n+1}) + \log P(Z_0, \dots, Z_{n+1}) \\ &\quad - \log P(X_0, \dots, X_{n+1}) \end{aligned} \quad \dots 1.3-1$$

where the term $P(X_0, \dots, X_{n+1})$ is the probability of input sequence and does not vary with (Z_0, \dots, Z_{n+1}) . Therefore, maximizing $P(Z_0, \dots, Z_{n+1})$ is equivalent to maximizing $g(X)$ where

$$\begin{aligned} g(X) &= \log P(X_0, \dots, X_{n+1}/Z_0, \dots, Z_{n+1}) \\ &\quad + \log P(Z_0, \dots, Z_{n+1}) \end{aligned} \quad \dots 1.3-2$$

Thus X_i can be classified by using information from all other patterns preceeding and following X_i . Storing $P(X_0, \dots, X_{n+1}/Z_0, \dots, Z_{n+1})$ distribution needs a large memory. For example, consider that each feature X_i has 'q' elements, each of which further take on 'k' values. The values X_i can take on is equal to k^q . The number of values, sequence of 'n' characters can take on will then be equal to

$$(k^q)^n$$

Further each Z_i can have 26 possible classifications; the term $P(X_i, \dots, X_{n+1}/Z_0, \dots, Z_{n+1})$ in equation 1.3-2 can thus have $26^n (k^q)^n$ values.

For a typical case when $n = 5$, $k = 17$, $q = 20$; we see this is large number.

To reduce this amount of memory, conditional independence is assumed. Thus this assumption is made, which yields

$$\log P(X_0, \dots, X_{n+1}/Z_0, \dots, Z_{n+1}) = \sum_{i=0}^{n+1} \log P(X_i/Z_i) \quad \dots 1.3-3$$

The assumption of conditional independence is valid, since cursive script [T05-77] is not used in this thesis. Thus 1.3-2 becomes:

$$g(X) = \sum_{i=0}^{n+1} \log P(X_i/Z_i) + \log P(Z_0, \dots, Z_{n+1}) \quad \dots 1.3-4$$

The term $P(X_i/Z_i)$ will be called likelihood in later references. Its estimation can be simplified by assuming the class conditional independence among the elements of feature-vector X_i , if $X_i = (X_{i1}, X_{i2}, \dots, X_{iq})$ then

$$\log P(X_i/Z_i) = \sum_{j=1}^q \log P(X_{ij}/Z_i) \quad \dots 1.3-5$$

Estimation of this term requires excessive computations if all sequences, being constructed from each character-class of Z_i , are considered. One feasible method of its estimation is to assume English language to be a Markov source. Details of this are discussed in chapter 6.

Maximizing expression 1.3-4 is the basis of the algorithms discussed in this thesis. It will be used at many places in the following chapters.

1.4 THE PROPOSED TEXT RECOGNITION SYSTEM

Since researchers [CHU-75, SH1-77, T05-77] have indicated that the use of context in the form of probabilities of character combinations can appreciably improve the recognition of characters in the text, the proposed system also uses context. Recognition of individual characters in a word can be considered as the basis for any general text-recognition system. Therefore, to simulate the simplest case, the system considers upper case alphabetic characters only. Blanks (or spaces) are used to distinguish the word

boundaries, other punctuation marks and special symbols have been eliminated.

The experiments conducted are based on the assumption that the system correctly classifies the spaces between the words and then successively locates the letters within the word. The text for the experiments was on three English language passages. These passages are shown in Appendix - A. The passages were set up using different hand-printed and machine-printed data-sets.

Usually the contextual text recognition systems comprise two stages. The first stage recognizes individual character and following the terminology of Schurmann [SCH-76] we also named this stage as Single Character Recognition Subsystem (SCRS). This stage is used as an aid to the second stage. The second stage improves the performance of text-recognition system by using context. Some researchers called this stage as contextual postprocessing system.

Figure 1.2 outlines the components of the system we proposed. The system is divided into three stages. Stage-1, includes preprocessing followed by extraction of features from the patterns. Stage-2, called as single character recognition subsystem (SCRS), recognizes a character or gives several alternatives of that. English text is input to this system. If the performance of this system is to the satisfaction of user, onward processing of the text can be terminated at this

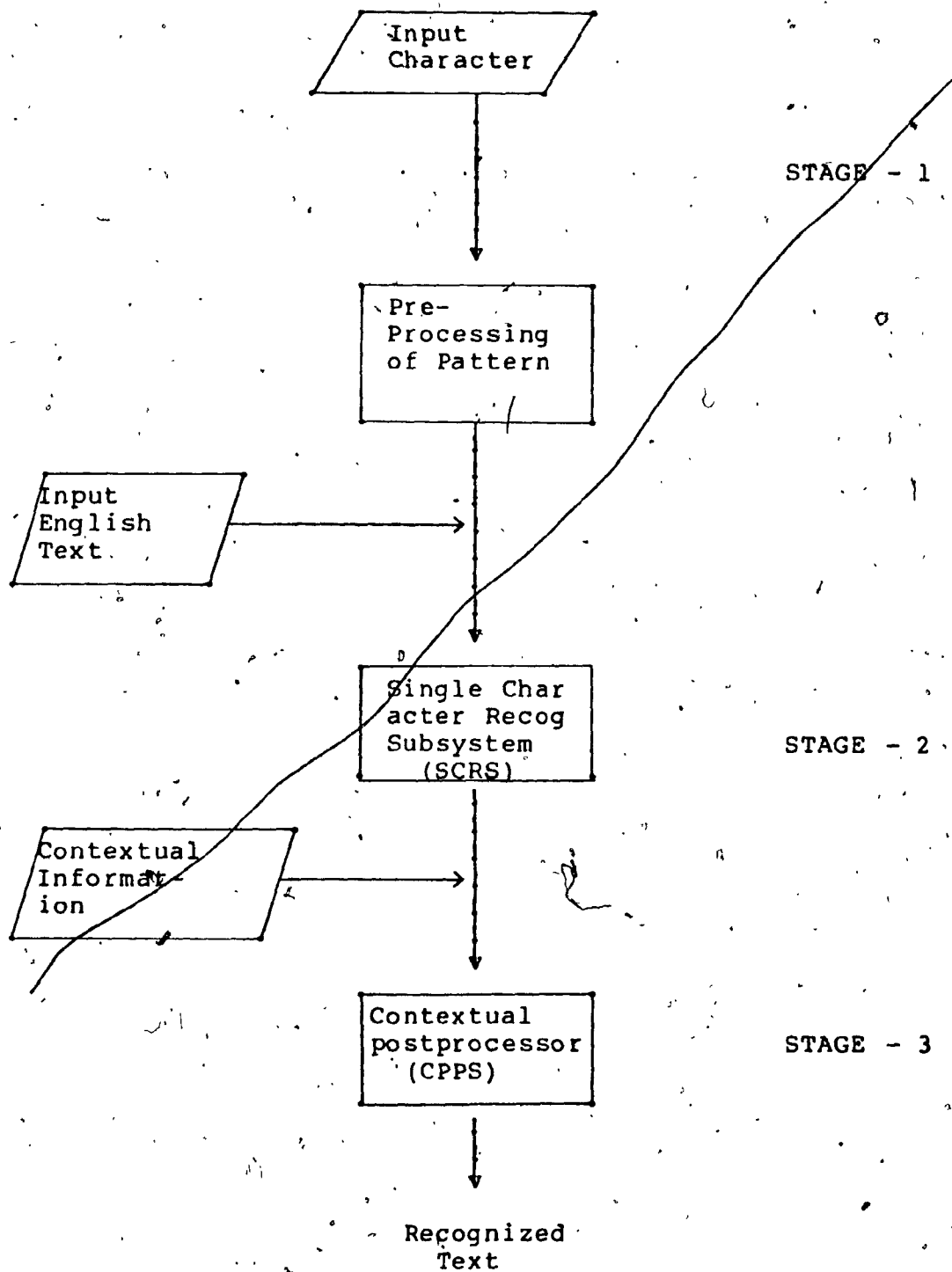


FIGURE 1.2: Outline of the Proposed Text recognition system

stage. Stage-3 is a context-aided text recognizer and it is referred to as Contextual Postprocessing Subsystem (CPPS). The output from this subsystem is the English text as recognized by the system.

Details of the recognition system are given in the following chapters.

CHAPTER 2

HISTORICAL REVIEW

2.1 PREVIEW OF THE CHAPTER

This chapter is divided into three more sections. Section 2.2 reviews the field of character recognition in general. Section 2.3 specifically reviews the work done on the use of contextual information in text recognition. An expanded review on contextual algorithms is presented in section 2.4.

2.2 CHARACTER RECOGNITION

Character recognition is a major topic in the area of pattern recognition. Fu [FU1-68] defined pattern recognition as a field that is concerned with machine recognition of a group or a single object on the basis of certain subjective requirements. Pattern recognition research blossomed in the early 1960's and has enjoyed over two decades of vigorous growth. Contributions to the growth have come from many disciplines, including statistics [DUD-73, SEB-62], communication theory [BRA-64], switching theory [NUM-68], control theory [COO-64], operations research [MUR-62], biology [BED-71], philosophy [MKE-41], psychology [KOF-35], and linguistics [NAR-62].

The problem of pattern recognition, in general, involves the tasks of study, modelling and construction of mechanisms that analyze, detect, recognize and describe patterns in analogue or digital data. In fact the problem of pattern recognition has been considered as one of discriminating the input data, between different populations through the search for features or distinguished attributes among members of a population [SKL-73].

In early efforts we find geometrical methods of pattern recognition; for example, shape and live patterns [DIN-55, SEL-58]. Sherman [SHE-59] used a quasi-topological method for the recognition of line patterns while Dimond [DIM-57] developed a topological process for reading hand-printed numerals, provided they are drawn in accordance with certain constraints that restrict size, proportion and location. Greanias et al. [GRE-63] described a system of recognizing member of a sixteen character alphabet (numerals and some miscellaneous printer symbols).

Unger [UNG-59] devised a binary feature vector system for geometric and topological classification based on answering a set of "yes" or "no" questions with respect to any given input pattern. He successfully tested his recognition system on a number of hand-printed alphanumeric characters. Doyle's [DOY-60] scheme was based on two notions: (a) the decision was reserved until all test results were tabulated, and (b) adaptive discrimination was based on learning from

real data.

The different mathematical techniques used to solve pattern recognition problems can be grouped into two general approaches; namely, the decision theoretic (or discriminant) approach [FU2-74] and the syntactic (or structural) approach [KAS-76]. In the decision-theoretic approach, a set of characteristic measurements (called features) is extracted from the patterns and the recognition of each pattern is made judiciously by partitioning the feature space [FU1-68]. Surveys on this subject have appeared in [DEU-75, KAN-74, SUE-78].

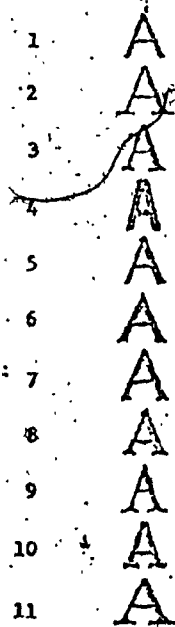
The syntactic [NAR-66] approach to pattern recognition provides a capability for describing a large set of complex patterns by using small set of simple pattern primitives and a set of geometrical rules. For example Knoke [KNK-67] considered pattern recognition as a problem of translating string languages and devised a program similar to a table-driven or variable-syntax compiler for the recognition of hand-printed characters and line drawings.

Toussaint [TO4-77] has discussed the problems underlying the testing and training phases of different statistical pattern recognition methods currently in use. He considered feature extraction as two independent sub-problems: search and evaluation. An exhaustive bibliography of this field and some details about statistical approaches to current problem

solving methods, both parametric and nonparametric is given in [T03-74].

Glucksman [GL2-71] introduced an economical method for deciding the class of a pattern described by a vector of measured features. He used characteristic loci features and was able to achieve 94.017% correct recognition on IBM data base having 9 fonts and 52 pattern classes. A non-parametric method for designing linear discriminant functions for pattern classification was formulated by Smith et al. [SME-72] as a linear programming problem. Glucksman [GL1-67] also described a measurement scheme for machine-printed characters of eleven different fonts (see FIGURE 2.1) by using a linear separation algorithm.

For any recognition system to be practical, it should be able to tolerate a certain amount of distortion in the character set. Ullman [ULL-72] described several rules and methods of dealing with distortion. A recognition technique which attempted to cope with the problems of misrecognition, distortion and variation of stroke thickness associated with hand-printed character was reported in [HOS-68]. The technique was improved by using a contour tracing method and a two stage classification scheme [HOS-72]. This scheme when tested on OCR 'B' font character set (a machine-printed data-base of English language character set) resulted in 36 rejects and 5 errors among 4431 characters.



1	A	Gill Sans
2	A	IBM Executive
3	A	IEEE Text
4	A	News Gothic
5	A	Lydian
6	A	Arbutus
7	A	Westminster Old Style
8	A	Cheltenham Bold
9	A	Garamond
10	A	Scarab
11	A	Fortune

FIGURE 2.1: Eleven fonts used by Glucksman.

Nugent [NUG-67] presented a method for machine recognition of cursive writing. This method used a simplified feature extractor. It partitioned symbols into strokes and categorized these strokes by geometric measures and their topological relations to previous strokes.

Owing to their significant dis-similarity even among the same class, hand-written characters gave a poor recognition rate by techniques like area matching [ULL-73] and cross-correlation methods [LUG-64].

Finding template matching inadequate, Grenander [GRE-70] used elastic templates. An alternate rubber-template approach is given by Widrow [WID-73].

Other systems of print-recognition such as scanning n-tuple [KAM-63, PAR-68], central moments [ALT-62], Fourier analysis [JON-62] and holographic techniques [GAB-71] have achieved varying degrees of success and commercial exploitation, but are also inappropriate for the recognition of hand-printed characters, because these methods perform better if the characters are well-written. Parks et al. [PAR-68] demonstrated the effectiveness of analogue processing techniques in dealing with printed characters of low quality. Some curve following techniques for hand-printed character were described in [BAR-60]. Parks [PAR-69] described a successful recognition system using two sequential processes of image transformations: (1) the first process detected and

mapped the position of short continuous line segments of specified orientation and reduced the representation of the image to binary form and (2) the second process detected and mapped the position of morphological features by operating upon the line segment maps. The morphological feature types were: (1) line endings (2) changes of direction of line segments (3) junctions of two or more segments and (4) crossovers of any combination of the three types of features.

The literature on character recognition is dominated by considerations of implementation [FIS-62, HOL-68]. Literature also exists on special-purpose electronic and optical image processing system [BED-71, DUF-73, MCO-63, UNG-58]. Some descriptions of special hardware for pattern recognition applications are found in Unger [UNG-58], McCormick [MCO-63] and Duff et al. [DUF-73].

2.3 THE USE OF CONTEXT IN TEXT RECOGNITION

Several natural languages own a unique characters set and a number of punctuation symbols. A text in such languages is constructed, if several sequences of one or more characters be combined with some punctuation symbols. Other languages like Chinese constitute text by combining several primitive strokes (we called them characters for sake of generality).

Considering the recognition of text as pattern recognition problem, the task could be done if the characters

and the punctuation symbols are correctly classified. Machines which recognize text by recognizing individual characters and symbols will make errors on the text, if a character is not recognized correctly. The question arises: would the use of context help in reducing these errors? We will investigate the answer to this question.

Context is "the parts of a written or spoken communication which precede or follow a letter, word, sentence, or passage, and affect its meaning [WEB-74]". Psychological studies have shown that single letter can be perceived better when they are parts of words [WHE-70].

Endeavors for text recognition, particularly in the English language have invoked the use of contextual information in the algorithms [CHU-75, SH1-77, T05-77].

Contextual information in the form of estimates of probability distribution of occurrence of single character (unigrams), character pairs (bigrams) and character triplets (trigrams) has been gathered in [T06-78]. These probability distributions are given a general name N-grams. In [HAN-76, RAV-67, RIS-71, SH4-79, SUE-79, T05-77], it is shown that N-grams are useful in machine recognition of English text. Efforts have also been made in automatic correction of texts, namely by, deletion, substitution, and insertion of characters in text [CAR-66, MIL-57, T05-77].

Suen [SUE-79] reported that none of the existing commercial OCR (Optical Character Recognition machines make much use of contextual information. Several methods have been tried for machine recognition of text [SH3-79]. In one of the approaches of machine recognition of English text, a word is recognized first [SH3-79]. A question that arises at this point is, what kind of information is contained in a word that a machine might be able to make use of? Toussaint [T05-77] distinguished between six classes of word features: graphological, phonological, statistical, syntactical, semantic, and pragmatic.

Graphological features are those that characterize the shape of a word or the shape of letter-cluster within a word. Phonological features are present because humans have an idea of what sounds to expect in certain situations, as in the acoustic similarity of rhyme. Statistical features are present in words to the extent that the more frequently occurring words are more familiar to us and are more easily recognized [ABO-59]. Syntactic features consists of markers or declaraction of parts of speech. Semantic features are markers indicating the words, i.e., taxonomic categories. Pragmatic features consist of information about how the user uses a word. In case of cursive script some of the pragmatic features are intimately related to some of the graphological features.

Contextual information can also be expressed in terms of syntactic rules. Kashyap and Mittal [KAS-76] used many such syntactic rules in speech recognition.

Another approach to the utilization of contextual information in text recognition is the table (dictionary) look-up method. In this method the words conforming the text are compared with those already stored in a dictionary in the computer memory through string-to-string matching [ALB-67, WAG-74]. The drawback of such an algorithm is the combinatorial explosion in computations. Hybrid approaches use both dictionaries and the N-gram probabilities [SH3-79].

The various approaches for text recognition which have so far been examined by researchers can thus be categorized into three groups: dictionary look-up methods, methods using N-gram probability distributions, and hybrid methods.

2.4 CONTEXTUAL ALGORITHMS: PAST APPROACHES

In this section the three approaches mentioned earlier in Section 2.3 will be discussed in further detail.

2.4.1 Dictionary Look-up Methods

The dictionary-look-up method is based on the assumption that every word in the text is selected from a dictionary. A word of text is classified by either of the following methods:

1. By string-to-string matching with every word in the

dictionary having the same length (number of letters in a word) as the input word has, and finding the best match [ALB-67].

2. Computing some discriminant function for every letter in the word. The algorithm then computes scores using the same discriminant function for all same length words stored in the dictionary. The word giving highest score is the output word designated by the algorithm [BLE-59].

Gold [GOL-59] used the first method in his system for recognizing hand-sent Morse codes. One of the earliest application of the second method to text recognition was by Bledsoe and Browning [BLE-59]. Other examples of similar approaches are Abe and Fukumura [ABE-71], Carlson [CAR-66], Cornew [COR-69], Shinghal et al. [SH3-79], and Vossler and Branston [VOS-64].

2.4.2 Probability Distribution Methods Using N-grams

Markov assumption for text representation was used for using N-gram probabilities. This approach is based on the assumption that true class of a character is related in a probabilistic manner to the true class of a small number of surrounding characters. Its use leads to the estimation of the probabilities of possible single characters (unigrams), character pairs (bigrams), character triplets (trigrams) or in general N-grams of characters from English texts. For instance, it can be assumed that the probability of occurrence

of each character depends on the v ($v \geq 0$) previous characters, therefore, one has to estimate these probabilities (e.g., $P(A, B)$) from the frequencies of $(v+1)$ -character combinations.

Harmon [HAR-62] used this method to detect errors in the recognition of cursive script. Edwards and Chambers [EDW-64], and Carlson [CAR-66] also employed this technique in character recognition. Raviv [RAV-67] and Abend [ABN-68] independently derived the formal decision theoretic solution for the optimum use of contextual information at syntactic level. Raviv [RAV-67] also made some simplifying assumptions. He assumed Markov dependence among characters to be recognized and then used sequential compound decision theory to make a decision on one character at a time (sequential contextual decoding). Duda and Hart [DUD-68] treated the alternative choices for the character in question with their confidences as the output of the classifier to make a decision on a short sequence of characters (contextual block decoding). Toussaint and Donaldson [TOI-72] applied a simple suboptimal block contextual decoding algorithm to hand-printed character recognition which searched over only the most probable bigrams and trigrams. This reduced significantly the amount of computation without severely reducing the recognition accuracy. A discussion of the sequential decoding and hybrid decoding methods is given in [CHU-75, TOI-72]. These algorithms are dependent on three functions: namely, "depth of search",

"holding", and "threshold". These functions not only determine the complexity of decoding but also the amount of contextual information to be used.

Based on Markov assumptions, an algorithm known as Viterbi algorithm [FOR-73, NEU-75] has been used for contextual recognition of text. Chung [CHU-75] and Shinghal et al. [SH4-79] have experimented with a modified version of the Viterbi algorithm.

2.4.3 Hybrid Methods

These methods use a dictionary, and N-gram approaches such as bigrams. For example, McElwain and Evens [MEL-55] used such approach to correct garbled Morse codes. One hybrid approach was considered by Riseman and Ehrich [RIS-71]. They proposed the use of bigrams to eliminate from dictionary search, a large number of sequences formed from the character alternatives. Shinghal and Toussaint [SH3-79] have also introduced a method based on this combined approach. They called this method the Predictor-Corrector algorithm and experimentally showed that it achieves the error-correcting capability of the dictionary look-up method at half the cost.

CHAPTER 3

PREPROCESSING, FEATURE EXTRACTION AND TRAINING

3.1 PREVIEW OF THE CHAPTER

In this chapter we describe the main characteristics of the data sets used. In an attempt to satisfy the objective that a text recognition system should perform well on different type-fonts, three different character sets were selected. The character sets included, one hand-printed and two machine-printed ones. Hand-printed characters were selected from Munson's multiauthor data base [MUN-68]. The two machine-printed character sets were selected from OCR 'A' and OCR 'B' data bases. The source and specifications of Munson's data are stated in section 3.2. The patterns contained in this data set were of different sizes and had noise. Therefore, a preprocessing scheme was adopted to include these two measures; namely: (a) elimination of noise and (b) size normalization. Sections 3.2.1 and 3.2.2 describe the methods adopted for carrying out these two measures. The method for feature extraction is given in section 3.2.3. Section 3.2.4 states the training procedure selected for the recognizer. The characteristics of OCR 'A' and OCR 'B' machine-printed data sets are described in Sections 3.3 and 3.4 respectively. Methods, similar to those described in sections 3.2.1 through 3.2.4, adopted for these

data set are also included in sections 3.3 and 3.4.

3.2 MUNSON'S DATA SET OF CHARACTER PATTERNS

Complete details of this data set were already described by Munson in [MUN-68]. In this study we selected the digitized patterns of only the 26 upper-case characters 'A' to 'Z'. The characters had been hand-printed by 49 subjects. Each subject printed 3 specimens of each one of the 26 letters, thus giving $3 \times 26 \times 49 = 3822$ characters. These character imprints were unconstrained and untutored. The binary images of the characters, each on a 24×24 grid were stored on a magnetic tape.

Some typical images of characters from the data-base are shown in FIGURE 3.1. In following three subsections, we will describe the preprocessing scheme and the method for feature extraction.

3.2.1 Noise Elimination

By observing characters shown in FIGURE 3.1, one can notice the distortion among them. In addition to the size, there are notches and bumps along the boundaries of the characters. These discrepancies in a character were given a common name: noise. In order to provide an effective and efficient description of patterns, smoothing was often required to remove noise.

AA			EEE
AAAA			EEEE EE
AAAAA			EEEE E
AAAAA			EEEEEE
AAAAA			EEEE
AAA AA			EE
AAA AA			E
AAA AA			E
AAA AAA			EE EE E E
AAAAAAAAAAAA	BBBB		E EE EEEEE
AAAAAAAAAAAA	BBBBBBBB		EEEEEEEE E E
AAAAAAAAAAAA	BBBBBB BB		EEEE E
AAA AAA	BBBB BBBB		
AA AAA	BBB BBB		EE
AA AAA	BBBB BBBB		EE
AA AAA	BBBBBBBB		EE
AA AAA	BBB BBB		EE
AA AAA	BBB BB		EE EEE EEE
AA AAA	BBBB BB		EE EEEEE E
AA AAA	BBB BBB		EEEEEEEE
AA AAA	BBB BBBB		EEEE
AA AAA	BBBB		EEEE
AA AAA	BBB		

FIGURE 3.1: Some typical characters from Munson's data base.

Smoothing consisted of two operations; namely: thinning, and filling. Thinning is the process of replacing a black point with a white point while Filling is the process of replacing a white point with a black point in the input pattern.

There are two approaches to smoothing. One is to carry out the preliminary smoothing operation on the input fields prior to pattern classification. Another is to incorporate smoothing into the recognition logic by making patterns insensitive to variations confined to any small areas in them. A number of algorithms belonging to these approaches have been suggested by Dineen [DIN-55], Doyle [DOY-60], Freyer and Richmond [FRE-62] and Unger [UNG-59]. Deutsch [DEU-72], Stefanelli and Rosenfeld [STE-71], and Triendl [TRI-70] have presented algorithms for thinning and skeletonization.

It was felt that algorithms described by the above researchers would not meet the requirements of the recognition system of this thesis. However, our smoothing scheme is based on the methods proposed by Unger [UNG-59]. The terminologies used in the implementation of the scheme are defined below:

The points comprising a pattern are black and white points.

S-window:

is a small window of size 3 X 3 used in filling and thinning operations of smoothing. This window is shown in

FIGURE 3.2.

Mid-point:

The point X shown in FIGURE 3.2 is called the mid-point of the window.

Neighbours:

The points surrounding the mid-point are called its neighbours. For example, the points a, b, c, d, e, f, g and h are the neighbours of the mid-point X.

Corner:

Three adjacent points, all not along the same line form a corner of a window. For example, the points b, c and e constitute one corner of S-window.

Hole:

It is a white point surrounded by a minimum of 4 black points.

Directions:

In usual geographical terminologies the point X has 8 directions; namely in order: East (E), North-East (NE), North (N), North-West (NW), West (W), South-West (SW), South (S), and South-East (SE).

L-window:

is a large window of size 5 X 5. S-window is a proper subset of it, and fits in the middle with one point along the sides. It was used in detecting hole in the pattern by considering 2 immediate neighbours of point X in eight directions. The L-window and eight directions numbered as 1 through 8 are shown in FIGURE 3.3.

a	b	c
d	x	e
f	g	h

FIGURE 3.2: S-window: the points involved in smoothing operation.

4	NW			3	N			2	NE
		a'	y	a'	y	a'			
		y	a'	a'	a'	y			
5	W	a'	a'	x	a'	a'		1	E
		y	a'	a'	a'	y			
		a'	y	a'	y	a'			
6	SW			7	S			8	SE

FIGURE 3.3: L-Window: points involved in Hole detection.

In above the squares marked as a' are the points of consideration. The square marked as y did not take an active part in the algorithm.

C-Hole:

A hole is called as C-Hole if it has at least another adjacent white point and is surrounded by black points in a minimum of 6 directions.

The objectives of smoothing were the following seven points.

1. Check the C-hole before thinning.
2. Eliminate isolated black points by changing them to white.
3. Eliminate small bumps along straight line segments.
4. Check C-hole before filling.
5. Fill in the isolated holes in otherwise black areas.
6. Fill in small notches in straight line segment.
7. Patch the broken character.

Typical bumps, notches, holes, C-holes and breaks etc., are shown in FIGURE 3.4.

Since the characters contained in Munson's data are very noisy, therefore, it was decided to include thinning and then filling in the smoothing procedure. To achieve the smoothing objectives listed above, the procedures S-1 through S-3 were developed. They are described below.

It is to be noted that the pattern of a character was processed by moving a S-window across the pattern. The movement of S-window was sequential, since every point in the pattern takes part in the processing.

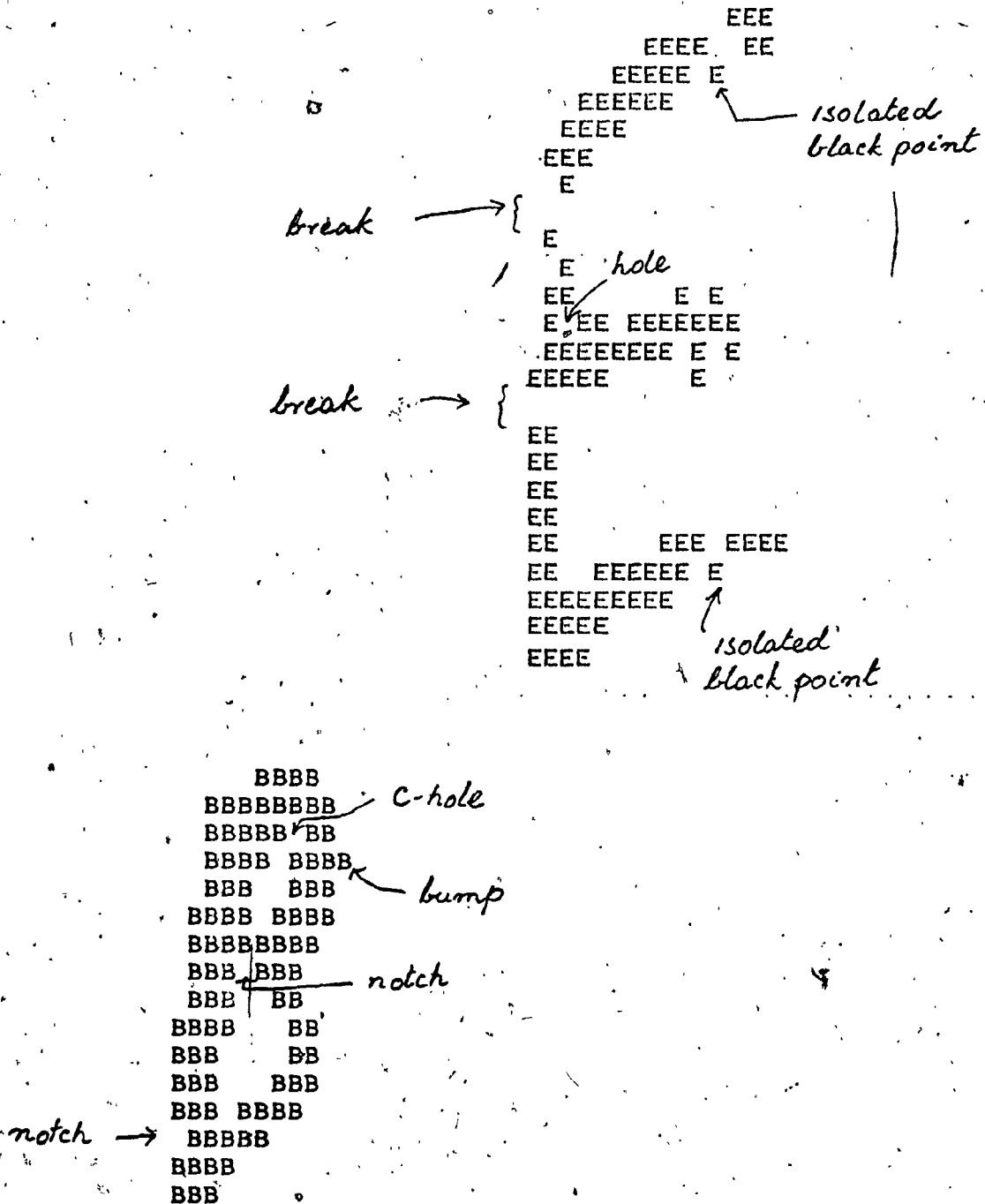


FIGURE 3.4: Points of interest in smoothing scheme.

S-1: Thinning Procedure

Comments: Subject to the conditions specified, the procedure replaces a black point with a white point. The objectives 1, 2 and 3 of smoothing (given above) are accomplished by the procedure.

Steps:

- a1) Assume that point 'X' is being examined.
- b1) Count the number of black points in S-window.
- c1) If the count is greater than or equal to a preset threshold (it was set to 7 for Munson's data) exit, i.e., do not eliminate the point.
- d1) Check the possible existence of a C-hole by invoking the boolean function HOLTHIN, which is described below.
- e1) If the value of HOLTHIN is true exit, i.e., the point X is a C-hole, so it should not be eliminated.
- f1) Compute the boolean thinning function. This function has been taken from [UNG-59].

$\text{thinfun} := X [(a + b + d)(e + g + h) + (b + c + e)(d + f + g)]$

- g1) If the value of thinfun is true exit;
otherwise set
 $X := \text{white};$ and exit.

The steps f1 and g1 above, explained that the point 'X' is changed to white if the points contained in at least 2 adjacent corners are white.

- i1) Repeat steps a1 through g1 for all points contained in the pattern.

Function HOLTHIN (X)

Comments: The function determines whether the point under examination is a C-hole.

Steps:

a1) Set HOLTHIN := false;

b1) Find the count of black points in 8 directions using L-window i.e., count number of a (see FIGURE 3.3).

c1) Using the result in b1 above, find directions count in which there is least one black point.

d1) Find the number of corners.

e1) If the directions count is greater than or equal to 6 and exactly 2 corners in NW (North-West) and SE (South-East) directions are present then
set HOLTHIN := true;

In this step, 6 and 2 are preset thresholds decided after a number of trials to retain very thin limbs of a character.

f1) Exit.

S-2: Filling Procedure

Comments: The procedure replaces a white point with a black point in the pattern, if the conditions laid down are satisfied. The objectives 4 through 6 of smoothing (given above) are accomplished by the procedure.

Steps:

a2) Assume that point X is being examined.

- b2) Count the number of black points in S-window.
- c2) If the count is greater than 7, or less than or equal to 2 then go to step e2, i.e., do not invoke the function HOLFIL.
- d2) Check the possible existence of C-hole by invoking the boolean function HOLFIL, which is described below.
- e2) If the value of HOLFIL is true then exit, i.e., the point 'X' is a C-hole. So it should not be filled.
- f2) Compute the boolean filling function:

$$\text{fillfun} := X + b g (d + e) + d e (b + g)$$
 This function is taken from [UNG-59].
- g2) If the value of the function fillfun is not true then exit; otherwise
 set $X := \text{black}$;
 and exit;
 This step explains that point 'X' is changed from white to black, if the four points i.e., b, d, e and g are black.
- 12) Repeat steps a2 through g2 for all points contained in the pattern.

Function HOLFIL (X)

Comments: This function determines whether the point under examination is a C-hole.

Steps:

- a21) Set HOLFIL := false;
- b21) Find the count of black points in 8 directions using L-window.

c21) If the count is less than 13 then exit, i.e., the point is not a C-hole and it can be filled. The number 13 denotes that black points are present in minimum 6 directions.

d21) Otherwise increase the size of hole by changing

g := white;

h := white; and also set

HOLFIL := true;

e21) Exit.

S-3: Patching Procedure

Comments: The procedure replaces the current blank line (row or column) with the previous non-blank line if the line following the current line is also non-blank. The last objective of Smoothing Procedure (given above) (i.e., the 7th) is accomplished by the procedure.

Steps:

- a3) Locate, if there is any, a blank row or column in the grid in between two consecutive non-blank rows or columns.
- b3) Map the previous row or column over this blank line.

FIGURE 3.5 shows the noise-eliminated pattern of the pattern given in FIGURE 3.4.

3.2.2 Size Normalization

Following noise elimination, the patterns were subjected to size-normalization operation. This operation increases or

BBBB
BBBBBBBB
BBBBBB BB
BBBBBB BB
BBB BBB
BBBB BBBB
BBBBBBBB
BBBBBBBB
BBB BB
BBBBBB BB
BBB BB
BBB BBB
BBBBBBBB
BBBBBB
BBBB
BBB

FIGURE 3.5: Noise eliminated pattern of the one given in
FIGURE 3.4.

reduces the height and breadth of a pattern to match certain preset grid. Size-normalization was useful in reducing the sensitivity of the feature extraction scheme (described in section 3.2.3) to the variations in size of the pattern in the grid.

Prior to this study a number of researchers have already adopted different size normalizing schemes [HUS-72, NAG-70, SH1-77]. For example, Hussain et al. [HUS-72] used a mathematical function based on the height and breadth of the character to increase (by duplication) or decrease (by elimination) rows and columns in the image.

Following this approach the normalization method adopted in this thesis also used a mathematical function. The normalizing process involved two steps in the following order: first normalizing height; second normalizing breadth. It is to be noted that the order, in which first and second steps are taken, is not important. Before describing the algorithm formally, we will define several keywords used in the algorithm.

Line:

Any row or column in the grid is called a line.

Left-most Column (L):

The first column of the grid with at least 2 black points.

Right-most Column (R):

The last column of the grid with at least 2 black points.

Top-most Row (T):

The first row of the grid with at least 2 black points.

Bottom-most Row (B):

The last row of the grid with at least 2 black points.

Breadth:

Total number of columns occupied by the pattern on the grid; i.e.,

$$\text{Breadth} = R - L + 1.$$

Height:

Total number of rows occupied by the pattern i.e.,

$$\text{Height} = B - T + 1.$$

Size

The height multiplied by the breadth, used in the manner for matrix representation, gives the size of the pattern. Thus 13 X 17 is the size of a pattern whose breadth is 13 and height is 17.

Start Position:

The point of intersection of variables T and B gives the Start Position of the pattern in the grid.

STANDB:

It is the fixed threshold value for breadth normalization. For one kind of data-set this value is fixed.

STANDH:

It is the fixed threshold value for height normalization. For one kind of data set this value is also fixed.

NLAB:

It is the number of lines to be deleted or added in

breadth.

NLAH:

Number of lines to be deleted or added in height.

The size-normalizing algorithm was then the following:

A. Height Normalization

The steps involved are the following.

Steps:

1. Assume the STANDH is given; which was considered as 24 for Munson's data set.
2. Locate L of the input pattern.
3. Compute the height HIGH of the pattern by counting the number of lines in the grid beginning from start position until the bottom-most row is reached.
4. Find the quantity NLAH, i.e.,

$$NLAH = STANDH - HIGH \quad \dots 3.2-1$$

If NLAH is positive then NLAH designates the number of lines to be added. If it is negative, that many lines have to be deleted.

5. Compute the step size by using the expression:

$$S = \lceil HIGH / (NLAH + 1) \rceil + 1 \quad \dots 3.2-2$$

$\lceil \rceil$ designate the ceiling of the number enclosed in $\lceil \rceil$.

6. Positive sign of the variable S implies that every S-th row has to be repeated (added). The negative sign designates the deletion of every such row.

7. Depending upon the sign of the variable 'S', add or delete every S-th row and take the count of the number of lines repeated or deleted.
8. If the magnitude of NLAH is equal to the count then terminate the operation and the resulting pattern is height-normalized, otherwise go to step 4, recompute the step size using new values of NLAH and HIGH and repeat steps 6 through 8. The height-normalized pattern of the one given in FIGURE 3.5 is shown in FIGURE 3.6.

B. Breadth Normalization

The steps involved are the following.

Steps:

1. Assume that STANDB is given, which was considered as 24 for Munson's data set.
2. Locate T of the height-normalized pattern.
3. Compute the breadth BRDTH of the pattern by counting the number of lines in the grid beginning from the start position until the right-most column is reached.
4. Find the quantity NLAB; i.e.,

$$NLAB = STANDB - BRDTH \quad \dots 3.2-3$$

If NLAB is positive then NLAB designates the number of lines to be added. If this number is negative that many lines have to be deleted.

5. Compute the step size by using the expression:

$$S = \lceil BRDTH / (NLAB + 1) \rceil + 1 \quad \dots 3.2-4$$

```

      BBBB
    BBBBBBBB
    BBBBB BB
    BBBBB BB
    BBB BB
    BBB BB
  BBBBB BBBBB
  BBBBBBBB
  BBBBBBBB
  BBBBBBB
  BBB BB
  BBB BB
  BBB BB
  BBBBB BB
  BBB BB
  BBB BB
  BBB BBB
  BBBBBBBB
  BBBBBBBB
  BBBBBB
  BBBB
  BBBB
  BBB

```

FIGURE 3.6: Height-normalized pattern of the one given in FIGURE 3.5.

6. As before, the positive sign of the variable S implies the addition of every S -th column; i.e., requirements for increase in size. Negative sign implies the deletion of S -th column i.e., reduction in size.
7. Depending upon the sign of the variable S , add or delete every S -th column and take the count of the number of lines repeated or deleted.
8. If the magnitude of $NLAB$ is equal to the count then terminate the operation and the resulting pattern is both height and breadth normalized (or size-normalized); otherwise go to step 4; recompute the step size using the new values for $NLAB$ and $BRDTH$ and repeat the steps 6 through 8.
9. If the height to the breadth ratio i.e., $HIGH/BRDTH$ is greater than 3 then breadth of the character is retained, but the height was already normalized. This arrangement prevented the letter ϕ printed without serifs from being converted into virtually all-black comprising over the whole grid.

FIGURE 3.7 shows the size-normalized pattern.

3.2.3 Feature Extraction

This section described the method of extracting the features from preprocessed and size-normalized patterns. A feature can be defined as a measurement made on a pattern. Usually several such measurements are made on a pattern. A

地。

set of such measurements or features is referred to as a feature-vector. A good set of features results in a good discrimination between the pattern classes, but it is not very sensitive to minor variations in the different patterns of same pattern class. To save on computations, it is important that feature-vector be of low dimensionality and its extraction from the pattern be simple.

Most of the methods like characteristic loci [GL1-67, KNL-69], contour tracing [CLE-68], and topological methods [MUN-68] suffer from the complexity of feature extraction scheme, high dimensionality of feature-vector or both.

For explanation purpose consider the problem of the recognition of hand-written characters. The usual discriminatory features are the sequences of strokes, the direction of strokes, the arrangement of strokes. These are generally not easy to measure [PAR-68].

In this thesis, a simple and low-dimensional feature extraction scheme was used. The scheme was the following:

The STANDB X STANDH grid of the smoothen and size-normalized pattern was divided into 'q':

$$q = (\text{STAND} / \text{DIV}) \times (\text{STANDH} / \text{DIV})$$

non-overlapping regions; each of DIV X DIV squares. In our experiments the value of DIV was 4 and for Munson's data

1.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

FIGURE 3.9: A typical 4 X 4 region: count of black points in 6 squares make one feature element in the vector space.

STANDB = STANDH = 24. FIGURE 3.8 shows these regions for Munson's data. A typical region is shown in FIGURE 3.9.

The feature-vector F

$$F = [f_1, f_2, \dots, f_q]$$

was obtained, where f_i , $1 \leq i \leq q$ is equal to the number of black points in the i -th region. FIGURE 3.8 shows the ordering of $q = 36$ regions in 24×24 grid used for Munson's patterns.

This method of feature extraction has the following advantages:

1. It is simple as it involves only a count of black points in the region. Thus it is easy to implement on the machine.
2. Dimensionality of feature-vector is 'q' which is presumed to be not very large. In the case of Munson's data the value of q is 36.
3. Since each region is 4×4 i.e., there are 16 points in the region. Each region constitutes an element of the vector. An element takes on the values between 0 and 16; that is, 17 possible values; which is not a very large number either (see FIGURE 3.9).
4. Small salt-and-pepper noise and minor discontinuities in the image can be tolerated.

Feature vectors were extracted for every pattern in the data set, in the manner described above. The feature vector for each character was stored on magnetic tape for later use

in the recognition process.


3.2.4 Training of the Classifier

As described in section 1.1, it was necessary to design the recognition algorithm to achieve satisfactory performance. The performance of the recognizer is supposed to improve as more and more patterns are observed. This process is called training or learning and the patterns used as input are called the training patterns.

The training procedure could either be supervised or unsupervised. The distinction in both is that, with supervised learning one knows the state of nature (class name) for each pattern, whereas with unsupervised learning it is not the case [ULL-72].

In the present work a supervised training scheme was used. The scheme was implemented by estimating the likelihoods (see section 1.2) for the samples available in the data set.

The 3822 sample patterns of Munson's data were separated into two groups: 3744 patterns written by the subjects 1 through 49 (excluding the subject 28) constituted the training set. The 78 characters written by subject 28 constituted the testing set. Subject 28 was arbitrarily chosen for testing purpose.



In order to know the identity of a character during the training phase, each character was labelled with the name of its class, i.e., labelled as 'A', 'B', ..., 'Z'. These patterns were used in computing a 3-dimensional $17 \times q \times 26$ matrix 'm'; we called it the training matrix such that

$$m_{ijk} = \log P(f_j = i/C_k) \quad \dots 3.2-5$$

for $0 \leq i \leq 16$; $1 \leq j \leq q$; and $2 \leq k \leq 27$.

where 'q', the size of feature vector was 36 in case of Munson's data as discussed in the beginning of this section. C_1 was blank and C_2 to C_{27} were correspond to character names 'A', 'B', ..., 'Z'. To avoid zero entries, Bayesian estimates of likelihoods were estimated.

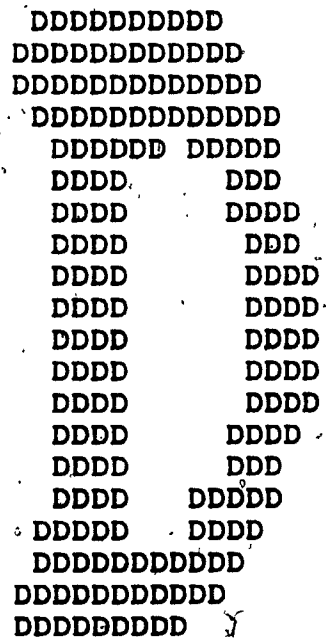
For any Feature vector $F = [f_1, f_2, \dots, f_q]$ the expression $\log P(F/C_k)$ can be computed from the training matrix.

3.3 OCR 'A' PATTERNS

To test the algorithms on machine-printed data as well, we selected OCR 'A' and OCR 'B' data sets. OCR 'A' data set was designed by American National Standards Institute [OCA-77]. Only upper case, digitized patterns of 26 letters 'A' to 'Z' were selected. There were 5791 patterns altogether. Some sample patterns are shown in FIGURE 3.10-A. Each sample was contained on 24×16 grid.

DDDDDDDDD
 DDDDDDDDDDD
 DDDDDDDDDDDDD
 DDDDDDDDDDDDD
 DDDDD DDDDD
 DDD DDD
 DDD DDDDD
 DDD DDD
 DDD DDDDD
 DDD DDDDD
 DDD DDDDD
 DDD DDDDD
 DDD DDDDD
 DDD DDDDD
 DDD DDD
 DDDDD DDD
 DDDDDDDDDDD
 DDDDDDDDDDD
 DDDDDDDDD

FIGURE 3.10-A: Some typical patterns from OCR 'A' data set.



DDDDDDDDDDD
 DDDDDDDDDDD
 DDDDDDDDDDD
 DDDDDDDDDDD
 DDDDD DDDDD
 DDDD DDD
 DDDD DDDD
 DDDD DDD
 DDDD DDDD
 DDDD DDDD
 DDDD DDDD
 DDDD DDDD
 DDDD DDDD
 DDDD DDD
 DDDD DDDD
 DDDDD DDDD
 DDDDDDDDDDD
 DDDDDDDDDDD
 DDDDDDDDD

FIGURE 3.10-B: Size-normalized pattern of character 'D' of
FIGURE 3.10-A.

As this data was single-font machine-printed data, we skipped the noise elimination step. In well written data sets noise elimination merely affects a few points, and thus will add to computations in preprocessing.

Size normalizing algorithm was the same as described in section 3.2.2 with a few exceptions:

1. In majority of samples the 4 bottom-most rows were blank. Therefore, before normalization we eliminated those rows.
2. The value of the variable STANDH was 20.
3. The value of the variable STANDB was 16.

The size-normalized pattern of character 'D' is shown in FIGURE 3.10-B.

The feature extraction method was also the same as described for Munson's data. Since the characters were normalized to a smaller size; obviously, the number of non-overlapping regions was less. The value of 'q' i.e., the number of elements in the feature vector was 20. Thus the size of the training matrix was 17 X 20 X 26.

Steps involved in training of the recognizer were essentially the same as described earlier except the training set was constituted of 5141 samples and the testing set included 650 samples. The frequency distribution of characters both in training and testing sets is given in TABLE 3.1.

TABLE - 3.1

Frequency Distribution of characters in training
and testing set of OCR 'A' data set

Character	Total number of such samples in	
	Training set	Testing set
A	199	25
B	199	25
C	199	25
D	199	25
E	199	25
F	199	25
G	198	25
H	198	25
I	199	25
J	199	25
K	199	25
L	198	25
M	197	25
N	195	25
O	198	25
P	196	25
Q	197	25
R	197	25
S	198	25
T	197	25
U	194	25
V	196	25
W	197	25
X	198	25
Y	198	25
Z	198	25

Total

5141

650

3.4 OCR 'B' PATTERNS

Another data set of a total of 2834 sample patterns of upper case letters 'A' to 'Z' was selected from OCR 'B' data base. There were 109 samples of each of letters 'A' to 'Z'. A sample pattern from the data is shown in FIGURE 3.11-A. This data was designed by European Computer Manufacturers Association [OCB-71].

Again in this data set the noise elimination step was skipped as it was single-font machine-printed data. Other specifications for size normalization and training were the same as for OCR 'A' data set. A sample of size-normalized pattern is shown in FIGURE 3.11-B. The training set, however, contained 18 samples of each of 'A' to 'Z'. The testing set contained 91 samples of each of the letters 'A' to 'Z'.

These preprocessed and size-normalized patterns of the characters contained in data sets described above were then used in setting up the hand-printed and machine-printed passages. The source passages and the method for setting-up are discussed in Chapter 4.

```

.AAAA
AAAAAA
AAAAAA
AAAAAAA
AAAAAAA
AAAAAAA
AAAAAAA
AAAA AAA
AAAA AAAA
AAAA AAAA
AAAAAAA
AAAAAAA
AAAAAAA
AAAAA AAAA
AAAA AAAA
AAAA AAAA
AAAA AAAA
AAA AAA

```

FIGURE 3.11-A: Atypical pattern of character 'A' from OCR 'B' data set.

```

AAAAA
AAAAAAA
AAAAAAA
AAAAAAA
AAAAAAA
AAAAAAA
AAAAAAA
AAAA AAAA
AAAA AAAA
AAAA AAAA
AAAAAAA
AAAAAAA
AAAAAAA
AAAAA AAAA
AAAAA AAAA
AAAAA AAAA
AAAAA AAAA
AAAAA AAAA
AAA AAAA
AAA AAA

```

FIGURE 3.11-B: Size-normalized pattern of character of FIGURE 3.11-A.

CHAPTER 4

COMPILATION OF PASSAGES FOR TEXT RECOGNITION

EXPERIMENTS

4.1 PREVIEW OF THE CHAPTER

The text recognition system described in this thesis was tested on three passages printed in five (3 subjects of Munson's data, plus two machine printed data sets) character-fonts, which were already described in sections 3.2 through 3.4. In section 4.2, a brief description of the passages used is given. Chi-square tests conducted to compare three passages are described in section 4.3. In these tests the probability distribution of single characters (unigrams) and character-pairs (bigrams) estimated by Toussaint and Shinghal [TO6-77] is used. These estimates are shown in TABLE 4.1 and TABLE 4.2 respectively. The method for setting-up the passages for text recognition experiments is described in section 4.4.

4.2 PASSAGES

The experiments described in this thesis were conducted on three passages of English text. The first passage selected was a short story written by Tolstoy. The second passage was chosen from Gadsby by Wright [WRI-39]; the third was chosen from Time newsmagazine of November, 1977. These passages will

TABLE - 4.1

Unigram Bayesian Probabilites $\times (10^{**7})$

Unigram		Probability $\times 10^{**7}$
.	-----	1732081
A	-----	644937
B	-----	124776
C	-----	269053
D	-----	307978
E	-----	1031249
F	-----	203876
G	-----	148126
H	-----	443164
I	-----	634895
J	-----	15867
K	-----	44715
L	-----	334645
M	-----	207024
N	-----	586861
O	-----	632405
P	-----	172685
Q	-----	11307
R	-----	500323
S	-----	551057
T	-----	772827
U	-----	224216
V	-----	90287
W	-----	144696
X	-----	18715
Y	-----	143455
Z	-----	8706

TABLE - 4.2
BIGRAM BAYESIAN PROBABILITIES * (10 ** 7)

	A	B	C	D	E	F	G	H	I	J	K	L	M
	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	198838 39109	79660 136165	84790 71246	51193 3678	44943 45080	67617 109084	27399 288961	76187 20037	137324 13862	8902 99476	8127 192	39006 11834	68912 462
A 46141	139 123503	14940 449	27269 11984	22426 152	1158 68111	4530 58506	11847 92803	1425 7007	22240 14240	846 4781	7011 709	71041 15529	15600 537
B 1305	8286 218	836 11746	182 87	299 87	40709 6815	146 3017	91 1383	211 12648	6998 585	2102 110	87 87	15424 10968	247 87
C 10305	33362 130	107 51294	5142 156	146 439	44787 9058	159 1044	94 30836	36650 9038	17921 91	91 87	6141 91	8833 2828	110 100
D 168883	7841 722	208 14262	152 166	2418 126	47485 4726	309 8254	2294 156	1025 10025	29733 1399	244 524	97 87	1956 3544	1243 87
E 357642	47358 93425	1419 4501	29896 10500	73277 3310	23991 129538	11209 88819	6851 24258	2021 2773	12655 17797	442 6884	1230 13426	34143 10308	23282 286
F 85294	11086 133	94 31719	91 97	87 87	14894 13104	10148 397	91 4875	94 7017	19187 87	91 97	94 87	4065 667	94 87
G 47638	9943 4508	94 9090	87 91	273 87	22501 11343	169 3752	2141 699	15779 4006	10210 87	91 117	139 87	3176 1057	859 87
H 45802	67269 1497	231 34993	143 117	234 94	211028 5334	208 1129	159 9569	120 4426	53383 91	87 328	104 87	859 4586	1188 87
I 10679	16788 157309	6389 52452	47443 4778	19555 696	24519 21221	15867 81714	16860 76988	231 742	823 19050	133 107	3114 1194	30189 133	21710 4202
J 182	1995 97	87 3300	94 100	87 87	3603 91	87 87	87 87	91 4774	130 87	87 87	87 87	87 87	87 87
K 11496	917 4231	143 331	117 113	159 87	13800 572	152 2965	224 130	198 345	6968 87	87 169	110 87	572 426	126 87
L 59782	2997 345	358 24844	494 1018	16925 91	51642 849	4511 9452	423 7115	166 7727	40546 2717	87 979	1044 94	41138 30716	1451 123

TABLE - 4.2 (CONTD.)

	A	B	C	D	E	F	G	H	I	J	K	L	M
	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
M	28630	35296 641	4416 21864	276 16310	107 94	50209 1581	302 6278	133 120	166 8967	20522 94	87 100	110 87	5686 91
N	152189	22361 5093	335 33759	29440 462	85580 624	43690 872	4215 35605	58181 65368	745 3837	21853 3570	816 445	3844 227	1728 185
O	75233	2965 112378	6760 13296	9322 14940	14044 130	2620 83706	78504 20349	5829 29827	1253 58943	5585 11017	488 22107	3710 774	34475 520
P	8065	18341 787	149 23568	113 8557	97 87	31738 29307	156 4000	91 6438	7987 6330	7681 91	87 152	117 87	1148 87
Q	133	87 87	87 87	87 87	87 87	87 87	87 87	87 87	87 87	87 87	87 87	87 87	87 87
R	98008	40325 9471	1484 41067	6838 2900	122579 97	2112 25899	5663 23432	1344 9540	1344 9540	44250 4319	276 813	5572 263	11785 110
S	231547	15272 1360	361 25704	10148 12397	660 507	62467 331	1233 26670	195 66416	19630 20108	39329 113	87 1419	2747 87	5305 104
T	161736	32220 732	266 67402	1406 250	169 91	71845 26286	458 22904	152 11353	248017 15887	83882 117	94 5074	110 87	1581 289
U	6447	7964 25124	5621 699	12153 9129	6708 97	9706 32607	1015 31614	8680 27038	126 139	6132 198	104 87	390 240	23555 309
V	312	8407 87	5006	117 87	91 87	56322 126	87 117	87 94	87 214	17762 87	87 91	87 87	87 91
W	14536	23669 5106	152 15102	110 100	231 87	24476 1998	240 2184	87 354	29020 130	24623 91	87 91	240 87	208 87
X	1673	1721 91	87 250	1676 4774	87 110	1155 97	136 97	87 2776	270 247	2444 100	87 94	87 97	87 87
Y	107772	1012 602	299 9042	1363 2141	266 87	6011 452	117 2923	195 1559	107 149	1952 107	87 273	130 87	1728 172
Z	527	1471 87	97 305	94 87	91 87	3280 87	91 100	87 104	120 175	670 104	91 107	91 87	175 253

be referred to as Pas-1, Pas-2, and Pas-3 respectively.

Even though 'E' is the most frequent letter in English text [T06-77], in Pas-2 the letter 'E' does not occur at all. The third passage Pas-3 is assumed to be a representation of newsmagazine writing. These three passages, each from different source and topic, were purposely selected in order to observe the effect of using context. These passages were reformatted by deleting all punctuation marks and special symbols and leaving only a blank between words. All numeric information was translated into equivalent words. For example, numeral '2' was translated into 'two'. The passages are shown in APPENDIX - A. The distribution of characters in the passages is shown in TABLE 4.3.

4.3 COMPARISON OF PASSAGES

In order to compare the usage of English language and to study the difference in writing styles of writers two chi-square tests were conducted. With the help of chi-square test procedure one can decide whether to accept or reject the null hypothesis. Null hypothesis is an assumption that the number of samples examined belong to the same population and in statistical notion denoted as H_0 . Correspondingly the other possibility is called the alternative hypothesis H_1 [KUS-77].

TABLE - 4.3

Frequency Distribution of Characters in
Three Passages

Character	Tolstoy	Gadsby	Time
A	82	119	99
B	14	24	28
C	37	38	18
D	53	38	42
E	142	0	144
F	15	22	30
G	20	24	21
H	81	58	76
I	73	113	77
J	1	0	1
K	6	6	15
L	50	46	35
M	32	27	27
N	78	98	83
O	96	115	74
P	23	27	17
Q	1	1	0
R	61	46	66
S	94	83	86
T	76	106	99
U	33	42	17
V	15	3	4
W	15	24	17
X	1	0	3
Y	10	33	23
Z	0	0	4

Let the 27 characters i.e., 'a' and 'A' to 'Z' be represented by C_1, C_2, \dots, C_{27} . The experiment measured, for the three passages, the frequency of occurrence of character C_i and character-pairs (C_i, C_j) for $(1 \leq i \leq 27)$. Based on the results of this experiment chi-square test was conducted to investigate whether the three passages satisfy two hypotheses: H_{01} and H_{02} , details of which are given below:

4.3.1 Test Hypothesis H_{01}

a) Using Unigram

To determine "whether these passages represent commonly used English"; a test was performed. The test uses chi-square distribution with hypothesis:

the frequency of occurrence of characters in each of the passage is from the same distribution.

Assume that:

O_{jk} = Observed frequency of occurrence of character C_j in passage k ; for $1 \leq k \leq 3$.

E_{jk} = Expected frequency of character C_j in passage k .

$N_k = \sum_{j=1}^{27} O_{jk}$ = Total number of characters in passage k .

E_{ij} was estimated as follows:

$$E_{jk} = P(C_j) \times N_k + 0.5 \quad \dots 4.3-1$$

for $1 \leq k \leq 3; \quad 1 \leq j \leq 27$.

It is assumed that probabilities shown in TABLE 4.1 are unigram probabilities of commonly used English. Therefore, $P(C_j)$ can be replaced by Unigram (j). Expression 4.2-1 can thus also be written as:

$$E_{jk} = \text{Unigram}(j) \times N_k + 0.5 \quad \dots 4.3-2$$

In above 0.5 was added as correction factor.

The test requires to compute:

$$\chi^2 = \sum_{j=1}^{27} (O_{jk} - E_{jk})^2 / E_{jk} \quad \dots 4.3-3$$

for $1 \leq k \leq 3$.

Under the hypothesis H_{01} , the value of chi-square was computed for all three passages. These values, listed in TABLE 4.4, were compared with the tabulated values of chi-square, with degrees of freedom shown in the table, at a significance level of less than 0.001 and arrived at the following decision:

Pas-1: Do not reject the null hypothesis H_{01} , i.e., the frequency of occurrence of characters in this passage follows the distribution function specified by the unigram.

Pas-2: Reject the null hypothesis H_{01} , i.e., the frequency of occurrence of characters in this passage do not follow the distribution function specified by the unigram probabilities.

Pas-3: Do not reject the null hypothesis, i.e., the frequency of occurrence of characters in this passage follow the distribution function specified by the unigram probabilities.

As shown above we did not reject the null hypothesis, if the test is made on Pas-1 and Pas-3. This concludes that English text represented by these two passages could be considered as in widely usage English. The hypothesis is rejected for Pas-2. Intuitively, this could be because letter 'E' does not occur in it at all, whereas it is the most frequent letter.

b) Using Bigram

Hypothesis H_{01} was tested using probabilities of character-pairs (bigrams) i.e., $P(C_i, C_j)$. Thus expression 4.3-1 will be changed to:

$$E_{ijk} = P(C_i, C_j) \times N_k + 0.5 \quad \dots 4.3-4$$

where

E_{ijk} = Expected frequency of character-pairs C_i, C_j in passage k. Correspondingly

O_{ijk} = Denotes the observed frequency of character-pair C_i, C_j in passage k.

As in the test above, we assumed the probabilities shown in TABLE 4.2 to be the bigram probabilities of commonly used English. Therefore, $P(C_i, C_j)$ can be replaced by

TABLE - 4.4

H₀₁ : Passages represent commonly used English?

Probability	Chi-square Values		
	Pas-1	Pas-2	Pas-3
Unigram	45.71394 (22)*	170.71230 (20)	53.90481 (21)
Bigram	335.17533 (119)	590.69703 (117)	266.74794 (122)

* quantities enclosed within parantheses are the degrees of freedom.

Bigram (i,j). Expression 4.3-4 can thus be written as:

$$E_{ijk} = \text{Bigram}(i,j) \times N_k + 0.5 \quad \dots 4.3-5$$

The test thus requires to compute:

$$\chi_k^2 = \sum_{i=1}^{27} \sum_{j=1}^{27} (O_{ijk} - E_{ijk})^2 / E_{ijk} \quad \dots 4.3-6$$

Under the hypothesis H_{01} , the value of chi-square was computed for all three passages. These values are listed in TABLE 4.4. These values were compared with the tabulated values of chi-square, with degrees of freedom shown in the table, at significance level of less than 0.001. The decision derived is same as the one made when unigram probabilities were used.

That is:

"Do not reject the null hypothesis H_{01} for both passages Pas-1 and Pas-3. This decision implies that frequency of occurrence of character pairs in these passages follow the distribution function specified by the bigrams,"

and

"Reject the null hypothesis H_{01} for Pas-2, which implies that frequency of occurrence of character pairs in this passage do not follow the distribution function specified by the bigrams."

Since using bigram probabilities we arrived at the same decision as was made in case of unigrams, therefore, we conclude the same in this case as well.

4.3.2 Test Hypothesis H_{02}

a) Using Single Character Frequencies (Unigrams)

This test determines whether the passages are mutually similar. The test needs to find whether the distribution of the frequency of occurrence of characters in one passage is different from that in other passage. Thus the test was performed using chi-square test with the hypothesis:

the distribution describing frequency of occurrence of characters in passage 'k' is the same as that in passage '1'.

Since the test involves relationship between the passages, for $1 \leq k, 1 \leq 3; i \neq j$, we will be using contingency tables.

In this test, the joint expected frequency E_{ki} was estimated by the equation:

$$E_{ki} = [(O_{ki} + O_{1i}) / (N_k + N_1)] \times N_k + 0.5 \quad \dots 4.3-7$$

for $1 \leq k, 1 \leq 3; k \neq 1$ and $1 \leq i \leq 26$

where

O_{ki} = Observed frequency of occurrence of character C_i in passage k .

O_{1i} = Observed frequency of occurrence of character C_i in passage 1.

N_k = Total number of characters in passage k .

N_1 = Total number of characters in passage 1.

The test thus requires to compute:

$$X^2 = \sum_{i=1}^{27} (O_{ki} - E_{ki})^2 / E_{ki} \quad \dots 4.3-8$$

Using this expression chi-square was computed for all passages pair-wise i.e., $1 \leq k, 1 \leq 3; k \neq 1$. They are shown in TABLE 4.5. The smallest value of $X^2 = 1439.75$ was for Pas-1 and Pas-3 with 24 degrees of freedom. This computed value of X^2 is still much higher than the corresponding tabulated value of X^2 at 24 degrees of freedom at significance level of less than 0.001. Thus the hypothesis was rejected for all passages pair-wise.

Rejection of this hypothesis implies that the distribution describing frequency of occurrence of characters, in passage 'k' is different from that in passage 'l'. It is thus concluded that the passages are not mutually similar.

b) Using Character Pairs Frequencies (Bigrams)

The same hypothesis was tested using frequencies of character pairs (C_i, C_j) for $1 \leq i, j \leq 27$. Thus expression 4.3-7 will now be changed to

$$E_{kij} = [(O_{kij} + O_{lij}) / (N_{ki} + N_{li})] \times N_{ki} + 0.5 \quad \dots 4.3-9$$

for $1 \leq k, 1 \leq 3; k \neq 1$, and $1 \leq i, j \leq 27$.

Thus the test requires to compute:

$$X^2 = \sum_{i,j=1}^{27} (O_{kij} - E_{kij})^2 / E_{kij} \quad \dots 4.3-10$$

T A B L E - 4.5

Ho2 : Passages are mutually similar?

	Chi-square Values		
Frequency distribution	pas-1 & Pas-2	Pas-2 & Pas-3	Pas-3 & Pas-1
Single Character	2780.01437 (23)*	2780.65165 (25)	1439.75108 (24)
Character Pairs	4889.23115 (617)	9302.92315 (608)	15670.82715 (452)

* quantities enclosed within parantheses are the degrees of freedom.

Using this expression, χ^2 was computed for all passages pair-wise. The values and the degrees of freedom of each pair are shown in TABLE 4.5. Again the computed values are much higher than corresponding tabulated values with the degrees of freedom specified in the table at a level of significance less than 0.001. This implies the rejection of the hypothesis for all passages pair-wise.

We thus conclude that the distribution describing frequency of occurrence of character-pairs in passage 'k' is different from that in passage 'l' which implies that the passages are not mutually similar.

4.4 SETTING-UP PASSAGES FOR TEXT RECOGNITION EXPERIMENTS

Since the passages have different frequency distribution of characters and also belong to different source (as concluded in section 4.3), it was decided to use these passages to be hand-written and printed by the letters of Munson, OCR 'A', OCR 'B' data sets.

Subjects 6 and 18 from training set of Munson's data were arbitrarily chosen. Subject 28 was already chosen as testing set.

Each passage was set-up with the characters of a subject from Munson's data by randomly selecting one out of three

specimen characters from each of 'A' to 'Z'. The reason, behind doing so, was to simulate the normal handwriting style.

Thus the passages were set-up in 3 different handwritings of subjects 6, 18, and 28 respectively. These passages will be referred to as:

Pas (1,Mun ₆)	Pas (1,Mun ₁₈)	Pas(1,Mun ₂₈)
Pas (2,Mun ₆)	Pas (2,Mun ₁₈)	Pas(2,Mun ₂₈)
Pas (3,Mun ₆)	Pas (3,Mun ₁₈)	Pas(3,Mun ₂₈)

Thus Pas (1,Mun₆) is the passage of letters written by subject 6.

FIGURE 4.1 shows one line of text from Pas (1,Mun₂₈). The subscript show the number of specimen (1 to 3) of the letter chosen.

In a similar fashion by randomly choosing 26 characters from training and testing sets of both OCR 'A' and OCR 'B' data the 3 three passages were set-up. They will be referred to as:

Pas (1,OCRA)	Pas(1,OCRB)
Pas (2,OCRA)	Pas(2,OCRB)
Pas (3,OCRA)	Pas(3,OCRB)

In the following chapters the text recognition experiments conducted using set-up passages mentioned above will be described.

I₃N₁ O₃L₁D₂E₃N₃ T₁I₃M₂E₃S₁ L₃O₁N₁G₂ L₃O₁N₃G₁ B₂E₁F₃O₁R₃E₂
T₁H₂E₁ C₂O₁M₂I₂N₃G₂ O₂F₂ C₂H₁R₃I₃S₃T₂

FIGURE 4.1: A line of Text from Passage Pas-1 written by subject 28.

CHAPTER - 5

SINGLE CHARACTER RECOGNITION SUBSYSTEM

5.1 PREVIEW OF THE CHAPTER

Following preprocessing and feature extraction, the second stage proceeds. This is the stage where the preliminary recognition or the classification of an input character is made. This stage is called as Single Character Recognition Subsystem or SCRS. Section 5.2 describes it in more detail. To study the misclassification of input patterns belong to different data sets confusion matrices were constructed. Section 5.3 explains how this information was gathered.

5.2 SINGLE CHARACTER RECOGNITION SUBSYSTEM (SCRS)

As outlined in FIGURE 1.2, Single Character Recognition Subsystem (SCRS) was the key stage of the proposed text recognition system. This is the stage where preliminary classification of the input pattern was done. In expression 1.3-4 for a given X_i ($1 \leq i \leq n$), Z_i can be any of the 26 possible letters 'A' to 'Z'. In SCRS the values of likelihoods i.e., the term $P(X_i/Z_i)$ are computed over these 26 letters for X_i . Of these 26 values, the d-highest are chosen as 'd' alternatives of X_i . The number of alternatives is a variable and will be called as 'depth of search'. The

depth of search will be denoted by 'd'. The subsystem does not allow any rejection at all and thus indicates that all the legitimate pattern classes are acceptable. Since we have already assumed that blank is perfectly recognizable, therefore, by legitimate classes we mean that any of the characters from 'A' to 'Z'. The design of the subsystem is given in FIGURE 5.1-A, and a sample of output from SCRS is given in FIGURE 5.1-B.

Let us consider the problem of recognizing input feature vector sequence X_0, \dots, X_{n+1} . From expression 1.3-4

$$g(X) = \sum_{i=0}^{n+1} \log P(X_i/Z_i) + \log P(Z_0, \dots, Z_{n+1})$$

Since SCRS is a maximum selector and considers every character-independent to each other, therefore, the second term in the above expression is a constant, thus we maximize:

$$g'(X) = \sum_{i=0}^{n+1} \log P(X_i/Z_i) \quad \dots 5.1-1$$

Expression 5.1-1 is like maximizing without using context. Again X_0 and X_{n+1} are considered as blank, thus one need to maximize:

$$g'(X) = \sum_{i=1}^n \log P(X_i/Z_i) \quad \dots 5.1-2$$

Therefore, the functions of SCRS can be formalized as below:

1. Assume that feature vector of an input pattern X_i is given.
2. Compute 26 likelihoods for class k , $k=2,3,\dots,27$, using

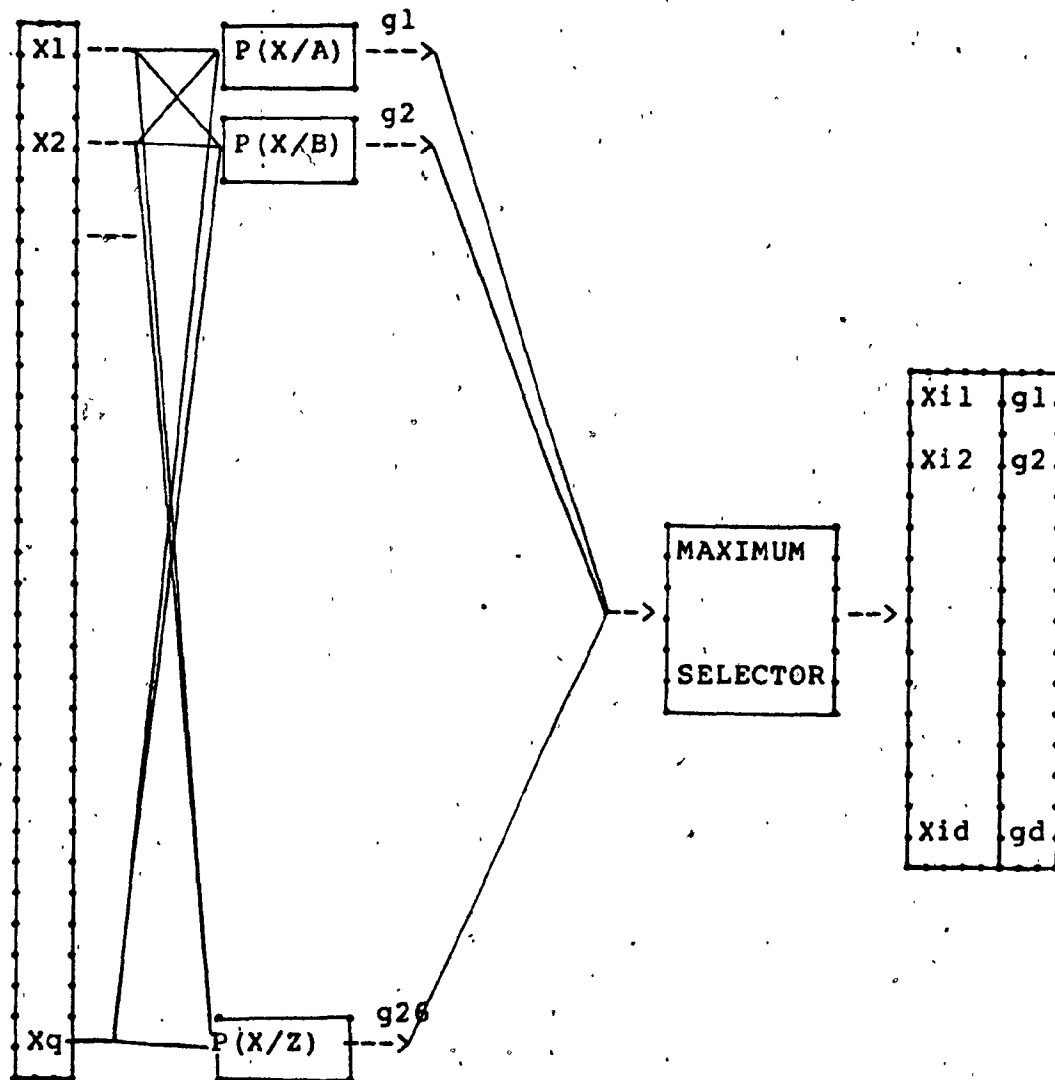


FIGURE 5.1-A: Design of Single Character Recognition Subsystem (SCRS).

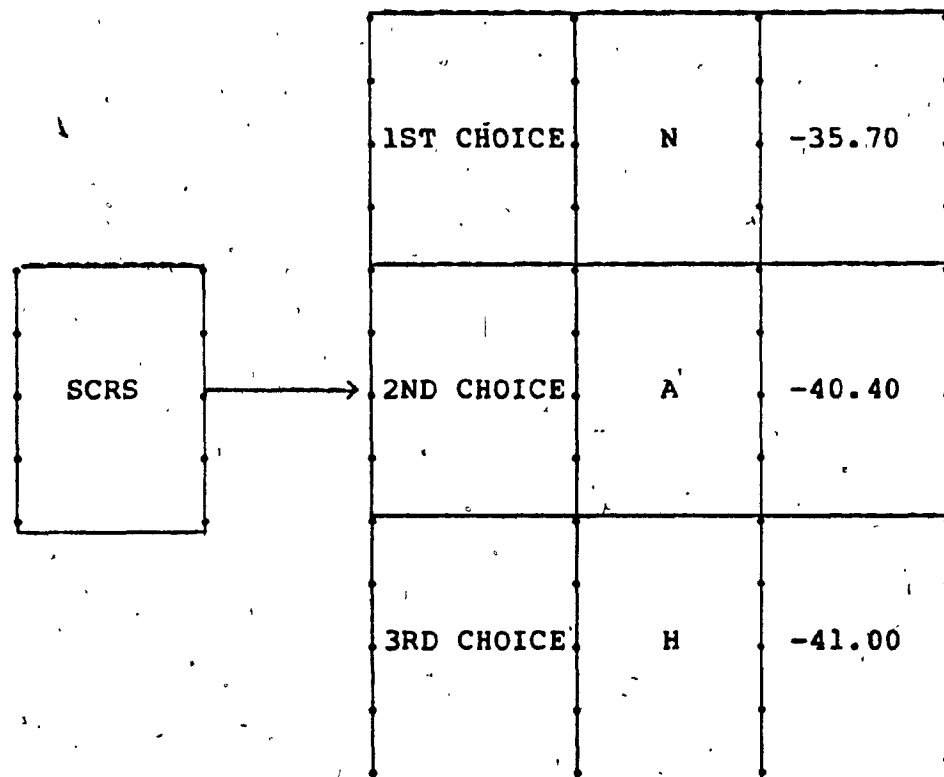


FIGURE 5.1-B: 3-Ordered choices for letter A; output from SCRS operating on testing set from Munson's data, i.e., subject 28.

the expression 3.2-5 and select 'd' ($1 \leq d \leq 26$) maximum alternatives for pattern X_i .

3. If the sequence X_1, \dots, X_n is to recognize, consider $d = 1$ and repeat steps 1 and 2 for every X_i . Maximizing expression 5.1-2, each X_i will be classified as Z_i . Concatenation of Z_i 's, for $i=1, \dots, n$ will yield the decided word. This decided word will be called as designated word in later references. These three steps can thus be implemented as below:

Comments: To accomplish steps 1 and 2, compute the elements of likelihood vector likli for the i -th pattern and find d -alternatives.

```
For k:=2 to 27 do
```

```
  Begin
```

```
    For j:=1 to q do
```

```
      likli (k-1) := likli (k-1) + log P ( $X_{ij}/L_k$ );
```

```
    End;
```

```
  if d = 26 exit;
```

```
For k:=2 to 27 do
```

```
  Begin
```

```
    For p:=1 to d do
```

```
      Begin
```

```
        aip := k-1; /* likli (k-1) is the largest
                      in the vector */
```

```
        likli (k-1) := -∞; /* very small quantity */
```

End;

End;

Comments: To accomplish step 3, assume that for every X_i its corresponding recognized character is available from steps 1 and 2. Using an operator CONCAT, construct the string 'S', by concatenating the string already formed and the current recognized letter $L_{a_{il}}$.

For example, if

$a_1 := 'HA'$

$a_2 := 'T'$

$a_3 := a_1 \text{ CONCAT } a_2$

then a_3 becomes 'HAT'

if $d = 1$ then

Begin

$S(L_{a_{1l}}) := L_{a_{1l}};$

For $i := 2$ to n do

$S(L_{a_{il}}) := S(L_{a_{(i-1)l}}) \text{ CONCAT } L_{a_{il}};$

End;

EXPERIMENT - 1

The value of 'd' should be 1 if one needs to study the recognition performance at this stage. Thus for $d = 1$, the input sequence X_0, \dots, X_{n+1} will be recognized as Z_0, \dots, Z_{n+1} , where Z_0 and Z_{n+1} are blank. For $d = 1$, an experiment was conducted to observe:

A. Overall character recognition on both training and testing

sets belonging to three data sets.

B. Text recognition on all three sets of passages.

TABLE 5.1-A and 5.1-B show the result of these experiments.

As seen from TABLE 5.1-A, the results on characters from OCR 'A' data show a 100% recognition. Therefore, further experiments were not conducted over it.

Training set of OCR 'B' data set obtained 100% character recognition rate while the testing set gave 95.478% correct. It should be noted that the training set was very small (18 samples of each of 'A' to 'Z'), while the testing set was fairly large (91 samples of each letter). Better recognition rates, both on OCR 'A' and OCR 'B' data, were obtained due to the fact that they were machine-printed single font characters.

Overall character recognition rates both on training and testing sets belonging to Munson's data are very low as compared to those obtained on OCR 'A' and OCR 'B'. The obvious reason for such a low performance was that the characters were unconstrained and untutored hand-printed characters. To study which subject's handwriting deteriorates the results, the same experiments were repeated on every subject of Munson's data. These results are shown in TABLE 5.1-B.

Among training set, subject 47 appeared to have good handwriting while subject 48 gave the lowest recognition

Table 5.1-A

Recognition Rates on 3 Data Sets Obtained by Using SCRS Only.

Data Set	Character Recognition Rate		Pas-1		Pas-2		Pas-3	
	training set	testing set	training set	testing set	training set	testing set	training set	testing set
Munson	88.034	70.513	87.239	71.686	88.200	69.442	87.207	73.631
OCR'A'	99.903	100.000	100.000	100.000	100.000	100.000	100.000	100.000
OCR'B'	100.000	95.478	100.000	94.500	100.000	95.791	100.000	93.978

T A B L E - 5.1-B

Recognition rates on Munson's data obtained
by using SCRS only
(No Context)

Subject 1	Character Recog. rate 2	Recognition Rate on Passages		
		Pas-1 3	Pas-2 4	Pas-3 5
1	89.359	91.073	91.217	89.964
2	88.462	87.376	84.172	86.040
3	96.154	98.738	97.713	98.540
4	84.615	86.384	86.459	87.226
5	96.154	93.327	96.157	92.974
6	94.872	91.794	94.511	91.606
7	93.590	89.720	92.589	89.325
8	85.897	86.114	88.015	87.318
9	85.897	84.310	78.042	83.303
10	89.744	90.622	89.844	89.325
11	88.462	79.351	92.315	80.474
12	80.769	76.826	87.100	78.193

col. 1: Serial number of the subjects.

col. 2: Average recognition rate on character set written by
corresponding subject (i.e., without passages being
constructed from them).

Columns 3, 4 and 5 list the average recognition rate without
using context on Pas-1, Pas-2, and Pas-3 respectively.

T A B L E - 5.1-B (CONTD)

Recognition rates on Munson's data obtained
by using SCRS only
(No Context)

Subject	Character Recog. rate	Recognition Rate on Passages		
		Pas-1	Pas-2	Pas-3
13	92.308	91.073	94.053	90.055
14	89.744	81.605	90.485	80.018
15	98.718	100.000	99.817	99.179
16	89.744	88.638	87.008	90.967
17	93.590	96.033	94.053	97.263
18	73.077	72.137	72.187	74.544
19	88.462	89.270	84.630	88.047
20	76.923	74.932	73.102	77.829
21	89.744	85.032	85.453	87.226
22	96.154	96.664	96.432	98.358
23	93.590	90.532	95.608	90.055
24	88.462	88.458	91.857	88.869
25	82.051	88.188	87.283	88.139
26	85.897	85.753	80.970	83.485
27	87.179	88.729	86.917	89.964
28*	70.513	71.686	69.442	73.631
29	85.897	89.540	92.040	89.872
30	89.744	91.434	88.198	89.872

* This subject is included in the testing set, he was not in training set.

T A B L E - 5.1-B (CONTD)

Recognition rates on Munson's data obtained
by using SCRS only
(No Context)

Subject	Character Recog. rate	Recognition Rate on Passages		
		Pas-1	Pas-2	Pas-3
31	93.590	94.500	92.772	92.609
32	92.308	89.179	90.942	89.416
33	91.026	95.582	96.523	95.894
34	89.744	91.524	95.059	91.058
35	82.051	83.769	85.087	83.217
36	82.051	85.573	89.296	86.131
37	83.333	79.892	71.455	79.288
38	92.308	92.155	92.498	91.788
39	85.897	84.941	87.283	85.128
40	80.769	77.097	77.127	75.091
41	85.897	75.834	85.544	75.639
42	94.872	92.876	92.315	91.332
43	91.026	93.147	94.785	94.069
44	92.308	92.155	88.838	90.785
45	88.462	84.040	83.532	83.303
46	91.026	94.319	93.138	95.894
47	79.487	74.121	77.676	74.635
48	71.795	67.899	76.121	68.887
49	83.333	85.212	85.361	83.759

rates.

These results obtained by SCRS vary significantly from data to data, i.e., they appeared to be data dependent. This kind of situation can not be borne in practice. Using this subsystem could we improve the recognition performance? Tonssaint [T04-77] described that possible answer is to include the use of context. Using context, results shown by Shinghal [SH4-79] reinforce this answer. In the present study, we also used the context. In chapters 6 and 7 postprocessing system which is based on contextual information is introduced.

Computational Complexity

The computational complexity of SCRS is a function of n and d . The selection of d largest elements among 26 needs:

$$\sum_{j=1}^{\min(26-d, d)} (26-j) \quad \dots 5.2-1$$

comparisons. For a word of n letters, this quantity will be repeated n times, i.e.,

$$n \sum_{j=1}^{\min(26-d, d)} (26-j) \quad \dots 5.2-2$$

If a word of n -letters is to be recognized at this stage, the number of comparisons will be $25n$ (value of $d = 1$).

5.3 CONFUSION MATRICIES

From the experiment-1-A above, we gathered the information about every input character and its corresponding recognized output. In pattern recognition terminology, we recorded the frequency distribution of any character to be confused with any of the possible character classes. The resulting two dimensional matrix was called the confusion matrix. As shown in TABLE 5.2, the rows, from top to bottom, are labelled 'A' to 'Z', designate the input to the SCRS. The columns, from left to right, are also labelled as 'A' to 'Z' designate the possible corresponding output. The elements of the matrix are the frequency count of a given input versus the SCRS output for $d = 1$. The diagonal (top left to bottom right) gives the frequency distribution of correct recognition. TABLE 5.2 shows the confusion matrix constructed from training set of Munson's data.

Similarly the Confusion Matrix was constructed from the testing set of OCR 'B' data. It is usual practice to construct the confusion matrix from the training set, but since the training set was small and gave 100% character recognition, we used the testing set to construct the confusion matrix. The confusion matrix for OCR 'B' data is shown in TABLE 5.3.

Thus SCRS classifies the input patterns primarily. To aid its performance the third stage i.e., CPPS is added. The

5.2

MUNSON'S DATA

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
A	125	0	0	0	0	0	1	5	0	1	3	1	3	0	0	0	1	1	0	0	0	0	3	0	0		
B	1	129	0	1	0	0	1	0	1	0	1	0	1	0	1	4	0	2	0	0	0	0	0	0	3		
C	0	0	133	0	0	0	0	0	0	0	0	7	0	0	4	0	0	0	0	0	0	0	0	0	0		
D	0	0	1	126	1	0	0	0	3	0	0	3	0	0	4	4	0	1	0	1	0	0	0	0	0		
E	0	0	2	1	124	4	1	0	2	0	0	5	0	0	0	0	0	1	1	0	0	0	0	0	3		
F	1	0	0	0	0	141	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0		
G	1	1	5	0	0	0	124	1	0	0	0	1	0	0	1	0	8	0	0	0	2	0	0	0	0		
H	6	1	0	0	0	0	0	114	0	0	0	1	15	0	0	0	0	1	0	0	0	0	0	1	5	0	
I	0	0	0	0	2	0	0	0	132	6	0	1	0	0	0	8	0	0	0	3	0	0	0	0	0	0	
J	1	0	0	0	0	0	0	0	6	115	0	1	0	0	0	0	0	0	0	12	8	0	0	0	1	0	
K	4	0	0	0	0	1	0	2	0	0	127	1	1	0	0	0	0	1	0	0	0	2	1	2	2	0	
L	1	2	1	0	1	0	0	0	2	1	1	132	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0
M	7	0	0	0	0	1	0	3	0	0	0	0	124	2	0	1	0	2	0	0	0	2	0	1	1	0	
N	4	0	0	1	0	0	1	8	0	0	1	0	5	114	0	0	0	0	0	0	5	3	0	2	0	0	
O	0	0	3	4	0	0	1	2	0	0	0	0	0	0	133	0	0	0	0	0	1	0	0	0	0	0	
P	2	0	0	1	0	3	0	0	0	0	0	0	0	0	0	137	0	0	0	1	0	0	0	0	0	0	
Q	1	0	1	2	0	0	6	0	0	0	0	0	0	1	5	0	124	1	2	0	0	0	0	0	1	0	
R	7	1	0	0	1	1	0	2	0	0	7	1	1	1	0	2	2	114	0	0	0	0	1	1	0	2	
S	0	1	0	1	1	1	1	0	3	6	1	0	0	0	0	0	0	0	129	0	0	0	0	0	0	0	
T	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	1	0	0	0	135	0	0	0	0	0	0	
U	0	0	0	0	0	1	3	0	0	2	0	3	5	2	0	1	1	0	0	118	5	1	2	0	0	0	
V	0	0	0	0	0	1	0	0	0	1	2	0	0	4	0	0	0	0	0	3	129	1	0	3	0	0	
W	0	0	0	0	0	0	1	0	0	1	0	0	1	9	0	0	0	0	0	3	8	121	1	0	0	0	
X	5	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	2	0	0	0	2	0	125	6	0	
Y	0	0	0	0	0	1	0	1	0	1	2	0	0	0	0	0	0	0	0	1	0	4	0	1	133	0	
Z	0	1	0	0	3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	138	

TABLE - 5.3

		CONFUSION MATRIX FOR TESTING SET OF O C R - B DATA																									
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	90	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
B	0	76	0	1	5	0	0	2	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	1	0	0	0
C	0	0	88	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0
D	0	0	0	0	91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	75	0	0	0	0	0	0	0	13	0	0	0	1	0	2	0	0	0	0	0	0	0	0
F	0	0	0	0	0	90	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
G	0	0	3	0	6	0	78	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0	91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	87	4	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	2	0	0	0	0	0	0	89	0	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	83	0	0	0	0	0	2	0	0	0	0	0
P	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	82	0	2	0	0	0	0	0	0	0	0
Q	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	4	0	84	0	0	0	0	0	0	0	0	0
R	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	84	0	0	0	0	0	0	0	0
S	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	89	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	91	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	90	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87	0	1	3	0	
W	0	0	0	0	0	0	0	5	0	2	0	0	0	0	0	0	0	0	0	0	0	2	0	82	0	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	91	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	91	0	0
Z	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	89

results obtained by SCRS will be used by CPPS, it will be shown in the following chapters.

CHAPTER 6

CONTEXTUAL POSTPROCESSOR MODIFIED VITERBI ALGORITHM

6.1 PREVIEW OF THE CHAPTER

In chapter 5 we recognized the text without using context i.e., using likelihoods only. This chapter and chapter 7 investigate how the context can help in improving the recognition performance. Thus the third subsystem CPPS used contextual information for deciding the identity of the input pattern. Section 6.2 describes this subsystem. Several algorithms used as CPPS and the experiments thereon are then described. The three passages described in chapter 4 were used as input text in these experiments. Section 6.3 describes modified Viterbi algorithm. The experiments conducted, using this algorithm are stated in section 6.4.

6.2 CONTEXTUAL POSTPROCESSOR (CPPS)

Contextual postprocessing subsystem (CPPS) is the third stage of the proposed system. It is context aided text recognizer. Performance of several contextual algorithms was tested at this stage. In all these algorithms we considered English language to assume the properties of Markov process of order one.

When $d = 1$ CPPS does not take part in classification. This is the situation where context was not used in classification and the text was recognized by the SCRS only.

The algorithms investigated could be divided in two categories.

1. The one based on Markov approaches [CHU-75, RAV-67, SH2-78, SH4-79].
2. The one based on Dictionary approaches [ALB-67, BLE-59].

Markov approaches are far less expensive than Dictionary approaches, but the character recognition performance was poorer than dictionary approach [SH4-79]. To have a compromise between the cost and the recognition performance a combination of two methods, which can also be considered as third category and called as hybrid method, was investigated as well.

The computational complexity and the recognition performances of the CPPS are depend upon the algorithm used. So the complexity of the algorithms used, will be investigated wherever we discussed them. The output from this subsystem is the text recognized by the proposed system using contextual postprocessing.

The description of CPPS thus completes the design of the proposed text recognition system. FIGURE 6.1 shows this design.

LEGEND

- Main stream
 - - - System's training
 ···· Aid to CPPS

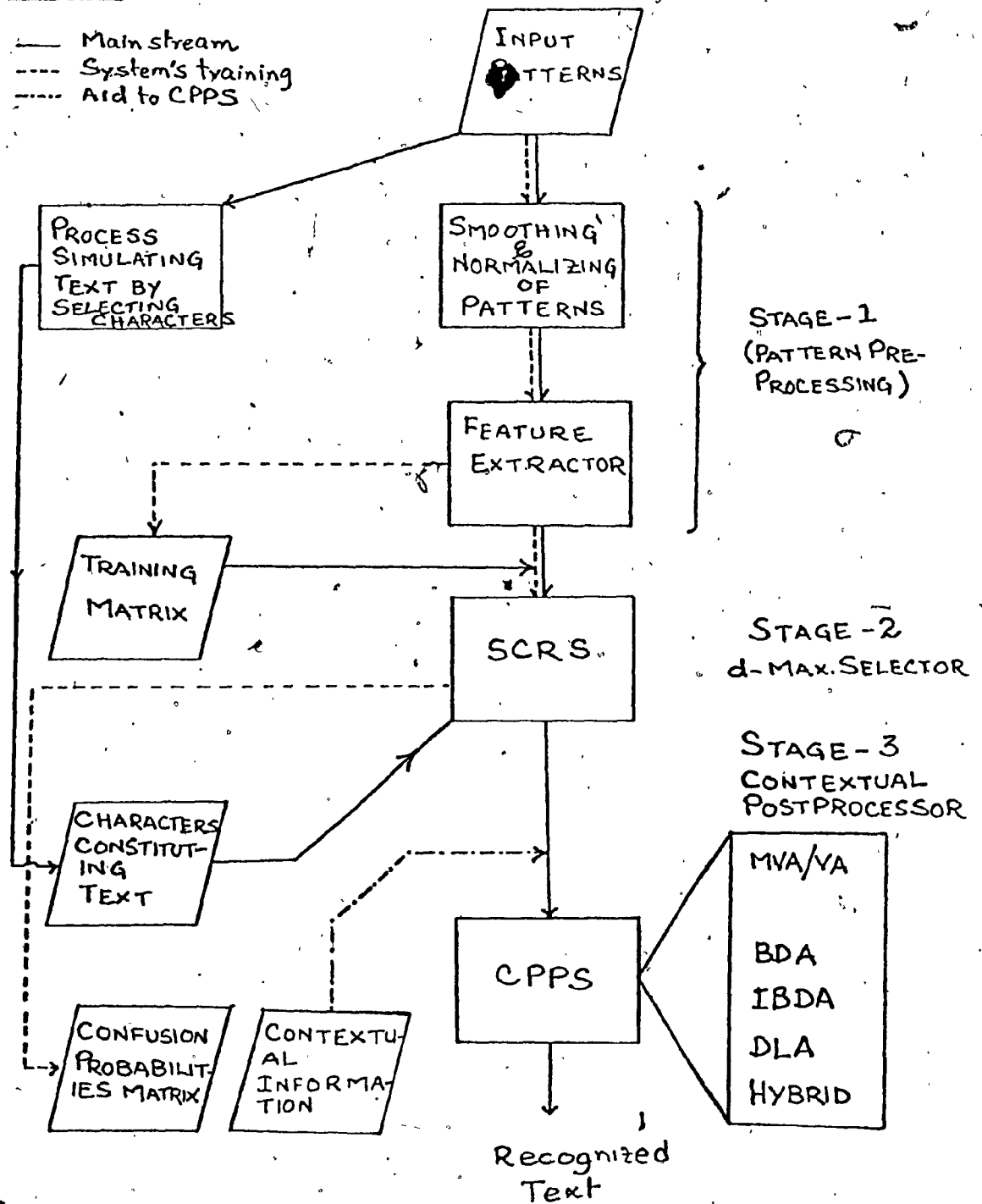


FIGURE 6.1: Context Aided Text Recognition System (detailed design).

Now the question arises based on the mathematical model developed in section 1.3; what kind of contextual information do we need?

Referring back to expression 1.3-4, in order to maximize it, the values of $P(X_i/Z_i)$ and $P(Z_0, Z_1, \dots, Z_{n+1})$ have to be estimated before hand. The probabilities $P(X_i/Z_i)$ are already estimated in section 3.2.4. Section 6.2.1 describes how the probabilities $P(Z_0, \dots, Z_{n+1})$ were estimated.

6.2.1 Estimation of Transition Probabilities.

$P(Z_0, \dots, Z_{n+1})$ is a priori probability of occurrence of the character sequence $Z_0, Z_1, \dots, Z_n, Z_{n+1}$. We already assumed that blank is the only delimiter used in separating the words. Further assumption was that the blanks are always perfectly recognizable; thus so would Z_0 and Z_{n+1} are always blank or more precisely:

$$P(Z_0/X_0) = P(Z_{n+1}/X_{n+1}) = 1$$

and

$$P(Z_i/X_i) = 0 \text{ for } 1 \leq i \leq n$$

...6.2-1

The character sequence thus reduced to the size of a word and each character in the word can assume any name of the letters 'A' to 'Z'.

$P(Z_1, \dots, Z_n)$ becomes the a priori probability of occurrence of the word Z_1, Z_2, \dots, Z_n in English language. One method of estimating $P(Z_1, \dots, Z_n)$ is to assume English

language to be a Markov source. If the text is considered to be a Markov source of order one then transition probabilities of type $P(C_i/C_j)$ can be used. This requires the probability distributions characterizing transition probabilities. These transition probability distributions can be estimated from the probability distributions of N-grams. As shown by Suen [SUE-79] the N-gram probability distributions have been estimated by many researchers.

The estimates of unigram and bigram probabilities as given in TABLE 4.1 and TABLE 4.2 were used. From these, the transition probability estimates were computed as follows:

$$P(C_i/C_j) = P(C_i, C_j) / P(C_j) \quad \dots 6.2-2$$

Transition probability $P(C_i/C_j)$ is the probability of occurrence of character C_i immediately after the occurrence of character C_j in English text. These estimates, thus computed, are shown in TABLE 6.1.

6.3 Modified Viterbi Algorithm (MVA)

The MVA provides an efficient method of finding the most likely sequence in maximizing a posteriori probabilities and it makes the decision on one word at a time.

Toussaint and Chung [TO2-74] proposed some modifications in Viterbi algorithm and called their method as Modified Viterbi Algorithm (MVA). The Viterbi Algorithm (VA) [VIT-67] was proposed as method of decoding convolutional codes.

TABLE - 6.1
TRANSITION BAYESIAN PROBABILITIES * (10 ** 7)

	A	B	C	D	E	F	G	H	I	J	K	L	M
	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1147969	459909	489530	295560	259470	390379	158190	439860	792830	51390	46920	225199	397860
	225790	786139	411330	21229	260259	629789	1668289	115680	80029	574310	1109	68320	2669
A	715430	2160	231649	422819	347720	17959	70240	183689	22100	344839	13120	108710	1101519
	1914959	6959	185819	2360	1056089	907160	1438949	108650	220799	74130	10990	240780	8330
B	104589	664069	67000	14590	23959	3262570	11699	7290	16909	560849	168460	6969	1236139
	17469	941369	6969	6969	546180	241789	110839	1013660	46879	8820	6969	879020	6969
C	383009	1239979	3980	191109	5429	1664620	5910	3490	1362189	666080	3379	228239	328299
	4830	1906460	5800	16320	336559	38799	1146090	335920	3379	3230	3379	105109	4089
D	5483609	254600	6749	4940	78509	1541829	10029	74490	33279	965430	7920	3149	63509
	23439	463089	5390	4089	4089	153449	268009	5070	325509	45429	17010	2819	115069
E	3468049	459229	13760	289899	710570	232639	108690	66430	19600	122720	4290	11929	331079
	905939	905939	43650	101820	32100	1256130	861279	235229	26889	172579	66749	130190	99960
F	4183620	543759	4609	4460	4270	730539	497749	4460	4609	941109	4460	4609	199390
	6519	1555799	4760	4270	642740	19469	239120	344180	4270	4270	4270	32719	4270
G	3216049	671249	6350	5870	18429	1519040	11410	144540	1065239	689280	6139	9380	214410
	304339	613669	6139	5870	765769	253299	47190	270449	5870	5870	7899	5870	71360
H	1033520	1517929	5210	3230	5280	4761849	4690	3589	2710	1204589	1959	2349	19380
	33780	789620	2640	2120	120359	25479	215919	99870	2050	7400	1959	103480	26810
I	168199	264419	100629	747260	308000	386189	249920	265560	3639	12359	49049	475499	341949
	247720	826150	75260	10959	334239	1287050	1212610	11690	300049	1689	18810	2089	66180
J	114699	1257330	54829	59240	54829	2270750	54829	54829	57349	81930	54829	54829	54829
	61130	2079789	63020	54829	54829	57349	54829	54829	3008760	54829	54829	54829	54829
K	2570949	205079	31980	26169	35560	3086210	33989	50099	44279	1558310	19460	127920	28180
	946210	74019	25270	19460	127920	663089	29070	77160	19460	37790	19460	95270	19460
L	1786430	896379	10700	14799	505759	1543190	134799	12640	4959	1211609	2599	1229300	43360
	10309	742400	30420	2719	25369	282449	212609	230900	81189	29249	2809	917870	3680

TABLE - 6.1 (CONTD.)

	A	B	C	D	E	F	G	H	I	J	K	L	M
	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
M	1382929	1704920	213309	13330	5170	2425269	14590	6420	8020	991290	4200	5310	19469
	30960	1060940	787830	4539	76370	303250	5800	433140	4539	4830	4200	209730	274649
N	2593270	381029	5709	501650	1458269	744470	71819	991390	12689	372369	13900	65500	29440
	86780	575250	7869	10630	14860	606700	1113859	65379	60830	7580	3870	114750	3149
O	1189629	46879	106890	147410	222069	41430	1241360	92169	19809	88309	7719	58660	545139
	1776989	210249	236239	2059	1323610	321769	471639	932050	174210	349569	12240	44200	8219
P	467040	1062110	8629	6539	5619	1837910	9030	5269	462519	444800	5040	6780	66479
	45570	1364799	495529	5040	1697139	231639	372820	365560	5269	8799	5040	45000	5040
Q	117630	76940	76940	76940	76940	76940	76940	76940	76940	76940	76940	76940	76940
	76940	76940	76940	76940	76940	76940	76940	76940	76940	76940	76940	76940	76940
R	1958890	805980	29659	136669	253259	2450000	42210	113190	26859	884429	5520	111370	235549
	189300	820810	5795	1969	138550	517650	468340	190679	86319	16250	5259	338140	2199
S	4201870	277140	6549	184159	11979	1133589	22380	3540	356220	713699	1580	49849	96269
	24679	466450	224969	9200	6010	483979	1205250	364899	2050	25749	1580	73169	1889
T	2092780	416909	3440	18190	2189	929639	5929	1970	3209219	1085389	1220	1419	86930
	9469	872150	3230	1180	340130	296370	146899	205570	1510	65659	1129	177180	3739
U	287539	355190	250700	542020	299180	432889	45270	387129	5619	273489	4639	17389	1050550
	1120529	31179	407150	4329	1454269	1409979	1205889	6199	8829	3879	10700	13779	9410
V	34560	931139	9639	12959	10080	6238109	9639	9639	9639	1967280	9639	9639	9639
	9639	554449	9639	9639	9639	13960	12959	10410	23699	9639	9639	41090	10080
W	1004589	1635770	10500	7600	15960	1691549	16590	6010	2005580	1701709	6010	16590	88119
	352879	1043709	6910	6010	138079	150939	24470	8979	6290	6290	6010	21359	6010
X	893939	919579	46489	895540	46489	617149	72669	46489	142270	1305899	46489	46489	46489
	48620	133579	2550899	58779	51829	51829	51829	1483299	131979	53429	50229	51829	95649
Y	7512599	70540	20840	95009	18540	419020	8160	13589	7460	136059	6059	9060	120460
	41960	630299	149250	6059	31510	482590	108679	10389	7460	19030	6059	6339	11989
Z	605329	1689639	111419	107970	104529	3767519	104529	99929	137840	769579	104529	201010	104529
	99929	350330	99929	99929	99929	99929	114859	119460	201010	119460	122900	152769	290599

Forney [FOR-73] extended it as a contextual text recognition algorithm. The text recognition capability of this algorithm was further investigated by Neuhoff [NEU-75]. He concluded that the algorithm provides computationally simple and fairly accurate character recognition.

Toussaint and Chung [TO2-74] considered Viterbi algorithm as a special case of modified Viterbi algorithm. Chung [CHU-75] investigated it and concluded that instead of including all possible pattern classes for input patterns in the examination, one could concentrate on only a few most probable pattern classes without resulting in any deterioration in the results. The modified Viterbi algorithm thus leads to a considerable saving in computations as compared to the Viterbi Algorithm.

In the present research, MVA has been extensively used. An exhaustive description of the algorithm and its implementation procedure are included in the following section.

6.3.1 Description of the ALGORITHM

So far X_i was considered as feature vector of a pattern. This vector belongs to some pattern and has to be classified into one of the possible classes anyway. Therefore, in later references the words feature-vector and pattern will be used synonymously.

Consider English text which consists of a character set comprising the symbol blank ' \emptyset ' and upper case letters 'A' to 'Z'. Let Z_k be the k -th character in the output text, where Z_k can assume any value in the character set.

To classify the sequence X_0, X_1, \dots, X_{n+1} the decision rule described in section 1.3 requires to maximize a posteriori probability $P(Z_0, \dots, Z_{n+1} / X_0, \dots, X_{n+1})$.

In general, one would have to compute and compare $P(Z_0, \dots, Z_{n+1} / X_0, \dots, X_{n+1})$ for all possible sequences Z_0, \dots, Z_{n+1} . If English language is assumed to be a Markov source, we need only to maximize expression 1.3-4.

Following the description given by Shinghal [SH4-79] the MVA is based on four assumptions given below:

1. English language is a Markov source of order r , for $r \geq 1$.

Thus

$$P(Z_k / Z_1, \dots, Z_{k-1}) = P(Z_k / Z_{k-r}, Z_{k-r+1}, \dots, Z_{k-1})$$

2. The patterns X_0 and X_{n+1} are named blank; all other X_i , for $1 \leq i \leq n$, are named with the letters.
3. Words in the text are properly delineated, i.e., spaces are always classified correctly, thus

$$P(\emptyset / X_i) = 1 \text{ for } i = 0, n+1$$

and

$$P(\emptyset / X_i) = 0 \text{ for } 1 \leq i \leq n$$

4. The process of character recognition is memoryless, i.e.,

$$P(X_i/Z_0, Z_1, \dots, Z_{n+1}, X_0, \dots, X_{n+1}) = P(X_i/Z_i)$$

Thus we will use these assumptions to equation 1.3-4 to obtain the maximizing function for the experiments. From equation 1.3-4

$$g(X) = \sum_{i=0}^{n+1} \log P(X_i/Z_i) + \log P(Z_0, \dots, Z_{n+1}) \quad \dots 6.3-1$$

from assumption 3

$$g(X) = \sum_{i=1}^n \log P(X_i/Z_i) + \log P(Z_0, \dots, Z_{n+1}) \quad \dots 6.3-2$$

In assumption 1 if $r = 1$ then

$$P(Z_k/Z_1, \dots, Z_{k-1}) = P(Z_k/Z_{k-1}) \quad \dots 6.3-3$$

from assumption 4

$$\log P(Z_0, \dots, Z_{n+1}) = \sum_{i=1}^{n+1} \log P(Z_i/Z_{i-1}) \quad \dots 6.3-4$$

from equation 6.3-4 and 6.3-2 we get

$$g(X) = \sum_{i=1}^n \log P(X_i/Z_i) + \sum_{i=1}^{n+1} \log P(Z_i/Z_{i-1}) \quad \dots 6.3-5$$

we called 6.3-5 as the discriminant function which is used to maximize a posteriori probabilities.

The description is further illustrated in FIGURE 6.2, where a 3-letter word HAT with depth of search $d = 4$ is assumed. The SCRS offered 'd' possible alternatives for each letter to CPPS. The information relevant to making a decision on the word can be expressed as a directed graph. Except the

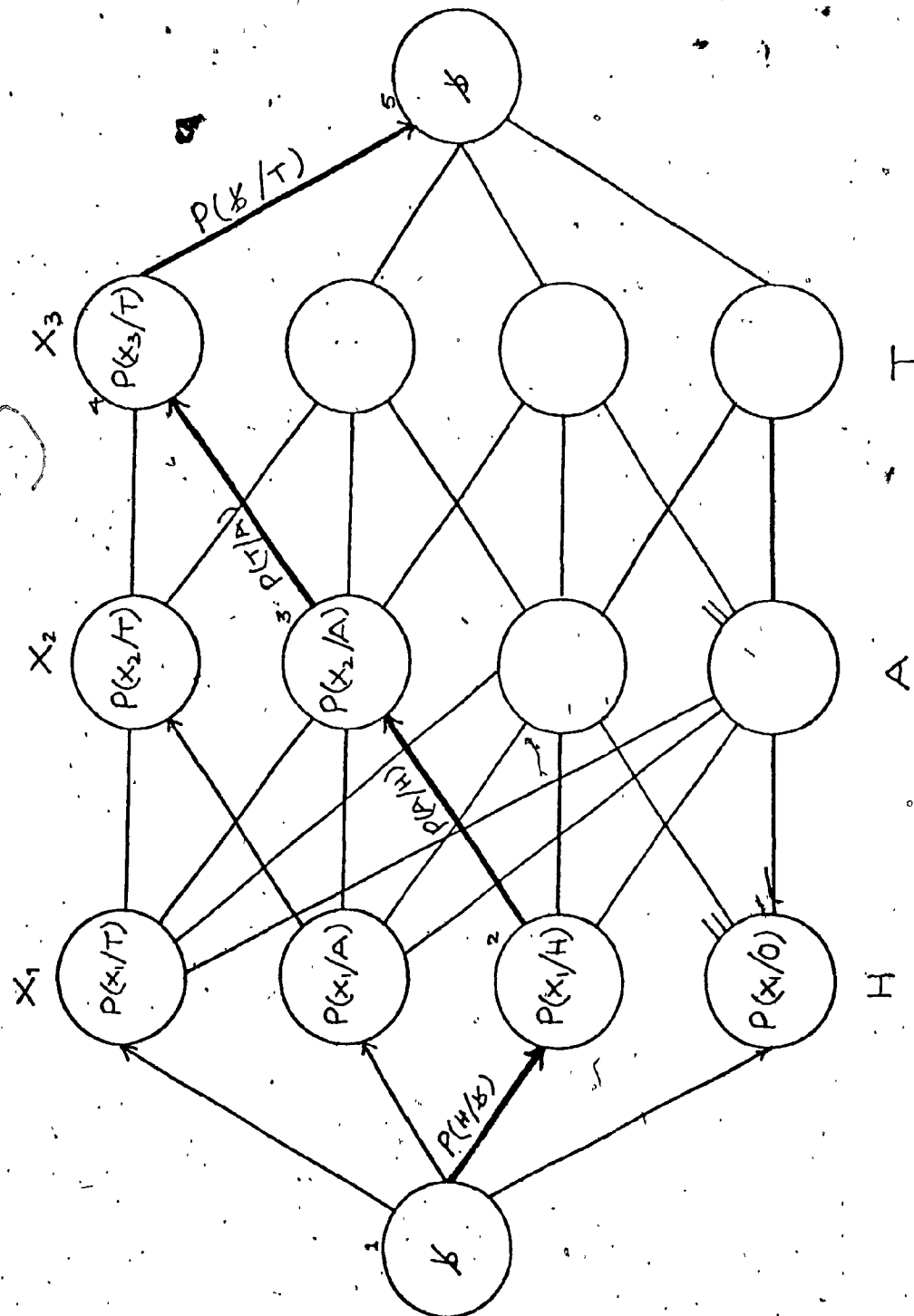


FIGURE 6.2: Illustration of Modified Viterbi Algorithm.

'start' and 'end' nodes which are identified by the blank character 'Ø', all nodes and edges have some probabilities associated with them. The edge probabilities i.e., $P(Z_i/Z_{i-1})$ are the Markov transition probabilities and represent static information (see section 6.2). The node probabilities which represent dynamic information are the conditional probabilities $P(X_i/Z_i)$ (also called likelihoods). From the figure it is clear that any path from the 'start' node to the 'end' node represents a sequence of letters but not necessarily a valid English word. Consider the darkened path in the same figure. Going through nodes 1, 2, 3, 4, and 5, if the sum of all the edge and node probabilities on the path be taken, we obtain:

$$g(X) = \log P(X_1/H) + \log P(X_2/A) + \log P(X_3/T) + \\ [\log P(H/'Ø') + \log P(A/H) + \log P(T/A) + \log P('Ø'/T)]$$

where the part of the equation enclosed in square brackets is the contextual information.

In above, if the value of $g(X)$ turned out to be the maximum, then the designated word given out by the algorithm will be the word 'HAT'.

If instead of 'd', all 26 letters of English language are considered, the algorithm is reduced to Viterbi algorithm [TQ2-75]. In this case, at each stage in the sequence, 26 most likely sequences are computed. At the final stage the most likely sequence is chosen.

Thus from the set of the character alternatives, a set of alternative words are formed. Out of these words the algorithm selects the most probable one.

6.3.2 Implementation of the Algorithm

Excluding blank, all the possible pattern classes could be considered as possible alternatives for the input pattern. This consideration leads to the separation of implementation process into two main steps: invoking SCRS to get 'd', $1 \leq d \leq 26$ maximum alternatives for each input pattern in the sequence and then maximizing $g(X)$.

When 'd' ($1 \leq d \leq 26$) alternatives for every pattern X_i have been found by SCRS, the 'd' most probable letter-sequences ending with 'd' alternatives of X_i , for $1 \leq i \leq n$, are found by computing $g(X)$. Of these 'd' letter-sequences the one which most probable in terminating with X_{n+1} — a blank is the word decided for the input sequence $X_0, X_1, \dots, X_n, X_{n+1}$. This word decided is the designated word. The algorithm can informally be explained as below:

Steps

1. Ignoring blanks get X_1, \dots, X_n , one letter-sequence (word), let its length (i.e., number of letters in the word) be n .
2. Invoking SCRS get 'd' maximum likelihoods and their corresponding letter alternatives for every letter in the

sequence.

3. Compute the discriminant function for each alternative of first position letter (i.e., X_1) given that preceding character was blank (i.e., X_0).
4. If $n = 1$ go to step 6.
5. Loop this step for i equals 2 to n , compute the discriminant and kept selecting the most probable sequences from position 2 to position i for sequence X_2, \dots, X_i using their 'd' alternatives and the discriminant computed a position before. For position 2 the position before is step 4.
6. Compute the discriminant that these 'd' alternatives follow character (X_{n+1}) was blank and output the one giving maximum discriminant as the designated word and output this word.

In the above algorithm step 1 is straightforward; step 2 is already explained in section 5.2. Steps 3 through 6 are formalized as given by Shinghal [SH4-77]:

Step [A1-1]

Comments: This step finds the maximum probability that can be constructed from the 'd' alternatives for each letter position found by SCRS in step 2. It proceeds position by position through the sequence, finding at each stage the most probable string upto and including the current position.

```

/* Initialize using the 'd' alternatives for feature vector  $X_i$ 
*/

```

Loop-1

```

  For j:=1 to d do

```

```

    Begin

```

```

       $M_1(L_{a_{1j}}) := \log P(X_1/L_{a_{1j}}) + \log P(L_{1j}/\emptyset);$ 

```

```

       $S_1(L_{a_{1j}}) := L_{a_{1j}}$ 

```

```

    End; /* Loop-1 */

```

```

/* skip to end-of-word processing if n=1 */

```

```

  If n=1 then go to step [A1-2];

```

```

/* select the 'd' most likely letter sequences ending in the
'd' alternatives for each feature vector  $X_2, X_3, \dots, X_n$ .

```

```

 $M_i$  saves the values of  $g(X_0, X_1, \dots, X_i)$  and  $S_i$  saves the
alternative letter-strings. */

```

```

/* In the following loops, the subscripts:

```

```

    i runs over sequence positions

```

```

    j runs over alternatives for position (i-1)

```

```

    k runs over alternatives for position i */

```

Loop-2

```

  For i:=2 to n do

```

```

    Begin

```

```

      For j:=1 to d do

```

Begin .

for k:=1 to d do

$g_k := M_{i-1}(L_{a_{i-1,k}}) + \log P(X_i/L_{a_{ij}}) +$
 $\log P(L_{a_{ij}}/L_{a_{i-1,k}});$

$M_i(L_{a_{ij}}) := \text{MAXPOS}(g_1, g_2, \dots, g_d, u);$

$S_i(L_{a_{ij}}) := S_{i-1}(L_{a_{i-1,u}}) \text{ CONCAT } L_{a_{ij}};$

End;

End; /* Loop-2 */

Step [A1-2]

/* Select the most likely letter-sequence at X_{n+1} (blank) as the designated word */

For k:= 1 to d do

$g_k := M_n(L_{a_{nk}}) + \log P(' \emptyset' / L_{a_{nk}});$

$M_{n+1}(' \emptyset') := \text{MAXPOS}(g_1, g_2, \dots, g_d, u);$

Stop as the designated word is in $S_n(L_{nu})$.

Following is the explanation of the symbols used in [A1-1] and [A1-2].

- L_1, L_2, \dots, L_{26} are the 26 English letters; i.e.,

$L_1=A, L_2=B, \dots, L_{26}=Z.$

- MAXPOS is a function such that if

$c := \text{MAXPOS}(g_1, g_2, \dots, g_d, u)$

is written, c is given the value of largest g_k , $1 \leq k \leq d$, and u is given the value of the subscript k . For example if g_2 is the largest among the g_k , $1 \leq k \leq d$, then c becomes g_2 and u becomes 2.

- CONCAT is an operator to concatenate letter sequence, and it is explained in section 5.2.

In the procedure above, when $d = 26$, the modified Viterbi algorithm becomes the Viterbi algorithm. In such a case the SCRS does not order the letter alternatives, as all the 26 alternatives are being considered anyway. The other special case is when $d = 1$. In that case, it is like classifying without contextual information i.e., classifying using SCRS only. This special case is already explained in section 5.2.

6.4 EXPERIMENTS WITH MVA/VA (EXPERIMENT - 2)

Using MVA as contextual postprocessor (CPPS), a text-recognition experiment was conducted using the set of three passages described in section 4.4.

MVA used the expression 6.3-5 which assumed English language to be a Markov source of order one. The same experiment was repeated for different values of 'd' in order to find the maximum number of letter alternatives needed for

TABLE - 6.2-A

Recognition rates obtained by using Viterbi Algorithm
(MVA for $d = 26$)

Passage	Subjects from Munson's Data			OCR 'B'
	6	18	28	
Pas-1	96.664	78.539	87.376	95.762
Pas-2	97.347	76.578	83.074	96.249
Pas-3	96.077	80.383	88.230	95.073

the best recognition performance.

The recognition results obtained on the passages for $d = 26$, i.e., the case where MVA reduces to Viterbi algorithm, are shown separately in TABLE 6.2-A. Comparing these results with those given in TABLE 5.1-A, it was observed that significant improvements in recognition rates are obtained with using context over not using it. It would further be clearer from TABLE 6.2-B which records the percentage of improvement with respect to the results obtained by SCRS.

The results of TABLE 6.2-B, thus show that contextual information helped in improving the character recognition rates to as maximum as 21 percent. Results for Pas-1 and Pas-3 are better than for Pas-2. The obvious reason is that Pas-2 does not represent a common English usage. Further, the improvement is more obvious if the recognition rate without context (at SCRS stage) are low. In such cases since the letters were badly written and their decision regions (likelihoods) were overlapping each other, the context helped in distinguishing the regions clearly and hence better recognition was observed. While in cases, where the letters were nicely written, which implies that the decision regions made from them were already distinguished enough, the context caused little improvement.

Compared to subject 18; for subject 28, the increase in recognition rate is very significant. This owes to the

T A B L E - 6.2-B

Percentage gain in recognition rates by using
VA over not using Context

Passage	Subjects from Munson's Data			OCR 'B'
	6	18	28	
Pas-1	5.305	8.875	21.887	1.335
Pas-2	3.001	6.083	19.631	0.478
Pas-3	4.881	7.833	19.827	1.165

TABLE - 6.3

Recognition rates obtained by using
Modified Viterbi Algorithm

		Subjects from Munson's data OCR 'B'			
		6	18	28	
Passage	Depth				
Pas-1	2	95.401	76.014	87.015	94.299
	3	95.401	71.547	87.286	95.762
	4	96.664	77.547	87.376	95.762
	5	96.664	78.629	87.376	95.762
	6	96.664	78.539	87.376	95.762
Pas-2	2	96.981	75.480	82.708	95.608
	3	96.981	76.578	83.440	96.249
	4	97.530	76.395	83.074	96.249
	5	97.347	76.670	83.074	96.249
	6	97.347	76.670	83.074	96.249
Pas-3	2	95.620	77.464	88.047	93.522
	3	95.620	79.745	88.321	95.073
	4	96.077	79.197	88.230	95.073
	5	96.077	80.474	88.230	95.073
	6	96.077	80.383	88.230	95.073

correct recognition of frequently used letters (e.g., R, S, and T) at SCRS-stage. If the most frequently used letters are correctly recognized, the context will help in recognizing letter-combinations (bigrams) made with them.

TABLE 6.3 shows the results obtained from MVA for $2 \leq d \leq 6$ on the same passages. As seen from the table the optimal value of the parameter 'd' is 4. If the rates obtained from SCRS were better (over 90%), the rates given by CPPS using MVA get stabilized before 'd' reached the value of 4. As 'd' increases 4 we did not observe any significant improvement in the results.

Computational Complexity of MVA

Using the method given by Shinghal [SH4-79] the computational complexity of the algorithm is:

$$(3d - 1) (1 + (n - 1)d) \quad \dots 6.4-1$$

where $d(3d-1)$ computations are needed for d-alternatives of 2 letters; 'd' additions to evaluate the probability of a letter being followed by a blank; and $(2d-1)$ computations to evaluate that a blank follows 'd' n-letter sequences. The amount of computations required in invoking SCRS is already given by equation 5.2-2.

Thus for three possible cases of the algorithm, the amount of computations will be:

for $d = 1$

as given, by equation 5.2-2, this amount will be $25n$, since the dynamic programming is not done.

for $2 \leq d \leq 25$

the amount of computations will be

$$n \sum_{j=1}^{\text{Min}(26-d, d)} (26-j) + (3d-1)(1+(n-1)d) \quad \dots 6.4-2$$

for $d = 26$

the SCRS invoked, without letting it do any ordering, therefore, the value of equation 5.2-2 will be zero.

Substituting $d = 26$ in equation 6.4-1 we get

$$77 \times (26n - 25) \quad \dots 6.4-3$$

The use of contextual information is further studied in chapter 7.

CHAPTER 7

CONTEXTUAL POSTPROCESSOR

BLOCK DECODING AND HYBRID METHOD

7.1 PREVIEW OF THE CHAPTER

This chapter further investigates the contextual algorithms. Block decoding algorithm investigated by earlier researchers [DUD-68, T01-72] is described in section 7.2. Using this method the same text recognition experiments were repeated. Introducing several improvements in this method an algorithm known as Imbricate Block Decoding algorithm (IBDA) is proposed which is described in the same section. Dictionary method and the experiments using this are described in section 7.3. Section 7.4 describes a hybrid method which is a combination of dictionary method and MVA. This section further describes the experiments conducted using this method. Section 7.5, in the last, concludes the performance of different contextual algorithms discussed in this thesis.

7.2 BLOCK DECODING ALGORITHMS

Duda and Hart [DUD-68] introduced the concept and made a decision on a short sequence of characters, by considering alternate choices for the character in question with their confidences. Toussaint and Donaldson [T01-72] applied suboptimal block contextual decoding algorithm to hand-printed

character recognition, which searched over only the most probable bigrams and trigrams. In section 7.2.1 we used these approaches. Section 7.2.2 describes improvements in this algorithm.

7.2.1 Block Decoding Algorithm (BDA)

This method is similar to several previous approaches [CHU-75, DUD-68, TOL-72]. All these approaches, usually listed under the common name of Block Decoding Algorithms (BDA).

In this method a sequence, say, x_1, \dots, x_n of patterns is divided into several blocks of length $b \geq 1$, where 'b' is not necessarily the size of the sequence (word). The value of 'b' is decided by the user. This method is a pseudo Markov method as it assumes that each block is independent of all other blocks. For each pattern in a block, a set of possible alternative classes is considered. The algorithm then selects the most likely sequence for the block by maximizing some function for each letter-sequence (i.e., decodes the input sequence). The function is biased by adding the log transition probabilities. The formal description of this algorithm is given below.

The algorithm can be formalized into following steps.

Steps

1. Assume that x_1, x_2, \dots, x_n be the input sequence of n patterns.

2. Invoke SCRS and select 'd' for $1 \leq d \leq 26$, alternatives for each pattern X_1 . This step is already explained in section 5.2.
3. Approximate the N-gram by a suitable approximation of a sum of low order log probabilities.

) Step-3 made the algorithm more general in the sense that it does not reduce the N-gram to only one type of product approximation as is the case with Markov assumption. For example, assume the same three letter sequence i.e., the word 'HAT', is to be recognized. Using bigram probabilities one can compute $C(X)$ where

$$C(X) = \log P(X_1/H) + \log P(X_2/A) + \log P(X_3/T) + \log P(HA) + \log P(AT) \quad \dots 7.2-1$$

instead of bigram, if transition probabilities be used the function $C(X)$ will be:

$$C(X) = \log P(X_1/H) + \log P(X_2/A) + \log P(X_3/T) + \log P(A/H) + \log P(T/A) \quad \dots 7.2-2$$

If Markov assumption is used then the function $C(X)$ will be

$$C(X) = \log P(X_1/H) + \log P(X_2/A) + \log P(X_3/T) + \log P(H/'\phi') + \log P(A/H) + \log P(T/A) + \log P(''\phi'/T) \quad \dots 7.2-3$$

It has been shown by Toussaint and Donaldson [T01-72] that other approximation to $P(HAT)$ can also be used. They further reported that the use of trigrams instead of bigrams accounts for 80% of the improvement.

It should be noted that the approximations 7.2-1 to 7.2-3 hold within a block of sequence only. If the sequence is divided into more than one blocks and expression 7.2-3 be used, the approximation over the entire sequence will not be exactly Markov, since the blocks are independent. In our experiments we used expression 7.2-3.

Using this expression the algorithm was implemented as follows:

Block Decoding Procedure

Comments: The procedure includes 3 steps; namely: initialization step, decoding step and the decision step.

The procedure divides the input sequence into 'b' blocks, and invokes SCRS to obtain 'd' alternatives for each pattern in the block. It then decides the identity of all patterns in the block by maximizing the function $C''(X)$.

/* Define the following variables:

b := block size

strt := starting position of the sequence which is being processed

k := counter for number of sequences made*/

/*initialization step*/

strt := 1; wrdleft := n; k := 0;

Step-1

```
/* initialize the block using 'd' alternatives for each
   pattern */
```

```
if b > wrdleft then b:=wrdleft;
```

```
BLK (d,b) := GETBLK (strt,b,n);
```

```
/* GETBLK is a procedure which gets the block of 'b' patterns
   from sequence of n patterns from strt position and
   saves it in matrix BLK */
```

Step-2

```
/* decoding step */
```

```
/* This segment finds the heighest probability. that can be
   constructed from the 'd' alternatives for each letter
   position in the block. It proceeds position by position
   through the block, in the last stage finding the most
   probable string of length equal to 'b'. Thus the decoding
   process is b-stage process and at stage i a new sequence
   upto and including position i is formed. Thus to decode a
   block of sequence of length 'b', with each pattern having
   d-alternatives, every stage is repeated 'd' times. The
   'b' stages are ST-1, ST-2, ..., ST-b */
```

ST-1

```
if strt = 1 then
```

```
Begin /* it is the first pattern of the sequence */
```

```

For j1:=1 to d do
  Begin
    g(j1) := log P(X1/BLK(j1,1)) + log P(BLK(j1,1)/'Ø');
    S1'(BLK(j1,1)) := BLK(j1,1);
    if n = 1 then
      Begin
        k := k + 1;

        Mk(S1') := g(j1);
        Sk := BLK(j1,1);

      End;
    End;
  End;

Else Begin /*it is other than the first block of the
          sequence*/

  For j1:=1 to d do
    Begin
      g(j1) := log P(Xstrt/BLK(j1,1));
      S1'(BLK(j1,1)) := BLK(j1,1);
    End;
  End;

ST-2 /* construct the d(b-1) sequences */

if n = 1 then goto ST-3

for ji:=1 to d do /* 2 ≤ i ≤ (b-1) */
begin

```

$$g(j_i) := g(j_{i-1}) + \log P(X_{\text{strt}+i-1}/\text{BLK}(j_i, i));$$

$$S'_i(\text{BLK}(j_i, i)) := \text{BLK}(j_i, i);$$

if $b \geq (i+1)$ then

$$g(j_i) := g(j_i) + \log P(\text{BLK}(j_i, i)/\text{BLK}(j_{i-1}, i-1))$$

ST-b

/* It is the last stage. At this stage the 'd' strings of 'b' letters are constructed and the one which is most likely is kept. The operator CONCAT has the same function as described in section 6.2 */

for $j_b := 1$ to d do

begin

$$g(j_b) := g(j_{b-1}) + \log P(X_{\text{strt}+b-1}/\text{BLK}(j_b, b));$$

$$S'_b(\text{BLK}(j_b, b)) := \text{BLK}(j_b, b);$$

$$\text{temp}_{j_b} := g(j_b);$$

End; /* Stage b */

$k := k + 1;$

$$M_k(S'_b) := \text{MAXPOS}(\text{temp}_{1_b}, \text{temp}_{2_b}, \dots, \text{temp}_{d_b}, u);$$

$$S_{0 \rightarrow \text{strt}+b-1}^k := S_{0 \rightarrow \text{strt}-1}^k \text{ CONCAT } S'_b(u);$$

End (i); /* end of $2 \leq i \leq (b-1)$ stages */

$$S'_i(\text{BLK}(j_i, i)) \text{ for } 1 \leq j_i \leq d; 1 \leq i \leq b;$$

is the string constructed from the letters in the block while going through all 'b' stages, S_0 is blank.

S^k represent the string against maximum $g(j_b)$'s.

Thus d^{b-1} such strings will be constructed when all these b stages for a sequence of 'b' patterns with d alternative each, are executed.

Change the values of strt and wrleft as 'b' patterns among n have been processed; such that $strt+b-1 \leq n$. Repeat steps 1 and 2 until the entire sequence is finished.

Step-3 /* decision step*/

Comments: The steps 1 and 2 are repeated until the entire sequence is decoded. Thus steps 1 and 2 construct d^{b-1} sequences. The decision step selects the one which is most likely to occur. The function MAXPOS, as described in section 6.2, searches the maximum among several given values (in this case k values) and also locates its position.

for $p:=1$ to k do

begin

$G_p := M_p + \log P ('b' / (S_b - S_{b-1}))$

End;

$M := \text{MAXPOS} (G_1, G_2, \dots, G_k, u);$

stop as the designated word is in $S_1 \rightarrow strt+b$ ending with a blank.

Experiment using BDA (EXPERIMENT - 3)

Using this algorithm as CPPS the text recognition experiment was repeated on set of three passages. The results obtained are shown in TABLE 7.1.

As seen from the table, the character recognition rates obtained from this algorithm are very poor as compared to those obtained from VA and MVA. This method took the maximum amount of comparisons (see below under computational complexity) and storage among all algorithms discussed so far in this thesis. For block size and depth (2,2), (2,3), (3,2) and (3,3) results are very poor, even lower than those obtained by SCRS i.e., without using context. The obvious reason was the independent nature of blocks. The results however, get stabilized for $d = 4$ and $b = 5$ which are better than those obtained from SCRS, but at the expense of heavy complexity and enormous memory. The results are still poorer than both VA and MVA at $d = 4$. As an example, for $d = 4$ and $b = 5$, 85.392% correct character recognition rate on Pas (1, Mun₂₈) was observed, whereas VA has given 87.376% and MVA has also achieved 87.376% correct recognition.

This algorithm does not seem to have any merits except the facility to use any discriminant function at the decoding step. For example we can use any discriminant function among 7.2-1 to 7.2-3. Computation and memory requirements are enormous and they increase exponentially with increase in 'd'.

TABLE - 7.1

Recognition rates obtained by using
Block Decoding Algorithm

Subject 6

		depth of search			
Block Size		2	3	4	5
Pas-1	2	86.835	86.835	86.925	86.925
	3	89.360	89.360	90.532	90.532
	4	92.606	92.606	93.508	93.508
	5	93.327	93.327	94.590	94.590
Pas-2	2	87.374	87.374	87.557	87.557
	3	92.864	92.864	93.321	93.138
	4	93.230	93.230	93.596	93.596
	5	95.425	95.425	95.883	95.883
Pas-3	2	84.763	84.763	84.945	84.945
	3	90.237	90.237	90.602	90.602
	4	92.153	92.153	92.518	92.518
	5	93.704	93.704	94.069	94.069

T A B L E - 7.1 (CONTD)

Recognition rates obtained by using
Block Decoding Algorithm

Subject 18

		depth of search			
Block Size		2	3	4	5
Pas-1	2	65.915	66.456	66.456	66.456
	3	70.604	71.506	71.506	72.317
	4	73.219	74.301	74.301	75.203
	5	74.482	75.383	75.383	76.465
Pas-2	2	68.344	68.344	68.344	68.344
	3	71.455	72.461	72.461	72.736
	4	73.559	74.016	74.016	74.219
	5	74.108	75.023	74.840	75.114
Pas-3	2	68.431	69.434	69.069	69.252
	3	72.536	74.453	74.361	75.364
	4	75.091	76.825	76.460	77.737
	5	75.912	77.920	77.463	78.650

T A B L E - 7.1 (CONTD)

Recognition rates obtained by using
Block Decoding Algorithm

Subject 28

		depth of search			
Block Size		2	3	4	5
Pas-1	2	78.269	77.998	77.998	77.998
	3	81.425	81.785	81.785	81.785
	4	84.130	84.310	84.400	84.400
	5	85.032	85.392	85.392	85.392
Pas-2	2	72.370	72.370	72.370	72.370
	3	78.042	78.957	78.683	78.683
	4	78.866	79.506	79.506	79.506
	5	81.244	82.068	81.702	81.702
Pas-3	2	76.369	76.186	76.095	76.095
	3	82.664	82.938	82.755	82.755
	4	83.759	83.850	83.850	83.850
	5	86.040	86.040	86.040	86.040

TABLE - 7.1 (CONTD)

Recognition rates obtained by using
Block Decoding Algorithm

OCR 'B'

		depth of search			
Block Size		2	3	4	5
Pas-1	2	84.941	85.663	85.663	85.663
	3	88.638	89.360	89.360	89.360
	4	90.983	92.335	92.335	92.335
	5	90.606	93.598	93.598	93.598
Pas-2	2	86.093	86.459	86.459	86.459
	3	91.125	91.674	91.674	91.674
	4	91.857	92.498	92.498	92.498
	5	93.962	94.602	94.602	94.602
Pas-3	2	83.759	84.215	84.215	84.215
	3	88.412	89.690	89.690	89.690
	4	90.146	91.332	91.332	91.332
	5	91.788	92.883	92.883	92.883

for the same size of word.

Computational Complexity

Considering the block size be 2 and assume 'd' alternatives be available then the number of comparisons and additions required to decode this sequence are respectively, $d(d-1)$ and $2d^2$. Thus the amount of computations required only by decoding procedure, is:

$$d^2 - d + 2d^2$$

$$\text{or } 3d^2 - d$$

For example, if $b = 5$ then $d(d^4 - 1)$ and $5d^5$ will be the number of comparisons and additions respectively; thus the amount of computations required is

$$d(d^4 - 1) + 5d^5$$

$$\text{or } 6d^5 - 4d$$

or in general

$$(b + 1) d^b - (b - 1) d$$

...7.2-4

This amount is still much lower than the one computed by Chung [CHU-75].

Thus for an input sequence of n patterns the amount of computations will be

$$(\lceil n/b \rceil - 1) \times [(b+1) d^b - (b-1) d]$$

$$+ [(b'+1) d^{b'} - (b'-1) d]$$

...7.2-5

$\lceil n/b \rceil$ is the ceiling of the number i.e., if $n = 13$ and $b = 5$

then $\lceil n/b \rceil$ will be 3.

Second term in equation 7.2-5 is the amount of computations for the last block and $1 \leq b' \leq b$.

7.2.2 Imbricate Block Decoding Algorithm (IBDA)

Block decoding algorithms did not get their recognition because of heavy complexity, and excessive time and memory requirements. Keeping these three factors in view we investigated this algorithm further and by modifying the basic assumptions we proposed a modified algorithm referred to as Imbricate Block Decoding (IBDA). This algorithm assumes Markov dependence and proceeds in blocks of length 'b'. From this assumption, while decoding (or recognizing) the sequence contained in the block, we need to overlap the consecutive blocks. This assumption thus does not allow blocks to be independent any further. The value of 'b' is decided by the user and it is not necessarily be the size of a word.

Efficiency in storage, computations and recognition performance is realized by:

- A. Similar to BDA defining a depth of search parameter 'd' where $1 \leq d \leq 26$, to reduce the number of possible classes.
- B. Assuming the Markov dependence over the entire sequence.
- C. Approximating the N-gram by a suitable approximation of a sum of low-order log probabilities.

- D. If $d > 2$ then reducing the number of alternatives to d' , where $2 \leq d' \leq d$ through a pruning algorithm described later in this section.

It is important to note that the point 'C' is not algorithm dependent. The algorithm is described below:

Steps

1. Assume X_1, X_2, \dots, X_n be the sequence of input patterns, where n designate the length of the sequence; assign n to a variable, say, wordleft.
2. Invoke SCRS and select ' d ' alternatives for each pattern X_i .
3. Let ' b ' be the size of the block, its value is heuristic and is decided by the user.
4. If wordleft $\leq b$ then $b := \text{wordleft}$, i.e., if length of input sequence is less than block size then set the block size to wordleft for that input sequence.
5. Invoke the procedure GETBLK to get a block of ' b ' letters with their ' d ' alternatives from input sequence. Its format is

GETBLK (strt, b, n);

where strt is the positional value of first pattern of block in the sequence.
6. If $n = 1$ or $d \leq 2$ or $b = 1$ skip the PREDECODING SCAN, otherwise pass the block through predocoding scan to keep the most appropriate alternatives only. This scan, as

described below, holds at least 2 alternatives for each pattern, except the first, in the block.

7. Using the expression 7.2-3 select the most likely sequence out of $d \times (d'=2)^{b-2}$ (in best case) or d^{b-1} (in worst case) possible sequence. If $n=1,2$ or $b=1,2$ then 'd' sequences will always be formed.
8. If n is greater than 'b' then after deciding on 'b' patterns go to step 4 i.e., shift the block by $(b-1)$ patterns (visually), while keeping the decision about the b -th pattern along with its d'_b alternatives. This means that new block of 'b' patterns (or less, if the length of sequence left is smaller than b) is constructed with the b -th pattern with its d'_b alternatives as the first column and next $(b-1)$ letters, with their 'd' alternatives for its subsequent columns. If $b=1$ then shifting is done by one block and current block is considered to be dependent upon the previous block.
9. Repeat the steps 4 through 8 until the decision on the entire sequence of n letters is being made.

To implement the algorithm we adopted the similar procedure as described for BDA. The only difference between two was the step 6 i.e., the predecoding scan, and the step 8. From step 6, the 'd' alternatives for every feature vector were reduced to d'_i , where $2 \leq d'_i \leq d$, and $2 \leq i \leq b$. Thus the b stages of decoding does not have to repeat themselves, 'd' times each. This procedure is described below. From step 8,

i.e., shifting the block by $(b-1)$ patterns, the Markov assumption was implemented. Rest of the details were the same as described earlier for BDA.

Predecoding Scan

If the depth of search 'd' is greater than 2, then to reduce the computations and to eliminate the number of less probable letter sequences, the block of 'b' patterns with 'd' alternatives each, is processed through a predecoding scan. This pruning is done by proceeding from left to right and calculating the sum of 'd' (maximum) likelihood values for each pattern in the preceding column to be followed by each pattern in the following column and the contextual information about them. That is

$$g'_k(X) := \log P(X_i/Z_i) + \log P(X_{i+1}/Z_{i+1}) + \log P(Z_{i+1}/Z_i) \quad \dots 7.2-6$$

for $1 \leq k \leq d, d > 2$

is computed. The pattern alternatives corresponding to two maximum $g'_k(X)$'s are retained in the following column of the block, and the rest are eliminated. This arrangement further reduces the number of 'd' alternatives (initially) to a manageable size (less than d for each X_i ; maximum 'd').

This predecoding scan can be formalized as below:

Comments: This procedure reduces the d ($d > 2$) alternatives for each letter to d' where $2 \leq d' \leq d$.

if $d > 2$ then

Begin

for $i:=1$ to $(b-1)$ do

Begin

for $k:=1$ to d do

$$g_k'(X) := \log P(X_i/Z_i) + \log P(X_{i+1}/Z_{i+1}) + \log P(Z_{i+1}/Z_i);$$

For $i:=1$ to 2 do

$$\text{BLK}'(u, i+1) := \text{MAXPOS}(g_1, \dots, g_d, u);$$

End;

End;

Experiment using IBDA (EXPERIMENT - 4)

Using this algorithm, the text recognition experiment was conducted on the same set of 3 passages. The results obtained are shown in TABLE 7.2. The results shown are encouragingly better than block decoding algorithm (see TABLE 7.1) and are very close to those obtained by MVA.

As seen from the table, the values $b = 5$ and $d = 4$ gave the better recognition rates. For these values character recognition rates for 3 subjects and OCR 'B' data were:

T A B L E - 7.2

Recognition rates obtained by using
Imbricate Block Decoding Algorithm

Subject 6

		depth of search			
	Block Size	2	3	4	5
Pas-1	2	95.221	95.221	96.483	96.483
	3	95.311	95.311	96.573	96.573
	4	95.221	95.221	96.483	96.483
	5	95.401	95.401	96.664	96.664
Pas-2	2	96.798	96.798	97.347	97.347
	3	97.072	97.072	97.621	97.621
	4	96.798	96.798	97.347	97.164
	5	96.889	96.889	97.438	97.438
Pas-3	2	95.073	95.073	95.438	95.438
	3	95.347	95.347	95.712	95.712
	4	95.347	95.347	95.712	95.712
	5	95.529	95.529	95.894	95.894

T A B L E - 7.2 (CONTD)

Recognition rates obtained by using
Imbricate Block Decoding Algorithm

Subject 18

		depth of search			
Block Size		2	3	4	5
Pas-1	2	75.203	76.014	76.014	76.104
	3	75.834	76.465	76.375	77.187
	4	75.653	76.646	76.646	77.728
	5	76.104	77.096	77.006	77.908
Pas-2	2	75.206	75.663	75.663	75.663
	3	75.663	76.304	76.304	76.578
	4	75.389	75.938	75.846	76.121
	5	75.572	76.487	76.487	76.761
Pas-3	2	76.642	78.650	78.285	78.467
	3	77.190	79.288	78.741	79.836
	4	77.281	79.288	78.832	80.018
	5	77.372	79.380	78.832	80.109

T A B L E - 7.2 (CONTD)

Recognition rates obtained by using
Imbricate Block Decoding Algorithm

Subject 28

		depth of search			
Block Size		2	3	4	5
Pas-1	2	86.114	86.023	86.023	86.023
	3	86.925	87.466	87.466	87.466
	4	86.745	87.196	87.196	87.196
	5	87.015	87.466	87.466	87.466
Pas-2	2	80.238	80.512	80.512	80.512
	3	81.701	82.800	82.068	82.068
	4	81.976	82.434	82.434	82.434
	5	82.342	83.166	82.617	82.708
Pas-3	2	86.679	86.314	86.131	86.131
	3	87.774	87.774	87.774	87.774
	4	87.682	87.956	87.774	87.774
	5	87.956	88.047	87.956	87.956

T A B L E - 7.2 (CONTD)

Recognition rates obtained by using
Imbricate Block Decoding Algorithm

OCR 'B'

		depth of search			
Block Size		2	3	4	5
Pas-1	2	94.229	95.221	95.221	95.221
	3	94.229	95.221	95.221	95.221
	4	94.229	95.491	95.491	95.491
	5	94.229	95.491	95.491	95.491
Pas-2	2	95.791	95.791	95.791	96.249
	3	95.700	95.700	95.700	96.157
	4	95.608	95.608	95.608	96.066
	5	95.608	95.608	95.608	96.066
Pas-3	2	93.613	94.252	94.252	94.343
	3	93.613	94.343	94.343	94.343
	4	93.522	94.708	94.708	94.799
	5	93.613	94.526	94.526	94.617

i (Pas-i, Mun₆) (Pas-i, Mun₁₈) (Pas-i, Mun₂₈) (Pas-i, OCRB)

1	96.664	77.006	87.466	95.491
2	97.438	76.487	82.617	95.608
3	95.894	78.832	87.956	94.526

The algorithm is better than BDA as far as recognition performance, computations and memory requirements are concerned. The recognition performance of MVA and IBDA is similar, but IBDA is still more expensive than MVA as it takes more computations to decide on a sequence.

Computational Complexity of IBDA

Using equation 7.2-4. The amount of computations required for decoding a block of 'b' is

$$(b+1) \sum_{i=1}^b d_i - \sum_{i=1}^{b-1} d_i \quad \dots 7.2-7$$

for $2 \leq d_i \leq d$

If all d_i 's are equal to d then because of Predecoding Scan the amount of computation will be little over than required by BDA. But experimentally it was observed that we never met such case. In majority of situations pruning done at predecoding scan level reduced ' d ' alternatives for every x_i , ($2 \leq i \leq b$) to 2 or 3.

7.3 DICTIONARY METHOD

Dictionary (Table) look-up methods are among the oldest methods of using contextual information. This method is suitable in limited vocabulary text-recognition systems. Past approaches have already been reviewed in Section 2.4. In the present work a more efficient approach than those taken previously, is developed. Basic concepts are the same as described by Toussaint in his dictionary method [T05-77] which was an extension of Bledsoe-Browning method [BLE-59]. Similar to dictionary method described in [T05-77] and [SH3-79], the method described here also achieves the efficiency, not by reducing the memory requirements but by reducing the computational complexity. The algorithm developed here will be called Dictionary Look-up algorithm (DLA). It is comprised of two procedures. Their description is given below:

1. Construction of Dictionary Procedure

Consider a dictionary $D(N)$ of N English words be divided into n subdivisions; namely: $dic(1), dic(2), \dots, dic(n)$. The subscripts $1, 2, 3, \dots, n$ represent the number of letters in words belonging to that subdivision. Each subdivision is called as subdictionary. We will be using both words subdivision and subdictionary interchangeably. In each subdictionary, the words (each word is different from others) of same lengths are stored, Therefore:

subdictionary $\text{dic}(1)$ contains words of length 1

subdictionary $\text{dic}(2)$ contains words of length 2

subdictionary $\text{dic}(n)$ contains words of length n

Let m_1, m_2, \dots, m_n be the number of words in each
subdictionary; m_1, m_2, \dots, m_n may not necessarily be equal.

Thus total number of words in the dictionary are:

$$N = \sum_{i=1}^n m_i \quad \dots 7.3-1$$

Assuming the address of $\text{dic}(1)$ be 1, the dictionary n
can be accessed by computing the formula

$$\sum_{i=1}^{n-1} m_i + 1 \quad \dots 7.3-2$$

Following the mathematical model given by expression
1.3-4 a word of n letters in $\text{dic}(n)$ can be represented as

$$z_1, z_2, \dots, z_n$$

Using transition probabilities the value $\text{VAL}(z_1, \dots, z_n)$
of the word is defined by the expression:

$$\text{VAL}(z_1, \dots, z_n) = \sum_{i=1}^{n+1} \log P(z_i / z_{i-1}) \quad \dots 7.3-3$$

Thus the second term of the expression 6.3-5 gives the
value of the word. The value of any word is a constant and it
is also stored in the dictionary.

2. Classification Procedure

To classify the input pattern sequence X_0, X_1, \dots, X_{n+1} , the procedure consists of two steps. In the first step only the sequence X_1, \dots, X_n is recognized as Z_1, \dots, Z_n using SCRS for $d = 1$. The word recognized is considered as the designated word if the value VAL of the sequence computed from expression 7.3-3 is present in the dictionary $\text{dic}(n)$. If the classification was unsuccessful at this step, second step is invoked.

Second step must classify the sequence and to do so it needs to maximize the expression 6.3-5 over all words in subdictionary $\text{dic}(n)$. Assume the quantity given by this expression be called SCORE; then

$$\text{SCORE} = \sum_{i=1}^n \log P(X_i/Z_i) + \sum_{i=1}^{n+1} \log P(Z_i/Z_{i-1}) \quad \dots 7.3-4$$

Following the assumption given in Section 6.3 the patterns X_0 and X_{n+1} are labelled blank, and Z_0 and Z_{n+1} also blank.

In this step the algorithm thus selects the subdictionary $\text{dic}(n)$ at address given by equation 7.3-2 and picks that n -letter word which has the highest SCORE in $\text{dic}(n)$.

TABLE - 7.3

Dictionary Constituents

Length of word n	Number of such words in subdictionary dic(n)
1	2
2	21
3	43
4	81
5	82
6	55
7	37
8	33
9	24
10	12
11	6
12	4
13	1
<hr/>	
Total	401

Experiment using Dictionary Method (EXPERIMENT - 5)

In all 401 distinct words were found to be present in three passages. A dictionary of these words was compiled. The constituents of the dictionary are shown in TABLE 7.3. Using the transition probabilities (see section 6.2.1) the value VAL of every word in the dictionary was computed and stored corresponding to them. The words within any subdictionary $dic(i)$ were arranged in descending order of their values.

Consider x_1, x_2, \dots, x_n be the input sequence, where n represents its length. The value of n gives the subdictionary to be looked-up. The description of look-up algorithm was structured into following two procedures:

1. Preliminary Classification:

This procedure includes three steps

- a1. Invoke SCRS for $d = 1$ and get the sequence recognized.
- b1. Using equation 7.3-3 find the value of this sequence.
- c1. Using binary search find whether this value is present in subdictionary $dic(n)$. If so, decide the word recognized as the designated word; otherwise go to secondary classification step.

2. Secondary Classification

This procedure consists of following two steps.

- a2. Using equation 7.3-4 compute the score of every word in subdictionary $dic(n)$.

T A B L E - 7.4

Recognition rates obtained by using
Dictionary Look-up Algorithm

Passage	Subjects from Munson's Data			OCR 'B'
	6	18	28	
Pas-1	99.910	99.820	99.008	100.000
Pas-2	100.000	99.085	99.396	100.000
Pas-3	100.000	99.818	99.544	100.000

b2. Output the dictionary word corresponding to maximum SCORE as designated word.

The experiment was conducted on all three passages both written by 3 subjects and OCR 'B'. The results obtained are listed in TABLE 7.4.

On machine-printed passages, the algorithm was able to achieve 100% correct recognition whereas on hand-printed passages recognition performance was over 99%. The recognition performance of this method appeared to be the best among those studied so far in this thesis.

DISCUSSION

This algorithm uses two stages of classification. An experiment was conducted to study the performance of both steps. In this experiment only the Pas-1 of 3 subjects and Pas(1, OCRB) were used. The results of the study are shown in TABLE 7.5. It is surprising that the results obtained in preliminary classification by SCRS are again seem to play an important role. The recognition performance at preliminary classification step is better in case of subject 6 and OCR 'B' only. Thus the introduction of this step for subject 6 and OCR 'B' is saving exhaustive computations and search of secondary classification. In case of subjects 18 and 28 preliminary step is not doing so. This fact thus concludes that the preliminary classification step should only be introduced, if the recognition performance of SCRS on a

T A B L E - 7.5

Words in Pas-1 recognized at preliminary
classification step of DLA

length of word n	total no. of such words	words correctly recognized			
		6	18	28	OCR 'B'
1	10	8	9	9	10
2	45	35	16	15	45
3	52	34	23	9	50
4	46	34	8	7	41
5	29	19	4	4	21
6	23	16	2	4	16
7	25	13	5	2	3
8	12	7	-	1	5
9	8	5	-	-	6
10	1	-	-	-	-
11	3	2	-	-	-
12	-	-	-	-	-
13	-	-	-	-	-
<hr/>					
Total	254	173	67	51	197
percentage of words recog.		68.110	26.378	20.079	77.559
percent. n. < 4		43.701	22.047	15.748	57.480

particular subject is over 90%. Below 90% it is not advisable to use this step.

Computational Complexity:

The classification procedure of the dictionary method involves two steps. Their computational complexity is given below:

1. The preliminary classification step needs only n additions to compute the value of n -letter word recognized by SCRS for $d = 1$. To binary-search this value in subdictionary $\text{dic}(n)$ which contains m_n words require $\log_2 m_n + 1$ comparisons. Thus total computations required to recognize a word of n letters:

$$n + \log_2 m_n + 1 \quad \dots 7.3-5$$

2. The computational complexity for the second step alone, include to compute the SCORES of words and searching the maximum score.

Considering equation 7.3-4, the computation of SCORE of a n -letter word require $n+1$ additions. The first quantity in this equation require n additions. The second quantity which is already stored in the dictionary is added to the first require another addition. There are m_n n -letter words. Thus total number of computations needed for m_n scores is

$$(n+1) m_n$$

To select the word with the maximum score needs m_n comparisons. Thus the total computations of second step is

$$\begin{aligned} (n+1) m_n + m_n - 1 \\ n m_n + 2m_n - 1 \end{aligned} \quad \dots 7.3-6$$

If the DLA involves both steps of the classification and if the word was not classified at preliminary step then the word needs

$$n + \log_2 m_n + 1 + n m_n + 2m_n - 1 \quad \dots 7.3-7$$

computations.

Thus to decide on the addition of the preliminary step one should need to be sure that this step should classify that many words such that

$$(Z-z) [n + \log_2 m_n + 1] \leq z [n m_n + 2m_n - 1] \quad \dots 7.3-8$$

Where Z is the total number of words in the text to be recognized; and z is the number of words recognized by the preliminary classification step.

7.4. HYBRID METHOD

The hybrid methods use both dictionary and the contextual information. An hybrid approach introduced by Shinghal et al. [SH3-79] is a combination of dictionary and modified Viterbi algorithm. The algorithm is known as Predictor-Corrector algorithm (PCA) and it achieved the character recognition rate of 96.4% using machine-printed characters of Ryan's data set (Pattern Recognition Data Base

TABLE - 7.6

Recognition rates obtained by using
Hybrid Method

Passage	Subjects from Munson's Data			OCR 'B'
	6	18	28	
Pas-1	100.000	100.000	99.098	100.000
Pas-2	100.000	99.268	99.817	100.000
Pas-3	100.000	99.818	99.818	100.000

No. 1.1.1A, IEEE computer society).

Following the approach of PCA, we also combined the MVA and DLA, and called this method as Hybrid method. MVA and DLA are described in sections 6.3 and 7.3 respectively.

Thus the hybrid approach involved two steps: prediction step and the correction step. The prediction step uses MVA and predicts the probable word sequence by searching through d-alternatives provided by SCRS. The value of the predicted word is then calculated and it becomes the designated word if such value exists in the corresponding subdictionary. The second step is invoked only if the decision about the identity of the word was not made at step 1. At this step DLA is used which substitutes the predicted word with the word (in the dictionary) of maximum score. In DLA the preliminary classification step was not included as MVA in step 1 has already done that.

Experiment using Hybrid Method (EXPERIMENT - 6)

Using this hybrid method of combined approach, the text-recognition experiment was conducted on 3 passages. The results obtained are shown in TABLE 7.6. Results shown are the best among all previous algorithms discussed in this thesis. The results on passages written by subject 6 and OCR 'B' show that if the character recognition rates obtained by SCRS are over 90%, perfect performance can be obtained by using this method. Comparing the results shown in TABLE 7.6

with those given in TABLE 5.1-A and TABLE 5.1-B, the gain in recognition performance is quite obvious.

The computational complexity of the algorithm depends upon the values of 'd' i.e., the depth of search.

If $d = 1$, the algorithm reduces to DLA. The optimal value of d was found to be 2. As 'd' increases the percentage of the words to be looked-up in the dictionary decreases. When 'd' reaches the value as defined to be optimal for MVA, the number of words to be searched by the dictionary becomes constant.

It should be noted that any other method like VA, BDA, IBDA, or any other product approximation approaches could also replace MVA.

The computational complexity of this algorithm depends upon the type of the algorithm used at the prediction step. It further depends on the number of words recognized at both steps. However, the mathematical model for computational complexity of both of the steps is already given in sections 6.4 and 7.3.

7.5 FINAL REMARKS

In chapters 6 and 7 five contextual algorithms were studied. Imbricate Block Decoding Algorithm (IBDA) and the preliminary classification step in Dictionary Look-up method

are the prime contribution of this research. Preprocessing and size-normalizing algorithms to enhance the quality of the character patterns were also designed and implemented. The objectives for the proposed text recognition system, we set in section 1.2, were fairly met by MVA and Hybrid method. Improvements in recognition performance have proved that contextual postprocessing helped in achieving better recognition performance. The amount of computations required by each method was also estimated, which will help the user to choose any algorithm subject to his requirements and the quality of data.

CHAPTER 8

CONCLUSION AND SOME SUGGESTIONS FOR FURTHER RESEARCH

8.1 PREVIEW OF THE CHAPTER

This chapter concludes the study made in this thesis and presents some suggestions for further study and research. The results of the experiments conducted in chapters 5, 6 and 7 are reviewed and compared in section 8.2. During the course of experiments, it was observed that keeping the same basic concepts and methodologies, several improvements in the intermediate structure of the system could be made in order to improve the system's modularity, economics and performance. Some of the observations are listed in section 8.3. Section 8.4 presents a few suggestions for further research in this area of study.

8.2 SUMMARY REVIEW

In this section the system is reviewed first and then its performance in the light of experiments is evaluated. Conclusions thus derived are discussed simultaneously.

SYSTEM'S REVIEW

A machine-based text-recognition system for any type-font printed alphameric characters has been developed. The design

of the system includes: (1) preprocessing stage (Stage - 1), (2) single character recognition subsystem "SCRS" (Stage - 2), and (3) contextual postprocessor or "CPPS" (Stage - 3).

Stage - 1 further includes three steps, namely: (a) elimination of noise, (b) size normalization, and (c) feature extraction. These steps have already been described in chapter 3. Steps (a) and (b) are data-dependent and therefore, they are optional. In case of good single-font machine-printed (like OCR 'A' and OCR 'B' data sets) and very good hand-printed, well tutored and constrained, characters these steps could be skipped. It is possible, because in such cases our feature-extraction scheme will be more mathematical than stochastic. Smoothing process i.e., steps (a) and (b), however enhance the quality of input data.

Stage - 2 or (SCRS) is the base of the system. Preliminary classification of the character is made at this stage. It also gives significant information about the type of algorithm to be used as contextual processor in stage 3. For example, if we obtain 99% to 100% correct character recognition rate from SCRS, then we do not need to go for any further classification. Characters included in OCR 'A' and the subject 15 of Munson's data base are examples of such a case (see TABLE 5.1-A & 5.1-B). If the preliminary recognition rates are high enough i.e., between 94% to 99% then SCRS can help in deciding the algorithm to be used as contextual postprocessor. For example, subject 6 (some other

as well) do not need any highly sophisticated, and complex algorithm for secondary classification. At this stage average recognition rate for subject 6 was 92.637% (see TABLE 5.1-B) using simply the modified Viterbi algorithm we were able to achieve 96.757% correct recognition on average (see TABLE 6.2).

Stage 3, i.e., CPPS serves as a secondary classifier and uses contextual information as an aid. If the preliminary classification results are not satisfactory (as specified by the user) then this stage could be added. It has already been shown that its use improves the recognition rate remarkably. Five algorithms were tested as contextual postprocessor.

REVIEW ON EXPERIMENTAL RESULTS AND CONCLUSION

With and without using contextual information six experiments were conducted altogether. It was observed that the choice of the algorithm to be used depends upon the classification rates of SCRS. Other factor in decision of choice is of course, user's specifications about recognition performance. TABLE 8.1 presents the summary of results observed from these experiments.

As shown by TABLE 8.1, better performance was observed when SCRS operates on machine-printed data. These results also show that for single font machine-printed data, we do not need highly sophisticated recognition algorithm. Simply the SCRS, obtained the recognition rates of 100% and 95.478% (see

TABLE 5.1-A) respectively, while operating on the testing sets from OCR 'A' and OCR 'B'.

An optimal value of training set was investigated and it is concluded that in case of good quality single type-font printed characters very little training is required. This minimum value should be however, at least 40% of total number of sample characters in the data set.

SCRS is a good check-point for examining the need for CPPS. If it is observed that the use of CPPS is essential then depending upon user's specifications, SCRS would further help in deciding over the kind of algorithm to be used as CPPS.

Behaviour of all the algorithms on machine-printed passages was far more better than on hand-printed passages. For example, SCRS, which is simply a maximum selector, was able to obtain 100% and 94.756% (on average) correct recognition on three passages compiled from OCR 'A' and OCR 'B' data sets respectively. This concludes that for hand-printed passages simple methods are not appropriate.

Viterbi algorithm and its modifications have satisfactory performance on both machine and hand-printed data. As an example, using MVA, character recognition performance on PAS (1, MUN28), PAS (2, MUN28), and PAS (3, MUN28) for $d = 4$, was 87.376%, 83.074% and 88.230% correct respectively.

Table 8.1

Summary of Results and Comparison of Character Recognition Rates Obtained by Using Different Algorithms.

Method	Passage	6	18	28	OCR'B'
SCRS	1	91.794	72.137	71.686	94.500
	2	94.511	72.187	69.442	95.791
	3	91.606	74.544	73.631	93.978
	average	92.637	72.956	71.586	94.756
VA	1	96.664	78.539	87.376	95.762
	2	97.347	76.578	83.074	96.249
	3	96.077	80.383	88.230	95.073
	average	96.696	78.500	86.227	95.695
MVA (d = 4)	1	96.664	77.547	87.376	95.762
	2	97.530	76.359	83.074	96.249
	3	96.077	79.197	88.230	95.073
	average	96.757	77.701	86.227	95.695
BDA b = 5, d = 4	1	94.590	75.383	85.392	93.598
	2	95.883	74.840	81.702	94.602
	3	94.069	77.463	86.040	92.883
	average	94.847	75.895	84.378	93.694
IBDA b = 5, d = 4	1	96.664	77.006	87.466	95.491
	2	97.438	76.487	82.617	95.608
	3	95.894	78.832	87.956	94.526
	average	96.665	77.442	86.013	95.208
DLA	1	99.910	99.820	99.008	100.000
	2	100.000	99.085	99.396	100.000
	3	100.000	99.818	99.544	100.000
	average	99.970	99.571	99.316	100.000
Hybrid d = 2	1	100.000	100.000	99.098	100.000
	2	100.000	99.268	99.817	100.000
	3	100.000	99.818	99.818	100.000
	average	100.000	99.695	99.578	100.000

Achieving better performance, Imbricate Block Decoding algorithm (IBDA) is far less expensive than Block Decoding Algorithm (BDA). As shown in the table, using this method 87.466%, 82.617%, and 87.956% correct recognition was observed on three passages written by subject 28. Comparing with Viterbi algorithms (VA, & MVA), this method is still expensive and thus lacking practicality.

Dictionary method is an expensive method and could only be recommended in limited vocabulary situations. The performance of this method on machine-printed passages was 100% and over 99% on hand-printed passages.

Using hybrid method best recognition performance was observed. Again on machine-printed passages 100% and close to 100% recognition performance was achieved on hand-printed passages. This method ensures better recognition, since it makes an efficient use of dictionary method. Apart from dictionary method none of the above methods provide that degree of surity. But in cases, where characters are badly written and SCRS performance is very low, the corrector step of this algorithm will be burdened and overheads will be close to those for dictionary method.

The average length of the words in Pas-1, Pas-2, and Pas-3 was 4.366, 4.286, and 4.317, respectively. The word recognition rates on some of the important algorithms discussed in this thesis are shown in TABLE 8.2. The word

T A B L E - 8.2

Word Recognition rates obtained by using
Some Selected Algorithms

Method	Passage	subjects from Munson's data			OCR 'B'*
		6*	18	28	
MVA	1	87.008	37.795	62.992	83.858
	2	89.804	41.569	54.902	85.882
	3	86.275	39.216	68.235	81.176
IBDA	1	87.008	36.614	62.598	82.268
	2	89.412	41.569	54.510	83.137
	3	85.490	38.824	66.275	78.431
DLA	1	99.606	99.213	97.638	100.000
	2	100.000	97.647	98.431	100.000
	3	100.000	99.608	98.824	100.000
HYBRID	1	100.000	100.000	98.031	100.000
	2	100.000	98.431	99.608	100.000
	3	100.000	99.608	99.608	100.000

Number of words in Pas-1 = 254

Number of words in Pas-2 = 255

Number of words in Pas-3 = 255

* A list of original and the words misrecognized in passages written by subject 6 and OCR 'B' is shown in Appendix-B.

recognition rates obtained from SCRS and BDA were too low to report.

Comparing the results of SCRS, the performance of algorithms at CPPS stage thus approved the use of contextual information as an aid to text recognition system.

In the last, we suggest one to use MVA (with $d = 4$) or hybrid method (with $d = 2$) as CPPS. Choice of one among these depends upon user's requirements. If one needs to ensure close to 100% recognition, no matter what cost he has to pay, we suggest him to use hybrid method, otherwise MVA is a better compromise between cost and performance.

Results of different algorithms on passages Pas-1 and Pas-3 were better than those on Pas-2. This implies that the use of context provides better results if the passages are written in common writing style. It should be noted that the statistical test made on the passages had already concluded that Pas-2 did not represent a commonly used English (see Section 4.3). Restricting the use of very common letters (like E, T, and A), as done in Pas-2 (E was not used at all), creates an uncommon letters combinations which have very small probability of occurrence.

8.3. SUGGESTED IMPROVEMENTS IN THE PRESENT SYSTEM

In order to ensure system reliability, following improvements are suggested.

1. In the experiments only the upper case letters 'A', 'B', ... 'Z' and the blank were used for system testing. Extensions to the recognition of all letters (both upper and lower case) could be performed. Also efforts could be made to include other punctuation symbols and delimiters like colon, comma, hyphen e.t.c., apart from blank.
2. Preliminary recognition rates over hand-printed data are comparatively low. They could possibly be improved by introducing better preprocessing and size-normalizing strategies. For example some better hole-detecting algorithm could be searched.
3. Characters in Munson's data were tilted towards one side or the other. Some rotating procedure could be introduced. This procedure would then reduce the structural variety and let the character follow a standard pattern. The features thus extracted will carry more information, and hence better results from SCRS are expected. Further the inter-class and intra-class distribution of the black points in the characters is not homogeneous. Some strategies could be searched in order to bring such homogeneity. For example a superimposed pattern for each character class could be constructed and

the distribution of black points among the characters pertain to one class could be adjusted.

4. Since preprocessing algorithms are usually data dependent, therefore, we expect that the algorithm developed in this study may not work well on all type-fonts. For example, when the character is too thin or represented in skeleton form, certain modifications may probably have to make. A suggested modification is to change the order of thinning and filling steps in smoothing schemes.
5. A heuristic pruning method was applied in reducing the number of alternatives in IBDA. The amount of computations of the algorithm still increases exponentially (though with much lower order). Probably would be better if, instead of heuristic approach, some function of low dimensionality (linear or quadratic) be investigated for searching the most probable sequence.
6. As explained in section 7.3, the preliminary classification step in DLA should be included only if the performance of SCRS is very high (over 90% correct).

8.4 SUGGESTIONS FOR FURTHER RESEARCH IN THIS AREA

There are several possible directions for future research in the area we studied. At present, we thought of the following:

1. English language is assumed to be a Markov source of order $r = 1$. Some other assumptions of dependency, like

considering $r > 1$, could be tested and an optimal assumption should be found.

2. Word-length-and-position independent unigram and bigram were used in the present experiments. Other statistics depending on a number of dependencies could be tested. Further, the statistics computed by other researchers should also be tested. A survey on such statistics is given in [SUE-79].
3. In the experiments only one kind of contextual information was used; some other kinds of informations like word-probabilities could be utilized while deciding on a sequence among several alternatives.
4. The proposed Imbricate Block Decoding algorithm (IBDA) should be investigated further, to make it more practical, for example emphasis could be given to improve the performance of predecoding scan.
5. System is tested on English language only. But since we are not considering its phonemic attributes, one could investigate whether the system could be adopted for other natural languages, using their contextual information.
6. If the segmentation of the characters be made, the proposed system could easily be adopted for the recognition of cursive scripts as well.
7. In all the experiments 'depth of search' value was fixed; could it be variable, some should investigate that.
8. The output text could be used as input to speech synthesizer which can produce equivalent phonemes and a

voiced-output could be produced.

APPENDIX - A

PASSAGE - 1

IN OLDEN TIMES LONG LONG BEFORE THE COMING OF CHRIST THERE REIGNED OVER A CERTAIN COUNTRY A GREAT KING CALLED CROESUS HE HAD MUCH GOLD AND SILVER AND MANY PRECIOUS STONES AS WELL AS NUMBERLESS SOLDIERS AND SLAVES INDEED HE THOUGHT THAT IN ALL THE WORLD THERE COULD BE NO HAPPIER MAN THAN HIMSELF BUT ONE DAY THERE CHANCED TO VISIT THE COUNTRY WHICH CROESUS RULED A GREEK PHILOSOPHER NAMED SOLON FAR AND WIDE WAS SOLON FAMED AS A WISE MAN AND A JUST AND INASMUCH AS HIS FAME HAD REACHED CROESUS ALSO THE KING COMMANDED THAT HE SHOULD BE CONDUCTED TO HIS PRESENCE SEATED UPON HIS THRONE AND ROBED IN THE MOST GORGEOUS APPAREL CROESUS ASKED OF SOLON HAVE YOU EVER SEEN AUGHT MORE SPLENDID THAN THIS OF A SURETY HAVE I REPLIED SOLON PEACOCKS COCKS AND PHEASANTS GLITTER WITH COLORS SO DIVERSE AND SO BRILLIANT THAT NO ART CAN COMPARE WITH THEM CROESUS WAS SILENT AS HE THOUGHT TO HIMSELF SINCE THIS IS NOT ENOUGH I MUST SHOW HIM SOMETHING MORE TO SURPRISE HIM SO HE EXHIBITED THE WHOLE OF HIS RICHES BEFORE SOLONS EYES AS WELL AS BOASTED OF THE NUMBER OF FOES HE HAD SLAIN AND THE NUMBER OF TERRITORIES HE HAD CONQUERED THEN HE SAID TO THE PHILOSOPHER YOU HAVE LIVED LONG IN THE WORLD AND HAVE VISITED MANY COUNTRIES TELL ME WHOM YOU CONSIDER TO BE THE HAPPIEST MAN LIVING THE HAPPIEST MAN LIVING I CONSIDER TO BE A CERTAIN POOR MAN WHO LIVES IN ATHENS REPLIED SOLON

PASSAGE - 2

A CHILDS BRAIN STARTS FUNCTIONING AT BIRTH AND HAS AMONGST ITS MANY INFANT CONVOLUTIONS THOUSANDS OF DORMANT ATOMS INTO WHICH GOD HAS PUT A MYSTIC POSSIBILITY FOR NOTICING AN ADULTS ACT AND FIGURING OUT ITS PURPORT UP TO ABOUT ITS PRIMARY SCHOOL DAYS A CHILD THINKS NATURALLY ONLY OF PLAY BUT MANY A FORM OF PLAY CONTAINS DISCIPLINARY FACTORS YOU CANT DO THIS OR THAT PUTS YOU OUT SHOWS A CHILD THAT IT MUST THINK PRACTICALLY OR FAIL NOW IF THROUGHOUT CHILDHOOD A BRAIN HAS NO OPPOSITION IT IS PLAIN THAT IT WILL ATTAIN A POSITION OF STATUS QUO AS WITH OUR ORDINARY ANIMALS. MAN KNOWS NOT WHY A COW DOG OR LION WAS NOT BORN WITH A BRAIN ON A PAR WITH OURS WHY SUCH ANIMALS CANNOT ADD SUBTRACT OR OBTAIN FROM BOOKS AND SCHOOLING THAT PARAMOUNT POSITION WHICH MAN HOLDS TODAY BUT A HUMAN BRAIN IS NOT IN THAT CLASS CONSTANTLY THROBBING AND PULSATING IT RAPIDLY FORMS OPINIONS ATTAINING AN ABILITY OF ITS OWN A FACT WHICH IS STARTLINGLY SHOWN BY AN OCCASIONAL CHILD PRODIGY IN MUSIC OR SCHOOL WORK AND AS WITH OUR DUMB ANIMALS A CHILDS INABILITY CONVINCINGLY TO IMPART ITS THOUGHTS TO US SHOULD NOT CLASS IT AS IGNORANT UPON THIS BASIS I AM GOING TO SHOW YOU HOW A BUNCH OF BRIGHT YOUNG FOLKS DID FIND A CHAMPION A MAN WITH BOYS AND GIRLS OF HIS OWN A MAN OF SO DOMINATING AND HAPPY INDIVIDUALITY THAT YOUTH IS DRAWN TO HIM AS IS A FLY TO A SUGAR BOWL

PASSAGE - 3

IT IS ONE OF THE LEAST HOSPITABLE PLACES ON EARTH A STEADY WIND MOANS OVER THE CROCODILES BASKING ON THE BANKS OF BLUEGREEN LAKE TURKANA AND FLATTENS THE KNEEHIGH BEACH GRASS WHERE THE LONGHORNED ORYX GRAZE BEYOND STRETCHES THE DESERT OF NORTH EAST KENYA BAKED BY THE AFRICAN SUN IN A WADI OR DRIED UP STREAM BED NOT FAR AWAY A SANDY HAired MAN MOVES SLOWLY HIS LOOSE SHORTS AND SHIRT FLAPPING IN THE BREEZE HIS HEAD BARE TO THE SUN HIS EYES SEARCHING THE ARID SOIL AT HIS FEET SOME FIFTY FEET AWAY SANDALS SCUFFING DUST INTO THE AIR BEHIND HER HIS WIFE KEEPS PACE HER EYES SWEEPING THE GROUND AN AFRICAN FLANKING THE LEADER ON HIS RIGHT AND HALF A DOZEN OTHERS MAKE UP THE REMAINDER OF THE PARTY. SUDDENLY THE LEADER STOPS STOOPS AND SNATCHES A SMALL BROWNISH FOSSILIZED BONE FRAGMENT OUT OF THE SAND NIMEIPATA HE SAYS IN SWAHILI TO THE MAN BESIDE HIM I HAVE GOT IT THEN MEAVE HE CALLS TO THE WOMAN WHO RUNS TO JOIN HIM TOGETHER THEY EXAMINE THE BONE FOR A MOMENT REPLACE IT ON THE EXACT SPOT WHERE IT WAS FOUND MARK IT WITH A STAKE AND RESUME THEIR SEARCH THE INTENT MAN IN THE DESERT IS RICHARD ERSKINE LEAKEY HEIR TO ONE OF THE GREATEST SURNAMES IN ANTHROPOLOGY AND AT THIRTY TWO A FORMIDABLE SCIENTIST IN HIS OWN RIGHT HE AND HIS DUSTY BAND ARE LOOKING ALMOST LITERALLY FOR FOOTPRINTS IN THE SANDS OF TIME FOR CLUES TO THE MYSTERY OF MANS ORIGINS

APPENDIX - B

(List of original and the words misrecognized)

(by using MVA and IBDA)

T A B L E - A

List of original and the words misrecognized by MVA &
IBDA operating on passages written by subject 6
(for b = 5 and d = 4)

Pas-1

Original word	Word misrecognized as	Original word	Word misrecognized as
ATHENS	ATHINS	AUGHT	MUGHT
BEFORE	BIFORE	CERTAIN	CIRTAIN
COCKS	COCKE	COMMANDED	COMMANDID
CONDUCTED	CONDUCTID	CONSIDER	CONSIWER
CROESUS	CROISUS	DIVERSE	DIVERSI
ENOUGH	INOUGH	EYES	EVIS
FAMED	FAMID	GREAT	GRIAT
HAD*	HAP	INDEED	INDIED
JUST	TUST	LIVES	LIVIS
NUMBERLESS	NUMBIRESS	PHEASANTS	PHIASANTE
PHILOSOPHER	PHILOSOPHIR	PRESENCE	PRISINCE
REACHED	RIACHED	REIGNED	RIIGNID
SEATED	SEATID	SLAVES	ELAVES
SOLDIERS+	SOLVIERS	SURETY	SURITY
THERE	THIRE	WELL	WILL
WIDE	WIVE	WISE	WISI

+ This word was misrecognized by using MVA only.

* This word was misrecognized by using IBDA only.

T A B L E - A (CONTD)

List of original and the words misrecognized by MVA &
IBDA operating on passages written by subject 6
(for b = 5 and d = 4)

Pas-2

Original word	Word misre- cognized as	Original word	Word misre- cognized as
ADULTS	MOULTS	ANIMALS	ANIMALE
BOOKS	BOOKE	BRAIN	BRMIN
CHILDHOOD*	CHILOHOOD	DAYS	PAYS
DID	PID	DISCIPLINARY	PISCIPLINARY
DO	OO	DOG	OOG
DOMINATING	POMINATING	DORMANT	OORMANT
DUMB	PUMS	FACTORS	FACTORE
FORMS	FORME	GOD	GOP
HOLDS	HOLDE	ITS	ITE
KNOWS	KNOWE	PUTS	PUTE
RAPIDLY+	RAPIPLY	RAPIDLY*	RAPIVLY
STARTS	STARTE	THOUGHTS	THOUGHTE
TODAY	TOPAY		

+ This word was misrecognized by using MVA only.

* This word was misrecognized by using IBDA only.

T A B L E - A (CONTD)

List of original and the words misrecognized by MVA &
 IBDA operating on passages written by subject 6
 (for b = 5 and d = 4)

Pas-3

Original word	Word misre- cognized as	Original word	Word misre- cognized as
BANKS	BANKE	BESIDE	BESIVE
BLUE	BLUI	DESERT	DISERT
DUSTY	OUSTY	EAST	IAST
EXAMINE	IXAMINE	EYES	EYIS
EYES	IKIS	FEET	FEIT
FORMIDABLE	FORMIDABLI	HEAD*	HEAP
HIER	HITR	INTENT	INTINT
KEEPS	KEEPE	KENYA	KENYM
LEADER	LIADER	LEADER	LEADIR
LEAKEY*	LEAKIY	LEAST	LIAST
LITERALLY	LITIRALLY	LOOSE	LOOSI
MYSTERY	MYSTIRY	PLACES	PLACIS
REMAINDER	REMAINDIR	SEARCH	SIARCH
SEARCHING	SIARCHING	SHORTS	SHORTE
SLOWLY	ELOWLY	SNATCHES	INATCHES
STEADY	STEAPY	STREAM	STRIMM
SUDDENLY	SUPPINLY	THEN	THIN
TOGETHER	TOGITHER	WIFE	WIFI
WHERE	WHIRE		

* This word was misrecognized by using IBDA only.

T A B L E - B

List of original and the words misrecognized by MVA &
IBDA operating on passages written by OCR 'B'
(for b = 5 and d = 4)

Pas-1

Original word	Word misre- cognized as	Original word	Word misre- cognized as
APPAREL	APPABEL	ART	ABT
BEFORE	BEFOBE	BRILLIANT	BBILLIANT
CHRIST	CHBIST	COLORS	COLOBS
COMPARE	COMPABE	CONQUERED	CONQUEBED
CROESUS	CBOESUS	COUNTRIES	COUNTHIES
COUNTRY	COUNTBY	DIVERSE*	DIVEBSE
FAR	FAB	GORGEOUS*	GOBGEOUS
GREAT	OBEAT	GREEK	OBEEK
MORE	MOBE	NUMBERLESS	NUMBEBLESS
PRECIOUS	PPECIOUS	PRESENCE	PPESENCE
REACHED	BEACHED	REIGNED	BEIONED
REPLIED	BEPLIED	RICHES	BICHES
ROBED	BOBED	RULED	BULED
SURETY	SUBETY	SURPRISE	SUPPPISE
TERRITORIES	TEBBITOBIES	THERE	THEBE
THRONE	THBONE	WORLD	WOBLD

* This word was misrecognized by using IBDA only.

T A B L E - B (CONTD)

List of original and the words misrecognized by MVA &
IBDA operating on passages written by OCR 'B'
(for b = 5 and d = 4)

Pas-2

Original word	Word misrecognized as	Original word	Word misrecognized as
BIRTH	BIBTH	BORN	BOBN
BRAIN	BBAIN	BRIGHT	BBIGHT
DISCIPLINARY	DISCIPLINABY	DORMANT	DOBMANT
DRAWN	DBAWN	FACTORS	FACTOBS
FIGURING	FIOUBING	FOR*	FOB
FORM	FOBM	FORMS	FOBMS
FROM	FBOM	GIRLS	GIBLS
IGNORANT	IONOBANT	IMPART	IMPABT
NATURALLY	NATUBALLY	OR*	OB
ORDINARY	OBDINABY	OUR	OUB
PAR	PAB	PARAMOUNT	PABAMOUNT
PRACTICALLY	PPACTICALLY	PRIMARY	PPIMABY
PRODIGY	PPODIGY	PURPORT	PUPPOBT
RAPIDLY	BAPIDLY	STARTS	STABTS
STARTLINGLY	STABTLINGLY	SUBTRACT	SUBTHACT
SUGAR	SUGAB	THROBBING	THBOBBING
THROUGHOUT	THBOUGHOUT	WORK*	WOBK

* This word was misrecognized by using IBDA only.

T A B L E - B (CONTD)

List of original and the words misrecognized by MVA &
 IBDA operating on passages written by OCR 'B'
 (for b = 5 and d = 4)

Pas-3

Original word	Word misrecognized as	Original word	Word misrecognized as
AIR	AIB	ANTHROPOLOGY	ANTHBPOLOGY
ARE	ABE	ARID	ABID
BARE	BABE	BREEZE	BBEEZE
BROWNISH	BBOWNISH	CROCODILES	CBOCODILES
DESERT*	DESEBT	DRIED	DBIED
EARTH	EABTH	FAR	FAB
FOR*	FOB	FOOTPRINTS	FOOTPPINTS
FRAGMENT	FBAGMENT	FORMIDABLE	FOBMIDABLE
GRASS	OBASS	GRAZE	OBAZE
GREATEST	OBEATEST	GREEN+	OBEEN
GREEN*	GBEEN	GROUND	OBOUND
HAIRD	HAIBED	HEIR	HEIB
LITERALLY	LITEBALLY	LONGHORNED	LONGHOBNE
MARK	MABK	MYSTERY	MYSTEBY
NORTH	NOBTH	OR*	OB
ORIGINS	OBIGINS		

+ This word was misrecognized by using MVA only.

* This word was misrecognized by using IBDA only.

T A B L E - B (CONTD)

List of original and the words misrecognized by MVA &
IBDA operating on passages written by OCR 'B'
(for b = 5 and d = 4)

Pas-3 (contd)

Original word	Word misrecognized as	Original word	Word misrecognized as
ORYX	OBYX	OTHERS*	OTHEBS
PARTY	PABTY	REMAINDER	BEMAINDER
REPLACE	BEPLACE	RESUME	BESUME
RICHARD	BICHABD	RIGHT	BIGHT
RUNS	BUNS	SEARCH	SEABCH
SEARCHING	SEABCHING	SHIRT	SHIBT
SHORTS	SHOBTS	STREAM	STHEAM
STRETCHES	STHETCHES	SURNAMES	SUBNAMES
THEIR	THEIB	THIRTY	THIBTY
TURKANA	TUBKANA	WHERE	WHEBE

* This word was misrecognized by using IBDA only.

BIBLIOGRAPHY

BIBLIOGRAPHY

- ABE-71 Abe, K; Fukumura, T: "Word-searching methods in character recognition system with dictionary," Sys., Comp. Contr., vol. 5, No. 2, 1971, pp. 1-9.
- ABN-68 Abend, K.: "Compound decision procedures for unknown distributions and for dependent states of nature, in Pattern Recognition," L. N. Kanal Ed., Thompson, Washington, D. C., 1968, pp. 204-249.
- ALB-67 Albegra, C. N.: "String similarity and misspellings," Comm. of ACM, vol. 10, May 1967, pp. 302-313.
- ALT-62 Alt, F. C.: "Digital pattern recognition by moments," in Optical Character Recognition, G. L. Fischer et al. Eds., Washington D. C.: Spartan Books, 1962, pp. 153-179.
- BAR-60 Baran, P.; Estrin, G.: "An adaptive character reader," IRE WESCON Convention Record, 4, 1960, pp. 29-41.
- BED-71 Beddoes, M. P.; Suen, C. Y.: "Evaluation and a method of presentation of the sound output from the Lexiphone - a reading machine for the blind," IEEE Trans. Bio-Med. Engng., vol. 18, March 1971, pp. 85-91.
- BLE-59 Bledsoe, W. W.; Browning, I.: "Pattern recognition and reading by machine," Proc. Eastern Jt. Comp. Conf. New York: Nat. Jt. Comp. Committee, Dec. 16, 1959, pp. 225-232.
- BRA-64 Braverman, D.: "Theories of Pattern Recognition," in Advances in Communications Systems, vol. 1, A. V. Balkrishnan Ed., Academic Press, New York, 1964.
- CAR-66 Carlson, G.: "Techniques for replacing characters that are garbled on input," Spring Jt. Comp. Conf., AFIPS Conf. Proc., vol. 28, Washington D. C.: Spartan, 1966, pp. 189-192.
- CHU-75 Chung, S. S.: "Using contextual constraints from the English language to improve the performance of character recognition machines," M.Sc. thesis, School of Comp. Sc. McGill Univ. 1975.

- CLE-68 Clemens, J. K.; Mason, S. J.: "Character Recognition in an Experimental reading machine for the blind," in Recognizing Patterns, P. A. Koller & M. Eden Eds., M.I.T. Press, 1968, pp. 156-167.
- COO-64 Cooper, D. B.; Cooper, P. W.: "Nonsupervised adaptive Signal Pattern Recognition," inform. Contr., vol. 7, 1964, pp. 416-444.
- COR-69 Cornew, R. W.: "A statistical method of spelling correction," Inform. Contr. 12, 1969, pp. 79-93.
- DEU-72 Deutsch, E. S.: "Thinning algorithms on rectangular, hexagonal, and triangular array," Comm. of ACM, vol. 5, No. 9, Sept. 1972, pp. 827-837.
- DIM-57 Dimond, T. L.: "Devices for reading hand-written characters," Proc. Eastern Jt. Comp. Conf., Dec. 1957, pp. 232-237.
- DIN-55 Dinneen, G. P.: "Programming Pattern Recognition," Proc. Western Jt. Comp. Conf., March 1955, pp. 94-100.
- DOY-60 Doyle, W.: "Recognition of Sloppy hand-printed Characters," Proc. Western Jt. Comp. Conf., vol. 17, May 1960, pp. 133-142.
- DUD-68 Duda, R. O.; Hart, P. E.: "Experiments in the recognition of handprinted text: Part II -- context analysis," Fall Jt. Comp. Conf. AFIPS Conf. Proc. vol. 33, Washington D. C., 1968, pp. 1139-1149.
- DUD-73 Duda, R. O.; Hart, P. E.: "Pattern Classification and Scene Analysis," Wiley, New York, 1973.
- DUF-73 Duff, M. J. B.; Watson, D. M.; Fountain, T. J.; Shaw, G. K.: "A cellular logic array for image processing," Pat.-Recog., vol. 5, 1973, pp. 229-274.

- EDW-64 Edwards, A. W.; Chambers, R. L.: "Can a priori probabilities help in character recognition?" JACM 11, 1964, pp. 465-470.
- FIS-62 Fischer, G. L. Jr.; Pollock, D. K.; Radack, B.; Stenvens, M. E.: Eds. Optical Character Recognition, Washington D. C.: Spartan Books, 1962.
- FOR-73 Forney, G. D.: "The Viterbi algorithm," Proc. IEEE vol. 61, No. 3, March 1973, pp. 268-278.
- FRE-62 Freyer, W. D.; Richmond G. E.: "Two dimensional spatial filtering and computers," Proc. Nat. Electronics Conf. 18, 1962, p. 529.
- FU1-68 Fu, K. S.: "Sequential Methods in Pattern Recognition and Machine Learning," Academic Press, New York, 1968.
- FU2-74 Fu, K. S.: "Syntactic Methods in Pattern Recognition," Academic Press, New York, 1974.
- GAB-71 Gabor, D.: "Character recognition by holography," Nature, 208, No. 5009, Oct. 1965, pp. 422-423; see also U. K. Pat. 1143086, Feb. 1969; U. S. Pat. 3600054, Aug. 1971.
- GL1-67 Glucksman, H. A.: "Classification of Mixed-font Alphabets by Characteristic Loci," Digest of Ist. Ann. IEEE conf., sept. 1967, pp. 138-141.
- GL2-71 Glucksman, H. A.: "Multicategory Classification of Pattern Represented by High-order Vectors of Multilevel Measurements," IEEE Trans. on Comp., vol. 20, Dec. 1971, pp. 1593-1598.
- GOL-59 Gold, B.: "Machine Recognition of hand-sent Morse code," IRE Trans. on Inform. Theory, vol. IT-5, No. 1, March 1959, pp. 17-24.
- GRE-63 Greanias, E. C.; Meagher, P. F.; Norman, R. J.; Essinger, P.: "The recognition of handwritten numerals by contour analysis," IBM Jr., of Res. and Dev., vol. 7, No. 1, Jan. 1963 pp. 14-22.

- GRE-70 Grenander, U.: "A unified approach to pattern analysis," in Advances in Computers, W. Freiberger, Guest Ed., vol. 10, New York: Academic Press, 1970, pp. 175-216.
- HAN-76 Hanson, A. R.; Riseman, E. M.; Fisher, E.: "Context in word recognition," Pat. Rocog. vol. 8, 1976, pp. 35-45.
- HAR-62 Harmon, L. D.: "Automatic reading of cursive script," in Optical Character Recognition Eds. Fischer, Pollock, Radack, and Stevens Eds., Spartan Books, Washington, D. C., 1962, pp. 151-152.
- HOL-68 Holt, A. W.: "Comparative religion in character recognition machines," Computer Group News, vol. 2, No. 6, 1968, pp. 3-11.
- HOS-68 Hosking, K. H.; Thompson, H., Conf. on Pat. Recog., Nat. Phy. Lab., July 1968, London:IEE.
- HOS-72 Hosking, K. H.: "A contour method for the recognition of handprinted characters," Proc. Conf. on Pat. Recog., Nat. Phy. Lab., in Machine Perception of Patterns and Pictures, The Inst. Of Phy. Conf. Series, No. 13, Walker, P. A. Ed., London, Apr. 1972, pp. 19-27.
- HUS-72 Hussain, A. B. S.; Toussaint, G. T.; Donaldson, R. W.: "Results obtained using a simple character recognition procedure on Munson's handprinted data," IEEE Trans. on Comp. vol. C-21, 1972, pp. 201-205.
- JON-62 Jones, C. M.; Kellacher, J. F.; Dillard, J. M.; Bernstein, S. L.: "Fourier transform methods in pattern recognition," CR Research Note AD 285794, July 1962, Armed Services Technical Information Agency, Arlington Hall, Virginia, U. S. A.
- KAM-63 Kamensky, L. A.; Liu, C. N.: "Computer-automated design of multifont print recognition logic," IBM Jr. of Res. and Dev., 7, No. 1, Jan. 1963, pp. 2-13.

- KAN-74 Kanal, L.: "Pattern Recognition," IEEE Trans. on Inform. Theory, vol. IT-20, Nov. 1974, pp. 1968-1974.
- KAS-76 Kashyap, R. L.; Mittal, M. C.: "Word recognition in a multi-talker environment using syntactic methods," Proc. 3rd. Int. Jt. Conf. on Pat. Recog., Cornado, 1976, pp. 626-631.
- KNK-67 Kpoke, P. J.; Richard, G. W.: "A Linguistic Approach to Mechanical Pattern Recognition," IEEE Digest of First Ann. Comp. Conf., Sept. 1967, pp. 142-144.
- KNL-69 Knoll, A. L.: "Experiments with 'characteristic loci' for recognition of handprinted characters," IEEE Trans. on Comp. (Short Notes), vol. C-18, No. 4, Apr. 1969, pp. 366-372.
- KOF-35 Koffka, K.: "Principles of Gestalt Psychology," Harcourt Brace, New York, 1935.
- KUS-77 Kustner, W. G. H.; Kastner, M. H. H. (Eds.): "The VNR Concise Encyclopedia of Mathematics", VNR Co., New York, 1977.
- LUG-64 Lugt, A. V.: "Signal Detection by Complex Spatial Filtering," IEEE Trans. Inform. Theory, vol. IT-10, Apr. 1964, pp. 139-145.
- MCO-63 McCormick, B. H.: "The Illinois pattern recognition computer-ILLIAC III", IEEE Trans. Electron. Comp., vol. EC-12, 1963, pp. 791-813.
- MEL-62 McElwain, C. K.; Evens, M. B.: "The degarbler - a program for correcting machine read morse code," Inform. Contr., vol. 5, 1962, pp. 368-384.
- MKE-41 McKeon, R., Ed., "The Basic Works of Aristotle," Random House, New York, 1941, pp. 700ff.
- MIL-57 Miller, G. A.; Friedman, E. A.: "The reconstruction of mutilated English texts," Inform. Contr., vol. 1, 1957, pp. 38-55.

- MUN-68 Munson, J. H.: "Experiments in the recognition of hand-printed text: Part I -- Character Recognition," Fall Jt. Comp. Conf., AFIPS Conf., Proc., vol. 33, Washington, D.C.: Thompson, Dec. 1968, pp. 1125-1138.
- MUR-62 Muroga, S.; Toda, I.; Kondo, M.: "Majority Decision Functions of up to Six Variables," J. Math. Comp., Oct. 1962; also in Japanese in 1959 and 1960.
- NAG-70 Nagy, G.; Toung, N.: "Normalization Techniques for Hand-Printed numerals," Comm. of ACM, vol. 13, No. 8, Aug. 1970, pp. 475-481.
- NAR-62 Narasimhan, R.: "A Linguistic Approach to Pattern Recognition," Digital Comp. Lab. Report 121, Univ. of Ill., Urbana, Ill., 1962.
- NAR-66 Narasimhan, R.: "Syntax directed interpretation of classes of pictures," Comm. of ACM, vol. 9, No. 3, March 1966, pp. 166-173.
- NEU-75 Neuhoff, D. L.: "The Viterbi algorithm as an aid in text recognition," IEEE Trans. Inform. Theory vol. 21, 1975, pp. 222-226.
- NUG-67 Nugent, W. R.: "The on-line Recognition of Cursive writing using Geometric-Topological Invariants of Stroke Succession," IEEE Digest of the First Ann. Comp. Conf. Sept. 1967, pp. 145-148.
- NUM-56 Numann, J. V.: "Probabilistic Logics and the Synthesis of Reliable Organism from Unreliable Components," Automata Studies, Princeton Univ. Press, Princeton, N.J., 1956, pp. 43-97.
- OCA-77 OCR 'A': "Character Set and Print Quality for Optical Character-Recognition (OCR-A), ANSI, 1977.
- OCB-71 OCR 'B': "The Alphanumeric Character Set for OCR-B for Optical Recognition, European Comp. Manf. Assoc., 1971.

- PAR-68 Parks, J. R.; Elliott; Cowin, G.: "Simulation of an alphanumeric character recognition system for unsegmented low quality print," IEE Nat. Phy. Lab., Conf. on Pat. Recog., July 1968.
- PAR-69 Parks, J. R.: "A multi-level system of analysis for mixed font and hand-blocked printed characters recognition," in Automatic Interpretation and Classification of Images, A. Grasselli Ed., Academic Press, New York, 1969, pp. 295-322.
- RAV-67 Raviv, J.: "Decision making on Markov chains applied to the problem of pattern recognition," IEEE Trans. Inform. Theory, vol. IT-13, Oct. 1967, pp. 536-551.
- RIS-71 Riseman, E. M.; Ehrich, R. W.: "Contextual word recognition using binary diagrams," IEEE Trans. on Comp. vol. C-20, No. 4, Apr. 1971, pp. 397-903.
- SCH-76 Schurmann, J.: "Multifont Word Recognition System with Application to Postal Address Reading," Proc., 3rd. Int. Jt. Conf. on Pat. Recog. Coronado, Nov. 76, pp. 658-662.
- SEB-62 Sebestyen, G. S.: "Pattern Recognition by an Adaptive Process of Sample Set Construction," IRE Trans. Inform. Theory, vol. IT-8, No. 5, Sept. 1962, pp. S82-S91.
- SEL-58 Selfridge, O. G.: "Panademonium: A paradigm for learning," Symposium on Mechanization of Thought Process. Nat. Phy. Lab., Middlesex, England, Nov. 1958.
- SHE-59 Sherman, H.: "A Quasi-topological Method for Machine Recognition of Line Patterns," Proc. Int. Conf. on Inform. Proces., ICIP, UNESCO Conf. in Paris, France, 1959, pp. 232-238.
- SH1-77 Shinghal, R.: "Using contextual information to improve performance of character recognition machines," Ph.D. thesis, School of Computer Science, McGill Univ., 1977.

- SH2-78 Shinghal, R.; Rosenberg, D.; Toussaint, G. T.: "A simplified heuristic version of a recursive Bayes algorithm for using context in text recognition," IEEE Trans. on Sys., Man and Cybern., vol. SMC-8, No. 5, May 1978, pp. 412-414.
- SH3-79 Shinghal, R.; Toussaint, G. T.: "A bottom-up and top-down approach to using context in text recognition," Int. J. of Man mach. Studies, vol. 11, No. 2, March 1979, pp. 201-212.
- SH4-79 Shinghal, R.; Toussaint, G. T.: "Experiments in text recognition with the modified Viterbi algorithm," IEEE Trans. on Pat. Analysis and Mach. Intl., vol. PAMI-1, No. 2, Apr. 1979, pp. 184-193.
- SKL-73 Sklansky, J. (Ed.): "Pattern Recognition: Introduction and Foundations," D. H. & R. Inc., Pennsylvania, 1973.
- SME-72 Smith, E. A.; Philips, D. R.: "Automated Cloud tracking using precisely aligned digital ATC pictures," IEEE Trans. Comp., vol. C-21, 1972, pp. 715-729.
- STE-71 Stefanelli, R. Rosenfeld A.: "Some parallel thinning algorithms for digital pictures," J. of ACM, vol. 18, No. 2, 1971, pp. 255-264.
- SUE-78 Suen, C. Y.: "Advances in Optical Character Recognition," Proc. Cand. Comp. Conf., May 1978, pp. 263-268.
- SUE-79 Suen, C. Y.: "N-gram statistics for natural language understanding and text processing," IEEE Trans. on Pat. Analysis and Mach. Intl. vol. PAMI-1, No. 2, Apr. 1979, pp. 164-172.
- T01-72 Toussaint, G. T.; Donaldson, R. W.: "Some simple contextual decoding algorithms applied to recognition of handprinted text," Proc. Ann. Cand. Comp. Conf., Session 1972, Montreal, Canada, 1-3 June, 1972, pp. 422101-422115.

- T02-74 Toussaint, G. T.; Chung, S.: "On some algorithms for using context in machine recognition of handprinted text," Second Ann. Comp. Sc. Conf., Detroit, MI, (abs.), Feb. 12-14, 1974, p. 62.
- T03-74 Toussaint, G. T.: "Bibliography on Estimation of Misclassification," IEEE Trans. Inform. Theory, July 1974, pp. 472-479.
- T04-77 Toussaint, G. T.: "Recent Progress in statistical Methods applied to Pattern Recognition," Second Jt. Comp. Conf. on Pat. Recog., Copenhagen, Aug. 1974, pp. 478-489.
- T05-77 Toussaint, G. T.: "The use of context in pattern recognition," Proc. IEEE Comp. Soc. Conf. on Pat. Recog. Image Proc., Troy, New York, 6-8 June, 1977, pp. 1-10, also in Pat. Recog. vol. 10, 1978, pp. 189-204.
- T06-78 Toussaint, G. T.; Shinghal, R.: "Table of Probabilities of Occurrence of Characters, Character-Pairs, and Character-triplets in English Text," Tech. Report No. SOCS 78.6, School of Comp. Sc. McGill Univ, Aug. 1978.
- TRI-70 Triendl, E. E.: "Skeletonization of Noisy Hand-drawn symbols using parallel operations," Pat. Recog. vol. 2, no. 3, Sept. 1970, pp. 215-226.
- ULL-72 Ullmann, J. R.: "Correspondence in Character Recognition," Proc. Nat. Phy. Lab., Middlesex, Apr. 1972, in Machine Perception of Patterns and Pictures, P. A. Walker, Ed., London, pp. 34-44.
- ULL-73 Ullmann, J.R.: "Pattern Recognition Techniques," Crane, Russak & Co., New York, 1973.
- UNG-58 Unger, S. H.: "A Computer oriented toward spatial problems," Proc. IRE, vol. 46, 1958, pp. 1744-1750.
- UNG-59 Unger, S. H.: "Pattern Detection and Recognition," Proc. IRE, vol. 47, Oct. 1959, pp. 1737-1752.

- VIT-67 Viterbi, A. J.: "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," IEEE Trans. Inform. Theory, vol. IT-13, Apr. 1967, pp. 260-269.
- VOS-64 Vossler, C. M.; Branston, N. M.: "The use of context for correcting garbled English text," Proc. ACM, 19th Nat. Conf., Aug. 1964, pp. D2.4-1 to D2.4-13.
- WAG-74 Wagner, R. A.; Fischer, M. J.: "The string-to-string correction problem," JACM, vol. 21, 1974, pp. 168-173.
- WEB-75 Webster, N.: "The International WEBSTER New Encyclopedia DICTIONARY of the English Language and Library of Useful Knowledge," The English Language Institute of America, Inc., Chicago, 1975.
- WHE-70 Wheeler, D. D.: "Processes in Word Recognition," in Cognitive Psychology, vol. 1, 1970, pp. 59-85.
- WID-73 Widrow, B.: "The 'rubber mask' technique," Pattern Recognition, vol. 5, 1973, pp. 175-211.
- WRI-39 Wright, E. V.: "Gadsby", Los Angeles: Wetzel Publishing Co., 1939.