A Computational Theory of Contextual Knowledge in Machine Reading

Stephen James Hanlon

Division of Artificial Intelligence
School of Computer Studies
University of Leeds

September 1994

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others.

ABSTRACT

Machine recognition of off–line handwriting can be achieved by either recognising words as individual symbols (word level recognition) or by segmenting a word into parts, usually letters, and classifying those parts (letter level recognition). Whichever method is used, current handwriting recognition systems cannot overcome the inherent ambiguity in writing without recourse to contextual information.

This thesis presents a set of experiments that use Hidden Markov Models of language to resolve ambiguity in the classification process. It goes on to describe an algorithm designed to recognise a document written by a single–author and to improve recognition by adapting to the writing style and learning new words. Learning and adaptation is achieved by reading the document over several iterations. The algorithm is designed to incorporate contextual processing, adaptation to modify the shape of known words and learning of new words within a constrained dictionary.

Adaptation occurs when a word that has previously been trained in the classifier is recognised at either the word or letter level and the word image is used to modify the classifier. Learning occurs when a new word that has not been in the training set is recognised at the letter level and is subsequently added to the classifier.

Words and letters are recognised using a nearest neighbour classifier and used features based on the two–dimensional Fourier transform. By incorporating a measure of confidence based on the distribution of training points around an exemplar, adaptation and learning is constrained to only occur when a word is confidently classified.

The algorithm was implemented and tested with a dictionary of 1000 words. Results show that adaptation of the letter classifier improved recognition on average by 3.9% with only 1.6% at the whole word level. Two experiments were carried out to evaluate the learning in the system. It was found that learning accounted for little improvement in the classification results and also that learning new words was prone to misclassifications being propagated.

ACKNOWLEDGEMENTS

I should like to thank my supervisor Dr. Roger Boyle for his direction, encouragement and patience during the course of this work. I would also like to thank Maggie, Bea and Roger Boyle for their support and warm friendship over the past four years.

Thanks also to Richard Thomas who supervised my undergraduate project which led directly to this work being done. I would also like to thank Eric Atwell and the CCALAS group for the useful discussion during my time working on this project.

Thanks also to the school computer support team for their technical help and for providing a great working environment; Peter Jowett, Steve Harris, Simon Saunders, Dave Harkess, Jim Jackson, Savio Pirondelli, Mark Conmy and Carlos Fandango.

Thanks to Joe Lee for lending me a PC for most of the time I have pursued this work and thanks also to Roland Cross whose support of cspike has been, in many times, beyond the call of duty.

Thanks to everyone involved with the OSCAR project at the University of Essex, Andy Downton, Graham Leedham and Simon Lucas for providing a stimulating and interesting environment for work. Thanks especially to Graham for all the encouragement and to Andy for giving me the time to complete this work. Thanks also to the HCS posse, especially Graeme Sweeney, Lee Silver and Neel Patankar who were great friends while I was at Essex.

Thanks to Maggie Egan for all the love and support during the good and bad times at Leeds and Essex.

Finally all my thanks to my brother, Peter and my parents for their tremendous love, support and help.

This research was supported by a grant from the Science and Engineering Research Council.

To Mum, Dad and Peter.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

Candidate Set :
The group of possible words or letters for an image. This is generated and reduced through a combination of classifiers and contextual information.

On–line recognition :
Recognition of handwriting achieved by interpreting the dynamic strokes of the writer's pen. Three dimensions of information are recorded for each sample, i.e. the X and Y position of the pen and whether the pen is 'up' or 'down'.

Off–line recognition :
Recognition of handwriting achieved by interpreting a two–dimensional image of handwriting.

PDA :
Personal Digital Assistant. A hand–held computer containing personal information and administrative software. Communication is through either a small keyboard or a digital pen and tablet.

Semantic Context :
Where the 'meaning' of a neighbouring word reduces the number of candidates for the current word. In handwriting recognition systems this can be implemented by calculating the probability distribution of co–occurrences of words. The conditional probability of a word given that words close by can then be used to accept or reject a candidate. Such semantic information can also be derived from dictionary entries.

Syntactic Context :
The use of surrounding word–class tags to reduce the number of possible word candidates. Commonly implemented for handwriting recognition systems by modelling the probability distribution of tag bigrams and trigrams and subsequently finding the most probable sequence of tags.

Chapter 1

INTRODUCTION

Machine reading of handwriting is the automatic conversion of handwritten words and symbols into a representation that can be manipulated by machine. In many applications this frequently means converting handwriting into its equivalent ASCII form. In order to achieve this, the handwriting must be entered into the machine; then the relevant information (usually the words or letters) must be located and interpreted.

This process seems trivial for trained humans for whom the written and printed word is still one of the most widely used forms of communication. Consider a traditional form of human transcription of handwritten documents; the process of typing a written document into a word–processor. The human typist reads the handwritten sheet of paper and then enters the information via a keyboard. In this case, no prior knowledge of the document may be needed to carry out the task of transcription, however a knowledge of the language and spelling of words in that language could improve the performance (in terms of both accuracy and speed). The machine reading of handwriting studied in this thesis (and in current systems) does not compare with human transcription, however we look at how it can be improved by using knowledge other than the written words.

Handwriting recognition can be seen as an alternative to speech processing, indeed handwriting recognition systems exploit many techniques also used in speech processing. There are areas, however, where speech recognition is not the most natural (or appropriate) method of entering information to a machine. For example, Cohen (1992) describes a situation where a doctor may not want to dictate patient notes to a computer in front of the patient, in this case writing the notes would be more appropriate. There are also many applications where speech processing cannot be used, for example the application of handwriting recognition could be applied to many forms–processing tasks. Research into applications such as postal address and cheque recognition have attracted much commercial interest; for example, the Centre of Excellence in Document Analysis and Recognition (CEDAR) has attracted large amounts of funding from the U.S. Postal Service (CEDAR 1994).

Machine reading of handwriting is just one area within the larger area of handwriting recognition and analysis. This covers all areas where computers are used to recognise, interpret and analyse handwritten text. Figure 1.1 shows the different domains of handwritten document processing, from Leedham (1994).

Both the areas of recognition and analysis can be split into on–line and off–line recognition, on–line recognition being the recognition of strokes and pen movements recorded by an electronic tablet, while off–line recognition takes an image of the written documents and attempts to interpret the static images of writing.

The application domains for these two processing techniques overlap, however the choice between on–line and off–line recognition comes from the application requirements. Applications where immediate recognition or verification is necessary use on-line recognition techniques, for example, signature verification or portable personal organisers. Off-line techniques are applied to domains where the writer of a document may not be available when the document is being processed or where large sets of documents are being processed, for example, insurance forms–processing or cheque recognition. In these cases, the documents are scanned to generate an image of the writing and then the writing is recognised. However, the difference between on–line and off–line handwriting recognition is already well documented and the reader is directed to Tappert, Suen & Wakahara (1990) for a review the two areas.

In relation to Figure 1.1, the work reported in this thesis comes under the branch of handprinted and cursive script recognition. The data-capture method was to scan images of writing, so the work reported is for off-line recognition.

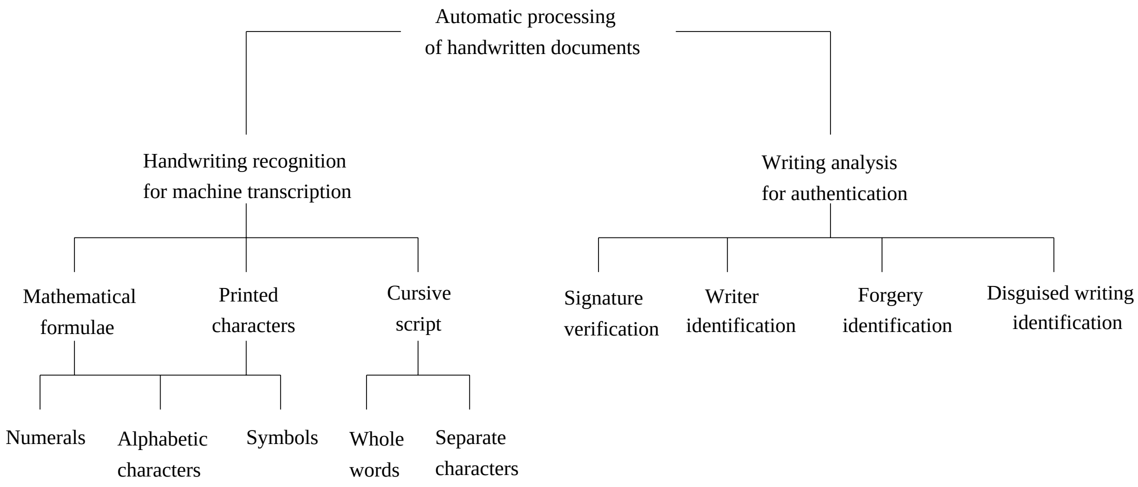


Figure 1.1: Different categories of handwritten document processing, from Leedham (1994).

1.1 Word and letter level recognition

Several different methods have been suggested for recognising handwritten words; these can be divided into whole word approaches and sub–word approaches. Sub–word recognition commonly involves splitting a word into letters, recognising the letters and then recombining the letter results, for example, Bozinovic & Srihari (1989). The power of such systems is their ability to recognise words that have not been seen in any predefined dictionary. However the difficulty arises when cursive words have to be segmented into individual letters since finding the start and end of letters in a handwritten word is difficult. This problem is made worse when the writing quality is poor and letters illegible. Lecolinet & Crettez (1991) divides letter segmentation into *explicit* and *implicit* segmentation, where explicit segmentation is done before any recognition and implicit segmentation is carried out during the recognition stage.

Alternatively words can be recognised as whole symbols avoiding the problem of segmentation. By ignoring the segmentation stage, the problems of badly written characters and merged or overlapping characters can also be avoided. However, reading words as whole symbols means that the system is constrained to a dictionary of valid words. This constraint

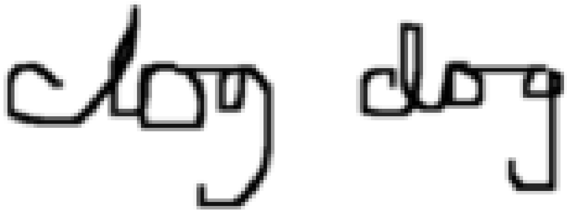


Figure 1.2: Similar words ”clog” and ”dog”

meant that for many years whole word recognition methods were not practical, given disk and memory constraints on computers. Since these constraints have been relaxed over recent years there has been more interest in whole word recognition approaches (for example, Farag (1979) had only a small dictionary of valid words, compared with recent systems such as Bramall & Higgins (1993) with a dictionary of many thousands of words).

1.2 Context in handwriting recognition

In the *Concise Oxford English Dictionary* (1990) ‘context’ is defined as

> the parts of something written or spoken that immediately precede and follow a word or passage and clarify its meaning.

This is similar to how context is used in handwriting recognition. Many handwritten words are visually similar and distinguishing between them from the visual cues can be difficult. Figure 1.2 shows the commonly used example of the words ‘dog’ and ‘clog’. These words are visually similar while semantically they are quite different. It is because of such visual ambiguity that context is used in handwriting recognition. Indeed, without context, human recognition of cursive script has been cited as being about 72% (Edelman, Ullman & Flash 1990).

There are many sources of context used in human recognition. However, for the information to be useful in an computer algorithm, it has to be in a form that can be processed efficiently. This has restricted many of the models to simple statistical techniques avoiding computationally expensive search techniques used in other forms of knowledge representation.

The visual ambiguity in handwriting (for example, Figure 1.2) is one reason for classifiers generating *candidate sets*. In handwriting recognition the candidate set could be either a list of possible letters or a list of possible words, depending upon the classifier. Each item in the lists could be associated with a *ranking* which gives a similarity ordering between the word image and the classifier.

Given a classifier that produces a candidate set, the most simple form of contextual constraint that can be applied is a dictionary look–up. In this case, the candidate words are compared to a lexicon (dictionary) for the problem domain. Words that exist in the dictionary are accepted and words that are not are rejected. This only works for classifiers that read words from their individual letters since a whole word classifier would always return words that were in the domain lexicon[1].

A more complicated form of context can be syntactic or semantic information which has been collated from other sources (such as tagged corpora of language and annotated dictionaries). Such methods of context use surrounding words to restrict the choice of the current word, either by reducing the candidate set, or by re–ordering the set of candidates. These forms of context commonly use a statistical model where a probability distribution of words is used to model the expected distribution of words.

The contextual information is processed by choosing the most probable sequence of words from the sequence of candidate sets. By finding the sequence of words that best fits the statistical model, we say that the classifier "understands" the passage. Figure 1.3 shows the sequence of contextual processing used at Nottingham Trent University, UK for their handwriting classifier (Wells, Evett, Whitby & Whitrow 1992). This sequence of processing represents the general hierarchy of language based context, where syntactic information is derived from word information and semantic information is derived from word and

[1] Unless, of course, it had been trained on words that would not occur within the application domain. However, this seems inappropriate and is not considered further.

syntactic information. A similar sequence of processing stages has also been proposed by Cohen (1992) for interpretation of constrained forms.

1.3 Adaptation in handwriting recognition

Adaptation for handwriting recognition happens when the classifier adapts towards the style used by the writer in order to increase the recognition rates. This has been studied for on-line recognition, in particular by Tappert (1984) and Schomaker, Abbink & Selen (1994).

The adaptation described in this thesis is to compensate for variation in the writing of a single person. For example, the time of day can affect the way someone writes a signature and hence affects the performance of a signature verification system (Fairhurst 1994).

Changing styles of writing for an individual writer are a problem because classification of the written words becomes difficult when the words are written differently. Traditionally, off–line classifiers are trained and then used with no subsequent re-training. This means that the training session must be varied enough to capture all the subtle differences in the possible writing styles, or conversely, the classifier must be able to generalise the training patterns so that changes in the writing style do not affect the classification process.

Schomaker et al. (1994) noted that a word can be written in many different ways and that each time a word is presented to a classifier there would always be something that is visually different. From this starting point, the classifier can be adapted to the style of writing.

Adaptation can be performed within an on–line environment since the medium is interactive and the writer can be asked to confirm any changes to the shape of words before the classifier commits the changes. In off–line recognition, it is difficult to implement this form of supervised learning.

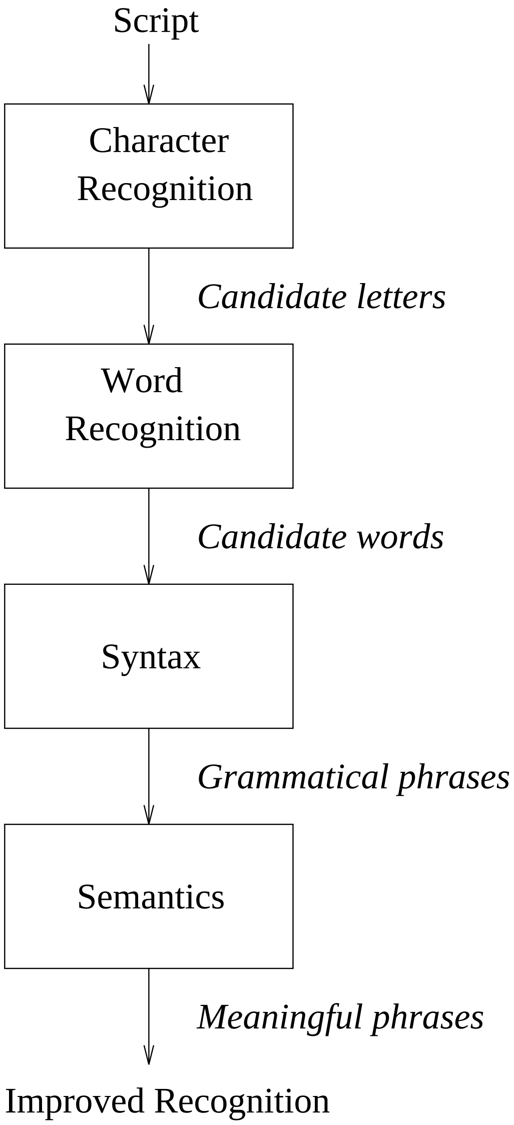


Figure 1.3: Sequence of contextual information for handwriting recognition, from Wells (1992)

1.4 Research Problem

The previous sections have introduced the recognition, use of context and adaptation in handwriting recognition. This thesis considers the problems of off–line handwriting recognition and the use of context and adaptation in order to improve and extend the recognition performance.

The Directed Reading algorithm (DR) is presented and used to control traditional uses of contextual information along with learning and adaptation. This is done while iterating over the document attempting to recognise the writing and adapt the classifier. The algorithm uses both global and analytical methods for reading words; words recognised as whole words are used to modify the letter classifier and words recognised at the letter level modify and extend the whole word classifier.

In order to implement the DR algorithm, a holistic word classifier was developed that used low harmonic Fourier features. This extends the printed text classifier proposed by O'Hair & Kabrinsky (1991) for single–author handwriting recognition.

The original contribution of this work can be summarised as :

- The Directed Reading algorithm, an algorithm that combines word and letter classifiers and controls unsupervised adaptation of off-line handwriting recognition, and
- A 2D Fourier transform classifier for whole word handwriting recognition.

1.5 Thesis Structure

The thesis is divided into eight chapters, including the introduction. Chapter 2 presents a review of off–line handwritten word recognition and Chapter 3 reviews the use of context in handwriting recognition.

Chapter 4 presents a set of experiments evaluating the use of Hidden Markov Models of syntax for resolving ambiguities resulting from the classification process.

Chapter 5 introduces the Directed Reading algorithm. The algorithm controls the process of reading and adaptation while iterating over a document.

In Chapter 6, the method of classification is presented. O'Hair and Kabrinsky's Fourier classifier (O'Hair & Kabrinsky 1991) was used because of it's holistic representation of the word shape. The classifier had been previously used on printed text with good results. The work presented in this chapter tests the classifier on 1000 single–author handprinted words.

Chapter 7 describes the experiments carried out to evaluate the Directed Reading algorithm in terms of it's ability to learn new words and to adapt the shape of known words.

Chapter 8 summarises the work done, draws conclusions, and suggests future research directions following the work reported in this thesis.

Chapter 2

OFF-LINE HANDWRITTEN WORD RECOGNITION

Recognition of handwriting can be traced back to the 1950's (Harmon 1972). Over the past forty years, however, much of the research effort has been on the recognition of isolated handprinted characters. Exceptions to this include Harmon (1962), Ehrich & Koehler (1975), Farag (1979) and Srihari & Bozinovic (1987). Handwritten word recognition falls into two categories, *global* recognition, where global features are extracted from the word image without segmenting the word; and *analytical* recognition, where a word is segmented into *sub–words* (usually letters) which are then classified and recombined to find the original word.

Recently many researchers have used Hidden Markov Models to recognise written words and as the number of workers in the area has increased, it has led to a more thorough evaluation of the problem of word recognition. Lorette (1993) considers the complexities of handwriting recognition, from the type of system needed to the variability of the writing itself. He notes that there are two kinds of information in the written word, the *significans* characterising the writer and the *significant* indicating the meaning (or the identity of the word). Off–line recognition is mainly concerned with recognising the significant, or the identity of the word[1]; however, in on–line recognition there is work in using the style of the writing (the significans) to help the recognition process — this is also proposed in this thesis for off–line recognition.

Lorette then describes three factors that affect the complexity of a handwriting recognition system : the number of writers, type of writing, and the size of vocabulary. Figure 2.1

[1] an exception to this is Cohen (1994) where identity is secondary to interpretation.

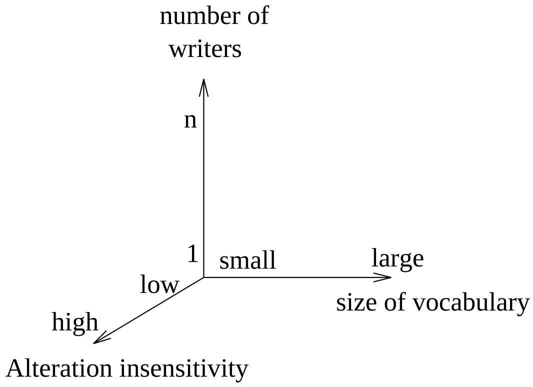


Figure 2.1: Complexity in off-line handwriting recognition. From Lorette (1993).

is taken from Lorette (1993) showing the complexity of different handwriting classifiers. Note that the type of writing is reflected in the diagram as alteration insensitivity. When the writing type is highly constrained, for example written in boxes, the writing style for a single writer does not alter dramatically compared to cursive writing where the same writer may write a word in a different style in different situations. Lorette notes that along with external constraints such as guiding boxes and lines, writing can also change because of *alteration of physical writing conditions* such as using a different pen, and *writer–specific variations* due either to mental state (for example the time of day) or if the writing is spontaneous, fast or slow.

Lorette shows (see Figure 2.2) that the reading process is one of encoding a message onto paper and then decoding that written message in order to read the original message. This model of writing and reading is similar to that used by Kopek & Chou (1994) where documents are recognised using Shannons communication theory (Shannon & Weaver 1964). In this case, the document is treated as a noisy communication channel and the aim of the reader (decoder) is to reconstruct the original message by compensating for the noise introduced during the printing and scanning phases). This model is shown in Figure 2.3 and can be seen to be similar to that of Lorette.

This chapter looks at the recognition of handwriting at both the whole word (global) level and the sub–word (analytical) level, and includes a review of Hidden Markov Model meth-

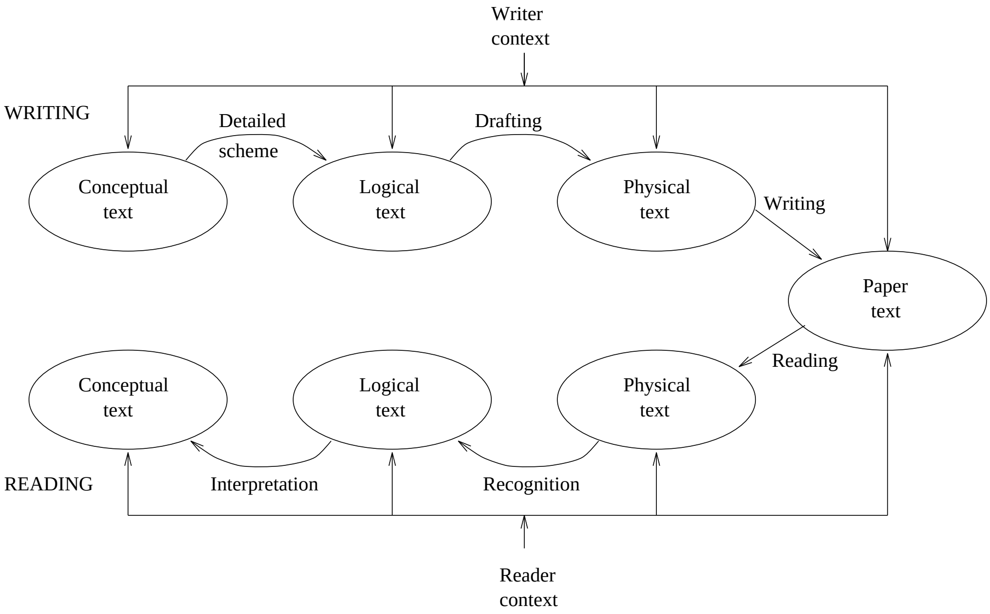


Figure 2.2: Writing and reading of text, from Lorette (1993)

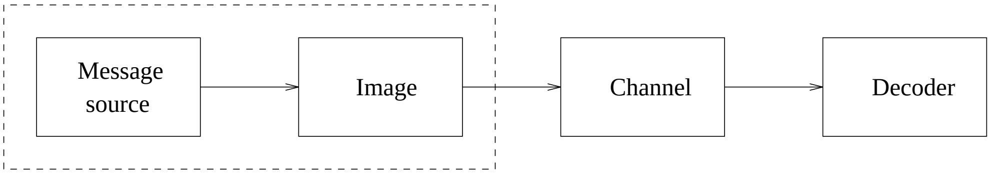


Figure 2.3: Communication theory view of document recognition, from Kopec (1994)

ods.

2.1 Analytical methods of recognition

Analytical recognition of words involves recognising a word in terms of its component parts. Lecolinet & Crettez (1991) note that there are two types of segmentation for analytical recognition :

- Explicit segmentation : where segmentation is carried out prior to recognition. For example, Srihari & Bozinovic (1987)
- Implicit segmentation : where recognition and segmentation processes take place in parallel. For example, Tappert (1982) and Fukushima (1993)

Explicit segmentation of words proves difficult since in cursive handwriting the joining of adjacent letters frequently leads to incomplete letter formations and ambiguous letter start and end points. A segmentation algorithm has to make a decision where to segment the letters which is difficult without prior knowledge of the word identity. One solution to this is to select many candidate segmentation points and delay a decision of the best combination of the segmentation points until after feature extraction (Srihari & Bozinovic 1987) . Implicit segmentation aims to solve this problem by carrying out the segmentation and recognition at the same time. For example, Tappert (1982) tests every combination of possible letter template against the word shape (in this case, an on–line written word). One problem with such methods is the ambiguity caused by letter pairs such as ‘cl’ being similar in shape to ‘d’ (as in Figure 1.2).

Another approach to analytical recognition is to avoid segmenting the word shape into letters and instead segment and classify ‘graphemes’; defined as those parts of a word shape that ‘look like a letter’ and ‘do not look like a ligature’(Lecolinet & Crettez 1991). Cheriet (1993) describes a system that locates key letters in a word and classify by these ‘parts’. Both these methods correspond to the approach of Simon & Baret (1989) of finding regular and singular features in a word image. Once the regular features have been located (such as the word axis – usually containing ligatures), they can be removed and the remaining singular features can be extracted and classified.

(Lecolinet & Crettez 1991) detect graphemes using two competing processes which locate ligatures and characters after which the word can be segmented into regions most likely to contain characters. Recognition of key letters (Cheriet 1993) works in a similar way; key letters (the singular features) are extracted from a word image through detection and removal of the ligatures (the regular features). Classification of the remaining characters is supplemented with whole word features such as detection of ascenders and descenders.

The remainder of this section looks at three analytical recognition systems in more detail: Bozinovic and Srihari's system was one of the first major off–line handwriting recognition systems; Fukushima uses the 'selective attention model' for segmenting and recognising cursive words using implicit segmentation; finally, Srihari's system for recognising U.S. postal addresses compares a Hidden Markov Model and a hypothesise and test approach to recognising words.

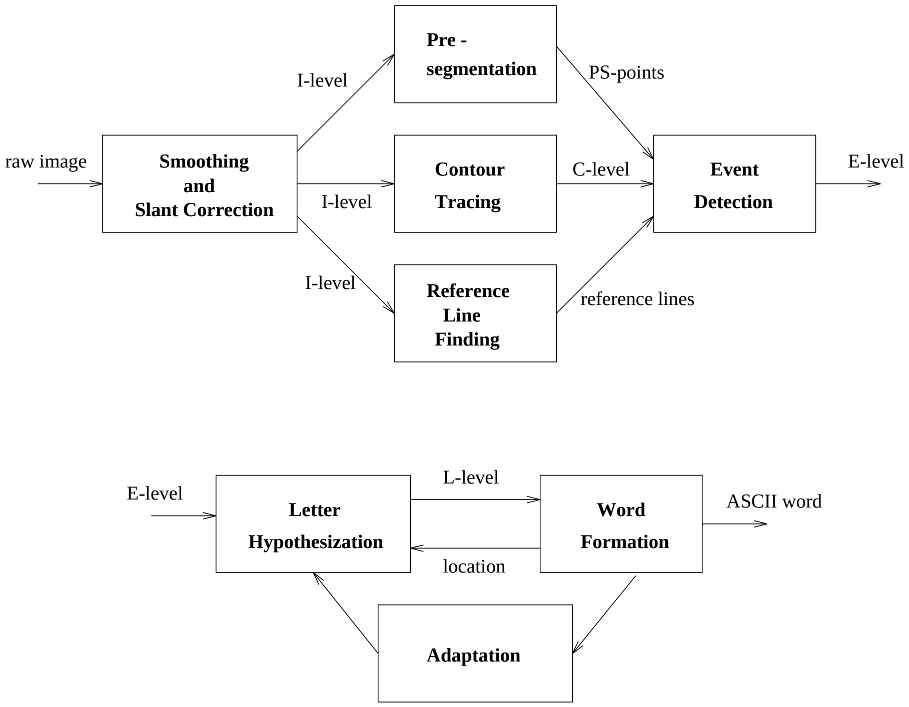


Figure 2.4: Bozinovic and Srihari system organisation

Bozinovic and Srihari (1987, 1989)

The off–line handwriting system described by Srihari & Bozinovic (1987) was one of the first full off–line recognition systems (this was also reported in Bozinovic & Srihari (1989)). A schematic view of the system is shown in Figure 2.4. The preclassification stage is shown above the recognition stage.

Preclassification involved a preliminary smoothing and slant correction process generating a preprocessed image (I–level). This preprocessed image was then passed into three modules : reference line finding, for detection of ascender, word body and descender regions of the image; presegmentation which made many *loose segmentation* decisions, finding presegmentation points (PSP's); contour tracing took a chain code (Freeman 1961) of the word and stored it in a topology tree.

These three modules then passed the results onto the event detection section. An event was a feature which was detected using the chain codes, word regions and segmentation points.

Features included those based on strokes, loops, dots and cusps in the word image. This 'E-level' representation was then passed to the next phase to be classified.

Classification was achieved by a process of letter hypothesis followed by word hypothesis. Letter hypotheses were converted into a word classification through the application of a stack search algorithm. The algorithm builds word hypotheses by appending possible word prefixes from left to right and accepting full words that appear in the lexicon. If the search results in no candidates being found, the word is rejected as unclassified.

In testing, the classifier was shown to recognise 77% of words correctly (most probable candidate) when trained with 66 word images and tested with 64 word images. In this case, the lexicon from which candidate words could be taken was 710 words. The classifier was further tested taking the top two candidates and results increased to 81%.

An adaptive loop was also implemented where words that had been correctly classified were then used to adapt the parameters of the system. When a word is classified, the events that generated that classification were also known. Adaptation was implemented following classification by adjusting the relationship (i.e. the conditional probability) between the features and possible letters.

The adaptation was tested by first classifying the 64 words, adapting the classifier and then *re–classifying* the 64 words. In this case, the results improved to 78% correct. This showed that a marginal increase in performance could be achieved through the adaptation of the classification parameters. However, the word images presented in (Bozinovic & Srihari 1989) are neatly written with all characters clearly visible. The segmentation routine would most likely fail if presented with poorly written words. However the principle of flexible segmentation reappears in work such as (Lecolinet & Crettez 1991) and (Chen, Kundu & Zhou 1992) where very loose definitions of segmentation are used. Similarly, relating the probability of word parts and possible letter candidates has recently been mirrored in the use of Hidden Markov Models to recognise word shapes.

Fukushima (1993)

The classifier described by Fukushima (1993) is based on the 'selective attention model' (Fukushima 1987). The selective attention model is a multilayer neural network which has the ability to segment and classify at the same time.

The network is a hierarchical multilayered network with forward and backward connections between each layer. Forward connections are used for pattern recognition while the backward connections are used for the selective attention and segmentation. If the network is considered with just forward connectors, it is similar to the Neocognitron (Fukushima 1988) where each level of the network takes its input from the lower level and recognises more global features, by looking at the pattern at higher levels of abstraction. Classification is achieved in the final level where the cell with the highest activation represents the class of the input stimulus.

The backward paths from the recognition layer to the input stimulus are controlled by *gate signals*. These implement the associative recall in the network. When a pattern is presented to the network, the cell with the highest activation in the recognition layer controls the backward paths such that only the stimulus that caused this classification are active at the input of the network. By isolating the stimulus for each classification, a segmented word can be recognised by classifying and then segmenting all the characters in a left to right fashion.

However, the problem of ambiguous characters in the image (such as the 'dog', 'clog' example) is not discussed in this work. Also the images of cursive script presented in (Fukushima 1993) are extremely neat with clear characters and ligatures leading to easily segmented words. Performance figures and lexicon sizes are not reported; instead sixteen correctly classified word images are presented. The method of classification and segmentation applied by Fukushima means that badly written words would not be recognised since letters and ligatures would be less discernible.

Srihari (1993)

There has been much work on postal address recognition, some other approaches are described in (Downton, Tregidgo & Kabir 1991, Miletzki, Uebel & Schulte-Hinsken 1994);

unlike these systems, Srihari, Govindaraju & Shekhawat (1993) uses a broad definition of a 'word' and also achieves good results on multi–author recognition of North American state names.

Srihari, Govindaraju & Shekhawat (1993) describes a system for reading handwritten U.S postal addresses, the aim of the recognition process being to determine the delivery location of a mail piece. The delivery location is the ZIP code plus four characters which together represent the state, city, post office, block face and PO box.

Recognition of the addresses follows a sequence of preprocessing, to remove underlining and split into lines and word separation. At this point words are defined to be any of the following : text, city, state-abbreviation, digit–string, digits–dash, ZIP+4 code, barcode, PO box and noise. Each word is classified into one of the above types through a first classification procedure to determine the type of each word and then uses syntax rules for address blocks to determine the exact type of each word.

Once each word image has been tagged with the word type, a recognition attempt can be made. Digit strings are classified by segmenting the word image into regions and recognising the regions (using knowledge of the number of characters expected in the block).

Two methods of word recognition are used for the street names. Firstly a method of hypothesis generation and testing is used where a continuous reduction of the lexicon is made at different stages in the algorithm. Segmentation is done by generating a series of segmentations for that word. Candidate letters are then generated which are then recombined and passed onto a dictionary checking process where valid words are chosen and ranked.

The second method of classification was to divide the word image into segments and to recognise those segments using a Hidden Markov Model based on that of Chen et al. (1992). These were then tested on random images extracted from the US postal stream and when the two classification methods were compared, hypothesis generation and testing outperformed the HMM method (hypothesis generation achieving 78.5% and HMM generating 68.0% for a 1000 word dictionary). However when combined, the classification result increased to 84.4%.

This system shows that recognition of multi–author words can be improved with the application of different classifiers. However, the combination strategy between the classifiers was not discussed.

2.2 Global methods of recognition

Global recognition of writing avoids the problems of segmentation by looking at the whole word shape. Whole word recognition was first proposed by Harmon (1962) where single words were written on an electronic tablet between two guiding lines. Features based on crosses of the centre line, dots, ascenders and descenders were extracted and then looked up in a 'truth table'. Harmon reported 96.9% correct recognition of the words 'zero' through to 'nine'. However, it was recognised that whole word recognition was difficult with the restrictions of memory at the time.

Farag (1979) reported an on–line whole word recognition system which used a Markov model to calculate the probability of a sequence of strokes being a particular word without segmenting that word into letters. As with Harmon's system, the vocabulary was limited to 10 words. With this small lexicon a first order model generated 98% correct results while a second order model produced 100% results. Brown (1981) recognised words as whole words, showing that the preprocessing of the word image increased the recognition rate by as much as 10%. Again, features used in this system were stroke based.

It was not until Hull (1988) proposed his computational theory of text recognition that whole word recognition was considered feasible. His model used gross word features to generate hypotheses which were then tested by looking again at the word image. This has been used in the area of postal address recognition (for example, (Ho, Hull & Srihari 1990).

Simon's singular and regular features of writing (Simon & Baret 1990) can be used to recognise words at the word level. The applications described in the previous section used characters as the singular features, while Simon's work extracts features from the whole word and uses these to classify the word.

In this section, three approaches to whole word handwriting recognition are presented. Firstly, Hull's theory of handwriting recognition is described, then Lecolinet's model of top–down and bottom–up recognition is presented which is, in part, similar to Hull's model of reading. Finally, Senior's whole word recognition is presented which builds on a recurrent neural network originally designed for speech recognition.

Hull's computational theory of reading text

Hull developed a theory of word recognition based on a review of psychological models of reading (Hull 1988). The predictive part of his model was developed from two sets of psychological work, that of Hochberg (1970) and Rayner (Rayner 1978, Rayner, Carlson & Frazier 1983). Hochberg's model was a top-down model of reading. In such models, the reader is constantly making hypotheses about words which are to appear. These have since been questioned by some studying the psychology of reading (Stanovich 1980) because the amount of processing needed in order to constrain a future word choice is not confirmed when tested experimentally. However, some form of preprocessing is demonstrated by Rayner's study of eye–movements which suggests that the words to the right of fixation provide information which is subsequently used to help recognition of the word. In this case, rather than building word hypotheses before fixating a word, a strategy for feature extraction is being made.

Hull's theory of recognition consisted of three parts :

- Computation of a gross visual description. Used to build hypotheses about the word and to direct further analysis of the word image.
- High–level knowledge processing. Where higher level information would have an influence on the reading process.
- Goal–directed feature testing. Here the word image is again processed based on the information derived in the first two processes.

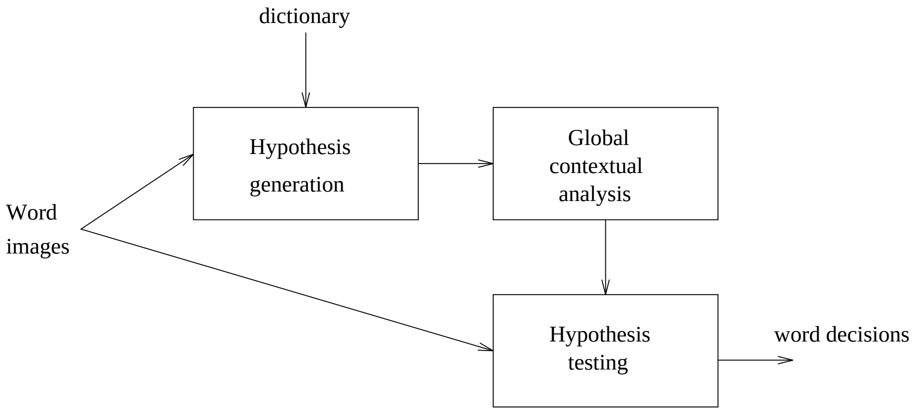


Figure 2.5: Hull's outline algorithm framework, from Hull (1988)

The theory was implemented as three stages : *Hypotheses generation*, *hypotheses testing* and *global contextual analysis*. The framework is shown in Figure 2.5.

Hypothesis generation created a set of candidate words — or a neighbourhood — which were then used to guide further classification. However, the experiments reported by Hull (1988) were on synthesised features of words from the Brown corpus. Later work (Ho et al. 1990) tested the theory on images of text, however only recently (Cohen 1994) has the theory been applied to handwritten words. Cohen's implementation is based on a model of context and is reviewed in the next chapter.

Backward matching for word recognition

(Lecolinet 1993 a) describes top–down and bottom–up approaches to cursive script recognition, these are [2] :

- Bottom–up strategies — these take a word image and classify through applying successive levels of abstraction. This is commonly achieved through analytical recognition (as discussed in the previous section).
- Top–down strategies — where words are recognised from holistic analysis of the word shape. However Lecolinet notes that many top–down approaches use a bottom–up process for feature extraction.

He then describes a system (Lecolinet 1993 b) that uses a 'backward matching' process which takes the top–down and bottom–up methods and makes them compete for information in the word image [3].

Lecolinet notes that different letters in words make a different contribution when recognising the word. *Lexical significance* describes the fact that lower frequency letters in a vocabulary are more informative and *visual significance* where, for example, letters with ascenders and descenders are easier to recognise than those without (particularly in words written quickly). He then notes that in French none of the 6 most frequent letters have a descender or an ascender while 11 of the 14 less frequent letters have. Knowledge of the most informative characters is used to order the letter hypotheses tested in a word image.

Bottom–up recognition is used to detect robust features in the word. Top–down recognition takes a small set of word candidates and tests these words against the word image. Word hypotheses are constructed of the letters contained in that word, however they are tested in the order of 'visual importance' rather than left–to–right. Letter recognition is carried out by testing for visual features of a letter being close to each other, these are again tested in order of their visual significance. To control the testing of hypotheses, a tree is made up

[2] These terms correspond to the analytical and global recognition techniques described earlier.

[3] This is similar to his grapheme system where classifiers compete for ligatures and graphemes.

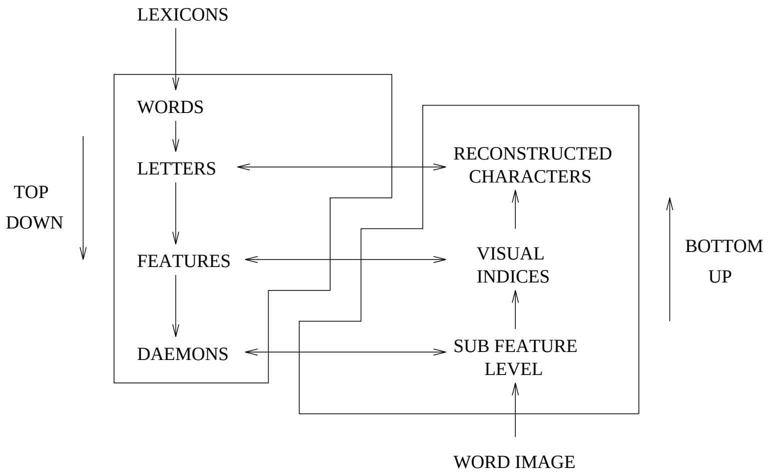


Figure 2.6: Top–down and Bottom–up processes, from Lecolinet (1993)

with levels for words, letters and features with the most important features to the left of the tree. A depth–first search of the tree seeks consistent combinations of word, letter and feature evidence. The interaction between top–down and bottom–up levels can be seen in Figure 2.6.

Lecolinet & Likforman-Sulem (1994) reports results for this system tested with lexicons between 20 and 70 words. For a lexicon of 20 words, recognition rates were 84% correct (95% in top 5 candidates) and for 70 words results were 65% correct (88% in top 5).

While the goal of Lecolinet's work is general document recognition and postal address recognition, the number of writers in the test set is not presented with the results. However, any system that is to recognise postal addresses must be able to cope with a wide variety of writers. It is interesting to note the size of the test lexicons which are considerably smaller than would occur in the intended applications. For example, the U.K. postal address file (used for gathering context for address recognition (Downton et al. 1991)) contains well over 100,000 different words in 1,500,000 addresses.

Word recognition using recurrent networks

Senior & Fallside (1993 a) describes a system for reading whole words using a recurrent neural network which can learn patterns through time. Previously, this network architec-

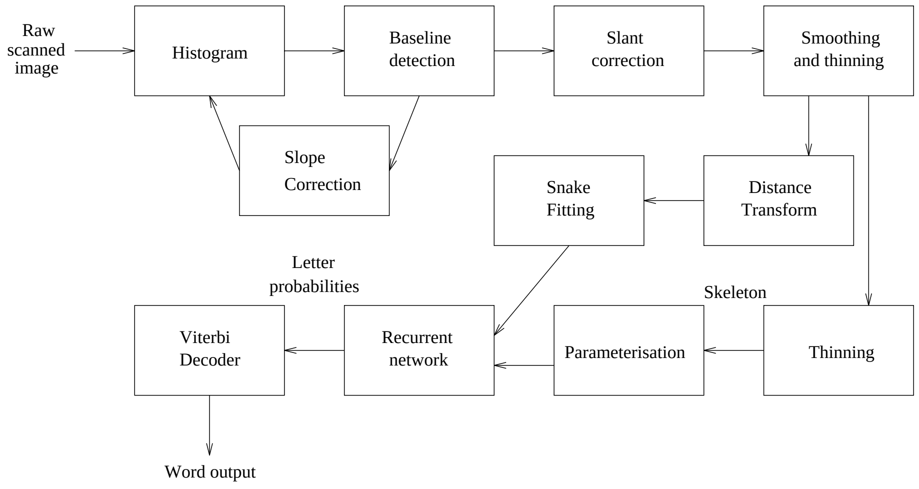


Figure 2.7: Senior's word recognition system, from Senior (1993b)

ture had been applied to speech recognition where the network could capture and recognise the changing articulation of words over time.

The initial system consisted of a sequence of preprocessing techniques followed by recognition using a recurrent network and Hidden Markov Model (HMM). Later improvements to the design included a feature spotting module to augment the left-to-right features used in the original system (Senior & Fallside 1993 b). A schematic of the modified system is shown in Figure 2.7.

Words are initially preprocessed by first correcting the slope (normalising the word image so it lies horizontally) and correcting the slant. The word image is then smoothed to remove the noise introduced in the previous processing and to prepare the word for skeletonisation. Stroke based features can then be extracted from the normalised image. The word image is split into many small rectangles and strokes in these areas are detected. A word is then represented as a sequence of vertical slices with localised stroke information stored from the top of the slice to the bottom.

The recurrent network used to classify words is a single layer of perceptrons where some of the output nodes are connected to the input layer via a single time delay. With this architecture, recognition of input nodes at time t affects the recognition of input nodes at time, $t + 1$. The output nodes of the network represent characters and so for every word slice presented to the network, the most likely character for that input (given the preceding slices) is output. For example, classifying the letter 'w' could result in the output 'iiuuunnwww' from the network (if it had been segmented into 10 slices). A Hidden Markov Model is then used to decode the output of the network into valid words. There is one network for each word that can be recognised by the system and recognition is performed by finding the model that most closely describes the output of the recurrent network.

Senior & Fallside (1993 b) describes a further feature spotting technique based on deformable splines. These larger stroke–based features are extracted from the word image and presented to the network along with the word slices. This further information is reported to increase single author recognition by 10% to 78.7% on a dictionary of 825 words.

2.3 Hidden Markov Models

Recently there has been a growing interest in using Hidden Markov Models (HMM's) for handwriting recognition. These have been used extensively for speech recognition where segmentation of an utterance into words (and phonemes) is difficult. This is analogous to the problem of segmenting cursive script where segmentation points are difficult to locate.

Senior's whole word recognition system presented earlier is one of many recent uses of HMM's in handwriting recognition. HMM's are used to model stochastic processes where the probability of observing an event depends upon the observations immediately preceding it (Rabiner 1989). Markov models can be used for recognition by having a separate model for each class. The probability of an observation sequence can then be calculated for each model. For handwriting recognition, the observation sequence is a set of left–to–right features extracted from the word image.

In the above case there is usually one HMM for each word in the dictionary. However, Chen & Kundu (1993) describes a system where there is only one model and the *path* through the model indicates the class of word.

This section looks at three HMM handwritten word recognition systems. Kundu, He & Bahl (1989) was one of the first HMM handwriting recognition systems. Caesar, Gloger, Kaltenmeier & Mandler (1993) and Kaltenmeier, Caesar, Gloger & Mandler (1993) report on a HMM which does not segment the word image into letters but uses extra holistic features to supplement the left-to-right nature of the observation sequence. Finally, Gilloux, Leroux & Bertille (1993) report on two sets of experiments : the first where the lexicon is small and each word is recognised by an individual model, and a second where the lexicon is allowed to be large and recognition is achieved by a sequence of HMMs representing letters.

The one-dimensional input to a HMM makes the approach suitable for on–line recognition; this is discussed in Lorette (1994).

Kundu *et al* (1989)

Kundu et al. (1989) presented a Hidden Markov Model approach to word recognition in which the Markov Model was used to post–process a sequence of letter taggings. In general, a word image was segmented into characters and then each character was classified as being one of 90 observations. The process of classification was carried out by a HMM which calculated the most probable sequence of letters (the hidden states) given the sequence of letter observations.

The problems of segmenting the word image into characters was sidestepped by writing the words with large gaps between the written letters. Each letter had a set of features extracted based on moments and feature spotting methods (loops, cusps, t-joins etc). The training data consisted of letters which were written by 100 different writers. Features were extracted from these letters and then clustered into 90 discrete observation sets. The probability of the letters belonging to a particular class could then be calculated following the clustering.

The hidden states of his model were the 26 characters, and so the full Markov model consisted of a state–transition matrix (the probability of one letter following another), a confusion matrix (the probability of a letter given a particular observation) and the initial probabilities of letters at the start of words.

This system achieved 85% correct recognition of words, where the word was also counted as correct if it appeared in the list of candidate words. This recognition rate was increased to 92.5% with post–processing of the incorrectly classified words. However, a method for automatically determining the 'correctness' of the classification was not discussed.

The constraint of letter segmentation was overcome by Chen et al. (1992). Words were allowed to be over and under segmented with some restrictions, for example a letter could not be segmented more than three times and a letter could only be merged with one other letter. Also, a null state for non–information strokes, such as ligatures, was introduced into the model.

This classifier was tested on the U.S. postal city names with considerably poorer results than those in the initial classifier reported by Kundu. When testing with a lexicon of 271 U.S. city names (taken from the USPS mailstream) the classifier produced results of 26.6% correct. However, many of these errors are words not in the lexicon. In this case, the words were recognised by comparing the top 20 paths through the model with all words in the lexicon, applying a cost function depending upon the number of characters inserted and deleted. Recognition of words using this method was 64.9% (top candidate) and 80.9% (top 5 candidates).

These results show that the Hidden Markov Model approach to handwriting recognition was promising within the constrained domain used by Kundu et al. (1989). However, when tested against multi–author words, the simple approach of applying a HMM to word recognition is not sufficient to classify words. Indeed, with more sophisticated HMM architectures, the recognition rates of HMM classification is still around 21–25% for a lexicon of 100 to 1000 words (Chen & Kundu 1993). The proposed method of testing each candidate path against each word in the lexicon would also be computationally expensive. Particularly in the test domain of U.S. postal addresses where the lexicon would be many thousand words and the time to recognise each envelope would be small.

Caesar, Kaltenmeier *et al* (1994)

Caesar et al. (1993) describes a system for recognising U.S. city names. The original system used the left-to-right sequence of observations while the system described in Caesar, Gloger, Kaltenmeier & Mandler (1994) used a second classifier with a 'word length' feature further to reduce the number of candidate words.

The model used was a Semi-Continuous HMM (SCHMM) where each letter was represented by a different model and the most probable sequence of models was chosen (similar to the above model used by Kundu). Letters were said to be either *upper–case* or *lower–case* and either *print–style* or *script–style*. This led to four variations for each character. Kaltenmeier et al. (1993) described the model topology where the model for each character was specifically trained to recognise each of the four variations. For the case of unrecognisable characters, a fifth *joker* variation was built into each letter model.

Classification was performed by first normalising the word image. This included skeletonisation, slant, baseline and size normalisation. A window was then passed over the word image from left to right where a feature vector (based on five regions : ascender, above middle, middle, below middle and descenders) was extracted, however the precise features are not reported. The most probable sequence of models is then calculated as it was in the system described by Kundu.

Training the model involved taking a tagged set of word images (where each letter is tagged with its identity and its variation), extracting the features and then performing vector quantisation and linear discriminant analysis before using the features in the forward–backward algorithm (Rabiner 1989) to train the models.

Caesar et al. (1994) reports 91% and 85% for U.S. and German city names respectively with a combined lexicon size of 100 entries. However, it is unclear if this represents a top–choice recognition rate, or if the correct word appears in a candidate set. The bias towards U.S. city names is attributed to the large number of American city names in the training set compared to German city names.

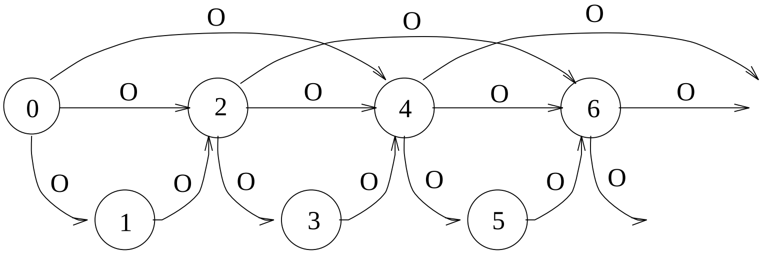


Figure 2.8: Word Markov Model, from Gilloux (1993a), the ‘O’s represent observations associated with transitions rather than states.

Gilloux et al (1993)

Gilloux presented two methods of using HMM’s for recognising handwritten words. Unlike the above methods, these relied on a word being segmented into pseudo–letters before classification. However, following segmentation the two types of classification reflect the two approaches of global recognition of the word (with each word being represented in a different Markov model), and analytical recognition where the word is classified through its component characters.

Feature extraction in both systems involved finding the vertical baselines in the word image (to find ascenders and descenders) and then splitting the word into pseudo–letters. For each pseudo–letter, loops, ascenders and descenders were detected and the relative position of these features was extracted. These were then associated with one of 27 observation symbols (26 characters and space).

The first type of recognition was for a domain with a small number of possible words, for example, cheque recognition. In this case, each word can be modeled by a separate HMM. The whole word Markov model associated observations with *transitions* rather than states which meant that insertions, substitutions and deletions of pseudo–characters could be encoded into the model structure (shown in Figure 2.8).

On a lexicon size of 27 word classes, such a model achieved 79% correct recognition for the top choice which increased to 98% for the top four choices. These results were from 2492 cursive words written by an unspecified number of writers on real cheques.

A lexicon of 27 words could be applicable to cheque recognition where the lexicon is the legal sum to be written. However, in an application such as postal address recognition, there is a large number of words that have to be recognised. In this case Gilloux proposes a system that concatenates letter HMM's together (Gilloux, Bertille & Leroux 1993). Gilloux extracts a set of features left-to-right along images of handwritten city images. City names are classified globally, spaces in city names such as 'Boulogne sur seine' are extracted and recorded as features. Classification is achieved by finding the most probable sequence of letter models (using the Viterbi algorithm, (Forney, Jr. 1973)).

Gilloux simulates the use of this classifier in a postal address recognition system. It is proposed that the classifier is used as some post–processing following post code recognition. The classifier is tested by building simulated candidate sets and then using the classifier to recognise these 10 city names. The candidate sets contain the correct city name and nine randomly selected from a lexicon of city names. The assumption being that similar post codes lead to visually different city names. Despite admitting that this assumption is incorrect, Gilloux shows that the classifier recognises the correct city 91% of the time.

The classifier is then tested with a 100 and 1000 word dictionary, that is, 99 and 999 random words are considered along with the correct word. In this case, the performance drops to 77.6% correct classifications for a 100 word dictionary (92.4% in top 10 candidates) and 42.5% correct for a 1000 word dictionary (78.8% in top 10 words).

These training and test images come from live mail addresses and so the recognition rates are comparable with other approaches to multi–author word recognition. However, the reported training set consists of 904 city name images and test set 226 images, so the reported results suffer from a lack of training and testing.

2.4 Summary

It has been noted that there are two methods for off–line word recognition. The first segments a word into letters (or at a suitable sub–word level), classifies these segments and reconstructs the word identity. The second approach is to recognise words as whole symbols and to avoid any segmentation process.

The HMM approach also divides into these two categories, as demonstrated by Caesar. However, many current HMM systems are more akin to ‘segment and classify’ routines than to the whole word approach.

We see also that applications are generally split into single and multi author recognition. The systems described by Bozinovic & Srihari (1989) and Senior & Fallside (1993 b) are both aimed at single author recognition. In the first system the lexicon is small, between 66 and 80 words while Senior’s work uses a lexicon of up to 1000 words.

For single author recognition we are faced with the problem of training the classifier. A small lexicon allows the writer to train the classifier; however training a classifier for around a thousand words is tedious and impractical. One possible solution to this is to learn words during the recognition process. The classifier can be trained on a small number of frequent words and later learn words as they appear during normal use of the system.

Bozinovic & Srihari (1989) also showed that the use of an adaptive feedback loop can extend the training of known words into the recognition process. We propose to augment the adaptation of known words with learning of new words to overcome the training problem for single author recognition.

Chapter 3

CONTEXTUAL KNOWLEDGE FOR HANDWRITING RECOGNITION

The previous chapter showed that word recognition usually results in a candidate set of possible words. Srihari & Bozinovic (1987) also showed that adaptation could be used to improve recognition rates. If we define context to be information used in the recognition process which is not derived directly from the current word image, then it follows that adaptation of a classifier with the aim of improving recognition rates can be considered a form of context.

The aim of this chapter is to review contextual processing and to compare adaptation with these other sources of information.

3.1 Introduction

When classifying a written symbol, the result is usually a candidate set of words or letters. These candidate sets represent the visual ambiguity between classes being recognised and are usually dependent upon the chosen feature set or classifier.

If we consider recognising the set of alphanumeric characters, we would expect (and usually get) the characters ’0’ (zero) and ’O’ (oh) to be confused. Commonly (for example, Fairhurst & Cowley (1993)) the two classes are merged into one larger class. This merged class can be considered a candidate set containing the two characters ’0’ and ’O’, other

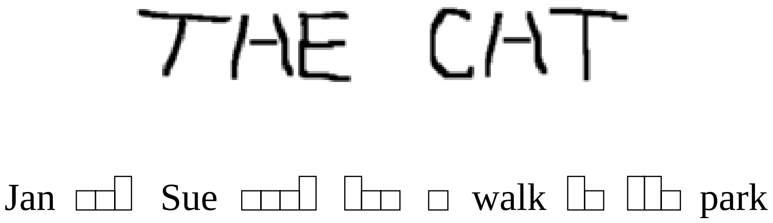


Figure 3.1: Examples of ambiguous text

methods could store the two characters separately. However in both cases the ambiguity needs to be resolved using some other source of information.

Recognising handwritten words generally results in larger candidate sets since there is far more scope for ambiguity. Figure 3.1 shows different types of ambiguous symbols. In the first, from Selfridge (1955), the letters A and H (in ‘the’ and ‘cat’ respectively) are the same, however the letter can identified using information in the surrounding letters. The second example in Figure 3.1 shows a passage where many words are unknown except for the overall shape of the word (Haber & Haber 1981). The unknown words can be inferred because of the sources of syntactic, semantic and orthographic (word shape) information already present in the sentence. Similarly, the example in Figure 1.2 shows the two words ‘dog’ and ‘clog’ where surrounding information would be needed to disambiguate the words.

Along with the inherent ambiguity in the written word, there is ambiguity introduced in the classification process. Depending upon the features used, two visually different words can result in being in the same candidate set. A trivial example of this would be a classifier which used features based only on ascender and descender counts. In this case, the words ‘recognised’ and ‘dig’ would result in the same feature vector. Of course, we use more features than in this example and try to encode the word shape using the most descriptive set of features which can practically be extracted. However, this ambiguity means that in some cases, extracting features from visually similar words results in the same feature vector for both words. If the classifier cannot be changed, then this form of ambiguity has to be resolved using contextual information.

The examples in Figure 3.1 show words and sentences where the identity of a word can be inferred from the surrounding information. The information used to infer these words can be encoded using a model of natural language, for example, a syntactic model may promote the candidate word 'and' between the words 'Jan' and 'Sue' over other possible candidates such as cat, sat, cut, not. Other forms of context have been developed for constrained domains such as postal address recognition and cheque recognition. For example Downton et al. (1991) described how the redundancy in British address blocks can be used to infer the address written on an envelope. In this case, a dictionary of postal addresses indexed using the post code, was used to provide further information to supplement the classification process. Similar methods have been used for U.S. postal addresses and recognising cheques, for example Cohen (1992).

Finally a third form of context can be exploited when words are recognised at the letter level. In this case, each letter may have a number of candidates and can be combined in a number of ways to generate different candidate words. These words can then be tested against a dictionary of valid words rejecting all candidates that are not in the dictionary. Since linear dictionary search is computationally expensive, methods have been developed that make this search more efficient.

The types of contextual information described above all use information other than that in the written word to help in classifying the image. The above examples show how a model of syntax can promote plausible candidate words and how a dictionary can be used to help exploit redundancy or choose valid candidate words. We aim to use adaptation to change the classifier itself and hence change the results in subsequent word classifications. In this case we have information derived from previously classified words effecting the result of subsequent classifications, in the same way that context uses surrounding word classifications and external information to change possible candidates.

However, unlike other sources of context, adaptation can only overcome the ambiguity in the classifier and not the inherent ambiguity in the written word. The aim, therefore, is that adaptation should complement other sources of context by promoting the correct words and so enhance further contextual processing.

Note that the methods of language modelling used in handwriting recognition systems is similar to that used in speech recognition. The reader is referred to Waibel & Lee (1990) and Atwell et al. (1993) for a description of algorithms and applications of language models in speech recognition.

This chapter considers context in more detail and contrasts the use of existing contextual techniques with that of adaptation and includes a discussion of how different methods of context can be ranked in terms of sensitivity to misclassification errors.

3.2 Contextual knowledge in human reading

Evidence of contextual information in human reading was provided by Tulving & Gold (1963). They measured the time taken between seeing a word flashed on a screen and identifying the word. Prior to flashing the word on the screen a set of words would be displayed which provided contextual priming for word identification. For example, if the word to be identified were ‘performer’, an 8 word lead–in could be ‘the actress received praise for being an outstanding...’.

The number of words in the lead–in was varied (between 1 and 8 words) in the experiments and it was found that the time taken to name the word was reduced as the number of words in the lead–in increased, i.e. the reader was given more relevant context.

A second set of experiments used lead–in passages which did not provide context for the displayed word (for example, the lead–in for the word ‘performer’ may be ‘the rain in spain falls mainly on the...’). These experiments showed that as the amount of inappropriate context was increased, the time taken to name the word also increased.

Psychology researchers have tried to describe how contextual information is used in human reading. These models are commonly described as top–down, bottom–up or interactive. Top–down models, for example Goodman (1967) build hypotheses about a word’s identity before testing these against the visual cues from the word. Bottom–up models use the visual stimulus to generate word hypotheses which are then processed using other contextual information.

Both top–down and bottom–up models have been criticised for since they do not account for performance in psychological tests (a discussion of the problems can be found in Stanovich, 1980). Interactive models of reading have been proposed to better account for experimental results, for example, those proposed by Morton (1969), Rumelhart (1977) and Stanovich (1980).

Interactive models of reading take many sources of information and process them at the same time. An example is the Logogen model proposed by Morton (1969) where visual, auditory and contextual information are fed into logogens which act as 'experts' building up evidence of a particular word as information is provided. Once enough relevant information to a logogen has been collected, it fires and that word is recognised.

A single logogen represents a single piece of semantic information, so the words 'KEYBOARD' and 'keyboard' would be represented by the same logogen, while the words 'KEY' as in that on a keyboard, and 'KEY' which is used in a lock would have different logogens.

Interactive models, for example Rumelhart (1977), are considered better models of reading since they account better for the measured results of reading experiments in people. Rumelhart's model had many knowledge sources, both visual and contextual, being used simultaneously to recognise a word.

Stanovich (1980) argued against the commonly held belief that it was the fluent reader that used more contextual knowledge when reading. He concludes :

> Thus according to the interactive-compensatory model, the poor reader who has deficient word analysis skills might possibly show a *greater* reliance on contextual factors. In fact, several studies have shown this to be the case.

Evett & Bellaby (1994) compared aspects of human and machine reading cursive script and found that human word recognition outperformed machine recognition of cursive words by around 14% (61.5% machine recognition compared to 85.6% for the human). Stanovich's observation reflects the continuing reliance upon and development of contextual information for machine recognition of handwriting.

3.3 Cohen's computational theory of context in a constrained domain

Cohen's computational theory (Cohen 1994) was developed to recognise multi–author handwriting in a constrained domain. As mentioned above, many applications for forms processing have scope for exploiting redundancy and information that can be inferred from the location of the writing in a document. Cohen's model is essentially the same as that proposed by Hull (1988) with the addition of a highly defined sequence of contextual processing derived from a set of constrained application domains.

Cohen's theory disregards syntactic and semantic knowledge in favour of redundancy and spatial information[1]. The basic principle is that of *purposive reading*, i.e. only reading the relevant parts of the document in order to gain an *interpretation* of what was written rather than a transcription of the writing.

The interpretation process is shown graphically in Figure 3.2, where domain knowledge and extracted features are combined with a global description (and positional information) to form hypotheses that may be then be tested against the image after which a decision is made to interpret the writing.

An algorithm to match this theory was developed and was demonstrated against six application domains. The algorithm contained six stages :

- **Create phrase location hypotheses :** Document is split into possible 'phrases', i.e. regions.
- **Compute features from phrases :** Gross features of phrases are extracted. The fea-

[1] This appears to be because psychological models of reading did not provide a model of syntax and semantics that could be translated into an algorithm Cohen (1994, p.2).

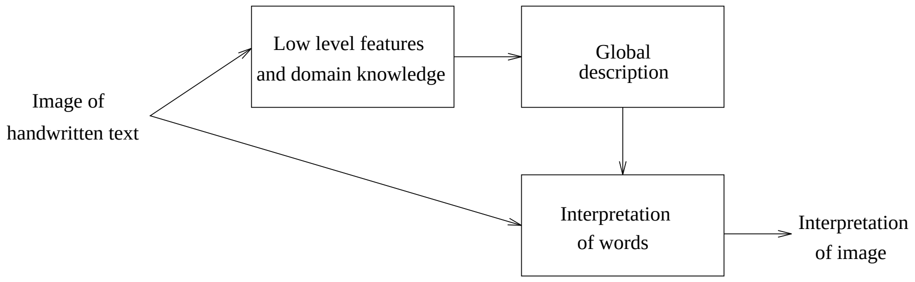


Figure 3.2: High–level description of Cohen's text interpretation (from Cohen, 1994)

tures are commonly length and existence of special characters and digits.

- **Assign phrase classifications :** Assign labels to different sections of the document, e.g. script, digit string or fraction.
- **Generate parsing hypotheses :** Attempt to parse sequence of 'phrases'. For example, *cursive string – cursive string – digit string* could be parsed as *city – state – ZIP code* in U.S. postal addresses.
- **Generate interpretation hypotheses :** Build hypotheses about the information in the image from recognised or categorised phrases in the image.
- **Choose most feasible interpretation hypothesis :** Choose a hypothesis based on the confidence values assigned in the different stages and return the result.

Note that the aim of this system is to extract the information the writer intended to communicate, not to recognise the writing. Hypothesising based on global information such as this is analogous to the statement by Stanovich about the role of context in poor readers. However, this places a great reliance for the correct hypothesis to exist at some stage in the processing. If the correct hypothesis does not exist then a completely incorrect interpretation could be made; this is inconvenient if the application is postal sorting and potentially lethal if it is prescription recognition.

Cohens computational theory is restricted to interpreting forms written by multiple authors, whereas we are interested in transcribing the writing of a single author without the positional constraints commonly found in forms processing.

3.4 Exploiting redundancy for contextual information

Off-line applications tend to be based around forms processing which usually contains short sections of writing giving specific constrained information. This section considers the use of redundancy within address recognition and cheque recognition. Redundancy in handwriting recognition occurs when the same piece of information has been written twice, or when you can infer one piece of information from another. Redundancy can usually be exploited within application specific domains where the position of written text can constrain the meaning.

Downton et al. (1991) describes a system for recognition of British postal addresses. The postcode is recognised using character recognition techniques augmented with the syntax of valid postcodes. Once a candidate set of syntactically correct postcodes has been generated, these are tested against a dictionary of valid postcodes implemented in a trie structure for efficient searching. This removes all invalid postcodes and returns the address associated with the postcode.

The remaining valid postcodes are then checked against the written address. The first and, if possible, the last characters from the post town are extracted and classified. If the evidence extracted from the post town corresponds to the address associated with the valid postcode, then the classification is taken to be correct. Otherwise it is rejected. Downton *et al* predicted that 35% of post towns would be correctly verified.

Cohen, Hull & Srihari (1991) reports a system which exploits redundancy in United States postal addresses in a similar way. However, this system relied upon street, city or state name being correctly recognised unlike the system described by Downton which gathered evidence from easily segmented parts of the address block. The system described in Cohen (1994) relied less on word recognition and more on accumulating enough evidence to correctly interpret the address block.

Cheque recognition is another application where recognition performance can be enhanced by exploiting redundancy. Again, Cohen (1994) describes recognition of cheques in terms of text interpretation. However, a more direct use of redundancy was given by Moreau, Plessis, Bougeois & Plagnaud (1991). Here the cursive amount written on the cheque is recognised using whole word techniques. Recognition at this stage can also be facilitated using syntactic knowledge (Paquet & Lecourtier 1993). Recognition of the printed amount is achieved using a digit classifier. The amount can then be verified by comparing the results of the two classification processes and can be rejected if the two results do not match.

As mentioned earlier, redundancy relies upon the same, or similar, information being duplicated in the document being read. Such processing is unsuitable for recognition of continuous writing since redundancy in natural language cannot be directly extracted using a word's position in the document (although syntactic techniques provide clues to the types of words in a sentence). In contrast, however, there is redundancy in the sense that the same word may be repeated many times in a passage. For example, in the previous sentence, the word 'the' is repeated twice. Zipf's shows that a small number of words occur frequently while the majority of words occur with a low frequency (Zipf 1945). When recognising continuous handwriting we aim to exploit the redundancy resulting from duplicated words since recognition and adaptation to a word will hopefully improve recognition rates if that word appears later in the text.

3.5 Language models of context

As mentioned earlier, psychological models of reading attribute some use of contextual information in the human reading process. In machine recognition of handwriting, some models of context are designed to reflect the human use of context, in particular the use of syntax, semantics and lexical access (dictionary lookup). Haber & Haber (1981) showed that even with only a few visual cues (in their case, the shape of the word), the word identity can be inferred using syntactic and semantic knowledge (for example, the 'Jan and Sue' example in Figure 3.1). These models of context can be described as language models and are normally used to aid recognition when the writing is continuous, for example, the transcription of handwritten documents where the lexicon is potentially large and the style of writing is unconstrained. However, many of the principles are reflected where the application domain appears restricted, for example, the use of syntax in the recognition of British postcodes and cursive amounts on cheques in the previous section. Similarly, this section ends with the recognition of constrained fields on U.S. census forms to which a 'language model' has been applied to increase the recognition performance.

3.5.1 Lexical context

The previous chapter showed that words can be recognised as either whole words or from the individual letters. When classifying at the letter level, there may be more than one candidate for each letter due to the ambiguity in the letter shapes. This ambiguity at the letter level leads to a number of possible word candidates for that word.

If the words are known to be from a fixed dictionary, then the word candidates can be compared to those in the dictionary (or *lexicon*) and those not in the lexicon can be removed. Using a restricted lexicon in this way means that a valid word (or set of words) is always chosen, however a linear search of a large dictionary can be very inefficient and so search methods have been developed to enable fast searching of dictionaries. Also methods of estimating the validity of words based on the probability of letters occurring together have been developed.

The n-gram method builds up a statistical model of letter transitions from a large body of text (or a corpus) and then uses the transition information to check words against the dictionary. In the simplest case, for example (Ehrich & Koehler 1975) letter transitions are stored in a binary transition matrix. Rather than storing binary transitions, the frequency of transitions can be stored, normalised and used to reduce and reorder the candidate set.

The main problem with n-gram methods is that words which are not in the dictionary can be marked as true. For example, if the lexicon contains two words 'cat' and 'ape', the following transitions are marked as true (c,a), (a,t), (a,p), (p,e). Given these transitions the non-words 'ca', 'ap' and 'pe' are valid, along with the valid words 'cat', 'cap' and 'cape' which do not appear in this restricted lexicon. As the number of words in the lexicon increases, so does the number of non-valid words increase.

The other method of storing dictionaries is to store them in a trie structure. Wells et al. (1992) showed that a 26-way tree structure could be stored using bit arrays to reduce the amount of memory used by the lexicon. When trie structures were compared with n-gram dictionaries, it was found that the trie structure was faster and more efficient (Ford & Higgins 1992).

When choosing between n-gram and trie dictionaries, we are choosing between the space needed to store the dictionary against the time to calculate the correct word. However, as the cost of memory and disk space reduces it is reasonable to store the dictionary explicitly (or in an n-gram dictionary) even where the lexicon is very large. For example, the phrase dictionary used by Breuel (1994 b) described at the end of the section.

3.5.2 Syntactic context

Top–down models of reading suggest that we use words that we have already processed to make hypotheses about the oncoming words. This method of using preceding words to aid processing the current word can be modeled on a computer and was used by Hull (1989) to show that it could reduce the number of words in a candidate set.

Hull assumed that when classifying a word, the identity of the previous word would be known. Given the identity of the preceding word, the candidate set for the current word could be reduced. So, for example if the preceding words were ‘he sat on the ...’ and the candidates were ‘chair’ and ‘share’, syntactic context would promote the word ‘chair’ since it is more frequently used as a noun. Note that the word ‘share’ could be used as both a noun and a verb. This ambiguity in the syntactic use of each word is modelled by storing the probability of the word having a particular tag; these are stored in the confusion matrix. The tag transition matrix stores the probability of word tags following each other.

The above work made the assumption that words were known; however, this is often not the case and later work (Hanlon & Boyle 1992 a, Hull 1992, Keenan 1992) assumed that the result of classifying words in the sentences were candidate sets.

As with Hull's work on the computational theory of reading (Hull 1988) the above work on syntax was restricted to using synthesised features of word shapes; this meant that the candidate set always contained the correct word. However, later work with synthesised features of words (Hanlon & Boyle 1992 b, Hull 1992) showed that there was potential for the use of a Hidden Markov Model in handwriting recognition.

(Hanlon & Boyle 1992 a) tested a HMM of syntax on real candidate sets generated from a whole word text classifier. This work manipulated the ambiguity in the system by using different feature sets and two different training sets. The features extracted were one dimensional Fourier features (Shridhar & Badreldin 1984) and features based on those synthesised by Hull. Training set feature vectors were clustered using a modified k–means algorithm (Zhang & Boyle 1991). Classification was achieved by finding the closest cluster to the unknown feature vector and using the words in that cluster to generate the candidate set. This meant that the possible candidate sets were known *a priori* and so the confusion matrix could be calculated prior to classification.

The probabilities for the tagging model were derived from the LOB corpus (Johansson, Atwell, Garside & Leech 1986) and the Viterbi algorithm (Forney, Jr. 1973) was used to find the most probable sequence of tags given the sequence of candidate sets. Candidate set reduction was achieved by removing all words which did not correspond to the most probable tagging of the sentence.

Unlike Hull's work which only reported candidate set reduction, Hanlon also reported classifier performance and tagging performance, i.e. the percentage of candidate sets correctly tagged. The classifier was trained on words derived from the LOB corpus (listed in Appendix A) and tested using restricted sentences taken from the corpus (listed in Appendix B). It was shown that as the lexicon increased (from 100 to 1000 words), the tagging performance increased for lexicons of 100 to 400 words after which it levelled at about 70% correct. Although the tagging was not perfect[2], some candidate sets were correctly reduced

[2] In fact, a HMM cannot fully represent the syntax of English, as shown by Chomsky (1955).

even when mistagged. Since many words can be used in different syntactic ways the correct word was not always removed when the candidate set was misclassified.

Candidate set reduction was reported to be about 50% (for a lexicon of 1000 words) while the error introduced was 10%, however, as the lexicon increased, the candidate set reduction decreased. This was due to the syntax model increasingly choosing the most probable tag for a candidate set and surrounding information having less contextual effect. This occurs because the candidate sets increased in size (on average larger than 10 words) and the confusion probabilities for individual words were distributed over a number of clusters (up to four clusters since there were four training images).

Results for a second order model were found to be marginally worse than those for the first order model. This was reported as being partly due to the test sentences not being statistically similar to the model used. However, it was later noted that the second order model was propagating misclassifications which were not being promoted in the first order model. The same effect of poorer second order results due to propagation of misclassification errors was found independently by Srihari (1993).

Keenan, Evett & Whitrow (1991) reported a similar statistical approach to tagging candidate sets. As in the above work, tag transition probabilities were calculated from the LOB corpus. Keenan used a 'moving window' technique to find the most probable tag for each candidate set, where adjacent candidate sets were reduced depending upon the tag transition probabilities. Keenan looked at bigram probabilities and trigram probabilities (analogous to the first and second order Markov models) and found that the trigram results were marginally better than bigram results. Results reported in (Keenan et al. 1991) showed that the trigram model was 'more stable' than the bigram model. For a test set of 372 words, 1246 candidates were generated. Following syntactic processing, the candidates were reduced to 521 for bigrams and 544 for trigrams, however bigram models resulted in 76 candidate sets without the correct word (i.e. over 20% error introduced by the syntax model) while application of the trigram model resulted in 61 candidate sets without the correct word (i.e. 16% error rate introduced).

In these experiments, the correct word was always present in the candidate set. Keenan (1992) reported a set of experiments where the correct word was removed from some of the candidate sets to simulate classifier error. In this case, the performance of the syntactic context fell considerably when errors increased above 10%[3]. However, Keenan does not report results of this experiment with the trigram model.

An alternative approach to syntactic tagging was proposed by Crowner & Hull (1991) where a second higher level of three tags could be used 'N' for noun related tags, 'V' for verb related tags and 'B' for function words and punctuation. Sentences were first 'parsed' by finding valid sequences of tags. The most probable resulting tag sequence was then used to reduce the candidate sets. Reductions of over 24% were reported on a test set of 217 words.

Srihari, Ng, Baltus & Kud (1993) looked at two methods of integrating syntax in a handwriting classifier. The first method was to model the probability of tag transitions and find the most probable sequence of tags (as described above). Her second method was to use a hybrid of syntactic hypertags and a probabilistic model.

Hypertags represent phrases within a sentence. For example, from Srihari, Ng, Baltus & Kud (1993), the sentence "The new leader of our group will be coming" is tagged as :

[**NPh1** (The **DET**)(new **JJ**)(leader **NN**)]

[**Prep** (of **IN**)(our **PP$**)(group **NN**)]

[**VPhX** (will **MD**)(be **BE**)(coming **VBG**)]

This gives two layers of syntactic information where the transitions between hypertags are modelled probabilistically. This is an extension of the model proposed by Crowner & Hull (1991) however, results for this syntax model have not been reported to date.

[3] Srihari, Ng, Baltus & Kud (1993) notes that such classification errors are around 30% for off–line systems

3.5.3 Semantic context

Semantic context aims to exploit the fact that semantically associated words appear close to each other. For example, the word ‘overdraft’ is likely to appear close to the word ‘bank’. Rose (1993) (and, Rose, Evett & Lee (1994)) measured the effect of different semantic sources of information for handwriting recognition. Rose reported three types of encoding and exploiting semantic information. The first, word frequency information, simply associates a score with a candidate depending upon the frequency of that word in a large corpus of text. This does not use surrounding information but does result in a large percentage (81%) of words being promoted to the top of the list of candidates.

A second measure used word collocations where a score is associated with a pair of words representing the strength of association between words. These scores were calculated from the Longman corpus (Summers 1991). Words were ranked according to the strength of association with the words up to four away. This resulted in 62% of words being promoted correctly with 22% tied with the same score.

Finally association ratios were used which are similar to collocation scores except that the order of the words is preserved. This was found to give only 42% correct words promoted at the expense of many words tied with the same score (51%).

The two collocation measures showed that information could be derived from modelling the way that words can appear close to each other. However, the work did not show the effect of how misclassifications in the recognition process could effect the results since the candidate sets were simulated rather than the result of classification routines.

3.5.4 U.S. Census recognition

Previous applications of language models have been applied to general recognition regardless of specific applications. Breuel (1994 b) describes a system for recognising responses to three questions on U.S. census forms relating to the respondents occupation. Consequently this involved a large number of writers, a large unconstrained lexicon and a variety of writing styles. The task was to transcribe the writing literally, rather than interpret the written response.

Recognition was achieved by segmenting the written words into letters and using an MLP classifier Breuel (1994 a). Since there were a small number of words in each response, a phrase model was used where the probability of each phrase (response) was calculated from a corpus of responses. The phrases in the corpus accounted for only 78% of the actual responses and because some of the responses only occurred once in the corpus (and so the probability was not calculated) the effective coverage of the phrase model was 64%.

A word based model was also used which gave a larger coverage of the input by storing the possible words in a dictionary. It was found that the phrase based model contributed more information than the word based model. However when these two sources of information were incorporated in a decision tree, performance was further improved giving an error rate of 6% for 1500 phrases and 50% rejection. Without any context, Breuel reports that the classification results were expected to be less than 5%.

This application is interesting because it shows that language model context can be used in a real application. However, the context used is basically dictionary lookup. In the case of the phrase model, the context is dictionary lookup of phrases augmented with the probability of that phrase occurring.

3.6 Discussion

The different types of context described in this chapter can be grouped into four categories; redundancy, dictionary, syntax and semantics. Kornai (1994) noted that language models of context cause local errors to be propagated into global errors. We can see that different types of context propagate errors to a different degree. When exploiting redundancy, classifier errors mean that either the validation fails where the two hypotheses contradict, or validation promotes an incorrect hypothesis. Errors cause a similar effect when using dictionary context; classification errors mean that either no word is found or incorrect words are taken as possible candidates.

In the two above methods of context, the errors are restricted to the words being classified. The remaining sources of context use information about other, possibly misclassified, words to reorder or reduce the candidate sets. As shown by Hanlon & Boyle (1992 a) and Srihari, Ng, Baltus & Kud (1993), errors in the recognition process corrupt the contextual information, and as the models use more surrounding information, for example, 2nd order HMM's, the results deteriorate further.

This propagation of errors could also occur with semantic information. The semantic information described by Rose et al. (1994) is based mainly on collocation information where a misclassification would cause either no information to be extracted or the wrong information to be extracted. So, for example, when reading the sentence 'I went to the bank for a mortgage' the word 'bank' may be misclassified. In such a case, there is no supporting evidence to promote the word 'mortgage'. In other situations, the misclassification could cause other, incorrect, candidates to be promoted instead of the correct word.

Despite the clear fact that errors can be introduced by such models of context when the classifier makes a mistake, there has been little work into what the effects of misclassifications upon context are. However, we can define a rough measure of 'stability' of contextual information based on the scope of the information used to generate hypotheses[4].

We define the contextual scope of a word to be those surrounding words that are affected by the identity of the current word given some method of context. So for syntactic context, the identity of one word in the sentence can affect the choice of other words in that sentence. In contrast, dictionary context has no scope beyond the current word since the identity of that word given a dictionary context model only affects that word.

As the scope of context increases, it follows that the potential for the propagation of errors increases. So for dictionary lookup there is no propagation of error since there is no scope, whereas a misclassification can cause a syntax model to propagate an error through the sentence.

[4] This is different to the stability of trigram models over bigram models mentioned by Keenan (1992)

	Context type	Scope of context	Calculation of context
Most stable	Dictionary	Single Word	Pr(word candidate | global dictionary)
	Syntax	Sentence	Pr(word candidate | surrounding syntactic structure)
	Semantics	'Close' words	Pr(word candidate | words previously recognised)
Least stable	Adaptation	Document (and further)	Pr(word candidate | words previously classified)

Table 3.1: Different types of context

Table 3.1 ranks different sources of context in order of 'stability'; note that adaptation has been included in the table. The adaptation model used by Bozinovic & Srihari (1989) used correctly identified words and adjusted classifier parameters, hence affecting the recognition of subsequent words. This can be seen in terms of context models where the classification of a word affects subsequent words within the contextual scope of that word. Similarly, misclassification of a word will cause the classifier to be erroneously adapted and affect the classification of subsequent words.

In these terms, adaptation can be seen as an additional source of contextual information, i.e. information derived from another source to help classify the current word. Given this, we can also see that the scope for propagating errors is far higher than the other sources of context and so we must develop methods for reducing the potential error propagation.

To overcome the errors we can either increase the candidate set size until the probability of the correct word being contained is within some bounds of the model (for instance, 10% for Keenan's syntax model) or a better classifier can be developed which will guarantee the correct word being in the candidate set.

Note that by increasing the candidate set size, we reduce the amount of information the context model can derive from the words in the candidate set itself. For all types of contextual knowledge, increasing the candidate set results in increased computation time (sometimes, increasing exponentially).

The approach taken in this thesis is to explore the use of the adaptive source of context and how it can be used to improve the performance of the classifier.

Chapter 4

A HIDDEN MARKOV MODEL OF SYNTAX FOR TEXT RECOGNITION

4.1 Introduction

A set of experiments is now presented which were carried out to evaluate how much a Hidden Markov Model (HMM) of English syntax could improve whole word recognition. Previous work carried out independently by Hull (1992) and Hanlon & Boyle (1992 b) showed that a HMM of syntax could be used to reduce the size of a word's candidate set given the context of the surrounding sentence. Such a probabalistic model of syntax could be built from an existing corpus of English text which was then used to find the most probable sequence of tags in a sentence

In the experiments described in this chapter, a classifier for *printed text* was implemented using features extracted from the whole word shape. From this a candidate set of similar looking words was generated. A model of syntax was then used to find the most probable sequence of tags in the sentence given the sequence of candidate sets.

4.2 Hidden Markov Model for tagging candidate sets

Markov models have been used extensively in pattern recognition where some pattern can be predicted from earlier patterns. As discussed in Chapter 2, Markov models have been used to provide contextual information in both speech processing and in handwriting recognition.

A Hidden Markov Model is a probabilistic model consisting of a state transition matrix, a confusion matrix and a set of initial probabilities. The hidden states are the features of the model which are generated by a Markov process, that is, the probability of the system being in a particular state at time t depends only on the immediately preceding states. The observations are events probabilistically related to the hidden states and are those events of the system which can be measured. Usual HMM problems include finding the probability a model generated a sequence of observations, finding the most probable sequence of hidden states given a sequence of observations and generating a HMM given a large sequence of observations.

When tagging candidate sets, the hidden states are the syntactic tags and the observations are the candidate sets. The probabilities for the tag transition matrix and initial tag probabilities are calculated from the whole LOB corpus. First order probabilities are calculated from the frequency of tag bigrams and second order probabilities from the frequency of tag trigrams. The confusion matrix probabilities, $Pr(observation|tag)$ are calculated by summing $\frac{m}{4}Pr(word|tag)$ for all tags in the cluster where $word$ appears in the cluster m times (up to a maximum of four printed words).

A set of 22 tags were used, which were logical groupings of the 134 tags used in the LOB corpus.

Thus a sentence, $S = w_1, w_2 \ldots w_n$ is represented as a list of candidate sets, $\hat{S} = c_1, c_2 \ldots c_n$, for which we wish to find the sequence of tags, $T = t_1, t_2 \ldots t_n$, which maximises :

$$\max_T [Pr(t_1)Pr(c_1|t_1) \prod_{i=2}^{n} Pr(t_i|t_{i-1})Pr(c_i|t_i)]$$

This is calculated using the Viterbi algorithm using the standard HMM approach (Forney 1973).

4.3 Data used in the experiments

The experiments used in this chapter formed the basis for the experiments described in later chapters. All of the data comes from the LOB corpus (Johansson et al. 1986) which was then filtered to extract data suitable for the experiments in this thesis.

This section describes the lexicon sizes, why an increasing lexicon size was used, the data set used from the LOB corpus and how it was extracted. The next section will then show how the words and sentences described in this section were classified.

4.3.1 Lexicon sizes

The experiments described in this thesis use a maximum lexicon size of 1000 words. The words chosen were the most frequently occuring 1000 words in the corpus and represented about 70% of the corpus.

The choice of 1000 words is small in comparision to other work which has used larger lexicons, for example Keenan (1992) and Rose (1993) used several thousand words while work reported by Bramall & Higgins (1993) used tens of thousands of words.

When the words in the LOB corpus are sorted in order of frequency there are many non–words that appear with a high frequency. This is mainly due to the method of tagging in the corpus where punctuation and apostrophised word endings such as ’s are tagged as seperate words. Because of this, the non–words were removed before taking the most frequent words. All the words used in the experiments are listed in Appendix A.

Many of the results reported here are for an increasing lexicon of 100 to 1000 words. This was to measure the effects of increasing the lexicon. These smaller lexicons were generated by taking the most frequent words from 100 to 1000 words in steps of 100 words.

4.3.2 Test sentences

The test sentences used in the experiments were also taken from the LOB corpus. These were constrained to contain only those words in the lexicon. Sentences boundries were marked in the LOB corpus however because the corpus is taken from many different sources (for example, newspapers) there were some sentences with a small number of words. Also sentences with punctuation were removed since the punctuation had been tagged while our language model did not include punctuation. The constraints on the sentences meant that many of the sentences were small clauses rather than full sentences. All the sentences used in the experiments are listed in Appendix B.

4.4 Word Recognition

Words were recognised using vector quantisation; features were extracted from a training set of words and were clustered in feature space. Unknown words were classified by extracting features and finding the closest cluster centre in feature space. The members of the closest cluster were used to generate the candidate set.

Two different sets of features were used, Fourier based descriptors and a set based on ascenders, descenders and moment based features. Further, the training set of words used to generate the clusters was printed twice to determine the influence of different noise effects. One set contained each word printed four times in a roman font, the other set contained each word printed in four different fonts.

4.4.1 Features used

4.4.1.1 Word shape features

A nine dimensional vector of word shape features consisted of three moment based features, four counts of ascender and descender and the aspect ratio of the word image.

Since words were not segmented into individual characters, a letter count was impossible to determine. Instead a relative measure of word length was given by the aspect-ratio, i.e. $length/height$.

Ascender and descender counts were taken by first splitting the word image into top, middle and bottom sections, where the top section contained any ascenders of the word and the bottom section contained any descenders. The word was then split down the vertical centre of gravity and ascenders and descenders were counted in the left and right regions.

A measure of the internal structure of the word image was calculated using moment based features. These were taken from Dudani, Breeding & Mcghee (1977) and have been used in character recognition and other applications (Kundu et al. 1989). These features are invariant over translation, rotation and scaling. The measures used were M_2, M_3 and M_4:

$$\mu_{pq} = \frac{1}{N}\sum_{i=1}^{N}(u_i - \bar{u})^p(v_i - \bar{v})^q$$

$$r = (\mu_{20} + \mu_{02})^{1/2}$$

$$M_2 = [(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2]/r^4$$

$$M_3 = [(\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2]/r^6$$

$$M_4 = [(\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2]/r^6$$

4.4.1.2 Fourier features

Fourier descriptors have been used to classify whole words by taking a two dimensional Fourier transform of the word image (O'Hair & Kabrinsky 1991), and individual characters by taking a one dimensional Fourier transform of the outline of the letter (Shridhar & Badreldin 1984). One-dimensional Fourier features were chosen to describe the shape of the word envelope.

In order to extract one dimensional Fourier features of a word image, a chain code of the word envelope must first be taken. Letters in the word were then joined to create a

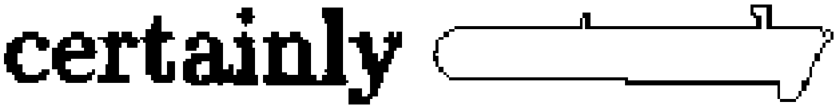


Figure 4.1: Word image and word envelope

connected region from which a chain code could be extracted. Figure 4.1 shows the word image 'certainly' and the extracted chain code.

The chain code was then represented as a set of points (x_i, y_i). The sequences $x_1, x_2 \ldots x_n$ and $y_1, y_2 \ldots y_n$ are interpreted as two one-dimensional signals, $x(m)$ and $y(m)$ where $x(L) = x(0)$ and $y(L) = y(0)$. A one-dimensional Fourier transform of each is taken and then normalised, i.e. from (Shridhar & Badreldin 1984).

$$a(n) = \frac{1}{L-1} \sum_{m=1}^{L-1} x(m) e^{in\omega_0 m}$$

$$b(n) = \frac{1}{L-1} \sum_{m=1}^{L-1} y(m) e^{in\omega_0 m}$$

These descriptors are not invariant to rotation, shift or size, so the following normalisations are done.

$$r(n) = [|a(n)|^2 + |b(n)|^2]^{1/2}$$

$$s(n) = r(n)/r(1)$$

The power spectrum showed that the most information was represented by approximately fifteen low frequency descriptors and a small number of high frequency descriptors. The shape of the word was then stored as a 15 dimensional feature vector containing the low frequency descriptors $s(2) \ldots s(16)$.

4.4.2 Clustering

The lexicon is generated with each word printed four times, which is then scanned and a set of features extracted from each word image. The resulting feature vectors are then clustered using an adapted K-means clustering algorithm based on (Zhang & Boyle 1991). The number of clusters for these experiments was defined to be $n-30$ where n is the number of words in the lexicon.

The result of the clustering algorithm is a number of clusters each containing a set of words as members. If the image acquisition and feature extraction were perfect, clustering would result with all four instances of a word in the same cluster. However, in practice, we find that noise effects cause some words to appear in more than one cluster. This has the effect of distributing the probabilities of words across different clusters and hence affects the tagging.

4.5 Results

Results were measured in three ways. The tagging performance was measured by comparing the Viterbi output with the tags in the LOB corpus. Two measures proposed and used by Hull (1992) to measure the candidate set reduction and the error rate in the tagging were also used.

Measuring the tagging performance indicates how well the Hidden Markov Model tags the candidate sets in contrast to candidate set reduction and error rate which indicates the consequence of using syntactic tags as information to constrain possible candidates.

4.5.1 Tagging performance

The simplest measure of results is to calculate the percentage of tags generated by the Viterbi algorithm which correspond to the sentences in the LOB corpus. The graph in Figure 4.2 shows the tagging performance for the first and second order experiments.

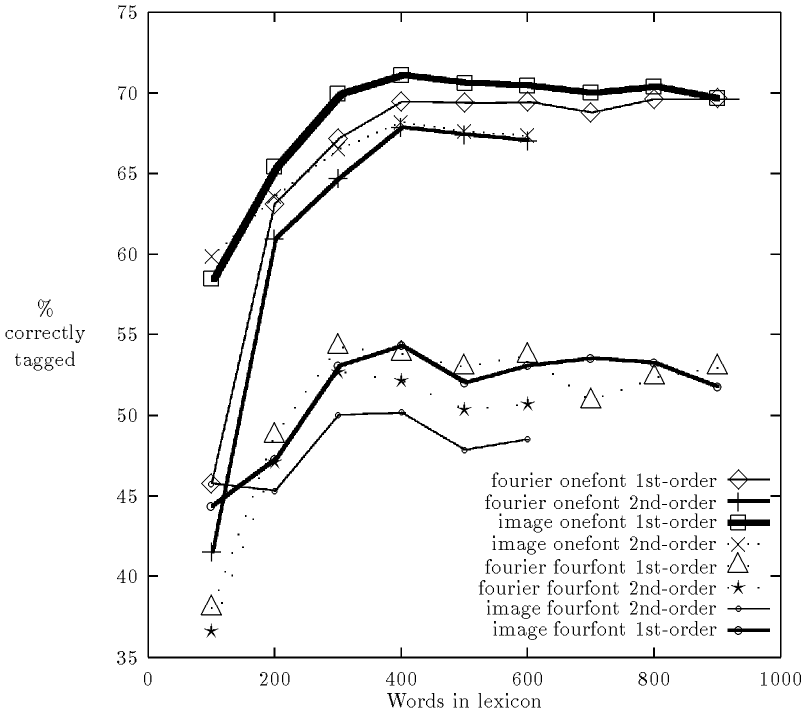


Figure 4.2: Percentage of words correctly tagged

These results show that the choice of fonts in the training set can dramatically improve performance. The system tagged significantly better when trained with one font rather than four different fonts.

Perhaps the most interesting result is that the second order results were marginally worse than the first order results. In order to find why this was so, the probability distribution of tag bigrams in the test sentences was compared to the distribution in the tag transition matrix of the HMMs. This gave a measure of how well the tag transition matrix modelled the structure of the test sentences. It was found that the structure of the test sentences were better represented by the first order transition matrix than the second order matrix. This was due mainly to the choice of test sentences; by constraining the words in the test sentences, the structure of the test sentences were found to be much different to the structure of sentences in the overall LOB corpus. Hence we would expect sentences with richer syntactic structure to be better modelled by a second-order model.

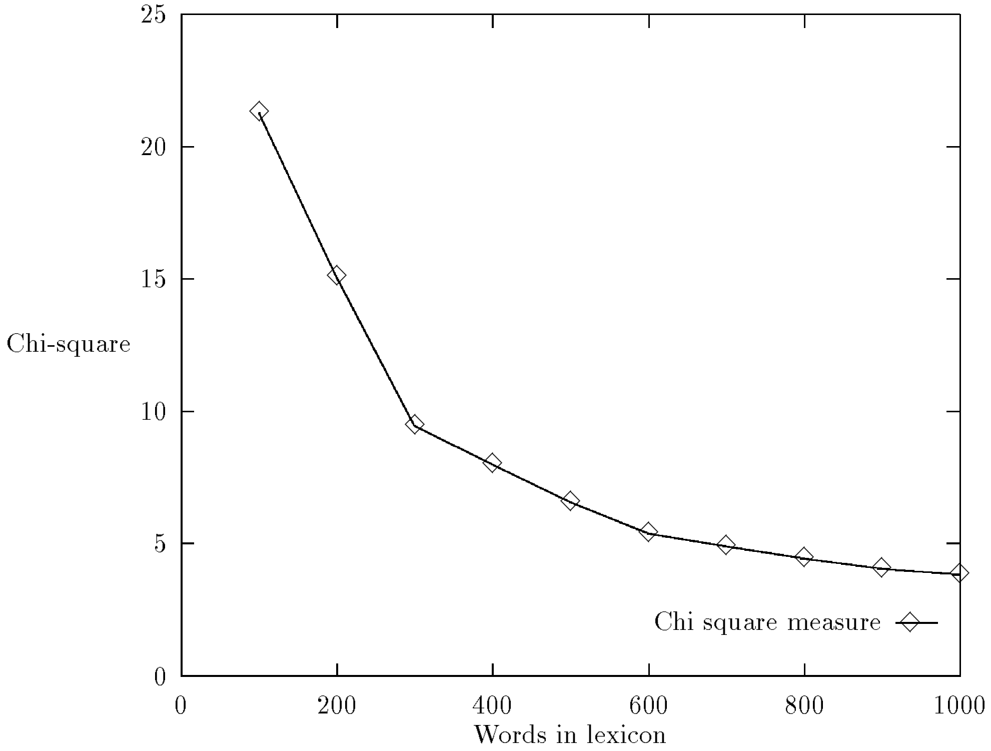


Figure 4.3: χ^2 comparison of transition matrix and test sentences

It was also found by comparing transition probability distributions that the tag transition matrix better represents sentences with a larger lexicon than those with a small lexicon. Figure 4.3 shows a χ^2 measure of the test sentences and the first order transition matrix and it shows the two distributions converging as the lexicon grows. So, the results appear to be better as the transition matrix better approximates the tag transitions in the test sentences.

4.5.2 Error rate

The error rate is defined as the percentage of candidate sets which do not contain the correct word. When tagging word images, this error occurs in classification where a word may be mis-classified, and after mis-tagging where the correct word may be removed from the candidate set. Consequently we use two error rate measures with word images, *classification error* is the percentage of candidate sets in error before tagging and *reduction*

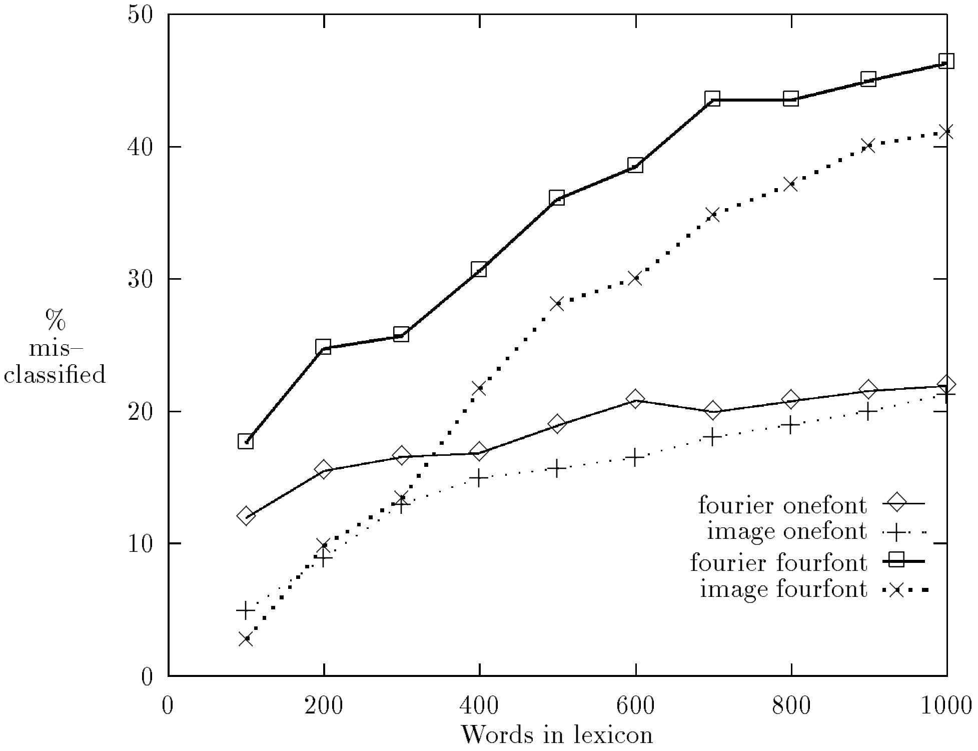


Figure 4.4: First order classification errors.

error is the additional percentage of candidate sets in error after tagging.

Figure 4.4 shows the classification errors for first order experiments while Figure 4.5 shows the reduction error rate for the first order experiments. Classification errors for second order experiments were identical and reduction errors slightly higher than those shown for first order.

The classification error shows that classification routines are introducing more errors as the number of words in the lexicon increases, reflecting the intuitive result that whole word features become less effective as the lexicon increases. If a large lexicon is to be used, a more robust set of features would have to be chosen. However, reduction errors are decreasing as the lexicon increases showing that when a word image is correctly classified, syntactic reduction of candidates introduces only a small amount of error.

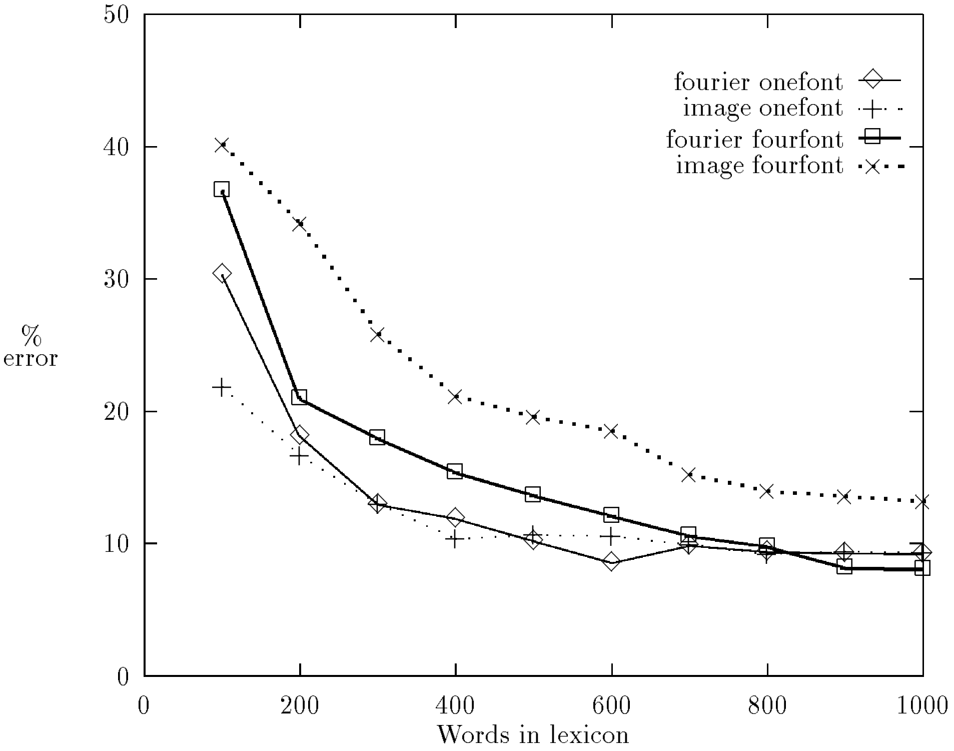


Figure 4.5: First order reduction errors.

Considering the increasing error in classifying words, the tagging performance is encouraging and shows that the HMM approach to tagging candidate sets is robust when coping with high levels of noise in the observation sequence.

4.5.3 Reduction of candidates

The percentage reduction of candidate sets shows how useful the tagging is when used in a text recognition system. This reduction is defined as the percentage reduction in the average candidate set size before and after tagging. Figure 4.6 shows the candidate set reduction for first order experiments. Second order results showed a slightly larger reduction.

These results show that the set reduction tends to become worse as the lexicon increases.

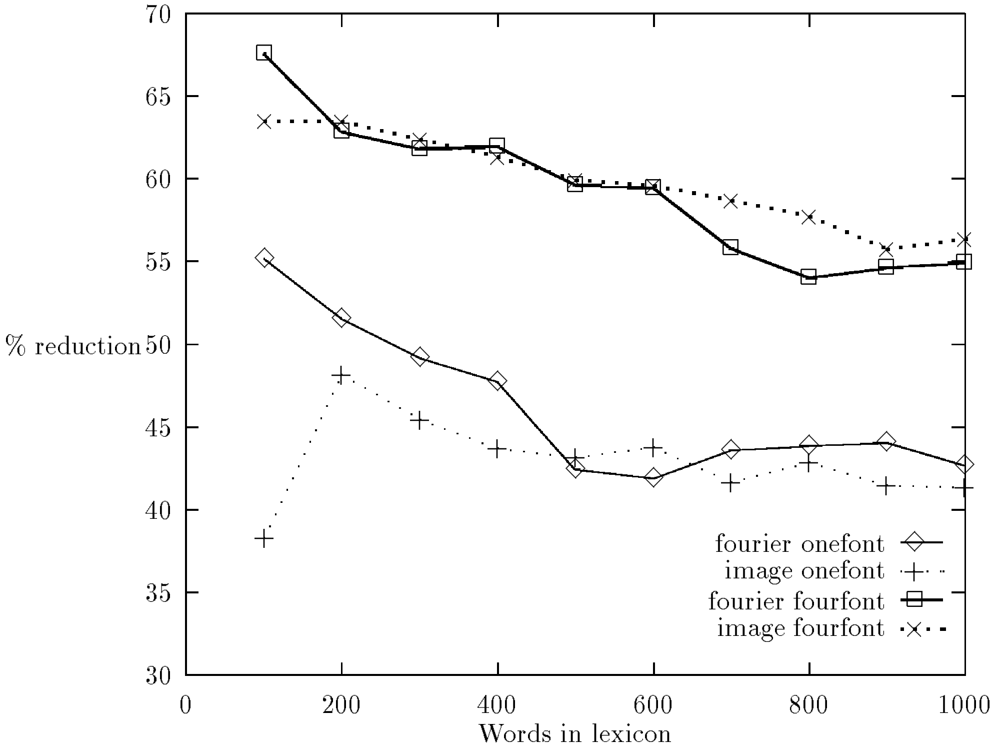


Figure 4.6: Candidate set reduction

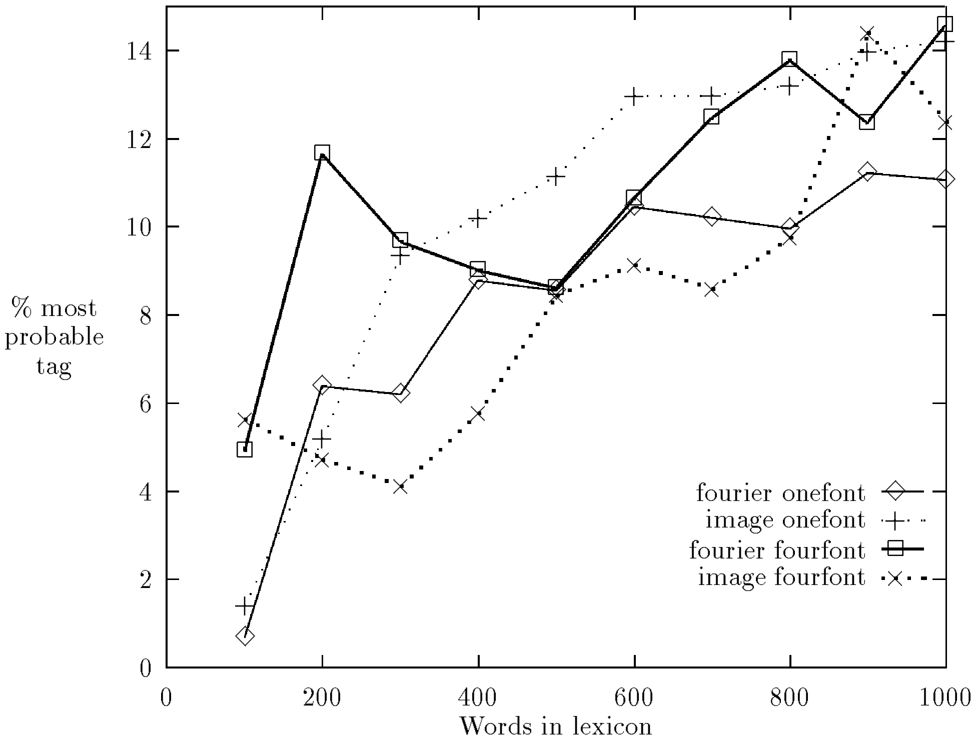


Figure 4.7: Percentage of most probable tags chosen by Viterbi algorithm

That is, the model is choosing tags which represent more candidate words and hence less words are removed. This suggests that perhaps for larger lexicons, the model chooses tags which are more probable given the observation, rather than tags which are less probable and promoted because of surrounding context.

We measure the number of times the Viterbi algorithm chooses the most probable tag given the observation, this is a crude measure of how the Viterbi algorithm is choosing the tags. The graph in Figure 4.7 shows the percentage of most probable tags chosen.

The trend shown in the graph is that the Viterbi algorithm increasingly chooses the most probable tag for an observation as the lexicon grows. This is then reflected partly in the falling candidate set reduction.

4.6 Discussion

This chapter has described a set of experiments designed to evaluate the use of a Hidden Markov Model in a text recognition system. Two classifiers based on different feature sets were tested with a number of points resulting from these tests.

The training set of the classifiers was important to the results. When trained over multiple fonts, the classifier produced many incorrect candidate sets which resulted in incorrect tagging.

Chapter 5

THE DIRECTED READING ALGORITHM

5.1 Introduction

This chapter introduces the Directed Reading (DR) algorithm which is designed to control adaptation when reading off–line single author handwritten documents. The main feature of the algorithm is a feedback loop where words and letters that are correctly classified during one iteration are used to adapt the classifier. Chapter 3 discussed how adaptation could be considered another source of context. As with all forms of context in handwriting recognition, the aim of utilising the additional sources of information is to increase the number of words correctly recognised, although this creates additional computational overhead.

The DR algorithm was originally presented by Thomas, Hanlon & Boyle (1990) where the domain of writing was handprinted computer programs with a Pascal-like structure. Keywords were restricted to a maximum of three letters and were written in block capitals. The system used a multi-layer perceptron to classify words and letters however it was restricted to being a prototype since the number of words and letters used in the experiments was small and insufficient for any training process. The algorithm described in this chapter is based on that described by Thomas with changes in the control structure and a more rigorous method of measuring the classifier results.

This chapter will introduce the idea of adaptive handwriting recognition in the domain of off-line document recognition and show that adaptation can be considered as a form of context which can be utilised in a handwriting recognition system. The design of the algorithm is then given showing how adaptation can be achieved by reading words at both the word and letter levels and by iterating over the document image. The notion of adaptation is then extended to include the unsupervised learning of new words within a constrained dictionary. The algorithm is then defined and finally approaches to measuring performance are given which will be used to evaluate the algorithm. Implementation details of the classifier and the evaluation of the system are given in the following chapters.

The algorithm expliots both whole word and segmented letter recognition to achieve adaptation and learning. Letter recognition is achieved by using a separate Fourier classifier discussed in the next chapter.

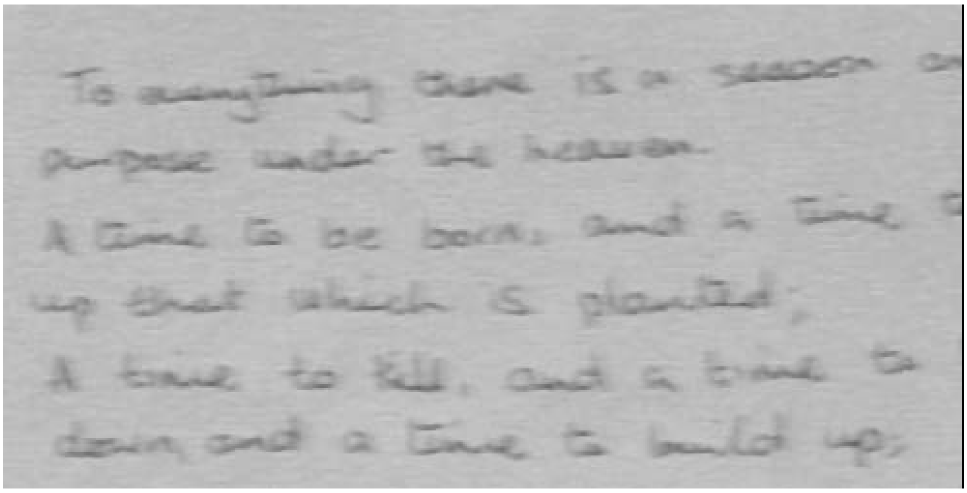
5.2 Adaptive Handwriting Recognition

Adaptive handwriting recognition is defined here to be when the classifier can change the internal representation of words, letters or graphemes to be closer to the style of writing on the document being read. Following this adaptation we expect the recognition rate to improve[1]. This can be seen in the human reading process, for example consider the piece of text in Figure 5.1. The writing style is poor and the scanning process for this document has severely deteriorated the quality of the words and letters. However, once a number of key words have been recognised it is possible to use the writing style along with other contextual knowledge to recognise the other words.

In this example we have a single document and the human reader may have to reconsult words a number of times in order to read the writing. The aim of our adaptive classifier is to analyse the document and by recognising a proportion of the words, hopefully adapt to the writing style and consequently classify the remaining words.

This concept of reconsulting parts of a document that are difficult to read is based on ob-

[1] This definition of adaptive recognition is essentially the same as that of Tappert (1984). His adaptation was achieved by the writer requesting further training of the classifier, here we intend the classifier to adapt without such supervision.



To everything there is a season a
purpose under the heaven.
A time to be born, and a time t
up that which is planted;
A time to kill, and a time to
down, and a time to build up,

Figure 5.1: A poor quality image of handwriting

servation rather than psychological evidence[2]. However, Frazier & Rayner (1982) showed a similar effect with the recognition of garden–path sentences where the reader will re–read a sentence if the grammatical structure is unusual (for example, when reading the sentence *the horse ran past the barn fell*).

Here it is observed that the reader reconsults the words to make sense of the sentence. The choice of words to reconsult depends upon the readers understanding of the sentence and the hypotheses being tested by reconsulting the words. This process of reading a word twice is analogous to the models of reading and context proposed by Hull (1988) and Cohen (1994).

At the lower level of word recognition rather than word interpretation, the words that have to be reconsulted are those which have not yet been recognised. By adapting the classifier to the style of writing in words that have been correctly classified, we aim to recognise the remaining words. This gives the basic structure of the algorithm, i.e. to recognise as many words as possible, then to adapt the classifier given the words that have been read and then reconsult the unclassified words.

The classifier used by Bozinovic & Srihari (1989) was shown to improve when adapted to

[2] I am not aware of psychological work into the recognition of poor handwriting.

writing style. Other systems have used adaptive techniques to improve recognition rates. Verikas, Bachauskene, Vilunas & Skaisgiris (1992) describe an adaptive text recognition system which uses a hierarchical method of classification. The aim being to improve recognition of degraded images of text. However, as well as being restricted to printed characters, the adaptation only occurs during the training process which is unsuitable if we are to apply the technique to handwriting recognition where the adaptation would be a continuous process.

Tappert (1984) and Schomaker et al. (1994) have applied adaptive techniques to the recognition of on–line handwriting. Tappert's system used elastic matching techniques for recognising characters (Tappert 1982) where each character had a prototype. The classifier is initially trained by presenting the writer with a training sequence of words which is entered. Adaptation occurs when the writer chooses to further train the classifier. In this case, the writer is prompted to enter more 'updating text' which is added to the previous training data. The increase in performance between trained and updated classifiers is not reported, although recognition rates for the updated classifier are about 94%.

In this application, adaptation is basically increasing the training set for the classifier. By prompting the writer to enter more words, the system knows which prototypes to adjust. For off–line recognition this prompting is not possible since there is no interaction. However, we note that this application highlights the use of adaptation to extend the training of the classifier.

Schomaker et al. (1994) describes an existing on–line recognition system which is author independent. It is noted that there is much information in an author independent classifier which is irrelevant when recognising the writing of a single person. Schomaker showed that by extracting strokes from different writers, the strokes could then be clustered depending upon the writer.

5.3 Adaptation Through Reading Words and Letters

The algorithm will iterate over a document repeatedly trying to classify words that have not been resolved. If the classifier were restricted just to recognising whole word shapes, it would severely restrict the amount of adaptation possible in the classifier. Following the first iteration, a number of words will have been confidently classified which are then used to adapt the whole word classifier[3]. In the following iteration, the remaining unclassified words are presented to the modified classifier. However, only those words which had been correctly identified in the first iteration would have been modified in the classifier, so subsequent iterations would only recognise this subset of words. So we have a situation where the classifier will only read and adapt words that are successfully classified in the first iteration.

The solution is to use information from another source, in this case it can either be context or another classifier. If context is used, candidate sets could be reduced until they contained only one word. However, the words that can be used in adaptation would still be constrained to be those words that have been recognised confidently in the first iteration, or those that have been inferred through use of contextual constraints. In contrast, if other words are recognised using a different classifier it would allow different types of words to be recognised.

The two classifiers could then reinforce each other by each acting as a trainer for the other, i.e. when the first classifier recognises a word, it is used to adapt the second classifier and vice-versa. We choose to exploit the two types of word recognition described in Chapter 2; global recognition of the whole word shape and analytical recognition by segmenting and classifying the individual letters.

Words that are recognised as whole symbols can be segmented and used to adapt the letter classifier and words recognised at the letter level can be used to adapt the whole word classifier. This is controlled by restricting the recognition of the document to one level for each iteration.

[3] Assuming that the classifier is not perfect, this set of words is a subset of the whole document.

As shown in Chapter 2, whole word classifiers are generally constrained to recognise the words that it has been trained on. In contrast, letter level classifiers are not constrained to a dictionary since they can (in theory) recognise any sequence of characters. In practice, letter level classifiers are constrained to recognise words in some dictionary, although that dictionary may be many thousands of words. Using two classifiers means that the system has two lexicons, one for the whole word classifier and another far larger lexicon for the letter level classifier.

If we assume that the valid words for the letter level classifier includes all words in the whole word classifier, we can now define another feature of the algorithm, that of *learning* words at the letter level. The whole word classifier has a lexicon L_W while the letter level classifier can recognise words in the global lexicon L_G. When a word is correctly classified at the letter level it may, or may not be, in the whole word lexicon. If the word is a member of L_W, then that word can be adapted in the whole word classifier. However, if the word is not in L_W, then we can add this new word to the whole word classifier and proceed to train this word with each new occurrence of it in the text.

To summarise, the use of word and letter classifiers facilitates adaptation of both words and letters as each classifier teaches the other. It also allows the whole word classifier to learn new words following recognition at the letter level.

5.4 Incorporating other sources of context

At the end of an iteration, each word will have a candidate set in one of three states :

1. **A single unique classification :** where only one word has been classified confidently
2. **A set of word candidates :** where a number of words are confidently classified for a word image
3. **No classification :** where no words are confidently classified for the word image

Words with a single confident candidate are used for adapting the classifier, while words with candidate sets are used to generate hypotheses about the sentence. Here, surrounding contextual information can be used to reorder the words in the candidate sets. The aim of using context would be to promote the correct word to a sufficiently high probability so that it could be considered a unique confident classification and subsequently used to adapt the classifiers.

Context may either reduce or reorder candidate sets. In the DR algorithm, adaptation only takes place when a single word has been classified, in which case the aim of any contextual knowledge used by the DR algorithm should be to reduce the candidate set to one word. Simply reordering the candidate set would not affect the adaptation and so would not alter classifications in subsequent iterations.

Another point to note is that some contextual methods make hypotheses using two or more words or candidate sets, here an assumption of independence between the classifications is made which constrains when the classifier can be adapted. In the original DR algorithm, words were adapted as soon as they had been correctly classified, however, if this classifier is adapted in this manner, different words could have candidate sets generated from different classifiers. Candidate sets could then be dependent on earlier classifications during the same iteration. Because of this, all adaptation is done after context has been used meaning that all classifications (and hence all candidate sets) were made with one classifier.

5.5 The Directed Reading Algorithm

Figure 5.2 below gives the Directed Reading algorithm.

The algorithm starts by looking at whole words (line 1) and then iterates over a

classify (line 2.1) / context (2.2) / adapt (2.3)

loop until as many words as possible have been read.

```
1         recognition level = whole words
2         repeat
2.1          classify all unknown words at current recognition level
2.2          apply contextual constraints on candidate sets
2.3          if (any new unique classifications) then
2.3.1           use whole word images to adapt word classifier
2.3.2           use segmented letter images to adapt letter classifier
2.3.3        endif
2.4          if (need to change recognition level) then
2.4.1           toggle recognition level
2.4.2        endif
3         until (no more classifications at word level
          and no classifications at letter level)
```

Figure 5.2: The Directed Reading Algorithm

The choice for changing the level of classification is done following adaptation. If words have been classified then the letter and word classifiers will have been adapted in step 2.3. In this case, the new classifiers are used to recognise the words at the same level. If no words were recognised then the level of classification is changed.

Termination of the algorithm occurs when there have been no classifications at either the word or letter levels. In this case, no adaptation has taken place and the classifiers remain unchanged. In this case the classifiers will remain unchanged over subsequent iterations and no more words will be recognised. As soon as this situation occurs, the algorithm terminates.

5.6 Merging results from word and letter classifiers

The problem of combining classifier results is an important consideration as more systems use multiple classifiers when recognising writing. Different approaches include Ho, Hull & Srihari (1994) who merge results depending upon the classifier itself and Fairhurst &

Cowley (1993) who pipeline character classifiers.

There are three different types of candidate set that can be generated from the two classifiers used in the DR algorithm resulting in nine possible combinations of candidate set, unique classification or no classification. Since the candidate sets generated at the word level would be of a different nature to the candidate sets generated from letter level classification, a statistical analysis of the classification results would be necessary before choosing a combination scheme.

However, it must be noted that any combination scheme would only be useful if it reduced the overall number of words in the candidate sets; and unless used in conjunction with some other form of context, would only be useful if it reduced the number of candidates to one. The reason is the same as that for context having to reduce the number of candidates. If the result of combining the results gives another candidate set, that word will not be used in adaptation and will be reclassified during the next iteration of the algorithm.

5.7 Evaluating the Algorithm

The aim of using any contextual information is to increase the number of words correctly recognised. In the case of the DR algorithm, increases in recognition rates would be achieved during successive iterations of the algorithm. So, analysing the effect of the DR algorithm can only be achieved by measuring the results after each iteration.

As mentioned earlier, the number of words in the global lexicon (L_G) and the word classifier (L_W) determines what kind of learning takes place in the classifier. When $L_G = L_W$, no new learning of words can be achieved since the word classifier has been trained on all words. Increases in the recognition rates would be through adaptation alone.

When the lexicon, L_W is a subset of L_G, successful recognition causes both adaptation and possible learning (if the word is unknown and is recognised at the letter level). The effect of learning (as compared to adaptation) can be gauged by measuring the number of words recognised at the word level which had not been in the initial word classifier lexicon. In this case, the words will have been recognised at the letter level and used to train the whole word classifier. The whole word classifier could subsequently recognise other words as symbols.

The aim of evaluating the algorithm is to measure the effects of adaptation and learning at the end of each iteration. By controlling the words in L_G and L_W, the effects on each can be determined.

5.8 Summary

This chapter introduced the design of the Directed Reading (DR) algorithm. It showed that adaptation could be considered in the same way as other forms of context and that a feedback loop implementing context was an extension of other computational models of context. It was then shown how adaptation could be extended by reading words and letters allowing words to be adapted and new words to be learned (within a constrained dictionary). Before the algorithm was presented in full, the use of other contextual information was discussed where it was noted that the overall effect of context within the DR algorithm should be to reduce the candidates rather than reorder them. Similarly the effect of merging whole words and letter classifications should be to reduce rather than reorder the candidate sets.

Chapter 6

CLASSIFICATION

6.1 Introduction

The previous chapter described the Directed Reading algorithm. The algorithm iterates over an image of handwritten text modifying its representation of individual words and letters as confident classifications are made. This chapter now presents the classifier used in implementing the algorithm on binary images of handwritten words.

Traditional statistical classification of handwriting relies upon using feature spotting techniques (for example extracting loops, ascenders, descenders and cusps in word images), however such features are based upon finding invariant features in handwriting and relying upon their invariance to recognise multi–author writing. In contrast, the DR algorithm is designed to recognise a single author's writing, and hence the features that will adapt cannot be chosen a priori.

Rather than choosing a set of heuristic features to spot in word and letter images, we use the Discrete Fourier Transform of the word images to generate a feature vector. This chapter presents the Fourier transform (section 5.2) and describes how O'Hair & Kabrinsky (1991) used the low harmonics for text recognition (5.3). It then shows how the classifier was modified to read handwritten words (5.4) and how it was further modified for inclusion in the DR algorithm by addition of confidence levels (5.5). Finally the classifier is tested in section 5.6, and the effects of adding new words to the classifier are discussed in section 5.7.

The classifier described in this chapter is used in the DR algorithm for recognition of both word and letter images. While the Fourier classifier was used, other classifiers for letter recognition, such n–tuple classifiers could have been expoited.

6.2 Describing shapes in Fourier Space

The Fourier transform is based on the principle that a signal can be described in terms of a sequence of simple periodic patterns. The transform itself decomposes the signal into the periodic patterns while the inverse transform maps the patterns back to the original signal[1].

The two–dimensional Fourier transform and it's inverse are given below :

$$\begin{aligned} F(u,v) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-2\pi i(xu+yv)}dx\,dy \\ f(x,y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{2\pi i(xu+yv)}du\,dv \end{aligned}$$

The function $f(x,y)$ is a continuous two–dimensional signal, $F(u,v)$ is the *harmonic function* with spatial frequencies u and v. When processing images the signal is not continuous and so the Discrete Fourier Transform (DFT) can be used, i.e. for an image $f(m,n)$ with MxN pixels :

$$\begin{aligned} F(u,v) &= \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)exp[-2\pi i(\frac{mu}{M}+\frac{nv}{N})] \\ f(m,n) &= \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)exp[2\pi i(\frac{mu}{M}+\frac{nv}{N})] \end{aligned}$$

There are a number of properties associated with the Fourier transform; however, one of the most useful properties for pattern recognition is that gross features of the signal can be described using a small number of low frequency harmonics, i.e. $F(u,v)$ where both u and v are less than some low value.

[1] An introduction to the Fourier Transform can be found in Gonzalez & Wintz (1987), along with many other introductions to image processing.

The one–dimensional Fourier transform has been used to recognise characters. This is done by extracting the chain code of the character which can then be converted into a one–dimensional signal to which the Fourier transform is applied.

Shridhar and Badreldin (1984) showed that the low frequency harmonics can be normalised so that the transform is invariant to scale translation and rotation. A feature vector can then be generated by taking 10 – 15 normalised low frequency harmonics and classification achieved using nearest neighbour techniques.

6.3 Classification of text using 2-D Fourier transforms

As described in the previous section, low Fourier harmonics of a 2D image represent the gross features of the image. O'Hair and Kabrinsky (O'Hair, 1991) used this principle for recognition of printed words as whole symbols.

Their algorithm took an image of text and calculated the 2D Fourier transform of the raw image. From this, variations of low Fourier harmonics were extracted and used to form a feature vector, which could then be manipulated using standard nearest neighbour feature space methods. Words were represented by centroids (exemplars) calculated as the mean position of the training words in feature space, and classification was achieved by extracting a Fourier feature and finding the closest word exemplar.

O'Hair's work showed that Fourier features could recognise text with a reasonable amount of robustness – results for different permutations of harmonics to generate the feature vector were all above 95% for lexicons of 1000-5000 words[2]. However these results used noiseless images of text and the supporting experiments for noisy images only added random white pixels to images of text rather than also adding random black pixels.

The Fourier classifier was chosen for further development for the following reasons :

[2] O'Hair tested various feature vector lengths, but found that combinations of feature vectors comprising of more than 5 harmonics in both horizontal and vertical dimensions gave results higher than 95%.

1. The classifier treated words as whole symbols rather than splitting the words into individual letters.

2. Classification was achieved without selective feature extraction. As noted earlier, full adaptation in a feature spotting classifier can only be achieved if all possible features are known *a priori*, in the case of the Fourier classifier, global word shape features are captured in the low Fourier harmonics.

3. In the basic algorithm described by O'Hair, each word is represented by a single point in feature space (the centroid). This enabled adaptation of word shapes to be implemented as moving a single point in feature space. This is described in more detail in section 5.5.

4. At the time of choosing a classifier for the words, there were few whole word classifiers in the literature. Those that did exist relied upon feature spotting techniques. More recent developments using Hidden Markov Models to describe the shape of a word could possibly solve the problem of holistic classification[3].

6.4 Extending the Fourier classifier for handwritten words

This section describes the Fourier classifier used for recognising words and letters in the DR algorithm. It describes the preprocessing of word images, calculation of centroids and classification. Experiments to determine the type of normalisation and feature vector size are also presented; accordingly, the section starts with a description of the training and test images used while developing the classifier.

6.4.1 Training and test data

The experiments pursued in this chapter were to evaluate the classifier with a lexicon of up to 1000 words. The lexicon was taken from the LOB corpus (Johansson, 1986) by calculating

[3] HMM word recognition is reviewed in Chapter 2

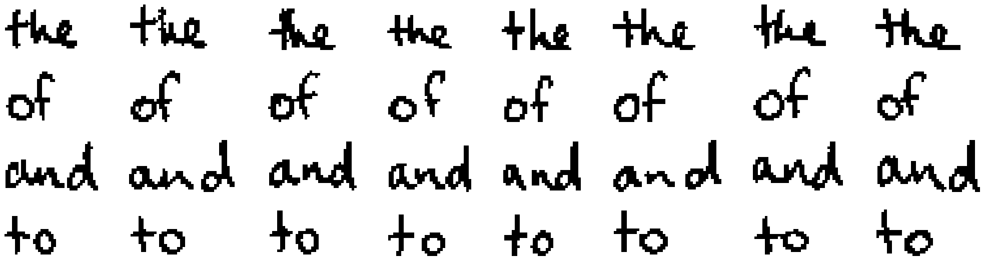

the the the the the the the the
of of of of of of of of
and and and and and and and and
to to to to to to to to

Figure 6.1: Training images of the first four words in the lexicon

the most frequent words and then removing non–words, for example, punctuation was removed since it was tagged in the corpus as a whole word and digits were also removed. The most frequent 1000 of the remaining words were extracted and used throughout the work described in this thesis.

The words were then each written ten times using a black biro on white paper. The pages were then scanned on a flat–bed scanner at 75 dots per inch (dpi). Each word was segmented from the images of the pages and each word image was then truthed. The choice of 75dpi was made due to disk space constraints at the start of this work. The common scanning resolution is about 300dpi as used in the CEDAR database of postal addresses (Hull, 1994) and other comparable work in this area (e.g. Senior & Fallside (1993 a)[4].

Figure 6.1 shows a sample of the first few words in the training set. During segmentation of the page images containing the word images, some top and bottom lines were either corrupted through noise or were not segmented correctly. This software problem meant that a number of word classes were missing from the final database of word images. Since the original set of words had been written in one session, it was thought better to use the words from the original session with a number of words missing rather than write the missing words and have a set of images written over two sessions. In total 21 word classes were missed from the training set these were :

all, almost, does, has, later, letters, men, number, or, place, problems, process, produced,

[4] It is difficult to predict how much the results would change if the experiments were repeated using a higher scanning resolution

production, programme, quite, something, their, thought, told, until.

Some of the results in this chapter describe the lexicon being increased from 100 to 1000 words. In these cases, the number of words actually used in the different sections were :

Lexicon size	Number of words used
100	96
200	185
300	285
400	385
500	484
600	584
700	683
800	782
900	879
1000	979

Table 6.1: Lexicon sizes in reported experiments

For each word, one word image was used as a test image and the remaining 8 or 9 word images were used as training images[5]. The ratio of 9 to 1 training to test images was not ideal, more images were used for training in order to generalise the classifier as much as possible. Given a larger database of word images an equal ratio of testing and training images would have been used.

6.4.2 Image preprocessing

Once the words and letters have been segmented, the images were normalised before the features were extracted. This process ensures that the images are presented to the feature extraction process in a uniform manner, thus improving the chance of different instances of the same word having the same, or similar, feature sets.

The normalisation technique was to scale the word or letter image to a predefined size. A set of experiments was carried out to determine the best size for the image normalisation.

[5] Some of the words were written nine times rather than ten and some words were lost in the segmentation process.

Three different height and width combinations were tested, these were :

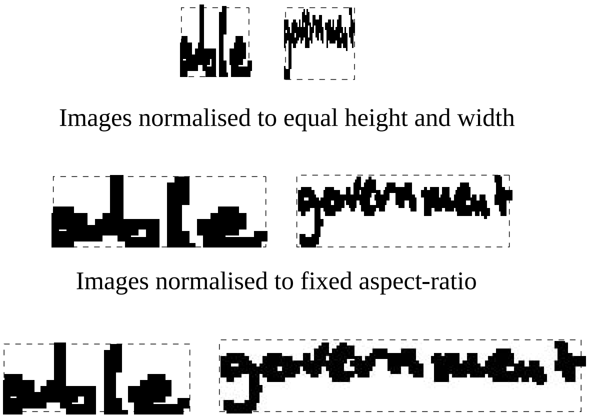
1. Same height and width, with all images scaled into a 48x48 image. This was expected to give poor results for long words since much of the horizontal information is lost by shrinking the image along the X axis.
2. Fixed height and width. Here the length of the word image along the X axis was set to three times the height of the image. All images were normalised into a 48x144 image. This size was chosen since the average height of the images was 15.92 pixels while the average length of the images was 44.59 pixels so the normalised size preserved the average aspect–ratio.
3. Fixed height and variable width. In this case, the length of the image in the X axis was variable while the height of the image was constrained to be 48 pixels. This meant that the word image was not corrupted in either the vertical or horizontal directions.

The height of word images was set to 48 pixels when testing the different normalisation techniques because of the 16x16 low Fourier harmonics extracted later from the Fourier transform. In the case of narrow words (such as 'i' and 'a') the normalised word had to be at least 16 pixels wide for the feature vector to be extracted: making the length three times the height of the feature vector was found to satisfy this condition.

Figure 6.2 shows the three types of normalisation for the words 'able' and 'government'. For all three techniques, the word 'able' can be recognised; however, the word 'government' becomes illegible to the human observer when normalised into a square.

A classifier using 16x16 Fourier harmonics as features and the above three normalisation techniques were used. Each word was represented by a centroid, calculated as the average position in feature space of the training feature vectors. Classification was carried out by finding the single closest centroid to the test word feature vector.

The classifiers were tested on all 979 test words. The results of the different normalisation techniques were :



Images normalised to equal height and width

Images normalised to fixed aspect-ratio

Images normalised to variable aspect-ratio

Figure 6.2: Normalised words ‘able’ and ‘government’

Normalisation technique	Percentage of words correct
Equal height and width	82%
Width = 3*height	82%
Variable aspect-ratio	80%

Table 6.2: Classification results with different normalisation techniques

The results show that keeping the aspect–ratio the same gives marginally worse results than normalising into a square or rectangle. One of the problems with normalising a word while preserving the aspect–ratio comes from the nature of the features being extracted from the image. Figure 6.3 shows the DFT power spectrum of the words ‘able’ and ‘government’ when the aspect–ratio is preserved. Figure 6.4 shows the extracted feature vectors from these images. The gross word shape information is in the low harmonics of the transform (at the centre of the graphs), this information is spread across approximately 400 low harmonics for the word ‘able’ and across over 900 harmonics for the word ‘government’. This meant that taking a fixed number of harmonics to generate a feature vector leads to information being lost for long words.

The aspect–ratio of words tends to be about the same for different instances of the same word. However it is rarely exactly the same, which means that the X axis information is different for different instances of the same word.

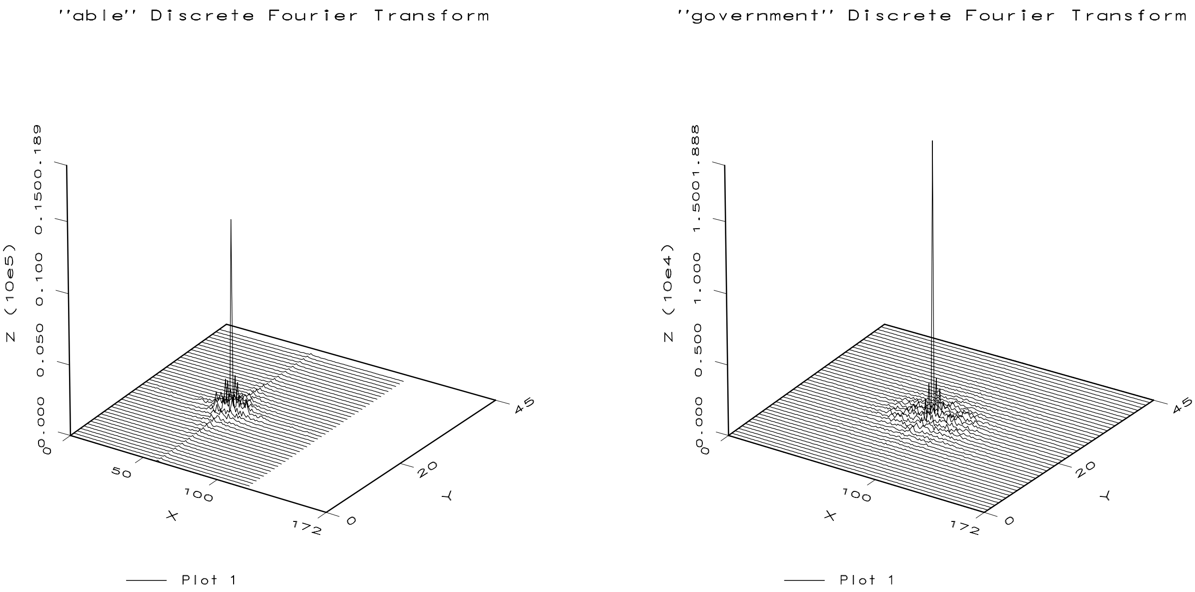


Figure 6.3: Discrete Fourier Transforms of 'able' and 'government'

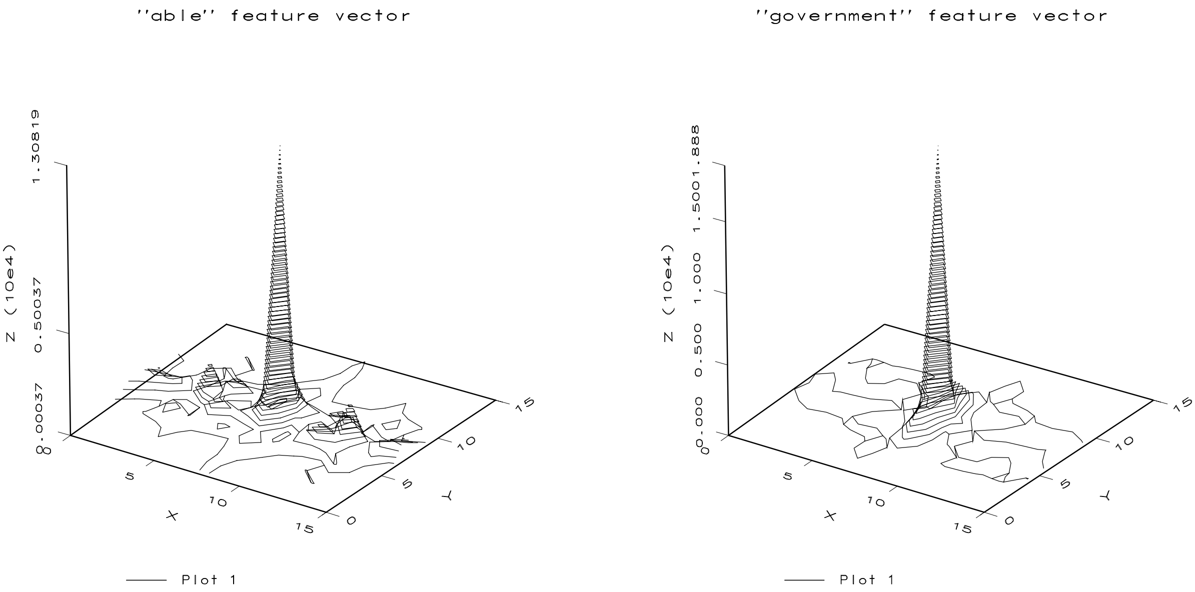


Figure 6.4: Extracted feature vectors of 'able' and 'government'

It can be seen from Table 6.2 that normalising word images into a fixed size produced the best results. However, although normalising into a square and a rectangle produced the same results with this lexicon, larger lexicons would contain longer words which would become corrupted when normalising into a square image. Since future experiments were aimed at using larger lexicons, a normalised image where the width was three times the height was chosen (i.e. 144x48 pixels).

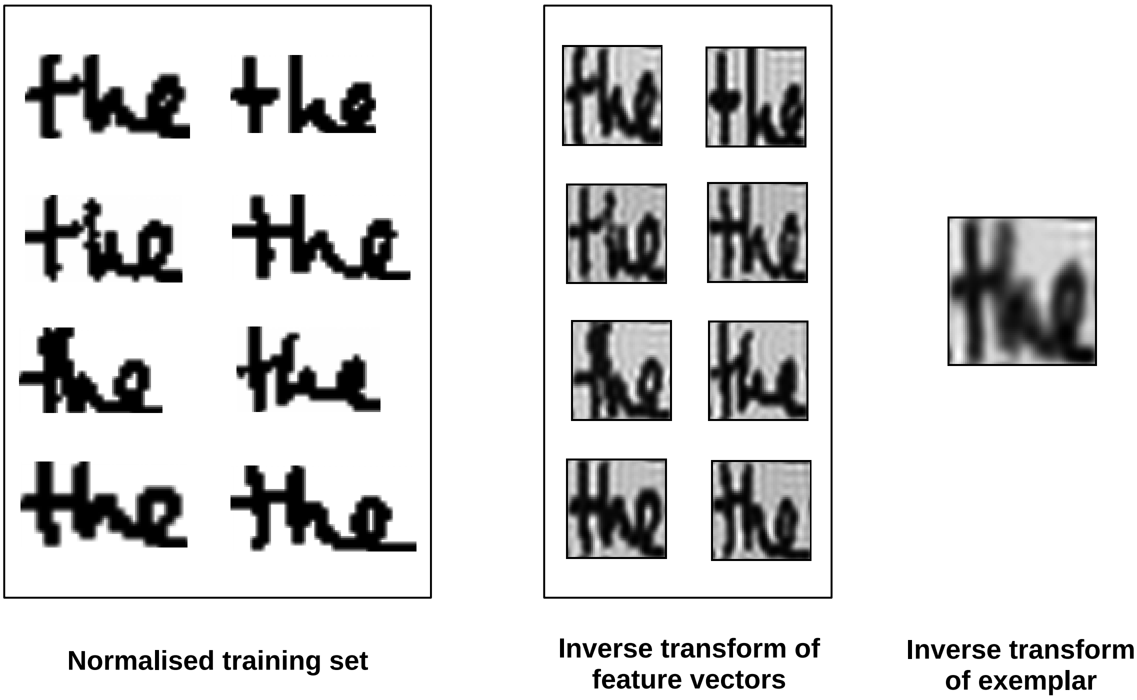
6.4.3 Extracting the Feature vector

As shown in the previous section, once the word or letter image has been normalised, the Discrete Fourier Transform (DFT) of the image was calculated. The DFT was used rather than the Fast Fourier Transform (FFT) because the FFT requires the input image to be square with sides of length 2^n and the normalised images used were not square. Figure 6.3 showed the power spectrum of the transforms of the words ‘able’ and ‘government’ in Figure 6.2. The transforms were normalised so that the low frequency harmonics were in the centre of the image.

O’Hair showed that the low harmonic Fourier coefficients could be used as feature vectors. In the previous section the feature vector was chosen to be the 16x16 low frequency harmonics. A classifier was tested that used the 8x8 low Fourier harmonics rather than 16x16; it was found that this reduced the correct classification rate by over 3%. The size of the feature vector therefore remained at 16x16 low Fourier harmonics.

6.4.4 Calculating centroids

Following O’Hair’s classifier, the exemplar for each word was chosen to be the centroid of the training feature vectors, i.e. the mean position of the feature vectors in feature space. Since the exemplars are points in Fourier space, the inverse Fourier transform can be taken giving a visual representation of the average of the training images. This can be seen in Figure 6.5 where the training images for the word ‘the’ are shown along with the inverse



Normalised training set

Inverse transform of feature vectors

Inverse transform of exemplar

Figure 6.5: Training set, Features and exemplar for the word “the”

transform of their feature vectors and the inverse transforms of the exemplar.

The ability to take the inverse transform of the exemplar can be especially useful when the DR algorithm starts to manipulate the position of exemplars in feature space, since a modified feature vector can be viewed in image space.

6.5 Classification using Fourier features

The previous sections described how the whole word classifier was used for recognition of handwritten words. This section further develops the classification of a new word. The DR algorithm only uses confident classifications to adapt the classifier and so the classifier must either return a measure of the confidence, or only return confident classifications.

6.5.1 Confidence levels

When classifying a new word or letter, the feature vector is calculated and the distance from each exemplar in Euclidean space can be calculated. However we are interested in finding the closest *confident* exemplars for a particular exemplar.

The spread of the training points around the exemplar can be measured by calculating the mean distance of points from the exemplar. So for a set of training points, $\underline{f_c^1}, \underline{f_c^2}, ..., \underline{f_c^n}$ the mean Euclidean distance from the exemplar, E_c, is :

$$\mu_c = \frac{1}{n}\sum_{i=1}^{n} d(\underline{f_c^i}, E_c)$$

and the variance of the distances is :

$$\sigma_c = \frac{1}{n}\sum_{i=1}^{n}(d(\underline{f_c^i}, E_c) - \mu_c)^2$$

For an arbitrary training vector, $\underline{f_c^i}$ from class c, if we define

$$\xi = \frac{d(\underline{f_c^i}, E_c) - \mu_c}{\sigma_c}$$

it is found that few training feature vectors have a value of ξ greater than 2.0. Figure 6.6 shows the frequency distribution of ξ.

When classifying new words, the value ξ (the number of standard deviations from the mean) was used as the confidence measure. Words where ξ is less than 2.0 were initially considered confident while those where ξ was larger than 2.0 were not.

6.5.2 Generating candidate sets

Candidate sets were generated from the set of close and confident centroids with respect to the unknown feature vector. To determine the effect of the confidence measure, a classifier was generated with the mean and standard deviation stored for each centroid. Classification of words was achieved by determining the 10 closest centroids and then removing

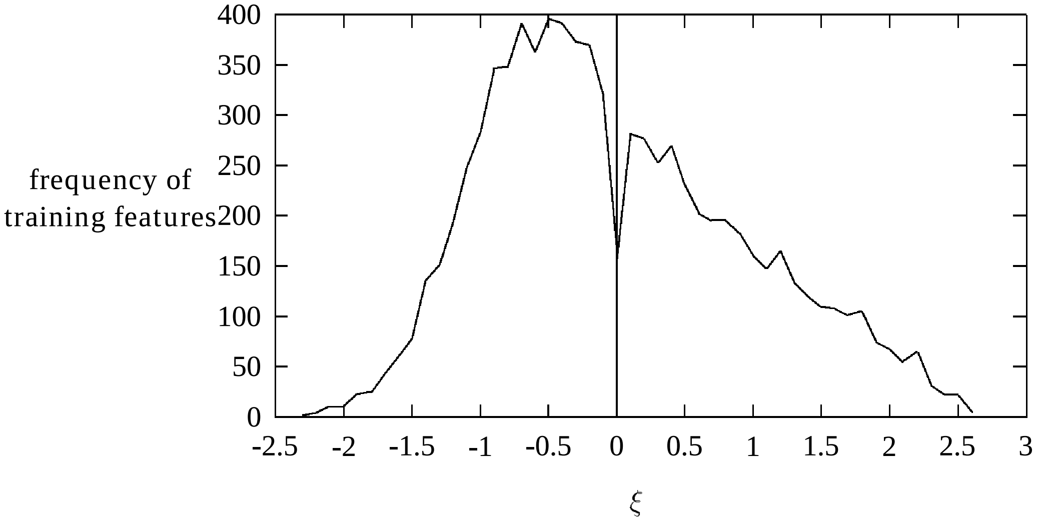


Figure 6.6: Plot of ξ for all training word images

Number of words	979	
Unique and correct	144	15%
Candidate set containing correct word	580	59%
No classification (reject)	72	7%
Unique and incorrect	71	7%
Candidate set not containing correct word	112	12%

Table 6.3: Recognition rates with $\xi = 2.0$

those classes that were not confident. This classifier was then tested on the full lexicon of 979 words. The results are given in Table 6.3.

The DR algorithm will only ever use unique and correct classifications in the adaptation, so it can be seen that the effective classification results have been significantly reduced following the introduction of the confidence level. Analysis of the 580 correct candidate sets showed that the correct word was at the top of 270 candidate sets. This indicated that there was possibly scope for improvement with a different confidence threshold, so the experiment was repeated with an increasing confidence measure, from $\xi(w) < 0.0$ to $\xi(w) < 2.0$ in steps of 0.5.

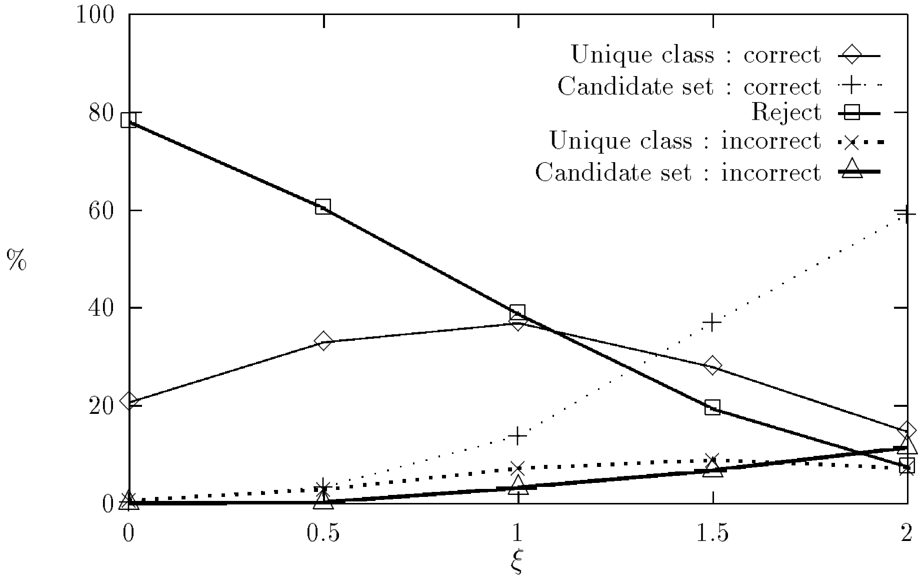


Figure 6.7: Results for increasing confidence threshold

Figure 6.7 shows the classification results with an increasing confidence threshold. As the confidence threshold increases the main features of the graph are :

1. The percentage of words with no classification drops,
2. The percentage of candidate sets containing the correct word increases, and
3. The percentage of words with a single classification peaks at $\xi = 1.0$.

The aim of choosing the confidence threshold was to maximise the number of correct classifications and minimise the error rate, i.e. reduce the number of classifications resulting in a unique incorrect result or a candidate set that does not contain the correct word. From this, a new confidence threshold of $\xi = 1.0$ was chosen.

6.6 Evaluating the Fourier classifier

This section describes experiments evaluating the performance of the Fourier classifier under two different training conditions. In the first, the classifier was trained with all but one of the training images. The remaining image for each word was used to test the classifier. So in the first case, the classifier had been trained and tested with words from the same lexicon.

It was also necessary to measure the classifier's performance when presented with words that were not in the classifier lexicon, since the DR algorithm can have a global lexicon larger than the word classifier lexicon. In the second set of experiments, the classifier is trained on one set of words and tested with a different set of words.

6.6.1 Experiment One

The first set of experiments measured the classifier performance when it had been trained on all words in the test set. In this case, the classification performance is being measured. As in the experiments in the previous section, nine word images were used for training and the remaining image was used in the test set, also the same three sets of results are of interest :

- When no classification is made,
- Unique classifications (correct and incorrect), since this is an indication of the words that the DR algorithm would use in the adaptation
- Candidate set classifications (correct and incorrect). Measuring the number of candidate shows where other sources of contextual information could be utilised to increase the number of correct classifications by reducing the candidate set size.

The classifier was tested with a lexicon of the most frequent 100 words in the LOB corpus. This was then increased in steps of 100 words and finally tested with a lexicon of 1000

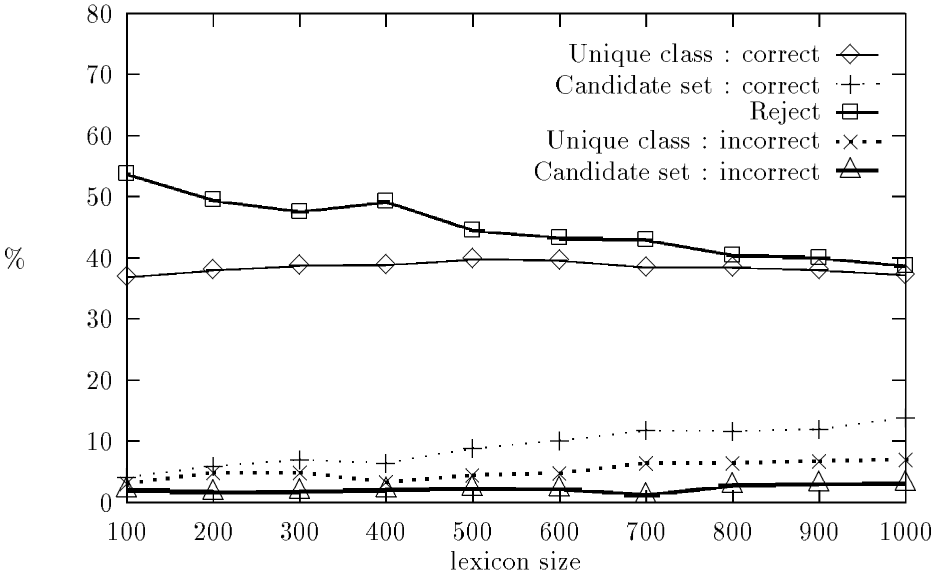


Figure 6.8: Results for increasing lexicon

words. The increasing lexicon was to determine the any trends in the results as the lexicon increased.

The results are shown in Figure 6.8. Note that the results when the lexicon is 1000 words is the same as those in the previous section when ξ was 1.0; in this case, the experiment was being run with the same parameters as here.

It can be seen that as the lexicon increases, the number of words with no classification drops from 53% to 38%, this is reflected by an increase in both the percentage of words containing the correct word (from 4% to 14%) and an increase in unique misclassifications (from 3% to 7%). The remaining 1% change is reflected in the increase of candidate sets not containing the correct candidate word (increasing from 2% to 3%).

These results are surprising because the percentage of unique and correct classifications are essentially constant : increasing from 36.8% with 100 words to 39.8% at 500 words and back to 37.2% for 1000 words. It is difficult to predict if this behaviour will continue for a further increasing lexicon. However it is likely that the number of words with no classification will continue to fall and the number of candidate sets containing the correct word will continue to increase. So for larger lexicons, the use of further contextual processing to reduce the candidate sets would be necessary. For those words classified with a candidate set, the number of words in the candidate set was on average 2.2 words for the 100 word lexicon and 2.3 words for the 1000 word lexicon.

6.6.2 Experiment Two

We are also interested in the performance of the classifier when the word is not in the word classifier lexicon. This is because the DR algorithm lets the global lexicon be larger than the word level lexicon so that new words can be learned. In this case, the whole word classifier must be able to reject words that are not in the classifier lexicon. This second measure of the classifier was designed to measure how well it rejected words that were not in the whole word lexicon.

In order to measure this, 100 random words were extracted from the lexicon as test words and the remaining words used for training. The classifier was then trained on increasing subsets of the remaining words and the results were recorded from these. The number of test words was kept constant so that the results could be compared as the training lexicon increased. Since the random 100 words were not being used to train the classifier, all the training images for these words were used to test the classifier (around 9 images per word). So in these experiments, the classifier was tested on 900 word images for each lexicon size.

The training and test words were chosen randomly from the lexicon so that different word types (particularly the word lengths) was distributed over the testing and training set. This is necessary since frequent words are, on average, shorter than the less frequent words.

The results of this experiment are reported in terms of rejections, unique classifications and candidate sets. However, the most important result to consider is the percentage of words that are uniquely classified since these represent confident classifications when there should be none.

Figure 6.9 shows the results of the classifier with the increasing lexicon. Notice that the

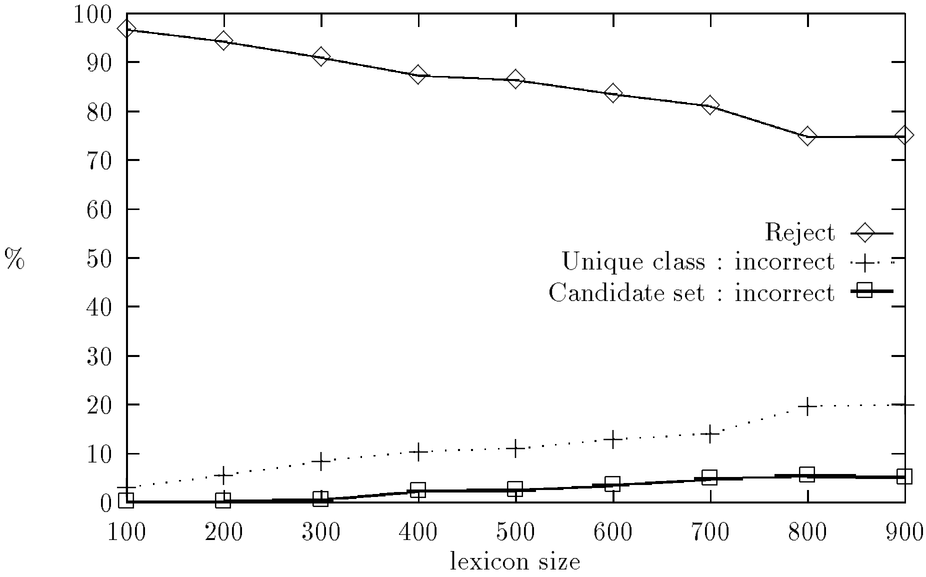


Figure 6.9: Classifier performance when tested with unknown words

training lexicon increases from 100 to 900 words since the remaining words were used for testing the classifier.

The results show that as the lexicon increases, the number of words rejected drops from 96% to 74%. Many of the misclassified words are unique (20% for 900 words), this error rate is important because it means that (in this case) one in five words would be misclassified and then used to adapt the classifier. In order to rectify this, a different approach to the confidence threshold could be applied or extra contextual information could be used to reduce the number of incorrectly classified words.

The remaining misclassifications are where a candidate set has been generated. These errors would not affect the DR algorithm directly; however, they would most likely cause an error in some other context processing where the candidate sets are used to generate word hypotheses.

6.7 Adaptation of Centroids

Chapter 5 showed how the DR algorithm uses confidently classified words and letters to adapt the internal representation of those words and letters, while this chapter has shown how the classifier represents words and letters using the 2 dimensional Fourier transform. We now consider the effect of adaptation on the centroids and the measure of confidence.

When a new word is added to the classifier, either as a previously unseen word or to modify an existing word shape, the centroid, mean and standard deviation need to be recalculated.

As shown before, for a word or letter, w, there is a set of n training images from which the feature sets have been extracted, $\underline{f^i}(w) = (f_1, f_2 ... f_j)$. The exemplar

$$\underline{E}(w) = \sum_{i=1}^{n} \frac{\underline{f^i}(w)}{n}$$

represents the shape of w.

When a new instance of w is added to the classifier, the centroid is modified by finding the new average of the training points.

If the new feature vector is $\underline{f^{n+1}}$, the new exemplar is : $\underline{E'}(w) = \frac{(\underline{E}(w) \cdot n) + \underline{f^{n+1}}(w)}{n+1}$

The effect of this is to shift the exemplar from the old mean of feature points to a new mean. If the classification has been incorrect then the effect is to move the exemplar away from the true centroid. Figure 6.10 shows the shifting effect of adding a new point in feature space.

One issue that arises when calculating the position of the new centroid is how many of the training points should be used. The DR algorithm is meant to adapt to the shape of the words currently being written however, if all training images are used it will still partly represent the old word shape. The centroid can be forced to represent the most recent images of a word by calculating the exemplar from a fixed number of previous feature vectors, thus representing a moving average of a possibly evolving style.

The equation for calculating the modified exemplar suggests also that this can be calculated without storing all the previous training features. If old feature vectors are to be removed

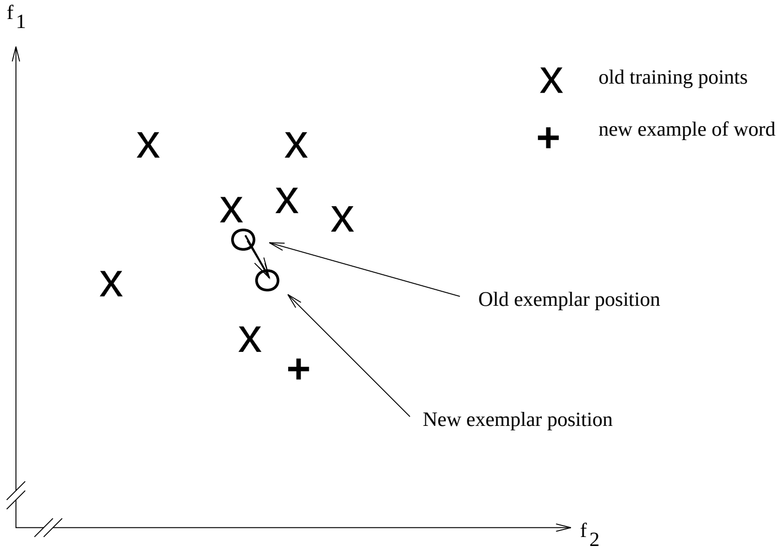


Figure 6.10: Adapting the position of the centroid

as new features are added then only the previous n feature vectors have to be stored for each word and letter exemplar.

Since the mean and the standard deviation of the distance of training points to the centroid are used to calculate the confidence measure, all the feature vectors used to calculate the current centroid have to be stored.

The effect of changing the number of feature vectors used to calculate the centroid is considered in the following chapter.

6.7.1 Effect of adaptation on the confidence measure

Figure 6.10 shows the movement of the centroid following the addition of a new feature vector (where this is shown in two dimensions). Depending upon the position of the new feature vector, the mean and standard deviation will either increase or decrease. If the distance between the new feature vector $\underline{f}$ and the centroid is less than the old mean distance, i.e. $\xi(\underline{f}) < 0$, then the mean will decrease. Similarly, if $\xi(\underline{f}) > 0$, the mean will increase. Since the confidence measure is a function of the mean and standard deviation,

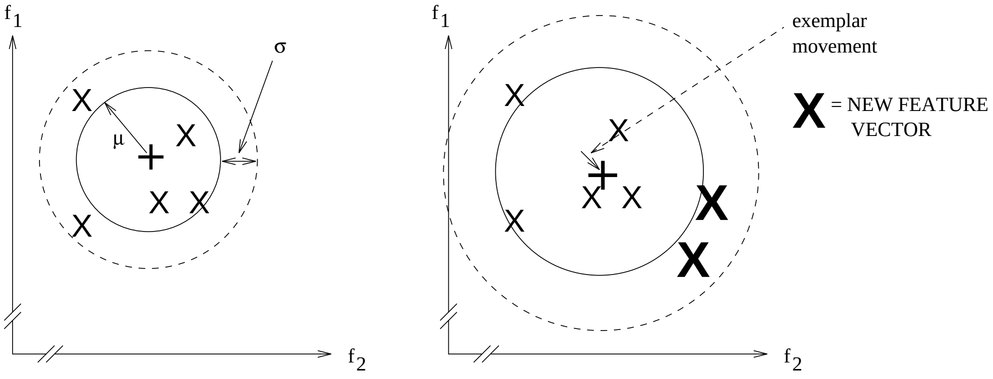


Figure 6.11: Confidence changes following addition of two feature vectors

increasing the mean will increase the cluster size for that word and so potentially increase the number of words that could be included in that cluster. Figure 6.11 shows the addition of two new feature vectors to a particular cluster. This addition increases the size of the cluster since the mean and the standard deviation have both increased and although the exemplar is a better representation of the word shape, many more words could be classified as confident in this cluster due to the increase in size.

If the number of feature vectors used to adapt a word is fixed, (and the writing style of the test data is different to that in the training data) then the mean distance of training points around the centroid is expected to increase and then decrease. It would increase as the centroid moves to a new position and then decrease as its new position in feature space is reinforced by similar feature vectors being added to the cluster.

If the writing style of the test data was the same as the training data, the exemplars would not be expected to move much and consequently there would be little change in the mean.

This change of the mean and standard deviation is important to the DR algorithm since words that are classified at the word level will always be within 1.0 standard deviations of the mean, since these are the only confident words. However, when words are classified at the letter level, the whole word feature could potentially be anywhere in feature space, significantly altering the mean and standard deviation.

6.8 Summary

This chapter has introduced the Fourier classifier proposed for the Directed Reading algorithm. The basic classifier used by O'Hair and Kabrinsky was described and then modified for use with handwritten words. Modification of the classifier involved finding a suitable image normalisation size and determining how large the feature vector should be.

Once the classifier had been designed it was tested on a set of 985 handwritten words. Words were classified by finding the nearest word centroid in feature space. The results of this produced classification rates of up to 83%. These results are comparable with other whole word classifiers (described in Chapter 2) while being far simpler in design.

A measure of confidence measured in relation to the spread of training points around the word centroid was then associated with classifications so that they could be used in the DR algorithm. By using the confidence measure in the classifier, the number of words being misclassified was reduced at the expense of reducing the number of classifications that were correctly (and uniquely) classified. If more than one word was confident, a candidate set of words was generated which could be used with other types of contextual information.

Finally, the classifier was tested with an increasing lexicon to determine the recognition performance when the classifier had been trained on all the words, and when the classifier was trained on different words. These showed that with this set of word images, (single author, written in one session) the classifier performed well, with an error rate of 10% and rejection rate of 38% on a lexicon of 1000 words.

Chapter 7

EXPERIMENTAL EVALUATION OF DIRECTED READING

The previous two chapters have described the Directed Reading algorithm and the classifier proposed for implementing it. This chapter now describes the implementation of the DR algorithm and the experiments used to evaluate it.

7.1 Objective of the Experiments

As with evaluating other sources of context, the main question when testing the DR algorithm is "by how much does the DR algorithm improve the recognition performance compared to the classifier in isolation?"

The basic measure of performance used in the systems described in Chapter 2 was the percentage of words correctly classified. For the DR algorithm, improvement in performance can be measured as the increase in the number of words classified correctly following each iteration of the algorithm.

Given that improvement in the DR algorithm can be attributed to adaptation to known words and learning of new words, two sets of experiments were proposed: the first to measure adaptation of known words and the second to measure new word learning. These could be implemented as testing the classifier using two extremes of training. In the first case, the whole word classifier is trained on all words that could be presented to it and adaptation comes from recognition at both the word and letter levels. When measuring learning, the letter classifier is trained and the word classifier is left untrained, i.e. no words can be read as whole words in the first iteration. Learning in the classifier can be measured by the number of words read as whole words in subsequent iterations.

A final set of experiments was performed with a small amount of word–level training to evaluate the effect of training the word classifier on a small number of frequent words.

7.2 Implementation Details

7.2.1 Introduction

The classifier was written in C++ on a Silicon Graphics Indigo workstation using the Khoros image processing library (Konstantinides & Rasure 1994) to implement the image processing routines, and the NAG mathematical library for the Discrete Fourier Transform function.

Since the DFT was computationally expensive, the feature vectors for all the training and testing words were extracted once and stored separately. Similarly, the test words were presegmented into letters and the extracted feature vectors were stored on disk. The aim of this pre-processing was to avoid re-calculating the Fourier transform during repeated testing of the algorithm. The images and feature vectors were stored without any modification or intervention.

During a test of the algorithm, all word and letter exemplars were loaded into memory, making the procedure rather large when running with a large classifier lexicon. In practice, a lexicon of 1000 words would fit into memory with a test set of approximately 600 words. If the parameters were larger than this, then the program would quickly run out of memory once adaptation started to take place, when new exemplars and old training vectors are loaded in. The memory restraints could be removed if only relevant exemplars and patterns were loaded at any one time.

Lexicon size	Number of words oversegmented	Number of words undersegmented
100	1 (0.7%)	17 (12%)
200	19 (4%)	36 (9%)
300	1 (0.2%)	21 (4%)
400	4 (1%)	12 (3%)
500	8 (1%)	27 (5%)
600	4 (1%)	24 (5%)
700	5 (1%)	21 (4%)
800	5 (1%)	28 (6%)
900	5 (1%)	25 (5%)
1000	12 (2%)	29 (5%)

Table 7.1: Percentage of under and over segmented words

7.2.2 Letter segmentation

Word images were stored as binary images and for the purpose of the experiments described in this report, the words were assumed to be discrete handprinted words, i.e. there is a white space between each character.

Segmentation of characters in the image was therefore implemented by labelling discrete regions in the word image and then extracting these as the segmented characters. The only additional rule in the segmentation process was to associate ‘i’ and ‘j’ dots with the body of the character. If a dot was found, then close segmented characters were checked to see if they were possible ‘i’ or ‘j’ bodies.

This segmentation process does not take account of merged characters (or segmented characters) due to a poor choice of threshold during the binarisation process. Due to this, there were a number of mis-segmented words presented to the classifier. The percentage of mis-segmented words is given in Table 7.1.

The table shows that the overall error introduced by taking the simple segmentation approach introduces a small error. The larger number of words undersegmented reflects the fact that some of the words had letters joined, either by the author or through merging introduced in thresholding.

7.2.3 Further rules introduced into the DR algorithm

The algorithm described in Chapter 5 does not contain any rules for error correction or error prevention. The table in the previous section showed that there would be at least one source of error that could be predicted, i.e. that the number of segmented characters may be incorrect.

Should one of these words be classified correctly at the word level, then the word would be incorrectly split into letters and then the incorrect letters would be used to adapt the letter exemplars. This was prevented by comparing the number of segmented letters against the chosen word of the classifier. If these counts were the same it was taken as evidence that the letter segmentation was successful and the letters were used to adapt the exemplars. If the numbers did not match, the letters were not used in adapting the classifier, however, the word classification was taken to be correct since the source of error was assumed to be in the segmentation routine and not the classifier.

If a badly segmented word is classified at the letter level, then there is no other source of evidence to compare the segmentation results with. In this case, there is no method of recovering from the error and images of the letters and of the whole word are used in the adaptation.

However, it was found that this simple check of segmented letter count against expected letter count meant that errors caused by misclassified words were prevented from propagating through the classifier. For example, if the word ‘dog’ were classified at the whole word level as ‘clog’ the number of segmented letters (3 in dog) would not (hopefully) match the number of expected letters (4 in clog) and the adaptation would be avoided.

7.3 Experiment description

The experiments were intended to measure adaptation and learning, however the data used in measuring these two aspects of the algorithm and the approach were basically the same. As in the previous chapter, all experiments were carried out by running the algorithm ten times with an increasing lexicon, the aim of this being to determine any trends over the increasing lexicon.

The design of the algorithm meant the classifier was modified at the end of each iteration; enabling the results of the classifier to be measured at the end of each iteration

7.3.1 Training and test data

The lexicon used for these experiments was the same as described in the previous chapter (and listed in Appendix A), i.e. the lexicon was taken from the most frequent words in the LOB corpus, and was split into subsets of increasing size.

In contrast, the test data was a set of sentences chosen from the LOB corpus (listed in Appendix B). The sentences were selected so that they contained only words in the current lexicon. Since there were ten test lexicons, there were ten sets of test sentences. The reason for choosing sentences rather than some random (or selected) set of words was that these experiments followed from the syntax experiments described in Hanlon & Boyle (1992 b).

Another constraint on the type of sentence chosen from the corpus was that the sentences all contained a single clause. This was to avoid problems with locating and identifying punctuation in the word image. At 75dpi, a couple of noisy pixels can look very similar to a punctuation mark, for example, a full–stop or colon. By choosing single clause sentences this problem was avoided, however it meant that the structure of sentences and the words in the chosen sentences were constrained.

Following the description of the algorithm, the classifier lexicon L_W is a subset of the global lexicon L_G. As mentioned in Chapter 5, when the classifier lexicon is smaller than the global lexicon, there will be words to be classified which have not been trained as whole words. In this case the classifier is expected to read the new words at the letter level and add a word exemplar to the whole word classifier. If the classifier has been trained with all the words in the global lexicon, then the adaptation is restricted to the movement of exemplars in feature space. In this case the improvement in successive iterations is based solely on the movement of exemplars and is used as a measure of possible adaptation in the classifier.

If we consider the DR algorithm and the Fourier classifier, we see that in order to learn a new word, the word has to have been classified twice (in order to estimate the variance for the confidence measure). In order to measure this learning, the word must appear in the test set at least three times in order to register a whole word classification. Similarly, whole word adaptation can only be measured if there is at least two instances of a word in the test set. The first used to adapt the whole word classifier (this can be recognised at either the word or letter level) and the second which would be recognised at the word level.

If the distribution of words in the test sentences is examined, we find that there is a small number of highly frequent words and a large number of low frequency words (when plotted this produces the standard Zipf curve). Given this distribution, we are interested in the potential for learning and adaptation in the test data.

We can define the potential for learning to be the number of words that occur frequently enough to allow initial recognition at the letter level and then subsequent recognition at the word level; that is, all words that occur three or more times. Similarly, the potential for word level adaptation can be measured as the number of words that appear more than once.

Table 7.2 lists the potential for adaptation and learning for each test set, i.e. different lexicon sizes. Potential for adaptation and learning is split into three columns: the first, *classes* is the number of different words that appear a sufficient number of times; the second column, *words* gives the total frequency of the words that could be adapted or learned. The final column *coverage* represents the maximum possible improvement that could be achieved with either the adaptation or learning. For adaptation this is calculated as the total word frequency minus the number of different word classes (i.e. the number of words that would have to be recognised in order to adapt the whole word classifier) which is divided by the number of words in the test set.

Before a word can be learned, it has to be classified twice at the letter level. In this case, the

Lexicon	Number	Adaptation			Learning		
Size	of words	Classes	Words	Coverage	Classes	Words	Coverage
100	142	26	119	(65%)	18	103	(47%)
200	465	100	424	(70%)	63	350	(48%)
300	486	101	419	(65%)	59	335	(45%)
400	356	72	260	(53%)	38	192	(33%)
500	592	109	469	(61%)	54	359	(42%)
600	498	93	372	(56%)	54	294	(37%)
700	499	83	365	(57%)	46	291	(40%)
800	481	92	358	(55%)	51	276	(36%)
900	456	91	325	(51%)	38	219	(31%)
1000	570	101	407	(54%)	52	309	(36%)

Table 7.2: Potential for adaptation and learning improvement with test data

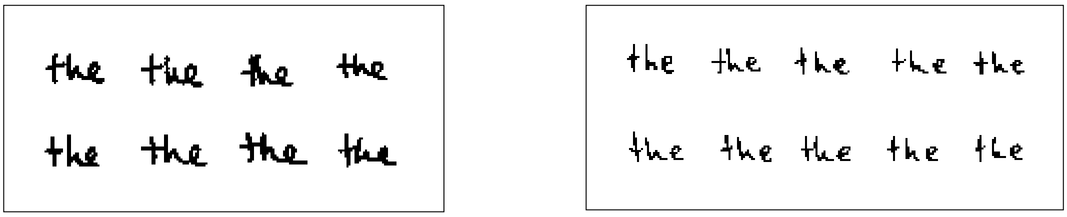
coverage for the learning is calculated as the total word frequency minus twice the number of different word classes divided by the total number of words.

It follows from the implementation of adaptation and learning that more words can be potentially recognised due to adaptation than through learning. However, the table shows that over 50% of the words in each of the test sets have the potential of being classified following adaptation. Similarly, on average 40% of the words in the test sets can be recognised following learning. Note that the set of words that could be classified following learning is a subset of those words that could be classified following adaptation. However, this shows that the distribution of words in the test sentences should facilitate some learning and adaptation.

7.4 Training and testing images

As described above, the experiments used the same lexicon and subsets of the lexicon as in the experiments in the previous chapter. This meant that the word images used to test the classifier in the previous chapter could be used in the experiments in this chapter.

The set of test words were the same as the experiments described in Chapter 4, i.e. all the



training set

test set

Figure 7.1: Training words 'the' and test words 'the'

sentences in the LOB corpus containing words in the constrained lexicon. Test word images were generated using the same process as the training words described in Chapter 6.

One difference between the training set and the test set was that the test words had been written more carefully with particular attention to the gaps between the words. Along with this, the words were written about three months after the training set had been written. This meant that the test set of images appeared quite different to the training set of images, as shown in Figure 7.1. This difference to the human observer means two things, firstly that the test words would be in a different style to those in the training set. This was not intended during the writing session, but later became apparent. Also, the extracted feature vectors could be statistically different to those in the training set, i.e. they would cluster in a different area in feature space. Again this is not convenient when testing the classifier performance, but is useful when testing the adaptive features of the classifier.

7.5 Measuring adaptation of known words

7.5.1 Experiment Description

The first set of experiments carried out measured the amount of adaptation achieved after each iteration. In this case the classifier was trained on each word that could appear in the test sentences. The changes in classification rates after each iteration could then be attributed to the classifier adapting to the style of the words in the test sentences.

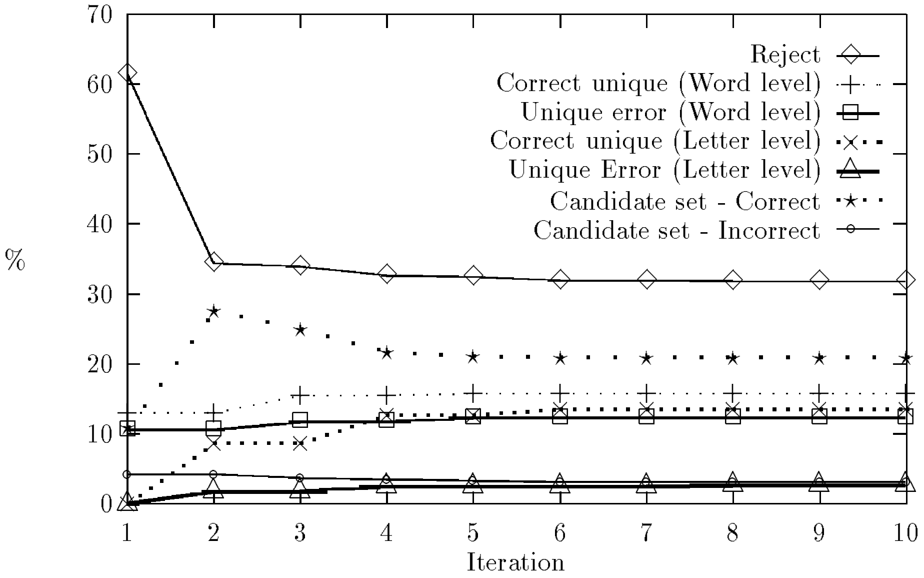


Figure 7.2: Adaptation results for lexicon of 1000 words

As mentioned earlier, the experiments were run ten times. For each different lexicon size, the classifier was tested on a different set of test sentences.

7.5.2 Results

The graph in Figure 7.2 shows the results for the experiment when run with a test and training set of 1000 words. The results for lexicons of 100 – 900 words showed the same pattern of results as in the graph for 1000 words.

Following the second iteration, both the word and letter classifiers have attempted to recognise the words. Following this, all changes in performance can be attributed to adaptation in the classifier.

The DR algorithm as implemented here, did not re–classify a word once it has been recognised (either correctly or not). This meant that the number of words recognised after each iteration would increase monotonically. The overall change in performance could then be calculated as the difference between the results after the algorithm had completed and the

lexicon size	word correct	word incorrect	letter correct	letter incorrect	cand set correct	cand set incorrect	no class
100	0.00	0.00	2.11	0.70	-2.11	0.00	-0.70
200	1.54	2.63	3.95	1.75	-4.82	-0.22	-4.82
300	0.41	4.12	4.94	0.41	-4.12	-2.06	-3.70
400	1.69	0.85	3.95	0.00	-4.80	-1.13	-0.56
500	1.88	2.05	5.46	0.34	-4.61	-2.56	-2.56
600	1.44	1.85	2.67	0.41	-4.11	-0.41	-1.85
700	2.40	0.80	3.61	0.60	-4.01	-0.80	-2.61
800	1.25	2.29	3.74	0.83	-3.53	-2.29	-2.29
900	2.85	1.75	3.73	0.44	-6.36	-2.19	-0.22
1000	2.81	1.75	4.91	0.88	-6.67	-1.05	-2.63

Table 7.3: Performance change when $L_W = L_G$ (measured in %)

results after the second iteration[1]. The change in performance for all lexicon sizes given in Table 7.3.

This table shows two sets of interesting results. The first, that more words are recognised at the letter level following adaptation rather than at the word level, and secondly, that this reduction appears to be from recognising words previously in the candidate sets rather than reducing the number of rejected words. The second point can be seen by comparing the high percentage reduction for correct candidates and the low reduction in rejections. The small reduction of rejected words was mainly through recognition at the letter level. For example, with a lexicon of 1000 words (570 test words), 196 words were rejected after the two iterations. Of those rejected words only 12 were later recognised : one (incorrectly) at the word level and 5 correct, 3 incorrect and 3 candidate sets (containing the correct word).

The first result was more words are later read at the letter level than at the word level through adaptation; on average, correct recognition at the word level increased by 1.6% while correct recognition at the letter level increased by 3.9%. One reason for this is that there are few letter classes compared to the relatively larger number of sparsely distributed

[1] In retrospect, it would have been better to adapt the classifier after both word and letter recognition since the letter classifier is modified immediately after word recognition, so the results of the letter classifier are from a modified classifier

word classes in the experiments. When a letter is adapted, it could have an effect on many words containing that letter. In contrast, when a whole word is adapted, this only has an effect when another instance of that word is subject to classification.

Another point about these results is that the increase in recognition comes from a reduction in the number of candidate sets rather than a reduction in the rejections (no class). This means that despite the adaptation, there remains a large number of words remaining unclassified (for example, about 30% in Figure 7.2).

When the misclassifications following the last iteration were examined, it was found that 31 of the 85 mistakes were due to the classifier choosing the word 'with'; these 31 mistakes were distributed over 24 different words. These errors were mainly due to errors in the first iteration where 25 words were mistaken for the word 'with'. However, propagation of these errors was avoided since only 7 of the erroneous words were four letters long. When the initial mean and variance were checked for the word 'with' it was found to have a higher confidence threshold than all other words, i.e. the sum of the mean and variance was higher for the word 'with' than for all other words. A high variance in the training set for the word was found to have caused this. Such empirical evidence can be used to pinpoint words and perhaps letters where a more specific 'special–case' classifier could be exploited. In this way words that are likely to cause an error can be predetermined and recognised with their own special–case classifier.

Figure 7.2 shows that a high proportion of the candidate sets contained the correct word. This is encouraging since the proportion of candidate sets represents the scope for other sources of context to contribute information. When looking at the candidate set sizes, it was found that after two iterations of the algorithm the average candidate set size was 2.9 words and when the algorithm had completed, the average candidate set size was 3.1 words. This shows that after adaptation the overall sizes of clusters has grown to accommodate more words as predicted in the previous chapter. The candidate set sizes lay between 2 and 5 words at the start of the algorithm, while at the end of the algorithm one word had a candidate set size of 6 words and another, a candidate set of 14 words which accounted for much of the increase in size.

7.6 Measuring classifier learning

7.6.1 Introduction

The previous experiment showed that there was some adaptation taking place as the algorithm iterated. The following experiments were to measure if any learning had taken place.

The classifier learns a new word when it has been recognised at the letter level and is subsequently used to train the whole word classifier. This can only happen when the whole word lexicon is smaller than the global lexicon. It follows that as the difference between the two lexicons increases, the number of new words that can be added to the whole word classifier (i.e. the scope for learning) increases. It would be possible to test every combination of global lexicon size against different whole word lexicons to determine the effect of changing this scope for learning. However, the aim of these experiments was to determine if any learning took place and to measure how much; this was tested by running experiments with an increasing global lexicon (as before) each with two different sizes of whole word lexicon :

- **Whole word lexicon = empty :** In this case the classifier is only trained on letters. During the first iteration, the words can only be recognised at the letter iteration. If any words have been recognised these will be used to adapt both the word and letter classifiers. Any whole words that are subsequently recognised have been learned solely from the letter classifications.
- **Whole word lexicon = 100 words :** Another set of experiments was carried out to examine the effect of a small initial word lexicon. In this case, the lexicon was the most frequent 100 words. During the first iterations, words are recognised at the word and letter levels and adaptation should take place. Recognition of whole words not in the initial word lexicon can then be measured.

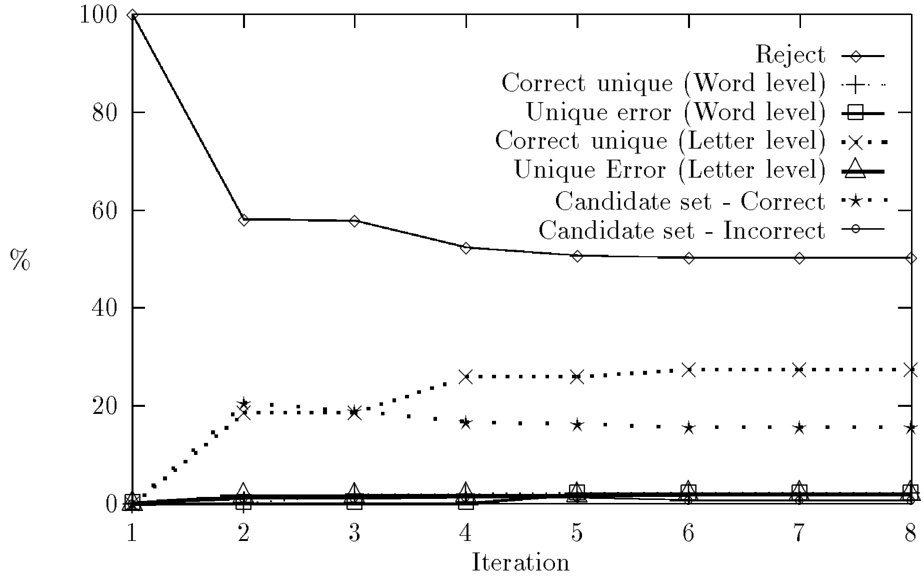


Figure 7.3: Results for empty initial whole word lexicon

7.6.2 Results for empty whole word lexicon

Figure 7.3 shows a graph of the results when the classifier was tested with words from the lexicon of 900 words. The results for the classifier when tested with sentences with lexicons of 100 to 700 words were similar and are tabulated later; the results when the lexicon was 800 words were quite different and are also described later, while the results for training on 1000 words were not obtained due to memory constraints on the machines used to test the classifier.

The results show that much of the adaptation occurs at the letter level, increasing the performance in reading words at that level by 9%. However, we were interested in the number of words correctly classified as whole words : about 2% of words are recognised correctly at the word level, and by the fifth iteration this was matched with about 2% of words being misclassified at the word level. In the test set of 456 words (lexicon size 900 words), this represented 10 words being classified correctly and 9 words incorrectly as whole words. The 10 words classified correctly were the words ‘of’ (seven times), ‘in’ (twice) and ‘is’ (once) while the words misclassified were the words ‘been’ (seven times) and ‘at’ (twice).

It was found that the average percentage of words correctly classified because of learning was 2.0% while the average percentage of words misclassified through learning was 0.4%. This showed that some learning had taken place and that correct words had subsequently been classified as whole words. If we consider this in terms of the potential for learning (in Table 7.2) we see that only a few of the words have been learned. So, for example, when the lexicon was 900 words, the number of potential new words that could be learned was 52; however we see that only three words (of, in, is) were learned.

These results show that relatively short words are likely to be learned at the whole word level. The words learned and later correctly used for classification in all experiments reported in this section are given in Table 7.4.

Frequency	Word	Frequency	Word
25	it	1	they
14	to	1	talk
14	of	1	opened
9	he	1	on
7	is	1	have
4	in	1	good
3	that	1	gone
2	this	1	coming
1	true	1	better

Table 7.4: Whole words learned and subsequently used for classification

It is not surprising that there is a tendency to learn shorter words and later classify them. In the first case, shorter words are more likely to be recognised at the letter level than longer words, for example, if the probability of a letter being classified correctly is 0.9, then the probability of a two letter word being classified becomes $0.9^2 = 0.81$ and the probability of a four letter word being classified is $0.9^4 = 0.6561$. So the probability of recognising a longer word at the letter level is lower than that of a shorter word. Secondly, the choice of test data, i.e. the sentences listed in Appendix B, meant that short function words appeared many times and longer words appeared infrequently. This meant that a higher proportion of short words were used to adapt the classifier than long words.

As before, the results for the increasing global lexicon sizes are listed in Table 7.5. Note that the tabulated results show the difference in performance between the second and last iterations of the algorithm, to show what effect the algorithm has on performance.

lexicon size	word correct	word incorrect	letter correct	letter incorrect	cand set correct	cand set incorrect	no class
100	0.70	0.70	4.93	2.11	-4.93	-0.70	-2.82
200	2.63	0.00	2.85	1.32	-3.51	-1.10	-2.19
300	1.85	0.21	5.97	1.03	-5.14	-1.44	-2.47
400	2.54	0.85	4.80	0.56	-4.24	-0.85	-3.67
500	3.92	0.85	8.19	0.85	-6.48	-1.02	-6.31
600	1.03	0.00	6.37	0.62	-2.05	-0.62	-5.34
700	3.01	0.00	7.21	0.60	-5.21	-1.00	-4.61
800	0.83	24.12	5.41	2.08	-8.94	-0.42	-23.08
900	2.19	1.97	8.77	0.44	-5.04	-0.44	-7.89

Table 7.5: Performance change with empty initial whole word lexicon

It can also be seen from the table that the results for a global lexicon of 800 words does not follow the trend of the other results. This can be seen particularly in the large increase in words incorrectly classified at the whole word level. Figure 7.4 shows the results following each iteration of the algorithm when run with a global lexicon of 800 words.

The first point to note is the large number of iterations. While the average number of iterations for all other experiments was 10 (and never more than 16), here the number of iterations is 30. Also, by the tenth iteration, the results for letter level classification (correct and incorrect) became stable and the results for correct whole word were stable. The reason for the many iterations is the incorrect word level classifications. The graph shows the misclassifications steadily increasing over many iterations.

This is a good example of a small error being propagated into a larger global error. In the second iteration (letter level) the words ‘it’ and ‘an’ were misclassified as ‘at’ and used to adapt the whole word exemplar for the word ‘at’. Following this, the next iteration showed a large number of words misclassified at the word level; usually the overall increase in

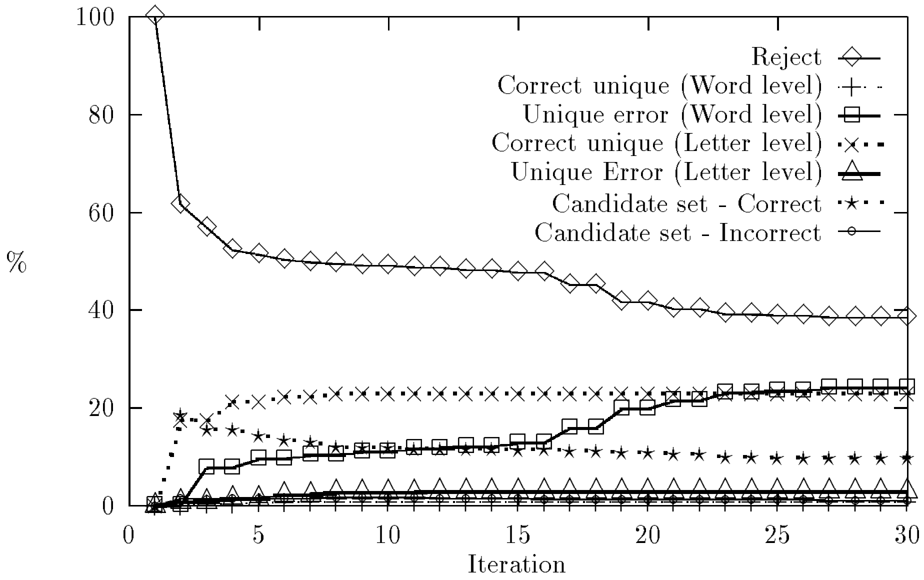


Figure 7.4: Results : empty classifier lexicon, global lexicon = 800 words

whole word misclassifications was less than 1%, here in one iteration it had increased by over 7%. All of these words were misclassified as the word ‘at’. Many of these words were two letter words so both the misclassified words and letters were used to modify the classifier causing the error to propagate. By the time the algorithm had completed, all the erroneous words had been classified as the word ‘at’.

Although this happened once in thirty runs of the experiments, it shows that using the confidence of a classification in isolation does not give a suitable measure of the effect of adding an exemplar to the classifier. Instead of making the decision about adding a word to the classifier using only a measure of the clusters size, a wider source of information could be used (e.g. the relative position of neighbouring clusters). For example, if the cluster overlap between words was to increase with the addition of a new word it could be penalised; or addition of words could be accepted only if it did not decrease the performance of the classifier on words that it had already been trained on. Changing the decision criteria for adding a word to the classifier could be an important factor in the DR algorithm, however it was not considered further in this piece of work.

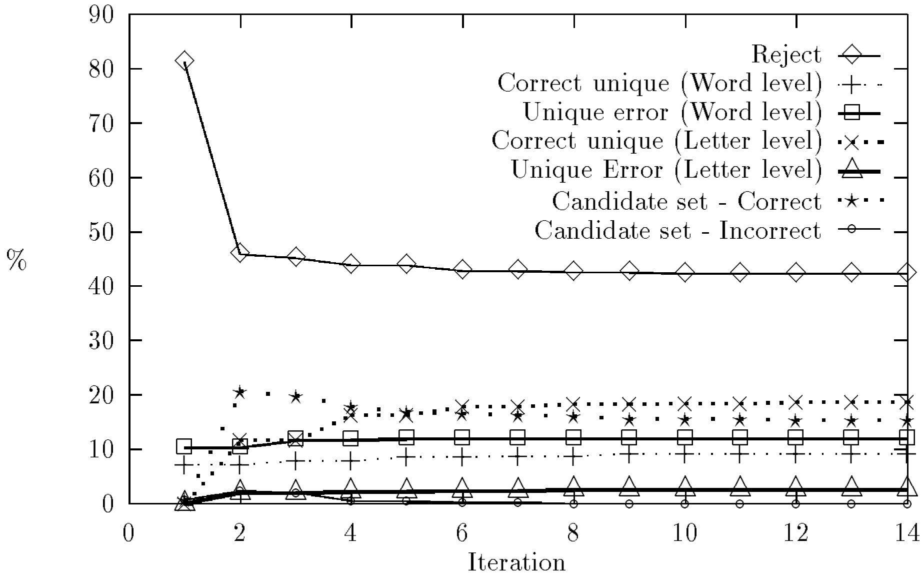


Figure 7.5: Results when initial whole word lexicon = 100 words

7.6.3 Results when word lexicon = 100 words

The above set of experiments was repeated with an initial word lexicon of 100 words (compared to an empty initial lexicon). The experiments were run with an increasing lexicon of 100 to 1000 words, the results for the global lexicon of 900 words is given in Figure 7.5. The results for the other experiments are tabulated later, however the results in Figure 7.5 can be compared to those in Figure 7.3 (when the word lexicon was empty).

The percentage of words being classified at the whole word level was (predictably) higher than in the previous set of experiments, however it can be seen that the percentage of misclassified whole words is higher than the percentage of correct whole words. The increase in the whole word recognition was marginal, about 0.5% increase in correctly classified whole words and about a 1% increase in incorrectly classified whole words.

The difference in results after the second iteration and the final iteration of the algorithm is given in Table 7.6.

lexicon size	word correct	word incorrect	letter correct	letter incorrect	cand set correct	cand set incorrect	no class
100	0.00	0.00	2.11	0.70	-2.11	0.00	-0.70
200	0.66	1.54	4.61	2.19	-4.17	-1.10	-3.73
300	1.44	7.61	7.00	0.62	-7.82	-2.06	-6.79
400	0.56	3.11	3.39	0.85	-3.11	0.00	-4.80
500	1.02	3.92	3.92	0.85	-1.71	-1.71	-6.31
600	2.26	2.05	6.78	0.41	-4.72	-1.23	-5.54
700	1.60	2.81	5.81	2.00	-4.21	-1.80	-6.21
800	0.62	0.42	4.57	0.42	0.21	-1.87	-4.37
900	1.97	1.54	7.02	0.66	-5.26	-2.41	-3.51
1000	0.88	2.28	5.26	0.70	-2.81	-1.40	-4.91

Table 7.6: Performance change when whole word lexicon = 100 words

7.6.4 Discussion

These two experiments were to investigate the *learning* in classifiers rather than the adaptation of already known words to a particular writing style. Taking the raw results from these two experiments it can be seen that little learning of new words as been achieved.

Here, learning is measured by looking at the number of words classified at the word level when no training at the word level has taken place. In order for this to happen, the word must be recognised at the letter level at least twice (in order to estimate the variance of the features in the cluster). The results show that there are many words classified correctly at the letter level which are then used to adapt the whole word classifier. For example, in the first set of experiments (when the word lexicon was empty) when tested with a global lexicon of 900 words, there were 81 different words correctly classified at the letter level. Of these, only 10 were correctly classified more than twice (i.e. could be learned) of which, three were later recognised at the word level. We see that there has to be a higher recognition rate at the letter level in order to recognise enough words to adapt the word classifier. So, although a high rejection rate is desirable in order to avoid mistraining, we find that this does not facilitate learning.

7.7 Summary

The experiments described in this chapter were meant to evaluate the DR algorithm in two areas. The first, adaptation, is when the classifier modifies the exemplar of a word in order to better represent the style of writing and so increase recognition rates. The second, learning, is when the whole word classifier has it's lexicon extended through the addition of new words which had previously been recognised at the letter level.

Both adaptation and learning could be measured in terms of the percentage improvement made in recognising a sequence of handwritten words. The recognition rates were measured after the second iteration of the DR algorithm – when both word and letter classifiers had attempted to recognise the words – and then when the algorithm had completed. In addition to these results, the performance was measured after each iteration.

In general, the results showed that performance, i.e. the percentage of words correctly classified, increased more at the letter level than at the whole word level. On average, this increase in performance at the letter level was 5.0% while at the word level it was 0.4% (averaged over all experiments reported in this chapter). This is important since the original aim was to improve word level recognition; although discrete handprinted words were used, it was accepted that segmentation of letters is difficult. If letter level adaptation is contributing more to the increase in performance, then either the whole word classifier needs more work, and perhaps a better measure of confidence, or more robust sub–word units could be developed to exploit the adaptation at the letter level.

When measuring learning, it was found that a small percentage of words was being learned and used in later classifications. This was caused by a small number of words being recognised more than once which could train the whole word classifier. The recognition rate at this level depends upon both the letter classifier and the method of rejecting words.

In one case, these experiments showed that small errors when learning a new word could be propagated over a document after many iterations. This shows that in comparison, unsupervised learning of new words is less stable than the unsupervised adaptation of known words. This is because when adapting, erroneous information only moves the exemplar away from its true position, while when learning, erroneous classifications cause the exemplar to be placed in the wrong position in feature space. If, as in this case, two different words are misclassified as one word, the mean and variance of that cluster becomes large along with the potential error rate.

Chapter 8

SUMMARY AND CONCLUSIONS

This thesis has considered adaptation in off–line handwriting recognition. In order to test the adaptation, the Directed Reading (DR) algorithm was designed and implemented which controlled the unsupervised learning of new words and adaptation of known words. In order for the algorithm to be tested, a classifier which used low harmonic Fourier features was implemented which gave good results for a single author.

This chapter first presents a summary of the work presented in this thesis, followed by the conclusions which can be drawn from this work and finally presents possible future directions for work.

8.1 Summary

This work first reviewed the research area for off–line handwritten word recognition. It was found that recognition of handwritten words falls into two categories : global (whole word) recognition, where the word is classified as a single symbol; and analytical (letter level) recognition, where the first process is to segment the word into individual letters followed by recognition of the individual letters. A third category, Hidden Markov Model recognition, was found to fall into the previous two categories of whole word or letter level recognition. None of these systems could recognise a medium sized lexicon (of about 1000 words) with results close to 100%. The system suggested by Bozinovic and Srihari, however, did suggest the use of adaptation better to model the style of writing.

In order to increase the recognition performance, other sources of information are used, commonly called contextual information. The review of word recognition techniques was followed by a review of contextual methods, usually used either to reorder or reduce a list of candidate words (the candidate set). It was found that these fall into two general areas : language model sources of context commonly where statistical models of word collocations are used to modify the candidate set; or application specific context, usually relying on redundancy in the document image to provide a second source of the same information. This does not exclude language model context from applications, for example, contextual information acquired through dictionary search is frequently used in applications such as postal address recognition, while being classed as a language model by some.

Chapter 4 presented a set of experiments implementing and evaluating a Hidden Markov Model of syntax aimed at reducing the number of words in candidate sets. These experiments were performed on images of printed text providing a guide to how syntactic context could be utilised in a handwriting recognition system.

Chapter 5 described the development of the Directed Reading algorithm. This was designed to use both language model sources of context and to adapt the classifier while reading a document over a number of iterations. A measure of 'stability' of context was discussed which enabled the different types of context (including the adaptation in the DR algorithm) to be compared.

The DR algorithm exploited context in the form of adaptation and analytical recognition techniques when reading a single–author document. Because both global and analytical methods of recognition were used to recognise words, two lexicons were defined : the global lexicon, L_G which contained all possible words that could occur in the problem domain; and the whole word lexicon, L_W which contained all the words that could be read at the word level. When $L_W \subset L_G$, the classifier could adapt known word shapes if the recognised word was a member of L_W, and learn new words if the word recognised at the letter–level was in L_G but not in L_W.

The algorithm enabled the classifiers :

- to read words at the word and letter levels,
- to have an expanding whole word recogniser lexicon,
- to generate word hypotheses which could be tested over subsequent iterations,
- and to adapt to a possibly evolving handwriting style.

In order to implement the DR algorithm, a classifier was implemented based on that used by O'Hair and Kabrinsky to recognise printed text. The algorithm used one exemplar (centroid) per word which meant that adaptation of a word image could be implemented by moving the exemplar in feature space. Another feature of this classifier was that the features were not extracted through feature–spotting methods, instead the Fourier features described the gross shape of the word. This meant that arbitrary changes to the word shape could occur which did not need to be pre-defined.

A measure of confidence was introduced based on the spread of training points in a cluster. This allowed words to be rejected if not 'close' to the cluster centre. The results showed that the error rate for a lexicon of 1000 words was about 20% with about 40% being correctly classified with no other words in the candidate set and another 40% of words being rejected. However, in another set of experiments, the results showed that about 20% of words were being incorrectly classified when they should have been rejected — this would adversely affect the results of the word learning.

The DR algorithm was then tested using the Fourier classifier. Adaptation and learning were recorded, where adaptation was measured by taking the increase in recognised words which the classifier had been previously trained on, and learning which was measured by taking the number of words classified as whole words which had not previously been trained.

The results showed that there was significantly more adaptation at the letter level than at the whole word level and that the learning that took place was split almost equally between correct and erroneous classifications. It was also shown that adaptation could be considered more ‘stable’ than learning new words.

8.2 Conclusions

Syntax was shown to be an effective source of contextual information, however as the lexicon increased, the error rates increased. Also, the number of words removed from a candidate set was found to reduce as the size of the lexicon increased. This showed that for a large lexicon further contextual information was necessary and hence prompted the investigation into the DR algorithm.

When evaluating the DR algorithm, two sets of experiments were presented : the first, evaluating the Fourier classifier; and the second evaluating learning and adaptation. From the results of these experiments a number of conclusions can be made. The results of these experiments can be summarised as follows :

- The whole word classifier worked well in isolation. When the measure of confidence was included, the error rate was still low although the rejection rate was high (about 40%).
- Adaptation occurred more at the letter level than at the word level in all experiments.
- Learning of new words accounted for little change in the results, and where learning had taken place, there was usually an equal number of words later classified in error than classified correctly.
- In one case, learning errors made during early iterations caused the algorithm to iterate many times strengthening and propagating that error.

From these points a number of conclusions can be made. Firstly, that the measure of confidence was insufficient for deciding if a word should be used in adaptation. The confidence measure was calculated using the spread of points in an individual cluster.

When deciding if a word should be used in the adaptation, it was only this local information about the cluster that was used without considering the wider consequences of the changes. As suggested in Chapter 7, a better criterion for the addition of new words and letters would be one which took into account the global effects of the change. However, the measure of confidence is important when rejecting words since the high percentage of unclassified words means that the algorithm re–classifies many words in each iteration. A better measure of the confidence could be based on a better model of the shape of the cluster and perhaps reject fewer words. This could be based on either a weighted Euclidean distance (modeling the variance in each dimension) or on Mahalanobis distance (modeling the correlation between different features). The conclusion, however, is that a different criterion for deciding if a word should be added should be used.

Another observation was that the overall performance of the classifier was worse when classifying the test sentences compared to when it was tested in isolation (with one word extracted from the training set). The Fourier classifier acts more like a fuzzy template matcher than a classifier based on feature spotting. When the style of writing changes slowly, we would expect the classifier (and perhaps the measure of confidence) to work well since many words would be close to existing exemplars. However, in the experiments described in this thesis, the change in writing style had been exaggerated and so the words were not close to existing exemplars. In cases where the writing style changes dramatically, (or enough to cause many rejects) the classifier could perhaps be augmented with another general word classifier, rather than working in isolation, where the Fourier classifier rejected many words with the result that the DR algorithm did not have enough new examples of each word to adapt for later iterations.

This also leads to an explanation for the greater adaptation at the letter level compared to that at the word level. The low number of words added to the word classifier caused by the high reject rate, meant that few words were adapted and so little improvement was made at the word–level. The few words recognised at the word level, however, were segmented and the letters used to adapt the letter–level classifier. This meant that the letter classifier received more examples of letters that the word classifier. This resulted in more words correctly classified in later iterations.

The main work described in this thesis had been to study whole word adaptation and recognition. However, this has shown that the letter–level (or perhaps a sub–word level) is just as important, if not more so, because a few correct classifications at the whole word level can lead to a large amount of adaptation at the letter level.

Finally, as mentioned in Chapter 7, there is a link between the learning of new words and the application domain (or more precisely, the frequency and distribution of words used in the domain). This is because learning only takes place when there is more than one image of the word classified which can be used for training. In a one–off test set, such as that in Chapter 7, only a few words occur more than twice. This means that the DR algorithm is more suited to applications where it would be used over a long period where the domain specific words are presented over a period of time.

8.3 Future Work

This thesis has introduced the DR algorithm and has explored some of the issues associated with it. From the above conclusions, a number of directions for further work can be suggested. These possible further areas include :

- Classifier issues, including work on different measures of confidence, development of a letter classifier and segmentation.
- Development of a criterion for adaptation.
- Use of other sources of context.

The performance of the algorithm was related to the performance of the classifier and the effectiveness of the confidence measure. The large amount of adaptation that occurred at the letter level leads to the fact that the algorithm could be better implemented with a classifier designed specifically for letter recognition, rather than using the same classifier architecture for both word and letter recognition.

In this thesis, the issue of segmenting whole words into characters has been avoided by using discrete handprinted characters. However, a reliable segmentation algorithm would be necessary to enable the use of letter level information. The problem is to balance the difficult task of segmenting a word image into letters against the fact that more adaptation takes place at the letter level. In the light of this, a different segmentation strategy could be used, perhaps segmenting words into regular sections, such as beginning, middle and end.

As mentioned in the previous section, the word clusters may be better represented using a weighted distance measure. This would have an effect of the way the confidence was calculated. So, for example, the confidence is currently calculated using the Euclidean distance between training points and the exemplar, and then finding the mean and variance of these distances. An extension to this in order to better model the distribution of points, could be to store the mean and variance in each dimension.

A further extension to this work would be to consider other potentially viable exemplars in feature space, rather than just those that lie within the cluster boundaries of $\mu + 2\epsilon$. In such a case, exemplars (and hence possible candidates), could be weighted by their distance from the feature vector of the unknown word. Such a measure would increase the likelihood of the correct word appearing in the candidate set, albeit with a low confidence.

This leads to further work in the criterion for choosing if a word should be used to adapt the classifier. Currently this is done using only the confidence measure, however the previous section showed that this may not be sufficient and that global information would have to be calculated from either surrounding clusters, or more generally, from all clusters. One approach could be to adapt the classifier and then test against all the relevant training data. If the modified classifier fares worse than the initial classifier, then the new word image could be removed from any adaptation.

The algorithm was also only tested on a small set of words. For it to be properly tested, it should be used over a period of time, where words have the opportunity to be learned and adapted. The first problem then is how should the algorithm be used. As the algorithm is defined at the moment, pages would be written and scanned, then read and the classifier adapted. However, in a real–world situation, this would probably be an unlikely scenario and the algorithm would be better implemented in some PDA equipment and used in conjunction with some on–line front–end. The application area and use of the algorithm are serious considerations for future work.

There should also be some investigation into how much written text should be considered in each iteration. This could be as small as a sentence, or as large as a page (in these experiments, the algorithm iterated over all the test sentences). The more text the algorithm iterates over, the longer it takes since it is continually reclassifying some words. A balance between the time taken for the algorithm to iterate over a document and the learning / adaptation taking place could be considered.

Finally, the integration of context in the DR algorithm could be further explored. Results in Chapter 7 showed that a high proportion of candidate sets contained the correct word, while candidate set sizes were generally small. This should facilitate the use of other contextual information when reducing the remaining candidate sets.

BIBLIOGRAPHY

Atwell, E., Arnfield, S., Demetriou, G., Hanlon, S., Hughes, J., Jost, U., Pocock, R., Souter, C. & Ueberla, J. (1993), Multi-level disambiguation grammar inferred from English corpus, treebank and dictionary, *in* S. Lucas, ed., 'Grammatical Inference : theory, applications and alternatives', Colloquium Digest no. 1993/092, Institution of Electrical Engineers, London, pp. 91–98.

Bozinovic, R. M. & Srihari, S. N. (1989), 'Off–line cursive script recognition', IEEE Proceedings on Pattern Analysis and Machine Intelligence **11**(1), 68–83.

Bramall, P. E. & Higgins, C. A. (1993), A blackboard approach to on–line cursive handwriting recognition for pen based computing, *in* 'Third International Workshop on Frontiers in Handwriting Recognition', Buffalo, U.S., pp. 295–304.

Breuel, T. (1994 a), A system for off–line recognition of handwritten text, Technical Report Technical Report 94–02, IDIAP, IDIAP, C.P. 609, 1920 Martigny, Switzerland.

Breuel, T. M. (1994 b), Language modelling for a real–world handwriting recognition task, *in* L. J. Evett & T. G. Rose, eds, 'Language models in Speech and Handwriting Recognition', AISB Workshop series.

Brown, M. (1981), Cursive script recognition, PhD thesis, Univ. Michigan, Ann Arbor, MI.

Caesar, T., Gloger, J., Kaltenmeier, A. & Mandler, E. (1993), Recognition of handwritten word images by statistical methods, *in* 'Third International Workshop on Frontiers in Handwriting Recognition', Buffalo, U.S., pp. 409–416.

Caesar, T., Gloger, J. M., Kaltenmeier, A. & Mandler, E. (1994), Handwritten word recognition using statistics, *in* 'European Workshop on Handwriting Analysis and Recognition : A European Perspective', IEE, pp. 5/1–5/7.

CEDAR (1994), Research summary, Technical report, Center of Excellence for Document Analysis and Recognition, University of New York at Buffalo.

Chen, M., Kundu, A. & Zhou, J. (1992), Off–line handwritten word recognition using a single contextual Hidden Markov Model, *in* 'Proc. of the Computer Vision and Pattern Recognition Conf.', Champaign, IL, pp. 669–672.

Chen, M. Y. & Kundu, A. (1993), An alternative to variable duration HMM in handwritten word recognition, *in* 'Third International Workshop on Frontiers in Handwriting Recognition', Buffalo, U.S., pp. 82–91.

Cheriet, M. (1993), Reading cursive script by parts, *in* 'Third International Workshop on Frontiers in Handwriting Recognition', Buffalo, U.S., pp. 403–408.

Chomsky, N. (1955), Syntactic Structures, Morton, The Hague.

Cohen, E. (1992), Interpreting handwritten text in a constrained domain, PhD thesis, University of Buffalo, SUNY.

Cohen, E. (1994), 'Computational theory for interpreting handwritten text in constrained domains', Artificial Intelligence **67**, 1–31.

Cohen, E., Hull, J. J. & Srihari, S. N. (1991), 'Understanding handwritten text in a structured environment: determining zip codes from addresses', International Journal of Pattern Recognition and Artificial Intelligence **5**(1 & 2), 221–264.

Crowner, C. & Hull, J. (1991), A hierarchical pattern matching parser and it's application to word shape recognition, *in* 'First International Conference on Document Analysis and Recognition', Saint-Malo, France, pp. 323–331. September 30 – October 2, 1991.

Downton, A. C., Tregidgo, R. W. S. & Kabir, E. (1991), 'Recogniton and verification of handwritten and hand–printed British postal addresses', International Journal of Pattern Recognition and Artificial Intelligence **5**(1 & 2), 265–291.

Dudani, S. A., Breeding, K. J. & Mcghee, R. B. (1977), 'Aircraft identification by moment invariants', IEEE Transactions on Computers **26**, 39–45.

Edelman, S., Ullman, S. & Flash, T. (1990), 'Reading cursive scriptby alignment of letter prototypes', International Journal of Computer Vision **5**(3), 303–331.

Ehrich, R. W. & Koehler, K. J. (1975), 'Experiments in the contextual recognition of cursive script', IEEE Transactions on Computers **24**, 182–194.

Evett, L. J. & Bellaby, G. J. (1994), The integration of knowledge sources for word recognition, *in* 'European Workshop on Handwriting Analysis and Recognition : A European Perspective', IEE, pp. 16/1 – 16/8.

Fairhurst, M. C. (1994), Automatic signature verification : making it work, *in* 'European Workshop on Handwriting Analysis and Recognition : A European Perspective', IEE, pp. 3/1–3/5.

Fairhurst, M. C. & Cowley, K. D. (1993), 'Parallel multi–layer classifier architectures of increasing heirarchical order', Pattern Recognition Letters **14**, 141–145.

Farag, R. F. H. (1979), 'Word-level recognition of cursive script', IEEE Transactions on Computers **28**, 172–175.

Ford, D. M. & Higgins, C. A. (1992), A tree based dictionary search technique and comparison with n-gram letter graph reduction, *in* R. Plamondon & C. G. Leedham, eds, 'Computer Processing of Handwriting', World Scientific, Singapore, pp. 291–312.

Forney, G. D. (1973), 'The Viterbi algorithm', Proceedings of the IEEE **61**, 268–278.

Forney, Jr., G. D. (1973), 'The Viterbi algorithm', Proc. IEEE **61**, 268–278.

Fowler, H. W. & Fowler, F. G., eds (1990), The Concise Oxford Dictionary of Current English, eigth edition (ed. r e allen) edn, Clarenden Press, Oxford.

Frazier, L. & Rayner, K. (1982), 'Making and correcting errors during sentence comprehension: Eye movements an the analysis of structurally ambiguous sentences', Cognitive Psychology **14**, 178–210.

Freeman, H. (1961), 'On the encoding of arbitrary geometric configurations', IRE Transactions on Electronic Computers **EC–10**(2), 260–268.

Fukushima, K. (1987), 'A neural network model for selective attention in visual pattern recognition and associative recall', Applied Optics **26**(23), 4985–4992.

Fukushima, K. (1988), 'Neocognitron : A hierarchical neural network capable of visual pattern recognition', Neural Networks **1**(2), 119–130.

Fukushima, K. (1993), Connected character recognition with a neural network, *in* 'Second International Conference on Document Analysis and Recognition', Tsukuba Science City, Japan, pp. 240–243.

Gilloux, M., Bertille, J.-M. & Leroux, M. (1993), Recognition of handwritten words in a limited dynamic vocabulary, *in* 'Third International Workshop on Frontiers in Handwriting Recognition', Buffalo, U.S., pp. 417–422.

Gilloux, M., Leroux, M. & Bertille, J.-M. (1993), Strategies for handwritten words recognition using Hidden Markov Models, *in* 'Second International Conference on Document Analysis and Recognition', Tsukuba Science City, Japan, pp. 299–304.

Gonzalez, R. C. & Wintz, P. (1987), Digital Image Processing, Addison-Wesley.

Goodman, K. S. (1967), 'Reading : A psycholinguistic guessing game', Journal of the Reading Specialist **6**, 126–135.

Haber, R. N. & Haber, L. R. (1981), 'The shape of a word can specify it's meaning', Reading Research Quarterly **XVI**(3), 334–345.

Hanlon, S. J. & Boyle, R. D. (1992 a), Evaluating a Hidden Markov Model of syntax in a text recognition system, *in* 'British Machine Vision Conference', University of Leeds, U.K., pp. 462–471. 23rd – 25th September 1992.

Hanlon, S. J. & Boyle, R. D. (1992 b), Syntactic knowledge in word level text recognition, *in* R. Beale & J. Finlay, eds, 'Neural Networks and Pattern Recognition in Human–Computer Interaction', Ellis Horwood, pp. 173–189.

Harmon, L. D. (1962), 'Handwriting recogniser recognises whole words', Electronics **35**, 29–31.

Harmon, L. D. (1972), 'Automatic recognition of print and script', Proceedings of the IEEE **60**, 1165–1176.

Ho, T. K., Hull, J. J. & Srihari, S. N. (1990), A word shape analysis approach to recognition of degraded word images, *in* 'Proceedings of the United States Postal Service 4th Advanced Technology Conference', Vol. 1, pp. 217–231.

Ho, T. K., Hull, J. J. & Srihari, S. N. (1994), 'Decision combination in multiple classifier systems', IEEE Proceedings on Pattern Analysis and Machine Intelligence **16**(1), 66–75.

Hochberg, J. (1970), Components of literacy: Speculations and exploritory research, *in* H. Levin & J. P. Williams, eds, 'Basic Studies on Reading', Basic Books Inc., pp. 74–89.

Hull, J. (1988), A computational theory of visual word recognition, Report number 88-07, University of NY at Buffalo.

Hull, J. (1989), Feature selection and language syntax in text recognition, *in* J. C. Simon, ed., 'From Pixels to Features', Elsevier Science Publications, North-Holland, pp. 249–260.

Hull, J. (1992), Incorporation of a Markov Model of language syntax in a text recognition algorithm, *in* 'Symposium on Document Analysis and Information Retrieval', University of Nevada, Las Vegas. 16th – 18th March, 1992.

Johansson, S., Atwell, E., Garside, R. & Leech, G. (1986), The tagged LOB corpus, Norwegian Computing Centre for the Humanities, Bergen.

Kaltenmeier, A., Caesar, T., Gloger, J. M. & Mandler, E. (1993), Sophisticated topology of Hidden Markov Models for cursive script recognition, *in* 'Second International Conference on Document Analysis and Recognition', Tsukuba Science City, Japan, pp. 139–142.

Keenan, F. G. (1992), Large Vocabulary Syntactic Analysis for Text Recognition, Unpublished phd thesis, Nottingham Trent University.

Keenan, F. G., Evett, L. J. & Whitrow, R. J. (1991), A large vocabulary stochastic syntax analyser for handwriting recognition, *in* 'First International Conference on Document

Analysis and Recognition', Saint-Malo, France, pp. 794–802. September 30 – October 2, 1991.

Konstantinides, K. & Rasure, J. R. (1994), 'The Khoros software development environment for image and signal processing', IEEE Transactions on Image Processing **3**, 243–252.

Kopek, G. E. & Chou, P. A. (1994), 'Document image decoding using Markov source models', IEEE Proceedings on Pattern Analysis and Machine Intelligence **16**(6), 602–617.

Kornai, A. (1994), Language models: Where are the bottlenecks?, *in* L. J. Evett & T. G. Rose, eds, 'Language models in Speech and Handwriting Recognition'.

Kundu, A., He, Y. & Bahl, P. (1989), 'Recognition of handwritten word : First and second order Hidden Markov Model based approach', Pattern Recognition **22**(3), 283–297.

Lecolinet, E. (1993 a), Cursive script recognition by backwards matching, *in* 'Sixth International Conference on Handwriting and Drawing', Paris, July 4–7, 1993, pp. 89–91.

Lecolinet, E. (1993 b), Top–down and bottom–up strategies for cursive script recognition, *in* 'Sixth International Conference on Handwriting and Drawing', Paris, July 4–7, 1993, pp. 117–119.

Lecolinet, E. & Crettez, J.-P. (1991), A grapheme–based segmentation technique for cursive script recognition, *in* 'First International Conference on Document Analysis and Recognition', Saint-Malo, France, pp. 740–748.

Lecolinet, E. & Likforman-Sulem, L. (1994), Handwriting analysis : segmentation and recognition, *in* 'European Workshop on Handwriting Analysis and Recognition : A European Perspective', IEE, pp. 17/1–17/8.

Leedham, C. G. (1994), Using forensic handwriting analysis techniques to enhance automatic handwritten script recognition and processing, *in* 'European Workshop on Handwriting Analysis and Recognition : A European Perspective', IEE, pp. 2/1–2/3.

Lorette, G. (1993), Is recognition and interpretation of handwritten text a scene analysis problem?, *in* 'Third International Workshop on Frontiers in Handwriting Recognition', Buffalo, U.S., pp. 295–304.

Miletzki, U. E., Uebel, W. & Schulte-Hinsken, S. (1994), Automated recognition of handwritten addresses on mail–pieces, *in* 'European Workshop on Handwriting Analysis and Recognition : A European Perspective', IEE, pp. 19/1–19/9.

Moreau, J.-V., Plessis, B., Bougeois, O. & Plagnaud, J.-L. (1991), A postal check reading system, *in* 'First International Conference on Document Analysis and Recognition', Saint–Malo, France.

Morton, J. (1969), 'Interaction of information in word recognition', Psychological Review **76**, 165–178.

O'Hair, M. A. & Kabrinsky, M. (1991), Beyond the OCR: reading whole words as single symbols based on the two dimensional, low frequency Fourier transform, *in* 'First International Conference on Document Analysis and Recognition', Saint-Malo, France, pp. 350–358.

Paquet, T. & Lecourtier, Y. (1993), 'Recognition of handwritten sentences using a restricted lexicon', Pattern Recognition **26**, 391–407.

Rabiner, L. R. (1989), 'Tutorial on Hidden Markov Models and selected applications in speech recognition', Proc. of the IEEE **77**(2), 257–286.

Rayner, K. (1978), Foveal and parafoveal cues in reading, *in* J. Requin, ed., 'Attention and Performance', Vol. VII, Lawrence Erlbaum Associates, New Jersey, pp. 149–161.

Rayner, K., Carlson, M. & Frazier, L. (1983), 'The interaction of syntax and semantics during sentence processing: Eye movements in the analysis of semantically biased sentences', Journal of Verbal Learning and Verbal Behaviour **22**(3), 358–374.

Rose, T. G. (1993), Large Vocabulary Semantic Analysis for Text Recognition, Unpublished phd thesis, Nottingham Trent University.

Rose, T. G., Evett, L. J. & Lee, M. J. (1994), Corpus based contextual analysis for speech and handwriting recognition, *in* L. J. Evett & T. G. Rose, eds, 'Language models in Speech and Handwriting Recognition', AISB Workshop series.

Rumelhart, D. E. (1977), Towards an interactive model of reading, *in* S. Dormic, ed., 'Attention and Performance VI', Erlbaum, Hillsdale, New Jersey, USA.

Schomaker, L., Abbink, G. & Selen, S. (1994), Writer and writing–style classification in the recognition of online handwriting, *in* 'European Workshop on Handwriting Analysis and Recognition : A European Perspective', IEE, pp. 1/1–1/3.

Selfridge, O. G. (1955), Pattern recognition and modern computers, *in* 'Proceedings of the Western Joint Computer Conference', MacMillan Co., New York, pp. 91–93.

Senior, A. W. & Fallside, F. (1993 a), An off–line cursive script recognition system using recurrent error propagation networks, *in* 'Third International Workshop on Frontiers in Handwriting Recognition', Buffalo, U.S., pp. 132–141.

Senior, A. W. & Fallside, F. (1993 b), Using constrained snakes for feature spotting in off–line cursive script, *in* 'Second International Conference on Document Analysis and Recognition', Tsukuba Science City, Japan, pp. 305–310.

Shannon, C. E. & Weaver, W. (1964), The mathematical theory of communication, The University of Illinois Press.

Shridhar, M. & Badreldin, A. (1984), 'High accuracy character recognition algorithm using Fourier and topological descriptors', Pattern Recognition **17**, 515–524.

Simon, J. C. & Baret, O. (1989), 'Formes regulieres et singulieres; applicatoin a la reconnaissance de l'ecrituremanuscrite', C. R. Acad. Sci. Paris, Serie II **309**, 1901–1906. In French.

Simon, J. C. & Baret, O. (1990), Handwriting recognition as an application of regularities and singularities in line pictures, *in* 'First International Workshop on Frontiers in Handwriting Recognition', CENPARMI, Concordia University, Canada., pp. 23–38.

Srihari, R. (1993), Personal Communication.

Srihari, R. K., Ng, S., Baltus, C. M. & Kud, J. (1993), Use of language models in on–line sentence / phrase recognition, *in* 'Third International Workshop on Frontiers in Handwriting Recognition', Buffalo, U.S., pp. 284–294.

Srihari, S. N. & Bozinovic, R. M. (1987), 'A multi-level perception approach to reading cursive script', Artificial Intelligence **33**, 217–255.

Srihari, S. N., Govindaraju, V. & Shekhawat, A. (1993), Interpretation of handwritten addresses in US mailstream, *in* 'Second International Conference on Document Analysis and Recognition', Tsukuba Science City, Japan, pp. 291–294.

Stanovich, K. E. (1980), 'Toward an interactive–compensatory model of individual differences in the development of reading fluency', Reading Research Quarterly **XVI/1**, 32–71.

Summers, D. (1991), Longman/Lancaster English language corpus: criteria and design, Technical report, Longman.

Tappert, C. C. (1982), 'Cursive script recognition by elastic matching', IBM Journal of Research and Development **26**, 765–771.

Tappert, C. C. (1984), Adaptive on–line handwriting recognition, *in* 'Proc. 7th Int. Conf. Pattern Recognition', pp. 1004–1007.

Tappert, C. C., Suen, C. Y. & Wakahara, T. (1990), 'The state of the art in on-line handwriting recognition', IEEE Transactions on Pattern Analysis and Machine Intelligence **12**, 787–808.

Thomas, R. C., Hanlon, S. J. & Boyle, R. D. (1990), Directed reading, *in* 'First Australian Conference on Cognitive Science', University of New South Wales. 4-7 November 1990.

Tulving, E. & Gold, C. (1963), 'Stimulus information and contextual information as determinants of tachistoscopic recognition of words', Journal of Experimental Psychology **66**, 319–327.

Verikas, A. A., Bachauskene, M. I., Vilunas, S. J. & Skaisgiris, D. R. (1992), 'Adaptive character recognition system', Pattern Recognition Letters **13**, 207–212.

Waibel, A. & Lee, K.-F. (1990), Readings in Speech Recognition, Morgan Kaufmann, San Mateo, California.

Wells, C. J., Evett, L. J., Whitby, P. E. & Whitrow, R. J. (1992), The use of orthographic information for script recognition, *in* R. Plamondon & C. G. Leedham, eds, 'Computer Processing of Handwriting', World Scientific, Singapore, pp. 273–291.

Zhang, Q. & Boyle, R. (1991), 'A new clustering algorithm with multiple runs of iterative procedures', Pattern Recognition **24**(9), 835–848.

Zipf, G. K. (1945), 'The meaning–frequency relationship of words', Journal of General Psychology **33**, 251–256.

Appendix A

LEXICON

This appendix gives a list of the words used in the experiments described throughout this thesis. The list of words was generated from the Lancaster Oslo Bergen corpus of English language (Johansson *et al* , 1986).

The words are listed in order of frequency in the corpus.

the
of
and
to
a
in
that
is
was
it
for
he
i
as
with
be
on
his
at
by
not
had
this
but
from
have
are
which
she
you
her
or
they
an
were
there
we
been
one
all
has
their
would
when
no
if
so
will
him
who
more
can
said
out
do
what
about
up
some
my
only
could
them
time
into
than
me
two
then
other
its
mr
these
now
may
any
should
made
first
over
very
our
like
new
did
must
such
after
man
much
years
before
most
where
many
well
even
also
being
those
people
way
back
too
good
little
down
how
last
between
your
just
work
still
see
know
make
might
because
through
same
own
long
get
here
go
great
three
never
life
year
come
both
old
another
without
us
again
each
used
under
came
take
say
found
though
men
does
thought
right
day
world
went
while
course
far
part
against
think
house
few
however
per
off
home
since
away
given
fact
small
going
always
left
case
himself
put
use
every
during
young
london
government
place
something
told
quite
until
number
almost
later
find
hand
sir
less
why
asked
give
once
nothing
yet

present
general
whole
large
end
possible
took
got
point
taken
days
children
often
next
look
better
side
looked
country
head
face
upon
school
perhaps
eyes
enough
need
rather
mind
local
party
form
things
british
war
second
water
cent
although
seemed
set
having
high
night
important
want
four
best
different
already

ever
whether
room
known
seen
national
shall
felt
together
within
public
miss
knew
council
thing
further
certain
door
show
round
order
done
called
church
become
saw
turned
power
money
towards
word
means
let
words
west
least
really
group
among
moment
girl
week
brought
book
able
love
service
view
kind
whose

matter
either
held
am
age
necessary
system
light
half
seems
therefore
mother
help
gave
above
air
mrs
full
behind
today
common
anything
times
tell
interest
early
area
family
white
table
probably
reason
labour
john
father
question
voice
lord
itself
field
car
state
others
ago
effect
began
themselves
social
true
outside

position
open
keep
sense
change
soon
across
town
company
five
members
making
indeed
office
thus
wife
name
particular
act
sometimes
body
along
taking
meeting
britain
french
feel
seem
problem
main
example
education
child
around
period
woman
free
england
experience
special
modern
heard
value
usually
political
hands
became
terms
particularly
certainly

morning
idea
six
english
turn
shown
short
read
mean
death
committee
result
minister
line
american
business
art
food
past
feet
following
due
yes
difficult
added
century
whom
alone
leave
countries
real
land
wanted
trade
hard
society
problems
concerned
music
doubt
stage
human
sure
god
cases
near
knowledge
increase
hope
forward

president
road
rate
months
level
close
story
herself
clear
several
results
except
doing
development
talk
gone
nor
market
looking
living
control
city
working
women
united
type
top
policy
pay
various
total
industry
stood
play
job
heart
everything
cut
building
says
future
boy
reached
required
kept
believe
answer
private
hours
evidence

ten
met
bring
university
figure
east
section
expected
subject
south
paper
carried
police
longer
likely
hall
black
start
meet
feeling
coming
big
history
followed
cost
conditions
personal
deal
person
movement
complete
bed
amount
nature
street
court
considered
board
influence
friends
union
secretary
individual
getting
figures
evening
simple
decided
similar
return

lost
action
suddenly
royal
economic
bill
situation
king
students
sort
lay
easy
provided
greater
blood
training
myself
houses
german
else
trying
married
late
force
fine
using
tax
sat
minutes
front
especially
natural
finally
written
works
try
run
nearly
live
instead
earlier
low
friend
available
strong
health
wrong
tried
oh
film

conference
call
started
red
ground
authority
sea
saying
peace
paid
law
values
stand
sent
research
methods
language
account
understand
thinking
sound
parents
foreign
commonwealth
books
bad
support
recent
original
opened
material
cold
yesterday
unless
rest
picture
theory
received
final
process
method
lines
attention
window
truth
report
questions
including
higher
beyond

schools
remember
quickly
lot
hear
described
built
bit
below
average
waiting
solution
reading
purpose
hotel
africa
wide
success
obtained
makes
considerable
ask
appeared
study
shows
middle
information
george
desire
design
chance
trouble
son
dark
college
changes
allowed
lead
heavy
fig
farm
centre
basis
visit
dead
association
ordinary
major
lady
rose

range
hundred
anyone
agreement
walked
moved
led
fell
colour
beginning
based
increased
hardly
happened
flat
date
care
returned
ready
points
paris
class
provide
passed
chapter
third
produced
previous
letter
floor
hold
groups
deep
berlin
price
million
length
husband
apart
surface
simply
prepared
military
member
hour
comes
charles
spent
size
plan

capital
accepted
suggested
showed
needed
hair
green
consider
appear
according
stop
rule
inside
happy
generally
europe
staff
jones
interesting
impossible
easily
defence
completely
workers
wall
opinion
ideas
wish
village
test
recently
gives
germany
county
america
agreed
surely
western
population
normal
miles
importance
hospital
fire
character
addition
weeks
rise
obviously
forms

clearly
term
states
single
queen
practice
north
mark
fall
direction
throughout
standing
prime
parts
chief
summer
giving
fish
expression
boys
wrote
wind
technical
merely
lower
european
demand
arrived
arm
station
someone
production
former
difficulty
behaviour
worth
usual
stay
seven
principle
pattern
neither
marriage
lived
interested
girls
garden
france
continued
cause

writing
services
programme
numbers
letters
difference
central
areas
whatever
treatment
serious
science
regard
move
model
fair
extent
suppose
scale
record
practical
list
learn
foot
space
quality
published
india
buildings
actually
soviet
note
meant
instance
brown
african
supply
speech
series
medical
follow
fear
earth
drink
caught
blue
tea
stopped
spoke
russian

physical
industrial
fully
appears
doctor
developed
worked
ways
straight
obvious
interests
established
club
classes
authorities
attempt
accept
news
hot
distance
died
closed
charge
remains
piece
needs
manner
income
effective
direct
current
circumstances
brother
agree
standard
slowly
slightly
scheme
regarded
places
ought
latter
immediately
expect
everyone
essential
degree
claim
chairman
tests

speak
remain
growing
exactly
entirely
effort
associated
approach
applied
temperature
river
pretty
pass
notice
meaning
function
eye
attitude
speed
reported
remained
presence
placed
ones
official
offer
nuclear
firm
duty
decision
talking
reduced
progress
older
ministry
milk
leaving
leading
knows
baby
afternoon
write
style
step
spirit
limited
commission
army
scene
reasons

Appendix B

TEST SENTENCES

This appendix gives the sentences used in the experiments described in Chapter 7. It can be seen that some of the ‘sentences’ are in fact clauses and not full sentences. This is due to the extraction method where any group of words delimited by a full stop was considered a sentence. The sentences were extracted across the whole corpus.

This list is divided into the ten sets of sentences used in the experiments. The lexicon size refers to the first n words in the lexicon listed in Appendix A, and the sentences are constrained to contain only words in that lexicon.

Sentences from 100 word lexicon

a
all did
and he is
and now
and so on
and so would you
and that is not all
but it is not
but it was not to be
but not to me
but that is all over
but that is not all
but there is more to it than this
but there was more to it than that
by
he did well
it did
it may be
it was
it was like that now
like me
most of it
no
no more
not at all
not so
now
now they are not
one
or can you
out and about
over and out
she must not
so he did
so it would
so there
that could be it
that was it
the first two years
then he
there was no more
this is it
this may be so
this time
we

Sentences from 200 word lexicon

a few will do
after two long years
again
all right
and away you go
and it made me think again
and she was right
and so it went on
and there was something she could do
another two to go
as
but all was still
but he never did
but of course he could see nothing
but she could see no way to get out of it
but the first place she made for was his home
but they must be good
but this is not work
far from it
for men may come and men may go
from now on they would be on their own
going in and out
going out into the world
good
good will man
he could not think so
he did as he was told
he does not see me
he found no one
he had come to the right place
he said nothing
he said nothing more
he went out
he will do the same
here at last
himself
his last years
his life was almost over
it all came back to her
it came about in this way
it did not know this
it did not take him long
it had nothing to do with her
it was good to see it go
it was just the same
it was no good
it was so here
it will be something to get well for
it would not be the last
just my work
just the once
just this
more down
not little old me
nothing
nothing in it at all
now off you go
number and case
of course not
one can do it right where he is
one can do nothing with them
or even later
she found out
she was right
she was very good about it
she was very old
she went over
so much to think about
so she had been right
so we are back where we were
some day a man will
something was on it
that was the first day
the other thought came back
then we can work on and away home
there are too many
there had been so much to see
there was nothing to be said
there was still so very far to go
there were only a few people about
these came my way later in life
they will get them
think what you like of me
this time of year it always is
this was his day
three
to know is also good
us
use any two or three
we both will
we three
we went in
when we get home
with that he went out
work again
you are still young
you can take it from me that he can and does
you know that
you will do very well without it

Sentences from 300 word lexicon

a girl on her own
all too often this is not the case
and every time we turned up something new
> had taken place
and together they did
at least not yet
at the present time no such order has been made
because she was second best
but a country is made up of people
but he was not done
but that was the end
but there was the world without as well as
> the world within
but we got one at last
children
different
even when we came it was different to now
ever
general
good form
good girl
he brought her to the party too
he felt that it was better like that
he found that she had done all and more
> than he had asked of her
he himself did not want to go at all
he knew at once that she would not
he looked at her
he looked at her for a long time
he looked down at her
he looked round
he looked round at her
he looked up
he means it
he turned and left the room without a word
he turned round
he turned to face her
he turned to go
he turned towards her
her face turned
here she had first been in love
his mind was made up
in the first place he must want something
it had been three years though before she had
> found another love
it had brought them together
it was all over and done with
it was not enough
last years at school
let them see him
love
money
never saw a thing
no need to look for that
nothing was certain
now she has got it
on a point of order
on to the next room
one last word
one thing is certain
people were so kind
perhaps she would have children
she could not think what had brought him
she had done it for so long
she looked him up and down
she thought of his words
she would find some way round them
> when the time came
so he had better say so
that was better
the book was always the same book
the end
the end came up
the mind
the moment had come
the new look
the show must go on
the war
the war years
then church
then he looked up at me
then she turned to the children
there was no need
there were few people in the church
they have four children
they looked at each other
think what you need
this is an order
this should be done
this then was her room
this was the moment
those days are now over
we are a public service and an important one
we have been let down
we have never taken that view
we left without another word
we shall see
you have seen it
you look
you would be better without me

Sentences from 400 word lexicon

a child
a common view
a modern voice
a week later the mother came back
all my love was for his wife
another point of interest is the money asked
around the house
but for others it may not be possible
but her father had seen
but it was true
but she did tell him all the same
but they made no sense
common sense
education
he could feel it
he held it up
he held out his hand
he is this body
he was from a good family
heard car
her car was already outside the door
his father would see through it though
his name
in the car he said nothing
indeed it has always been so
indeed so
it might be open
it might help us
it seemed full
it was an old woman
it was now five
it was to be free
it was true
just more sense
look at it again act by act
making money
meeting
must be true what they say
my father went with me
no change
no one seems to know
now go and change
open it up
said he would help
she has put up the value of her money
she held up her hand
she is a good woman
she looked around
she looked around her
she probably did
she put both hands to his face
some change must come
soon the head can be seen
taking it back
that is not quite true
the car was on time
the door was open
the full table
the modern view
the most important thing in the world is the family
the old woman had eyes that could see
the terms
the whole family came to like him
their first meeting
there was nothing particular about it
these were left far behind
they held nothing
this is not true
this is the main point really
this man had been at the meeting too
this was all part of the act
we are of the true world
we made love in those few days many times
we think of it as something special
you may need my help

Sentences from 500 word lexicon

a difficult year
a very short story that is not what it seems
all it wanted was to be left alone
and so was the girl herself
and some people get down to the job themselves
answer this today
believe it or not
best forward
but in his old age he did not feel so sure
but she knew that was not the answer
certain things become very clear
common market
death for no reason
difficult
due back in a few days
everything is over
far more than ever she could hope to give him
food was short
for the moment everything had been said
future effect
future problems
gone in
hands and feet
he asked for the full name of the boy
he got what she wanted
he has no business here
he is concerned with them as they are and
> also for what they are
he kept himself to himself
he knew he was doing the right thing in not
> doing so
he looked forward to meeting her
he made no answer
he only has to make sure you say it all
he reached for her hand
he stood quite still
he wanted me to act
he wanted to think
he was alone
he would not leave the old man
his business
his policy
how to turn your work
it felt as if she were alone in the world
it has long been clear party policy that this
> should not be done
it is something that has to be done because
> the other side is doing it
it looked real
it made him more human
it says so in the book
it was all quite clear
it was enough just to talk
it was morning
it was very near his old home
leave it be child
might be hours
my heart was too full for words
now all that seemed gone
now read on
on art
only a matter of hours now
people talk far too much and say the same
> things over and over again
policy
private company
real problems
results
service industry
she could not answer
she is so sure about things
she seemed concerned
she stood up
she told herself that she would do her best
> never to see either of them again
she was not alone
so she was all alone
sure
that would make her look forward all the
> more to the next day
the
the answer is
the answer is no in both cases
the case for art education
the girl stood very still
the heart of the matter
the light was gone
the morning began just the same as the others
the next she was gone again
the other did not answer at once
the place and its people were to play an
> important part in my life
the second stage of labour is over
the story so far
the type does not change much
the woman had gone by
the words were clear
the world of music
the young woman living alone
there need be no doubt about that
there was no answer
they are not looking at us
they believe that what they are doing is the
> right thing to be doing
they seemed to come from down the road
they wanted to help
this is a difficult question
thought for food
to
to talk about
very few women have
wanted it right away
we must end the idea of war
what his story will be
yes
you have not known her long enough for
> it to mean anything to you

Sentences from 600 word lexicon

a simple enough question
all that day more were coming in
all that was nearly a century ago
as often as not they were not even married
be natural
better make it black though
big business
but figures alone can not tell the whole story
but he suddenly felt a great deal better
but he was too late
but she decided she could not face it
coming out now
getting it over
he followed her into the room near by
he might return
he must be left with something to live by
he thought about that sort of death for a moment or two
he turned suddenly
he used to come every day and talk to me for a few minutes
he was coming up again
he was told it might be a fine
he was wrong
he would have made it easy but for the little man
health
her friends
his father lay on his back
his nature
how to make friends
in themselves they will not increase the number of students at all
it has made a good start
it was a police car
it was coming now
it was hard to figure her
it was not part of their blood
just you take it easy
most had been married for between two to six years
most of the time he lay on his back with his eyes open
movement of thought
never has the health of children been better
not very likely
often two friends will work at it together
she got up and went to the front door
she knew now that something was really wrong
she met his eyes then
she was in black
she was trying to sort things out in her mind
so wrong
social and personal
sometimes he would be there for three or four minutes
sometimes you do meet some one who says little himself
strong case
ten years
that sort of way of life
that was what the figure looked like
that was what you would have expected
the bed was still made up
the bill is a short one
the day is coming when he will do so
the friend
the health service will cost more
the influence of the social group
the paper
the same with history
then came on board
there had been a feeling of hope then which had gone later
there was no one on board when she went down
there were ten minutes left
they are in the big money
they will not work in union
this is not the complete answer
we can not leave the situation like this
we have considered this development
we was just getting down to business
what was getting him down in a big way was being
> told what to do about his own children
when he heard her coming he went to the far end of
> the deal table and sat against it with his back to the door
wrong
you have done nothing wrong

Sentences from 700 word lexicon

a considerable effect

a man of peace

a solution

a sound idea

and there is enough truth in that to set you thinking

ask for him

authority

cold war front

dead

design

design methods

first they see the commonwealth as a whole

full support is necessary

general information

he came round and opened the back door of the car

he could remember how he had not been able to understand
> why his father did nothing about this

he felt that the situation was getting beyond him

he had the door of the car opened

he knew their truth as few could do

he may find his wife cold

he must be heard with attention

he opened a door

he opened the door and went in

he tried each in turn without success

he was like that about a lot of things

he went quickly to the door

her eyes opened

high and wide

his mother came into the hall as he opened the front door

it seemed hours before her chance came

it was a little bit of everything

just peace

just three questions

like the sea

methods

my case got off to a bad start

my chance had come at last

no one is very good or very bad

nor had she told her parents that she was coming

open the window before you leave

peace and war

peace policy

perhaps we had read too few books

purpose

report

several others must be dead

she had her truth

she is still subject to the control of her mother in law

she knew that was how she appeared

she opened door after door

she rose at once

she was dead

so much for the new material

so were the people at the table beyond that

something made her look out of the window

sometimes like a boy lost in the dark

still no sound

still no sound from outside

success

that is bad enough

the answer would seem to be that it is a bad thing before death
> and a good thing after

the common way of using one is to point it at the subject
> and take a reading

the door opened before they reached it

the hard way of peace

the new books

the old lady went away

the past was dead

the sea house

the sea is not full

the work of the schools

then he tried it out for sound

therefore they were to love truth and peace

they rose

this is a reading world

this is indeed a success story

this is only half the truth

this is your moment of truth

this was a lead

this was beyond me

too cold

unless they tried to come back

waiting

we have a

we hope to hear more of this meeting between our two countries

yet even this did not yet trouble me very much

yet it was still bad enough

you and me and the rest of us

you are of the heavy world

Sentences from 800 word lexicon

a good rule
a letter
almost every day we hear something about its importance to us
and she walked on
and then he walked off the stage
and then suddenly it happened
but it is already out of date
but nothing happened
but only a few short weeks ago it was a different story
but surely that is the wrong way to go about things
by our political staff
care and study
chapter five
chapter four
chapter one
chapter six
chapter ten
chapter two
consider the first question
deep study
deep water
good beginning
he moved a little away from her
he needed her
he said that there were many forms
he spent most of his life among these people
he thought of the years a long time ago when his father had seemed happy
he walked all morning
her hair had just been done
her mother would provide all the material
importance of example
in an hour
it came to just over six hundred miles
it could easily be all or nothing with him
it happened this way
it is an interesting question
it was important this last figure should be increased
it was quite interesting
it was the first time he had done so in over three weeks
later he returned
never let it stop
no character
not for months had it happened
opinion
probably he had had inside knowledge from one of them
put to the test
range
ready for life
round and round that field we walked all day
she felt sure she had at last found a man who would make her happy for life
she had looked at him and then walked quickly away
she led the way into his room
she moved across the room
she told me what had happened
she turned about and walked across the hall
she turned and walked back to the road
size of the market
so he opened it and walked in
something moved behind him
that was impossible
that was the beginning
the evening passed all too quickly
the letter
the problem we shall consider is the following
then she got up and moved towards the door
there were people inside
there were two men inside
they have been going on for a long time and much money has been spent
they last for weeks in water
they were back in just over an hour
this gives a very important group of people
this is obviously common sense
time passed
time passed so quickly
to him it was a war of ideas
too wide a range
we accepted this
when he returned an hour later the car was gone
when she is needed she is there
you are out of date
you might just as well try to change the colour of your hair
you needed them no more
you would wish me to report on them

Sentences from 900 word lexicon

always keep on the move
and away you go into your speech
arrived early
at least you learn something about human nature
but at last he spoke
but he knew she must like one of the boys on his either side better than him
but this is a matter of principle
consider the difference between
do the same with your speech
he arrived at the door and stopped
he caught her arm
he had someone to put in its place
he moved over to the window and looked down upon the garden
he said no more until they had stopped right outside the flat
he stopped
her
here the groups could live without so much fear
it is much more than even the development of a single new town
it was concerned with the commonwealth technical training week
  > which opened yesterday
it was now ten past seven
it was the people who lived in it
it was the police station
it will also demand a big increase in staff
last term had been bad enough
marriage rate
neither man looked at her
neither side can go forward alone
never before has there been such a big programme of school building
not so this great power station
now fear caught her
now she knew how much it was worth
old boys
on the record
people on the move
perhaps it spoke of both
play space
practical
public services
seven years
she caught hold of his arm
she has a foot of water in her
she only arrived today
she was too far away for him to see any expression on her face
she went into the garden
social services
standing room only for nothing to pay
standing with her back to it was a woman
tea
that meant the whole of a long day to live through first
the difference
the flat fish does not want to rise
the future of technical education
the government are sure that this is the right principle
the music stopped
the order book is a record
the other stopped
the programme
the trouble with long standing problems is that most people get used to them
the wind of change
the woman walked in the same direction on the other side of the road
the world of science
there are four letters for you
they could see how much it meant to him
they have seven children
they should have left them to fall down
this book is worth reading
this week in your garden
three on short list
treatment of boys
was clearly shown
we do have something to drink
we had to stay by her in any case
we must learn all we can about them
we stopped all that

Sentences from 1000 word lexicon

a form on which to claim will then be sent to her
afternoon tea
all the rest is done by the eye and hand
and died
and he can set a whole group talking
and the reason was obvious
at some time or other you will speak in public
baby book
but it is essential that the government should stand firm
but when they were outside neither of them made any attempt to
  > turn in the direction of her hotel
certainly one had the right to expect better
everyone else had the power
half an hour later he reported back
he began to rise slowly from the table
he closed his door
he got to work immediately
he had lines to speak
he knows what he will get from his children
he knows what you are going to say
he may be older
he turned slowly and went out
he worked for me for two years
he would speak to the secretary about it
her fear of him was pretty obvious
industrial health
into the eye of the wind
it knows no fear
it remained thus for a whole summer
it seems pretty obvious that it is the job of a government to
  > look after the needs of the people
it seems to be obvious
it was growing cold
local authorities are caught both ways
make the paper speak
ministry of health
no news
nor could any power on earth stop her from talking
not a word about the baby
not exactly
not most effective
nothing obvious
on duty
on the whole men accept their places
one is due to style
one of the small ones
one of those places where we used to go
other interests
ought to be a better way of doing things
our children must first have something to say before they can write it
our family doctor
people taking charge
placed on record
progress in science
royal commission
she would visit her parents that afternoon
so everyone knows where he is
so much is established
some parents can not accept this change in their children
talking about health
that was their attitude
the authorities and the world around
the first step
the industrial court
the ministry is in a position to know more than even the best
  > local education authority
the point appears to come out very clearly in those cases
  > where we make a decision
the reasons are quite simple
the report of the public schools commission was followed
  > by the public schools bill
then slowly she turned and looked up into his face
there appears to be much we can learn from each other
there are many ways of being hard to understand
there was more than enough land for their needs
there were three main reasons for this
there will never again be a club like it
they were told to go straight on
they worked hard for the rest of the morning
this is good news
this is the necessary first step
this still took place entirely on her left hand
this system worked well until last year
this time his offer was accepted
this was of particular interest for two reasons
time to start talking
two men were talking
under no circumstances
we agree
we can charge what we like
where the baby was
within three days she had worked it all out
you see what he is talking about