Stochastic Segment Modeling for Offline Handwriting Recognition

A dissertation

submitted by

Premkumar Natarajan

In partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in

Electrical Engineering

TUFTS UNIVERSITY

Date

May 2012

ADVISER: Dr. Joseph Noonan

Abstract

Much of the world's information in industry/office settings continues to be initially recorded in the form of handwritten documents. Examples include notes taken at meetings, lectures, and even medical records. One important drawback inherent to such handwritten notes is that the information contained in them is opaque to electronic data management systems. That drawback can be overcome by employing technology that is capable of automatically generating electronic transcriptions of the handwritten text. Such technology is referred to in the research literature as *offline handwriting recognition* technology.

Over the past decade, the hidden Markov model (HMM) has become the paradigm of choice for the task of offline handwriting recognition. In this dissertation, we present a new Stochastic Segment Modeling technique for recognition of offline handwriting. The technique builds upon an existing HMM-based system and incorporates broader, long-term context into the recognition process. Such long-term context is typically encoded in the form of structural features extracted from segments of handwritten text.

Traditionally, structural features have been used only in recognition approaches that rely on accurate segmentation of words into smaller units (sub-words or characters). However, such segmentation-based approaches do not perform well on real-world handwritten images, because breaks and merges in glyphs typically create new connected components that are not observed in the training data. To mitigate the problem of having to derive accurate segmentation from connected components, we present a novel framework where a HMM-based recognition

system trained on shorter-span fixed-width-window features is used to generate candidate 2-D character images (the “Stochastic Segments”). A separate classifier that uses structural features extracted from the stochastic character segments generates a new set of scores that are independent of the HMM scores. Finally, the scores from the HMM system and from the structural feature classifier are used in combination to generate a final hypothesis that is better than the results from either the HMM or from structural matching alone. We demonstrate the efficacy of our approach by reporting experimental results on a large corpus of handwritten Arabic documents.

Dedicated:

To my parents and to my brother for their abiding faith in me

And to my wife Vidhya (Subbulakshmi) for her love and her selfless support of my every endeavor

Acknowledgements

Our accomplishments are shaped as much by the generous help and advice of our mentors and friends as they are by our own efforts. It is my good fortune that I have been blessed with gifted mentors and talented, caring friends. Many of them have had a substantial direct or indirect role in guiding me through my academic pursuits.

My parents taught me, through example, the importance of perseverance. Growing up, I saw them work tirelessly yet cheerfully to give me and my brother a good start in life. There is no question that since those early years my efforts at school and elsewhere have been motivated by the strong desire to not let their efforts be in vain. Over the years, my brother and I have been blessed with their patient mentoring and constant encouragement of our ventures.

After coming to the US to pursue graduate studies and a career in research and development I was, once again, fortunate in the mentors I found: Dr. Joseph Noonan, Dr. John Makhoul, and Richard Schwartz.

It is popular knowledge that Dr. Noonan is a gifted teacher; what is less known perhaps is his genuine concern for the wellbeing and personal development of his students. He opened my eyes to "technical pragmatism," i.e., how to go from an equation to a practical solution. I greatly cherish his guidance through my Master's degree years and his subsequent encouragement of my Doctoral studies. It is my singular good fortune to have come to know him.

Dr. John Makhoul is one of those all-too-rare human beings – a technical giant and an exceptionally talented mentor. Like all champions, he has the ability to effortlessly kick his game into the next gear when required. Over the years, just observing him in action has been an amazing educational experience. In the years since I joined BBN he has served as mentor, role model, teacher, and friend. My immense gratitude to him is best demonstrated by continuing to advance the culture of excellence at BBN which he has played a key role in establishing.

Rich Schwartz was my first technical supervisor when I joined BBN. Rich is, simply put, a technical genius in the area of statistical learning. Rich is also a phenomenal teacher. It is impossible to convey fully the impact of the technical education that I have received from Rich. As far as the work in this dissertation is concerned, the training I received from Rich is, without question, the dominant shaping force.

Through the last decade, I have had the privilege of working closely with Rohit Prasad on many exciting technical projects. Rohit is another exceptional technical talent at BBN and my conversations with him have helped preserve and advance my technical edge and agility. Importantly, without his support, it would have been impossible for me to have found the energy and time to work on this dissertation. I am grateful to Krishnakumar Subramanian and Anurag Bharadwaj for their generous help and work on experiments. Michael Decerbo and I had many enjoyable discussions during a time when we were the only two researchers working on OCR at BBN and it is still a pleasurable experience to reminisce over our discussions.

Several other people have helped me over the years including Dr. Dennis Fermental, Dr. Sos Agaian, George Preble, Warren Gagosian, Yvette Landry, Paolo Forte, Gilon Miller, and Janie Bess. To all of them, I offer my heartfelt gratitude.

Working on this dissertation was often a demanding, candlelight journey; one that I could not have undertaken with the sustained support of my wife Vidhya and my parents-in-law, whether it was taking care of our delightful girls (Latha, Madhu, and Ramya), the delicious food on the dinner table, or the timely offerings of myriad refreshments.

Table of Contents

1 Introduction

Notwithstanding the ubiquity of handy electronic devices, such as tablets and notebook personal computers, the pen and paper continue to remain popular mediums of recording information in professional, industrial, and academic environments. Important everyday information, such as road signs, street names and address information, billboards, posters, and banners, will also continue to exist in "hardcopy", i.e., non-digital format. In today's world, where convenience equals instant electronic access to information, there is a pressing need to store and archive all available information within unified electronic databases so that they can be easily and effectively accessed when required. Indexing hardcopy information to make it searchable and accessible through a computer system requires an automatic capability for generating electronic transcriptions of the associated text content.

Since the middle of the 20th century, researchers have responded to the need for automatic machine reading and transcription of text by attempting to combine a digital image acquisition device such as a scanner or a camera with software that processes the digitized image to automatically produce a transcription of the text content of the image. When the acquired digital image is of a machine-printed text document, the processing software is often referred to as Optical Character Recognition (OCR) software. When the acquired digital image is of a handwritten text document, then the processing software is variously referred to as Intelligent Character Recognition (ICR) [Kim88], Hand Writing Recognition

(HWR), or, sometimes, just Handwriting Recognition (HR). In this dissertation we will use the acronym HWR to refer to handwriting recognition.

In the case of handwritten text a further subdivision may be considered between handwritten text that is natively electronic in nature and text that is natively available in hardcopy form (i.e., paper). The former is often referred to as *online* handwriting and the latter as *offline* handwriting [Pla00, Sei96]. Of the two, recognition of online handwriting is, in principle, similar to the problem of speech recognition and is by far the easier task. In fact, a fair claim can be made that for many popular applications such as note-taking with Microsoft tablets, online handwriting is largely a solved problem for English.

In contrast, recognition of offline handwriting, regardless of language, remains one of the most challenging pattern recognition tasks in the language processing field. Many of the challenges in offline handwriting recognition have their genesis in one fundamental difference between the two modes of writing: online handwriting is essentially a one-dimensional (1-D) signal (i.e., a function of one independent variable) whereas offline handwriting is a two-dimensional (2-D) signal. Online writing is accomplished using an electronic stylus and pad which record the information as a time-ordered sequence of (x, y) points whereas offline handwritten documents are only available as a set of (x, y) points with no natural ordering or structure other than the spatial organization of the text pixels. Other significant attribute differences that add to the challenge of offline HWR relative to online HWR include:

1. **Stroke information:** Stroke information is reliably known in the case of online handwriting. In the case of offline handwriting the computer system has to first "guess" which of the pixels in the scanned or camera-captured document are parts of a text stroke versus which pixels belong to the background.

2. **Pen up/down information:** Electronic writing pads that are used in combination with the stylus capture dynamic pen-up/down information that has been shown to be very useful for improving recognition accuracy.

3. **Controlled generation process:** System designers have control over the type of electronic stylus and writing pad used in producing online handwriting. It is not possible to impose or assume any such constraints in the case of offline handwritten documents and offline recognition systems need to deal with a diversity of stylus types.

4. **Variability:** In the case of offline handwriting, not only do the types of styli used vary widely (ball-point pens, ink pens, felt-tip pens, pencils), but so does type of paper medium on which such styli are used. Furthermore, independent of the quality and texture of different types of paper media, the background itself can vary - the paper might or might not have horizontal rules, it might or might not have a grid background, and, if both sides are used for writing, there could be bleed-through of ink from the other side. Finally, paper can get folded, creased, crumpled, and be subjected to other real-world acts that make it a noisy medium of information in stark contrast to the controlled, clean

generation and acquisition process provided by the electronic stylus and its accompanying electronic writing pad.

In considering machine performance of a cognitive task, it is often interesting to consider human performance on the same task. Often, human performance on a task serves as the ultimate target for any machine-based execution of the same task. In the context of this dissertation, a natural question to ask is this: "*How well do humans do when given the task of recognizing handwritten documents authored by an unfamiliar writer?*"

As it turns out, this question has been studied and answered, in part, by researchers in the field of cognitive science and psychology and the following excerpt from [Ede90] compactly summarizes salient facts from experimental studies of human reading ability.

"*In comparison, people recognize correctly 96.8% of handprinted characters [Neisser and Weene, 1960], 95.6% of discretized handwriting [Suen 1983], and about 72% of cursive strings (see [Edelman 1988 appendix 1]).*"

Clearly, the recognition of unconstrained, cursive offline handwriting is a challenging task even for humans to perform.

We conclude this introductory chapter with a description of the structure of the rest of this dissertation. In the next chapter, Chapter 2, we present an extensive survey of the state-of-the-art in offline handwriting recognition. In Chapter 3, we describe a script-independent, hidden Markov model (HMM) based methodology for handwriting recognition and its embodiment in the BBN Byblos system.

Next, in Chapter 4, we present experimental results that characterize the performance of the BBN Byblos HMM-based handwriting recognition system on English and Arabic handwritten data sets. In Chapter 5, we introduce the stochastic segment modeling approach, both in terms of its probabilistic framework and an engineering implementation that is based upon the BBN Byblos system. Subsequently in Chapter 6, we present the results of an extensive set of experiments designed to assess the usefulness of the proposed SSM approach for offline HWR. Finally, in Chapter 7, we discuss the key conclusions of our research effort and highlight avenues for further research using the SSM framework.

2 Survey of the State-of-the-Art in Offline Handwriting Recognition

Despite the richness of the associated technical challenges, recognition of unconstrained, natural offline handwritten text has received limited attention from pattern recognition researchers. A survey of the literature reveals that the only two tasks in offline handwriting recognition that have attracted widespread research attention are the tasks of recognizing handwritten addresses on postal envelopes and recognizing the handwritten strings in bank checks. While a rich trove of robust and accurate techniques have been developed for the postal address recognition [Bur93, Sri93, Kor97, Sri97, Kim98] and check recognition applications [Heu97, Kim97, Kor96], their general applicability is severely limited because of their reliance on constraints that are unique to these applications. For example, in the case of postal address recognition one can use the known layout of the text on the envelope to robustly segment words and to determine their type (person name, street name, city, state, and zip code). Further processing of the words relies on applying appropriately constrained vocabularies and exploiting redundancy. Specifically, the zip code, city, and state fields contain reinforcing redundant information that can be used to improve the overall recognition accuracy. Similar obvious layout and redundancy constraints recognition are available in the case of checks and enable robust, accurate recognition of handwritten numerical strings (courtesy amounts in checks, zip codes, etc.) [Kim97, Kim 98, Kor96].

Another limitation of previous and much of the ongoing work in the recognition of offline handwritten text is that it is focused on language or script-specific

systems. Furthermore, even the set of languages that have been investigated is quite limited and a survey of recent work in handwriting recognition shows that the vast majority of research efforts in offline handwriting recognition are focused on English and Arabic.

In one of the early learning-based works on the recognition of unconstrained, offline cursive handwritten English text [Sen98], Senior outlines a neural network-based methodology for recognition of offline handwritten text in English. While the work in [Sen98] is sophisticated in terms of the recognition techniques that are proposed and used, Senior's work presumed the ability to segment individual words with very high accuracy and the experimental analyses in [Sen98] were performed using a corpus of very limited complexity – a single-writer corpus of documents in which each word was written within a wide border of white space to facilitate nearly flawless segmentation with a simple technique. Furthermore, the vocabulary comprised only 1334 unique words and included all the words in the test set. Finally, while the work investigates Recurrent Neural Networks (RNN), Time-Delay Neural Networks, and simple vector quantized (VQ) HMMs, an assertion is nevertheless made that the "scanned image *must* be segmented into separate words" (Chapter 4 of [Sen98]) before further processing can occur (emphasis added). As will be made clear by the following, that assertion has since been proven unnecessary by the successful application of HMMs to the task of offline HWR [Nat06].

In [Vin04], Vinciarelli and colleagues dealt with offline recognition of unconstrained English handwritten texts using HMMs and statistical language

models. Interestingly, in presenting their offline HWR system in [Vin04] the authors assert that prior to their work, offline handwriting systems discussed in the research literature deal "almost without exception" with single words – an assertion that is vindicated by a review of the relevant literature [Che94, Mad01, Sen98, El99, Lu96] and by several published surveys [Ste99, Pla00, Vin02]. The related work reported in [Ber05] presents the results of applying the well-known ROVER [Fis97] combination technique to the outputs of an ensemble of HMM handwriting recognizers. Attempts have also been made to combine online and offline handwriting recognition techniques to improve performance on one task or the other. For example, in [Vin03], Vinciarelli presents an interesting approach that leverages an offline handwriting recognizer to improve the performance of an online handwriting recognizer.

In both [Vin04] and [Ber05], experimental results are reported on three different English corpora. The first is the same single-writer corpus used by Senior; the second is the IAM corpus [Mar02] which is composed of 1,539 scanned pages of text written by 657 different writers, and third is the Reuters database [Seb02] which is, again, a single-writer database of 70 documents.

Much of the work reported in [Vin04, Ber05] extends techniques originally developed for speech recognition to the new task of offline handwriting recognition and the reported experimental results are a compelling testimony to the effectiveness of the HMM paradigm for the offline HWR problem. As an aside, it is interesting to note that as late as 2005, the leading research teams in the area of offline HWR were still grappling with relatively simple corpora such as

the three listed above. Even in the IAM corpus, the rich diversity of writers is moderated by a relative lack of complexity in the handwritten content. The 1,539 pages in the IAM corpus are handwritten versions of the 500 unique pages in the well-known Lancaster-Oslo-Bergen (LOB) corpus of British English and only span approximately 11K unique words.

More recent work in HMM-based offline HWR has also focused on the application and extension of advances from speech recognition. For example, in [Dre09a], Dreuw applies the well-known speaker-adaptive training approach, originally developed by researchers at BBN [Ana97], to perform writer-adaptive training for offline Arabic HWR. Dreuw also presents the results of confidence-based discriminative training for model adaptation in offline Arabic HWR in [Dre09b]. Prasad et al have applied the consensus network based hypothesis combination approach [Man99] to the task of large vocabulary Arabic offline HWR in [Pra10a].

For their experimental assessments Dreuw, et al. used the IFN/ENIT database comprising a total of 32,492 words written by 916 different writers spanning a vocabulary of 937 unique Tunisian town names whereas Prasad et al report their experimental results using the DARPA MADCAT Arabic corpus comprised of approximately 34K documents written by 300 different writers.

Lorigo and Govindaraju [Lor06] present a recent and comprehensive survey of techniques for offline Arabic handwriting recognition in which the surveyed techniques are organized in four categories: Rules-based Approaches [Abu94, Ami03], Neural Networks (NN's) [Fah01, Har03], HMMs [Mil01, Pec03], and

hybrid techniques [Alm04, Sou04] that use straightforward combinations of one or more of the preceding techniques with the goal of overcoming the limitations of a single technique.

As is evidenced by the review above, over the past decade and especially in recent years, research in offline handwriting recognition has emphasized the use of HMMs for modeling handwritten text [Lor06]. While several researchers have applied HMMs to the task of machine-print and handwriting recognition [Kun89, Vlo92, Che94, Bun95, Kor97], the seminal work by Makhoul et al. [Mak98] in developing a script-independent methodology for optical character recognition (OCR) of machine printed text document is unique in its emphasis on the script-independence of the underlying methodology. In fact, in all the surveyed efforts that preceded [Mak98] the application of HMMs to the tasks of off-line machine printed and handwriting recognition was specific to the particular script under consideration.

In [Mak98], the authors applied an HMM system that was originally developed for speech recognition but with one significant change – the front-end feature extraction component was re-implemented to deal with images instead of speech signals. The framework in [Mak98] has been significantly advanced over the past fifteen years to incorporate major new advances such as adaptation [Nat99], feature combination [Nat01], and character duration modeling [Nat05]. In 2006, Natarajan et al. [Nat06] extended the basic HMM-based OCR system in [Mak98] to perform offline handwriting recognition in Arabic, English, and Chinese using the same single, script-independent methodology.

The extensive survey we have presented so far substantiates the fact that the work in [Nat06] is the first demonstration of the feasibility of script-independent recognition of unconstrained offline handwritten text. From a methodological perspective, the approach embodied by the HMM-based systems in [Mak98] and [Nat06] for OCR and for offline HWR, respectively, differs from other approaches principally in the emphasis that is placed on script independence. Since the publication of [Nat06], significant advances have been accomplished to the basic capability and are described in [Nat08, Cao09a, Cao09b, Cao09c, Nat09, Sal09, Pra10a, Pra10b, Nat12].

The work in this dissertation builds upon the early work in [Mak98] and upon [Nat06]. While the basic HMM paradigm offers several advantages, it is, ultimately, a one-dimensional modeling paradigm whereas offline handwriting is a truly two-dimensional pattern. Furthermore, the so-called percentile features used in both [Mak98] and [Nat06] (described in detail in Chapter 3) are computed by considering the text pixel content within uniform narrow slices – usually 2 to 3 pixels wide – of a line of text. As a result, the features themselves largely capture only 1-D information which essentially represents the vertical distribution of text pixels at a given (horizontal) position within the line of text.

From a modeling perspective, the ideal approach would be to first consider all possible 2-D segmentations of a document image, then generate a classification score for each segment, and finally determine the set of segmentations that gives the best overall classification score. Unfortunately, the ideal approach is computationally intractable because of the immense number of candidate

segmentations that need to be considered and scored. Given that the 2-D structure of handwritten text contains significant discriminative information and given the concomitant intractability of the "ideal" solution, it is desirable to find an efficient approach that would allow 2-D information structural information to be considered and used during the recognition process.

To that end, in this dissertation we describe a novel, computationally efficient *Stochastic Segment Model* (SSM) framework that allows the combination of an arbitrary number of features and classification techniques for the task of offline HWR. The proposed SSM framework directly addresses and exploits the two-dimensional nature of the information contained in handwritten text by (1) enabling the use of features that adequately capture the 2-D nature of handwriting, and, (2) providing a mechanism for incorporating an arbitrary number of classifiers that efficiently and effectively use the information in such 2-D features. We also present experimental evidence that validates our initial hypothesis and illustrates the promise of the new framework. Furthermore, we describe techniques for automatically adapting the SSM framework to each test document and present associated experimental results. But before proceeding to describe the stochastic segment modeling technique, we first present the basic HMM OCR methodology.

3 HMM-Based OCR

The proposed new stochastic segmentation model (SSM) framework builds upon the BBN Byblos system – a HMM-based OCR and offline HWR system developed at BBN. Therefore, we first review the principles and operation of the Byblos HMM system before presenting the SSM framework.

3.1 Background and Overview

Approaches to offline HWR (and, indeed, to OCR) can be classified into two broad categories: *segmentation-based* approaches [Sar02, Lor05] and *segmentation-free* approaches [Mak98, Pec03, and Nat06]. Both approaches assume the availability of pre-processing techniques that ingest a scanned document image and identify the location of text lines within the image. The location of a text line is typically specified in terms of a tight bounding box around the pixels associated with a particular text line. Segmentation-based approaches require a further pre-processing step which, ideally, divides a text line image into its constituent character glyphs. Because consistent and precise segmentation of a text line image into such character glyphs is an impossible and often ill-defined task for many real world documents, segmentation-based approaches attempt to first divide character glyphs into smaller segments. All further processing is performed on these smaller segments which are reconstituted into characters using combination rules. From a historical perspective, the vast majority of approaches to HWR [Lu96] embody segmentation-based approaches. In some cases, segmentation is done at the word-level and is combined with holistic word modeling [Dzu98, Hen01, Mad01] but limits on the amount of

training data that is available and the computational complexity associated with whole word models limit their application to small vocabulary recognition tasks.

In contrast, segmentation-free approaches such as the frame-based hidden Markov modeling approach described in this chapter do not require such pre-segmentation of the text glyphs. In HMM-based systems, segmentation is a natural by-product of the algorithms used for training and recognition. HMMs [Rab89] are capable of modeling the variability of a feature vector as a function of one independent variable. In speech [Mak95], there is one natural independent variable: time. In both offline HWR (and in OCR of machine-printed text), there are two independent variables since text images are two-dimensional (2-D); therefore, 1-D HMMs cannot be used directly. We structure the offline HWR problem as a combination of two 1-D pattern recognition tasks. The first 1-D task, also called *line finding*, is to locate the individual lines of text on a page, and the second 1-D task is to recognize the text content of each line.

Even the recognition problem at the level of a single line is in truth 2-D; however, we make it into a 1-D problem by extracting a feature vector that is a function of only one dimension (usually horizontal position within the line). The feature vector is extracted from narrow vertical strips along each line of text in exactly the same manner as described in [Nat01] for machine-printed OCR. The fact that the extracted feature vector does not depend on the script being recognized is one reason BBN's approach is script-independent. The other reason is that the HMM modeling approach itself does not change with the script being recognized. In particular, the fact that there is no separate character segmentation component,

neither in training nor in recognition, allows the same system to recognize scripts where the characters are separate or connected.

In the rest of this chapter, we describe the basic probabilistic framework of the Byblos HMM offline HWR system and briefly review the key steps in the operation of that system. To illustrate the script-independence of the Byblos system, we present offline HWR results in two languages with different scripts: English and Arabic. English handwriting presents the challenge of dealing with a cursive script (for handwritten data) and Arabic presents the further challenge of dealing with context-based variations in the shape of character glyphs.

3.2 HMM-based HWR: Probabilistic Framework

We represent a line of text within a scanned image as a sequence of feature vectors, X. The aim is to find the character sequence that maximizes $P(C|X)$, the probability of a sequence of characters C given the feature vector sequence X. Using Bayes' rule $P(C|X)$ may be written as

$$P(C|X) = P(X|C)\,P(C)/P(X) \qquad (1)$$

We call $P(X|C)$ the *feature model* and $P(C)$ the *language model* (or grammar). $P(X|C)$ is a model of the feature vector sequence X, given a sequence of characters C, and is approximated as the product of the component probabilities, $P(X_i|c_i)$, where X_i is the sequence of feature vectors that corresponds to character c_i. The feature model for each character is given by a specific HMM.

$P(C)$, the language model, is the prior probability of the sequence of characters C. The language model used in the Byblos OCR system is an *n*-gram Markov model

which computes $P(C)$ by multiplying the probabilities of consecutive groups of n characters or words. Given a sequence of characters C consisting of M characters, the *n*-gram language model is formally specified by the following equation:

$$P(C) = \prod_{i=1}^{M} P(c_i | c_{i-1}, c_{i-2}, \ldots, c_{i-n+1})$$

where the c_i's are the component characters in the character sequence C. Of course, for the first $n-1$ characters in the sequence, only the available context is considered in the *n*-gram computation. Because they are so frequently used, *n*-grams for the cases when $n = 2$ and $n = 3$ have special names – bigrams and trigrams, respectively.

Finally, $P(X)$ in (1) is the *a priori* probability of the data and does not depend on C; therefore, we can maximize $P(C|X)$ by maximizing the product $P(X|C)\ P(C)$.

3.3 Hidden Markov Models

A hidden Markov model (HMM) [Rab89] is essentially a Markov chain with one significant difference: in a Markov chain a state is associated with a unique, deterministic output value whereas in an HMM each state is associated with a probability distribution over all possible output values. We use the word "output" because Markov models are generally thought of as generative models that "produce" the observed data as output. Figure 1 shows a simple 4-state HMM with transitions and their probabilities, and the output probability distribution associated with each of the four states. These probability distributions are defined over the feature vector x which is typically a high-dimensional vector. The model

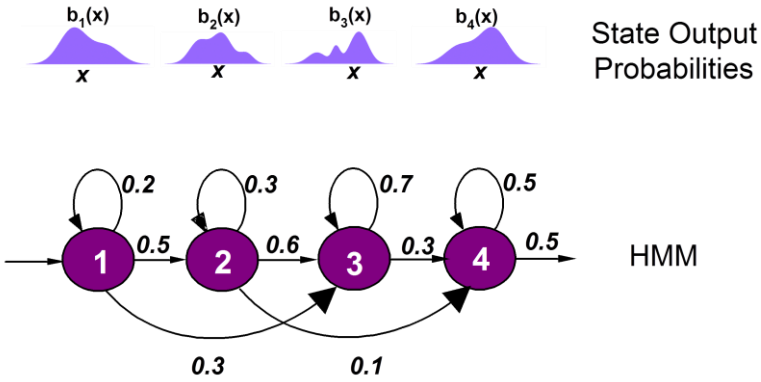


Figure 1: An example of a 4-state, left-to-right HMM of the type used in the Byblos system. It includes one-state skips and self loops.

shown in Figure 1 is known as a left-to-right model because there is a flow from left to right as one traverses the model in producing the output sequence.

Given a sequence of feature vectors extracted from a line of text, the recognition problem is to find the sequence of states or characters that "generated" the observed sequence of feature vectors. However, because of the probabilistic nature of the output that is generated by a state, almost any sequence of states could in principle, generate the observed output. Because it is not possible to uniquely map a sequence of feature vectors to a sequence of states/characters, the sequence of states that actually generated the vectors is hidden from the observer – hence the term *hidden* Markov model. Nevertheless, we can compute the *probability* that the observed sequence of feature vectors could have been generated by a particular sequence of states. Of particular interest is the sequence of states that has the highest probability of having generated the observed feature-vector sequence. By using the Markov property of the HMM, it is possible to find

that optimal state sequence very efficiently using the Viterbi algorithm [For73] or other search algorithms [Aus91, Sch96]. The resulting sequence of characters is taken as the output of the recognition component.

3.4 Byblos OCR System

The Byblos OCR system is a statistical, HMM-based recognition system that uses the Byblos HMM engine [Ngu95, Mak98] which was originally developed for speech recognition at BBN. Figure 2 shows a block diagram of the system. As indicated in the diagram, the OCR system can be sub-divided into two basic functional components: training and recognition. Both training and recognition have the same pre-processing and feature extraction stages.

3.4.1 Pre-processing

Document images typically contain non-information bearing variability (often

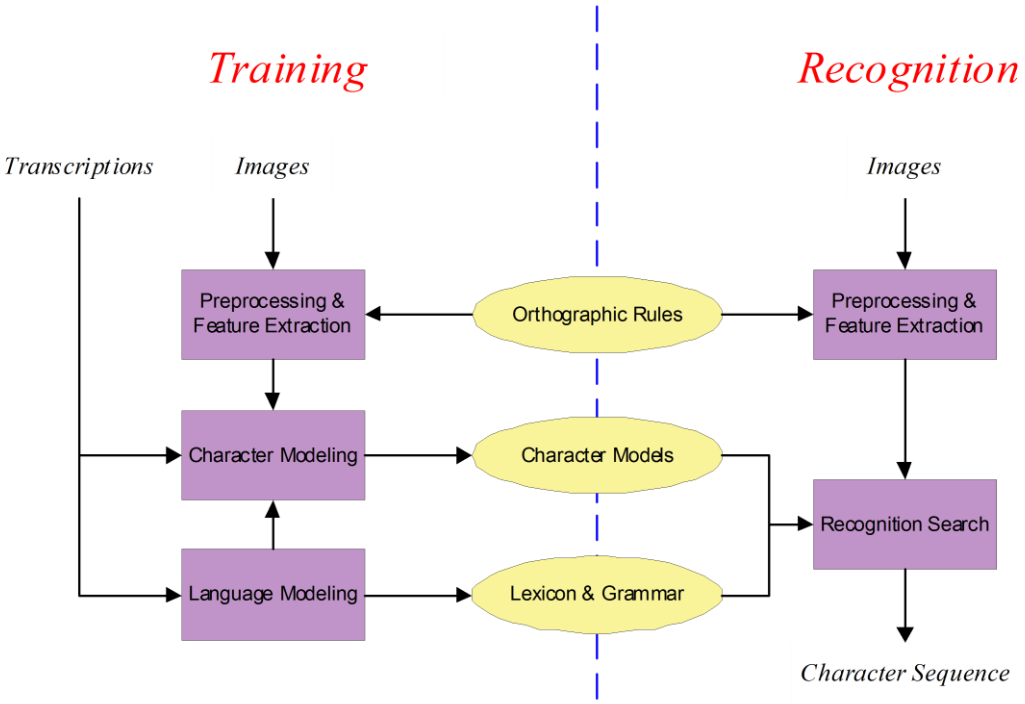


Figure 2: Block diagram of BBN Byblos OCR system.

referred to as “noise”) that is introduced by one or more physical processes such as scanning, faxing, writing style, presence or absence of ruled lines, crumpling, and folding. The goal of pre-processing is to minimize noise before the image is further processed. We note in passing that any particular type of variability could be undesirable in one context whereas that same variability might provide useful information in another. For example, variations in writing style (slant, overhanging strokes, flourishes, etc.) are undesirable from the perspective of recognizing the content of handwritten pages whereas those very variations are the target of feature extraction when the task is that of identifying the writer of a particular document.

The pre-processing stage in the Byblos OCR system is designed to minimize variations that are undesirable from the perspective of recognizing the handwritten text. To that end, we first apply a set of simple filters such as median filters that minimize artifacts such as salt-and-pepper noise and stains. Next, we deskew the image using the approach described in [Nat01]. We then apply techniques that detect and remove ruled-lines while minimizing any concomitant distortions in the shapes of the character glyphs [Cao09a, Cao09b].

The final pre-processing operation that we apply is a slant normalization technique to make the vertical strokes within character glyphs perpendicular to the baseline. To normalize the slant, we estimate and then apply a non-linear, 2-D transform to each connected component (CC) within an input (black-white) text image. The estimation of the non-linear transform is based upon the approach presented in [Tay01]. We apply the slant correction procedure iteratively until the

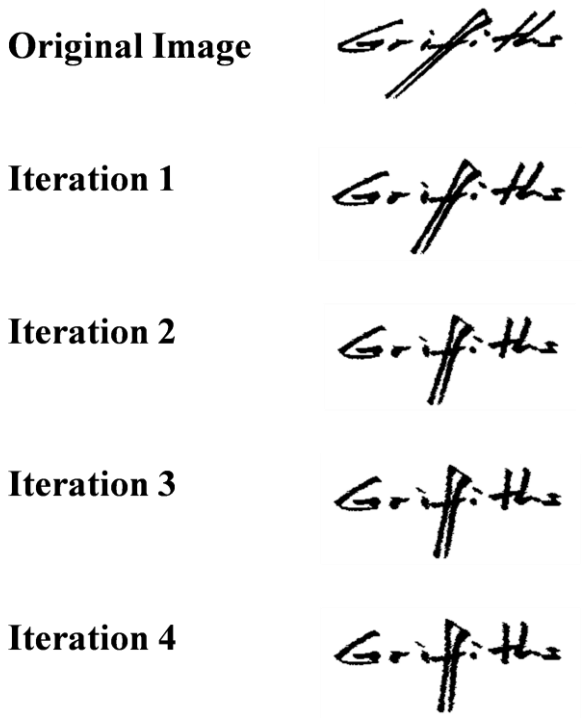


Figure 3: Example of slant normalization procedure.

estimated slant is below a certain threshold. A word from the original image and its slant-corrected versions for each of four successive iterations of the correction procedure are shown in Figure 3.

While repeated application of the transform progressively reduces the slant of text, a closer inspection of the image indicates that the transform makes the perimeter of the text more jagged. A section of the original word image and the corresponding slant corrected section after four applications of the non-linear transform are shown in Figure 4. Clearly, one area of future work is to improve the slant correction procedure to reduce the manifestation of such jagged perimeters.

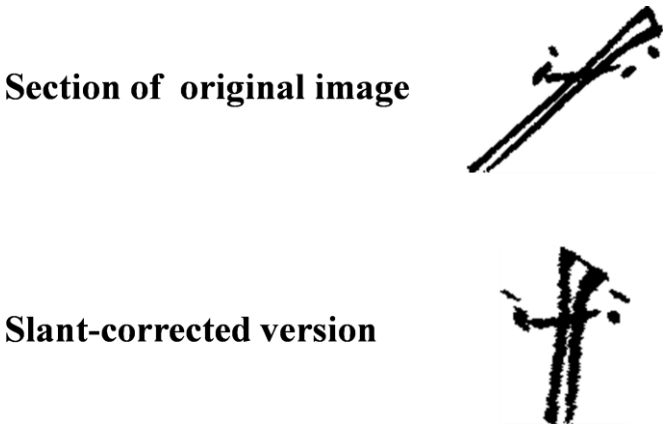


Figure 4: Jagged edges due to slant correction.

3.4.2 Line Finding

After pre-processing, the image is segmented into lines of text. For machine-printed text, the Byblos OCR system uses a HMM-based line finding technique that takes advantage of certain regularities that are characteristic of machine-printed text lines and is, therefore, not well suited to handle the irregular nature of unconstrained handwritten text. More generally, finding text lines in handwritten documents remains a challenging task that continues to attract significant research attention [Pen11, Pha11, Buk11, Saa11]. Under the DARPA MADCAT program, the research team led by BBN has developed several different line finding techniques [Man11]. Furthermore, in [Man11], we present a novel framework for combining the output of several different line finding techniques to significantly improve the accuracy of finding handwritten text lines. Both the individual techniques and the combination framework are used in the Byblos HWR system.

3.4.3 Feature Extraction

The feature extraction procedure, as applied to a sample line of English text, is graphically illustrated in Figure 5. For each line of text, the features are computed

from a sequence of overlapping analysis windows. The width of the window is set proportionately to line height (the width is typically set to 1/15 of line height) and a prescribed overlap is maintained between successive windows (the current overlap is equal to 2/3 of the window width).

For each window, also called a *frame*, several features are computed. The most

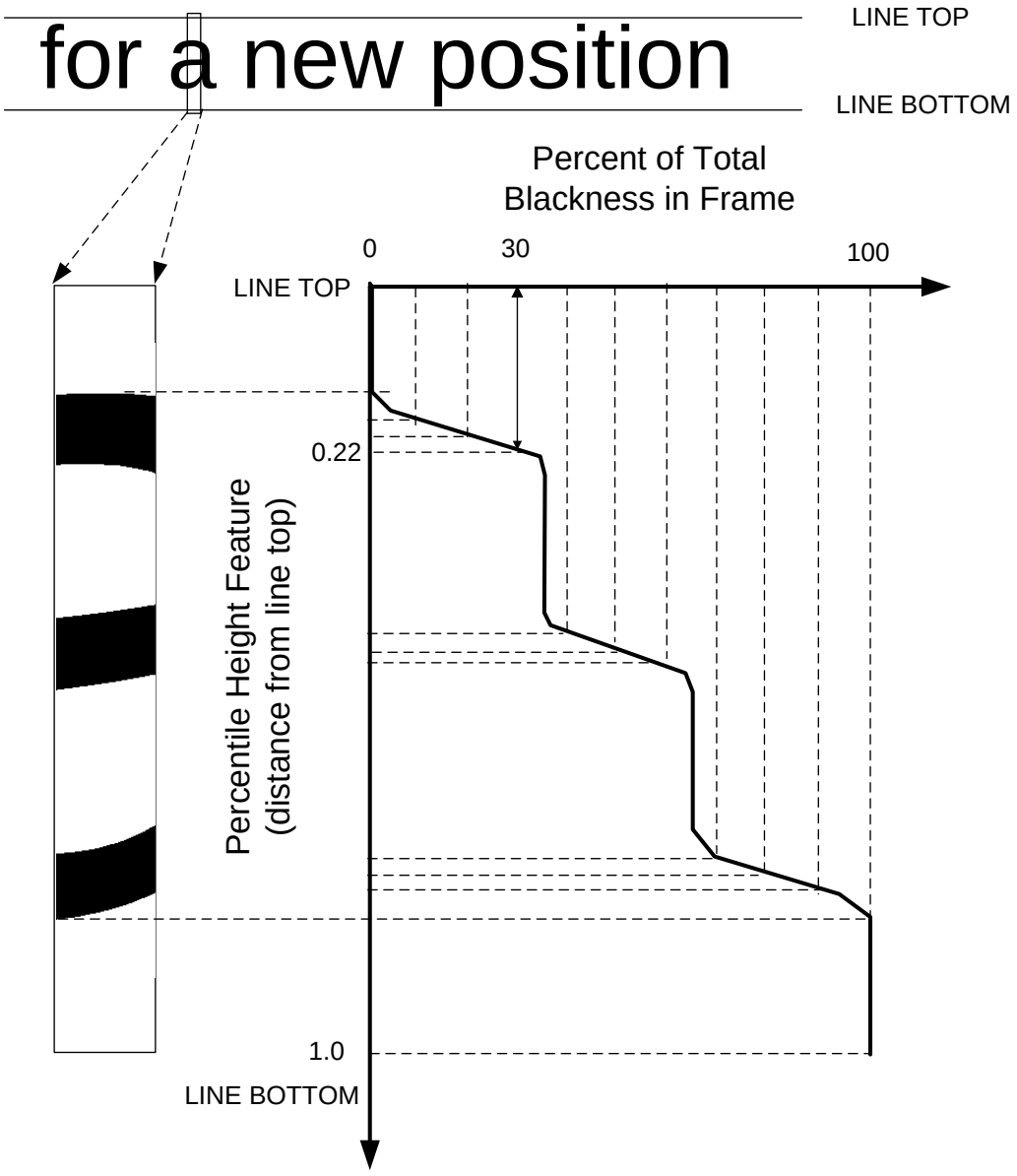


Figure 5 : Extraction of percentile features.

important ones are the *percentile* features [Nat01] computed from the binary pixels in the window. Blackness is integrated from top to bottom. After dividing by overall blackness, we get a normalized function that rises monotonically from 0 at the top of the window to 1.0 at the bottom. In the example shown in Figure 5, the feature value corresponding to the 30^{th} percentile is 0.22. This means that 30% of the blackness in the window occurs at 0.22 of the height of the window, measuring from the top and going down. The percentile features tend to be relatively insensitive to various types of noise. The percentile function is subdivided into 20 equal parts and the corresponding percentile values are taken as features.

Once the percentile features are computed, two additional sets of features are computed from the percentile values: the vertical derivatives of percentile features of a single frame and the horizontal derivatives of percentile features of adjacent frames.

In addition to the percentile features and its derivatives, a set of *angle* and *correlation* features are computed from the pixel values. The frame is divided from top to bottom into 10 (overlapping) blocks, and each block is made square by including material that falls outside the window on both sides. Within a square block, angle and correlation values are computed from a scatter plot of the text pixels, using standard linear regression to find the best linear fit that minimizes the mean square error. Basically, the angle feature measures the local slope of the text stroke within the square block while the correlation feature measures how the text pixels are distributed around the best linear fit.

Finally, for each frame, we compute a single, scalar-valued energy feature which is simply the percentage of black pixels within the frame. Collectively, the percentile, angle, correlation, and energy features are referred to as PACE features.

The feature extraction program thus computes a total of 81 features per frame: 20 percentiles, 20 vertical derivatives, 20 horizontal derivatives, 10 angle features, 10 correlation features, and the energy feature. Linear Discriminant Analysis (LDA) [Fuk90] is then used to reduce the number of features per frame from 81 to 15. The decision to use 15 LDA features was made empirically after running a set of experiments and choosing the number of features that resulted in the minimum character error rate. The resulting vector of 15 LDA features is a compact numerical representation of the data in the frame, and is the feature vector used in our recognition experiments.

3.4.4 Training

The OCR system models each character with a multi-state, left-to-right HMM; the model for a word is the concatenation of the models for the characters in the word. Each state has an associated output probability distribution over the features, as shown in Figure 2. Each output probability distribution is modeled as a weighted sum of Gaussians, or what is called a *Gaussian mixture*. A Gaussian mixture is completely parameterized by the means and variances of the component Gaussians, along with the weight of each Gaussian in the mixture. The number of states and the allowable transitions are system parameters that can

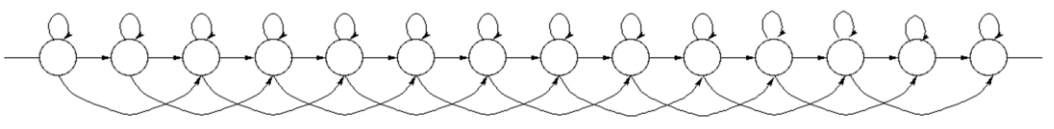


Figure 6: OCR character model is a 14-state, left-to-right HMM with self-loops and one-state skips.

be set. For our experiments we have used 14-state, left-to-right HMMs with the topology shown in Figure 6.

Training, which is the process of estimating the parameters (transition probabilities and feature probability distributions) of each of the character HMMs, is performed using what has been known alternately as the Baum-Welch [Bau72], forward-backward, or expectation-maximization (EM) algorithm [Dem77, Red84], which iteratively aligns the feature vectors with the character models to obtain maximum likelihood estimates of HMM parameters. The algorithm is guaranteed to converge to a local maximum of the likelihood function. The feature probability distributions in our system are characterized by the means, variances, and weights of the Gaussian mixtures.

We assume that, for each text-line image in the training data, we are given the corresponding ground truth, which is simply the sequence of characters on that line. No information is given about the location of each character on the line; that is, no pre-segmentation is necessary.

Depending on the amount of available training data, it may not be possible to get robust estimates of all the HMM parameters for all the characters. That is one reason why we use LDA to reduce the size of our feature vector. Another method

that is used to reduce the total number of parameters in the system is to share some of the Gaussians across different character models. The motivation for such sharing is made clear by the following example. Consider the two English letters *c* and *o*. Clearly, the first halves of the two characters exhibit the same curved shape and therefore the features corresponding to the first halves would also exhibit similar distributions of feature values.

Gaussian-sharing in our Byblos system can be performed through one of three related configurations: Tied Mixture (TM) configuration [REFs], Character Tied Mixture (CTM) configuration [Bel89, Hua89, Lu99], and State Tied Mixture (STM) configuration. In the TM configuration a single set of Gaussians (referred to as a codebook of Gaussians or just a codebook) is shared between all states of all character models. The individuality of each state output probability distribution is characterized solely by the specific component mixture weights. The TM configuration represents the most extreme mode of sharing and is appropriate when the amount of training data is not adequate to support robust estimation of models with larger numbers of free parameters.

In the CTM configuration we train one codebook of Gaussians for each character model; the Gaussians in a character codebook are thus shared among the states of the model for that character but there is no sharing across characters. The CTM configuration offers a greater number of free parameters than the TM configuration and with it the possibility of better performance, subject to the availability of sufficient data for training all the parameters.

Our discussion thus far has implicitly assumed that a single glyph shape is associated with each character and that that shape is modeled by a single HMM. But in the case of cursive handwriting in general and especially in the case of natively cursive scripts such as Arabic, the shape of the glyph associated with a particular character can often change as a function of the context in which it appears, where context is typically defined by the characters that precede and/or follow the character under consideration. In such cases, it is often desirable to use a separate HMM to model the glyph shape associated with each salient context – an approach that is referred to in HMM literature as "context-dependent" modeling. The STM configuration applies only to context-dependent models and it imposes a shared set of Gaussians for each numbered state of all the context-dependent HMMs associated with a particular character.

3.4.5 Recognition

After pre-processing a line of text and performing feature extraction, as described above, the recognition process consists in a search for the sequence of character models that has the highest probability of having generated the observed sequence of feature vectors, given the trained character models, a possible word lexicon, and a statistical language model (typically an *n*-gram language model) of the possible character or word sequences. The recognition search is a two-pass [Aus91, Sch96] (a forward pass and a backward pass) beam search for the most likely sequence of characters. The width of the search beam is a system parameter that can be set. Typically, lowering the beam width increases the speed but degrades the accuracy of recognition. The forward pass is an approximate but

efficient procedure for generating a small list of character sequences that are possible candidates for being the most likely sequence. The Byblos HMM system uses an approximate bigram language model, i.e. an *n*-gram language model, where $n = 2$, in the forward pass.

The backward pass is a more detailed search for the most likely character sequence within this small list. Even though we typically use the same set of character HMMs in the forward and backward passes, the search program itself does not impose any such constraint, i.e., if needed we can use two different sets of character HMMs, one in the forward and the other in the backward pass. When required, larger, more computationally complex HMMs are used in the backward pass for greater discriminative power. The backward pass can use either a bigram or a trigram ($n = 3$) language model but typically a trigram language model is employed.

The use of a word lexicon (or, vocabulary) during recognition is optional; its use generally results in a lower error rate when the out-of-vocabulary (OOV) rate is low or close to zero. The term out-of-vocabulary refers to words that are present in the test set but are not included in the lexicon. The lexicon itself is estimated from a suitably large text corpus and the language model, which provides the probability of any character or word sequence, is also estimated from the same corpus. Note that the text corpus used for estimating the language model (and, indeed, the lexicon) need not be limited to the manual transcriptions associated with the images in the training set because only the sequence of words in the text is needed to estimate the language model probabilities. The only caveat is that the

transcriptions associated with the images in the test set not be used for estimating the lexicon or language model.

In the next chapter we present the results of some experiments that exercise the Byblos offline HWR system on English and Arabic handwritten text data sets.

4 Frame-based HMM OCR: Experimental Results

In this section we present, in separate sub-sections, the data sets, the metric, the experimental procedure, and results for English and Arabic. While only the Arabic results are of direct relevance to the work in this dissertation, we have included experimental results on English in order to demonstrate the script-independence of the Byblos system.

4.1 Data

4.1.1 English Handwritten Text Corpus

For our English offline HWR experiments, we used the IAM database [Mar02]. As described earlier in Chapter 2, the IAM English database consists of unconstrained handwritten English sentences from the LOB corpus [Joh78]. The database was collected by distributing forms with printed text to writers, and having them write the text on the forms in their own handwriting. A total of 1539

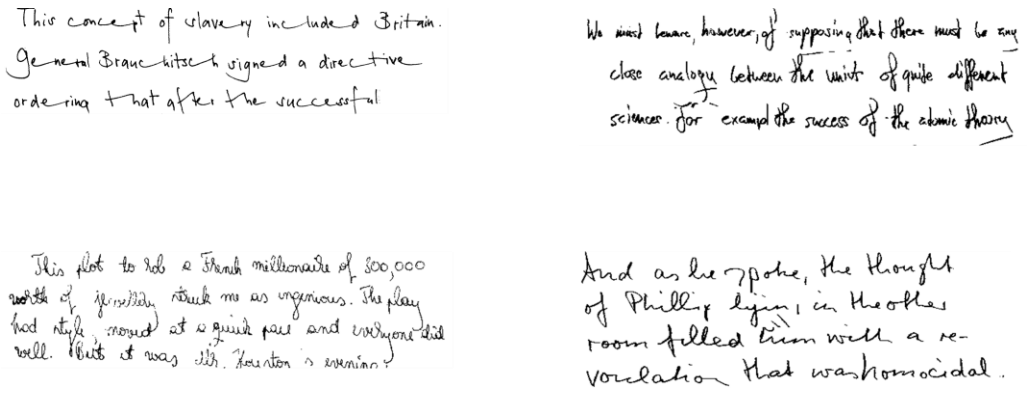


Figure 7: Sample images from the IAM database.

images from 657 different writers, scanned at a resolution of 300 dpi were used in our experiments. For our experiments, we split the corpus into three sets: a training set, a development set, and a held-out test set. In dividing up the corpus

Characteristic	Train	Development	Test
Number of Images	1239	150	150
Number of Lines	10726	1316	1311
Total Number of Words	95512	11632	11308
Unique Number of Words	10217	3135	3093
Number of Writers	506	108	97

Table 1: Characteristics of the IAM training, development, and test sets.

into the three sets, we ensured that no writer appears both in test and in training (i.e., a writer-independent test condition). Further, in order to ensure that the language models are fair, all handwritten instantiations of a form are assigned entirely to a single subset (training, dev, or test). In other words, each passage of text is either in training or in (dev) test but never in both. Figure 7 contains samples from the IAM database. The samples illustrate the rich diversity in writing styles contained within the IAM database.

Table 1 lists the characteristics of the training, dev, and test sets that we used in our experiments. The IAM database includes annotations of the bounding boxes for each word. For the experiments reported here, we used the word bounding box information to determine the top and bottom of the lines of text within each image.

4.1.2 Arabic Handwritten Corpus

For our Arabic offline HWR experiments, we used the IFN/ENIT corpus [Pec02]. This corpus was collected by distributing forms with pre-selected text of Tunisian

city names and postal codes to multiple writers and having them write the text on the forms in their own handwriting. A total of 26,459 images consisting of 937 unique city names from 411different writers are available for training and test. These images have been distributed into 4 sets by the creators of the corpus: set *a*,

Characteristic	**Train**	**Development**	**Test**
Dataset	*a*, *b*, and, *c*	*a*, *b*, and, *c*	*d*
Number of Images	17983	1741	6735
Number of Unique City Names	937	492	850
Number of Writers	277	30	104

Table 2: Characteristics of training, development, and test sets selected from IFN/ENIT corpus.

set *b*, set *c*, and, set *d*. For our experiments, we used sets *a*, *b*, and, *c* for training/development and set *d* was used for test purposes. This split is exactly the same as reported in the ICDAR 2005 Arabic handwriting recognition competition [Mar02]. As shown in Table 2, we held-out 1741 images from 30 writers in the training set for additional test/development purposes. Figure 8 contains samples from the IFN/ENIT database. These samples illustrate the broad diversity in writing styles contained within the database.

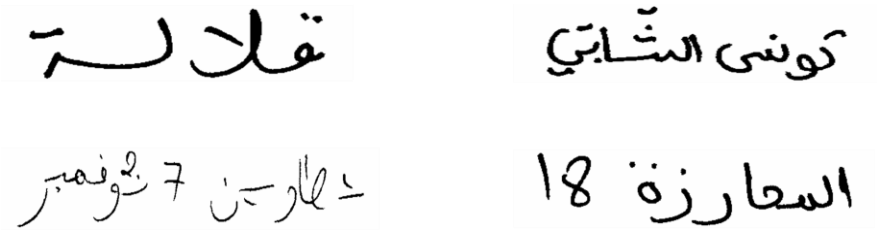


Figure 8: Four sample text images from the IFN/ENIT corpus.

Despite its obvious limitations as a research corpus, its use in the ICDAR competitions has made the IFN/ENIT corpus a *de facto* platform for comparison of recognition accuracy in Arabic. Therefore, we decided to use the IFN/ENIT database for our initial Arabic handwriting recognition experiments in order to assess the performance of our HMM-based system relative to other published techniques.

4.2 Metric (Word Error Rate)

System performance is measured using the word error rate (WER) or the character error rate (CER), defined as:

$$WER\ (CER) = \frac{deletions + insertions + substitutions}{total\ number\ of\ words\ (characters) in\ reference}$$

where the reference is the manually generated ground truth transcriptions.

4.3 Experiment configuration and system performance

4.3.1 English HWR

Glyph HMM Configuration: Each HMM character model comprises 14 states as shown in Figure 4. The English HWR configuration includes 77 distinct character HMMs and we used a CTM configuration with a separate codebook of 256 Gaussians for each character HMM in the training set. We experimented

with context-dependent as well as context-independent models. For any given character, context is defined by the identities of the characters to its left (preceding) and right (following). For example, in the word "cat", character "a" is said to be in the context of preceding "c" and following "t," whereas in the word "halibut", the same character "a" is in the context of preceding "h" and following "l." In our system, a total of 9,814 contexts were considered.

Language Model Configuration: We experimented with both character and word tri-gram language models (LM) which were trained on transcriptions from the IAM training data set. For our experiments using word-based LMs, we used a 10K-word vocabulary derived from the IAM training transcriptions alone. The out-of-vocabulary (OOV) rate on the test set for the 10K IAM vocabulary was 10.01%.

We ran five training and test experiments using data from the IAM English database. The results of the five experiments are tabulated in Table 3 below, with each row of the table summarizing the configuration and performance (in terms of

Glyph HMM Configuration	**Language Model Configuration**	**Word Error Rate**
Context-independent, CTM	Character tri-gram	68.5%
Context-independent, CTM	Word tri-gram	52.8%
Context-dependent, CTM	Word tri-gram	49.3%
Context-dependent, STM	Word tri-gram	46.1%
Context-dependent, STM, Slant Correction	Word tri-gram	40.1%

Table 3: Summary of English offline HWR results on the IAM database.

WER) of the models used in each of the five experiments.

As indicated in the first row of Table 3, in the very first experiment we trained a set of context-independent character HMMs using the IAM handwritten training data set. The models were tested against the IAM handwritten test data and yielded a WER of 68.5%, with an associated CER of 40.1%. The only other previous work using this database in combination with an HMM-based approach incorporating statistical language models is [Vin04] where the authors report a WER of 57%. While it is not possible to directly compare our results with that in [Vin04] because of differences in the training and test sets, the fact that our very first experiment yielded a WER that was within a reasonable margin of the best previous work was an encouraging indication of the promise of our approach.

In HMM-based OCR systems, one can use a character-based LM or a word-based LM. The specific choice of one or the other is usually a function of the OOV rate – when the OOV rate is high relative to the WER, a character LM is preferred; otherwise a word LM is preferred. Here, the OOV rate of 10.01% is clearly much smaller than the WER. Therefore, in our second experiment we replaced the character tri-gram LM with a word tri-gram LM and observed a satisfying decrease in WER to 52.80%.

In handwritten texts, the shape of a particular character glyph typically varies based on its context. Therefore, in our third experiment we trained *context-dependent* character HMMs. In the context-dependent configuration a separate HMM is used to model each contextually distinct instance of a character. Using

context-dependent HMMs and a word LM, we obtained a WER of 49.30%. Once again, our intuition was empirically validated by the result of the experiment.

In our fourth experiment, we used the STM configuration with of 128 Gaussians per mixture. The STM configuration enables better modeling of the structural evolution of a character glyph in the direction of writing. With the STM model, we obtained a WER of 46.1%. Cumulatively, the word LM, context-dependent HMMs, and the STM configuration lowered the WER from 68.5% to 46.1% – a substantial net reduction of 32.8% relative.

We ran a final fifth experiment in which we again used the STM configuration but with slant corrected versions of the training and test images instead of the raw versions from the IAM database as was done in the previous experiments. The application of slant correction further lowered the WER to 40.1%, which represents a relative reduction of 41.5% over the initial WER of 68.5%

As mentioned earlier, the only other reported prior work on this database using a HMM-based approach with statistical language models is [Vin04] where the authors report a WER of 57%. It is not possible to directly compare our results with that in [Vin04] because of unknown differences in the training and test sets.

4.3.2 Arabic HWR

We ran several Arabic HWR experiments using data from the IFN/ENIT corpus. The results of the experiments are listed in Table 4, with each row showing the configuration and performance (in terms of WER) of the models used in each of the experiments. Recall that the IFN/ENIT corpus consists of handwritten Tunisian city names or postal codes. Each image in the IFN/ENIT corpus

Glyph HMM Configuration	**Language Model**	**SER – % on Set *d***
Context-independent, CTM	Character	58.9
Context-independent, CTM	Compound word	12.9
Context-dependent, CTM	Compound word	11.8
Context-dependent, CTM, Slant correction	Compound word	11.2
Context-dependent, STM, Slant correction	Compound word	11.0
Context-dependent, STM, Slant correction, unsupervised writer adaptation	Compound word	10.6

Table 4: Summary of Arabic Recognition Results on the IFN/ENIT corpus.

comprises a single line of text which in turn contains one or more words and, occasionally, number strings. Therefore, in assessing performance on this corpus, we consider whether the entire line (or, text string) has been recognized correctly by measuring the so-called string error rate, or SER. For each test sample, the string error rate is either zero or one.

The first recognition experiment used 14-state context-independent character HMMs and a character tri-gram LM, both trained on the IFN/ENIT corpus. For training the character HMMs, we used a CTM configuration with a separate codebook of 256 Gaussians for each character in the training set. The error rate on the city names in set *d* was 58.9% with this configuration.

In our second experiment we replaced the character tri-gram LM with a compound-word LM by stringing the words constituting a city name into a single, distinct token. With the compound word LM, we obtained an error rate of 12.9%,

more than a factor of 4 better than using a character tri-gram LM! While the improvement is impressive, it is important to keep in mind that the IFN/ENIT corpus has a very limited vocabulary of less than a 1000 city names; a fact that makes a word LM extremely powerful in eliminating unwanted character sequences.

Next, we performed a third experiment in which we trained context-dependent character HMMs. Using context-dependent HMMs and the compound word LM, we obtained an error rate of 11.8%, a 1.1% absolute improvement over using context-independent character HMMs. The improvement in performance with context-dependent HMMs over context-independent HMMs is significant (nearly 9% relative) but smaller than the improvement obtained on English data under the same modeling conditions. We believe this is due to the fact that for Arabic both our OCR and HWR systems use the contextual forms of characters to train character HMMs. The contextual form in Arabic encodes canonical changes in shape of a character glyph due to the neighboring characters. Consequently, the primary context-dependent variability in the case of Arabic characters is that due to the natural differences in styles across different writers. Therefore, it is not surprising that context-dependent HMMs yield a smaller improvement for Arabic than for English.

In our fourth Arabic HWR experiment, we applied slant correction to the training data and re-trained the context-dependent character HMMs. Testing on the slant corrected test images using context-dependent HMMs and compound word LM resulted in an error rate of 11.2% on city names in the development set. Once

again, slant correction yielded a lower relative reduction in error rate for Arabic than for English. A possible cause for this relatively lower improvement in error rate is that, in comparison to English, a larger fraction of the strokes that compose Arabic glyph shapes are horizontal or near-horizontal.

We then trained context-dependent character HMMs with a STM configuration. We used a separate set of 128 Gaussians to model the output distribution at each state for all contexts of a particular character. The error rate on city names in the development set dropped to 11.0%.

In our sixth and final experiment, we used the recognition results from the previous experiment to perform unsupervised adaptation of the character HMMs for each writer. Adaptation [Nat99] is the process of adjusting the parameters of an initial trained model so as to improve performance on a particular document or a subset of documents. HMM adaptation techniques have been extensively studied in the context of speech recognition and can be broadly divided into two categories: *supervised* adaptation and *unsupervised* adaptation. In supervised adaptation manual transcriptions for the adaptation data are provided whereas in unsupervised adaptation we first use the trained model to recognize the document and then use the recognized text as the transcriptions for the adaptation data. Using the adapted model we can then recognize the document again, typically with higher accuracy. Here, we used the Maximum Likelihood Linear Regression (MLLR) [Leg95] adaptation technique. A maximum of 8 transformations was applied to adapt only the means of the Gaussians associated with each character

HMM. The adapted models resulted in an error rate of 10.6%, an absolute improvement of 0.4%.

While the goal of our experiments was to demonstrate the script-independence of our HMM-based offline HWR methodology, we note in passing that the performance of the Byblos HMM system on set *d* is better than the best reported result in the ICDAR 2005 Arabic handwriting competition [Mar02]. Having established the script-independent and state-of-the-art performance of our HMM-based offline HWR system, we now proceed to present the stochastic segment modeling approach.

5 Stochastic Segment Modeling

Recognition approaches that require accurate segmentation of the text into smaller canonical units do not usually perform well on handwritten text or even on machine-printed text that has been subjected to certain types of real-world degradations. There are two primary causes for poor performance of segmentation-based approaches. First, segmenting handwritten text, especially for scripts such as Arabic, is very difficult. Second, most real-world images (handwritten or machine printed) are prone to degradations that result in breaks and merges in glyphs – a phenomenon that generates an essentially infinite number of new, previously unseen connected components and thereby makes the segmentation task even more challenging, if not impossible.

As discussed in Chapter 2, offline HWR systems based on hidden Markov models have been shown to outperform segmentation-based approaches. An important benefit of HMM based systems is that they are segmentation-free; i.e., no pre-segmentation of word/line images into smaller units such as sub-words or characters is required. However, there are known limitations with HMM based approaches [Ost96]. These limitations are due to two reasons: (a) the assumption of conditional independence of the observations given the state sequence, and (b) the restrictions on feature extraction imposed by the need for local frame-based observations.

While [Ost96] is focused on the speech recognition task, the limitations noted therein are directly relevant to offline HWR systems which use pixel-level features from narrow slices of the text image. In particular, the narrow windows

provide very limited contextual information thereby weakening the conditional independence assumption in these systems.

Inspired by the work in [Ost96], we present a novel offline HWR framework for combining structural matching and HMM-based recognition, which we will demonstrate has more discriminative power than a straightforward combination of the structural (i.e., long-span) and percentile (i.e., short-span) features at each frame. The key steps in the new framework can be summarized as follows: first, an HMM-based system trained on the percentile features is used to demarcate candidate 2-D character images which we call *stochastic segments*. Second, structural features are extracted from these automatically discovered stochastic segments, and a separate classifier that operates on these structural features is applied to generate a new set of scores for each of the candidate stochastic segments. Third, the scores from the percentile-feature based HMM system and from the classifier operating on the broader stochastic segment-based features (also referred to as structural features) are combined to generate the best hypothesis.

5.1 Stochastic Segment Modeling – Theoretical Framework

In this section, we formulate the framework for stochastic segment modeling. We begin with the baseline HMM system, which is described as follows. Given a line image I we extract a feature vector sequence X using fixed size frames that correspond to successive positions of a sliding analysis window that moves across each text-line image along the direction of writing. These frames are typically narrow and capture short span features. As described earlier in Section 3.2, the

goal of the recognition task is to find a sequence of characters C that maximizes $P(C|X)$, the probability of a sequence of characters C given the feature vector sequence X. Using Bayes' rule $P(C|X)$ may be written as:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

In the equation above $P(X|C)$ is the character glyph model which can be modeled by a hidden Markov model and $P(C)$ is the language model. As is usual with the HMM approach, $P(X|C)$ is approximated by the product of component probabilities $P(X_i|c_i)$ and $P(C)$ is computed using the standard *n*-gram model.

We now introduce the stochastic segment modeling framework. Let S be a segmentation (i.e., a set of character image segments) of a text-line image I into a character sequence C. Using Bayesian inference, we can rewrite $P(C|I)$ as follows:

$$\begin{aligned} P(C|I) &= \sum_S P(C,S|I) \\ &= \sum_S P(C,S|X) \\ &= \sum_S P(C|S,X)P(S|X) \end{aligned}$$

We assert that $P(C|S,X) = P(C|S)$; i.e., once the set of character image segments S is available to the classifier, X does not provide any more information. By applying this assertion, we can rewrite $P(C|I)$ as,

$$\sum_S P(C|S)P(S|X)$$

Next, we define $P(S|X) = P(L, C|X)$. In other words the segmentation S is now fully characterized by two parts, the sequence of characters C and their horizontal extents (i.e, lengths) L. By applying this definition, we can rewrite $P(C|I)$ as,

$$\sum_S P(C|S)P(L, C|X)$$

Through simple Bayesian inference, the equation above can be expanded as,

$$\sum_S P(C|S)P(C|X)P(L|C, X)$$

Finally, we assert that once the sequence of characters C is available X provides no additional information about the associated lengths of the characters. In probabilistic terms, this implies $P(L|C, X) = P(L|C)$. By applying this assertion, we can finally rewrite $P(C|I)$ as,

$$\sum_S P(C|S)P(C|X)P(L|C) - (2)$$

The last step of the inference above is the probabilistic formulation of the Stochastic Segment Model (SSM) framework. For convenience of discussion, we identify this last step as Equation (2). We now consider each of the distinct probabilistic elements contained therein.

$P(C|S)$, the first element in Equation (2), is the probability of a sequence of characters C given the particular segmentation S and is approximated by the geometric mean of the component character probabilities $P(c_i|S_i)$, where c_i is a character within the sequence C and S_i is the image segment associated with c_i.

$P(c_i|S_i)$ may be computed using any of a variety of classifiers, e.g., SVM or NN, that produce scores that are correlated with posterior probabilities. The SVM, NN, or other classifier used to compute $P(c_i|S_i)$ is referred to as the segment classifier and the classifier training process is described in Section 5.2. Of particular significance is the fact that the component character probabilities (and, therefore, $P(C|S)$) can be computed by combining evidence from multiple sets of segmental/structural 2-D features extracted from the image segments and even from multiple classifiers, each operating over some combination of the available 2-D features.

The second element of Equation (2), $P(C|X)$, is the probability of the character class label C given the sequence of underlying feature vectors X. As indicated earlier, $P(C|X)$ is simply the posterior from an HMM which includes the character glyph score *and* the grammar/language model score.

The third and last element of Equation (2), $P(L|C)$, is a probabilistic model of the horizontal extents (i.e., lengths) of the characters in the sequence and it can be estimated separately from training data as a distribution of observed character lengths, appropriately normalized for comparison across different fonts and writing styles.

The width of each segment in S spans a portion of the text-line image that corresponds to several (anywhere from 10 to 30) short-span feature vectors X. As indicated earlier, $P(C|S)$ is computed by employing models that operate on long-span structural features that are computed from entire character image segments.

Therefore, the introduction of $P(C|S)$ enables a tight, interactive coupling between the long- and short-span features. In other words, the stochastic segment modeling framework integrates long-term context with local context within a single probabilistic formulation. In order for Equation (1) to be computationally tractable, we only consider a limited set of possible segmentations S that are small perturbations around the *N*-best list of maximum-likelihood segmentations (and associated character or word hypothesis sequences) produced by the baseline HMM system.

5.2 Stochastic Segment Modeling – Engineering Details

In terms of engineering implementation, the stochastic segment modeling framework involves the following key steps:

Stochastic Segment Generation: In order to train the stochastic segment classifier, we need to create a training corpus of annotated stochastic segments. Annotation here simply means assigning a character label to each segment. Developing an adequately sized annotated corpus through manual means is prohibitively expensive. Therefore, we propose an automatic means for generating such a corpus. First, a traditional frame-based HMM system is trained and then applied to the training itself to generate a set of recognition hypotheses for the text-line images in the training data. The HMM recognizer automatically produces either an *N*-best list or lattice of candidate character sequences corresponding to each input text-line image. For each candidate character sequence, the HMM automatically provides an associated segmentation of the input text-line image. Using the frame-to-character alignment generated by the frame-based HMM, we

map the frames to pixel locations in the underlying image and extract the *stochastic image segments* (2-D character images) along with the associated character labels. Figure 9 shows an example of the stochastic segmentation produced by an HMM engine.

Segment Classifier Training: For each extracted segment, we can compute several so-called structural (i.e., long-span) features that represent shape characteristics of the character. Such features capture a much broader context than the narrow frame-based features. For training the segment classifier, we use the structural features computed from the automatically generated segments and associated

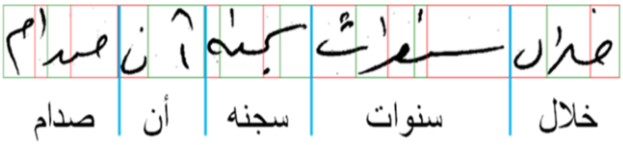


Figure 9: Example showing the one-best stochastic segmentation of characters in a line image. The segmentation was generated by the Byblos HMM engine trained with short-span PACE features.

labels from the first step. The classifier can be any one or a combination of well-understood techniques such as support vector machine (SVM) classifiers or Neural Network (NN) classifiers.

Recognition: Similar to training, recognition is also a two-step process. First, we use the narrow frame-based HMM to produce several candidate character sequences (in *N*-best or lattice form) along with the associated automatic segmentations for an input text line image. Next, for each segment we apply the trained segmental classifier(s) to compute a score(s) for the associated character label. We then combine the scores from the HMM and SVM for each hypothesis

and use the combined score to re-rank the candidate character sequences corresponding to the input text line image. The score combination itself is implemented as yet another classifier which can be trained/optimized using a held-out portion of the training corpus.

In the rest of this Chapter we describe experimental procedure for exercising and assessing the performance of the SSM framework.

5.3 Stochastic Segment Framework – Experiment Design

As described earlier, for each text line image the Byblos HMM-based offline HWR system produces several candidate character sequences and associated segmentations as a byproduct of the training and recognition processes. Therefore, extraction of stochastic segments simply involves using the pixel boundaries of the narrow HMM frames to extract the candidate 2-D character images. In Figure 9, we have shown an example segmentation of a line into constituent characters as determined by an HMM system.

5.3.1 Segment Modeling using SVMs

The stochastic segments can be modeled by a variety of different classifiers including support vector machines (SVMs), neural networks (NNs), or even Bayesian classifiers. In our experimental system, we decided to use SVMs for modeling the stochastic segments in combination with the Gradient-Structure-Concavity (GSC) features that have previously been shown to be effective for offline handwriting recognition [Sal09, Fav96].

GSC features are symbolic, multi-resolution features that capture structural information at three levels of granularity. The finest granularity is provided by

the gradient features which represent the local orientation of strokes. Gradient features are computed by simply convolving two Sobel operators on the binary image. The two operators approximate the derivatives in the *x* and *y* directions,

Structural features capture information coarser level of granularity and are designed to extract and represent stroke pattern information such as corners (perpendicular co-occurrence of strokes), up/down gradients, etc. Structural features are computed from the gradient features rather than from the raw binary image itself.

The concavity features are actually a collection of three sub-classes of features that represent stroke relationships at long distances. All three of the features – G, S, and C – are computed using simple linear operators on the raw binary image or on the gradient features (image).

We use the character labels and the GSC features extracted from stochastic segments to train a multi-class SVM with the radial basis function (RBF) Kernel. The multi-class SVM is trained such that, given a GSC feature set extracted from a stochastic segment, it outputs a vector of probabilities, each of which is the probability of the given feature set belonging to a particular character.

5.3.2 *N*-best Rescoring with SSM Framework

A schematic diagram of the *N*-best rescoring procedure used to implement our proposed stochastic segment modeling framework is shown in Figure 10. As shown in the figure, given a text-line image (produced by the line finding step), we first process it using the baseline HMM system to generate a set of *N*-best

hypotheses and character segmentation for each hypothesis in the *N*-best list. For each hypothesis, say C_n in the *N*-best list, we compute the HMM likelihood score,

$$P(X|C_n)$$

and the language-model score

$$P(C_n)$$

Here C_n stands for the n^{th} hypothesis in the list of *N*-best hypotheses for the input text-line image. We use *n*-gram language models, typically with the value of *n* set

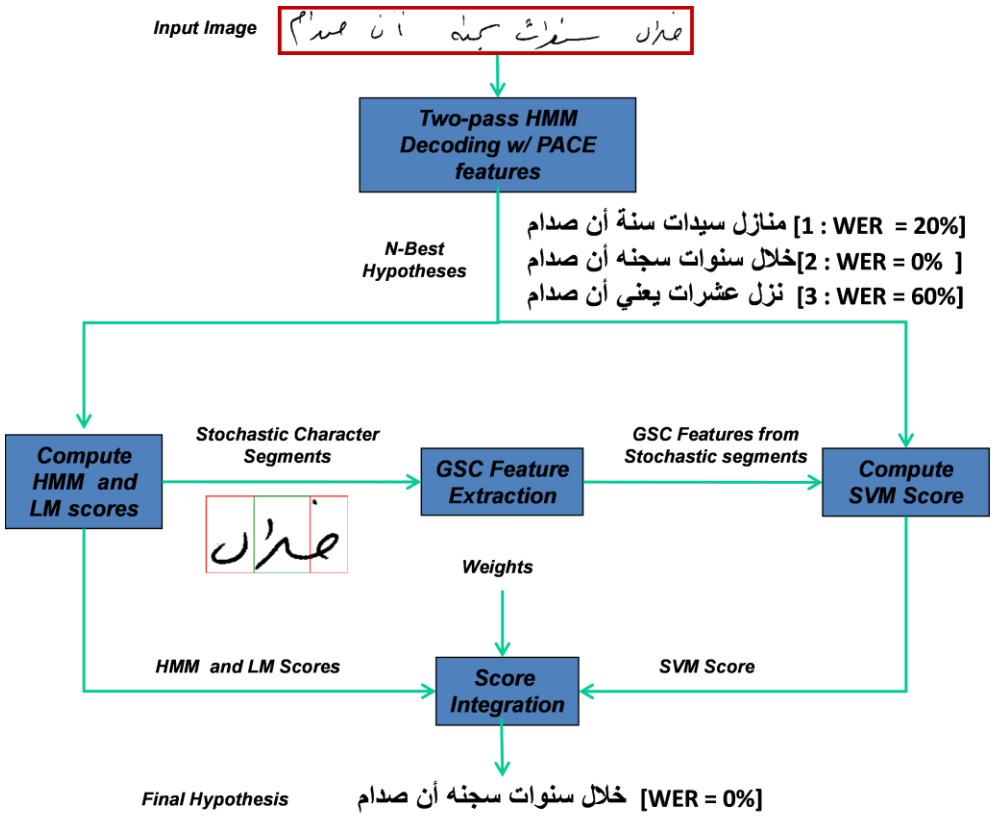


Figure 10: Illustration of the rescoring procedure in which the SVM scores are combined with the HMM (glyph) and LM scores. The example text line image, the N-best hypotheses, the final hypothesis, and associated WERs are all taken from an actual SSM experiment.

to 3 and the language-model score, therefore, is just the *n*-gram probability score for the hypothesis. When combined together, either through direct multiplication of the two scores, or as is more common, through the weighted addition of their logarithmic forms, the likelihood score and the language model score are correlated to the posterior probability for the character given the short-span feature vector sequence X.

Using the character segmentation information from the HMM, we extract the set of 2-D character images corresponding to each of the *N*-best hypotheses. We then compute a sequence of long-span or structural (in our case, GSC) feature vectors that correspond to the sequence of 2-D character images in each hypothesis. Next, for each character within each hypothesis in the *N*-best list, the segment classifier (here, SVM) is used to compute a score

$$P(c_{n,i}|S_f)$$

that is correlated with the posterior probability of the character $c_{n,i}$; where $c_{n,i}$ refers to the i^{th} character in the n^{th} hypothesis within the *N*-best list. The composite "segment-based" score

$$P(C_n|S_f)$$

for the n^{th} hypothesis in the *N*-best list is computed as the geometric mean of the $P(c_{n,i}|S_f)$ scores for all characters in the hypothesis.

Finally, we generate a single score H_n that integrates the three different scores (HMM likelihood, language model, and segment/structural) using the following formula:

$$H_n = \propto_1 \times \log[\, P(X|C_n)] + \propto_2 \times \log[\, P(C_n)] + \propto_3 \times \log[P(C_n|S_f)]$$

The terms $\propto_1$, $\propto_2$, and $\propto_3$ in the equation are the integration weights shown above the "Score Integration" box in Figure 10. Clearly, if more than one segment classifier is used, then the number of terms in the equation above would increase to accommodate the scores from the additional classifiers. Finally, the *N*-best hypotheses are re-ranked in order of decreasing $H_n$and the top ranking hypothesis is selected as the single-best SSM recognition result for the input text line image. In the next chapter, we present experimental results that demonstrate the reduction in error rate when the single-best HMM recognition result is compared with the single-best SSM recognition result.

6 Stochastic Segment Modeling – Experimental Results

In this section, we describe two sets experiments that were performed using the stochastic segment modeling framework described in the previous section. In the first set of experiments, the Byblos HMM system was trained using just percentile features and the segmental SVM classifier was trained using GSC features. This first set of experiments was designed to evaluate: (a) the classification accuracy of SVMs using GSC features, (b) the quality of the character segmentation from the HMM, and (c) establish the viability of rescoring the *N*-best list by using the segmental SVM scores in combination with the scores generated from the HMM.

The second experiment was designed to assess the ability of the SSM framework to: (a) deliver performance improvement even when the basic HMM system is trained using all available features (i.e., the percentile *and* GSC features) and (b) provide improvements in large data regimes where the underlying HMM system itself operates at relatively higher levels of accuracy.

6.1 Data Sets

We performed our Arabic handwriting recognition experiments using data from two different Arabic corpora. One corpus, called the AMA corpus, is a set of 5000 Arabic handwritten documents comprises handwritten copies of 200 distinct Arabic passages, each written by 25 native writers of Arabic. The AMA collection contains a scanned TIFF (Tagged Image File format) image of each of the 5000 pages, an associated XML file which contains writer and page metadata, the pixel coordinates of the bounding box around each word in the document, and a set of offsets representing Parts-of-Arabic-Words (PAWs). The AMA corpus

Set	Number of Images	Number of Writers
Training	848	10
Dev	125	10
Test	48	4

Table 5: LDC Arabic mini-corpus used for rescoring experiments.

contains a total of 930K total words which are spanned by a vocabulary of 9400 unique words. We used the manually annotated PAW units from the AMA corpus to assess the classification accuracy of the SVM in combination with GSC features.

The second corpus is the so-called LDC Arabic Handwriting corpus, a large corpus of approximately 30K Arabic handwritten pages scanned at a resolution of 300 dpi. The pages contain a total of 2.8M words which are spanned by a lexicon of 85K unique words.

For our initial set of experiments, in addition to the AMA corpus, we used a small subset of the LDC Arabic corpus which we will call the LDC Arabic mini-corpus. The LDC mini-corpus consists of 1250 pages written by 14 different authors. The text content of the 1250 pages is taken from various newswire articles, weblog posts, and newsgroup posts. The collection of 1250 pages

Set	**Number of Images**	**Number of Writers**
Training	28K	182
Dev	880	26
Test	880	26

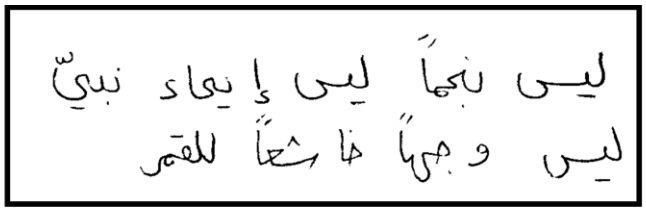
Table 6: LDC Arabic corpus – Training, Dev, and Test split

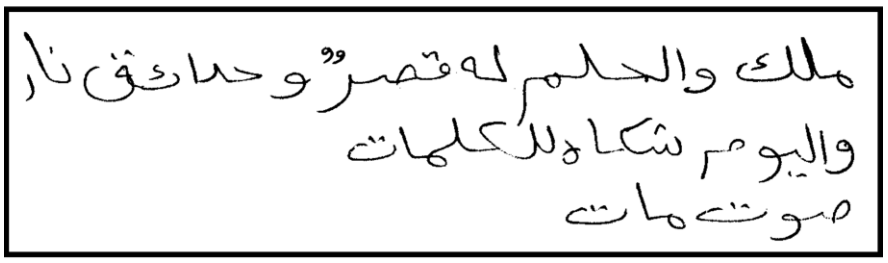
contains a total of 83K words and is spanned by a vocabulary of 7.6K words. In order to ensure a fair test set with no writer or document content in training, 229 images were held-out of the training and development sets. 125 images from this set were randomly chosen as the development set. A total of 48 images by 4 different authors constitute the test set. The details of the LDC mini-corpus are shown in the Table 6.

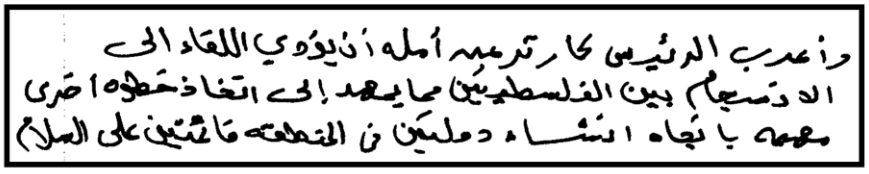
For our second set of experiments, we used the entire LDC Arabic corpus by dividing it into training, development (or, dev), and test sets. A total of 28K pages written by 182-Different scribes were used for training and a set of 1760 pages from 52 scribes was split equally into an 880-page development set and an 880 page test set. Only the development set is used for optimizing the model and recognition search parameters; no optimization is performed on the test set. Of the 26 writers each in the dev and test sets, 13 writers are present in the training set and 13 writers are not present in the training set.

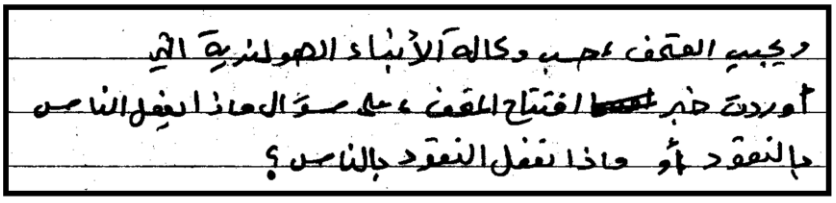
In the next two pages, we show several text line images taken from the AMA and the LDC corpora. The images are illustrative of the tremendous diversity of handwriting in both corpora. In addition to the diversity of the handwriting itself, the LDC Arabic corpus contains a wide variety of textual content taken from available newswire and weblog text data sets.

Example Images from the AMA Corpus









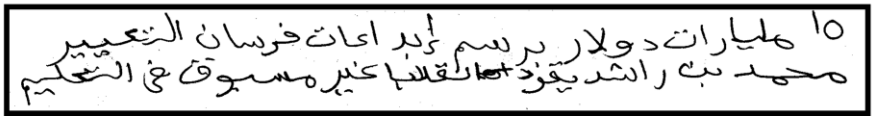


Figure 11: Sample text image lines taken from the AMA corpus. While the textual content is restricted to 200 selected passages, the images in the corpus span a variety of different writing styles and writing instruments.

Example Images from the LDC Corpus

Figure 12: Sample text image lines taken from the LDC corpus which contains a wide variety of writing styles and writing instruments. The textual content is also diverse and is taken from newswire articles and weblogs.

6.2 Initial SSM Experiments

The goals of the first set of experiments are to:

1. Measure the classification accuracy of an SVM classifier that uses GSC features to model a set of PAWs.
2. Assess the quality of the candidate stochastic segments from the HMM.
3. Establish the viability of the SSM approach by rescoring the *N*-best list using the segmental SVM scores in combination with the scores generated from the HMM.

6.2.1 Classification with GSC Features and SVM Classifiers

In order to measure both the efficacy of the GSC features and of the SVM classifier, we instrumented a classification experiment using manually annotated PAWs from the AMA data set. We used radial basis function (RBF) SVMs and trained them using GSC features extracted from each entire PAW image and the PAW labels from the training set. The PAW images and labels were randomly chosen from the AMA corpus. A total of 6498 training samples from 34 PAW classes were used to train the SVM classifier. The test set consists of 848 PAW

Types of Units	Number of Classes	Error Rate - %
PAWs	34	17.2
Stochastic Segments	40	25.3

Table 7: PAW and segment classification accuracy using an SVM classifier and GSC features.

images from the same set of 34 PAW classes. From the vector of probability scores produced by the SVM for each class label, we chose the class label with the highest probability as the classification label for the PAW image. With that setup, we observed an error rate of 17.2%.

6.2.2 Generating Stochastic Segments using HMMs

Having established the utility of the GSC-SVM classification paradigm, we proceeded to investigate the consistency of segments that are stochastically generated by the Byblos HMM engine. Therefore, in our second experiment, we first trained the Byblos HMM engine using the PACE features and used it to extract stochastic segments from the word images in the AMA dataset. We then used the extracted stochastic character segments in place of the PAWs and computed GSC features from each character segment. A total of 13,261 automatically segmented character training samples from 40 character classes were used for training a RBF-kernel SVM. The trained SVM was then tested against a test set of 3315 test samples and we observed an error rate of 25.3%.

Some caution is appropriate in directly comparing the classification error rate on the manually generated PAWs with the classification error rate on the automatically generated stochastic segments because the training and test sets are different. As shown in Table 11, the results on the PAW classification task were better than the results for the stochastic segment classification task. Two possible factors come to mind as possible causes of the higher error rate on the stochastic segments.

One factor which could have resulted in the poorer performance for the character segment classification is that, since the character boundaries are stochastically determined, they may not have been consistent; the associated noise introduced in the features would prevent sharp, discriminative feature distributions. To explore the consistency of the segmentation, we visually reviewed a substantial number of automatically extracted character segments and found them to be reasonably accurate and consistent. In Figure 3, we show multiple instances of the same character appearing in different words written by different authors and see that the character segmentation is pretty consistent across these multiple instances. The review of segmentation performance indicates that inconsistency in stochastic segmentation is unlikely to be a major contributor to higher error rate associated with stochastic segments.

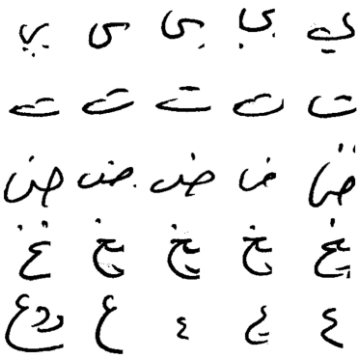


Figure 13: Each row shows different "stochastic segment" instances of a single character. Each instance was automatically segmented within the word in which it occurs. Despite the variability in glyph shapes across the different instances, the accuracy and consistency of segmentation appears to be high.

The second possible factor is that, on average, the PAWs encode substantially longer structural context, and are therefore more easily discriminated than

character segments. In fact, longer context in other language classification tasks such as speech recognition and machine translation is known to provide more discriminative information (and, thereby, greater recognition or translation accuracy) than local or short-term context. Therefore, in the current case, it seems highly likely that the longer structural context is the primary enabler of the lower classification error rate in the case of PAW classification.

6.2.3 Validating the SSM Framework through *N*-best Rescoring using SVM Scores

The third and final experiment in our first set of experiments was designed to validate that the recognition word error rate can be lowered using the SSM framework. We used the LDC mini-corpus for this experiment. First, we trained the Byblos HMM recognizer using the data in the mini-corpus. Using the trained recognizer, we generated stochastic segments for all the images in the training data set. We then extracted GSC features for each stochastic segment and trained SVM classifiers for each segment-class (character class). During recognition rescoring, we compute the SVM classification scores for each stochastic segment in the *N*-best hypothesis list generated by the Byblos HMM recognizer. The amount of data used for training, development, and validation is shown in Table

Scores Used for Rescoring	**WER (%)**
Glyph (HMM) + Language Model	55.1
Glyph + Language Model + SSM	52.8

Table 8: Experimental results on the LDC mini-corpus using scores from the SVM segment models to re-rank the *N*-best hypotheses. An absolute improvement of 2.3% in the WER is observed.

9. The SVM was trained using 900 randomly chosen stochastic segment images from the training set for each character class. The results for this experiment, along with those for the Byblos HMM system are shown in Table 15.

From the numbers in the two rows of Table 15, we see that the addition of the SVM segment scores for rescoring improves overall system performance by 2.3% absolute (4.1% relative). This experiment is the first known application of stochastic segmentation for any handwriting recognition task including the task of recognizing unconstrained, cursive offline handwritten text.

6.3 Large Data Training Experiments

The *N*-best rescoring experiment summarized in Table 15 clearly demonstrates the effectiveness of the SSM framework. Nevertheless, it is important to note that the HMM system in this initial experiment was trained using just the PACE features whereas the HMM + SSM system uses PACE features (for the HMM) and new GSC features for the SSM. While the SSM framework it intended to enable the integration of new kinds of features (especially, two-dimensional, structural features) into the overall recognition process, it is still important to understand whether the framework itself, by allowing the incorporation of additional classifiers, can improve performance even when all available features are used to train the baseline HMM system itself.

Furthermore, previous experience in speech recognition indicates that many new techniques provide improvements over a direct HMM approach when limited training data is available but that those improvements evaporate when much larger amounts of training data are made available. Furthermore, the gains from

new techniques often disappear when large recognition vocabularies and massive amounts of text data are used to train the *n*-gram language models used in HMM systems such as the BBN Byblos system.

Therefore, notwithstanding the promising initial results from the application of the stochastic segmentation approach to the recognition of offline handwriting, two important questions remain to be answered. The first question is:

Question 1: Does the decrease in word error rate come from the use of the GSC features alone or does the SSM framework also contribute to the lowering of the error rate?

Question 2: Does SSM-enabled improvement in recognition performance endure even when much greater amounts of training data are available to train the Byblos HMM system?

We decided to investigate the answers to those questions jointly. To that end, we first trained the Byblos HMM system on the 28K training images in the LDC Arabic handwriting corpus. Consistent with our standard practice, each text line image was analyzed using a sliding vertical window and 33 script-independent PACE features were extracted from the pixel content underlying each location of the analysis window (also called a frame). We also extracted 96-dimension GSC features. Linear Discriminant Analysis (LDA) [Nat01, Sal09] was then applied to reduce the dimension of the feature space from 129 to 15.

Consistent with our practice thus far, each individual character glyph was modeled by a 14-state, left-to-right, context-dependent HMMs with state-tied

Gaussian mixture distributions. The HMM parameters were trained using the standard maximum likelihood (ML) estimation technique [Rab89]. In total 1.5 million Gaussians were trained for 175 unique characters that included Arabic characters, numerals, punctuations and English characters. English characters were included in the model set because Arabic handwritten documents often contain a sprinkling of English words and numerals.

The Byblos HMM system uses *n*-gram language models. Bigram and trigram language models (LM) were also estimated from 90 million words of Arabic newswire and web data. The decoding lexicon consisted of 92K of the most frequent words in this Arabic text corpus.

Recognition was performed in three passes. The first pass uses the character HMMs and a bigram word language model to perform a fast beam search. Next, the backward pass uses a trigram language model to create an *N*-best list. Usually, the backward pass model has more parameters than the forward pass model. However, in these experiments we used the same HMM in the forward and backward passes. In the third and final recognition step, the *N*-best list from the

System Configuration	**WER (%)**
HMM system: Un-adapted HMM	31.3
SSM system: un-adapted HMM & SVM	30.7

Table 9: Performance of the HMM and SSM system without adaptation on the LDC Arabic data set.

backward decoding is rescored using a combination of scores from various knowledge sources including the HMM and the LM.

As shown in the first row of Table 9, the basic HMM system (without adaptation) results in a word error rate (WER) of 31.3% on the test set. For our SSM experiments, we trained segmental models on the training data by using the character boundaries produced by the HMM system. Specifically, a 175-class SVM was trained on the stochastic character segments in the training corpus. We used GSC features to train the SVM. Next, the *N*-best list produced by the HMM system was rescored with segmental scores obtained from the segmental SVMs. As shown in the second row of Table 16, we obtain a 0.6% absolute reduction in WER over the baseline HMM system for using the segmental scores.

Note that the gain with rescoring with the segmental model is lower than that reported in [Nat09]; a difference that is due to the fact that the HMM recognizer used for the experiments reported here was trained on PACE+GSC features, whereas in [Nat09] we used only the short-span PACE features for HMM training. Our goal here is to establish the value of the SSM framework itself by eliminating any benefit due to the additional use of GSC features and the current experiment reported was designed in light of that goal. The 0.6% absolute reduction in WER observed here can be attributed solely to the SSM classifier rather than due to the combination of new features and the SSM classifier as in [Nat09]. Therefore, the answer to the first question posed at the beginning of Section 6.3 is a positive one –the SSM classifier combination framework in and of itself contributes to the lowering of the WER.

In a satisfying response to the two questions posed at the beginning of this section, we have demonstrated that the SSM approach by itself helps to lower the WER and that the improvement due its application persists even in the face of massive amounts of training data. Nevertheless, there is one final question that we need to answer before we can safely assert that the SSM approach provides an enduring reduction in WER:

Does the reduction in WER achieved through the application of the SSM framework endure even after the system is adapted to the test data?

In the next section, we analyze the results of a sequence of experiments that were designed to answer this final question.

6.4 Adaptation Experiments

6.4.1 HMM Adaptation

In the speech recognition community it is standard research practice to report final WER numbers after the initial trained models have been subjected to unsupervised adaptation on the test data itself. This practice of reporting final numbers after unsupervised adaptation is rooted in years of empirical evidence showing that unsupervised adaptation often obliterates the gains from other sources such as new features or pre-processing. In the case of the SSM framework, any improvements obtained by adapting the SVM classifier itself are desirable because those improvements would further bolster the case for the usefulness of the SSM framework. By the same token, the SSM framework is only interesting if the overall performance improvement endures any adaptation-related improvements in the performance of the basic HMM system itself.

Therefore, we first applied unsupervised page-wise adaptation to the Byblos HMM system using the well-known maximum likelihood linear regression (MLLR) adaptation technique [Leg95]. Recall that in unsupervised adaptation, the recognition hypotheses from the initial trained models are used in place of manually generated transcriptions in order to *adaptively* re-estimate the parameters of the trained models. In our case, we used the single-best recognition hypotheses from the SSM recognition framework in place of manual transcriptions.

The results of our adaptation experiments are summarized in Table 10. To facilitate easy comparison, the first and third rows of the table simply list the performance of the trained HMM and SSM systems without adaptation. As can be seen from those two rows, the WER of the HMM system is 31.3% and that of the SSM system is 30.7% – an improvement of 0.6% absolute, or 1.9% relative.

In designing an unsupervised adaptation approach, it is important to keep in mind that available adaptation data is typically much smaller in amount than the training data that was used to train the models in the first place. Furthermore, the adaptation data contains errors, and when the error rates are high, a significant fraction of the adaptation data is incorrect. Therefore we need to use regularization techniques [Li06] that restrict the degree or extent to which model parameters are changed during the adaptation process.

In our case, we allowed a maximum of 128 regression classes for the MLLR adaptation step and we re-ran the Byblos HMM recognizer using the adapted HMMs. As shown in row 2 of Table 10, the application of page-wise HMM adaptation results in a WER of 30.4%, which is 0.9% absolute lower than the WER observed without adaptation.

Next, we applied the SVM segment classifiers to the stochastic segments generated by the adapted HMMs. It is important to note that we used the original SVMs trained using stochastic segments that were generated using the HMM system without adaptation. In other words, the SVM classifier parameters were not adapted to the test data. With that configuration, as shown in the fourth row of Table 10, we observed a WER of 29.9% – an absolute reduction of 0.5% from the WER of the adapted HMM-only system and an associated reduction of 0.8% from the 30.7% WER obtained using the SSM framework with un-adapted HMMs.

The fact that even with HMM-adaptation the SSM framework retains most of its

System Configuration	WER (%)
HMM system: Un-adapted HMM	31.3
+ *page-wise HMM adaptation*	30.4
SSM system: un-adapted HMM & SVM	30.7
+ *page-wise HMM adaptation*	29.9
+ *page-style SVM adaptation*	29.6

Table 10: Performance of HMM and SSM systems with and without adaptation on the LDC Arabic corpus.

improvement, by itself, establishes the value of the SSM framework. In the following sub-section we consider one final avenue for further improvement in the error rate – that of efficiently adapting the parameters of the SVM segment classifier itself.

6.4.2 Segment Model (SVM) Adaptation

For our final adaptation condition, in addition to adapting the HMM system, we adapted the SVM parameters using techniques described in [Li06]. Our first SVM adaptation experiment produced a negative result, i.e., directly adapting the SVM model parameters to each page resulted in worse performance than with the initial trained models. The most likely cause for the failure of direct SVM adaptation is that the richness of SVM parameters makes SVM parameter estimation more sensitive to errors in the adaptation transcripts. The nature of discriminative models is such that the small amounts of available adaptation data result in significant over-fitting of the SVM model parameters to the errors in the adaptation transcripts.

Therefore, we investigated an adaptation approach which mitigates the two suspected problems – small amount of adaptation data and the associated high error rate – by transforming the adaptation problem into an adaptive model selection problem. We first automatically clustered the training set images into K clusters using a simple set of "style" features [Cao9c]. The style features include: (a) height and width of the stochastic segments, (b) thickness of strokes, (c) density of text on a page, and (d) contour-dependent features. The goal of the clustering step is to organize the training images according to the style of

handwriting contained in each image. Next we trained a global segment model using automatically generated segments from the entire set of training images. Subsequently, K *style-adapted* models are estimated by adapting the parameters of the global segment model using the images associated with each of the style-clusters. Finally, we train a separate classifier that is capable of classifying a new image to one of the K style clusters.

During recognition, we extract the same set of style features from each test image and use the trained classifier to find the style cluster that contains images that are, on average, most similar to the test image. Next, the test image is processed using the adapted HMM recognizer and a *N*-best hypothesis list is generated along with the stochastic segments associated with each hypothesis. Finally, the style-adapted SVM associated with the most similar style cluster is selected and the SSM framework is exercised using the style-adapted segmental SVM classifier. Satisfyingly, as can be seen from the final row of Table 17, the style-adaptive approach produces a further 0.3% reduction in WER, thereby validating our hypothesis and bringing the total improvement in WER due to the SSM framework to 0.8%, from 30.4% to 29.6%.

7 Conclusions and Future Work

In this dissertation we have described a new stochastic segment modeling framework for integrating long-range 2-D information with short-span approximately 1-D features such as the PACE features. While the SSM framework is the focus of this dissertation, the underlying script-independent methodology for offline HWR [Nat06] is, in itself, a major new milestone in the area of handwriting recognition.

The SSM framework incorporates an existing HMM system; it does not require manually segmented training data nor does it require an explicit rule-based pre-segmentation step. All needed segmentations are automatically inferred. Our implementation of the SSM framework is a computationally efficient *N*-best rescoring approach. The experimental results presented in this dissertation demonstrate that the use of stochastic segment modeling gives a 2.6% relative gain over the baseline HMM system, *even after the underlying HMM system was adapted to the test image*. The fact that the SSM framework delivers gains that endure the adaptation of the underlying HMM system points to their continued relevance in the future.

As part of our work, we explored techniques for adapting the segment classifier/models themselves. Direct adaptation of the segment classifier resulted in unsatisfactory performance but we proposed and validated a technique for adaptively selecting appropriate segment models for each test image. The adaptive selection technique was shown to be successful in lowering the WER.

It is difficult to compare the work reported here directly with the work reported by other researchers because the vast majority of previously published research on offline handwriting recognition has focused on smaller data sets with several simplifying attributes. For example, the now standard IFN/ENIT database presents a small vocabulary research task where most of the images contain one or two words.

In 2010, NIST conducted, OpenHART (for Open Handwritten Arabic Recognition Test) the first-ever worldwide competitive evaluation of offline Arabic HWR using the same test set that the BBN Byblos system has been tested against during each year of the DARPA MADCAT program. The results of the OpenHART evaluation clearly demonstrate that the recognition accuracy of the Byblos HMM system is, by far, the best of all available offline HWR systems. Therefore, it is safe to claim that any improvement in the performance of the Byblos HMM system represents an advance in the state-of-the-art in offline handwriting recognition.

In addition to advancing the state-of-the-art in offline handwriting recognition and providing useful immediate improvements to recognition accuracy, the SSM framework also offers a fertile basis for future research efforts. We now highlight three possible directions for future research using the SSM framework as a basis.

First, the adaptation experiments outlined in this dissertation were limited to adaptation of the parameters of the HMM component within the SSM framework. Past experience with related statistical pattern recognition approaches suggests that automatic adaptation of the parameters of the segment classifier itself to each

test page will yield meaningful improvements in accuracy. Furthermore, such adaptation need not be restricted to just the page level and can, in fact, be performed on sets of documents that have been automatically aggregated based upon some underlying similarity in their content (e.g., writing style).

Second, we have used *N*-best rescoring to prove the usability and effectiveness of the stochastic segmentation framework. Clearly, the use of lattices will allow for more comprehensive consideration of a broader set of alternative segmentations and hypotheses which has the potential to produce even greater improvements in recognition accuracy.

Finally, the stochastic segmentation approach can be extended to other recognition tasks such as writer identification and writer verification. For both writer ID and writer verification, discriminative structural features computed from the stochastic segments derived automatically from training data can be used to train appropriate classifiers. At runtime, the same stochastic segmentation and feature extraction procedure can be used in combination with the trained classifiers to perform writer identification or verification. By allowing a consideration of writer-specific details at the glyph-segment level, a stochastic segment based approach could lead to significant improvements in identification and verification accuracy.

8 Bibliography

[Abu94] Abuhaiba I S I, Mahmoud S A, and Green R J, "Recognition of Handwritten Cursive Arabic Characters," *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, pp. 664-672, 1994.*

[Alm04] Alma'adeed S, Higgens C, and Elliman D, "Off-Line Recognition of Handwritten Arabic Words Using Multiple Hidden Markov Models," *Knowledge-Based Systems, vol. 17, pp. 75-79, 2004.*

[Ami03] Amin A, "Recognition of Hand-Printed Characters Based on Structural Description and Inductive Logic Programming," *Pattern Recognition Letters, vol. 24, pp. 3187-3196, 2003.*

[Ana97] Anastasakos T, McDonough J, and Makhoul J, "Speaker Adaptive Training: A Maximum Likelihood Approach to Speaker Normalization," *IEEE International Conference on Acoustics, Speech, and Signal Processing, Munich, Germany, 1997.*

[Aus91] Austin S, Schwartz R, and Placeway P, "The Forward-Backward Search Algorithm," *IEEE International Conference on Acoustics, Speech, Signal Processing, Toronto, Canada, Vol. V, 697-700, 1991.*

[Bau72] Baum L E, "An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes," *Inequalities, Vol. 3, 1-8, 1972.*

[Bel89] Bellegarda J and Nahamoo D, "Tied Mixture Continuous Parameter Models for Large Vocabulary Isolated Speech Recognition," *IEEE International Conference on Acoustics, Speech, Signal Processing, Glasgow, Scotland, Vol. 1, 13-16, 1989.*

[Ber05] Bertolami R and Bunke H, "Ensemble Methods for Handwritten Text Line Recognition Systems," *International Conference on Systems, Man, and Cybernetics, pp. 2334-2339, 2005.*

[Buk11] Bukhari S S, Shafait F, Breuel T M, "Text-Line Extraction Using a Convolution of Isotropic Gaussian Filter with a Set of Line Filters," *International Conference on Document Analysis and Recognition, Beijing, China, 2011.*

[Bun95] Bunke H, Roth M, and Schukat-Talamazzini E G, "Off-line Cursive Handwriting Recognition using Hidden Markov Models," *Pattern Recognition, Vol. 28, pp. 1399-1413, 1995.*

[Bur93] Burges C.J.C., Ben J.I., Denken J.S., Lecun Y, and Nohl C.R., "Off-line Recognition of Handwritten Postal Words using Neural Networks," *International Journal for Pattern Recognition and Artificial Intelligence, 1993; 7(4):689–704.*

[Cao09a] Cao H, Prasad R, and Natarajan, P, "A Stroke Regeneration Method for Cleaning Rule-lines in Handwritten Document Images," *Proceedings of the International Workshop on Multilingual OCR (MOCR), Barcelona, Spain, 2009*.

[Cao09b] Cao H, Prasad R, Natarajan, P, and Govindaraju, V, "Nested State Indexing in Pairwise Markov networks for Fast Handwritten Document Image Rule-line Removal," *Proceedings of the 16th IEEE International Conference on Image Processing (ICIP), 2009.*

[Cao09c] Cao H, Prasad R, Saleem S, and Natarajan P, "Unsupervised HMM Adaptation using Page Style Clustering," *International Conference on Document Analysis and Recognition, Barcelona, Spain, 2009*.

[Che94] Chen M Y, Kundu A, Zhou J, "Off-line Handwritten Word Recognition using a Hidden Markov Model Type Stochastic Network," *IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(5):481–496, 1994.*

[Dem77] Dempster A P, Laird N M, and Rubin D B, "Maximum-likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society Ser. B (methodological), Vol. 39, 1-38, 1977*.

[Dzu98] Dzuba G, Filatov A, Gershuny D, and Kill I, "Handwritten Word Recognition – the Approach Proved by Practice," *Proc. 6th International Workshop on Frontiers in Handwriting Recognition, pp. 99–111, Taejon, Korea, 1998.*

[Ede90] Edelman S, Ullman S, and Flash T, "Reading Cursive Handwriting by Alignment of Letter Prototypes," *International Journal of Computer Vision, Vol. 5:3, pp. 303-331, 1990.*

[ElY99] El-Yacoubi A, Gilloux M, Sabourin R, and Suen C Y, "Unconstrained Handwritten Word Recognition using Hidden Markov Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(8):752–760, 1999.*

[Fah01] Fahmy M M M and Al Ali S, "Automatic Recognition of Handwritten Arabic Characters Using Their Geometrical Features," *Studies in Informatics and Control J., vol. 10, 2001*.

[Fav] Favata J T and Srikantan G, "A Multiple Feature/Resolution Approach to Handprinted Digit and Character Recognition," *International Journal of Imaging Systems Technology, 7:304–311, 1996.*

[Fav96] Favata J T and Srikantan G, "A Multiple Feature/Resolution Approach to Handprinted Digit and Character Recognition," *International Journal of Imaging Systems Technology, 7:304–311, 1996.*

[Fis97] Fiscus J, "A Post-processing System to yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)," *IEEE Workshop on*

Automatic Speech Recognition and Understanding (ASRU), pp. 347-354, Santa Barbara, USA, 1997.

[For73] Forney G D, "The Viterbi Algorithm," *Proc. IEEE, Vol. 61, 268-278, 1973.*

[Fuk90] Fukunaga K, "Introduction to Statistical Pattern Recognition," *Academic Press, New York, Chapter 10, Second Edition, 1990.*

[Har03] Haraty R and Ghaddar C, "Neuro-Classification for Handwritten Arabic Text," *Proceedings of ACS/IEEE International Conference on Computer Systems and Applications, 2003*.

[Hen01] Hennig A and Sherkat N, "Cursive Script Recognition using Wildcards and Multiple Experts," *Pattern Analysis and Applications, 4(1):51–60, 2001.*

[Heu97] Heutte L, Pereira P, Bougeois O, Moreau J, Plessis B, and Courtellemont P, "Multi-bank Check Recognition System: Consideration on the Numeral Amount Recognition Module," *International Journal of Pattern Recognition and Artificial Intelligence, World Scientific, pp. 595–617, 1997.*

[Hua89] Huang X D and Jack M A, "Semi-continuous Hidden Markov Models for Speech Recognition," *Computer Speech and Language, Vol. 3, 1989*.

[Joh78] Johansson S, Leech G N, and Goodluck H, "Manual of Information to Accompany the Lancaster-Oslo/Bergen Corpus of British English, for Use with Digital Computers," *Department of English, University of Oslo, Norway, 1978*.

[Kim88] Kimura M, Ejima T, Aso H, Yashiro H, Son N, and Suzuki M, "An Intelligent Character Recognition System with High Accuracy and High Speed by Integrating Image-type and Logical-type Information Processing," *IAPR International Conference on Pattern Recognition (ICPR), Rome, Italy, 1998.*

[Kim97] Kim G and Govindaraju V, "Bankcheck Recognition Using Cross Validation between Legal and Courtesy Amounts," *International Journal for Pattern Recognition and Artificial Intelligence, pp. 657–673, 1997.*

[Kim98] Kim G and Govindaraju V, "Handwritten Phrase Recognition as Applied to Street Name Images," *Pattern Recognition, vol. 31, no. 1, pp. 41-51, Jan 1998.*

[Kor96] Kornai A, Mohiuddin KM, and Connell SD, "Recognition of Cursive Writing on Personal Checks," *Proc 5th International Workshop on Frontiers in Handwriting Recognition, Essex, UK, 1996; 373–378.*

[Kor97] Kornai A, "An Experimental HMM-based Postal OCR System," *Proc International Conference on Acoustics, Speech and Signal Processing, Munich, Germany, 1997; 3177–3180.*

[Kun89] Kundu A and Bahl P, "Recognition of Handwritten Script: A Hidden Markov Model-based Approach," *Pattern Recognition, Vol. 22, pp. 283-297, 1989.*

[Leg95] Legetter C J and Woodland P C, "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models," *Computer Speech and Language, pp. 171-185, vol. 9, 1995.*

[Li06] Li X and Bilmes J, "Regularized Adaptation of Discriminative Classifiers," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Toulouse, France, 2006*.

[Lor05] Lorigo L and Govindaraju V, "Segmentation and Pre-Recognition of Arabic Handwriting," *Proceedings of the International Conference on Document Analysis and Recognition, pp. 605-609, 2005*.

[Lor06] Lorigo L and Govindaraju V, "Offline Handwriting Recognition: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, Issue 5, pp. 712–724, 2006.*

[Lu96] Lu Y and Shridhar M, "Character Segmentation in Handwritten Words – An Overview," *Pattern Recognition, 29(1):77–96, 1996.*

[Lu99] Lu Z, Schwartz R, Natarajan P, Bazzi I, and Makhoul J, "Advances in the BBN BYBLOS OCR System," *Proc. International Conference on Document Analysis and Recognition, 337-340, Bangalore, India, 1999.*

[Mad01] Madhvanath S and Govindaraju V, "The Role of Holistic Paradigms in Handwritten Word Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(2):149–164, 2001.*

[Mak95] Makhoul J and Schwartz R, "State of the Art in Continuous Speech Recognition," *Proc. National Academy of Sciences USA, Vol. 92, 9956-9963, 1995*.

[Mak98] Makhoul J, Schwartz R, LaPre C, and Bazzi I, "A Script-Independent Methodology for Optical Character Recognition," *Pattern Recognition, Vol. 31, No. 9, 1285-1294, 1998.*

[Man99] Mangu L, Brill E, and Stolke A, "Finding Consensus Among Words: Lattice-Based Word Error Minimization," *EuroSpeech'99, pp. 495-498, Budapest, September 1999.*

[Mar02] Märgner V, Pechwitz M, El Abed H, "ICDAR 2005 Arabic Handwriting Recognition Competition," *8th International Conference on Document Analysis and Recognition, ICDAR 2005, Aug. 29-Sep. 01, 2005, Seoul, Korea, (2005).*

[Mar02] Marti U V and Bunke H, "The IAM Database: An English Sentence Database for Offline Handwriting Recognition," *International Journal on Document Analysis and Recognition, vol. 5, no. 1, pp 39-46, Jan 2002.*

[Mar05] Margner V, Pechwitz M, and ElAbed H, "ICDAR 2005 Arabic Handwriting Recognition Competition," *Proc. Int'l Conf. Document Analysis and Recognition, pp. 1274-1278, 2007.*

[Mar07] Margner V, Pechwitz M, and ElAbed H, "Arabic Handwriting Recognition Competition," *Proc. Int'l Conf. Document Analysis and Recognition, pp. 70-74, 2005.*

[Mil01] Miled H and Ben Amara N E, "Planar Markov Modeling for Arabic Writing Recognition: Advancement State," *Proceedings of the International Conference on Document Analysis and Recognition, pp. 69-73, 2001*.

[Nat01] Natarajan P, Lu Z, Bazzi I, Schwartz R, and Makhoul J, "Multilingual Machine Printed OCR," *International Journal of Pattern Recognition and Artificial Intelligence, 15:1, 2001, pp. 43–63.*

[Nat05] Natarajan P, Sundaram R, Prasad R, and MacRostie E, "Character Duration Modeling for Speed Improvements in the BBN Byblos OCR System," *International Conference on Document Analysis and Recognition, Seoul, Korea, 2005.*

[Nat06] Natarajan P, Saleem S, Prasad R, MacRostie E, and Subramanian K, "Multi-lingual Offline Handwriting Recognition using Hidden Markov Models: A Script-independent Approach," *SACH'06 Proceedings of the 2006 Summit on Arabic and Chinese Handwriting Recognition, Springer-Verlag Berlin, 2008.*

[Nat08] Natarajan P, Saleem S, Prasad R, MacRostie E, and Subramanian K, "Multi-lingual Offline Handwriting Recognition Using Hidden Markov Models: A Script-Independent Approach," *Springer Book Chapter on Arabic and Chinese Handwriting Recognition, ISSN: 0302-9743, Vol. 4768, pp. 231-250, March 2008.*

[Nat09] Natarajan P, Subramanian K, Bhardwaj A, and Prasad R, "Stochastic Segment Modeling for Offline Arabic Handwriting," *International Conference on Document Analysis and Recognition, Barcelona, Spain, 2009.*

[Nat12] Natarajan P, Prasad R, Cao, H, Subramanian K, Saleem S, Belanger D, Vitaladevuni S, Kamali M, and MacRostie E, "Arabic Text Recognition using a Script-Independent Methodology: A Unified HMM-based Approach for Machine-print and Handwritten Text," *Springer Book on Guide To Arabic Script Recognition , To Appear in 2012.*

[Nat99] Natarajan R, Bazzi I, Lu Z, Schwartz R, Makhoul J, "Robust OCR of Degraded Documents," *International Conference on Document Analysis and Recognition, September 19-22, 1999, Bangalore, India.*

[Ngu95] Nguyen L, Anastasakos T, Kubala F, LaPre C, Makhoul J, Schwartz R, Yuan N, Zavaliagkos G, and Zhao Y, "The 1994 BBN/BYBLOS Speech Recognition System," *Proc. ARPA Spoken Language Systems Technology Workshop, Austin, TX, Morgan Kaufmann Publishers, 77-81, 1995.*

[Ost96] Ostendorf M, Digilakis V V, and Kimbal O A, "From HMM's to Segment Models: A Unified View of Stochastic Modeling for Speech

Recognition," *IEEE Transactions on Speech and Audio Processing, 4(5):360-378, 1996.*

[Pec02] Pechwitz M, Snoussi Maddouri S, Märgner V, Ellouze N, Amiri H, "IFN/ENIT-Database of Handwritten Arabic words," *7th Colloque International Francophone sur l'Ecrit et le Document, CIFED 2002, Oct. 21-23, 2002, Hammamet, Tunis, (2002).*

[Pec03] Pechwitz M and Margner V, "HMM Based Approach for Handwritten Arabic Word Recognition Using the IFN/ENIT-Database," *Proceedings of the International Conference on Document Analysis and Recognition, pp. 890-894, 2003*.

[Pen11] Peng X, Cao H, Prasad R, Natarajan P, "Text Extraction from Video Using Conditional Random Fields," *International Conference on Document Analysis and Recognition, Beijing, China, 2011*.

[Pha11] Phan T Q, Shivakumara P, Su B, Tan C L, "A Gradient Vector Flow-Based Method for Video Character Segmentation," *International Conference on Document Analysis and Recognition, Beijing, China, 2011*.

[Pla00] Plamondon R and Srihari SN, "On-line and Off-line Handwriting Recognition: A Comprehensive Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence 2000, 11(1):68–89.*

[Pra10a] Prasad R, Kamali M, Belanger D, Rosti A.-V, Matsoukas S, Natarajan P, "Consensus Network Based Hypothesis Combination for Arabic Offline Handwriting Recognition," *IAPR International Conference on Pattern Recognition (ICPR), Istanbul, Turkey, 2010*.

[Pra10b] Prasad R, Bhardwaj A, Subramanian K, Cao H, and Natarajan P, "Stochastic Segment Model Adaptation for Offline Handwriting Recognition," *IAPR International Conference on Pattern Recognition (ICPR), Istanbul, Turkey, 2010.*

[Pro00] Procter S, Illingworth J, and Mokhtarian F, "Cursive Handwriting Recognition using Hidden Markov Models and a Lexicon-driven Level Building Algorithm," *IEE Proceedings on Vision, Image and Signal Processing, 147(4):332–339, 2000.*

[Rab89] Rabiner L, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE, Vol. 77, 257-286, 1989*.

[Red84] Redner R A and Walker H F, "Mixture Densities, Maximum Likelihood and the EM Algorithm," *SIAM Review, Vol. 26, 195-239, 1984*.

[Saa11] Saabni R, El-Sana J, "Language-Independent Text Lines Extraction Using Seam Carving," *International Conference on Document Analysis and Recognition, Beijing, China, 2011.*

[Sal09] Saleem S, Subramanian K, Kamali M, Prasad R, and Natarajan P, "Improvements in BBN's HMM-based Offline Arabic Handwritten Recognition System," *Proc. International Conf. Document Analysis and Recognition, pp. 773-777, 2009.*

[Sar02] Sari T, Souici L, and Sellami M, "Off-Line Handwritten Arabic Character Segmentation Algorithm: ACSA," *Proceedings of the International Workshop on Frontiers in Handwriting Recognition, pp. 452-457, 2002.*

[Sch96] Schwartz R, Nguyen L, and Makhoul J, "Multiple-Pass Search Strategies," *in Automatic Speech and Speaker Recognition: Advanced Topics, C-H. Lee, F.K. Soong, K.K. Paliwal, Eds., Kluwer Academic Publishers, 429-456, 1996.*

[Sch96] Schwartz R, Nguyen L, and Makhoul J, "Multiple-Pass Search Strategies," *in Automatic Speech and Speaker Recognition: Advanced Topics, C-H. Lee, F.K. Soong, K.K. Paliwal, Eds., Kluwer Academic Publishers, 429-456, 1996.*

[Seb02] Sebastiani F, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys, vol. 34, no. 1, pp. 1-47, 2002.*

[Sei96] Seiler R, Schenkel M, and Eggimann F, "Off-line cursive handwriting recognition compared with on-line recognition," *Proc International Conference on Pattern Recognition, pp. 505–509, 1996.*

[Sen98] Senior A W, and Robinson A J, "An off-line cursive handwriting recognition system," *IEEE Transactions on Pattern Analysis and Machine Intelligence 1998; 20(3):309–321.*

[Sou04] Souici-Meslati L and Sellami M, "A Hybrid Approach for Arabic Literal Amounts Recognition," *The Arabian Journal of Science and Engineering, vol. 29, pp. 177-194, 2004.*

[Sri93] Srihari S.N., "Recognition of handwritten and machine-printed text for postal address interpretations," *Pattern Recognition Letters 1993; 14:291–302.*

[Sri97] Srihari S.N., Kuebert E.J., "Integration of Handwritten Address Interpretation Technology into the United States Postal Service Remote Computer Reader System," *Proc 4th International Conference on Document Analysis and Recognition, Ulm, Germany, 1997; 892–896.*

[Ste99] Steinherz T, Rivlin E, and Intrator N, "Off-line Cursive Script Word Recognition – a Survey," *International Journal on Document Analysis and Recognition, vol. 2, no. 2, pp 1-33, Feb 1999.*

[Sub10] Subramanian K, Manohar V, Cao H, Prasad R, Natarajan P, "Subword-based Stochastic Segment Modeling for Offline Arabic Handwriting Recognition," *International Workshop on Frontiers in Arabic Handwriting Recognition, Istanbul, Turkey, 2010.*

[Tay01] Tay Y H, Lallican P M, Khalid M, Viard-Gaudin C, and Knerr S, "An Offline Cursive Handwritten Word Recognition System," *Proceedings of IEEE Region 10 Conference, 2001*.

[Vin02] Vinciarelli A, "A Survey of Offline Cursive Word Recognition," *Pattern Recognition, vol. 35, no. 7, pp. 1433-1446, June 2002.*

[Vin03] Vinciarelli A and Perone M, "Combining online and offline handwriting recognition," *International Conference on Document Analysis and Recognition, pp. 844-848, 2003.*

[Vin04] Vinciarelli A, Bengio S, and Bunke H, "Offline Recognition of Unconstrained Handwritten Texts Using HMMs and Statistical Language Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(6), 2004*.

[Vlo92] Vlontzos J A and Kung S Y, "Hidden Markov models for character recognition," *IEEE Transactions on Image Processing, Vol. 1, pp. 539-543, 1992*.