Holistic Word Recognition for Handwritten Historical Documents

Victor Lavrenko, Toni M. Rath and R. Manmatha*
[lavrenko, trath, manmatha]@cs.umass.edu
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
Amherst, MA 01003

Abstract

Most offline handwriting recognition approaches proceed by segmenting words into smaller pieces (usually characters) which are recognized separately. The recognition result of a word is then the composition of the individually recognized parts. Inspired by results in cognitive psychology, researchers have begun to focus on holistic word recognition approaches. Here we present a holistic word recognition approach for single-author historical documents, which is motivated by the fact that for severely degraded documents a segmentation of words into characters will produce very poor results. The quality of the original documents does not allow us to recognize them with high accuracy - our goal here is to produce transcriptions that will allow successful retrieval of images, which has been shown to be feasible even in such noisy environments.

We believe that this is the first systematic approach to recognizing words in historical manuscripts with extensive experiments. Our experiments show a recognition accuracy of 65%, which exceeds performance of other systems that operate on non-degraded input images (non historical documents).

1. Introduction

Despite results from cognitive psychology, which indicate that humans largely recognize words *holistically* when reading text, much of the handwriting recognition research has focused on *analytical* character recognition approaches. In this paradigm, words are broken down into characters (or other units), which are then individually recognized to determine the correct word label.

However, this approach requires one to determine the character boundaries [1], which can only be achieved by having already recognized the characters. This paradox has led researchers to consider over-segmentations, multiple segmentations and other similar strategies to address the problem. More recently, the holistic approach [10] has gained in popularity as an attractive and more straightforward solution. Holistic recognition approaches treat words as an inseparable unit. No segmentation is performed and the whole word is recognized at once.

While holistic approaches may be most popular for their simplicity and their parallels to the human reading apparatus, our approach here is driven by an entirely different motivation: most handwriting research is conducted on recently acquired data of good quality. There are, however, documents that are significantly degraded, e.g. the manuscripts of George Washington at the Library of Congress, which we used in our experiments (see Figure 1 for an example). Such documents are of great interest to a broad community of researchers, scholars and the general public. However, their poor quality makes it difficult to recognize them using the analytical character recognition approach that employs character segmentation.

Here we provide an approach to the recognition of whole words in such collections. A document is described using a Hidden Markov Model [14], where words to be recognized represent hidden states. The state transition probabilities are estimated from word bigram frequencies. Our observations are the feature representations of the word images in the document to be recognized. We use feature vectors of fixed length, using features ranging from coarse (e.g. word length) to more detailed descriptions (e.g. word profile). The collection that was used to train and evaluate the recognition system consists of 20 page images with a total of 4856 words. We believe that this is the first systematic approach to recognizing words in historical documents.

* This work was supported in part by the Center for Intelligent Information Retrieval and in part by the National Science Foundation under grant number IIS-9909073 and in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

Our focus is not on producing perfect transcriptions, but rather on achieving reasonable recognition accuracy, which enables us to perform retrieval of handwritten pages from a user-supplied ASCII query. Currently it is not feasible to produce perfect or even near-perfect recognition results for document corpora like the George Washington collection. However, it has been demonstrated [6, 5] that satisfactory retrieval can still be performed in related noisy environments such as speech and printed words using the noisy outputs of recognizers.

An additional application for our model is the automatic alignment of a transcription with a page like in [18] (e.g. for training purposes). In this scenario the recognition lexicon is constrained to that of a supplied transcript, which is also used to estimate unigram/bigram frequencies. The recognition output can then be used to align lexicon terms and their respective location in the page image.

The remainder of this article is organized as follows: the next section puts our work in context with related work, then we introduce our holistic word features in section 2, followed by the description of our recognition model 3. Next, we present our experimental evaluation (section 4) and then conclude the paper in section 5.

1.1. Related Work

In recent years, research in handwriting recognition [13] has advanced to a level that makes commercial applications (e.g. Tablet PCs) feasible. However, this success has been mostly limited to the *online* handwriting recognition case, where the pen movements are recorded as a user writes. *Offline* handwriting recognition, that is, recognition of text from an image of the writing, has only been successful in small-vocabulary and highly redundant domains such as automatic check processing and mail sorting (e.g. [9]). More recently, the community has started to look at large-vocabulary tasks [19].

While the recognition output of such systems will not satisfy a human reader, it can be used for text retrieval. Results from information retrieval on ASR (automatic speech recognition) output [5] and OCR (optical character recognition) output [6] indicate that retrieval performance does not drop significantly even for 30% word error rates when compared to retrieval performed on undistorted text.

The authors of [10] provide a survey of the holistic paradigm in human reading and its applications in handwritten word recognition. They show that holistic word recognition is a viable alternative to the popular analytical (character segmentation-based) approach to handwriting recognition, and point out parallels in human reading studies. In particular the *word-superiority effect* [2], which states that humans are able to recognize certain words faster than it takes them to recognize an individual character. This is strong evidence in favor of a holistic recognition process.

In [11], the authors discuss the application of a Hidden Markov model for recognizing handwritten material that was produced specifically for this purpose. First, they asked a number of subjects to write out a set of pages. To improve the quality of the writing, the subjects were asked to use rulers and not to split words across lines. Recognition was performed with a Hidden Markov model with 14 states for each character. These Markov models were concatenated to produce word and line models. A statistical language model was used to compute bigrams and the authors showed that this improved the results by about 10%. The authors showed a recognition rate of about 60% for vocabulary sizes ranging from 2703 to 7719 words. The paper also contains a discussion of recognition rates obtained by other researchers - these varied from a recognition rate of 42.5% for a 525 word vocabulary and 37% for a 966 word vocabulary reported in [12] to a recognition rate of 55.6% in a 1600 word vocabulary reported by [8].

The work in this paper focuses on recognizing historical handwritten manuscripts using simple HMMs - one state for each word. We show that results of comparable quality may be obtained for this problem. Although there is great value in the preservation of handwritten historical documents, little research has been undertaken in this area, presumably due to the challenges in this domain - which include degraded manuscripts with ink bleed and poor scanning quality. The domain was tackled by [16], who use word spotting - they form equivalence classes of words by matching them in pairs. Equivalence classes with certain word frequencies can then be used for indexing by having a human annotate them. While this process can provide fast and good-quality retrieval, the cost for creating equivalence classes is currently prohibitive for very large collections.

Although [16] allows the retrieval of handwritten documents based on ASCII queries, the technique makes use of image matching techniques, rather than attempting to recognize word images. The results in [18] have shown how difficult this can be: the authors aligned a page of Thomas Jefferson with its manually generated transcript using recognition. To achieve reasonable performance, they had to assume that a manually generated transcript of the page was available, and they had to restrict the lexicon (to an average of 13 words). Given these restrictions, the overall alignment accuracy was 83%. We note that while the use of a transcript is a reasonable strategy for the task of alignment, it is not reasonable when the task is recognition. In the work here, we assume that the recognizer has no knowledge of the transcript of the test page and further that the entire vocabulary is used.

All handwriting recognition systems rely on preprocessing routines in order to normalize the variations that

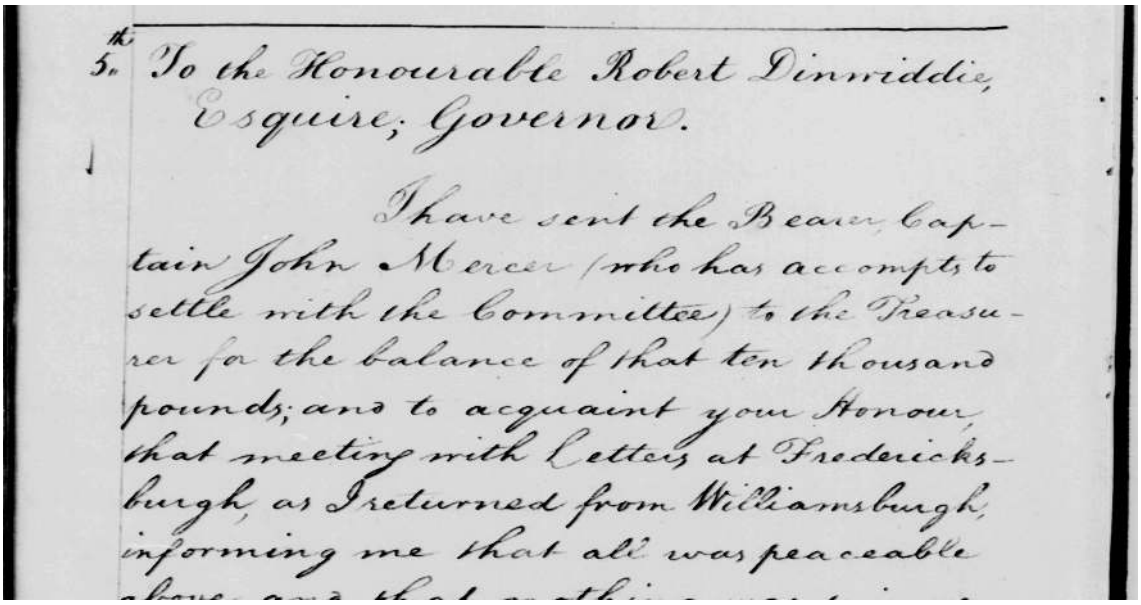


Figure 1: Part of a scanned document from the George Washington collection.

are present even in single-author handwriting. We perform slant/skew/baseline normalizations that are commonly used in the literature (for example, see [11]). While some of our features are generally used for the recognition of handwritten characters [17], we use them to represent the shape of entire words.

2. Features

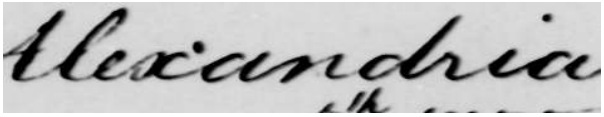
Many word images can be distinguished easily by looking at simple holistic features such as the width in pixels. However, differing word images with the same coarse features require a more detailed description in order to distinguish between them. Previous work [15, 16] has shown the value of profile-based features (e.g. projection profiles) for this task. Consequently, the feature set we use here consists of a coarse-to-fine range of features.

Our observation model for a given word image describes its features as being drawn from a state-conditional distribution. In order to easily describe this process, we represent each word image I_j by a feature vector f_j of fixed length ($D = 27$ dimensions). While scalar features such as word image length can be easily used as a feature vector dimension in this framework, the profile-based features – which vary with the width of a word image – have to be turned into a fixed-length description. This is achieved by using the low order coefficients from a DFT (Discrete Fourier Transform) of each of the original profile-based features.

Together, the scalar and profile-based features form a vector of fixed-length for word images of all sizes. In our experiments, we also normalized the range of each feature dimension to the unit interval. The entire process of feature vector generation is illustrated in Figure 2. In the following sections, we will first describe the image normalizations we perform prior to the feature extraction. Then our feature set is described, starting with the scalar features.

2.1. Image Normalization

One of the major challenges in recognizing handwriting is its variability. Even in the case of single-author handwriting, there are differences between the same words written at different times. We compensate for part of these variations by normalizing the slant and skew in handwriting.



(a) original image, as segmented from document,

(b) after cleaning and normalization.

Figure 3: Image cleaning and normalization.

Figure 3(a) shows an original image, which has been seg-

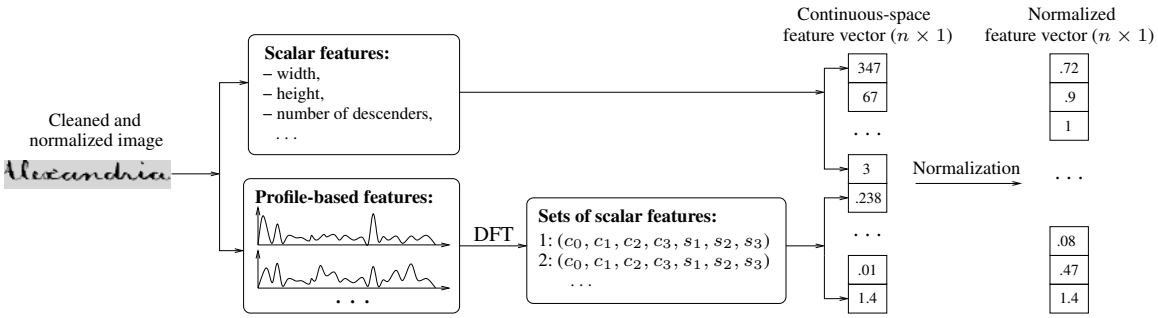


Figure 2: Feature generation process.

mented from a document. In the first step, we remove parts from other words that reach into the bounding box. Then we estimate and remove slant and skew (cursive angle and inclination of writing). In the last step, the background of the image is set to a constant color and the image is padded in order to move the baseline[1] to a predefined location. This ensures that all images are divided into two parts of the same proportions by their respective baselines. Figure 3(b) shows a typical result of these normalization steps.

2.2. Scalar Features

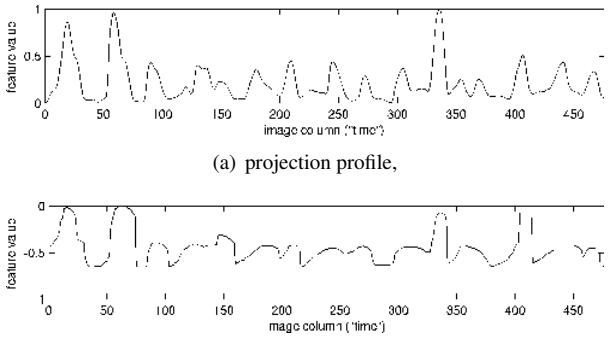
Each of the features described here, can be expressed by a single number. Part of them have been used previously (see e.g. [16]) to quickly determine coarse similarity between word images. For a given image with tight bounding box (no extra space around the word) we extract:

1. the height h,
2. the width w,
3. the aspect ratio w/h,
4. the area $w \cdot h$, and
5. an estimate of the number of descenders in the word, i.e., strokes below the baseline (e.g. lower part of 'p').
6. an estimate of the number of ascenders in the word.

While the aspect ratio and area features are redundant, their distributions differ from those of the height and width features.

2.3. Profile-Based Features

The variable-length features we use, give a much more detailed view of a word's shape than single-valued features can. All of the profile features below have been successfully used in a whole-word matching approach [16]. Each feature results from recording a single number per image column of the word, thus creating a "time series" (x-axis = time) of the same length as the width of the image.



(a) projection profile,

(b) upper word profile.

Figure 4: Two of the three utilized profile features. Both features were directly extracted from image 3(b).

We generate three time series:

1. Projection Profile: Each time series value is the sum of the pixel intensities in the corresponding image column (see Figure 4(a) for an example).
2. Upper Word Profile: Each value is the distance from the top of the word's bounding box to the first "ink" pixel in the corresponding image column (see Figure 4(b)).
3. Lower Word Profile: The same as the upper word profile, but the distance is measured from the bottom of the image bounding box.

The quality of these features strongly depends on good normalization, as detailed in section 2.1. For example, slant can affect the visibility of parts of words in terms of the word profile features (e.g. the 'l' leaning over the 'e' in Figure 3(a)).

While these time series features capture the shape of a word in great detail, they vary in length, and thus cannot

1 The baseline is the imaginary line that people write on.

be used in our framework, which requires fixed-length feature vectors. A time series can be adequately approximated by the lower-order coefficients of its Discrete Fourier Transform (DFT) [3]. The DFT representation also takes into account that images can have different lengths, since one period of the DFT basis functions is equal to the number of sample points.

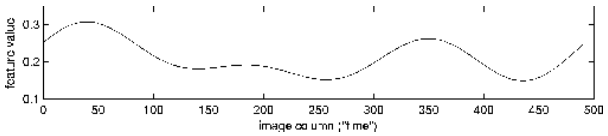


Figure 5: Projection profile time series from Figure 4(a), reconstructed using 4 lowest-order DFT coefficients.

We perform the DFT on the time series $\mathbf{f} = f_0 \ldots f_{n-1}$ to get its frequency-space representation $\mathbf{F} = F_0 \ldots F_{n-1}$:

$$F_k = \sum_{l=0}^{n-1} f_l \cdot \mathrm{e}^{-2\pi i l k/n}, \qquad 0 \le k \le n-1. \qquad (1)$$

From the DFT representation we extract the first 4 real (cosine) components and 3 imaginary (sine) components[2] for use as scalar features. Figure 5 shows a reproduction of the time series in Figure 4(a) using these features. For our purposes, this approximation suffices, since the goal is not to represent the original signal in all details, but rather to capture the global word shape with a small number of descriptors.

3. Mathematical Model

In this section we formalize handwriting as a Markov Process with hidden states corresponding to words in the vocabulary and observable states representing a noisy (handwritten) rendition of those words. Let V be the set of words in a given language, and suppose a given author is trying to create a manuscript of length n. First, we shall assume that the author has an extremely short memory span and is able to keep in mind only the last word she has written. Given that the last word was $w_{j-1} \in V$, the author picks the next word w_j according to some probability distribution $P(w_j|w_{j-1})$. Then the author decides on a general shape in which she will write out the word. As we described in section 2, the shape of the word is represented by a D-dimensional real-valued vector of features $f_j \in \mathbb{R}^D$. The shape f_j depends on which word

2 For real-valued signals, the first imaginary coefficient of the DFT is always 0.

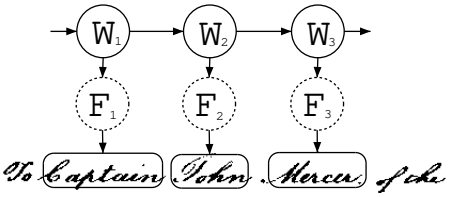


Figure 6: A Hidden Markov Model for capturing the handwriting process. Words (W_i) represent the hidden states in the author's head. From these states the author samples feature vectors (F_i) which describe the intended shape of the word. The author then writes out the word according to the shape feature vector.

w_j the user decided to generate; we model this dependency by a word-conditional probability density: $p(f_j|w_j)$. The density accounts for the variation in the handwritten shapes of the same word. Once a shape is determined, the user writes the word down, producing a word image I_j. The entire process is illustrated in Figure 6.

The above process can be captured by a Hidden Markov Model [14]. The words $w_1 \ldots w_n$ represent the state sequence. Each state depends only on the previous state and the transitions are governed by the distribution $P(w_j|w_{j-1})$. Feature vectors f_j represent the observations, which are conditionally independent of each other given the identity of the word w_j. We assume that the final word image I_j is a deterministic function of f_j. Under this process, the probability of generating the sequence of words $w_1 \ldots w_n$ together with their handwritten forms $I_1 \ldots I_n$ is given by the following expression:

$$P(w_1 \ldots w_n, I_1 \ldots I_n) = \prod_{j=1}^{n} P(w_j|w_{j-1}) p(f_j|w_j) \qquad (2)$$

In the above formula, we assume that the shape vector f_j completely determines the handwritten form I_j, so there is no need to model $P(I_j|f_j)$. As a matter of convenience we take w_0 to be a special unwritten word that always marks the beginning of discourse. In the remainder of this section we describe the components of the model and discuss how the model can be used for transcribing a handwritten document.

3.1. Using the Model for Transcription

Suppose we are given a sequence of word images $I_1 \ldots I_n$ and want to generate its transcription $v_1 \ldots v_n$. The model outlined above would guide us to search for the sequence $w_1 \ldots w_n$ that is the most likely state sequence given the observations $I_1 \ldots I_n$. In other words, the ideal transcription would satisfy:

$$v_1 \ldots v_n = \arg \max_{w_1 \ldots w_n} P(w_1 \ldots w_n | I_1 \ldots I_n) \quad (3)$$

Note that the conditional probability in equation (3) and the joint probability in equation (2) are different by a factor independent of the state sequence $w_1 \ldots w_n$. Consequently, we can simply search for the state sequence that maximizes equation (2). While there are exponentially many candidate sequences, we can quickly find the best sequence by using the Viterbi algorithm[4]. In order to use the algorithm we need to specify the two components of our model: the state transition probabilities $P(w|v)$ and the observation density $p(f|w)$.

3.2. State Transition Probabilities

The standard procedure for learning the parameters of the Hidden Markov Model is the Baum-Welch algorithm[14]. However, in our case the learning can be substantially simplified, since our states correspond to words in the English vocabulary V. This allows us to learn the state transition probabilities $P(w|v)$ by simply counting how many times word w follows the word v in a large collection of text. Similarly, we can estimate the initial state probabilities $P(w|w_0)$ by the relative frequency of w in the same collection. If T is a collection of text, the probabilities would take the form:

$$\begin{aligned} P_T(w|w_0) &= \left(\frac{\text{number of times } w \text{ occurs in } T}{\text{total number of words in } T} \right) \\ P_T(w|v) &= \left(\frac{\text{number of times } v, w \text{ occurs in } T}{\text{number of times } v \text{ occurs in } T} \right) \end{aligned} \quad (4)$$

The choice of collection T has the most direct impact on the quality of the resulting probability estimates. Ideally, we would like to have unlimited quantities of text by the writer of the manuscript. However, in practice we usually have large volumes of text from other sources (T_O) and only a small amount of text from the target author (T_A). We used simple averaging to smooth the probability estimates from T_O and T_A[3] :

$$\begin{aligned} \hat{P}(w|w_0) &= \frac{1}{3} \left[P_{T_A}(w|w_0) + P_{T_O}(w|w_0) + \frac{1}{|V|} \right] \\ \hat{P}(w|v) &= \frac{1}{3} \left[P_{T_A}(w|v) + P_{T_O}(w|v) + \hat{P}(w|w_0) \right] \end{aligned} \quad (5)$$

3 Note: rather than using uniform weights of $\frac{1}{3}$ for each component, we could have tuned the weights to optimize likelihood of a held-out portion of text. In our experiments tuning did not lead to significant improvements in the overall accuracy of the model.

The uniform probability $\frac{1}{|V|}$ is necessary to avoid zero probabilities. Note that we interpolate the bigram and the unigram probabilities in equation (5).

3.3. Observation Probabilities

We chose to model the observation probabilities $p(f|w)$ by a multi-variate normal distribution. Each state w corresponds to a Gaussian with mean μ_w and covariance matrix Σ_w. The likelihood of seeing observation f in state w is given by the following expression:

$$p(f|w) = \frac{\exp\left\{-\frac{1}{2}(f-\mu_w)^\top \Sigma_w^{-1}(f-\mu_w)\right\}}{\sqrt{2^D \pi^D |\Sigma_w|}} \quad (6)$$

Here $|\Sigma_w|$ denotes the determinant of the covariance matrix, and D is the number of features (dimensions) used to represent the word image. In order to estimate the parameters μ_w and Σ_w for each word w we need a training set of transcribed manuscripts. Suppose that $g_{w,1} \ldots g_{w,k}$ are the different feature vectors for images of the word w in the training set. Then the mean μ_w can be estimated as:

$$\mu_w[d] = \frac{1}{k} \sum_{i=1}^{k} g_{w,i}[d] \;, \quad d = 1 \ldots D \quad (7)$$

Here d denotes a particular dimension of the feature vectors. The covariance matrix Σ_w can be estimated accurately only if the training set contains sufficiently many examples $g_{i,w}$ of the word w. In our experiments this was never the case, so we approximated the covariance matrix as $\Sigma_w \approx \sigma_{avg} \cdot I$. Here I is the identity matrix and σ_{avg} is the mean feature variance, computed as:

$$\sigma_{avg} = \frac{1}{D} \sum_{d=1}^{D} \left(\frac{1}{N_{tr}-1} \sum_{i=1}^{N_{tr}} (g_i[d] - \mu[d])^2 \right) \quad (8)$$

In equation(8), vector μ represents the average of all vectors in the training set. The inside summation goes over all N_{tr} vectors g_i in the training set and represents the variance along dimension d. The outside summation goes over the dimensions $d = 1 \ldots D$ of the feature vectors.

4. Experimental Evaluation

In this section we describe a set of experiments we carried out to test the effectiveness of our model on the task of transcribing a collection of George Washington's manuscripts. We start by describing the dataset, training resources and the evaluation procedure. Then we look at a sample transcription and analyze the errors that were made by our model. Finally, we take a detailed look at how the amount of training data affects recognition errors.

4.1. Experimental Setup

Our evaluation corpus consists of a set of 20 pages from a collection of letters by George Washington. The collection is accurately segmented into words, and we have manually transcribed every word image. We do not lowercase the transcribed words, so *"Fort"* and *"fort"* are treated as two different words. We do not transcribe punctuation and non-alphanumeric characters. There are a total of 4856 words in the collection, 1187 of them unique. This is a relatively small amount of data, and we have to use it both for learning the parameters of the model and for evaluating the performance. To improve the stability of our results we carried out 20-fold cross-validation as follows. During each iteration we pick one page as our testing page, estimate the model from the remaining 19 pages and test the performance on the testing page. We also evaluate the model when fewer than 19 pages are used for estimation.

4.1.1. Evaluation Measures. We use Word Error Rate (WER) as our measure of performance. WER is a proportion of the words that were not recovered exactly as they were in the manual transcript. As we perform cross-validation, we report the mean error rate across 20 pages along with the standard deviation. Because of the relatively small size of our dataset, a large proportion of the errors is caused by out-of-vocabulary (OOV) words. These are the words which occur only in the testing page, and not in any of the training pages. Since we model the words in their entirety, we cannot possibly provide the correct transcription for these words. To separate the OOV errors from mismatches we report two types of WER, one that includes OOV words and one that omits them from evaluation.

4.1.2. Additional Resources. In addition to the 20 pages of transcribed manuscripts, we have access to a large electronic collection of writings by George Washington and Thomas Jefferson. The collection contains over 4.5 million words, which is enormous compared to the 4.6 thousand words we have in the 20 handwritten pages. We experimented with different ways of estimating the bigram model $P(w|v)$, which governs state transitions in our model. For one experiment, we used only the training set to estimate bigrams. For a second experiment we added the Jefferson portion of the Washington-Jefferson collection. For a third experiment we used the entire collection, including the Washington portion. We took particular care to remove the 20-page testing set from this collection, so there is no possible overlap between the training and testing sets. The Washington-Jefferson collection will be used as "other sources" (T_O) in our experiments. It is worth noting that the bigrams from this collection do not represent a perfect fit to the 20-page set. The reason is that in the 20-page set a large number of words are split (hyphenated) across different lines. None of the split words are present in the electronic Washington-Jefferson collection.

4.2. Example Transcription

Figure 7 gives an example of transcription that could be generated by our model. The example is a letter from the last page of the dataset. The model was estimated from the preceding 19 pages and used the Washington-Jefferson collection for bigrams (along with bigrams from the 19 training pages). Our model made 7 errors in the transcript of the letter, this corresponds to a 17% word error rate, which is substantially better than our average performance. The errors are underlined in Figure 7. Out of 7 mistakes, two are due to out-of-vocabulary (OOV) words: *"hereof"* and *"5th"* were not found in the 19 training pages and were transcribed as *"twenty"* and *"16th"* respectively. Our model replaced the word *"meet"* with a similar-looking *"not"*. It is not immediately clear what could account for replacing *"Captain John Mercer"* with *"Captain Peter Bacon"*.

4.3. Impact of State Transition Probabilities

Transcription accuracy is greatly influenced by the amount and the quality of the training data. In the remainder of this section we try to quantify the effect of training data and provide a reasonable estimate for the amount of resources that would be necessary to achieve satisfactory transcription. We start by looking at our state transition model. We consider 5 possible ways to estimate transition probabilities $P(w|v)$:

1. **None:** In this case we make no attempt to model the transitions. $P(w|v)$ is simply $1/|V|$ for all words w and v. For every feature vector f, the model will select the word w with the highest $p(f|w)$, even if w is very unlikely in this position. Note that Viterbi decoding is unnecessary in this case.
2. **Unigram:** We model relative frequencies of the words, but not transitions from one word to another. $P(w|v)$ is the same as $P(w|w_0)$, independent of v. Probabilities are estimated from the training 19 pages. Viterbi is, again, unnecessary, the model will pick a word w with the highest $p(f|w)P(w|w_0)$.
3. **19 pages:** We estimate the bigram model from the 19 pages in the training set and interpolate them with the unigram model.
4. **19+Jeff:** Get the bigrams from the Jefferson portion of the electronic collection, interpolate them with the 19-page bigrams and unigrams.
5. **19+J+W:** Get bigrams from both Washington and Jefferson together, interpolate with the 19-page bigrams and unigrams.

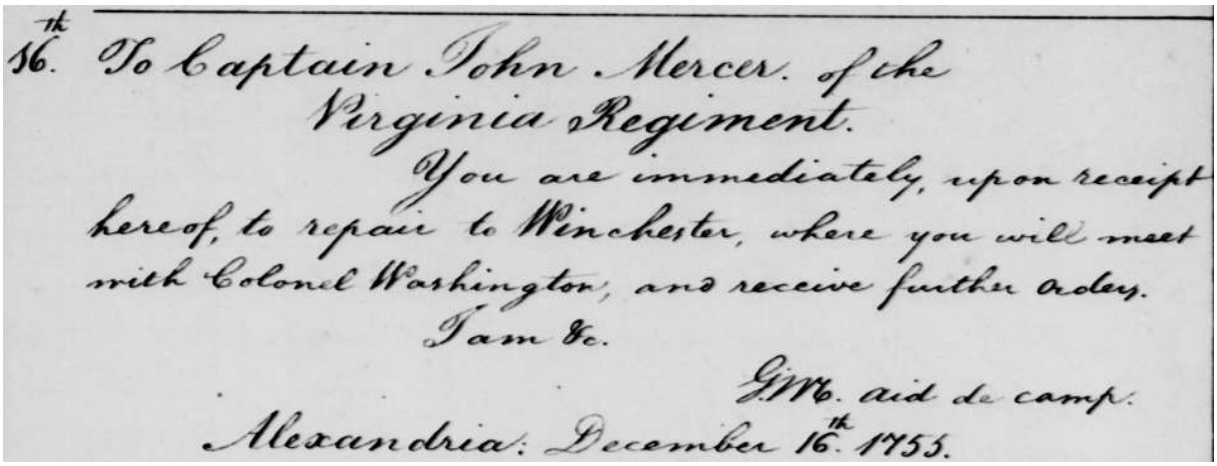


Figure 7: Part of a scanned letter by George Washington and the automatic transcription generated by our model.

6. **Target:** Estimate the bigrams from the testing page, interpolate with unigrams from the testing page. This is a cheating experiment, meant to provide an upper bound on bigram performance.

State Transition Model		Word Error Rate	
Source	Size	exclude OOV	include OOV
None	0	0.531 $\pm$ 0.05	0.603 $\pm$ 0.05
Unigram	4.6K	0.448 $\pm$ 0.05	0.533 $\pm$ 0.05
19 pages	4.6K	0.414 $\pm$ 0.06	0.503 $\pm$ 0.07
19+Jeff	191K	0.388 $\pm$ 0.05	0.481 $\pm$ 0.06
19+J+W	4,533K	0.349 $\pm$ 0.06	0.449 $\pm$ 0.07
Target	243	0.045 $\pm$ 0.03	0.063 $\pm$ 0.04

Table 1: Effect of different ways of estimating the state transition probabilities. See section 4.3 for a detailed description of sources. Size refers to the aggregate number of words from which bigrams were estimated.

For each of these conditions we perform 20-fold cross-validation with 19 training and 1 testing pages. We report mean word error rate and standard deviation, both with and without OOV. Table 1 shows the results. We observe that without modeling state transition we get mean WER of over 53%. Adding prior (unigram) probabilities from the training pages reduces WER to 45%, and adding the training bigrams drops it to 41%. The drop in word error rate from 53% to 41% is statistically significant and makes a strong case for the importance of modeling state transitions. By adding the bigrams from the Jefferson collection we are able to bring the error rate down to 39%, and with Washington the error reduces to 35%, marking a substantial improvement over the 41% WER we achieved by using the 19 training pages alone. The last condition, using the bigrams from the testing page, is a cheating experiment. While it does not represent a valid transcription experiment, it suggests that our system could be used to *align* manuscripts and their transcripts with an alignment error rate of around 6%.

4.4. Impact of Observation Probabilities

Now we turn our attention to the observation probabilities $p(f|w)$ which model the generation of feature vectors from word states. This part of the model is trained entirely from the manually transcribed manuscript pages, there are no external resources. We want to get a sense of how the number of training pages affects transcription accuracy. In order to do that we modify our cross-validation procedure to use $n < 19$ training pages in each split. State transition bigrams are interpolated with the Washington-Jefferson collection. For each n we record the mean number of words in the training portion, the mean number of out-of-vocabulary words on the testing page, and the mean error rates. Results are presented in Figure 8 and in Table 2. We observe that with small number of training pages the OOV errors are extremely high, while the non-OOV errors remain below 45% even with just one training page. This is under-

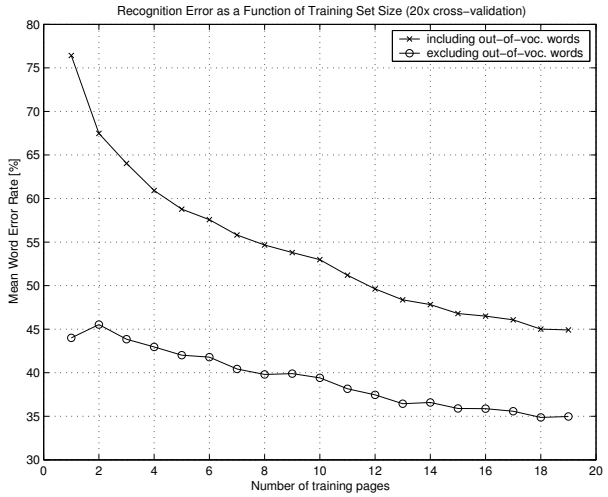


Figure 8: The number of training pages has a strong effect on transcription accuracy, especially on the out-of-vocabulary errors.

standable, since more than half of all the words are out-of-vocabulary. With 10 training pages one out of five words is out-of-vocabulary, and the non-OOV error rate is a respectable 39%. With 19 training pages the OOV error is down to 45%, while non-OOV WER is at 35%.

4.5. Summary of Results

Under favorable training condition our model achieved a mean word error rate of 0.349, which corresponds to recognition accuracy of 65%. This compares very favorably with previously published results [8, 11, 12] which were obtained with similar vocabulary sizes but on much cleaner collections. One may argue that the resources we used to achieve 65% accuracy may not always be available in an operational setting. However, tables 1 and 2 demonstrate that the model can achieve a reasonable accuracy (55-60%) even with relatively meager resources, such as 10 training pages and a contemporary text collection for estimating the bigrams (e.g. the Jefferson collection). Finally, the last experiment in Table 1 suggests that the current model could be used to align manuscripts and transcripts with an accuracy of at least 93%.

4.6. Model Improvements

Our current recognition suffers from a number of weaknesses. Here we would like to address some of these issues and provide suggestions as to how they could be resolved.

1. Out of vocabulary terms: The current system can only recognize words that are in the training vocabulary. Among the several strategies that are possible, automatic generation of training instances would fit naturally into our current model. We could create artificial training instances for words which have few or no occurrences in the training set (e.g. see [7]). The simplest way would be to create words from a representative font (possibly with multiple different styles for each letter) which is extracted from the training documents. In order to further increase the training set size and to simulate real conditions once could distort either the artificial word images or their feature representations. When doing this, it is important to ensure that the distortions of word images will not be reversed by the preprocessing stage.
2. N-gram model: To a great extent the quality of the recognition results depends on the accuracy of the N-gram (unigram, bigram, ...) frequency estimates that are used. Higher accuracy can be achieved by extending the training corpus that is used to estimate such frequencies. However, for historical documents one not only has to consider factors such as the particular topic of discussion and writing style of the author, but also the spelling of words. This means a suitable corpus for N-gram estimation would also have to be picked from the same period of time as the document that is to be recognized.
3. Hyphenations: Our current document collection contains many hyphenations, which break words into two parts. Given an already large vocabulary, it is virtually impossible to obtain training data for all hyphenations of words in the vocabulary. This problem could be addressed by implementing a hyphenation detector into the document preprocessing stage. When a hyphenation is detected, the two parts of a word image are concatenated across lines in order to form a single word.
   The detection and removal of hyphenations would also have an additional positive effect on the recognition accuracy: bigram estimates from a text corpus are not in general created for hyphenated words. With no more hyphenated word images, all of the segmented images in a document correspond to complete words.

5. Conclusions

We have presented a handwriting recognition approach for single-author historic manuscripts with large vocabularies. In order to address the poor quality of the page images we have chosen a holistic word recognition approach that does not require character segmentation. The error rates we achieve are comparable to those of multi-writer recognition systems for high-quality input pages. While the accuracy is not yet sufficient to produce automatic transcripts that will

Train.	Training words		Testing words		OOV words		Word Error Rate	
pages	total	unique	total	unique	total	unique	excluding OOV	including OOV
1	216	132	242	150	143	112	0.440 $\pm$ 0.07	0.764 $\pm$ 0.05
3	731	321	242	150	88	78	0.439 $\pm$ 0.07	0.640 $\pm$ 0.07
5	1223	457	242	150	71	65	0.420 $\pm$ 0.07	0.588 $\pm$ 0.07
10	2381	702	242	150	56	51	0.394 $\pm$ 0.07	0.530 $\pm$ 0.08
15	3679	1012	242	150	42	39	0.359 $\pm$ 0.06	0.468 $\pm$ 0.06
19	4613	1151	242	150	37	35	0.349 $\pm$ 0.06	0.449 $\pm$ 0.07

Table 2: The number of training pages has a strong effect on transcription accuracy, especially on the out-of-vocabulary errors.

be acceptable for human readers, successful retrieval of historic documents and transcription alignment can already be performed.

Acknowledgments

We would like to thank the Library of Congress for providing scanned versions of George Washington's handwritten manuscripts. Maximo Carreras co-authored the Viterbi decoding implementation we used in our experiments.

References

[1] R. G. Casey and E. Lecolinet: *A Survey of Methods and Strategies in Character Segmentation*. IEEE Trans. on Pattern Analysis and Machine Intelligence **18**:7 (1996) 690-706.

[2] J. M. Cattell: *The Time Taken Up by Cerebral Operations*. Mind **11** (1886) 220-242.

[3] C. Faloutsos: *Multimedia IR: Indexing and Searching*. In: Modern Information Retrieval, R. Baeza-Yates and B. Ribeiro-Neto; Addison-Wesley, Reading, MA, 1999.

[4] G. D. Forney: *The Viterbi Algorithm*. Proc. of the IEEE **61** (1973) 268-278.

[5] J. S. Garofolo, C. G. P. Auzanne and E. M. Voorhees: *The TREC Spoken Document Retrieval Track: A Success Story*. In: Proc. of RIAO 2000, Content-Based Multimedia Information Access, vol. 1, Paris, France, April 12-14, 2000, pp. 1-20.

[6] S. M. Harding, W. B. Croft and C. Weir: *Probabilistic Retrieval of OCR Degraded Text Using N-Grams*. In: Proc. of the 1st European Conference on Research and Advanced Technology for Digital Libraries. Pisa, Italy, September 1-3, 1997, pp. 345-359.

[7] E. Ishidera and D. Nishiwaki: *A Study on Top-down Word Image Generation for Handwritten Word Recognition*. In: Proc. of the 7th Int'l Conf. on Document Analysis and Recognition, vol. 2. Edinburgh, Scotland, August 3-6, 2003, pp. 1173-1177.

[8] G. Kim, V. Govindaraju and S. N. Srihari: *An Architecture for Handwritten Text Recognition Systems*. Int'l Journal on Document Analysis and Recognition **2**:1 (1999) 37-44.

[9] A. Kornai, K. M. Mohiuddin and S. D. Connell: *Recognition of Cursive Writing on Personal Checks*. In: Proc. of the 5th Int'l Workshop on Frontiers in Handwriting Recognition. Colchester, UK, September 2-5, 1996.

[10] S. Madhvanath and V. Govindaraju: *The Role of Holistic Paradigms in Handwritten Word Recognition*. Trans. on Pattern Analysis and Machine Intelligence **23**:2 (2001) 149-164.

[11] U.-V. Marti and H. Bunke: *Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwriting Recognition System*. Int'l Journal of Pattern Recognition and Artifical Intelligence **15**:1 (2001) 65-90.

[12] T. Pacquet and Y. Lecourtier: *Recognition of Handwritten Sentences using a Restricted Lexicon*. Pattern Recognition **26**:3 (1993) 391-407.

[13] R. Plamondon and S. N. Srihari: *On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey*. IEEE Trans. on Pattern Analysis and Machine Intelligence **22**:1 (2000) 63-84.

[14] L. R. Rabiner: *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. Proc. of the IEEE **77**:2 (1989) 257-286.

[15] T. M. Rath and R. Manmatha: *Features for Word Spotting in Historical Manuscripts*. In: Proc. of the 7th Int'l Conf. on Document Analysis and Recognition (ICDAR), Edinburgh, Scotland, August 3-6, 2003, vol. 1, pp. 218-222.

[16] T. M. Rath and R. Manmatha: *Word Image Matching Using Dynamic Time Warping*. In: Proc. of the Conf. on Computer Vision and Pattern Recognition, Madison, WI, June 18-20, 2003, vol. 2, pp. 521-527.

[17] Ø. D. Trier, A. K. Jain and T. Taxt: *Feature Extraction Methods for Character Recognition - A Survey*. Pattern Recognition **29**:4 (1996) 641-662.

[18] C. I. Tomai, B. Zhang and V. Govindaraju: *Transcript Mapping for Historic Handwritten Document Images*. In: Proc. of the 8th Int'l Workshop on Frontiers in Handwriting Recognition 2002, Niagara-on-the-Lake, ON, August 6-8, 2002, pp. 413-418.

[19] A. Vinciarelli, S. Bengio and H. Bunke: *Offline Recognition of Large Vocabulary Cursive Handwritten Text*. In: Proc. of the 7th Int'l Conf. on Document Analysis and Recognition, vol. 1. Edinburgh, Scotland, August 3-6, 2003, pp. 1101-1105.