Aligning Transcripts to Automatically Segmented Handwritten Manuscripts

Jamie Rothfeder, R. Manmatha, and Toni M. Rath

Department of Computer Science,
University of Massachusetts Amherst, Amherst, MA 01003, USA,
{jrothfed,manmatha,trath}@cs.umass.edu *

Abstract. Training and evaluation of techniques for handwriting recognition and retrieval is a challenge given that it is difficult to create large ground-truthed datasets. This is especially true for historical handwritten datasets. In many instances the ground truth has to be created by manually transcribing each word, which is a very labor intensive process. Sometimes transcriptions are available for some manuscripts. These transcriptions were created for other purposes and hence correspondence at the word, line, or sentence level may not be available. To be useful for training and evaluation, a word level correspondence must be available between the segmented handwritten word images and the ASCII transcriptions. Creating this correspondence or alignment is challenging because the segmentation is often errorful and the ASCII transcription may also have errors in it. Very little work has been done on the alignment of handwritten data to transcripts. Here, a novel Hidden Markov Model based automatic alignment algorithm is described and tested. The algorithm produces an average alignment accuracy of about 72.8% when aligning whole pages at a time on a set of 70 pages of the George Washington collection. This outperforms a dynamic time warping alignment algorithm by about 12% previously reported in the literature and tested on the same collection.

1 Introduction

Off-line handwriting recognition and retrieval still remains an unsolved problem in the general case for both modern and historical handwriting. In recent years, there has been some work on large vocabulary datasets on both modern [17, 11] and historical documents [6, 13]. Evaluating handwriting recognition and retrieval techniques on large datasets requires annotated (ie. with ground truth) large vocabulary datasets for training and testing. Creating such large annotated datasets is challenging. For large modern datasets (like the IAM database [10]) this has been achieved by having a number of different people copy out in a specified manner articles that they have been provided. Such restrictions include requiring people to use a ruler while writing and to make sure that each

* J. Rothfeder and T. M. Rath are now at IBM and Google respectively.

line corresponds to a line in the original article. Deriving word by word correspondences from the given line by line correspondences is not that difficult especially for clean modern databases like the IAM dataset [10]. The situation is more challenging with historical handwritten documents. In many situations the only text that is available is the handwritten one and the creation of ground truth requires a labor intensive process of manually transcribing each word. This is for example, how the publicly available George Washington dataset of 20 handwritten pages [6] was produced. However, this is a very time consuming process. Given the repetitive boring nature of the task errors are also produced during the transcription process.

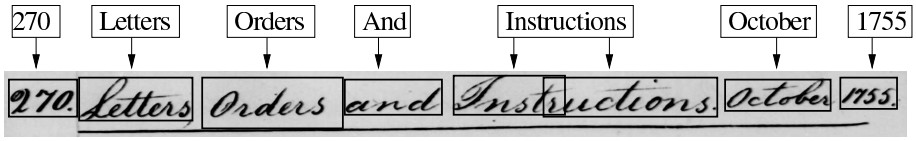
For some historical documents an electronic transcription is sometimes available [1]. These may be scholarly transcriptions made for use in historical studies or other related endeavors. For example, electronic transcripts for a portion of George Washington's papers [2] are available from the Library of Congress. The alignment between the scanned images and the transcriptions is not available. That is we do not know the correspondence between the words or lines in the scanned image and the words or lines in the transcript. In many cases the situation is even worse and we do not accurately know which pages line up. This is because each letter that in Washington's papers is transcribed as a unit. On the other hand in Washington's manuscripts, a letter often ends halfway on a page and a second letter begins right after that [3]. An automatic procedure to align the words on the transcript with the words on the handwritten page would be very useful but is very challenging to do. This may be done for example by automatically segmenting the handwritten words and then trying to find an alignment between the segmented boxes and the words in the transcript. If there were no errors in the segmentation or transcription, a simple linear alignment would suffice. That is, by assigning the first transcript word to the first word-image, the second transcript word to the second word-image and so on. This alignment assumes the start and end points are specified. In practice errors in segmentation, or transcription ensure that this approach will not work. The segmentation errors produced make linear alignment impractical. This will happen with any practical segmentor for even a low rate of segmentation errors throws off a linear alignment and would produce useless training/evaluation data. Another source of error comes from words which are broken up at the end of the line and continued on the next page. A segmentor would treat these as two words while in the transcript they only occur as a single word. Besides segmentation errors, there may also be errors in the transcriptions. These errors may occur because of a mistake on the part of the transcriber (historical documents are sometimes hard to decipher) or for example because the transcriber expanded an abbreviation in the original document.

[1] Printed transcriptions pose an even greater challenge since optical character recognition errors will also have to be taken into account during alignment.

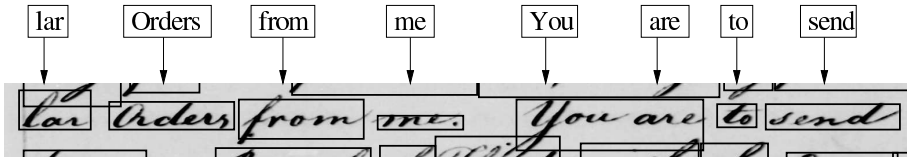
[2] There are actually multiple writers in this collection for George Washington employed secretaries to help him with his work

[3] This is probably because these are copies of the actual letters that were sent out.

Here we propose a new algorithm to align the output of an automatic word segmentor on a handwritten page with a transcript. The data we have consists of a set of pages from George Washington's manuscripts. Each page is automatically segmented using the automatic scale space segmentation algorithm reported in [9, 8]. It is shown in [8] that for segmenting historical documents, this algorithm outperforms a gap metrics based algorithm. We also have a transcript corresponding to each page of the manuscript which has been generated by manually labeling the words. Our goal is to assign one or more words from the transcript to each of our automatically segmented word-images. Fig 1 shows an illustration of alignment for two different lines. Each word image is assigned to one or more transcript words. In the case of oversegmentation (or fragmentation), we wish to assign the same transcript word to all fragments of the word image. In the case of undersegmentation (or multiple words in a box), all corresponding transcript words should be assigned to the bounding box.



(a) Alignment: Notice how we deal with the oversegmented word "Instructions".



(b) Alignment: Notice how we deal with the undersegmented words "You" and "are".

Fig. 1. Two automatically segmented lines and transcript alignments.

We treat the problem as one of aligning two sequences - a sequence of errorful word images and a sequence of words from the transcript. The alignment uses a linear Hidden Markov model and is solved using the Viterbi algorithm to produce the most likely sequence. The HMM models the probability of generating (observing) the word images given the words. The transition model accounts for segmentation errors. The algorithm produces an average alignment accuracy of about 72.8% when aligning whole pages at a time on a set of 70 pages of the George Washington collection. This outperforms a dynamic time warping alignment algorithm (by 12%) previously reported in the literature and tested on the same collection.

Sequence alignment problems have been solved before in a number of language technology areas and bioinformatics. For example, HMM's have been used for sequence alignment in speech recognition [15], the alignment of synthesized speech with speech [7], the alignment of speech recognition output with closed

captions in video [12], machine translation [1] and bioinformatics [2] and for aligning parallel corpora in machine translation [4]. For print OCR [3] created groundtruth by using a machine readable description to print the document and then matching character bounding boxes with bounding boxes derived from a scanned image of the document. [18] aligned an imperfect transcript obtained from a scanned image of a printed page with the characters in unsegmented text.

There has, however, been little work in aligning handwritten text to transcripts. Tomai et al. [16] assume that a line by line correspondence of the transcript and handwritten line is provided and that a word by word alignment is required. They use a handwriting recognizer to produce a ranked list of words from a vocabulary for each recognized word image. Different segmentations are then made of each line and the segmentation that has the highest probability, given the line transcript, is selected. Kornfield et al. [5] consider the problem of alignment when line by line correspondences are available and also only when page by page correspondences are provided. They show that the first case is much easier than the second case. They treat word images and transcripts as two time series and then use dynamic time warping to align them.

The rest of the paper is organized as follows: Section 2 introduces the idea of using an HMM to align text with handwritten documents. Then, section 3 describes the the two components of our observation model. Next, we discuss the transition model in section 4. Datasets are discussed in 5. Experiment results are reported in the next section. Finally we conclude the paper.

2 Using a Hidden Markov Model to Align Text

Let H be a handwritten page. Let $S_1, \ldots, S_x$ be a sequence of random variables corresponding to the word-images from H. Let the transcript of H be of length y, this is a sequence of words corresponding to near-perfect segmentation. Given that we have some knowledge about the types of errors that the automatic segmentor produces, our goal is to assign one or more transcript words to each S_i, thus aligning the transcript to the document. To do this we construct the Hidden Markov Model shown in Fig. 2, where the hidden variables are $S_1, \ldots, S_x$, and the observed variables are the feature vectors, $\mathbf{F}_1, \ldots, \mathbf{F}_x$, extracted from each of the word-images (the features are the same as described in [6]). The full joint distribution for our HMM is given by:

$$P(S_1 = s_1, \ldots, S_n = s_n, \mathbf{F}_1 = \mathbf{f}_1, \ldots, \mathbf{F}_n = \mathbf{f}_n) \tag{1}$$

$$= \prod_{i=1}^{n} P(S_i = s_i | S_{i-1} = s_{i-1}) P(\mathbf{F}_i = \mathbf{f}_i | S_i = s_i) \tag{2}$$

After constructing our HMM in this way, we run the Viterbi algorithm to decode the sequence of assignments to each of our S_i, thus assigning one transcript word to each of the word-images. After this, a postprocessing step may be employed to assign more than one transcript word to some of the word-images

(this is needed in the case of undersegmentation). This postprocessing step is not discussed in this paper.

The next sections describe the two components of our HMM: the observation model and the transition model.

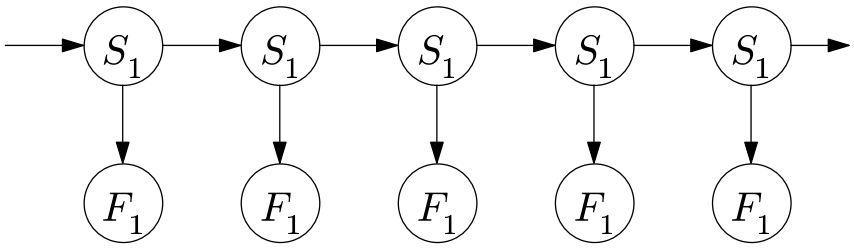


Fig. 2. Graphical model of Hidden Markov Model used for Alignment

3 Observation Model - Feature Likelihoods

Let $\mathbf{f}_i$ be the feature vector corresponding to the the random variable S_i. Our feature vector is 27 dimensional. It is the same set used in Lavrenko et al [6] and consists of scalar features like length of the word or the number of ascenders as well as profile features. Profile features include projection profiles and upper and lower profiles. To obtain a constant length representation, a discrete Fourier Transform is computed over the profiles and only the low order coefficients are used. For more details on the features used see Lavrenko et al [6]. For every vocabulary word w_j in our transcript, we compute the feature-likelihood $P(\mathbf{f}_i|w_j)$, which is a multivariate normal distribution give by:

$$P(\mathbf{f}_i|w_j) = \frac{exp\left\{-\frac{1}{2}(\mathbf{f}_i - \mu_w)^T \Sigma_w^{-1}(\mathbf{f}_i - \mu_w)\right\}}{\sqrt{2^D \pi^D |\Sigma_w|}} \tag{3}$$

with mean μ_w and covariance matrix Σ_w. $|\Sigma_w|$ is the determinant of the covariance matrix, and D represents the number of features extracted for each of the word images, in our case $D = 27$.

The next step is to estimate μ_w and Σ_w. To do this, we use a training set where the word images boxes are manually corrected (see section 5 for more on experimental datasets). We also need a transcript for each document - the words from which form our vocabulary. The transcripts, along with the manually corrected word-images provide us with a set of pairs in the form $\{\text{word-image}, \text{ASCII-annotation}\}$. If we let the length of our feature vectors be k, then μ_w is a vector of length k containing the mean of all of the feature vectors extracted from word images that have been labeled w_i. In other words, for each word w_i in our vocabulary, we extract a set of feature vectors, $\mathbf{g}_{w,1}, \ldots, \mathbf{g}_{w,k}$ from

the training word images that have been labeled with w. Then μ_w is computed as follows:

$$\mu_w[d] = \frac{1}{k}\sum_{i=1}^{k} \mathbf{g}_{w,i}[d], d = 1, \ldots, D \tag{4}$$

where d is a dimension of the feature vector.

The covariance matrix Σ_w can only be estimated accurately for w_j if there is a sufficient number of word-images in our training set which have been annotated with w_j. Unfortunately, this is never the case and we approximate the covariance matrix using one value, $\Sigma_w \approx \sigma_{avg} * I$, for all words. I is the identity matrix and σ_{avg} is the mean feature variance given by the following:

$$\sigma_{avg} = \frac{1}{D}\sum_{d=1}^{D}\left(\frac{1}{N_{tr}-1}\sum_{i=1}^{N_{tr}}(\mathbf{g}_{w,i}[d] - \mu[d])^2\right) \tag{5}$$

where μ is the average of value of all feature vectors in the training set and N_{tr} is the number of all feature vectors, $\mathbf{g}_{w,i}$ in our training set.

Some improvements may be obtained by smoothing the probability estimates especially when the number of training samples is small. A discussion of smoothing for this particular problem requires far more space than is available here and is, therefore, omitted.

4 Transition Model

As mentioned before, errors can be either oversegmentations, undersegmentations, missed word, or extra bounding boxes. If we have an oversegmentation then part of a transcript word will need to be assigned to two or more adjacent word-images. In the case of undersegmentation, two transcript words will need to be assigned to one bounding box. Extra bounding boxes may be dealt with by assigning a transcript word to more than one word image (as in oversegmentation), and missed-words can be dealt with the same way that undersegmentation is. The following describes a transition model designed to support this behavior.

We can use our transition model to account for segmentation errors by assigning positions in our transcripts in the following way (also see Fig. 3 for an illustration):

1. No error: $S_i \leftarrow w_j$ and $S_{i+1} \leftarrow w_{j+1}$
2. Oversegmentation: $S_i \leftarrow w_j$ and $S_{i+1} \leftarrow w_j$
3. Undersegmentation: $S_i \leftarrow w_j$ and $S_{i+1} \leftarrow w_{j+k}, k > 1$

Where S_i is the random variable corresponding to a word image on the page and w_j is the vocabulary word at position j in the transcript.

The setting of parameters for our model is discussed in section 6.2.

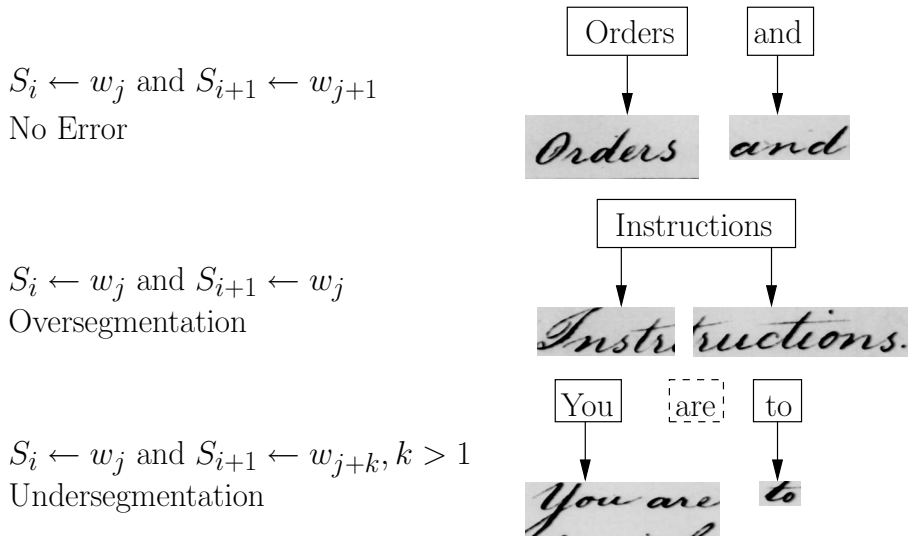


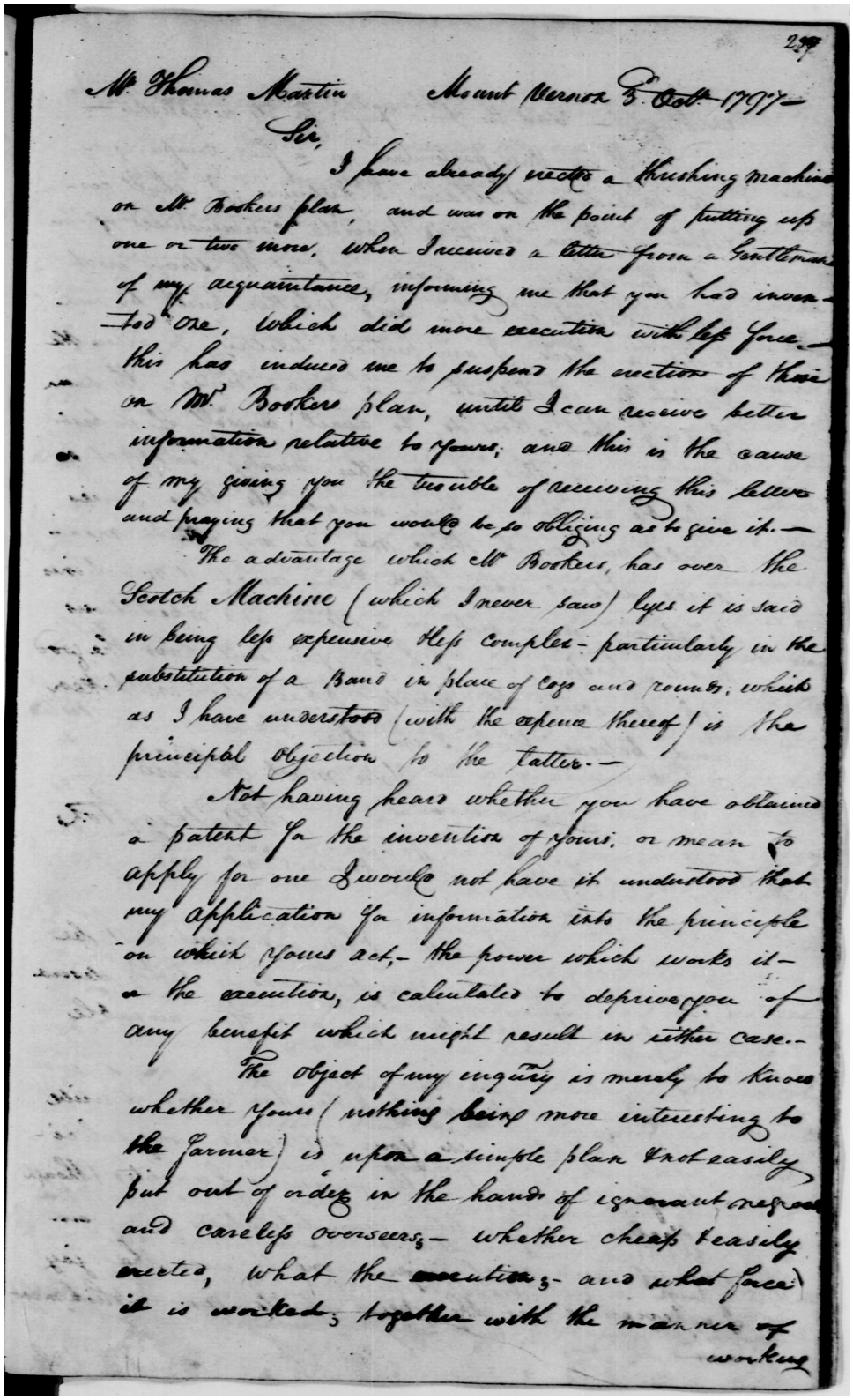
Fig. 3. Different alignment errors and their representations in the transition model. The dangling "are" in the undersegmentation example is not handled by the HMM directly. It could be assigned later using a postprocessing technique not discussed in this paper.

5 Dataset

5.1 Characteristics of the 100 Page George Washington Collection

The 100 Page George Washington Collection (GW100) is a collection of digitized pages, where each page contains one or more letters authored by George Washington and written by a few different secretaries. The pages that comprise this collection were sampled from different portions of the Library of Congress, which contains roughly 140,000 pages scanned at 300 dpi. The letters were scanned from microfilm, and are of varying quality due to the degree of blotches, bleed-through, and faded ink which are present on every page (see Fig. 4 for an example of a document-image).

The data that were used for the experiments comprise a set of word-images that were automatically segmented from each of the pages in GW100. The 100 documents were randomly split into three partitions so that 20 went to a training set, 10 to the validation set, and the remaining 70 to the evaluation set. Transcripts were manually created for each page. In addition, a transcript mapping to word-images (ie. the true alignment) was also created manually using the BoxModify tool [14]. These mappings on the training and validation set were used to estimate parameters. On the test set they were only used for evaluation. These transcripts contain a small number of typographical errors.



Mr. Thomas Martin Mount Vernon 2. Octr. 1797

Sir,

I have already erected a thrashing machine on Mr. Borkers plan, and was on the point of putting up one or two more, when I received a letter from a Gentleman of my acquaintance, informing me that you had invented one, which did more execution with less force — this has induced me to suspend the erection of those on Mr. Borkers plan, until I can receive better information relative to yours; and this is the cause of my giving you the trouble of receiving this letter and praying that you would be so obliging as to give it. —

The advantage which Mr Borkers, has over the Scotch Machine (which I never saw) lyes it is said in being less expensive less complex - particularly in the substitution of a Band in place of Cogs and rounds; which as I have understood (with the expence thereof) is the principal objection to the latter. —

Not having heard whether you have obtained a patent for the invention of yours, or mean to apply for one I would not have it understood that my application for information into the principle on which yours act, - the power which works it - & the execution, is calculated to deprive you of any benefit which might result in either case. -

The object of my enquiry is merely to know whether yours (nothing being more interesting to the farmer) is upon a simple plan & not easily put out of order in the hands of ignorant negroes and careless overseers; - whether cheap & easily erected, what the execution is - and what force it is worked; together with the manner of working

Fig. 4. Image 2360237 from the George Washington Collection. Notice the bleedthrough, faded ink, and blotches.

6 Experiments and Results

We now discuss the experiments and results. First, we discuss the evaluation procedure. This is followed by a brief discussion of how the parameters are estimated. We then evaluate the alignment algorithm on the test set and discuss the results.

6.1 Evaluating Alignment Performance

The goal of the algorithm is to produce an ASCII labeling of word images that can be used as training data for handwriting recognition or handwriting retrieval algorithms, and it is important that our evaluation measure reflects this. For the groundtruth we used a set of automatically segmented word images labeled in the following way:

- Each correctly segmented word image is labeled with the ASCII term corresponding to the word image that is contained within its bounding box.
- Each oversegmented word image is labeled with the ASCII term corresponding to the word image that is contained within the sum of its parts.
- An undersegmented bounding box is labeled with each of the ASCII terms corresponding to the word images that are contained within.

Since the output of our alignment algorithm is a labeling of word images, it is easily compared with the groundtruth. Let $B = \{b_1, \ldots, b_r\}$ be the ASCII words in the labeling of a bounding box and let $G = \{g_1, \ldots, g_s\}$ be the corresponding groundtruth labeling. The score for B is given by the number of matching labels in B and G divided by the greater of $|B|$ or $|G|$. Or, more precisely:

$$\sigma(B, G) = \left(\sum_{i}^{r} t(i) \right) / max(|B|, |G|) \tag{6}$$

where

$$t(i) = \begin{cases} 1 \; \exists j : b_i = g_j \\ 0 \; \text{Otherwise} \end{cases} \tag{7}$$

A score for a set of pages is computed by first assigning a score to each bounding box and then averaging this score for all the bounding boxes in all the pages.

6.2 Model Parameters

The optimal values of our parameters for both the observation model (feature-likelihoods) and the transition model were discovered separately via parameter sweeps.

For the observation model, we performed an exhaustive search using the training and validation set to estimate the value of σ. The optimal value was found to be 0.1.

Transition Model Parameters Normally, the parameters for the transition model in an HMM can not be estimated by a simple exhaustive parameter search because the state space can be quite large. In our case, however, we are assigning transcript words to bounding boxes such that once w_j has been assigned to S_{i-1}, $w_1 \dots w_{j-1}$ are no longer in the state space of S_i. We can further shrink this state space by assuming that undersegmentation errors involving more than three words in one bounding box are extremely rare. Our revised state space for $S_i = \{w_j, w_{j+1}, w_{j+2}, w_{j+3}\}$. This leaves us with only three parameters to estimate.

The mean percentages of oversegmentation, undersegmentation and missed words in the training set are 0.05, 0.06, and 0.03 respectively. Our initial setting for the transition model was based on these values as follows:

- $P(S_i \leftarrow w_j | S_{i-1} \leftarrow w_j) = 0.05$ - Compensates for oversegmentation.
- $P(S_i \leftarrow w_j | S_{i-1} \leftarrow w_{j-1}) = 0.86$ - Corresponds to correct segmentation.
- $P(S_i \leftarrow w_j | S_{i-1} \leftarrow w_{j-2}) = 0.08$ - Compensates for undersegmentation and missed words.
- $P(S_i \leftarrow w_j | S_{i-1} \leftarrow w_{j-3}) = 0.01$ Compensates for undersegmentation and missed words.

However, a parameter sweep using the training and validation set showed that performance over the validation set was optimized by choosing:

- $P(S_i \leftarrow w_j | S_{i-1} \leftarrow w_j) = 0.001$ - Compensates for oversegmentation.
- $P(S_i \leftarrow w_j | S_{i-1} \leftarrow w_{j-1}) = 0.998$ - Corresponds to correct segmentation.
- $P(S_i \leftarrow w_j | S_{i-1} \leftarrow w_{j-2}) = 0.0008$ - Compensates for undersegmentation and missed words.
- $P(S_i \leftarrow w_j | S_{i-1} \leftarrow w_{j-3}) = 0.0002$ Compensates for undersegmentation and missed words.

It was originally expected that the parameters could be estimated directly from the segmentation errors. This parameter sweep shows this not to be the case. The reason for this is due to very large feature likelihood scores being produced by the observation model. Because the features are normalized before computing the parameters of the observation model, σ_{avg} is a very small value. This produces a very narrow Gaussian. Thus, the likelihood score for any two similar feature vectors may differ by a few orders of magnitude and the transition model must compensate for this.

6.3 Results

When the algorithm was run on the test set, the mean score over the 70 pages was found to be 72.8%. The result was compared with that in Kornfield et al. [5]. The experimental dataset is identical to the one used in our experiments. Alignment is performed by treating the bounding boxes and transcripts like two time series and then using dynamic time warping (DTW) to align them.

They evaluated their algorithm by first concatenating the words together that have been assigned to a bounding box, and then computing the Levenshtein distance between the bounding box and the corresponding groundtruth (this measure is close to the measure we use for evaluation). An average score of 60.5% over all bounding boxes was reported when page level alignment was done. Our alignment algorithm, therefore, outperforms their algorithm by about 12%. HMM's incorporate transition probabilities which may explain the better performance.

Kornfield et al. [5] also show that if line break information is available (i.e. line level alignment information is provided) their performance substantially increases to 74.5% (any algorithm will show a substantial improvement in performance when the input is line aligned). Our performance given page alignments is close to their performance using line alignments. The HMM based alignment algorithm presented here would perform even better if line break information were available since the alignment would be much more accurate for shorter sequences. Line break information is rarely available and hence using line break information is not practical for real world problems.

7 Conclusion and Future Work

Aligning transcripts to handwritten data is useful for creating training data. We proposed a new HMM based automatic alignment algorithm for aligning word images and transcripts at page level. This outperformed a previously reported algorithm using dynamic time warping. Future improvements may be obtained by using smoothing for the probability estimates or by using better models. Improvements in segmentation may also improve performance.

8 Acknowledgments

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.

References

1. Y. Deng and W. Byrne. Hmm word and phrase alignment for statistical machine translation. In *Proceedings of HLT-EMNLP*, 2005.
2. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 2001.
3. J. D. Hobby. Matching document images with ground truth. *International Journal on Document Analysis and Recognition*, 1(1):52–61, 1997.
4. M. Kay and M. Roscheisen. Text-translation alignment. *Computational Linguistics*, 19(1):121–142, March 1993.

5. E. M. Kornfield, R. Manmatha, and J. Allan. Text alignment with handwritten documents. In *Proceedings of Document Image Analysis for Libraries (DIAL)*, pages 23–24, 2004.
6. V. Lavrenko, T. M. Rath, and R. Manmatha. Holistic word recognition for handwritten historical documents. In *Proceedings of the Workshop on Document Image Analysis for Libraries DIAL'04*, pages 278–287, 2004.
7. F. Malfrre, O. Deroo, and T. Dutoit. Phonetic alignment: Speech synthesis based vs. hybrid hmm/ann. In *Proceedings of the ICSLP*, pages 1571–1574, 1998.
8. R. Manmatha and J. L. Rothfeder. A scale space approach for automatically segmenting words from historical handwritten documents. *IEEE Transactions on PAMI*, 28(8):1212–1225, August 2005.
9. R. Manmatha and N. Srimal. Scale space technique for word segmentation in handwritten manuscripts. In *Proc. of the Second Int'l Conf. on Scale-Space Theories in Computer Vision*, pages 22–33, Corfu, Greece, September 26-27 1999.
10. U. V. Marti and H. Bunke. A full English sentence database for off-line handwriting recognition. In *Proc. of the 5th Int. Conf. on Document Analysis and Recognition, Gangalore, India*, pages 705–708, 1999.
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):65–90, 2001.
12. P.J.Jang and A. G. Hauptmann. Learning to recognize speech by watching television. *IEEE Intelligent Systems*, 14(5):51–58, 1999.
13. T. M. Rath, V. Lavrenko, and R. Manmatha. A search engine for historical manuscript images. In *Proceedings of ACM SIGIR'04*, pages 369–376, 2004.
14. T. M. Rath, J. L. Rothfeder, and V. B. Lvin. The BoxModify tool, 2004. (Computer program).
15. D. K. Roy and C. Malamud. Speaker identification based text to audio alignment for an audio retrieval system. In *ICASSP '97*, pages 1099–1102, Munich, Germany, 1997.
16. 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*, pages 413–418, Niagara-on-the-Lake, ON, August 6-8 2002.
17. A. Vinciarelli, S. Bengio, and H. Bunke. Offline recognition of unconstrained handwritten texts using hmms and statistical language models. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 26(6):709–720, 2004.
18. Y. Xu and G.Nagy. Prototype extraction and adaptive ocr. *IEEE Trans. PAMI*, 21(12):1280–1296, December 1999.