Using a Hidden-Markov Model in Semi-Automatic Indexing of Historical Handwritten Records

Thomas Packer, Oliver Nina, Ilya Raykhel
Computer Science
Brigham Young University

Abstract

Indexing of historical records is a process that uses human effort to read text images and convert them into a machine readable format that facilitates search. The Church of Jesus Christ of Latter-day Saints has been using volunteers to index millions of microfilm images of genealogy records collected throughout the world. This indexing process is time-consuming. We adapt a technique for holistic handwritten word recognition originally published by Victor Lavrenko et al., and use it to semi-automatically index US census record images to improve the efficiency of the manual indexing process used in the LDS Church's "Internet Indexing" project. The data used for this project differs from Lavrenko's paper in that we recognized words in three columns of a structured census image instead of unconstrained handwritten words. The approach resulted in 90% accuracy for the chosen columns using as little as one page of manually indexed training data.

Introduction

The motivation for the research described in this paper is the belief that we can improve the efficiency of the manual process of indexing historical records, for example the process used in the "Internet Indexing" project organized by the Church of Jesus Christ of Latter-day Saints [9]. The high-level strategy used in this research was to apply existing results of computer science, such as machine vision and machine learning, to demonstrate that it is possible to improve the efficiency of human effort in transcribing such historical documents. The specific technique we used was holistic handwritten word recognition based on a hidden Markov model, as described by a paper by Victor Lavrenko et al [1].

Related Work

This project is directly related to the paper "Holistic Word Recognition for Handwritten Historical Documents" by Lavrenko et al. [1] published in 2004[1]. In this paper by Lavrenko et

[1] The Lavrenko paper should be seen as a starting point in designing the current experiment, not as something we are trying to beat. It should be obvious that our data was specifically chosen to demonstrate how easy it is do get high word classification accuracy on at least certain parts of a census image.

al., unconstrained handwriting is recognized using a hidden Markov model (HMM). The implementing and applying of that HMM is very similar to what we describe below in Methods. It differs in that they had to simplify their observation probability model because they did not have enough training data with respect to the large number of word shapes they were trying to identify. It also differs in that they interpolated their probability model trained on one writer with a probability model trained on another writer to gain the value of additional training data at the expense of that training data not being from the same writer.

The primary way in which their model differs from ours, and a point at which we required some creativity, is the fact that they applied an HMM to natural language sentences. We applied the same kind of HMM to tabular data. This required us to build three observation probability models instead of one (one for each column, instead of a single model for words at any position within a sentence). But this also allowed us to be flexible in ordering these three sets of states within the HMM. In other words, we could re-order the columns any way we wished in an attempt to discover and exploit the varying levels of dependency between different pairs of columns. For example, if the text in the father's birthplace column provides information that could help predict the text in the mother's birthplace column, then it is beneficial to make sure these columns are adjacent to each other in the structure of the HMM. In the end, however, the order did not seem to be significant (at least based on our small sample of test data).

The other work from which we drew knowledge of how to implement our methods include papers providing more detail on how HMMs work [4], how to implement the Viterbi algorithm [7] used in finding the most likely sequence of word classifications for the test data and extracting holistic word features for the observation probability model.

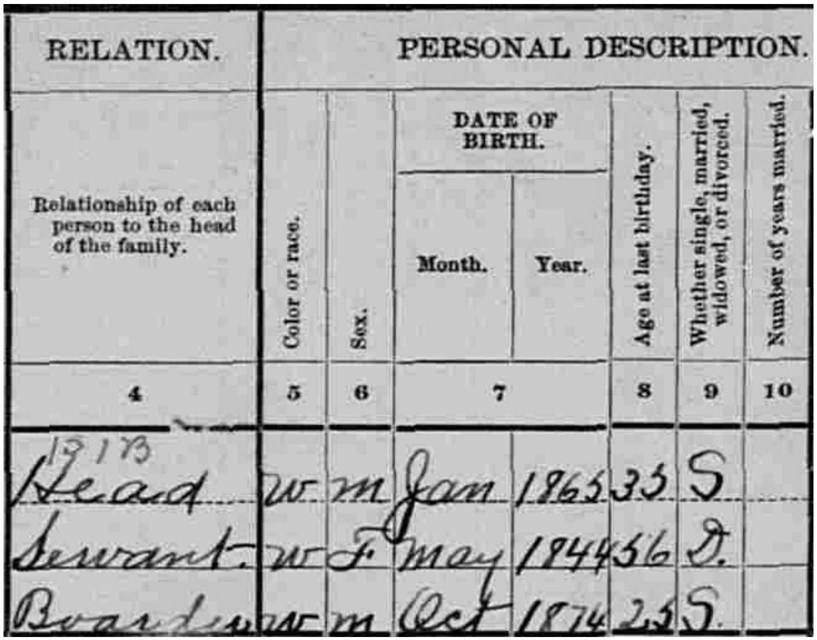
Methods

As with most machine learning projects, the process we followed consisted of three general phases:

1. A human indexer provides the labels (i.e. the classes or meanings) of handwritten words in some collection of documents, called the dataset.
2. Each word image in the training data is reduced to a set of features on which a classification algorithm is trained.
3. The trained algorithm is applied to automatically classify the remaining words in the dataset. The accuracy of this classification process is determined by comparing the automatic labels to correct, hand-made labels.

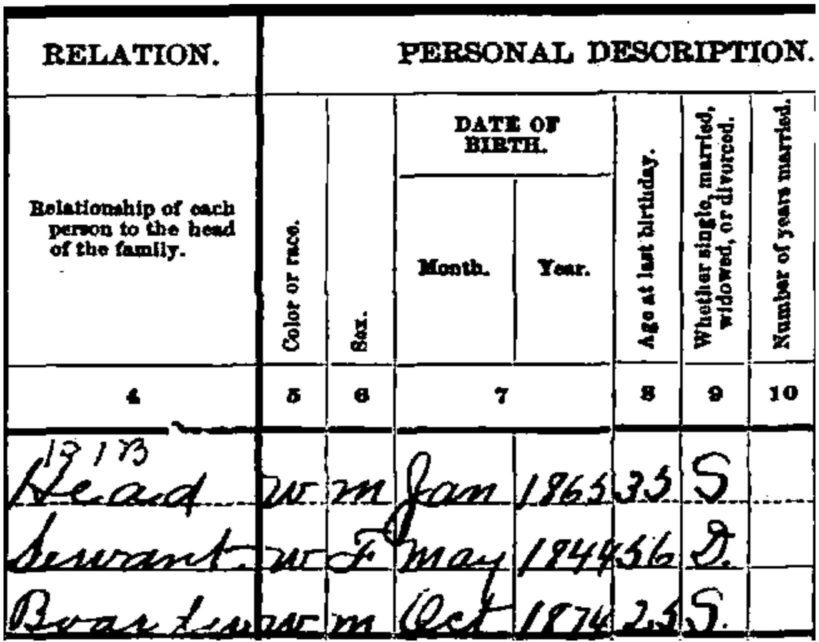
Data

Three digitized images of US Census records were taken from search.labs.familysearch.org consisting of 123 lines. These three images were selected so that the handwriting on all three was done by the same writer. Figure 1 shows a portion of a census image, including the three columns we used (4, 6 and 9).



RELATION.	PERSONAL DESCRIPTION.						
Relationship of each person to the head of the family.	Color or race.	Sex.	DATE OF BIRTH.		Age at last birthday.	Whether single, married, widowed, or divorced.	Number of years married.
			Month.	Year.			
4	5	6	7		8	9	10
19173 Head	w	m	Jan	1865	35	S	
Servant.	w	F	May	1844	56	D.	
Boarder	w	m	Oct	1874	25	S.	

Figure 1: Portion of an original census image.



RELATION.	PERSONAL DESCRIPTION.						
Relationship of each person to the head of the family.	Color or race.	Sex.	DATE OF BIRTH.		Age at last birthday.	Whether single, married, widowed, or divorced.	Number of years married.
			Month.	Year.			
4	5	6	7		8	9	10
19173 Head	w	m	Jan	1865	35	S	
Servant.	w	F	May	1844	56	D.	
Boarder	w	m	Oct	1874	25	S.	

Figure 2: Portion of a thresholded census image.

We used Kittler's algorithm [8] to threshold the image to produce the black and white image from which word rectangles were exrtracted. Figure 2 shows the result of applying Kittler's algorithm to the image in Figure 1.

This step is different from the Lavrenko paper in that they did not use thresholding. Rather, they computed the bounding box and the profile features from the pixel values of the original digitized image. This experiment also differs from Lavrenko’s in that we do no de-skewing and de-sloping. Not doing this kind of pre-processing probably makes it a little harder to recognize words in the Relationship column because, as you can see in Figure 4, tall letters, e.g. “f” partially, hide the profiles of nearby short letters, e.g. “e”.

We selected the following three columns among the 28 available because they had a relatively small number of possible values per column (i.e. a small vocabulary size) and because these values seemed to have high correlation with each other based on manual inspection. For example, the value "wife" in the relationship column is obviously well correlated with the "female" sex and the "married" marital status. Below we list the names, original order and number of values (vocabulary size) found in our training data for the three columns we used.

1.	Relationship to Head	4th column	14 values
2.	Sex	6th column	2 values
3.	Marital Status	9th column	4 values

We semi-automatically extracted word image rectangles in the following manner. First, we cut each of the three fields from each row in the three images using a free image manipulation program (Gimp). These manually selected rectangles were a close approximation to the minimal bounding box for each word that might be produced automatically using computer vision techniques. We pasted these rectangles into another image with a distinct background as shown in Figure 3. Then we automatically extracted these rectangles and read them into memory where they could be associated with the manual transcriptions used for training and testing the classification system. (To fully demonstrate the value of the methods in this paper to the process of indexing, we would need to demonstrate that the whole process of word rectangle extraction can also be done automatically.)

We then manually made transcriptions of all the extracted words, providing these to the training and validation (or testing) algorithms as labels for the training data and validation data, respectively. Then we read in the images and transcriptions into objects in the Java programming language to group them together, in which object we also placed the feature vectors, described below.

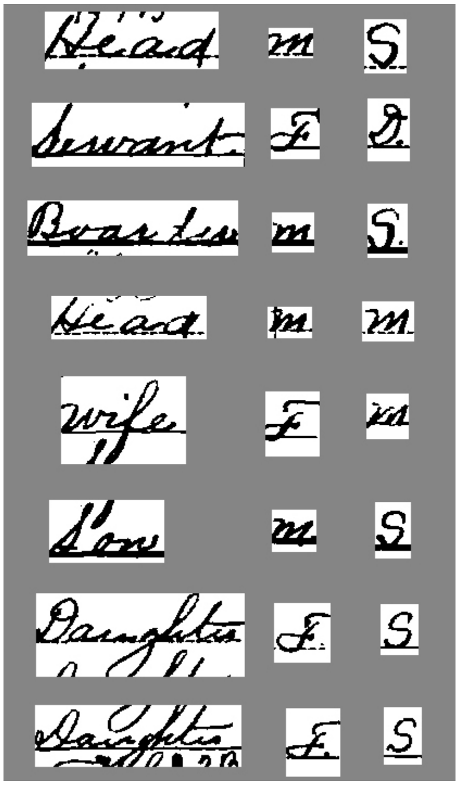

Head m S
Servant F D
Boarder m S
Head m m
wife F m
Son m S
Daughter F S
Daughter F S

Figure 3: Manually extracted bounding boxes.

Observation Probability Model

A hidden Markov model is composed of two kinds of probability distributions. The first distribution estimates the probability of seeing a handwritten word shape given the intended word label. This is called the observation probability and is discussed in this section. The second probability distribution estimates the probability of seeing a word label given its position in a sequence of other word labels and will be discussed in the following section.

In machine learning and statistical modeling, descriptions of events (e.g. handwritten word shape) generally must be reduced to a vector of one or more numerical values. We will refer to these values as features, each of which is some kind of numerical description of the shape of a word. In this research, we computed the following features from the word image rectangles described above. These are the same features used in the Lavrenko paper. Each image rectangle is assumed to be the minimal bounding box of the word shape.

1. **Width** of the word image rectangle.
2. **Height** of the word image rectangle.
3. **Aspect ratio** of the word image rectangle (width / height).
4. **Area** of the word image rectangle (width * height).
5. **Profile features** (21 values):
    1. **Upper** Profile (distance from top of rectangle to first black pixel).
    2. **Lower** Profile (distance from bottom of rectangle to first black pixel).
    3. **Projection** Profile (number of black pixels appearing in each column).

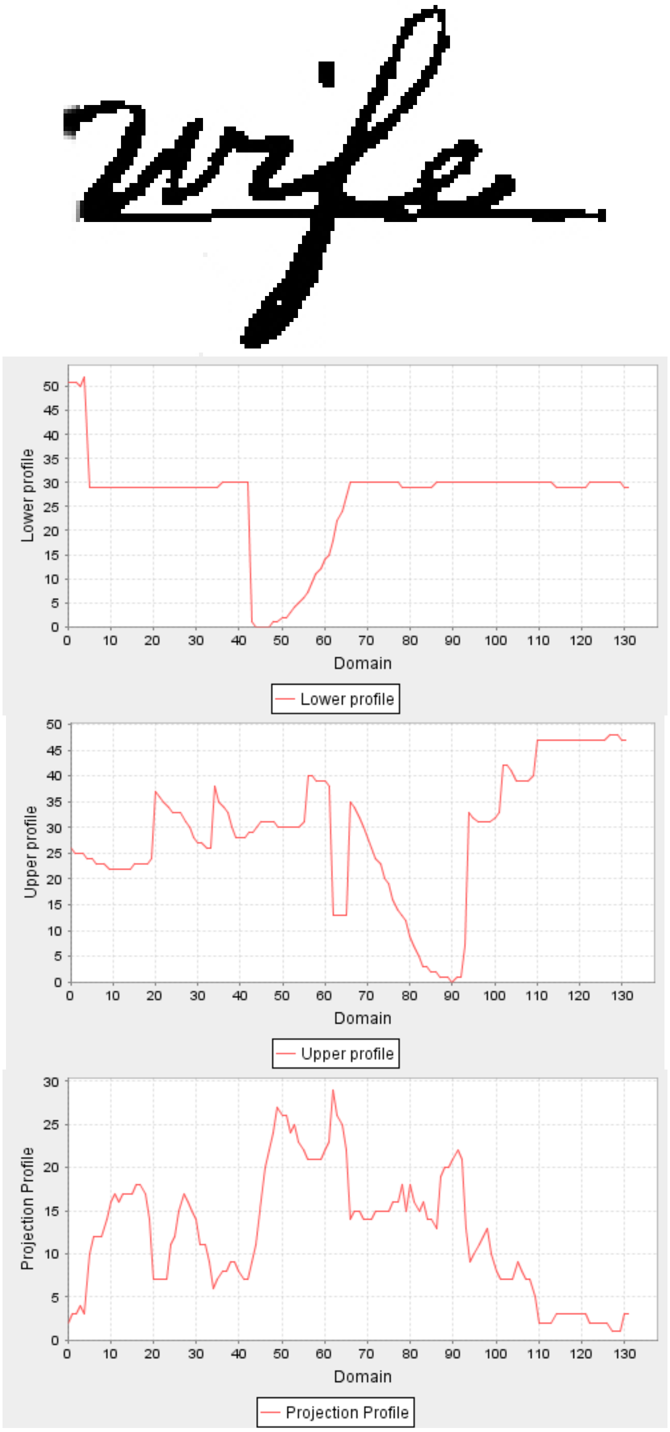


Figure 4: Profiles for the word "wife".

There were 21 scalar features derived from the profiles. We used a Discrete Fourier Transform (DFT) to transfer the profiles to the frequency domain. For each profile we consider the first 4 real and first 3 imaginary components of the DFT frequency description and use those values as scalar features.[2] These scalar features provide a good approximation of the shape of the profile.

All 25 feature values were computed for each word (three words per row in each census image) in our training and test set.[3] Training the observation probability model consisted of computing means and co-variances among these feature values for each word in the training data. During testing (i.e. during automatic classification of each word in the test set), observation probabilities were calculated based on these statistics for those words in the test set, using the following formula for a multi-variate normal distribution:

$$p(\mathbf{x}|\omega_i) \quad = \quad (2\pi)^{-d/2}\,|\mathbf{C}_i|^{-1/2} \quad e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_i)^T\mathbf{C}_i^{-1}(\mathbf{x}-\mathbf{m}_i)}$$

The formula in the Lavrenko paper (above) is identical to the formula we used (below), except for a missing minus sign in the exponent.[4]

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

In each case, the formula gives the probability of a vector of features given the identity of the word (i.e. the word type). $|C_i|$ is the determinant of the covariance matrix used to describe how feature values vary with respect to each other for each particular word type (hence it is subscripted by the word type index), and C_i^{-1} is the inverse of this covariance matrix.

Transition Probability Model

While a minimal machine learning approach that does not consider dependencies between columns might be sufficient to produce reasonable classification accuracies, we wanted to verify that we could demonstrate an improvement in accuracy by exploiting dependencies between the words in different columns of a census image. For example, the row with the word "wife" is more likely to have "f" in the sex field and "m" in marital status field than it is to have “m” and “s”, respectively, regardless of what the word image might look like. This may be scratching the surface of what improvements might be possible through exploiting such implicit dependencies. For example, dependencies also exist between rows in a census image. If the head of a household has "Massachusetts" as his birthplace, all of his children will likely have the same label in the column for "Father's Birthplace".

[2] This was done to be consistent with Lavrenko’s experiments. The reason they used fewer imaginary components is because the first imaginary component will always be zero, and thus provides no information. So to be technically correct, we are using real components 0 through 3 and imaginary components 1 through 3.

[3] There were a total of 25 features per word, which is the sum of 21 profile features and 4 bounding box features.

[4] The authors of the Lavrenko paper acknowledge that their formula contained a typo within email correspondence.

This recognition of inter-cell dependencies suggests possible benefits from building a model that classifies all the words in a row simultaneously based on all the information provided by that row. There are multiple ways to do that. In this research we have chosen the hidden Markov model (HMM) as a reasonable approximation to a joint probability distribution.

HMMs are often used in natural language processing tasks. There are two key differences between a natural language sentence and a row in a census image with respect to an HMM:

1. A sentence has a pre-specified, natural order while the fields in a row is artificially ordered and are somewhat arbitrary. We should select an ordering that is as non-arbitrary as possible.
2. In an English sentence, the set of possible words that might be written at each position is assumed to be the same for all word positions, (though the probability of each word choice will vary, given the type of the previous word). In a census row, the vocabulary for each column can be assumed to be a distinct set of word types.

To address the first difference, we ran our training and testing scenario on every possible ordering of the chosen columns. The results are summarized in the following section.

As one reviewer correctly noted, though we think we can manually pick a column order that would exploit the greatest amount of inter-column dependencies, we decided to try multiple orderings because it's easier to pick an optimal order based on final accuracies rather than based on our intuition of which column pairs should provide the best predictive power for each other. Furthermore, our choice to use hidden Markov model should be justified considering that it does not capture all possible inter-column dependencies simultaneously. This is discussed in our concluding section.

We used a simple bigram word probability model to estimate word probabilities with dependence between columns. While part-of-speech tagging (and other sentence-based NLP processes) use a single bigram probability distribution, as mentioned above, we set up a separate transition probability distribution for each column.

These are the formulas for calculating transition probabilities:

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

Figure 5: Formulas for computing the probability of a word given that it is the first word in sequence (top) and for computing the probability of a word given the previous word in sequence.

In figure 5, T is the training set, $P_T(w|w_0)$ is the state transition probability for the first word in the sequence (given no previous word), and $P_T(w|v)$ is the state transition probability from the word v to the word w.

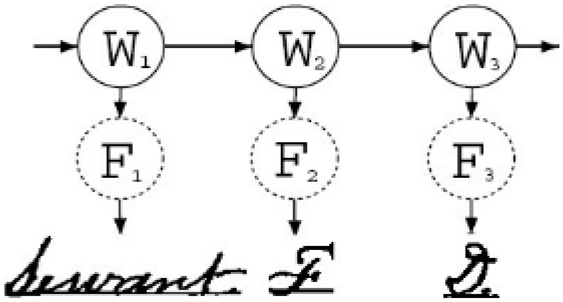


Figure 6: A hidden Markov model is used to calculate the joint probability of a sequence. Observed states are visual word feature vectors (F's); hidden states are word transcriptions or types (W's).

To classify all three words in a row simultaneously, we create a hidden Markov model that models the transitions between columns and the generation of observed word shapes and then apply the Viterbi sequence labeling algorithm [7] to find the classifications that result in the highest joint probability:

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

Figure 7: Joint probability formula for the HMM.

Bringing both probability models together in figure 7, $P(f_j|w_j)$ is the observation probabilities computed using the visual futures and the multi-variate Gaussian distribution, and $P(w_j|w_{j-1})$ is the bigram transition probabilities.

Results

To produce our first graph (Figure 8), we performed 24-fold cross validation, which consisted of 24 separate experiments with 5 rows per test set and 118 disjoint rows per training set. This approach was used to minimize the impact of the small amount of available data on both training and evaluation. Classification accuracies were computed for each experiment as the number of correct word classifications given for each test set divided by the total number of words in the test set. Implicitly, this accuracy is an average over all three columns. These experiment accuracies were then averaged over all 24 experiments and are graphed in Figure 8.

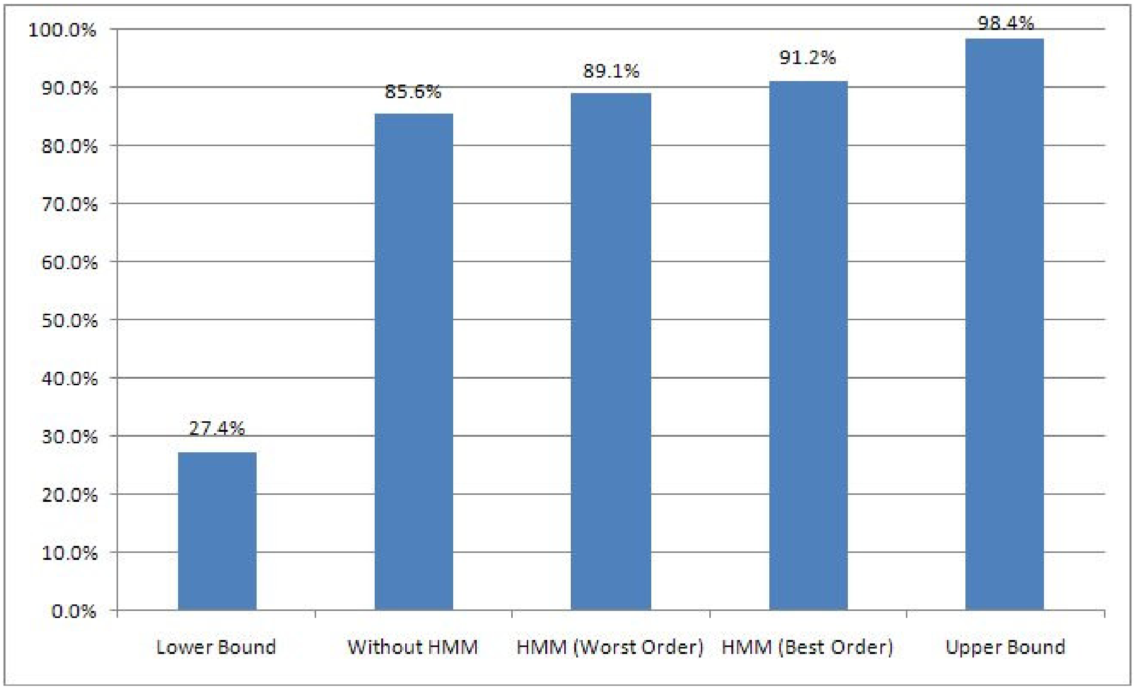


Figure 8: Classification accuracies for different models predicting the values of three fields in a census image.

On the right side, the upper bound (98.4%) was computed by training and testing the classifier using the actual word label as the only feature. It is not 100% because for some of the cross validation folds the test set contained words not present in the training set. The two accuracies of the HMM (Best Order and Worst Order) were 91.2% and 89.1% respectively, and were obtained by applying all the methods described above on two different orderings of columns. When we applied only the observation probability model, the accuracy went down about 4.5% to 85.6%. The lower bound on the left side (27.4%) is the result of randomly selecting a class label for each word.

While just the visual features alone provide respectable accuracy, the addition of a Hidden Markov Model raises it further to a value comparable to that reported in Lavrenko et al. [1] on a comparably-sized vocabulary. The primary result published in Lavrenko et al. [1] was 65%, but it is hard to compare our results to this number because of the vast difference in vocabulary size. Unconstrained handwriting is much harder to classify automatically.

As can be seen, re-running the experiment with different state orders for the HMM produced slightly different levels of accuracy, within a range of about 2%. The best results were achieved for the following order: "Marital Status → Sex → Relationship to Head". However, we are not sure that these differences in accuracy are statistically significant.

Next, we decomposed these classification accuracies and report accuracies for individual columns for the two most interesting cases: “Without HMM” and “HMM (Best Order)”. Figure 9 graphs this comparison.

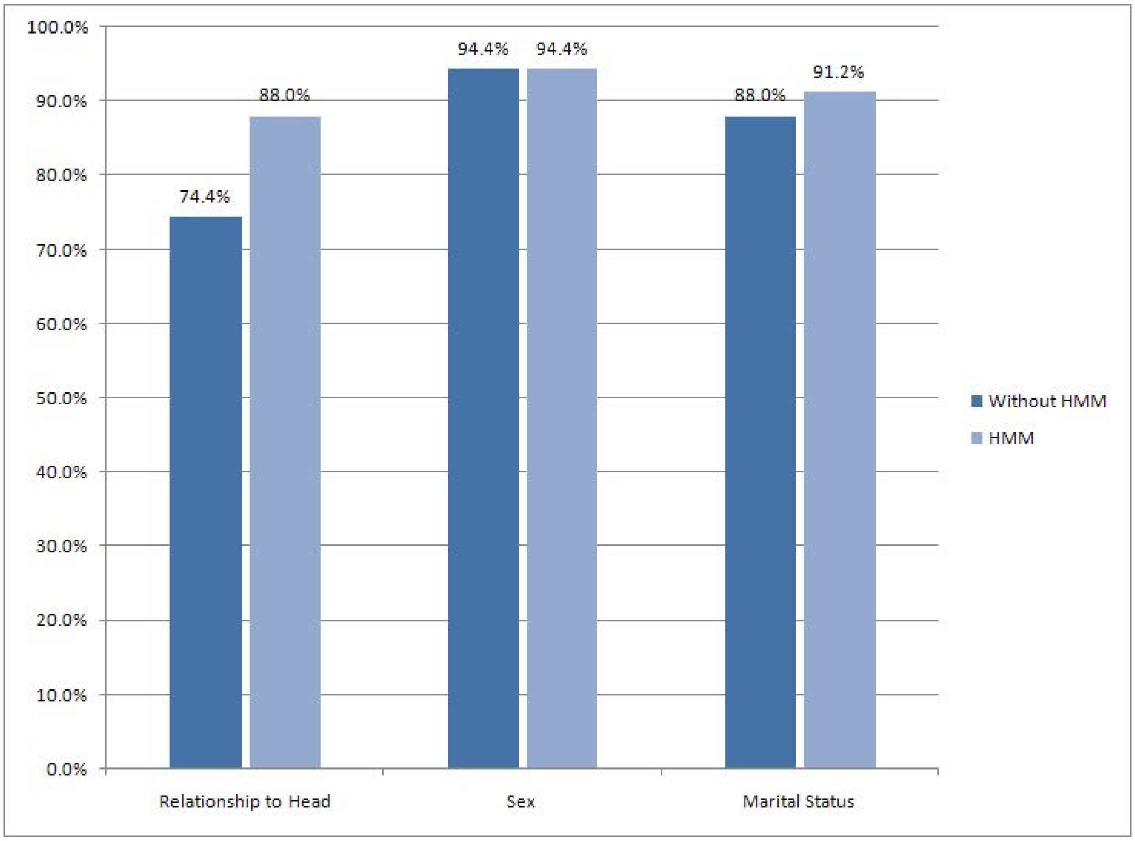


Figure 9: Classification accuracy for each column, with and without the transition probability model (i.e. the full HMM).

It is interesting to note that the easiest column, sex, having just two values, does not improve with the addition of transition probabilities for this particular ordering;[5] but knowing its value does greatly help to classify the Relationship to Head column as seen by the 13.6% boost in accuracy using the transition probability model.

Finally, we are interested to know how much training data is actually necessary to achieve accuracies near 90%. This information would be useful in determining how much more efficient a human indexer might be when working in such a semi-automatic labeling scenario. The following two graphs plot the classification accuracy for all columns (Figure 10) and for the Relationship to Head column alone (Figure 11) with respect to the amount of training data used to train the HMM (Best Order).

[5] This is obviously related to the fact that marital status was the previous word in the model, and the value of marital status does not provide much information about gender. This might not always be true, for example consider a dataset in which there is a larger number of widows: considering demographic statistics on women living longer than men, there might be a correlation between widows and the female gender.

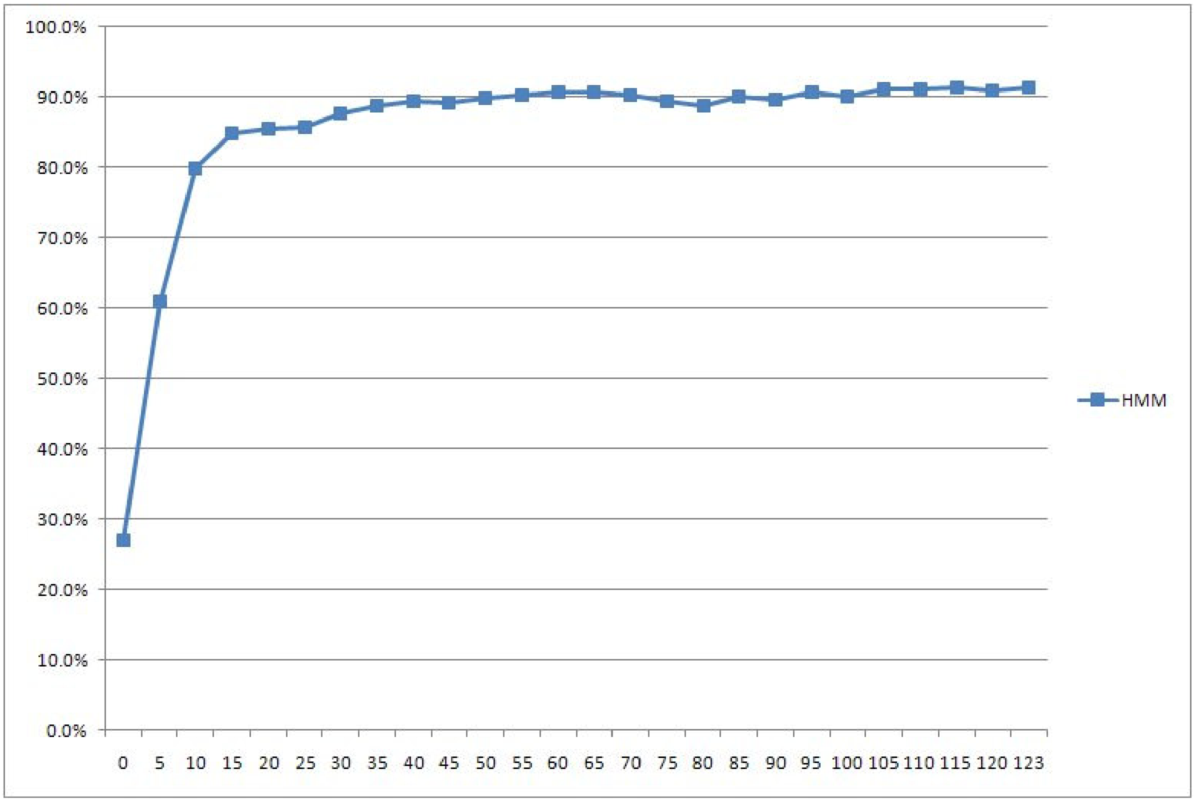


Figure 10: Classification accuracy for the HMM classifier for all three columns, plotted for varying numbers of rows used as training data. Each data point represents an increase of 5 rows of training data from the previous point.

It is interesting to note that for the three chosen columns, we can achieve 80% accuracy using as little as 10 rows of training data, and we achieve about 90% accuracy with as little as one page (about 50 rows) of training data.

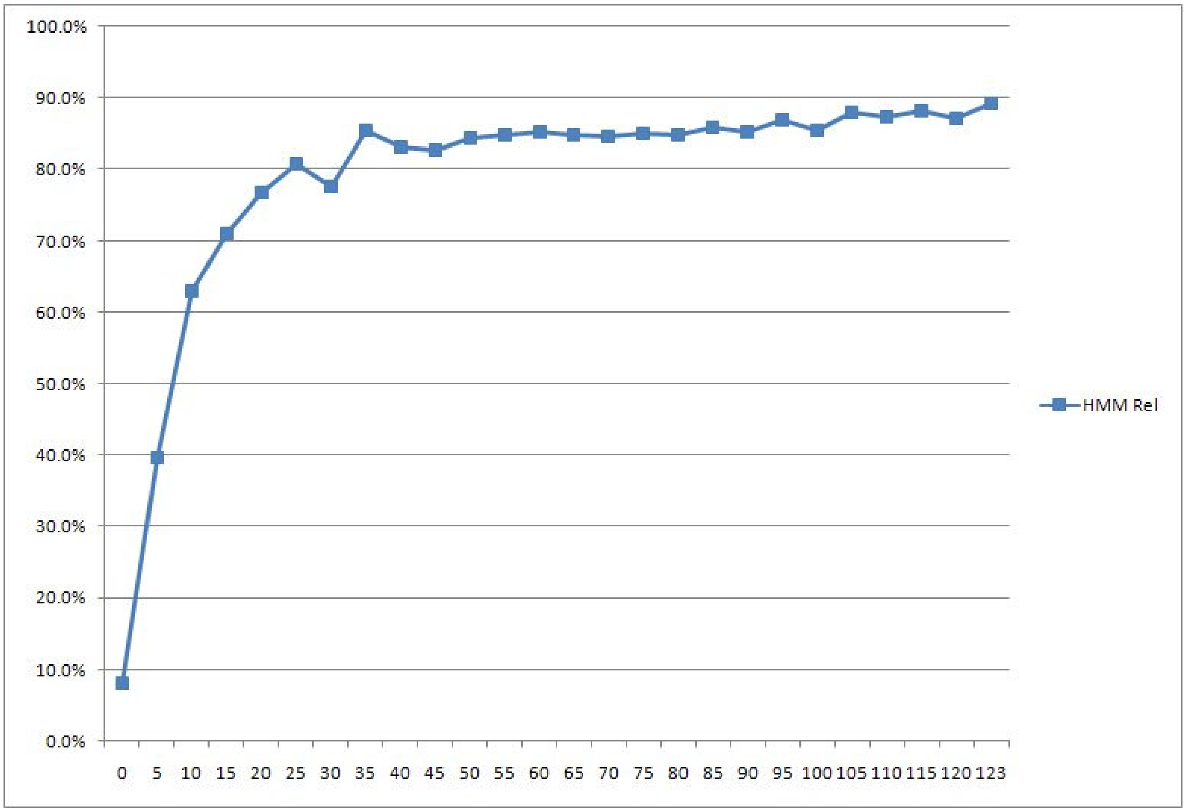


Figure 11: Classification accuracy for the HMM classifier for the Relationship to Head column, plotted for varying numbers of rows used as training data.

Figure 11 shows that the harder column, Relationship to Head, achieves above 80% accuracy with as little as 35 rows of training data. The plot appears to continue trending upward throughout, which suggests that additional training data beyond our three-pages would continue to improve its accuracy.[6]

Conclusion and Future Work

This research indicates that it is possible to apply state-of-the-art holistic word recognition and natural language processing techniques to efficiently index words from tabular, hand-written historical documents. The goal was to demonstrate high enough accuracy in recognizing the words in a US census record that it is possible to make a manual indexing process more efficient. Our achieving word classification accuracies of 90% using 50 rows of training data means that, instead of the human indexer needing to manually transcribe every word in the three chosen columns for an indefinite number of pages, the indexer would only need to label a single page, and then to correct a mere 10% of word labels on the subsequent pages, which labels would be automatically suggested by an HMM-based labeling algorithm.

Future work could be conducted to quantify how much these results actually speed up the

[6] A special thank-you must be given to the anonymous reviewer who suggested we run the additional experiments needed to generate Figures 9 – 11 which we believe greatly increased the value of this study.

indexing process. Possible difficulties encountered in practice may come from the fact that retraining may be necessary for different styles of handwriting. But these difficulties could be mitigated by applying other common machine vision techniques, such as image preprocessing, which were not explored in this research.

We also believe that the results presented here could be of benefit to the semi-automatic indexing other kinds of structured data, not just censuses, including other kinds of tabular data with inter-column (or inter-row) dependencies. While our goal was not to create a working application, the conversion of our tool to such an application should be straightforward, provided that the step of identifying word image rectangles is also automated.

There are a number of improvements that would be beneficial as part of future research and development:

- Automate the word bounding box extraction from the census images such as has been done in [10].
- Apply more preprocessing to input images: de-skew, de-slant, isolate and other image cleaning such as has been done in [1].
- Use more than three columns. We would like to ultimately allow some level of automation for all the columns in a census image needing to be transcribed, including the more difficult name and place columns. While the relationships between other columns is less obvious, inter-row correlations will likely be useful for many of these other columns.
- Use more training data, possibly from different census takers to assure high accuracy for columns with larger vocabularies.
- Use more advanced visual scalar features in the observation probability model.
- Consider relationships between nearby rows.
- Use a joint probability model instead of a hidden Markov model that is not as dependent on column ordering. This would overcome an apparent deficiency in the hidden Markov model in that the HMM only considers correlations among a subset of all pairs of columns. Using a true joint probability model would find correlations among all column pairs simultaneously.
- Alternatively, use tri- or higher order n-grams instead of bigrams in the transition probability model.
- Apply domain knowledge to constrain the model, such as genealogical ontologies used in [11].

Bibliography

[1] V. Lavrenko, T. M. Rath and R. Manmatha. Holistic Word Recognition for Handwritten Historical Documents. *Proceedings of the First International Workshop on Document Image Analysis for Libraries (DIAL'04)*, 2004.

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

[3] U.-V. Marti and H. Bunke. Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwriting Recognition System. *International Journal of Pattern Recognition and Artificial Intelligence.* vol. 15:1, pp. 65-90. 2001

[4] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE* vol. 77:2, pp.257-286. 1989

[5] T. M. Rath and R. Manmatha. Feature for Word Spotting in Historical Manuscripts. *Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 218-222., 2003

[6] T. M. Rath and R. Manmatha. Word Image Matching Using Dynamic Time Warping. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 521-527. 2003

[7] A. J. Viterbi. Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm, *IEEE Transactions on Information Theory*, vol. 13:2, pp. 260–269., 1967

[8] J.V. Kittler et al. Threshold Selection Based on a Simple Image Statistic. *CVGIP(30)*, vol. 2, pp. 125-147. 1985

[9] Internet Indexing Project Home Page.
http://www.familysearch.org/eng/indexing/frameset_indexing.asp

[10] H. Nielson and W. Barrett. Consensus-based table form recognition of low-quality historical documents. International Journal on Document Analysis and Recognition, vol. 8, pp. 182-200. 2006

[11] K. Tubbs and D. W. Embley. Recognizing records from the extracted cells of microfilm tables. Proceedings of the 2002 ACM symposium on Document engineering, pp. 149-156. 2002