BAYESIAN LEARNING OF 2D DOCUMENT LAYOUT MODELS FOR PRESERVATION METADATA EXTRACTION

Song Mao and George R. Thoma
U.S. National Library of Medicine
Bethesda, Maryland 20894, USA
Email: smao,gthoma@mail.nih.gov

ABSTRACT
Digital preservation addresses the storage, maintenance, accessibility, and technical integrity of digital materials over the long term. Preservation metadata is the information required to perform these tasks. Given the volume of these journals and high labor cost of manual metadata entry, automated metadata extraction is necessary. Document layout analysis is a process of partitioning document images into hierarchically structured and labeled homogeneous physical regions. Descriptive metadata such as bibliographic information can then be extracted from these segmented and labeled regions using OCR. While numerous algorithms have been proposed for document layout analysis, most of them require manually specified rules or models. In this paper, we first define the hierarchical 2D layout model of document pages as a set of attributed hidden semi-Markov Models (HSMM). Each attributed HSMM represents the projection profile of the character bounding boxes in a physical region on either the X or Y axis. We then describe a Bayesian-based method to learn 2D layout models from the unstructured and labeled physical regions in a set of training pages. We compare the zoning and labeling performance of the learned HSMM-based model, a learned baseline model, and two rule-based systems on 69 test pages and show that the HSMM-based model has the best overall performance, and comparable or better performance for individual fields.

KEY WORDS
2D Document layout models, preservation metadata extraction, Bayesian learning.

1 Introduction and Prior Work

Compared to paper documents, scanned or online digital documents can be easily maintained, retrieved, and transmitted via Internet. However, if the supporting software and hardware become obsolete, digital documents, unlike paper documents, could be completely lost. It can become a serious problem for digital library systems that store large amount of digital documents. Digital preservation addresses the storage, maintainance, accessibility, and technical integrity of digital materials over the long term. At the National Library of Medicine, we are interested in preserving scanned and online medical journals.

Preservation metadata is a crucial component in any digital preservation system since it saves all the information required for preserving digital documents [1]. While some technical metadata of a scanned document page can be extracted directly from its header, descriptive metadata such as bibliographic information of a scanned journal article is usually not directly available. Layout analysis of scanned document pages is a process of partitioning a document page into hierarchically structured and labeled homogeneous physical regions. Metadata can then be extracted from these segmented and labeled regions using OCR.

While numerous algorithms have been proposed for document layout analysis, most of them rely on manually created rules or models. Spitz [2] described a system for layout recognition based on style related rules. Kim *et al.* [3] use a set of geometric and contextual rules to segment and label document pages. Language models have been used by a few researchers for analyzing the layout of document pages. Kopec and Chou [4] used a probabilistic finite state automaton to represent the layout of Yellow Pages. Krishnamoorthy *et al.* [5] proposed a hierarchical document page segmentation algorithm using a set of block grammars. Mao and Kanungo [6] use a set of stochastic context-free grammars to recognize the layout of bilingual dictionary pages. All the rules or models described above are manually specified, and therefore a new set of rules or models may have to be manually created for a new class of documents.

In this paper, we describe a method in which we learn the layout models from unstructured and logically labeled physical regions in a training set of document page images. The models to be learned are a set of attributed hidden semi-Markov models (HSMMs). Each attributed HSMM represents the character projection profile of physical regions on either the X or Y axis. The whole model represents a hierarchical 2D layout of a class of document pages.

This paper is organized as follows. In Section 2, we briefly define the general form of our layout models. In Section 3, we describe the details of a learning algorithm for inducing attributed hidden semi-Markov models based on Bayesian model merging. In Section 4, the experimental protocol is given. Finally, in Section 5, we present results and a detailed discussion.

2 The Model

Physical regions in a document page appear in spatial order on a page and are separated by white spaces of different sizes. They usually have different widths and heights and contain characters, figures or image components of different sizes and attributes. For example, the title region in the first page of an article usually appears at the top center of the page, has small height, and has characters of large size. It is typically followed by a white space which is in turn followed by an author region that has characters of small size. A large physical region such as a column could consist of several regions such as paragraphs. A good model should be able to represent all such useful information.

We use attributed hidden semi-Markov models (HSMM) to represent document layouts as follows: **1)** each such model represents the projection profile of character bounding boxes of a physical region on either the X or Y axis as determined by a dimensionality attribute. A physical region can either be a white space or text region. Non-text regions such as figures and tables can also be modeled but are not considered in this paper. The use of character bounding box as the basic image unit for document page decomposition was initially proposed by Ha *et al.* [7]. **2)** Each state of an attributed hidden semi-Markov model represents a text region or a white space region (a gap between two text regions or a page margin). A state can also refer to multiple zones if these zones overlap in the projection profile. **3)** The underlying language model of the HSMM, the finite state automaton, is used to model the order of physical regions on the projection profile. **4)** State observations are used to represent observed features from the characters in associated physical region, **5)** state duration is used to model the size (width or height) of the corresponding physical region.

Formally, an attributed hidden semi-Markov model M is a 5-tuple $M = (A_i, B_i, C_i, \pi_i, \rho)$, where A_i is the state transition probability matrix of M, B_i is a state observation probability matrix of M, C_i is the duration (or size) probability matrix of M, π_i is the initial probability vector: the probability that M starts in a given state, and ρ denotes the dimensionality attribute of the model. Our 2D hierarchical layout model consists of a set of such attributed hidden semi-Markov models arranged in a tree grammar framework. We have described the details of the grammar model in [8].

We explain our layout model using the example page shown in Figure 1. The layout model consists of several attributed hidden semi-Markov models each of which represents the character projection profile on either the X or Y axis. Note that both the text and white space regions are represented in the model. Since the models are hidden semi-Markov models, no state has a transition to itself.

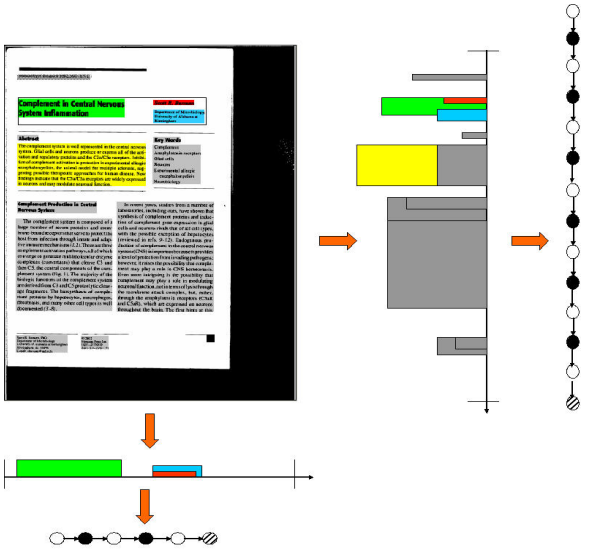


Figure 1. The model at the left is used to model the character projection profile, delineated by zone boundaries, of the whole page on Y axis. The fourth state (consisting of title, author, and affiliation) in the model can be expanded into the model at the bottom since the physical regions represented by the state can be further split along X axis into two columns. Note that the states that contain text are colored in solid black. The terminating states are shown as shaded circles.

3 The Learning Algorithm

In the last section, we described the general form of the layout model of document pages, i.e., a set of attributed HSMMs. The exact topology and parameters of the layout model for different document classes are usually different. For a given document class, our goal is to learn the exact topology and parameters of the layout model from unstructured and labeled physical regions in a set of training document pages. We pose the learning problem as a search or optimization problem in a Bayesian framework.

Stolcke and Omohundro [9] used a similar approach for inducing hidden Markov models (HMMs) to analyze speech signals. Their work is based on the approach taken by Thomason and Granum [10] in which HMM models are induced by maximizing some likelihood criteria. In our approach, state duration distributions are considered during the learning process. This is a significant improvement over HMMs where model state duration distributions are inherently geometric distributions, which in most cases is not true for the sizes of document physical regions. In our approach, instead of 1D string model, a 2D layout model is learned by recursively applying the learning algorithm on both X and Y projection profiles of character bounding boxes.

3.1 The Recursive Bayesian Learning Algorithm

We assume that the rectangular bounding boxes of zones and characters of a set of training document pages are given. Note that the given bounding box and logical labels of physical regions do not have to be perfect as long as error occurs with relatively small probability. We can obtain an initial projection profile of the bounding boxes of characters, which are delineated by zone boundaries, on either the X or Y axis. We assign states to physical regions in the projection profile as described in Section 2. Figure 1 shows how to get a string of states from the projection profile of a document page image.

To obtain an initial model from a set of input strings of states, we can construct an attributed hidden semi-Markov model which produces exactly these strings. The initial states of the model correspond to the first state in each input string and each string is represented by a unique path in the model. The probability of an initial state represents the relative frequency of the corresponding string among all strings. Within each path, a state has a unique transition to the next state with probability 1. To compute the observation distribution of each state in an attributed hidden semi-Markov model, the physical region represented by the attributed hidden semi-Markov model is first partitioned into a sequence of thin strips of same size perpendicular to the axis determined by the dimensionality attribute. A count of character bounding boxes in each strip is considered as an observation of the state. A empirical distribution of such counts is then considered as the observation distribution of the state. The number of strips in each state is considered a duration (or size) observation of the state with probability one. Since the probability of each path in the model is equal to the relative frequency of the corresponding string, the model is a maximum likelihood model, i.e. $\arg\max_M P(X|M)$, where X denotes some input data.

We now want to merge the states of the model such that a Bayesian posterior probability function is maximized as follows:

$$\begin{aligned} M^* &= \arg\max_M P(M|X) = \arg\max_M \frac{P(M)P(X|M)}{P(X)} \\ &= \arg\max_M P(M)P(X|M). \end{aligned} \quad (1)$$

The online version of a best first merging algorithm with look ahead [9] is used to search for an optimal model. We constrain the search not to have loops, not to merge a text state with a gap or margin state, and not to merge two states that have very different spatial order. The logical label counts of each state are retained during the merging process. The logical label of each state in the final model is determined by the logical label that has the most count. While this algorithm can be used to learn the attributed hidden semi-Markov models from 1D input strings, we can learn a 2D model by applying it recursively to the X or Y projection profiles of physical regions of document pages.

Let $\mathcal{M}$ be the 2D model to be learned and let I be a set of training document page images. Let $D_s(I)$ be a set of physical regions in I that have the state label s. Let $modelMerging(D_s(I))$ be the merging algorithm operated on $D_s(I)$ and return an attributed hidden semi-Markov model. Let $D_0(I)$ be a set of full page domain of I. Let $RDV(D_s(I), \mathcal{M})$ be a duration Viterbi algorithm [11] that is used to recursively segmenting a document page using the attributed hidden semi-Markov model in $\mathcal{M}$. It returns a set of domain sets $\mathcal{D}(I)$ that can be further split at another direction. Set $\mathcal{M} = \phi$ where ϕ is the empty set. The recursive learning algorithm $RecursiveLearning(D_s(I), \mathcal{M})$ perform the following steps:

1. $M = modelMerging(D_0(I)), \mathcal{M} = \mathcal{M} \cup \{M\}$.
2. $\mathcal{D} = RDV(D_0(I), \mathcal{M})$.
3. For each domain set $D_s(I)$ in $\mathcal{D}$, loop:
   (a) If $D_s(I)$ can be further split at another dimension, call $RecursiveLearning(D_s(I), \mathcal{M})$.
   (b) Else, continue.

In the following subsections, we will discuss the detailed descriptions of priors and likelihood functions used in the optimization process as shown in Equation 1.

3.2 Priors for Hidden Semi-Markov Models

From the learning point of view, an attributed hidden semi-Markov model M can be decomposed into three components as $M = (M_g, M_t, \theta_M)$ where M_g is the general model form, M_t is the exact model topology given M_g, and θ_M is the parameters of M given M_t and M_g. The prior of M can therefore be written as

$$\begin{aligned} P(M) &= P(M_g, M_t, \theta_M) \\ &= P(M_g) \cdot P(M_t|M_g) \cdot P(\theta_M|M_t, M_g). \end{aligned}$$

Stolcke and Omohundro [9] proposed state-based priors for M_t and θ_M in a HMM as follows:

$$P(M) = P(M_g) \prod_{q \in Q} P(M_t^{(q)}|M_g) P(\theta_M^{(q)}|M_g, M_t^{(q)}).$$

where Q is the state vocabulary. The Dirichlet prior is used to compute $P(\theta_M^{(q)}|M_g, M_t^{(q)})$ in [9] and a state-based simple prior is used to compute $P(M_t^{(q)}|M_g)$. Since the state duration is explicitly represented in the hidden semi-Markov models, we can compute $P(\theta_M^{(q)}|M_g, M_t^{(q)})$ and $P(M_t^{(q)}|M_g)$ as follows:

$$\begin{aligned} P(\theta_M^{(q)}|M_g, M_t^{(q)}) &= \frac{1}{B(\alpha_t, \ldots, \alpha_t)} \sum_{i=1}^{n_t^{(q)}} \theta_{q_i}^{\alpha_t - 1} \cdot \\ & \quad \frac{1}{B(\alpha_e, \ldots, \alpha_e)} \sum_{j=1}^{n_e^{(q)}} \theta_{q_j}^{\alpha_e - 1} \cdot \end{aligned}$$

$$\frac{1}{B(\alpha_d,\ldots,\alpha_d)}\sum_{k=1}^{n_d^{(q)}}\theta_{q_k}^{\alpha_d-1},$$

$$P(M_t^{(q)}|M_g) = p_t^{n_t^{(q)}}(1-p_t)^{|Q|-n_t^{(q)}}\cdot p_e^{n_e^{(q)}}(1-p_e)^{|Q|-n_e^{(q)}}\cdot p_d(1-p_d)^{V^{(q)}}.$$

where the quantities with subscript t, e, b represent the contributions of state transition, state observation, and state duration parameters, respectively. We compute $V^{(q)}$ as

$$V^{(q)} = \frac{d_{max}^{(q)} - d_{min}^{(q)}}{d_{max}^{(q)}}N. \quad (2)$$

where $d_{max}^{(q)}$ and $d_{min}^{(q)}$ are the largest and smallest duration symbols for state q, respectively, and N is the normalization factor of the state duration. We set $pd = 0.8$ in our experiments. The state duration prior in Equation 2 is selected to penalize the merging of two states that has very different sizes. This is based on our observations that document regions with similar logical labels tend to have similar sizes.

3.3 Likelihood Functions

In order to compute the posterior probability, we need to compute the likelihood function, i.e., the probability of observing the data given a model. The likelihood function can be expressed as

$$P(X|M) = \int_{\theta_M} P(\theta_M|M)P(X|M,\theta_M)d\theta_M.$$

where $P(X|M)$ can be approximate by the sum of the probability of the Viterbi path for each element in X [9]. Since the Dirichlet prior is used for $P(\theta_M|M)$, Equation 3 can be rewritten as

$$P(X|M) \approx \prod_{q\in Q}\int_{\theta_M^{(q)}} P(\theta_M^{(q)}|M)P(v^{(q)}|M,\theta_M^{(q)})d\theta_M^{(q)}, = \prod_{q\in Q}\frac{B(v_1^{(q)}+\alpha_1^{(q)},\ldots,v_n^{(q)}+\alpha_n^{(q)})}{B(\alpha_1^{(q)},\ldots,\alpha_n^{(q)})}.$$

where vs are Viterbi counts and αs are parameter prior weights.

4 Experimental Protocol

We first learn the layout models from the title pages of 19 articles of a medical journal. We then use the learned models to segment and label title, author, affiliation, and abstract fields in the title pages of another 69 articles of the same journal. The test dataset consists of 198 title textlines, 181 author textlines, 600 affiliation textlines, and 2079 abstract textlines.

The width of strips used for partitioning a physical region is 12 pixels. The number of character bounding box counts are uniformly quantized into 1 to 10 levels. Each quantization level represents 5 bounding boxes. Any count greater than 50 will be quantized to 10. The normalization factor for state duration is set to 100 strips.

We also learn a baseline 2D layout model that is similar to our 2D model except that it is based on hidden Markov models. From now on, we call our 2D model the HSMM-based model and the baseline model the HMM-based model. The training dataset and the values of common parameters of the recursive learning algorithm are the same for learning both models. We use the learned baseline model by replacing the duration Viterbi algorithm with the Viterbi algorithm in the recursive learning algorithm as described in Section 3.1. We then compare the segmentation and labeling performance using both models and two rule-based systems [3, 12] developed previously.

5 Experimental Results and Discussion

We report the experimental results in two steps: the model learning results and segmentation and labeling results using the learned models.

5.1 The Learning Results

The learning algorithm automatically induces three attributed model components (HSMMs) from a set of training document pages as shown in Figure 2. The model component (b) represents the projection profile of the characters in the whole page on the X axis. It consists of the states representing the left margin, left column, column gap, and right column of the page. The left column state L and right column state R in (b) are expanded into the model components shown in (c) and (d), respectively. They represent the projection profile of the characters in the left column and right column on the Y axis. The solid black states represent text regions and white states represent white spaces. The shaded states represent the terminating state. In the model component shown in (c), state 2 represents the title field, state 4 represents the author field, state 6 and 8 represent the affiliation field, and state 16 and 10 represent logical zones of other type. In the model component shown in (d), state 4, 22, and 26 represent the abstract field, other text states represent logical zones of other type.

Since the given bounding boxes and logical labels of physical regions are not perfect and observation of a state may not be the optimal representation of the state, some states are incorrectly merged. For example in (c), state 1 has an incorrect transition to state 4. But those transitions usually have very small probabilities and are ignored in the recognition algorithms. Table 1 shows some statistics of the learning results.

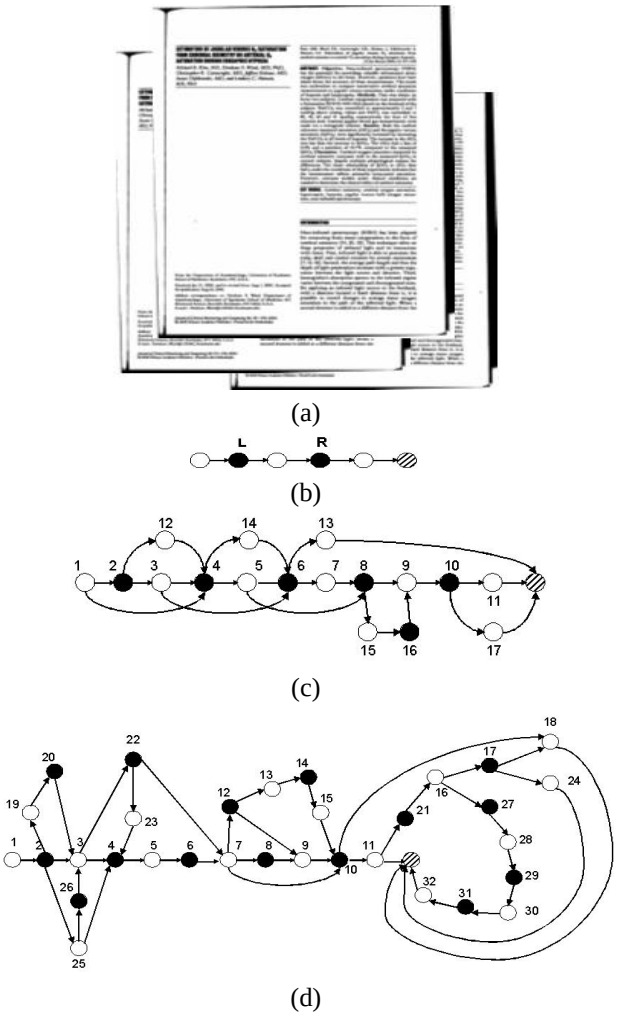


Figure 2. A sample document page used in the experiments is shown in (a). The learned model components that representing the projection profile of the characters in the whole page on the X axis (b), the left column on the Y axis (c), and the right column on the Y axis (d).

5.2 Results and Summary

We report segmentation and labeling accuracy for each of the title, author, affiliation, and abstract fields using the HSMM- and, HMM-based models, and two rule-based methods. We also report an overall accuracy that is the ratio of the total number of correctly located and labeled title, author, affiliation, and abstract textlines to the total number of title, author, affiliation, and abstract textlines. Figure 3 shows a graph representation of the performance results.

Figure 4 shows the segmentation and labeling result of a sample test page using HSMM-based layout model and the HMM-based layout model. The logical labels of the physical regions are signified with the thickness of their rectangular bounding boxes. From the thinnest to the thickest bounding boxes, the represented logical labels are title, author, affiliation, and abstract.

Table 1. This table reports number of the initial states, number of the states in the final model, and batch size.

model component	Number of Initial states	Number of Final states	Batch size Batch size
1	95	HSMM: 5, HMM: 5	5
2	189	HSMM: 17, HMM: 28	5
3	205	HSMM: 32, HMM: 29	5

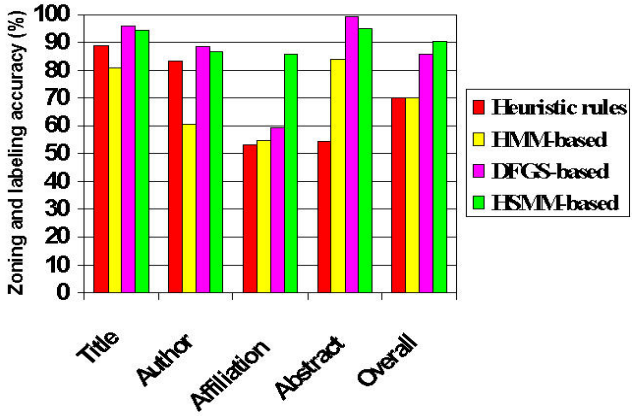


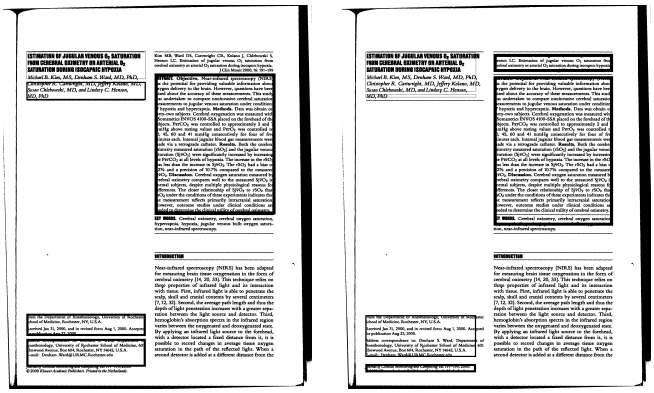
Figure 3. Segmentation and labeling accuracy using the HSMM-based model, the HMM-based model, heuristic rules, and rules inferred in the Dynamic Feature Generating System (DFGS). Note that in the heuristic-rule-based and DFGS-based system 1) more features are used, and 2) prior physical segmentation is required.

We can see that the HSMM-based model significantly outperforms the HMM-based model. This is mainly due to the fact that the duration of states in our model are explicitly represented and learned from training dataset. We can also see that the HSMM-based model has comparable zoning and labeling performance as the DFGS-based system for title, author, and abstract fields. But it has significantly better performance for affiliation field since it was able to merge the over-segmented affiliation zones in the training document pages.

In summary, we have described a recursive learning algorithm for learning a 2D layout model from unstructured and labeled physical regions in a set of training document pages. Experimental results show that the HSMM-based layout model significantly outperforms a learned baseline model (HMM-based layout model). We will consider more features such as font size, font attribute, and key word as state observations in our learning algorithm. We will also test our algorithms on classes of documents with different layout styles.

References

[1] *Building a National Strategy for Preservation: Issues*



(a) (b)

Figure 4. Segmentation and labeling errors in an example document using the HSMM-based model (a) and using the HMM-based model (b). Note that the only error in (a) is that an author textline is labeled as title. However in (b), all the author textlines are labeled as title, all the affiliation textlines are labeled as author, the footer textlines are recognized as affiliation, and textlines immediately above and below abstract field are also labeled as abstract.

in Digital Media Archiving, Council on Library and Information Resources and the Library of Congress, Washington DC, 2002.

[2] A. L. Spitz, "Style directed document recognition," in *Proceedings of International Conference on Document Analysis and Recognition*, Saint-Malo, France, September 1991, pp. 611–619.

[3] J. Kim, D. X. Le, and G. R. Thoma, "Automated labeling in document images," in *Proceedings of SPIE Conference on Document Recognition*, San Jose, CA, January 2001, pp. 111–117.

[4] G. E. Kopec and P. A. Chou, "Document image decoding using Markov source models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 602–617, 1994.

[5] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan, "Syntactic segmentation and labeling of digitized pages from technical journals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 737–747, 1993.

[6] S. Mao and T. Kanungo, "Stochastic language models for automatic acquisition of lexicons from printed bilingual dictionaries," in *Document Layout Interpretation and Its Applications*, Seattle, WA, September 2001.

[7] J. Ha, R. Haralick, and I. Phillips, "Document page decomposition by the bounding-box projection technique," in *Proceedings of International Conference on Document Analysis and Recognition*, Montreal, Canada, August 1995, pp. 1119–1122.

[8] S. Mao, A. Rosenfeld, and T. Kanungo, "Stochastic attributed k-d tree modeling of technical paper title pages," in *Proceedings of IEEE Conference on Image Processing*, Barcelona, Spain, September 2003.

[9] A. Stolcke and S. M. Omohundro, "Best-first model merging for hidden markov model induction," Tech. Rep. TR-94-003, International Computer Science Institute, Berkeley, CA, January 1994.

[10] M. G. Thomason and E. Granum, "Dynamic programming inference of markov networks from finite sets of sample strings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 491–501, 1986.

[11] L. R. Rabiner, B. H. Juang, S. E. Levinson, and M. M. Sondhi, "Recognition of isolated digits using hidden Markov models with continuous mixture densities," *AT&T Technical Journal*, vol. 64, pp. 1211–1234, 1985.

[12] S. Mao, J. Kim, and G. R. Thoma, "A dynamic feature generation system for automated metadata extraction in preservation of digital materials," in *The First International Workshop on Document Image Analysis for Libraries*, Palo Alto, CA, January 2004, pp. 225–232.