Handwritten Text Recognition for Bengali

Joan Andreu Sánchez
PRHLT Research Center
Universitat Politècnica de València
València 46022 (Spain)
jandreu@prhlt.upv.es

Umapada Pal
CVPR Unit
Indian Statistical Institute (India)
umapada@isical.ac.in

***Abstract*—Handwritten text recognition of Bengali is a difficult task because of complex character shapes due to the presence of modified/compound characters as well as zone-wise writing styles of different individuals. Most of the research published so far on Bengali handwriting recognition deals with either isolated character recognition or isolated word recognition, and just a few papers have researched on recognition of continuous handwritten Bengali. In this paper we present a research on continuous handwritten Bengali. We follow a classical line-based recognition approach with a system based on hidden Markov models and *n*-gram language models. These models are trained with automatic methods from annotated data. We research both on the maximum likelihood approach and the minimum error phone approach for training the optical models. We also research on the use of word-based language models and character-based language models. This last approach allow us to deal with the out-of-vocabulary word problem in the test when the training set is of limited size. From the experiments we obtained encouraging results.**

***Keywords*-Bengali, Handwritten Text Recognition, HMM, *n*-grams, MPE training.**

I. Introduction

Handwritten text recognition of Bengali is a problem that has started recently to be researched [1]. Recognition of handwritten Bengali is difficult problems because it has many different like-characters symbols. Most of the existing research papers in Bengali are performed on segmenting the characters from words and then recognizing them isolatedly. A number of papers have focused on typeset text [2], while other papers have been performed for character level segmentation and recognition in Bengali [3]. It is worth noting that due to the presence of touching symbols, the segmentation of characters from a word is not feasible, and even the words are usually difficult to segment. Overlapping and touching characters frequently occur in Bengali and create more hindrance in segmenting characters from the words and words from other words. Therefore, holistic approaches that recognize characters and words as a whole are necessary.

In the past years stochastic approaches like hidden Markov models (HMMs) have been widely applied to perform text recognition tasks but only a few pieces of work using HMMs are performed in Bengali handwritten word recognition [4] and almost all these methods consider the recognition as word wise HMM model creation [5]. Character-based HMM models for optical modeling have been successfully used for recognition of continous handwritten English/Latin scripts [6]. One of the advantages is that they allow recognizing characters and words as a whole, without any previous segmentation. HMMs avoid the problem of pre-segmentation of words into characters so the errors of pre-segmentation can be eliminated.

Though, character-based HMM models as optical models are popular in the literature of text recognition, the process may not be directly useful in Indian scripts, especially in Bengali because of its complex properties. Thus, sufficient data for each combination will be necessary for training the respective class models. Recently, Roy et al.[7] used character-based HMM for Bengali text recognition. The authors proposed the zone segmentation approach to reduce the number of classes in HMM for word recognition and thus the performance was improved.

Different technologies exist for recognition of continous handwritten text. Current state-of-the-art technology is based on NN-based techniques [8]. However, classic techniques based on HMM and *n*-gram language models are a competitive alternative as it has been recently confirmed [9]. In addition, NN-based systems currently are not able to include the power of language models in a NN-based decoding system, and consequently, for difficult tasks the word error rate my be higher than HMM/*n*-gram based systems.

In this paper we explore the use of holistic approaches to recognize handwritten Bengali text. The proposed system is based on HMM and *n*-gram language models. Given the large amount of characters that can exist in Bengali, it is necessary to take profit as much as possible of the training data. Therefore, in this paper we explore several ways of doing this, both in the training process and in the decoding process.

First, we research on the use of discriminative techniques for training the optical models (OM). Since the number of characters is large, it is necessary to discriminate as much as possible between the classes. Concretely we research on the use of the minimum phone error (MPE) criterion for training the OM. Second, we research on the use of word-based language models (LM) and character-based LM for recognizing Bengali. The later method allows the system to deal with the out-of-vocabulary problem at character level. Third, we explore the MPE criterion both at word level and at character level. Our hypothesis is that the MPE training at character level is a better method since it allows the training process to discriminate between classes at character level in spite of discriminating at word level, where the discriminating classes may be not related since they are in the context of a word.

This paper is organized as follows: first, some characteristics of the Bengali language are described. Then, the main characteristics of the Bengali HTR system are describes. The dataset, the results and the discussion are described in Section IV. Finally, some conclusion and future work are described in Section V.

II. Properties of Bengali

Bengali, the second most popular language in India and the fifth most popular language in the world, is an ancient Indo-Aryans language. More than 200 million people in the eastern part of Indian subcontinent speak in this language. Bengali script alphabet is used in texts of Bengali, Assamese and Manipuri languages. Also, Bengali is the national language of Bengalidesh.

The alphabet of the modern Bengali script consists of 11 vowels and 39 consonants. These characters are called as *basic characters*. Basic characters of printed characters are shown in Fig. 1. Writing style in Bengali is from left to right and the concept of upper/lower case is absent in this script. It can be seen that most of the characters of Bengali have a horizontal line at the upper part of the character that is called *matra*.

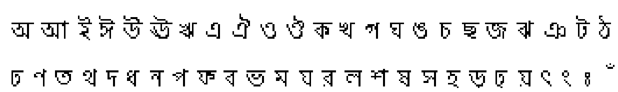


Figure 1. Examples of Bengali basic characters.

In Bengali script a vowel following a consonant takes a modified shape. Depending on the vowel, its modified shape is placed at the left, right, both left and right, or bottom of the consonant. These modified shapes are called *modified characters*.

A consonant or a vowel following a consonant sometimes takes a compound orthographic shape which is called *compound character*. Compound characters can be combinations of two consonants as well as a consonant and a vowel. Compounding of three or four characters also exists in Bengali. There are more than 200 compound characters in Bengali [7]. Examples of some Bengali compound character formation are shown in Fig. 2.

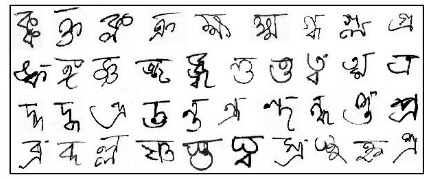


Figure 2. Examples of some Bengali popular handwritten compound characters.

A Bengali text word (or line) can be horizontally partitioned into three zones. The upper-zone denotes the portion above the matra, the middle zone covers the portion between matra and baseline, the lower-zone is the portion below baseline. Different zones in a Bengali text line are shown in Fig. 3.

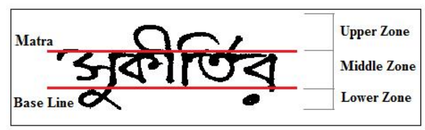


Figure 3. Three zones of Bengali script – upper, middle and lower zone separated by matra and base line.

Recognition of Bengali text is not similar to Latin text due to the variation of character-modifiers presence in 3 zones: upper, middle and lower zones as well as presence of complex shaped compound characters. When a consonant character, say “ক” (appears only in middle zone) gets combined with a vowel, the vowel forms a modifier which can appear in either in middle zone (like “কা”), or middle and upper zone (like “কি”) according to the nature of vowel. Hence, the combinations of consonants and vowel make a large number of possible character combinations and as a result recognition of Bengali text is a challenging task.

III. HTR for Bengali

The HTR system used in this paper followed the classical architecture with a line image feature extraction and hidden Markov model (HMM) and language model (LM) training/decoding [10]. A document image preprocessing step on the full page was not necessary since the images used in the experiments were in binary and fairly clean, the line images were manually annotated and automatically extracted, and the the errors were manually corrected. Since our HTR system was based on HMMs, each pre-processed line image was represented as a sequence of feature vectors (see Fig. 5).

Given a handwritten line image represented by a feature vector sequence, $\mathbf{x} = x_1\ x_2\ \dots\ x_m$, the HTR problem can be formulated as the problem of finding a most likely word sequence, $\mathbf{w} = w_1\ w_2\ \ldots\ w_l$, i.e., $\mathbf{w} = \arg\max_{\mathbf{w}} P(\mathbf{w} \mid \mathbf{x})$. Using the Bayes' rule we can decompose this probability into two probabilities, $P(\mathbf{x} \mid \mathbf{w})$ and $P(\mathbf{w})$, representing optical-lexical knowledge and syntactic knowledge (LM), respectively:

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} P(\mathbf{w} \mid \mathbf{x}) = \arg\max_{\mathbf{w}} P(\mathbf{x} \mid \mathbf{w}) P(\mathbf{w}) \quad (1)$$

$P(\mathbf{x} \mid \mathbf{w})$ is typically approximated by concatenated character models, like HMMs [11] in this paper, while $P(\mathbf{w})$ is approximated by a LM, like n-grams [11]. The LM can be a word-based LM or a character-based LM. This approach has been successfully used for many languages like Latin languages [9], and Arabic [12] just to mention a few. In this regard, HTR for Bengali is a less researched problem. We followed this approach for a Bengali dataset as we describe in the following section.

The search (or decoding) of $\hat{\mathbf{w}}$ is optimally carried out by using the Viterbi algorithm [11]. The training of the OM is usually carried out by defining a merit function. The usual function is the likelihood of the sample, and in such case, the forward-backward algorithm is used for obtaining the maximum-likelihood estimation [11]. An alternative merit function that can be used is based on the maximal mutual information [13], [14], that for M observation sequences $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ is the defined as:

$$\mathcal{F}(\theta) = \sum_{i=1}^{M} \log \frac{p_\lambda(\mathbf{x}_i | M_{w_r})\ P(w_r)}{\sum_{\hat{w}} p_\lambda(\mathbf{x}_i | M_{\hat{w}})\ P(\hat{w})} \quad (2)$$

where M_w is the composite model corresponding to the word sequence w and w_r represents the sentence reference. Note that the numerator in (2) is the likelihood merit function. The denominator in (2) adds up all word sequences in the task. This value is approximated with the most probable word sequences that accounts for the observation $\mathbf{x}_i$ that, in turn, is computed with a word lattice. The denominator represents the competitor transcripts with regard to the reference transcript in the numerator [14]. The lattice that is used in the denominator is obtained from a lower order n-gram for obtaining "confusable" hypotheses. Note that if a word-based n-gram LM is used, then the competitor characters in the same time frame may be useless since they have to appear in the context of word. Alternatively, the lattice in the denominator can be computed with a character-based LM. In this way the competitor models may appear in the same context. We explored this hypothesis in the experiments. Another alternative merit function that can be used is the phone error that is defined in terms of (2). This optimization criterion is called minimum phone error (MPE). We used MPE as training criterion in the experiments.

IV. Experiments and Results

A Bengali dataset was used for this research that consisted of 98 page images written by several writers. Fig. 4 shows some examples of this dataset. The images

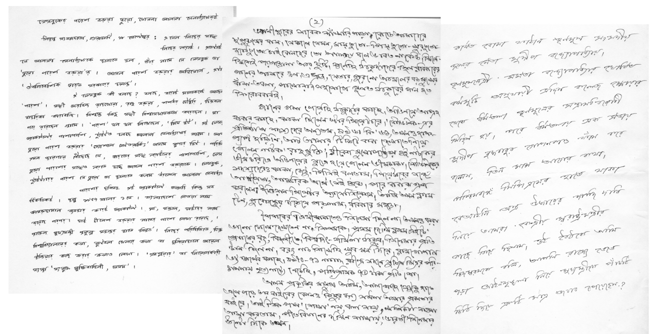


Figure 4. Example of same pages from the dataset.

were in black and white. This dataset was divided in several partitions for performing cross-validation experiments. Table I shows some statistics about the partitions. It is remarkable the large amount of running out-of-vocabulary (OOV) words that was 30.1% in average. Note that these OOV words are sure errors in a word-based open vocabulary decoding process. Usually the following word after an OOV word also involves an error and this means that about 60% word error rate (WER) was expected in a word-based open vocabulary decoding process. This OOV problem introduces a new problem that sometimes is overcome with character-based LM.

Table I
Basic statistics of the different partitions in the Bengali dataset.

Number of:	P0	P1	P2	P3	P4	Total
Pages	20	20	20	20	18	98
Lines	465	435	435	249	201	1,785
Run. words	4,407	3,750	3,623	2,206	1,708	15,694
Run. OOV	1,503	947	1,477	667	201	-
Lex. OOV	1,175	750	1,254	502	169	-
Lexicon	1,953	1,439	1,970	931	408	4,962
Char. set size	217	199	214	184	121	291
Run. Characters	5,148	5,506	5,446	5,063	5,242	42,712

We performed cross-validation experiments with the partitions previously described. We used the feature extraction method described in [15]. This feature extraction method was based on moments computed from a sliding window from left to right on the line image. Some context is additionally taken and the context is stacked. Fig. 5 shows a Bengali line and a grey-level representation of their corresponding feature vectors. Note that no special treatment was performed on the modifiers that may be placed out of the main body of the line (e.g., ◌ৃ or ◌ঁ). This problem would need a special treatment, since the OM should be able to cope with a bidimensional relation among characters. The problem of recognizing these characters was left to the lexical and

LM models in this research.

A baseline HTR system was trained with HTK. Characters that were composed by two strokes like ো were modeled with two HMM, but the information was retained in the lexical model for reversing the operation and to left just one character in the hypothesis produced by the recognition system.

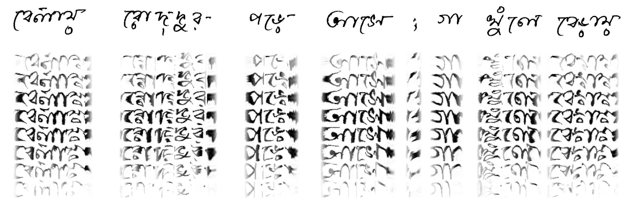


Figure 5. Example of a Bengali line on the top and the representation of the feature vectors obtained.

Two approaches were researched for language modelling: the first approach was based on word-based LM, and the second approach was based on character-based LM. In addition, we researched on two criteria for the training of the OM, namely, ML training and MPE training.

A. Word-based LM and ML training

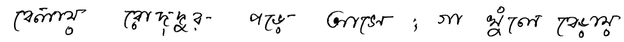
A word-based 2-gram LM was trained with the training dataset described in Table I, a different LM for each partition. The OM were trained by using the likelihood as a merit function. Table II shows in the "Baseline" row the WER and the Character Error Rate (CER) obtained in this first experiment.

Table II
RESULTS OBTAINED IN THE EXPERIMENTS IN ALL PARTITIONS (ALL PARTS. COLUMN) AND ONLY IN PARTITION P0.

	All Parts.		P0	
	WER	CER	WER	CER
Baseline	63.4	39.3	67.9	42.8
+ Gen. HMM	62.7	38.5	66.7	40.7
+ CV	49.7	31.8	52.2	32.8
+ WL-MPE	-		50.4	29.8
+ CL-MPE-1	-		43.8	25.8
+ CL-MPE-2	-		43.0	25.4

Note that 63.4% WER was close to the expected 60% that was mentioned previously when the OOV word problem was described. The first line of Fig. 6 shows a handwritten Bengali sentence and the second line shows the reference sentence. The third line shows the sentence hypothesized by the system.

We observed in this first experiment that many characters appeared just a few times in the dataset. Thus, 88 characters out of approximately 290 appeared 3 times or less in the dataset (see Table III for some of these low frequent characters). Therefore the HMM for these characters were not well trained. Note in third line in Fig. 6 that the low frequent character দ্ was not recognized in the second word.



বেলায় রোদ্দুর পড়ে আসে ; গা খুলে বেড়ায়
বেলায় রোদে তার পরে আসে ; গা বিকেল বেলায়
বেলগাছে রোদ্দুর পড়ে আসে ; গা বিকেল বেলায়

Figure 6. Example of Bengali sentence and recognition results.

In order to deal with this problem we substituted all the characters with low frequency (those that appeared 3 times or less) by a generic HMM. The goal was that the lexical and the LM dealt with this problem. We performed the same recognition experiment and the obtained results are shown in second row ("+ Gen. HMM" row) in Table II.

Table III
LOW FREQUENT BENGALI CHARACTERS IN THE DATASET. LEFT COLUMN IS THE FREQUENCY.

3	ল্ড	র্ভ	র্গ	স্প্র	জ্ব	প্ল	ঞ্চ্য	ন্দ্য	থ্য	ঙ্খ	...
2	ঢ়	স্ন	স্ট্য	ষ্প	শ্ল	র্ল	র্দ্ধ	র্চ	স্ল	ম্দ	...
1	ড়্য	দ্	স্ব্য	স্ত	স্ঠ	স্ক্র	স্-	স্	শ্ট	শ্ছ	...

Some improvement was obtained both in the WER and in the CER in this experiment. Line four in Fig. 6 shows that this time the low frequent character দ্ was recognized in the second word.

As mentioned previously, many errors were due to the running OOV words. This problem can be alleviated with a better LM and/or a good lexicon. A better LM can be obtained with more training data. Therefore, we evaluated the contribution of a good lexicon, and for this purpose, a closed vocabulary experiment was performed. This experiment was the same as an open vocabulary experiment, but the words in the test set were added as an additional lexicon. Note that the LM was the same as in the previous experiments, except for the new lexicon. Table II shows in "+ CV" row the obtained results. The WER was this time 49.7% that represented 13 absolute points less than the result in the "+ Gen. HMM" row. The CER was 31.8% that was worst than an experiment reported in [1]. In [1], results are reported with semi-ortho syllables on a larger database, and the results were about 25% error rate at semi-ortho syllable level.

B. Word-based LM and MPE training

As mentioned in the previous section, MPE criterion is an alternative merit function for training the OM. The goal is to learn the system to discriminate among different OM. This means that the system needs to see in the same time frame the correct character and the confusable character. Fig. 7 illustrates this situation with an example. The red (or bold) path in the lattice (that is the correct transcript) is used in the numerator of expression (2), while the whole lattice (including the red path) is used in the denominator. It is worth noting that expected values are used in the computation of (2), and therefore the contribution of the best path is not equally weighted in the numerator and the denominator (see [14] for additional information). Those characters that appear in the same time frame compete each other. Only some paths are plotted, but the lattice may include thousands of these paths depending on the parameters for generating the lattice. The more paths the lattice includes, the better to distinguish between characters.

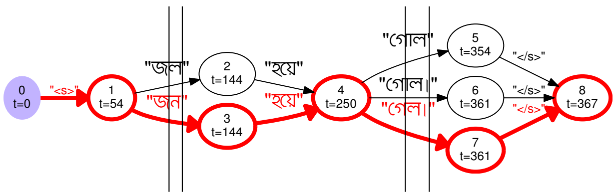


Figure 7. Example showing the information used for training discriminatively. Each node includes the time information. Vertical lines delimit horizontal position where the lattice has some confusing characters.

We used HTK for training the OM with this training criterion. Row "+WL-MPE" shows the results obtained in this experiment, this time just for partition P0[1]. As we described in Section III, the denominator in (2) is obtained from a lower order n-gram for getting "confusable" hypotheses. In the "+WL-MPE", we used a word-based 1-gram. Note that the WER decreased from 52.2% to 50.2%.

The hypotheses represented in denominator of (2) have to represent the most confusable hypotheses with regard to the reference transcript. Note that if the lattice is composed from a word-based LM, then the competitors character have to appear in the context of a word. Therefore, we hypothesized that a character-based lattice could include more confusable character. Fig. 8 illustrates this situation.

We tested this hypothesis with lattices obtained with an 1-gram character-based LM and with a 2-gram character-based LM. Rows "+CL-MPE-1" and "+CL-MPE-2" in Table II represent respectively the obtained results. We can observe that the results were clearly better with an absolute decreasing of 9.2 points in WER with regard to the "+CV" row.

C. Character-based LM and ML training

Character-based LM can be used for dealing with OOV words, since the basic units to be recognized are characters. A character-based LM was trained with the training dataset. The OM were trained with HTK using the likelihood as a merit function. Note that these OM were the same used in Section IV-A. The only difference of this experiment with regard to the experiments in Section IV-A is that we used a character-based LM. This time the decoding process was performed with the iAtros system[2], our local decoder. We tested several values of n for the n-grams. The idea of having a large n is to have the same context as a 2-gram word-based LM and to compare the results. Table IV shows the results obtained for different values of n. Note that small changes were obtained in the WER and CER when n was 7 with regard to $n = 6$. Note that the CER was 30.4% and the WER was 61.5%. Although the CER was better with the close vocabulary experiment with the word-based LM (32.8%), the WER was clearly worst (52.2% in the word-based LM and 61.5% in the character-based LM). This is one of problems when a character-based LM is used, and only when the CER is really small the text is readable, namely, the obtained result are not necessarily well-formed words.

Table IV
RESULTS OBTAINED IN THE EXPERIMENTS WITH CHARACTER-BASED LM IN PARTITION P0.

	P0	
	WER	CER
1-gram	83.6	40.5
2-gram	78.8	38.9
3-gram	69.8	34.1
4-gram	64.8	32.5
5-gram	63.0	31.7
6-gram	61.7	30.5
7-gram	61.5	30.4

D. Character-based LM and MPE training

We explored also the use of the MPE criterion as merit function in the training of the OM using character-based LM. We tested with a character-based 1-gram LM and 2-gram character-based LM for computing the denominator in (2). These OP were the same used in '+CL-MPE-1" and "+CL-MPE-2" in Table II. For computing (2), we used lattices like those that can be seen in Fig. 8. Table V shows the results obtained in this experiment.

Table V
RESULTS OBTAINED IN THE EXPERIMENTS WITH CHARACTER-BASED LM AND MPE TRAINING IN PARTITION P0.

	P0	
	WER	CER
7-gram	61.5	30.4
+ CL-MPE-1	58.8	28.7
+ CL-MPE-2	58.0	28.0

Note that the CER decreased more 2.0 absolute points, while the WER was still very high. This confirms

[1] These experiments were not performed for all partitions because they required additional time, and we decided to perform experiment in depth with one partition before going in breadth with all partitions. These experiments are left for future work.

[2] https://www.prhlt.upv.es/page/projects/multimodal/idoc/iatros/download.php

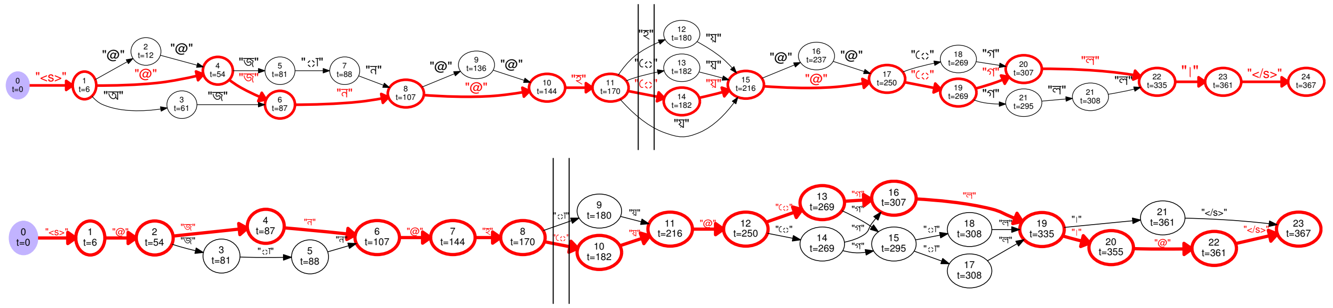


Figure 8. Example showing the information used for training discriminatively. The lattice was obtained from a character-based LM. A 1-gram character-based LM was used in the upper graph and a 2-gram LM was used in the lower graph. This graphs is just for illustration because the real graphs tend to have thousands of paths.

that character-based LM can be a good alternative for dealing with the OOV problem. However, there is still room for improvement for the WER.

V. Conclusion

This paper researched on the use of HTR for Bengali. This language is characterized by the large number of characters. We investigated both the use of word-based LM and character-based LM. The latter option allowed us to obtain competitive results at character level, but the WER was not as good as the WER obtained with word-based LM. We investigated also the use of the MPE training criterion for training the OM. This MPE training was based on lattices. The results obtained when the OM were trained with this criterion were better than the results obtained when the ML criterion was used. For future work we intend to research more in depth the MPE training criterion, since the process for obtaining the lattice can be exploited for guaranteeing that more confusable character are included in the lattice. In this way, the OM can be able to better discriminate between the correct and the confusion classes.

Acknowledgment

This work has been partially supported through the European Union's H2020 grant READ (Recognition and Enrichment of Archival Documents) (Ref: 674943) and partially supported by MINECO/FEDER, UE under project TIN2015-70924-C2-1-R.

References

[1] U. Garain, L. Mioulet, B. Chaudhuri, C. Chatelain, and T. Paquet, "Unconstrained bengali handwriting recognition with recurrent models," in *ICDAR*, 2015, pp. 1056–1060.

[2] P. Natarajan, E. MacRostie, and M. Decerbo, *The BBN Byblos Hindi OCR System*. London: Springer London, 2010, pp. 173–180.

[3] U. Pal, R. Roy, and F. Kimura, "Multi-lingual city name recognition for indian postal automation," in *13th International Conference on Frontiers in Handwriting Recognition*, 2012, pp. 169–173.

[4] T. Bhowmik and U. R. andS. K. Parui, "Lexicon reduction technique for Bangla handwritten word recognition," in *Document Analysis Systems*, 2012, pp. 195–199.

[5] S. Belhe, C. Paulzagade, A. Deshmukh, S. Jetley, and K. Mehrotra, "Hindi handwritten word recognition using hmm and symbol tree," in *Proceeding of the Workshop on Document Analysis and Recognition*, ser. DAR '12, 2012, pp. 9–14.

[6] A. Toselli and E. Vidal, "Handwritten text recognition results on the bentham collection with improved classical n-gram-hmm methods," in *Proceedings of the 3rd International Workshop on Historical Document Imaging and Processing*, 2015, pp. 15–22.

[7] P. Roy, P. Dey, S. Roy, U. Pal, and F. Kimura, "A novel approach of bangla handwritten text recognition using HMM," in *14th International Conference on Frontiers in Handwriting Recognition*, 2014, pp. 661–666.

[8] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. on PAMI*, vol. 31, no. 5, pp. 855–868, 2009.

[9] A. Toselli and E. Vidal, "Handwritten text recognition results on the Bentham collection with improved classical n-gram-HMM methods," in *International Workshop on Historical Document Imaging and Processing (HIP)*, August 2015.

[10] A. H. Toselli, A. Juan, D. Keysers, J. González, I. Salvador, H. Ney, E. Vidal, and F. Casacuberta, "Integrated Handwriting Recognition and Interpretation using Finite-State Models," *IJPRAI*, vol. 18, no. 4, pp. 519–539, June 2004.

[11] F. Jelinek, *Statistical Methods for Speech Recognition*. MIT Press, 1998.

[12] M. Parvez and S. Mahmoud, "Off-line arabic handwritten text recognition: a survey," *ACM Computing Surveys*, vol. 45, no. 2, pp. 1–35, 2013.

[13] P. Gopalakrishnan, D. Kanevsky, A. Nadas, D. NahamooM, and A. Picheny, "Decoder selection based on cross-entropies," in *ICASSP*, 1998, pp. 20–23.

[14] D. Povey and P. Woodland, "Improved discriminative training techniques for large vocabulary continuous speech recognition," in *ICASSP*, 2001.

[15] M. Kozielski, J. Forster, and H. Ney, "Moment-based image normalization for handwritten text recognition," in *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, Sept 2012, pp. 256–261.