2009 10th International Conference on Document Analysis and Recognition

Improvements in BBN's HMM-based Offline Arabic Handwriting Recognition System

Shirin Saleem, Huaigu Cao, Krishna Subramanian, Matin Kamali, Rohit Prasad, Prem Natarajan
BBN Technologies, Cambridge MA, USA
{ssaleem,hcao,ksubrama,mkamali,rprasad,pnataraj}@bbn.com

Abstract

Offline handwriting recognition of free-flowing Arabic text is a challenging task due to the plethora of factors that contribute to the variability in the data. In this paper, we address some of these sources of variability, and present experimental results on a large corpus of handwritten documents. Specific techniques such as the application of context-dependent Hidden Markov Models (HMMs) for the cursive Arabic script, unsupervised adaptation to account for the stylistic variations across scribes, and image pre-processing to remove ruled-lines are explored. In particular, we proposed a novel integration of structural features in the HMM framework which exclusively results in a 9% relative improvement in performance. Overall, we demonstrate a relative reduction of 17% in word error rate over our baseline Arabic handwriting recognition system.

1. Introduction

Commercial off-the-shelf (COTS) OCR software can accurately recognize clean machine-printed text with simple layouts. However, the recognition of handwritten text continues to be a challenging research problem due to the various sources of variability in the data. Different people have different styles of writing which results in inconsistent shapes for the same glyphs. In fact, random variations in shapes are encountered across different instances of glyphs written by the same scribe. Furthermore, the readability of handwritten text is adversely affected by sloppy writing, even for humans. Some of the artifacts of sloppy writing are slanted characters, non-linear baseline across words and presence of scratches and fragments in the text. In addition to the above challenges, the recognition of Arabic text poses its own set of problems due to the inherent connectedness of the script. Character segmentation for Arabic is hard because of the cursive nature of the script. There are also several characters in Arabic that are distinguished only in the placement and number of dots and strokes. All of these factors make the recognition of free-style handwritten Arabic text an inherently difficult problem.

In this paper, we address some of these challenges, and present techniques that result in improved performance of BBN's Hidden Markov Model (HMM) based optical handwriting recognition (OHR) system on Arabic handwritten text. The advantage of an HMM based approach [1, 2] is that it does not require a pre-segmentation of the characters unlike other segmentation-based approaches [3]. Research in off-line Arabic handwriting recognition has previously focused on constrained databases like IFN/ENIT [4] which is a closed vocabulary corpus of images of isolated Tunisian town/village names. In this paper, for the first time, we show results on a large vocabulary free-flowing Arabic handwriting collection. We give an overview of this corpus in Section 2. In Section 3, we present the configuration of the OHR system. A novel integration of structural features (that capture characteristics such as loops, writing, etc in the script) within the HMM framework is proposed in Section 4. We also present results using unsupervised adaptation and image-preprocessing to detect and remove ruled-lines in Section 5 and 6 respectively.

2. Corpus Description

We use handwritten data collected by the Linguistic Data Consortium in our experiments. The data consists of scanned images of handwritten Arabic text of newswire articles, web log posts and newsgroup posts, as well as their corresponding ground truth annotations. The scribes were chosen from different demographic backgrounds. Varied writing conditions such as the type of writing instrument (pen or pencil), type of paper (ruled or un-ruled), and writing speed (careful, normal and fast) were introduced. The images

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.282

were scanned through a high quality scanner, at a resolution of 600 dpi. The ground truth annotations included the co-ordinates of bounding boxes around individual words and the corresponding tokenized transcriptions.

Table 1. Description of dev and test sets

Set	Scribe Set	#Images	#Scribes	#Words
Dev	Not In Training	109	6	9.6K
	In Training	109	6	9.6K
Test	Not in Training	112	6	10K
	In Training	112	6	10K

A total of 8250 pages by 58 different scribes were used for training. Additionally, a set of 442 images generated from a disjoint set of documents was used for validation purposes, which was split into development and test sets. Each of these sets comprised of the same documents written by scribes in the training set, and by new scribes who were never seen in training. There is no overlap of either documents or scribes between the development and test sets. The details of these sets are shown in Table 1. Figure 1 shows examples of two images from the corpus which to the best of our knowledge is largest collection of free-flowing Arabic handwritten documents with annotations. Note that the data exhibits several characteristics that make text recognition hard such as overlapping line/word boundaries, non-linear baseline within lines/words, slanted characters, scratches and poor legibility.

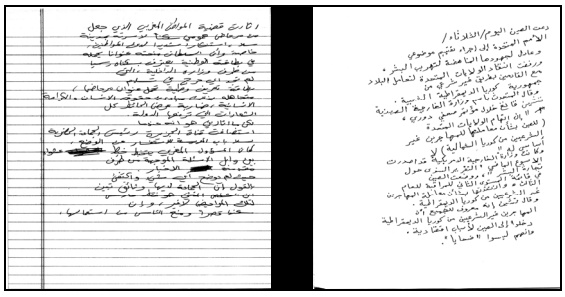


Figure 1: Examples of Images from Corpus

3. Configuration of Baseline OHR System

In this section, we describe the configuration of the BBN Byblos OHR system which is based on the work presented in [5]. The Byblos OHR system models handwritten text as the output of Hidden Markov Model (HMM) based character models. In the following, we provide a detailed description of feature extraction, training, and recognition.

Feature Extraction: Feature extraction is the first step in both training and recognition. In order to convert the 2-dimensional images into a 1-dimensional sequence of features needed to build HMMs, we typically determine the location of the top and bottom boundaries of the lines of text, and then compute the feature vector for each of these lines. If the lines are regularly spaced as in machine-printed data, HMM-based or connected component based line-finding algorithms [6] achieve near-perfect accuracy. However, in free-flowing handwritten data, adjacent lines of text often overlap. Also, handwritten text exhibits significant variations in baseline within a text line. This is an issue for feature extraction since the width of the frame is set proportional to the height of the line. All of the above factors make text line finding/separation for handwritten text a difficult problem. Since line finding is not central to the purposes of this paper, for our experiments, we used the rectangular bounding boxes on the individual word images to obtain a piece-wise linear approximation of the envelope for the text line. Features were computed from the piece-wise linear envelope around the line image. The left and right boundaries of the word were not used for feature extraction. The line image was segmented into a sequence of thin, overlapping vertical windows called frames. A total of 33 of the following script-independent features were extracted from each frame: Percentiles of intensity values, Angle, Correlation, and Energy. Linear Discriminant Analysis (LDA) was then applied to reduce the dimension of the feature space from 33 to 15. This set of transformed features is called PACE and is described in detail in [1].

Training: We used a 14-state, left-to-right HMM to model each individual character. Each state of the HMM has an output probability distribution over the features modeled as a Gaussian mixture. The maximum likelihood estimate of the parameters of the HMM are obtained by iteratively aligning the sequence of feature vectors with the sequence of character models using the Expectation Maximization algorithm.

In handwritten Arabic text, the shape of the character glyph often varies depending on the characters that precede and follow it. Such context-dependence of glyphs is typical of cursive connected scripts, but can vary even more widely because of a writer's personal style. Context-dependent HMMs offer a robust, data driven approach for modeling contextual information. In [7] we showed the superiority of context-dependent models over context-independent models for machine-printed Arabic text. We trained Position-dependent tied mixture (PDTM) HMM models, where a separate set of Gaussians is

estimated for each state of all the context-dependent HMMs associated with a particular character. In total 254K Gaussians were trained for 176 unique characters (including Arabic characters, numerals, punctuations and English characters).

Recognition: The BBN Byblos recognition engine performs a two-pass search using glyph HMMs and the language model. A trigram language model trained on 90 million words of Arabic newswire data was used for recognition. The decoding lexicon consisted of 92,000 of the most frequent words in our Arabic text corpus. The out of vocabulary (OOV) rate of the test set measured against the 92K lexicon is 7.5%. The forward pass is a fast match beam search using the HMMs and an approximate bi-gram language model. The output of the forward pass consists of the most likely word-ends per frame. The backward pass operates on the set of choices from the forward pass to restrict the search space and an approximate trigram language model to produce an N-best list of hypotheses. The N-best list is then re-ranked using a combination of the acoustic scores, and a language model score. The weights for re-ranking were tuned on the development set.

Table 2. Summary of results on test set with baseline OHR system

Decoding Set	%WER
Scribes in Training	43.8
Scribes not in Training	28.6
Overall	36.2

Table 2 shows the performance in terms of word error rate (WER) on the test set. For WER computation, all punctuations and digits were stripped from both the recognition results and the reference transcripts. Contrary to intuition, the %WER of the scribes in training is significantly worse than those never seen in training. Analysis showed that this is an artifact of the scribe selection in the test set. Outliers in each set (a scribe with unusually high %WER in the "Scribes in Training" set, as well as a scribe with unusually low %WER in the "Scribes not in Training" set) contribute to the counter-intuitive results in Table 2. Experiments on a different test set held out from the training data showed the performance on scribes not seen in training to be 31% relative worse than those that were represented in training.

4. Structural Features

In the Arabic script, many letters share common primary shapes and differ only in the number and position of the dots and strokes. Structural features capture intuitive aspects of writing such as loops, branch-points, endpoints, and dots. One such family of features are the GSC (Gradient, Structure and Concavity) [8] features. GSC features are symbolic, multi-resolution features that combine three different attributes of the shape of a character – the gradient representing the local orientation of strokes; structural features that extend the gradient to longer distances and provide information about stroke trajectories; and concavity that captures stroke relationships at long distances. While GSC features have successfully been used in recognition of isolated digits and handwritten words in the past [9] using segmentation-based approaches, they have never been used in the HMM framework. In this section, we describe a novel integration of these structural features in BBN's HMM-based text recognition framework.

For computing the GSC features, first a gradient map is constructed from the normalized image by estimating gradient value and direction at each pixel. Next, Gradient features are obtained by counting the pixels which have almost the same gradient. The structure features enumerate complex patterns of the contour. To compute the concavity features, pixels which lie in certain special regions such as holes and strokes are detected. The image is then divided into bins and the number of such pixels in each bin is counted.

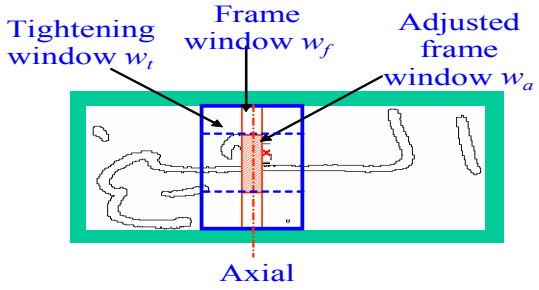


Figure 2. Illustration of frame tightening for feature computation

The width of the sliding window used to compute the GSC features is wider than that used for percentiles. Since the baseline of a word image may fluctuate within portions of the same word, we also algorithmically tightened the upper and lower boundaries of the sliding window. This ensures that the features are normalized and minimizes the variation of the feature space. Figure 2 shows thematically the tightening of the frame. We define the width of the frame (*wf*) as $wf.width = wf.height/12$, where *wf.height* is the height of the word image. A tightened window *wt* is obtained by expanding *wf* to both the left and right sides so that $w_t.width = 5w_f.width$. The upper and

lower boundaries of w_f are redefined by the bounding box of black pixels within w_t. We thus obtain an adjusted window w_a. The region within w_f which is outside of w_a does not have any black pixels. The GSC features are computed from the adjusted frame window w_a. The tightened window w_a is divided evenly into 12 overlapping vertical bins, and 4 sets of GSC features are computed for each bin. A total of 48 of each of the GSC features are computed for each frame. As described in section 3, we use an LDA to reduce the dimension of the feature vector to 15. This number was empirically determined.

We tried different combinations of the PACE and individual GSC features with and without frame tightening. Frame tightening consistently showed improved performance. The combination of Gradient and Concavity features with the PACE features yielded the biggest gain - a 9% relative reduction in WER as show in Table 3.

Table 3. Summary of improvements on test set with structural features

Features	%WER
PACE	36.2
+ Gradient & Concavity	33.0

5. Unsupervised Adaptation

Adaptation has been widely used to combat the variability in speech in automatic speech recognition and to adapt to fonts and degradations in text recognition of machine-printed documents. In handwritten text variability occurs due to inter-scribe differences in writing style, font and slant.

Table 4. Summary of improvements on test set with unsupervised adaptation

Adaptation	%WER		
	Overall	Scribes in Training	Scribes not in Training
None	33.0	39.6	26.4
Page-wise	31.5	37.8	25.1

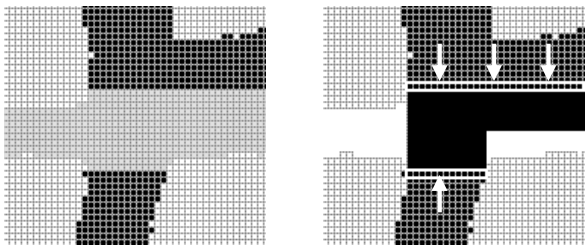
We used text recognition output (top-best hypothesis from the re-ranked n-best list) of each page to adapt the means of the Gaussians of the HMMs using Maximum Likelihood Linear Regression (MLLR) estimation [10]. A maximum of 10 regression classes were used for transformation. The adapted model was then used to re-decode the same page. As shown in Table 4, unsupervised adaptation resulted in a 4.5% relative reduction in WER. As one would expect, adaptation improves results for scribes not in training more than for scribes in training.

6. Ruled-Line Detection and Removal

The performance of the text recognition system was found to be considerably worse on pages with horizontal rules compared to pages without such rules (Table 5). This is due to the sensitivity of the majority of the features discussed in Sections 3 and 4 on variations of the pixels in the frame. Therefore, we pre-processed the images to detect and remove such lines using an algorithm described in this section.

Ruled-line detection: The ruled-line detection program works by finding the local maxima from the horizontal projection profile of the intensity of the input image. The input image is initially divided into 10 equal vertical strips. Ruled-lines are detected at each strip as follows.

- The projection profile of the intensity is normalized by dividing it by the width of the strip. The normalized projection profile is denoted by PROJ(Y).
- A smoothing template is applied to the projection profile. The smoothed value PROJ_SMOOTHED(Y) at a certain Y is the average of all the values within a window of W pixels wide, centered at Y. Different values of W (set to $2dpi/300$ and $4dpi/300$) are used to detect lines of different widths.
- We search the smoothed projection profile for pixels Y^* such that PROJ_SMOOTHED(Y^*) > 0.5 and the local maxima is within the radius of $12dpi/300$
- For each Y^* detected, we search within $|Y - Y^*| < 3 \cdot dpi / 300$ of the strip for black pixels and mark them as "ruled-line pixels"



(a) Ruled-line detection (b) Classification of ruled-line pixels

Figure 3. Classification of ruled-line pixels as black or white

Ruled-line removal: The detected ruled-lines are classified as "black" or "white" heuristically. If a non ruled-line pixel is black and adjacent to a ruled-line pixel, then the pixels starting from the one immediately connected to the non ruled-line pixel to the one at the center of the ruled-line are classified as "black". The rest of the pixels are classified as "white". An illustration is shown in Figure 3.

It was found that the line-removal algorithm results in the removal of some pixels in the glyphs that are connected to the line leading to disconnected character segments in the image as shown in Figure 4. However, since the feature extraction algorithm does not rely on connected component analysis, the creation of disconnected components does not significantly affect system performance.

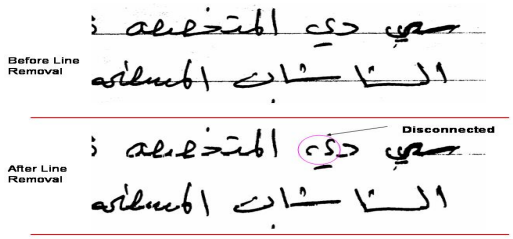


Figure 4. Example of line-removal

Line removal was run on all pages annotated as ruled in the training set. These processed images were used in combination with the un-ruled images to train the glyph models. On the test set images, we ran the fully automatic ruled-line detection and removal algorithm. The precision and recall of the ruled-line detection algorithm were 100% and 98.2% respectively on the test set indicating that there were very few ruled-line detection errors. In Table 5, we summarize the %WER on images with and without ruled-lines separately. Ruled-line removal results in a large gain on the set with page lines. However, the gain on the overall set is 4.8% since only 25% of the test images consist of ruled lines.

Table 5. Summary of improvements on test set with line removal algorithm

Ruled Line Removal	%WER		
	Overall	Ruled	Un-ruled
None	31.5	36.2	29.9
Applied	30.0	31.5	29.5

8. Conclusions and Future Work

In this paper, we presented BBN's Arabic offline handwriting recognition system. Several diverse techniques that addressed the inherent variability of handwritten text as well as the nuances of the Arabic script were presented. Overall, we demonstrated a 17% relative reduction in word error rate over the baseline system. In the future, we propose to further improve performance by exploring discriminative features and models, writer-adaptive training, and baseline detection and correction techniques.

9. Acknowledgements

We would like to thank Dr. Venu Govindaraju, Dr. Srirangaraj Setlur, and Dr. Zhixin Shi at the Center for Unified Biometrics and Sensors at SUNY Buffalo for providing the GSC feature extraction library for this research.

10. References

[1] P. Natarajan, Z. Lu, I. Bazzi, R. Schwartz, and J. Makhoul, "Multilingual Machine Printed OCR," *International Journal of Pattern Recognition and Artificial Intelligence*, 2001, pp. 43–63.

[2] R. Al-Hajj, L. Likforman-Sulem, and C. Mokbel, "Arabic handwriting recognition using baseline dependant features and hidden markov modeling," *International Conference on Document Analysis and Recognition*, 2005, pp. 893-897

[3] L. Lorigo, V. Govindaraju, "Segmentation and Pre-Recognition of Arabic Handwriting" , *International Conference on Document Analysis and Recognition*, Seoul, Korea, 2005.

[4] M. Pechwitz, S. Snoussi Maddouri, V. Märgner, N. Ellouze, H. Amiri, "IFN/ENIT-Database of Handwritten Arabic words," *7th Colloque International Francophone sur l'Ecrit et le Document*, Hammamet, Tunis, 2002.

[5] P. Natarajan, et al, "Multi-lingual Offline Handwriting Recognition Using Hidden Markov Models: A Script-Independent Approach," *Springer Book Chapter on Arabic and Chinese Handwriting Recognition*, Vol. 4768, March 2008. pp. 231-250.

[6] Z. Lu, R, Schwartz, C, Raphael, "Script-independent, HMM-based text line finding for OCR," *International Conference on Pattern Recognition*, Barcelona, Spain, Vol. 4, 2000.

[7] R. Prasad, S. Saleem, M. Kamali, R. Meermeier, P. Natarajan, "Improvements in Hidden Markov Model Based Arabic OCR", *International Conference on Pattern Recognition*, Tampa, U.S.A, December 2008

[8] S. Tulyakov, V Govindaraju, "Probabilistic model for segmentation based word recognition with lexicon," *International Conference on Document Analysis and Recognition*, 2001

[9] J. T. Favata and G. Srikantan, "A multiple feature/resolution approach to handprinted digit and character recognition." *International Journal of Imaging Systems and Technology*, 1996

[10] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaking adaptation of continuous density HMMs," *Computer Speech and Language*, 9, 171-186, 1995.