A Robust, Language-Independent OCR System

Zhidong Lu, Issam Bazzi, Andras Kornai, John Makhoul,
Premkumar Natarajan, and Richard Schwartz

BBN Technologies, GTE Internetworking, Cambridge, MA 02138

ABSTRACT

We present a language-independent optical character recognition (OCR) system that is capable, in principle, of recognizing printed text from most of the world's languages. For each new language or script the system requires sample training data along with ground truth at the text-line level; there is no need to specify the location of either the lines or the words and characters. The system uses hidden Markov modeling (HMM) technology to model each character. In addition to language independence, the technology enhances performance for degraded data, such as fax, by using unsupervised adaptation techniques. Thus far, we have demonstrated the language-independence of this approach for Arabic, English, and Chinese. Recognition results are presented in this paper, including results on faxed data.

Keywords: character recognition, OCR, speech recognition, Hidden Markov Models, Arabic OCR, Chinese OCR

1. INTRODUCTION

The use of Hidden Markov Models (HMM) in developing continuous speech recognition (CSR) technology has several useful aspects, including language-independent training and recognition methodology; automatic training on non-segmented data; and simultaneous segmentation and recognition. We have developed a language-independent OCR system that uses existing continuous speech recognition technology [4,16,22]. Except for the preprocessing and feature-extraction stages, our OCR system utilizes the BBN BYBLOS continuous speech recognition system [18] without any modification to perform the training and recognition. In section 2, we present an overview of the OCR system and describe our approach of using HMMs for character recognition. In section 3, we demonstrate the language independence of our system by presenting results for Arabic, English, and Chinese. In section 4, we show the robustness of our system in recognizing documents of poor quality by presenting results on faxed English data.

A number of research efforts made use of HMMs for off-line printed and handwriting recognition, but recognition was always performed on a single language. Our approach is most similar to those of [1,7,9,11,12,17] in that we are extracting features from thin slices of the line image, which is the key to making the recognition system language-independent. However, often additional feature extraction steps are taken, as in Elms and Illingworth [9], who also extract features from horizontal slices, which necessitates some pre-segmentation at the character level, making their system (developed for recognition on printed Roman characters) inappropriate for languages with connected script. Aas and Eikvil [1] used a bounding box around each word to be recognized and extracted features from vertical thin slices to perform recognition on a single printed Roman font. Kornai [11] also used features extracted from vertical thin slices to perform recognition on handwritten addresses from the CEDAR corpus. There has been little work in using HMMs for the recognition of Arabic script [2]. Allam [3] used contour tracing to locate groups of connected characters; the recognition was then performed on each such group as a whole, using features extracted from vertical slices. The feature extraction methods in Ben Amara and Belaid [6] and Yarman-Vural and Atici [25] were specific to Arabic and may not be easily generalizable. Except for Park and Lee [19], there has been almost no work in using HMMs for the recognition of off-line printed oriental script.

Our approach of using HMMs departs from other OCR approaches in three ways. First, our approach is focused on the problem of language-independent recognition: the major components of the system (feature extraction, training, and recognition) are designed to be script-independent. Second, training and recognition are performed using an existing continuous speech recognition system with no modification except of course for preprocessing and feature extraction. Third, there is no need to perform any presegmentation either at the character or at the word level. This contrasts with other work where presegmentation is often performed at the character level, and almost always at the word level (which would be quite problematic for Arabic). A segmentation-free approach is important for the recognition of degraded documents where characters are often connected, and makes it easy to train the OCR system on new corpora and to apply the system to new scripts.

2. OVERVIEW

2.1 Basic System

Our OCR system includes two parts: the training system and the recognition system as shown in a block diagram in Figure 1. The system depends on the estimation of character models, as well as a lexicon and grammar, from training data. The training system takes as input training data scanned text images coupled with ground truth. After a preprocessing stage in which the page is deskewed and lines of text are located, each line is divided into narrow overlapping vertical windows. Then we extract a set of simple features for each window (see 2.2). The character-modeling component then takes the feature vectors and the corresponding ground truth to estimate the character models (see 2.3). Since each line of text may be transcribed in terms of words, the character-modeling component also can make use of a lexicon obtained from a large text corpus. A language model (grammar) for recognition can also be estimated from the same text or a larger set of text. The training process also makes use of orthographic rules that depend on the type of script. For example, the rules tell whether the text lines go horizontally or vertically, as in traditional Chinese, and if vertically, whether the text is read from left-to-right, as in Roman script, or right-to-left, as in Arabic script.

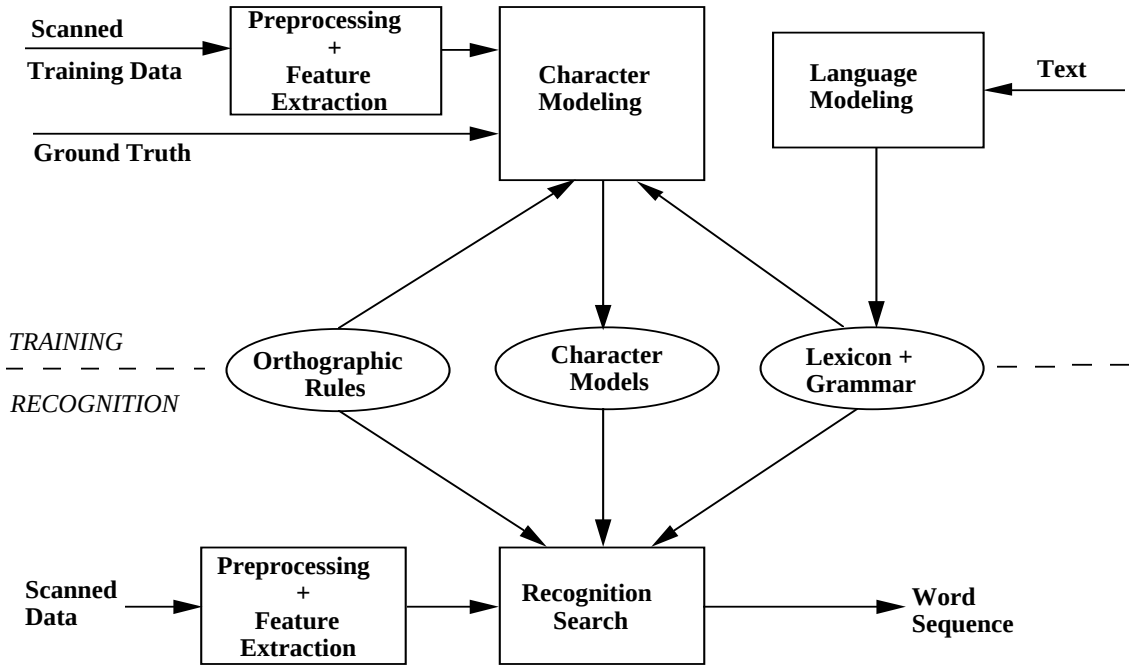

Figure 1: Block diagram of the OCR system.

The recognition system uses the same preprocessing and feature extraction components used in training. During recognition the output of the feature extraction, together with the different knowledge sources estimated in the training (character models, lexicon, and grammar), are used to find the character sequence that has the highest likelihood by Viterbi search. In our system, only the knowledge sources that are shown as ellipses in Figure 1 depend on the particular language or script. Because the whole training and recognition system, shown as rectangular boxes, is designed to be language-independent, the same basic system can be used to recognize most of the world's languages with little or no modification. The whole system is based on modeling each of the characters to be recognized by a hidden Markov model (HMM). This probabilistic approach has the advantages that it does not require separate segmentation at the character and word levels, the training is performed automatically on data that is not presegmented, and the whole system does not depend on the language or script being recognized.

2.2 Preprocessing and Feature Extraction

In order to use HMMs, we need to compute a feature vector as a function of a single independent variable. In speech, when we divide the speech signal into a sequence of windows (which we call frames) and compute a feature vector for each frame, the independent variable is time. In on-line handwriting recognition, feature vectors are computed from the pen coordinates, which are also a function of time [24]. In OCR, however, we usually face the problem of recognizing a whole page of text, so there is no natural way of defining a feature vector as a function of a single independent variable. In fact, different approaches have been taken in the literature [2]. At this stage in our work, we have chosen a line of text as our major unit for training and recognition (in what follows we assume lines to be horizontal, without loss of generality). Prior to finding the lines, we find the skew angle of the page and rotate the image so that the lines are horizontal. Then we use an HMM method to find the top and bottom of each line. Once lines are located, we are ready to perform feature extraction by using horizontal position along the line as the single independent variable.

We scan a line of text from left to right (right to left for Arabic). At each horizontal position, we compute a feature vector on a narrow vertical strip, which we call a frame. We divide a line into a sequence of overlapping frames. Each frame is a narrow vertical strip whose width is a small fraction (typically about 1/15) of the height of the line, and the height is normalized to minimize the dependence on font size. The overlap from one frame to the next is a system parameter currently equal to two-thirds of the frame width (see Fig. 2). Fig. 2 also shows that each frame is divided into 20 equal overlapping cells (again, the cell overlap is a system parameter). The basic features we compute are simple and script-independent:

- intensity (percentage of black pixels within each cell) as a function of vertical position;
- vertical derivative of intensity (across vertical cells);
- horizontal derivative of intensity (across overlapping frames);
- local slope and correlation across a window of 2-cell square.

Note that we have specifically excluded features that require any form of partial recognition, such as sub-character pieces (e.g., lines, curves, dots), as well as features that are specific to a particular type of script.

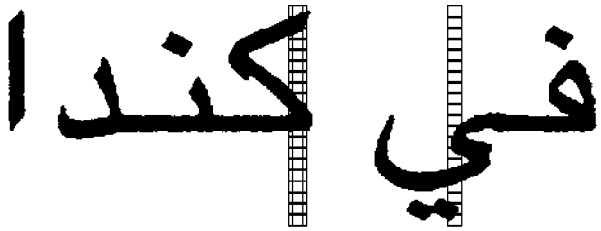

Figure 2: Dividing a line of text into frames and each frame into cells.

Although the intensity features alone would represent the entire image, we include other features, such as vertical and horizontal derivatives, local slope, and local correlation, so as to include more global information and to help overcome the limitation imposed by the conditional independence assumption inherent in HMMs. The result is a set of 80 basic features per frame. We then perform linear discriminant analysis (LDA) [10] on these basic features and choose the top 15 dimensions as the final features for our system. The LDA features not only save computation and memory but also improve performance in accuracy.

2.3 HMM Character Structure

The central model of the OCR system is the HMM of each character. For each model, we need to specify the number of states and the allowable transitions among the states. Associated with each state is a probability distribution over the features. The model for a word then is a concatenation of the different character models.

Our HMM character structure is a left-to-right structure that is similar to the one used in speech (see Figure 3). The loop and skip arcs in Figure 3 allow relatively large, nonlinear variations in horizontal position. In our current system, 14-state models are used for all characters. This structure was chosen subjectively to represent the characters with the greatest number of horizontal transitions. While it would be possible to model different characters with different numbers of states, we find it easier to use the same number of states for all characters.

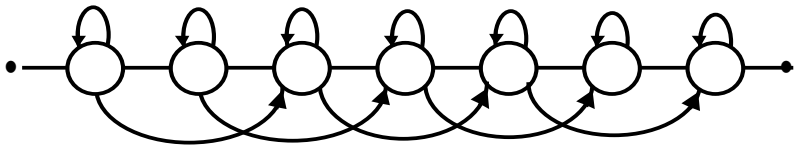

Figure 3: 7-state Hidden Markov model (HMM). 14-state models are used for OCR.

To model the probability structure of each state of character HMMs we employed what is known in the speech recognition literature as a phoneme-tied-mixture (PTM) Gaussian structure. For each character, the probability density of the 15-dimension feature vector is usually modeled with a mixture of 128 Gaussian densities. The parameters of these densities are initialized with a clustering process [15] on a subset of the training data and trained thereafter. Thus, for each of the characters, there are 128 mixture weights (one for each Gaussian) that represent the probability density for that character.

2.4 Training and Recognition

The training algorithm is the same as the one used in our BYBLOS speech recognition system [18]. Line by line, the features extracted from the vertical slices, coupled with the ground truth transcriptions, are used as input to train the character models. No presegmentation is needed either at the character level or at the word level. The forward-backward training algorithm is used to estimate the model parameters [22]. The resulting models maximize the likelihood of the training data given the ground truth transcriptions. In addition to the character HMMs, we also compute a lexicon and a language model. These are generally estimated from large text corpora which are independent of the image data. The language model is usually a bigram or trigram model that contains the probabilities of all pairs or triples of words. Because it requires only text to estimate a language model, we can use much larger amounts of text than present in the image corpus to develop more powerful language models.

The recognition algorithm is also identical to the one used in our speech recognition system. The recognition process is mainly a search for the most likely sequence of characters given the input feature vector sequence, the lexicon, and the language model. Since the Viterbi algorithm would be quite expensive when the state space includes a very large vocabulary and a bigram or trigram language model, we use a multi-pass search algorithm [23]. If we use a language model to improve the performance of the OCR system, the out-of-vocabulary (OOV) problem will be severe, since virtually no lexicon of a fixed size is large enough for natural language. We have developed a technique [4] to solve the OOV problem. With a moderate-size lexicon (e.g. 30,000 words), this technique can decrease error rate by a factor of 2.

3. LANGUAGE-INDEPENDENT OCR

One of the primary benefits of using HMMs to model characters is that the fundamental models are not designed for any particular language, thus making it possible to apply the same technology to most of the world's scripts. The only requirement we have is that the text be written in lines. We trained our system to perform OCR on Arabic, English, and Chinese. The Arabic and English systems provide high quality unlimited vocabulary omnifont recognition [4]. The Chinese OCR system can recognize 3,870 characters, including 3755 simplified Chinese characters from GB2312-80 Level I and 115 Roman/punctuation characters. Our systems are easily trainable. We carried out experiments to show that we can train our system on a moderate size training corpus and get much improved performance on that specific type of data. The character error rates (CER) reported here were computed as the total error rate of substitution, deletion, and insertion errors. In the following we present some results on Arabic (3.1), English (3.2), and Chinese (3.3).

3.1 Arabic

Arabic characters are connected in printing which makes the OCR task hard for any methodology that requires presegmentation. The shapes of Arabic characters are also context-dependent in that the character's shape depends on whether it is connected to other characters from neither, left, right, or both directions. Our system, equipped inherently with segmentation-free algorithm and context-dependent models, is good at dealing with the above problems. We used the DARPA Arabic OCR Corpus to train and test our omnifont system. The corpus contained 345 pages of images collected from books, magazines, newspapers, and computer fonts. The image quality is variable (see Figure 4 for a sample). We used an 89-character set. Character trigram and word unigram language models were estimated from the text of this corpus.

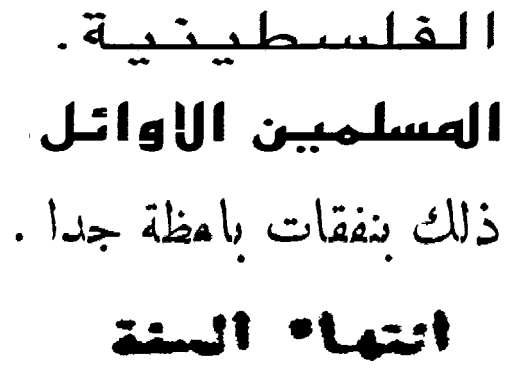

Figure 4: Sample images from DARPA Corpus

In order to show that our system can perform well on new data sets with training, we collected a corpus of about 100,000 characters from the Arabic newspaper An Nahar. This corpus is mainly a unifont corpus with some minority fonts.

We used a training set of 100,000 characters from the DARPA Arabic OCR Corpus to train our omnifont Arabic system. Test results based on a disjoint test set from the same corpus are shown in the first two rows of Table 1. Without a lexicon, a character error rate (CER) of 4.7% was obtained. With a 30K-lexicon, a CER of 3.3% was obtained using our unlimited-vocabulary technique [4] which solved the out-of-vocabulary (OOV) problem, a 30% relative improvement. In another experiment, we trained the system with 50,000 characters from the Arabic newspaper An-Nahar and tested on the same newspaper without a lexicon. A CER of 0.8% was obtained as shown in the third row of Table 1. The training set of 50,000 characters is less than a page of the newspaper.

Corpus	Font	Lexicon	Character Error Rate (%)
DARPA Corpus	Ominfont	No	4.7
DARPA Corpus	Omnifont	30K	3.3
Newspaper	"Unifont"	No	0.8

Table 1: Arabic OCR Results

3.2 English

We used the University of Washington English Document Image Database I to train and test our omnifont system. There are 958 image zones from books, journals, and magazines. A 90-character set was used. Character trigram and word unigram language models were also estimated from the text of this corpus. Figure 5 shows samples from the corpus.

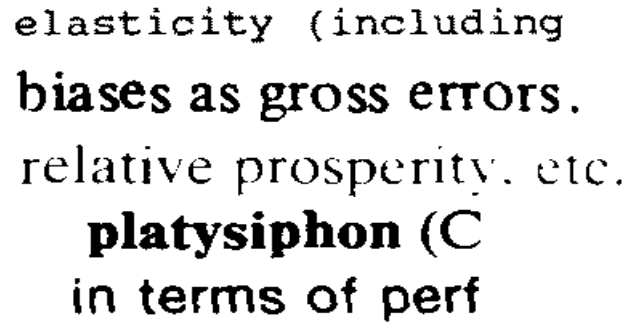

Figure 5: Sample images from UW Database I

We trained our omnifont English OCR system with a training set of 100,000 characters from the UW Database and tested the system on a disjoint test set from the corpus. A CER of 2.1% was obtained without a lexicon. With a 30K lexicon, a CER of 1.1% was obtained using the unlimited-vocabulary technique [4]. The improvement is nearly a factor of 2 with a lexicon. Comparing Arabic to English using our unlimited-vocabulary technique, we note that the average error rate for Arabic was about three times that of English (3.3% versus 1.1%). We speculate that the higher error rate for Arabic is due to several causes: the greater similarity of Arabic characters; the connectivity of Arabic characters and the existence of ligatures; the wider diversity of fonts in the Arabic corpus; and the lower quality of some of the Arabic data.

3.3 Chinese

To further demonstrate the language-independence of our approach, we extended our system to Chinese, which differs from languages like English and Arabic in that the script does not have a small number of characters from which all words are put together. When collecting a training corpus, it is therefore extremely difficult to find samples of all characters in actual use. Also, Chinese characters in general have very complicated structure, and it is not obvious from the outset that the simple models described in 2.3 are appropriate to model these complex characters.

In order to cover all the Chinese and Roman characters to be recognized, we first collected a computer-generated corpus by printing and scanning images of all the 3,755 unique characters in GB2312-80 Level I of simplified Chinese and of 115 Roman/punctuation characters in 4 fonts (Fangsong, Hei, Kaishu, and Song). The size of the character set is 3,870. Each of the 3,870 characters occurs 14 times in each of the four fonts. The characters occur in random sequences. Figure 6 shows a sample from the computer-generated corpus.

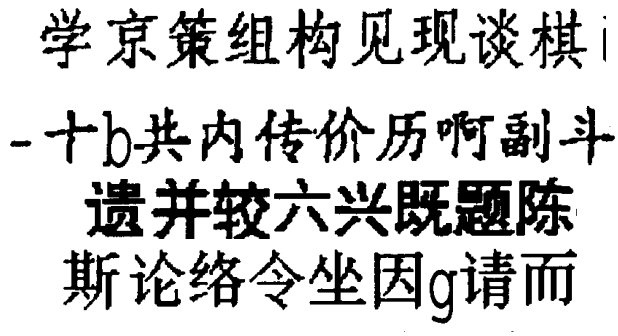

Figure 6: A sample from the computer-generated Chinese corpus

The first experiments were performed on this corpus to test whether the HMM technology works for Chinese OCR at all. Each character was modeled by a 14-state HMM, just like for English and Arabic. No language model was used. In the unifont experiments, where the system was trained and tested on a single font at a time, the character error rate (CER) ranged from 0.1% to 0.4% for the four fonts, with an average of 0.3%. In the multifont experiment, where the system was trained and tested on a pool of data from the four fonts, the average error rate was 0.5%. Both of these results demonstrated that the complexity of the characters did not present any problem for the models we were using.

In order to get a sense of how different the four fonts were from each other, we ran a cross-font experiment in which the system was trained on three of the fonts and tested on the fourth. The result was a CER of 10%, meaning the fonts are quite different from each other. This experiment suggested that in order to obtain good performance on real data, which is bound to look different from the synthetic data, we would have to collect training data from actual printed sources. Therefore we collected a real printed corpus of 60,000 characters from the Chinese newspaper People's Daily. The corpus has only 2,600 unique characters and is mainly a unifont corpus with some minority fonts. Figure 7 is a sample from this corpus.

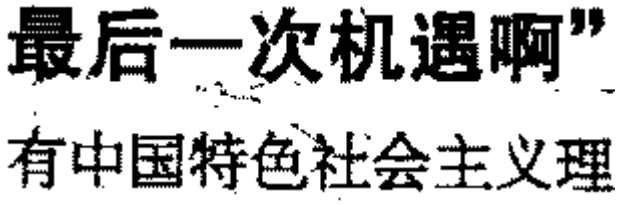

Figure 7: A sample from the printed Chinese corpus

For our experiments with real Chinese data, we used both the real and the computer-generated data for training but only real data (a disjoint set) for testing. From the real corpus we used 16,000 characters for training, which covered about 1600 unique characters. From the computer-generated data we used 3,870 unique characters, each in the four fonts. The test set contained 1225 unique characters, 2% of which did not appear in the real training data. However, we assumed we would have a chance at recognizing some of these characters based on the computer-generated training even though we knew that the computer fonts and real fonts were rather different.

The results of this experiment were quite satisfying. The overall character error rate was 1.5%**.** The error rate for those characters with no real training data was 7%. Characters that had between 1 and 5 training tokens in the real data had an average error rate of about 3%, while characters with more than 5 training tokens had an average error rate of 1%. This result shows that it is possible to achieve low error rate on Chinese character recognition of real images even though we did not have training data for all of the characters, and many of the characters had very little real training data.

4. OCR ON DEGRADED DATA

Given that the most ubiquitous image sources are fax machines, copying machines, and handheld scanners, which can produce low quality output compared to the 600dpi flatbed scanners used in many research undertakings, it is important to achieve good performance on degraded images. Initially we considered both photocopying and fax, but our first experiments on photocopies showed that there was no degradation due to multiple generations of copying, at least not on the high-end copiers we had. Therefore, we decided to concentrate on dealing exclusively with fax images. One basic approach to noisy data would be to explicitly model the noise and then modify a system trained on clean data. However, this presumes a better understanding of the fax degradation process than we have, so we decided on the simple and robust method of training the system directly on the degraded data (4.1). This approach is ideal for our system which is easily trainable and already has ready-to-use unsupervised adaptation techniques (4.2) that require no transcribed data.

4.1 Training on fax

We collected an English corpus of fax images by faxing pages from the UW English Document Image Database I to different fax machines and scanning in the results. About 100,000 characters are used as the training set. Figure 8 is a sample of the original and faxed images. Notice that in addition to the overall blurring effect there are also broken and connecting characters in the faxed image.

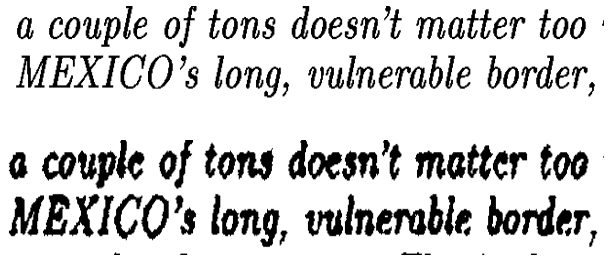

Figure 8: Original image (top) and its faxed version (bottom)

As a baseline experiment, we measured the degradation due to recognizing fax images when the training had been done on clean images. In a closed-vocabulary experiment using the English data, we found that the character error rate for the faxed data was 5.3%, compared to 0.8% for the clean data before faxing - a six-fold increase in error. Simply by training the system on fax data (which was trivial since we already had the transcriptions from the original data before it had been faxed) we could reduce the error rate from 5.3% to 2.2%.

4.2 Adaptation

Each image of fax data has different properties. Thus, even when the system has been trained on fax data, there are some differences in each new page. We hypothesized that we could take advantage of the fact that we were dealing with a full page, and hence with one type of degradation, to improve performance further through adaptation techniques borrowed from speaker adaptation in speech recognition. If we can determine that a portion of speech was produced by a single speaker, we can use unsupervised adaptation techniques to improve recognition accuracy for that portion. The process works as follows.

First, we recognize the speech using a speaker-independent model. Then, assuming that the recognized words are correct, we adapt the speech model to the voice of the new speaker, essentially retraining the speech model to the voice of the speaker. Using the newly adapted model, we re-recognize the speech. Since in speech we find that the adapted model results in a 20% reduction in word error rate, we decided to adapt to each page of text, because both the fonts and fax degradation are generally uniform within, but not across, pages.

Of the several adaptation techniques found in the speech recognition literature we decided to try Maximum Likelihood Linear Regression (MLLR) adaptation [13], with which we have had significant experience. MLLR adaptation is used when the amount of data available for adaptation is not sufficient to reliably adapt all the parameters of the HMM. The method limits itself to updating the means of the mixture components that make up the state output distributions. The rationale behind MLLR is that the difference between speakers, or pages of text, is mainly characterized in the estimates of the mixture component means. As there is no adaptation of transition probabilities, mixture component weights, or mixture component covariances, these parameters take their values from the original model set. Starting with the system trained on fax data, with MLLR the error rate dropped from 2.2% to 1.7% - a reduction of about 20% in error rate, which is comparable to the benefit we receive for speaker adaptation in speech recognition.

Training Set	Adaptation	Test Set	Character Error Rate (%)
Original	No	Original	0.8
Original	No	Faxed	5.3
Faxed	No	Faxed	2.2
Faxed	Yes	Faxed	1.7

Table 2: Fax Results

Table 2 gives a summary of the results on fax data presented. As the table shows, the final result (1.7% CER) is only around a factor of two higher than the best case of training and testing on original clean images. This result was obtained by training the system on fax data and including unsupervised adaptation as part of the recognition process.

5. CONCLUSION

In this paper, we presented a language-independent OCR system performing open-vocabulary OCR on Arabic, English, and Chinese. The system is based on Hidden Markov Models and utilizes the same advanced technology that is used for speech recognition. We showed that our HMM-based OCR system was easily trainable on new sets of data and portable to recognize new scripts. With training, the system could achieve robust performance on degraded data.

REFERENCES

1. K. Aas and L. Eikvil, "Text page recognition using grey-level features and hidden Markov models," Pattern Recognition 29, 977-985, 1996.
2. B. Al-Badr and S. Mahmoud, "Survey and bibliography of Arabic optical text recognition," Signal Processing, Vol. 41, No. 1, pp. 49-77, 1995.
3. M. Allam, "Segmentation versus segmentation-free for recognizing Arabic text," Proc. SPIE, Vol. 2422, 228-235, 1995.
4. I. Bazzi, C. Lapre, R. Schwartz, and J, Makhoul, "Omnifont and unlimited vocabulary OCR system for English and Arabic," Proc. International Conference on Document Analysis and Recognition, Ulm, Germany, Vol. 2, 842-846,1997.
5. J. Bellegarda and D. Nahamoo, "Tied Mixture Continuous Parameter Models for Large Vocabulary Isolated Speech Recognition," IEEE Int. Conf. Acoustics, Speech, Signal Processing, Glasgow, Scotland, Vol. 1,13-16, May 1989.
6. N. Ben Amara and A. Belaid, "Printed PAW recognition based on planar hidden Markov models," 13th Int. Conf. Pattern Recognition, Vienna, Austria, Vol. II, 220-224, 1996.

7. W. Cho, S.-W. Lee, and J.H. Kim, "Modeling and recognition of cursive words with hidden Markov models," Pattern Recognition 28, 1941-1953, 1995.
8. R.B. Davidson and R.L. Hopley, "Arabic and Persian OCR training and test data sets," Proc. Symp. Document Image Understanding Technology (SDIUT97), Annapolis, MD, 303-307, 1997.
9. A.J. Elms and J. Illingworth, "Modelling polyfont printed characters with HMMs and a shift invariant Hamming distance," Proc. Int. Conf. Document Analysis and Recognition, Montreal, Canada, 504-507, 1995.
10. R.A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems, " Annals of Eugenics 7, 179-188, 1936
11. A. Kaltenmeier, T. Caesar, J.M. Gloger, and E. Mandler, "Sophisticated topology of hidden Markov models for cursive script recognition," Proc. Int. Conf. Document Analysis and Recognition, Tsukuba City, Japan, 139-142, 1993.
12. A. Kornai, "Experimental HMM-based postal OCR system, " Proc. Int. Conf. Acoustics, Speech, Signal Processing, Munich, Germany, Vol. 4, 3177-3180, 1997.
13. C.J. Leggetter and P.C. Woodland, " Maximum Likelihood linear regression for speaker adaptation of continuous density hidden Markov models," Computer Speech and Language, Vol. 9, pp. 171-185, 1995
14. J. Makhoul, S. Roucos, and H. Gish. "Vector Quantization in Speech Coding," Proc. IEEE 73, 1551-1588, 1985.
15. J. Makhoul and R. Schwartz, "State of the Art in Continuous Speech Recognition," Proc. Natl. Acad. Sci. USA, Vol. 92, pp. 9956-9963, October 1995.
16. J. Makhoul, R. Schwartz, C. LaPre, C. Raphael, and I. Bazzi, "Language-Independent and Segmentation-Free Techniques for Optical Character Recognition," Document Analysis Systems Workshop, Malvern, PA, pp. 99-114, October, 1996.
17. M. Mohamed and P. Gader, "Handwritten word recognition using segmentation-free hidden Markov modeling and segmentation-based dynamic programming techniques," IEEE Trans. Pattern Analysis and Machine Intelligence 18, 548-554, 1996.
18. L. Nguyen, T. Anastasakos, F. Kubala, C. LaPre, J. Makhoul, R. Schwartz, N. Yuan, G. Zavaliagkos, and Y. Zhao, "The 1994 BBN/BYBLOS Speech Recognition System," Proc. ARPA Spoken Language Systems Technology Workshop, Austin, TX, Morgan Kaufmann Publishers, pp. 77-81, January 1995.
19. H.S. Park and S.W. Lee,"Off-line recognition of large-set handwritten characters with multiple hidden Markov models," Pattern Recognition 29(2), 231-244,1996.
20. I.T. Phillips, S. Chen, and R.M. Haralick, "CD-ROM document database standard," Proc. Int. Conf. Document Analysis and Recognition, Tsukuba City, Japan, pp. 478-483, Oct. 1993.
21. L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proc. IEEE, Vol. 77, No. 2, pp. 257-286, Feb. 1989.
22. R. Schwartz, C. LaPre, J. Makhoul, C. Raphael, and Y Zhao, "Language-Independent OCR Using a Continuous Speech Recognition System," Proc. Int. Conf. on Pattern Recognition, Vienna, Austria, pp. 99-103, August 1996.
23. R. Schwartz, L. Nguyen, and J. Makhoul, "Multiple-Pass Search Strategies," in Automatic Speech and Speaker Recognition: Advanced Topics, C-H. Lee, F.K. Soong, K.K. Paliwal, Eds., Kluwer Academic Publishers, 429-456, 1996.
24. T. Starner, J. Makhoul, R. Schwartz, and G. Chou, "On-line Cursive Handwriting Recognition Using Speech Recognition Methods," IEEE Int. Conf. Acoustics, Speech, Signal Processing, Adelaide, Australia, V-125-128, 1994.
25. F.T. Yarman-Vural and A. Atici, "A heuristic algorithm for optical character recognition of Arabic script," Proc. SPIE, Vol. 2727, Part 2, 725-736, 1996.