Recent Improvements in the BBN OCR Syste

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

BBN Technologies, GTE
Cambridge, MA 02138, USA

ABSTRACT

We describe several improvements that we have made in the BBN BYBLOS OCR System. First, we adopted continuous density hidden Markov models (HMMs) rather than discrete density HMMs. This resulted in improved accuracy when more training data is available. It also allowed us to use unsupervised speaker adaptation algorithms (borrowed from speech recognition) for adaptation to font, style, and quality. Second. we sped up the character recognition by a factor of about 50 so that a full page of 2,000 characters requires about 30 to 40 seconds for processing. Third, we tested the system on Chinese characters. This required development of tools to create a training corpus from available sources. It also required techniques for dealing with an open set of characters, where some of the characters may have no real training data. The end result was 1.2% character error on newspaper data.

1. INTRODUCTION

We have previously described the use of the BBN BYBLOS speech recognition system for language-independent OCR [1-4]. After finding lines of text, we compute a feature vector for each narrow vertical slice of the image of each line. We then have a one-dimensional input sequence of feature vectors, which we model as the output of a hidden Markov model (HMM). Beyond the initial image manipulation and feature extraction, the remainder of the processing (model training and recognition) uses the BBN BYBLOS speech recognition system [5] with no source code modifications. We have previously reported man advantages to this approach. First, there is no explicit character segmentation process in either the training or recognition. This allows us to train on a large corpus with minimal effort. Second, the recognition process does not suffer from segmentation errors, and therefore is quite able to handle scripts where the characters are connected - such as Arabic. Finally, we can use all of the mature techniques developed in speech recognition for creating more powerful models or making the system robust to various degradations.

At the preceding SDIUT97 workshop [3], we reported ver promising results on Arabic using the ARPA/SAIC Arabic Corpus [6] and on English OCR using the University of Washington Corpus [7]. This system used discrete HMM models to model the features that result for the characters. It also ran quite slowly, taking 30 minutes or more for a full page of text. It had only been used for alphabetic languages with relatively fe characters. And it had only been tested on relatively clean images.

Since the SDIUT97 workshop, we have made several improvements to the system including:

- upgrading the models from discrete densities to continuous densities, and adding techniques for unsupervised adaptation to font, style, and quality,
- making it much faster,
- extending the system to Chinese OCR, including the development of tools for rapid creation of training corpora, and
- developing techniques for dealing with degraded documents.

This paper discusses the first three items above, while the fourth – dealing with degraded input – is dealt with in a separate paper.

Discrete density HMMs are relatively straightforward to implement but research speech recognition systems no commonly use continuous density HMMs in order to provide more detailed models when more training data is available. There are several reasons that continuous density HMMs are preferable for this effort. First, we had observed that the accuracy of our discrete density HMM system did not improve significantly when we increased the amount of training data. We expect that the continuous density system would be able to benefit from larger training corpora. Second, there are several speaker adaptation algorithms that have been developed for continuous density HMMs. These algorithms can be used for adaptation to the variations in OCR: font, style, and quality. Finally, it was important to us that, wherever possible, we use the same software base for our OCR and speech recognition efforts.

The initial research OCR system was not practical in that it took a very long time (about 1/2 hour) to process a full page of text. This was, in part, due to not having paid attention to speed issues, and also because HMM systems perform a huge search over all possible character sequences with all possible segmentations. While the basic Viterbi beam search reduces the search space, the computation can be quite large. In addition, when we converted the system to use continuous density HMMs, the computation increased significantly because of the large number of Gaussian densities that needed to be evaluated. We streamlined the initial image processing stages and worked on techniques to speed up the recognition phase by a factor of about 50. As a result, the system takes between 30 and 45 seconds to process a full page of text. While this is not as fast as some systems, it makes the system usable.

We were concerned that it would be hard to extend the system to Chinese because of the large number of characters. We didn't know whether the system would be able to deal such detailed characters. Another concern was that a modest sized training corpus of real images would not cover all of the characters even once, which would present a problem for estimating models for new fonts. We also had no corpus of Chinese images and had to develop an inexpensive way of creating one. Our solution was to use sources for which we could find both images and online versions. This required developing techniques to automatically align the characters in the transcripts to the images.

We believe these extensions to the system are important in that they indicate that the system can be made practical and can, in fact, be used for many different languages. In section 2 we review the basic concepts in the HMM-based OCR system. Section 3 describes the results obtained in moving to continuous density HMMs. In Section 4, we describe the techniques that we used to speed up the system by a factor of 50. And Section 5 describes our work on Chinese OCR.

2. Review of OCR Using BYBLOS

We have previously described how we adapted our BBN BYBLOS continuous speech recognition system to the OCR problem. We review the basic techniques here very briefly. For a more complete discussion, see [3]. The fundamental idea, as illustrated in Figure 1, is to convert the two-dimensional image of a line of text into a one-dimensional sequence of feature vectors, so that we can treat the entire image as the output of the combined HMMs for the characters in the line. For each narro vertical slice of the image we compute features as a function of the vertical position. We also compute the derivatives of these features in both the vertical and horizontal directions. Finally we compute several local angles and correlation features to represent angular information. The result is a vector of 80 features for each slice or "frame". (We use the terms ***window, frame, slice,*** interchangeably. Also it is obvious that one can compute other features for each frame.) Once we have done this, we can use the BYBLOS speech recognition software without modification, since we can treat characters as if they were phonemes and words and sentences in text just like words and sentences in speech.

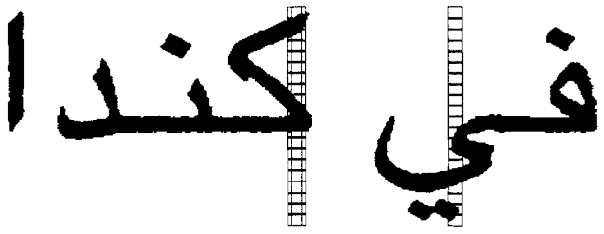

Figure 1: We divide a line of text into overlapping windows or 'frames'. We then compute a vector of features from the pixels within that window.

Figure 2 illustrates the overall BBN BYBLOS OCR system. (This block diagram is essentially the same for speech recognition as for OCR.) The system consists of two major parts: the training system and the recognition system.

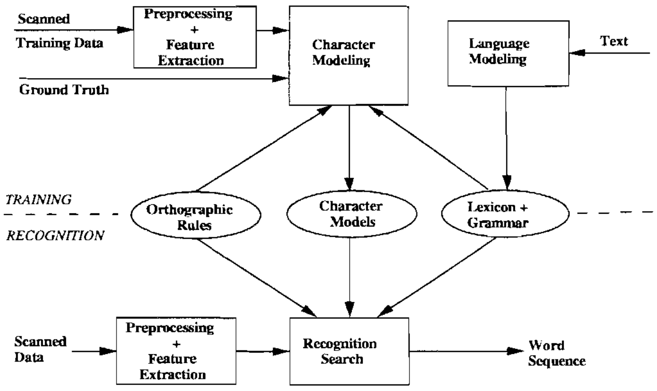

Figure 2: Block diagram of the OCR system.

The training system is provided with scanned images of lines of text and the matching text level transcriptions for those lines. The images are processed first by rotating the image appropriately and then by finding the locations of the lines. Then, features are extracted from each line image. In particular, we divide the line into narrow overlapping vertical slices of the image and extract a feature vector from each slice. The training system uses the forward-backward algorithm [8] to estimate the character models. A language model (prior model on character sequences) for recognition can also be estimated from the same text or a larger set of plain text. This can take the form of a probability distribution on likely character sequences, a lexicon of likely words, and a prior distribution on likely word sequences. The orthographic rules provide simple information about the language, such as the writing direction of the text lines.

In the recognition process, the image is preprocessed in the same way as during training. Then, the goal is to search for the most likely sequence of characters, given the extracted feature vectors from the scanned images, together with the different knowledge sources estimated in training, such as character models, lexicon, and grammar. We use a multi-pass fast search algorithm [9-12 instead of using the Viterbi algorithm because the Viterbi algorithm would be quite expensive when the state space includes a very large vocabulary and a bigram or trigra language model.

The goal of the system is to be language-independent in that it can easily be trained on matching images and text for almost an language and then used to recognize that language. The features extracted are not designed for any particular language and since the training does not require that the images be aligned at the character level, it is extremely easy to create a sufficient training corpus for a new language.

3. Continuous Density System

As stated above, we wanted to use continuous density HMMs for the OCR system for several reasons. In our original system, the models used were "tied-mixture models". First, we divided the entire input feature space into disjoint regions. Then, we defined a single gaussian distribution from the data within each region. The state distributions for all of the states of all of the characters are then estimated as a mixture of these gaussians, where the mixture weights are estimated using the Forward-Backward estimation algorithm. The advantage of this kind of system is that it is very easy to create and the computation is small because we only need to identify the likely gaussians for each input vector once. Although this system technically uses continuous (gaussian) densities, it functions as a discrete density system, since the set of gaussians is shared by all of the states.

By "continuous densities HMMs", we mean that we can use different sets of Gaussian distributions as the basis of the probability density associated with different states of the HMM. For example, we can define a distinct set of gaussians for each of the characters and this same set of gaussians is used for all of the states associated with that character. Thus, only the mixture weights vary from state to state within the character. In speech recognition, this type of HMM is called a "Phoneme-Tied Mixture" (PTM) HMM. Typically we use a set of 256 Gaussians for each character. (For speech recognition we often define systems that have more distinct sets of gaussians.)

We use the same multiple-step procedure for estimating these models that we use in our speech recognition system. First, we compute the feature vectors for a large number of images. We divide the feature space into 256 regions using a binar clustering and k-means process [13]. We define a gaussian for each region simply by computing its sample mean and diagonal variance. Then, we take this set of gaussians as the initial estimates of the gaussians for each of the characters. When we use the Forward-Backward algorithm to reestimate the distributions, the gaussians for each character diverge and the mixture weights for the different states are estimated.

While this provides a reasonable model, we find that a second round of estimation is beneficial. We use the model estimated above to determine the alignment of image frames to the states of the corresponding character HMMs. Then we choose a rando sample of the vectors for a character to initialize a new set of means for that character, which are estimated by going through all of the data several times. Finally, we use the Forward-Backward algorithm again to get final estimates of all of the model parameters. While this process sounds somewhat involved, it results in reliable estimates of the model parameters without the need for any manual alignment of any data. Thus, the human effort is minimized, which is one of our primar goals.

One difference between text images and speech is that text images typically have less variability. In particular, the background is often uniformly white, while in speech, "silence" is a continuous noise signal. Therefore, if we did nothing special, many of the estimated gaussian densities would have zero variance. To avoid this problem we simply added a small constant value to the estimated variances.

To reduce the dimensionality of the 80-dimensional input feature vector, we use linear discriminant analysis (LDA) [14] and choose the 15 features with the largest eigenvalues. This reduces computation and storage while improving the robustness of the estimates of the gaussian densities.

With the continuous-density HMMs, we have more parameters to describe the character models. We found that the continuous density system obtained the same performance as the discrete system when trained on small amounts of training data, but (in contrast with the discrete system) improved further for larger amounts of data. Also the continuous-density HMMs allow us to use the unsupervised adaptation techniques developed for speech recognition. Unsupervised adaptation is useful in dealing with degraded documents.

3.1 Improvement in Performance

One advantage of using the continuous-density system is that more training would improve the performance of the system. We can demonstrate the improvement in the performance of the continuous-density system with a test on English OCR. In the rest of this paper, we refer to the character error rate (CER), which is the total rate of the substitution, deletion, and insertion errors, as the accuracy measure for our OCR system. We used the University of Washington English Document Image Database I [7] to train and test our omnifont system. There are 958 image zones from books, journals, and magazines. However, in most of our experiments, we used a small subset of this data to train our models. A 90-character set was used. We used a character trigram prior language model, which was also estimated from the text of this corpus (excluding the test data). Figure 4 shows some samples from the corpus.

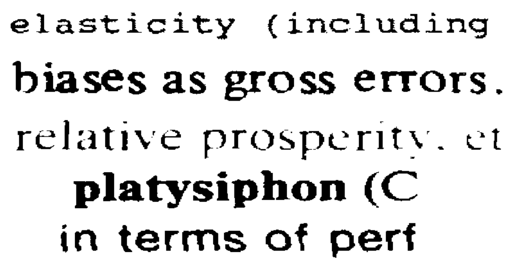
elasticity (including
biases as gross errors.
relative prosperity, etc.
platysiphon (C
in terms of perf

Figure 4: Sample images from UW Database I

We trained our omnifont English OCR system with either a training set of 100,000 characters or a training set of 600,000 characters from the UW Database and tested the system on a disjoint test set from the UW Database.

The CER for different training conditions and different models is shown in Table 1. A CER of 2.1% was obtained without a lexicon on the discrete-density system trained on the 100,000-character training set. No improvement was observed when we trained the discrete-density system on the 600,000-character training set. A CER of 2.1% was also obtained without a lexicon on the continuous-density system trained on the 100,000-character training set. But a big improvement was observed when we trained the continuous-density system on the 600,000 character training set where a CER of 1.2% was obtained. This is because the continuous-density system has the capability to accommodate more variations or degradations in it models.

System	Corpus Size	Char Error Rate (%)
Discrete-density	100K or 600K	2.1
Continuous-densit	100K	2.1
Continuous-densit	600K	1.2

Table1: Improvement in English OCR

4. Recognition Speed

In speech recognition, a system is considered fast if it operates in "real time", which is about 12 phonemes per second. Our speech recognition system uses 5-state HMM models for each phoneme, and the average number of observation vectors is about 8 per phoneme. In contrast, our BYBLOS OCR system uses 14-state models for each character, and the average number of observation vectors per character is about 25. So we could expect that the OCR system would run about 10 times slower or about one character per second. But OCR systems routinely run at 100 characters per second or more.

We increased the speed of the BYBLOS OCR syste dramatically by a combination of techniques. First, we integrated the various preprocessing steps, such as image rotation, linefinding, and feature extraction so that we avoided unnecessary I/O. Then we sped up the procedures for finding lines and for feature extraction by simple modifications. The bulk of the computation is taken up by the recognition search. The recognition search was sped up by a combination of three techniques: fast gaussian computation, two-pass search, and aggressive beam search pruning. Each of these will be described below.

4.1 Fast Gaussian Computation

We found that most of the computation in our continuous densit recognition system is in the Gaussian computation, i.e. finding the closest Gaussians to compute the output probability for the feature vector of each frame. Usually there are 256 Gaussians defined for each of the characters. We had developed a technique called "Fast Gaussian Computation" (FGC) to speed up the gaussian computation for the BYBLOS speech recognition system [15].

Our algorithm is a simplification of the algorithm presented in [16]. In the FGC method we first divide up the feature space into disjoint regions using a binary clustering algorithm. Typically, we might define 1024 regions. Given any feature vector, it is possible to determine the region that contains this vector using a binary search. Then, for each frame in the training set, we determine which region the frame is in, what gaussian was used, and which character the frame is part of. We then record in each region of the space, the list of all of the gaussians that were ever used for each of the characters.

During recognition, we first determine in which region the feature vector lies. This is done only once per frame. Then, when we consider a particular character, we only need consider those gaussians that have appeared within that region in the training data. Typically, the number of gaussians required is a small fraction of the full set. Thus the computation of gaussians is reduced by an order of magnitude without loss in performance. The entire system is sped up by a factor of three or more.

4.2 Two-pass Search

Our recognition search algorithm is a fast two-pass search, consisting of a coarse forward pass and a fine backward pass [10]. The forward pass computes the probability of each character (or word) ending at each frame in the input. The backward pass processes the line in the reverse direction using more detailed models. But whenever the search algorith transitions (backwards) to another character, it need onl consider those characters that were found to be likely to end at that frame of the input. The product of the forward and backward scores provide an ideal pruning score to minimize the computation.

The forward pass usually takes most of the time because it searches among all the character models while the backward pass searches only among the candidates selected by the forward pass. Since the purpose of the forward pass is only to find a list of likely characters, we find that it can be done using much less detailed models without any noticeable loss in the accuracy of the backwards pass. One useful feature developed for the BYBLOS speech recognition system is that the HMMs for phonemes (characters here for OCR) used in the two passes can be different. So we can use simple models in the forward pass to speed it up and use fine models in the backward pass to increase the accuracy. For example, we find that if we use 7-state discrete HMMs in the forward pass and 14-state HMMs *in the backward* pass, the computation in the forward pass is reduced dramatically, resulting in overall speed increase, while the accuracy is the same.

4.3 Pruning the Search

The goal of the recognition search is to find the character sequence that has the highest likelihood. The basic HMM search uses a time-synchronous "beam search" technique in which we discard any hypotheses that appear to be unlikely to result in the highest score. We find that we can often discard a large percentage of the hypotheses without any loss in accuracy. We found that our system could be tuned to discard a much larger number of hypotheses without significant loss.

4.4 Speed vs. Accuracy

Table 2 shows the results of the speed-up of a unifont Arabic OCR system. The speed-up factor of 50 was obtained with onl a loss of 0.17% absolute in CER. We obtained similar results in English. About half of the computation is taken up by the various preprocessing stages, and the other half is taken up by the recognition search. The resulting speed of about 45 characters per second is not extremely fast. But it is now within the range of usability and it is clear that it could be sped up by another small factor if necessary.

System	Speed (char/sec)	Char Error Rate (%)
Original	0.7	0.77
Fast	45	0.94

Table 2: Speed-up Results on an Arabic OCR syste

5. Chinese OCR

To further demonstrate the language independence of our approach, we extended our system to Chinese, which differs fro languages like Arabic and English in that the script does not have a small number of characters from which all words are constructed. Therefore, a training corpus, is unlikely to contain samples of all the characters that are expected in test data. Also, Chinese characters in general have very complicated structure, and it is not obvious from the outset that the simple models described in Section 2 are appropriate to model these complex characters.

5.1 Computer-Generated Chinese Corpus

In order to test the basic model on Chinese characters, 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 was 3,870. We printed and scanned 14 samples of each of the 3,870 characters in each of the four fonts. The characters occur in random sequences. Figure 5 shows a sample from the computer-generated corpus.

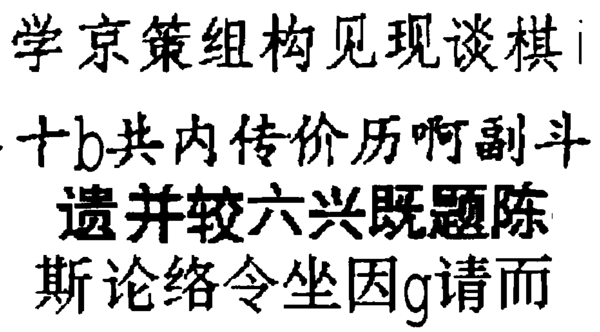
学京策组构见现谈棋 i
-十b共内传价历啊副斗
遗并较六兴既题陈
斯论络令坐因g请而

Figure 5: A sample of the computer-generated Chinese corpus

5.2 Unifont and Multifont Experiments

The first experiments were performed on the computer-generated 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 Arabic and English. 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 and the system could distinguish a large number of characters.

In order to get a sense of how different the four fonts were fro each other, we ran a cross-font experiment in which the syste was trained on three of the fonts and tested on the fourth. The result was a CER of 10%, indicating that 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 computer-generated data, we would have to collect training data from actual printed sources.

5.3 OCR on Real Data

We did not have any corpus of real images with transcriptions for Chinese. So we needed to collect our own corpus. But we did not want to transcribe a large corpus, because this requires substantial effort – especially in Chinese where the training set might need to be larger and where the characters are harder to type. Therefore, we looked for sources where we could scan an image and could also find a text version. We found several newspapers, magazines, and books that we could buy that were also on line. We decided to use the Chinese newspaper People's Daily as our first source. We collected a real printed corpus of 60,000. The corpus has only 2,600 unique characters and is mainly a unifont corpus with some minority fonts. Figure 6 is a sample from this corpus. As can be seen, this data contains significant variability.

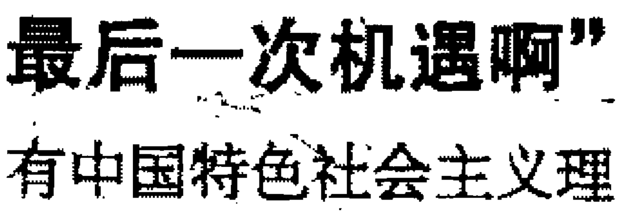
最后一次机遇啊"
有中国特色社会主义理

Figure 6: A sample from the printed Chinese corpus

However, the internet versions did not have linefeeds within a paragraph. So we developed a technique to determine the line-by-line transcription of the images from the paragraph transcriptions. The procedure starts by using an existing model of the characters (based on computer-generated images in this case) to recognize the characters in the images. Although the CER was quite high (about 10%), we found that we could align the recognized characters with the true characters in the online version automatically. We used the program that is normall used for aligning reference and hypothesis strings to compute error rates. Since we are only concerned with the location of the new line in the text version, we have no problem with substitution or insertion errors. The only uncertainty is when a character in the true text is deleted so we do not know which line of the image it is on. This only occurs on 5% of the lines, and we know that the problem has occurred so we can examine those lines or simply discard them from the training data. This procedure thus makes it even easier to create a training corpus for a new language, since we can determine the alignment of text to the lines of the image automatically.

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 just 16,000 characters for training, which covered only about 1600 unique characters. From the computer-generated data we used 3,870 unique characters, each having 14 training tokens in each of the four fonts. The test set contained 1225 unique characters, many of which did not appear in the real training data. Frequency weighted, 2% of the character samples in the test data had no training.

We found that the most effective way to combine the computer-generated training data and the real data was to simply train on both the data sets, with a larger weight on the real data. This way, any character with real data training is based mostly on the real data, while any character with no real training still has a model based on the computer-generated data. The CER results for several conditions are given in Tables 3 and 4.

Training Set	Language Model	CER
Computer	None	10%
Computer	Character trigra	7%
Computer + Real	Character trigra	1.2%

Table 3: CER for Chinese on newspaper data.

Number of samples of real training data	CER
0	5%
1 to 5	3%
> 5	<1%
All characters	1.2%

Table 4: CER as a function of amount of real data for each character when trained on a mix of computer-generated and real data.

We can see that the use of a trigram language model on characters reduces the error rate from 10% to 7% when the character models are constructed entirely from computer-generated data. When we add the training from real data, we see that the CER drops to 1.2 %, because most of the character samples in the test have at least one sample of training. Table 4 gives a breakdown of the CER as a function of the amount of real training data for that character. We see that even characters with no training now have a CER of 5%. The improvement from 7% is due to the fact that the overall error rate is lower, and frequentl the recognition is helped by an adjacent character being correct. Once we have even a few training samples, the error rate decreases. And with more training it is lower still. It is interesting to note that, in general, the Chinese characters have very low error rates. The characters with the highest error rates were punctuation and English characters.

This result shows that it is possible to achieve a low error rate on Chinese character recognition of real images even though we did not have training data for all of thc characters, and many of the characters had very little real training data.

6. Conclusions

In this paper, we presented several advances in the BBN BYBLOS OCR system, which can perform open-vocabular OCR on Arabic, Chinese, and English. The system is based on Hidden Markov Models and utilizes the same advanced technology that is used for speech recognition. Many of the changes were to incorporate the more advanced features of research speech recognition systems, including continuous-density HMMs, unsupervised adaptation, and fast search algorithms. We reported the improvement in Arabic and English OCR on the newly updated continuous-density system to sho that the continuous-density system improves significantly with a larger training corpus. We sped up the system by a factor of 50 without a significant loss in CER. The system is fast enough for practical use for Arabic and English OCR. Also we have extended the system to recognize Chinese with high accuracy on real newspaper data.

7. References

[1] 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.

[2] 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.

[3] J. Makhoul, R. Schwartz, C. LaPre, I. Bazzi, Z. Lu, P. Natarajan, "A Language-Independent Methodology for OCR," Proc. Symp. Document Image Understanding Technology (SDIUT97), Annapolis, MD, 1997.

[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] L. Nguyen, T. Anastasakos, F. Kubala, et al., "The 1994 BBN/BYBLOS Speech Recognition System", Proc. of ARPA Spoken Language Systems Technology Workshop, Austin, TX, Jan. 1995, pp. 77-81.

[6] 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. 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.

[7] 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.

[8] 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.

[9] S. Austin, R. Schwartz, P. Placeway, "The Forward-Backward Search Algorithm", Proc. of IEEE ICASSP-91, Toronto, Canada, May 1991, pp. 697-700.

[10 L. Nguyen, R. Schwartz, et al., "Search Algorithms for Software-Only Real-time Recognition", Proc. of ARPA Human Language Technology Workshop, Princeton, NJ, Mar. 1993, pp. 411-414.

[11 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.

[12 L. Nguyen, R. Schwartz, "Efficient 2-Pass N-Best Decoder", Proc. EuroSpeech '97, Rhodes, Greece, Sept. 1997, pp. 167-170.

[13 J. Makhoul, S. Roucos, and H. Gish. "Vector Quantization in Speech Coding," Proc. of IEEE, Vol. 73, pp. 1551-1588, November, 1985.

[14 R.A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems, " Annals of Eugenics 7, 179-188, 1936.

[15 J. Davenport, R. Schwartz, L. Nguyen, "Towards a Robust Real-Time Decoder," Proc. of ICASSP '99, Phoenix AZ, March 1999, p. II-645.

[16 M. Padmanabhan, E. E. Jan, L. R. Bahl, M. Picheny, "Decision-tree based feature-space quantization for fast gaussian computation", Proc. of 1997 IEEE Workshop on Automatic Speech Recognition and Understanding, Santa Barbara, CA, Dec. 1997, pp. 325-330.