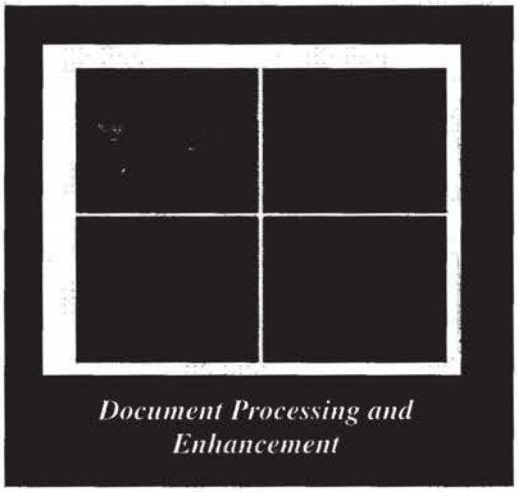
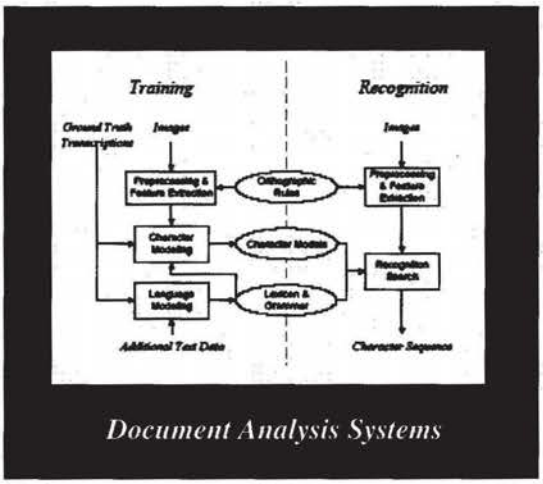
Proceedings

2005 Symposium on Document Image Understanding Technology



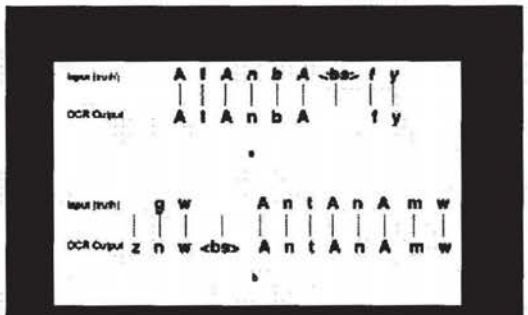
Document Processing and Enhancement



Document Analysis Systems

Word Shape Token	Word 1	Word 2	Word 3
0000 0001 0100 0000 0000 0000 0010	مناسب suitable	منتفع gainfully employed	
0000 0100 0000 0010	مكعب make cubic	يلاعب he jests	يلعب he plays
0000 0000 0000 0000 0001 1000 0100 0100 0000 0101	مستهلك customer of you		
0000 0000 0000 0000 0001 1000 1000 0000 0000	مستورد importer		
0000 0000 0000 0001 0000 0000 0000 0001 0000 0001 0000 0001 0000 0000 1001	ستحصلين you will be given a grant	ستستمتعين you will enjoy yourself	

Arabic Document Analysis



Extraction and IR

November 2-4, 2005
Inn and Conference Center
Marriott Hotel
Adelphi, Maryland

Proceedings

SDIUT05

The 2005 Symposium on Document Image Understanding Technology

Inn and Conference Center
Marriott Hotel
Adelphi, Maryland
November 2-4, 2005

Sponsored by: The United States Department of Defense
Advanced Research & Development Activity
Organized by: The Laboratory for Language and Media Processing
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742

A limited number of additional copies of these proceedings are available for $60 from:
University of Maryland
Institute for Advanced Computer Studies
College Park, MD 20742
Phone (301) 405-6444
Fax: (301) 314-9115
Email: sdiut05@lamp.cfar.umd.edu

Table of Contents

MESSAGE FROM THE ORGANIZERS 5

KEYNOTE SPEAKERS

GOOGLE PRINT 9
Luc Vincent, Google

EDUCE: ENHANCED DIGITAL UNWRAPPING FOR CONSERVATION AND EXPLORATION OF INACCESSIBLE TEXTS 11
W. Brent Seales and Yun Lin, University of Kentucky

SESSION 1 DOCUMENT ANALYSIS SYSTEMS

A FLEXIBLE EXPERIMENTATION AND CONFIGURATION PLATFORM FOR MULTILINGUAL HMM OCR 15
Ehry MacRostie and Premkumar Natarajan, BBN Technologies

DOCUMENT INFORMATION PROCESSING: TOWARDS SOFTWARE AND TEST COLLECTIONS 23
G. Agam, S. Argamon, O. Frieder, D. Grossman and D. Lewis, Illinois Institute of Technology and David D. Lewis Consulting**

DOCLIB: A DOCUMENT PROCESSING RESEARCH TOOL 31
Kevin Chen, Stefan Jaeger, Guangyu Zhu and David Doermann, Booz Allen Hamilton* and University of Maryland*

A PROCESS FLOW FOR REALIZING HIGH ACCURACY FOR OCR TEXT 39
Kazem Taghza, Julie Borsack and Tom Nartker, University of Nevada, Las Vegas

SESSION 2 DOCUMENT PROCESSING AND ENHANCEMENT

A DOCUMENT IMAGE ENHANCEMENT MODULE: PERSPECTIVE WARP CORRECTION 49
C. Monnier, Vitaly Ablavsky, Steve Holden and Magnus Snorrason, Charles River Analytics, Inc. and Boston University**

DOCUMENT RECOGNITION AND TRANSLATION WITH HANDHELD MOBILE DEVICES **55**
Esin Darici Haritaoglu and Ismail Haritaoglu, Polar Rain Inc.

SESSION 3 PROCESSING OF ARABIC DOCUMENTS

A SYSTEM FOR DISCRIMINATING HANDWRITING FROM MACHINE PRINT ON NOISY ARABIC DOCUMENTS **65**
John C. Femiani, Mariano Phielipp and Anshuman Razdan, Arizona State University

NLP-ENHANCED EXPLOITATION OF DEGRADED ARABIC TEXTS **71**
Michael Shalev, Encyclopaedia Britannica

INITIAL RESULTS IN OFFLINE ARABIC HANDWRITING RECOGNITION USING LARGE-SCALE GEOMETRIC FEATURES **79**
Ilya Zavorin, Eugene Borovikov and Mark Turner, CACI International Inc.

CLASSIFICATION OF MACHINE PRINT AND HANDWRITING IN MIXED ARABIC DOCUMENTS **89**
Karthik Sridharan, Faisal Farooq and Venu Govindaraju, CEDAR, SUNY at Buffalo

SESSION 4 EXTRACTION AND INFORMATION RETRIEVAL

ACCURATE DOCUMENT CATEGORIZATION OF OCR GENERATED TEXT **97**
Robert Price and Anthony Zukas, Content Analysis LLC and SAIC**

EFFECT OF DEGRADED INPUT ON STATISTICAL MACHINE TRANSLATION **103**
Faisal Farooq and Yaser Al-Onaizan, CEDAR, SUNY at Buffalo and IBM T.J. Watson Research Center**

MOBILE INTERACTIVE SUPPORT SYSTEM FOR TIME-CRITICAL DOCUMENT EXPLOITATION **111**
George Nagy and Daniel Lopresti, Rensselaer Polytechnic Institute and Lehigh University**

SESSION 5 ARABIC DOCUMENT ANALYSIS

HANDWRITTEN ARABIC WORD SPOTTING USING THE CEDARABIC DOCUMENT ANALYSIS SYSTEM **123**
Sargur Srihari, Harish Srinivasan, Pavithra Babu and Chetan Bhole, CEDAR, University at Buffalo, SUNY

WORD SPOTTING IN ARABIC SCRIPT DOCUMENTS 133
A. Lawrence Spitz and Jim Yaghi, DocRec Ltd.

CHALLENGES IN ADAPTING MACHINE-PRINT ARABIC OCR FOR HANDWRITING 139
Steven G. Schlosser and Robert C. Vogt, NovoDynamics, Inc.

SESSION 6 CROSS DOMAIN APPLICATIONS

ROBOT NAVIGATION TECHNIQUES FOR ENGINEERING DRAWING ANALYSIS 157
Thomas C. Henderson and Chimiao Xu, University of Utah

PERCEPTUAL ORGANIZATION IN SEMANTIC ROLE LABELING 167
P. Sarkar and E. Saund, Palo Alto Research Center

PICTOGRAPHIC RECOGNITION TECHNOLOGY APPLIED TO DISTINCTIVE CHARACTERISTICS OF HANDWRITTEN ARABIC TEXT 173
Mark A. Walch and Donald T. Gantz, The Gannon Technologies Group and George Mason University**

SESSION 7 DOCUMENT ANALYSIS APPLICATIONS

A DOCUMENT TRIAGE SYSTEM FOR ARMY APPLICATION 187
F. Fisher, K. Marcus, R. Chang and J. Turner, Army Research Laboratory

MULTI-LANGUAGE HANDWRITING DERIVED BIOMETRIC IDENTIFICATION 197
Donald T. Gantz, John J. Miller and Mark A. Walch, George Mason University and The Gannon Technologies Group**

TRADEOFF STUDIES ABOUT STORAGE AND RETRIEVAL EFFICIENCY OF BOUNDARY DATA REPRESENTATIONS FOR LLS, TIGER, AND DLG DATA STRUCTURES 211
David Clutter and Peter Bajcsy, University of Illinois

DEMONSTRATION AND POSTER ABSTRACTS

ABBYY SOFTWARE HOUSE OCR, FORMS PROCESSING AND DATA CAPTURE TECHNOLOGY REVIEW 225
Imelda Valenzuela, ABBYY

VERUS A MIDDLE EASTERN LANGUAGE OCR 229
Steven G. Schlosser, NovoDynamics, Inc.

IDG: A Business Information Extraction, Management, and Routing Front-End for Content Management Systems 231
Vikas Krishna, Savitha Srinivasan, Neil Boyette, Isaac Cheng, Jeffrey Kreulen and Tapas Kanungo, IBM Almaden Research Center

Fast Skew and Slant Correction for Arabic Written Word or Line 235
Essam M. Zaki and Mohamed El-Adawi, Sakhr Software USA Inc.

EVALUATION WORKSHOP ABSTRACTS

Performance Evaluation of Multilingual Document Exploitation Systems 239
Steven G. Schlosser, NovoDynamics, Inc.

Issues with Automatic OCR Evaluation and Metrics 241
Kristian J. Concepcion.

Evaluation Issues in Image Refiner 243
Kristen Summers and Eugene Borovikov, CACI

The Sporadic Nature of OCR Evaluations 245
Tapas Kanungo, IBM Almaden Research Center

Ground Truth Representation Used in Testing and Optimization of the Optical Word Recognition System 247
Mike Ladwig, Northrop Grumman Corporation

Metrics for Word Spotting 249
Sargur Srihari, CEDAR University of Buffalo

Evaluating with Informational Confidence 251
Stefan Jaeger and David Doermann, University of Maryland

AUTHOR INDEX 253

Message from the Organizers

Over the past 10 years, SDIUT has brought together thousands of researchers from academia, industry and government to continue to explore the unique needs and challenges facing the government with respect to processing a tremendous volume on hard copy material they encounter every day. With continued effort, and increased awareness of the need for such capabilities, we will be able to turn the ideas being discussed in this meeting, into meaningful solutions for the future.

This year, the Symposium will host over 20 technical talks, almost a dozen posters and demos and 2 keynote talks, followed by a one day workshop on the Evaluation of Document Image Processing Technologies, run by Ms. Jen Doyon of the Mitre Corporation. The workshop will bring together and focus experts in the field of document image processing on evaluation issues with the hope to making progress toward enhanced ground truth representation and corresponding evaluation techniques. Expanding on the character and word accuracy based metrics in current use facilitates a more complete approach to evaluating the performance of established recognition systems. It also enables component-level evaluation of specialized technologies such as script / language identification, line and word finding, signature detection, handwriting recognition, word spotting, forms processing and others. We hope that this session will help define solutions that can be incorporated into future government programs.

The organization of this Symposium would not have been possible without the endless hours of work by the staff of the University of Maryland Institute for Advanced Computer Studies (UMIACS) and in particular of Ms. Denise Best. Thank you Denise for your wonderful work.

Thank you once again for your participation and contributions to this Symposium.

The 2005 SDIUT Organizing Committee.
November 2, 2005

Keynote Speakers

GOOGLE Print

Luc Vincent
Google, Inc.

EDUCE: Enhanced Digital Unwrapping for Conservation and Exploration of Inaccessible Texts

W. Brent Seales **Yun Lin**
Computer Science Department
University of Kentucky
{seales, ylin}@netlab.uky.edu

This work addresses digital unwrapping of a class of damaged and fragile objects that are impenetrable for conventional imaging equipment and cannot easily be opened physically for a clear digitization. Existing document processing/analysis techniques are not able to reveal a considerable amount of the contents wrapped inside. We develop a general approach for building a readable image of an opaque, rolled or folded document from a non-destructive volumetric scan. The problem is framed by localizing, constructing and manipulating a texture image induced by a surface embedded in a 3D voxel space. Our approach is characterized as an iterative energy-based optimization framework. The goal is to form an optimal texture image by optimizing the surface geometry using a $2\frac{1}{2}$ Active Contour model and a novel texture-guided contouring algorithm. As one of the critical steps in the processing framework, a physically-based surface parameterization algorithm is used to map the convoluted embedded surface to an image plane which preserves angles, lengths and minimizes the distortion of text in the image. Initial experimental results on sample scrolls show the feasibility of this method, which we believe is necessary for scholars seeking to study inaccessible texts. Our vision is that these methods will lead to viable techniques for scanning everyday opaque objects such as books without opening them.

Computed Tomography (CT), a widely used technique in healthcare, is nondestructive, penetrating and revealing, which makes it a good candidate to use in folded document digitization. Our processing framework takes volumetric data from a custom CT scanner (with tunable X-ray energy and scan resolution) as input. The output data of a scan is treated as a cuboid of aligned voxels in x,y,z directions. Ink, document substrate and empty space can be distinguished from the intensity variations in the unstructured voxel set.

We build an optimization algorithm to correctly localize the substrate surface within the voxel set. The algorithm enforces two levels of optimization constraints. One is imposed by a $2\frac{1}{2}$ Active Contour segmentation model and the other, which we call texture-guided contouring, is based on texture image analysis which takes advantage of the surface parameterization that maps the textured surface mesh to an image plane. The steps of the algorithm proceed as follows:

1. Construct an optimal mapping M_1 from a volume V to a piecewise smooth surface S :

   $$M_1 : V \rightarrow S$$

   by minimizing the energy E_1, where E_1 is defined as:

   $$E_1(V, S) = I(V, S) + C(S)$$

   E_1 is a function of the volume V and an embedded surface S. $I(V, S)$ represents an Image component which evaluates how well the surface locates the intensity features contained in the volume. $C(S)$ represents a Shape component which evaluates the smoothness of the surface.

2. A filter is applied on V and S that produces a *nonregularized* texture map T_1:

   $$F : V + S \rightarrow T_1$$

   S is represented as a triangulated mesh and its triangular components map to R^2 independently, where the texture map T_1 is defined without inheriting any adjacency relationships.

3. A mapping M_2 is constructed from the surface S representing the document to a planar surface U, which converts the nonregularized texture map T_1 to a regularized texture map T_2:

   $$M_2 : S + T_1 \rightarrow U + T_2$$

   T_2 is expected to represent the "image" of the information carried on the surface of the document. We approximate a conformal mapping by using mass-spring system simulation.

4. Further optimization is enforced by minimizing the energy E_2 for T_2, which can be framed as:

$$E_2(T_2) = P(T_2) + R(T_2)$$

P is a generalized function evaluating low level image features while R is a generalized function based on high level document image understanding techniques. The surface is optimized iteratively through steps 2,3,4. During each iteration the surface position is updated along its normal direction.

5. Return U, T_2

We build a $2\frac{1}{2}$ Dimensional Active Contour model to construct M_1 in step 1. We take advantage of cross-sections to define an initial surface by approximating 2D salient contours in representative slices. Optimization follows and in this phase optimization is only performed based on raw volume data and the surface, ignoring the texture constraint for the moment. An optimal mapping is defined as minimizing the energy function E_1, which is a linear combination of an Image component $I(V, S)$ and and a Shape component $C(S)$. $I(V, S)$ calculates the gradient of the surface with respect to the volume, which attracts the surface to contours with large intensity gradients. $C(S)$ is evaluated as a combination of 5 components, which are first-order and second-order partial derivatives of the parametric surface S. These components correspond to continuity, curvature and twist. The surface S is approximated by a mesh decided by a finite set of points. We solve the optimization problem using a greedy algorithm, which is an extension of planar active contouring to $2\frac{1}{2}$ dimensions: points move on the discrete grid to any position within a defined planar neighborhood while the constrained energy functional is built in three dimensions.

The mapping M_2 that unwraps the embedded surface plays a significant role in the quality of the resulting texture. The goal is to introduce as little distortion as possible to the text that may be present in the document. The point is that the surface representing a document can have *arbitrary shape*. We approximate a conformal mapping by using mass-spring system simulation. A surface mesh, modeled as a mass-spring system, performs realistically when simulation principles are founded on physical laws. An optimal mapping is achieved when the system reaches equilibrium status with minimal potential energy. The mass-spring system strives to maintain the length of each edge while performing mapping by deformation. For a triangle, when its three sides are preserved, its three angles are also preserved. Since the spring system minimizes edge lengths globally, i.e., the length summation of all edges, it approximates but does not produce a strict conformal mapping.

This surface parameterization method has an important advantage over other methods: arbitrary surfaces can be handled. For surfaces that are rolled, folded, or wrapped, such as a scroll, we use constrained dynamics to perform an unrolling operation followed by flattening. The virtual unrolling "holds" one edge of the surface and allows the free portion to descend under simulated gravity.

We assume a correctly positioned surface will induce a texture map T_2 that contains both high and low frequency components. Further optimization of surface location is achieved by adding a novel texture constraint in step 4 that ultimately results in optimization directly over the texture. We adopt a sharpness measure for the texture image based on frequency domain analysis and entropy evaluation. After each loop, the surface S is updated along the normal in a limited neighborhood, yielding a new surface S'. A texture map T_1' is generated for S' in step 2 and a regularized texture T_2' by unwrapping and flattening S' in step 3. If its texture evaluation energy is reduced, the update S' is accepted. General low-level analysis is a starting point. Our framework is extensible by including the possibility to apply advanced pattern analysis based on a priori knowledge of the document's historical and cultural milieu.

We have implemented the key algorithms outlined above and have conducted a series of tests on replicas and experimental artifacts scanned using custom CT scanners. In the first testing phase, we scanned and processed regularly shaped canvas scrolls and irregular papyrus scrolls. Motivated by that success, we created a set of replicas from papyrus and ink and embedded them in polyurethane plastic. Two representative examples include a papyrus strip rolled and a papyrus strip connected into a Mobius band with ink (locally) on both sides. The results of applying our approach successfully reveals the material content without physically opening them. These successes demonstrate the potential application of our algorithms to more difficult but important cases such as ancient scrolls and damaged books. The post-processing algorithms presented here are a necessary though not sufficient step toward the goal of reading texts, books, and documents from volumetric scans. We anticipate improvements in scanning technologies in the areas of resolution and sensitivity to materials, which together with work such as ours will move forward the analysis of documents from penetrating scans.

Document Analysis Systems

A Flexible Experimentation and Configuration Platform for Multilingual HMM OCR

Ehry MacRostie **Premkumar Natarajan**
{emacrost,pnataraj}@bbn.com
BBN Technologies
Cambridge, MA

Abstract

In this paper we present the BBN Flex multilingual OCR model configuration and experimentation platform which allows non-expert users to configure and test HMM models for the BBN Byblos OCR system. The Flex platform is designed to provide customers with model training capabilities for any machine printed language. Furthermore, training capabilities can be updated as new modeling techniques become available. This paper describes the HMM OCR modeling paradigm and the design of the Flex platform. Modeling techniques currently available on the platform as well as new techniques that we plan to add are discussed. Finally, we describe a test of the system on Korean machine printed data in which we achieved a character error rate of 6.84%.

1 Introduction

One of the major challenges for Optical Character Recognition (OCR) systems is coping with the variety of data encountered in real-world settings. Document images may be printed by a number of different physical means in any number of different fonts. Digitization of the document by means of a scanner, fax or camera may introduce distortion and noise to the document. Binarization of color or grayscale digital images can further degrade image quality. Furthermore, all of these problems may be encountered in documents in languages for which no OCR system exists. In government settings such real-world problems are often more pronounced.

The BBN Byblos OCR system is a data-driven Hidden Markov Model (HMM) system that can be configured for a wide variety of languages and document degradation types [1,4,5]. The goal of the recent effort described in this paper is to simplify the model configuration process so that customers can easily configure the Byblos system for their data and language requirements. In addition, we wanted to create a system that allows us to easily deploy new model training techniques developed at BBN that increase the performance and speed of our OCR system.

This paper is organized as follows. In Section 2 we describe our basic HMM modeling paradigm for OCR. In Section 3, the Flex platform is described and some details regarding our use of the Unicode character encoding standard are discussed. Section 4 addresses HMM modeling techniques supported in the current version of this platform. In Section 5 we discuss some modeling techniques that we plan to incorporate into the platform in the future. In Section 6 we describe the results of testing the system on a corpus of Korean documents.

2 The HMM OCR Modeling Paradigm

In this section we briefly review the BBN Byblos OCR system. For a more detailed description the reader is referred to [1]. A pictorial representation of the system is given in Figure 1. In the figure, the ellipses represent script-dependent knowledge sources. The conceptual script-independent system components are identified by rectangular boxes.

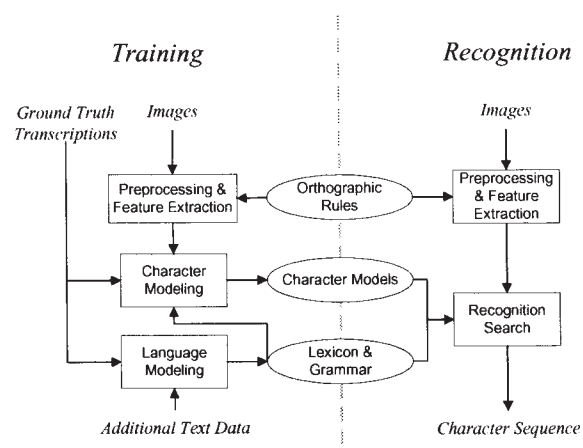


Figure 1: HMM OCR overview

The OCR system can be divided into two basic functional components: training and recognition. Both training and recognition share a common pre-processing and feature extraction stage in which we

first de-skew the scanned image and then locate the regions (bounding boxes) of individual text lines. We then compute a feature vector as a function of the horizontal position within each of these line regions. Each line of text is horizontally segmented into a sequence of thin, overlapping, vertical strips called frames. For each frame we compute a language-independent, feature vector that is a numerical representation of the frame.

The OCR system models each character with a multi-state, left-to-right HMM. As described in Section 3.3, the system handles languages with right-to-left, bi-directional and other rendering orientations by normalizing them to a series of left-to-right feature vectors internally.

Figure 2: Sliding window feature extraction

Each state of the HMM has an associated output probability distribution over the features. The number of states and the allowable transitions are system parameters. For our full-character experiments we use 14-state HMMs with the topology shown in Figure 3. While the number of states can be changed, the basic topology containing self-loops and state skips is the same for all configurations and all languages.

Training is performed using the Baum-Welch or Forward-Backward algorithm that aligns the feature vectors with the character-models to obtain maximum likelihood estimates of HMM parameters. The state output probabilities of the HMM are modeled as mixtures of Gaussians. The HMM parameters estimated during training are the means and variances of the Gaussians, the mixture component weights, and the state transition probabilities. During recognition we search for the sequence of characters that is most likely given the feature-vector sequence and the trained character-models, in accordance with the constraints imposed by a lexicon and/or a statistical grammar. The use of a lexicon during recognition is optional but its use generally results in a lower Character Error Rate (CER). The lexicon is estimated from a suitably large text corpus. Typically the grammar (language model), which provides the probability of any character or word sequence, is also estimated from the same corpus.

A significant advantage of HMM-based systems is that they provide a language-independent framework for training and recognition. At the same time, they do not require the training data to be segmented into words or characters – they automatically train themselves on un-segmented data.

Figure 3: Left-to-right HMM topology

This data-driven approach allows custom models to be trained for new languages or domains with minimal data preparation and transcription costs. The Flex platform allows non-expert users to perform training on-site with their own data.

3 The Flex Platform

3.1 Overview of the Flex Platform

The Flex platform consists of three main components, the Byblos model training toolkit, the Condor High Throughput Computing system (from the University of Wisconsin), and a Microsoft ASP.NET Web User Interface (UI) layer. The entire system runs on a single Windows 2000 Server machine providing access from many computers via its web interface.

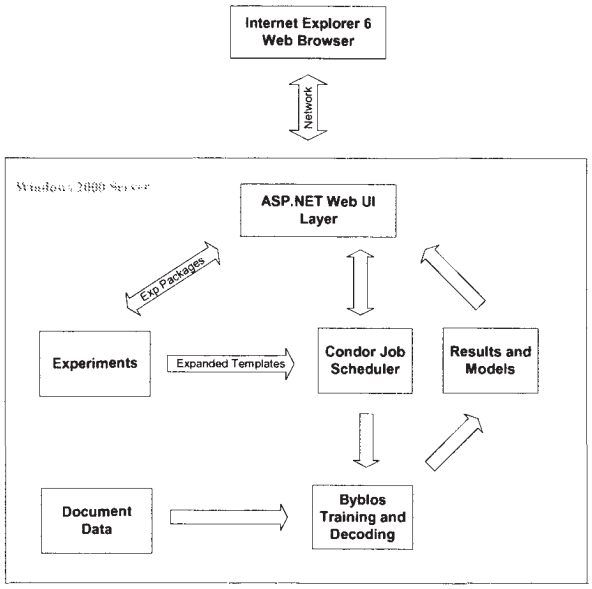


Figure 4: Platform components

The Condor and ASP.NET components are standard off the shelf software packages that provide the parallel processing and application layer support. Condor provides the parallel processing environment for the system. Though the current version of the Flex platform runs on a single machine, it can easily be configured to include a number of parallel compute nodes. The application logic is implemented in an ASP.NET layer. This layer manages experiments, exposes selected Byblos parameter values to the user, and displays recognition results. The Byblos OCR training suite is a collection of programs developed at BBN that includes

all of the HMM OCR model training and recognition functionality depicted by the rectangular boxes in Figure 1.

The platform infrastructure is built around logical units called experiments. An experiment encapsulates one type of high-level process and typically contains a number of sequential and/or parallel steps. In their generic form, all experiments consist of a central script that controls the sequence of programs to be executed, an XML parameter file that links the user interface to the experiment, and a number of generic parameter files that contain runtime parameters for the programs. The user interface is designed so that any number of experiments can be dropped into the system without any reconfiguration. Many common tasks such as splitting a corpus into test and training sets and converting transcriptions into a format that is usable by Byblos have been implemented within the experiments. The goal is to not require the user to perform any data manipulation other than loading the data onto the server.

3.2 User Interface

The system user interface is designed to be as simple as possible while at the same time enabling the user to easily see what is happening in the training process. We have implemented several new visualization tools that allow the user to easily view the status and progress of the training process.

For each new data set, the user first selects the appropriate type of training. This selection requires some knowledge of the training techniques available on the platform. A selected experiment type is then configured using a standard HTML form.

The parameters available using the user interface vary depending on the experiment type. Some common parameters include the data corpus location, the language being processed, and the image resolution of the document images being processed. Character modeling experiments allow the user to select whether or not to train a new Linear Discriminant Analysis (LDA) matrix and whether the reference transcriptions should be normalized using the Unicode NFKC normalization type. For the sub-character experiment, the user must provide the total number of sub-characters to use for training as described in Section 4.4.

Once the experiment is configured, it is then submitted for processing by the Condor system. As jobs run on the queue, their progress can be monitored using a job monitor tool.

One of the new visualization tools developed for this platform was the experiment explorer. The experiment explorer, shown in Figure 5, displays the parameter files and log files for each step in the experiment. It provides a graphical view of the experiment structure as well as a convenient way to monitor and debug an experiment.

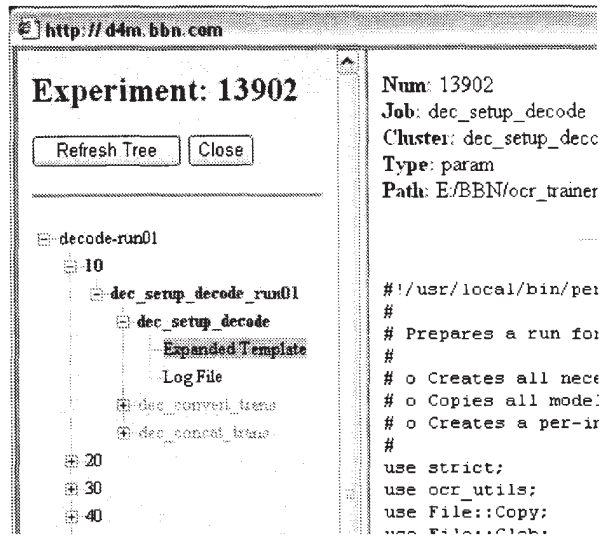


Figure 5: Experiment explorer display

After training, the models are evaluated against a test set. The test results are displayed in a results viewer tool shown in Figure 6. In the results viewer, the original line image, the corresponding recognition result, and the reference transcription are displayed together. The result and reference transcriptions are aligned and color coded so that errors can easily be seen. Line-by-line, page-by-page, and overall error rates are also displayed.

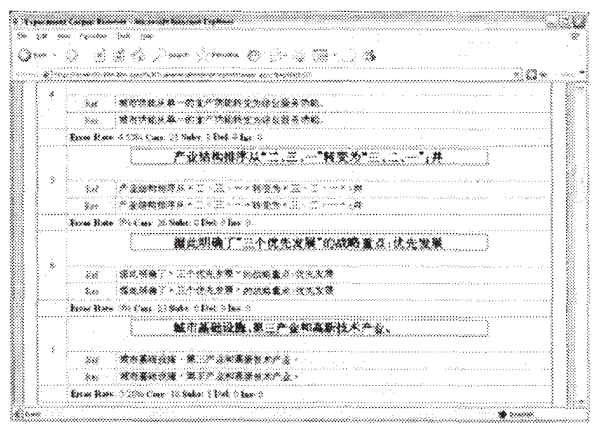


Figure 6: OCR recognition results display

3.3 Data Requirements

In the Flex platform we assume that the user will provide a corpus of document images and the corresponding line-by-line transcriptions.

A critical benefit of our HMM OCR technology is that it does not require manual character-level segmentation of document images for training. Also, line locations are found automatically in black and white input images using our HMM line finding method

[1]. This greatly minimizes the effort required to transcribe a training corpus.

We have standardized our OCR system on the Unicode character encoding standard in order to support a wide variety of languages and to facilitate easy integration with other downstream processes. Even within the Unicode standard there are a number of character-rendering issues that our system must deal with.

One of the most common issues is the reading direction of text. While most Western languages are read from left to right, Arabic and some other Middle Eastern languages are read right to left. Our system accounts for this by reversing the direction of feature extraction to match the reading direction and training models in the usual way. This requires a priori knowledge of the language of each document.

Reading direction becomes a little more complex when bi-directional text rendering [8] is taken into account. Unicode is always encoded electronically in speaking order. When a line of text contains right to left text interspersed with left to right strings, the left to right strings are rendered as such within an overall right to left line.

For our modeling purposes in order to align the feature vectors with their corresponding Unicode transcriptions, we reverse the underlying Unicode speaking order for embedded bi-directional strings temporarily for modeling and recognition then map them back to the standard Unicode ordering in the recognition output.

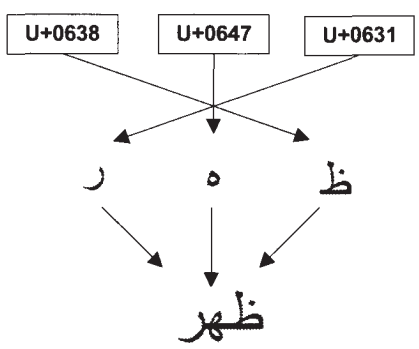


Figure 7: Right-to-left rendering in Arabic

There are a handful of characters that are rendered out of sequence with the general rendering direction of the script from which they come. An example of this is the Devanagari vowel sign 'I' (encoded as U+093F in Unicode) used in Hindi and other Indic languages. When it appears to the right of a consonant in the character encoding, the 'I' is rendered to the left of the consonant even though standard rendering for Devanagari is left to right. An example of out-of-sequence character rendering is shown in Figure 8.

A related, though more frequent character rendering issue occurs in languages such as Thai and Hindi. These languages have vowels and diacritics that are rendered in the same horizontal space as surrounding consonants. Since our feature extraction procedure proceeds in one direction using a fixed-height window, features from multiple characters are folded into the same feature vectors when two characters are rendered in the same horizontal space.

In order to deal with cases of overlapping and out-of-sequence character rendering, we take a combinatorial approach. Instead of detecting vertical or out-of-sequence rendering during decoding, we detect these cases in the training transcriptions and create unique models for each one. In practice this is done by clustering adjacent characters in the training transcriptions based on rendering information. The clusters are then used as the base modeling units.

Using such clustered modeling units, on a corpus of clean scanned Hindi documents, we have achieved character error rates of 1.4% using the full-character model configuration [5].

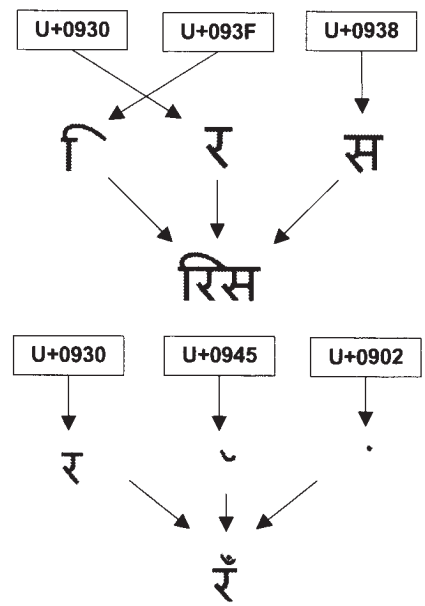


Figure 8: Vertical rendering and out-of-sequence rendering in the Devanagari script

4 Supported Modeling Techniques

This section describes modeling techniques that are currently supported by the Flex platform. Full-character modeling is the base modeling technique that usually provides the highest decoding accuracy. Several other modeling configurations that increase decoding speed over the full-character configuration are described; these include Fast Gaussian Trees, Tied Mixture Models and Sub-character models.

4.1 Full-character Modeling

Character Tied Mixture (CTM) training is the basic full-character training method used by the OCR training system. It is the preferred training method for most alphabetic scripts. For scripts with large numbers of

unique characters, sub-character modeling offers significantly faster decoding speeds.

The CTM training method trains a separate HMM for each unique character in the training corpus. Typically we use 14-state, left-to-right HMMs with self-loops and one-state skips. A codebook of 256 Gaussians is associated with the HMM for each character and there is no sharing of Gaussians across different characters.

CTM models usually produce the most accurate decoding results and the slowest decoding speed. Apart from the model parameters, decoding speed for CTM models depends on the total number of unique glyphs that we choose to model for a particular language. These glyphs are usually simply the characters in the target language, but in the case of languages with vertical character rendering some glyphs are combinations of characters.

4.2 Fast Gaussian Trees

Broadly speaking, the time spent during recognition is spent either in computing the HMM score (i.e., evaluating the Gaussian likelihoods) or in executing the search logic itself. In a basic HMM system, typically the Gaussian computations tend to dominate the overall computation time. One efficient method for reducing the Gaussian computation time is to first select (using a fast computationally cheap procedure) a preferred list of Gaussians for each feature vector and then only compute the likelihoods for those preferred Gaussians. Our implementation of this technique is referred to as Fast Gaussian Computation (FGC) trees or just Fast Gaussian Trees. For each feature vector, we walk down the FGC tree to find the preferred list of Gaussians for that vector.

FGC trees are automatically estimated from training data using trained CTM models. They do not alter the base CTM models and can be turned off during recognition.

While the decoding speed is faster than straight CTM models, recognition accuracy is typically a little worse using FGC trees. One important feature of the FGC tree architecture is that we can trade-off accuracy for speed by adjusting the length of the list of preferred Gaussians; the shorter the list, the greater the gain in speed and (typically) the greater the loss in accuracy.

4.3 Tied Mixture Modeling

Another way to reduce the time spent in Gaussian computations is to share Gaussians across different characters. We call this sharing approach the Tied Mixture (TM) model. In the TM model, instead of using a separate set of Gaussians for each character model, a single set of Gaussians is shared among all models. Mixture weights for each model are used to distinguish models from one another. As a result, only one set of Gaussian likelihoods needs to be evaluated for each input feature vector. In the standard CTM decoding, we use a separate set of 256 Gaussians for each character HMM, whereas in the TM mode we typically use a single set of 512 Gaussians. This is a huge reduction in the number of Gaussians in the system. As is to be expected, this reduction in the total number of parameters leads to a worse performance but once again the trade off between accuracy and speed can be controlled by adjusting the number of Gaussians in the TM model.

Our recognition engine uses a patented two-pass beam search. The goal of the first pass is not to generate the single-best hypothesis; rather, the goal is to prune out as many unlikely hypotheses as we can without rejecting the correct answer. The pruned hypothesis list is the starting point for the second pass where more detailed models (e.g., CTM models) can be used. We have found that the TM models are most effective method when used in the first pass in combination with CTM models in the second pass.

As with FGC trees, combining TM and CTM models increases decoding speed at a small cost to decoding accuracy. FGC trees and TM models can be used jointly to obtain a greater increase in speed than either approach can deliver when used separately.

4.4 Sub-Character Modeling

Unlike the FGC trees and TM models, the primary aim of the sub-character modeling approach is to reduce the time spent in the search logic. For large character sets, the fan-out at each node in the search network is also large. As a result, for languages with large character sets, the time spent in the search procedure (i.e., propagating scores and hypotheses from a node to all following nodes and sorting node scores at each time instant), can become a large fraction of the total compute time – especially when FGC trees and TM models are used to reduce the Gaussian computation time.

The sub-character configuration is based on the observation that when looking at a group of characters closely, many sub-regions are similar or identical across different characters. For example the first half of the character '**o**' and the first half of the character '**c**', look virtually the same, and therefore should not require different models. This similarity is exploited by modeling similar sub-characters with a single model then chaining these models together to model the full characters. Typically, each sub-character model is a 3-state HMM and a full character is spelled using up to five sub-characters to create a 15-state model.

In keeping with our overall approach of not requiring language experts, the list of sub-characters and the spelling of each full character using these sub-characters are both performed automatically using a clustering procedure. The optimal number of sub-characters could vary depending on the language. We

have used sub-character models for Chinese and Japanese data sets containing approximately 4000 and 3000 full characters, respectively. For these data sets, we have found that 1024 sub-characters is the best configuration [4].

5 Planned Improvements

Duration modeling and Adaptation are two modeling techniques that have been developed at BBN but have not yet been integrated into the Flex platform. Duration Modeling increases decoding speed (without any loss in accuracy) by adding a temporal constraint for each character. Adaptation allows existing models to be tuned for a new domain by using a small amount of training data from that domain.

5.1 Duration Modeling

The speed improvements described above involved reducing the number of Gaussians computed during character recognition. Another approach that has been explored is to use a constraint on the length of a character to speed up the forward pass of decoding. This is known as Duration Modeling [6]. For the purpose of duration modeling, the duration (or length) of a character hypothesis is simply the number of frames that are aligned with that hypothesis. Please refer to Section 2 for a description of frames.

As described earlier, the Byblos recognition engine uses a two-pass search algorithm. In the fast match pass, unlikely hypotheses are pruned from the search space of hypotheses. The goal here is to remove as many wrong hypotheses as possible before the detailed match pass. Duration Modeling adds a constraint on the length of a character during the fast pass match to prevent new hypotheses from being generated at improbable locations. For example, the duration model would prevent a character hypothesis that has one-tenth the duration observed in training.

Character durations are modeled as simple histograms on the durations observed during training. We use a separate histogram for each character. The trained histogram is then smoothed using an appropriate smoothing window, and the smoothed histogram is used to apply a hard constraint in the search algorithm.

Experimental results on Chinese data have shown that Duration Modeling can improve decoding speed by 37% without decreasing decoding accuracy [6].

5.2 Adaptation

Another potential addition to the Flex platform is character model adaptation. Often models exist for a particular language of interest; however the characteristics of the target domain of documents to be processed do not match the characteristics of the training corpus that was used to train the existing models. This mismatch often results in degraded performance. For example, the models may be trained on data from newspapers and magazines while the domain of interest might be photocopied, faxed, or camera-captured documents.

Adaptation is a method that allows us to use a small amount of new training data from the domain to tune the parameters of existing trained models to that domain. Previous work at BBN [7] has shown that Maximum Likelihood Linear Regression (MLLR) adaptation can provide significant improvements in performance.

6 A Case Study: Training Korean Models

During development, the Flex platform was continually regression tested using a number of data corpora that we have used in the past for model training. These corpora included data sets from Hindi, Arabic, Thai, Chinese and Japanese. Once the initial version of the platform was developed, we wanted to test the system on a completely new corpus/language to verify that our end-to-end training procedure could handle a new language.

We tested using a corpus of 1054 clean Korean images generated electronically from text. Before this test, we had never trained our system on any Korean data.

The corpus was not manually inspected ahead of time. The black and white documents and the corresponding UTF-8 transcriptions were simply loaded into the system and the standard procedure was used to validate the data and to run the model training process. The system automatically split the corpus into training and test sets – 811 documents in training and 243 in test. The ratio of training to test is a configurable parameter available to the user so these amounts could have been different. A total of 1525 units were modeled including individual Unicode code points and character clusters as described in Section 3.3.

The first experiment we ran was full-character training. This took approximately 18 hours of processing time on our dual-processor 2.6 GHz machine but only about 30 minutes of setup time for the user. The experiment ran through to completion without error. We ran recognition followed by optimization of the decoding weights then decoded again using the new weights. The character error rate on the fair test set using the CTM models was 6.84%.

Experiment	Character Error Rate
Full-character (CTM)	6.84%
CTM plus TM &FGC	7.41%
Sub-character	7.52%

Table 1: Error rates for clean Korean images

The standard procedures were followed to train FGC trees, TM models, and CTM models. Additionally, we used the standard procedure to train sub-character

models. The recognition results for all of these experiments are shown in Table 1.

7 Conclusions

The Flex OCR model configuration platform greatly simplifies the creation of new models for the BBN Byblos OCR system. It will allow non-expert users to train OCR models in their own facilities on their data. The platform includes a number of new tools for visualizing various aspects of the model training process. Updated modeling techniques that provide better performance and recognition speed can easily be added to the platform as they become available.

The BBN Byblos OCR system is now standardized on the Unicode character encoding system which allows the system to support a wide variety of languages.

Though this platform was developed for OCR modeling we believe that the distributed computing environment and the user interface can be easily re-configured to support other similar tasks.

Acknowledgements

This work was funded by the Army Research Laboratory through a LASER ACTD effort. We would also like to acknowledge the support and encouragement of Mr. Luis Hernandez of ARL.

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, 15:1, 2001, pp. 43–63.

[2] Z. Lu, R. Schwartz, P. Natarajan, I. Bazzi, and J. Makhoul, "Advances in the BBN BYBLOS OCR System," Proc. of Intl. Conf. Doc. Analysis and Recognition, Bangalore, India, 1999, pp.337–340.

[3] P. Natarajan, R. Schwartz, M. Decerbo, T. Keller, "Porting the BBN BYBLOS OCR System to New Languages," Symposium on Document Image Understanding Technologies, 2003, pp. 47–52.

[4] E. Macrostie, P. Natarajan, M. Decerbo, and R. Prasad, "The BBN Byblos Japanese OCR System," Internamtional Conference on Pattern Recognition, Cambridge, UK, Aug. 23-24, 2004.

[5] P. Natarajan, E. MacRostie, M. Decerbo "The BBN Byblos Hindi OCR System," Document Recognition and Retrieval XII, Proceedings of SPIE Vol. 5676, 2005, pp 10-16.

[6] P. Natarajan, R. Sundaram, R. Prasad, E. MacRostie, "Character Duration Modeling for Speed Improvements in the BBN Byblos OCR System," International Conference on Document Analysis and Recognition, Seoul, Korea, 2005, pp. 1136-1140.

[7] P. Natarajan, et. al., "Robust OCR of Degraded Documents" International Conference on Document Analysis and Recognition, Bangalore, India, 1999, pp. 357-361.

[8] Davis, M. "The Bidirectional Algorithm," Unicode Standard Annex #9, Revision 15, March 2005.

Document Information Processing: Towards Software and Test Collections

G. Agam, S. Argamon, O. Frieder, D. Grossman
Illinois Institute of Technology
Department of Computer Science
10 W. 31st Street, Chicago, IL 60616

D. Lewis
David D. Lewis Consulting
858 W. Armitage Ave., #296
Chicago, IL 60614

Abstract— **Analysis of large collections of complex documents is increasingly critical to forming accurate and precise intelligence appraisals of enemy activity. Complex documents include handwritten notes, diagrams, and graphics in addition to printed and formatted text. Collections may comprise many different types of complex documents and data, hence extracting useful intelligence from such collections is difficult and time-consuming. Existing automated analysis and search tools are of minimal help. Point solutions exist for some relevant subproblems, but most of these systems are specialized to work with only a single type of information, and in some cases are specific to a particular genre.**

We describe issues surrounding our ongoing development of an *integrated* system supporting complex document information processing (CDIP). When completed, the system will support: (a) segmenting and analyzing complex document components to produce data representations supporting querying and data mining, (b) importing such analyzed documents into database structures that allow efficient querying and data mining, (c) answering complex queries against information spanning multiple kinds of complex document data, and (d) discovery of patterns and relationships in complex collections, optionally guided by user queries. We also discuss our initial efforts to build a complex document test collection for evaluating both our system and others that may appear in the future.

I. Introduction

In the modern geopolitical climate, very large collections of complex documents are increasingly important to form accurate and precise intelligence appraisals of enemy activity. Such collections are rife, including many documents recently captured in Iraq and Afghanistan by U.S. forces, those captured in Ramallah by Israeli forces, and more. Effectively extracting useful intelligence from such large collections is currently an incredibly difficult and time-consuming task, since traditional automated approaches are not applicable. Analysts currently dealing with complex document collections must manually sift through enormous and jumbled collections of typed and handwritten texts, diagrams, lists, and more for useful facts. For instance, in the hunt for Saddam Hussein a team of military officers laboriously scoured thousands of complex documents by hand and gradually built a complicated multicolored chart showing links between people and organizations, to make sense of the enormous mass of information confronting them. As no integrated automated tools currently address this problem, there is a growing and clear need for computer systems for *complex document information processing* (CDIP). This paper describes our efforts towards development of a useful CDIP system.

We consider *complex documents* to include handwritten notes, diagrams, and graphics in addition to printed and formatted text. Furthermore, collections can consist of multiple types of documents; as a simple example, the Enron collection contains e-mails, policy documents, business statements, and more. Currently, no extant systems focus on complex documents; rather various point solutions exist for subproblems. Solutions exist for OCR of document images – solutions exist for recognition of entities in text – solutions exist for handwriting recognition – but *no* solutions exist that integrate all needed methods for processing complex documents.

The essential problem is that *existing point solutions take no regard for the entire complex document problem.* Research progress is also slowed by lack of any accepted complex document test collection. This paper describes our work on developing an architecture that supports complex document information processing, building a complex document collection for evaluation, testing our architecture using core (existing) point solutions, and improving overall performance by identifying bottlenecks and developing new point solutions or integration techniques, as needed.

A. Background

Though no integrated CDIP system exists, to the best of our knowledge, point solutions do exist for all key components of the complex document problem. Some are addressed by NIST-funded data collection and com-

petitive evaluation efforts such as TREC, CLEF, NTCIR, and MUC. They include:

- optical character recognition (OCR)
- extracting document structure
- handwriting and signature recognition
- extraction of tables/lists and their data
- finding/interpreting figure captions
- search/retrieval of images and graphics
- cross language information retrieval
- entity and structured information extraction
- document clustering

However, such systems are specialized—each only works with a single type of information, and is often specific to a particular genre (such as news stories). Thus, when faced with complex documents, users must partition the data manually and use multiple systems in a piecemeal fashion, manually collating results. Three main problems arise when using such systems piecemeal for CDIP.

First, when a specific system is unaware of the larger context of the information which it is given, its results may be degraded, as well as incomplete. For example, if a printed-text OCR system is given a printed page with overwritten handwriting, it will likely give a high error-rate on the printed text, as well as possibly producing extra, spurious, recognition results for the handwritten portion. An integrated system that first separates printed text from handwriting and processes them separately would do much better.

Second, when multiple information processing systems are used, where each system has no awareness of the others, the difficult tasks of collating, cross-checking and determining consistency of information items must be done by hand. This is both expensive and error-prone when dealing with very large numbers of documents.

Third, component systems normally require the determination of operation parameters which are set to optimize performance. Optimizing the performance of individual components, however, does not guarantee optimal performance of the overall system.

B. Project goals

Today's fragmented solutions offer text, structured data, and XML search, but lack a unified search regardless of the inherent structure of the data. Today, analysts are forced to search for key pieces of evidence by hand, log them into a journal, and then gradually identify connections among information items. We are currently developing an integrated suite of software tools targeted at *complex document information processing* (CDIP). These tools will ultimately support:

- Posing of complex queries against information spanning structured, semi-structured and unstructured data in a complex document collection,
- Analysis of each component of a complex document collection with appropriate techniques (e.g., document analysis for scanned text, language processing for online text image analysis and OCR output, statistical transformations for numerical data) to produce attributes that support querying and data mining,
- Importing of analyzed documents into database structures that allow efficient querying and data mining, and
- Discovery of patterns and relationships in complex collections, optionally guided by user queries.

All tools are being implemented using a modular design supporting a variety of applications; our goal is ensuring that such tools are suitable for use with existing analytical tools. We focus on enabling users to effectively access *all* the information contained in a complex document — not only the image portion, or only the text portion, or the only numerical portion.

To optimize the overall performance of the system we use an integrating principle of a probabilistic document model which integrates, in a principled manner, interpretations and reliability estimates thereof from all system components. In this way we intend to meliorate, if not overcome, the risk of error propagation. In addition, this approach enables "goal-directed interpretation" in which the process of responding to user queries can initiate reinterpretation of document image data.

The remainder of the paper describes our vision for useful analytic CDIP tools in more detail. In the following section, we describe issues in CDIP system design, including its functional requirements and system architecture. We then deal with the involved question of how to evaluate CDIP technologies in general, and our plans for the building the first extant complex document test corpus and how it will be used to evaluating and developing the system further. We conclude summarizing our current state of development and further issues for exploration.

II. CDIP System Design

A. Functional requirements

The focus is on developing a prototype tool suite implementing three core CDIP technologies of *text analysis*, *integrated retrieval*, and *data mining*. Specific attention is being paid to modular design, to ensure that

the developed software modules will easily integrate into different task-level applications.

Text analysis augments text documents and components with canonical class labels or structured interpretations. This eases querying, since the user can sometimes refer to labels from a finite set rather than anticipating all vocabulary variations. It also aids data mining, by providing sets of attributes that are smaller, less ambiguous, and more task specific than natural language words and phrases. We currently concentrate on two key text analysis technologies: text categorization [1] and stylistic analysis [2], [3].

Integrated retrieval from different kinds of data sources is a key mediator function—our IIT Mediator system currently supports a variety of traditional data sources such as unstructured text, semistructured XML/text data, as well as structured database querying [4]. A rule-based source selection algorithm selects those data sources most relevant to an information request, enabling the system to take full advantage of domain-specific searching techniques, such as translation of a natural language request into a structured SQL query. Results are then fused into an integrated retrieval set [5]. Such integrated domain-specific search is unique, to the best of our knowledge, in the current state of the art. The prototype builds upon this foundation to allow sophisticated querying of information extracted from complex documents at previous levels. Spatial relations between document elements, such as adjacency, are be straightforward to encode and query, as are keywords denoting topic, style, and graphic categories. This will enable searching for clusters of related data elements, as in queries such as "Find three orders mentioning attacks on different countries, all signed by the same person."

Data mining can leverage the fact that the IIT Mediator already indexes many different kinds of data using a relational database model, allowing straightforward integration of structured and unstructured data. We are developing tools that will augment this structure with stylistic, graphic classification, and template-based data. This approach allows application of traditional data mining and machine learning methods to discover relationships between different data such as association rules [6] and document clusters [7]. We will further develop routines to find correlations in document descriptors (for example, possible relationships between the author of a document and particular language styles).

These constitute relatively robust, broadly applicable technology sets for whose elements a number of existing implementations are available. They also produce outputs particularly compatible with other components of the system. In the near future, we will also examine incorporation of richer, but more domain-specific, technologies such as named entity recognition and information extraction.

B. Processing Modules

The prototype architecture is designed as a generic framework for integrating component technologies with appropriate APIs and data format standards to allow 'plugging in' different subsystems for performing component tasks. We emphasize that, throughout, software development work in the prototype is on interfacing and integrating existing available components, rather than on developing new ones.

Image analysis in the prototype comprises modules for low-level image processing, which segment different components of a complex document. Among these processes, printed text is separated from handwriting tables, and graphics, signatures and logos are identified, characters and words in printed text are segmented from each other, and features characterizing these different components are extracted for higher-level processing. Additional modules are included to determine spatial relations between document image components for: geometric analysis of document structures, parsing of tables and lists, and finding geometric associations.

Symbol analysis consists of processes for extracting symbolic content and relations from processed image data. This symbolic content can serve as input to the information database and conceptual analysis functions which provide knowledge-level analyses (retrieval, data mining, reliability assessment) using these primitives. Symbol analysis covers a range of different functions, including letter/word identification (i.e., OCR), handwriting attribution, classification of maps and diagrams, and parsing of tables and lists in the document. While some of these functions are more language-based and others more graphically-based, in the system these processes will interact via a probabilistic document structure model.

High-level information access and analysis functions consist of information retrieval from a collection's heterogeneous data and data mining to find interesting and meaningful patterns in that data. Retrieval in the prototype is based around IIT Mediator technology maintains a database of extracted information and handles user requests. The functionality at this level is integrated retrieval from heterogeneous data extracted from complex documents. As well, we are studying and

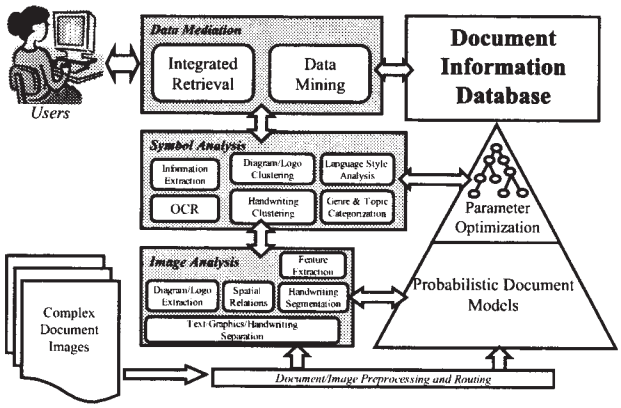


Fig. 1. Architecture of the full envisioned CDIP prototype.

working on techniques for pattern detection on complex document collections. Our main goals are: extracting information, finding relational and temporal links and patterns, and organizing this information via a usable interface. While such tasks have been addressed previously in other contexts, and some have made some use of speech recognition and OCR on news data, the challenges are much greater with diverse, complex material from multiple sources. For instance, it may be necessary to maintain multiple ambiguous interpretations in the output of handwriting recognition and OCR, which can stress data mining and pattern finding techniques. Thus as our project progresses, scalability in the face of large and ambiguous datasets will be a key ongoing consideration.

C. Software architecture

A key component of our approach is to tightly integrate the processes of document image interpretation, symbol extraction and relation, information retrieval, and data mining. An integrated approach promises increased reliability for all of these processes. Constraints on image interpretation, based on data consistency concerns at the high-level, can improve reliability of image interpretation. Similarly, gaps in the database can potentially be filled in at retrieval time, by reinterpreting image data using top-down expectations based on user queries. Our envisioned architecture for the complete system is given in Figure 1. It is divided into three main levels, each comprising several modules: *image processing*, which segments and extracts structures from images (for OCR we will integrate existing software in the obvious fashion), *symbol analysis*, which performs linguistic and classification analysis of extracted information for annotation in the database, and *data mediation*, which includes integrated information retrieval and data mining using the complex document information database constructed by the first two levels.

Image processing for CDIP must separate out different image components from one another, prepare them for further processing (OCR and classification), and annotate geometric relations between them. Separation of text from graphics and handwriting, as well as geometric normalization of each component, can be done using directional and regulated morphological operations [8], [9]. Both global and local features of each component are extracted, for use by symbol analysis (see below). Spatial relations between components are classified and then used to construct a document structure model which can provide useful annotations in the document database (such as that certain components are likely titles, or logos, or signatures, for example). Key will be the use of knowledge-based constraints at this level to help constrain the interpretative processes. For example, probabilistic document structure models may be used which give probabilities of image-level features, conditioned on structural labels for image regions [10]. Dynamic programming can then be used to solve for a maximum likelihood structure under the model. If multiple possible document structures are allowed, the highest probability model will be chosen for each given document.

The main function of the symbol analysis layer is to annotate document component data (both graphic and textual) with meaningful metadata which enables more effective utilization at the mediation level. Diagram clustering [11] will provide symbolic labels for similar graphics in the corpus, allowing linking of different documents with the same symbols or logos. Manual annotation can easily be added as well to make such labels meaningful to human analysts. Similarly, handwriting and signature analysis will be used to identify the authors of handwritten notes (either by recognition based on known samples, or by clustering to determine that different samples are from the same hand). Too, the linguistic style of textual components (extracted by OCR) will be analyzed [2], [3] to determine a profile of the text and its author, for genre (command, report, request), tone (authoritative, submissive), and for author characteristics (gender, age, background). In all of these cases, symbolic labels will be inserted appropriately into the corpus database which can then be searched over, or used as data fields for data mining. In the near future, we also intend to incorporate more in-depth linguistic

processing including part-of-speech tagging [12], named-entity tagging (e.g., [13]), and information extraction (e.g., [14]), which will enable inclusion of more detailed structured data in the corpus database.

To support flexible querying of the complex document information at the level of *concept analysis*, the prototype will maintain a database of all the information extracted from the corpus (this also includes unstructured text). The IIT Mediator is built using the SIRE information retrieval infrastructure [15], [16] which represents all information, whether unstructured text, semistructured XML, or structured data, in a relational database format. This uniform substrate gives us a foundation to build upon to allow more sophisticated querying of information extracted from complex documents at previous levels. Spatial relations between document elements, such as adjacency, are straightforward to encode and query, as are defined keywords denoting topic, style, and graphic categories. In addition, this opens up larger possibilities for future work, of searching for clusters of related data elements, as in queries such as "Find three letters discussing corporate mergers, signed by the same person."

Finally, our relational approach to data representation allows the application of traditional data mining and machine learning methods to discover new associations in the data, such as association rules [17] and document clustering [18], [19]. While our work on data mining from complex document collections is just beginning, we will work on developing algorithms that look for correlations in document descriptors (for example, possible relationships between the author of a document and particular language styles). Additionally, the relational structure will ensure that our approach can be combined with other external data (biographical data on authors, etc.) to give users a more complete picture.

In addition to all the above, the use of an architecture that integrates all CDIP modules, from image processing to information retrieval and data mining, will also enable top-down application of interpretative constraints based on user interaction and feedback. For example, if only partially relevant results are found for a user's query, a further attempt may be made to find information of the requested type by re-examining the original documents in light of the particular type of information requested (and what is currently known).

III. Data Sets and Evaluation

A central task for a CDIP project is evaluating the effectiveness of the system, and how that effectiveness responds to changes in effectiveness of components (OCR, IR, table and diagram extraction, etc.). A major tool in evaluation are test collections, i.e. data sets manually annotated with desired or correct interpretations.

Test collections have played a large role in the evolution of the fields on which CDIP depends: document analysis, information retrieval, computational linguistics, and data mining. Test collections are used both to evaluate technologies in these areas, and to improve the effectiveness of these technologies through tuning parameters and fitting statistical models. Indeed, much recent progress in these fields has resulted from the incorporation of new statistical and machine learning methods enabled by the availability of labeled data sets.

In contrast, we believe progress on unified CDIP systems has lagged precisely because of the lack of a high quality, publicly available test collection. The construction of such a collection is a major goal of our project. We describe our progress in this area below, along with how we will use the resulting collection in evaluating capabilities of our CDIP system.

A. Document Collections for CDIP Evaluation

A good test collection for CDIP should cover the richness of inputs a CDIP system may face: a range of document formats, structures, sizes, and genres, manifested with varying print and imaging qualities. Documents should include handwritten text and annotations, diverse fonts, and elements such as images, tables, logos, and other graphic complexities. The volume of documents, and the number of redundant or useless documents, should be large enough to stress the component technologies and the system as a whole. Further, given our interest in intelligence applications, we want our collection to contain private communications within and between groups of people planning activities and deploying resources. Finally, the data in the collection should be publicly available to researchers with minimal costs and licensing restrictions.

Documents that become public through legal duress (lawsuits, criminal trials, FOIA requests, etc.) provide one of the best sources of such material. We have chosen to base our collection on the Legacy Tobacco Documents Library, a collection of approximately 7 million documents (42 million pages) that became public through legal proceedings against US tobacco companies. We have arranged to obtain these documents in TIFF format, with associated metadata, from the University of California - San Francisco [20].

Besides satisfying the criteria we listed in the previous section, the tobacco documents have the advantage of an associated active research community. More than 200 peer-reviewed papers have been published using documents from LTDL and similar sources [21], and the US National Cancer Institute has funded more than a dozen groups to study various topics using tobacco documents [22]. We plan to leverage this interest to aid CDIP evaluation, as discussed in the next section.

B. Methods and measures

Evaluation of the CDIP prototype and its components requires not just documents, but also defined tasks, desired system outputs for those tasks, and measures of how well the system produces such outputs. We are defining such evaluation tasks for three key aspects of our prototype's performance: document analysis, information retrieval, and data mining.

1) Document Analysis: One set of evaluations will measure the effectiveness of our document analysis components. These results will help direct future research and development by both: (a) aiding evaluation of which component algorithms perform best on complex documents, and (b) enabling us to determine which (if any) document analysis components are responsible for retrieval or mining errors at the high-level. Testbed tasks will be constructed by manually annotating samples of 50-100 diverse document images for each type of desired output: text segments, document structure, diagram tags and relations, recognized characters, and so on. We are making initial use of a stratified sample of about 1200 tobacco documents produced by tobacco document researchers at University of Georgia [23]. This sample has been annotated with ground truth for printed and handwritten text.

Outputs generated by CDIP components will be compared to the manual ground truth and difference recorded. Standard effectiveness measures (e.g. character- and word-level error rates for OCR) will be used.

2) Information Retrieval: Test collections for information retrieval require, in addition to documents, a set of *topics* (specifications of certain documents of interest, and queries a user might enter to find them) and corresponding *relevance judgments* (lists of which documents should be retrieved for each topic). We are in discussions with several tobacco document groups to formalize as topics some of the subject areas in which they are searching for documents, and to cooperate on producing relevance judgments.

While these efforts are at an early stage, several aspects of the topics already suggest they will provide a good challenge for CDIP capabilities. Many topics specify not only documents on a particular topic, but documents of a particular genre (e.g. memos or minutes) or with a particular audience (internal vs. external communications). Such topics provide a challenge to both document image analysis and linguistic stylistic analysis. Handwritten documents and annotations are also of particular interest, raising the question of what can be done to improve their accessibility short of (currently impossible) arbitrary handwriting recognition.

We will measure the ability of the CDIP prototype to retrieve relevant documents in response to queries for each topic, and compute standard effectiveness measures such as simple average precision.

3) Data Mining: Another type of processing that we will support is a full scan of a set of complex documents to look for *interesting* patterns. Evaluation of data mining will focus on both effectiveness and efficiency. Effectiveness will be measured against small amounts of manually annotated data. We will investigate active learning methods in producing annotated data for training and experimental design methods for reducing the amount of data to annotated in order to determine differences among systems. In some cases, it may be possible to use one part of a complex data set as ground truth for pattern finding on another. For example, one document may list the membership of a group at a particular time, while determining that from other documents would be of interest. Tobacco document researchers have also established ground truth for some items of interest, e.g. links between various tobacco industry personnel and outside organizations.

Efficiency is an issue not just because of the size of complex document collections, but because of the ambiguity introduced by OCR, handwriting recognition, and document structure analysis. This ambiguity can vastly increase the range of patterns which a data mining system must consider, impacting running time and memory (as well as accuracy). We will focus in particular on how algorithms for association rules and temporal sequence patterns scale with the degree of ambiguity in interpretation.

IV. Current Status

A. The IIT Mediator

The current prototype of the IIT Mediator[1] implements the essential functionality of integrated retrieval we have described, for document data represented in a structured format (as above). It deals in a seamless manner with a variety of data sources, including free-text, semistructured XML, and structured databases (complex document information will be integrated soon, as lower-level processing and extraction modules reach maturity). The IIT Mediator keeps separate stores of metadata for each type of source that it has access to, which enables it to determine which sources are most likely to have data relevant to a given query. Simple natural-language query processing extracts functional roles from the query that a rule-based processor then translates into source-specific queries. Results are then integrated via an information-fusion algorithm [5]. Currently, the Mediator sends the query in its natural-language form to all available unstructured sources, essentially performing a metasearch. The idea is that the Mediator results, in addition to possibly containing an answer to the user's query, will at least be no worse than those obtained from a traditional metasearch [24].

B. Symbolic document processing

For symbolic analysis, we have focused to date primarily on topic and style based text categorization. Work on these tasks requires solving two main problems: developing algorithms to build accurate classification models, and constructing sets of features that are both useful for text categorization and computable. For the former we have both used existing machine learning methods, as well as a novel text categorization method based on Bayesian logistic regression [25]. We have also developed a comprehensive infrastructure for processing and representing linguistic annotations for texts in a database format, which includes tools for extracting a wide variety of linguistic textual features. We have further developed resources for features based on Systemic Functional Linguistics [26], which enable more accurate and more insightful stylistic text classification, in many cases [27], [28].

C. Image processing

The image processing components of our system will rely on commercial software components as well as additional components we are developing. Commercial OCR engines include in addition to OCR/ICR additional processing capabilities that are necessary for document analysis. These capabilities include noise filtering and skew correction, adaptive image segmentation, detection of page orientation, and document zone determination. Detected zones include text blocks, tables, and graphics/pictures. The performance of commercial engines on noisy document images is yet to be determined. In addition to a commercial OCR engine we will be using the *DocLib* system developed by the Language and Media Processing Laboratory of the University of Maryland, as an integration framework. DocLib includes in addition to low-level processing capabilities, several high level-modules such as logo-detection and script-identification. The components we are developing for this system include document image de-warping modules [29], [30], separation of text from handwriting using directional regulated morphological operations [8], [9], signature clustering, and table identification in noisy documents.

[1]Demo version available at `http://www.ir.iit.edu/mediator`.

V. Summary

Complex document information processing is of increasing importance to intelligence analysis in the modern world, and yet little work has been done on developing integrated solutions to this problem. We have laid out an integrated architecture which unifies the various component technologies, giving a clear roadmap for research on developing useful CDIP systems.

Acknowledgments

This work is currently supported by a Challenge Workshop grant from ARDA.

References

[1] D. Lewis, R. E. Schapire, J. P. Callan, and R. Papka, "Training algorithms for linear text classifi ers," in *SIGIR '96: Proc. of the 19th Int. Conference on Research and Development in Information Retrieval*, 1996.

[2] S. Argamon, M. Šarić, and S. S. Stein, 'Style mining of electronic messages for multiple author discrimination," in *Proc. ACM Conference on Knowledge Discovery and Data Mining*, 2003.

[3] S. Argamon, M. Koppel, J. Fine, and A. R. Shimony, 'Gender, genre, and writing style in formal written texts," *Text*, 2003.

[4] D. Grossman, S. Beitzel, E. Jensen, , and O.Frieder, 'IIT Intranet Mediator: Bringing data together on a corporate intranet," *IEEE IT Professional*, vol. 4, no. 1, pp. 49–54, 2002.

[5] S. Beitzel, E. Jensen, A. Chowdhury, D. Grossman, O. Frieder, and N. Goharian, 'On fusion of effective retrieval strategies in the same information retrieval system," *Journal of the American Society of Information Science and Technology*, vol. 55, no. 10, 2004.

[6] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, "Fast discovery of association rules," in *Advances in knowledge discovery and data mining*. American Association for Artificial Intelligence, 1996, pp. 307–328.

[7] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *Proc. SIGKDD Workshop on Text Mining*, 2000.

[8] G. Agam and I. Dinstein, "Adaptive directional morphology with application to document analysis," in *Mathematical Morphology and its Applications to Image and Signal Processing*, P. Maragos, R. W. Schafer, and M. A. Butt, Eds. Kluwer Academic Publishers, 1996, pp. 401–408.

[9] ——, "Regulated morphological operations," *Pattern Recognition*, vol. 32, no. 6, pp. 947–971, May 1999.

[10] J. Liang, I. Phillips, and R. Haralick, "A methodology for document image structures extraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 719–734, 2001.

[11] S. Aksoy and R. M. Haralick, "Graph-theoretic clustering for image grouping and retrieval," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1999, pp. 63–69.

[12] E. Brill, "Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging," *Computational Linguistics*, vol. 21, no. 4, pp. 543–565, 1995.

[13] X. Carreras, L. Marques, and L. Padro, "Named entity extraction using adaboost," in *Proceedings of CoNLL-2002*, Taipei, Taiwan, 2002, p. 167170.

[14] D. Freitag, "Machine learning for information extraction in informal domains," *Machine Learning*, vol. 39, 2000.

[15] O. Frieder, A. Chowdhury, D. Grossman, and M. C. McCabe, "On the integration of structured data and text: A review of the SIRE architecture," in *DELOS Workshop on Information Seeking, Searching, and Querying in Digital Libraries*, Zurich, Switzerland, December 2000.

[16] C. Lundquist, O. Frieder, D. Holmes, and D. Grossman, "Integrating structured data and text: A relational approach," *Journal of the American Society for Information Science*, vol. 50, no. 5, April 1999.

[17] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, J. B. Bocca, M. Jarke, and C. Zaniolo, Eds. Morgan Kaufmann, 12–15 1994, pp. 487–499.

[18] A. Ruocco and O. Frieder, "Clustering and classification of large document bases in a parallel environment," *Journal of the American Society of Information Science*, vol. 48, no. 10, October 1997.

[19] ——, "Parallel document clustering process. US Patent #5,864,855. Patent filing by the Office of the Judge Advocate General, Department of the Army." January 1999.

[20] H. Schmidt, K. Butter, and C. Rider, "Building digital tobacco industry document libraries at the university of california, san francisco library/center for knowledge management," *D-Lib Magazine*, vol. 8, no. 2, 2002. [Online]. Available: http://www.dlib.org/dlib/september02/schmidt/09schmidt.html

[21] N. Hirschhorn, "Research reports and publications based on tobacco industry documents, 1991-2005," May 2005. [Online]. Available: http://tobacco.health.usyd.edu.au/site/gateway/docs/Hirschhorn_publications_list.pdf

[22] "Review and analysis of tobacco industry documents," June 1999, national Cancer Institute Program Announcement. [Online]. Available: http://grants.nih.gov/grants/guide/pa-files/PAR-99-114.html

[23] W. Kretzschmar, C. Darwin, C. Brown, D. Rubin, and D. Biber, "Looking for the smoking gun: Principled sampling in creating the tobacco industry documents corpus," *Journal of English Linguistics*, vol. 32, pp. 31–47, 2004.

[24] D. Dreilinger and A. E. Howe, "Experiences with selecting search engines using metasearch," *ACM Transactions on Information Systems*, vol. 15, no. 3, pp. 195–222, 1997.

[25] A. Genkin, D. D. Lewis, and D. Madigan, "Large-scale bayesian logistic regression for text categorization," *submitted*, 2004.

[26] M. A. K. Halliday, *Introduction to Functional Grammar*, 2nd ed. Edward Arnold, 1994.

[27] S. Argamon and J. Dodick, "Conjunction and modal assessment in genre classification," in *Notes of AAAI Spring Symposium on Attitude and Affect in Text*, March 2004.

[28] C. Whitelaw, N. Garg, and S. Argamon, "Using appraisal taxonomies for sentiment analysis," *submitted*, 2005.

[29] G. Agam and C. Wu, "Structural rectification of non-planar document images: application to graphics recognition," in *Graphics Recognition: Algorithms and Applications*, ser. LNCS, D. Blostein and Y.-B. Kwon, Eds., vol. 2390, 2002, pp. 289–298.

[30] C. Wu and G. Agam, "Document image de-warping for text/graphics recognition," in *Structural, Syntactic, and Statistical Pattern Recognition*, ser. LNCS, T. Caelli, A. Amin, R. Duin, M. Kamel, and D. de Ridder, Eds., vol. 2396, 2002, pp. 348–357.

DOCLIB : A Document Processing Research Tool

Kevin Chen

Booz | Allen | Hamilton
134 National Business Pkwy
Annapolis Junction, MD 20701
chen_kevin@bah.com

Stefan Jaeger
Guangyu Zhu
David Doermann

Institute for Advanced Computer Studies
University of Maryland,
College Park, MD 20742, USA
jaeger@umiacs.umd.edu

Abstract

Often, valuable document processing intellectual capital is lost due to staff transitions or project restructuring prior to technology transfer. Furthermore, hardware and software integrity, dependencies, and compatibility are critical components that often impede technology migration. While many open source tools attempt to mitigate these issues, they do not always address specific design needs and tailored-process that Government organizations must adhere to. This paper addresses the need for a common document processing research vehicle through which institutions can develop and share research-related software and applications across academic, business, and Government domains.

1 Introduction

Although research does not generally commission the development of production quality software, the absence of a software development framework and guiding process can significantly impede progress upon the migration of technologies. The transfer of technology across organizational boundaries requires a significant time investment. These efforts are often exacerbated when there is incomplete documentation, unpredictable methods required to build and execute software, inconsistent software dependencies, and/or a misuse of software. The setup, configuration, and execution of research-related software can be difficult without a structured and shared development and delivery process. Although a time investment is required during software acquisition, a lack of infrastructure and process causes an avoidable loss of intellectual capital within the document processing community. A set of core research tools developed with an eye toward production, quality, and stability can allow valuable intellectual resources to be more focused on creative and innovative document processing solutions. This, in turn, can maximize return on investment (ROI) for funding organizations. This paper prescribes a systematic development approach and shared architecture which has been successfully used in collaboration across academia, business, and Government environments. The DOCLIB Collaboration Model also includes an "add-on" construct that allows research related applications to be developed and shared.

The DOCLIB Collaboration Model has been effectively used as a communication portal between University of Maryland, the Department of Defense, and industry. The use of this collaboration model has resulted in a shared document image processing C++ toolkit termed core-DOCLIB. This toolkit is currently funded by a specific Government division.

Most existing software packages satisfy only a sub-set of the document processing user base and do not prescribe a robust development process to facilitate technology transfer [11,12,13,14]. For instance, Intel's OpenCV Toolkit provides high-level computer vision methods for Windows users only [11]. This limitation creates a barrier across the enterprise and discourages collaboration. Alternatively, MATLAB does not provide the ease-of-use and efficiencies needed for a collaborative model [14]. The core-DOCLIB toolkit, however, supports various platforms (e.g.: Windows, Solaris, Linux, etc) and compilers (e.g.: c++.net, gcc, g++, Forte) and readily conforms to security pre-requisites most Government agencies must adhere to. Moreover, the external API was designed to be easy-to-use for those users new to the object oriented world.

Our intention is to advertise the DOCLIB collaboration model across varying Government divisions. Decisions relating to sharing of the core-DOCLIB software, shared development environment, and add-ons are in progress.

2 Approach

Compliance to a structured development process that directly addresses many of the technology migration obstacles currently facing the industry was the driving force behind the core-DOCLIB success. The DOCLIB Collaboration Model involves the use of the core-DOCLIB toolkit and the adaptation of a shared process used to foster collaboration.

Core-DOCLIB is a C++ toolkit that provides document/image-processing capabilities through a documented, easy to use interface. This library seeks to provide a functional, stable, and robust environment that supports the collaborative development of research capabilities. Use of the DOCLIB Collaboration Model has facilitated technology migration by mitigating debugging time, number of code reviews, and configuration management challenges.

Document processing research applications that are built on top of the core-DOCLIB library are referred to as DOCLIB add-ons. A DOCLIB add-on standardizes the delivery and use of research applications. Robust test scripts, standard naming conventions, auto-generated documentation, and an organized directory hierarchy are some of attributes that are included as part of an add-on. Adherence to the add-on conventions fosters discipline within organizations resulting in less time-consuming technology migration efforts. The application of the DOCLIB Collaboration Model to 'add-ons' has improved collaboration efforts between the Department of Defense and the University of Maryland over the past two years.

3 Core-DOCLIB

The establishment and use of the DOCLIB Collaboration Model fueled the creation of the core-DOCLIB C++ toolkit. This toolkit has served to satisfy document processing research requirements set forth by academia and the Department of Defense. It was necessary for core-DOCLIB to be scalable, flexible, and extendable to meet specific organizational needs. For example, use of an Image Factory was introduced in the core-DOCLIB architecture to satisfy a need for plug-and-play image formats. This design allows image types to be added or removed without modifications to the existing core-DOCLIB code.

DLImage DLImage is the centralized image object used to store image data and header information when loading an image. The DLImage object is a generic image format that is independent of the input image types (i.e. TIFF, JPEG, GIF, etc.). DOCLIB Image processing algorithms are designed as independent modules that can be added or deleted without affecting other core-DOCLIB features (See Figure 1).

Core-DOCLIB currently supports the following image processing features:

- Resize an image
- Rotate an image
- Subimage
- Flip an image
- Contour an image
- Reverse black/white pixels on an image
- Paste an image
- Project
- Calculate connected components
- Draw Line, Box, Pixels onto an image
- Dilate an image
- Erode an image
- Sharpen an image
- Blur an image
- Conversions:
  - 256 Color quantized
  - Percent Thresh hold Binarization
  - Threshold Binarization
  - Color To Gray
  - Threshold Gray To Binary
  - NIBlackBinarization
  - Binary To Color
  - Gray To Color
  - Binary To Gray
  - YCrCb Binarization
- Skeletonize
- Deskew
- Centroid

Image Factory This factory inherits the Factory Design Pattern concept and extends the factory concept (See Figure 2). The Image Factory enables a higher lever of abstraction, encapsulating the implementation of all image objects. Moreover, the Image Factory provides the logic to determine an image's file format without the user specifying.

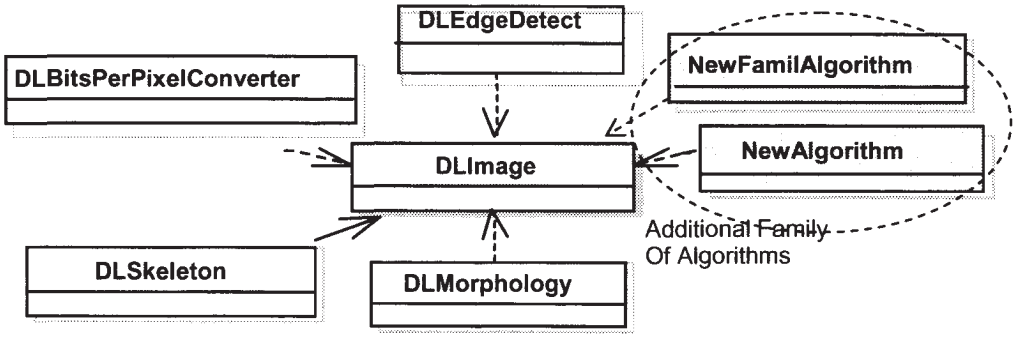


Figure 1: DOCLIB Algorithm Hierarchy

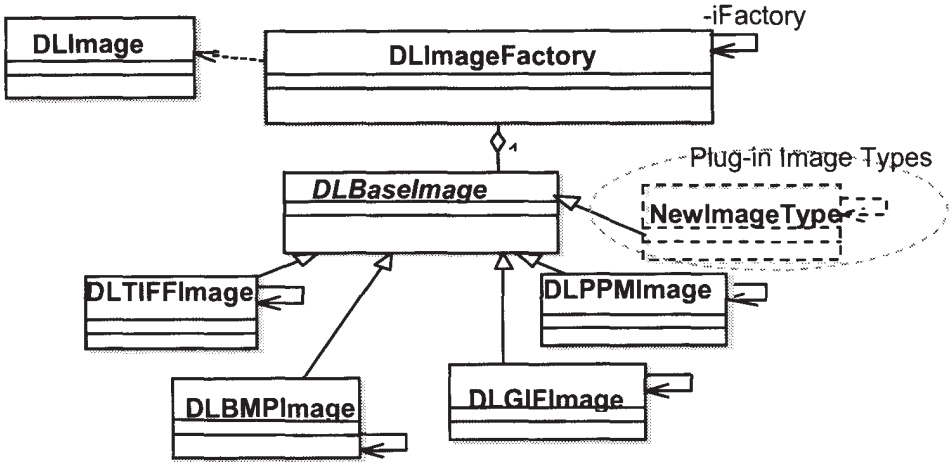


Figure 2: Image Factory class diagram

All image type objects are declared statically, therefore, the image type objects are constructed when the application starts. Image type objects register their existence at the Image Factory during construction (see Figure 3). Once the new image object has been registered with the Image Factory, the new image type will be one of the supported images in core-DOCLIB. The core-DOCLIB Image Factory architecture has been proven and allows the Department of Defense to 'plug-in' proprietary image types into core- DOCLIB without any code modification.

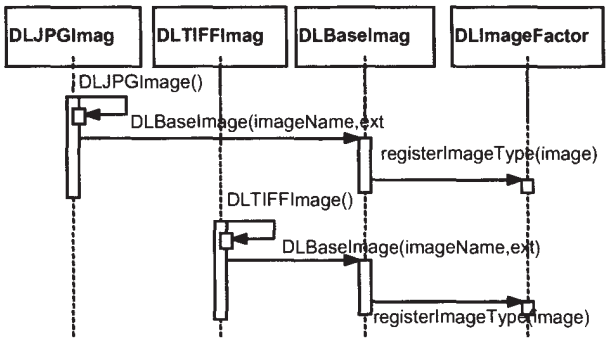


Figure 3: Image plug-in registration process

3.1 DLDocument

The DLDocument class was defined as a container for the results of basic image operations and metadata calculations performed by other DOCLIB classes and add-ons. While DLImage is a container for the basic image operations implemented in DOCLIB, DLDocument offers data structures and methods for manipulation of metadata, i.e. data that is on a higher cognitive level than just the image pixels, such as ground-truth data for classifier training or logical segmentations into lines or characters. DLImage and DLDocument are two independent modules standing side by side and interacting with each other, as illustrated in Figure 4. In particular, DLDocument and its subclasses have access to their own image data and can perform basic image operations using the methods in the DLImage class.

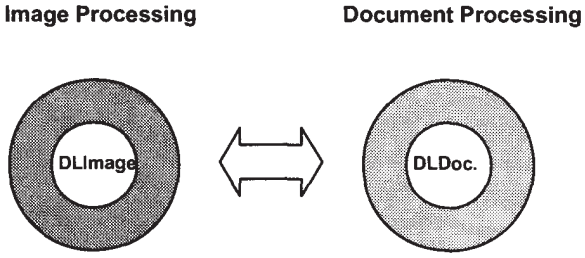


Figure 4: DLImage and DLDocument

DOCLIB's DLDocument class understands documents as hierarchical entities composed of pages containing regions of interest, which we call zones. Accordingly, DOCLIB's document hierarchy consists of the following objects:

- DLDocument
- DLPage
- DLZone

The zones are hierarchical structures in themselves and can contain sub zones (see Figure 5).

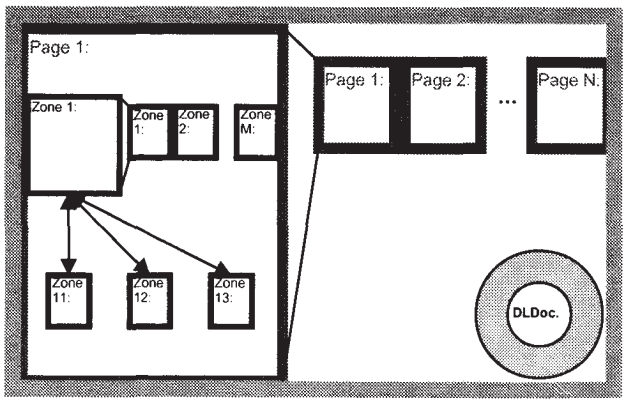


Figure 5: DOCLIB's document hierarchy

The DLDocument class represents the highest level in the DOCLIB document hierarchy. It is a container for attributes that describe the layout and content of one or more pages within a document. Each DLDocument object has a unique document ID and features a list of document tags allowing developers to dynamically append or remove additional self-defined attributes. Several constructors are available, depending on the information present at the time of construction. The pages of a document are stored in a simple list structure. Developers have access to this list via a set of functions for appending, inserting, or deleting either one single page or a list of pages.

The DLPage class represents the metadata and logical content of a document page. Its main purpose is to accommodate information specific to pages rather than whole documents. The structure of DLPage, however, is similar to DLDocument. It contains a page ID and a dynamic attribute list that developers can use to store self-defined attributes. The main difference to the DLDocument class is a list of zones, which are general containers defining specific regions on a page, such as text blocks, images etc. Similar to DLDocument, DLPage provides several functions for appending, inserting, or deleting zones on a page. In addition, it provides methods for accessing page-related information, e.g. its spatial dimensions or the corresponding DLImage.

A DLZone object is a general container for objects on a document page. DLZone offers a general bounding box concept that allows zones to have sub zones, so-called child zones. These child zones are stored in the same list structure used by DLDocument and DLPage to store pages and zones, respectively. A zone and its sub zones span a dynamic tree structure in which zones (nodes) can easily be appended, inserted, or deleted. In addition, DLZone provides methods for merging and splitting zones. Similar to methods in DLPage, DLZone offers methods for accessing information specific to zones. A Zone object is basically defined by its origin, width and height. DOCLIB allows developers to dynamically change this information. Each DLZone object automatically performs a consistency check whenever the developer tries to do so, and denies the operation when it would lead to an inconsistent state, for instance when a zone would extend the boundaries of its parent.

In order to allow programmers to adapt DOCLIB to their own needs, and customize DOCLIB objects to the requirements of a specific application domain, developers can subclass DLZone and augment it with more specific information. For example, typical user-defined subclasses of DLZone could be DLTextLine and/or DLCharacter etc.

4 Software Development Process

The DOCLIB Collaboration Model has shown that adapting a structured development process assists throughout the technology migration process. Several well-defined practices were established and integrated as part of the development and release proceedings.

Distribution portal Coordinating software changes or enhancements without a distribution mechanism can be quite challenging. Given the complexity of document processing related research, it is often that bug fixes/code enhancements or modifications are transferred to end-users. A web portal that allows users to check status, download specific versions of the libraries, and seek help is invaluable to coordinating and communicating during technology transfer.

Source control One of the most important aspects of collaborative code development is the sharing of the code itself. Without all users having access to the same code base, each party will inherently develop similar but different baselines in parallel. Several open source version control software packages facilitate the management of software development initiatives. These systems provide developers with the ability to obtain and modify source code for their projects. If errors occur, a configuration Management (CM) tool provides methods for reverting code to a previous known stable state.

Documentation standards Thorough and understandable documentation is crucial to the success of many technology transfer efforts. Unfortunately, it is also the most often ignored. Regimented documentation standards and use of auto-generating documentation packages (ie. Doxygen [7]) allow user documentation to be quickly and easily created. This information is often the single source customers will use to gain insight into the operation and functionality of software.

Bug tracking Bug tracking helps to ensure the development of quality software products. Bug tracking gives users and developers the ability to document

bugs, request new features, and check the status of the resolutions to their issues. We use the popular Mantis bugtracker for DOCLIB [10].

Software integrity It is difficult to guarantee software integrity within the document processing research field given the complexity of the science and the variability in data. The need to verify system behavior was evident early on during the development of core-DOCLIB. A regression test application was prescribed that thoroughly tested all behavior within the system. This module is often used by software developers before committing changes to existing code. This approach has also proven invaluable in identifying obscure bugs before formal core-DOCLIB software releases.

The execution of regression tests also facilitates the observance of memory leaks throughout the system. Advertisement of the regression tests has helped build buy-in and commitment to customers of core-DOCLIB and/or add-on modules. This mechanism has proved very useful in minimizing the number of bugs occurring throughout core-DOCLIB.

5 DOCLIB Add-Ons

DOCLIB has proven its success as an underlying image processing library in the DoD community and at University of Maryland for the last two years. DOCLIB not only helps resolve compatibility issues among different research groups but also helps to reduce the amount of code that needs to be imported into the Government workspace.

The core of DOCLIB provides basic image processing methods that have been successfully tested in numerous practical applications and are considered standard by researchers and programmers alike. Add-on modules, however, are solutions to problems that have hitherto eluded practical approaches, and are still actively investigated within the research community. Methods competing with other techniques are also implemented as add-ons, either because the research community has not yet given the final verdict on their underlying theory or they are fine-tuned to specific problem types and can thus not be considered standard for the general case. DOCLIB provides most of its document processing functionality as add-ons to the DOCLIB core. Table 1 of the appendix lists some of the currently available add-ons for DOCLIB.

6 Conclusion

A shared software development toolkit and structured development process can significantly enhance technology migration experiences within the document image processing community. The DOCLIB Collaboration Model prescribes a methodology and underlying toolkit to satisfy these needs.

Recipients of DOCLIB add-ons can be reassured that a structured development process was adhered to. Moreover, the location of documentation, source code, test scripts, etc will be consistent regardless of the organization developing it. Once an organization is a user of core-DOCLIB, only add-on related code will have to be transferred to receiving organizations.

The DOCLIB Collaboration Model has been effectively used as a communication portal between University of Maryland, the Department of Defense, and industry. The use of this collaboration model has resulted in a shared document image processing C++ toolkit termed core-DOCLIB. This toolkit is currently funded by a specific Government division. Our intention is to advertise this collaboration model across varying Government divisions. Decisions relating to sharing of the DOCLIB software, shared development environment, and add-ons are in process.

Acknowledgments

The partial support of this research under DOD contract MDA90402C0406 is gratefully acknowledged.

References

[1] J. Hochberg, P. Kelly, T. Thomas, and L. Kerns, Automatic script identification from document images using cluster-based templates, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 19, No. 2, pp. 176–181, Feb. 1997.

[2] L. O'Gorman, The Document Spectrum for Page Layout Analysis, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 15, No. 11, pp. 1162-1173, Nov. 1993.

[3] S. Jaeger, Informational Classifier Fusion, 17th Int. Conf. on Pattern Recognition (Cambridge, UK), pp. 216-219, 2004.

[4] L. Golebiowski, Automated Layout Recognition, Symposium on Document Image Understanding Technology, Greenbelt (Maryland), pp. 219-228, 2003.

[5] M. Steinbach, G. Karypis, and V. Kumar. A Comparison of Document Clustering Techniques, Technical Report #00-034, Department of Computer Science and Engineering, University of Minnesota, 2000.

[6] R. Ziemer, Removing Repetitious Horizontal Background Lines, Internal Report, BAH, 2004.

[7] http://www.doxygen.org

[8] G. Zhu, S. Jaeger, D. Doermann, A robust stamp detection framework on degraded documents, Proc. of SPIE Conf. on Document Recognition and Retrieval XIII, 2006, to appear.

[9] G. Zi, Groundtruth Generation and Document Image Degradation, Technical Report: LAMP-TR-121/CAR-TR-1008/CS-TR-4699/UMIACS-

TR-2005-08, University of Maryland, College Park, May 2005

[10] http://www.mantisbt.org

[11] www.intel.com/technology/computing/opencv/

[12] C. H. Lee, T. Kanungo. The Architecture of TRUEVIZ: A groundTRUth/metadata Editing and VIsualiZing toolkit. Technical Report: LAMP-TR-062/CS-TR-4212/CAR-TR-959, University of Maryland, College Park, February 2001.

[13] T. Kanungo, C. Lee, J. Czorapinski and I. Bella, TRUEVIZ: A groundTRUth/metadata Editing and VIsualiZing Toolkit for OCR, *SPIE Conference on Document Recognition and Retrieval*, 2001.

[14] http://www.mathworks.com/

Appendix

Table 1: DOCLIB Add-ons

Application Type	Add-on Module Name	Descriptiom
Script Identification	ScriptIdentification	The DOCLIB script identification add-on performs script identification for machine printed document images. It is a modified version of the Hochberg algorithm described in [1]. It allows classifying a document image as being printed in one of the following scripts: Amharic, Arabic, Armenian, Burmese, Chinese, Cyrillic, Devanagari, Greek, Hebrew, Japanese, Korean, Latin, or Thai. Script Identification can also be retrained to focus on different language mixes.
Layout Analysis	PageLayoutDOCLIB	The page layout add-on provides an automated means of training a classifier to recognize a document layout or set of layouts. The classifier can then be used to score an unknown image. The software is easily modified to include other types of object information. More details are given in [4].
Page Segmentation	DLDocstrumDOCLIB	For page segmentation, DOCLIB provides a module implementing the Docstrum method for structural page layout analysis [2]. The Docstrum method is based on bottom-up, nearest-neighbor clustering of page components. It detects text lines and text blocks, and is advantageous over many other methods in three main ways: independence from skew angle, independence from different text spacings, and the ability to process local regions of different text orientations within the same image.
Line Processing	DLGetLinesDOCLIB	Given a deskewed, oriented text image with dark machine print text on a light background, the line detection add-on will calculate text line information based on the connected component information. The image must be deskewed and oriented in order to receive meaningful results. The algorithm utilized has proven robust given text images with low resolution, broken characters, connected scripts (e.g. Arabic), multi-component character scripts (e.g. Chinese), and noise.

Application Type	Name	Descriptiom
	ZRemoveLinesDOCLIB	The line removal module is used to read in a document image and locate the background horizontal lines [6]. It then creates an output file in which the background lines are removed while the text remains intact. This add-on is designed to work on handwritten documents where the user writes on lined paper. Consistent background lines with inconsistent writing are expected. Less than optimal (though still useful) output may result when this is not the case. Various parameters are defined in the module which, when modified, may improve performance depending on the data. The module does not rely on color information. Hence it is particularly handy when the background horizontal lines are similar to the color of the handwriting. The algorithm requires binary input images that are reasonably deskewed (roughly, between -9 and 9 degrees, inclusive).
Object Recognition	logoTokenMatchDOCLIB	This module implements a logo recognition algorithm. Given a potential logo it performs a gray scale correlation against a set of candidate logos. For each candidate a score between 0 and 100 is generated corresponding to the degree of similarity. The best match is provided along with the score. In order to improve performance, the algorithm stops comparing against candidate logos when the best score is beneath a pre-defined threshold.
	StampDetection	The stamp detection module detects and locates elliptic stamps or seals on an input image by identifying and characterizing elliptic connected edges using both their magnitude and orientation information [8]. For a given document, it outputs an image list of detected stamps together with their corresponding confidence values. The module shows good performance for images with moderate noise and can be efficiently used in batch mode to process a large set of images.
Classifier Combination	InformationFusion	This add-on is an implementation of a new technique for information fusion and classifier combination, developed at the University of Maryland for hard and noisy environments. The underlying information-theoretical idea is to first normalize the confidence values provided by different methods for the same problem, before actually combining them. The normalization is performed in such a way that each confidence value matches its informational content. The recognition candidate providing the maximum information over all methods will then be selected as the most likely candidate for a given classification problem and presented test pattern. The method has already been successfully applied to several document processing tasks, including character recognition and script identification. Several papers describing the theoretical background have been published [3].

Application Type	Name	Descriptiom
Degradation	DLDegradationDOCLIB	DOCLIB's degradation tool degrades images with different types of noise [9]. Its main purpose is to emulate noise typically introduced during the scanning of paper documents. The generation of well-defined noisy environments allows adaptation of recognizers to real-word noise usually encountered in practical documents. For instance, the following types of noise may be added to a document image: blur, jitter, horizontal and vertical lines, rotation, speckle, pixel flips, etc.
Performance Evaluation	F_ScoreDOCLIB	The F-Score module is used to compare two object sets, the putative set and the truth set. Based on this comparison, the precision, recall, and F-measure values are calculated along with the F-score(s) [5].

A Process Flow for Realizing High Accuracy for OCR Text

Kazem Taghva and Julie Borsack and Tom Nartker
Information Science Research Institute
University of Nevada, Las Vegas
Las Vegas, NV 89154-4021

Abstract

1 Introduction

The Information Science Research Institute (ISRI) has performed in-depth research of optical character recognition (OCR), information retrieval, and related topics since 1989. ISRI ran many experiments comparing OCR output to manually corrected versions of the same collections. Although our experiments have shown that average precision and recall are not negatively affected by OCR errors, we also identified and characterized problems that OCR text may cause when applied in retrieval and other information access tasks.

In nearly all large hard copy document conversion tasks, information retrieval is just one aspect of the entire project. For example, classification of documents and extraction of specific information may also be required of the collection. Therefore, the required level of accuracy for these information access tasks is an important consideration. Measuring OCR text accuracy though is a formidable undertaking without automation.

In this paper we present a means to automatically measure OCR quality of documents. We also recommend a hardcopy document conversion process that will perform automated OCR correction, determine a document's OCR quality, and provide a correction tool that can guarantee a certain level of accuracy for converted collections.

2 OCR and Information Access

Although most information today is created electronically, compiling a collection of documents generated from various text processing applications proves to be more difficult than one would expect. In fact, most large collections are produced using imaging and optical character recognition. These processes retain "the original" document form (the images) and create a consistent and automated method for data capture. Unfortunately, OCR generated text can cause difficulties for information access (IA). Two widely applied IA tasks are information retrieval (IR)[4, 6, 10] and information extraction (IE)[3]. In this section we introduce a few of the difficulties we have encountered in our research when using OCR text with these technologies. We also note the level of accuracy we believe is required to perform these tasks effectively.

In Sections 3 and 4 we present two technologies that automate OCR correction and document quality control (QC). In Section 5, we introduce an additional semi-automated technology that can be used to further improve document quality if necessary. The application of these components after OCR will greatly improve the collection for any subsequent IA tasks.

2.1 Problems for Information Retrieval

ISRI studies on effects of OCR errors on retrieval have pointed out that certain advanced functionalities of information retrieval systems, such as ranking, are not robust enough to overcome OCR errors. It is our view that more advanced retrieval techniques and applications require a higher accuracy rate. For example, increased overhead in the index (especially for large collections)[16, 14, 20], potential unpredictability in document ranking[14, 15] and the unretrievability of poor quality documents[15, 17] are issues that should be considered.

We have shown that using automated post-processing can improve document quality with regard to these issues and in turn improve retrieval effectiveness[8]. Although the required level of accuracy may vary from application to application, we found that a 95% *word accuracy* should be the minimum for document collections with an average page length of ten pages or greater to attain results similar to what one would expect from a clean collection.[11]. If short abstracts are to be indexed, there may not be enough redundancy to compensate for

misrecognized words. A higher word accuracy rate may be needed to retrieve short documents.

2.2 Problems for Extraction

Information extraction is even more heavily affected by poor OCR. Unlike IR, where redundancy compensates for poor recognition, any misrecognized characters in the data to be extracted will cause an instance of the targeted data to be missed. Another issue for IE that usually plays no role for IR is zoning errors. The more complex a document structure becomes, the more difficult it is for an OCR to determine reading order. When reading order is incorrect or if the zone type is misconstrued, zoning errors are the result. Zoning errors affect IE by eliminating the logical order expected by an extraction algorithm. So not only does IE demand higher character accuracy, it also expects the targeted data to have a predictable order.

ISRI has recently performed IE research that validates these claims. Our current research focuses on the identification of privacy information in free text. The Licensing Support Network (LSN), a large document collection made up mostly of OCR'd documents, should not disclose any personal privacy information once the collection is made publicly available via the Internet. Several "privacy types" must be identified and potentially redacted prior to a document's release. In this section we discuss three of our more well-developed extraction techniques that identify personal addresses, birth dates, and employee identification numbers, and discuss the effect of OCR on the results.

Our address extraction task applies the Hidden Markov Model (HMM) for discovering non-commercial addresses in free text[5, 7]. With the OCR text, we encountered several difficulties. The order of the address components in the OCR output did not always correspond to the correct order of the address on the image. Incorrect zoning affected the tagging applied by the HMM because the order of the tags did not fit the algorithm's expected sequence. Misrecognized characters and the loss of contextual information also interfered with address identification. Our test results indicate that OCR data significantly degrades tasks such as searching for specific features such as addresses in free text. For a more complete discussion of this research, please refer to [19].

Birth dates and employee identification numbers (employee id) are also protected from disclosure under Exemption 6 of the Freedom of Information Act (FOIA), 5 USC Section 552(b)(6)[9]. These algorithms look for relationships among features since if a feature or the privacy relationship is missing, the data is not considered private[2, 1]. For example, a date in isolation is not private nor does it become private if it is identified as a birth date with an identifier such as "born on." A birth date becomes private when it is associated with an individual. The statement *John Doe was born on 5/17/55* would be considered private but all these features, person name, relationship, and date, must exist to be tagged as such. The employee id extractor is similar in spirit in that to be considered private, it must be associated with an individual. So it is clear that certain keywords are required for each PA type to be identified and that the proximity of these features to each other is vital to the extraction algorithm.

We evaluated the results using the standard measures of recall, precision, and the F1 measure but in two ways. First, we computed these measures on a per-document basis. At this level, the effectiveness of our programs as document classifiers is determined. That is, we can determine how effective our extractors are at answering the question, *which document contains at least one example of private information of the given type?*

Second, we also evaluate our programs with respect to how well they identify specific instances of a given PA type. Thus, we can determine how well these techniques identify instances of a given PA type. Figure 1 presents the results of experiments using both the birth date and employee id extraction programs.

Although some degradation is apparent at the document level, it is when one compares the results between the clean and OCR instances that the affect of misrecognition is significant.
As with the address HMM IE algorithm, two types of OCR-errors affected the extractors ability to identify birth dates and employee ids: character recognition errors and zoning errors. We estimated that 37.5% of the errors were caused by misrecognition and 63.5% were due to erroneous zoning.

Since most extraction tasks require identifying specific pieces of information from text and will more likely require some proximity, if any part of that information is incorrect it will negatively impact the extraction algorithm. For extraction tasks, close to 100% OCR accuracy is needed. Some of the character errors can be automatically corrected using an aggressive OCR error correction system like Manicure (See Section 3). But we believe if MCT (Section 5) is configured specifically for a particular application, semi-automated correction can significantly improve OCR text for extraction.

3 Automated Post-Processing

Through experimentation and analysis, we identified certain OCR problems that could be corrected through automated post-processing. Mani-

Clean				
Task	Collection	Recall	Prec.	$F1$
Per Document				
employee id	empid-617	0.979	0.903	0.940
employee id	empid-579	0.949	0.968	0.958
birth date	dob-745	1.000	0.930	0.964
birth date	dob-076	0.974	0.961	0.967
Per Item				
employee id	empid-617	0.695	0.683	0.689
employee id	empid-579	0.815	0.783	0.799
birth date	dob-745	0.990	0.850	0.915
birth date	dob-1076	0.678	0.779	0.725

OCR				
Task	Collection	Recall	Prec.	$F1$
Per Document				
employee id	empid-617	0.979	0.903	0.940
employee id	empid-579	0.928	0.968	0.947
birth date	dob-745	0.925	0.925	0.925
birth date	dob-076	0.869	0.957	0.910
Per Instance				
employee id	empid-617	0.644	0.556	0.597
employee id	empid-579	0.712	0.406	0.517
birth date	dob-745	0.938	0.835	0.884
birth date	dob-1076	0.464	0.534	0.497

Figure 1: Comparing clean text to OCR text for data extraction

cure (Markup ANd Image-based Correction Using Rapid Editing) is ISRI's software solution for performing automated OCR post-processing[18]. Manicure is specifically designed to prepare text collections from printed materials for information access applications. We found that by post-processing the *complete* document, misrecognized words could be corrected and other post-cleanup could be performed with a very high level of accuracy[13, 12]. Manicure automatically detects and corrects OCR spelling errors by using dictionaries, approximation matching, the knowledge of typical OCR errors, and frequency and distribution of words and phrases in a document.

Over the past several years, Manicure's most applied resource has been its ability to automatically correct misspellings in OCR text. The number of corrections made seemed impressive but there was no measurable proof that Manicure made a significant difference in error correction. In 2003, ISRI prepared an experiment that tested the corrections made by Manicure. These tests showed that the word accuracy improvement of Manicure output over raw OCR output ranged from 0.51% up to 2.55% (depending on the quality of the document) for all non-stopwords. This percentage of improvement is comparable to correcting as many as 30% of the misspellings. In addition, these results showed that the improvement provided by Manicure increases as page quality decreases[8].

These tests confirmed that Manicure improves the quality of OCR text for retrieval and has the potential to increase average precision. For many conversion projects, a Manicure'd collection may be all that is needed. But Manicure also provides a semi-automated correction interface that can bring a document to any desired level of accuracy. As mentioned in Section 2, some applications require a very high level of accuracy to achieve satisfactory results. The following Sections explain how a document's accuracy can be automatically determined using Manicure and then describes the Manicure Correction Tool (MCT), an interface that can be used to quickly and efficiently bring a document up to the desired level of accuracy.

4 Measuring Accuracy

Having an automated way of measuring OCR quality is a critical component to any large document conversion project. Yet, the only measures available were the estimated character accuracy provided by the OCR or time-consuming ground-truthing and sampling. ISRI found a solution to measuring OCR quality by building on information we already had. Using word accuracy, actually *non-stopword* accuracy, instead of character accuracy was the first necessary modification to enhancing the quality control (QC) process.

To estimate non-stopword accuracy of recognized documents, statistics about non-stopwords must be collected. For example, the QC procedure must know the number of misspelled and correctly spelled words in a document to calculate its percentage of non-stopword correctness. Since Manicure has access to this data as it processes a document, it made sense to include the QC procedure within Manicure.

The intelligence applied in Manicure QC is actually discovered during the automated correction performed by Manicure.

No other QC system can approximate actual OCR document accuracy in the same way Manicure QC does. Figure 2, the pseudocode for the algorithm, shows the additional information provided to and exploited by Manicure QC which vastly improves its ability to verify word correctness.

Stated simply, the QC process takes the number of correct words and misspellings on each page and for the entire document and computes a ratio. The document's accuracy is then compared to a predetermined threshold (set for a particular application). This process is completely automated. Documents that do not pass QC can be marked for subsequent review.

The QC procedure included in Manicure is a nec-

```
Given document D,
Assume w1,w2, . . . ,wn is the list of words in D,

Set CorrectSum (C) to 0;
Set MisspellingSum (M) to 0;

For each word w in D do:
    if w is in the dictionary then
              mark w as ``correct'';
              increment C;

    else if w is an acronym then
              mark w as ``correct'';
              increment C;

    else if w is a proper noun then
              mark w as ``correct'';
              increment C;

    else if w is a stopword (e.g., the, and, of)
              mark w as ``reject'';

    else if w is one of the known patterns (e.g., email,identifiers)
              mark w as ``reject'';

    else
              mark w as ``misspellings'';
              increment MisspellingSum;
    end if
end For;

Set Accuracy Level (AL) to C/C+M;

If AL is greater than Threshold (Td)
  D passes QC
else
    D fails QC;
```

Figure 2: Pseudocode for Manicure QC

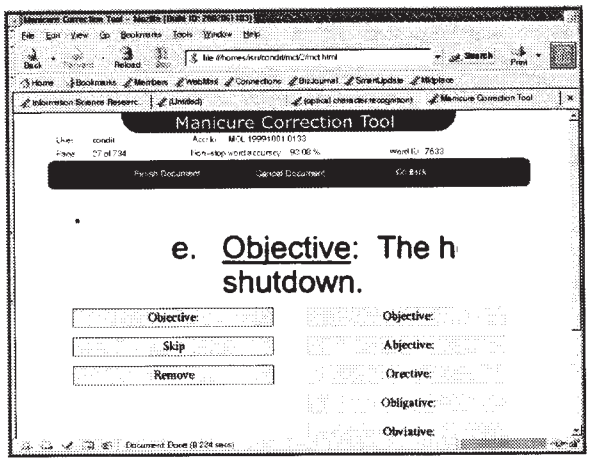


Figure 3: MCT image-based OCR correction

essary prerequisite for documents being submitted to the LSN since based on ISRI research, a document must meet 95% non-stopword accuracy. In some preliminary runs, Manicure QC identified problems with the document conversion process that was causing documents to fail. Further, through verification, we found that if a document passes Manicure QC, it meets the 95% accuracy requirement we believe to be necessary to assure effective retrievability. For a complete description of the Manicure QC evaluation, see [11].

5 The Manicure Correction Tool

MCT is the semi-automated interface that can be used in conjunction with Manicure QC to manually improve documents that fail quality control. The use of MCT would depend on the requirements of a particular application and may or may not be incorporated into the process work flow. But if there are accuracy requirements, MCT provides an easy to use and efficient means for correction.

Documents processed by Manicure are initially scanned to produce electronic images of each page. MCT uses these electronic images in its user interface (See Figure 3). Note that the misspelled word is highlighted on the image in context. Below the fragment of text on the left is the word as it was recognized by the OCR device and on the right is a list of candidate words for correction. A user of MCT need only select the correction from the candidate list and move to the next misspelling. As corrections are made, non-stopword accuracy increases (See counter at top of Figure 3) until the minimum accuracy level is reached.

Behind the candidate list is an elegant algorithm for ordering corrections. It includes dynamic string matching together with a Bayesian technique for selecting OCR confusions. This approach typically produces the most likely candidate at the top of the list.

First, MCT uses information from Manicure to augment the dictionary (acronyms, proper nouns). In this way, the dictionary becomes dynamic and specific to the document being corrected. Further, MCT applies a priori data and Bayesian techniques to build a "learned confusion matrix," which is used to order the corrections. The corrections list will then present the most likely candidates based on the knowledge provided to it from Manicure. Figure 4 presents pseudocode in narrative form for the way MCT orders corrections.

6 Conclusion and Future Work

The process of document conversion is more than the generation of ASCII text for a set of input pages. The process must take into account the eventual application for the generated text. As we emphasized throughout the paper, the quality of the application's effectiveness is heavily dependent on the quality of the text produced. So any large document conversion process must at a minimum have some form of the components discussed in this paper: 1) Method of conversion (scan & OCR) 2) Complete and consistent QC and 3) Automated and/or semi-automated cleanup and correction. Manicure, Manicure QC, and MCT provide this process flow for obtaining the best quality OCR output for any collection with a minimum amount of effort and human intervention.

With our continued research on information access of OCR generated text, we have identified both new areas for research and changes to our systems that would enhance their usability. First, erroneous zoning was a major issue in all our IE experimentation. One open area for research would be the automated correction of incorrect zones. Correcting zoning errors would be a major contribution to information extraction in this domain. Also, Manicure automatically corrects non-stopwords. This of course is crucial for IR but other text elements prove to be important for other IA tasks. Consider the problem of identifying an employee id or a birth date that has been misrecognized. Although Manicure can recognize special strings, it does not attempt to "fix" anything that is not plausibly a word. Fortunately, MCT can be designed to highlight any string in the text. Modifications can be made to MCT to display special strings for verification, including numeric strings.

Our goal is to completely automate the document conversion process. We believe we accomplished this goal for IR. Since other IA tasks may require a very high accuracy rate, the addition of a semi-automated correction system tuned to a specific application is a satisfactory solution for now.

```
Assume document D has failed with m1,m2, ..., mk misspellings,
number of correct words C, and number of misspellings, M, then,

For each misspelling m in D do
    let p1, p2, ..., pt be the list of correct candidates for m;
    re-rank and prune this list using a priori information,
       Bayesian formula, and similarity threshold, Ts;
    sort new p list in order of similarity to m;
    offer new p list to the user for selection with image of m;
    increment CorrectSum, C;
    decrement MisspellingSum, M;

    Set Accuracy Level (AL) to C/C+M;
    If AL is greater than Threshold (Td) break;
end For;
```

Figure 4: Pseudocode for ordering corrections in MCT

References

[1] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM International Conference on Digital Libraries*, 2000.

[2] Sergey Brin. Extracting patterns and relations from the world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*, 1998.

[3] Jack G. Conrad and Mary Hunter Utt. A system for discovering relationships by feature extraction from text databases. In W. Bruce Croft and C. J. van Rijsbergen, editors, *Proc. 17th Intl. ACM-SIGIR Conf. on Research and Development in Information Retrieval*, pages 260–270. SIGIR, Springer-Verlag, July 1994.

[4] William B. Frakes and Ricardo Baeza-Yates, editors. *Information Retrieval, Data Structures & Algorithms.* Prentice Hall, Englewood Cliffs, NJ, 1992.

[5] D. Freitag and A. McCallum. Information extraction with HMMs and shrinkage. In *Proceedings AAAI-99 Workshop Machine Learning and Information Extraction*, 1999.

[6] D. Harman. *Information Retrieval, Data Structures and Algorithms.* Prentice Hall, Englewood Cliffs, NJ 07632, 1992.

[7] T.R. Leek. Information extraction using hidden markov models. Master's thesis, UC San Diego, 1997.

[8] Thomas Nartker, Kazem Taghva, Ron Young, Julie Borsack, and Allen Condit. OCR correction based on document level knowledge. In *Proc. IS&T/SPIE 2003 Intl. Symp. on Electronic Imaging Science and Technology*, volume 5010, pages 103–110, Santa Clara, CA, January 2003.

[9] Department of Justice. The Freedom of Information Act (FOIA), 5 USC Section 552(b)(6). `http://www.usdoj.gov/oip/exemption6.htm`. Viewed Sept. 28, 2005.

[10] G. Salton. *Automatic Text Processing.* Addison-Wesley, Reading, MA, 1989.

[11] ISRI Staff. Measuring and delivering 95% nonstopword document accuracy. Technical Report 2003-04, Information Science Research Institute, University of Nevada, Las Vegas, September 2003.

[12] Kazem Taghva, Julie Borsack, Bryan Bullard, and Allen Condit. Post-editing through approximation and global correction. *International Journal of Pattern Recognition and Artificial Intelligence*, 9(6):911–923, 1995.

[13] Kazem Taghva, Julie Borsack, and Allen Condit. An expert system for automatically correcting OCR output. In *Proc. IS&T/SPIE 1994 Intl. Symp. on Electronic Imaging Science and Technology*, pages 270–278, San Jose, CA, February 1994.

[14] Kazem Taghva, Julie Borsack, and Allen Condit. Results of applying probabilistic IR to OCR text. In *Proc. 17th Intl. ACM/SIGIR Conf. on Research and Development in Information*

Retrieval, pages 202–211, Dublin, Ireland, July 1994.

[15] Kazem Taghva, Julie Borsack, and Allen Condit. Effects of OCR errors on ranking and feedback using the vector space model. *Inf. Proc. and Management*, 32(3):317–327, 1996.

[16] Kazem Taghva, Julie Borsack, Allen Condit, and Srinivas Erva. The effects of noisy data on text retrieval. *J. American Soc. for Inf. Sci.*, 45(1):50–58, January 1994.

[17] Kazem Taghva, Julie Borsack, Allen Condit, and Padma Inaparthy. The effects of OCR errors on short documents. Technical Report 94-10, Information Science Research Institute, University of Nevada, Las Vegas, February 1995.

[18] Kazem Taghva, Allen Condit, Julie Borsack, John Kilburg, Changshi Wu, and Jeff Gilbreth. The MANICURE document processing system. In *Proc. IS&T/SPIE 1998 Intl. Symp. on Electronic Imaging Science and Technology*, San Jose, CA, January 1998.

[19] Kazem Taghva, Jeffrey Coombs, and Ray Pereda. Address extraction using hidden markov models. In *Proc. IS&T/SPIE 2005 Intl. Symp. on Electronic Imaging Science and Technology*, San Jose, CA, January 2005.

[20] Kazem Taghva, Tom Nartker, Allen Condit, and Julie Borsack. Automatic removal of "garbage strings" in ocr text: An implementation. In *The 5th World Multi-Conference on Systemics, Cybernetics and Informatics*, Orlando, Florida, July 2001.

Document Processing and Enhancement

A Document Image Enhancement Module: Perspective Warp Correction

Camille Monnier[1], *Vitaly Ablavsky*[2], *Steve Holden*[3], *and Magnús Snorrason*[4]

[1,3,4]Charles River Analytics Inc.
Cambridge, MA 02138 USA
{cmonnier, sholden, mss}@cra.com

[2]Computer Science Dept. Boston University
Boston, MA 02215 USA
ablavsky@cs.bu.edu

Abstract

Optical Character Recognition (OCR) is critical to document management workflow in both Army and civilian scenarios. It is also one of its weakest links, having no turnkey or standardized solution. Low quality documents imaged under adverse conditions exhibit severe image artifacts, such as uneven illumination, low contrast, speckle noise, degraded character glyphs (broken, joined, aliased), rotational skew, and page warp (perspective skew and curved baselines). These artifacts catastrophically degrade OCR performance. Processing large volumes of such documents using manual clean-up is prohibitively expensive.

We are currently developing a system named AIDA which pre-processes document images (bitmaps) to boost OCR performance. AIDA consists of a set of document image quality metrics, each detecting a specific artifact type, and a suite of corresponding image enhancement algorithms. AIDA's greatest novelty is a knowledge-based approach to automatically determine the optimal enhancement operator sequence and parameter values, based solely on computed quality metrics. In its current stage of development, AIDA successfully detects and corrects artifacts including rotational page skew, perspective skew, uneven illumination and speckle noise in both English and Arabic documents, and has specifically been validated on Arabic datasets of interest to the Army.

We present the method and results from AIDA's perspective rectification module. Documents captured by hand-held devices such as digital cameras often exhibit perspective warp artifacts. These artifacts pose problems for OCR systems which at best can only handle in-plane rotation. We propose a method for recovering the planar appearance of an input document image by examining the vertical rate of change in scale of features in the document. Our method makes fewer assumptions about the document structure than do previously published algorithms.

1. Introduction

When a document image is captured with a hand-held device the relative orientation affects the resulting appearance. If the camera's optical axis is co-aligned with the normal of the document plane, such as in a flatbed scanner, the only artifact will be the rigid rotation. This can be corrected using a variety of methods, such as by examining projection profiles [1], Hough transforms [2,3], by line extraction [4], or by nearest-neighbor searches [5], and some modern OCR systems now include this capability as a pre-processing option.

When the optical axis is at an angle with the normal of the document plane, the deformation is no longer simply a rigid rotation, but becomes a perspective distortion that causes variations in text size perpendicular to the axis of rotation.

Methods to correct perspective warping using vanishing point detection have been developed in both architectural photogrammetry [7,8] and in image document analysis [9], most of which involve extracting multiple straight lines that are grouped and intersected to compute a vanishing point. Unfortunately, while it is generally straightforward to extract lines from architectural imagery, it is much more difficult to reliably extract lines from document images. Although it is possible to detect a horizontal vanishing point along the direction of text baselines [9], most methods for rectifying the vertical (baseline-orthogonal) component of a perspective projection are heavily based on assumptions about document structure. Typical approaches for detecting a vertical vanishing point in image document analysis include:

- Assume that a document page has visible left and right boundaries; use those two lines for finding the second vanishing point
- Assume that the text is justified and use the text edge boundaries to define the vertical vanishing point

These methods place rather strong constraints on the subject matter, and do not take full advantage of the rich information available on the page.

The deficiency of the above methods is illustrated in Figure 1 below:

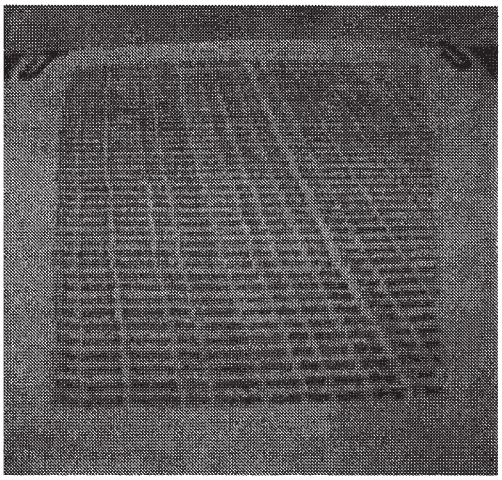


Figure 1: Counter-perspective illusion

The document in Figure 1 has uneven left and right margins and when viewed at an angle appears to have no perspective distortion at all. Although "illusory cues" can be found in this particular instance because each line happens to repeat the same sentence, one can trivially construct examples where such cues would be absent.

2. Overview of the approach

Figure 2 below outlines the algorithm and the subsequent chapters explain the ingredients.

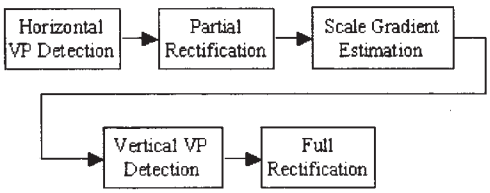


Figure 2: Algorithm description

Text lines are grouped together and their intersection point is used to compute the first vanishing point, which is sufficient for rectification of the horizontal distortion. Next we examine the scale gradient along the direction perpendicular to text baselines to estimate the second vanishing point, which will complete the rectification process.

3. Detection of the Horizontal Vanishing Point from Text Baselines

Rectification of perspective distortion has successfully been accomplished by detecting vanishing points in a scene. Work has been done regarding the 3D reconstruction of structure from imagery containing man-made architectures via the use of vanishing points [7,8]. In the case of document images, where long, straight edges are rare, it is required to perform additional analysis to extract the visual cues necessary for vanishing point detection. We have implemented a version of Pilu's algorithm [9] for detecting and grouping text lines in order to compute the horizontal vanishing point and rectify the perspective distortion along text lines (Figure 3).

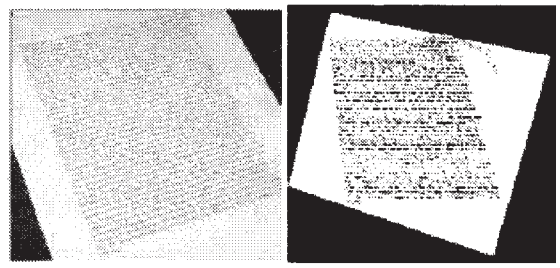


Figure 3: Rectification with respect to the first vanishing point

4. Perspective Rectification from a Vanishing Point

Given the location of a vanishing point $\mathbf{V} = \begin{bmatrix} V_x \\ V_y \end{bmatrix}$, it is possible to write a planar homography that corrects the deformation produced by a perspective projection up to an aspect ratio [7,9]. In the case of a document image, we expect to detect at most two vanishing points in the directions parallel or perpendicular to text baselines. We compute the homography for correcting the distortion in the direction of the text lines as in [9]. Pilu writes the homography as a rotation to align the vanishing point with the image X-axis followed by a transform $\mathbf{M}_x$ that places the detected vanishing point at infinity along the X-axis (1).

$$\mathbf{M}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{V_x} & 0 & 1 \end{bmatrix}, \quad \mathbf{M}_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{V_y} & 1 \end{bmatrix} \tag{1}$$

We compute the homography for the vanishing point perpendicular to the text lines as a translation along the X-axis to set the vanishing point's X coordinate as the origin, followed by a transform $\mathbf{M}_y$ that places the detected vanishing point at infinity along the Y-axis (1).

5. Feature Extraction

We assume that text lines may be non-justified, and that line endpoints are thus unreliable for computing the vanishing point. We instead rely on text baselines as our primary source of data, choosing as features the average inter-line distance and average line height (scale), and the rate of change of scale over the image surface (texture gradient).

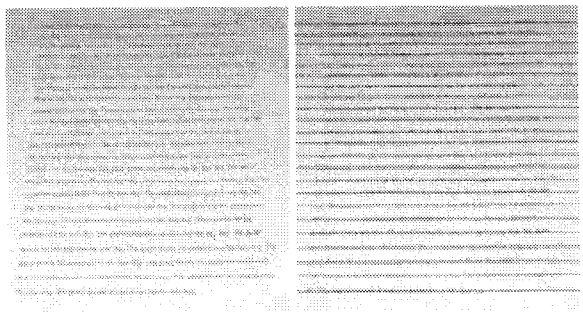


Figure 4: Camera image of a slanted (?=42°) text document (left), and recovered text baselines (right).

5.1. Baseline Detection

We estimated baselines by implementing a variation on Pilu's method [9] for detecting text lines in a document image. The image is binarized and processed for connected components, which are then grouped using a line-growing algorithm (Figure 4). While this is not necessarily the most robust baseline detection method available, it is useful because it extracts the features we need for detecting both the horizontal and vertical vanishing points.

5.2. Line Spacing Gradient Estimation

We estimate the gradient of inter-line spacing as a linear function over image pixels along the y image axis. If a horizontal vanishing point is detected, the correction is applied prior to estimating this metric. In order to reduce noise, we fit the y-locations of detected baselines to a linear function of the form $\Delta y' = ay' + b$; this mitigates the impact of outliers on the final estimation, and helps smooth noisy data points (Figure 5). Note that any known uniform distance in the document (such as text height) may be used to estimate the gradient, though larger features tend to provide much more stable estimates.

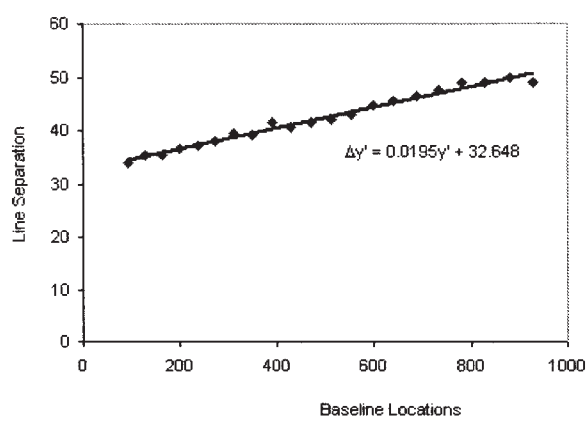


Figure 5: Inter-baseline distances of the above document image, γ = 0.0195

6. Vertical Vanishing Point Estimation

We approach the problem in terms of the measurable quantities described in Section 5. Our solution assumes that text lines are for the most part evenly spaced in the original document, and that the document image is approximately centered about the camera's optical axis.

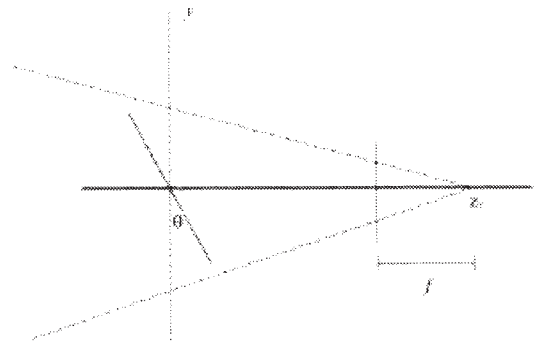


Figure 6: Geometry of the perspective projection of a rotated plane

Treating the deformation as the perspective transformation of a page in the XY plane that has been rotated an angle ? about the X-axis (Figure 6), and discarding the Z component, we obtain the 2D deformation:

$$\begin{bmatrix} x' \\ y' \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & 0 \\ 0 & \frac{-\sin\theta}{f} & \frac{z_c}{f} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2}$$

Where z_c is the camera's position along the Z-axis and f represents the distance from the camera to the image plane. Writing y in terms of discrete lines, with line number n and uniform spacing d, we obtain

$$y = nd \tag{3}$$

which we can substitute into (2) to generate the relationship

$$y' = \frac{fnd\cos\theta}{(z_c - nd\sin\theta)} \tag{4}$$

The vanishing point along the Y-axis can be defined as the mapping of a point at infinity to a finite point in the image plane. Taking the limit of (4) as n goes to infinity, we obtain the equation for the vanishing point

$$y'_{vp} = \lim_{n \to \infty} y' = -f \cot\theta \tag{5}$$

We wish to describe the vanishing point in terms of measurable quantities $\Delta y' = \frac{dy'}{dn}$ (the separation distance between distorted lines) and $\gamma = \frac{d(\Delta y')}{dy'}$ (the rate of change of the separation distance between distorted lines). Setting

$$\Delta y' = \frac{dy'}{dn} = \frac{f d z_c \cos\theta}{(z_c - nd\sin\theta)^2} \tag{6}$$

and

$$\frac{d(\Delta y')}{dn} = \frac{2 f d^2 z_c \sin\theta \cos\theta}{(z_c - nd\sin\theta)^3} \tag{7}$$

Combining (6) and (7) we can write

$$\gamma = \frac{d(\Delta y')}{dy'} = \frac{d(\Delta y')}{dn} \cdot \frac{dn}{dy'} = \frac{2d\sin\theta}{(z_c - nd\sin\theta)} \tag{8}$$

We can then rewrite equation (5) in terms of our measurable quantities $\Delta y'$ and γ by substituting equations (6) and (8) into an expression for $\cot\theta$:

$$\cot\theta = \frac{2\Delta y'(z_c - nd\sin\theta)}{f\gamma z_c} \tag{9}$$

Substituting into (5), we have

$$y'_{vp} = \frac{2\Delta y'(z_c - nd\sin\theta)}{\gamma z_c} \tag{10}$$

Because we assume that the document image is approximately centered about the optical axis, it is reasonable to state

$$z_c >> nd\sin\theta \tag{11}$$

for points near the origin. We thus obtain our estimate for the vanishing point,

$$y'_{vp} \approx \frac{2\Delta y'}{\gamma} \tag{12}$$

where $\Delta y'$ is the scale value near the origin, and γ is the rate of change in scale in the image. Applying the vanishing-point correction, we obtain the rectified image (Figure 7). Note that the vanishing point is given with respect to the origin, which is assumed to be near the center of the image. For documents that have a significant horizontal skew (Figure 9), it is necessary to estimate the vanishing point's horizontal offset. We are able to estimate the vanishing point's horizontal offset using a single vertical cue, such as left-most baseline endpoints. While we avoid searching for vertical cues in the computation of the vanishing point's Y-component, we only require a single vertical cue to determine the corresponding X-coordinate.

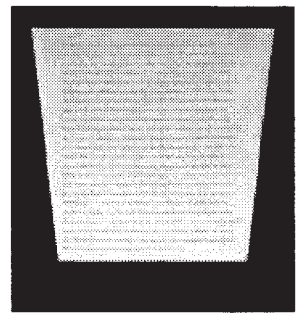


Figure 7: Rectified document image

7. Results

Our method for the rectification of perspective-skewed document images works well over a large range of slant angles (Figure 8), and requires only a single vertical cue to successfully rectify arbitrary perspective distortions (Figure 9). Our preliminary results indicate that a significant boost in OCR performance can be expected for documents rectified in this manner (Table 1). While very little difference in performance was observed at small skew angles, at larger angles OCR performance increased by as much as 5% in our trials. In the sample set shown below, a slight loss in performance (0.2%) is observed in the small-angle case due to blurring caused by the transformation. Improvement in OCR performance across various document images is directly related to the proportion of text that lies furthest from the camera (and is thus the most affected by the perspective skew). Consequently, at skew angles exceeding 35°, we notice an attenuation of OCR improvement as the effective resolution of the text furthest from the camera is often too low for warp correction alone to aid recognition. In practice, our OCR did not exhibit difficulty in recognizing text closest to the camera in all but the most severely skewed images.

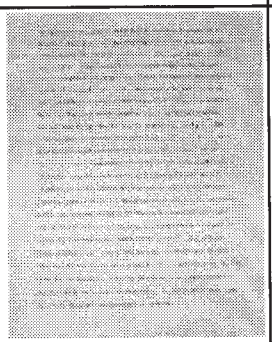
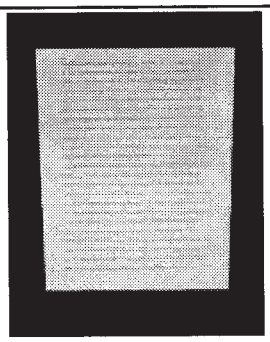
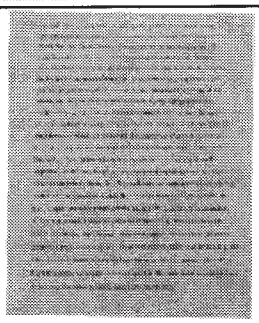
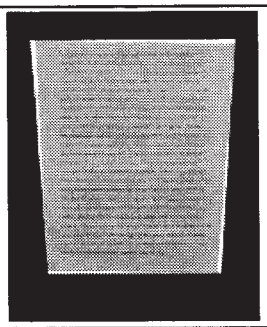
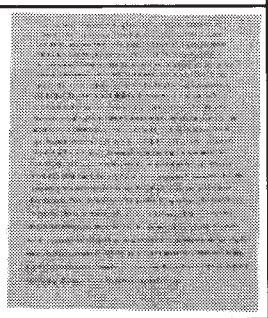
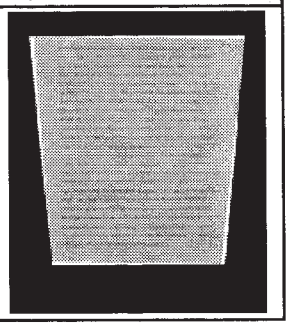
Param.	Original	Rectified
$\theta = 10°$ $\gamma = .0067$ $\Delta y' = 50.6$		
$\theta = 22°$ $\gamma = .0127$ $\Delta y' = 45.6$		
$\theta = 34°$ $\gamma = .0171$ $\Delta y' = 40.7$		

Figure 8: Secondary rectification of document images slanted about the X-axis

Table 1: Sakhr Automatic Reader's performance before and after rectification

Skew Angle	OCR Performance (Skewed)	OCR Performance (Rectified)
10°	93.3	93.1
22°	85.4	90.9
34°	81.0	85.9

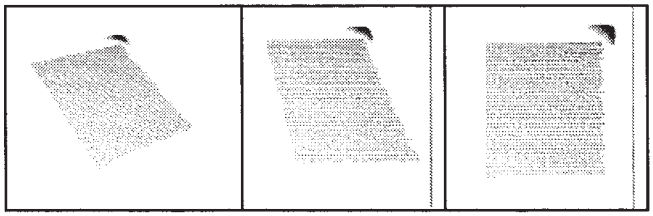


Figure 9: Binarized document image taken from an arbitrary perspective (left), partially rectified document image (middle), fully rectified image (right)

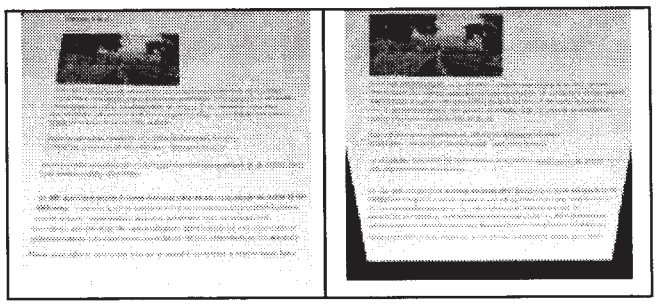


Figure 10: Document image at 60° (left), rectified image (right)

8. Conclusions

We present a method for rectifying perspective distortion in the direction perpendicular to text lines in document images. Provided the original document has a reasonably well-distributed number of evenly spaced or uniformly tall text lines (Figure 10), our algorithm is capable of reliably correcting the distortion with no additional input. In cases where the image's center does not coincide with the optical axis, a vertical text cue must be detected to determine the vanishing point's horizontal offset. While it is reasonable to assume that most text is left-justified (right-justified in the case of Arabic), we are pursuing a method to more robustly estimate a scale gradient along the X-axis. Additionally, while our method is numerically stable with respect to noise, it is likely that its performance can be further improved by exploiting an alternate estimate of the texture gradient such as scale-space analysis [6].

9. References

[1] A. Bagdanov and J. Kanai, Projection Profile Based Skew Estimation Algorithm for JBIG Compressed Images, *Proceedings of the 4th International Conference on Document Analysis and Recognition*, Ulm, Germany, 1997, pp. 401-405.

[2] H.-F. Jiang, C.-C. Han, K.-C. Fan, A Fast Approach to the Detection and Correction of Skew Documents, *Pattern Recognition Letters*, 18 (7), 1997, pp. 675-686

[3] Amin, A., Fischer, S., 2000. A document skew detection method using the Hough transform. *Pattern Analysis and Applications* 3 (3), 243-253.

[4] Cao,Y., Wang, S. Li, H., Skew detection and correction in document images based on straight-line fitting, *Pattern Recognition Letters* 24 (12), 2003, pp. 1871-1879

[5] Lue, Y., Tan, C.L., A nearest-neighbor chain based approach to skew estimation in document images, Pattern Recognition Letters, 24 (14), 2003, pp. 2315-2323

[6] Lindeberg, T. "Scale-Space Theory: a Basic Tool for

Analyzing Structures at Different Scales," *Journal of Applied Statistics*, vol. 21, no. 2, pp. 225-270, 1994.

[7] Liebowitz D, Criminisi A, Zisserman A. Creating architectural models from images. In *Proc. Eurographics*, Milan, Italy, Sept. 1999, pp.39-50.

[8] F.A. van den Heuvel, Vanishing point detection for architectural photogrammetry, Int. Arch. Photogrammetry Rem. Sens. 32 (part 5) (1998) 652-659.

[9] Pilu, M. "Extraction of Illusory Linear Cues in Perspectively Skewed Documents," *CVPR 2001, also a Hewlett-Packard UK tech report.*

The research reported in this document/presentation was performed in connection with contract/instrument DAAD17-03-C-0009 with the U.S. Army Research Laboratory. The views and conclusions contained in this document/presentation are those of the authors and should not be interpreted as presenting the official policies or position, either expressed or implied, of the U.S. Army Research Laboratory or the U.S. Government unless so designated by other authorized documents. Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

Document Recognition and Translation with Handheld Mobile Devices

Esin Darici Haritaoglu **Ismail Haritaoglu**
Polar Rain Inc.
465 Fairchild Drive Suite: 226
Mountain View, CA 94043

Abstract

We describe a mobile system which is capable of taking a pictures of a document or a sign (written in foreign language) with digital camera attached to commercial off the shelf (COTS) mobile digital personal assistant (PDA) and process the captured image to extract foreign text from images, recognize text (Latin and Arabic alphabet), and translate the document into English so that Soldiers can understand what is written on these documents or information signs without requiring a wireless connection. We have achieved more than 98% accuracy for both scripted Arabic and Latin OCR engines and overall process including detection, extraction, recognition and translation takes less than 15 second for a half-page document written with 12 point fonts.

1 Introduction

In this paper, we would like to introduce our mobile document recognition and translation system and challenges we have faced and solutions we have provided during development of our system[1] Our system consists of a PDA device and attached digital camera. All processing modules including Latin or Arabic OCR and multilanguage translation engines are working on PDA and do not require any wireless communication to complete the process. Our system is capable of providing OCR and translation results less than 15 second for half page documents. Example of sign translation is shown in Figure 1.

One of the most challenging parts was that all solutions for OCR and translation should fit into the small memory space and limited processing power of mobile devices. Additionally, the image quality of available PDA based digital cameras and limited camera operation (manual focus, handheld motion while capturing images) makes the problem more challenging. We will explain our mobile Arabic and Latin OCR and translation algorithm which yield very accurate OCR and domain specific translation results.

Soldiers engaged in battlefield operations in foreign countries encounter a variety of foreign language documents and signs of unknown content. These documents and signs might be relevant to the Soldier's current mission or to the overall battlefield process. Currently US Department of Defense and Security agencies interest is to use such mobile translation device in foreign countries to translate documents and road/warning signs so that Soldiers are able to read road/information signs and documents and quickly triage and translate foreign documents so that mission relevant documents and information can be utilized in a timely manner.

Our system permits the user to translate captured foreign documents quickly so that mission relevant documents can be utilized in a timely manner, and documents containing information relevant to commanders on the battlefield can be passed to the appropriate evaluation personnel. The overall concept is to prevent documents containing time-critical information from being combined with large quantities of other documents that will be stored for future processing.

In addition to a document translation system, our prototype device will be capable of permitting users to capture road/warning/information sign and documents and translate to English quickly so that users can read or interpret foreign sign containing warning, navigation or other information relevant to mission. The translation tool will help user for accurate decision making by providing translation when and where they need most.

Core elements of our technology are the ability to extract text information accurately and recognize foreign text quickly with a limited resource mobile device. We have developed very fast and accurate image processing algorithm to be able to extract text information reliably

[1] The research reported in this document/presentation was performed in connection with contract/instrument W911QX-04-C-0015 with the U.S. Army Research Laboratory. The views and conclusions contained in this document/presentation are those of the authors and should not be interpreted as presenting the official policies or position, either expressed or implied, of the U.S. Army Research Laboratory or the U.S. Government unless so designated by other authorized documents. Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

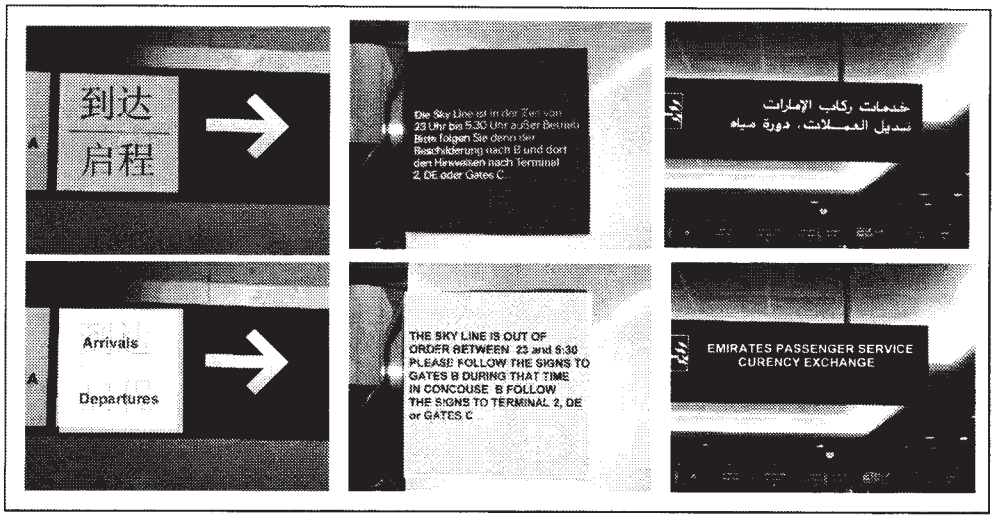


Figure 1: Example of sign translation with mobile devices to help navigation

from pictures and documents and recognize each character or word in the text. It does not require any special hardware and works on a COTS PDA.

Once foreign text detected and characters and words are extracted, they are recognized using Polar Rain's mobile OCR engine. Polar Rain's Mobile OCR engine is capable of recognizing machine printed Latin, Arabic and Cyrillic documents and road/information signs. It is capable of generating acceptable and usable OCR results as small as 125 dpi document captured with PDA's digital cameras. Once the text recognized it is translated to English. Polar Rain has been developing custom domain-specific and generic mobile translation engines which is capable of translating Spanish, Arabic, Serbian to English.

All operation is working on PDA with small digital camera and Soldiers are able to get translation ready in less than 15 second. All processing is done in mobile device without requiring an wireless connection. End users can use the device for any road/sign information translation as simple as taking a picture for other purposes, such as traveling and navigation in foreign in foreign countries

In this paper, we will first introduce our system hardware and some of the limitations in Section 2. Then in Section 3, we will go over our Latin and Arabic OCR engines that we have developed and their accuracy for test databases with different size text. In section 3, we will explain mobile translation and triage engines that we are using in the system and performance analysis for different sets of documents and road images.

2 Mobile Translation and Triage System without Wireless Connection

Our goal was to develop an image based document/text translation and triage system using mobile device with which users only need to capture written document or information/or road sign and our system provides translation of written text in those images. Although image capture and text detection, OCR and translation technologies has been investigated for at least 25 years, the challenging problem is to make the system running on a small mobile device which has many restrictions such as limited main memory, and limited processor power. This makes the problem harder and selection of algorithm is crucial to obtain acceptable performance, e.g., complete the processing time in an acceptable amount of time with acceptable OCR and translation accuracy. In addition, we would like to implement the system without any wireless connection. If the system has a wireless connection the images can be transferred to host computer where image processing and translation are applied and results are returned to user over wireless link. Although this approach is good for both commercial and military application [9, 10, 8], the wireless link may not be available all the time on operation fields therefore those systems may not be useful if there is no wireless connection.

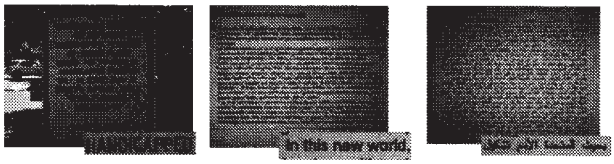


Figure 2: Example of images captured with VEO 1.3MP PDA camera

We have implemented our system for Commercial of the shelf (COTS) Personal Digital Assistant (PDA) which has attached digital camera. Although there are sufficient alternatives for PDA's on the market, there is only one or two good quality digital camera which can be attached and used with PDA's. In our system we have chosen VEO 1.3MegaPixel SD digital camera with manual focus. We can capture up to 1280x1024 pixel resolution. Although there is some color quality problem, it seems that VEO 1.3 SD camera gives the best per-

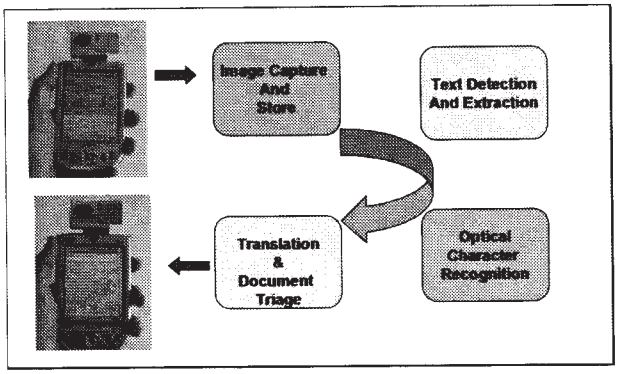


Figure 3: Mobile Document Translation Device and Modules

formance in our application over other alternatives. Recently, some of the PDAs has built-in camera on board but none of them has ability to focus document images in short distance. VEO allows us to adjust the focus so that we can capture document/pages in short distance (as small as 8 inches) so that we can capture quarter of a page in sharp focus. In Figure 2, an example of document and road sign images taken with VEO are shown.

Figure 3 shows overall system diagram. It has four main modules:

- **Image Capture and Store** Our system allows users to capture images with 1280x1024 resolution. Inherently, motion blur and de-focus are two important problem while capturing image with VEO or other mobile digital cameras. Our system has image enhancement modules to determine the amount of blur and de-blur some parts if necessary.

- **Text Extraction and Layout Analysis** Our system has two main text detection and extraction modules for road/information signs and document pictures. Road sign detection modules allow us to detect text on road and information signs which has various color of the text, orientation and illumination. Road sign detection and extraction modules use sophisticated algorithms to detect scene text which is inherently difficult to detect. Our system uses very fast text detection and extraction modules for document images. Document images has more text and less non-text regions and detection and extraction of text does not required sophisticated image enhancement to extract the text. Our system has text detection and extraction for both Arabic and Latin text which uses different approaches to extract the each character as Arabic text has scripted nature which is different than Latin text.

- **Character Segmentation and Recognition** We have developed two different OCR for mobile devices: Latin and Arabic OCR. Both OCR uses shape similarities, size and font sensitive information while recognizing character. They also use some language depended features to increase the accuracy. We have tested both OCR engines using both road signs and document images and we have obtained more than 98% OCR accuracy for Latin and Arabic text.

- **Text Translation and Triage** We have developed multi language translation from Spanish, Arabic and Serbian to English. In addition we have developed mission sensitive dictionary which allows us to triage documents/text quickly.

3 Text Detection and Extraction for Document and Road Signs

As text detection and extraction in document images is slightly less complicated than text detection and extraction in road sign images, we have developed two different approaches to detect text in road sign and document images. Main motivation is that road/information images have less text but complicated background difficult to detect, however document images has less difficult background but more text. Therefore we would like to allocate more time to detect text in road/information sign processing but less time for recognition. In contrast, more time for recognition and less time for detection in document images.

We have developed very robust but computationally expensive text detection and extraction for road sign images. We have applied this method to text images without any major changes and we have obtained good results. However we have noticed that we can use less expensive vision techniques to obtain the similar results on document images in less amount of execution time. For example, the assumption we can make is that the document images are not as complicated as road signs and there is unique background color (such as white) and unique foreground color (such as black). In addition, the quality of the VEO images makes the text extraction little easier. We have developed two different text detection and extraction algorithms (a) Road Sign Text Detection (RSTD) (b) Document Text detection (DTD). As road sign images may have very complex background and variation, we prefer image enhancement before text detection and segmentation. We apply the hierarchical connected components algorithm to obtain segmentation for road sign images. Symmetrical Neighbor Filter (SFN) enhancement was one of our innovation for road sign detection which we implemented in a fast way to achieve both memory and processing efficient version . It helps to smooth the images by removing the interior region noise while sharpening the edges between the complex background and text regions. Figure 4 shows an example of road sign detection and recognition obtained with our system

However we do not need such complicated algorithm for text documents. We have developed a fast appear-

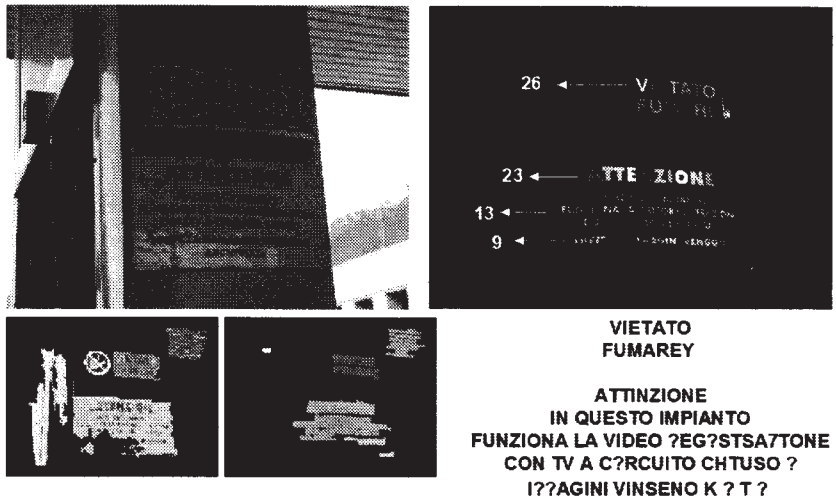


Figure 4: Example of text detection and recognition in road sign images

ance based method which is applied to entire image and utilize local contrast between text and background. Instead of single global thresholding method (which tends to fails when there is camera noise or imager problem), we compute a threshold locally using text appearances. This method yields very good detection and extraction method for document images but does not perform well for road sign detection. Figure 5 shows the performances of both detection method on a PDA with 624 Mhz xScale processor.

4 Optical Character Recognition on Mobile Device

We have implemented shape based optical character recognition for Latin and Arabic text. In order to use both memory and processor power efficiently, we have developed an adaptive feature based recognition algorithm based on character shape invariant and shape depended features. One of the challenging part of the OCR is to develop such recognition system on limited memory and processor power.

4.1 Latin OCR

we have implemented 2D normalized shape template matching method. The basic idea behind that method is to defer the computationally expensive feature comparison as much as possible. We first use computationally fast features to eliminate the non-similar characters and then apply computationally expensive features only to those characters that pass the first level.

Feature selection is critical to obtain a good recognition rate. However, as we are targeting to implement it for PDA, memory usage and speed are other important factors. While the selected features should give the most distinguishable power for all the character that we need to recognize, the recognition and comparison algorithm should be fast enough to produce the results in limited time and the feature space should not require too much memory. After conducting extensive experiments, we selected 2D-shape histograms which describe the relative orientation and distance between the points in the characters, as main feature for recognition engines. We trained the system using an adaptive training method. We used scene-characters that are detected in image databases in training sets. If there is not enough number of characters, we combined computer generated-characters (where the documents are printed and captured by the camera instead of captured on road sign). We trained 53 characters using more than 6250 single character with different size. Each characters in the training set has 45-55 sample using different fonts and different size (8-72 pixel in height).

We have conducted many test and training session to understand the performance of the recognition system for various Latin characters which is written in different fonts and different sizes. We have used four main fonts (Arial, Verdona, Helvatica, Tohoma) and character sizes (10 point to 56 point) while we are generating characters. We have printed text documents in a one page and captured the entire page with a single image with 640x480 and 1280x1024. This setting provided us almost 75dpi resolution for 640x480 image and 150dpi resolution for 1280x1200 resolution. As we are processing pixel (not the points), we use number of pixel as size for the characters. For example a character printed with 12,16,24,32 point font size appears in document images with 6,9,14,18 pixel in height size for 640x480 document images, 14,20,30,38 pixel in height size for 1280x1200 images. Generally, our recognition system works good for characters which has pixel size 15 or higher.

4.1.1 Latin Character Recognition Performance Evaluation

We have generated three sets of character data for training and testing (a) characters extracted from docu-

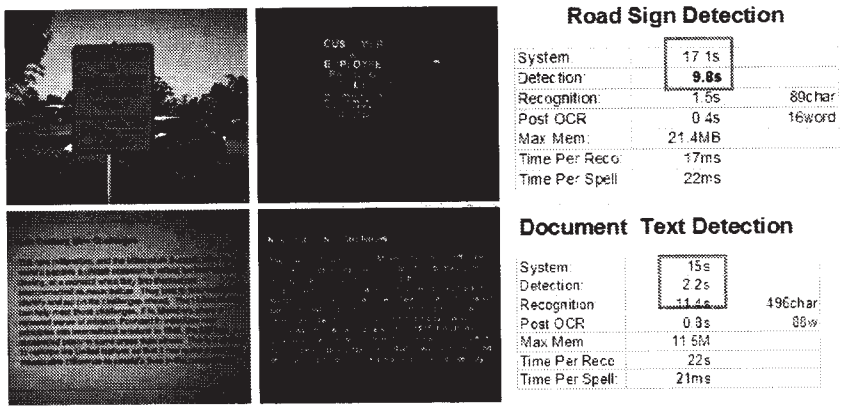


Figure 5: Text detection and extraction performance on a PDA which has 624 Mhz intel Xscale Processor

ment images which is generated randomly and printed 10-18 font size with four different fonts (b) characters extracted from road sign images (used in highways and outside information) (c) characters extracted from document images which contains English text and printed with 10 or 12 points single font. The first set has almost 7600 characters (four different font type with different apparent sizes changing from 8 pixel to 85 pixel in height). The second set has more than 8000 characters and written in the sign with different fonts (we have observed that the font variation is not significant as most of the road has consistent font type). A third set consists of 16235 characters and contains single font. An example of images are shown in Figure 6.

Below are the some character recognition results we have obtained. On the average we have obtained 98.7% recognition accuracy for document images. We have obtained slightly lower recognition rate 97.8% for road sign images. We observed that characters extracted from road sign images have more variation than characters extracted from document images. We were able to obtain recognition rate of almost 99.2% for character Set-II which has apparent size smaller than 19 pixel

	All Sizes	$(s \leq 16)$	$(s \leq 30)$	$(31 \leq s)$
Set I (CG)	0.982	0.929	0.980	0.995
Set II (RS)	0.968	0.968	0.966	0.971
Set III (UN)	0.984	0.991	0.993	0.974

Table 1: OCR accuracy of our system for Latin alphabet

As it is seen above, the best recognition performance is obtained with 10-12 point size printed UN document images. Extracted characters have small appearance size however they only contain single font type.

Overall we have reduced the detection processing down to 1.7 second on the average using 624Mhz xScale based PDA with Intel Compiler. As you can see from the results, the detection process is almost fixed and does not depend on how many characters are extracted from the image. However, recognition process is directly related to how many characters are extracted from the image. For every character extracted by detection process, recognition function is executed, therefore the more character the image contains the longer the overall

Figure 6: The system has been tested by using two set of image database from which each character is extracted (a) sample road sign images (b) sample document images

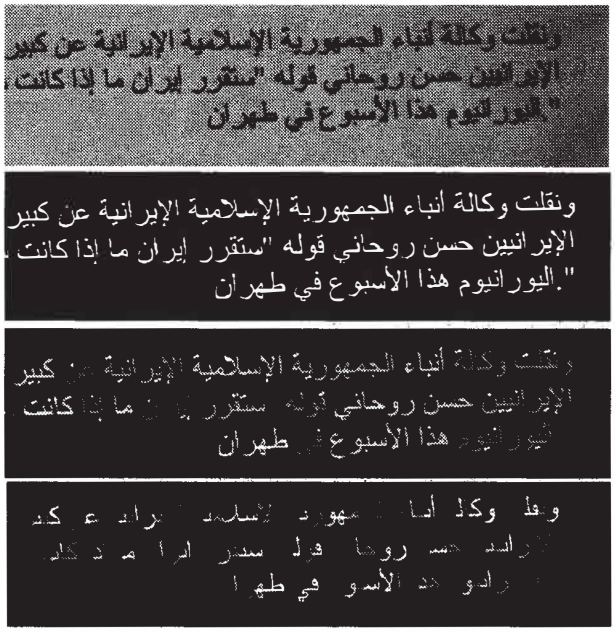


Figure 7: An example of Arabic character recognition process: original text image (top row), detected text only regions (second row), word segmentation (thirds row) and final Arabic character segmentation (last row)

recognition time. In other words, the recognition-time is shorter for images which contain bigger apparent size character than the images which contain smaller apparent size characters. One of the important speed measures is processing time per single character recognition. With our method, we are able to recognize single character in 8 ms on the average.

4.2 Arabic OCR

In addition to Latin OCR, we have developed fast and accurate Arabic OCR which is able to recognize scripted Arabic characters and words. In summary, we first detect the Arabic text in document images and extract the Arabic text from images. Then we apply connected components algorithm to find each connected components. Each connected component may contain single or multiple Arabic characters or a sub-piece of an Arabic characters. After we obtain the connected components, we segment words. After the words are detected we further apply Arabic character segmentation algorithm to detect each individual Arabic character. Example of Arabic character recognition and dot integration are shown in Figure 7.

There are mainly 26 Arabic characters and 10 Arabic digits in Arabic alphabet. However, because of scripted nature of Arabic language, each character can have 4 different character shape depending on it's position (isolated, middle, initial or end character). In addition, some of the basic Arabic characters can take some diacritics, dots etc. These dots make the character totally different character. Therefore Arabic characters without dots may have many ambiguities in recognition. In order to build a robust recognition system for Arabic, we have develop Arabic characters with and without dot information. We trained our system to recognize Arabic OCR characters. Each Arabic OCR character may represent a single Arabic text Character or multiple Arabic text characters depending on the position and whether or not it has dots (Arabic Text characters are the ones used in Arabic text). In our recognition system, we provide OCR-Character to Text character mapping, and if there are one-to-many type of mapping, dot classification algorithm helps to resolve disambiguates.

4.2.1 Performance Evaluation

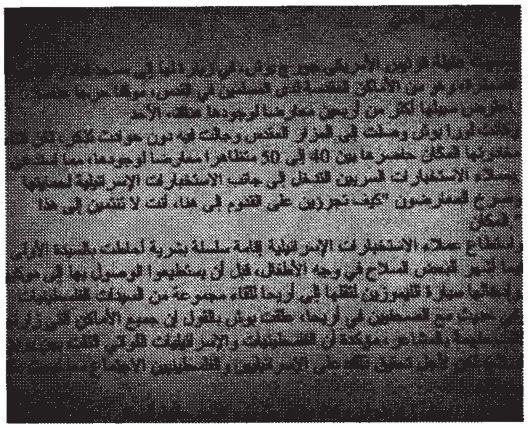
We have conducted extensive tests to evaluate the Arabic character segmentation and recognition accuracy. We have generated four different test sets, each test set have word-level ground truth data that allows us to compare the results. We have computed three accuracy metrics: character recognition accuracy(OCR), word recognition accuracy without any spelling correction (WORD-NSP), word recognition with Spelling(WORD-SP). Here is the summary of the results:

- **set-I**: Set-I contains 21 document images taken with Veo camera with 1280x1024 resolution. The Arabic text are CNN news segments, UN documents etc and printed on 12 and 16 point size Arabic fonts (roughly equal to 150dpi). It contains 4049 scripted Arabic words and 19210 characters. The words and characters appear on document images with various apparent sizes, where avg. character size is 17 pixel.

- **set-II**: Set-II contains 13 document images taken with Veo camera with 1280x1024 resolution. The Arabic text are CNN news segments and printed on 16 point size Arabic fonts and cover half page document(roughly equal to 175dpi). It contains 1927 scripted Arabic words and 9043 characters. The words and characters appear on document images with various apparent sizes, where avg. character size is 18 pixel.

- **set-III**: Set-III contains 11 document images taken with Veo camera with 1280x1024 resolution. The Arabic text are CNN news segments and printed on 12 point size Arabic fonts and cover half page document(roughly equal to 145dpi). It contains 1735 scripted Arabic words and 8227 characters. The words and characters appear on document images with various apparent sizes, where avg. character size is 14 pixel.

- **set-IV**: Set-IV contains 31 document images obtained with scanner at 300 dpi resolution. This set allow us to compare our results with commercial Arabic OCR products. The Arabic text are CNN news and UN documents and printed on 12 and 16



Kenai: OCR

نورا بوش في القدس
واجهت عقيلة الرئيس الأمريكي جورج بوش في زيارة لها إلى مسجد قبة القدس
الصخرة وهو من الأماكن المقدسة لدى المسلمين في القدس موقفا حرجا عندما
اعترض سبيلها اكثر من أربعين معارضا لوجودها هناك احد
وكانت نورا بوش وصلت إلى المزار المقدس وجالت فيه دون حوادث يذكر لكن أثناء
مغادرتها المكان حاصرها بين لا إلى حال متظاهرا معارضا لوجودها
بعلاء الاستخبارات السريين التدخل الى جانب الاستخبارات الاسرائيلية لحمايتها
وصرخ المعارضون كيف تجروين على القدوم الى هنا ان ال تنتمين الى هذا
المكان
واستطاع عمال الاستخبارات الاسرائيلية اقامة سلسلة بشرية أحاطت بالسيدة الأولى
ما شهر البعض السطح في وجه الطقاء قبل أن يستطيعوا الوصول بها إلى موكبا
وادخالها سيارة الليموزين لنقلها الى أريحا للقاء مجموعة من السيدات الفلسطينيات
في حديث مع الصحفيين في أريحا علقت بوش بالقول إن جميع الأماكن التي زار
تن ملهمة بالمشاعر مؤكدة أن الفلسطينيات والاسرائيليات اللواتي التقت بهن يردن
لسلاء نكن الرجل تحقيق ذلك على الاسرائيليين والفلسطينيين اجتماع معا لبحث ثم
وتحقيقه
كانت نورا بوش زارت في وقت مبكر من يوم الحد حائط المبكى احد الأماكن

Dict: CNN (22K) **in second**

System	10.2
Detection	0.7
Recognition	6.4
Spelling:	2.3
Translation	0.5
Triage:	0.3
# of Char	846
# of Word	174
# of Spelling	43
# of Translation	198

9.2 MB Memory

Kenai-P: Translation

Laura bush at jerusalem faced [Eqylp] us president bush in visit its to mosque bloc jerusalem dome he is of holy places at muslims at jerusalem stand critical when objected recommendations from more than one forty course such there minimum had Laura bush amounted to [AlmzAr] the [wjAlt] its without accidents the the mentioned but during favours place [HASrhA] between not to elevated [mtZAhrA] course such thus summoned [bEmAlAl'] intelligence secret of to mention intelligence [A<lsrAnylyp] accomplishment voice oppositionists how [tjr&yn] on come to here the the [tntmyn] to such a place managed [EmAlAl'] intelligence [A<lsrAnylyp] set series human advised sayeda first the month some bek in they [AAlTfAl'] by that able reach by to [mwkbA] kholi cars [Altymwzyn ltqlhA] to jericho meet a package of women participating in an interview with press in jericho was completed suspended bush saying that all places [Alty] visited would [mlhmp bAlm$AEr] they affirmed that that behaviour participating [wA<lsrAnylyAt] forced al victim wear al-alam but man achieve a on [A<lsrAnylyyn] palestinians meeting together discuss the warned [wtHqyqh] had parker bush visited passed time early of on minimum time yard targets places

Figure 8: Example of results obtained with Polar Rain's Kenai-II Mobile Arabic to English Translation and Triage system. The execution times are in second and obtained with a PDA which has 64MB main memory and 624Mhz xScale processor

point size Arabic fonts. It contains 7695 scripted Arabic words and 34477 characters. Average character size is 25 pixels.

Using above test sets, we have achieved following ocr and word accuracy:

	OCR.	WORD-NSP	WORD-SP
Set I	0.989	0.906	0.946
Set II (12)	0.972	0.775	0.893
Set III (16)	0.929	0.544	0.757
Set IV (scan)	0.991	0.965	0.977

Table 2: Arabic OCR accuracy of our system

Our Arabic OCR is also very fast and runs on a 624 Mhz PDA with 6.2 second per image (per half page) which is rough 7.5 ms per character. Figure 8 and Figure 9 shows examples of Arabic and Spanish OCR and translations results with execution time statistics.

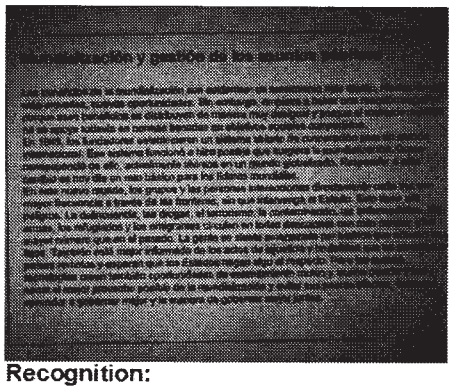
5 Document Translation and Triage

Overall goal of our system is to provide useful translation and triage results so that Soldier in the field can easily understand what is written on any information sign or on a piece of document that are written with a foreign language.

We have built Spanish to English and Arabic to English machine translation system for PDA. Translation engines runs fast enough to provide acceptable text translation less than 5 second (on the average) for text with 300 words. We have conducted offline training by using parallel sentences for documents and road signs in both Spanish and Arabic. In addition we have built transliteration tables in case the recognized word is not in our dictionaries. In addition, we have built necessary tools to support any mission specific translation and triage processing and fast new mission deployments. Currently we are conducting accuracy of our translation engines by comparing human translator and commercial translation engines. An example of Arabic translation obtain by our system, SYSTRAN and human operator are shown in Figure 10.

6 Conclusion

We have described our mobile document recognition and translation system (Kenai-II) which works on a COTS PDA with attached digital camera. Core elements of our technology are the ability to extract text information accurately and recognize each foreign text quickly with a limited resource mobile device. We have developed very fast and accurate image processing algorithm to be able to extract text information reliably from pictures and documents and recognize each character or word in the text. It does not require any special hardware and works on a COTS PDA. Once foreign text detected and characters and words are extracted, they are



System:	11.4s		
Detection:	1.6s		
Recognition:	8.4	1164 char	12 point
Post OCR	0.5	179 word	18 pixel
Translation	0.4s		1435
Translation Table	3.1MB		Spanish
Load Translation	0.8ms		

Original Text

Mundialización y gestión de los asuntos públicos
Los beneficios de la mundialización son evidentes: un crecimiento más rápido, niveles de vida más elevados, nuevas oportunidades. Sin embargo, empieza a haber una reacción negativa porque esos beneficios se distribuyen de manera muy desigual y porque el mercado mundial no se apoya

Recognition:

MUNDIALIZACIóN GESTIóN LOS ASUNTOS PúBLICOS LOS BENEFICIOS CONTRA (MANERA MUNDIAL MUNDO NUNCA) NACIóN SON EVIDENTES CRECIMIENTO LAS (TASA) RáPIDA (RáPIDO) NIVELES VIDA ARMAS (ESAS LAS UNAS) ELEVADOS NUEVAS OPORTUNIDADES SIN EMBARGO EMPIEZA HABER UNA REACCIóN NEGATIVA PORQUE ESOS BENEFICIOS DISTRIBUYEN MANERA MUY DESIGUAL PORQUE MERCADO MUNDIAL APOYA CON (LOS NOS)

Translation:

GLOBALISATION MANAGEMENT THE AFFAIRS PUBLIC THE BENEFITS GLOBALISATION ARE GROWTH THE SUPPLY MORE) QUICKLY) LEVELS SHIFT THE SUPPLY MORE) HIGH NEW OPTION BECAUSE) CITIES STAGES GASES SERIOUS) HOWEVER HOWEVER BEGINS HAVE A REACTION NEGATIVE BECAUSE THESE BENEFITS REPRESENTED WAY VERY UNEQUAL BECAUSE MARKET WORLD SUPPORTS STILL RULES BASED OBJECTIVES CHARTER THIS LACK WAY) SHARED THE FOUNDING ESTABLISHED SYSTEM OPEN COOPERATION TO WORLD INTERNATIONAL

Figure 9: Example of results obtained with Polar Rain's Kenai-II Mobile Spanish to English Translation system. The execution times are in second and obtained with a PDA which has 64MB main memory and 624Mhz xScale processor

SOURCE:

في اليوم الفوجيل (آل الجيش)،
الاسرائيلي هدم ليل الاربعاء الخميس عشرة منازل على الاقل خلال
عملية توغل في مخيم رفح للاجئين الفلسطينيين جنوب قطاع غزة .

Kenai-P:

gaza [2-1] (f-15 was)the [AfAd] source ismailia official al-youm thursday the army israeli destruction overcome Wednesday Thursday ten houses on [AlAql] during process forgotten in refugee rafah refugees south gaza strip

SYSTRAN:

Gaza 2-1 ([aaf] in) - source securities of drawings reported today Israeli Thursday that the army of [hdm] of night Wednesday Thursday ten houses on little during operation penetration in camp Palestinian Rafah for refugees of south Gaza Strip.

HUMAN:

Gaza 02/01 (AFP) An official security source reported today that on Wednesday night the Israeli army destroyed at least ten houses during an incursion into Rafeh Palestinian refugee camp in the Gaza Strip.

Figure 10: A comparison of Arabic to English translation with our system and Systran and human translator

recognized using Polar Rain's mobile OCR engine. Polar Rain's Mobile OCR engine is capable of recognizing machine printed Latin, Arabic and Cyrillic documents and road/information signs. It is capable of generating acceptable and usable results. We have achieved more than 98% accuracy for both scripted Arabic and Latin OCR engines. Once the text is recognized, it is translated to English. Polar Rain has been developing custom domain-specific and generic mobile translation engines which is capable of translating Spanish, Arabic, Serbian to English.

References

[1] A. Jain and B. Yu Automatic Text Location in Images and Video Frames in Proc. IEEE Pattern Recognition, Vol 31, 1998

[2] H. Li, and D. Doermann. A Video Text Detection System Based on Automated Training. ICPR, pages 223-226, 2000

[3] E. Haritaoglu, I. Haritaoglu Real Time Scene Sign/Text Detection System For Location Awareness And Navigation International Conference on Image Processing,2003.

[4] I. Haritaoglu Link from Real World to Digital Information Space International conference on Ambiguous Computing, October, 2001.

[5] I. Haritaoglu, Scene Text Extraction and Translation for Handheld Devices International conference on Computer Vision and Pattern Recognition (CVPR) December 2001.

[6] J. Ohya, A. Shio, and S. Akamatsu, Recognizing Characters in Scene Images IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol 16 Feb, 1994

[7] J.C. Shim, C. Dorai, R. Bolle, Automatic Text Extraction from Video for Content-Based Annotation and Retrieval In. Proc. Of ICPR, pp. 1998.

[8] J. Spohrer, Information In Places, IBM System Journal, Vol 38, No. 4 - Pervasive computing

[9] Anne Eisenberg, Point-Shoot-Translate into English: An Article on Newyork Times (March,14,2002)

[10] Michael J.Miller, Don't leave home without your hand-held translator: An Article on PC Magazine (September,9 ,2002)

Processing of Arabic Documents

A System for Discriminating Handwriting from Machine Print on Noisy Arabic Documents

John C. Femiani[1] Mariano Phielipp[2] Anshuman Razdan[3]

Abstract

A system for identifying handwritten and machine printed text from grayscale images of real-world Arabic documents is presented. The focus is on Arabic language documents because the problem is more difficult since Arabic typeset is similar to Arabic handwriting. Arabic typeset letters are connected and have similar shapes to handwritten words. Once the two types of text have been segmented it is anticipated that each can be recognized with higher accuracy using specialized treatment. We achieve positive results in segmentation by augmenting a set of attributes geared towards identification of machine printed text on binary documents (2D attributes) with attributes motivated by a 3D stroke-based interpretation of the grayscale information on documents.

1 Introduction

Our goal in this paper is to describe attributes used to distinguish typeset or machine-printed Arabic text from handwritten Arabic text in low-quality document images acquired at a resolution of 150 dots-per-inch (dpi). The authors hope that this research will ultimately be the first part of a system that provides the capability to collect, organize, and search for text in data from Arabic documents, allowing fewer language experts to accomplish more. The need to discriminate between typeset and handwritten Arabic comes from the nature of handwritten annotations: the fact that handwriting is present at all in a document can be a clue to its nature, as handwriting often contains corrections to the document or indicates the importance of the document. In spite of the fact that some tools for recognizing typeset Arabic can sometimes recognize handwriting as well, the accuracy is much worse. Therefore, special treatment may be given to the handwritten portion of a document by the use of more powerful software tools or human transcription.

The particular task motivating this research was the processing of a large dataset of documents collected en-masse and containing sensitive information. We conceived of a system whereby presence of handwriting would affect the handling of a document: All documents would be run through an Optical Character Recognition (OCR) tool to be catalogued in a keyword-searchable database. The documents could then be inspected and corrected manually based on the percentage of the document that contained handwriting (in descending order).

2 Data

The project was motivated by the influx of Arabic language documents following 9/11. Millions of documents had been acquired in digital form and the intent was to extract meaningful intelligence from the scanned documents. We received data in the form of 430 almost-randomly chosen degraded 150 ppi 8-bit gray-scale Arabic language document images that had been extracted from compressed PDF files and thus suffered compression artifacts. A significant portion of the document images was occupied by a dark region caused by the scanner bed being left open as pages were scanned. The document's page on average occupied around a quarter to a third of the entire document image, but some images contained several pages that had been placed on the scanner bed at once. The document's themselves exhibited a variety of font sizes, including images that contained only one large word as well as images with text that was only a few pixels high and thus undecipherable. Due to the classified nature of the document images, we are prohibited from publishing documents from our testing and training dataset. However, the documents we use are almost identical in nature to those available at the Iraq Research and Documentation Project (IRDP) [1] sponsored by Harvard University's Center for Middle

[1] Graduate Research Assistant, PRISM, Arizona State University, John.Femiani@asu.edu.
[2] Graduate Research Assistant, PRISM, Arizona State University, Mariano.Phielipp@asu.edu.
[3] Research Professor and Director, Decision Theatre, Arizona State University, razdan@asu.edu

Eastern Studies, with the exception that the IRDP documents have been binarized whereas we work with grayscale images. These documents were seized from Iraq and they are not classified. The US Senate has released digital copies of over 2.4 million documents to Harvard, who has made 42,000 publicly accessible.

A key issue in processing these documents is the presence of artifacts on the page that are not printed text, images, logos, or handwriting. This includes roughly uniform perturbations of the images intensity due to inaccuracies in the scanner as well as loss in image fidelity due to an unknown lossy compression method (probably JPEG) that was applied to all of the documents in our training and testing dataset. The documents had little or no skew caused by projection since they were all scanned on a flatbed scanner, but many of them had streaks or large tonal variation in certain regions presumably due to issues with the scanner or prior damage done to the documents themselves, making accurate segmentation of text a difficult task. Speckles on the document are an issue since Arabic text is heavily accented compared to English so the dots may be important in deciphering the text, but by their nature there are so many dots in an image that they would dominate the feature-collection phase of our classifier. We resolve that issue by flagging small components and we compute a reduced feature set on only those dots which are nearby larger components. Other artifacts included tears through the pages, stains, stamps placed over the text, patterned background images, and multiple pages at arbitrary orientations in the images. When we mention noise in this paper we are referring collectively to the set of issues mentioned above.

3 Related Work

Much work has been done on separating typeset characters or words on a page as this is a first-step for OCR [2, 3]. Accurate results have been reported for English typeset documents using 2D (binary) methods, even in the presence of a significant amount of noise [4]. In their work, document elements are split into three categories: Handwriting, Print, or Noise. Three trained Fisher linear classifiers are used and their results are combined based on their classification confidences. The classification accuracy is improved by combining the Fisher classifiers with a Markov Random Field (MRF) over the word-blocks where each word is connected to the nearest word to its left and right (for print), or the four nearest neighbors in any direction (for noise). The probability of a word being print versus noise based on context is systematically estimated based on training samples, and the most likely solution is found by the discrete local minimization method of Highest Confidence First (HCF). The authors demonstrate positive results on English and Chinese language documents, however positive results for English language documents do not always translate to good results in Arabic since Arabic typeset text and handwriting are both cursive, letters are often (but not always) connected at the baseline, and diacritical marks are difficult to distinguish from noise but are important to the meaning of Arabic writing [5]. In future work we plan to do a direct comparison of our method with the method of [4] on Arabic documents.

Stroke based models have been used to represent handwritten text with some success since they reflect the method used to produce the image [6]. A distinction is made between a stroke, trace, and a crossing such that a stroke is the concatenation of several traces to form the complete path of a pen from when it was placed on a page (called a "pen down" event) until it was removed from the page (a "pen-up" event). A trace is a portion of a stroke that does not cross itself or any other strokes, and a crossing is a region where traces meet, such as at the center of the letter "X". Traces and crossings are identified either by skeletonizing a component or by analyzing its contours [6]. The formation of strokes from the traces allows for recovery of the drawing order of handwriting, but strokes are difficult to recover efficiently or robustly except for certain simple configurations [6-11] .

The additional information in a grayscale image prior to binarization was modeled as a 3D surface in [12]. The surface was segmented into regions based on its gradient and Hessian, and the regions where used to segment strokes even in the case where one stroke would usually be lost due to blending with another. Cues from the intensity-surface of high resolution images were used for detecting stroke endpoints and stroke order in [7] which analyzes the evidence left by a ballpoint pen using forensic-style techniques to recover temporal order of strokes.

4 Our Method

The primitive element we classify is the four-way connected component. We are not concerned with breaking a component into individual letters or identifying entire words which may encompass multiple components since our goal is not to actually provide OCR. We model components using a stroke-based model that is similar to the models of [6] [10] [13] which we call a *stroke-graph* as illustrated in Figure 1. This model is an appropriate model for handwriting on a page, but one might point out that printed text was not produced by a pen-like instrument. Of course, we do not know yet which components on the page are printed or handwritten, and so we gather attributes from both a stroke-model and a raster-based model of the text. The nature of Arabic typeset is such that it appears to mimic the stroke-like nature of handwriting, unlike Latin characters such as E, K, H, F, etc.

We deal directly with the grayscale intensities as well

as with a binary mask indicating foreground versus background pixels. In this paper, however, we shall sometimes refer to "darkness" rather than intensity. When darkness is mentioned we refer to the complement of intensity. We treat the complement of the intensity of each pixel as its depth (z) measured orthogonal to the image plane, and thus we associate a set of 3D points (x, y, z) with traces in our stroke-graph.

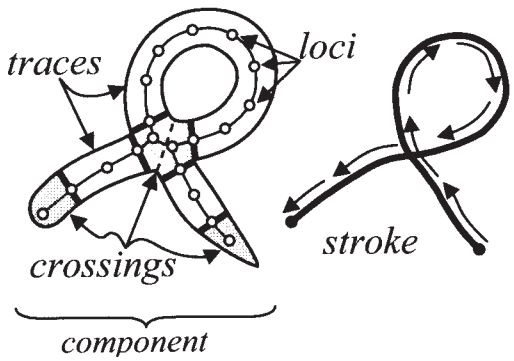


Figure 1. A single component of a stroke-graph is shown (left) with the *loci*, *crossings*, and *traces* indicated by solid arrows. A possible stroke interpretation is shown (right) with the drawing order indicated by solid arrows.

In our model, traces are represented by a chain of points we call *loci* roughly along their midline. Associated with each locus is the set of points from the image foreground that are closest to that locus. The traces are initially obtained by thinning iteratively the component [14] and then in order to compensate for noise and compression artifacts some loci are removed and others are translated according to the method explained in the paragraph accompanying Figure 1. In order to reduce the running time, we limit the number of thinning passes to the expected width of the text. Items which remain are not linear and can be removed from further consideration. Once traces are identified, the tangent to the trace at each locus is estimated by normalizing the mean of up to 5 secants centered at the tangent. Since the secants tend to get larger in magnitude as the points grow further apart, this method gives more weight to measurements taken further way from the locus, compensating for the fact that the actual positions of loci are corrupted by discretization error and error resulting from compressing the document. These tangents appear correct on visual inspection as shown in Figure 2.

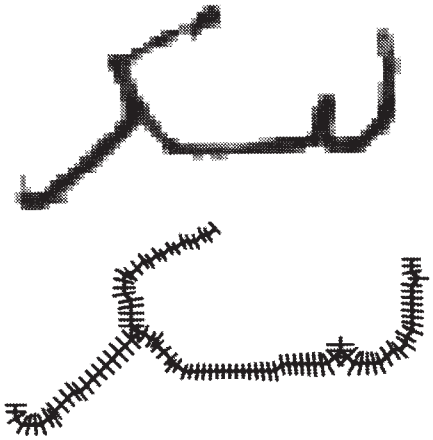


Figure 2 Top: A component with the background pixels clamped to white. Bottom: The medial axis is shown with cross-sections at each locus orthogonal to its approximated tangent.

The initial positions of the loci are not considered ideal since the midline of the trace may not be representative of the pen-path. The darkest point can not be used either since compression, noise, and the physics of writing may cause multiple darkness maxima. Our solution is to use the maximal point in a Gaussian approximation to the darkness cross-section. Each pixel associated with the locus is projected along the tangent direction. The resulting 1D function is modeled as a Gaussian, and the maximal point on the Gaussian is identified. Each locus is shifted orthogonal to its tangent so that the locus lies at the maximum of the Gaussian. The value of the Gaussian at its maximum is used as the locus's z-coordinate.

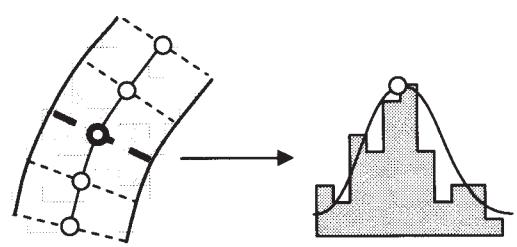


Figure 3 The pixels associated a locus (left, thick line) are project in the direction of the tangent so that their intensities form a 1D array of intensities (right, gray) that can be approximated by a Gaussian (right, solid line).

The region where multiple traces meet in a component is called a crossing, and it is initially represented by a connected set of points on the components medial axis with a valance other than two. Portions of the medial axis near the initial crossing region are removed from the neighboring traces and added to the crossing.

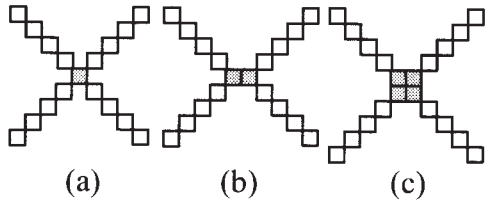


Figure 4 The ideal situation would be as in (a) where one pixel remains at the crossing (gray). The medial axis may contain clusters of points where traces meet as shown in (b) and (c).

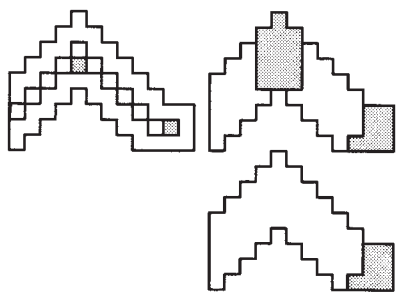


Figure 5. A crossing region (gray) is removed since it has a valance of two. The endpoint is also marked as a crossing since it has a valance of one rather than two.

Once crossings have been identified many short-traces will have entirely disappeared. If this is the case then the crossings at either end are either merged into a single larger crossing. If a merged crossing would have had a valence of two it is removed entirely and the connecting traces are joined as illustrated in Figure 5. The resulting network of traces and crossing is termed a *stroke-graph* and it is the basic unit from which many of our 3D Stroke-based attributes will be derived.

Our main concern has been determining a set of attributes based on these 3D stroke-graphs that distinguish typeset from handwritten components. We are looking for real-valued numbers that capture the difference between handwritten and printed text only. These attributes come from the binary raster information, the grayscale raster information, and from our 3D stroke based model.

4.1 2D Attributes

We began by using some attributes that were collected directly from the binary classification of pixels in the raster image into foreground and background regions. These attributes were chosen based on their reported effectiveness [4, 15] and their ease of implementation. Our goal was to allow our method to build off of methods that appear successful on binarized images. This way our tool would be able to cope with binary images or images so noisy that a stroke-model becomes meaningless.

4.2 3D-Stroke Attributes

Attributes are accumulated for an entire component based on the mean, variance, skewness, kurtosis, Gini coefficients [16], and the mean squared sample differences of attributes collected at the trace-level. Trace attributes are in turn based on attributes measured at each locus. We begin here by listing the locus-level attributes, and then proceed up to trace and component level attributes.

The 2D position of a locus refers to is x and y coordinates on the image-plane. The mean of these positions is not kept as we do not want the components position on the page to influence the classification: however the higher moments of position are used. The stroke model we use allows for the computation of a z-coordinate at each point along the trace's path. We illustrate our method for choosing z in Figure 3. All of the previously mentioned statistics of the set of z-coordinates are used, including the mean. The radius of a locus is the minimal distance from the locus to either contour – roughly half the trace's width. Two radii are kept for each locus: they are the distances from the nearest contour to the left and right of the locus as we traverse the contour[4]. The full set of statistics is used on the set radii. The error measures the discrepancy between the Gaussian approximation and the actual cross section and measured as the mean squared error taken from Figure 3. We only accumulate the mean of this quantity. The tangent angle of a locus is measured up from the detected horizontal axis of the page. Tangent angles are not accumulated: rather their respective tangent vectors are accumulated and converted to angles afterwards. The mean tangent is excluded from the set of attributes used to classify a component, but the mean difference between angles is used. Finally the number of loci in a trace, and the number of traces in a component are used as attributes as well.

One final source of attributes was added as a simple way of incorporating some of the components context within the document. The statistics kept for each component are also accumulated over a 1.5-inch neighborhood of each component and included as additional features. We achieved our best result by training two classifiers using the same algorithm (jRip) [15], but we derived additional features for a second classifier by using the results of the first classifier to determine if a 1.5 inch neighborhood is mostly hand or mostly print. We refer to the classifier that is run first

[4] The left and right are ill-defined since we do not know the direction in which the contour was traversed. Our implementation currently always begins in the leftmost column of the topmost row in which the component occurs.

as classifier 1, and the classifier based on its result as classifier 2.

4.3 Classification

We use the jRip propositional rule learner [17] based on the Repeated Incremental Pruning to Produce Error Reduction (RIPPER) algorithm first proposed by Cohen [18] in order to generate a classifier based on our attribute set and training data. We chose to use jRip over other methods because it generates a set of rules that are often simple enough for a human to understand. Additionally, it produced classified as many elements correctly as some of the most popular methods such support vector machines and neural networks (using WEKA). Another motivation behind using this classifier is that while it takes some time to train, it can classify components very quickly. Rather than explicitly model context as a MRF in the document, we collect all neighbors whose bounding boxes are within a predefined distance of each-other of 1 inch or 150 pixels. We trained two classifiers, one on features computed at each component, and the other with a feature set augmented by fraction of each component's neighbors classified as hand or print by the first classifier.

In order to generate training and testing data for our experiment, we generated an interactive tool for manual entry of ground-truth for components in the documents. We recorded each images filename and the coordinates of the darkest pixel of each component in the image along with information on the component as to whether it appeared to be handwriting, printed, image, noise (of various sorts), an embedded image, border, stamp, or a merged component consisting of both printed and handwritten parts. The tool allowed the user to select from a list of 15 categories and label components by either clicking on them, or by drawing a region around them. In this manner 47,287 components were selected of which were 20,491 printed, 20,567 were handwritten or contained components that were hand and print merged together due to overlap[5], and the remaining were items we did not wish to include in either category (such as holes, noise, borders, or logos).

Our method was implemented in C++ and compiled with all speed-optimization flags set. It took an average 4.3 seconds per document on a Pentium Xeon 3.2 GHz computer with 2 Gb of RAM. Of these, on average 2.72 seconds were spent on forming the initial image segmentation, growing connected components, finding skeletons and creating the stroke-graph. The remaining 1.31 seconds were spent collecting features and evaluating the 2 sets of decision rules generated by jRip. Our results are presented in **Table 1** using a confusion matrix. Each column corresponds to the training label of a component, each row corresponds to the class predicted by our classifier, and each entry in the matrix shows the number of instances trained according to the column and predicted according to the row. We were able to identify 94.56% of the components correctly[6], and we identified 94.31% of all handwritten items in our test set and 94.8% of all printed items.

Table 1 Results of our classification of the entire feature set, presented as a Confusion Matrix.

Predicted Typeset	Predicted Handwritten	
19325	1166	True Typeset
1069	19498	True Handwritten

5 Conclusions and Future Work

We have presented our system for discriminating between Arabic handwriting and typeset text. We have achieve positive results on our training and testing corpus, and conclude that incorporating 3D features based on a stroke-model of the text derived directly from grayscale images can improve our ability to separate printed from handwritten text. In future work we would like to compare our method carefully with the method of [4] and explore other ways of dealing with document layout and noise in the types of documents we see. Since handwritten or mixed components appear to have been successfully identified, we would like to enable queries for words through handwritten portions of text using a shape-based word-spotting algorithm. Also we would like to extend out ability to capture strokes and resolve crossings. Our current stroke based model solves simple examples of crossovers, but suffers from some extreme cases of ambiguity cause by sub-pixels attributes in text or image degradation due to compression.

Acknowledgements

The authors would like to thank the Partnership for Research in Spatial modeling (PRISM) and other sponsors, as well as the reviewers for their helpful comments.

[5] For the purposes of this project, merged hand & print are lumped together with handwritten components since are difficult to OCR and thus require the same extra attention as handwritten components. Also overlap tends occur between handwritten words rather than hand-over-print in our test data.

[6] The tool "WEKA" could not process all 40,000 entries and so it was trained with roughly 50% of the samples and reported accuracy greater than 96% on that subset.

References:

[1] M. Kanan, R. Rend, H. Mneimneh and R. Rabil "The Iraq Research and Documentation Project (IRDP)," http://www.fas.harvard.edu/~irdp/, September 15, 2005.

[2] Zheng, Liu and Ding "Single-character type identification," *Document Recognition and Retrieval IX* vol. 4670, no. 1, pp. 49-56, 2001.

[3] Pal and Chaudhuri "Automatic separation of machine-printed and hand-written text lines," *ICDAR '99.* pp. 645-648, 1999.

[4] Y. Zheng, H. Li and D. Doermann "Machine Printed Text and Handwriting Identification in Noisy Document Images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* vol. 26, no. 3, pp. 337-353, 2004.

[5] A. Amin "Off-Line Arabic Character Recognition: A Survey," *ICDAR97* pp. 1997.

[6] R. Plamondon and C. M. Privitera "The Segmentation of Cursive Handwriting: An Approach Based on Off-Line Recovery of the Motor-Temporal Information," *IEEE Transactions on Image Processing* vol. 8, no. 1, pp. 80-91, 1999.

[7] Doermann and Rosenfeld "Recovery of temporal information from static images of handwriting," *CVPR '92.* pp. 162-168, 1992.

[8] I.S. Abuhaiba and P. Ahmed "Restoration of Temporal Information in Off-Line Arabic Handwriting," *PR* vol. 26, no. 7, pp. 1009 1993.

[9] Bunke, Ammann, Kaufmann, Ha, Schenkel, Seiler and Eggimann "Recovery of temporal information of cursively handwritten words for on-line recognition," *ICDAR'97* vol. 2, pp. 931-935, 1997.

[10] Hew and Alder "Strokes from pen-opposed extended edges," *ICONIP '99* vol. 3, pp. 1155-1160, 1999.

[11] Kato and Yasuhara "Recovery of drawing order from single-stroke handwriting images," *PAMI* vol. 22, no. 9, pp. 938-949, 2000.

[12] L. Wang and T. Pavlidis "Direct Gray-Scale Extraction of Features for Character Recognition," *PAMI* vol. 15, no. 10, pp. 1053-1067, 1993.

[13] Nishida "An approach to integration of off-line and on-line recognition of handwriting," *Pattern Recognition Letters* vol. 16, no. 11, pp. 1213-1219, 1995.

[14] Lam, Lee and Suen "Thinning methodologies-a comprehensive survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* vol. 14, no. 9, pp. 869-885, 1992.

[15] Y. Zheng, H. Li and Doermann "Text identification in noisy document images using Markov random model," *ICDAR'03* pp. 599-603, 2003.

[16] G.J. Glasser "Variance Formulas for the Mean Difference and Coefficient of Concentration," *Journal of the American Statistical Association* vol. 57, no. 299, pp. 648-654, 1962.

[17] I.H. Witten and E. Frank "Data Mining: Practical machine learning tools with Java implementations," 2000.

[18] W.W. Cohen "Fast Effective Rule Induction," *Proc. of the 12th International Conference on Machine Learning* pp. 115-123, 1995.

NLP-Enhanced Exploitation of Degraded Arabic Texts

Michael Shalev
Encyclopædia Britannica,
Natural Language Technologies Division
311 S. Michigan Avenue,
Chicago, IL 60604
michaels@eb.com

Abstract

Arabic poses significant challenges to automatic exploitation due its highly inflectional nature (stems can have hundreds or even thousands of inflected forms), and its consonantal orthography (most vowels are omitted), which causes high ambiguity rates. These challenges are even more pronounced when dealing with texts that are degraded.

Arabic OCR character recognition accuracy is adversely affected by Arabic's cursive script, usage of diacritic dots, and the relatively large number of letter forms (each letter may have several allographs depending on the context of adjacent letters).

We applied a context-sensitive morphological analysis system to degraded Arabic texts received in lattice format to examine the potential contribution of linguistic analysis to text exploitation of degraded Arabic documents. The enhancement was measured at various levels of character recognition accuracy (corresponding to accuracy levels typical of OCR output). The resulting levels of word recognition and entity detection were compared to the baseline levels of the generated degraded texts.

The results show improvement in the number of recognized words and entities in the linguistically processed texts compared to the raw degraded documents at all levels of input letter accuracy.

1 Introduction

1.1 Arabic Morphology

Arabic[1] morphology is commonly viewed as being based on underlying consonantal roots (*jiðr*, جذر) that undergo vowel changes, and in some cases consonantal insertions and deletions, to create inflected and derived forms [5]. For example, the root /ktb/كتب combined with the vocalic pattern **CaCaCa**[2] (where '**C**' represents a consonant and '**a**' represents a vowel) derives the verb *kataba*[3] كَتَبَ 'to write'. This derivation may be further inflected into forms indicating semantic features, such as number, gender, tense etc.: *katab-tu* كَتَبْتُ 'I wrote', *katab-ta* كَتَبْتَ 'you (sing. masc.) wrote', *katab-ti* كَتَبْتِ 'you (sing. fem.) wrote', *?a-ktubu* أَكْتُبُ 'I write/will write' etc.

In addition, the definite article, as well as several prepositions and conjunctions are considered prefixes and not autonomous words; additionally, pronominal objects and possessive pronouns are suffixed to verbs and nouns, respectively.

The structure of Arabic morphology results in each word having a very high number of inflections, which may radically change the basic form of the words. Accordingly, the total number of Arabic inflected forms is estimated to be 70 million [1], [2].

1.2 Arabic Orthography

Arabic orthography is primarily consonantal - most vowels are not written. There is a supplementary system of vowel and gemination diacritics (written above, under and beside the text), – but these are omitted in most written texts[4]. As a result, the rate of form ambiguity, i.e. ratio of morphological analyses per

[1] Arabic here refers to Modern Standard Arabic (MSA).

[2] This pattern is known in Arabic grammar as وزن فعَلَ.

[3] Latin transcription of Arabic examples and terms follows English spelling conventions. The character: "?" denotes a glottal stop; and "ð" denotes the sound "th" in "this".

[4] This supplementary marking system is used in this document for some Arabic examples, to emphasize the difference between Arabic words.

given string, is considerably higher than in most languages.

2 Arabic OCR Challenges

In addition to morphological and orthographical features that complicate Arabic text exploitation in general, (e.g. search, translation and entity extraction), Arabic script and letter shapes (printed as well as handwritten) present additional challenges that are specific to OCR, including:

- Cursive Script: most characters are connected and their shape is context (position) sensitive: a letter may have different shapes when appearing in isolation, in word-initial, -median, or -final position; letters may have as many as 4 shapes.
- Word elongations (tatwil تطويل) and ligatures (letter combinations, such as lam-alif "لا") expand the number of possibilities.
- Diacritic marks serve to optionally mark vowels as well as to distinguish between several of the Arabic letters, where they are obligatory. Dust, dirt, and specks may be confused with diacritic marks, further complicating letter recognition.
- Punctuation marks are often omitted and cursive script is sometimes broken within a word, making the distinction between word boundaries and inner word spaces more difficult, and complicating segmentation.

Typically, OCR software generates several character candidates per image segment, each designated by a confidence score; such OCR internal structures are referred to as lattices.

While statistical and language-independent OCR strategies may suffice to provide acceptable accuracy levels for some languages (e.g. English) in which the above considerations are absent, or present to a lesser degree, when dealing with Arabic texts, whether printed, or even more significantly, handwritten, the accuracy levels offered by such methods decrease, and often may not yield sufficient accuracy to be of practical value.

The purpose of this study is to examine the contribution of applying linguistic post-processing to enhance the exploitation of Arabic OCR output. Exploitation enhancement is achieved by identifying correct and appropriate words within OCR output lattices, thereby making the texts more searchable, and therefore more suitable for automatic categorization and routing processes. Enabling such a linguistic-based accuracy enhancement requires a robust NLP system that can identify and score all 70 million inflected forms that appear in Arabic, and be able to propose the most appropriate word represented by the OCR word lattice.

While fully accurate OCR of Arabic handwriting will remain difficult to achieve, it should be noted that:

- Partial handwriting recognition, even at low accuracy rates, is potentially of great value.
- Partial recognition may allow identifying important keywords in documents.
- Improvement in word recognition over character lattices at various accuracy levels can contribute significantly to automatic processes.

3 Natural Language Processing

In pursuit to enhance and exploit degraded Arabic documents, several approaches were examined, mainly character and word n-grams [4], [6], light stemming (algorithmic prefix and suffix removal) [3] and closed-end partial lexicons.

The current study presents an approach based on applying a robust stemming/morphological analysis system to exploit and enhance degraded Arabic documents.

3.1 Morphological Analysis

The Britannica Morphological Analyzer (BMA) for Arabic was used in this study. BMA identifies all Arabic word forms, breaking down inflected forms into their base form (stem) and inflectional morphemes. BMA applies scoring based on frequency measures of each stem, as defined in the BMA lexicon, a frequency measure of the morphological variant, and a measure calculated by context-sensitive analysis, which checks for validity of syntactic phrase structure.

When BMA is applied as a stemmer in search engine indexing systems, the score is used to ensure that the correct stem is indexed in cases of ambiguity, so that only inflections relevant to the query (i.e. the stem appearing in the query) are returned, and not unrelated forms that happen to share the same spelling with one of the inflected forms of the query terms.

When applied to degraded text, BMA's scores are used differently: rather than using the scores to choose between *alternative morphological analyses* of a given string, the scoring system is used to choose between different *strings*, representing different word forms proposed for a given segment of the degraded text.

3.2 Named Entity Recognition (NER)

Named Entities (NEs) refer to proper names (e.g. people, places, organizations, expressions of time, etc.), and hold a special significance in text exploitation due to their high content value. NER attempts to identify and classify Named Entities by referring to NE

lexicons. Cross-language NER translates entities identified in Arabic into English.

Performing NER by linguistically enhancing degraded input intends to provide more valuable results from the perspective of text exploitation. EntX, Britannica's cross-language NER system, was used in this study.

4 Data

40 typical documents from Arabic media web sites (Al-Hayyat, Elaph, Alquds, etc.) were selected. The length of these documents ranges from 150 to 200 words, yielding a sample corpus of over 7,000 words.

The documents then underwent automatic degradation to four levels of character accuracy typical of handwritten OCR output (see section 5.1 below), yielding 160 sets of output lattices (four for each original document), and a total of over 28,000 words.

5 Method

5.1 Automatic Degradation

To simulate OCR results at different levels of accuracy, an automatic text degradation program was developed, modeled on OCR lattices of a sample corpus of handwritten documents. The program takes as input an Arabic text document and the desired level of degradation (defined as the letter accuracy[5] level), and returns sets of word lattices for the identified words/strings in each degraded document.

The degradation process has several parameters:

- ***Character accuracy***: the probability of the character being correctly identified by the OCR process (i.e. being shown as the topmost character candidate per position)
- ***Lattice depth***: the number of character candidates for a given character (0-5)
- ***Lattice quality***: the probability of the correct character appearing in the generated lattice, providing it wasn't selected as the first candidate
- ***Segmentation***: lattices are either split, or separate lattice joined to model incorrect identification of word boundaries. The segmentation quality reflects the percentage of words correctly segmented)

The lattices are populated with random characters and random relational confidence scores (aside from the topmost character as defined by letter accuracy, having the top score, 0).

Note that the fact that lattices are populated randomly makes the output somewhat more difficult to exploit, as compared to OCR output, where scores of correct ground truth character candidates in the lattice are likely to be better (lower) than those of incorrect characters.

Each document is degraded to four levels of character accuracy: 30%, 45%, 60%, and 75%, yielding 4 documents for each original text, in addition to the original, which serves as the ground truth for the degraded documents.

Given the character accuracy, the lattice quality and segmentation quality were adjusted to reflect typical OCR results, in a manner illustrated in table 1.

Our definition of character accuracy follows the one proposed by Rice, Kanai and Nartker [7], based on the Edit Distance algorithm (known as the Levenshtein algorithm).

The following table describes the character accuracy, lattice quality and segmentation quality of the degraded documents.

Table 1: Lattice quality and segmentation quality per Letter accuracy.

Character accuracy	Lattice quality	Segmentation quality
30%	55%	25%
45%	65%	65%
60%	75%	80%
75%	90%	90%

The output (i.e. 'identified' words) of each degraded document was compared to the ground truth to determine the word accuracy achieved in the simulated OCR output per document (for each letter accuracy level), which serves as the baseline.

5.2 NLP Processing

5.2.1 Word-Internal Lattice Enhancement

Each sequence of characters in the lattice is compared with a set of patterns reflecting typical noun and verb prefixes, suffixes, and internal patterns. In cases where a pattern is partially found, this process may introduce an additional character candidate to the lattice. Consider the following table:

[5] The probability of the correct letter appearing in the OCR input in this position (compared to the ground truth).

Table 2: Example Lattice[6] of البحرية ("the marine"). Correct letters appear in red. Optical scores appear under letter candidates - lower scores are better.

String Option 1	Letter	ا	ا	ف	ب	م	ي	ة
	Score	0	0	0	31	0	0	0
String Option 2	Letter	إ	ى	ب	ه	ر	أ	ة
	Score	85	20	35	58	10	51	35
String Option 3	Letter		ص	ط	م	ث	و	ر
	Score		57	4	70	15	32	70
String Option 4	Letter		ء	و	ح	ة	س	ى
	Score		216	76	75	138	183	78
String Option 5	Letter		ل	ه		س	س	ف
	Score			180		20	200	21
Output		ا	ل	ب	ح	ر	ي	ة

In string option 1 above (the first row) there is a word-initial sequence of two alifs ("اا"). The word-internal processing introduces a candidate lam ("ل") in the second position as the sequence "ال" is the Arabic determiner, which has a very high likelihood of appearing in word-initial position. This expands the set of strings that may be generated from the lattice. Following this process, the set of all possible strings or character combinations resulting from each lattice (generally ranges from hundreds to thousands) is generated. These strings are then subject to word level linguistic processing (see following section).

5.2.2 Word Level Linguistic Processing

All generated strings first undergo morphological analysis to determine which, if any, of the strings are morphologically well-formed. The stems of the set of well-formed strings are then subjected to lexical look-up to determine if they are known words. The lexicon contains statistical data regarding distribution of stems and inflected forms. If sufficient context of neighboring words exists, context-sensitive weighting is applied to the word candidates. Each step of the word level linguistic processing contributes to the linguistic score accorded to the word (see below).

5.2.3 Linguistic Scoring

A *linguistic score* is determined per string (as opposed to the *optical score* given to each *character* candidate by the OCR process), which considers the following factors: frequency and distribution probability of lexical stems and inflection patterns; and contextual weighting (if a sufficient context of neighboring words exists).

Thus the best linguistic scores are assigned to words that are recognized as belonging to widespread stems with common inflections which fit well in their syntactic context. Words that are recognized by BMA but are found to be derived from less frequent stems, and/or feature rare morphological structures or are inappropriate with their context are given higher (worse) scores. Words that are not recognized at all are discarded.

Table 3: output for three different strings considered for a certain lattice (given in phonetic transcription). Lower scores signify better confidence.

String candidate	Linguistic score	Gloss	Details
Alsban *almhtrfyn* (الشبان المحترفين)	**12**	*The young experts*	valid words, correct phrase structure - well formed agreement
Alsbak *almhtrfyn* (الشباك المحترفين)	**23**	ungrammatical *(*experts the window)*	valid words, incorrect grammatical context
Alsblk *almhtrfyn* (الشبلك المحترفين)	-	word discarded	invalid word

A composite score of the *linguistic score* of a word and an *optical score*[7] for this word is calculated and the string with the highest composite score is returned. The composite score is calculated by 75%-25% weighting in favor of the linguistic score (see further discussion below regarding weighting considerations).

6 Results

The word accuracy and named entity (NE) accuracy of each document was determined by manually comparing it with the respective ground truth. The 160 degraded documents were then subject to NLP processing (see section 5), providing 160 linguistically enhanced documents. These too were manually compared to the ground truths, and word and NE

[6] 45% character accuracy; purple cell contains a character introduced by linguistic post-processing

[7] The optical score of the string is calculated as the average optical score of the characters the string consists of.

accuracy were determined. The results appear in figure 1 below.

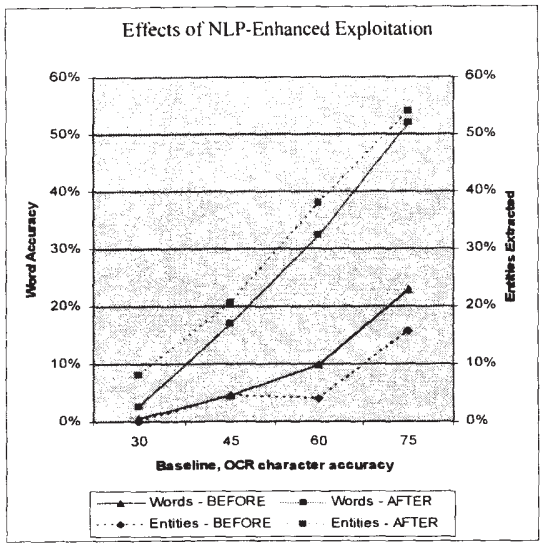


Figure 1: Effects of NLP processing of degraded documents: word and entity accuracy before and after NLP processing

The solid blue line represents the base level word accuracy level for each of the four degradation levels (30%, 45%, 60%, and 75%), while the dashed blue line represents the NE accuracy at these levels.

The solid magenta line shows NLP-enhanced word accuracy for each degradation level, and the dashed magenta line shows the NLP-enhanced NE accuracy. The following tables show the word- and NE-recognition for the four levels of character accuracy.

Table 4: Word-recognition as a function of character-accuracy

Character accuracy	Word recognition	
	Non-enhanced	NLP-enhanced
30%	0.5%	2.6%
45%	4.4%	17%
60%	9.8%	32.2%
75%	22.9%	51.9%

Table 4 above shows an improvement from 0.5% to 2.6% for highly degraded documents, to from 23% to 52% for the least degraded documents.

Table 5: Named Entity-recognition as a function of character-accuracy

Character accuracy	Named entity recognition	
	Non-enhanced	NLP-enhanced
30%	0%	8.2%
45%	4.2%	20.6%
60%	4%	37.9%
75%	15.7%	54%

Table 5 above shows an improvement from 0% to 8.2% for highly degraded documents, to from 15.7% to 54% for the least degraded documents.

7 Discussion

The results show significant improvement both in word and in named entity (NE) recognition accuracy following linguistic post-processing at all character accuracy levels.

While word accuracy is higher than NE accuracy in non-enhanced results, NE accuracy is higher than word accuracy in NLP-enhanced OCR. Average word length in Arabic is 4.63 characters, while average NE length is 6.4 characters[8]. This characteristic may help explain the different patterns observed between NLP-enhanced and non-enhanced entity accuracy versus word accuracy: since entities are by and large longer than general words, the probability of an incorrectly identified letter is greater for entities than for words; however, when applying NLP-enhancement, string length contributes positively to the probability of successfully identifying the word, as there are more cues in the input.

From a practical perspective of trying to increase the value of OCR to text exploitation, increasing recognition levels of entities, which are of high content value, is more valuable than the 'across-the-board' increase achieved in word accuracy.

NER was applied by looking at a pre-defined set of entities, a subset of the words identified by BMA. An alternative approach would be to actively search for user-defined entities, such as certain proper names, and trying to match them against given lattices; this may produce more false-positives, but may limit the documents with these specific entities that would be overlooked to a minimum.

This study used a pre-defined weighting allowing 75% to the linguistic score, and a 25% weight to the optical score. This was based on preliminary tuning, however, further work needs to be carried out to find the optimal balance between the two scores, which we

[8] These values are based on a sample corpus of documents/ articles collected from Arabic media web sites.

expect to be dependent on type of material and OCR quality.

Appendix A shows an example Arabic text, in raw degraded form and in linguistically enhanced form. Correct words and NEs are marked in both documents.

Appendix B shows an XML excerpt for a word processed by BMA, and shows the metadata that may serve other exploitation processes.

8 Acknowledgements

I would like to thank the team at Encyclopædia Britannica's Natural Language Technologies Division.

References

[1] K. R. Beesley, Finite-State Morphological Analysis and Generation of Arabic at Xerox Research: Status and Plans in 2001. In *ACL Workshop on Arabic Language Processing*. Toulouse, France, 2001.

[2] Y. Choueka, personal communication.

[3] K. Darwish and D.W. Oard, Term selection for searching printed Arabic. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. Finland, 2002.

[4] S. Harding, W. Croft and C. Weir, Probabilistic Retrieval of OCR Degraded Text Using N-Grams. In *European Conference on Digital Libraries*. Pisa, Italy, 1997.

[5] J.J. McCarthy, *Formal Problems in Semitic Phonology and Morphology*. (Garland, New York; London, 1985).

[6] S.H. Mustafa and Q.A. Al-Radaideh, Using N-grams for Arabic Text Searching. In *Journal of the American Society for Information Science and Technology*, Volume 55, Issue 11, (Wiley Periodicals, 2004).

[7] S.V. Rice, J. Kanai and T.A. Nartker. An Evaluation of OCR Accuracy. In *The Second Annual Symposium on Document Analysis and Information Retrieval*. Las Vegas, Nevada, 1993.

Appendix A – Sample Arabic Document

Figure 2 shows an example Arabic text degraded to 60% character accuracy.

قخك ظغف جنود مء بشمب البظلية اعهمريفأس ىغ ىزدثيؤ مإلبايا قسأ
الرماصؤ غصجي يلعرهق.وقئل اخجيش اشامأطأي فس ليجئ بن خفىة
جعآد غتلول لغن ظرتهلم عربتهم ثثبوغ ضاجفؤ خرلت عصى إكسذ عإغ
بلجدجنء التط تعو ثعسرا لأمطتيين مس الضاش.قألعإ الدجةن اث ظهريأ
ةادإا مظ قؤاء جضمإضينم تتع برصاخ متإيليث بنلسدينة.ويعد الهئوى ءت
ملئقيي وضال ضسبقع صحغي يوقب خلالج زلإااتستن محل قتا وليذد
ضلءبير من القتقى كدن إضرلو مةاي الغحرية وجف نفس
وذئنظقةعوقمل لراذ أأرخكيفن إن تحض الأقظتليي يلغأون الى سضع
كريات طآيقة من بلمتثفرات سي نقغك ثحآدط بوى يصكنهم تفوءر
للعبباح الأءرةة حقذلغ ذلظببباىٔ الثقيتذمتلان غماة بف مشاة ءءبأياق
اساأريزوة قف يدآزئ خءىا كوإ ثلحتئس عةتظري طي اثظثؤر قببصش
على جاخب نأطعلق وضم نز أربتزم قرك اسردادم.ومنئ اذتزى
اخأمبيكي للعإتأ عام لذ03 قحء خدى ذذاقل في المصءلى ذ7ىة جهتي
امريكعلوسقطذ مايدة افثمذظي 4ه0 روف صخش نسبهد ححطري كل5
طؤلومترا طن الراصسى بغداث.وقهش ماوت مو زلكتىينز ين
غدمسؤوةين عن روح قئنةذم أي الرمادي يةهم وجدخأ ان نسزوى
اتمقطومة يةلمدإةة ظعإدل ملمظجأخ بمدفنة الهطذجةكوقالت كءضؤطيج
ضائمي معخذلة ضي بت سح شحسعراف ان صلجكش نلاظضتكض قلدم
كفك تفاصيأ قميظب عض قلغدلات الهي وتعرض طمة سلجنود ضةءلين
ذلك طفنه ذحاحلض لحبلقثممف

Figure 2. Non-enhanced OCR output (16 correctly identified words appear in **brown**)

Figure 3 below shows the result of linguistic post-processing applied to the document; an additional 47 words were recognized, of which 12 are named entities (versus none in the original degraded document).

فخك سقف جنود من عذاب التسلية اعهمريقأس في سادسين منفصلين
فيك الرمادي غصجي يلعرهف.وقئل الجيش اشامأطأي في ليجئ بن
خفى جزأ قتلوا لدى ظرتهلء عربتهم بعبوة ضاجقؤ زرعت عصى
أكسف عرب بلجدجفء الخط تعد معسرا للمسلحين مس الضاش.قألعإ
الشجون ان ظهري سائدا من قوات جضمإضيفم تبع ارضاء ستجينين
بفلسدينة.ويعد الهدوء إو كلثاني وضال أسبوع الذي ينفع خلال زلإااتستن
محل هذا أريع ضلءبير من الفترى كدن إضرت مشاة القارية وجف
نفس وذئنفظقةعوقمل لرن أأرخكيقن إن تحض الأقظتليي يلغأون الى
سرع كريات كثير من المتفجرات المتفجرات نفغك محدد بول يمكنهم
تفجير العربات الأدقة وفرجى ذلظببباىٔ الثقيتذمتلان كمصر بل مشاة
البطريق اساأريزو قف يدآزئ ايضا أو الخميس لحضاري طي اثظثؤر
فببصر على جانب اصبعيه وضم فز عرفتهم غرب اسردادم.ومفئ فلتزر
اخأمبيكي للعراق عام لغمس قتل على الاقل في المصائر ذاى جهتي
امريكعلوسفطذ عابد اقتداءي زها جوي شخص رحبه حواري كلت
كيلومترا طن العاصفة بغداث.وفهش مالت من المارينز ين غدمسؤوةين
عن روح النظام أي الرمادي يفهم وجدوا ان مستوى المقاومة
بالمضافة يعادل الموجود بمدينة الهطذجةكوقالت كبرصدين غائمي
معاقل ضد بت ض شحسعراق ان الجيش نلاظضتكض فلجم كفل
تفاصيل فليلب عض الهجمات الهي وتعرض لم الجنود مقللين ذلك طفنه
محاولا لحبلقثممف

Figure 3. BMA-enhanced OCR Output (45 newly recognized words in blue, NEs are highlighted)

Figure 4 below shows the document returned by Britannica's EntX server, containing cross-language NER. The English entity digest appears at the head of the document, and Britannica's ETL (Embedded Translation Layer) provides a hidden layer of translation for all words/phrases that is displayed by placing the mouse (cursor) over any word/phrase (see middle of document by mouse cursor in figure 4).

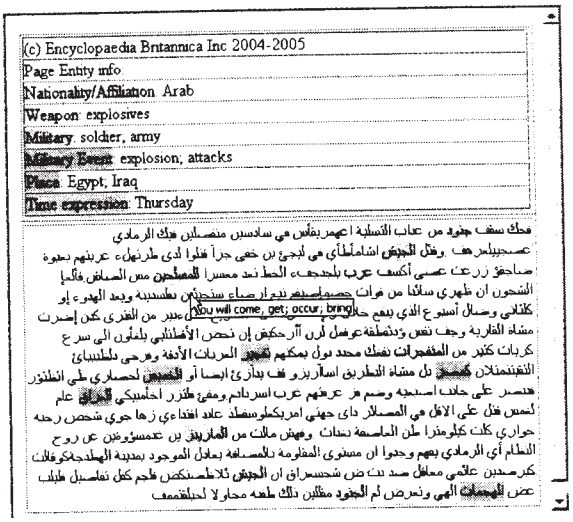


Figure 4. EntX processing of NLP-enhanced text, showing the entities highlighted by category, with their English equivalents appearing in the entity digest header.

Appendix B – BMA XML Excerpt

The following XML document shows an excerpt of the morphological and other linguistic metadata added by the Britannica Morphological Analyzer (BMA).


```
- <root>
  - <word>
    - <input>
        - <![CDATA[وكتبه]]>
      </input>
    - <result>
        <stem>كَتَب</stem>
        <transcription>kataba</transcription>
        <partOfSpeech>verb</partOfSpeech>
        <root>كتب</root>
        <conjugation>1</conjugation>
        <category />
      - <rawTranslation>
          - <![CDATA[ to write]]>
        </rawTranslation>
      - <prefix>
          <string_prefix />
        </prefix>
      - <suffix>
          <string_suffix>هُ</string_suffix>
          <person_suffix>3</person_suffix>
          <number_suffix>singular</number_suffix>
          <gender_suffix>masculine</gender_suffix>
        </suffix>
        <score>85</score>
      - <inflection>
          <vocalizedWord>وَكَتَبَهُ</vocalizedWord>
          <inflectionTranscription>wakatabahu</in-
            flectionTranscription>
          <person>3</person>
          <number>singular</number>
          <gender>masculine</gender>
          <state />
          <tense>past</tense>
          <voice />
        </inflection>
      - <morphTranslation>
          <conjTranslation>and</conjTranslation>
        - <inflectionTranslation>
            - <![CDATA[wrote]]>
          </inflectionTranslation>
          <objectTranslation>it</objectTranslation>
        </morphTranslation>
      </result>
    </word>
  </root>
```


Figure 5. BMA XML Output for the Arabic Word وكتبه ("*he wrote it*")

Initial Results in Offline Arabic Handwriting Recognition Using Large-Scale Geometric Features

Ilya Zavorin **Eugene Borovikov** **Mark Turner**
CACI International Inc
4831 Walden Lane
Lanham, MD 20706
{izavorin,eborovikov,miturner}@caci.com

Abstract

This paper presents preliminary results that we obtained during the first "proof-of-concept" stage of our on-going research on offline Arabic handwriting recognition. Our method relies on the Arabic script's large-scale features (e.g. loops, cross- and end-points) and employs a double-layered discrete Hidden Markov Model (HMM) for character and word recognition. This approach does not require manual segmentation of words into characters during training or testing. The prototype system uses manual feature extraction and yields high recognition rates on synthetically generated words.

1 Introduction

Although there have been extraordinary advances in automatic recognition of machine-printed Arabic text over the last several decades, the problem of automatic offline recognition of handwritten Arabic script remains largely unsolved. As Edelman et al. [3] point out, the two main reasons for the difficulty of this task are

- character segmentation ambiguity; and
- character shape variability

Arabic script is inherently cursive, meaning that a considerable part of a word image may consist of ligatures that connect consecutive characters. These ligatures may often be interpreted as parts of letters giving rise to multiple equally plausible segmentations. Letter shapes are position-dependent. Arabic text contains numerous diacritics, overlaps, and casheedas. Furthermore, even when written by the same person, the same characters may change both geometry and topology (see, for example, Figure 1 taken from the corpus described by Pechwitz et al. in [8]). All these characteristics, combined with other language-independent irregularities typical in handwritten text, make Arabic handwriting recognition problem extremely difficult.

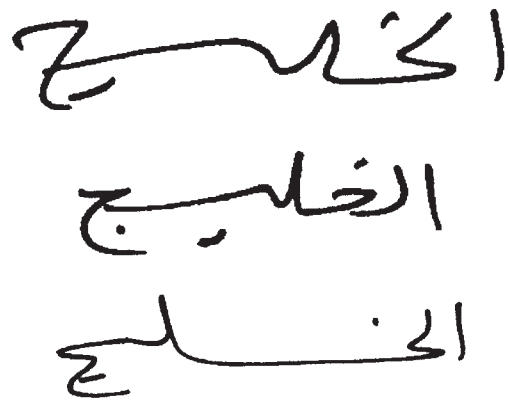


Figure 1: Same word written by the same writer

The core of a complete ICR system is its recognition unit. According to Steinherz et al. [14], one way of classifying recognition algorithms is by the size and nature of the *lexicon* involved, as well as by whether or not a *segmentation* stage is present. Lexicons may be

- small and specific containing about 100 words,
- limited, but dynamic, which may go up to 1000 words, and
- large or unlimited.

Segmentation-based methods take a sequence of primitive segments derived from a word image together with a group of possible words, and attempt to concatenate individual segments to best match candidate characters. In the segmentation-free case, no attempt is made to separate the image into pieces that relate to characters, although splitting into smaller pieces is still possible, followed by generation of a sequence of observations. In both cases, algorithms try to reproduce how a word was *written* by recognizing its components in the left-to-right (or right-to-left) fashion.

Alternatively, there is a class of *perception-oriented* methods that perform a human-like *reading* by utilizing some stable features to reliably find

characters anywhere in the image, and to use them to bootstrap a few word candidates for a final evaluation phase. Such methods have rarely been truned into complete recognition systems. Instead, they are sometimes used to enhance more traditional "directional" algorithms described above.

To date, most attention appears to have been directed toward small- and limited-lexicon methods, especially those applied to Latin scripts. Many different methods and a few complete recognition systems have been proposed in the literature. Overall, however, the field of handwritten text recognition has not yet matured [14].

There are relatively few review-style articles [10, 4] and even fewer attempts to experimentally compare different methods. One reason for this is lack of sufficiently large, general, and publicly available test datasets. Furthermore, the performance of a complete ICR system depends on many other components besides its recognition engine, e.g. preprocessing, segmentation, and post-processing modules. To our knowledge, there have been virtually no attempts to create a common framework that allows an objective comparison of different engines or complete recognition systems.

One of the most successful approaches to Arabic character and word recognition is based on the application of *Hidden Markov Models* (HMM) [5, 9, 2]. They model variations in printing and cursive writing as an underlying probabilistic structure, which cannot be observed directly [1]. When small lexicons are involved, elementary HMMs that model individual characters are combined to form individual larger *model-discriminant* HMMs for every word in the lexicon. When a word image is presented to such a system for recognition, it chooses the word interpretation whose model gives the highest probability given a sequence of observations, or symbols, generated from the given image. For larger lexicons only one *path-discriminant* HMM is build from the character sub-HMMs and recognition is performed by finding the most likely path through the model given the sequence of observations from a word image. Path-discriminant models are more general but less accurate than model-discriminant ones.

HMM-based methods are popular because they performed quite well on a similar problem of speech recognition, and because their mathematical machinery has been developed and refined quite well. At the same time, they still constitute an active area of research. One reason is that performance of an HMM depends strongly on the type of features used to generate observation sequences both for training of the models and for recognition. As pointed out in [14], the "ideal" feature set has not yet been found, and may not exist at all. Also, while most HMM-based methods represent a word image as a one-dimensional sequence of observation vectors, a more recent approach is to generalize to two-dimensional Markov Models (also called Markov Random Fields) [7]. While this approach appears to give better accuracy, it is also much more computationally expensive and so requires more sophisticated design and implementation than traditional HMM-based methods. In addition, MRF-based systems usually require much more training data than HMM-based ones.

Figure 2 shows the same Arabic word written by two different people (also taken from [8]). We can see that while the two handwriting styles are quite different, most of the large-scale features such as loops, turning points and long straight strokes are preserved. Therefore we made the decision to base feature extraction in our recognition system on such features. We believe that they are more robust than small-scale features such as local curvature and stroke thickness to inter-writer and intra-writer variations in size, orientation, and local distribution of characters as well as to poor image quality.

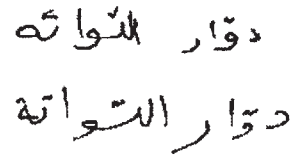


Figure 2: Same word written by different writers

Our approach, which we describe in more detail in Section 2 is based on that developed by Khorsheed in [5]. Our recognition system consists of four main components. *First*, it is based on large-scale geometric features. *Second*, it simulates the writing process by tracing connected components of a word image using a consistent set of rules. *Third*, it is trainable and uses HMMs. *Finally*, it can be adapted to use various types of high-level information at latter stages of the recognition process to resolve ambiguity.

2 Methodology

In Section 2.1, we present in some detail the original approach of Khorsheed [5]. We then describe our modifications in Section 2.2.

2.1 Original Algorithm

Khorsheed's approach is to construct a single "universal" discrete HMM that represents an entire lexicon that is being modeled. This is an example

of the path-discriminant approach described in Section 1. The universal model is a collection of states each of which represents a single Arabic character. These character states are interconnected according to character bigram statistics extracted from a given training lexicon. Each state is in itself a discrete HMM that has a left-to-right (or, perhaps more precisely, right-to-left) topology. Each hidden state in a character model heuristically represents a stage in writing this character. Therefore the number of hidden states is proportional to the character's complexity.

The recognition system works in two stages: training and recognition (see Figure 3). During training, a preprocessor converts an image of a word or a letter into a sequence of discrete observations symbols. These sequences are used to train individual character models that are then connected into a universal model. The well-known Baum-Welch HMM training algorithm [11] is used at this stage. During recognition, the same preprocessor converts a word image into an observation sequence that is then fed into the trained model. The most likely path through the model yields a sequence of character states that constutites the recognized word. A modification of the standard Viterbi algorithm [11] that computes several top paths rather than just the best one is used at this stage.

The preprocessor works in several steps. First, a skeleton of a word or letter image is computed and represented as an image consisting of background pixels, "simple" foreground pixels as well as "knot" points that are foreground pixels that are end or intersection points. Then this image is converted into an ordered sequence of links that connect the knots. This approach is somewhat similar to the method described in [15] where knots are used as "anchors" to align both topological and geometric features of a glyph skeleton. In order to perform the skeleton traversal, we use a set of tracing rules that simulate the writing process. However, it is important to realize that *any* set of rules would work as long as it is consistent and robust to variations. For instance, the main right-to-left direction of tracing is not essential. After the links are computed, the next step is to detect loops of various types. After that all links that are not part of a loop are approximated by piecewise-linear curves. The final step of the preprocessor is to convert the resulting sequence of line segments into a sequence of observations. Each segment is either labeled as a part of a large-scale geometric feature such as a loop, an intersection point or a turning point, or quantized using k-means clustering according to its orientation and length. The label or the index of the k-means cluster centroid is used to compute a discrete observation symbol.

A more detailed description of Khorsheed's algorithm may be found in [5].

2.2 Modifications

We have made several modifications to the original methodology.

2.2.1 Smaller Feature Set

The complete feature set used in the original algorithm is shown in [5, Table 1]. In our preliminary work with synthetic data we reduced the number of features by computing fewer types of turning points and by reducing the number of k-means clusters (see Table 1). Experiments indicate that such a reduced set is sufficient to process the data at hand. We do realize, however, that as we get more involved with real-life data, the feature set may have to be expanded to provide more discriminative power. Such expansion will require relatively small changes to the recognition system as long as we continue to base it on discrete HMMs.

2.2.2 Separate Models for Various Character Forms

In the original algorithm, a given character model is used for all (2 or 4) positional forms of that character. We decided to use separate models for all the different form. The main reason for such a change is that, although this makes the resulting universal model several times bigger, it also makes it more accurate by limiting the number of connections between character models and therefore prohibiting or discouraging certain paths that would be allowed by the smaller "form-independent" universal model. For instance, there will be relatively few connections from models that represent final forms of characters into those representing initials character forms, because these can only occur in words consisting of two or more subwords, e.g. علي بابا ("Ali-Baba").

2.2.3 Training by Words Without Segmentation

Although during the recognition stage, one does not need to pre-segment the data beyond separation into words, during training manual segmentation into characters is still required for the original algorithm. Thus training of Khorsheed's system proceeds as follows.

1. For each training word:

    1.1 Manually pre-segment the word image into character sub-images.

    1.2 Convert all character sub-images into observation sequences.

2. For each character:

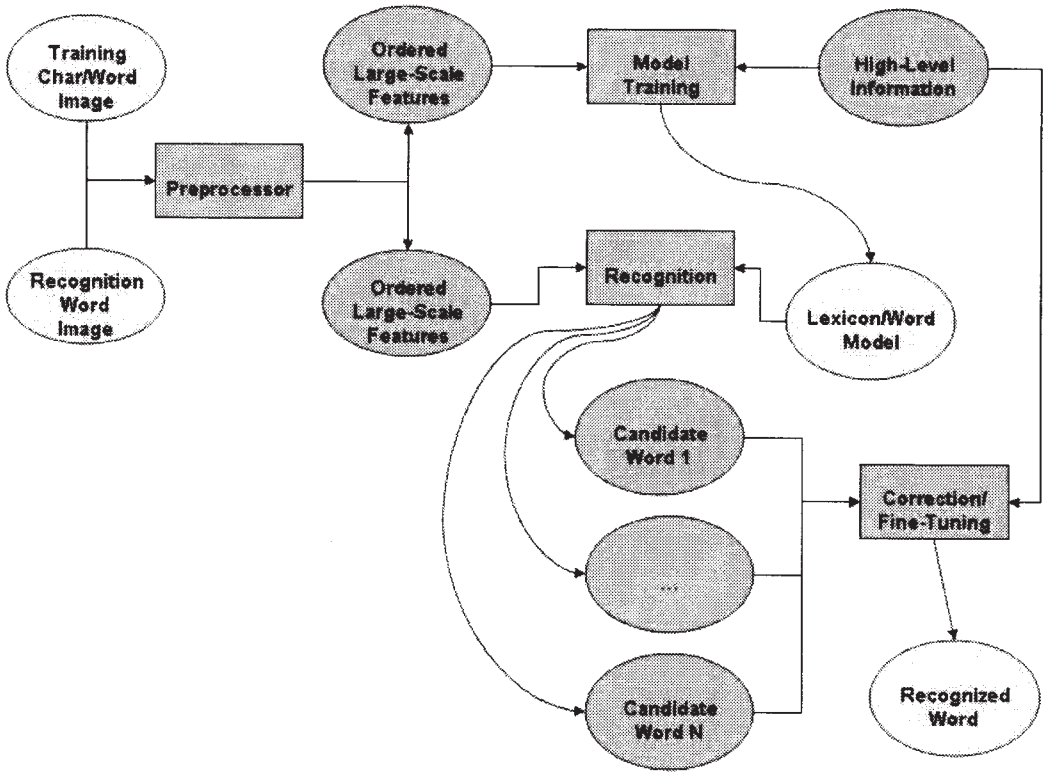


Figure 3: Block diagram of the recognition system

Observation Symbol	Feature Description	Observation Symbol	Feature Description
0	Dot	5	2-link complex loop
1	End point	6	3-link complex loop
2	Branch point	7	double loop
3	Cross point	8	Left turning point
4	Simple loop	9	Right turning point
		10-29	k-means cluster centroids

Table 1: Discrete features and the corresponding observation symbols

2.1 Compute the initial model. In particular, first determine the number of states.

2.2 Collect all training sequences for thet character from data generated in Step 1.

2.3 Train the character model.

3. Connect all character models into a universal model using character bigram statistics.

Note that in Step 2 there is a single instance of each character model that is trained once using all instances of that character extracted from all training words. Manual preprocessing of data is a big limitation for a real-life recognition system. We propose a way of modifying this approach to avoid the presegmentation. We assemble a word model for each training word possibly using several identical copies of the initial character model if the corresponding character appears in that word more than once. For instance, the name علي بابا ("Ali-Baba") contains two instances of the initial form of the letter ب ("Beh"). We then train the word models by corresponding observation sequences generated from the entire word images, just like when the model-discriminant approach to recognition described in Section 1 is used. Then we disassemble the trained word models into character components. Since the final universal model contains a single instance of each character model, after all training words have been processed, we have to combine all the instances of the same character models into one. This can be done by averaging. Thus, the training stage of the modified algorithm proceeds as follows.

1. For each character:

   1.1 Compute the initial model.

2. For each training word:

   2.1 Assemble initial character models into a word model.

   2.2 Convert entire training word image into an observation sequence.

   2.3 Use the sequence to train the word model.

   2.4 Disassemble the word model into instances of trained character models.

3. For each character:

   3.1 Average all instances of corresponding trained model into a single trained character model.

4. Connect all character models into a universal model using character bigram statistics.

In Section 3 we describe results of experiments with both manually segmented and unsegmented data.

3 Experiments

In this paper we present results of the initial "proof-of-concept" stage of our project that only involved experiments with synthetic data.

3.1 Manual Skeletonization and Tracing

In place of a fully automatic preprocessor we have implemented an interactive graphical tool designed to manually create ordered feature sequences of segmented letters. A user can superimpose an ordered piecewise-linear skeleton on top of an image of a "template" letter (see Figure 4. Each segment of the skeleton is labeled with a corresponding feature from Table 1 if this segment is a part of a loop, intersection etc. These skeletons are used to train individual character models and can also be concatenated into skeletons representing entire words. To simulate handwriting differences, both character and word skeletons can be randomly perturbed. Two types of perturbation are possible, namely,

- Adding random noise to geometric coordinates of endpoints of a segment (a.k.a. "shaking"). The amount of shaking is controlled by a standard deviation parameter.
- Random adding or removal of entire segments to/from a skeleton.

For instance, Figure 5 shows skeletons of the word محمد ("Mohammad") created by concatenation followed by various degrees of shaking of the letter skeletons shown in Figure 4.

3.2 Synthetic Data Tests

We performed three sets of tests with synthetic data. In the first two cases, we used machine-printed Arabic characters as templates when manually creating character skeletons. The *first* set of tests was based on training using segmented data. To train a total of 102 character models for the various positional forms of the main 29 letters of the Arabic alphabet, we manually created and labeled skeletons of each character form. Each of these skeletons was perturbed by various degrees of shaking and this data was used to train individual models that were then assembled into a universal model based on bigram statistics extracted from a set of about 2,000 unique words collected from a set of newspaper articles. The character models contained between 3 and 20 hidden states. The same word collection was used to evaluate the trained model: word skeletons were obtained by concatenation of the corresponding character skeletons, then they were shaken.

The *second* set of tests involved unsegmented training data. In this case, 90% of the word collection was used for training and the other 10%

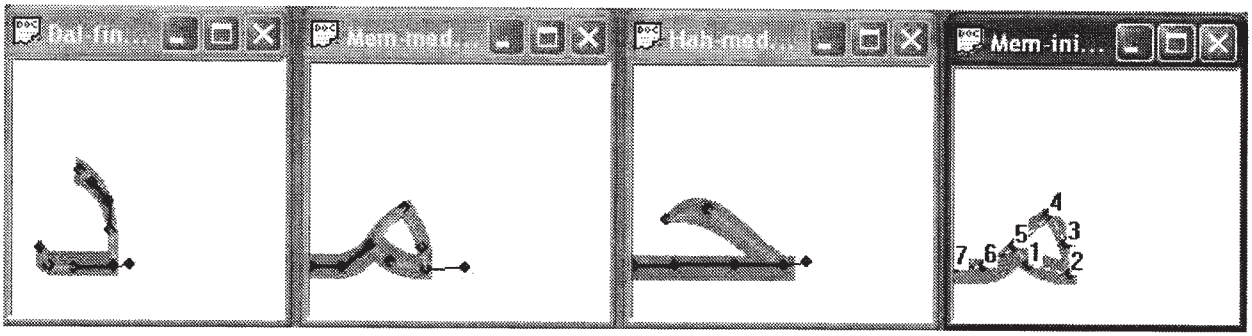


Figure 4: Manually created and labeled skeletons of letters م ("Mem", initial and medial forms), ح ("Hah", medial form) and د ("Dal", final form)

for testing. For each training word, we created a word skeleton by character skeleton concatenation and generated an initial word model by using the same initial (untrained) character models as in the first set of tests. In this case, the word skeletons were perturbed both by shaking and by random addition and removal of segments. Then training was performed as described in Section 2.2.3. Test data was then created in a similar way for the test words and the universal model was evaluated. Performance of the recognition system for the two sets of tests is reported in Table 2. As expected, recognition accuracy for the segmentation-based tests was higher than for tests without segmentation. However, even in the latter case, it was high enough to warrant the use of this methodology in processing of more realistic data.

The *third* test involved several handwritten samples from the corpus described in [8]. We wanted to show that, at least using data that is mostly manually preprocessed, it is possible to model one handwriting style using perturbed versions of other styles. From the corpus, we selected two different words that, in particular, had three letters in common. For each word, we found three handwritten samples of the word from three different writers. In addition, we generated a machine-printed copy of each word. Using the interactive tool described in Section 3.1, we manually created and labeled skeletons for all eight samples (see Figure 6). Using the same methodology as in the second set of tests described above, we trained two word models using observation sequences obtained from perturbed versions of the skeletons of the handwritten samples. Then we disassembled the word models and created a universal model using bigram statistics from the two words and averaging instances of those character models that appeared more than once. We then tested the universal model using the observation sequences obtained from (unperturbed) skeletons of the machine-printed versions of the two words. In both cases, our recognition system found the correct paths through the universal model. We also found that perturbation of training samples was critical: when we did not use shaking when training the word models, we could not find *any* path through the resulting universal model for one of the test observation sequences.

4 Future Work

This section explores various ways of extending our system, e.g. automating the initial image processing stage and the high-level information usage during the recognition stage.

4.1 Building Automatic Pre-processor

The current image preprocessing stage is largely manual. One of our immediate goals is to build a robust image preprocessor. We plan to use the algorithm by Zhang and Suen [16] for efficient thinning, followed by skeletanization via the curve fitting method by Liao and Huang [6]. Then after the proper segment labeling and ordering, the skeleton would be fed to the existing HMM-based recognition engine.

Although our system aims primarily at processing bitonal images, grayscale imagery can provide better recognition cues. Razdan et al. proposed an interesting approach to aid handwriting analysis and OCR [12, 13] by applying 3-dimensional modeling to analysis of handwritten documents. Their technique permits recovery of 3D spatial-temporal information from a 2D grayscale document image, potentially turning an off-line recognition problem into an on-line recognition problem that could be easier to solve. We could employ this technique in the pre-processing stage of our system to naturally order the sequence of HMM observations.

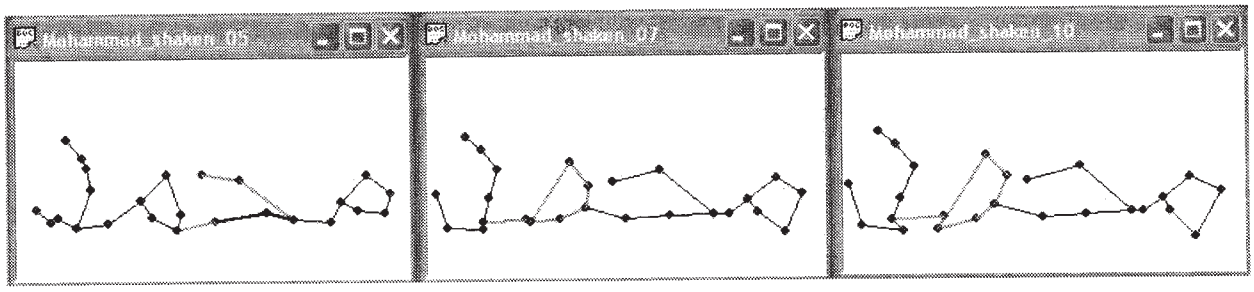


Figure 5: Shaken skeleton of the word محمد ("Mohammad") obtained by concatenation of skeletons of the letters shown in Figure 4

4.2 Using High-level Information

Ambiguity is unavoidable in handwriting recognition. Figure 7 shows an example of a word image which may equally likely be interpreted as "dear" or "clear" even by the best recognition system - a human - unless additional context information is provided. Therefore we believe that incorporation of

Figure 7: Example of ambiguity

multiple types of high-level information into a handwriting recognition system will be essential for producing high quality results. This information may include limited lexicons, various language models (such as the letter and word n-grams), as well as context, location and co-location information.

References

[1] Richard G. Casey and Eric Lecolinet. A survey of methods and strategies in character recognition. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 18(7):690–706, July 1996.

[2] Michael Decerbo, Ehry MacRostie, and Premkumar Natarajan. The bbn byblos pashto ocr system. In *HDP '04: Proceedings of the 1st ACM workshop on Hardcopy document processing*, pages 29–32, New York, NY, USA, 2004. ACM Press.

[3] Shimon Edelman, Tamar Flash, and Shimon Ullman. Reading cursive handwriting by alignment of letter prototypes. *International Journal of Computer Vision*, 5(3):303–331, 1990.

[4] Mohammad Khorsheed. Off-line Arabic character recognition - a review. *Pattern Analysis and Applications*, 5(1):31–45, 2002.

[5] Mohammad S. Khorsheed. Recognising handwritten Arabic manuscripts using a single hidden Markov model. *Pattern Recognition Letters*, 24(14):2235–2242, 2003.

[6] C.-W. Liao and J. S. Huang. Stroke segmentation by Bernstein-Bezier curve fitting. *Pattern Recognition*, 23(5):475–484, 1990.

[7] Hee-Seon Park and Seong-Whan Lee. A truly 2-d Hiddne Markov Model for off-line handwritten character recognition. *Pattern Recognition*, 31(12):1849–1864, 1998.

[8] M. Pechwitz, S. Snoussi Maddouri, V. Märgner, N. Ellouze, and H. Amiri. IFN/ENIT-database of handwritten Arabic words. In *Proceedings of CIFED*, pages 129–136, 2002.

[9] Mario Pechwitz and Volker Märgner. HMM based approach for handwritten Arabic word recognition using the IFN/ENIT-database. In *Proceedings of ICDAR*, 2003.

[10] Rejean Plamondon and Sargur Srihari. Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 22(1):63–84, January 2000.

[11] Lawrence R. Rabiner. A tutorial on HMM and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[12] Anshuman Razdan, John Femiani, and Jeremy Rowe. 3D methods to aid handwriting analysis and OCR. In *Proceedings of the Symposium on Document Image Understanding Technology*, page 287, April 2003.

[13] Anshuman Razdan, John Femiani, and Jeremy Rowe. 3d techniques for analyzing handwritten manuscripts for digital libraries. In *Inter-*

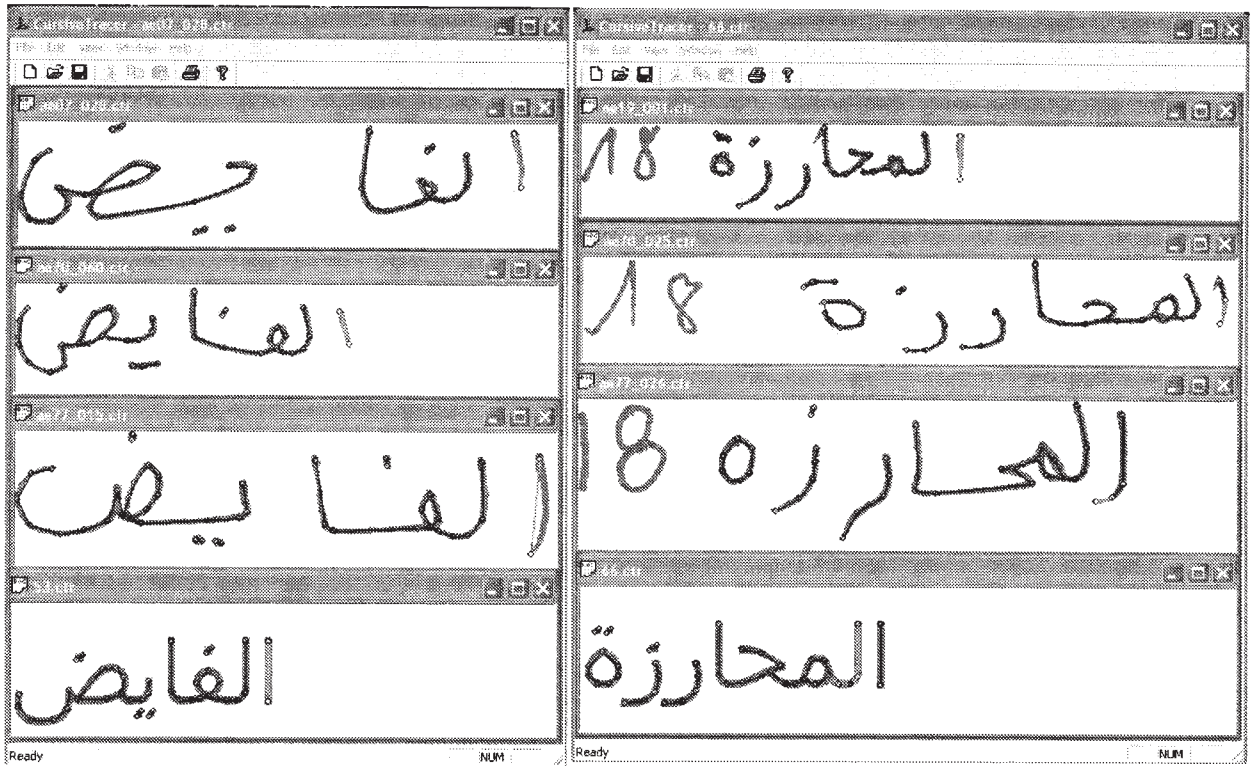


Figure 6: Two words, three handwritten and one machine-printed styles of each, manually preprocessed

national Conference on Digital Libraries, pages 430–436, New Delhi, February 24-27 2004.

[14] Tal Steinherz, Ehud Rivlin, and Nathan Intrator. Off-line cursive script word recognition - a survey. *International Journal of Document Analysis and Recognition*, 2(2–3):90–110, 1999.

[15] Mark A. Walch and Donald T. Gantz. Pictographic matching: a graph-based approach towards a language independent document exploitation platform. In *HDP '04: Proceedings of the 1st ACM workshop on Hardcopy document processing*, pages 53–62, New York, NY, USA, 2004. ACM Press.

[16] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239, 1984.

Training ...	Character Accuracy		Word Accuracy		
	Precision	Recall	Number of Word Seq's	Number of Matches	Recognition Rate
with Segmentation	99.5%	99.6%	107950	105691	97.9%
without Segmentation	93.1%	92.1%	10800	8059	74.6%

Table 2: Performance of the recognition system on synthetic data when training on both segmented and unsegmented data

Classification of Machine Print and Handwriting in Mixed Arabic Documents

Karthik Sridharan **Faisal Farooq** **Venu Govindaraju**
CEDAR, Suite 202, SUNY at Buffalo
Amherst, NY 14228, USA
{ks236,ffarooq2,govind}@cedar.buffalo.edu

Abstract

In this paper we present a system for classification of machine printed and handwritten text in mixed Arabic documents. The classification is performed at the word level. We propose a feature extraction algorithm for each word image based on Gabor filters followed by classification using an Expectation Maximization(EM) based probabilistic neural network. The main contributions of this paper are a Gabor filter based feature extraction for classification and design of an EM based neural network that reduces overfitting of training data. An overall precision of 94.62% *was obtained using our method as compared to* 83.33% *using a simple backpropagation neural network and* 90.26% *using an SVM.*

1 Introduction

The processing of document images prior to recognition plays a significant part in the development of Handwriting Recognition (HR) systems. In a document that has both machine print and handwritten text, it is important to distinguish between the two. This task is challenging in Arabic because the script is cursive in both machine print and handwriting and there are no special rules that can easily distinguish them.

In this paper we describe a method that extracts texture features from Arabic words. An EM based neural network is used for classification to deal with the sparse training data that does not have representatives from all fonts and writing styles.

A neural network based classifier was suggested in [1] that used nine texture features to distinguish machine print from the handwritten text in bank checks. Srihari et al. [2] describe a block separation method where the classification is based on the frequency of the heights of the different components in the segmented block. It is assumed that a block with widely differing heights is handwritten and a block with uniform component heights is machine printed. This method does not apply readily to Arabic script.

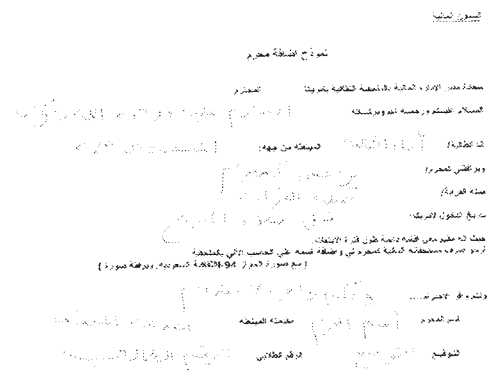


Figure 1: A sample document

Our hypothesis is that in Arabic handwriting, horizontal runs and gradients are not as uniform as in machine print. The advantage of our method is that it can be implemented at the word level as it captures the local structure of components in the document. A discrimination method that operates at the word level was described in [3], using slope and stroke width histograms. However, the method was not trainable and thresholds were selected empirically.

Figure 1 shows a block diagram of Our approach. It has 3 stages : (i) word extraction (ii) feature extraction and (iii) classification. In the word extraction stage, we binarize [7] the image and extract individual word images from the document [8]. Each word image is normalized by scaling to a fixed height while preserving the aspect ratio, hence the width of the word images vary. The baseline of the connected components is estimated and corrected. Lines are segmented into words by clustering connected components based on thresholds chosen by the normalized variance of white space in the lines. Directional Gabor filters are used to extract features from the word image. Classification is performed by a probabilistic neural network which is trained using an

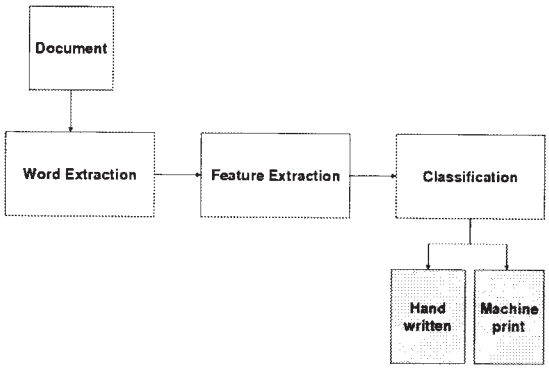


Figure 2: Components of the system

EM algorithm. This neural network combines solutions according to their posterior distribution to avoid overfitting based on the training data.

2 Feature Extraction

Gabor filters are directional filters that have been used successfully for classification of textures and automatic script identification [4]. Since direction of strokes and uniformity is a key feature in Arabic script, the use of Gabor filters seem to be ideally suited for the task.

Gabor functions are Gaussian functions modulated by a complex sinusoid. In $2D$, a Gabor function is given by:

$$h(x,y) = g(x',y').\epsilon^{2\pi jFx'}$$

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\epsilon^{-\frac{1}{2}[(\frac{x}{\sigma_x})^2+(\frac{y}{\sigma_y})^2]}$$

where (x',y') are rotated components of (x,y),

$$x' = xcos\theta + ysin\theta$$

$$y' = -xsin\theta + ycos\theta$$

and F is the radial frequency which for a given scale s is given by $F = F_0/s$. The output of the filter,

$$G_{\theta,s}(x,y) = \int I(s,t)h_{\theta,s}(x-s,y-t)dsdt$$

is an image with the components in the chosen direction becoming prominent. Since machine print Arabic has more uniformity as compared to handwriting and the same characters repeated in the text have strokes in the same direction, Gabor filters for feature extraction is a prudent choice. Figure 2 shows a sample word image extracted. Figure 3 shows the output of the Gabor filter for each direction at a single scale when applied to the word image in Figure 2.

Figure 3: Word image

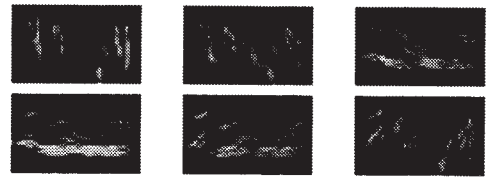


Figure 4: Gabor Filter Output for 1 scale and 6 orientations

Since the word images all vary in their width the Gabor filter cannot be applied directly. For classifiers like neural networks or support vector machines (SVM) the feature vectors need to be of fixed size. This problem can be resolved by noting that the main information obtained from the Gabor filter output is the strength of the word image in each direction and scale which is given by the sum of the output of the filter for each direction and scale resulting in a vector of size [number of scales × number of directions]. In order to make it font independent we normalize the output by dividing the sum of filter output by the sum of the output of an isotropic Gaussian filter,

$$Gauss(x,y) = \int I(s,t)g(x-s,y-t)dsdt$$

For direction θ and scale s

$$Gabor(\theta,s) = \frac{\int G_{\theta,s}(x,y)dxdy}{\int Gauss(x,y)dxdy}$$

In our implementation we use a set of 6 filter banks with 3 directions and 2 scales. Thus for each word image we extract a 6-dimensional feature vector for classification.

3 Classification

The training set is generally sparse and does not cover all fonts. Traditional classifiers like SVMs and backpropagation neural networks tend to overfit sparse data. This drawback is overcome by the Bayesian Neural Networks(BNN) [5] by integrating over the posterior distribution of the weights. That is, instead of finding one solution, many solutions are found and are weighted according to their posterior probabilities given the training data. The BNN outperforms many classifiers including the SVM. However BNNs need to sample high dimensional weight vectors from their posterior distribution. Markov Chain Monte-Carlo sampling methods, such as Langevian Monte Carlo method and

Hamiltonian sampling methods can be used for the purpose. However these methods are computationally expensive.

A BNN for a binary classification can viewed as linear combination of potential solutions according to their posterior probabilities. Since sampling is computationally intensive, we propose a new neural network where a layer of neurons use an error function which apart from penalizing neurons responsible for errors in classification, also penalizes neurons that are similar to each other.

The idea is to make the neurons compete in finding different possible solutions. The part of the error function that penalizes solutions that lead to bad classification is given by the Euclidean distance between the target and the output. The part of the error function that penalizes similar solutions is given by the sum of the square of the cosine of the neuron weight vector with respect to the weight vectors of the other neurons in the layer. A bias term is included in the weight vector to make sure that all the hyperplanes given by the neurons need not pass through the origin. Thus the error function of a single neuron is given by

$$E_{i,k} = \frac{1}{2}(t_k - o_k)^2 + \frac{\beta}{2}\sum_{j!=i} \frac{w_i^T w_j}{|w_i||w_j|}^2 \qquad (1)$$

where

$$cos(v,u) = \frac{v^T u}{|v||u|}$$

and t_k is the target for the kth instance and o_k is the weighted sum of the output of all the neurons according to their posterior probabilities. Therefore,

$$o_k = \sum_i P(w_i) o_{k,i}$$

The transfer function used is the classic sigmoid function. One way of looking at this error function is as the negative log likelihood of the posterior. In this view we would be modeling the likelihood of the output to follow a Gaussian distribution with mean around the target and the prior to be a Gaussian distribution of the cosine function of the weight of the neuron with the other neurons with zero mean. Zero mean of the cosine signifies that we are trying to model the neurons such that the cosine of their weights with the other neurons is as orthogonal as possible since $cos(90) = 0$. Parameter β decides the trade off between the error on classification and how "different" the solutions should be. By minimizing the error function we obtain weights that are as orthogonal (different) to each other as possible and yet classify well.

We also need to weight the solutions given by the neurons according to their posterior probabilities. We propose the use of an Expectation Maximization (EM) algorithm [6] for learning the weights and probabilities. In the E step of the algorithm we find the posterior probabilities of the neurons given the training data. In the M step we optimize the weight parameters to reduce error. While the E step is exact, the M step uses the gradient descent approximation. If we take the error function to be negative log likelihood then in probability space for the kth instance and i^{th} neuron we get posterior probability of the weight as

$$\begin{aligned} & P(w_i | o_k = t_k, w_{j:j\neq i}, I_k) \\ &= \frac{P(o_k = t_k | w_i, w_{j:j\neq i}, I_k) P(w_i | w_{j:j\neq i}, I_k)}{\sum_i P(o_k = t_k | w_i, w_{j:j\neq i}, I_k) P(w_i | w_{j:j\neq i}, I_k)} \\ &\propto \epsilon^{-\frac{\eta}{2}(o_k - t_k)^2} \epsilon^{-\frac{\beta\eta}{2}\sum_{j!=i} \frac{w_i^T w_j}{|w_i||w_j|}^2} \\ &\propto \epsilon^{-E_{i,k}} \end{aligned}$$

where η is the learning rate for the gradient descent on weights and β is a free parameter. I_k denotes the kth input. The EM algorithm uses a batch update of weights so all the instances are considered at once and the parameters of all the neurons are updated simultaneously. Let w_i^t represent the weight of the ith neuron after t updates by the algorithm. The probability of getting all the classifications correct given that the weights of the neurons in the previous step $t-1$ is W^{t-1} ($W^{t-1} = \{w_i^{t-1} | i = 1:n\}$ where n is the number of neurons) is

$$\prod_k P(o_k = t_k | W^{t-1}, I_k)$$

$$= \prod_k \sum_i P(o_k = t_k, h_i^t | W^{t-1}, I_k)$$

and h_i is a hidden variable.

We assume that the instances are independent. Taking log on both sides,

$$\sum_k log(P(o_k = t_k | W^{t-1}, I_k))$$

$$= \sum_k log(\sum_i P(o_k = t_k, h_i^t | W^{t-1}, I_k))$$

Now we simply multiply and divide the term in the sum by some positive non-zero $\alpha_{i,k}^t$ where $\sum_i \alpha_{i,k}^t = 1$ (The α's are probability distribution of the latent or hidden variables) and hence,

$$\sum_k log(P(o_k = t_k | I_k))$$

$$= \sum_k log(\sum_i \frac{P(o_k = t_k, h_i^t | W^{t-1}, I_k)\alpha_{i,k}^t}{\alpha_{i,k}^t})$$

By Jenson's inequality we have

$$\sum_k log(P(o_k = t_k|I_k))$$

$$\geq \sum_k \sum_i \alpha^t_{i,k} log(\frac{P(o_k = t_k, h^t_i|W^{t-1}, I_k)}{\alpha^t_{i,k}})$$

Further splitting the joint probability $P(o_k = t_k, h^t_i|W^{t-1}, I_k)$ as the product of $P(o_k = t_k|W^{t-1}, I_k)$ and $P(h^t_i|o_k = t_k, W^{t-1}, I_k)$ we get,

$$\sum_k log(P(o_k = t_k|I_k))$$

$$\geq \sum_k \sum_i \{\alpha^t_{i,k} log(\frac{P(o_k = t_k|W^{t-1}, I_k)}{\alpha^t_{i,k}})$$

$$+log(\frac{P(h^t_i|o_k = t_k, W^{t-1}, I_k)}{\alpha^t_{i,k}})\}$$

Let $O = T$ represent the event when all instances are classified correctly and let

$$F(\alpha^t, W^{t-1}) =$$

$$\sum_k \sum_i \{\alpha^t_{i,k} log(\frac{P(o_k = t_k|W^{t-1}, I_k)}{\alpha^t_{i,k}})$$

$$+log(\frac{P(h^t_i|o_k = t_k, W^{t-1}, I_k)}{\alpha^t_{i,k}})\} \quad (2)$$

Therefore we have

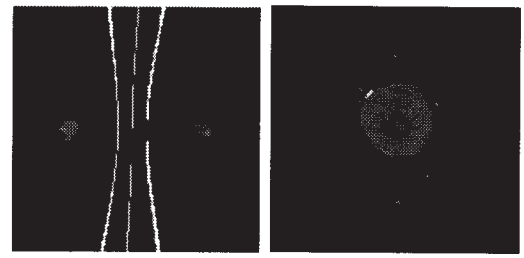
$$log(P(O = T|I)) \geq F(\alpha^t, W^{t-1})$$

Since we cannot build an algorithm that directly optimizes $log(P(O = T|I_k))$ we instead optimize $F(\alpha^t, W^t)$ as this guarantees that the likelihood always increases. Further the expectation maximization algorithm alternately optimizes the α's and the weights keeping one constant while optimizing the other thus performing a co-ordinate ascent. In the E step the α's are optimized and in the M step the weights are optimized and thus the EM algorithm iterates between successive E and M steps.

In the E step we need to minimize $F(\alpha^t, W^t)$ with respect to α^t. Hence we differentiate $F(\alpha^t, W^t)$ with respect to each $\alpha^t_{i,k}$ keeping weights constant and solve the equation for the differential equal to 0. However it is important to note that since α's are 0, they should sum to one for each instance. Hence we use the Lagrange multiplier λ in the equation and get

$$\frac{dF(\alpha^t, W^{t-1})}{d\alpha^t_{i,k}} =$$

$$log(\frac{P(o_k = t_k|W^{t-1}, I_k)P(h^t_i|W^{t-1}, I_k}{\alpha^t_{i,k}}) - 1 + \lambda$$



(a) Linear Classification

(b) Non-linear Classification

Figure 5: 2D classification Results

$$= 0$$

Therefore to minimize $F(\alpha^t, W^{t-1})$ with respect to the α's, we have

$$\alpha^t_{i,k} = P(o_k = t_k|W^{t-1}, I_k)P(h^t_i|W^{t-1}, I_k)\epsilon^{\lambda-1}$$

But sum of α^t_k's is equal to 1 for each data instance, thus

$$\epsilon^{\lambda-1}\sum_i P(h^t_i|W^{t-1}, I_k)P(o_k = t_k|W^{t-1}, I_k) = 1$$

To minimize $F(\alpha^t, W^{t-1})$ with respect to the *alpha*'s we simply set

$$\alpha^t_{i,k} = \frac{P(o_k = t_k|W^{t-1}, I_k)P(h^t_i|W^{t-1}, I_k)}{\sum_j P(o_k = t_k|W^{t-1}, I_k)P(h^t_j|W^{t-1}, I_k)}$$

Since the error function is the negative log of probability we have ,

$$P(o_k = t_k|W^{t-1}, \pi^{t-1}, I_k) \propto \epsilon^{-E_{i,k}}$$

where $E_{i,k}$ is got from Equation 3.

Now let

$$\pi^t_i = P(h^t_i|W^{t-1}, I_k)$$

be the prior associated with each neuron. Therefore, the update of α's in the E step to minimize $F(\alpha^t, W^{t-1})$ is given by

$$\alpha^t_{i,k} = \frac{\epsilon^{-E_{i,k}}\pi^{t-1}_i}{\sum_j \epsilon^{-E_{j,k}}\pi^{t-1}_j} \quad (3)$$

From 2 we have

$$F(\alpha^t, W^t) = \sum_k \sum_i \alpha^t_{i,k} log(\frac{P(o_k = t_k|W^{t-1}, I_k)\pi_i}{\alpha^t_{i,k}})$$

In the M step we minimize $F(\alpha^t, W^t)$ with respect to each of the weights W^t and the priors. To find the optimal priors we differentiate $F(\alpha^t, W^t)$ with respect to each of the π's. However, since the priors

Table 1: Performance analysis of the system.

	Back-Prop. Neural Net		SVM		EM Neural Net	
	Precision(%)	Recall(%)	Precision(%)	Recall(%)	Precision(%)	Recall(%)
Handwritten	62.26	95.19	74.26	97.12	94.68	85.58
Machine-print	97.83	79.02	98.82	87.76	94.93	98.25
Overall Performance(%)	83.33		90.26		94.62	

must add up to 1, we use the Lagrange multiplier λ' to ascertain this condition. Thus for minimizing $F(\alpha^t, W^t)$ with respect to the prior,

$$\frac{dF(\alpha^t, W^t)}{d\pi_i^t}$$

$$= \sum_k \frac{d(\alpha_{i,k}^t log(P(o_k = t_k | W^{t-1}, I_k)\pi_i^t)}{d\pi_i^t} + \lambda' = 0$$

$$\Rightarrow \sum_k \frac{\alpha_{i,k}^t}{\pi_i^t} + \lambda' = 0;$$

Therefore,

$$\pi_i^t = -\frac{\sum_k \alpha_{i,j}^t}{\lambda'}$$

But the priors sum to one. Therefore

$$\lambda' = -\sum_i \sum_k \alpha_{i,k}^t$$

For minimizing $F(\alpha^t, W^t)$ we set priors as

$$\pi_i^t = \frac{\sum_k \alpha_{i,k}^t}{\sum_i \sum_k \alpha_{i,k}^t}$$

$$\Rightarrow \pi_i^t = \frac{\sum_k \alpha_{i,k}^t}{K} \quad (4)$$

Where K is the number of instances of training data.

To minimize $F(\alpha^t, W^t)$ with respect to each of the weights W^t we use the gradient descent. The gradient of $F(\alpha^t, W^t)$ with respect to each of the weights w_i^t is given by

$$\frac{dF(\alpha^t, W^t)}{dw_i^t} \propto \sum_k \frac{d(\alpha_{i,k}^t log(e^{-E_{i,k}} \pi_i^t)}{dw_i^t}$$

$$\propto -\sum_k \alpha_{i,k}^t \frac{dE_{i,k}}{dw_i^t}$$

Since $\alpha_{i,k}^t$ is constant for all i and k, we need to perform gradient descent on error function $E_{i,k}$. This means we simply follow the negative gradient of error function $E_{i,k}$ which is given by

$$-\frac{dE_i}{dw_i^{t-1}}$$

$$\propto -\eta \sum_k (t_k - o_k)(1 - o_{k,i}) o_{k,i} I_k$$

$$-\beta\eta \sum_{j!=i} \frac{(w_i^{t-1})^T w_j^{t-1}}{|w_i^{t-1}||w_j^{t-1}|} \frac{w_j^{t-1} - w_j^{t-1} \frac{w_i^{t-1}(w_j^{t-1})^T}{|w_i^{t-1}|^2}}{|w_i^{t-1}||w_j^{t-1}|} \quad (5)$$

Where $o_{k,i}$ represents the output of the ith neuron. Since we follow the negative gradient, the M step is partial. The EM algorithm is initialized with setting weights to some random vectors and making all priors equal. In the E step we use Equation 3 to update the posteriors based on the weights and priors from the previous iteration. In the M step we use Equation 4 to set the priors based on the calculated posterior and use the gradient given by Equation 5 to perform gradient descent on the weights of the neural network.

To initially test the performance of the neural network, we used some 2D test cases. Figure 5(a) shows the performance of the method on the linear classification problem. The central gray band stands for 0.5 probability region and the side gray bands stand for 0.75 and 0.25 probability regions respectively. The results obtained are similar to the Bayesian decision boundary. It curves out at places where data is not available thus showing that the system is unsure at places where training data is unavailable. Figure 5(b) shows the decision boundary found by the method on a non-linear test set. Here we see the gray band around the data points in the center which is the decision boundary which stands for 0.5 probability region.

4 Performance Analysis

There is a lack of labeled handwritten datasets for training and testing purposes in Arabic. We have collected handwriting samples from forms that have prompts in machine print. Figure 1 shows an example of the document. We collected 34 documents from 18 different writers. We used 5 documents for training purpose and the remaining for testing.

We measured the performance of our system by the precision and recall metrics, commonly used by the Information Retrieval (IR) community. Precision in our case would be the ratio of handwritten

words labeled correctly to all words that are labeled as handwritten by our system. Recall is measured as the ratio of handwritten words labeled correctly to all handwritten words in the test set. Similarly the corresponding metrics for machine print are also calculated. We used a training data of just 50 words each of handwritten and machine print images and a testing data of documents containing a total of 286 machine print and 104 handwritten words in them. Table 1 shows the summary of our experimental results. In order to evaluate the performance of our classification step, we compared the results by using a back-propagation neural network and an SVM for classification. The overall precision of our system is 94.62%. Our system outperformed a backpropagation neural network (83.33%) and also an SVM (90.26%).

5 Conclusion

Discrimination of handwritten and machine printed text is required in many document analysis and forensic applications. We have presented an algorithm for discriminating Arabic handwriting from machine print, which is a difficult task when compared to other languages. However, our method is trainable and relies on the uniformity of strokes and curves in machine print compared to handwriting. Given the training data, our method can be adapted to other languages and scripts as well. We compared our proposed classification technique with commonly used back-propagation neural network and state-of-the-art SVM. Our EM based neural network outperformed the standard systems. The EM based neural network addresses the problem of sparse data, thus our method is robust even when large amounts of training data is not available.

References

[1] E.B.D.S. Jose, B. Dubuisson and F. Bortolozzi, Distinguishing between Handwritten and Machine Printed Text in Bank Cheque Images,*Document Analysis Systems, LNCS* (2002) 2423, 58–61.

[2] P.W. Palumbo, S.N. Srihari, J. Soh, R. Sridhar and V. Demjanenko, Postal Address Block Location in Real-Time, *IEEE Computer* (1992), 34–42.

[3] F. Farooq, V. Govindaraju, and M. Perrone. Processing of Handwritten Arabic Documents, *Proceedings of the 12th Conference of the International Graphonomics Society* (2005) 183–186, Salerno, Italy

[4] P.B. Pati, S.S. Raju, N. Pati and A.G. Ramakrishnan, Gabor filters for Document analysis in Indian Bilingual Documents, *Proceedings of International Conference on Intelligent Sensing and Information Processing*, (2004) 123–126, Chennai, India.

[5] R.M. Neal, Bayesian Learning for Neural Networks, *Springer Verlag* (1996), New York.

[6] Dempster A., Laird N., and Rubin D. *Maximum Likelihood from Incomplete Data via the EM Algorithm*, Journal of the Royal Statistical Society, Series B, 39(1):138, 1977.

[7] N. Otsu, A threshold selection method from gray-level histograms,*IEEE Transactions on Systems Man and Cybernetics* (1979) 9(1), 62-66.

[8] F. Farooq, V. Govindaraju, and M. Perrone. Pre-Processing Methods for Arabic Handwritten Documents, *Proceedings of the International Conference on Document Analysis and Recognition* (2005) 267–271, Seoul, Korea

Extraction and Information Retrieval

Accurate Document Categorization of OCR Generated Text

Robert Price[1] **Anthony Zukas**[2]
[1]Content Analyst LLC, Reston, VA
[2]SAIC, Reston, VA

Abstract

The purpose of this research was to examine the application of the Latent Semantic Indexing (LSI) algorithm to the categorization of Optical Character Recognition (OCR) generated documents (or text). LSI is a robust dimensionality reduction technique for the processing of textual data. The technique can be applied to collections of documents independent of subject matter or language. Given a collection of documents, LSI indexing can be employed to create a vector space in which both the documents and their constituent terms can be represented. In practice, vector spaces of several hundred dimensions are typically employed. The resulting vector space possesses some unique properties that make it well suited to a range of information-processing problems. Of particular interest to the document conversion community is the fact that the technique is highly resistant to noise in text that is generated by the OCR conversion process. Noise in OCR generated documents nominally takes the form of missing characters or improperly interpreted characters that result in word misspellings. Normally, human operators are employed to perform corrective post processing of noisy OCR-generated text prior to categorization or other document workflow processes that require highly accurate text (e.g., Boolean searches). A technology such as LSI that could eliminate human review and editing of OCR-generated text, while maintaining highly accurate categorization or information retrieval rates, would result in a dramatic increase in workflow throughput with substantial labor savings.

1 Introduction

Previous work [1] has demonstrated the high performance of Latent Semantic Indexing (LSI) [5] on the Reuters-21578 [9] test set. In this paper we have examined the ability of LSI to categorize documents that contain corrupted or noisy text (e.g., misspellings, transliterations differences, OCR errors). In an earlier case study for a US Intelligence Agency related to a pending FOIA release, utilizing documents derived via OCR, indicated the LSI technology could possibly provide good retrieval performance on OCR generated text. Attempts at using other technologies to examine the document set for potential "mosaic effect" connections had failed.

A key feature of LSI that makes it attractive for categorizing noisy text is its capability of automatically extracting the conceptual content of text items [5, 6]. With knowledge of their content, these items then can be treated in an intelligent manner. For example, documents can be routed to individuals based on their job responsibilities. Similarly, e-mails (or documents with noisy text) can be filtered accurately. Information retrieval operations can be carried out based on the conceptual content of documents, not on the specific words that they contain (or variants thereof generated by noise sources, e.g., OCR). Of particular note is that the LSI algorithm does not employ any auxiliary data structures as part of its processing (e.g., thesauri, grammars, taxonomies or ontologies).

2 Discussion

Numerous pattern matching technologies have been studied and applied towards finding relevant search patterns in noisy text [2,4]. If the image conversion process is poor (i.e., originals are of poor quality) and the characteristics of the OCR engine are well understood a model of the overall conversion process can be created and employed to train a pattern matching engine for searching the OCR text. Such models are too expensive and time consuming to produce in practice. If the image conversion process is relatively clean (i.e., originals are of good quality, image enhancement software and production quality scanning equipment is used) then pattern matching technologies may be able to overcome some inconsistencies in the OCR text and achieve some reasonable confidence in the precision/recall of the search results. Often the real world is somewhere in between [11].

This problem is also very akin to the conversion of text from speech and subsequent retrieval of the text by search [3]. Techniques described in this paper are also applicable to this domain.

3 Approach

To formally examine the robustness of the LSI algorithm under conditions of noisy text a

categorization-style testing approach was selected since the authors had previous experience testing the LSI algorithm using this mode of operation. The Reuters categorization test sets were selected due to their availability and active use in other categorization testing efforts by other researchers. The Reuters categorization test sets come in two versions (Reuters-21578 [9] and Reuters RCV1-V2 [10]) and consist of document training and document test sets. The training set provides a knowledge base to train the algorithm undergoing examination and the test documents are used in evaluating the categorization accuracy of the subject algorithm. In the case of the LSI algorithm the training documents were employed to train the LSI vector space. The makeup of the Reuter's corpuses is discussed in the following section.

In the case of LSI the training documents are used to create a matrix that relates the documents and the words that occur in them. The rows of the matrix correspond to *terms* that occur in the documents. The columns correspond to individual *documents*. The number entered in the *i*th row and *j*th column of the matrix corresponds to the number of times that the *i*th term appears in the *j*th document. The matrix produced in this manner can be very large. In practical applications it can involve hundreds of thousands of terms and even larger numbers of documents. Fortunately, the matrix is very sparse and is amenable to dimensionality reduction.

A powerful mathematical technique, known as singular value decomposition (SVD), is used to reduce this matrix to a product of three matrices. One of these matrices has non-zero values only on the diagonal. Small values on this diagonal, and their corresponding rows and columns in the other two matrices are then deleted. This truncation process generates a matrix of greatly reduced dimensionality. For any given dimensionality, this technique can be shown to produce an optimal approximation of the original matrix. The columns of the associated matrices can be used to create a vector space in which both terms and documents are represented. The dimensionality of this vector space can be chosen to work well in a particular application. Typically, LSI spaces with a dimensionality of several hundred are employed. The dimensionality reduction has the effect of extracting semantic information that is latent in the processed text, hence the name latent semantic indexing. In some cases unlabeled background data is used to enhance the knowledge base of the training set [7]. This approach was not utilized in this work.

To test the robustness of LSI in conditions of increasing document degradation a procedure was developed that degraded percentages of text in the test documents by inserting, deleting, and substituting characters randomly at specified error rates based on experience with numerous OCR engines and OCR degradation models published in the literature [4]. In our model, for any given document, 66 percent of the errors were random alphanumeric substitutions, 17 percent were deletions, 12 percent were insertions of random alphanumeric characters and 5 percent were random insertions of spaces. Although true OCR errors are more systematic, the intent here was to show to what extent the text of the documents could be degraded and still retain useful categorization results. Appendix A illustrates the impact of the degradation algorithm on one test document at various stages of degradation.

The collections of degraded test documents were then presented to the LSI algorithm for categorization and the accuracy results recorded.

4 Test Corpus and Performance Measures

To understand the performance of LSI on degraded text, the ModApte version of the Reuters-21578 test set [9] was utilized. A second test using the larger more recent RCV1-V2 test set was also performed to examine the impacts of a larger training set and much larger test set.

The ModApte version has been used in a wide number of studies [8] due to the fact that unlabeled documents have been eliminated and categories have at least one document in the training set and the test set. We followed the ModApte split defined in the Reuters-21578 data set in which 71% of the articles (6552 articles) are used as labeled training documents and 29% of the articles (2581 articles) are used to test the accuracy of category assignments.

The larger Reuters RCV1-V2 [10] consists of 804,414 documents broken up into 23,149 training documents and 781,265 test documents. We employed the Topic category orientation of the test set which consists of 103 Topic categories.

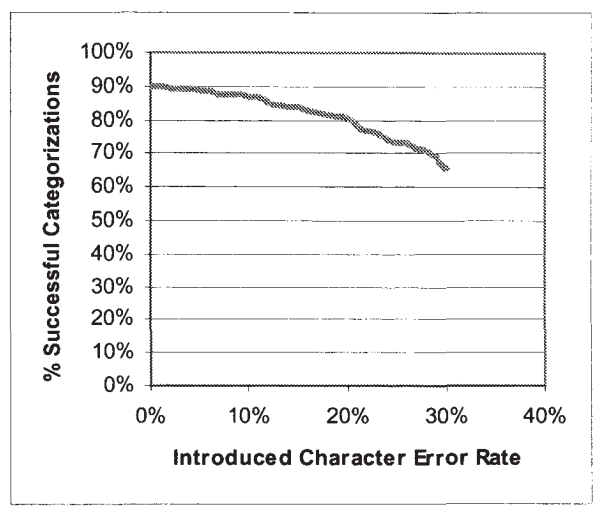


Figure 1. Reuters-21578 document categorization accuracy versus introduced character error rate.

For evaluating the effectiveness of category assignments to documents by LSI we adopted the *break-even point* (the arithmetic average of precision and recall) as reported in [12], and the total ('micro-averaged') precision P and recall R (defined in [13]).

5 Results

The categorization accuracy of LSI on the Reuters-21578 test set was reported in detail in [1]. Figure 1 shows the performance of the LSI algorithm on the Reuters-21578 test set with regards to categorization accuracy for various percentages of document degradation. As can be observed in Figure 1 the categorization accuracy falls off at a very slow rate for an introduced character error rate of 0 to 30 percent.

To compare results across the two Reuters data sets the authors plotted baseline categorization accuracy versus the degradation levels of the degraded text. Figure 2 (Reuters-21578) and Figure 3 (Reuters RCV1-V2) show the plots of the overall categorization accuracy compared to the baseline versus the introduced character error rate.

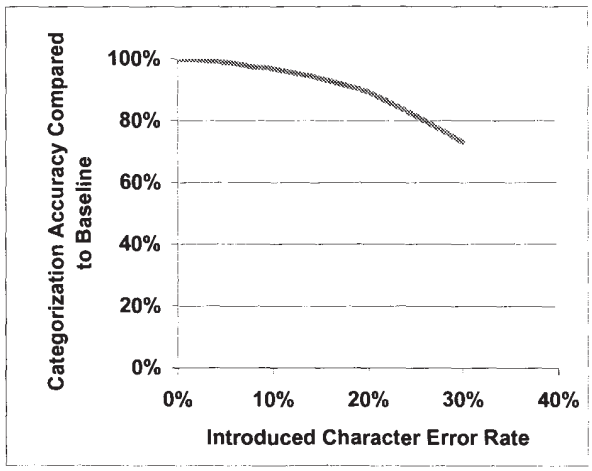


Figure 2. Reuters-21578 categorization accuracy compared to baseline versus introduced character error rate.

It is interesting to note that the two curves are very similar up to an introduced character error rate of 15 percent. Examination of Figure 3 revealed an interesting artifact in the Reuters RCV1-V2 curve. After an introduced character error rate of 15 percent, the curve for the Reuters RCV1-V2 test set falls off much faster than the curve for the Reuters-21578 test set. As noted previously the Reuters RCV1-V2 test set contains 718,265 test documents versus 2,581 test documents for the Reuters-21578 test set. One explanation for the differences (and of continuing author investigation) is the larger number of new unique word combinations generated by degrading the RCV1-V2. There is a 33:1 ratio for the number of test documents versus training documents in the Reuters RCV1-V2 document set. This same ratio is 2.5:1 for the Reuters-21578 document collection (ModApte variation). It could be possible that a larger training set (than the one supplied with the RCV1-V2 test collection) is required at higher levels of document degradation.

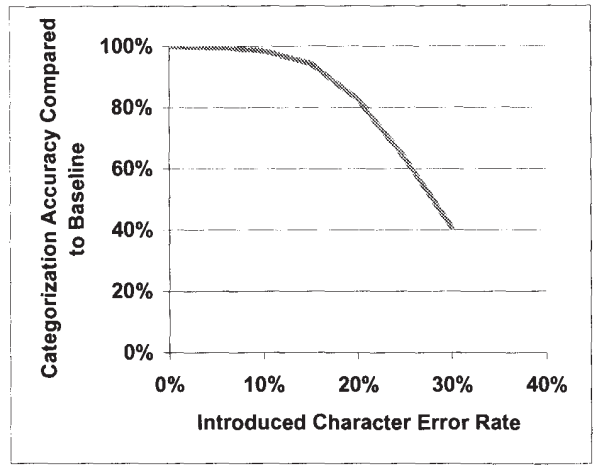


Figure 3. Reuters RCV1-V2 categorization accuracy compared to baseline versus introduced character error rate.

Figures 4 and 5 show plots of the baseline categorization accuracy versus the percentage of corrupted words. The interest in these plots stemmed from the observation that at a 20% introduced character error rate roughly three-quarters of the words in a test document are corrupted. It can be observed that the LSI algorithm maintains a very flat curve out to a point where approximately 60% of the words are corrupted in a test document.

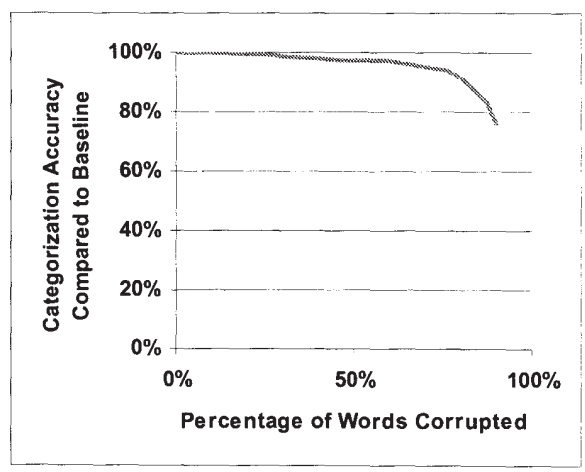


Figure 4. Reuters-21578 categorization accuracy compared to baseline versus the percentage of words corrupted.

To the authors' knowledge this level of information retrieval on noisy text has never been achieved using

other technologies and is worthy of further investigation. Further examination of Figures 4 and 5 also show the more rapid falloff at the tail of the curve between the two document collections as observed in Figures 2 and 3.

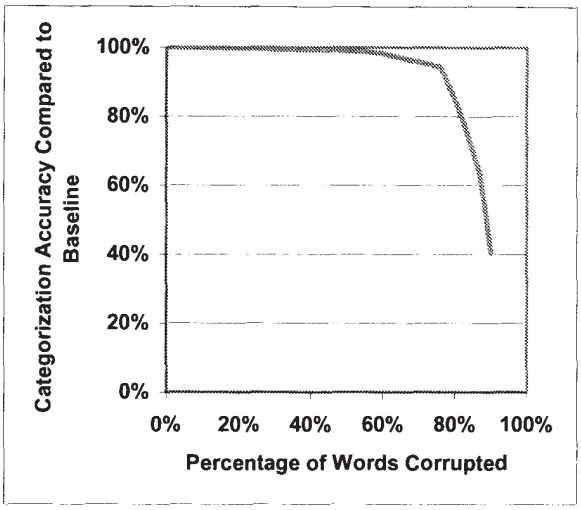


Figure 5. Reuters RCV1-V2 categorization accuracy compared to baseline versus the percentage of words corrupted.

6 Conclusions

The authors find these results very encouraging. The evidence that the accuracy of the LSI algorithm falls off very slowly in categorization testing, even at high levels of text errors, indicates LSI could be an excellent solution to the general indexing of noisy text. In most applications where OCR text is targeted for information retrieval, the text conversion process most often requires a human-in-the-loop to either (1) extract document meta data which is indexed (rather than the OCR text) or (2) reviewers read the OCR text and provide human-intensive review and correction prior to full-text indexing. The results of this study indicate that reasonably accurate OCR conversion processes combined with LSI indexing could eliminate the human review process normally associated with the indexing of OCR text.

Future research will address the application of LSI in other areas, e.g., indexing of machine readable text as a result of speech-to-text conversion, the use of LSI-based text summarization of noisy text to replace human-generated meta data, and the potential of using LSI to mitigate the effects of noisy data on machine translation systems.

Acknowledgements

We thank Roger Bradford, and Dr. Janusz Wnek for their useful comments when reviewing this paper.

References

[1] A. Zukas and R. Price, Document Categorization Using Latent Semantic Indexing. In Proceedings: 2003 Symposium on Document Image Understanding Technology, Greenbelt MD, April 2003 87-91.

[2] J. Jeuring, Polytypic Pattern Matching, In Proceedings of the Seventh International Conference on Functional Programming Languages and Computer Architecture, ACM (1995) 238-248.

[3] A. Singhal and F. Pereira, Document Expansion for Speech Retrieval, Proceedings SIGIR (1999) 34-41.

[4] R. Baeza-Yates and G. Navarro, A Practical Index for Text Retrieval Allowing Errors, In CLEI, volume 1 (1997) 273-282.

[5] S. Deerwester et al. Indexing by Latent Semantic Analysis, Journal of the American Society for Information Science, 41(6), pp. 391-407.

[6] T. Landauer and D. Lanham. Learning Human-like Knowledge by Singular Value Decomposition: a Progress Report, Advances in Neural Information Processing Systems 10 (1998), Cambridge: MIT Press, pp. 45-51.

[7] K. Nigam. Using unlabeled data to improve text classification, PhD. Thesis, Carnegie Mellon University, May 2001.

[8] S. Rice, F. Jenkins, T. Nartker. The Fifth Annual Test of OCR Accuracy. Technical Report TR-96-01, Information Science Research Institute, University of Nevada, Las Vegas, NV, 1996.

[9] D. Lewis. Reuters-21578 Text Categorization Test Collection. Distribution 1.0. README file (version 1.2). Manuscript, September 26, 1997. http://www.daviddlewis.com/resources/testcollection/reuters21578/readme.html.

[10] D. Lewis, Y. Yang, T. Rose, F. Li. RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research, 5(2004):361-397, 2004. http://www.jmlr.org/papers/volume5/lewis04a/lewis04a.pdf.

[11] H. Baird, Document Image Models and Their Uses, ICDAR, Proc. Intl. Conf. Document Anal. Recog. (1992) 62-67.

[12] Y. Yang and X. Liu. A Re-examination of Text Categorization Methods, Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), (1999) 42-49.

[13] Y. Yang. An Evaluation of Statistical Approaches to Text Categorization, Journal of

Information Retrieval, Volume 1, No. 1/2 (1999) 67-88.

Appendix A

China wheat purchases reach 80 pct of 1996 target. China's total state purchases of wheat reached 18.53 million tonnes by the end of August, accounting for 80.26 percent of the target for 1996, the official People's Daily said on Sunday. Of the total, purchases of wheat at state-fixed prices were 13.11 million tonnes, more than 90 percent of the year's target, the newspaper said. The state also buys at market prices. The state bought 1.95 million tonnes of rapeseed, fulfilling 66.21 percent of state plans, it said.China's grain output was expected to be a record 475 million tonnes in 1996, the Xinhua news agency said on Sunday.

Figure A-1. Sample document with no errors.

hina wheat purchases reach8O pct of 1996 target.ChinB's total stae purchaWes of wheat reached 18.53 BilliOn tonnes by the 6nd of August, accountingMfo 8.26 percent of th target for 1e96, the officiZl People's Daily said on Sunday.Of the total, purchases of wheatam statO-fixed prices were 13.11 million tonnes, more than 90 percent of the year's target, the newspapeV said. vZe stateBalso buysMat mapket prices.The ytate bXught 1.95 million tonnes of rapeseed, fulfi7li0g 66.21 phrcent of state plans, it said.ChinI's grain output wGs expected to be a record 45 million tonnes in 1996, th3 Xinhua news agency said on Sunday.

Figure A-2. Sample document with 5% errors.

Wina wheat purchases reach 80ZpctvoQ 1996 target.hina's totVl state purchases4of whHat reayheP Y8.53 milfio7 tNnnes bx theMendFofU guPt, accouting loXU80.26 percFnt of ths t3rgetdfor 1996,3the official stopleesMDaol6 said6on Sunday.Of the total, purchases of wReat at state-fixejlprices wer2 13.11 mi3lYon tonnes,more uhan 90wp4rcentFof th0 yealE3 tajgep, thl nespapsGd ip Thv state alfo iuys at Tarket RricjsUT0e state boGght 1.95 millMon tons of rpese d, fulpilling 66.21 per enteof state plans,Sit snia.C ina1s graiT ogtput was exGecUed to be a record 47Hmil Won tonnFs in Q996, thedXinhuP nwws ageBcy0said on unfaM.

Figure A-3. Sample document with 15% errors.

CHinKwhet pGrcreM reacv 80 p SPmav996 taretoChinyY yo4aj dwatr purchaleUOof wheHt r1lhGe o8.5k miTion tonnes bh the Pnd of A g0st, ac0ubtzn5pf1q J .g6bpXr8e2tUfqthestarget for 1N96, tueo8iKCaA PeopleOsqVaily sacd Pn ScndayfOflth stota0t Curchases lf wPeam pw hD t7Gfxew LricsVwere 16.11 mtlyuoo tnn0, moLe than90 AOre8t of the yead'shtargzt, the newPla0er sbso The stxtK alsp buysdat by et Hivq.ThB ptctV bouAhtr1.9V million t4nnesof rmp9sQeu,kfulfpllin 06P21 perOxnt ol s at YlkGs, 0t saidOChYnaes grain 61put wQs eApected tocbeYo record L5 millionbqoPnes e f9968 the XQnhua nFls ageOy sai4 on Sundayj

Figure A-4. Sample document with 30% errors.

Effect of Degraded Input on Statistical Machine Translation

Faisal Farooq[1]
CEDAR, SUNY at Buffalo
Amherst, NY 14228
ffarooq2@cedar.buffalo.edu

Yaser Al-Onaizan
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
onaizan@us.ibm.com

Abstract

In this paper, we study the effects of degraded (noisy) input on the accuracy of Natural Language Processing(NLP) systems such as Machine Translation(MT). The noisy input is often the only available input as it is the output of another NLP application such as an Optical Character Recognition (OCR) system. We propose a method for reducing the noise in the input by correcting the output of an OCR system. We show that correcting OCR output reduces the degradation of the translation quality of a state-of-the-art Statistical Machine Translation (SMT) system when the corrected output is used, instead of the original output, as input to the MT system. We use a novel method for correcting the input based on a direct "phrase-based" translation system in contrast to traditional HMM source-channel models.

1 Introduction

Many NLP systems such as Information Retrieval or Speech-to-Speech Translation systems are multimodal, where the input is often in different multimedia format such as scanned images of written text, audio, or video files. The building blocks of such systems are often other NLP applications pipelined together to form the wider NLP system. For instance a cross-lingual information retrieval system might be composed of an optical character recognition(OCR) system, a machine translation system, and a search engine. Therefore, such components must be robust against noisy input.

Component systems are often developed independently and possibly by different groups. In most cases, the internals of one component are not accessible to the developers of the next component in the pipeline. In such cases, these components (e.g. OCR) must be treated as black boxes where only their output is observed (Figure 1a). Unfortunately, the output of these systems is error-prone, especially for non-Roman languages (e.g., Arabic) which have not been researched as extensively. Therefore, the next component must be robust enough to handle noisy input. The problem is compounded when the input to OCR systems, in the form of images, is also noisy.

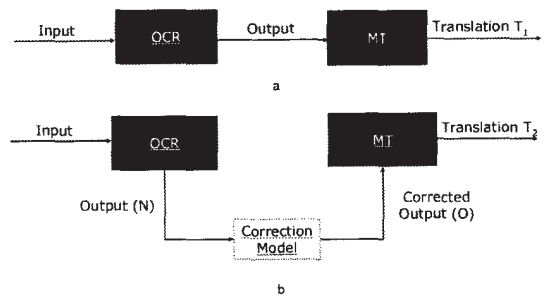


Figure 1: a) Typical NLP architecture b) NLP architecture with proposed correction model.

In this paper, we describe a probabilistic model for correcting the output of OCR systems in order to minimize the degradation of translation quality of our Statistical Machine Translation as shown in Figure 1b. We discuss the manner in which such models can be trained. We evaluate the utility of our correction model by using it to correct the output of an off-the-shelf, commercially-available, state-of-the-art Arabic OCR system. In section 2 we describe the related work. In section 3 we introduce our proposed technique. We describe our training data, test data and experimental settings in sections 4.1,4.2 and 5. We describe our models in sections 6 and 7 and the evaluation of our system performance in section 8. We finally conclude in section 9.

[1]The author was a summer intern at IBM T.J. Watson Research Center, NY

Table 1: OCR Error Rates for the three datasets.

	Training Data	Development Test Set	$MT'03$ Eval Set
WER(%)	25.20	21.21	28.87
CER(%)	10.60	8.16	11.00

2 Related Work

There has been some work on post-processing of the OCR output. A general survey of the research in this area can be found in [1]. Jones et al. [2] describe a multi-pass OCR post-processing system which carries out individual word corrections, combined edit distance corrections and bigram probability based correction in different passes. Perez-Cortes et al. [3] use a stochastic finite state machine to test hypothesis of words. If the machine accepts the word, then no correction is made, otherwise smallest set of transitions that could not be traversed show the most similar string in the model. Pal et al. [4] describe a method for OCR error correction of Devanagiri script using morphological parsing. Some of these techniques are highly dependent on the language and use features that are specific to the language in question, which makes such techniques difficult to adapt to a different language. The method we propose here is trainable. Although we show results on Arabic OCR and the effect on Arabic-to-English translation only, the technique is adaptable to other languages where training data is available. Okan et al. [5] introduce a generative probabilistic OCR model that describes an end-to-end process in a noisy channel framework. The technique they describe is implemented as a cascade of finite state transducers. We propose a different approach to this problem. We cast this problem as a simplified translation task where the source language is the error-full OCR output and the target language is the corrected output. We also present in section 8, a more extensive evaluation of the utility of our correction technique in a more realistic NLP application (i.e., Machine Translation).

3 Proposed Technique

The erroneous output from the OCR can be thought of as an output from a noisy channel problem[6]. It can be considered as a black-box through which the signal (truth) when passed gets corrupted and emerges out as the degraded output. We can learn the corruption models (correction model in the direct model framework) and then use this knowledge to correct the output. We leverage our experience in building SMT systems by casting this correction task as a simplified translation task. We propose a direct phrase-based translation approach that has proven effective in machine translation (e.g., [9], [7]).

As is done in a typical SMT system, given *sentence pairs* in the source (*foreign*) and the target (*english*) languages, we first align the source and target words. Word alignments can be obtained using one of several techniques such as HMM alignments [8], Maximum Entropy alignments [10], or Maximum-Posterior alignments [11]. Once a sentence pair is word-aligned, we extract phrase pairs (also referred to as block in the statistical machine translation literature) in a manner similar to [9]. These phrases are then used in combination with an n-gram language model to translate the source language into the target language.

We model the correction as a simplified translation task. The source language in our case is the OCR output (error-full) and the target language is the truth (correct). Since there is no word reordering as is the case of a typical MT task, the alignments can be obtained easily by minimum edit distance alignments or monotone HMM alignments. We used a simple dynamic programming minimum edit distance alignment. We approached this task in two ways. First, where the correction was achieved a character at a time, i.e., in terms of translation, the units of translation were characters instead of words as in a typical SMT system. The dynamic programming alignment between the truth and the corresponding OCR output was obtained at a character level. In the second method the same technique was applied, however, the units were words. Thus, we implemented two models - a character-based and a word-based. These models are explained further in sections 6 and 7.

4 Training and Test Data

4.1 Training Data

We use a commercially available Arabic OCR to recognize Arabic machine printed images for which we have the ground-truth. In order to create our correction training data, images of Arabic documents were needed. Since we did not have access to any publicly available large dataset of Arabic images and the corresponding text version, we automatically gener-

ated images from the arabic text of the LDC parallel corpus [14]. We used a Microsoft Word macro to open the *UTF-8* Arabic text files, format them using Bold Arabic Transparent Font at $14pt$. and print them into postscript files. The postscript files were then converted into jpg images at 300 dpi. This gave us a rich source of ~ 4000 images containing 23205 segments or $\sim 1.1M$ words. These images were recognized using the Arabic OCR and the OCR output, paired with the truth, was used as the training data for our models.

4.2 Test Data

The development and blind test sets were created in the same manner described above. The development test set was created from editorials from an online Arabic newspaper (www.asharqalawsat. com). It consists of 390 segments, $\sim 11K$ words. The blind test set was the NIST MT'03 [15] evaluation set. This test set is typically used to report the quality of machine translation systems. This set consisted of 663 segments or $\sim 15K$ words. This set was used two-fold 1) as a blind test set for the correction task, 2) to test the effect of input degradation on the SMT system. This was achieved by calculating the MT quality for the original set, then the quality of MT by translating the OCR output acquired from the corresponding images and finally for the corrected output.

5 Experimental Settings

We use the evaluation metrics reported by [5] for OCR evaluation. The evaluation metrics are the word error rate (WER) and the character error rate (CER) and are standard in measuring the accuracy of OCRs. The metrics are defined as follows:

$$WER = \frac{WordEditDistance(W_{truth}, W_{Output})}{|W_{truth}|}$$

$$CER = \frac{CharEditDistance(Truth, Output)}{|Truth|}$$

where edit distance is the Levenshtein Edit Distance and is defined as the minimum number of point mutations required to change one unit(character or word) into another unit, and a point mutation is one of – substitute, insert or delete a unit. $|Truth|$ and $|W_{truth}|$ is the number of characters and words respectively in the ground-truth. The error rates for the OCR for the three sets is given in Table 1.

Our correction model is evaluated first by measuring the reduction in recognition error rates. The error reduction is measured by comparing the error rate in N and O shown in Figure 1. Since MT is word-based, and any improvement in the WER

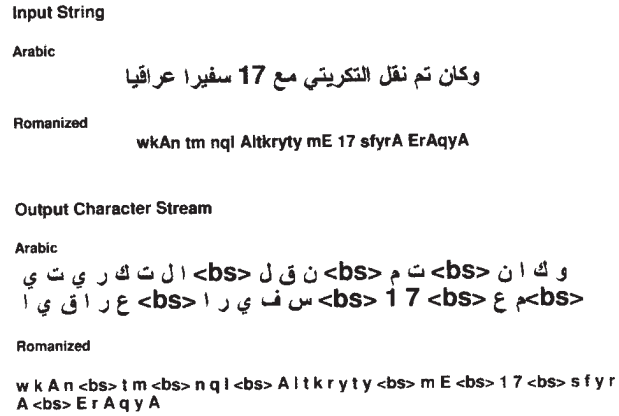


Figure 2: Conversion of input into a character stream.

would potentially affect the MT performance, in our further discussion we focus solely on the WER of the OCR or the corrected output. Secondly, we evaluate our correction model in the machine translation context by measuring the overall improvement in MT when the corrected OCR output is used instead of the original OCR output. This is achieved by comparing the translation quality (as measured by BLEU [12]) of T_1 and T_2 shown in Figure 1.

6 Character-based Model

Most OCR systems are conventionally segmentation based i.e. the text lines are segmented into words and words into their constituent characters prior to actual recognition. The OCR usually repeats the errors it makes in recognizing a certain character. In order to model the character-based nature of the OCR system, we defined the correction task as a translation of the character sequence that the OCR produces into the corresponding ground truth. Thus, the units of our input are characters instead of words in the case of a typical MT task.

6.1 Conversion to Character Stream

The first step for this task requires conversion of the input text into a stream of characters. This was achieved by separating individual characters by spaces. We used a special symbol <bs> to denote the word boundaries in order to differentiate it from the space between the characters. Figure 2 shows the same example of conversion in Arabic and romanized for illustration.

This was done for segments on both sides - the ground truth as well as the OCR output. Thus, our parallel corpus consisted of a character stream and its corresponding output from the OCR. It should be noted that the conversion on both sides into characters was performed after and not before recognition.

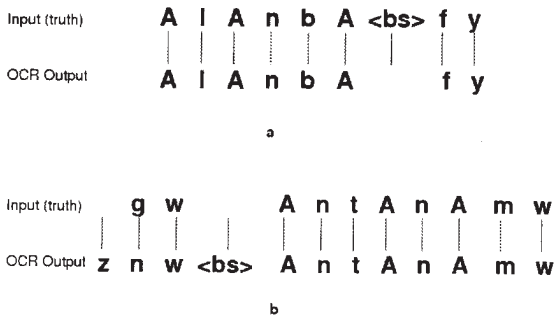


Figure 3: Examples of merge/split errors.

Source (OCR)	Target (Truth)	Count
ن ي(n y)	ن ي(n y)	17437
ن ي(n y)	غ ي(g y)	22
ن ي ب و(n y b w)	ن ي ب و(n y b w)	2
ن ي ب و(n y b w)	ن ي ق و(n y q w)	2
ن ي ب و س(n y b w s)	ن ي ق و س(n y q w s)	35

Figure 4: Phrases extracted from training data after conversion to character stream.

6.2 Training

In order to extract the phrases from the training data we align the truth and the OCR output character streams obtained above. Since there is no word re-ordering, we use a dynamic programming based minimum edit distance alignment. The edit distance was calculated using the regular Levenshtein edit distance formula with equal cost for deleting, inserting or substituting a unit. Using this, one-to-one alignments are obtained for the paired truth and output datasets. An unaligned character in the truth models a character deletion. Similarly an unaligned character in the OCR output models an insertion. Moreover, an unaligned <bs> in the truth models a merge error where the OCR merges two words and an unaligned <bs> in the OCR output models a word split. Figure 3a shows an example where two words were merged by the OCR into one and the <bs> is unaligned. Similarly, Figure 3b shows an example where a word was split and a character inserted (z). This time the <bs> on the OCR side is unaligned.

Using this alignment we extract all phrase pairs of lengths 1 through n ($n = 6$ in our case) characters. Figure 4 shows example phrases extracted from the training data.

6.3 Decoding

To correct the OCR output for a given test sentence, we "translate" this sentence by decoding using two weighted components - the phrases obtained above and the language model. We trained a character trigram language model by converting the gigaword corpus [13] into characters using the same procedure as above. The original corpus has 319 files totalling $\sim 4.3GB$ and $\sim 39.1M$ words. The language model is an interpolated language model. The *lambdas* were calculated from held out data, which is the last 10% of the gigaword corpus.

The decoder is a beam search decoder. For the correction problem, the decoding is carried out in a monotone fashion such that characters at the beginning of the sentence must be decoded first. Decoding hypotheses are extended in cardinality-synchronous fashion such that at any given time step n, only hypotheses that cover exactly n characters (i.e., of cardinality n) are extended. For example, at time step 0 only hypothesis of cardinality 0 (in the special init hypothesis) are extended. The decoding is continued until all characters in the sentence are covered. The final decoding is the hypothesis of cardinality l with the least cost, where l is the length of the source text. More formally, the final decoding ϵ for the source f is the one that satisfies the following equation:

$$\epsilon = \underset{\epsilon}{argmax}\ [-w_{ph} \times\ log_{10}P(\epsilon|f) \\ -w_{lm} \times\ log_{10}P(\epsilon)]$$

where $P(\epsilon)$ is the trigram character language model probability and $P(\epsilon|f)$ is the phrase-based direct model. The weights w_{ph} and w_{lm} of these components were trained using the first 50 segments of the development test set. the weight are trained using maximum BLEU training such that weights are adjusted to maximize the BLEU score. The weights that resulted in the maximum BLEU score for these 50 segments were selected for correcting the rest of the test set and any subsequent test sets using the monotone decoding as described earlier. Since, there were only two weights to be trained, we also adopted a grid-search technique. The translation accuracy(measured by WER) was calculated by varying the phrase component weight from 1.0 to 0.5 in steps of 0.1 and correspondingly the language model weight was varied from 0.0 to 0.5.

Table 2: Reduction in WER of development test set using word-based model before and after normalization.

	OCR Output	Corrected Output	Improvement (abs.)	Improvement (rel.)
WER % (Before Norm.)	21.21	20.08	1.13	5.32
WER % (After Norm.)	17.7	13.4	4.3	24.29

6.4 Results

The character-based model resulted in a 1% absolute (12% relative) reduction in the CER of the development test set. There was no improvement in the WER, which remained at 21.21%. This could be attributed to a number of reasons. In addition to the character trigram language model being weak, the phrase features were also not helpful. It was noted that the OCR had a post-processing step where the recognized output was converted to the closest possible dictionary word. This led to the fact that besides correctly recognizing the character, the rest of the confusion matrix was rather smooth than sharp as expected making the phrase features less reliable. Due to this post-processing, the actual error that the OCR would make on a character was hidden. By the creation of a dictionary word, the OCR output character could be replaced by another character and thus would not make it possible to model the error the OCR would make often.

7 Word-based Model

As noted in section 6.4, the character-based model proved to be weak to correct OCR errors. To overcome this, a word-based model is pursued, which proved to be useful. The word trigram language model would be much more beneficial since it provides more context.

7.1 Training

The training data was again aligned using the minimum edit distance, however, only this time based on words rather than characters. After aligning the truth and the OCR output, we extracted word phrases just as we extracted character phrases as described earlier in section 6.2. The drawback with this, however, would be that in order to get features for all possible words, we need much more data. In order to overcome this, we adopted the following procedure:

1. Extract phrasepairs of length=1 pairs $P = \bigcup p_w$, where p_w is a phrase pair containing source word w and all target word(s) from the training data.
2. Get the vocabulary V_t of the test set.
3. $\forall w_i \in V_t$, if $\exists p_{w_i} \in P$ such that $count(p_{w_i}) < T$ then $P = P \cup \{w_i, neighbour(w_i)\}$, i.e. add w_i and $neighbour(w_i)$ to P.

Given a very large language model vocabulary V_{lm}, the function $neighbour(w_i)$ yields all words $w_i' \in V_{lm}$ such that the minimum edit distance of w_i' from w_i is less than a threshold T'. For example, if the OCR output word was *AntAnAmw* (antanamo), $neighbour(AntAnAmw)$ would yield $\{AntAnAmw,\ gwAntAnAmw,\ ntAnyAhu, \ldots\}$ (antanamo, guantanamo, netanyahu,...). The thresholds T and T' were chosen empirically.

7.2 Decoding

Translation was carried out in the same manner as in case of a character-based model, however, the units this time were actual words. For this we trained a word trigram language model from the gigaword corpus[13] in a similar fashion as in section 6.3. However, this time there was no conversion into character streams. The weights for the phrase and the language model components were obtained by the grid search technique described in section 6.3.

7.3 Results

The word model performed better than the character model in general (1.13% for the dev. test set and 13.71% for the blind test set absolute improvement in WER as compared to no improvement by the character model). Since Arabic is highly inflectional, many of the errors were prefix/suffix errors. Our trigram language model had difficulty predicting the right set of prefix/suffix combination to choose for Arabic words. Other errors were less severe. For example, confusing various forms of *alef* (i.e., *alef* without *hamza* or with *hamza* above or below it). Since various forms of *alef* are mapped to one canonical form by the MT engine (in a process we call *normalization*), these errors were ignored by normalizing the OCR output as well as the corrected output that our model generated. The WER of the OCR after normalization was 17.7%. However, after normalizing the corrected output, the WER dropped

Table 3: Reduction in WER of $MT03$ test set using word-based model before and after normalization.

	OCR Output	Corrected Output	Improvement (abs.)	Improvement (rel.)
WER % (Before Norm.)	28.87	15.28	13.59	47.07
WER % (After Norm.)	13.71	11.54	2.17	15.83

from 20.08% to 13.4%. Thus, we were able to get a 4.3% absolute reduction in the WER after correction compared to 1.13% absolute reduction before normalization as shown in Table 2.

A similar improvement is also observed for our blind test set. The WER before normalization was reduced from 28.87% to 15.28%, and from 13.71% to 11.54% after normalization as shown in Table 3.

8 Evaluation

In order to evaluate the effect of errors introduced by the OCR in an Arabic to English Machine Translation system, we calculated the quality of MT before correction and after correction of the OCR output for the $MT03$ dataset. The translation is performed on the normalized text and the BLEU scores are calculated with the NIST provided reference[15]. Normalization is carried out as described in section 7.3. Table 4 shows the BLEU score for translation of the $MT03$ test set before and after correction. The BLEU score for the actual $MT03$ test set(truth) is also reported for comparison. As shown we achieve a statistically significant improvement in the BLEU scores.

9 Conclusion

We described a novel system for correcting the erroneous output of any OCR system. Our methods are generic and can be extended to any language. We achieved considerable reduction in word error rate. In addition, we demonstrated the utility of our system in a NLP application. OCR errors reduce the machine translation quality. However, this degradation can be minimized when our correction model is used to post-process the output of the OCR system. We were able to get a statistically significant increase in BLEU scores when our correction model is used (0.401 vs. 0.378).

However, the BLEU score obtained when translating the original text (truth) is still better than that of the corrected OCR output (0.465 vs. 0.401). A further reduction of the OCR word error rate might be necessary to reduce that gap. We believe that acquiring more training data for the word model will further reduce the error rates. The training data can be easily generated in the manner described in this paper. Such data can be easily obtained for any language or domain given that enough text is available electronically for that language/domain.

We also believe that our character-based model can be improved if used in conjunction with a word-based language model, instead of the character-based language model that was used.

References

[1] K. Kukich, Techniques for automatically correcting words in text, *ACM Computing Surveys* (1992) 24(4), 377–439.

[2] M.A. Jones, G.A. Story and B.W. Ballard, Integrating multiple knowledge sources in a a Bayesian OCR post-processor, *International Conference on Document Analysis and Recognition* (1991) 925–933, St. Malo, France.

[3] J.C. Perez-Cortes, J.C. Amerngual, J. Arlandis and R. Llobet, Stochastic error-correcting parsing for OCR post-processing, *International Conference on Pattern Recognition* (2000) 4405–4408, Barcelona, Spain.

[4] U. Pal, P.K. Kundu and B.B. Chaudhuri, OCR error correction of an inflectional Indian language using morphological parsing, *Journal of Information Science and Engineering* 16(6), 903–922.

[5] O. Kolak, W. Byrne and P. Resnik, A Generative Probabilistic OCR Model for NLP Applications, *Proceedings of HLT-NAACL* (2003) 55–62, Edmonton, Canada.

[6] L.R. Bahl, F. Jelinek, R.L. Mercer, A Maximum Likelihood Approach to Continuous Speech Recognition, *IEEE Transactions on PAMI* (1983) 5(2), 179–190.

[7] P. Koehn, F.J. Och and D. Marcu, Statistical Phrase-Based Translation, *Proceedings of the Human Language Technology Conference (HLT-NAACL)* (2003), Edmonton, Canada.

Table 4: BLEU scores for the $MT03$ test set(± 0.0164 for statistical significance).

	Ground Truth	OCR Output	Corrected Output
BLEU Score	0.4651	0.3776	0.4013

[8] S. Vogel, H. Ney, and C. Tillmann, HMM Based Word Alignment in Statistical Translation, *Proceedings of the 16th International Conference on Computational Linguistics (COLING'96)* (1996), 836–841, Copenhagen, Denmark.

[9] C. Tillmann and H. Ney, Word Re-ordering and a DP Beam Search Algorithm for Statistical Machine Translation,*Computational Linguistics* (2003) 29(1), 97–133

[10] A. Ittycheriah and S. Roukos, A Maximum Entropy Word Aligner for Arabic-English Machine Translation, *HLT-NAACL* (2005), Vancouver, Canada.

[11] B. Zhao, N. Ge and K. Papineni, Inner-Outer Bracket Models for Word Alignments using Hidden Blocks, *HLT-NAACL* (2005), Vancouver, Canada.

[12] K. Papineni, S. Roukos, T. Ward and W. Zhu, BLEU: a Method for Automatic Evalution of Machine Translation,*Proceedings of the 40th Annual Meeting of Association of Computational Linguists (ACL)* (2002), 311–318, Philadelphia, USA.

[13] Arabic Gigaword Corpus, *Linguistic Data Consortium (LDC), http://www.ldc.upenn.edu* (2003) Catalog LDC2004T17, Philadelphia, USA

[14] Arabic News Translation Text, *http://www.ldc.upenn.edu* (2004) Catalog LDC2003T12, Philadelphia, USA

[15] MT'03 Arabic to English MT Evaluation Set, *http://www.ldc.upenn.edu/* (2003), Philadelphia, USA

Mobile Interactive Support System for Time-Critical Document Exploitation

George Nagy
Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180
nagy@ecse.rpi.edu

Daniel Lopresti
Department of Computer Science
Lehigh University
Bethlehem, PA 18015
lopresti@cse.lehigh.edu

Abstract

Computer Assisted Visual Interactive Recognition, or CAVIAR, has proven to be an effective methodology for integrating human and machine expertise in addressing challenging pattern recognition tasks. In this paper, we examine the application of the CAVIAR paradigm to problems arising in real-time document analysis in the field. We identify hurdles we expect to encounter, propose potential solutions, and describe an evaluation framework to help determine whether this is a fruitful line of research.

1 Introduction

Research aimed at fully automating the processing of document images has received a tremendous amount of attention over the past 40 years. As a quick perusal of any of the dozens of surveys to date will reveal, however, progress in automatic recognition and interpretation has been slower than predicted. We expect that further improvement in accuracy in domains such as cursive handwriting and degraded documents will be even more protracted because the challenges that remain are much harder. As in speech recognition, bridging the gap between machine and human knowledge to allow the former to draw even with the latter appears problematic. The context brought by humans to any classification task is much greater than what can be obtained from even the largest collections of training samples available to our community. Endowing fully automated systems with broad knowledge remains a far-off goal.

There exists, however, a significant body of applications where it is not necessary to fully automate the task of document analysis. Rather, the focus is on a relatively small number of high-value documents (perhaps just one) – say from a cache discovered in the field as part of an ongoing criminal justice, military, or intelligence operation – where the computer plays the role of an assistant to help the user acquire information that would otherwise remain inaccessible. While such documents could be collected and returned to a central repository for scanning and batch processing in the traditional manner, there is often a significant advantage in being able to exploit the information in real-time, immediately and in situ.

Recently, Nagy and Zou, et al. have begun exploring a concept known as CAVIAR: Computer Assisted Visual Interactive Recognition [3, 6, 24, 23]. Over the last few years this work has led to the development of a successful interactive system for recognizing faces and flowers, both problems of a level of difficulty (i.e., automated current accuracy) comparable in certain ways to document recognition in the field. Experiments on sizable databases of faces and flowers indicate that interactive recognition is more than twice as fast as the unaided human, and yields an error rate ten times lower than state-of-the-art automated classifiers. The benefit margin of interactive recognition increases with improved automated classification. Parsimonious human interaction throughout the interpretation process is much better than operator intervention only at the beginning and the end, e.g., framing the objects to be recognized or dealing with rejects. Furthermore, this interactive architecture has been shown to scale up: it can start with only a single sample of each class, and it improves as recognized samples are added to the reference database.

The notions embodied in CAVIAR differ in fundamental ways from past efforts at mobile and/or interactive recognition. Whether such an approach can be equally effective in the domain of documents as it is for flowers and faces is unproven. In this paper, we discuss the application of this paradigm to document analysis in the field, identify the hurdles we expect to encounter, propose potential solutions, and describe an evaluation framework to help deter-

mine whether this is a fruitful line of research.

2 Related Work

The questions we plan to examine arise from adapting CAVIAR to tasks from document analysis. There are, however, other projects that share similar goals and assumptions; below we briefly cite a few of these. The Army Research Laboratory's Forward Area Language Converter (FALCon) system provides mobile optical character recognition (OCR) and translation capabilities [10, 21], but, so far as we know, employs a traditional user interface. A growing amount of research is being conducted on camera-based document acquisition (e.g., [7, 11]). This work employs the camera as a new form of capture device, but, as with FALCon, treats the later processing stages as though they will be fully automated

Camera-based systems for locating and recognizing text in traffic signs and providing translation services for non-native visitors in foreign lands are perhaps more similar to what we have in mind [8, 22]. Still, we have yet to encounter such a system with an interaction paradigm as integral and as sophisticated as CAVIAR's. Reading services for the vision-impaired are likewise focused on page-at-a-time processing, but employ an auditory user interface as opposed to a visual one for obvious reasons [9, 19]. A somewhat similar notion is recent work on developing tools to support forensic document analysis [20]. We note, however, that such systems are designed to solve a different, much more specific problem, are intended for off-line use by domain experts (as opposed to occasional users whose primary jobs lie elsewhere), and have no need for mobility.

3 Rationale for a Human-Machine Collaborative Approach to Document Analysis

A divide-and-conquer strategy for visual recognition should partition difficult domains into components that are relatively easier for both human and machine. There are pronounced differences between human and machine cognitive abilities. Humans excel in gestalt tasks, like object-background separation. We apply to recognition a rich set of contextual constraints and superior noise-filtering abilities. Computer vision systems, on the other hand, still have difficulty in recognizing "obvious" differences and generalizing from limited training sets. We can also easily read degraded text (e.g., CAPTCHA's [2]) on which the best optical character recognition systems produce only gibberish.

Computers, however, can perform many tasks faster and more accurately. Computers can store thousands of images and the associations between them, and never forget a name or a label. They can compute geometrical properties like higher-order moments whereas a human is challenged to determine even the centroid of a complex figure. Spatial frequency and other kernel transforms can be easily computed to differentiate similar textures. Computers can count thousands of connected components and sort them according to various criteria (size, aspect ratio, convexity). They can quickly measure lengths and areas. They can flawlessly evaluate multivariate conditional probabilities, decision functions, logic rules, and grammars. On the other hand, the study of psychophysics revealed that humans have limited memory and poor absolute judgment [16]. A detailed comparison of these differences appears in Table 1.

4 Technological Issues

Here we briefly summarize some of the technological issues that must be addressed in the implementation of a CAVIAR-like system to support document analysis and exploitation:

1. The rapid development of high-quality, low-cost digital cameras suitable for full-page imaging in color or a broad range of grays. Smaller versions of these cameras are now available as plug-ins for Personal Digital Assistants (PDA's) and within a year sufficient resolution should be available for camera phones. These devices make document-acquisition in the field simpler than with conventional page-width scanners.

2. Displays are approaching the resolution-limit of the human eye (discounting head movement). In addition to the small displays built into PDA's, cell phones, and helmets, larger flexible displays that can be incorporated into a user's clothing are on the verge of becoming available.

3. Interaction with a direct-action device like a stylus, and especially a thumb, is faster than with a mouse. Touch-sensitive screens are just now becoming available for cell phones.

4. The storage capacity of mobile devices is already sufficient for most DIA and OCR tasks.

5. Nagy, et al. have implemented interactive recognition on both stand-alone and wireless networked mobile platforms. Both are adequate for PDA's, but neither is yet acceptable for cell-phone based systems because these do not have

Table 1: Comparison of relative strengths of human vs. machine in visual pattern recognition.

Human	Machine
• dichotomies	• multi category classification
• figure-ground separation	• nonlinear, high-dimensional classification boundaries
• part-whole relationships	• store and recall many labeled reference patterns
• salience	• accurate estimation of statistical parameters
• extrapolation from limited training samples	• application of Markovian properties
• broad context	• estimation of decision functions from training samples
• gauging relative size and intensity	• evaluation of complex sets of rules
• detection of significant differences between objects	• precise measurement of individual features
• colored noise, texture	• enumeration
• non-linear feature dependence	• computation of geometric moments
• global optima in low dimensions	• orthogonal spatial transforms (e.g., wavelets)
	• connected component analysis
	• sorting and searching
	• rank-ordering items according to a criterion
	• additive white noise
	• salt & pepper noise
	• determination of local extrema in high-D spaces

enough computational resources for stand-alone operation, and their bandwidth is too low for acceptable interactive response time on image computations. However, according to technology forecasts, both of these shortcomings will disappear in a year or two.

6. The need for network connectivity depends greatly on the targeted application domain: civilian, military, or covert. While it is clearly desirable for such a system to be operable completely autonomously, there may be substantial value in networking document acquisition and exploitation activities.

5 Data Entry on Hand-Held Devices

It is clear that an important constituent of mobile interactive document analysis is manual text entry. The alternatives seem to be restricted to virtual keyboard on a touch sensitive screen activated by a handheld stylus, finger-operated keyboards incorporated in the operators clothing (on arm or thigh), and automatic speech recognition. We believe that the stylus is the most appropriate solution, because in addition to text entry it can also mediate the graphical communication essential in some aspects of document image analysis.

The virtual keyboard appeared in the seventies to accelerate the digitization of maps and line-drawings by avoiding having the operator shift constantly between pointing device and keyboard. It consisted of a picture of a keyboard on a piece of paper that could be shifted to the area of the drawing being vectorized. The current virtual keyboard usually appears in a fixed partition of the touch-sensitive screen of a handheld device. The text being entered appears on another partition. Edwards surveyed input interface issues in mobile devices in 1997 [4]. Data input is usually a local operation, so it makes little difference whether the device is networked or not.

The key consideration for stylus data entry are speed, (perceived) operator comfort, and ramp-up time. The first two factors are influenced by the amount of space allocated to the keyboard, to the recognized or entered text, and to control functions. The third factor depends heavily on the keyboard layout. The QWERTY layout, developed to prevent binding of type bars in mechanical typewriters, is suboptimal even for typing, and even more so for one-handed stylus entry.

There has been much work on evaluating alternative keyboards and word-completion algorithms (for a recent overview, see [1]). An upper bound on the speed of individual character entry is imposed

by Fitts' Law, which is a nonlinear relationship between pointing time and the distance and size of the target. The relevant distance is that between the screen areas ("keys") corresponding to consecutive letters. The letter transition frequency is given by a language model. As an example of the compromises necessary in keyboard design, it is possible to reduce the average distance by having several space keys, but this decreases the size of the keys. Several researchers have optimized keyboard designs according to various language models [14, 13, 15, 12]. The computed speeds hover about 40 words per minute, but actual text entry is much slower.

The speed increase obtainable by word completion also depends on the language model. Ancona [1] demonstrates a keyboard with separate keys for the ten most common words (with a cumulative word frequency of 28%). After each tap on the screen, the ten most likely words appear in the selection area of the screen. If the correct word is included, it can be selected with one additional tap. If not, another letter is tapped, which brings up ten new words. With a vocabulary of 13,000 words, the expected number of taps per word is claimed to be 3.3. Some of the notions here are clearly similar to those incorporated in CAVIAR.

The performance of word-completion systems depends on how well the stored lexicon is matched to the user input. Multiple lexicons – for different languages and applications – can be either stored on board, or downloaded via a wireless connection.

Another important source of ideas is the technology developed for vectorizing maps and engineering drawings [17, 18]. Manual vectorization was first conducted from hardcopy with a digitizing table or tablet. The operator traced the lines with cross-hairs under a magnifying glass with a MARK button. After the advent of large-size roller-feed scanners and bitmapped displays, all service bureaus and in-house operations converted to on-screen vectorization from scanned copy. One advantage was that already vectorized lines could be displayed with a different color, and departures between manually entered lines and original were clearly visible. The operator could zoom in on dense portions of the drawing. However, the aspects of interest here are the algorithms developed for semi-automated data entry.

For colored maps, different color layers were first separated according to RGB values. Vectorizing algorithms were manually initialized to a line segment or curve, and then could automatically follow that line at least to the next intersection point. The systems would also attempt to automatically recognize map and drawing symbols (for schools or resistors). If it failed, the operator would override it. The character recognition software recognized cleanly lettered labels (elevations, part numbers, resistor values), but left labels confused by overlaid line art or poor lettering to the operator.

If most of the labels cannot be recognized by OCR because of poor document quality or unusual character shapes, it is still possible to rapidly mark their location and orientation, rotate them to horizontal, and move them to a single area of the screen. This accelerates manual label entry. A single E-sized drawing may contain 3,000 alphanumeric symbols, which is more than a densely printed page of text.

Most such data-entry systems are part of larger GIS or CAD software, and are typically designed for standard workstations. All graphical operator interaction is therefore mediated by the mouse. As demonstrated by Engelbart and colleagues at SRI long ago, direct-action devices, like a touch-sensitive stylus, allow faster and more accurate interaction [5]. Indeed, we found that to be the case in comparing desk-top CAVIAR systems with handheld CAVIAR.

Like CAVIAR, these interactive systems exhibit clear speed advantages over completely manual data entry, and are robust enough (unlike automated systems) for operational application. Although some of these systems are laboriously trainable, one key difference compared to CAVIAR is that no commercial system that we are aware of incorporates adaptive algorithms that take advantage of routine operator input. Unlike CAVIAR, they also fail to provide visible models for the entry of complex 2-D patterns.

6 Test Data Collection

Any comparison of the proposed mobile interactive document exploitation system with existing and forthcoming automated and manual data entry systems requires a test database. It is, of course, desirable that the test database reflect the characteristics of the documents that are to be processed. For the purposes of this discussion, we assume that the domain of interest is handwritten Arabic documents; however, it should be clear how the the same issues and potential solutions generalize to other genres and languages.

We believe that existing handwritten Arabic databases cannot fulfill the objectives we have in mind for CAVIAR research because they are too small, they were designed specifically for testing a particular class of recognition algorithms, and they do not reflect the characteristics of the target data. For the sake of concreteness, we make the following assumptions:

1. The proposed system is to be applied to transcription and further processing of documents similar to those in a growing collection of handwritten Arabic documents, referred to here as the *Target Database*. Although we propose a mobile, personal device operated by a non-specialist, the test database will also be used for evaluating batch-mode DIA and OCR systems.

2. The amount of data and corresponding metadata, including ground truth (GT), will be comparable to an existing database used for evaluating typeset Arabic documents, i.e., 400,000 words. We take this as a fixed point for now. Handwritten documents have fewer words than printed text: say 100 words/document on average. The documents are not restricted to a single page, but each document is created by a single author.

3. Handwritten documents have statistically different textual content than typeset documents; therefore we cannot mirror the existing database. Handwritten material is less likely to form a grammatical narrative: it may have unusual abbreviations, short lists of phrases, unlabeled strings of digits, underlines, corrections, cross-outs and erasures, and unstylized layouts.

4. Character formation in free writing in any script is markedly different from copying or taking dictation from tape or a person. The conditions under which the document is composed, including the degree of stress or haste, will perceptibly alter even the same person's writing. We believe that this must be taken into consideration to avoid disappointment when the system is deployed.

5. The most important components of systematic variability are (a) country of education, (b) level of education (years of schooling), and (c) age of the writer.

6. Digitization and character recognition are affected by the medium: pen or pencil, n^{th} generation copy, fax, paper quality, and physical conditions (writing on a desk is different from writing in a hand-held notebook).

7. The chosen approach for (semi-)automated transcription should depend on the distribution of the amount of writing among individuals. If there are many long passages or multiple documents by the same writer, the system ought to take advantage of this. Language context and writer/style adaptation are powerful aids to recognition.

8. A service bureau with Arabic software can perform scanning and ground-truthing more efficiently than students. Such a service bureau could be located in a country with a significant Arab-speaking population (US, Canada, UK), or in a friendly Arab country (Jordan, Egypt) with an advanced computer infrastructure.

In order to plan data collection, it would be desirable to have the following information. The relevant document statistics could either be estimated (guessed) by a designated expert, or obtained from the agency responsible for processing the target documents.

Ideally, a sizeable random sample of the original documents (or very good copies) with translations from the Target Database, with accompanying metadata, would be made available. Most of the necessary information could be derived from such a sample. If, as is likely, such a sample cannot be released, then specification of the following distributions (in the form of histograms) would be useful:

- Number of *words per document*.
- Number of *documents per writer* and of *words per writer*.
- *Educational profile* of the writer.
- *Country of schooling* of the writer.
- *Vocabulary:* a lexicon indexed by frequency and, if possible, by writer.
- *Media:* pen, pencil, copy, fax, photo, etc.

In order to gauge feasibility, in Table 2 we provide a rough estimate (perhaps accurate to $\pm 50\%$) of the cost of deriving the required statistics from a hardcopy database of documents. These costs do not include the significant cost of collecting the documents, nor planning/management costs.

The handwriting data could consist of either: existing originals (or high-quality copies thereof), or handwritten documents generated for this purpose by about 1,000 subjects.

Possible sources of existing data include schools (assignments, term papers, course notes, administrative records), businesses (orders, reports, correspondence), and government agencies (dead files of forms and free-form correspondence). Privacy concerns may preclude certain of these approaches, however.

Table 2: Estimated costs of a hardcopy document ground-truthing activity.

Category	Cost
Scanning (200 or 300 dpi, 8-bit grayscale), for 4,000 mostly single-page documents, with quality control, at $1.00 per document	$4,000
Transcription, with verification and correction, at $2.00 per document	$8,000
System acquisition or customization for detailed (word-box level) zoning	$10,000
Interactive box location, verification and correction	$8,000
Database creation (accession numbers, bitmaps, boxes, GT)	$12,000
Estimated total	$42,000

If new data must be generated, a possible scenario for distributed data collection is:

1. Prepare from five to eight printed single or multi-page protocols on various topics, e.g., economics, politics, technology, military, religion. The topics should correspond to their proportions in the target database. The protocols should include questions that require a narrative answer, including requests for summarization, organization, decision, rebuttal, comment or memory-aids in the subjects' own words. It is essential to avoid having the subjects copy fixed material, and to give rise to varied vocabulary, syntax, spelling, and document length.

2. Recruit ~1000 Arab-literate subjects according to specified demographics, possibly in different countries, and request each to respond to one or more forms. In U.S. university settings, this would require permission for the cognizant institutional review board on human factors experimentation, and obtaining signed consent forms.

3. Provide each writer a writing surface, paper and writing instrument according to the randomized experimental design.

4. Collect the forms, possibly with the demographic data attached to each sheet. Scan the image data and organize the metadata in a widely readable format (ASCII perhaps). The size of the database after digitization at 200 dpi (8-bit depth) should be about 16 GB. Because of the prevalence of white space, after compression it may fit on a single CD.

For DIA and OCR research and development, this method would far superior to the customary collection of copies from prescribed forms. The latter, however, would be less expensive because the handwritten data need not be transcribed but only proofread to catch copying errors.

Part of the resulting database should be sequestered for independent testing. The remainder could be released to the research and development community. Used in this manner, the proposed database would have a relatively long shelf-life and provide performance measures far more representative of operational systems than the customary greenhouse test data.

7 Examples of Interactive Document Image Analysis

We mention some DIA tasks where automated algorithms work accurately only on exceptionally clean documents, but where a little interaction can quickly produce acceptable results.

Most OCR algorithms, especially for handwriting, are designed for binarized images, because scripts generally avoid discrimination based only on shades of gray or color. Instead of using thresholding hardware built into the scanner, today documents are usually digitized to 8-bit gray scale or RGB, and subsequently converted to binary images. Global binarization algorithms work only if the foreground and background reflectance are uniform throughout the document, which may not be the case if, for example, part of a folded documents suffers prolonged exposure to sunlight, or if there are dark areas around the edges of a photocopy. Local binarization algorithms measure the distribution of reflectance in a window translated through the page, and set the local threshold between foreground and background reflectance peaks according to the estimated dispersions of the components of the mixture distribution. The window size and reflectance distribution estimates invariably depend on explicit or implicit assumptions about the relative density and configuration (strokes) of the foreground (ink) and background. These assumptions generally hold only for a narrow class of documents. Fortunately, the binarization algorithms are simple, therefore an operator can easy set the appropriate window size and foreground density either for the whole document, or for selected areas. This still allows local algorithmic thresholding, and therefore requires much less

interaction than setting the threshold manually everywhere, and is much more robust than fully automated local thresholding.

Line finding is another instance where interaction may be effective. The first step is usually estimating global document skew, i.e., the angle of the written lines with respect to the paper or digitizer axes. While very accurate skew estimation and correction algorithms have been developed for printed matter, they do not work well on handwriting because the orientation of individual lines varies, the margins are not straight, there may be only a few words on a page, and there may be several columns of words or phrases at different angles. Humans can, however, judge skew remarkably well, and convey this information to the computer by a few well chosen stylus taps or by rotating a superimposed grid. After the computer-proposed skew correction and line finding is corrected, the occasional merged pair of lines – due to overlapping ascenders and descenders – can be likewise rapidly separated.

Word segmentation is relatively easy for printed text, except for extremely tightly-set, micro-justified print. In handwriting, however, large spaces often appear within words and, towards the end of a line, words are often squeezed together. In Arabic and other scripts, some inter-letter spaces are mandatory. Underlines that link word sequences can further complicate the task. Again, humans can usually spot missed word boundaries even in unfamiliar languages and scripts. If the writing lines are already properly segmented, then a simple interface can be designed to correct linked and broken words.

At the character recognition level, there are also several opportunities for effective interaction. First, humans can often tell where perfect accuracy is important, as in telephone numbers, email addresses, and proper nouns. If the automated algorithms fails on important words, phrases or numbers, they can be either entered manually using the virtual keyboard, or selected by a stylus tap from the top recognition candidates.

The human can provide global assistance to the character recognition system. The operator may be able to recognize the language or script of a document even from a few words, perhaps by using common sense or contextual information related to the source of the document (e.g., indicate that the document probably contains a mixture of Korean and German.) He or she can indicate the average slant, and in Western scripts, the prevalent case (e.g., if a writer uses only capital letters). The operator may also decide which of the available lexicons would provide the best language model. (The lexicons will be automatically updated with entries from the processed documents that have been deemed correct.)

Most importantly, entering only part of a document may provide enough training data – to a recognition system designed with this in mind – for fine-tuning the classification algorithms. The underlying assumption is that if the remainder of the document (and perhaps also additional documents) is from the same source, the adjusted parameters will yield more accurate recognition. If that is not the case, the operator can easily separate the portions of the document written by different individuals (assuming that the second writer was not attempting to mimic the first). Of course, human handwritten data entry in an unfamiliar language and script is also problematic. However, it requires far less training than learning to speak, write and understand a language. Off-shore data-entry is sometimes carried out by operators who do not know English. Anecdotal evidence suggests that on printed matter at least, non-speakers are more accurate, because they must look at each letter.

8 Evaluation Plans and Issues

The requirements for evaluating a CAVIAR-like approach to document analysis differ from traditional, fully-automated techniques. In the latter case, the focus is entirely on classification accuracy, whereas in the former, the system is comprised of a human user and a machine working in collaboration: both time and accuracy are important measures. It is unlikely the system will be more accurate than a human working alone, it must be faster. Similarly, it is unlikely the system will be faster than a fully-automated algorithm, it must be more accurate. Note also that such work naturally demands the use of human test subjects, unlike most pattern recognition research.

One of the most intriguing aspects of the earlier CAVIAR work was its demonstration of an inherent potential to incorporate adaptation, yielding a system that improves with use. This also provides the opportunity to study human learning: the operator and the system learn together, a little like a blind person and her Seeing Eye dog. As was done in the case of CAVIAR for flowers, we plan to study this effect.

It is also evident that one of the most plausible targets for a CAVIAR-like system involves documents written in a language that the user is unable to read, perhaps using a script that he/she does not recognize. A typical end-to-end scenario, then, might including the following processing stages:

1. Categorization: source, age, type of document, size, quality, language, script.

2. Image-level preprocessing including various levels of segmentation.

3. Word recognition.

4. Translation.

5. Information importance assessment/transformation into actionable knowledge.

Our initial experiments will be conducted on workstations with simulated mobile-device window sizes.

Among other expected benefits, we believe that the development cycle for interactive recognition systems may be faster than for wholly automated systems. One reason to suspect this will be true is that it is no longer necessary to take care of rare contingencies: they can be dealt with by the human operator, who can always override every automated option. This will allow deployment of a whole family of support systems, specialized to different levels of operator familiarity with the target language(s) and script(s).

We intend to develop the interfaces necessary to accomplish the above interactive tasks in parallel with data collection. While this would have been a monumental endeavor only a few years ago, today excellent tools are available for the purpose. Perhaps the most difficult part of the project is devising and incorporating a customized logging system for timing and recording every interaction and its effect. This must be done in sufficient detail to allow replaying the whole interactive document information acquisition sequence in order to find out where and what goes wrong.

The individual files log must be compiled and aggregated for statistical analysis. This will also form the basis for evaluating the interactive system and for comparing it with entirely automated and entirely manual data entry.

We emphasize that we do not address the extraction of data from a large backlog of accumulated documents. Unlike much of the academic research on document analysis, our interest here is not in archival documents, but in those whose value decreases exponentially with time. We propose a system that may prevent future backlogs of nearly worthless documents by on-site, just-in-time information extraction.

References

[1] M. Ancona, S. Locati, M. Mancini, A. Romagnoli, and G. Quercini. Comfortable textual data entry for PocketPC: the WTX system. In *Advances in Graphonomics, Proceedings of IGS 2005*, Salerno, Italy, June 2005.

[2] H. S. Baird and D. P. Lopresti, editors. *Human Interactive Proofs: Second International Workshop*, volume 3517 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Germany, 2005.

[3] S-H. Cha, A. Evans, A. Gattani, G. Nagy, J. Sikorski, C. Tappert, P. Thomas, and J. Zou. Computer Assisted Visual Interactive Recognition (CAVIAR) technology. In *Proceedings of the IEEE Electro/Information Technology Conference*, May 2005. http://www.ecse.rpi.edu/homepages/nagy/PDF_files/EIT2005_Evans_etal.pdf.

[4] J. Edwards. New interfaces: Making computers more accessible. *IEEE Computer*, pages 12–14, December 1997.

[5] W. K. English, D. C. Engelbart, and M. L. Berman. Display-selection techniques for text manipulation. *IEEE Transactions on Human Factors in Electronics*, HFE-8(1):5–15, March 1967.

[6] A. Evans, J. Sikorski, P. Thomas, J. Zou, G. Nagy, S.-H. Cha, and C. C. Tappert. Interactive visual system. Technical Report CSIS No. 196, Pace University, 2003. http://csis.pace.edu/~ctappert/ivs/ivs-nagyetal.pdf.

[7] F. Fisher. Digital camera for document acquisition. In *Symposium on Document Image Understanding Technology*, Columbia, MD, 2001. http://www.dtic.mil/matris/sbir/sbir022/a038.pdf.

[8] H. Fujisawa, H. Sako, Y. Okada, , and S. Lee. Information capturing camera and developmental issues. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition (ICDAR'99)*, pages 205–208, Bangalore, India, September 1999.

[9] T. Hedgpeth, M. Rush, J. Black, and S. Panchanathan. The iCare project reader. In *Sixth International ACM SIGACCESS Conference on Computers and Accessibility*, October 2004. http://cubic.asu.edu/grants/downloads/RDE/iCareReader.doc.

[10] M. Holland and C. Schlesiger. High-mobility machine translation for a battlefield environment. In *Proceedings of NATO/RTO Systems Concepts and Integration Symposium*, volume 15, pages 1–3, Monterey, CA, May 1998.

[11] C. Jacobs and P. Simard. Low resolution camera based OCR. *International Journal on Document Analysis and Recognition*. To appear.

[12] C. L. James and K. M. Reischel. Text input for mobile devices: Comparing model prediction to actual performance. In *Proceedings of the ACM Conference on Computer-Human Interaction*, pages 365–371, 2001.

[13] D. J. Langendorf. Textware solution's Fitaly keyboard v1.0 easing the burden of keyboard input. *WinCELair Review*, February 1998.

[14] S. Mackenzie and W. Soukoreff. Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction*, 17:147–198, 2002.

[15] T. Masui. An efficient text input method for pen-based computers. In *Proceedings of the ACM Conference on Computer-Human Interaction*, pages 328–335, 1998.

[16] G. Miller. The magical number seven plus or minus two; some limits on our capacity for processing information. *Psychological Review*, 63:81–97, 1956.

[17] G. Nagy, L. Li, A. Samal, S. Seth, and Y. Xu. Cooperative text and line-art extraction from a topographic map. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition (ICDAR'99)*, pages 467–470, Bangalore, India, September 1999.

[18] G. Nagy, L. Li, A. Samal, S. Seth, and Y. Xu. Integrated text and line-art extraction from a topographic map. *International Journal on Document Analysis and Recognition*, 2(4):177–185, June 2000.

[19] J. P. Peters, C. Thillou, and S. Ferreira. Embedded reading device for blind people: a user-centred design. In *IEEE Emerging Technologies and Applications for Imagery Pattern Recognition (AIPR 2004)*, pages 217–222, 2004.

[20] S. Srihari and Z. Shi. Forensic handwritten document retrieval system. In *First International Workshop on Document Image Analysis for Libraries (DIAL'04)*, pages 188–194, 2004.

[21] K. Swan. FALCon: Evaluation of OCR and machine translation paradigms, August 1999. http://www.arl.army.mil/seap/reports/kreport.pdf.

[22] J. Yang, X. Chen, J. Zhang, Y. Zhang, and A. Waibel. Automatic detection and translation of text from natural scenes. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '02)*, May 2002.

[23] J. Zou. *Computer Assisted Visual InterActive Recognition*. PhD thesis, Rensselaer Polytechnic Institute, 2004.

[24] J. Zou and G. Nagy. Evaluation of model-based interactive flower recognition. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 2, pages 311–314, 2004.

Arabic Document Analysis

Handwritten Arabic Word Spotting using the CEDARABIC Document Analysis System

Sargur Srihari, Harish Srinivasan, Pavithra Babu and Chetan Bhole
Center of Excellence for Document Analysis and Recognition (CEDAR)
University at Buffalo, State University of New York
Amherst, New York 14228
srihari @cedar.buffalo.edu

Abstract

An algorithm and a system for searching handwritten Arabic documents to locate key words is presented. Three main components of the system are a word segmenter, a shape based matcher for words and a search interface. The user types in a query in English within a search window, the system finds the equivalent Arabic word, e.g., by dictionary look-up, locates word images in an indexed (segmented) set of documents. A two-step approach is employed in performing the search: (1) prototype selection: the query is used to obtain a set of handwritten samples of that word from a known set of writers (these are the prototypes), and (2) word matching: the prototypes are used to spot each occurrence of those words in the indexed document database. A ranking is performed on the entire set of test word images– where the ranking criterion is a similarity score between each prototype word and the candidate words based on global word shape features. A database of 20,000 word images contained in 100 scanned handwritten Arabic documents written by 10 different writers was used to study retrieval performance. Using five writers for providing prototypes and the other five for testing, using manually segmented documents, 55% precision is obtained at 50% recall. Performance increases as more writers are used for training.

1 Introduction

Spotting handwritten words in documents written in the Latin alphabet has received considerable attention [1],[2], [3]. A systematic comparison of seven methods was made in [4] who showed that the highest precision was obtained by their method based on profiles and dynamic time warping (DTW). A method based on word shape was shown to perform better than the method based on DTW, both in terms of efficiency and effectiveness, in [5]. This paper discusses the performance of the word shape method presented in [5] when applied to handwritten Arabic documents.

The word spotting algorithm is implemented as a tool within an interactive software system known as CEDARABIC which is designed for the analysis of handwritten Arabic documents. Most functionalities of CEDARABIC are the same as that found in a predecessor system known as CEDAR-FOX which was developed for forensic examination by U.S. law enforcement [6]. Being a software branch of the CEDAR-FOX system CEDARABIC also has capabilities for image processing operations such as enhancement, thresholding, underline removal, etc., as well as for writer verification, writer identification and signature verification. The interfaces have been modified to be appropriate for analyzing Arabic language documents by an English-speaking user. The system includes a GUI for various functions including scanning, selecting a region of interest, feature computation, ground truthing(manual segmentation and entering word truth), searching of a word based on its English equivalent, etc. The primary focus of this paper is on the word-spotting tool of CEDARABIC–whose goal is to find each occurrence of a user-specified query word in a given set of documents.

In the CEDARABIC system the task of word spotting has two phases: indexing and search. In the indexing phase each document image in the database is indexed by segmenting into its component word images. The search phase has two parts: in the first part the typed in query is used to generate a set of query images corresponding to handwritten versions of it. In the second part each query image is matched against each indexed image to find the closest match.

The paper consists of the following sections: (i) scanned document collection used to design the system and and test its performance, (ii) user interface of CEDARABIC, (iii) word segmentation algorithm, (iv) word-shape matching algorithm, (v) conclusion and acknowledgment.

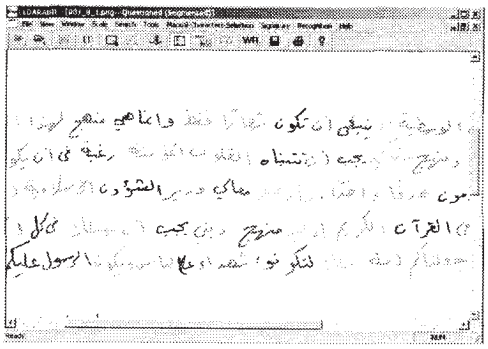


Figure 1: Sample ".arb" file opened with CEDARABIC. Each segmented word is colored differently from adjacent words.

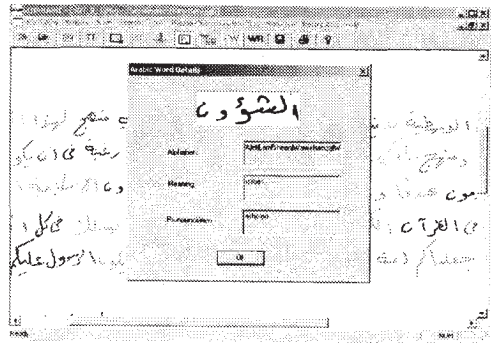


Figure 2: Truth for a particular word. This window pops up when a particular word image is right-clicked. The truth consists of three parts: the individual characters, its meaning in English and its Arabic pronunciation shown in Engish.

2 Document Image Database

A document collection was prepared with 10 different writers, each contributing 10 different full page documents in handwritten Arabic. Each document comprises of approximately 150 – 200 words each, with a total of 20,000 word images in the entire database. The documents were scanned at a resolution of 300 dots per inch. This is the resolution to be used for optimal performance of the system. Figure 1 shows a sample scanned handwritten Arabic document written by writer 1.

For each of the 10 documents that were handwritten, a complete set of truth comprising of the alphabet sequence, meaning and the pronunciation of every word in that document was also given. The scanned handwritten document comprising of word images were mapped with the corresponding truth information given, and saved as a separate file with a ".arb" extension. The naming convention for the file is explained with this example. File 007_9_1.*arb* is the ninth document written by writer number 7. When some writers use more than one full page to transcribe a document, the last digit 1 or 2 identifies the page of the document. These ".arb" files have all the necessary information in them to perform word spotting using the tool that is described in the following section. The preparation of these ".arb" files involved careful manual word segmentation of each scanned document. The software tool described below also provides for a way to manually segment each word from a raw unprocessed document.

3 User Interface

Most of the functionalities of the CEDARABIC system, including word spotting, are the same as that found in a system for Latin script, known as CEDAR-FOX, which was developed for forensic examination by U.S. law enforcement [7] [8]. Being a modification of the CEDAR-FOX system for the analysis of handwritten documents in the Latin alphabet, CEDAR-FOX also has capabilities for image processing operations such as enhancement, thresholding, underline removal, etc., as well as for writer verification, writer identification and signature verification. The interfaces have been modified to be appropriate for analyzing Arabic language documents by an English-speaking user.

CEDARABIC has a GUI for analyzing and using the prepared ".arb" for performing word spotting and other information lookup tasks. A particular ".arb" file can be opened by selecting the "Q" or "K" option on the tool bar and Figure 1 shows a screen shot of CEDARABIC when a particular file is opened. By clicking on the tool bar icon for "Show Words", the different words get marked with different colors thereby differentiating one Arabic word image from another.

Right-clicking on any word image pops up a dialog- box displaying the complete set of truths associated with that word. Figure 2 shows a screenshot when a particular word image was right-clicked. The Arabic alphabet sequence, the English equivalent meaning, and the pronunciation of each word can be viewed with this tool.

The query for word spotting is typed as English text– one that corresponds to the English equivalent meaning of the word to be searched for.

The first step in word spotting is to identify prototypes (template word images) corresponding to the Arabic word for the queried text. These template word images are obtained from documents which are set aside as training documents. For example, we can say documents pertaining to writers 1 through 5 can be used solely for the purpose of training and they provide for the template word images. The default set consists of writers 1 through 5. These training ".arb" should contain the complete truth for each word image, so as to pick up those that match the query text. The word images for those that matched serve as template word images to per-

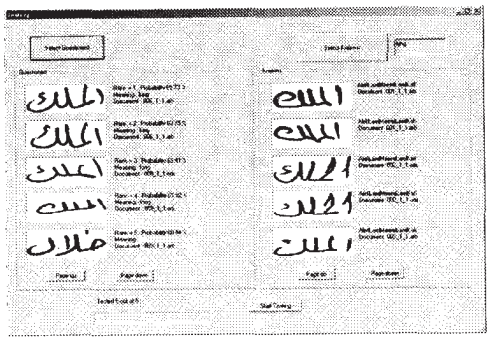


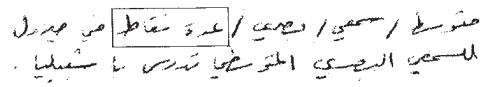
Figure 3: Word spotting results. The template word images on the right side, obtained in the first step, are used to spot the images shown on the left. The results are sorted in rank by probability. The images on the right are the selected prototypes and those on the left are the ranked set of test images.

form word spotting in other documents that do not have the truth. The matching words are automatically selected and the user is also given an option of selecting the ones that need to be used as template word images. Once this is done, upon clicking "Use Selected for Training" these images will be used to spot similar occurrences for words in other documents purely based on the shape of the word. The template word images serve as a knowledge base to learn the writing styles of the particular word and then use this to identify other similar words purely based on matching the image. A set of test documents can now be specified, and the tool will look for the occurrence of the particular word in every document specified and rank the retrieved words in the order of probability of match. Figure 3 shows the result of one such word spotting task performed. In this example the search is for the word "king" whose Arabic equivalent is pronounced as "al malik". The images on the right are the template word images and those on the left are the word images in order by rank.

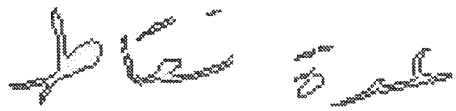
4 Word Segmentation

Automatic segmentation into words is based on taking several features on either side of a potential segmentation point and using a neural network for deciding whether or not the segmentation is between two distinct words. Some of the differences between the tasks of segmenting Arabic script and segmenting Latin script are: presence of multiple dots above and below the main body in Arabic and the absence of upper case letters at the beginning of sentences in Arabic.

The process of word segmentation begins with obtaining the set of connected components for each line in the document image. Figure 4 shows a region of a document image the connected components (exterior and interior contours) of a small section. The



(a) Region of a document image

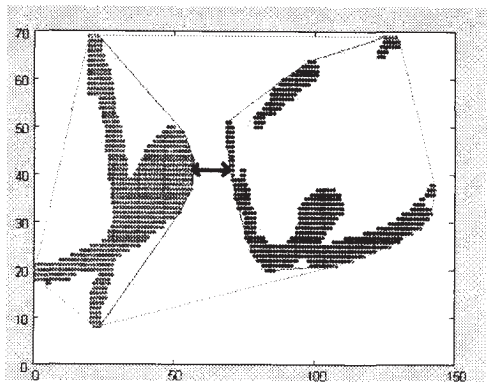


(b) Connected component exterior and interior contours

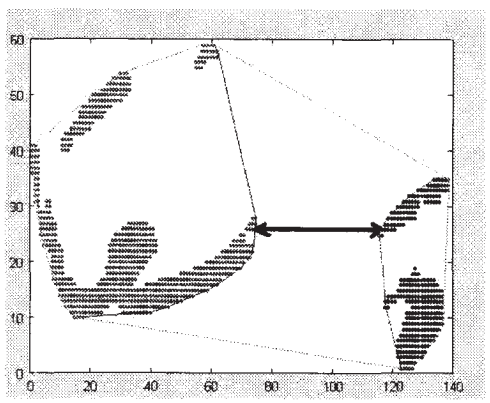
Figure 4: Connected components of a small section of the image.

interior contours or loops in a component are ignored for the purpose of word segmentation as they provide no information for this purpose. The connected components are grouped into clusters, by merging minor components such as dots above and below a major component. Also particular to Arabic, most words starts with the Arabic character "Alef". The presence of an "Alef" is a strong indicator that there may be a word gap between the pair of clusters. The height and width of the component are two parameters used to check if the component is the character "Alef". Figure 6 shows samples of the Arabic character Alef. Every pair of adjacent clusters are candidates for word gaps. 9 features are extracted for these pairs of clusters and a neural network is used to determine if the gap between the pair is a word gap. The 9 features are: width of the first cluster, width of second cluster, difference between the bounding box of the two clusters, flag set to 1 or 0 depending on the presence or absence of the Arabic character "Alef" in the first cluster, the same flag for the second cluster, number of components in the first cluster, number of components in the second cluster, minimum distance between the convex hulls enclosing the two clusters and the ratio between, the sum of the areas enclosed by the convex hulls of the individual clusters, to the total area inside the convex hull enclosing the clusters together. Figure 5 shows two pairs of adjacent clusters tested for word gaps.

A neural network was trained using these 9 features with feature vector labeled as to whether it is a word gap or not. This is similar to the neural network approach used for English postal addresses [9].



(a) Not word gap



(b) Word gap

Figure 5: Two pairs of adjacent clusters. Dotted lines indicate convex hulls around the individual clusters. Solid lines indicate convex hull around the two clusters taken together. (a) The gap between the convex hulls is truly not a word gap, and (b) The gap between the convex hulls is truly a word gap.

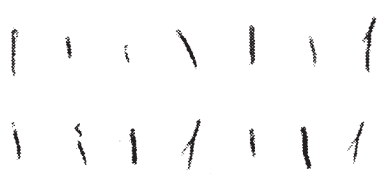


Figure 6: Samples of Arabic character "Alef". The hight and width are two parameters that are used to detect the presence of "Alef" in the clusters.

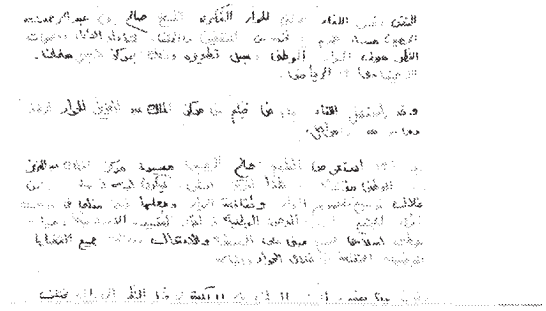


Figure 7: A document segmented into words by the neural network. Figure 8(a) 8(b) 8(c) and 8(d) show the first four clusters of the document for the first line from the left and the classification result.

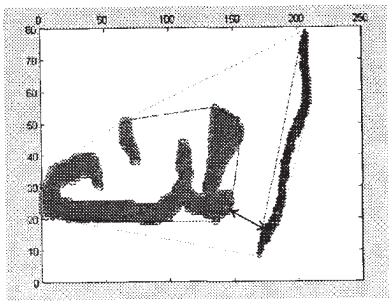
Table 1: NN Features

Fig	f1	f2	f3	f4	f5	f6	f7	f8	f9	NN
8(a)	156	42	12	0	0	2	1	24.59	0.58	-0.75
8(b)	42	13	7	0	1	1	1	8.00	0.52	-0.99
8(c)	13	84	14	1	0	1	2	24.75	0.40	0.91
8(d)	84	82	51	0	0	2	2	51.00	0.56	0.80

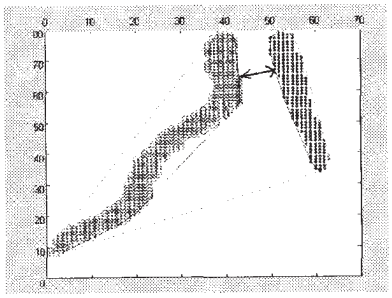
However the features are different. Figure 7 shows a document segmented into words by the neural network. Figure 8 shows the first 4 pairs of clusters whose features are extracted and fed to the neural network. These 4 pairs of clusters correspond to the words starting from the left on the first line of the document in figure 7. Table 1 shows the 9 features for these pairs respectively and the last column shows the output of the neural network and table 2 shows what each feature is. A positive output from the neural network indicates a presence of word gap and a negative output indicates the absence of a word gap. From the table 1, it is evident that though the distance between the clusters for pair 1 and pair 3 are almost the same, the neural network gives the correct classification. This is due to presence of "Alef" that strongly indicates the presence of a word gap.

Table 2: Features

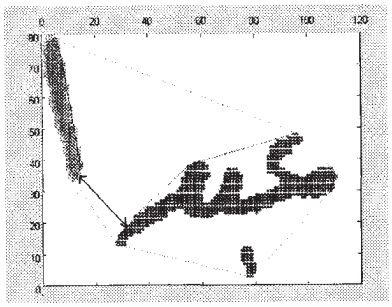
Number	Feature
1	Width of 1st cluster
2	Width of 2nd cluster
3	Bounding box gap between clusters
4	Alef flag for 1st cluster
5	Alef flag for 2nd cluster
6	No of components 1st cluster
7	No of components 2nd cluster
8	Min. Convex hull distance
9	Ratio of sum of cluster areas to total area



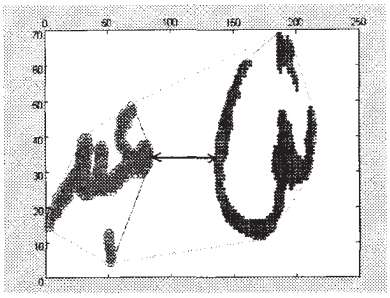
(a) Not word gap



(b) Not word gap



(c) Word gap



(d) Word gap

Figure 8: Four pairs of adjacent clusters. Note that the distance between the clusters in (a) and (c) are the same (the scale in the plot are different). The neural network still classifies (c) as a word gap due to the presence of the "Alef".

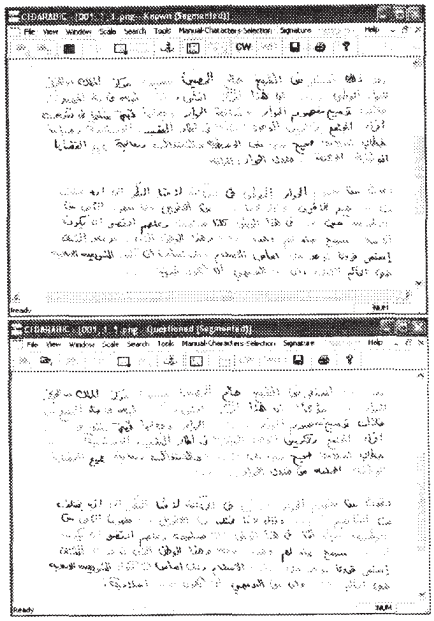


Figure 9: Example of a reasonably well-segmented Arabic document by automatic segmenter designed for English. The truth is shown on the right screen-shot and the result of the automatic word segmenter is shown on the left. In this case 86% of the automatically segmented words are correct.

4.1 Word Segmentation Performance

An example of a reasonably well-segmented Arabic image by the automatic segmenter, together with the perfectly segmented image (truth), are shown in Figure 9. There are 130 words in the truth shown on the right-hand side. The result of the automatic segmenter, shown on the left, has the following error counts: (i) 9 words have been split into two parts (9 errors), (ii) 3 adjacent pairs of words have been merged (6 errors) and (iii) one set of three words have been merged (3 errors). Since there are a total of 18 errors we can say that 86% of the word segmentations are correct.

An example of a poorly segmented image together with its truth are shown in Figure 10. In this case there are 120 words in the truth shown on the right. the error counts of the automatically segmented document on the left are as follows: (i) 17 words have been split (17 errors), (ii) 18 adjacent words have been merged (36 errors), (iii)3 merges of three adjacent words (9 errors), and (iv) one merge of four adjacent words (4 errors). Since there are 66 errors we can say that 45% of the word segmentations are correct.

Overall performance over ten writers writing 10 documents each, when the lines are segmented correctly, is about 60% using a set of seven segmentation features. A more complex set of features is expected to yield a higher level of performance.