1. **Character Recognition Systems Overview**

Character recognition systems differ widely in how they acquire their input (on-line versus off-line), the mode of writing (handwritten versus machine printed), the connectivity of text (isolated characters versus cursive words), and the restriction on the fonts (single font versus Omni-font) they can recognize. The different capabilities of character recognition are illustrated in Figure (1).

In this report, we are going to use the terms “OCR”, “ICR” and “NHR” for printed character recognition, offline handwritten recognition and natural handwriting recognition online, respectively.



Character Recognition

Off-Line

On-Line

(Handwritten)

Machine Printed

Handwritten

Single Font

Omni Font

Isolated Characters

Cursive Words

Isolated Characters

Cursive Words

**Figure (1):** Character recognition capabilities

**1.1. On-Line (Real-Time) Systems**

These systems recognize text while the user is writing with an on-line writing device, capturing the temporal or dynamic information of the writing. This information includes the number, duration, and order of each stroke (a stroke is the writing from pen down to pen up). Online devices are stylus based, and they include tablet displays, and digitizing tablets. The writing here is represented as a one-dimensional, ordered vector of (x, y) points. On-line systems are limited to recognizing handwritten text. Some systems recognize isolated characters, while others recognize cursive words. We are going to use the new term “Natural Handwriting Recognition” (NHR) for this technology.

**1.2. Off-Line Systems**

These systems recognize text that has been previously written or printed on a page and then optically converted into a bit image. Offline devices include optical scanners of the flatbed, paper fed and handheld types. Here, a page of text is represented as a two-dimensional array of pixel values. Off-line systems do not have access to the time-dependent information captured in on-line systems. Therefore offline character recognition is considered as a more challenging task than its online counterpart.

The word optical was earlier used to distinguish an optical recognizer from systems which recognize characters that were printed using special magnetic ink. In the case of a print image, this is referred to as Optical Character Recognition (OCR). In the case of handprint, it is referred to as Intelligent Character Recognition (ICR).

Over the last few years the decreasing price of laser printers has made computer users able to readily create multi-font documents. The number of fonts in typical usage has increased accordingly. However the researcher experimenting on OCR is unhappy to perform the vastly time-consuming experiments involved in training and testing a classifier on potentially hundreds of fonts in a number of text sizes and in a wide range of image noise conditions; even if such an image data set already existed. Collecting such a database could involve considerably more effort.

Although the amount of research into machine-print recognition appears to be tailing off as many research groups turn their attention to handwriting recognition, it is suggested that there are still significant challenges in the machine-print domain. One of these challenges is to deal effectively with noisy, multi-font data, including possibly hundreds of fonts.

The sophistication of the off-line OCR system depends on the type and number of fonts to be recognized. An Omni-font OCR machine can recognize most non stylized fonts without having to maintain huge databases of specific font information. Usually Omni-font technology is characterized by the use of feature extraction. Although Omni-font is the common term for these OCR systems, this should not be understood literally as the system being able to recognize all existing fonts. No OCR machine performs equally well or even usably well, on all the fonts used by modern computers.

**2. Offline Character Recognition Technology Applications**

The intensive research effort in the field of Character Recognition was not only because of its challenge on simulation of human reading but also because it provides widespread efficient applications. Three factors motivate the vast range of applications of off-line text recognition. The first two are the easy use of electronic media and its growth at the expense of conventional media. The third is the necessity of converting the data from the conventional media into the new electronic media.

OCR and ICR technologies have many practical applications which include the following, as examples, but not limited to:

* Digitization, storing, retrieving and indexing huge amount of electronic data as a results of the resurgence of the World Wide Web. The text produced by OCRing text images can be used for all kinds of Information Retrieval (IR) and Knowledge Management (KM) systems which are not so sensitive to the inevitable Word Error Rate (WER) of whatever OCR system as long as this WER is kept lower than 10% to 15%.
* Office automation for providing an improved office environment and ultimately reach an ideal paperless office environment.
* Business applications as automatic processing of checks
* Automatic address reading for mail sorting
* Automatic passport readers
* Use of the photo sensor as a reading aid and transfer of the recognition result into sound output or tactile symbols through stimulators.
* Digital bar code reading and signature verification
* Front end components for Blind reading Machines
* Machine processing of forms
* Automatic mail sorting (ICR)
* Processing of checks (ICR)
* Credit Cards Applications (ICR)
* Mobile applications (OCR/ICR)
* Blind Reader (ICR)

**3. Arabic OCR Technology and state of the art:**

Since the mid-1940s researchers have carried out extensive work and published many papers on character recognition. Most of the published work on OCR has been on Latin characters, with work on Japanese and Chinese characters emerging in the mid-1960s. Although almost a billion of people worldwide, in several different languages, use Arabic characters for writing (alongside Arabic, Persian and Urdu are the most noted examples), Arabic character recognition has not been researched as thoroughly as Latin, Japanese, or Chinese and it has almost only started in the 1970’s. This may be attributed to the following:

i) The lack of adequate support in terms of journals, books, conferences, and funding, and the lack of interaction between researchers in this field.

(ii) The lack of general supporting utilities like Arabic text databases, dictionaries, programming tools, and supporting staff.

(iii) The late start of Arabic text recognition.

(iv) The special challenges in the characteristics of the Arabic script as stated in the following section. These characteristics results in the fact that the techniques developed for other writings cannot be successfully applied to the Arabic writing: Different fonts, etc;

In order to be competent with the human capability at the digitization of printed text, font-written OCR’s should achieve an Omni-font performance at an average WER ≤ 3% and an average speed ≥ 60 words/min. per processing thread. While font-written OCR systems working on Latin script can claim approaching such measures under favorable conditions, the best systems working on other scripts, especially cursive scripts like Arabic, are still well behind due to a multitude of complexities [windows magazine 2007]. For example, the best reported ones among the few Arabic Omni font-written OCR systems can claim assimilation WER’s 3% and 10% generalization WER's under favorable conditions (good laser printed windows and Mac fonts) [Attia et al 2007, 2009], [El-Mahallawy 2008], [Rashwan et al 2007].

**4. Arabic OCR challenges**

The written form of Arabic language while written from right to left presents

many challenges to the OCR developer. The most challenging features of the Arabic

orthography are [Al-Badr 1995], [Attia 2004] :

**i) The connectivity challenge**

Whether handwritten or font written, Arabic text can only be scripted cursively; i.e. graphemes are connected to one another within the same word with this

connection interrupted at few certain characters or at the end of the word. This necessitates any Arabic OCR system to not only do the traditional grapheme recognition task but do another tougher grapheme segmentation one (see Figure 2) To make things even harder, both of these tasks are mutually dependent and must hence be done simultaneously.



**Figure (2):** Grapheme segmentation process illustrated by manually inserting

vertical lines at the appropriate grapheme connection points.

**ii) The dotting challenge**

Dotting is extensively used to differentiate characters sharing similar graphemes. According to Figure (3), where some example sets of dotting differentiated graphemes are shown, it is apparent that the differences between the members of the same set are small. Whether the dots are eliminated before the recognition process, or recognition features are extracted from the dotted script, dotting is a significant source of confusion – hence recognition errors – in Arabic font-written OCR systems especially when run on noisy documents; e.g. those produced by photocopiers.



**Figure (3):** Example sets of dotting-differentiated graphemes

**iii) The multiple grapheme cases challenge**

Due to the mandatory connectivity in Arabic orthography; the same grapheme representing the same character can have multiple variants according to its relative position within the Arabic word segment {Starting, Middle, Ending, Separate} as exemplified by the 4 variants of the Arabic character “ ع” shown in bold in Figure (4).



**Figure (4):** Grapheme “ ع” in its 4 positions; Starting, Middle, Ending & Separate

**iv) The ligatures challenge**

To make things even more complex, certain compounds of characters at certain positions of the Arabic word segments are represented by single atomic graphemes called ligatures. Ligatures are found in almost all the Arabic fonts, but their number depends on the involvement of the specific font in use. Traditional Arabic font for example contains around 220 graphemes, and another common less involved font (with fewer ligatures) like Simplified Arabic contains around 151 graphemes. Compare this to English where 40 or 50 graphemes are enough. A broader grapheme set means higher ambiguity for the same recognition methodology, and hence more confusion. Figure (5) illustrates some ligatures in the famous font “Traditional Arabic”.



**Figure (5):** Some ligatures in the Traditional Arabic font.

**iv) The overlapping challenge**

Characters in a word may overlap vertically even without touching as shown in Figure (6).



**Figure (6):** Some overlapped Characters in Demashq Arabic font.

**v) Size variation challenge**

Different Arabic graphemes do not have a fixed height or a fixed width. Moreover, neither the different nominal sizes of the same font scale linearly with their actual line heights, nor the different fonts with the same nominal size have a fixed line height.

**vi) The diacritics challenge**

Arabic diacritics are used in practice only when they help in resolving linguistic ambiguity of the text. The problem of diacritics with font written Arabic OCR is that their direction of flow is vertical while the main writing direction of the body Arabic text is horizontal from right to left. (See Figure (7)) Like dots; diacritics – when existent - are a source of confusion of font-written OCR systems especially when run on noisy documents, but due to their relatively larger size they are usually preprocessed.



**Figure (7):** Arabic text with diacritics.

**5. Current OCR/ICR Products**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Product** | **Type** | **License** | **Languages** | **Performance** | **Platform** | **Price** | **Notes** |
| Sakhr’s OCR Automatic Reader  (القارئ الالى) | OCR | commercial | -Arabic, English, French and 16 other languages. Farsi, Jawi, Dari, Pashto, Urdu (available optionally in extra language pack)  - Support bilingual documents(Arabic/English, Farsi/English and Arabic/French). | - 99% for high quality documents.  - 96% for low quality documents. | Windows |  |  |
| VERUS OCR  NovoDynamics | OCR | commercial | - Arabic, Farsi/Persian, Dari, Pashto English and French.  - Support bilingual documents. |  | Windows | 1295 $ |  |
| Readiris | OCR | commercial | - Latin based languages.  - Asian languages.  -Readiris (for middle east) support Arabic, Farsi and Hebrew. |  | -Windows , Mac OS. | - Readiris 12 (latin):  \*Pro : 129$  \* Corporate: 399$  - Readiris 12 (Asian) : \*Pro : 249$  \*Corporate : 499$  - Readiris 12 (middle east) : \*Pro : 249$  \*Corporate : 499$ | -Pro features: Standard scanning support and standard recognition features.  -Corporate features : volume scanning support and advanced recognition features. |
| **Product** | **Type** | **License** | **Languages** | **Performance** | **Platform** | **Price** | **Notes** |
| Kirtas’s KABIS III Book Imaging System: Employ **SAKHR** engine for Arabic | OCR | commercial | -English, French, Dutch, Arabic (Naskh & Kofi), Farsi, Jawi, Pashto, and Urdu.  - Support bilingual documents (Arabic/English), (Arabic/French), and (Farsi/English). |  | - Windows 2003 SERVER 64-bit |  | - SureTurn™ robotic arm uses vacuum system to gently pick up and turn one page at a time |
| Nuance OmniPage 17 | OCR | commercial | - English, Asian languages and other 120 languages.  - Doesn’t include Arabic.  - Support bilingual documents. | 99% character accuracy | -Windows  -OmniPage pro for Mac OS | - Professional 499 $  -Standard 149 $ |  |
| EDT WinOCR | OCR | commercial | - English, German, French, Spanish, Italian, Swedish, Danish, Finnish, Irish.  -Doesn’t support Arabic. | 99% accuracy | -Windows | 40 $ | Free trial is available |
| CuneiForm | OCR | Freeware | -Latin based languages.  - Support multilingual (Russian-English) |  | - Windows, Linux, Mac | Free |  |
| HOCR | OCR | General Public License | - Hebrew |  | Linux |  |  |
| Tesseract | OCR | Freeware | Can recognize 6 languages, is fully UTF8 capable, and is fully trainable |  | Windows and Mac |  |  |
| SimpleOCR | OCR | Freeware | English and French |  | Windows |  |  |
| ReadSoft | OCR | Commercial | European characters, simplified and traditional Chinese, Korean, Japanese characters |  | Windows |  |  |
| Microsoft office document Imaging | OCR | commercial | Language availability is tied to the installed proofing tools. |  | Windows |  | Uses ScanSoft OCR engine |
| **Product** | **Type** | **License** | **Languages** | **Performance** | **Platform** | **Price** | **Notes** |
| ABBYY FineReader | OCR/ICR | commercial | -More than186 languages.  - Support Arabic numbers  -Plans to support Arabic. | 99% accuracy | -Windows , Mas OS | 400 $ | -Dictionary for some languages  -Free trial is available |
| ExperVision TypeReader & OpenRTK | OCR/ICR | commercial | - Latin and Asian based languages  -Doesn’t support Arabic |  | Windows, Mac, Unix,Linux |  |  |
| Accusoft SmartZone | OCR/ICR | commercial | - For OCR: English, Danish, Dutch, Finnish, French, German, Italian, Norwegian, Portuguese, Spanish, and Swedish.  -For ICR: only English.  - doesn’t support Arabic |  | -Windows | - ICR/OCR Standard: 1999$  -ICR/OCR Professional: 2999$  - OCR standard : 999$  - OCR Professional: 1999$ | -Professional edition : Full speed  -Standard edition : Limited to 20% of Professional Speed  - Free trial is available |
| IRISCapture Pro | ICR | commercial | Latin based languages |  | Windows |  |  |
| A2IA | ICR |  | English, French, German, Italian, Portuguese and Spanish |  | Windows |  |  |
| LEADTOOLS ICR SDK Module | ICR |  | -Catalan, Czech, Danish, Dutch, English, Finnish, French, German, Hungarian, Italian, Norwegian, Polish, Portuguese, Spanish, Swedish |  | Window |  |  |

**6. Databases:**

**6.1 AHDB (Arabic Handwritten Database)**

**Database Form Design [Somaya et al 2002]:**

* + Each form contains 5 pages
  + The first 3 pages were filled with 96 words, 67 of which are handwritten words corresponding to numbers that can be used in handwritten cheque writing. The other 29 words are from the most popular words in Arabic writing (ان,من,فى,هذا….etc)
  + The 4th page contain 3 sentences of handwritten words representing numbers and quantities that can be written on cheques
  + The fifth page is lined, and it is completed by the writer in freehand on any subject of their choice
  + The color of the forms is light blue and the foreground black ink
  + The DB contains 105 form
  + The DB is available publically.

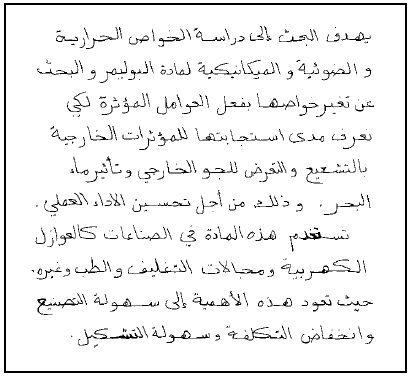


Figure (8) An example of free handwriting

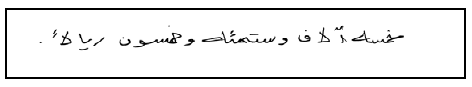


Figure (9) An Example of sentences contained in cheques

**6.2 Arabic Characters Data Corpus**

**Database Form Design: [**Huda Alamri et al 2008]

* The form consists of 7 × 7 small rectangles; one character inside each rectangle
* The DB includes 15800 character written by more than 500 writers



Figure (10) A4 sized form used to collect character samples

**6.3 A Novel Comprehensive Database for Arabic Off-Line Handwriting Recognition**

**Database Form Design:** [A. Asiri et al 2005]

* + It consists of 2 pages
  + The first page includes: a sample of an Arabic date, 20 isolated digits as 2 samples of each, 38 numerical strings with different lengths, one 35 isolated letters as one sample of each and the first 14 words of an Arabic word dataset
  + The second page includes the rest of the candidate words
* The forms were filled by 328 writers
* The database will be made available in the future for research purposes from the Centre for Pattern Recognition and Machine Intelligence (CENPARMI), at Concordia University.

Figure (11) Sample of the filled form

**6.4 DATABASES FOR RECOGNITION OF HANDWRITTEN ARABIC CHEQUES [**Yousef Al-Ohali 2000**]**

* The database was collected in collaboration with Al Rajhi Bank , Saudi Arabia
* It consists of 7000 real world grey-level cheque images(all personal information including names, account numbers, and signatures were removed)
* The DB is available after the approval of Al Rajhi bank.
* **The database is divided into 4 parts:**
* Arabic legal-amounts database (1,547 legal amounts)
* Courtesy amounts database (1,547 courtesy amounts written in Indian digits)
* Arabic sub-words database (23,325 sub-words)
* Indian digits database (9,865).

Figure (12) A sample of the Figure (13) segmented legal amount

Arabic Cheque database

**6.5 Handwritten Arabic Dataset Arabic-Handwriting-1.0** [Applied Media Analysis 200]

* 200 unique documents
* 5000 handwritten pages
* A wide variety of document types: diagrams, memos, forms, lists (including Indic and English digits), poems
* Documents produced by various writing utensils: pencil, thick marker, thin marker, fine point pen, ball point pen, black and colored
* Available in binary and grayscale
* Price : $500 for academic use and $1500 for standard use.

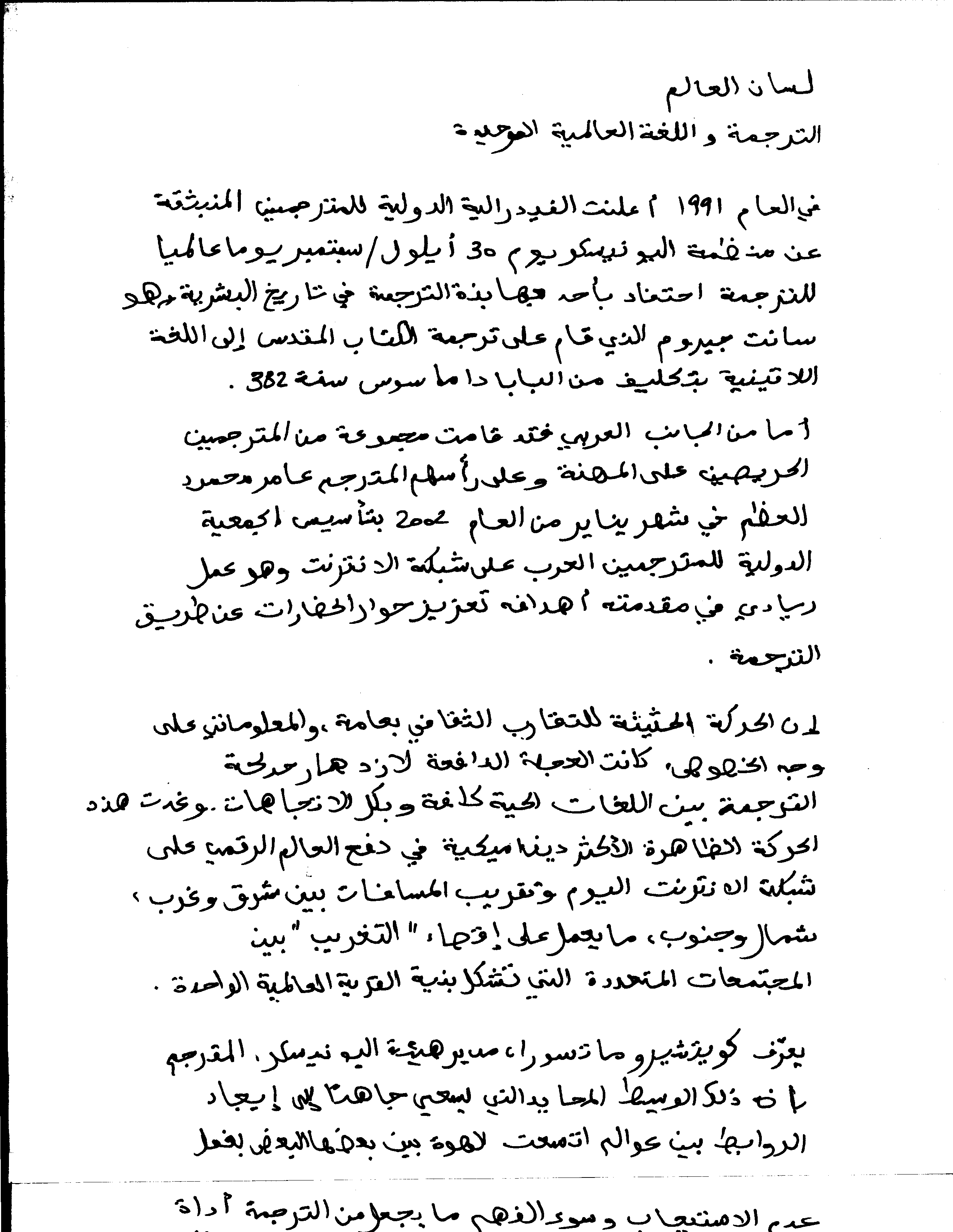


Figure (14) A sample from the Media Analysis Database

**6.6 IFN/ENIT-Database**

* Consists of 32492 Arabic words handwritten by more than 1000 different writers
* Written are 937 Tunisian town/village names. Each writer filled one to five forms with pre-selected town/village names and the corresponding post code.
* The DB is available free of charge for non-commercial use.

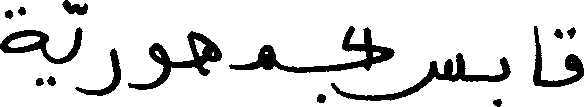
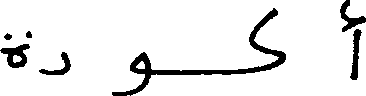
 

Figure (15) Samples from the IFN/ENIT DB

**6.7 (MADCAT) by LDC**

* It consists of the following: [Stephanie M. Strassel 2009]
  + The AMA Arabic Dataset developed by Applied Media Analysis (AMA 2007) which consists of 5000 handwritten pages, derived from a unique set of 200 Arabic documents transcribed by 49 different writers from six different origins.
  + The LDC acquired 3000 pages of handwritten Arabic images collected by Sakhr. Sakhr's corpus consists of 15 Arabic newswire documents each transcribed by 200 unique writers. LDC added line and word level ground truth annotations to each handwritten image, and distributed these along with English translations for each document to MADCAT performers.
* Beyond existing corpora, MADCAT performers requested additional new training data totaling at least 10,000 handwritten pages in the first year and 20,000 pages in the second year of the program, plus ground truth annotations for each page.
* Writing conditions for the collection as a whole are established as follows: Implement: 90% ballpoint pen, 10% pencil; Paper: 75% unlined white paper, 25% lined paper; Writing speed: 90% normal, 5% fast, 5% careful.
* The DB is not published yet.

** **

Figure (16) Processed document for assignment Figure (17) Handwritten version

**6.8 The DARPA Arabic OCR Corpus**

The DARPA Arabic OCR Corpus consists of 345 pages of Arabic text (~670k characters) scanned at 600 dots per inch from a variety of sources of varying quality, including books, magazines, newspapers, and four computer fonts. Associated with each image in the corpus is the text transcription, indicating the sequence of characters on each line. But the location of the lines and the location of the characters within each line are not provided. The corpus includes several fonts, for example: Giza, Baghdad, Kufi, and Nadim. The corpus transcription contains 89 unique characters, including punctuation and special symbols. However, the shapes of Arabic characters can vary a great deal, depending on their context. The various shapes, including ligatures and context-dependent forms, were *not* identified in the ground truth transcriptions.

**7. Measuring OCR output correctness**

Once the OCR results have been delivered, it is needed to get an idea of the quality of the recognized full-text. There are several way of doing this and a number of considerations to be taken [Joachim Korb 2008]

The quality of OCR results can be checked in a number of different ways. The most effective but also most labor extensive method is manual revision. Here analyzer checks the complete OCR result against the original and/or the digitized image. While this is currently the only method of checking the whole OCR-ed text, and the only way to get it almost 100% correct, it is also cost prohibitive. For this reason, most systems reject it as impractical.

All other methods of checking the correctness of OCR output can only be estimations, and none of these methods actually provides better OCR results. That is, further steps, which will include manual labor, will have to be taken to receive better results.

**7.1 Software log analysis vs. human eye spot test**

To get to such an estimation one can use different methods, which will yield different results. The simplest way is to use the software log of the OCR engine, a file in which the software documents (amongst other things) whether a letter or a word has been recognized correctly according to the software’s algorithm. While this can be used with other (often special) software and thus allow for the verification of a complete set of OCRed material, it is also of rather limited use. The reason for this is that the OCR software will give an estimation of how certain the recognition is according to that software's algorithm. This algorithm cannot realize any mistakes made because they are beyond the software's scope. For example: Many old font sets have an (alternative) 's', which looks very similar to an 'f' of that same font set. If the software has not (properly) been trained to recognize the difference it will produce an 'f' for every such 's'. The software log will give high confidence rates for each wrongly recognized letter and even the most advanced log analysis will not be able to realize the mistake.

The second method for estimating the correctness of OCR output is the human eye spot test. Human eye spot tests are done by comparing the corresponding digital images and fulltext of a random sample. This is much more time consuming than log analysis, but when carried out correctly it gives an accurate measurement of the correctness of the recognized text. Of course, this is only true for the tested sample, the result for that sample is than interpolated to get an estimation of the correctness for the whole set of OCRed text. Depending on the sample, the result of the spot test can be very close to or very far from the overall average of the whole set.

**7.2 Letter count vs. word count**

After deciding on the method for estimation, one has to decide what to count. One can compare either the ratio of incorrect to correct letters or the ratio of incorrect to correct words. The respective results may again be very different from each other.

In either method, it is important to agree on what counts as an error. One could for example, count every character (including blank spaces) that has been changed, added or left out.

For example: The word 'Lemberg' has been recognized as 'lern Berg'. In letter count, this would be counted as five mistakes: 1: 'l' for 'L', 2: ''r and 'n' for 'm', 3: one letter added, 4: blank space added, 5: 'B' for 'b'. Notice that the placement of 'r' and 'n' for 'm' counts as two mistakes!

In word count the same example would count as two mistakes. One, because the word has been wrongly recognized and two, because the software produced two words instead of one.

Currently, the letter count method is mostly used, because it produces the same difference in the average for each detected error. That is each detected error is counted as one error, regardless of its importance within the text. The problem with letter count is that it is impossible to make statements about searchability or readability from it.

The word count average, on the other hand, only changes if a new error also appears in a new word. That is to say, when two letters in a single word are recognized wrongly, the whole word still counts as a single error. If an error is counted, though, it usually changes the average much more drastically than it would in letter count, because there are fewer words in a text than there are letters.

While word count will give a much better idea of the searchability or readability of a text, it does not take into account importance of an error in the text. Thus an incorrectly recognized short and comparatively unimportant word like “to” will change the average as much as one in a longer word like “specification” or a medium sized word like “budget”. Thus, the predictions about searchability or readability of a text made from word count are not very accurate either.

Only a very intricate method that would weigh the importance of each error in a given text could help here. There are now projects working on this problem, but there is as yet no software that does this and employing people to do it would not be practical.

**7.3 Re-consider checking OCR output accuracy**

Because of the problems with all methods described above and because the simple estimation of the percentage of errors in a text does not change the quality of current OCR software, libraries planning large scale digitization projects should consider refraining from checking the quality of their OCR results on a regular basis. Even in smaller projects, where checking OCR results is more feasible, the amount of work put into this task should carefully considered.

This said, at least at the beginning of a project the OCR output should be checked to a certain extent to make sure that the software has been trained for the right fonts, the proper types of documents and the correct (set of) languages.

Also, to get a simple overview of the consistency of the OCR output and to find typical problems, it may be a good idea to put the software's estimated correctness values into the OCR output file or to keep it separately. A relatively simple script can then be used to monitor these values and to find obvious discrepancies. These can then be followed up to see where the problem is and what, if anything, can be done about it.

**8. Competitions**

**8.1 ICDAR Arabic Handwriting Recognition**

The ICDAR Arabic Handwriting Recognition Competition aims to bring together researchers working on Arabic handwriting recognition. Since 2002 the freely available IfN/ENIT-Database is used by more than 60 groups all over the world to develop Arabic handwriting recognition systems [Volker et al 2009].

**Evaluation Process:**

The objective is to run each Arabic handwritten word recognizer (trained on the IfN/ENIT-Database) on an already published part of the IfN/ENIT-Database and on a new sample not yet published. The recognition results on word level of each system are compared on the basis of correct recognised words / respective there dedicated ZIP(Post)-Code. A dictionary can be used and should include all 937 different Tunisian town/village names.

**8.2 ICDAR Online Arabic Handwriting Recognition**

The ICDAR Online Arabic Handwriting Recognition Competition aims to contribute in the evolution of Arabic handwriting recognition research. This competition is organized on the database on online Arabic handwritten text (ADAB). A comparison and discussion of different algorithms and recognition methods should give a push in the field of Arabic handwritten word recognition [Volker et al 2009]

**Evaluation Process**

The object is to run each Arabic handwritten word recognizer (trained on a part of version 1.0 of the ADAB-database) on an already published part of the ADAB-database and on a test set not included in the published part. The recognition results on word level of each system are compared on the basis of correct recognised words, i.e. there correspondent consecutive Numeric Character References (NCR). A dictionary can be used in the recognition process.

**8.3 ICDAR Printed Arabic OCR competitions:**

No competition is available for Arabic machine printed OCR like that for offline and online handwriting recognition

**9. Tools and Data Dependency:**

**9.1 OCR:**

1. ScanFix pre-processing tool (or similar): 15$ per license.
2. Nuance document analysis tool (Framing tools) (or similar): 30$ per license.
3. Word based language model: Needs corpus
4. Character based language model: Needs segmented, annotated corpus
5. Grapheme to ligature and ligature to grapheme convertor: Need to build a tool
6. Statistical training tools: HTK, SRI, Matlab.
7. Error analysis tools: Need to be implemented.
8. Diacritic Preprocessing tool
9. Language Recognition tool

**9.2. ICR:**

1. Pre-processing tool
2. Word based language model: Needs corpus
3. Character based language model: Needs segmented, annotated corpus
4. Grapheme to ligature and ligature to grapheme convertor: Need to build a tool
5. Statistical training tools: HTK, SRI, Matlab.
6. Error analysis tools: Need to be implemented.
7. Language Recognition tool

**10. Research Approaches**

**10.1. ICR** [Abdelazim 2005], [Volker et al 2009]

|  |  |  |  |
| --- | --- | --- | --- |
| **Author(s)** | **Description** | **Data** | **Results** |
| Abuhaiba et al.(1994) | Fuzzy Models (FCCGM) | 1410 letters | 99.4% |
| Amin et al.(1996) | NN | 3000 characters | 92% |
| Alimi(1997) | Neuro-fuzzy | 100 words | 89% |
| Dehghani et al.(2001) | Multiple HMM | Farsi-Cities | 71.82% |
| Maddouri et al.(2002) | TD-NN | 70 words 2070 images | 97% |
| Khorsheed(2003) | Universal HMM | ancient documents | 87% |
| Alma'adeed et al.(2004) | Multiple HMM's | AHDB | 45% |
| Haraty & Ghaddar(2004) | NN | 2132 letters | 73% |
| Souici-Meslati & Sellami(2004) | NN | 55 words | 92% |
| Farah et al.(2004) | ANNK-NN, fuzzy K-NN | 48 words (100 writers) | 96% |
| Safabakhsh & Adibi(2005) | CD-VD-HMM | 50 words | 91% |
| Pechwitz & Märgner(2003) . (ARAB-IfN) | |  | | --- | | SC-1D-HMM | | IFN/ENIT | 2003: 89% |
|  |  | IFN/ENIT | 2005: 74.69% |
| Jin et al.(2005) (TH-OCR) | Statistical methods | IFN/ENIT |  |
| Touj et al.(2005) (REAM) | Planar HMMs | IFN/ENIT |  |
| Kundu et al.(2007) (MITRE) | VD-HMM | IFN/ENIT | 61.70% |
| Ball (2007) (CEDAR) | HMM | IFN/ENIT | 59.01% |
| Pal et al.(2006) (MIE) |  | IFN/ENIT | 83.34% |
| Schambach(2003) (SIEMENS) | HMM | IFN/ENIT | 87.22% |
| Al-Hajj et al.(2006) (UOB-ENST) | HMM | IFN/ENIT | 2005: 75.93% |
| **Same group** | |  | | --- | |  | | IFN/ENIT | 2007: 81.93% |
| Abdulkadr(2006) (ICRA) | NN (Two-Tier approach) | IFN/ENIT | 2005: 65.74% |
| **Same group** |  | IFN/ENIT | 2007: 81.47% |
| Menasri et al.(2007) (Paris V) | hybrid HMM/NN | IFN/ENIT | 80.18% |
| Benouareth et al.(2008) | HMM | IFN/ENIT | 89.08% |
| Zavorin et al.(2008) (CACI) | HMM | IFN/ENIT | 52% |
| Dreuw et al.(2008) | HMM | IFN/ENIT | 80.95% |
| Graves & Schmidhuber (2008) | MDR-NN | IFN/ENIT | 91.43% |
| Kessentini et al.(2008) | HMM multi-stream | lexicon of 500 words | 86.2% |

**10.2. OCR** [Abdelazim 2005], [ El-Mahallawy 2008]

|  |  |  |  |
| --- | --- | --- | --- |
| **Author(s)** | **Description** | **Data** | **Results** |
| Abdelazim , et al(1990) | Probabilistic Correlation | Single font Database | 96% |
| El Badr (1995) | Bayesian Classifier-word based | 42,000 words | 73%-94% |
| R.C. Vogt (1996) | Template Matching | 220,000 words | 65% |
| H. Amir,et al (2003) | Generalized Hough Transform | isolated Arabic Characters | 93% |
| Gillies , et al (1999) | NN | 344 pages | 90% |
| Khorsheed et al (1999) | HMM | Closed vocabulary | 97% |
| Ozturk , et al (2000) | Multi-Layer BP neural | isolated Arabic Characters | 95% |
| Abdelazim , et al (2001) | Bayesian Classifier | 10 different font database | 96.5% |
| J. Makhooul, et al (2001) | HMM | DARPA Arabic OCR Corpus | 95%-99% |
| Klassen T.J., and Heywood M.I.(2002) | NN\_SOM | isolated Arabic Characters | 80%-90% |
| Abdulaziz Al-Khuraidly, et al (2003) | Moment Invariants, NN-  RBF | Naskh font only is used | 73% |
| Khorsheed et al (2007) | HMM | 116,743 words and 596,931 characters of six different computer-generated fonts | 85.9 % |
| Rashwan et al (2007,2009) | Autonomously Normalized Horizontal Differential Features for HMM-Based Omni Font-Written OCR | 270000 word is used for training  representing 6 different sizes and 9 fonts (Microsoft and Mac.)  72000 word is used for testing  representing 6 different sizes and 12 fonts( Microsoft and Mac. ) | 99.3% |

**11. Current Projects of National Interest:**

11.1. **Million Book Project by Alexandria Bibliotica:**

Alexandria Library uses Sakhr and Novodynamics OCRs for Arabic documents and ABBY OCR for Latin documents in their million book project digitization. Sakhr is better than Novodynamics for high quality documents but Novodynamics is significantly better for bad quality documents.

**11.2. The E-content Project: Dr. Hoda Baraka (Dr. Samya Mashaly)**

No Data is available till now

**11.3. Dar El Kotob Project (?).**

No Data is available till now

**12. Recommendations for Benchmarking and Data Resources:**

**12.1 Benchmarking**:

The recommended Benchmarking must be two-folded; one is to measure robustness and reliability of the product (software) and this requires 40,000 documents in one batch. These should include simple and complex documents, different qualities, etc.

The second test, for accuracy, should include at least 600 pages (200 high quality, 200 medium, and 200 poor quality) coming from books, newspapers, Fax outputs, Typewriters, etc.

**12.2 Training Data for a basic research tool**:

The amount of training data required (for researchers to build printed OCR systems) is configured in the following:

We need to focus on the Naskh fonts family. Within Naskh, there may be about 6 families. Each would have 6 different font sizes (8,10,12,14,16,18). The rule is that we need to have about 25 instances for each shape in each case. We assumed to have about **300** different shapes (characters and ligatures). So we need **300\*25=7500** instances. This is about 8 pages.

This should be done for each **font family** and for each **font size** as follows:

**8**pages\***6**faontsfamilies\***6**fontsizes= around **300** pages total.

These pages (for clean high quality training data) will be generated artificially, by balancing the data to cover all the **300** shapes. We will use nonsense character strings to cover the characters equally.

Then, to generate lower quality training data:

a- The **300** pages will be outputted from a Fax machine (once)

b- The **300** pages will be copied once (one output), then twice (second output).

The same process will be done for **600**, **300**, and **200** dpi. (This now gives **3600** pages: **300** clean, **300** from Fax, **300** copied once, **300** copied twice, then multiply those **1200** by **3** for the **3** different resolutions).

We will also obtain **2000** transcribed pages from Alex. Bib. with low quality old books, etc.).

**13. Survey Issues:**

13.1 List of Researchers and Companies to be contacted

1. Sakhr
2. RDI
3. ImagiNet
4. Orange- Cairo
5. IBM- Cairo
6. Cairo University
7. Ain Shams University
8. Arab academy (AAST)
9. AUC
10. GUC
11. Nile University
12. Azhar university
13. Helwan university
14. Assuit university
15. Other companies that are users of the technology

13.2 List of Key Figures in the Field to invite in the conference

1. **John Makhoul, (BBN)**
2. **Luc Vincent (Google)**
3. [**Lambert Schomaker**](http://www.ai.rug.nl/~lambert/)**: Rijksuniversiteit Groningen (The Netherlands)**

**14. SWOT Analysis**

**14.1. Strengths**

The expertise, good regional & int’l. reputation, and achievements of the core team *researchers* in DSP, pattern recognition, image processing, NLP, and stochastic methods.

**14.2. Weaknesses**

1. The is a late comer to the market of Arabic OCR.

2- The tight time & budget of the intended required products.

3- No benchmarking available for printed Arabic OCR

4- No training database available for research community for Arabic OCR

**14.3. Opportunities**

Truly reliable & robust Arabic OCR/ICR systems are a much needed essential technology for the Arabic language to be fully launched in the digital age.

2- No existing product is yet satisfactory enough! (See appendix I for Evaluation of commercial Arabic OCR packages)

3- The Arabic language has a huge heritage to be digitized.

4- Large market of such a tech. of over 300 million native speakers, plus other numerous interested parties (for reasons such as security, commerce, cultural interaction, etc.).

**14.4. Threats**

1. Back firing against Arabic OCR technologies in the perception of customers, due to a long history of unsatisfactory performance of past and current Arabic OCR/ICR products.

2- Other R&D groups all over the world (esp. in the US) is working hard and racing for a radical solution of the problem.

**15. Suggestions for Survey Questionnaire:**

1. **Specify the application that OCR recognition will be used for**
2. **What is the data used/intended to train the system?**
3. **What is the benchmark to test your system on?**
4. **Would you be interested to contribute in the data collection. At what capacity?**
5. **Would you be interested to buy Arabic OCR annotated data?**
6. **Would you be interested to contribute in a competition**
7. **How many persons working in this area in your team? What are their qualifications?**
8. **What are the platforms supported/targeted in your application?**
9. **What is the market share anticipated in your application?**
10. **Would your application support any other languages? Explain.**

**REFERENCES**

[1] Abdelazim, H. Y. “Recent trends in Arabic OCR,” in Proc. 5th Conference of Engineering Language, Ain Shams University, 2005.

[2] Al-Badr, B., Mahmoud, S.A., Survey and Bibliography of Arabic Optical Text Recognition, Elsevier Science, Signal Processing 41 (1995) pp. 49-77.

[ 3] A. Asiri, and M. S. Khorsheed, "Automatic Processing of Handwritten Arabic Forms Using Neural Networks", PWASET vol. 7, August 2005

[4] Attia, M., Arabic Orthography vs. Arabic OCR, Multilingual Computing & Technology magazine, USA, Dec. 2004.

[5] Attia, M., El-Mahallawy, M. “Histogram-Based Lines & Words Decomposition for Arabic Omni Font-Written OCR Systems; Enhancements and Evaluation”, Lecture Notes on Computer Science (LNCS): Computer Analysis of Images and Patterns, Springer-Verlag Berlin Heidelberg, Vol. 4673, pp. 522-530, 2007.

[6] Attia, M., Rashwan, m. A. A., El-Mahallawy, M.S.M., "[Autonomously Normalized Horizontal Differentials as Features for HMM-Based Omni Font-Written OCR Systems for Cursively Scripted Languages](http://www.rdi-eg.com/rdi/Downloads/Scientific%20Papers/ICSIPA2009_Camera-Ready.pdf)", [ICSIPA2009](http://www.sp.ieeemalaysia.org/ICSIPA09/), Kuala\_Lumpur-Malaysia, Nov.2009.

http://www.rdi-eg.com/rdi/technologies/papers.htm

[7] Applied Media Analysis. "Arabic-Handwritten-1.0", 2007

http://appliedmediaanalysis.com/Datasets.htm.

[8] Huda Alamri, Javad Sadri,Ching Y. Suen,Nicola Nobile, "A Novel Comprehensive Database for Arabic Off-Line Handwriting Recognition", ICFHR Proceedings, 2008

[9] J. Makhoul, I. Bazzi , Z. Lu, R. Schwartz,   and P. Natarajan, "Multilingual Machine Printed OCR," International Journal of Pattern Recognition and Artificial Intelligence Vol. 15, No. 1 43-63 © World Scientific Publishing Company BBN Technologies, Verizon, Cambridge, MA 02138, USA, 2001.

[10] Joachim Korb, "Survey of existing OCR practices and recommendations for more efficient work", 2008, TEL plus project

[11] Khorsheed, M.S. “Offline Recognition of Omnifont Arabic Text Using the HMM ToolKit (HTK)”, Pattern Recognition Letters, Vol. 28 pp. 1563–1571, 2007.

[12] Rashwan, M., Fakhr, W.T., Attia, M., El-Mahallawy, M., “Arabic OCR System Analogous to HMM-Based ASR Systems; Implementation and Evaluation”, Journal of Engineering and Applied Science, Cairo University, www.Journal.eng.CU.edu.eg, December, 2007

[13] Somaya Al-Ma’adeed, Dave Elliman, Colin A Higgins, "A Data Base for Arabic Handwritten Text Recognition Research", 2002,IEEE Proceedings

[14] Stephanie M. Strassel, "Linguistic Resources for Arabic Handwriting Recognition", Proceedings of the Second International Conference for Arabic Handwriting Recognition,. 2009.

[15] Volker Margner,Haikal El Abed, "Arabic Handwriting Recognition competition", ICDAR 2009

[16] Windows Magazine, middle east, "Arabic OCR packages", Apr. 2007, pp.82- 85.

[17] Yousef Al-Ohali,Mohamed Cheriet, Ching Suen,"Databases For Recognition of Handwritten Arabic Cheques",In: L.R.B. Schomaker and L.G. Vuurpijl (Eds.), Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition, September 11-13 2000, Amsterdam, pp 601-606