OCR OF ARABIC TEXT WITH SAKHR

Definition of terms ........................................................................................................................................ 2
Summary of processes.................................................................................................................................... 4
Scanfix preparatory procedures: ................................................................................................................... 6
Selecting sample PTIFFs to process and learn .......................................................................................... 6
Custom-processing the sample images ....................................................................................................... 8
Possible Processing Outcomes .................................................................................................................. 11
   Scenario A: Light Processing................................................................................................................... 11
   Scenario B: Heavy Processing.................................................................................................................. 13
   Scenario C: Adequate Processing........................................................................................................... 15
Sakhr preparatory procedures: .................................................................................................................... 16
   Sakhr startup .......................................................................................................................................... 16
   Drawing text frames............................................................................................................................... 17
   Creating a custom Sakhr font.................................................................................................................. 18
   Running OCR......................................................................................................................................... 21
   Evaluating and improving Sakhr results............................................................................................... 22
   Finishing up ............................................................................................................................................ 27
Sakhr batch job procedures: ....................................................................................................................... 27
   Using Automatic Reader Wizard.......................................................................................................... 27
## Definition of terms

<table>
<thead>
<tr>
<th><strong>Batch job</strong></th>
<th>Processing or running OCR on more than one image file at the same time.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Diacritical mark</strong></td>
<td>In Arabic, many characters (letters) have exactly the same form, and are distinguished only by the position of various dots over, above, or inside the main character block. This property poses special difficulty for OCR, as dots can be ignored by software as speckling or error, or even removed.</td>
</tr>
<tr>
<td><strong>Exclusion frame</strong></td>
<td>An exclusion frame can be drawn manually inside a text frame to instruct Sakhr not to attempt to recognize non-white space inside it.</td>
</tr>
<tr>
<td><strong>Font</strong></td>
<td>A complete set of characters of a particular size and style of type. Different publications are printed in different fonts, whose characters may look quite different from each other. If a new font is encountered, the OCR technician must teach the software to recognize it.</td>
</tr>
<tr>
<td><strong>Gap</strong></td>
<td>Arabic letters are printed in a cursive, connected form. Depending on the font and volume, letters can sometimes be printed with gaps in between. These should be filled in during the processing stage for ideal results.</td>
</tr>
<tr>
<td><strong>Jaggies</strong></td>
<td>Multiple repeating protrusions and/or cavities on the edges of scanned characters that appear when zoomed in to the pixel level. Jaggies should ideally be eliminated in the processing stage for optimal OCR results.</td>
</tr>
<tr>
<td><strong>Image frame</strong></td>
<td>An image frame can be drawn either automatically or manually, and instructs Sakhr to ignore non-white space within it as an image.</td>
</tr>
<tr>
<td><strong>Kashīdah</strong></td>
<td>The space between two connecting Arabic characters, usually in the form of a low horizontal line, but possibly slanting (depending on the characters it connects). The size and shape of the kashīdah can be manipulated in Sakhr to achieve optimal recognition results.</td>
</tr>
<tr>
<td><strong>Ligature</strong></td>
<td>The phenomenon whereby any of several Arabic characters, when placed consecutively in a certain order, can be printed in an irregular form. The OCR software must be taught to recognize these ligatures as composite forms, since attempting to read them one character at a time may result in error.</td>
</tr>
<tr>
<td><strong>Loop</strong></td>
<td>As do the English letters ‘g,’ ‘e’ and ‘p,’ some Arabic letters have islands of white space inside them. Due to poor printing, or quality deterioration during scanning and processing, these ‘loops’ can become filled in; this has a poor effect on the OCR results.</td>
</tr>
<tr>
<td>Term</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NLP</td>
<td>Stands for Natural Language Processing. Sakhr can use an NLP procedure after a recognition process to make further improvements in the recognized text, based on what it understands of the sentence syntax, grammar or content. NLP substantially slows the recognition process, and should be used at the discretion of the OCR technician.</td>
</tr>
<tr>
<td>OCR</td>
<td>Stands for Optical Character Recognition. This is the process whereby scanned images are electronically ‘read’ and converted to editable text files. The ultimate goal is to embed the text files into an image database, allowing the page images to be searched by their content.</td>
</tr>
<tr>
<td>OTIFF</td>
<td>Original TIFF image file, received directly from scanner</td>
</tr>
<tr>
<td>PTIFF</td>
<td>Processed TIFF image file, usually de-skewed so that all lines of text are perfectly horizontal and cropped so that the black edges from the scanner are eliminated. Eventually, PTIFFs will undergo a second phase of processing focused on improving the legibility of the individual characters themselves.</td>
</tr>
<tr>
<td>Text frame</td>
<td>Drawn either automatically or manually, the text frame indicates Sakhr’s recognition target area. Sakhr will attempt to recognize all non-white space inside a text frame as some sort of text.</td>
</tr>
<tr>
<td>TIFF</td>
<td>Stands for Tagged Image File Format. It is the type of image file AMEEL AND IRAQ RECOLLECTION use for preservation.</td>
</tr>
<tr>
<td>Vowels</td>
<td>Distinguished from diacritical marks, vowels are not actually elements of Arabic letters. Examples of vowels include the <em>fathah</em>, <em>dammah</em>, <em>kasrah</em>, and <em>sukūn</em>. Vowels are not usually present in printed volumes, but if they are, they should be ignored during the OCR process.</td>
</tr>
</tbody>
</table>
Summary of processes

The preparation phase is crucial to achieving maximum accuracy during OCR. The Sakhr recognition software package has the capability to recognize a generic Arabic font (*Naskh* or *Kūfī*) with a fair degree of accuracy. The OCR technician, on his/her part, must teach the software to recognize any peculiarities or unusual characteristics in the font of the scanned volume in question. These may include unusual ligature formations, oddly-shaped letters, punctuation marks, slanted *alifā*, or the letters *sīn* and *shīn* with only one or two peaks, for example.

Before running OCR, the technician selects three PTIFFs from the scanned volume to serve as sample pages. One of the pages should be the Table of Contents page. If the other two pages have substantial text, then the three pages are probably sufficient. If not, a fourth may be necessary.

A significant amount of time should be spent experimenting with the custom ‘erode,’ ‘grow,’ and ‘sand and fill’ function settings in the Scanfix software on the sample pages for each volume in order to determine the most appropriate settings for the volume as a batch job. The goal is to make the text as legible (for the OCR software) as possible: this means smoothing out jagged characters, plugging gaps between letters, and solidifying character blocks without filling in crucial white space within characters or connecting separate diacritical marks. When satisfied with the custom settings on the sample pages, the technician decides on appropriate settings for the entire volume and runs a batch job through Scanfix.

The technician then runs ‘full’ recognition scans in Sakhr on the three sample PTIFFs (which have now been custom processed to be as legible as possible). This procedure allows the technician to walk the software, character by character, through the course of OCR. When the software recognizes a character correctly, the technician positively reinforces this recognition. When the software recognizes a character incorrectly, the technician corrects the software and, if appropriate, instructs the software to avoid making the same mistake in the future by modifying the font library he/she has created specific to that volume.

In general, the OCR technician should only modify the font library for the sake of font peculiarities that occur over the course of the volume. For example, if the technician finds the left-hand side of the loop on the final form of the letter *nūn* to be faint or missing throughout a volume (this might happen as a result of an over-worn type block in the printing press), he/she should instruct the software to recognize this alternate form of *nūn*. However, if the technician finds the left-hand side of the loop on the final form of the letter *nūn* to be faint or missing on one occasion, but cannot locate more than a few similar instances throughout the volume (this might happen as a result of a one-time error during the printing process), it is likely that causing the program to learn the alternate form will precipitate more mistakes than it will avoid.
(It is taken for granted here that the font will be more or less consistent throughout any given volume; in cases where the font actually does vary considerably, the technician should divide the volume appropriately and repeat this process for each section in which the font remains internally consistent.)

After running ‘full learn’ recognition scans on all three sample PTIFFs, the technician decides if more ‘learning’ is necessary. If so, the above process is repeated; if not, a batch process of the entire volume (already processed in Scanfix according to the settings determined by the three sample pages) is run through Sakhr. The program is capable of rendering an image into text at a rate of approximately one page per five seconds. This rate is substantially slowed when NLP is used.

Finally, the success and efficacy of the OCR phase is heavily reliant upon the quality of PTIFFs – that is to say, upon the quality of scanning and initial phases of processing. No amount of utilizing the grow/erode functions can replace the need for sharp, clear OTIFFs straight from the scanner.
Scanfix preparatory procedures

Selecting sample PTIFFs to process and learn
To serve as sample learning pages, PTIFFs should be mostly if not entirely filled with text. Select three page images from different sections in the volume, to ensure as great a variety of printing and scanning quality as possible. One of the pages should be the Table of Contents page. If the other two pages have substantial text, then the three pages are probably sufficient. If not, a fourth may be necessary. The images do not have to be especially clean and sharp, but there should be no largely illegible sections. After making a selection, save a copy of these PTIFFs in a separate folder.

### Good choice for sample PTIFF

<table>
<thead>
<tr>
<th>Arabic Text</th>
<th>English Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>توضيح الصور الفضائية</td>
<td>Explanation of the image</td>
</tr>
<tr>
<td>توضيح الصور الفضائية</td>
<td>Explanation of the image</td>
</tr>
<tr>
<td>توضيح الصور الفضائية</td>
<td>Explanation of the image</td>
</tr>
<tr>
<td>توضيح الصور الفضائية</td>
<td>Explanation of the image</td>
</tr>
</tbody>
</table>

### Poor choice for sample PTIFF

<table>
<thead>
<tr>
<th>Arabic Text</th>
<th>English Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>توضيح الصور الفضائية</td>
<td>Explanation of the image</td>
</tr>
<tr>
<td>توضيح الصور الفضائية</td>
<td>Explanation of the image</td>
</tr>
<tr>
<td>توضيح الصور الفضائية</td>
<td>Explanation of the image</td>
</tr>
<tr>
<td>توضيح الصور الفضائية</td>
<td>Explanation of the image</td>
</tr>
</tbody>
</table>
1. Decide on three PTIFFs from the scanned volume to use as sample images.

After deciding on which PTIFFs to use as samples, the images must be processed in order to make the characters as legible as possible.

2. Open one of the sample PTIFFs in Scanfix:
Custom-processing the sample images

3. Zoom in on the image until the individual pixels that form the characters are discernible. The zoom commands are located on a vertical strip of buttons along the left-hand side of the screen.
4. Examine the zoomed-in portion for potential recognition errors. In this sample, several issues should come to mind:

- The word *tasānīf* contains the following problems: gap between the *tāʾ* and *sād*, incomplete *sād*, gap between the *yāʾ* and *fāʾ*, and incomplete *fāʾ*.
- The word *minhā* contains a gap between the *nūn* and *hāʾ*.
- The loop in the letter *ʿayn* in the word *badāʾiʿ* is almost entirely closed.
- The long *kashidah* in the word *al-nizām* has jaggies along the bottom.
- The *rāʾ* in the word *wa-ashʿār* has a vertical gap.
- The word *niyābah* contains a gap between the *bāʾ* and the *tāʾ* *marbūṭah*.

The horizontal gaps suggest that the image should be artificially ‘horizontally grown,’ and the vertical gaps and incomplete letters suggest that it should be ‘vertically grown,’ as well. For the jagged *kashidah*, the ‘sand and fill’ option can be used. Take care, however, not to allow loops on letters such as the *mīm* or *wāw* to be filled in. Also, uneven growth can cause dots and/or diacritical marks to lose their shape and be interpreted by the OCR software as blobs.

5. Click on ‘settings,’ ‘advanced settings,’ and then the ‘smoothing & completion’ tab.

6. Replace the 0 in the ‘sand and fill’ box with a 3. A setting of 3 should be appropriate for most situations when characters have jaggies along the edges.

7. Input settings for the ‘custom grow and erode’ box:
a. A capital ‘H’ stands for ‘grow horizontally’
b. A capital ‘V’ stands for ‘grow vertically’
c. A lowercase ‘h’ stands for ‘erode horizontally’
d. A lowercase ‘v’ stands for ‘erode vertically’

e. Enter a magnitude for each command. For example, ‘H1’ will tell Scanfix to ‘grow horizontally’ by a factor of 1. ‘v1.7’ will tell Scanfix to ‘erode vertically’ by a factor of 1.7.

f. Scanfix executes commands in the order they are entered. For example, the command ‘H1v1h1V1’ will cause the image to be grown horizontally, eroded vertically, eroded horizontally, and then grown vertically.

8. After deciding on appropriate settings, click ‘OK’ to close the advanced settings dialog box.

9. Click ‘Process’ to process the sample PTIFF according to the parameters you have outlined. The following are three possible outcomes of processing using different commands and parameters:
Possible Processing Outcomes

Scenario A: Light Processing

- ‘Sand and fill’ command, magnitude=3

BEFORE              AFTER

- Note that the problem of the ragged kashida in the word al-nizām has been solved by the ‘sand and fill’ option
• Characters are cleaner in general:

![Before](image1.png)

![After](image2.png)

• However, most of the gaps that previously existed between characters have not been filled by simple use of the ‘sand and fill option.’
Scenario B: Heavy Processing

- ‘Sand and fill’ command, magnitude=3
- ‘Custom grow and erode’ command, parameters=H3V3
• Note that the vertical gap within the letter \( rā’ \) in the word \( wa-ash ʻār \) has been filled in:

BEFORE

\[
\text{وانشاعر}
\]

AFTER

\[
\text{وانشاعر}
\]

• Note that many other gaps between or within characters have been filled in completely or almost completely.

• However, note also that all of the \( mīm \) characters and most of the \( wāw \) characters have been completely darkened, so that no open loops remain. This is an undesired result; the OCR engine may have difficulty reading these characters.
Scenario C: Adequate Processing

- ‘Sand and fill’ command, magnitude=3
- ‘Custom grow and erode’ command, parameters=H1V1

BEFORE

AFTER

- This is an acceptable compromise. Not all gaps between/within characters have been filled, but they have been improved. Also, the integrity of vulnerable characters such as \( \text{wāw} \) and \( \text{mīm} \) has been generally preserved. Overall, characters have been smoothed, and legibility has increased.

10. Save the processed sample images in the appropriate directory.
Sakhr preparatory procedures

Sakhr startup
11. Open Sakhr Automatic Reader Gold Edition 8.0 after first inserting the Sakhr dongle into a USB port.

12. Open a processed sample image from the appropriate directory. (Note: because Scanfix saves TIFFs in a slightly different way, you may have to select ‘All Files’ in the ‘Files of Type’ menu.)

13. The image will open inside the Sakhr interface.

Depending on the layout of text on the page, it may be necessary to draw text frames to instruct Sakhr in what order to read the text. The buttons for text frame commands are located at the upper right-hand corner of the screen.
## Drawing text frames

| Button 1 | Automatically draws text, exclusion and image frames. Use in situations when layout is straightforward and uncomplicated, or when no stylized text (e.g. in titles) exists. |
| Button 2 | Deletes all existing frames. |
| Button 3 | Allows text frame to be drawn manually |
| Button 4 | Allows exclusion frame to be drawn manually |
| Button 5 | Allows image frame to be drawn manually |
| Button 6 | Allows sequence of frames to be determined manually |
| Button 7 | Automatically ‘optimizes’ sequence of frames |
| Button 8 | Adds ‘breakpoint,’ a point from which sections can be added or deleted from the frame |
| Button 9 | Removes breakpoint |
| Button 10 | Allows user to add area between two breakpoints to the text frame |
| Button 11 | Allows user to subtract area between two breakpoints to the text frame |
Creating a custom Sakhr font

14. After text boxes are drawn, create a new Sakhr font and load it into the recognition engine to begin learning. From the ‘Tools’ menu, select ‘Options’ and then click on the ‘Engines’ tab.

![Options dialog box]

Make sure ‘Arabic’ is selected as the current OMNI engine, and that the font type is ‘Naskh.’ For the purposes of AMEEL AND IRAQ RECOLLECTION, ‘Kofi’ (Kūfi) will probably be used only rarely. Click on the button to the right of the ‘Additional Font’ box. In the new dialog box, click on ‘Font Manager.’ Select the button that appears labeled ‘New Font…’
In the new version of Sakhr, the settings are going to be different from this example shown above. For example, when uploading a font to the engine, you’ll go to tools, then choose, Active Profile Options.
15. Select the appropriate characteristics of the new font according to your observations of the typeface in the printed volume. These are very broad; fine-tuning of the new font will be accomplished later in the process.

- ‘Style’ should almost always remain set to ‘regular.’
- ‘Size,’ on the other hand, can vary according to volume.
- ‘Closest System Font’ is the most important menu, selected by observing the sample text in the white box and comparing it to the typeface in the printed volume.

Finally, type a name for the new font and click ‘Save.’ Then click ‘Add’ in the next dialog box, which adds the new font to Sakhr’s font library.
16. Before clicking ‘OK,’ check the settings in the ‘Recognition’ tab:

- In almost all situations, the radio button for ‘Diacritics Evaluation’ should be set to ‘skip.’
- The average noise size should be as large as possible to avoid interpreting dots as noise.
- The boxes for ‘Automatic Style Detection,’ ‘Use Arabic Linguistic rules,’ and ‘Neglect Cashida’ should be checked; everything else should be left unchecked.
- Set the ‘Failure Character’ to ‘@.’

Running OCR

17. Run OCR on the processed sample image by clicking on ‘Start’ from the ‘Recognize’ menu, or by clicking the appropriate button at the top of the screen.
18. After completing the OCR process, Sakhr displays two frames: on top, a zoomed-in view of the sample image, and on bottom, a text box displaying the OCR results. The position of the cursor in the OCR results determines the position of the thin red box in the sample image. Clicking on a word or character in the OCR results moves the thin red box to the appropriate location.

Evaluating and improving Sakhr results

19. After the OCR process is complete, check Sakhr’s official error statistics, located in the ‘Recognize’ menu. Record these statistics.
20. Start ‘full learn’ process by selecting ‘Full’ from the ‘Learn’ submenu of the ‘Recognize’ menu. Note: if this option is grayed-out, this means you have not properly loaded a font. A ‘full learn’ can be accomplished only on a user-created font, not the system default font.

The purpose of the full learn is for the user to follow Sakhr character by character and correct the program, when necessary. To indicate the appropriate character, Sakhr outlines it in a red box. At the same time, the character appears, in text form, in the ‘learn font’ box below.
If the character displayed in the ‘character’ field of the ‘Learn font’ dialog box matches the character outlined in red above, press ‘Accept.’ This will instruct Sakhr to move on to the next character. In this case, Sakhr has correctly recognized the *alif*.

Next character:

Sakhr has correctly recognized this character as well, which is a *lām*. Press ‘Accept’ to move on to the next character.
21. Eventually, there will be a character Sakhr fails to recognize.

Here, Sakhr has recognized the mīm as an alif with a raised hamzah. Double-click on the mīm button toward the bottom of the dialog box (or type the mīm in manually from the keyboard and press the ‘Learn button that will appear). The following dialog box will pop up:

This dialog box allows you to add the misrecognized character to your font. On the left, in the ‘symbol’ box, is indicated the character that you are improving (mīm). Below that is a picture of the actual piece of the image that Sakhr misrecognized. On the right, check to make sure that the correct radio button is selected for the form of the character (here, the ‘isolated’ form is correctly selected). You may leave the outlined space in the ‘Samples’ area where it is to save a new sample, or drag it over another sample to replace it. Then, click ‘OK’ to improve the character. Sakhr will automatically advance to the
next character in the ‘full learn’ process; continue until the entire page has been reviewed.

22. Other regularly-used commands in the ‘learn font’ dialog box:

1. Click on ‘1’ or ‘2’ to see more characters to choose from. Page 2 consists mostly of punctuation marks and numerals.

2. ‘Fast learn’ mode (blue) vs. ‘Step by step learn’ mode (green). Only use fast learn mode on pages where accuracy is already very high (i.e., almost never).

3. These commands move the cursor left and right within the image text, not by character but by character block.

4. ‘Segmentation’ commands – these are used to manipulate the size and shape of the red box outlining a character. For example, the ‘Add next segment’ command (top left) can be used to highlight a complete ligature when Sakhr has highlighted only part of it. The ‘Remove last segment’ command (top right) can be used to break down a ligature into its components.

5. *Kashīdah* commands – these extend and truncate the *kashīdah* in left and right directions.
Finishing up
23. When the entire page has gone through the ‘full learn’ process, save your modified font by clicking ‘Save’ in the ‘Learn font’ dialog box. Exit the dialog box. Exit all text windows and image windows that are open in Sakhr.

24. Load the second of your three sample PTIFFs. Set text frames as you did on the first image. Load your previously-saved font (instead of creating a new one) and run a recognition. Initiate a ‘full learn’ process as you did on the first image. When complete, save the font again.

25. Repeat for third sample PTIFF.

Sakhr batch job procedures

Using Automatic Reader Wizard
26. Using the font you have just created and improved, it is now time to quickly scan the remaining pages of the volume in a batch job. Click on the ‘Automatic Reader Wizard’ button in the toolbar on top of the screen (or select it from the ‘Tools’ menu).

This wizard allows you to set up and complete your batch process step by step.
27. Select the radio button to indicate you will be running your batch process using already-saved files, not directly from a scanner. Then click ‘Next.’

28. In the screen that appears, click on the right-most button to add files to the batch recognition job.
29. Locate the directory where the image files from the scanned volume are stored, then select all the files within it by holding down the ‘Shift’ key and using the arrow keys. Then press ‘Add.’

![Select images to be recognized](image1)

30. Check to make sure the image files appear in the correct order when you are returned to the Automatic Reader Wizard. Sakhr has a habit of placing the alphabetically last file in front, as seen in the image below. If this occurs, use the arrow buttons in the box (indicated here) to move the file to its proper position.

![Select images to be recognized](image2)
31. Click the ‘Next’ button to move to the next step. You will be instructed to press the ‘Modify’ button if you need to switch font engines. This is actually the ‘Settings’ button on the bottom of the screen. Press this button and click on the ‘Engines’ tab on top of the dialog box. Next to the ‘Additional Font’ field, press the button with an ellipsis on it to load your customized font.
32. Double-click on your customized font (or click on it and press ‘Add’). Before pressing ‘OK,’ click on the ‘Recognition’ tab to make sure all the recognition settings are correct (see #16). Then press ‘OK’ in the previous dialog box to return to the Automatic Reader Wizard. Press the ‘Next’ button.

33. You will be prompted to specify any special image operations you want Sakhr to perform. Make sure ‘Image Rotation’ is set to ‘None’ and that the two squares for ‘Fix Image’ and ‘Invert image colors’ are unchecked. Then click ‘Next.’
34. In the next step, instruct Sakhr to save the output from each image file as individual text files. To do this, make sure the square for ‘Divide output into files’ is *checked*. The ‘Number of Pages in each file’ should be 1. Finally, check the box for ‘Use the same files names’ [sic] to make sure the directory naming process is standardized. Click the ‘Next’ button.

35. Under ‘Save text options,’ use the button with the ellipsis to select the appropriate directory for saving your output text files. Under the ‘Format’ menu, make sure that ‘DOS text’ is selected. Under the ‘Code page’ menus, select ‘Unicode.’ Finally, click the ‘Finish’ button.
36. Sakhr will now run a ‘Load and recognize’ process automatically until all the selected image files are saved as Unicode text. Depending on the number of pages, this process will take several minutes to complete. If NLP support is being used, the process will be dramatically slowed and may cause the computer to freeze if there is not enough available RAM.
37. When doing the batch with the new version of Sakhr, 9.0, you’ll see that it’s going to be different when doing the txt files batch. For example; in the format section, it will be Plain Text in stead of DOS Text.

38. When doing the art files, it’s going to be the same and you’ll choose Automatic Reader text.

39. After Sakhr is done, it will automatically give you a statistics box to tell you how many images were recognized, ignored, and some other information.

40. Copy these statistics as well in the same spreadsheet you have done for the same volume.
38. Hit *print screen* at the far right on the top of the keyboard, and paste it on paint. Then save it on the output folder of the volume you’re working on.

39. Click on *view error messages* on the statistics box, as well and save the page will appear to you on the output folder as well.