Retrieving Arabic Printed Document: a Survey

Kareem Darwish and Ossama Emam

IBM Technology Development Center, Cairo, Egypt

***Abstract* — This paper surveys some of the literature pertaining to searching and retrieving OCR'ed printed documents with emphasis on Arabic documents. It examines peculiarities of Arabic morphology, orthography, retrieval, word clustering, display, OCR, and error correction. The paper surveys existing evaluation test-beds for retrieval of Arabic OCR texts. Lastly, it concludes with possible directions for future research.**

***Index Terms* — Arabic, Information Retrieval, OCR, Morpholgoy, Othography, Error Correction.**

I. Introduction

Since the advent of the printing press in 15th century, the number of printed documents has grown overwhelmingly. Only recently has electronic text become ubiquitous. Electronic text is usually easy to search and retrieve which led to the development of many text search engines. Nonetheless, there remains a huge volume of legacy documents which are available in print only. One way to search and retrieve printed documents is by digitizing them and performing Optical Character Recognition (OCR) to transform the digitized printed documents (a.k.a. document images) into electronic text. Although the OCR process is not perfect and produces many errors, especially for orthographically and morphologically complex languages such as Arabic, it produces a text representation of the document images that can be retrieved (a.k.a. searched). Figure 1 demonstrates a possible document flow for a retrieval system that searches OCR'ed document images.

Much research has been conducted to improve the retrieval effectiveness and visualization of OCR'ed documents. This paper surveys some of this research, lists some of the available resources, and explores futures directions to further improve the process of retrieval and visualization. The paper focuses primarily on Arabic OCR'ed documents. This paper is organized as follows: Section 2 provides a background on the properties of the Arabic language including morphology, orthography, OCR, and retrieval; Section 3 surveys available retrieval test collection and their construction; Section 4 discusses OCR error handling; section 5 talks about the display of search result; section 6 provides future directions; and section 7 concludes the paper.

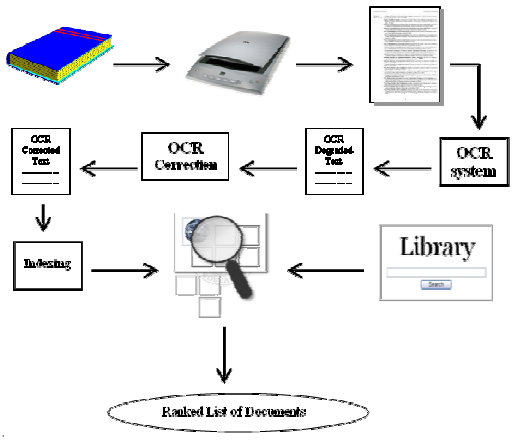

Fig. 1. Document flow in a printed document retrieval system

II. Arabic Properties

This section focuses on issues of morphology, retrieval, and Arabic OCR along with orthographic properties complicating OCR.

- **Arabic Morphology**

Arabic words are divided into three types: noun, verb, and particle (Abdul-Al-Aal, 1987). Nouns and verbs are derived from a closed set of around 10,000 roots (Ibn Manzour). The roots are commonly three or four letters and are rarely five letters. Arabic nouns and verbs are derived from roots by applying templates to the roots to generate stems and then introducing prefixes and suffixes. Figure 2 shows some templates for 3 letter roots. Figures 3 and 4 show some of the possible prefixes and suffixes and their corresponding meaning. The number of unique Arabic words (or surface forms) is estimated to be 6 x 1010 words (Ahmed, 2000). Figure 5 shows some of the words that maybe generated from the root ktb – كتب.

Further, a word may be derived from several different roots. For example the word AymAn – ايمان can be derived from five different roots. Figure 6 shows possible roots for the word AymAn – ايمان and the meaning of the word based on each. For the purposes of this paper, a word is any Arabic surface form, a stem is a word without any prefixes or suffixes, and a root is a linguistic unit of meaning, which has no prefix, suffix, or infix. However, often irregular roots, which contain double or weak letters, lead to stems and words that have letters from the root that are deleted or replaced.

Significant work has been done in the area of Arabic morphological analysis. Some of the approaches include:

1. The Symbolic Approach: In this approach, morphotactic (rules governing the combination of morphemes, which are meaning bearing units in the language) and orthographic (spelling rules) rules are programmed into a finite state transducer (FST). Koskenniemi proposed a two-level system for language morphology, which led to Antworth's two-level morphology system PCKIMMO (Antworth, 1990; Koskenniemi, 1983). Later, Beesley et al. developed an Arabic morphology system, ALPNET, which uses a slightly enhanced implementation of PC-KIMMO (Beesley et al., 1989; Kiraz 1998). However, this approach was criticized by Ahmed (2000) for requiring excessive manual processing to state rules in an FST and for the ability to only analyze words that appear in Arabic dictionaries. Kiraz (1998) summarized many variations of the FST approach.

2. Unsupervised Machine Learning Approach: Goldsmith (2000) developed an unsupervised learning automatic morphology tool called AutoMorphology. This system is advantageous because it would automatically learn the most common prefixes and suffixes from just a word-list. However, such a system would not be able to detect infix and uncommon prefixes and suffixes.

3. Statistical Rule-Based Approach: This approach uses rules in conjunction with statistics. This approach employs a handcrafted list of prefixes, a list of suffixes, and templates to extract a stem from a word and a root from a stem. Possible prefix-suffix-template combinations are constructed for a word. Hand-crafted rules are used to eliminate impossible combinations and the remaining combinations are then statistically ranked. RDI's system called MORPHO3 utilizes such a model (Ahmed, 2000). Such an approach achieves broad morphological coverage of the Arabic language, but required significant manual intervention to craft the rules.

4. Statistical Approach: This approach assumes that a word is constructed as a prefix-stem-suffix tuple. Given a word, the analyzer generates all possible segmentations by identifying all matching prefixes and suffixes from a table of prefixes and suffixes. Then given the possible segmentations, the trigram language model score is computed and the most likely segmentation is chosen. Two such systems are Sebawai, which was trained on the output of ALPNET (Darwish, 2002), and IBM-LM analyzer, which was trained on a manually segmented Arabic corpus from LDC and uses language modeling to improve analysis (Lee et al., 2003). Such approach achieves the broadest coverage with the least number of manually crafted rules, but likely requires a large number of training examples.

5. Light Stemming Based Approach: In this approach, leading and trailing letters in a word are removed if they match entries in lists of common prefixes and suffixes respectively. The advantage of this approach is that it requires no morphological processing and is hence very efficient. However, incorrect prefixes and suffixes are routinely removed. This approach was used to develop Arabic stemmers by (Aljlayl et al., 2001; Darwish and Oard, 2002A; Larkey et al., 2002).

Template	***Stem***	***Meaning***
CCC – فعل	ktb – كتاب	books or wrote
mCCwC – مفعول	mktwb – مكتوب	something written
CCAC – فعال	ktAb – كتاب	book
CCACyC – فعاعيل	ktAtyb – كتاتيب	Qur'an school
CACC – كاتب	kAtb – كاتب	writer
CcwC – فعول	ktwb – كتوب	skilled writer

Figure 2: Some templates to generate stems from roots with examples from the root (ktb – كتب)

Prefix	w – و	k – ك	f – ف l – ل	Al – ال	wAl – وال
Meaning	and	like	then to	the	and the

Figure 3: Some example prefixes and their meanings

Prefix	h – ه	k – ك	hm – km – كم هم	hA – ها	y – ي
Meaning	his	your (sg.)	their your (pl.)	her, its	my

Figure 4: Some example suffixes and their meanings

Prefix	ktb – كتاب	wktAbh – وكتابه	yktb – يكتب	ktAbhm – كتابهم	mktbp – مكتبة	AlkAtb – الكاتب
Meaning	book	and his book	he writes	their book	library	the writer

Figure 5: Some words that can be derived from the root ktb – كتب

Root	*Meaning*
Amn	peace or faith
Aym	two poor people
mAn	will he give support
ymn	Covenants
ymA	will they (fm.) point to

Figure 6: Possible roots for the word AymAn – ايمان along with meaning

- **Arabic Retrieval**

Due to the morphological complexity of the Arabic language, much research has focused on the effect of morphology on Arabic Information Retrieval (IR). The goal of morphology in IR is to conflate words of similar or related meanings. Several early studies suggested that indexing Arabic text using roots significantly increases

retrieval effectiveness over the use of words or stems (Abu-Salem et al., 1999; Al-Kharashi Evens, 1994; Hmeidi et al. 1997). However, all the studies used small test collections of only hundreds of documents and the morphology in many of the studies was done manually. Performing morphological analysis for Arabic IR using existing Arabic morphological analyzers, most of which use finite state transducers (Beesley et al., 1989; Beesley, 1996), is problematic for two reasons. First, they were designed to produce as many analyses as possible without indicating which analysis is most likely. This property of the analyzers complicates retrieval, because it introduces ambiguity in the indexing phase as well as the search phase of retrieval. Second, the use of finite state transducers inherently limits coverage, which the number of words that the analyzer can analyze, to the cases programmed into the transducers. Darwish attempted to solve this problem by developing a statistical morphological analyzer for Arabic called Sebawai that attempts to rank possible analyses to pick the most likely one (Darwish, 2002). He concluded that even with ranked analysis, morphological analysis did not yield statistically significant improvement over words in IR. A study by Aljlayl et al. (2001) on a large Arabic collection of 383,872 documents suggested that lightly stemmed words, where only common prefixes and suffixes are stripped from them, were perhaps better index term for Arabic. Similar studies by Darwish and Oard (2002B) and Larkey et al. (2002) also suggested that light stemming is indeed superior to morphological analysis in the context of IR. However, the shortcomings of morphology might be attributed to issues of coverage and correctness. Concerning coverage, analyzers typically fail to analyze Arabized or transliterated words, which may have prefixes and suffixes attached to them and are typically valuable in IR. As for correctness, the presence (or absence) of a prefix or suffix may significantly alter the analysis of a word. For example, for the word "Alksyr" is unambiguously analyzed to the root "ksr" and stem "ksyr." However, removing the prefix "Al" introduces an additional analysis, namely to the root "syr" and the stem "syr." Perhaps such ambiguity can be reduced by using the context in which the word is mentioned. For example, for the word "ksyr" in the sentence "sAr ksyr" (and he walked like), the letter "k" is likely to be a prefix. The problem of coverage is practically eliminated by light stemming. However, light stemming yields greater consistency without regard to correctness. Although consistency is more important for IR applications than linguistic correctness, perhaps improved correctness would naturally yield great consistency. Lee et al. (2003) developed IBM-LM, which adopted a trigram language model (LM) trained on a portion of the manually segmented LDC Arabic Treebank in developing an Arabic morphology system, which attempts to improve the coverage and linguistic correctness over existing statistical analyzers such as Sebawai (Darwish, 2002). IBM-LM's analyzer combined the trigram LM (to analyze a word within its context in the sentence) with a prefix-suffix filter (to eliminate illegal prefix suffix combinations, hence improving correctness) and unsupervised stem acquisition (to improve coverage). Lee et al. report a 2.9% error rate in analysis compared to 7.3% error reported by Darwish for Sebawai (Lee et al. 2003). A study by Darwish et al. (2005) suggested that using IBM-LM statistically significantly improved retrieval effectiveness.

The retrieval of OCR documents is discussed in section 3.

- **Arabic OCR and Orthography**

The goal of OCR is to transform a document image into character-coded text. The usual process is to automatically segment the document image into character images in the proper reading order using image analysis heuristics, apply an automatic classifier to determine the character codes that are most likely to correspond to each character image, and then to exploit sequential context (e.g., preceding and following characters and a list of possible words) to select the most likely character in each position (Darwish and Oard, 2002B). The character error rate can be influenced by reproduction quality (e.g., original documents are typically better than photocopies) (Baird, 2000), the resolution at which the document was scanned, and any mismatch between the instances on which the character image classifier was trained and the rendering of the characters in the printed document (Baird, 1993). Arabic OCR presents several challenges, including:

- Connected characters, which change shape depending on their position in the word, make the isolation of individual character images challenging.
- Word elongations (kashida) and special forms for certain letter combinations (ligatures such as lam-alef (ﻻ)) are often used in typed text (Trenkle et al., 2001), expanding the number of possibilities that the classifier must consider.
- 15 of the 28 Arabic letters include dots as an integral part of the character, and authors sometimes choose to additionally place diacritic marks on some letters. Dots and diacritic marks can easily be confused with speckle or dust, making detection of the correct character challenging.
- Due to the morphological complexity of Arabic, the number of legal words has been estimated to be 60 billion (Ahmed, 2000). This limits the value of sequential context somewhat, since it would be impractical to store a complete vocabulary of that size.

There are a number of commercial Arabic OCR systems, with Sakhr's Automatic Reader and Shonut's Omni Page being perhaps the most widely used (Kanungo et al., 1999).

III. Retrieval Evaluation Test Collections

A retrieval test collection is composed of a closed set of documents, a set of topics, which are at least 25 and are typically 50, and relevance judgment, which specify which documents are "relevant" to which topics and constitute the most expensive part of developing a test collection. Three methods have been used to produce the documents for test collections of OCR-degraded text:

• Systematically altering character-coded text using a character level confusion model that is trained on aligned pairs of character-coded and OCR-degraded texts. Large test collections can be efficiently produced using this technique by starting with an existing test collection for which topics and relevance judgments are already available. This avoids developing new relevance judgments. However, the degree of insight that can be obtained depends on the fidelity of the character confusion model, which might model some aspects of the process (e.g., character replacement) better than others (e.g., the effect of document skew during scanning). Harding, et al. used OCR errors that were simulated in this way to examine the effect of indexing character n-grams on retrieval from four English document collections (with 423 to 12,380 documents), finding that n-grams outperformed words (Harding et al., 1997).

• Typesetting character-coded text to produce a document image, optionally degrading the image to simulate speckle, page skew, bleed-through, varying illumination, and other factors (Baird, 2000; Kanungo, 1996), and then performing OCR. Although the operations on large document images adds some time to the process, large test collections can still be constructed relatively efficiently because it is possible to start with a collection for which topics and relevance judgments already exist. Baird used this technique to show that that retrieval effectiveness falls dramatically with increases in the character recognition error rate (Baird, 1993).

• Scanning a collection of printed documents, performing OCR, and then manually creating appropriate topics and relevance judgments. The size of a test collection created in this way will be limited by the resources available for the relevance judgment process. However, this technique can accurately model many aspects that may be present in real applications (e.g., unfamiliar fonts, damaged pages, and handwritten annotations). Taghva, et al. (1994) experimented with a 204-document English document image collection using this technique. The average length of the documents was 38 pages. He observed no significant effect of degradation on retrieval. Tseng and Oard experimented with different combinations of n-grams on a Chinese collection of 8,438 document images. The documents images were scanned from printed material. They observed that combinations of character 1-grams and character 2-grams performed best. Further, they reported that blind relevance feedback did not improve retrieval effectiveness (Tseng & Oard, 2001).

To develop relevance judgments, there are several methods reported in the literature. Some of the methods reported are:

• Exhaustive search: due to the required amount of manual processing, relevance judgments developed using this method was restricted to small collections and was reported not be feasible for larger collections (Jones & Van Rijsbergen, 1975).

• Pooling: pooling involves the participation of a "significantly" diverse set of systems in which the same topics are provided to all the systems and the top n retrieved results from each system are pooled and judged. This method is used by different evaluations such as the ones at TREC (Oard & Gey, 2002).

• Interactive Search and Judge (ISJ): ISJ technique, which was developed by Cormack et al., allows a judge to search the collection with different reformulations of topic expressions (Cormack et al., 1998). The judge continues to search until he/she is confident that all or most relevant documents are found.

• Iterative Search and Judge: in this technique, the judge is not required to manually reformulate topic expressions and the formulation is done automatically using relevance feedback. This method, which was developed and verified by Sanderson and Joho (2004), entails performing an initial search and then manually examining the top 100 retrieved documents. All the documents that are deemed relevant are used to reformulate the original queries. This process is repeated 5 times for each topic.

Three Arabic OCR test collections are mentioned in the literature. Darwish and Oard created a collection of 2,730 document images obtained from a medieval Arabic religious book called "the Sustenance of the Return." The document images were scanned at 300x300 dpi and two other versions of the collection were produced by down sampling the document images to 200x200 dpi and 200x100 dpi to simulate the fine and standard fax resolutions respectively. All versions of the collection were OCR'ed using Sakhr's Automatic Reader (version 4). Associated with the documents were a set of 25 topics for which the relevance judgments were created using exhaustive search. They reported that 3-grams and 4-grams are the best index terms for OCR degraded Arabic text (Darwish & Oard, 2002). Darwish created another collection from a large collection of 383,000 Arabic newswire articles by automatically degrading the collection using an OCR degradation model. Associated with the collection were a set of 50 topics and relevance judgments that were created using the pooling method (Darwish, 2003). Again character 3 and 4-grams were observed to the best index terms. The last collection was created by Abdelsapor et al. by randomly picking approximately 25

pages from 1,378 Arabic books from Bibliotheca Alexandrina (BA) forming a set of 34,651 printed documents (Abdelsapor et al., 2006). The books cover a variety of topics including historical, philosophical, cultural, and political subjects and the printing dates of the books range from the early 1920's to the present. The documents were converted to document images by scanning them in black and white at 300x300 dpi. The scanning was done as a part of the Million Book Project in which the BA is responsible for scanning 75,000 Arabic documents. The document images were subsequently OCR'ed using Sakhr's Automatic reader (version 6). The OCR text had character error rates ranging between 1% and 21% for different books. The fonts used in the books were divided into 12 different font classes, which correspond to the most popular fonts used in print, and a 13th class containing rare fonts. Associated with the collection are a set of 20 topics that were created using the iterative search and judge method. Darwish and Emam (2005) reported that blind relevance feedback did not benefit retrieval on this collection.

IV. OCR Error Handling

- **OCR Error Correction**

Much research has been done to correct recognition errors in OCR-degraded collections. There are two main categories of determining how to correct these errors. They are word-level and passage-level post-OCR processing. Some of the kinds of word level post-processing include the use of dictionary lookup, probabilistic relaxation, character and word n-gram frequency analysis (Hong, 1995), and morphological analysis. Passage-level post-processing techniques include the use of word n-grams, word collocations, grammar, conceptual closeness, passage level word clustering, linguistic context, and visual context. The following introduces some of the error correction techniques.

- Dictionary Lookup: Dictionary Lookup, which is the basis for the correction reported in this paper, is used to compare recognized words with words in a term list (Hong, 1995; Tseng and Oard, 2001). If a word is found in the dictionary, then it is considered correct. Otherwise, a checker attempts to find a dictionary word that might be the correct spelling of the misrecognized word. Jurafsky and Martin illustrate the use of a noisy channel model to find the correct spelling of misspelled or misrecognized words (Jurafsky and Martin, 2000). The model assumes that text errors are due to edit operations namely insertions, deletions, and substitutions. Given two words, the number of edit operations required to transform one of the words to the other is called the Levenshtein edit distance (Baeza-Yates and Navarro, 1996). To capture the probabilities associated with different edit operations, confusion matrices are employed. Another source of evidence is the relative probabilities that candidate word corrections would be observed. These probabilities can be obtained using word frequency in text corpus (Jurafsky and Martin, 2000; Lu et al., 1999). However, the dictionary lookup approach has the following problems (Hong, 1995):

a) A correctly recognized word might not be in the dictionary. This problem could surface if the dictionary is small, if the correct word is an acronym or a named entity that would not normally appear in a dictionary, or if the language being recognized is morphologically complex. In a morphological complex language such as Arabic, German, and Turkish the number of valid word surface forms is arbitrarily large which complicates building dictionaries for spell checking.

b) A word that is misrecognized is in the dictionary. An example of that is the recognition of the word "tear" instead of "fear". This problem is particularly acute in a language such as Arabic where a large fraction of three letters sequences are valid words. In handling this problem, the error correction reported in this paper does not assume that a word is correct because it exists in the dictionary of possible words and assumes that it could have been generated from another correct word.

- Character N-Grams: Character n-grams maybe used alone or in combination with dictionary lookup (Lu et al., 1999). The premise for using n-grams is that some letter sequences are more common than others and other letter sequences are rare or impossible. For example, the trigram "xzx" is rare in the English language, while the trigram "ies" is common. Using this method, an unusual sequence of letters can point to the position of an error in a misrecognized word. This technique is employed by BBN's Arabic OCR system (Lu et al., 1999).
- Using Morphology: Many morphologically complex languages, such as Arabic, Swedish, Finnish, Turkish, and German, have enormous numbers of possible words. Accounting for and listing all the possible words is not feasible for purposes of error correction. Domeij proposed a method to build a spell checker that utilizes a stem lists and orthographic rules, which govern how a word is written, and morphotactic rules, which govern how morphemes (building blocks of meanings) are allowed to combine, to accept legal combinations of stems (Domeij et al., 1994). By breaking up compound words, dictionary lookup can be applied to individual constituent stems. Similar work was done for Turkish in which an error tolerant finite state recognizer was employed (Oflazer, 1996). The finite state recognizer tolerated a maximum number of edit operations away from correctly spelled candidate words. This approach was initially developed to perform morphological analysis for Turkish and was extended to perform spelling correction. The techniques used for

Swedish and Turkish can potentially be applied to Arabic. Much work has been done on Arabic morphology and can be potentially extended for spelling correction.

- Word Clustering: Another approach tries to cluster different spellings of a word based on a weighted Levenshtein edit distance. The insight is that an important word, specially acronyms and named-entities, are likely to appear more than once in a passage. Taghva et al. (2001) described an English recognizer that identifies acronyms and named-entities, clusters them, and then treats the words in each cluster as one word. Applying this technique for Arabic requires accounting for morphology, because prefixes or suffixes might be affixed to instances of named entities. DeRoeck and Al-Fares (2000) introduced a clustering technique tolerant of Arabic's complex morphology. Perhaps the technique can be modified to make it tolerant of errors.
- Using Grammar: In this approach, a passage containing spelling errors is parsed based on a language specific grammar. In a system described by Agirre, an English grammar was used to parse sentences with spelling mistakes (Agirre et al., 1998). Parsing such sentences gives clues to the expected part of speech of the word that should replace the misspelled word. Thus candidates produced by the spell checker can be filtered. Applying this technique to Arabic might prove challenging because the work on Arabic parsing has been very limited (Moussa et al., 2003).
- Word N-Grams (Language Modeling): A Word n-gram is a sequence of n consecutive words in text. The word n-gram technique is a flexible method that can be used to calculate the likelihood that a word sequence would appear (Tillenius, 1996). Using this method, the candidate correction of a misspelled word might be successfully picked. For example, in the sentence "I bought a peece of land," the possible corrections for the word peece might be "piece" and "peace". However, using the n-gram method will likely indicate that the word trigram "piece of land" is much more likely than the trigram "peace of land." Thus the word "piece" is a more likely correction than "peace".

Dictionary lookup in combination with language modeling was successfully applied to Arabic with more than 60% reduction in word error rate (Magdy and Darwish, 2006). The effect of correction on retrieval was examined and was shown to correspond to the OCR error reduction (Magdy and Darwish, 2006).

- **Query Garbling with Weighted Structured Queries**

Query garbling attempts to map clean queries into the degraded representation of the documents using an OCR degradation model, which can produce possible ways a character or a character segment might have be corrupted by the OCR process. OCR might misrecognize a word in many different ways. For example, the word "eat" may be recognized as "eat", "cat", "sat", "eal", … etc. A problem that follows directly from that is which replacement should be used in the IR application. An approach to overcoming the problem is to conflate possible replacements via the use of structured queries. InQuery (Allan et al., 2000), PSE (Darwish, 2003), and Indri (Strohman and Croft, 2006), implement structured queries by treating all possible replacements as synonyms. The implementation computes a new joint term frequency and a joint document frequency for the possible replacements as follows (Darwish and Oard, 2003):

$$TF_j(Q_i) = \sum_{\{k|D_k \in T(Q_i)\}} TF_j(D_k)$$

$$DF(Q_i) = \left| \bigcup_{\{k|D_k \in T(Q_i)\}} \{d \mid D_k \in d\} \right|$$

where Q_i is a query term, D_k is a document term, $TF_j(Q_i)$ is the term frequency of Q_i in document j, $DF(Q_i)$ is the number of documents that contain Q_i, d is a document, and $T_j(Q_i)$ is the set of known replacements (in this case, translations) for the term D_k.

This represents a very cautious strategy in which a high DF for any replacement will result in a high DF (and thus a low weight) for new joint DF of that query term. Retrieval results are then dominated by query terms that have no "unsafe" (very common) replacements. For example, the Arabic query term "علي" can either mean "on" or the proper name "Ali." If "Ali" appears in few documents but "on" appears in many, the DF equation will treat "علي" as if it were at least as common as "on." When there is not a large disparity in DF, structured queries have a kind of query expansion effect. For example, the Arabic word "خبز" can be translated as "bread" or "bake," and structured queries would (with proper stemming) reward an occurrence of "bake bread."

- This risks a somewhat counterintuitive result. Using a translation resource with improved coverage of rare translations could actually harm retrieval effectiveness. To illustrate this, consider a case in which the query term "ax" has a 99.9% probability of being recognized as "ax," but a 0.1% probability of being misrecognized as the common term "the." In such a case, the common term leads to a high joint DF, effectively diminishing the value of the original query term.
- To overcome this problem, Darwish (2003) introduced weighted structured queries which incorporate translation probabilities in structured queries as follows:

$$TF_j(Q_i) = \sum_{\{k|D_k \in T(Q_i)\}} \left[TF_j(D_k) \times wt(D_k) \right]$$

$$DF_j(Q_i) = \sum_{\{k|D_k \in T(Q_i)\}} \left[DF_j(D_k) \times wt(D_k) \right]$$

where $wt(D_k)$ is the translation probability of the replacement. Query garbling in conjunction with weighted structured queries was shown to statistically significantly improve retrieval effectiveness for Arabic (Darwish and Oard, 2003).

Figure 7: Sample page

من ناحيتى المراقبة والنيران على السهل الساحلى.
وطرق الاقتراب التى تسلكها أى قوات عربية من ناحية الشرتى تنح ر دى طرق خمسة أهمها الثلاثة التالية:
ا- الطريق ا لأول وهو ا لأقصر من بغداد- هـ 2233 " المفرتى.، أو ا لانحراف إلى
الرطبة قبل 3أ3 دمشق الأردن،
محاور ا لارابى من العرا إلى سوريا وا لأردد
2- الطريق الاثانى من بغداد " أبو كمال- باليرا- دمشق- ا لأردد.
3- الطريق الثالست وهو الأطول من بغ داد- الموصل- دير الزور- حملرو- دمشق- ا لأردن " 68

Figure 8: Sample OCR output

V. Displaying Search Results

Due to the fact that OCR produces many error in the recognized text, displaying the original document images in the results as opposed to the OCR text is often more desirable, because the user would see the document in the original formatting without the OCR recognition errors. To display OCR'ed document images, the so-called image-over-text technology is often used to overlay the document images over the OCR output. However, displaying the Arabic document images corresponding to search results with proper highlighting of users' search words can be challenging due to the frequent OCR errors and Arabic's complex morphology and orthography. Figure 7 and Figure 8 show a sample document image and the corresponding OCR from the BA collection.

As mentioned in the error correction section, word clustering of morphologically similar Arabic words has been demonstrated by DeRoeck and Al-Fares (2000), and word clustering of misrecognized versions of the same word was done by Taghva and Stofsky (2001). To the best knowledge of the authors, there is no published work on clustering morphologically similar Arabic words that are misrecognized. Perhaps the technique reported on by DeRoeck and Al-Fares (2000) can be modified to make it tolerant of errors. A system reported on by Abdelsapor et al. (2006) performs highlighting of morphological variants based on light stemming in an image-over-text display system without regard to OCR errors. Figure 9 provides a screenshot from the system.

VI. Future Directions

Some of the possible future directions include:

- Error tolerant morphological analysis. This can be helpful in improving error correction of OCR degraded documents and can help in clustering OCR degraded and morphological similar words for highlighting query terms in search results.
- Improved retrieval algorithms. Darwish demonstrated that weighted structured queries can improve retrieval effectiveness significantly. Similarly, Singhal et al. (1996) have shown that for English, byte length normalization is more robust to character recognition errors than the cosine normalization usually used in vector space retrieval systems, and Tseng and Oard (2001) have seen similar results for Chinese. It is likely that the improved retrieval algorithms that are tuned specifically to OCR degraded text can improve retrieval effectiveness.
- Larger test collections. The existing OCR degraded Arabic retrieval test collections with real OCR output remain small. It is well known that the size of retrieval test collection can have a significant impact the effectiveness of different techniques. Therefore, the creation of larger test collection (with at least hundreds of thousands of documents) is instrumental to continued effective research in the area.
- Automatic layout analysis. In previously reported work, the document images were segmented manually, and no special processing was required to determine the appropriate reading order. Automatic layout analysis

will, however, be needed in many practical applications (e.g., searching printed newspapers).

- Image enhancement for low-resolution applications. Faxes, video captions, and scene text in video have significantly lower resolution than ordinary scanned documents, and video applications often also include unusual background characteristics. Image processing techniques such as mathematical morphology and multi-frame integration can be helpful in such cases.
- Deploying OCR documents in other retrieval and natural language applications. Thus far, most of the reported work focused on the search and retrieval of document images. However, further processing of OCR documents such as information extraction and machine translation maybe required. The degradation in text poses unique challenges with plenty of room for contribution.

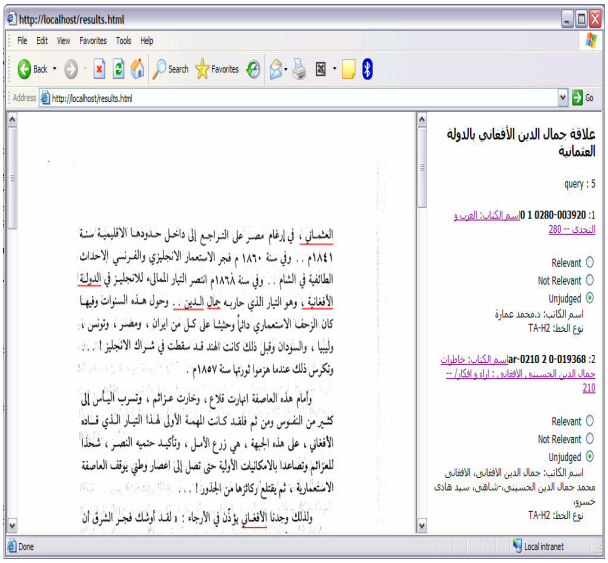

Figure 9: Web-based system for searching, displaying, and judging document images

VII. Conclusion

This paper presented a survey of some of the recent research aiming to improve the retrieval effectiveness and visualization of OCR'ed documents in general and Arabic OCR documents in specific. The paper examined issue pertaining to document handling including error-handling and orthographic and morphological processing. Further, it listed some of the available research resources and explored futures directions to further improve the process of retrieval and visualization.

Acknowledgement

The authors wish to acknowledge the assistance and valuable comments of Amgad Madkour.

References

Abdelsapor, A., N. Adly, K. Darwish, O. Emam, M. Nagi. Building a Heterogeneous Information Retrieval Collection of Printed Arabic Documents. LREC 2006.

Abdul-Al-Aal, Abdul-Monem, An-Nahw Ashamil. Maktabat Annahda Al-Masriya, Cairo, Egypt, 1987.

Abu-Salem, H., M. Al-Omari, and M. Evens. Stemming Methodologies over Individual Query Words for Arabic Information Retrieval. JASIS, 1999. 50(6): p. 524-529.

Agirre, E., K. Gojenola, K. Sarasola, and A. Voutilainen. Towards a Single Proposal in Spelling Correction. In COLING-ACL'98, 1998.

Ahmed, M. A large-scale computational processor of the Arabic morphology, and Applications. Master's Thesis, Faculty of Engineering, Cairo University, Cairo, Egypt, 2000.

Allan, J., Connell, M.,Croft, W.B., Feng,F., Fisher, D. and Li, X. INQUERY and TREC-9. In TREC-9, pp. 551-577, 2000.

Aljlayl, M., S. Beitzel, E. Jensen, A. Chowdhury, D. Holmes, M. Lee, D. Grossman, and O. Frieder. IIT at TREC-10. TREC-2001, 2001.

Al-Kharashi, I. and M. Evens. Comparing words, stems, and roots as index terms in an Arabic information retrieval system. JASIS. 45 (8): 548-560, 1994.

Antworth, E. PC-KIMMO: a two-level processor for morphological analysis. In Occasional Publications in Academic Computing. 1990. Dallas, TX: Summer Institute of Linguistics.

Baeza-Yates, R. and G. Navarro. A Faster Algorithm for Approximate String Matching. in Combinatorial Pattern Matching (CPM'96), Springer-Verlag LNCS. 1996.

Baird, H. Document image defects models and their uses. Proceedings of the Second International Conference on Document Analysis and Recognition (ICDAR), 62-67, 1993.

Baird, H. State of the art of document image degradation modelling. Proceedings of the 4th IAPR Workshop on Document Analysis Systems (DAS 2000), 2000.

Beesley, K., T. Buckwalter, and S. Newton. Two-Level Finite-State Analysis of Arabic Morphology. In the Seminar on Bilingual Computing in Arabic and English. 1989. Cambridge, England.

Beesley, K. Arabic Finite-State Morphological Analysis and Generation. In COLING-96. 1996.

Cormack, G., C. Palmer, and C. Clarke. Efficient Construction of Large Test Collections. In the Proceedings of the 21st ACM SIGIR Conference, 282-289, 1998.

Darwish, K. Building a shallow morphological analyzer in one day. ACL 2002 Workshop on Computational Approaches to Semitic Languages, July 11, 2002.

Darwish, K. and O. Emam. The Effect of Blind Relevance Feedback on a New Arabic OCR Degraded Text Collection. In International Conference on Machine Intelligence: Special Session on Arabic Document Image Analysis, 2005.

Darwish, K. and D. Oard. CLIR Experiments at Maryland for TREC 2002: Evidence Combination for Arabic-English Retrieval. In TREC, 2002A. Gaithersburg, MD.

Darwish, K., D. Oard, Term selection for searching printed Arabic. In the Proceedings of the 25th ACM SIGIR Conference, page 261 - 268, 2002B.

Darwish, K. and D. Oard. Probabilistic structured query methods. SIGIR 2003: 338-344.

Darwish, K., H. Hassan, O. Emam. Examining the Effect of Improved Context Sensitive Morphology on Arabic Information Retrieval. Proceedings of the ACL

Workshop on Computational Approaches to Semitic Languages, 2005. pages 25-30.
De Roeck, A. and W. Al-Fares. A Morphologically Sensitive Clustering Algorithm for Identifying Arabic Roots. Proceedings of the 38th Annual Meeting of the ACL, Hong Kong, 2000.
Domeij, R., J. Hollman, V. Kann, Detection of spelling errors in Swedish not using a word list en clair. Journal of Quantitative Linguistics, 1994. 1: p. 195-201.
Goldsmith, John, Unsupervised Learning of the Morphology of a Natural Language. http://humanities.uchicago.edu/faculty/goldsmith/, 2000.
Harding, S., W. Croft, and C. Weir. Probabilistic retrieval of OCR degraded text using n-grams. European Conference on Digital Libraries, 1997
Hmeidi, I., G. Kanaan, and M. Evens. Design and implementation of automatic indexing for information retrieval with Arabic documents. JASIS. 48 (10): 867-881, 1997.
Hong, T., Degraded Text Recognition Using Visual and Linguistic Context, in Computer Science Department. 1995, SUNY Buffalo: Buffalo.
Ibn Manzour, Lisan Al-Arab. www.muhaddith.org.
Jones, K. S. and C. J. Van Rijsbergen. Report on the need for and previous of 'ideal' test collection, TR #365, University Computer Laboratory, Cambridge, 1975.
Jurafsky, D. and J. Martin, Speech and Language Processing. 2000: Prentice Hall.
Kanungo, T. Document degradation models and methodology for degradation model validation. Ph.D. Thesis, Electrical Engineering Department, University of Washington, 1996.
Kanungo, T., G. Marton, and O. Bulbul. OmniPage vs. Sakhr: paired model evaluation of two Arabic OCR products. Proceedings of SPIE Conference on Document Recognition and Retrieval (VI), Vol. 3651, San Jose, California, Jan. 27-28, 1999.
Kiraz, G. Arabic Computational Morphology in the West. In The 6th International Conference and Exhibition on Multi-lingual Computing. 1998. Cambridge.
Koskenniemi, K. Two Level Morphology: A General Computational Model for Word-form Recognition and Production. 1983, Department of General Linguistics, University of Helsinki.
Larkey, L., L. Ballesteros, and M. Connell. Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. In SIGIR, 2002, Tampere, Finland. pp. 275-282
Lee, Y., K. Papineni, S. Roukos, O. Emam, and H. Hassan. Language Model Based Arabic Word Segmentation. ACL 2003: 399-406.
Lu, Z., I. Bazzi, A. Kornai, J. Makhoul, P. Natarajan, and R. Schwartz. A Robust, Language-Independent OCR System. in The 27th AIPR Workshop: Advances in Computer Assisted Recognition, SPIE. 1999.
Magdy, W. and K. Darwish. Word-Based Correction for Retrieval of Arabic OCR Degraded Documents. To appear in SPIRE 2006.
Moussa B., M. Maamouri, H. Jin, A. Bies, X. Ma. Arabic Treebank: Part 1 - 10Kword English Translation. Linguistic Data Consortium (2003).
Oard, D. and F. Gey. The TREC-2002 Arabic/English CLIR Track. TREC-2002, 2002.
Oflazer, K., Error-Tolerant Finite State Recognition with Applications to Morphological Analysis and Spelling Correction. Computational Linguistics, 1996. 22(1): p. 73-90.
Sanderson, M. and H. Joho. Forming Test Collection with no System Pooling. In the Proceedings of the 27th ACM SIGIR Conference, page 33-40, 2004
Singhal, A., G. Salton, and C. Buckley. Length normalization in degraded text collections. Proceedings of 5th Annual Symposium on Document Analysis and Information Retrieval, 149-162, April 15-17, 1996.
Strohman, T. and Croft, W.B. Low Latency Index Maintenance in Indri. In the Proceedings of the Open Source Information Retrieval Workshop (OSIR) 2006, pp. 7-11
Taghva, K., J. Borasack, A. Condit, and J. Gilbreth. Results and implications of the noisy data projects. Technical Report 94-01, Information Science Research Institute, University of Nevada, Las Vegas, 1994.
Taghva, K. and Eric Stofsky. OCRSpell: an interactive spelling correction system for OCR errors in text. IJDAR 3(3): 125-137, 2001.
Tillenius, M., Efficient generation and ranking of spelling error corrections. 1996, NADA.
Trenkle, J., A. Gillies, E. Erlandson, S. Schlosser, and S. Cavin. Advances in Arabic text recognition. Proceeding of Symposium on Document Image Understanding Technology, Columbia, Maryland, April 23-25, 2001.
Tseng, Y. and D. Oard. Document image retrieval techniques for Chinese. Proceeding of Symposium on Document Image Understanding Technology, Columbia, Maryland, April 23-25, 2001.