Inf Retrieval (2008) 11:405–425
DOI 10.1007/s10791-008-9055-y

Effect of OCR error correction on Arabic retrieval

Walid Magdy · Kareem Darwish

Received: 16 August 2006 / Accepted: 13 February 2008 / Published online: 11 March 2008
© Springer Science+Business Media, LLC 2008

Abstract Arabic documents that are available only in print continue to be ubiquitous and they can be scanned and subsequently OCR'ed to ease their retrieval. This paper explores the effect of context-based OCR correction on the effectiveness of retrieving Arabic OCR documents using different index terms. Different OCR correction techniques based on language modeling with different correction abilities were tested on real OCR and synthetic OCR degradation. Results show that the reduction of word error rates needs to pass a certain limit to get a noticeable effect on retrieval. If only moderate error reduction is available, then using short character n-gram for retrieval without error correction is not a bad strategy. Word-based correction in conjunction with language modeling had a statistically significant impact on retrieval even for character 3-grams, which are known to be among the best index terms for OCR degraded Arabic text. Further, using a sufficiently large language model for correction can minimize the need for morphologically sensitive error correction.

Keywords OCR · Language modeling · Information retrieval · Error correction

1 Introduction

Since the advent of the printing press in the fifteenth century, the amount of printed text has grown overwhelmingly. Although a great deal of text is now generated in electronic

The work for this paper was performed while both authors were at the IBM Technology Development Center, Cairo, Egypt.

W. Magdy (✉) · K. Darwish
Cairo Microsoft Innovation Center, Smart Village—Bldg B115, Km 28, Cairo-Alexandria Desert Rd, Abou Rawash, Egypt
e-mail: i-wmagdy@microsoft.com

K. Darwish
e-mail: kareemd@microsoft.com

character-coded formats (HTML, word processor files, etc.), many documents, available only in print, remain important. This is due in part to the existence of large collections of legacy documents available only in print, and in part because printed text remains an important distribution channel that can effectively deliver information without the technical infrastructure that is required to deliver character-coded text. These factors are particularly important for Arabic, which is widely used in places where the installed computer infrastructure is often quite limited. Printed documents can be browsed and indexed for retrieval relatively easily in limited quantities, but effective access to the contents of large collections requires some form of automation.

One such form of automation is to scan the documents (to produce document images) and subsequently perform OCR on the document images to convert them to text. Typically, the OCR process introduces errors in the text representation of the document images. The error level is affected by the quality of paper, printing, and scanning. The introduced errors are more pronounced in Arabic OCR (as compared to English) due to some of the orthographic and morphological features of Arabic. For example, the dataset reported-on in this paper, which is based on a fairly clean book that has been published 25 years ago and scanned at 300 300 dpi, has a word error rate of approximately 39%. Even higher word error rates were observed by the authors in their collaborative work with the Library of Alexandria on their Arabic digitization project. This is significantly higher than the average word error rate for the out-of-copyright English books (typically 100 years old) that are available through the Internet Archive. Orthographically, Arabic characters are connected and change shape depending on their position in a word. As for morphological complexity, Arabic allows the insertion of infixes to form words and the attachment of prefixes and suffixes that include pronouns, determiners, number markers (singular, dual, and plural), conjunctions, etc.

The introduced errors adversely affect retrieval effectiveness of OCR'ed documents. This paper examines the effect of word-based post-OCR error correction in conjunction with language modeling on Arabic retrieval effectiveness using different index terms on two collections of degraded Arabic documents. The correction uses a character segment based noisy channel model and language modeling to correct OCR errors. The paper compares the effect on retrieval effectiveness of performing "good" error correction and performing "moderate" error correction with and without language modeling, respectively. The effect of error correction strategies is also investigated when using different index terms, namely word surface forms, morphological variants, and sub-word character n-gram sequences. The paper provides suggestions on which error correction strategies and index terms to use or not use under different conditions to improve retrieval effectiveness. The paper will be organized as follows: Sect. 2 provides background information on Arabic OCR and retrieval along with OCR error correction; Sect. 3 presents the experimental setup; Sect. 4 reports and discusses experimental results; and Sect. 5 concludes the paper and provides possible future directions.

2 Background

2.1 Arabic morphology and OCR

The goal of OCR is to transform a document image into character-coded text. The usual process is to automatically segment a document image into character images in the proper reading order using image analysis heuristics, apply an automatic classifier to determine

the character codes that most likely correspond to each character image, and then exploit sequential context (e.g., preceding and following characters and a list of possible words) to select the most likely character in each position. The character error rate can be influenced by reproduction quality (e.g., original documents are typically better than photocopies), the resolution at which a 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. Arabic OCR presents several challenges, including:

Arabic's cursive script in which most characters are connected and their shape vary with position in the word. Further, multiple connected characters may resemble other single characters or combinations of characters. For example, the letter "??" (sheen) may resemble "???" (noon—ta combination).

The optional use of word elongations and ligatures, which are special forms of certain letter sequences.

The presence of dots in 15 of the 28 to distinguish between different letters and the optional use of diacritic which can be confused with dirt, dust, and speckle (Darwish and Oard 2002a, b). The orthographic features of Arabic lead to some characters being more prone to OCR errors than others.

The morphological complexity of Arabic, which results in an estimated 60 billion possible surface forms, complicates dictionary-based error correction. A surface form is any group of consecutive characters in the text that may include a word with the attachment of a conjunction, a determiner, and/or a pronoun. Arabic words are built from a closed set of about 10,000 root forms that typically contain 3 characters, although 4-character roots are not uncommon, and some 5-character roots do exist. Arabic stems are derived from these root forms by fitting the root letters into a small set of regular patterns, which sometimes includes addition of "infix" characters within the root (Ahmed 2000). Thus, stems with no infixes are identical to roots. Further attachment of prefixes and suffixes that include determiners, conjunctions, pronouns, and grammatical markers produces a word surface form. Again, a word can be identical to a stem if no prefixes or suffixes are attached.

There are a number of commercial Arabic OCR systems, with Sakhr's Automatic Reader and Shonut's OmniPage being perhaps the most widely used (Kanungo et al. 1999a, 1997). Most Arabic OCR systems segment characters (Gillies et al. 1999; Hassibi 1994a, b; Kanungo et al. 1997), while a few opted to recognize words without segmenting characters (Allam 1995; Lu et al. 1999). A system developed by BBN avoids character segmentation by dividing lines into slender vertical frames (and frames into cells) and uses an HMM recognizer to recognize character sequences (Lu et al. 1999).

2.2 OCR degraded text retrieval

Retrieval of OCR degraded text documents has been reported on for many languages, including English (Harding et al. 1997; Kantor and Voorhees 1996; Taghva et al. 1994a, b, 1995, 1996b); Chinese (Tseng and Oard 2001); and Arabic (Darwish and Oard 2002a, b).

For English, Doermann (1997) reports that retrieval effectiveness decrease significantly for OCR'ed documents with an error rate at some point between 5% and 20%. Taghva reported experiments which involved using English collections with documents ranging in number between 204 and 674 documents that were about 38 pages long on average (Taghva et al. 1994b, 1995). The documents were scanned and OCR'd. His results show

negligible decline in retrieval effectiveness due to OCR errors. Taghva's work was criticized for being done on very small collections of very long documents (Tseng and Oard 2001). Small collections might not behave like larger ones, and thus they might not be reflective of real life applications in which retrieval from a large number of documents is required (Harman 1992). Similar results for English were reported by Smith (1990) in which he reported no significant drop in retrieval effectiveness with the introduction of simulated OCR degradation in which characters were randomly replaced by a symbol indicating failure to recognize. These results contradict other studies in which retrieval effectiveness deteriorated dramatically with the increase in degradation. Hawking reported a significant drop in retrieval effectiveness at a 5% character error rate on the TREC-4 "confusion track" (Hawking 1996). In the TREC-4 confusion track, approximately 50,000 English documents from the federal registry were degraded by applying random edit operations to random characters in the documents (Kantor and Voorhees 1996). The contradiction might be due to the degradation method, the size of the collection, the size of the documents, or a combination of these factors. In general retrieval effectiveness is adversely affected by the increase in degradation and decrease in redundancy of search terms in the documents (Doermann 1998).

Several studies reported the results of using n-grams. A study by Harding et al. (1997), compared the use of different length n-grams to words on 4 English collections, in which errors artificially introduced. The documents were degraded iteratively using a model of OCR degradation until retrieval effectiveness of using words as index terms started to significantly deteriorate. The error rate in the documents was unknown. For n-grams, a combination of 2 and 3 grams and a combination of 2, 3, 4, and 5 grams were compared to words. Their results show that n-gram indexing consistently outperformed word indexing, and combining more n-grams was better than combining fewer. In another study by Tseng and Oard, they experimented with different combinations of n-grams on a Chinese collection of 8,438 document images and 30 Chinese queries (Tseng and Oard 2001). Although ground-truth was not available for the image collection to conclude the effect of degradation on retrieval effectiveness, the effectiveness of different index terms were compared. They experimented with unigrams, bigrams, and a combination of both. Chinese words were not segmented and bigrams crossed word boundaries. The results of the experiments show that a combination of unigrams and bigrams consistently and significantly outperform character bigrams, which in turn consistently and significantly outperforms character unigrams.

For Arabic, Darwish and Oard (2002a, b) reported that character 3-gram and 4-grams were the best index terms for searching OCR degraded text. They conducted their experiments on a small collection of 2,730 scanned documents.

In general, blind relevance feedback does not help for the retrieval of OCR degraded documents (Darwish and Emam 2005; Lam-Adesina and Jones 2006; Taghva et al. 1996a, b; Tseng and Oard 2001).

2.3 Building an OCR degraded collection

To build an OCR-degraded test collection, there are three common approaches:

1. Printed document domain: which involves building a collection by scanning printed documents and performing OCR. This approach is most desirable because the errors in the text are due to real OCR degradation and not a model of the degradation. However,

building large test collections of several hundred thousand documents with a set of topics and relevance judgments can be very expensive. Therefore, the collections reported in the literature were all small. One such collection is a Chinese collection of 8,438 documents which was developed by Tseng and Oard (2001). The documents in Tseng's collection varied widely in their degradation level and there was no accurately character-coded version (OCR ground truth) for the collection. Abdelsapor et al. (2006) developed a collection of Arabic OCR'ed document images by randomly picking approximately 25 pages from 1,378 Arabic books from Bibliotheca Alexandrina (BA) forming a set of 34,651 printed documents. Associated with the collection are set of 25 topics that were developed using an iterative search and judge method (Sanderson and Joho 2004). 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 1920s to the present. Again, no ground truth is available for the collection. Having ground truth helps show the effect of degradation on retrieval. Developing OCR ground truth is typically laborious, involving either correction of OCR errors in the OCR'd version of the collection or manual re-entry of the collection's text. Lam-Adesina and Jones (2006) reported on a collection that they developed from the Spoken Document Retrieval (SDR) track collection. The stories in the collection were printed using different formats and fonts, and the resulting hardcopies were scanned and OCR'ed. Associated with the collection of 21,759 news stories are rough or closed-caption quality transcripts and 50 topics that were developed for the SDR track (Lam-Adesina and Jones 2006). Darwish and Oard (2003) report on a small collection of 2,730 documents of scanned and OCR'ed document images for which ground truth exists. The collection is used in this paper and is thoroughly described later.

2. Image domain: which involves building a collection by synthesizing document images from a preexisting non-degraded collection, degrading the document images, and performing OCR on them. Synthesizing document images is done by typesetting the text into an image (Doermann and Yao 1995). To degrade document images, different document degradation models were developed (Baird 1990, 1993, 2000; Doermann and Yao 1995; Kanungo 1996; Kanungo et al. 1995, 1993). The models parameterize different aspects of the document images such as font size, page skew, horizontal and vertical offset, horizontal and vertical scaling, blur, resolution, pixel jitter, and sensitivity. With degradation modeling, document image collections of varying degradation levels with corresponding ground truth can be developed automatically. To verify suitability of the generated document image collections for further OCR research, tests were developed. It is claimed that a degradation model is valid if the confusion matrices that result from automatically degraded documents are similar to the ones that result from real documents (Kanungo and Haralick 1998; Li et al. 1997; Lopresti and Zhou 1994; Nagy 1994). However, Kanungo and Haralick (1998) criticized their approach on the basis that OCR algorithms might filter certain features in either the synthetic or the real documents making both produce similar confusion matrices. Kanungo et al. (2000) instead proposed a probabilistic method that focuses on the correctness of the model in isolation of OCR algorithms. The advantage of this approach for creating OCR-degraded collections is that it is inexpensive, the degradation level can be tuned, and OCR ground truth is automatically available. Although OCR researchers prefer real document images and real OCR output (Tseng and Oard 2001), the suitability of this approach for IR experimentation needs to be verified.

3. Text domain: building a collection by synthesizing OCR degradation. This approach has the advantage of being able to use a preexisting non-degraded collection with its topics and relevance judgments to rapidly build a new degraded collection. This approach was used in developing many degraded text collections (Croft et al. 1994; Harding et al. 1997; Harman 1995; Smith 1990; Taghva et al. 1996a). The degradation models ranged between ones that attempted to accurately model OCR degradation (Harding et al. 1997) to ones that randomly introduced errors (Smith 1990). Mittendorf and Schäuble (2000) argued that using synthetic OCR degradation do not lead to the variations of recognition probabilities, which affect ranking permutations the most, that are observed in real OCR degradation. Darwish (2003) introduced formal tests to verify that the modeled OCR-degradation has similar effect on retrieval as real OCR-degradation.

2.4 OCR error correction

Much research has been done to correct recognition errors in OCR-degraded collections. There are two main categories of approaches to correct these errors, namely 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. The work in this paper shows that this problem can be overcome ever for Arabic if the lookup dictionary is large.

(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.

Mittendorf and Schäuble (2000) argue that using dictionary lookup can be harmful to retrieval effectiveness because if a correctly recognized token does not exists in the dictionary it is likely to have a high inverse document frequency, hence a valuable search term, and the correction process might eliminate it. In effect a proper correction may be eliminated because the ranking formula did not rank it as the best correction.

Character n-grams: character n-grams maybe used alone or in combination with dictionary lookup (Lu et al. 1999; Taghva et al. 1994a). 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). The technique can be particularly helpful in limiting the number of candidate corrections and hence making correction more efficient.

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 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. This paper tests correction without accounting for morphology.

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 described an English recognizer that identifies acronyms and named-entities, clusters them, and then treats the words in each cluster as one word (Taghva et al. 1994a). Applying this technique for Arabic requires accounting for morphology, because prefixes or suffixes might be affixed to instances of named entities. DeRoeck introduced a clustering technique tolerant of Arabic's complex morphology (De Roeck and Al-Fares 2000). 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 (Magdy and Darwish 2006; 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". The work in this paper uses language modeling and does not automatically assume that a word is correct if it exists in the dictionary. This paper builds on the work of (Magdy and Darwish 2006) to ascertain the effect of error correction on retrieval effectiveness.

Multi-OCR output fusion: in this approach multiple OCR systems, which typically have different classification engines with different training data, are used to recognize the same text. The output of the different OCR systems is then fused by picking the most likely recognized sequence of tokens using language modeling (Magdy et al. 2007). This is akin to using classifier ensembles.

2.5 Arabic information retrieval

Most early studies of character-coded Arabic text retrieval relied on relatively small test collections (Abu-Salem et al. 1999; Al-Kharashi and Evens 1994); more recent results are based on a single large collection (from TREC-2001/2002) (Gey and Oard 2001; Oard and Gey 2002). Several types of index terms have been examined, including words, word clusters, terms obtained through morphological analysis (e.g., stems and roots), and character n-grams of various lengths. The effects of normalizing alternative characters, removal of diacritics and stop-word removal have also been explored (Darwish and Oard 2002a, b; Fraser et al. 2002; Larkey et al. 2002; Mayfield et al. 2001; McNamee et al. 2002). Early studies conducted on small collections suggested that roots were the best Arabic index terms (Abu-Salem et al. 1999; Al-Kharashi and Evens 1994). More recent studies using the larger TREC-2001/2002 Arabic test collection indicate that lightly stemmed words and character 3 and 4-grams result in better retrieval effectiveness than roots (Aljlayl et al. 2001; Darwish and Oard 2002a, b; Fraser et al. 2002; Larkey et al. 2002; Mayfield et al. 2001; McNamee et al. 2002). Retrieval effectiveness is known to be affected by the size, genre, and document length in the test collection, and by many details of system processing (e.g., character normalization, stop-word removal, and morphological analysis). As for OCR degraded Arabic text, a previous study suggests that 3 and 4 character grams and their combinations with index terms obtained through morphological analysis, such light stems, outperform all other kinds of index terms (Darwish and Oard 2002a, b).

3 Experimental setup

As shown in Fig. 1, documents are scanned, OCR'ed, OCR errors are optionally corrected, indexed, and searched. For evaluation, two collections are employed. The first is a small

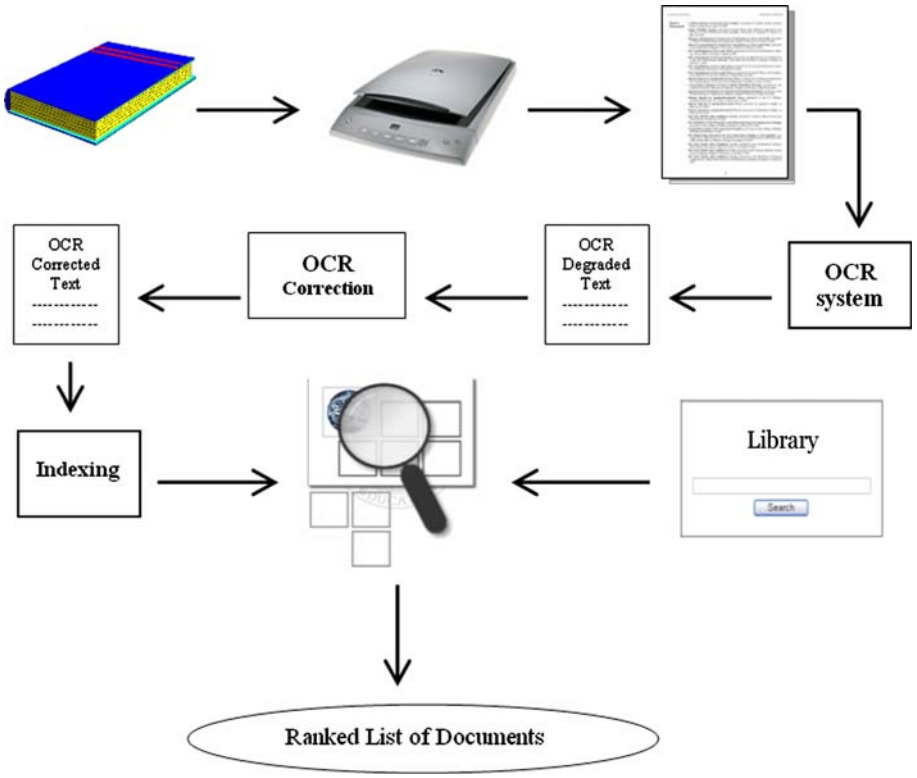


Fig. 1 Document flow in a printed document retrieval system

collection of OCR degraded text. As for the second, due to the lack of existence of a large collection of Arabic OCR text, a large existing character-coded Arabic collection is corrupted to simulate OCR errors in the documents (further explanation is provided in the following subsection). The effect of corrupting the collection and its subsequent correction on retrieval effectiveness is examined. For both collections, a portion of the collection is used to train a character based or a character segment based OCR error correction models. The following presents the collections, the error model which is used to corrupt the large collection, the error correction model that is used to correct both collections, and the design of experiments that test the effect of error correction on retrieval using different index terms.

3.1 The document collection

The first document collection is the Zad collection which is built from *Zad Al-Me'ad*, a printed fourteenth century religious book, which was scanned at 300 300 dpi and OCR'ed using Sakhr's Automatic Reader version 4.0 without any book-specific training. Further, a manually entered and corrected electronic copy of the Zad collection is available. The collection consists of 2,730 separate documents, 25 topics, which only include title queries,

and relevance judgments which were built by exhaustively searching the collection. The number of relevant documents per topic ranges between 3 and 72, averaging 20. The average query length is 5.4 words (Darwish and Oard 2002a, b). The first author of (Darwish and Oard 2002a, b) created the topics and performed the relevance judgments.

As for the large collection, the best presently available Arabic test collection was created for the TREC-2002 "Cross-Language IR (CLIR) track;" for brevity, it is referred to here simply as the TREC collection. It contains 383,872 articles from the Agence France Press (AFP) Arabic newswire. NIST developed 50 topics in cooperation with the Linguistic Data Consortium (LDC), and relevance judgments were developed at the LDC by manually judging a pool of documents obtained from combining the top 100 documents from all the runs submitted by the participating teams in TREC 2002 CLIR track. The number of known relevant documents ranges from 10 to 523, with an average of 118 relevant documents per topic (Oard and Gey 2002). The topic descriptions include a title field that briefly names the topic, a description field that usually consists of a single sentence description, and a narrative field that is intended to contain any information that would be needed by a human judge to accurately assess the relevance of a document (Harman 1995). As for the corruption of the collection, a unigram model is used, as described in (Darwish 2003). OCR degradation is modeled as a noisy channel in which the observed characters result from the application of some distortion function on the real characters. The model used here accounts for three character edit operations: insertion, deletion, and substitution. Formally, given a clean word $\#C_1..C_i..C_n\#$ and the resulting word after OCR degradation $\#D_1..D_j..D_m\#$, where D_j resulted from C_i, representing the null character, L representing the position of the letter in the word (beginning, middle, end, or isolated), and # marking word boundaries, the probability estimates for the three edit operations for the models, are:

$$P_{\text{substitution}}(C_i \rightarrow D_j) = \frac{\text{count}(C_i \rightarrow D_j|L_{C_i})}{\text{count}(C_i|L_{C_i})} \quad (1)$$

$$P_{\text{deletion}}(C_i \rightarrow \varepsilon) = \frac{\text{count}(C_i \rightarrow \varepsilon|L_{C_i})}{\text{count}(C_i|L_{C_i})} \quad (2)$$

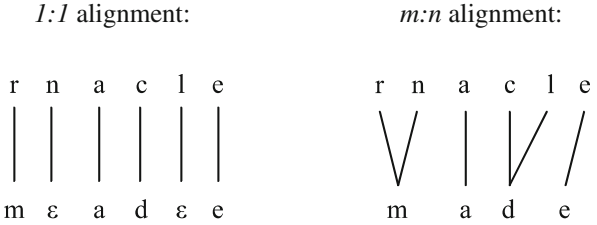
$$P_{\text{insertion}}(\varepsilon \rightarrow D_j) = \frac{\text{count}(\varepsilon \rightarrow D_j)}{\text{count}(C)} \quad (3)$$

The models are trained using 2,000 words obtained by automatically aligning the real OCR outputs from the 300 300 dpi version of the Zad collection with the associated clean text version.

The resulting character-level alignments are used to create a garbler that reads in a clean word $\#C_1..C_i..C_n\#$ and synthesizes OCR degradation to produce $\#D'_1..D'_j..D'_m\#$. For a given character C_i, the garbler chooses a single edit operation to perform by sampling the estimated probability distribution over the possible edit operations. If an insertion operation is chosen, the model picks a character to be inserted prior to C_i by sampling the estimated probability distribution for possible insertions. Insertions before the # (end-of-word) marker are also allowed. If a substitution operation is chosen, the substituted character is selected by sampling the probability distribution of possible substitutions. If a deletion operation is chosen, the selected character is simply deleted. Darwish (2003) validated that the effect of synthesizing OCR degradation using the aforementioned model on retrieval is consistent with the effect of real OCR degradation for the Zad collection.

3.2 Error correction model

For OCR model training, the goal is to learn an effective model of OCR degradation to enable effective correction of OCR errors. It is desirable to minimize the number of training examples, because the process of producing the examples is manual. Previously published papers indicate that training an error model with 2,000 examples produces a good model with as little as 5,000 examples producing nearly the best possible model (Darwish and Oard 2003). The model introduced by (Darwish and Oard 2003) is used as is in this work. For this work, 2,000 words were randomly picked from the corrupted TREC collection to train the error correction model and 4,000 words were used from the Zad collection.[1] The trained models are used to correct the respective collections. The 2,000 words amount to nearly 2–4 pages in an average size book and typically require 20–30 min of correction time. For all words (in training and testing), the different forms of *alef* (*hamza*, *alef, alef maad*, *alef with hamza on top*, *hamza on w*a, *alef with hamza below it*, and *hamza on ya*) are normalized to *alef*, and *ya* and *alef maqsoura* are normalized to *ya*. Also, all diacritics and kashidas are removed. The characters in the corrupted and manually corrected training examples may be aligned in two different ways, namely: 1:1 character alignment (as done in the synthetic degradation process), where each character is mapped to no more than one character (including the null character for deletion or insertion); or using *m*:*n* alignment, where any number of characters are aligned to any other number of characters. The second method is more general and potentially more accurate especially for Arabic where a character can be confused with as many as three or four characters. The following example highlights the difference between the 1:1 and the *m*:*n* alignment approaches. Given the training pair (rnacle,made):



For alignment, the Levenstein dynamic programming minimum edit distance algorithm is used to produce 1:1 alignments. The algorithm initially computes the minimum number of edit operations required to transform one string into another, and then the algorithm is back-traced to find the alignments. Given the output alignments of the algorithm, properly aligned characters (such as a a and e e) are used as anchors, 's (null characters) are combined to properly aligned adjacent characters (anchors) producing *m*:*n* alignments, and 's between correctly aligned characters are counted as deletions or insertions.

To formalize the error model, given a clean word $= \#C_1..C_k..C_l..C_n\#$ and the resulting OCR degraded word $= \#D_1..D_x..D_y..D_m\#$, where $D_x..D_y$ resulted from $C_k..C_l$, (representing the null character, and # marking word boundaries, the probability estimates for the three edit operations for the models are:

[1] Extra training data was used for the real OCR output because error types were more variant than those for the automatically corrupted data.

$$P_{\text{substitution}}(C_k..C_l \rightarrow D_x..D_y) = \frac{\text{count}(C_k..C_l \rightarrow D_x..D_y)}{\text{count}(C_k..C_l)} \quad (4)$$

$$\text{P}_{\text{deletion}}(C_k..C_l \rightarrow \varepsilon) = \frac{\text{count}(C_k..C_l \rightarrow \varepsilon)}{\text{count}(C_k..C_l)} \quad (5)$$

$$P_{\text{insertion}}(\varepsilon \rightarrow D_x..D_y) = \frac{\text{count}(\varepsilon \rightarrow D_x..D_y)}{\text{count}(C)} \quad (6)$$

When decoding a corrupted string composed of the characters $D_1..D_x..D_y..D_m$, the goal is to find a string composed of the characters $C_1..C_k..C_l..C_n$ such that P(|)·P() is maximum. P () is the prior probability of observing in text and P(|) is the conditional probability of producing from .

A modification to the above involved giving a small uniform probability to single character substitutions that are unseen in the training data (Magdy and Darwish 2006). This is done in accordance to Lidstone's law to smooth probabilities. The probability is set to be 100 times smaller than the probability of the smallest seen single character substitution.[2]

For the Zad collection, P() is computed from a web-mined collection of religious text by Ibn Taymiya, who was the main teacher of the medieval author of the Zad book. The collection contains approximately 16 million words, with 279,000 unique surface forms. As for the TREC collection, P() is computed from a web-mined collection of Arabic newswire documents from the BBC, Al-Ahram newspaper, Al-Jazeera news site, Al-Wafd newspaper, and Al-Moheet news site. The collection contains 12 million words, with nearly 260,000 unique surface forms.

P(|) is calculated using the trained model, as follows:

$$P(\delta|\chi) = \prod_{\text{all:}D_x..D_y} P(D_x..D_y|C_k..C_l) \quad (7)$$

The segments $D_x..D_y$ are generated by finding all possible 2^{n1} segmentations of the word . For example, given "macle" then all possible segmentations are (m,a,c,l,e), (ma,c,l,e), (m,ac,l,e), (mac,l,e), (m,a,cl,e), (ma,cl,e), (m,acl,e), (macl,e), (m,a,c,le), (ma,c,le), (m,ac, le), (mac,le), (m,a,cle), (ma,cle), (m,acle), (macle). The segmentation producing the highest probability is chosen.

All segment sequences $C_k..C_l$ known to produce $D_x..D_y$ for each of the possible segmentations are produced. If a sequence of $C_k..C_l$ segments generates a valid word which exists in the web-mined collection, then argmax P(|) · P() is computed, otherwise the sequence is discarded. Possible corrections are subsequently ranked.

3.3 Language modeling

For language modeling, a trigram language model is trained on the same web-mined collections that were mentioned in the previous subsection without any kind of morphological processing. Like the Zad and TREC collections, *alef* and *ya* letter normalizations are performed and diacritics and kashidas are removed. The language model is built using SRILM toolkit with Good-Turing smoothing and default backoff.

[2] Other uniform probability estimates were examined for the training data and the one reported here seemed to work best.

Given a corrupted word sequence $= \{_{1..i..n}\}$ and $= \{X_1..X_i..X_n\}$, where $X_i = \{_{i0..im}\}$ are possible corrections of $_i$ ($m = 10$ for all the experiments reported in the paper), the aim was to find a sequence $= \{_{1..i..n}\}$, where $_i$ X_i, that maximizes:

$$\underbrace{\left(\prod_{i=1..n,j=1..m} P\left(\chi_{ij}|\chi_{i-1,j},\chi_{i-2,j}\right)\right)}_{\text{Language Model}} \cdot \underbrace{P(\delta_i|\chi_{ij})}_{\text{Character Model}} \tag{8}$$

For each corrupted word $_i$, the top m ($m = 10$) correction $X_i = \{_{i0..im}\}$, as computed by Eq. 8, are generated. So given a sequence $= \{_{1..i..n}\}$ the top m corrections for each word are generated leading to $= X_1..(_i..X_n\}$. All possible sequences $_{1...m}((_i\ X_i)$ are generated and scored using Eq. 5. The highest scoring sequence is picked as the correct sequence .

3.4 Testing the models

Two types of tests are performed to measure the effect of error correction. The first type examines the change in Word Error Rate (WER) which is computed by examining a set of approximately 2,000 and 6,000 words for the Zad and TREC collections, respectively. The testing is done for the 1:1 and *m:n* character models with language modeling (LM) enabled or disabled. In all the results reported in this paper, the top correction is chosen. The second examines the effect of correction on retrieval effectiveness. The retrieval experiments are performed on the clean, OCR degraded/synthetically corrupted, and corrected versions of the Zad and TREC collections described above. Note that for the TREC collection, only the *m:n* character mapping is done. The authors' intuition is that since the TREC collection is corrupted using a 1:1 model, then using either models would not make much difference as the *m:n* model is a generalization of the 1:1 model. Multiple corrected versions of the collection are generated with all different correction models mentioned before. The resulting corrected collections are as follows:

For Zad collection, correction with:

1. 1:1 character error model.
2. *m:n* character error model.
3. 1:1 character error model + language model.
4. *m:n* character error model + language model.

For TREC collection, correction with:

1. *m:n* character error model.
2. *m:n* character error model + language model.

The collections are indexed and searched using words, character 3-grams, character 4-grams, and lightly stemmed words obtained using Al-Stem (Oard and Gey 2002). For all experiments, Indri is used with no blind relevance feedback, stopword removal, or stemming. Indri combines inference network model with language modeling (Metzler and Croft 2004). The figure of merit for evaluating retrieval results is mean average precision (MAP). Statistical significance between different retrieval results is performed using a paired 2-tailed *t*-test and Wilcoxon test with continuity correction with *p*-values of less than 0.05 to assume statistical significance. The Wilcoxon test *p*-values are being reported for completeness. There are some indications that the *t*-test is sufficiently reliable despite the fact that the normality condition might not be met (Sanderson and Zobel 2005).

4 Results and discussion

Tables 1 and 2 summarize the effect of correction on WER for the Zad and TREC collections, respectively. As stated earlier, two sets of 2,000 and 6,000 words are used to test the correction of the Zad and the TREC collections, respectively. The evaluation involved examining the word error rate before and after correction with language modeling enabled or disabled. The results show that error correction removes a large portion of the errors with language modeling having a positive impact on error correction. Also, error correction is more effective for the Zad collection compared to the TREC collection. This could be a result of better coverage of the dictionary and better comparability of the trained language model for the correction of the Zad collection.

Figures 2 and 3 and Tables 3 and 4 summarize the retrieval results of searching the original (clean), OCR'ed (corrupted/bad), and corrected versions of the Zad and TREC collections respectively using words, character 3-grams, character 4-grams, and lightly stemmed words. Tables 5 and 6 provide the *p*-values of the paired 2-tailed *t*-test and Wilcoxon text of comparing the results for the Zad and TREC collections respectively. The results confirm that character 3 and 4-grams are indeed the best index terms with 3-grams on uncorrected text outperforming words and light stems even after correction. For correcting the Zad collection with or without language modeling, the results (Table 3) show that retrieval effectiveness is statistically indistinguishable from the original uncorrupted and OCR degraded versions of the collections when indexing using words. However, for the TREC collection (Table 4), using language modeling statistically improves effectiveness over the corrupted version and makes effectiveness indistinguishable from clean version. Same is true for the use of light stems for the Zad collection with and without language modeling and the TREC with language modeling only. For character 3-grams, the error correction statistically significantly improves retrieval effectiveness over corrupted versions for the Zad and TREC collections (except for 3-grams *m*:*n* model without language modeling). Unlike character 3-grams, character 4-grams does not necessarily improve retrieval effectiveness statistically. For character 3 and 4-grams, retrieval effectiveness is generally statistically significantly worse than the clean text (except for 4-gram

Table 1 Word error rate (WER) and error reduction (ER) for correction with the different models for the Zad collection

Model	1:1		*m*:*n*	
	WER (%)	ER (%)	WER (%)	ER (%)
No correction	39.0	–	39.0	–
Base model	24.0	38.5	21.6	44.6
w/language modeling	15.4	60.5	11.7	70.0

Table 2 Word error rate (WER) and error reduction (ER) for correction with the different models for the TREC collection

Model	*m*:*n*	
	WER (%)	ER (%)
No correction	31.4	–
Base model	20.2	35.7
w/language modeling	15.8	49.7

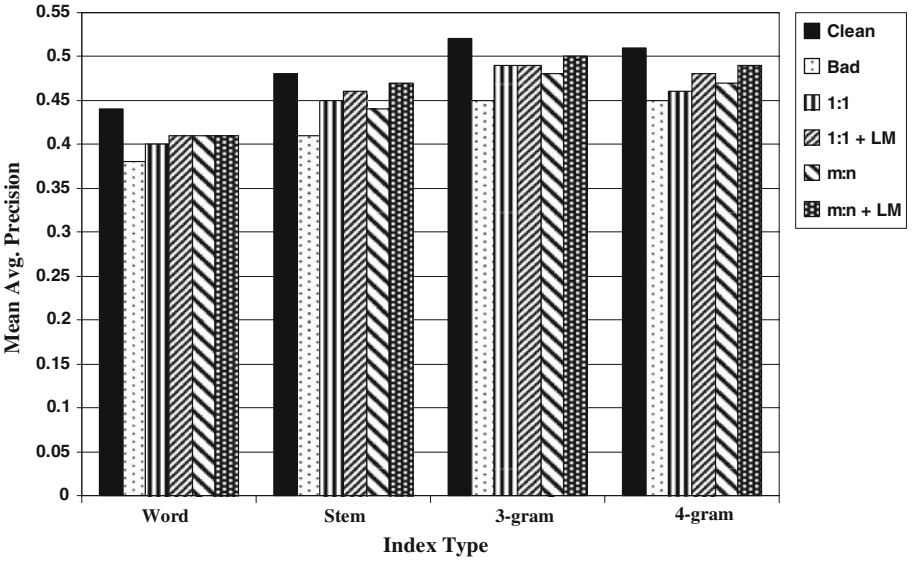


Fig. 2 Results in MAP of searching the original, bad, and corrected versions of the Zad collection (+LM indicates the use of language modeling)

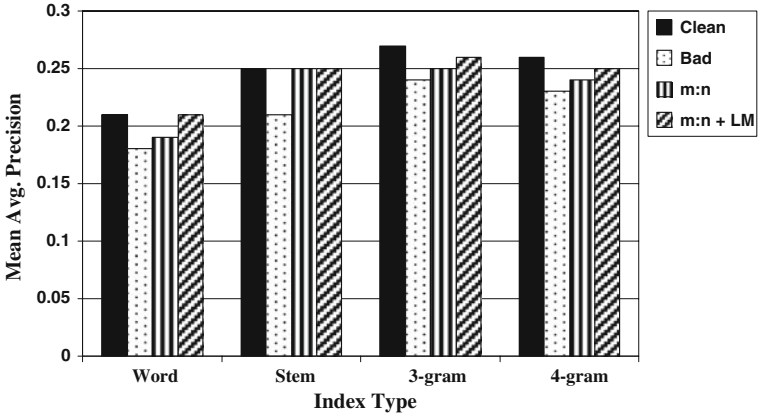


Fig. 3 Results in MAP of searching the original, bad, and corrected versions of the TREC collection (+LM indicates the use of language modeling)

with *m*:*n* model with and without language modeling and the 1:1 model with language modeling for the Zad collection).

The results suggest that given a moderately degraded Arabic collection with resulting word error rate greater than 20%, doing no correction and searching using character 3 or 4-grams without correction does not seem to be a bad strategy. This can be seen in comparing the results for character 3 and 4-grams on the corrupted version compared to the

Table 3 Results in MAP of searching the original, bad, and corrected versions of the Zad collection (+LM indicates the use of language modeling). The left|right squares below MAP for corrected versions indicate *t*-test values in comparing to the clean and bad collections respectively (using *t*-test values from Table 5), with black and grey indicating statistically significantly worse or better, respectively, and white indicating no statistical
significance

	Clean	Bad	*1*:*1*	*1*:*1* + LM	*m*:*n*	*m*:*n* + LM
Word	0.44	0.38	0.40 (0.19 | 0.10)	0.41 (0.11 | 0.14)	0.41 (0.14 | 0.09)	0.41 (0.11 | 0.08)
Stem	0.48	0.41	0.45 (0.11 | 0.02)	0.46 (0.19 | 0.02)	0.44 (0.06 | 0.01)	0.47 (0.33 | 0.00)
3-gram	0.52	0.45	0.49 (0.02 | 0.04)	0.49 (0.02 | 0.04)	0.48 (0.00 | 0.13)	0.50 (0.08 | 0.03)
4-gram	0.51	0.45	0.46 (0.01 | 0.68)	0.48 (0.08 | 0.09)	0.47 (0.06 | 0.24)	0.49 (0.17 | 0.04)

Table 4 Results in MAP of searching the original, bad, and corrected versions of the TREC collection (+LM indicates the use of language modeling). The left|right squares below MAP for corrected version indicate *t*-test values in comparing to the clean and bad collections respectively (using *t*-test values from Table 6), with black and grey indicating statistically significantly worse or better, respectively, and white indicating no statistical significance

	Clean	Bad	*m*:*n*	*m*:*n* + LM
Word	0.21	0.18	0.19 (0.04 | 0.55)	0.21 (0.13 | 0.00)
Stem	0.25	0.21	0.25 (0.10 | 0.28)	0.25 (0.25 | 0.00)
3-gram	0.27	0.24	0.25 (0.00 | 0.03)	0.26 (0.00 | 0.00)
4-gram	0.26	0.23	0.24 (0.01 | 0.30)	0.25 (0.00 | 0.00)

corrected and stemmed versions of Zad and TREC. The results also suggest that indexing using short n-grams such as 3-grams is a better strategy than moderate error correction with no language modeling.

As for using a language model, with the *m*:*n* model for both collections, error correction statistically significantly improves retrieval effectiveness, and for the corrected Zad collection, unlike the TREC collection, retrieval effectiveness is statistically indistinguishable from the effectives of retrieving from the clean version. This would suggest that "good" error correction, with word error rate less than 15%, can have a statistically significant positive effect on retrieval, and possibly improve to the level of retrieving clean documents.

Another interesting and important observation here is that correction in the experiments is done at the word level without any morphological analysis and the correction yielded good results. In fact, using the *m:n* character model with language modeling reduces word error rate by 70%. This seems to suggest that using a large language model for correcting a morphologically rich language like Arabic can minimize the need for morphological analysis. Further, indexing using character n-grams can benefit from good correction that

Table 5 *p*-Value of the paired 2-tailed *t*-test and Wilcoxon test comparisons of retrieval results for the ZAD Collection for Base Model. Black and Grey squares indicate that results are statistically significantly worse and better than corrected version, respectively

Paired 2-tailed t-test					
		1:1	*1:1* + LM	*m:n*	*m:n* + LM
Word	Clean	0.19	0.11	0.14	0.11
	Bad	0.1	0.14	0.09	0.08
Stem	Clean	0.11	0.19	0.06	0.33
	Bad	0.02	0.02	0.01	0
3-gram	Clean	0.02	0.02	0	0.08
	Bad	0.04	0.04	0.13	0.03
4-gram	Clean	0.01	0.08	0.06	0.17
	Bad	0.68	0.09	0.24	0.04
Wilcoxon test					
		1:1	*1:1* + LM	*m:n*	*m:n* + LM
Word	Clean	0.1	0.01	0.08	0.02
	Bad	0.28	0.06	0.03	0.05
Stem	Clean	0	0	0	0.05
	Bad	0.01	0.02	0.01	0
3-gram	Clean	0.02	0.02	0	0.12
	Bad	0.02	0.02	0.06	0.02
4-gram	Clean	0.02	0.03	0.06	0.18
	Bad	0.54	0.15	0.53	0.04

performs no morphological analysis. This is advantageous because character 3 and 4-grams are the best index terms for OCR degraded Arabic text.

5 Conclusion and future work

This paper examines the effect of OCR error correction on retrieval effectiveness of Arabic OCR degraded documents. When correcting without language modeling, the word error rate is nearly halved, but the effect on retrieval effectiveness is less pronounced with no guarantee of statistically significant improvement. This would suggest that given only moderate error correction, performing no correction and using character n-grams is not a bad strategy. However, given "good" error correction, like in the case of using language modeling, retrieval effectiveness can statistically significantly improve (often to the level of retrieval of the uncorrupted documents). Therefore, unless error correction is not "very good" (with error rate greater than 15%) then using n-gram index terms would be preferred for retrieval. Further, given a large language model, word-based error correction can be effective for Arabic, which is orthographically and morphologically complex, even in the

Table 6 *p*-Value of the paired 2-tailed *t*-test and Wilcoxon test comparisons of retrieval results for the ZAD Collection for Base Model. Black and Grey squares indicate that results are statistically significantly worse and better than corrected version, respectively

Paired 2-tailed t-test			
		$m:n$	$m:n$ + LM
Word	Clean	0.04	0.13
	Bad	0.55	0
Stem	Clean	0.1	0.25
	Bad	0.28	0
3-gram	Clean	0	0
	Bad	0.03	0
4-gram	Clean	0.01	0
	Bad	0.3	0
Wilcoxon test			
		$m:n$	$m:n$ + LM
Word	Clean	0.01	0.06
	Bad	0	0
Stem	Clean	0.01	0.08
	Bad	0	0
3-gram	Clean	0	0
	Bad	0	0
4-gram	Clean	0	0
	Bad	0	0

absence of morphological processing. Also, character 3 and 4-grams, which are the best index terms for OCR degraded Arabic text, can benefit from word-based correction with language modeling.

For future work, there are a few clear directions to follow. Investigating sub-word error correction techniques may prove useful for languages where the best index terms are n-grams. Further, a comparison of the effect of error correction as opposed to query garbling is warranted (Darwish and Oard 2003). Also, a serious exploration of the effect of correction on large real OCR document collections is warranted. Unfortunately, there are no reports in the literature of TREC size Arabic OCR document collections and much effort needs to be invested to create such collections. Lastly, investigating the effect of correction using language modeling but no character level model is warranted.

Acknowledgment The authors would like to sincerely thank Mr. Haytham Fahmy for his valuable comments.

References

Abdelsapor, A., Adly, N., Darwish, K., Emam, O., & Nagi, M. (2006). Building a heterogeneous information retrieval collection of printed Arabic documents. *LREC 2006*.

Abu-Salem, H., Al-Omari, M., & Evens, M. (1999). Stemming methodologies over individual query words for Arabic information retrieval. *JASIS, 50*(6), 524–529.

Agirre, E., Gojenola, K., Sarasola, K., & Voutilainen, A. (1998). Towards a single proposal in spelling correction. *COLING-ACL'98* (pp. 22–28).

Ahmed, M. (2000). *A large-scale computational processor of Arabic morphology and applications*. MSc. Thesis, Faculty of Engineering, Cairo University, Cairo, Egypt.

Aljlayl, M., Beitzel, S., Jensen, E., Chowdhury, A., Holmes, D., Lee, M., Grossman, D., & Frieder, O. (2001). IIT at TREC-10. In *TREC-2001, Gaithersbury, MD* (p. 265).

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

Allam, M. (1995). Segmentation versus segmentation-free for recognizing Arabic text. *SPIE, 2422*, 228–235.

Baeza-Yates, R., & Navarro, G. (1996). A faster algorithm for approximate string matching. *Combinatorial pattern matching (CPM'96)* (pp. 1–23). Springer-Verlag LNCS.

Baird, H. (1990). Document image defect models. In *IAPR Workshop on Syntactic and Structural Pattern Recognition* (pp. 38–46).

Baird, H. (1993). Document image defects models and their uses. In *Second international conference on document analysis and recognition* (ICDAR) (pp. 62–67).

Baird, H. (2000). State of the art of document image degradation modeling. In *The 4th IAPR workshop on document analysis systems (DAS 2000)*.

Croft, W. B., Harding, S., Taghva, K., & Andborsak, J. (1994). An evaluation of information retrieval accuracy with simulated OCR output. In *Proceedings of the 3rd annual symposium on document analysis and information retrieval* (pp. 115–126). University of Nevada, Las Vegas, Nev.

Darwish, K., & Emam, O. (2005). 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.*

Darwish, K. (2003). *Probabilistic methods for searching OCR-degraded Arabic text.* Ph.D. Thesis, Electrical and Computer Engineering Department, University of Maryland, College Park.

Darwish, K., & Oard, D. (2002a). CLIR Experiments at Maryland for TREC 2002: Evidence combination for Arabic-English retrieval. In *TREC-2002, Gaithersburg, MD*.

Darwish, K., & Oard, D. (2003). Probabilistic structured query methods. In *SIGIR-2003* (pp. 338–344).

Darwish, K., & Oard, D. (2002b). Term selection for searching printed Arabic. In *SIGIR-2002* (pp. 261–268).

De Roeck, A., & Al-Fares, W. (2000). A morphologically sensitive clustering algorithm for identifying Arabic roots. In *The 38th annual meeting of the ACL*, Hong Kong (pp. 199–206).

Doerman, D. (1997). The retrieval of document images: A brief survey. *ICDAR* (pp. 945–949).

Doermann, D. (1998). The indexing and retrieval of document images: A survey. *Computer Vision and Image Understanding, 70*(3), 287–298.

Doermann, D., & Yao, S. (1995). Generating synthetic data for text analysis systems. In *Symposium on document analysis and information retrieval* (pp. 449–467).

Domeij, R., Hollman, J., Kann, V. (1994). Detection of spelling errors in Swedish not using a word list en clair. *Journal of Quantitative Linguistics*, 195–201.

Fraser, A., Xu, J., & Weischedel, R. (2002). TREC 2002 cross-lingual retrieval at BBN. In *TREC-2002*. Gaithersburg, MD.

Gey, F., & Oard, D. (2001). The TREC-2001 cross-language information retrieval track: Searching Arabic using English, French or Arabic queries. In *TREC-2001* (pp. 16). Gaithersburg, MD.

Gillies, A., Erlandson, E., Trenkle, J., & Schlosser, S. (1999). Arabic text recognition system. In *The symposium on document image understanding technology* (pp. 220–233).

Harding, S., Croft, W., & Weir, C. (1997). Probabilistic retrieval of OCR-degraded text using N-grams. In *European conference on digital libraries* (pp. 345–359).

Harman, D. (1992). Overview of the first text retrieval conference. *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 36–47). Pittsburgh, Pennsylvania, United States.

Harman, D. (1995). Overview of the fourth text retrieval conference. *TREC-4* (p. 1).

Hassibi, K. (1994a). Machine printed Arabic OCR. In *The 22nd AIPR workshop: Interdisciplinary computer vision, SPIE proceedings* (Vol. 2103, pp. 126–134).

Hassibi, K. (1994b). Machine printed Arabic OCR using neural networks. In *The 4th international conference on multi-lingual computing*.

Hawking, D. (1996). Document retrieval in OCR-scanned text. *Sixth parallel computing workshop, paper P2-F*.

Hong, T. (1995). *Degraded text recognition using visual and linguistic context*. Ph.D. Thesis, Computer Science Department, SUNY Buffalo, Buffalo.

Jurafsky, D., & Martin, J. (2000). *Speech and language processing*. Prentice Hall.

Kantor, P., Voorhees, E. (1996). Report on the TREC-5 confusion track. *TREC-5* (p. 65).

Kanungo, T. (1996). *Document degradation models and methodology for degradation model validation*. Ph. D. Thesis, Electrical Engineering Department, University of Washington.

Kanungo, T., Marton, G., & Bulbul, O. (1999a). OmniPage vs. Sakhr: Paired model evaluation of two Arabic OCR products. *SPIE Conference on Document Recognition and Retrieval (VI)* (Vol. 3651, pp.109–120).

Kanungo, T., Baird, H., & Haralick, R. (1995). Validation and estimation of document degradation models. *Symposium on document analysis and information retrieval* (pp. 217–228).

Kanungo, T., Bulbul, O., Marton, G., & Kim, D. (1997). Arabic OCR systems: State of the art. *Symposium on document image understanding technology*.

Kanungo, T., & Haralick, R. (1998). An automatic closed-loop methodology for generating character ground-truth for scanned documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 21*(2), 179–183.

Kanungo, T., Haralick, R., & Phillips, I. (1993). Global and local document degradation models. *The 2nd international conference on document analysis and recognition (ICDAR93)* (pp. 730–734).

Kanungo, T., Haralick, R., Baird, H., Stuezle, W., & Madigan, D. (2000). A statistical, nonparametric methodology for document degradation model validation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(11), 1209–1223.

Lam-Adesina, A. M., & Jones, G. J. (2006). Examining and improving the effectiveness of relevance feedback for retrieval of scanned text documents. *Information Processing and Management 42*(3), 633–649.

Larkey, L., Allen, J., Connell, M. E., Bolivar, A., & Wade, C. (2002). UMass at TREC 2002: Cross language and novelty tracks. In M. V. Ellen, & P. B. Lori (Eds.), *The eleventh text retrieval conference, TREC 2002* (pp 721–732). Gaithersburg, MD: NIST Special Publication 500–251.

Li, Y., Lopresti, D., & Tomkins, A. (1997). Validation of document defect models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*, 99–107.

Lopresti, D. & Zhou, J. (1994). Using consensus sequence voting to correct OCR errors. *IAPR workshop on document analysis systems* (pp. 191–202).

Lu, Z., Bazzi, I., Kornai, A., Makhoul, J., Natarajan, P., & Schwartz, R. (1999). A robust, language-independent OCR system. In *The 27th AIPR workshop: Advances in computer assisted recognition, SPIE* (Vol. 3584).

Magdy, W., & Darwish, K. (2006). Arabic OCR error correction using character segment correction, language modeling, and shallow morphology. *Proceedings of the 2006 conference on empirical methods in natural language processing* (pp. 408–414), Sydney, Australia.

Magdy, W., Darwish, K., & Rashwan, M. (2007). Fusion of multiple corrupted transmissions and its effect on information retrieval. *The seventh conference on language engineering—ESOLEC'2007* (pp. 351–358).

Mayfield, J., McNamee, P., Costello, C., Piatko, C., & Banerjee, A. (2001). JHU/APL at TREC 2001: Experiments in filtering and in Arabic, video, and web retrieval. *TREC-2001. Gaithersburg, MD* (p.322).

McNamee, P., Piatko, C., & Mayfield, J. (2002). JHU/APL at TREC 2002: Experiments in filtering and Arabic retrieval. In *TREC-2002* (p. 358). Gaithersburg, MD.

Metzler, D., & Croft, W. B. (2004). Combining the language model and inference network approaches to retrieval. *Information Processing and Management Special Issue on Bayesian Networks and Information Retrieval, 40*(5), 735–750.

Mittendorf, E., & Schäuble, P. (2000). Information retrieval can cope with many errors. *Information Retrieval, Springer Netherlands, 3*(3), 189–216.

Moussa, B., Maamouri, M., Jin, H., Bies, A., Ma, X. (2003). Arabic treebank: Part 1—10Kword English translation. *Linguistic Data Consortium*.

Nagy, G. (1994) Validation of OCR datasets. *The 3rd annual symposium on document analysis and information retrieval* (pp. 127–136).

Oard, D., Gey, F. (2002). The TREC 2002 Arabic/English CLIR Track. In *TREC-2002*. Gaithersburg, MD.

Oflazer, K. (1996). Error-tolerant finite state recognition with applications to morphological analysis and spelling correction. *Computational Linguistics, 22*(1), 73–90.

Sanderson, M., & Joho, H. (2004). Forming test collection with no system pooling. In *The Proceedings of the 27th ACM SIGIR conference* (pp. 33–40).

Sanderson, M., & Zobel, J. (2005). Information retrieval system evaluation: Effort, sensitivity, and reliability. In *SIGIR 2005* (pp. 162–169). Salvador, Brazil.

Smith, S. (1990). *An analysis of the effects of data corruption on text retrieval performance. technical report DR90–1*. Cambridge, MA: Thinking Machines Corp.

Taghva, K., Borsack, J., & Condit, A. (1994a). An expert system for automatically correcting OCR output. *Proceedings IS&T/SPIE 1994 international symposium on electronic imaging science and technology* (pp 270–278). San Jose, CA.

Taghva, K., Borasack, J., Condit, A., & Gilbreth, J. (1994b). *Results and implications of the noisy data projects*. Technical Report 94–01, Information Science Research Institute, University of Nevada, Las Vegas.

Taghva, K., Borasack, J., Condit, A., & Inaparthy, P. (1995). *Querying Short OCR'd Documents*. Technical Report 94–10, Information Science Research Institute, University of Nevada, Las Vegas.

Taghva, K., Borsack, J., & Condit A. (1996a). Evaluation of model-based retrieval effectiveness OCR text. *ACM Transactions on Information Systems, 14*(1), 64–93.

Taghva, K., Borsack, J., & Condit, A. (1996b). Effects of OCR errors on ranking and feedback using the vector space model. *Information Processing and Management, 32*(3), 317–327.

Tillenius, M. (1996). Efficient generation and ranking of spelling error corrections. *NADA report TRITA-NA-E9621*.

Tseng, Y., & Oard, D. (2001). Document image retrieval techniques for Chinese. In *Symposium on Document Image Understanding Technology* (pp. 151–158). Columbia, MD.