

# Table of Contents

<b>Table of Contents .....</b>	<b>1</b>
<b>List of figures .....</b>	<b>3</b>
<b>List of tables.....</b>	<b>4</b>
<b>Abstract.....</b>	<b>1</b>
<b>1. Introduction .....</b>	<b>2</b>
1.1. Contribution.....	5
1.1.1. <i>Text Fusion</i> .....	5
1.1.2. <i>Omni-font OCR Correction</i> .....	5
1.1.3. <i>Information Retrieval Experiments</i> .....	6
1.2. Thesis Outlines .....	7
<b>2. Literature Survey .....</b>	<b>8</b>
2.1. Arabic OCR.....	9
2.2. OCR Error Correction Applications:.....	11
2.2.1. <i>OCR Error Correction:</i> .....	11
2.2.2. <i>Arabic Information Retrieval:</i> .....	15
<b>3. Fusion .....</b>	<b>17</b>
3.1. Introduction .....	18
3.2. Text Fusion.....	19
3.3. Text Alignment:.....	20
3.4. Fusion Effect on Error Rate Reduction: .....	24
3.4.1. <i>Experimental Setup:</i> .....	24
3.4.2. <i>Results</i> .....	25
3.5. Fusion Effect on Information Retrieval Improvement .....	28
3.5.1. <i>Data Set</i> .....	28
3.5.2. <i>Experimental Setup and Results</i> .....	32

3.6.	Conclusion and Future Work .....	35
<b>4.</b>	<b>Omni Font OCR Error Correction .....</b>	<b>37</b>
4.1.	Introduction .....	38
4.2.	Data set .....	39
4.3.	Error Correction and Experimental Setup .....	40
4.3.1.	<i>Candidates Selection</i> .....	41
4.3.2.	<i>Constant "C" selection</i> .....	41
4.3.3.	<i>Language Modeling</i> .....	42
4.3.4.	<i>Testing Correction Effectiveness</i> .....	44
4.4.	Experimental Results .....	46
4.5.	Conclusion and Future Work .....	52
<b>5.</b>	<b>Integrated System.....</b>	<b>53</b>
5.1.	Introduction .....	54
5.2.	Integrated System Architecture .....	55
5.3.	Experimental setup and results .....	58
5.4.	Conclusion and Future Work .....	60
<b>6.</b>	<b>Conclusion and Future Work .....</b>	<b>61</b>
6.1.	Conclusions .....	62
6.2.	Future work .....	64
	<b>References .....</b>	<b>65</b>

## List of figures

Figure 1-1 Transformation Layer.....	2
Figure 2-1 Arabic language orthographic and morphological challenges.....	10
Figure 3-1 Block diagram modeling Text Fusion process.....	19
Figure 3-2 Results in MAP of searching different versions of the collection .....	31
Figure 3-3 Results in MAP of searching different fused models, hashed bars refers to statistical significant retrieval results better than the original degraded versions .....	34
Figure 3-4 Preferred implementation for the fusion system .....	36
Figure 4-1 Accuracy vs. number of candidate corrections for ZAD set with different values of C. Accuracy refers to the presence of a proper correction among the N best candidate corrections. ....	42
Figure 4-2 Results in MAP for searching different versions of the ZAD collection .....	50
Figure 5-1 Possible implementations for the integrated system.....	57

## List of tables

Table 3-1 Example of character alignment using edit distance .....	21
Table 3-2 Example of word alignment using edit distance .....	22
Table 3-3 Error rates in different versions of the test data .....	25
Table 3-4 Fusion results of different fonts.....	26
Table 3-5 Produced versions of text set after applying error model with different CER .....	31
Table 3-6 Results of fusion of each two different text set versions. Each cell is formed of upper and lower values, upper value is the WER after fusion of the two models opposite to the cell, lower value represents the common error between these two versions of text and which is the limit if WER .....	33
Table 3-7 Results of fusion of some triples of models, Model-1-1-2 returns to fusion of two different versions of model-1 and one version of model-2 .....	34
Table 4-1 Percentage of words for which proper correction was not found in top N corrections .....	48
Table 4-2 WER for different values of $\beta$ for ZAD test set .....	48
Table 4-3 WER for different values of $\beta$ for TREC test set using different LMs	49
Table 4-4 Comparing Correction effectiveness with and without using character model for ZAD set .....	49
Table 4-5 Comparing Correction effectiveness with and without using character model TREC set.....	50
Table 4-6 p-values of paired 2-tailed t-test and Wilcoxon tests comparing the retrieval effectiveness when using language modeling with the ED and REF models for different index terms.....	51
Table 4-7 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.....	51

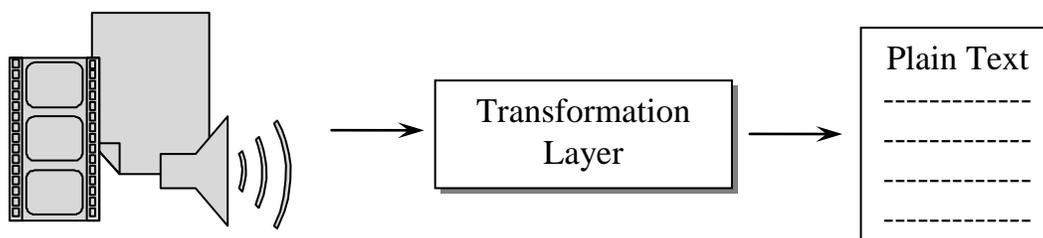
Table 5-1 Error rates and OOV rates for fused versions of ZAD.....	58
Table 5-2 WER and amount of Error reduction for different fused versions of OCRed text using two different methods for correction.....	59

## **Abstract**

Nowadays, massive digitization efforts are being exerted for converting variant types of data into text form. Optical character recognition (OCR) is considered the main enabling technology for converting printed documents and books into digital form. Unlike claims of OCR vendors, OCR error rates are far from perfect. These error rates are much higher in some languages such as Arabic due to its orthographical and morphological challenges. Previous work has focused on correction of OCR degraded text based on the presence of prior information about character error models. However, previous work neglected the possibility of the presence of different versions of the generated OCR text from different OCR systems, which are assumed to produce different types of errors. This thesis explores text fusion, which involves the use of language modeling to determine which OCR system (if any) properly recognized individual words. In addition, another technique for correction of OCR degraded text that is independent of character-level OCR errors, and hence independent of scanned document source. It is based on language modeling in conjunction with a uniform character model that uses edit distance only. Both techniques were applied Arabic OCR text from different domains (religious and news domains). Both techniques have proved their significant effect on error reduction, especially when integrating both techniques together error reduction have reached 86%. For all experiments, information retrieval effectiveness was tested, and improvement has been reported in most of cases.

# 1. Introduction

Since the advent of internet at the end of the twentieth century, there has been a push to represent different types of information in text form to facilitate search ability and storage. There are variant forms of information that require much effort to transform into text, such as images, audio, video. A transformation layer is usually presented to transform these different types of media into text form (figure 1.1). The transformation layer can have several forms according to the type of the input data. Automatic Speech Recognition (ASR), Optical Character Recognition (OCR), and text in image detection are used as the transformation layers for transforming information found in speech, document images, and videos into text respectively.



**Figure 1-1 Transformation Layer**

Unfortunately, these transformation technologies usually introduce errors in the resultant text. These errors are introduced as a reason for misrecognition of signals coming from the input data. The types of error depend on the type of input signal in addition to the features used for recognition. For speech signals, errors are mainly due to noisy environment, phonetically close phonemes, continuous speech (Hui Jiang, 2005; Amir et al., 2001). For OCR, errors are due to low quality of the printed pages to be recognized, difficulty of some fonts, and some confusion between characters of close shapes in addition to the orthographic and morphological challenges of some languages (Magdy and Darwish, 2006a and

2006b). For text in image detection, errors stems from textures in images and complex orientation of text in images.

Different approaches aim to correct the resultant error in these transformation systems. However, working in the transformation layer is considered to be very complicated and requires the integration of many technologies and classifiers. Furthermore, transformation layers are considered domain dependent, which means that one transformation layer for a specific signal cannot be used to transform a different type of signal, for example, ASR cannot be used for recognizing characters in document images. Hence some efforts focus on the preprocessing and post-processing of the incoming signal and the output text respectively for error reduction. Preprocessing depends on signal refining and adjustment, noise removal, skew detection and correction (for OCR), environment identification and subtraction ... etc. Post processing depends mainly on linguistic and morphological properties of the recognized text language, and it works with syntactic analysis and language modeling.

Errors from the transformation layer are more pronounced in some challenging languages such as Chinese and Arabic due to these languages orthographic and morphological properties (Kolak and Resnik, 2002). The introduced errors adversely affect linguistic processing and retrieval of the transformed text.

This thesis focuses on post processing of Arabic OCR text for error reduction and information retrieval enhancement.

Post processing for OCR text focuses on decreasing the errors through degraded text correction using error models and sometimes language models. In some applications where the main purpose of error reduction is the enhancement of the information retrieval effectiveness, correction of errors can be replaced with other techniques such as query degradation using character error models, or index expansion with candidate corrections for words (Darwish and Magdy, 2007).

In this thesis, different approaches for OCR text post processing are introduced. Different versions of OCR text fusion, Omni-font OCR error correction, and

integrated system of fusion and correction are introduced with testing their effect on error reduction and information retrieval effectiveness. Experiments are applied on Arabic language, as a reason of the orthographic and morphological challenges of this language. However, the introduced approaches are potentially applied on different languages. Furthermore, the approaches are also applied on text coming from different domains other than the OCR, such as ASR text.

## **1.1. Contribution**

This Research makes the following contribution:

1. Text Fusion.
2. Omni-Font OCR Correction.
3. Information Retrieval improvement for degraded text

### **1.1.1. Text Fusion**

Text fusion is a new technique that assumes the presence of more than one version of the degraded text (OCR output text), each with different types of errors. These different versions of degraded text are aligned by words in order to be fused using a language model that tries to determine which version (if any) has an uncorrupted version for each word in the text. The introduced technique consult a language model in order to select the best sentence path among different candidate paths coming from different versions of the text. These different versions of text are mainly coming from the same source image, but each version uses different OCR system and with different image resolution.

The main aim from text fusion is to prove that with the same source image, it is possible to obtain a better quality OCR text output when using different resolutions and different OCR systems in the OCR process. The final fused version has to have fewer errors than any of the used versions in the fusion process.

### **1.1.2. Omni-font OCR Correction**

Previous work in degraded text correction (Brill and Moore, 2000; Church and Gale, 1991; Domeij et al., 1994; Hong, 1995; Jurafsky and Martin, 2000; Magdy and Darwish, 2006a, 2006b) depends on the presence of some information about the error model. However, in case of using text fusion, and as the fused version will be a combination of different versions with different error models, using a character error model for correction will be infeasible. A presented approach

introduces a general method for error correction that does not require the training of a character error model. The correction method relies on a uniformly distributed character error model based on the edit distance between a misrecognized word and a candidate correction with the assistance of a domain specific language model. This approach is well suited for situations where document are obtained from a variety of sources. This approach is compared to the state-of-the-art approaches for degraded text correction.

### **1.1.3. Information Retrieval Experiments**

Many recent initiatives, such as Project Gutenberg<sup>1</sup>, Google Print<sup>2</sup>, the Open Content Alliance<sup>3</sup>, and the Million Book Project, have focused on digitizing and OCR'ing large repositories of legacy books books (Thoma and Ford, 2002; Simske and Lin, 2004; Barret et al. 2004). Such initiatives have been successful in digitizing and OCR'ing millions of books in a variety of languages. One important task associated with the digitization efforts revolves around effectively finding information inside books through search. The task of searching digitized books is potentially complicated by the OCR process, which typically introduces errors in the textual representations of books. The errors are affected by the quality of paper, printing, font, OCR training, and scanning. One of the contributions of this thesis is testing the effect of error reduction in each step on the information retrieval effectiveness. A classic Arabic book from the fourteenth century is used for all the information retrieval tests. Some other tests for the retrieval effectiveness were tested on news data, in order to check the performance over a variety of domains. Mean average precision (MAP) is used as the figure of merit for all the experiments.

---

<sup>1</sup> Project Gutenberg website, <http://www.gutenberg.org>

<sup>2</sup> Google Print website, <http://books.google.com>

<sup>3</sup> Open Content Alliance website, <http://www.opencontentalliance.org>

## **1.2. Thesis Outlines**

This thesis is organized as follows:

In Chapter 2, a literature survey describes prior work done in three main topics related to the subject of the thesis: Arabic OCR and Arabic language challenges; OCR error correction, Arabic OCR error correction in specific. Third part describes the state-of-the-art approaches for improving information retrieval effectiveness for OCR output text, and especially for Arabic OCR.

In Chapter 3, Text fusion is defined and the utilized approach is described, then experimental setup to test fusion effectiveness on error reduction and information retrieval.

Chapter 4 describes the main Omni-font error correction approach, describes experimental setup, and discusses results error reduction and retrieval.

Chapter 5 shows the integration of text fusion with text correction, and reports the enhancement in text quality when using both systems.

Chapter 6 concludes the thesis, and discusses potential future work.

## **2. Literature Survey**

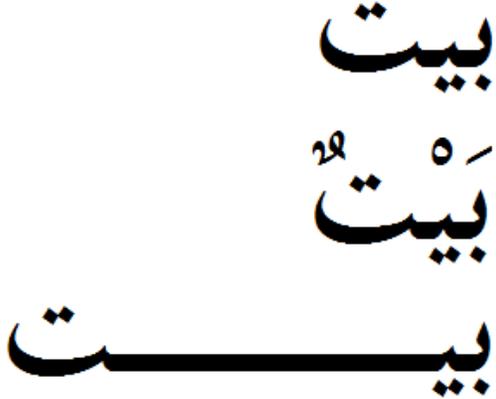
In this chapter, prior work on Arabic OCR, OCR errors correction, and information retrieval for degraded documents are reported. For Arabic OCR, Arabic language challenges are described from the orthographic and morphological points of view, and the state-of-the-art Arabic OCR systems are listed. For OCR error correction, previous work and approaches for error correction in degraded text are described, and prior art concerning Arabic degraded text is described in more details. The last part in this chapter describes the work done for improving information retrieval effectiveness for degraded text using different approaches including Arabic specific approaches.

## 2.1. Arabic 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 (figure 2.1), including:

- Arabic's cursive script in which most characters are connected and their shape vary with position in the word.
- 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 letters to distinguish between different letters and the optional use of diacritic which can be confused with dirt, dust, and speckle (Darwish and Oard, 2002).
- The morphological complexity of Arabic, which results in an estimated 60 billion possible surface forms, complicates dictionary-based error correction. Arabic words mostly contain prefix and suffix and they 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 between two letters of the root (Ahmed, 2000).

There are a number of commercial Arabic OCR systems, with Sakhr's Automatic Reader, Shonut's Omni Page, and RDI Arabic OCR being perhaps the most widely used. Retrieval of OCR degraded text documents has been reported for many languages, including English (Harding et al., 1997), Chinese (Tseng and Oard, 2001), and Arabic (Darwish and Oard, 2002).

 <p>Different shapes of the same Arabic character ('ain) depending on its position in word</p>	 <p>Three different Arabic characters with the same base shape but different numbers of dots</p>
 <p>Same Arabic word, written in first line with no diacritics or elongations. In second line with diacritics, and third line with elongation</p>	 <p>wasaya+ktub+unahaa (prefix)+(word)+(suffix) and will + write + they it = and they will write it</p> <p>One Arabic word represents the challenges in Arabic morphology. When translated to English, it is translated into five words</p>
<p><b>Figure 2-1 Arabic language orthographic and morphological challenges</b></p>	

## **2.2. OCR Error Correction Applications:**

OCR systems are not so accurate especially for Arabic text. Word Error Rates (WER) for Arabic OCR output ranges from 3% to 60% from the previously mentioned systems depending on font, scanning resolution, paper quality, and other factors. Low WER can be acceptable in many applications, but moderate and high error rates are often unacceptable especially when a user is the consumer of the OCR output or when application, such as information retrieval that rely on the correct recognition for properly matching.

In the presented work, the effect of error correction is measured on two ways

First: the effect of WER and Character Error Rates (CER).

Second: the effect of correction on retrieval effectiveness.

### **2.2.1. OCR Error Correction:**

Much research has been done to correct recognition errors in OCR-degraded collections. There are two main categories of correction techniques. 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 (Oflazer, 1996). 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 thesis, is used to compare recognized words with words in a term list (Church and Gale, 1991; Hong, 1995; Jurafsky and Martin, 2000). 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 (2000) illustrate the use of a noisy channel model to find the correct spelling of misspelled or misrecognized words. 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). 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.

- Character N-Grams: Character n-grams maybe used alone or in combination with dictionary lookup (Lu et al., 1999; Taghva et al., 1994). 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 described an English recognizer that identifies acronyms and named-entities, clusters them, and then treats the words in each cluster as one word (Taghva, 1994). 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 (1998), an English grammar was used to parse sentences with spelling mistakes. 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”.

Magdy and Darwish (Magdy and Darwish, 2006) tested the effectiveness post-OCR error correction. The correction uses an improved character segment based noisy channel model to correct OCR errors. They examined the use of single character and character segment based correction of Arabic OCR text combined with language modeling and shallow morphological analysis. Further, they tested character position and smoothing. The results showed the superiority of the character segment based model compared to the single character based model. Further, the use of language modeling yielded improved error correction particularly for the character segment based model. Accounting for character position and shallow morphological analysis had a negative impact on correction, while smoothing had a positive impact. Lastly, given a large in-domain corpus to extract a correction dictionary and to train a language model is a sufficient strategy for correcting a morphologically rich language such as Arabic with a 70% reduction in word error rate.

### **2.2.2. 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, M 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 (Darwish, 2003). The effects of normalizing alternative characters, removal of diacritics and stop-word removal have also been explored (Darwish and Oard, 2002); (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, M 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 (Darwish and Oard, 2002); (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, 2002).

Other work examined the effect of word-based post-OCR error correction on Arabic retrieval effectiveness (Magdy and Darwish, 2006). Although word-based error correction has an impact on text quality and word error rate was nearly halved, the effect on retrieval effectiveness was less pronounced with statistically significant increases for longer index terms and no statistically significant increases for shorter index terms.

Darwish and Magdy (Darwish and Magdy, 2007) compared the improvement in IR effectiveness with OCRed text using two different approaches both based on character error model; first approach tested the usage of an error model for query garbling, and the second one the use of OCR correction. The results showed that unless a “good” language model is built for the correction, then query garbling will be a better approach for improving IR effectiveness.

### **3. Fusion**

In this chapter, Text fusion is defined with a full description for the approach used for implementation. Two types of test were examined, the first tests fusion effect on error reduction, and the second tests its effect on the improvement of information retrieval effectiveness.

### **3.1. Introduction**

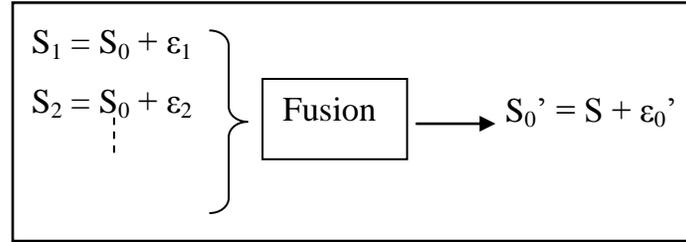
Previous work on OCRed text focused on two main aspects. The first involves improving Information Retrieval (IR) effectiveness on the degraded text using query garbling in conjunction with structured or balanced queries (Oard and Ertunc, 2002), (Darwish and Oard, 2003a, 2003b). The second focuses on correcting OCR errors to improve IR effectiveness (Taghva et al, 1994), (Tseng and Oard, 2001), (Lu et al., 1999), (Magdy and Darwish, 2006).

Previously mentioned OCR correction work depends on the presence of only one source of degraded text. In this chapter, a new technique is introduced that assumes the presence of more than one version of the degraded text, each with different types of errors. These different versions of degraded text are aligned by words in order to be fused using a language model to try to determine which version (if any) has an uncorrupted version for each word in the text. The introduced technique is applied on different versions of the same source text that is printed in different fonts and then scanned using different OCR systems and different resolutions.

This chapter is organized as follows: Section 2 defines text fusion and describes the main idea of how it works; Section 3 shows how text from different sources is aligned as the main preprocessing step for text fusion; Section 4 describes the experimental setup and shows results of fusion on error reduction; Section 5 shows fusion effect on information retrieval; and finally section 6 concludes the chapter and discusses possible future directions.

### 3.2. Text Fusion

Text fusion can be defined as follows: given a clean text set  $S_0 = \{s_{01} \dots s_{0j} \dots s_{0m}\}$  and  $n$  degraded versions  $S_i = \{s_{i1} \dots s_{ij} \dots s_{im}\}$ , where  $1 \leq i \leq n$  and  $s_{ij}$  is the degraded version of  $s_{0j}$ ,  $S_i$  can be represented as  $S_0 + \varepsilon_i$ , where  $\varepsilon_i$  is the set of edit operations necessary to transform from the clean version to the degraded version and  $\varepsilon_i$  could result from to the data entry process (OCR, ASR, typing ... etc). As illustrated in Fig. 1, the goal is to obtain a new version  $S_0' = S_0 + \varepsilon_0'$ , where  $S_0'$  is obtained by picking the closest  $s_{ij}$  to  $s_{0j}$  leading to  $\varepsilon_0' < \text{minimum}(\varepsilon_j)$ . In this work, a trigram language model is used to attempt to pick the closest  $s_{ij}$  to  $s_{0j}$  by finding  $S_0'$  that maximizes the language model probability.



**Figure 3-1 Block diagram modeling Text Fusion process**

Language model is used to pick up the most likelihood proper word among different candidates from different degraded text sets. Given a degraded word sequences  $\Xi = \{X_1 \dots X_i \dots X_m\}$ , where  $X_i = \{\chi_{i0} \dots \chi_{iN}\}$  are possible candidates for the proper word in the original text (*where  $N$  is the number of degraded sources*), the aim is to find a sequence  $\Omega = \{\omega_1 \dots \omega_i \dots \omega_m\}$ , where  $\omega_i \in X_i$ , that maximizes the tri-gram language model probability of the word sequence:

$$\prod_{i=1..m, j=1..N} P(\chi_{ij} | \chi_{i-1, j}, \chi_{i-2, j})$$

### 3.3. Text Alignment:

In order to apply fusion on different versions of text, words in these versions need to be aligned, where each word (set of words) in each OCR output version is aligned to the corresponding word (set of words) in OCR output of the other version. The problem of alignment stems from OCR errors such as word splitting, where a word is recognized with spaces between its characters leading to be seen as more than one word, and words appending, where more than one word are appended to each other as the OCR system misses the entire spaces.

Alignment of OCR outputs is not an easy problem, however in the same time, some clues are found in the OCR output, which help in the alignment process. The fact that lines coming from different OCR systems are already aligned, since they are coming from the same source image, is the main clue. Hence, alignment process for OCR output is simplified from text in page alignment to words in line alignment.

Edit distance is used for the alignment process. Magdy and Darwish (Magdy and Darwish, 2006) used Levenshtein edit distance for character alignment between clean and degraded versions of the same word. Their algorithm was developed in order to achieve m:n character mapping, which is very similar to word alignment problem, with words are to be aligned instead of the characters.

Magdy and Darwish algorithm for alignment depends on the Levenshtein distance table in order to identify the unchanged characters in both words and map them to each other, and then different characters in both words are then easily aligned.

Example:

For a clean word “Gambol” that is garbled to be “Gumbo”, Levenshtein edit distance is calculated as shown in table 3.1. Black cells indicate characters matching in both words, where if cell<sub>ij</sub> is black, this indicates that character<sub>i</sub> in word<sub>1</sub> and character<sub>j</sub> in word<sub>2</sub> are both matched and aligned to each other. These cells contains the minimum value for their entire row and/or column, otherwise, if the cell doesn’t contain the only minimum value at least in one of its entire row or

column, then corresponding characters to these cells are considered to be not aligned.

As shown in table, “A” row, and “N” column don’t have any unique minimum value, then they aren’t considered to be aligned to any character. The same is for “E” row that has no corresponding character in other word, and this appears from the minimum number in its row, isn’t the minimum one in the column, in this case, this characters is considered not to be aligned to any other character in the other word too.

**Table 3-1 Example of character alignment using edit distance**

		S	N	M	B	L
	0	1	2	3	4	5
S	1	0	1	2	3	4
A	2	1	1	2	3	4
M	3	2	2	1	2	3
B	4	3	3	2	1	2
L	5	4	4	3	2	1
E	6	5	5	4	3	2

The same technique is used for words alignment in couple of lines, where characters in both lines are aligned with considering spaces as normal character, and then only aligned spaces in both lines are used to be delimiters for splitting both lines into a set of aligned words.

**Example**

For an original text of a printed text “كافيك وكافي أتباعك”, it was recognized with two different OCR systems, and the output was as follows: “كافيك#وكافي#أتباعكافلا” and “كنيكوكافي#ناعك#فلا”, where # is for spaces.

Applying the previous algorithm on these two small lines, the constructed table with identified aligned characters is shown in table 3.2.

Table 3.2 shows that there is only one space in both lines that are aligned to each other, and in this case these aligned spaces will be the words delimiters, and lines are split into aligned set of words as follows:

كافيك وكافي أتباعكافلا  
 كنيكوكافي ناعك##فلا

Where word alignment here are m:n, as first part of the sentences are 2:1 word alignment, and the second part is 1:2 word alignment.

Table 3-2 Example of word alignment using edit distance

		ك	ئ	ي	ك	و	ك	ا	ف	ي	#	ن	ا	ع	ك	#	ف	ل	ا
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ك	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
ا	2	1	1	2	3	4	5	5	6	7	8	9	10	11	12	13	14	15	16
ف	3	2	2	2	3	4	5	6	5	6	7	8	9	10	11	12	13	14	15
ي	4	3	3	2	3	4	5	6	6	5	6	7	8	9	10	11	12	13	14
ك	5	4	4	3	2	3	4	5	6	6	6	7	8	9	9	10	11	12	13
#	6	5	5	4	3	3	4	5	6	7	6	7	8	9	10	9	10	11	12
و	7	6	6	5	4	3	4	5	6	7	7	7	8	9	10	10	10	11	12
ك	8	7	7	6	5	4	3	4	5	6	7	8	8	9	9	10	11	11	12
ا	9	8	8	7	6	5	4	3	4	5	6	7	8	9	10	10	11	12	11
ف	10	9	9	8	7	6	5	4	3	4	5	6	7	8	9	10	10	11	12
ي	11	10	10	9	8	7	6	5	4	3	4	5	6	7	8	9	10	11	12
#	12	11	11	10	9	8	7	6	5	4	3	4	5	6	7	8	9	10	11
أ	13	12	12	11	10	9	8	7	6	5	4	4	5	6	7	8	9	10	11
ت	14	13	13	12	11	10	9	8	7	6	5	5	5	6	7	8	9	10	11
ب	15	14	14	13	12	11	10	9	8	7	6	6	6	6	7	8	9	10	11
ا	16	15	15	14	13	12	11	10	9	8	7	7	6	7	7	8	9	10	10
ع	17	16	16	15	14	13	12	11	10	9	8	8	7	6	7	8	9	10	11
ك	18	17	17	16	15	14	13	12	11	10	9	9	8	7	6	7	8	9	10
ا	19	18	18	17	16	15	14	13	12	11	10	10	9	8	7	7	8	9	9
ف	20	19	19	18	17	16	15	14	13	12	11	11	10	9	8	8	7	8	9
ل	21	20	20	19	18	17	16	15	14	13	12	12	11	10	9	9	8	7	8
ا	22	21	21	20	19	18	17	16	15	14	13	13	12	11	10	10	9	8	7

Previous technique is used for word alignment between any two different versions of text, assuming the lines are already aligned, which is a very acceptable assumption. However, the previous technique can be extended for line alignment in case of non aligned lines, but in this case, it will take a whole page text, and new lines will be considered as the delimiters instead of the spaces.

### **3.4. Fusion Effect on Error Rate Reduction:**

In this section, Fusion will be tested on real data to check the amount of improvement that could be gained from this process. A set of ten pages of text will be printed in three different fonts, and then scanned using different resolutions and OCRed using two different OCR engines.

#### **3.4.1. Experimental Setup:**

For testing the effect of fusion process, ten pages were selected from an Arabic religious book from the 14<sup>th</sup> century (Zad Al-ma'ad fi Hadie Khair Al-ebad<sup>4</sup>), and were manually written again to obtain the clean version of the text. Pages were found to contain about 4,200 words. The clean version is then printed with three different Arabic fonts to obtain different sources of the same text. The used fonts were: Kufi, Mudir, and Simplified Arabic. Each of these three versions is then scanned twice, once at 300 dpi and another at 200 dpi, and then all images are OCRed using two different OCR systems: Sakhr Automatic Reader<sup>5</sup>, and RDI OCR system<sup>6</sup>, which lead to the presence of twelve different recognized versions of text to the same source data.

All versions were aligned to the original (clean) version of text using the previously mentioned alignment technique in order to calculate the error rates. Table 3.3 shows the error rates in each version.

As shown in table 3.3, error rates differ from one font to another and from one OCR system to another. In most cases, 300 dpi scanned version achieved better quality than in 200 dpi. It can be noticed Sakhr OCR system performance was much less than RDI system in all cases, but this returns that RDI system was trained for these specific types of fonts, unlike Sakhr, which was trained on Omni font.

---

<sup>4</sup> Referred to later as ZAD

<sup>5</sup> Referred to later as Sakhr

<sup>6</sup> Referred to later as RDI

**Table 3-3 Error rates in different versions of the test data**

Font Type	OCR System	300 dpi		200 dpi	
		CER	WER	CER	WER
Kufi	Sakhr	20.5%	44.7%	19.1%	41.8%
	RDI	1.6%	5.6%	4.8%	14.6%
Mudir	Sakhr	4.4%	10.7%	3.3%	8.8%
	RDI	0.8%	3.0%	8.1%	25.6%
Simplified	Sakhr	3.7%	9.1%	7.1%	16.5%
	RDI	2.4%	9.4%	19.7%	56.2%

A trigram language model was trained on a web-mined collection of religious books belonging to Ibn Taymia, the teacher of the author of ZAD book, to insure content similarity, using the SRILM toolkit (Stolcke, 2002). Testing the LM on the test version, it was found that the out of vocabulary (OOV) was less than 1%, which reflects the strength of the built language model.

Different fusion runs were performed on the previous data. For each font, versions coming from the two different OCR systems are fused for each resolution separately. For each OCR system, versions of different resolutions are fused for each different font. Finally, all versions for a given font are fused together giving one output version coming from four different versions. For all versions WER and CER were checked before and after fusion.

### 3.4.2. Results

Table 3.4 shows fusion results for different versions of the same source data. Table 3.4 is decomposed of three main tables; each consists of the fusion results for different versions of a certain font. For each cell in tables, the upper and lower parts represent the CER and WER respectively. Shaded cells are for the original versions, while un-shaded ones are for the fused versions. For fused cells, each one lies between the two versions that are fused together, and the cell in the

middle of each table represents the outcome of the fusion of the whole four versions of text.

Results show that fused version always has lower error rates than the original version. From tables, the fusion of the four versions doesn't improve the quality of text so much, however in most cases error rates are decreased but with no significant error reduction like that in two versions fusion.

Error reduction varies in fusion process depending on the overlapping errors between version, which is semi-random, as it depends mainly on the OCR engine, plus the quality and orientation of the scanned paper.

**Table 3-4 Fusion results of different fonts**

Kufi				Mudir			
	200 dpi		300 dpi		200 dpi		300 dpi
RDI	4.8%	1.27%	1.6%	RDI	8.1%	0.72%	0.8%
	14.6%	4.03%	5.6%		25.6%	2.45%	3.0%
	3.41%	2.07%	2.49%		1.19%	0.57%	0.52%
	8.73%	4.97%	6.10%		3.63%	2.00%	1.91%
Sakhr	19.1%	15.45%	20.5%	Sakhr	3.3%	2.21%	4.4%
	41.8%	34.70%	44.7%		8.8%	5.84%	10.7%

Simplified			
	200 dpi		300 dpi
RDI	19.7%	1.09%	2.4%
	56.2%	3.98%	9.4%
	4.39%	0.58%	0.70%
	11.90%	2.10%	2.50%
Sakhr	7.1%	1.02%	3.7%
	16.5%	3.37%	9.1%

Unexpected results can be seen in table 3.4, which is the resulting version from fusing two versions coming from the same OCR system but with different resolutions. The expected results was to have errors in the better resolution version as a subset of the errors in the low resolution version, and the expected fused version had to be of no better quality than the one of the higher resolution version. However, results shows that there is always improvement in text quality from fusing different versions of text coming from the OCR system but with different resolutions, it was discovered that errors in both version aren't totally dependent or subset of each other, and the OCR system performance changes with different resolution (in some cases lower resolution version gave better quality text than higher resolution one). Results show that this improvement in quality can reach 63% error reduction in WER, and this process is never harmful.

## 3.5. Fusion Effect on Information Retrieval Improvement

In this section, Fusion effect will be tested on different sets of text and the effect on information retrieval improvement will be checked.

### 3.5.1. Data Set

In order to perform information retrieval test, an electronic version of ZAD was available that was manually written and was error free. The electronic version consists of 2,730 separate documents. Associated with the documents are a set of 25 topics and relevance judgments, which were built by exhaustive judgment of the documents (which will be useful for IR tests). The number of relevant documents per topic ranges between 3 and 72 and averages around 20. The average query length is 5.4 words (Darwish and Oard, 2002).

As scanning the full pages of the printed version of the book would be a very exhausting process, especially when using different resolutions and different OCR systems for recognition. Magdy and Darwish (Magdy and Darwish, 2006a, 2006b, 2008) had an OCRed version of ZAD that was scanned at 300x300 dpi and then OCRed using an older version of Sakhr automatic reader (version 4). Eight pages containing about 4,200 words with Character Error Rate (CER) of 13.9% and Word Error Rate (WER) of 36.8%, were selected at random from the OCRed text and manually corrected. The degraded and clean versions were used to build an error model that was subsequently used to train a garbler that attempts to introduce errors similar to those of the OCR system. OCR degradation was modeled using a noisy channel model in which the observed characters result from the application of some distortion function on the real characters (Magdy and Darwish, 2006a, 2006b). 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  $C_i$ ,  $\epsilon$  representing the null character,  $L$  representing the position of the letter in the word (beginning, middle, end, or isolated – Arabic characters change shape

depending on their positions in words), and # marking word boundaries, the probability estimates for the three edit operations for the models, are:

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

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

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

The resulting character-level alignments were 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.

To obtain different levels of degradation, the character error rate (CER) was tuned with tuning variable  $k$ .

$$P_{\text{new}}(C_i \rightarrow D_j) = \begin{cases} k \cdot P_{\text{original}}(C_i \rightarrow D_j) & , C_i \neq D_j \\ P_{\text{original}}(C_i \rightarrow D_j) + (1-k) \cdot (1 - P_{\text{original}}(C_i \rightarrow D_j)) & , C_i = D_j \end{cases}$$

Where  $P_{\text{original}}$  and  $\text{CER}_{\text{original}}$  are the calculated edit operation probability and original CER respectively,  $k$  is the tuning factor, and  $P_{\text{new}}$  is the new edit operation probability.  $C_i = \varepsilon$  and  $D_j = \varepsilon$  for insertion and deletion respectively. The new CER  $\text{CER} = k \cdot \text{CER}_{\text{original}}$ .

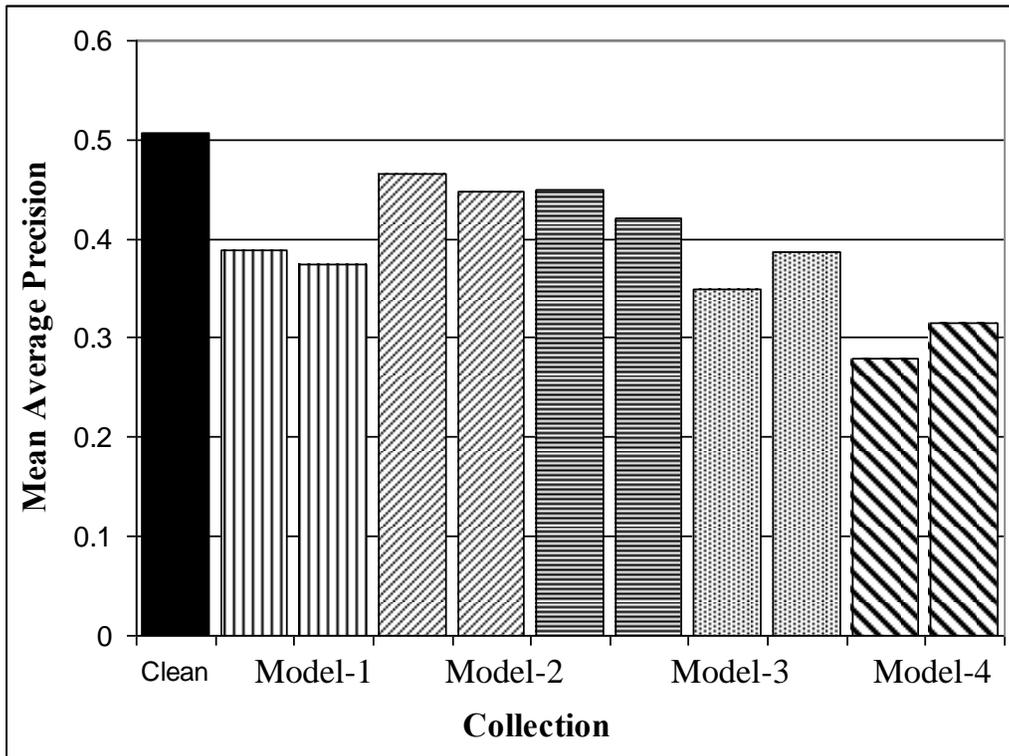
Degradation model was applied on the clean electronic version with different values of  $k$  ( $k = 1, 0.5, 0.66, 1.25, 2$ ). For each value of  $k$ , two degraded versions were produced to check the reliability of the degradation model and the randomness of the generated errors. Results of the generated versions are listed in Table 3.5, which lists the CER, WER, and Out Of Vocabulary (OOV) words (not in the language model training set) for the original OCR text and the synthetically degraded versions. The synthetically degraded version where  $k = 1$  has nearly identical CER, WER, and OOV to the original OCR text. For the rest of this paper, garbled versions will be referred to with the model number shown in Table 3.5 (Model-1 ... Model-5). Between any two garbled versions (either using the same model or different models), there are *common word errors* (CWE) where both models misrecognized a given word, which means that the maximum text improvement with fusion will be limited by the CWE. For example, the two versions of Model-1 have a CWE of 17%, which means that the minimum WER after fusion of these two versions would be 17%.

For all the generated versions, IR tests were performed with mean average precision as the figure of merit to check the effect of garbling on the retrieval effectiveness. Figure 3.2 shows the mean average precision of the garbled versions compared to the clean version.

The index term used for indexing and searching the collection was 4-grams. According to Darwish, character 4-grams are the best index term for Arabic OCR text (Darwish and Oard, 2002). Figure 3.2 shows that the retrieval effectiveness decreases as the WER increases in the collection set. For all degraded versions, IR effectiveness was observed to be statistically different from the clean version. A paired two-tailed t-test with  $p\text{-value} < 0.05$  was used to indicate statistical significance.

**Table 3-5 Produced versions of text set after applying error model with different CER**

<b>Data set</b>	<b>CER</b>	<b>WER</b>	<b>OOV</b>
<b>Original</b>	13.9%	36.8%	20.9%
<b>Model-1</b>	13.9%	36.3%	21.1%
	13.9%	36.4%	21.1%
<b>Model-2</b>	7.0%	20.3%	11.9%
	7.0%	20.4%	11.9%
<b>Model-3</b>	9.3%	26.1%	15.2%
	9.3%	25.9%	15.2%
<b>Model-4</b>	17.4%	43.2%	25.0%
	17.4%	43.3%	24.9%
<b>Model-5</b>	27.9%	59.2%	33.8%
	27.9%	59.2%	33.7%



**Figure 3-2 Results in MAP of searching different versions of the collection**

### 3.5.2. Experimental Setup and Results

Text fusion is tested for the fusion of two and three different degraded versions with different errors. No tests were performed on fusing more than three different versions, as it is unlikely to obtain more than the three independent sources of the same text set.

The retrieval experiments were performed on the clean and fused versions of the text. The collections were indexed and searched using character 4-grams (Darwish and Oard, 2002). For all experiments, Indri search engine toolkit (Abdul-Jaleel et al., 2004) was used with default parameters with no blind relevance feedback. Again, the figure of merit for evaluating retrieval results was mean average precision (MAP), with statistical significance testing done using a paired 2-tailed t-test.

Table 3.6 shows the fusion results of pairs of fused versions coming from different models. The original WER in each model is mentioned under the model name. The resulting WER after fusion and the CWE rate are listed in the top and bottom parts of the cell respectively. Results show that the language model usually selects the proper word between the two candidate words (when at least one of them is the proper word). In many of resulting fusions, the WER was more than halved. Another observation is that fusion is always useful even when fusing a lightly degraded set and a highly degraded set, but the improvement in text quality decreases as the degradation of the fused text sets increases. Fusion can be used in tandem with automatic error correction using a character error model and language modeling, which typically remove more than 50% of the errors [6], to further eliminate more errors.

Table 3.7 shows the fusion results when fusing 3 degraded versions, which yields better results compared to fusing 2 degraded versions.

Figure 3.3 shows the information retrieval results on all the previously mentioned fused versions, where  $M_{ij}$  returns to fusion output from Model-i and Model-j respectively. For all output sets, the MAP of the fused set is better than each

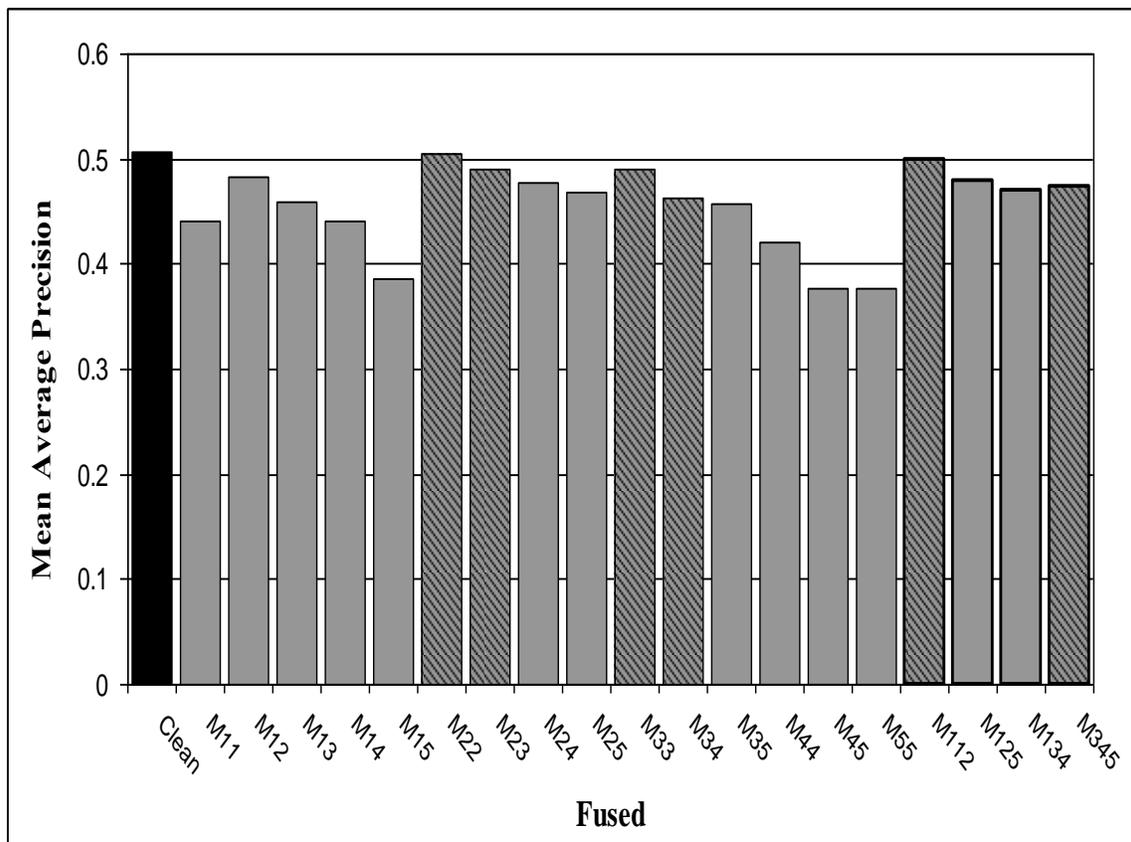
individual set, but not all the output sets from fusion were statistically significant better than the individual sets. Considering that character based error correction has been reported to have little effect on retrieval effectiveness [8], achieving retrieval effectiveness that is statistically indistinguishable from the retrieval effectiveness when searching the clean text version in a few cases is very promising. Further, combining fusion with character level and language model based correction may have significant impact on retrieval effectiveness. Perhaps, OOV words, which contribute more than half the remaining errors after fusion, can be the primary targets of such correction.

**Table 3-6 Results of fusion of each two different text set versions. Each cell is formed of upper and lower values, upper value is the WER after fusion of the two models opposite to the cell, lower value represents the common error between these two versions of text and which is the limit if WER**

Model-1	17.4%				
36.3%	17.0%				
Model-2	10.0%	6.0%			
20.3%	9.6%	5.7%			
Model-3	12.7%	7.4%	9.3%		
26.1%	12.2%	7.0%	8.9%		
Model-4	20.4%	11.7%	14.9%	23.9%	
43.2%	19.9%	11.2%	14.4%	23.4%	
Model-5	26.8%	15.4%	19.5%	31.6%	42.2%
59.2%	26.3%	14.9%	19.0%	31.0%	41.6%
	Model-1	Model-2	Model-3	Model-4	Model-5
	36.4%	20.4%	25.9%	43.3%	59.2%

**Table 3-7 Results of fusion of some triples of models, Model-1-1-2 returns to fusion of two different versions of model-1 and one version of model-2**

	Common Error	WER
Model-1-1-2	5.3%	5.9%
Model-1-3-4	7.8%	8.4%
Model-1-2-5	7.8%	8.5%
Model-3-4-5	11.7%	12.3%



**Figure 3-3 Results in MAP of searching different fused models, hashed bars refers to statistical significant retrieval results better than the original degraded versions**

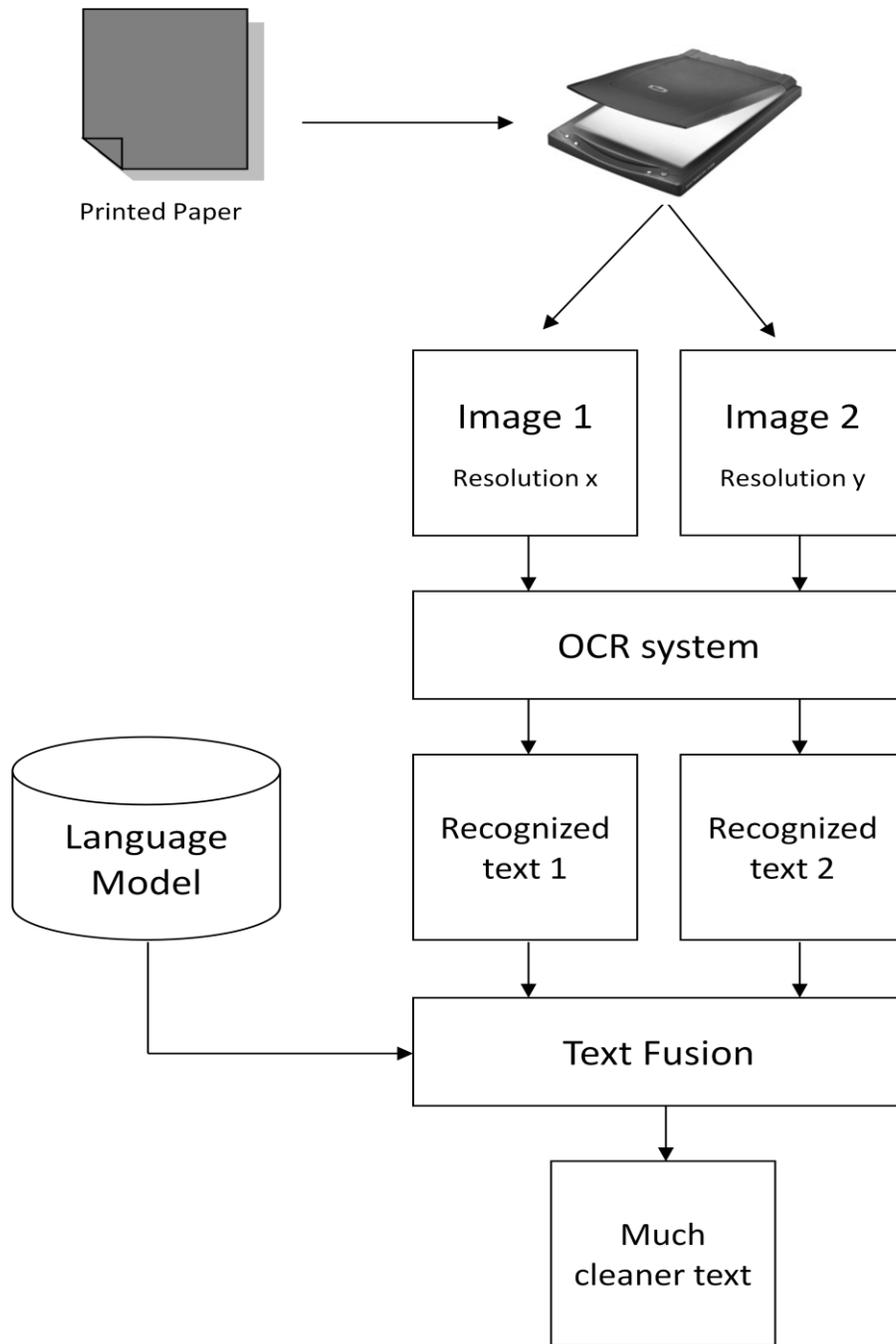
### **3.6. Conclusion and Future Work**

In this chapter, Fusion of different versions of the same source data, and from different OCR systems and with different resolutions and fonts were tested. Fusion was proved to be never harmful; it always results with a better version than the original versions. The strong and surprising observation was that fusion of different versions coming from the same OCR system but with different resolutions gives a better version of text too, and this conclusion was proved with using 3 different fonts and two different OCR systems, and this can be considered as the main conclusion of this chapter.

Figure 3.4 shows the proposed main application for fusion based on the conclusion of this chapter.

Also fusion effect on information retrieval effectiveness was tested, and it was found that the effectiveness depends upon the quality of the fused versions of text and the CWE between them.

For Future work, applying fusion in conjunction with character level and language modeling based correction can further eliminate much of the errors in the text. Furthermore, character based fusion can be one of the interesting things, where a much deeper alignment to character level after word alignment can be done, and fusion on character level can be tested.



**Figure 3-4 Preferred implementation for the fusion system**

## **4. Omni Font OCR Error Correction**

In this chapter a new technique is proposed for correction of OCR degraded text that is independent of character-level OCR errors, and hence independent of scanned document source. It is based on language modeling in conjunction with a uniform character model that uses edit distance only. The technique compares well to state-of-the-art correction techniques that are based on language modeling and source-specific character error models. Although the proposed technique yielded lower correction effectiveness, its impact on retrieval effectiveness is statistically significant and at par with state-of-the-art correction techniques. The main requirement of the proposed technique is the training of a “good” language model matching genre, style, and temporal coverage. The advantage of being independent of character level errors is clear in applications where printed documents vary in source, font, and degradation level, and which is the case of text output from fusion process.

## 4.1. Introduction

Recent work in OCR correction depends on the presence of a source specific character error model for the OCR output text, which makes the correction systems depends on font, OCR system, and printed document quality and requires training examples to build error models (Magdy and Darwish, 2006a, 2006b, 2008). In case of text fusion, and as the output of the process is text from different sources with different character error models, the creation of character error model independent technique for error correction is an essential task. Furthermore, in applications where documents are obtained from heterogeneous sources, building a character level model for every font type and size and every degradation level is impractical. This chapter introduces a general method for error correction that does not require the training of a character error model. The correction method relies on a uniformly distributed character error model based on the edit distance between a misrecognized word and a candidate correction with the assistance of a domain specific language model. This approach is well suited for situations where document are obtained from a variety of sources and for fused text correction. The approach is compared to previously reported approaches that use character level models to examine correction effectiveness and consequent retrieval effectiveness. Although the approach is tested on Arabic OCR text documents, the approach is potentially applicable to text that is degraded using different processes from different languages.

The chapter is organized as follows: Section 2 provides some information about the data used for testing the approach; Section 3 presents the error correction methodology; Section 4 describes the experimental setup; Section 5 reports and discusses results; and Section 6 concludes the chapter and provides possible future directions.

## 4.2. Data set

To test the correction system, two document collections from two different domains were used. The first collection was ZAD collection (mentioned in the previous chapter); The OCRed text version of the whole book was used for test. The second collection is the Text Retrieval Conference (TREC) 2002 Cross-Language IR (CLIR) track collection; for brevity, it is referred here simply as the TREC collection. It contains 383,872 articles from the Agence France Press (AFP) Arabic newswire. NIST developed 50 topics for the collection 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. Character error model built from ZAD (section 3.5.1) was used to degrade the TREC collection in order to obtain synthetically degraded collection of text.

After degradation, as set of 4,000 words were randomly picked and were found to have a 31% WER. The words were manually corrected to train a character level model to compare correction with and without training such a model. Another set of sentences, composed collectively of 6,000 words, were randomly picked, corrected, and set aside for testing.

For all words in both collections, the different forms of *alef* (*hamza*, *alef*, *alef maad*, *alef with hamza on top*, *hamza on wa*, *alef with hamza on the bottom*, and *hamza on ya*) were normalized to *alef*, and *ya* and *alef maqsoura* were normalized to *ya*.

### 4.3. Error Correction and Experimental Setup

Since the proposed approach does not use a trained source-specific character-level error model, Levenshtein edit distance is used instead with uniform probability distribution for different edit operations. In other words, all substitutions, deletions, and insertions are considered equally likely. For a given OCR'ed word  $w_{OCR}$ , a dictionary is checked and the closest  $m$  candidate corrections  $W_{cand} = \{ w_1, \dots, w_i, \dots, w_m \}$  are ranked according to their edit distance and unigram probability of observing a word in text according to the following similarity function  $S_{ED}$ :

$$S_{ED}(w_i) = \underbrace{e^{-C \cdot ED(w_{OCR}, w_i)}}_{CharacterModel} \cdot \underbrace{P(w_i)}_{UnigramModel} \quad (2)$$

Where  $ED$  is edit distance between  $w_{cand}$  and  $w_i$ ,  $P(w_{cand})$  is the unigram probability of  $w_{cand}$  in the dictionary, and  $C$  is a scaling factor to affect the relative contribution of edit distance ( $C$  is proportional to the effect of edit distance). This will be referred to as the ‘‘ED’’ model.

Best  $N$  candidates will be selected according to the previous formula and a language model is used to select the best candidate correction according to context.

To properly compare to state-of-the-art correction, an alternative segment based character error model was trained as described by Magdy and Darwish (Magdy and Darwish, 2006a). This model will henceforth be referred to as the ‘‘REF’’ model. Formally, for a given degraded word  $w_{OCR} = \#D_l..D_x..D_y..D_o\#$ , a set of possible correction  $W_{cand} = \{ w_1, \dots, w_i, \dots, w_m \}$ , where  $w_i = \#C_{il}..C_{ik}..C_{il}..C_{in}\#$ , the null character  $\varepsilon$ , and the word boundary marker  $\#$ , the probability estimates for the three edit operations for the models are:

$$P_{\text{substitution}}(C_k..C_l \rightarrow D_x..D_y) = \frac{\text{count}(C_k..C_l \rightarrow D_x..D_y)}{\text{count}(C_k..C_l)} \quad (3a)$$

$$P_{\text{deletion}}(C_k..C_l \rightarrow \varepsilon) = \frac{\text{count}(C_k..C_l \rightarrow \varepsilon)}{\text{count}(C_k..C_l)} \quad (3b)$$

$$P_{\text{insertion}}(\varepsilon \rightarrow D_x..D_y) = \frac{\text{count}(\varepsilon \rightarrow D_x..D_y)}{\text{count}(C)} \quad (3c)$$

The similarity function  $S_{\text{REF}}$  between the  $w_{\text{OCR}}$  and a candidate correction  $w_i$  combines the character transformation probability with the unigram probability of observing the proposed correction in the text as follows:

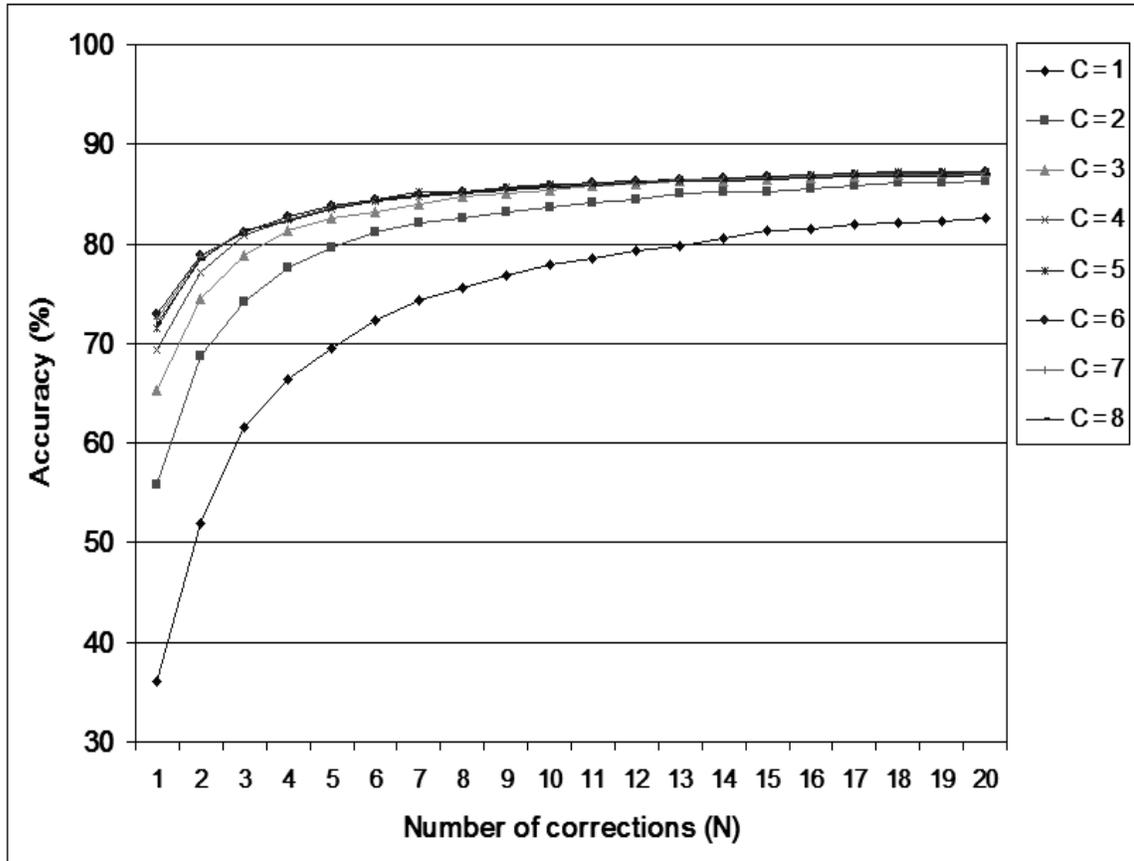
$$S_{\text{REF}}(w_i) = \underbrace{\prod_{\text{all:}D_x..D_y} P(D_x..D_y|C_k..C_l)}_{\text{CharacterModel}} \cdot \underbrace{P(w_i)}_{\text{UnigramModel}} \quad (4)$$

### 4.3.1. Candidates Selection

To get  $m$  initial candidates that are similar to the OCR'ed word, all words in dictionary are indexed as combinations between letters unigrams, bigrams, trigrams, and the word length. For example: “example”  $\rightarrow$  {e, x, a, m, p, l, e, #e, ex, xa, am, mp, pl, le, e#, #ex, exa, xam, amp, mpl, ple, le#, <NO>7</NO>}. A given OCR'ed word is used as a query with the same format, but instead the word length will be a range from length-1 to length+1 to allow the presence of deletion or insertion of characters. For experiments in this paper, the Indri search toolkit (Abdul-Jaleel, 2004) was used to index the dictionary and run queries. For each OCR word, the top 1,000 ( $m = 1,000$ ) retrieved words are scored according to the similarity function.

### 4.3.2. Constant “C” selection

In order to obtain the best value of the scaling factor  $C$  in equation (2), some offline correction experiments were performed with different values of  $C$ , namely 1, 2, 3, 4, 5, 6, 7, and 8. Experiments were performed on the ZAD collection, and the presence of the proper correction among best  $N$  candidate corrections were noticed as shown in Figure 1.



**Figure 4-1 Accuracy vs. number of candidate corrections for ZAD set with different values of C. Accuracy refers to the presence of a proper correction among the N best candidate corrections.**

From graph, the probability to find the proper correction increases as more candidates are taken, and the accuracy nearly saturates after  $N=10$ . As shown in the graph, higher values for  $C$  give better performance than lower values, and the best performance was at  $C=5$  where the accuracy reached 83.8% at  $N=5$ , 86% at  $N=10$ , and 87.3% at  $N=20$ . For the remainder for the paper,  $C$  will be used with a value of 5.

### 4.3.3. Language Modeling

For language modeling, a trigram language model was trained without any kind of morphological processing. The language model was built using the SRILM toolkit (Stolcke, 2002) with Good-Turing smoothing and default backoff.

Given a corrupted word sequence  $\Delta = \{\delta_1 \dots \delta_i \dots \delta_m\}$  and  $\Xi = \{X_1 \dots X_i \dots X_m\}$ , where  $X_i = \{\chi_{i0} \dots \chi_{iN}\}$  are possible candidate corrections of  $\delta_i$  (where  $N$  is the number of candidates corrections taken), the aim was to find a sequence  $\Omega = \{\omega_1 \dots \omega_i \dots \omega_m\}$ , where  $\omega_i \in X_i$ , that maximizes:

$$\underbrace{\left( \prod_{i=1..m, j=1..N} P(\chi_{ij} | \chi_{i-1, j}, \chi_{i-2, j}) \right)}_{\text{LanguageModel}} \cdot \underbrace{e^{-\beta \cdot ED(\delta_i, \chi_{ij})}}_{\text{CharacterModel}} \quad (5)$$

where  $\beta$  is the scaling factor to affect the relative contribution of the edit distance ( $\beta$  is proportional to the effect of edit distance).

When combining language modeling with the REF model, the goal is to maximize (Magdy and Darwish, 2006a)

$$\underbrace{\left( \prod_{i=1..m, j=1..N} P(\chi_{ij} | \chi_{i-1, j}, \chi_{i-2, j}) \right)}_{\text{LanguageModel}} \cdot \underbrace{\prod_{all: D_x \dots D_y} P(D_x \dots D_y | C_k \dots C_l)}_{\text{CharacterModel}} \quad (6)$$

#### 4.3.4. Testing Correction Effectiveness

To test the effectiveness of correction, two types of tests were performed. The first examined the reduction in word error rate, and the second observed the effect of correction on retrieval effectiveness. The first test was applied to both collections, while the later was applied to the ZAD collection only. The reasons why the second test was not performed on the TREC collection are explained later. In examining the reduction in word error rate for the ZAD and TREC collections, the top  $N$  candidate corrections, with  $N$  varying between 1 and 20, are examined to determine if the proper correction is among them. When using language modeling, the effect of the scaling factor  $\beta$ , which is proportional to the effect of edit distance, is examined at different values of  $\beta$  ( $\beta = \{1, 2, 4, 8\}$ ) and the top correction being considered by the language model were either 5 or 10.

The same language model mentioned in chapter 3 was for the ZAD collection. For the TREC collection, all the text from the TREC collection that was not part of the character level model training set and not from the test set was used to build a language model, which will be referred to henceforth as the AFP-LM model. Using TREC collection for training the LM was the reason behind not testing the IR effectiveness on TREC, as it is not practical to use the training set as the test set. Another language model was trained from a web-mined collection of Arabic newswire articles from the BBC, Al-Ahram newspaper, Al-Jazeera news site, Al-Wafd newspaper, and Al-Moheet news site. This language model will be referred to as the News-LM model. Unfortunately, the news articles in this collection do not span the same time period as the TREC collection.

Correction effectiveness was tested on sets of 2,000 and 6,000 words for the ZAD and TREC collections respectively.

The effect of correction on retrieval effectiveness was examined for the ZAD collection. The retrieval experiments were performed on the clean, corrupted, and corrected versions of the ZAD collection described above. The versions of the collection were indexed and searched using words, character 3-grams, character 4-

grams, and lightly stemmed words obtained using Al-Stem (Gey and Oard, 2001). For all experiments, Indri was used with default parameters with no blind relevance feedback. The figure of merit for evaluating retrieval results was mean average precision (MAP). Statistical significance between different retrieval results was performed using a paired 2-tailed t-test and a Wilcoxon test with continuity correction with  $p$ -values  $\leq 0.05$  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).

## 4.4. Experimental Results

Table 4.1 reports the percentage of words for which a proper correction was not found in the top  $N$  generated corrections for both test sets using the ED and REF models. The change in percentage of words for which correction failed is faster with increasing  $N$  for the ED model compared to the REF model. This results in a narrowing of the gap between the two models for the ZAD collection from 6.2% difference to 2.2% difference. The difference between the percentages for varying values of  $N$  for the TREC collection was surprising small with the ED model slightly outperforming the REF model for large  $N$ 's. These results are promising, because they suggest that using a language model to aid in picking the most likely correction is likely to lessen the impact of not using a trained character model. Further, the chances of finding proper corrections beyond 10 corrections are greatly diminished.

Tables 4.2 and 4.3 show the effect of using a trigram language model in conjunction with edit distance in reranking the top 5 and top 10 candidate corrections with different values of  $\beta$  for the ZAD and TREC test set respectively. The results show that using the top 10 corrections is better than using just the top 5. The best values for  $\beta$  were 2 and 4 for the ZAD and TREC collections respectively. Table 3 highlights the fact that the use of a better language model, such as the AFP LM that is trained on a set that matches style and temporal coverage of the text to be corrected, yields better correction effectiveness compared to the use of another less matching language model such the News LM model. In fact, Table 5 shows that compared to using no language modeling, utilizing the AFP LM had a visible effect on WER (more than 5% drop in WER), while utilizing the News LM had minimal effect on WER (less than 1% drop in WER). Since using the AFP LM for correcting the AFP collection would not be appropriate (it would tantamount to using the same set for training and testing) and

the use of the News LM yields minimal improvements, the authors elected to run IR experiments for the ZAD collection only.

Table 4.4 and Table 4.5 compare the correction effectiveness when employing a trained vs. a uniform character error model with and without language modeling for the ZAD and TREC collections respectively. The results show that using a trained character level model yields noticeably better correction effectiveness compared to using the proposed uniform character error model. However, as will be shown later, the difference in effect on retrieval effectiveness is less dramatic.

For IR experiments, language modeling was used to correct the ZAD collection with  $N=10$ , and the corrected versions (with ED and REF models) were compared to each other and to the clean and original OCR'ed versions. For the reasons mentioned earlier, IR experiments were performed on the ZAD collection only. Figure 2 summarizes the retrieval results of searching the clean, OCR'ed (bad), and corrected (with character model and edit distance) versions of the ZAD collection using words, light stems, character 3-grams, and character 4-grams. Table 4.6 reports the  $p$ -values of paired 2-tailed t-tests and Wilcoxon tests comparing the retrieval effectiveness when using language modeling with the ED and REF models for different index terms. Table 4.7 provides the  $p$ -values of the paired 2-tailed t-tests and Wilcoxon tests of comparing the results for both corrections models with language modeling to the clean and original OCR'ed versions. 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 both correction models, character 3-grams – as an index term – achieved the highest MAP and error correction statistically significantly improved retrieval effectiveness, and retrieval effectiveness was statistically indistinguishable from the effectiveness of retrieving from the clean version. The same was true for character 4-grams when using the REF model with language modeling. Table 6 shows that using either model produced statistically indistinguishable retrieval effectiveness. Contrary to the reports in the literature

(Magdy and Darwish, 2006b), the results suggest that “good” error correction with and without a source-specific character model could statistically significantly improve retrieval effectiveness (and possibly be statistically indistinguishable from retrieving clean documents). It is not clear whether this would be applicable to other languages, but the results indicate that this might be the case.

The results suggest that training a character error model yields more effective correction, but the effect of correction on retrieval effectiveness is uncertain. Further, training a character error model is often disadvantageous due to its dependency on font size and type, OCR system, scanned paper quality, and other factors. On the other hand, the correction technique proposed in this paper does not require the training of character level models and achieves comparable retrieval results. This approach is more practical for applications where printed pages are obtained from a variety of heterogeneous sources.

**Table 4-1 Percentage of words for which proper correction was not found in top N corrections**

Number of correction	Model	1	2	3	5	10	20
ZAD	ED	28.4	21.4	18.8	16.2	14.0	12.7
	REF	22.2	16.9	15.0	13.2	11.5	10.5
TREC	ED	12.6	7.6	5.7	4.2	3.1	2.6
	REF	11.1	6.6	5.6	4.6	4.0	3.7

**Table 4-2 WER for different values of  $\beta$  for ZAD test set**

	$\beta$	WER	
		5 Corrections	10 Corrections
ZAD	1	23.8%	21.0%
	2	20.2%	17.2%
	4	23.2%	21.3%
	8	34.1%	34.5%

**Table 4-3 WER for different values of  $\beta$  for TREC test set using different LMs**

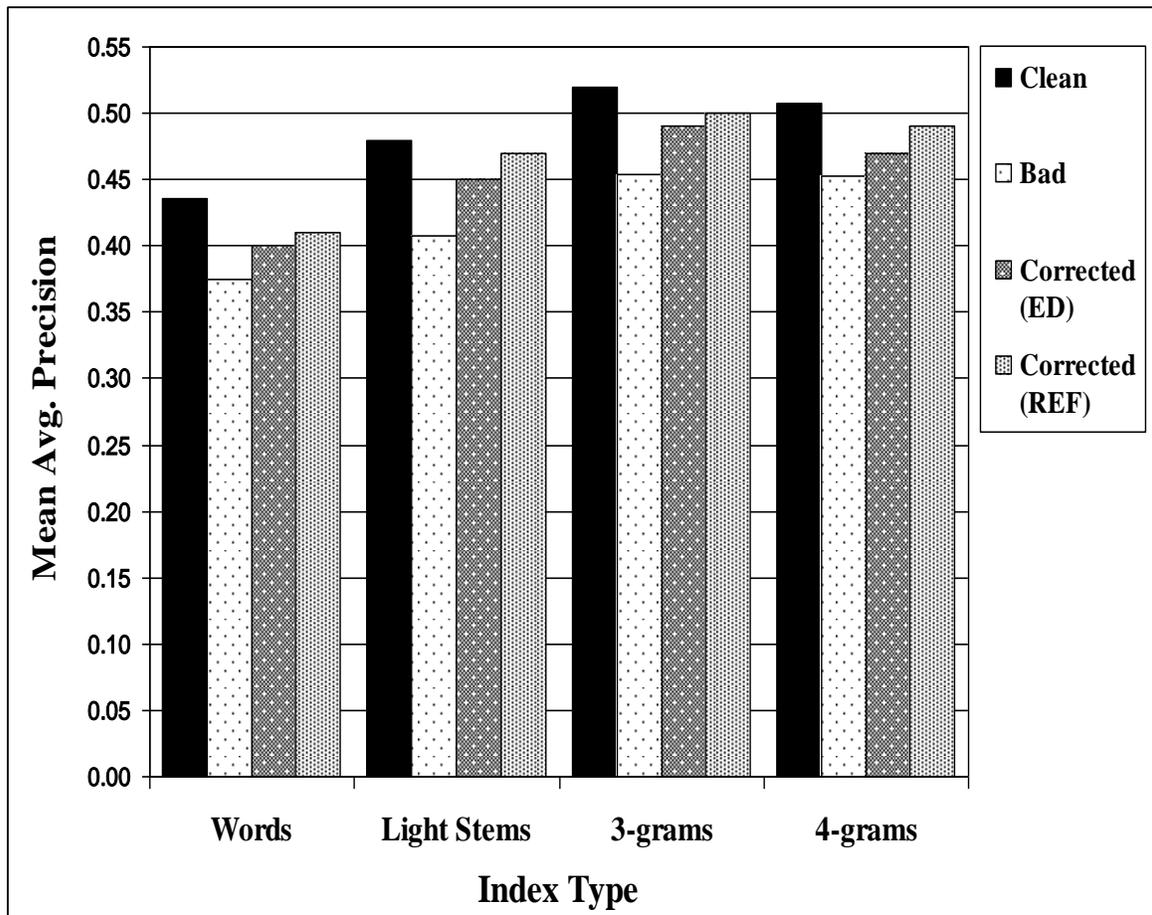
LM	$\beta$	WER	
		5 Corrections	10 Corrections
AFP LM	1	12.8%	13.2%
	2	9.0%	8.6%
	4	7.9%	7.3%
	8	11.7%	12.1%
News LM	1	23.2%	25.4%
	2	15.6%	16.2%
	4	11.8%	11.7%
	8	14.5%	15.0%

**Table 4-4 Comparing Correction effectiveness with and without using character model for ZAD set**

		WER	Error Reduction
<b>Uniform Character Model</b>	<b>Baseline</b> using unigram	28%	27%
	<b>LM</b>	17%	56%
<b>With Character Model</b>	<b>Baseline</b> using unigram	22%	44%
	<b>LM</b>	12%	70%

**Table 4-5 Comparing Correction effectiveness with and without using character model TREC set**

		WER	Error Reduction
Uniform Character Model	Baseline	12.6%	59%
	AFP LM	7.3%	76%
	News LM	11.7%	62%
Trained Character Model	Baseline	11.1%	64%
	AFP LM	5.9%	81%
	News LM	10.7%	65%



**Figure 4-2 Results in MAP for searching different versions of the ZAD collection**

**Table 4-6 p-values of paired 2-tailed t-test and Wilcoxon tests comparing the retrieval effectiveness when using language modeling with the ED and REF models for different index terms**

Index Term	t-test	Wilcoxon
Word	0.14	0.19
Light Stem	0.19	0.35
3-grams	0.33	0.42
4-grams	0.25	0.28

**Table 4-7 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**

		ED Model		REF Model	
		t-test	Wilcox	t-Test	Wilcox
Word	Clean	0.05	<b>0.01</b>	0.11	<b>0.02</b>
	Bad	0.18	0.10	0.08	0.05
Stem	Clean	0.08	<b>0.01</b>	0.33	<b>0.05</b>
	Bad	0.02	0.01	0	0.00
3-g	Clean	0.05	0.06	0.08	0.12
	Bad	0.04	0.03	0.03	0.02
4-g	Clean	0.06	0.05	0.17	0.18
	Bad	0.15	0.17	0.04	0.04

## 4.5. Conclusion and Future Work

The Chapter examined a technique for OCR error correction based on language modeling and a uniform character model that uses edit distance only and compares to state-of-the-art correction techniques based on language modeling and trained character error level models. Although the proposed technique yielded lower correction effectiveness, its impact on retrieval effectiveness is statistically significant and at par with state-of-the-art correction techniques. The main requirement of the proposed technique is the training of a “good” language model matching genre, style, and temporal coverage. The advantage of using a character model independent technique is clear in applications where printed documents vary in source, font, and degradation level and are potentially scanned and OCRed using different systems. Further, contrary to previously published work (Magdy and Darwish, 2006b), this paper showed that using “good” error correction can have a statistically significant impact on retrieval effectiveness.

For future work, the proposed technique needs to be tested on heterogeneous printed sources and potentially other degradation sources such as automatic speech recognition. A Factored language model might prove beneficial to incorporate morphological information and other factors such as part of speech tags to improve the correction ability. This can be instrumental in overcoming the problem of correcting out-of-vocabulary words. In addition, the automatic induction of a trained character error model might prove useful and deserves examination. Finally, word prediction might prove useful for cases where OCR grossly misrecognized words.

## **5. Integrated System**

This chapter describes the results of integrating the two proposed techniques for error reduction of OCRed text. The chapter shows the best way for integrating Omni-font correction engine to the fusion engine and its effect on the error reduction.

## **5.1. Introduction**

Previous chapters proposed two effective techniques for reducing the amount of errors in an OCRed text. Text fusion of different versions of the same data has proved to be never harmful, and the amount of gain in error reduction depends on the common errors among the fused versions. On the other side, Omni-font correction proved its effectiveness for error reduction overcoming the absence of character error model. In this chapter, best integration of both systems is described, and the gain from error reduction prospective is reported.

This chapter is organized as follows: Section 2 proposes the best integration of the two systems; Section3 describes the experimental setup and reports results; and Section 4 concludes the chapter and provides possible future directions.

## 5.2. Integrated System Architecture

There are two possible ways for integrating both systems as shown in figure 5.1. In figure 5.1(a) different versions of OCR'd text will be fused resulting a newer version of less errors than any of the original versions. The fused version is then corrected using Omni-font correction engine in order to reduce more errors. The other implementation for the integrated system is shown in figure 5.1(b), where all versions are corrected using the correction engine, then corrected versions are then fused to result a much less errors version of the text.

Implementation 5.1(a) is much preferred than implementation 5.1(b) according to following reasons:

Implementation (a) applies correction on only one version (fused version), which makes it much faster than the other implementation where correction is applied to the  $n$  versions.

Fusion process takes the full advantage of OOV words in the OCR'd versions. However in case of applying correction before fusion, there will no OOV words, which makes it more difficult to the fusion engine, and the possibility of system confusion becomes higher.

Given the two mentioned advantages, Implementation (a) is used as the integration between the fusion and correction engines. For the correction phase, there are two different options for applying correction. First one is correcting the OOV words only, although this option has less chance in correcting more words, it is considered much safe, as it will never change a correct word (assuming all correct words are in dictionary). The second option is applying correction over all the fused text words; this option has the opportunity of correcting all errors in text, however, it is considered more risky as it can change a correct word.

In this chapter, implementation (a) is tested with its different correction options, and results are observed.

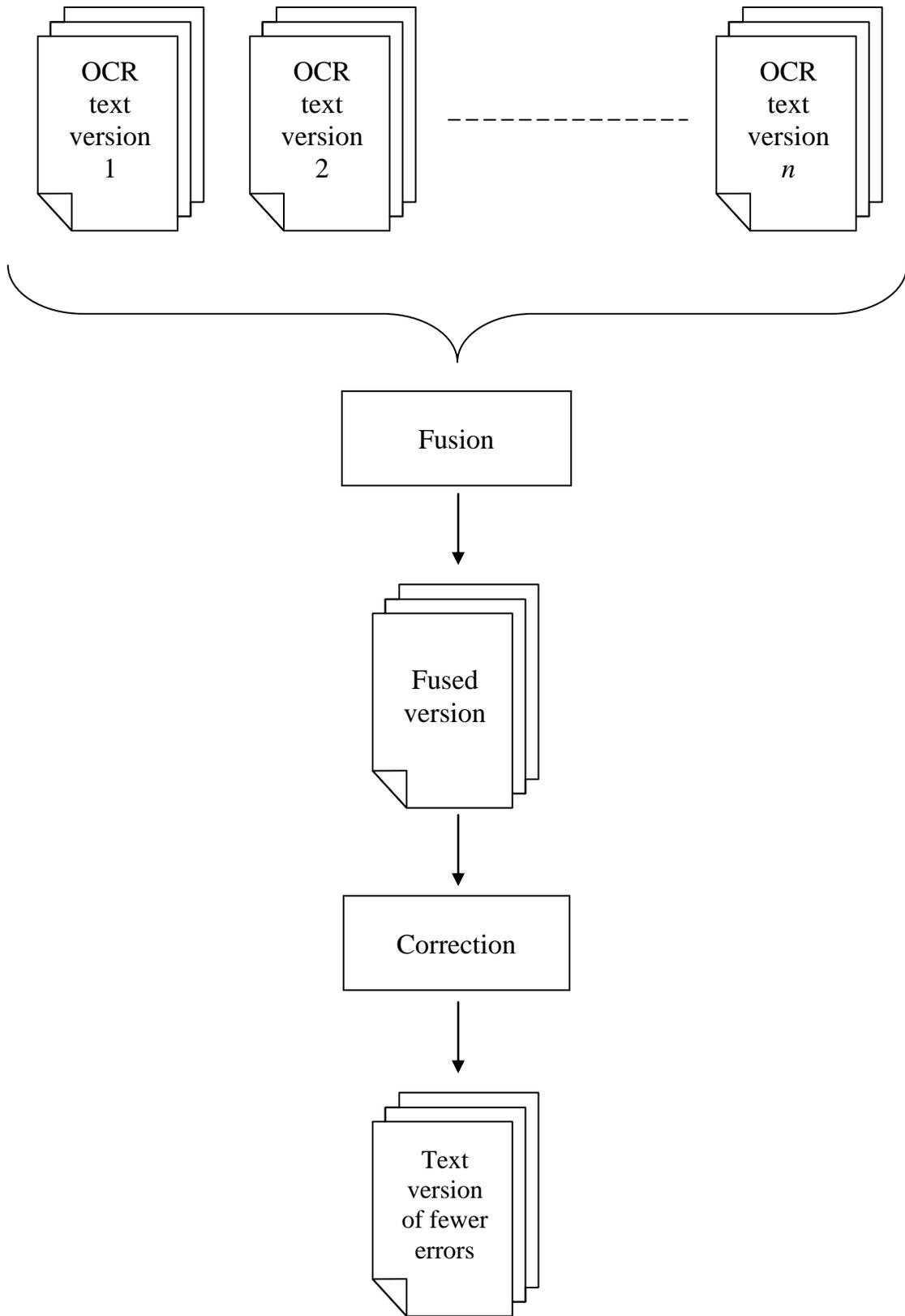


Figure 5.1 (a)

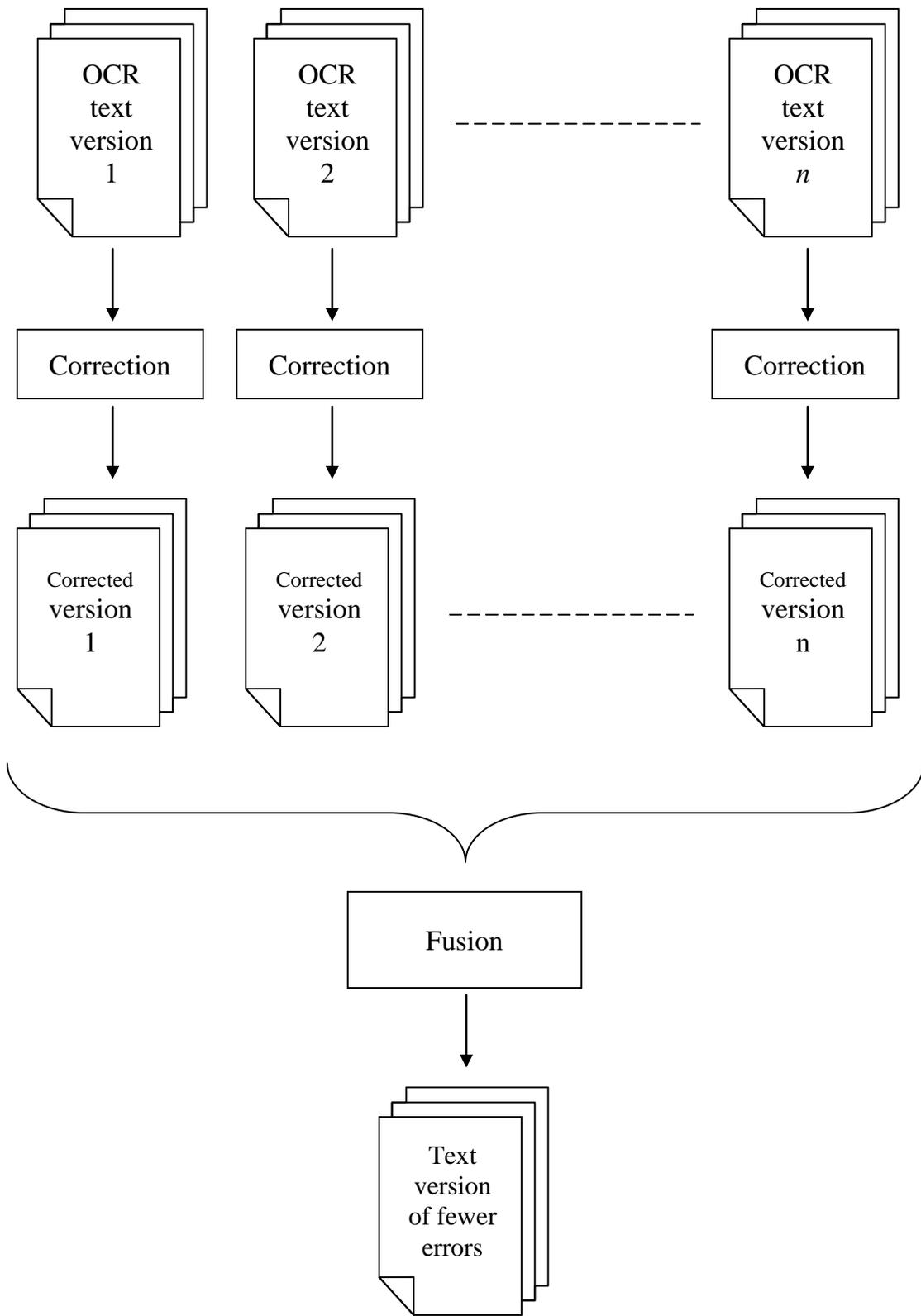


Figure 5.1 (b)

Figure 5-1 Possible implementations for the integrated system

### 5.3. Experimental setup and results

Omni-font correction was applied to the resulting fused versions of ZAD showed in section 3.4.2. Table 5.1 shows some statistics on the available versions for the test. For each version character error rate, word error rate, and out of vocabulary rate are calculated. Table 5.1 shows that 55% of the WER are OOV on average, which means that the maximum error reduction that can be achieved when applying correction to OOV only will be 55% on average.

Correction was applied to each fused version in two different manners. First one tests correction of the OOV words only in the fused version, and the second tests correction of the whole text based on the assumption that any word can be incorrect. Results are reported in table 5.2.

**Table 5-1 Error rates and OOV rates for fused versions of ZAD<sup>7</sup>**

Code name	CER	WER	OOV	Version resulting from the fusion of:
Clean	-	-	0.88%	The clean version of the text
K.200	3.41%	8.73%	3.45%	RDI and Sakhr Kufi versions scanned @200 dpi
K.300	2.49%	6.10%	3.14%	RDI and Sakhr Kufi versions scanned @300 dpi
K	2.07%	4.97%	1.76%	All RDI and Sakhr Kufi versions
M.200	1.19%	3.63%	1.40%	RDI and Sakhr Mudir versions scanned @200 dpi
M.300	0.52%	1.91%	1.35%	RDI and Sakhr Mudir versions scanned @300 dpi
M	0.57%	2.00%	1.02%	All RDI and Sakhr Mudir versions
S.200	4.39%	11.90%	3.82%	RDI and Sakhr Simplified versions scanned @200 dpi
S.300	0.70%	2.50%	1.75%	RDI and Sakhr Simplified versions scanned @300 dpi
S	0.58%	2.10%	1.33%	All RDI and Sakhr Simplified versions
RDI.K	1.27%	4.03%	3.98%	RDI 200 and 300 dpi Kufi versions
RDI.M	0.72%	2.45%	1.79%	RDI 200 and 300 dpi Mudir versions
RDI.S	1.09%	3.98%	2.47%	RDI 200 and 300 dpi Simplified versions
Sakhr.K	15.45%	34.70%	13.28%	Sakhr 200 and 300 dpi Kufi versions
Sakhr.M	2.21%	5.84%	2.65%	Sakhr 200 and 300 dpi Mudir versions
Sakhr.S	1.02%	3.37%	1.71%	Sakhr 200 and 300 dpi Simplified versions

<sup>7</sup> For more details about the original versions of the fused version, refer to table 3.4

From results in table 5.2, as expected, OOV correction never harms the text quality and as shown error reduction ranges from 15% to 53%. On the other hand, applying correction engine over the full text fails to improve the text quality in most of times, but on the contrary it increased the amount of errors significantly that it reached the double in some cases. However, full text correction proved its success with version with high amount of error rates; also it proved its total failure with versions of small amount of errors. As shown in table, “OOV correction” wins “full text correction” in most of time except with the version with the highest amount of error “Sakhr.K” where the WER is nearly 35%. In case of the version “M.300” OOV correction achieved to reach the maximum accuracy limit of the text quality, as it text quality accuracy reached 99.1% through an error reduction of 53% from the correction process, and this is the maximum reachable theoretical accuracy, as the test data has a 0.9% OOV for the language model used in fusion and correction processes.

**Table 5-2 WER and amount of Error reduction for different fused versions of OCRd text using two different methods for correction**

Version	Fused error rates	WER after Correction		Error Reduction	
		OOV Cor.	Full Cor.	OOV Cor.	Full Cor.
K.200	8.7%	7.1%	7.9%	18.9%	9.1%
K.300	6.1%	5.1%	7.2%	15.6%	-18.7%
K	5.0%	4.1%	6.2%	16.8%	-25.7%
M.200	3.6%	2.4%	4.8%	32.6%	-32.8%
M.300	1.9%	0.9%	3.7%	52.7%	-96.5%
M	2.0%	1.0%	3.8%	47.9%	-88.4%
S.200	11.9%	9.4%	9.5%	20.6%	19.8%
S.300	2.5%	1.3%	3.8%	48.7%	-54.0%
S	2.1%	1.8%	4.3%	15.1%	-103.8%
RDI.K	4.0%	3.3%	6.2%	19.0%	-53.7%
RDI.M	2.4%	1.5%	4.1%	39.9%	-68.6%
RDI.S	4.0%	2.4%	4.3%	39.8%	-9.1%
Sakhr.K	34.7%	29.7%	26.3%	14.6%	24.1%
Sakhr.M	5.8%	3.8%	6.2%	34.2%	-6.4%
Sakhr.S	3.4%	2.0%	4.4%	41.6%	-29.6%

## **5.4. Conclusion and Future Work**

In this chapter, the integration of text fusion and Omni-font correction engines was shown. It has been declared that using the correction engine after the fusion process will be more efficient and effective. The results showed that the amount of errors is reduced by one third on average when using correction for the OOV words only. However, applying full text correction proved to be more effective with text of higher rates of errors.

For future work, the second implementation for the integrated system need to be tested even it is expected to me less efficient and effective. Also, a much sophisticated system can be implemented to allow the candidate corrections from all versions to be fused together with using voting techniques for the common candidates from different versions.

## **6. Conclusion and Future Work**

In the research presented by this thesis, different techniques for post-processing of OCR text output were introduced. Two approaches for error reduction in OCR degraded text were tested. All experiments were tested on Arabic language as a reason for the challenges that this languages suffers from. Some experiments tested the approaches on different domains (religious and news domains) in order to prove the effectiveness of the approaches across different domains. Experiments in the thesis measured the impact of OCR degraded text post-processing through two ways: a) the impact on error reduction, and b) the impact in retrieval.

## 6.1. Conclusions

Through all experiments, it has been shown that:

1. Text fusion was introduced and tested on several versions of the same source text. Text fusion proved to be never harmful; it always results with a better version than the original versions. The strong and surprising observation was that fusion of different versions coming from the same OCR system but with different resolutions gives a better version of text too.
2. Text fusion effect on retrieval was tested and the results were sometimes better and sometimes not. The results showed that the main factor on improving the retrieval effectiveness was the amount of error reduction in the produced version, which by its way is dependent on the amount of common errors among the fused versions.
3. A technique for Omni-font OCR error correction was examined based on language modeling and a uniform character model that uses edit distance only and compares to state-of-the-art correction techniques based on language modeling and trained character error level models. Although the proposed technique yielded lower correction effectiveness, its impact on retrieval effectiveness is statistically significant and at par with state-of-the-art correction techniques. The main requirement of the proposed technique was shown to be the training of a “good” language model matching genre, style, and temporal coverage. The advantage of using a character model independent technique is clear in applications where printed documents vary in source, font, and degradation level and are potentially scanned and OCRed using different systems.
4. The integration of text fusion and Omni-font correction engines was tested. It was declared that using the correction engine after the fusion process will be more efficient and effective. The results showed that the amount of errors is reduced by one third on average when using correction for the OOV words

only. However, applying full text correction proved to be more effective with text of higher error rates.

5. Finally, error reduction could reach its limit (where the output WER = OOV of the LM) like in the case of fusing the RDI and Sakhr OCR output of the Mudir font that was scanned at 300dpi, then applying Omni-font correction for the OOV words. On the other hand, error reduction could reach 86% like in case of fusing all versions of the Simplified font, then correcting the OOV words; where the outcome version has an WER of 1.3%, while the minimum WER for all the fused versions was 9.1%.

## 6.2. Future work

The thesis has introduced different techniques for OCR errors reduction, and for improving the retrieval effectiveness of OCRed text. Although many experiments were examined through the thesis, there are further examinations that could be tested in order to achieve better. These examinations can be listed as follows:

1. Fusion proved its effectiveness with error reduction on the word level. However, applying fusion on the character level could be more useful, where in case of the misrecognition of a certain word among all fused versions; the possibility for constricting the proper word from the fusion of characters of the misrecognized words will be very high.
2. Using factored language model instead of normal language model will be an interesting test to overcome the limitation of the training data.
3. Different implementations for the integrated system of fusion and correction need to be tested. Also, a much sophisticated system can be implemented to allow the candidate corrections from all versions to be fused together with using voting techniques for the common candidates from different versions.
4. Collecting a huge amount of data for training a general language model is a potential test instead of using a domain specific language model. The general language model will increase the system robustness for any type of data.
5. Text fusion and degraded text correction can be tested on different types of text other than the OCR output text. ASR text is one of the important domains that the presented systems can be applied on for error reduction in errors. For domain such as recognized speech text, no changes will be needed for the presented systems unless for the alignment process in the fusion engine, that is currently based on aligned lines. For ASR, alignment could be based on silence between sentences.

## References

1. Abdul-Jaleel, N., Allan, J., Croft, W.B., Diaz, F., Larkey, L., Li, X., Metzler, D., Strohman, T., Turtle, H., and Wade, C. UMass at TREC 2004: Notebook. *Text REtrieval Conference*. (TREC 2004), page 657.
2. Abu-Salem, H., M. Al-Omari, and M. Evens. Stemming Methodologies Over Individual Query Words for Arabic Information Retrieval. *JASIS*, 50(6) (1999) 524-529.
3. Agirre, E., K. Gojenola, K. Sarasola, and A. Voutilainen. Towards a Single Proposal in Spelling Correction. *In COLING-ACL'98* (1998).
4. Ahmed, M. A Large-Scale Computational Processor of Arabic Morphology and Applications. *MSc. Thesis, in Faculty of Engineering Cairo University: Cairo, Egypt*. (2000).
5. Amir, A., A. Ehat, and S. Srinivasan. Advances in Phonetic Word Spotting", IBM Research Report RJ 10215, August 2001
6. Andreas Stolcke. SRILM - An Extensible Language Modeling Toolkit. *Proceedings of the Workshop on Statistical Machine Translation*, pages 72—77 (2002)
7. Al-Kharashi, I. and M Evens. Comparing Words, Stems, and Roots as Index Terms in an Arabic Information Retrieval System. *JASIS* 45(8) (1994) 548-560.
8. Baeza-Yates, R. and G. Navarro. A Faster Algorithm for Approximate String Matching. *In Combinatorial Pattern Matching (CPM'96)*, Springer-Verlag LNCS (1996).
9. Barret, W., L. Hutchison, D. Quass, H. Nielson, and D. Kennard. Digital Mountain: From Granite Archive to Global Access," *Proc. of International Workshop on Document Image Analysis for Libraries, Palo Alto, January 2004*, pp. 104-121, (2004).

10. Brill, E. and R. Moore. An improved error model for noisy channel spelling correction. *In the proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 286 – 293 (2000).
11. Church, K. and W. Gale. “Probability Scoring for Spelling Correction.” *Statistics and Computing*, 1: 93-103 (1991).
12. Darwish, K. and D. Oard. Term Selection for Searching Printed Arabic. *In SIGIR-2002* (2002).
13. Darwish, K. and D. Oard. CLIR Experiments at Maryland for TREC 2002: Evidence Combination for Arabic-English Retrieval. *In TREC-2002*, Gaithersburg, MD (2002).
14. Darwish, K. Probabilistic Methods for Searching OCR-Degraded Arabic Text. *PhD thesis, Maryland*, 2003
15. Darwish, K. and W. Magdy. Error correction vs. query garbling for Arabic OCR document retrieval. *TOIS 2007*, Volume 26
16. De Roeck, A. and W. Al-Fares. A Morphologically Sensitive Clustering Algorithm for Identifying Arabic Roots. *In the 38th Annual Meeting of the ACL*, Hong Kong, (2000).
17. Domeij, R., J. Hollman, V. Kann. Detection of spelling errors in Swedish not using a word list en clair. *Journal of Quantitative Linguistics* (1994) 195-201.
18. Fraser, A., J. Xu, and R. Weischedel. TREC 2002 Cross-lingual Retrieval at BBN. *In TREC-2002*. Gaithersburg, MD (2002).
19. Gey, F. and D. Oard. The TREC-2001 Cross-Language Information Retrieval Track: Searching Arabic Using English, French or Arabic Queries. *In TREC-2001*, Gaithersburg, MD (2001).
20. Harding, S., W. Croft, and C. Weir. Probabilistic Retrieval of OCR-degraded Text Using N-Grams. *In European Conference on Digital Libraries* (1997).

21. Hong, T. Degraded Text Recognition Using Visual and Linguistic Context. *Ph.D. Thesis, Computer Science Department, SUNY Buffalo: Buffalo* (1995).
22. Hui Jiang. Confidence measures for speech recognition: A survey. *Speech Communication Volume 45, Issue 4, Pages 455-470.* (2005)
23. Jurafsky, D. and J. Martin. Speech and Language Processing. Chapter 5: pages 141-163. *Prentice Hall* (2000).
24. Kolak, O. and P. Resnik. OCR error correction using a noisy channel model. Proceedings of the second international conference on Human Language Technology Research, (2002)
25. Larkey, L., L. Ballesteros, and M. Connell. Improving stemming for Arabic information retrieval: light stemming and cooccurrence analysis. *In proceedings of the 25th annual international ACM SIGIR conference*, pages 275-282 (2002).
26. 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).
27. Magdy, W. and K. Darwish. Arabic. OCR Error Correction Using Character Segment Correction, Language Modeling, and Shallow Morphology. *In EMNLP 2006*, pages 408 – 414 (2006a)
28. Magdy, W. and K. Darwish. Arabic. Word-Based Correction for Retrieval of Arabic OCR Degraded Documents. *In SPIRE* (2006b)
29. Magdy, W., K. Darwish, and M. Rashwan. Fusion of Multiple Corrupted Transmissions and its effect on Information Retrieval. *ESOLE 2007*
30. Magdy, W. and K. Darwish. Effect of OCR error correction on Arabic retrieval. *Inf Retrieval, DOI 10.1007/s10791-008-9055-y*, 2008
31. Mayfield, J., P. McNamee, C. Costello, C. Piatko, and A. Banerjee. JHU/APL at TREC 2001: Experiments in Filtering and in Arabic, Video, and Web Retrieval. In TREC-2001. Gaithersburg, MD (2001).

32. McNamee, P., C. Piatko, and J. Mayfield. JHU/APL at TREC 2002: Experiments in Filtering and Arabic Retrieval. In TREC-2002, Gaithersburg, MD (2002).
33. Moussa B., M. Maamouri, H. Jin, A. Bies, X. Ma. Arabic Treebank: Part 1 - 10Kword English Translation. *Linguistic Data Consortium* (2003).
34. Oard, D. and F. Gey. The TREC 2002 Arabic/English CLIR Track. In TREC-2002, Gaithersburg, MD (2002).
35. Oflazer, K. Error-Tolerant Finite State Recognition with Applications to Morphological Analysis and Spelling Correction. *Computational Linguistics* 22(1), 73-90 (1996).
36. Sanderson, M. and J. Zobel. Information Retrieval System Evaluation: Effort, Sensitivity, and Reliability. In *SIGIR 2005, Sheffield* (2005).
37. Simske, S. and X. Lin. Creating Digital Libraries: Content Generation and Re-mastering. Proc. International Workshop on Document Image Analysis for Libraries, Palo Alto, January 2004, pp. 33-45, (2004).
38. Stolcke, A. SRILM - An Extensible Language Modeling Toolkit. Proceedings of the Workshop on Statistical Machine Translation, pp. 72—77 (2002)
39. Taghva, K., J. Borsack, and A. Condit. An Expert System for Automatically Correcting OCR Output. In *SPIE - Document Recognition* (1994).
40. Thoma, G. and G. Ford. Automated Data Entry System: Performance Issues. Proc. SPIE Conference on Document Recognition and Retrieval IX, San Jose, 2002, pp. 181-190, (2002).
41. Tillenius, M., Efficient generation and ranking of spelling error corrections. *NADA* (1996).
42. Tseng, Y. and D. Oard. Document Image Retrieval Techniques for Chinese. In *Symposium on Document Image Understanding Technology, Columbia, MD* (2001).