A High Level Approach to Arabic Sentence Recognition

Abdulwahab Gebril Krayem

A thesis submitted in partial fulfilment of the requirements of Nottingham Trent University for the degree of Doctor of Philosophy

September 2013

ABSTRACT

The aim of this work is to develop sentence recognition system inspired by the human reading process. Cognitive studies observed that the human tended to read a word as a whole at a time. He considers the global word shapes and uses contextual knowledge to infer and discriminate a word among other possible words. The sentence recognition system is a fully integrated system; a word level recogniser (baseline system) integrated with linguistic knowledge post-processing module. The presented baseline system is holistic word-based recognition approach characterised as probabilistic ranked task. The output of the system is multiple recognition hypotheses (N-best word lattice). The basic unit is the word rather than the character; it does not rely on any segmentation or require baseline detection. The considered linguistic knowledge to re-rank the output of the existing baseline system is the standard n-gram Statistical Language Models (SLMs). The candidates are re-ranked through exploiting phrase perplexity score. The system is an OCR system that depends on HMM models utilizing the HTK Toolkit. The baseline system supported by global transformation features extracted from binary word images. The adopted features' extraction technique is the block-based Discrete Cosine Transform (DCT) applied to the whole word image. Feature vectors extracted using block-based DCT with non-overlapping sub-block of size 8x8 pixels. The applied HMMs to the task are mono-model discrete one-dimensional HMMs (Bakis Model).

A balanced actual scanned and synthetic database of word-image has been constructed to ensure an even distribution of word samples. The Arabic words are typewritten in five fonts having a size 14 points in a plain style. The statistical language models and lexicon words are extracted from The Holy Qur'an. The systems are applied on word images with no overlap between the training and testing datasets. The actual scanned database is used to evaluate the word recogniser. The synthetic database is a large amount of data acquired for a reliable training of sentence recognition systems. This word recogniser evaluated in mono-font and multi-font contexts. The two types of word recogniser have been used to achieve a final recognition accuracy of99.30% and 73.47% in mono-font and multi-font, respectively. The achieved average accuracy by the sentence recogniser is 67.24% improved to 78.35% on average when using 5-gram post-processing. The complexity and accuracy of the post-processing module are evaluated and found that 4-gram is more suitable than 5-gram; it is much faster at an average improvement of 76.89%.

ACKNOWLEDGEMENTS

First of all, I am thankful to Allah, who gave me the strength and the ability to complete my thesis without His divine help, I would have never been able to complete it.

In fact, there are many people to whom I must express my gratitude for their contributions to the successful completion of this dissertation.

I would like to express my sincere appreciation and thanks to my main supervisor, Prof. Nasser Sherkat, who greatly supported me in completing my work. I will be grateful for his assistance and guidance forever.

I would also like to thank my co-supervisors, Dr. Lindsay Evett and Dr. Taha Osman. I appreciate their comments and encouragement throughout this research work.

Thanks to all my friends and colleagues who have provided invaluable help and exchanged fructiferous views from time to time that has been vital to my progress.

My heartiest thanks are due to my highly respected Mother and my dearest thanks go to my wife, without whose constant support and encouragement, particularly in difficult times, the whole endeavour of writing my thesis would not have been fruitful.

Moreover, I am also thankful to my sons and daughters, and other relatives for their continuous supports and encouragements during my PhD study.

Last but certainly not least, I wish to express my special gratitude to my daughter Hajer for her regular insistence for keeping up my hard work.

TABLE OF CONTENTS

ABSTRACT ..... **IV**

ACKNOWLEDGEMENTS ..... **V**

TABLE OF CONTENTS ..... **VI**

LIST OF FIGURES ..... **XI**

LIST OF TABLES ..... **XIV**

LIST OF EQUATIONS ..... **XVIII**

LIST OF ACRONYMS ..... **XIX**

CHAPTER 1 ..... **1**

INTRODUCTION **1**

1.1 Introduction **1**

1.2 Features of Arabic script **2**

1.3 Arabic Language Challenges **5**

1.4 Problem Definition **6**

1.5 Research Objectives **9**

1.6 Structure of the Thesis **11**

CHAPTER 2 ..... **13**

RELATED CONCEPTS AND LITERATURE REVIEW **13**

2.1 Introduction **13**

2.2 Printed Arabic Text Databases **15**

2.2.1 DARPA 16

2.2.2 APTI 16

2.2.3 MMAC 16

2.3 Feature Extraction and 2D-DCT 18

2.4 Classification and DHMM 23

2.5 HMM-based Printed Text Recogniser 27

2.6 The Re-Ranking Post-Processing Techniques 31

2.7 Summary 36

CHAPTER 3 38

ARABIC WORDS DATABASE PREPARATION 38

3.1 Introduction 38

3.2 Considerations in Building OCR Text Database 39

3.3 Related Works 40

3.4 Corpus Acquisition 45

3.5 The Holy Qur'an Arabic 47

3.6 Database Implementation Methodology 48

3.6.1 Data collection form 50

3.6.2 Data Preparation and Acquiring 53

3.6.3 Pre-processing and Word Segmentation 55

3.6.4 Ground Truth 62

3.7 Summary 68

CHAPTER 4 70

HMM/DCT HOLISTIC WHOLE WORD RECOGNISER 70

4.1 Introduction 70

4.2 Thesis Methodology 71

4.3 Arabic Printed Datasets 73

4.3.1 Word Set Selection Criteria: 73

4.3.2 Samples Distribution: 75

4.3.3 Statistical analysis of the Arabic Qur'an text corpus: 76

4.3.4 Database implementation Process 79

4.4 Proposed Recogniser Methodology 85

4.5 Overfitting Problem 89

4.6 Experiment Parameters 91

4.7 Experimental Results 94

4.7.1 Performance Estimation 95

4.7.2 Classification accuracy 98

4.7.3 Classifier stability test 106

4.8 Comparison with other authors 110

4.8.1 Mono-font comparison 117

4.8.2 Multi-font comparison 117

4.9 Summary and Conclusion 120

CHAPTER 5 122

LINGUISTIC POST-PROCESSING FOR SENTENCE RECOGNITION 122

5.1 Introduction 122

5.2 Proposed Re-ranking Post-Processing System 124

5.2.1 Word-based n-gram models 125

5.2.2 Perplexity 127

5.3 Sentence recognition system 128

5.4 Linguistic post-processing system 129

5.5 Lexicon Dataset and Language Model Creation 129

5.5.1 Quranic-2279UW Database 130

5.5.2 Language Model Creation 132

5.6 Experiments and results 134

5.6.1 Word recogniser 134

5.6.2 Linguistic Post-processing Evaluation 135

5.6.3 Linguistic post-processing Complexity 143

5.7 Comparison with Literature Review 147

5.8 Summary and Conclusion 149

CHAPTER 6 151

CONCLUSION AND FUTURE WORK 151

6.1 Introduction 151

6.2 Conclusion 152

6.2.1 Databases 153

6.2.2 SLMs 155

6.2.3 HMM/DCT Holistic Word Recogniser 155

6.2.4 Integrated Linguistic Post-processing 157

6.2.5 Sentence Recognition System 157

6.3 Limitations 158

6.4 Thesis Contributions 161

6.5 Future Work 163

BIBLIOGRAPHY 166

LIST OF FIGURES

Figure 1.1: Arabic text: Printed and Handwritten sentences 4

Figure 1.2: The characteristics of Arabic text 5

Figure 1.3: The word shape of cursive words alone contains sufficient information to classify the image as one of the lexicon words [1] 8

Figure 1.4: The Sentence recognition system 10

Figure 2.1: DCT coefficients with the zigzag pattern 21

Figure 2.2: DCT coefficients selection (a) zigzag mask (b) zonal mask 21

Figure 2.3: The Markov Generation Model [28] 24

Figure 2.4: Simple Left-Right HMM [28] 25

Figure 2.5: A 4-state ergodic model with start and end states 25

Figure 3.1: The IAM database filled form [44] 42

Figure 3.2: The HIT-MW database collection form layout [45] 43

Figure 3.3: The FHT database form layout details [46] 45

Figure 3.4: Dependency graph for verse (67:1) [51] 47

Figure 3.5: Form layout 51

Figure 3.6: Group and Page reference files 54

Figure 3.7: The representation of a line in the (x, y) space using (ρ, θ) 58

Figure 3.8: Text section 60

Figure 3.9: Line segment 60

Figure 3.10: Segmented word using crop rectangle 61

Figure 3.11: Tidy segmentation of the whole word body 61

Figure 3.12: GT in XML file format for the word (بنورهم) 64

Figure 3.13: GT in XML file format for subset folder 66

Figure 3.14: Database directory structure 67

Figure 4.1: Upper and lower baselines of a text line (Simplified Arabic style of writing) 74

Figure 4.2: A block diagram of the proposed holistic whole word recogniser 85

Figure 4.3: Block diagram of the applied word-image normalization 86

Figure 4.4: DCT coefficients selection using zigzag masking 87

Figure 4.5: Left-to-right HMM model with 5-state (Bakis model) 93

Figure 4.6: Discrete HMM description file (prototype) 93

Figure 4.7: Multi-font-252UW-32CB Performance 103

Figure 4.8: Multi-font-252UW for Andalus top-1, effect of varying the size of codebook on recognition accuracy 104

Figure 5.1: The word lattice produced for the sentence "يكاد البرق يخطف أبصارهم" 125

Figure 5.2: Word image sample from true scanned and synthetic word images printed in Thuluth (Deco Type) font 131

Figure 5.3: Andalus font recognition accuracy before and after the post-processing 137

Figure 5.4: Simplified Arabic font recognition accuracy before and after the post-processing 138

Figure 5.5: Tahoma font recognition accuracy before and after the post-processing 139

Figure 5.6: Thuluth font recognition accuracy before and after the post-processing 140

Figure 5.7: Traditional Arabic font recognition accuracy before and after the post-processing 141

Figure 5.8: Fonts recognition accuracy improvement to top-1 after the post-processing 142

Figure 5.9: Font recognition accuracy worsening to out of the top-10 list after the post-processing 143

Figure 5.10: Sentence length frequency distribution in The Holy Qur'an 146

Figure 6.1: The translation problem due to word segmentation error 159

LIST OF TABLES

Table 1.1: The complete Arabic character set 4

Table 3.1: Data types 52

Table 3.2: Used Arabic fonts 52

Table 3.3: Font Style 53

Table 4.1: Qur'an frequency distribution of the characters and PAWs 77

Table 4.2: Qur'an unique words frequency distribution 78

Table 4.3: Samples of all used fonts typewritten in plain style 14 pt 79

Table 4.4: Word-image samples for the five used fonts 81

Table 4.5: Quranic-252UW database frequency distribution of the characters and PAWs 82

Table 4.6: Quranic-343UW database frequency distribution of the characters and PAWs 84

Table 4.7: Font mask size 92

Table 4.8: Calculated parameters for validating mono-font Quranic-252UW database 96

Table 4.9: Fivefold cross-validation WRR obtained for Andalus font 96

Table 4.10: Fivefold cross-validation WRR obtained for Tahoma font 97

Table 4.11: Fivefold cross-validation WRR obtained for Simplified Arabic font .. 97

Table 4.12: Fivefold cross-validation WRR obtained for Traditional Arabic font . 97

Table 4.13: Fivefold cross-validation WRR obtained for Deco type Thuluth font 98

Table 4.14: Cross-validation average WRR obtained for all font-types 98

Table 4.15: Calculated parameters for Mono-font-252UW 100

Table 4.16: Mono-font-252UW-32CB experiment results (WRRs) 100

Table 4.17: Validation and test set performance (WRR%) 101

Table 4.18: Multi-font-252UW-32CB experiment results (WRRs) 102

Table 4.19: Multi-font-252UW WRRs for Andalus using different codebook 104

Table 4.20: Multi-font-252UW-64CB experiment results (WRRs) 105

Table 4.21: Comparing top-1 WRR% for Multi-font-252UW-32CB and Multi-font-252UW-64CB 106

Table 4.22: Calculated parameters for Mono-font-343UW 107

Table 4.23: Mono-font-343UW-32CB experiment results (WRRs) 107

Table 4.24: Comparing top-1 WRR% for Mono-font-252UW-32CB and Mono-font-343UW-32CB 108

Table 4.25: Multi-font-343UW-32CB experiment results (WRRs) 109

Table 4.26: Multi-font-343UW-64CB experiment results (WRRs) 110

Table 4.27: Comparing WRR% for Multi-font-252UW-64CB and Multi-font-343UW-64CB 110

Table 4.28: Summary of the used databases in the literature review and this thesis 114

Table 4.29: Mono-font comparative summary with the literature review 115

Table 4.30: Multi-font comparative summary with the literature review 116

Table 5.1: Accuracy of the Word Recognition Systems without Post-Processing 135

Table 5.2: Andalus Font Recognition Accuracy with Post-Processing 137

Table 5.3: Simplified Arabic Font Recognition Accuracy with Post-Processing. 138

Table 5.4: Tahoma Font Recognition Accuracy with Post-Processing 139

Table 5.5: Thuluth Font Recognition Accuracy with Post-Processing 140

Table 5.6: Traditional Arabic Font Recognition Accuracy with Post-Processing 141

Table 5.7: Font Recognition Accuracy (from Top-1 to Top-10) Improvement to Top-1 after the Post-Processing 142

Table 5.8: Fonts Recognition Accuracy Worsening to Out of the Top-10 List after the Post-Processing 143

Table 5.9: Top-1 Choice Accuracies Improvement Using N-Gram 145

Table 5.10: n-gram Top-1 Average Accuracies and Complexities for All Fonts Using OCR Top-10 Outputs 145

Table 5.11: Top-1 Accuracy Improvement between the selected n-grams 145

Table 5.12: Accuracy and Complexity Comparison between the selected n-grams 146

Table 5.13: Comparison Summary between Works Reported in Literature Review 147

LIST OF EQUATIONS

Equation 2.1: 2D DCT coefficients 20

Equation 3.1: Straight line equation 57

Equation 3.2: Straight line polar equation 57

Equation 4.1: Word recognition rate (WRR) 94

Equation 5.1: Word-based n-gram model [28] 126

Equation 5.2: maximum likelihood estimate [28] 127

Equation 5.3: The cross-entropy H 128

Equation 5.4: The perplexity PP 128

LIST OF ACRONYMS

2D-DCT	Two-Dimensional Discrete Cosine Transform
ANN	Artificial Neural Network
APTI	Arabic Printed Text Image
BBDCT	Block-Based Discrete Cosine Transform
DAMP	DARPA Arabic Machine Print
DARPA	Defence Advanced Research Projects Agency
DCT	Discrete Cosine Transform
DHMM	Discrete Hidden Markov Model
FL	Fuzzy logic
GT	Ground Truth
HLM	HTK Language Modelling Toolkit
HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit
k-NN	k-Nearest Neighbours
MLP	Multilayer Perceptron
MMAC	Multi-Modal Arabic Corpus
NLP	Natural Language Processing
OCR	Optical Character Recognition
PAW	Piece of Arabic word
RR	Recognition Rate
SAIC	Scientific Application International Company
SLM	Statistical Language Model
SVM	Support Vector Machines
VQ	Vector Quantization
WER	Word Error Rate
WRR	Word Recognition Rate

CHAPTER 1

INTRODUCTION

1.1 Introduction

Transferring written text to a computer-readable form has always been one of the attractive research challenges. The objective of text recognition is to develop a robust and accurate system capable of achieving the level of human performance in reading. Off-line text recognition applications can improve the input operation speed and decrease the possibility of human errors by avoiding retyping a captured document. It takes as input a raster image of a text captured by a scanner or from screen and then transfers it into a machine-editable text. Consequently, several computer subjects are involved in text recognition, including image and signal processing, pattern recognition, natural language processing, and information systems and databases. Although researchers have been intensively investigated in the field, existing systems still did not achieve yet humans reading capabilities. Automatic off-line text recognition is inherently difficult due to the great variability of writing and printing styles. The inherently difficulties are due to letters writing styles, where they presented in different sizes and skews, and variation in width and shape of strokes. The research presented in this

thesis is a contribution toward a reliable recognition system for Arabic text.

1.2 Features of Arabic script

Arabic language is the revealed language of The Holy Qur'an, the holy book of Islam. It is the worship language of Islam used all over the Muslim world, and is the official language in nearly twenty Arabic countries located in the Middle East and North Africa. This makes Arabic script one of the most used around the world and one of the official languages are used at the United Nations. Moreover, there are a number of languages use Arabic alphabet and Arabic-like writing style such as Persian and Urdu, Kurdi and Jawi.

Arabic language has three forms: Classic Arabic – the language of The Holy Qur'an – the dialect of Arabic people in pre-Islamic times, standardized in early Islamic period by adding dots and vowel signs. It is still used as written language in limited formal circumstances. The second form is the Modern Standard Arabic an adapted form of the Classical Arabic. It is the language of today's publications and media. Spoken or Colloquial Arabic is the third form, used (only spoken) in daily interactions and every area of the Arabic countries has its own spoken Arabic dialects.

Arabic writing and English writing are similar in a number of aspects: use letters, spaces, numerals, punctuation marks, special symbols, and written in horizontal lines. However, there are essential differences among these two languages. Unlike English, writing direction of the Arabic scripts is from right to left and its words printed in a semi

cursive style (العربية). Both in printed and handwritten form, Arabic word can be composed of one or more sub-words called PAW as depicted in Figure 1.1. In reality, Arabic is calligraphic and not syllabic writing language. Arabic language has 28 one case main letters there shape is context sensitive, as shown in Table 1.1, hence each letter can have two to four shapes: isolated, initial, medial, or final (ب بـ بـ ـب). Moreover, there are groups of letters has the same main body shape distinguished by diacritical dots (ب ت ث). The character in its different location and PAWs are called a glyph. Another characteristic that is Arabic letters can be vowelized (الْجَدُّ), Figure 1.2 shows more details. These vowels (diacritics) usually omitted from the writing, but they could be present to resolve context ambiguity or in formal documents. Arabic handwritten style and some printing fonts usually comprise vertical arrangements of letters called ligatures. Arabic characters can have special form known as elongation. These letter shapes and vowels, together with some other secondaries (ء), lead to expand the number of Arabic letters over than 120 different shapes. Furthermore, from the syntax, inflection and vocabulary points view the factors such as feminine and masculine, singular, double, plural, and their respective grammatical positions need to be considering.

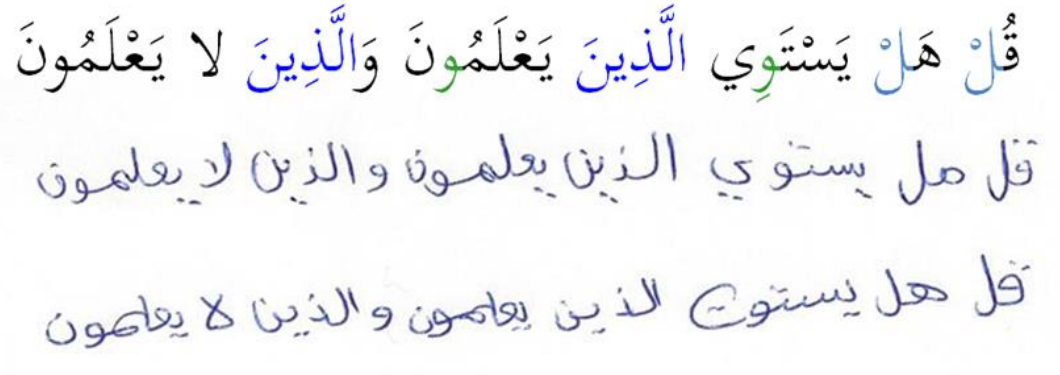


Figure 1.1: Arabic text: Printed and Handwritten sentences

Table 1.1: The complete Arabic character set

Character	Isolated	Initial	Middle	End	Character	Isolated	Initial	Middle	End
Alif	ا	ا	ـا	ـا	Dad	ض	ضـ	ـضـ	ـض
Ba’	ب	بـ	ـبـ	ـب	Tah	ط	طـ	ـطـ	ـط
Ta’	ت	تـ	ـتـ	ـت	Tha	ظ	ظـ	ـظـ	ـظ
Tha’	ث	ثـ	ـثـ	ـث	‘Ayn	ع	عـ	ـعـ	ـع
Jeem	ج	جـ	ـجـ	ـج	Ghyn	غ	غـ	ـغـ	ـغ
Ha’	ح	حـ	ـحـ	ـح	Fa’	ف	فـ	ـفـ	ـف
Kha’	خ	خـ	ـخـ	ـخ	Kaf	ق	قـ	ـقـ	ـق
Dal	د	د	ـد	ـد	Kaaf	ك	كـ	ـكـ	ـك
Thal	ذ	ذ	ـذ	ـذ	Lam	ل	لـ	ـلـ	ـل
Ra’	ر	ر	ـر	ـر	Mem	م	مـ	ـمـ	ـم
Zy	ز	ز	ـز	ـز	Noon	ن	نـ	ـنـ	ـن
Sin	س	سـ	ـسـ	ـس	Ha’	ه	هـ	ـهـ	ـه
Shin	ش	شـ	ـشـ	ـش	Waw	و	و	ـو	ـو
Sad	ص	صـ	ـصـ	ـص	Ya’	ي	يـ	ـيـ	ـي

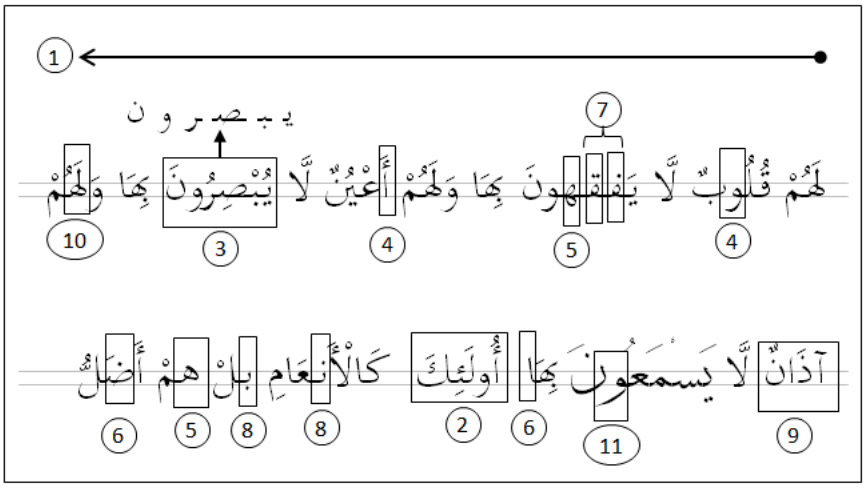


Figure 1.2: The characteristics of Arabic text

1.3 Arabic Language Challenges

The mentioned characteristics of the Arabic language demonstrate the complexity of the language. These complexities include writing style, characters shape similarity, diacritics and inflection poses unique challenges for text recognition and natural language processing. The main challenges to text recognition are the segmentation and classification. The segmentation difficulties are segmenting the semi-cursive script into words and words into letters, letter parts or primitives. In order to correctly segment and extract structural primitives it is vital to detect the imaginary line(s) called baseline(s). Baseline detection considered crucial problem, especially in Arabic writing. The presence of

noise and diacritics greatly increase the difficulties of word segmentation and baseline detection challenges.

Diacritics are a valuable and an important feature in Arabic language. They provide information for the inflectional features for words within the sentence, and resolving word meaning ambiguity by adding semantic information to words. Furthermore, the word syntactic position within the sentence is determined by its diacritics. The morphology richness of Arabic language is a major concern for development of Natural Language Processing (NLP) systems. It is classified as a highly inflectional language where the word stem affixed with prefixes, infixes, and suffixes to indicate case, gender, number, tense, etc.

1.4 Problem Definition

The researches in text recognition systems are mainly based on two approaches, analytical and holistic approach. With a few scattered exceptions, most recognising algorithms use the analytical technique. The first approach segments a word into smaller units such as characters or graphemes (sub-parts of characters). This strategy is adequate for the open vocabulary, as it identifies these smaller units which are concatenated to form word hypotheses which can be considered as valid words or may not belong to the language. Due to the cursive written form of Arabic script and the high variability of characters forms in addition when writing or the image is so poor, the segmentation efficiency hard and error-prone process. The segmentation issue is the major drawback of this approach where segmentation errors mislead classifier during character recognition.

Using holistic approaches, on the other hand, the word is treated as a whole without segmentation and attempts to recognise it using its global shape features. Holistic approaches eliminating the segmentation problem that is primary concerns for analytical approaches, and they may succeed on poorly written or digitised words. Its word recognition result depends on the lexicon words, and the recognition of the unknown word is a word in the lexicon with the highest probability in the recognition result. The price for this method advantages, speed and avoiding problems associated with segmentation, is to constrain the recogniser to limited-lexicon applications. This is because the holistic approaches consider every word as a different class, so when the lexicon size increase, the number of classes increases which in turn increases the probability of misclassification. The direct source of this misclassification is the deficiency in the ability of features set to discriminate between word classes.

These two approaches are reflecting the two different interpretations of psychologists' studies on human being visual word recognition in alphabetic languages. Analytical models support the theory of reading words letter by letter, the opposing view is the holistic model where words are recognised on the basis of their global shapes.

One of the cognitive studies observed that the human tended to read a word as a whole at a time and the contextual information take place to deduce and discriminate a word among other possible words. Holistic theories of reading propose that visual recognition of words affected by their global shape as shown in Figure 1.3 [1] [2] [3], while

the segmentation takes place in case of unfamiliar words [4]. The use of context can be realised in reading a misspelled word correctly in a passage or sentence level. This is clear when reading a text and facing a distorted written word, the human being has ability to delay reading of some words until more contextual knowledge is gathered to define the most probable right word [5].

All of the above are important motivations to build an automatic text recognition systems based on holistic Arabic whole word approach enhanced by contextual knowledge.

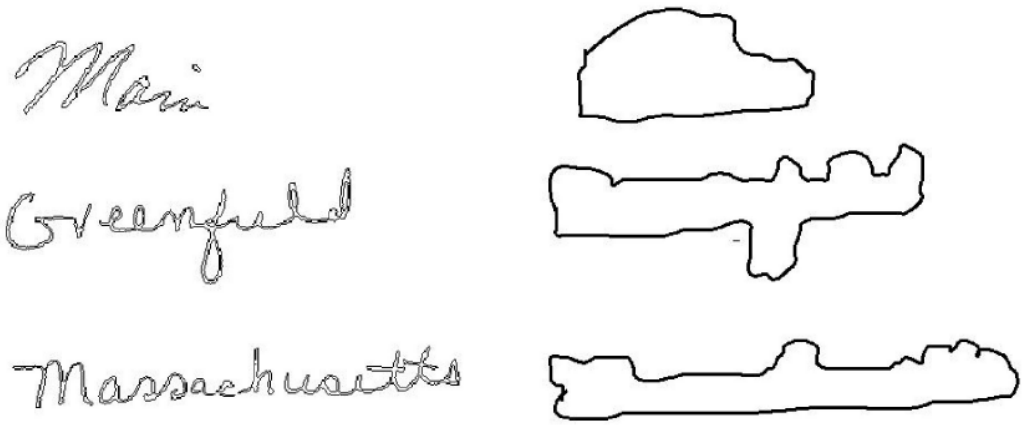


Figure 1.3: The word shape of cursive words alone contains sufficient information to classify the image as one of the lexicon words [1]

The aim of this research is to improve accuracy of holistic approach for offline printed text by design a sentence recognition system to deal with written words. The fundamental idea of using sentence level is the ability to apply contextual information on the sentence (word-hypothesis) to overcome the main weaknesses of the holistic approach;

vocabulary growth problem. The task of the sentence recognition can be decomposed into two subtasks: holistic Arabic whole word recognition and sentence-level post-processing using contextual information. The recognition phase completed using holistic word features, and the outputs are ordered word-hypothesis lattice of top-n matching words in the lexicon using language model.

1.5 Research Objectives

To achieve the aim of this dissertation, as stated in problem definition section 1.4, a word-based or holistic recogniser is needed to be constructed; it has tuneable parameters, its input is word features representing global word shape and its output is a ranked list of word-hypothesis. In many cases, the correct word may not be present in top-1 choices neither in the chosen top-n. To minimise the word error at or near the top-1, the candidates are re-ranked using one of the re-ranking post-processing techniques. Statistical language models (SLMs) are one of these techniques used to re-rank the candidates especially in case of sentence recognition (see Figure 1.4). To train and test the whole system it is required to have word-image database related to an electronic corpus; since the corpus is the fundamental element to create the SLM.

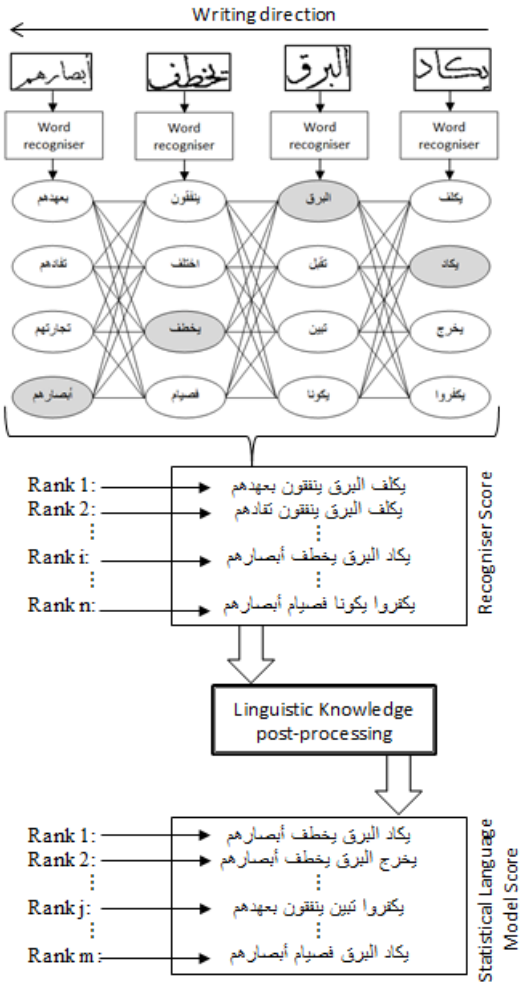


Figure 1.4: The Sentence recognition system

Taking into account main reasons the following objectives are addressed:

- Define an electronic corpus; a closed and an authentic Arabic corpus
- Developing benchmark data for printed Arabic word satisfy the development needs; a full Arabic sentences related to the corpus and printed in fonts cover different printing complexity of character shapes and investigated by other researches
- Designing and implementing a trainable classifier characterised as multiple recognition hypotheses, and language and corpus

independent. A holistic word-based recogniser satisfies the main idea; its input is word feature vector and outputs an N-best word lattice

- The system has to recognise word image printed in font size within the range from 10 to 16 pts that are regularly used in printed Arabic documents such as magazines [6]
- Feature extraction stage plays an important role in constructing a successful recognition system. The target technique intends to be simple; bypasses word segmentation and language independent
- Linguistic post-processing to improve reliability of the holistic word-based recognition system so it can handle larger corpora

1.6 Structure of the Thesis

This dissertation is organised as follows.

- Chapter 2 discusses previous works and researches on the areas of text-image databases, text recognisers, and re-ranking post-processing techniques
- Chapter 3 discusses previous works on databases implementations and demonstrates the methodology for word-image database creation
- In chapter 4 a discrete one dimensional Hidden Markov (Bakis) Model is constructed based on Hidden Markov Model Toolkit[1] (HTK) using features in Block-based DCT domain. For system evaluation (validation/training, test, and stability test), two databases are established. The two databases contain true scanned typewritten Arabic word-images for five different fonts. Validation on the training set is performed using the k-fold cross-validation technique. Result discussion and comparison with other related works are presented

[1] http://htk.eng.cam.ac.uk/

- Integrated Linguistic Post-processing is described in chapter 5, where the linguistic knowledge is applied to improve the accuracy of the sentence recogniser by re-ranking its top-n hypothesis using n-gram SLMs of order 1 to 5. The recognition system is evaluated using synthetic word-image database and The Holy Qur'an is used to enable the use of Statistical Language Models (SLMs) to improve recognition rate. The achieved improvements are discussed and compared with other related works
- The last chapter is 6, it presents the conclusion, limitation, and future work. The infrastructure utilities and units of the integrated system are discussed. Results are reviewed as well as the system limitation.

CHAPTER 2

RELATED CONCEPTS AND LITERATURE REVIEW

2.1 Introduction

Each classical recogniser has two key components, namely, feature extraction and pattern classification. Feature extraction is one of the important processes in the recognition systems; it has a strong influence on the classification stage. The major goal of feature extraction is to get the most significant information to represent the whole data in a compact representational set of features, which maximizes the recognition rate.

In conventional Research, which owns the majority in the literature, the recognition system is trained on features of whole or partial characters, and recognises each character then. Consecutive characters are combined to generate candidate word patterns. This technique is known as Segmentation-based Systems. Indeed, segmentation is one of the most challenges of cursive script, due to its variability. The Arabic text segmentation still remains an unsolved problem though many segmentation algorithms exist. One way of remedying this, is bypassing segmentation and looking at word or text line as major unit for training and recognition. Several researchers have proposed to use segmentation-free approaches where the unit scanned

from right-to-left, and at each horizontal position a set of features was extracted from a narrow vertical strip. These features are fed to a classifier which able to achieve segmentation and character recognition in a continued way, this approach analogues the Automatic Speech Recognition (ASR). Other researchers try to recognise the whole representation of a word instead of segmenting and recognising its characters or primitives individually. These systems are generally known as holistic approach.

The classification stage is the decision-making stage of any recognition system. It uses the features that were extracted in the feature extraction stage to classify the text segment according to predefined rules. The results of the classification stage are related to the information from image data. As the classifier sometimes produces not a single candidate but a set of possible candidate results, the role of the post-processing stage emerge for improving the selection of the right solution.

The text image database plays essential roles for text recogniser development and evaluation. Therefore, the contents must meet the needs of the recogniser nature. Moreover, relating database to a corpus provides the facility for the recogniser to employ linguistic knowledge in post-processing stage. Text databases are classified, based on the recogniser nature, to handwritten text and machine-printed documents; its units can be digits, characters, words or texts. In some cases it serves a specific domain application, such as bank check and postal address reading. In the field of text recognition, having a standard database is

vital for evaluating and comparing results achieved by different groups of researchers. It is worth to mentioning that there is no generally accepted database, for handwritten and machine-printed, that is freely available for researchers and can be considered as a benchmark.

2.2 Printed Arabic Text Databases

A rich research literature exists for Arabic handwriting and only few articles deal with Arabic typewritten. This describes the abundance of handwritten Arabic databases and seldom of machine-printed Arabic databases. A detailed discussion on the handwritten Arabic Text databases can be found in [7].

Datasets can be synthetic or real world data. Synthetic images are somehow similar to what can be seen in real-world documents. The real world data can be simulated by applying modelled degradation to ideal input images. It is a possible solution to generate automatically a quite large database beneficial in terms of evaluation. The main weakness of this approach is that the generated database may not be of the same variety as real-world data. Worse still, it may bias a recogniser towards unnatural image styles.

DARPA Arabic Machine Print (DAMP) document corpus was the most popular large-scale typewritten Arabic databases in the literature. APTI database [8] is the most popular synthetically generated machine-printed database for screen-based OCR. In addition, there are other private databases are developed to meet researchers needs.

2.2.1 DARPA

DARPA Arabic Machine Print (DAMP) corpus was collected by SAIC. It consists of 345 scanned pages (at 600 DPI) of Arabic text (nearly 670k characters) with ground-truth [9]. The scanned pages are collection of book chapters, magazine articles, newspapers and computer printed text in 4 different fonts. Apparently the data is unavailable today [9].

2.2.2 APTI

The APTI database developed by Slimane et al. [8] suited for the evaluation of screen based OCR systems. The database contains more than 45 million of single word images representing more than 250 million characters. Word images are synthetically rendered in low resolution (72 dpi) with a lexicon of 113,284 different Arabic words (648,280 character) presented in 10 fonts, 10 font-sizes and 4 font-styles. Each word image in APTI has its ground truth annotation provided in XML files. The dataset is divided into six equilibrated sets, where frequency distribution of each letter is evenly between the sets. The first five sets are publicly available for research, while the sixth set is kept internal used by the creators of APTI for evaluating the submitted systems in blind mode.

2.2.3 MMAC

MMAC corpus is presented by AbdelRaouf et al. [10]. The data is created from text and images of existing documents. It is consisting of 6 million Arabic words (282,593 unique words) selected from different sources covering old Arabic, religious texts, traditional and modern

language, different specialisations and from online chat rooms. In addition to words the corpus includes PAWs as well as naked words and PAWs. The diacritical markings are not considered in this corpus. The image data are presented in three different formats: single token, paragraph and full page.

Full-page real image dataset, 19 different documents, scanned with resolution 300DPI, 24bit RGB colour mode and tiff file format. Each document has an associated text files.

Paragraph image dataset, 552 paragraphs (around 8,000 words), presented in three different categories: real scanned (223 images), computer-generated (141 images) and computer-generated with artificial noise (188 images). The font type, sizes and styles are not specified. The truth text is provided for all paragraph images.

Single token image dataset, 1,779,582 rendered images, each token (Word, naked Word, PAW or naked PAW) is presented in three Arabic fonts; Simplified Arabic, Arabic Transparent, and Traditional Arabic. The font size is 14 point and the style appears regular. Each token is written to bitmap memory allocation with 300DPI resolution and stored in bmp grey-scale file format. Total number of word images is 847,779 generated for the 282,593 unique words. Each token image has its ground truth XML files. To simulate real-life scanned document the rendered images are degraded by skewing and adding artificial noise.

All databases mentioned above are large-scale benchmarking databases. However, many research groups build small datasets of their

own that fulfil their application needs; like those presented later in Section 2.5.

2.3 Feature Extraction and 2D-DCT

One of the important processes in pattern recognition is feature extraction; which is the process of extracting the most significant information to represent the whole raw-data in small number of new data suitable for the classification phase. It is used for reducing the dimension of the feature set which improves prediction accuracy and minimizes processing complexity. The good selected features are those that are efficiently discriminate between patterns of different classes, but are invariant for pattern within the same class. Feature extraction methods can be classified based on features type into three categories: Structural Features, Statistical Features, and Global Transformation [11].

Structural features are the representation of the global and local properties of patterns by geometrical and topological features. Topological features represent a pattern by extracting and counting number of topological features such as ascending, descending and middle strokes, openings, to the right, left, up and down, loops, cross points, branch points, line ends, etc. [12] [13] [14]. In geometrical features the patterns are represented by the measurement of the geometrical quantities such as word length, aspect ratio, Centre of gravity, mean grey value, stroke width, and more [15] [16] [17].

The statistical feature set is pixel based information derived from statistical distribution of pixels calculated over images or regions of

images. They include zoning, moments, projection histograms, among many others [13] [18] [19].

Global Transformation converts the pixel representation of the pattern to a more compact form which shortens the features dimensionality. In general, transformation schemes can be easily applied and provide feature invariants to global deformations like translation, dilation and rotation. Moreover, global transformation features are language independent features set. Several mathematical transforms have been adapted in text recognition include: Wavelets, Hough transform, Gabor transform, Fast Fourier Transform, DCT, and others [18] [20] [21] [22]. A good survey on feature extraction methods for text recognition is presented in [23], [7].

The DCT is often referred, as is the case in this research, to the 2D-DCT of type DCT-II; which is the most commonly used type. The DCT is widely used transformations in the area of speech and image storage and transmission, as well as feature extraction in pattern recognition. It has the property that most of the signal information is concentrated in just a few low frequency coefficients.

For an M x N image expressed by f(x, y), DCT coefficients are calculated as follows [24]:

Equation 2.1: 2D DCT coefficients

$$F(u,v) = \alpha(u)\alpha(v)\sum_{x=1}^{M}\sum_{y=1}^{N} f(x,y)\cos\left[\frac{(2x+1)u\pi}{2M}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

$$for\ u = 1, \dots, M\ \ and\ v = 1, \dots, N$$

$where\ \alpha(\omega)\ is\ defined\ by$:

$$\alpha(\omega) = \begin{cases} \sqrt{\frac{1}{N}} & for\ \omega = 1 \\ \sqrt{\frac{2}{N}} & otherwise \end{cases}$$

The size and dimension of the resultant DCT coefficient matrix is equal to the input image, this implies that the DCT does not reduce data dimension.

Figure 2.1 shows the properties of the DCT coefficients ordered in zigzag pattern for 8x8 block size. The upper left coefficient is called the DC coefficient with frequency (0, 0), holds most of the image energy, while the rest are AC coefficients with increasing frequency along the zigzag path and reaches its highest at lower right coefficient. The low frequencies are more visually significant in an image than higher frequencies which represent noise. This demonstrate that low frequency coefficients have a great effect for image reconstruction, and the high frequency coefficients can be discarded without sacrificing too much image quality.

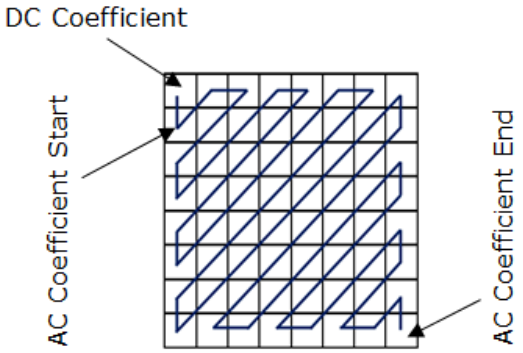


Figure 2.1: DCT coefficients with the zigzag pattern

For feature selection, after discarding some useless coefficients in low and high frequencies the low-dimensional feature vectors usually selected with conventional methods such as zigzag or zonal masking. The two approaches, zigzag and zonal masking, are shown in Figure 2.2 (a) and (b) respectively. DCT implementation can be entire image or Block-based implementation. In block-based operation the image is divided into blocks of m x n pixels size where each block represents a single image. The widely used block size is 8 x 8, the same size used by the JPEG compression standard.

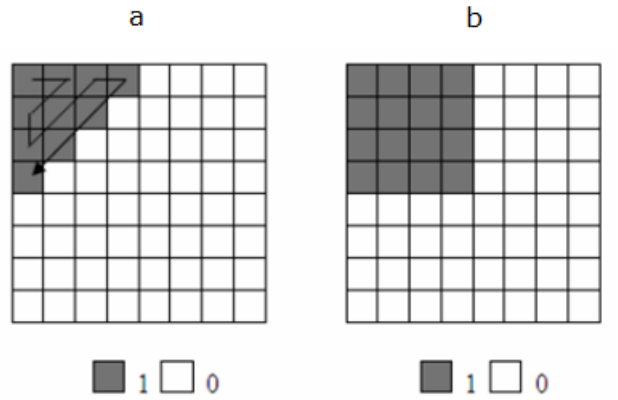


Figure 2.2: DCT coefficients selection (a) zigzag mask (b) zonal mask

It is worth mentioning that DCT has the advantage of fast formula calculations in the DCT and the inverse DCT, since they are based on the real numbers. Because of its strong energy compaction and real number calculation properties; it is successfully used in pattern recognition applications, such as face recognition applications [25], and script recognition [26].

AlKhateeb et al [27] apply the DCT to the entire image to obtain 100 DCT coefficients using the zigzag order. The features matrices are normalized into the range [-1, 1] before fed to ANN classifier and achieved recognition rate of 80.75% on word level. The used word images to experiment are from the well-known IFN/ENIT – database of handwritten Arabic words.

Another system of AlKhateeb, et al., in [18] recognises Arabic handwritten texts using a k-NN approach to classify the words by using Block-based DCT transform (BBDCT) features. An image of a word is segmented into non-overlapping blocks/frames of 20 pixels width, and the DCT coefficients of each frame computed as feature vectors of the word image. Finally, from each block the first five DCT coefficients in the zigzag order are used for classification. This system was tested on the IFN/ENIT – database and achieved a 61.45% recognition rate.

Generally, once the features of a pattern are extracted, the features extraction stage passes the selected features information to the classifier in form of one, or combination, of feature extraction representations.

2.4 Classification and DHMM

The classification stage is the crucial and essential stage in the recognition process. It is the main decision–making part; where it classify unknown sample, based on the extracted features, into one of a finite set of predefined classes. The classification methods can produce a unique solution or a set of approximate solutions as an output. A good OCR system that has the ability to cope with variations in the writing styles while being capable of distinguishing similar yet different characters or words. Several classification techniques and methods in Arabic OCR are reported; which intended to reduce the processing time while improving the recognition rate. Most of those classifying techniques are based on ANN, HMM, k-NN, FL, SVM, Hybrid approaches and others. A comprehensive survey has been presented in [7].

HMMs are doubly stochastic processes which have been widely and successfully used in the recognition of one-dimensional applications especially speech processing, in which the temporal information is available. This Efficiency motivated researchers to employ HMMs in character recognition. Online handwriting recognition; which is a 1-dimension function of time, is an example of the use of HMM approaches. The same approach is applied for 2-dimension text image signal and has shown comparable results to other classifying techniques. Interpreting off-line written text as sequence of ink signals moving on the writing direction, analogues to the temporal sequence in speech can be achieved by applying a fast segmentation process using a sliding window technique. In addition, HMMs classifiers have been successfully applied in

off-line text recognition without consideration to the temporal information, where word image either considered as a whole or segmented into its primitives.

HMM is a finite state machine which consist a set of hidden states connected to each other by conditional transition probabilities (see Figure 2.3). Each hidden state has an associated set of probabilities of emitting particular visible states according to some output probability density function (pdf). With no constraints have been placed on the transition matrix; the unit model topology can be defined with arbitrary number of model states and transitions between these states. Several types of model topologies have been proposed in the literature, ranging from left/right models to fully connected ergodic models. Figure 2.4 is depicting Left-to-Right or Bakis model and Figure 2.5 ergodic model.

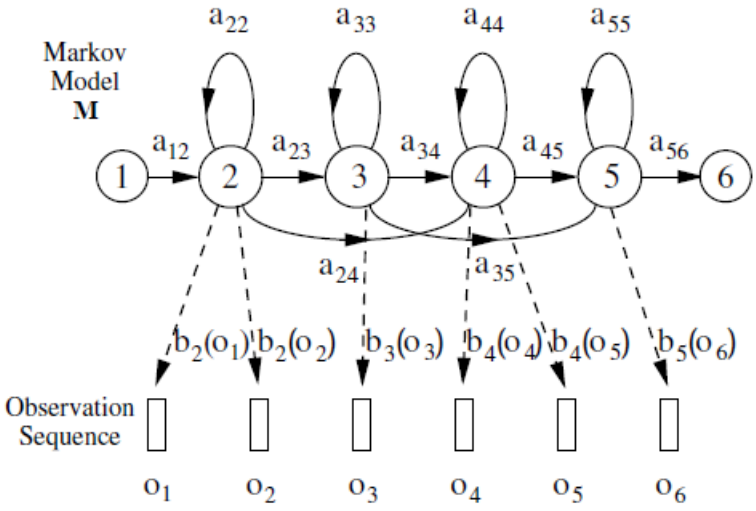


Figure 2.3: The Markov Generation Model [28]

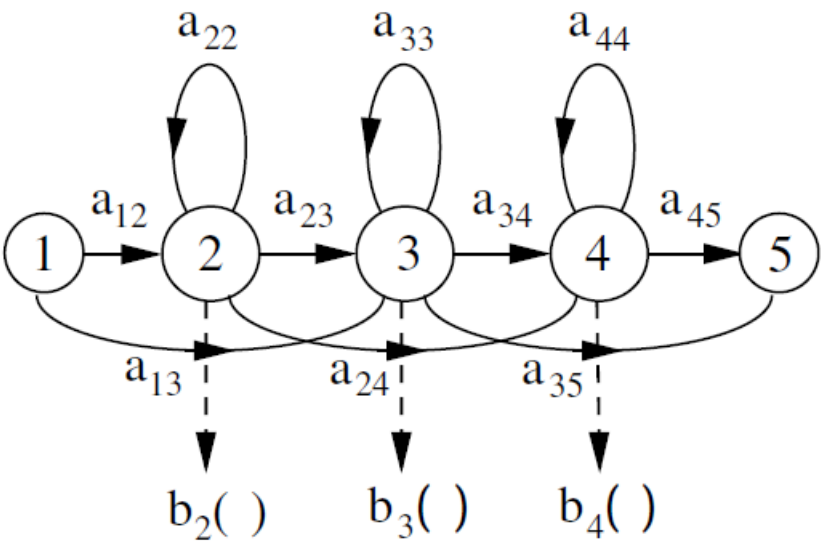


Figure 2.4: Simple Left-Right HMM [28]

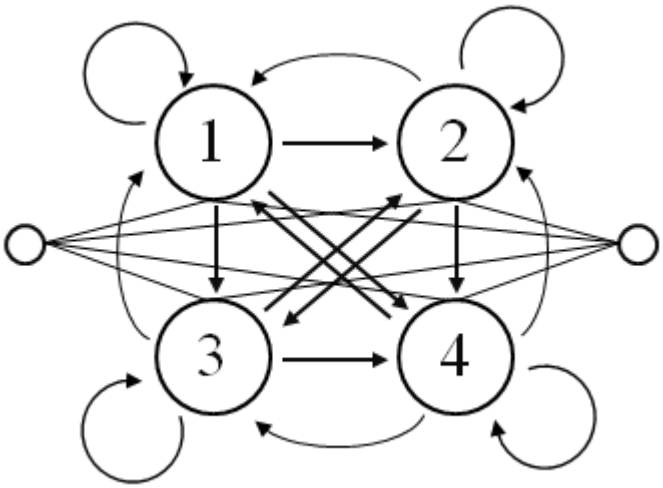


Figure 2.5: A 4-state ergodic model with start and end states

HMMs are mainly classified into three categories depending on the method of modelling output probabilities [28]: discrete HMMs [29], continuous HMMs [30], and semi-continuous HMMs [31]. However, discrete HMMs are more attractive because of its low computation cost [28]. Another advantage of using the DHMM is that it is being able to model any distribution especially in noisy signal (speech) when assumption of normal distribution is not satisfied [32]. Accordingly, The

DHMMs can represent more complicated shapes [29]. DHMMs are mainly suitable for modelling data which is naturally symbolic; such as letters and words, bitmap images, and DNA sequences. In general, using DHMMs demand that the feature vectors need to be encoded into discrete symbols using VQ technique. The DHMMs can also be used with continuous signals such as speech by mapping the continuous feature vectors to discrete symbols using VQ.

As can be seen from the Figure 2.3 above, the DHMM is defined by the following entities:

- $S = \{s_1,\ s_2, \dots, s_N\}$ a finite set of hidden states, N being the number of states
- $O = \{o_1,\ o_2, \dots, o_M\}$ a finite set of observations, M being the size of the symbol alphabet
- $A = \{a_{ij}\}$ a transition matrix, where a_{ij} is the transition probability from state s_i to state s_j, with $a_{ij} \geq 0$ and $\sum_{j=1}^{N} a_{ij} = 1$
- $B = \{b_j(o_k)\}$ an observation (output) matrix, where $b_j(o_k)$ is the probability of generating observation (symbol) o_k given the state s_j
- $\Pi = \{\pi_i\}$ an initial state distribution vector, representing probabilities of initial states, with $\pi_i \geq 0$ and $\sum_{j=1}^{N} \pi_i = 1$

Thus it can be seen from above that a complete specification of a DHMM can be described compactly by a set of parameters as,

$$\lambda = (A,\ B,\ \Pi).$$

2.5 HMM-based Printed Text Recogniser

A recently survey covering researches in handwriting Arabic text recognition is presented in [7], and [33] for machine-print recognition until 2006. Recently, most researchers have focused on handwritten rather than typewritten recognition. There are numerous publications on Arabic OCR. In this study, the most relevant works are exploited; that are use HMM classifier applied on the same computer-printed Arabic fonts.

Khorsheed [34] presented a font-independent recogniser. The system is designed using multiple left-to-right HMMs where each word represented by a separate model. Features are represented by segmental Vector Quantisation extracted from half Fourier spectrum coefficients of the normalized polar word-image. The system was designed using left to-right 6-state HMM in a serial model where only one transition from a state to its successor is allowed. The obtained features are invariant to the Poincaré group of transformations: dilation, translations and rotations. The used lexicon size is 145 words, more than 1,700 samples used to evaluate the word recogniser performance. Two different cases are applied to assess the performance of the system. The first case, samples were rendered in four different fonts: Andalus, Simplified Arabic, Thuluth, and Traditional Arabic. Font sizes ranging between 18 and 48 pt., angles range from 0 to 2π, and translation up to twice the sampled word size. The font sizes, angle and translation values generated randomly. The second case, two fonts (Thuluth and Traditional Arabic) replaced with two handwritten scripts. The experiments were performed

using 44% of the data set to train the word models, while the rest was used for assessing the recogniser. Each word model was trained using randomly selected observation sequences from the four fonts. The recognition rates of the first case is not reported. The highest word recognition rate (WRR) achieved in the second case for Simplified Arabic and Arabic Traditional are: 90% as the first choice and 98% within the top-ten choices.

In later work, Khorsheed reports a HMM-based mono-font off-line recognition system for the computer-generated cursive Arabic text line [35]. The HMM classifier is built on the HTK. A segmentation-free sliding window technique is used on text line. Features extracted from the text are statistical representing the intensity, intensity of horizontal derivative and intensity of vertical derivative. The system is open vocabulary dependent on character models and grammars. Two recognisers are designed in two different model schemes: mono-models and tri-models. In mono-models different shapes of a single letter are represented by a distinct HMM model in total number of 60 models. In tri-model there are 9,393 models each HMM model representing a combination of three letters and it is context-dependent. In all cases each HMM module, regardless of font type, has eight states per model and codebook includes 128 clusters. The states' transitions of the HMM module are not stated. The recogniser performance was evaluated using Arabic text database typewritten (not specified font sizes) in six different fonts: Andalus, Naskh, Simplified Arabic, Tahoma, Thuluth, and Traditional Arabic. The database containing more than 15,000 line images. The performance was assessed for each font with 1,500 line images for

training and 1,000 line images for testing. The line image height is normalized to 60 pixels to eliminate font type and size dependency. The experiments conducted separately for each font. Three different type of features are used: intensity, intensity of horizontal derivative and intensity of vertical derivative. The highest average recognition rates (RRs) achieved was measured for cell size (window width) 3 x 3 with 1 pixel vertical overlapping windows. The highest achieved results were 88.7% and 92.4% for Andalus font in mono-model and tri-model respectively. Another set of experiments were carried out on a multi-font where training data are selected randomly with same sample size from all fonts. Using fixed data set for testing, 200 lines from each font. The reported results did not discuss font results; it simply mentioned that the system shows a flat performance of ≈95% when using the tri-model.

In Al-Muhtaseb et al. [36] a system is implemented with similar techniques as in [35]; built on HTK Toolkit, segmentation-free, sliding window feature extraction, text line unit and mono-font. The text line images all normalized to height of 80 pixels. From each vertical strip, the 16 features were extracted that are representing one type of features (sum of black pixels). The HMM module topology is the same for all fonts where the transition is allowed to the current, the next, and the following states only. However the number of states and codebook size are font dependent; the HMM model can have five or seven states left-to-right HMM and codebook sizes can be 128 or 256. Each character shape is considered as a separate class in total of 126 classes for each font. To evaluate the recogniser a text line database was implemented in eight fonts (Akhbar, Andalus, Arial, Naskh, Simplified Arabic, Tahoma, Thuluth,

and Traditional Arabic), each font consists of 2766 lines of text. It appeared that synthetic images have been used in this database. In training phase the first 2,500 lines were used and for testing phase the remaining 266 lines were used. To assure presence of a sufficient number for all shapes of Arabic letters, the training sets are augmented with 5 copies of the 3 lines of the minimal Arabic script database [37]. The classifier having a unique code for each shape of each character achieves highest accuracy percentage is for Arial font with 99.85%, and Andalus register the lowest accuracy 96.83%. An improvement in accuracy in all fonts when combining, after recognition, different character shapes into one code; the highest is Arial (99.90%) and the lowest is Andalus (97.86%).

Slimane et al. [38] also applied an HMM recogniser with sliding window to the recognition of Arabic printed word image. This work focuses on screen-based OCR. The recogniser is constructed using HTK Toolkit and classified as a word unit, segmentation-free, multi-font and open vocabulary. Two proposed techniques are used: global multi-font system (font independent) and cascading system (font recogniser followed by mono-font word recogniser). The word images are normalised in grey-level with 45 pixels height and then with 8 pixels window width the feature vectors of 102 components are computed. Concerning the HMM topology, 64 models having an equal length of five states, not defined transitions, representing a similar character shapes groups. These similar character shapes are grouped according to predefined rules. At recognition time, all sub-models are used to compose an ergodic HMM to allow recognising potentially any word in an

open vocabulary manner. To evaluate the system some parts of the APTI database were used [8]. The ATPI database is word images synthetically generated in low-resolution "72 dot/inch." It is freely available large-scale benchmark database suited for the evaluation of screen-based OCR systems. Word images are generated using 10 different fonts: AdvertisingBold, Andalus, Arabic Transparent, DecoType Naskh, DecoType Thuluth, Diwani Letter, M Unicode Sara, Simplified Arabic, Tahoma, and Traditional Arabic. The 10 different fonts are select to be in size 24pt. Set-1 (18,897 words for each font) is used for training and the unseen set of the word in set-5 (18,868 words for each font) are for testing. The highest calculated performance of character recognition rate is 99.20% for Andalus font using mono-font recogniser, while the Global multi-font attains 98.60% for Arabic Transparent font. The word recognition rate also registered with highest calculated performance is 94.20% for Andalus font using cascading system mono-font recogniser, and 87.20% for Arabic Transparent font using Global multi-font.

2.6 The Re-Ranking Post-Processing Techniques

The text recogniser has text image and a lexicon as inputs, and its output is word-hypothesis best match the word image within the lexicon. In some cases, the outputs are ordered word-hypothesis lattice of best-n matching words in the lexicon. Among these hypotheses the correct word-hypothesis is not always placed at the first position nor at any selected first n-position in the ranked list. The recogniser can employ the post-processing stage to improve the accuracy; this can be achieved by

incorporating the context and shape information at the post-processing stage.

Using linguistic knowledge such as co-occurrence pattern of adjacent words and syntax and semantics is still in its early stages in Arabic text recognition and much more research efforts are needed. The successful applications of these linguistic models supporting the recognition process, especially for English text, recommend their effectiveness. In the case of Arabic OCR only a few publications investigate recognition rate improvement in the post-processing stage. Most of these approaches are based on combining different systems or classifiers. To the author's best knowledge only a small number of researches investigate the use of contextual information to improve re-ranking the list of candidates.

AlKhateeb et al. [39] use structure-like features (number of connected regions: sub-words and diacritical marks) to re-rank the top-10 result produced by a handwriting recognition system. They implement a Hidden Markov Model (HMM) classifier using the HTK Toolkit evaluated by the IFN/ENIT database of handwritten Arabic words (Tunisian town/village names). The top-10 candidates are re-ranked according to the values of their refined probability. The refined probability is a function of associated probability (obtained from the recogniser) and Gaussian-like function which in turn it is a function of structure-like features. In terms of top-1 recognition rate, the highest improved accuracy is 2.51% using database version v1.0p2 with set d (6735

words) for testing, where the recognition rate is enhanced from 86.73% to 89.24%.

Prasad et al. [40] applied n-gram language models (LMs) to re-score an n-best list. The kernel of the system is glyph HMM classifier. The data set is from the DARPA Arabic Machine Print (DAMP) document corpus collected by SAIC. It is collected from books, magazines, newspapers, etc. The corpus size is 297 scanned images of text used for developing, training and testing the OCR system; 60 for development, 60 for testing, and 177 for training purpose in addition to another 380 synthetically generated images of newswire text. Three types of LM are used in improving the recognition rate: character, word and PAW. The LMs are created from 2.6 million Arabic words of newswire data in addition to transcriptions from the training set images. The lexical sizes for character, word, and PAW n-grams are: 162 characters, 65K words, and 9K PAW respectively. The achieved performance by rescoring the N-best list (N not specified) using LMs is represented in terms of word error rate (WER). The maximum reduction in WER is 10.1% achieved by using the PAW trigram compared to 11.5% obtained by character 5-gram and 15.9 % for word trigram. The system improvement using the LMs is not specified.

Devlin et al. [41] proposed an alternative approach, where statistical machine translation (SMT) and n-gram LM are used to re-rank OCR generated sentences. The HMM based OCR system is Arabic handwritten character recogniser. The baseline system is trained on 2.3 million words of transcribed text field collected such as legal filings. The

testing sentences are 1,397 (22,261 words). The Arabic-to-English SMT system (hierarchical system) is trained on 45 million words (news data) from Arabic-to-English parallel corpus at Linguistic Data Consortium (LDC). The SMT uses English n-gram LM (n not specified) to re-score its outputs. This English n-gram is trained on 5 billion words from the LCD English GigaWord corpus. It should be noted that there is a significant mismatch between domains used to train the SMT and the OCR systems. The Arabic 5-gram LM is trained on three Arabic corpora: GigaWord (500 million words), in-domain baseline training text (2.3 million words), and Arabic sets used in training SMT (45 million words). The task of the SMT and the LMs is to minimize the WER by re-ranking the 20-best ranked list produced by the baseline system. The baseline system WER (24.88%) is improved on top to 24.21% and to 24.25% by using the Arabic 5-gram and SMT-LM respectively. The highest WER improvement is 23.81% resulted when the SMT-LM is used in combination with the Arabic 5-gram LM.

Al-Hajj et al. [42] introduced a re-ranking scheme via fusion of three homogeneous HMM-based classifiers. The classifiers have the same topology and differ only in the sliding windows orientation angle ($-\alpha$, 0°, and $+\alpha$). The classifiers outputs, list of word candidates with their scores, are fused at the decision level. Three decision makers are employed: the sum rule, the majority vote rule, and a combining classifier based on neural network namely Multilayer Perceptron (MLP). The MLP is trained to select only one HMM classifier and its Top-1 candidate. The three combination strategies are considering the Top-10 candidate words of each recogniser to produce a rescored word list. Experiments have been

carried out on the IFN/ENIT database, sets (a, b, and c) for training and set (d) for testing. The reference classifier (0°) accuracy is higher than classifiers using oriented frames (α = ±20°). Generally, the proposed combination schemes improved the accuracy. The highest improvements achieved at Top-1 choice are by using the MLP, where the reference classifier Top-1 recognition rate increases from 87.60% to 90.96%. The sum rule outperformed the other two schemes at Top-2 and Top-3 by 94.89% and 95.87% correspondingly.

Farah et al. [43] applied a syntactical analyser at post-classification stage to select the better suited word from the set of candidates. The suited word is the word which produces syntactically correct sentences. The syntactic analyser uses the grammar describing the Arabic legal amounts. The proposed system is to recognise handwritten Arabic legal amounts in cheques. It is composed of three classifiers, namely MLP neural network, k nearest neighbour (k-NN) and Fuzzy k-NN, combined in a parallel scheme at the combination stage. They take word holistic structural features as their input and produce top-3 words ranked list as their output. The experiments were carried out on database for Arabic literal amounts built by the authors (4,800 handwritten words represents 48 words of the lexicon written by 100 different writers). The three classifiers were trained using 1,200 words and tested by 3,600 words. The testing set (48 words x 75 writers) written by writers not seen in training phases. The classifiers results are combined to produce a list of up to nine ranked candidate words. The combined schemes are statistical decision systems, including maximum, minimum, average, product, and sum-rule. The Fuzzy k-NN classifier

achieved the highest recognition results 92.16%. The combination technique improves the recognition rate. The highest recognition rate improvement is 94% achieved by the sum-rule scheme. Recognition accuracy is further improved to 96% when the syntactic information integrated at the end of the post-processing stage.

2.7 Summary

A typical pattern recognition system takes the normalised input data, extracts the features, trains the classifier, evaluates the test pattern, and finally improves recognition accuracy based on linguistic or statistical data.

The background investigation illustrated many important facts about the status of research in Arabic text recognition. Arabic printed text has more or less been abandoned in favour of handwritten text. The analytical approach, especially sliding windows, is widely used for word recognition in favour of holistic one. The database is an actual part of any recogniser satisfies its requirements. However, there is no benchmark databases of text to test the performance of any system developed for Arabic text recognition. The effectiveness of exploiting contextual information is to improve re-ranking the list of candidates. Further research can be directed toward integrating context-dependent word correction for Arabic, as these studies are still at an early stage of development.

This study is motivated by these facts and aim to design recogniser employ holistic approach for word recognition and consider

passage-level post-processing for context-dependent word correction (re-rank the candidate lattice). Unfortunately, none of the available databases satisfy the requirements of this project. This raises the need to build database representative to the recognition task, it is composed by word samples derived from corpus.

CHAPTER 3

ARABIC WORDS DATABASE PREPARATION

3.1 Introduction

The database provides a very important infrastructure towards development and comparison of different recognition systems for hand or machine printed scripts. Accessing to a word-images database and statistical language models (SLMs) is essential during system development. Hence, the database should meet the needs of the proposed method in this study.

The goal of this chapter is to present a methodology for building a database of written words/sentences for the research in Arabic text recognition, in general and machine printed (multi-font) Arabic script recognition, in particular. Word segmentation is a challenge in Arabic text because of the semi-cursive nature of Arabic script (PAWs) and word-shape length is font dependent. To provide efficiency, accuracy, timelessness, security and economy; a fully computerized system has been developed for automatic: form creating, word segmentation, and ground truth generation.

3.2 Considerations in Building OCR Text Database

Different databases for different OCR systems exist. Each one is related to some application, e.g. handwritten numeral and literal amounts for bank checks application, handwritten town names for postal address reading and sorting system, and low resolution screen rendered text for reading text from screen images. The nature of the classifier creates conditions that influence the solution of the database implementation problem. Accordingly, there are number of points to be considered to design an OCR system, which in turn influence database implementation. The most important points are:

(1) High level knowledge integrity: integration and contribution of linguistic knowledge for improving the recognition throughput and accuracy. This require a corpus as the foundation of the database rather than collecting text from ”random” sources

(2) Major unit specification: character, word, sentence, script, or text line

(3) Style variation: different fonts might be used (machine printed and synthetic text), or many individual writing styles can occur (handwritten text).

(4) Font size and style: font point size range and style (normal, bold, italic, bold italic)

(5) Writing style: handwritten, machine printed or synthetic text

(6) Printing quality (pixels/inch)

(7) Scanning intensity: Binary-scale, grey-scale, Colour (RGB space)

(8) Image Resolution : DPI (Dots Per Inch)

(9) Noise handling ability: the scanned image can has noise due to various reasons

In this research context, some constraints should be satisfied in order to build word-images database:

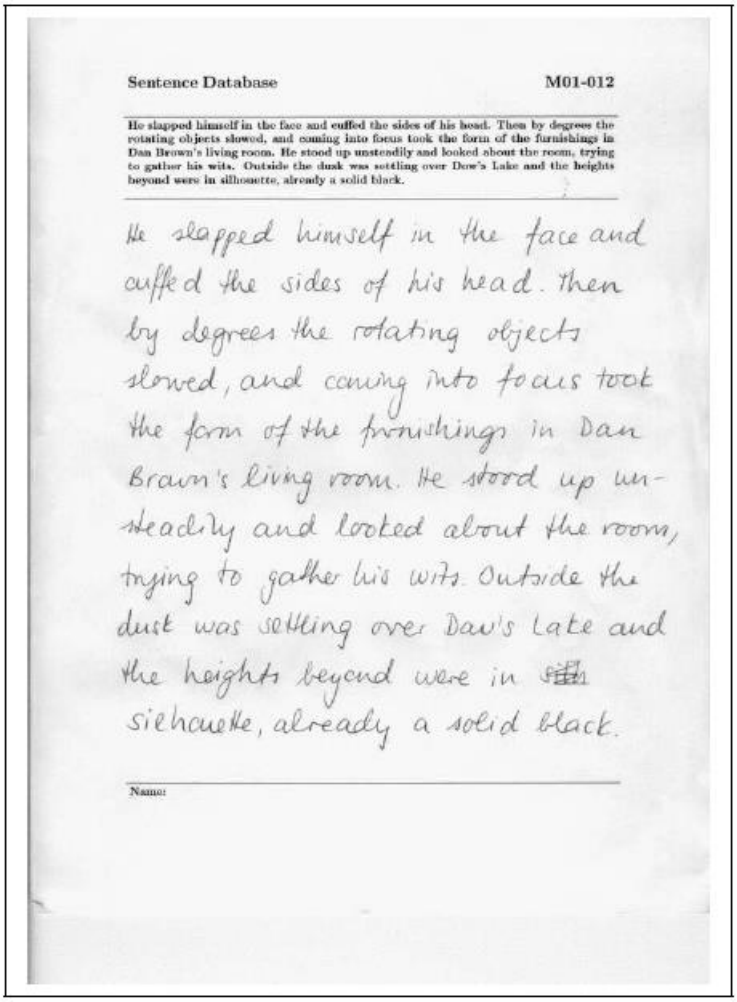
- Authentic text corpus comprises data set captures the semantic and syntactic of the script
- Two types of databases related to the corpus with enough samples for training and testing recogniser uses holistic word features:
    - Real scanned word-images to evaluate baseline recognition system
    - Rendered word-images representing full-sentences to assess the integrated system (baseline system integrated with linguistic post-processing)
- Samples reflecting off-line real-life Arabic text typewriting variability; printed in five font types, 14 points in plain style
- Samples are stored in binary-scale using TIF format with resolution of 300 dpi

3.3 Related Works

Different databases for different OCR system exist in different languages; they contribute in their related researches. This section presents an overview methodology of database implementation techniques for some scripts which includes English, Chinese, Farsi, and Arabic.

Marti and Bunke [44] designed the IAM database; a full English handwritten sentences from the Lancaster-Oslo/Bergen (LOB) corpus distributed over 4,881 lines of text. The collection data form, as depicted

in Figure 3.1, is automatically generated. The form layout has four parts separated by horizontal lines. The first part contains database title and text code identifying its category and number in that category and its starting sentence. The second part has a machine-printed text which the writer should copy in the following section. The last section is for writer where he can voluntarily print his name. The horizontal lines are used for document segmentation and skew correction operations. To ease the image pre-processing, the writer asked to write on a guide lines printed on another sheet of paper placed under the form. The form scanned at a grey level of 8-bit with resolution of 300 dpi and saved in TIFF-format. The aim of this database is to be applied in segmentation-free recognition techniques and in systems that use linguistic knowledge. For this reason, the ground truth file contains the labels of the printed and handwritten line of text in ASCII-format.



Sentence Database M01-012

He slapped himself in the face and cuffed the sides of his head. Then by degrees the rotating objects slowed, and coming into focus took the form of the furnishings in Dan Brown's living room. He stood up unsteadily and looked about the room, trying to gather his wits. Outside the dusk was settling over Dow's Lake and the heights beyond were in silhouette, already a solid black.

He slapped himself in the face and cuffed the sides of his head. Then by degrees the rotating objects slowed, and coming into focus took the form of the furnishings in Dan Braun's living room. He stood up unsteadily and looked about the room, trying to gather his wits. Outside the dusk was settling over Dan's Lake and the heights beyond were in silhouette, already a solid black.

Name:

Figure 3.1: The IAM database filled form [44]

The HIT-MW sentence database for Chinese language is implemented by Su et al. [45]. It is built to fit segmentation-free recognition systems and techniques that apply statistical language model. It is based on the China Daily corpus. As can be seen from Figure 3.2, the collection form layout is divided into three sections: writing guidelines, typewritten text, and handwriting. The upper two sections are separated by horizontal line, while the third section is bounded by rectangular box. Horizontal lines can help to remedies the document and writing skew since there is no writing rulers. Each form has 4-pair-digit code identifier; the first two-par shows the publishing year and month; the other two-pair are stands for

text location in the corpus. The forms are digitized in 300 dpi resolution saved as grey-scale in BMP-format image file. The ground truth file format is not specified. In addition to the handwriting text line, the ground truth file involves the text lines from the text section that corresponding with the handwriting text line.

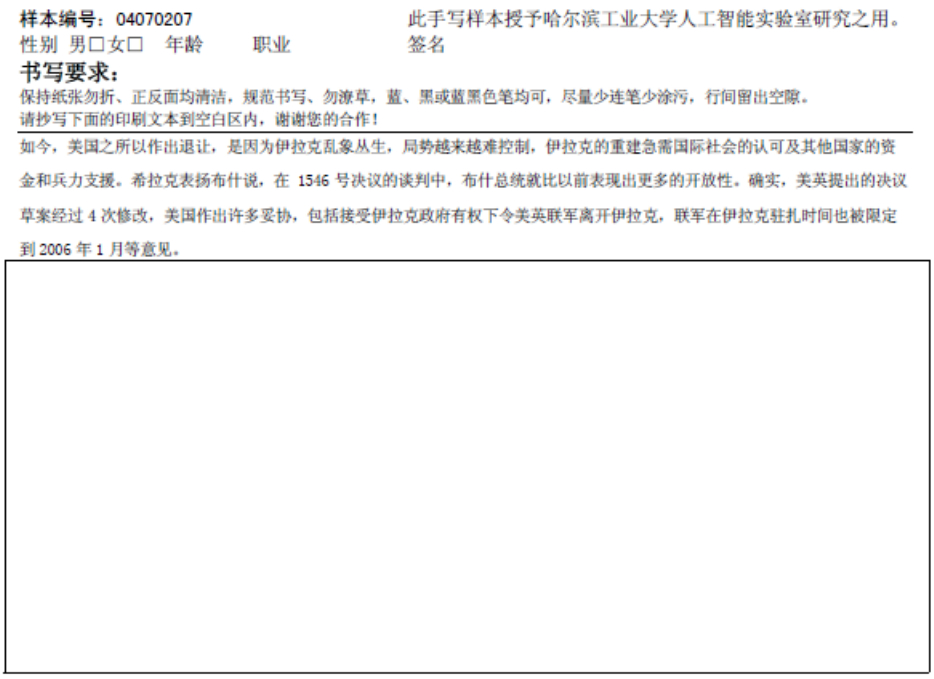

样本编号：04070207　　此手写样本授予哈尔滨工业大学人工智能实验室研究之用。
性别 男□女□　年龄　　职业　　签名
书写要求：
保持纸张勿折、正反面均清洁，规范书写、勿潦草，蓝、黑或蓝黑色笔均可，尽量少连笔少涂污，行间留出空隙。
请抄写下面的印刷文本到空白区内，谢谢您的合作！
如今，美国之所以作出退让，是因为伊拉克乱象丛生，局势越来越难控制，伊拉克的重建急需国际社会的认可及其他国家的资金和兵力支援。希拉克表扬布什说，在 1546 号决议的谈判中，布什总统就比以前表现出更多的开放性。确实，美英提出的决议草案经过 4 次修改，美国作出许多妥协，包括接受伊拉克政府有权下令美英联军离开伊拉克，联军在伊拉克驻扎时间也被限定到 2006 年 1 月等意见。

Figure 3.2: The HIT-MW database collection form layout [45]

In addition to these databases there are many other databases in the handwritten recognition domain inspired by the former database. These inspirations were in form design and unit sample acquiring procedure.

The FHT Farsi handwritten scripts database was implemented by Ziaratban et al. [46]. Texts sampled from corpus in different categories similar to the IAM database. The layout of form and its details are illustrated in Figure 3.3. All filled form pages were scanned with 300 dpi in 256 grey scales. The GT files are for the entire handwritten texts.

Mahmoud et al. [47] developed AHTD contains text selected from an Arabic corpus in different topics. The forms are scanned in grey scale at different resolution 200 dpi, 300 dpi, and 600 dpi. The page, the paragraph and the line levels have ground truth files in database format. Another Arabic Text Images Database (AHTID/MW) written by Multiple Writers has been developed by Mezghani et al. [48]. It comprises written Arabic words and text-lines by different writers. Each text line image and word image has XML ground truth file describing its contained information such as sequence PAWs and characters.

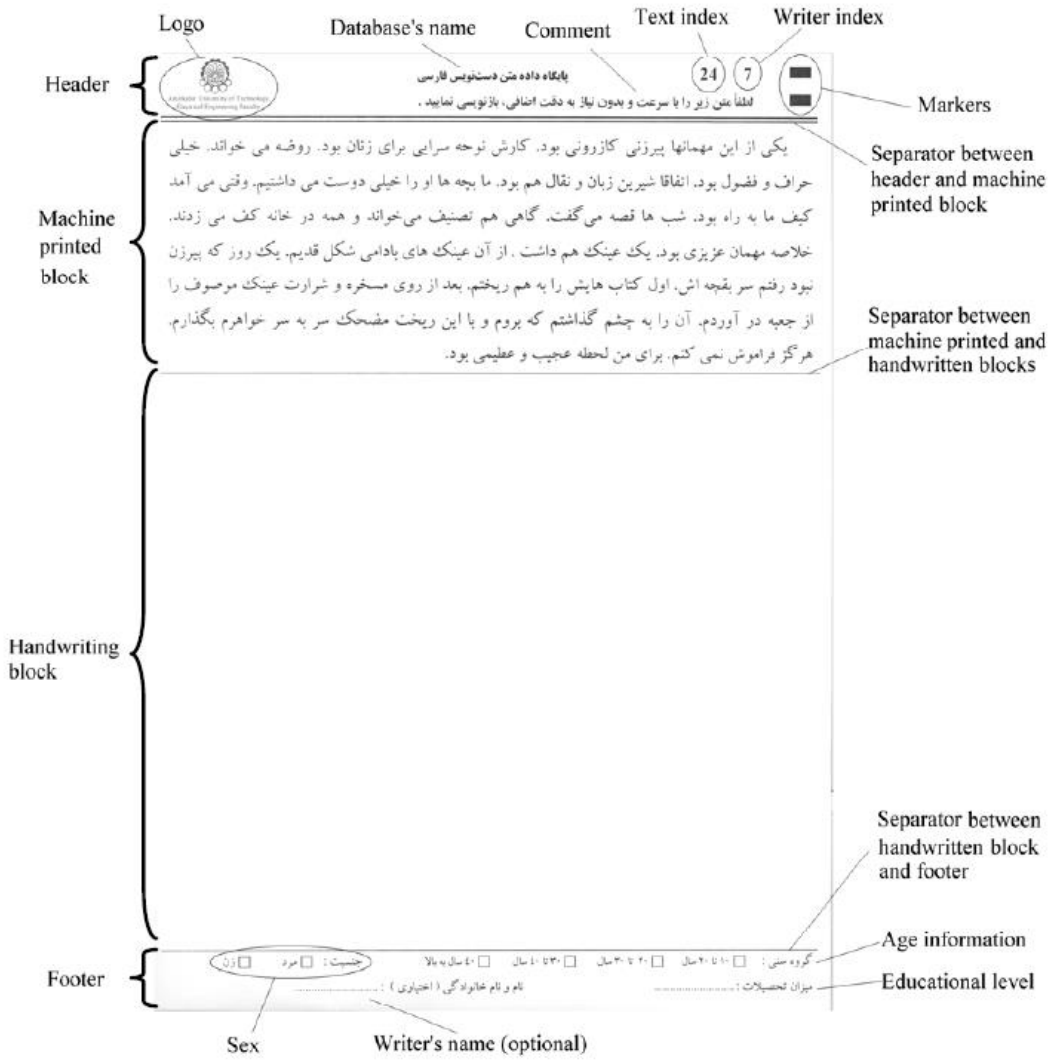


Figure 3.3: The FHT database form layout details [46]

A brief summary about related databases for different languages in time period 1993 – 2008 can be found in [46] and Pravez and Mahmoud [7] review the available Arabic text databases.

3.4 Corpus Acquisition

The utilization of contextual knowledge is one strategy to successful text recognition. Since this work is directed toward this goal, it is important to have an OCR oriented database that contains full image of

Arabic sentences in addition to SLMs. The first phase of database and SLM development is defining an electronically available authentic text corpus. After a comprehensive investigation, the text corpus is collected from The Holy Qur'an, which is available at the Tanzil Quran project[2] in the public domain. It provides highly verified UTF-8 format Quran text written in Classical Arabic. It allows downloading the Quranic text in various formats including Simple and Uthmani Scripts with different number of diacritics and symbols.

It should be noted that the linguists and grammarians use The Holy Qur'an as a reference because it contains pure Arabic vocabulary as well as Arabic grammatical rules [49] [50]. The Holy Qur'an, therefore, forms a robust NLP basis for researchers in Arabic language in general and complementing OCR in specific.

Given the importance of the Qur'an, the "Quranic Arabic Corpus[3]", an online linguistic resource organized by the University of Leeds, provide grammar, syntax and morphology for each word in the Holy Quran [51]. This corpus project is part of the Arabic language computing research group within the School of Computing at the University of Leeds. In terms of data, an authentic copy of the Arabic Qur'an text from Tanzil project is used. Figure 3.4 below shows a dependency graph that describes the syntax of verse (67:1). These set of the provided analysis can be used to facilitate and/or improve the OCR process.

[2] http://tanzil.net/download/
[3] http://corpus.quran.com/

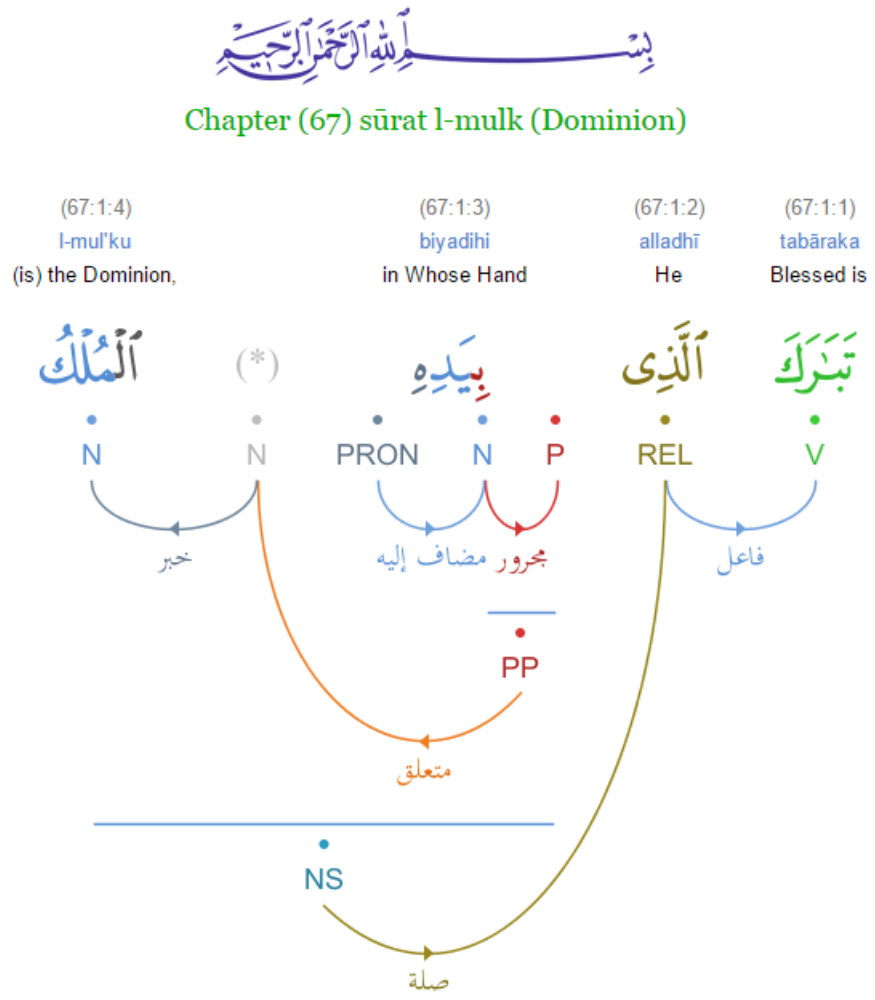


Figure 3.4: Dependency graph for verse (67:1) [51]

3.5 The Holy Qur'an Arabic

The Holy Qur’an is the religious book of Islam written in Quranic Arabic dating from seventh century, which forms a unique genre and represents the purist and the most authentic form of the classical Arabic language. The used version of The Holy Qur'an is according to Hafs' transmitted version (riwayat). The Holy Qur'an organised in a hierarchical model as follows: Document, Chapter/Surah, Verse/Ayah, and Word/Kalimah. In addition to this hierarchy, there are pause symbols indicating the end of complete meaningful sentences. Considering the

starting Basmalah of 112-Surah as verses and the functional words as well as letters of introductory verses for some chapters (الم، عسق) as words; The Holy Qur'an comprises of: 114-Surah, 6,348-Ayah, and 78,245-Kalimah, each Ayah can have from 1 to 129 words. It is considered as a closed corpus including 14,870 unique words.

3.6 Database Implementation Methodology

This section presents the implementation details of a database of isolated machine-written Arabic words for text recognition researches. Work requirements demand creating a word-images database related to the task. These requirements are: isolated words printed in different fonts and samples that meet the constraints stated above (section 3.2). Moreover, the needed attributes of the database are the followings:

- Sample names are coded according to pre-defined rules representing its attributes (character set group, font attribute, position in the corpus, etc.)
- word images are stored together with their ground-truth information
- Ability to add new fonts and font attributes according to the pre-defined rules

Once the collection form has been designed, which will be described in the next subsection, database implementation approach is summarized in the following steps:

- Define the desired set of words from the corpus and define frequency distribution of each

- Automatically generate Microsoft Word documents containing filled forms; each document contains one form filled by words typed with the same font type, size, and style. Each document provided with two reference files in text format for segmentation process, one for page-groups and the other for words in each page.
- Printout Word documents in predefined printing quality (pixels/inch)
- Scan the printed forms at desired dpi resolution
- Save scanned images in binary-scale using TIF format. Every document image is saved in file its name structure according to a predefined format
- Edges distortion eliminating in document image
- Skew detecting and correcting for document image using Hough transform technique
- Document segmentation; exploit vertices pixel coordinates of text section and lines, and words
- word image tidy segmentation, whole word body shape, by applying smearing strategy using Run-Length Smoothing Algorithm
- Word binary images are saved in its corresponding subset folder with resolution of 300 DPI in TIFF-format file. File names are according to predefined format
- Generate ground truth annotation in XML file format for each word images and subset folder

To build an OCR text database, it is necessary to fill, print, and save a series of document images continuously. Therefore a batch mode command is needed at every stage or group of steps for practical use.

The modules of this system are implemented using MATLAB (R2009a/64-bit) Image Processing Toolbox, as well as programming in Delphi 7 and Microsoft Visual Basic 2010. It should be note that in this study it is chosen to use binary word images without applying any pre-processing.

3.6.1 Data collection form

In reality, the form layout is a nontrivial task since it must be simple for automatic word segmentation and labelling tasks. Due to the difficulties in word segmentation, the form layout designed in a way that simplify line and word segmentation which has to be computerized.

The structure and foundation of the collection form is inspired by the ideas presented in section 3.3. The collection form layout is illustrated in Figure 3.5 emerged after several trails. The form is A4-size white plain paper divided into three sections separated by horizontal lines: header, text section, and footer. The header and footer sections are for documentation purpose. The separating lines between sections are dark black lines; to be able to employ them as horizontal references as well. The header involves database title "Qur'an-MP Database"; MP denotes Machine Printed text, and code-number used for documentation and automation process.

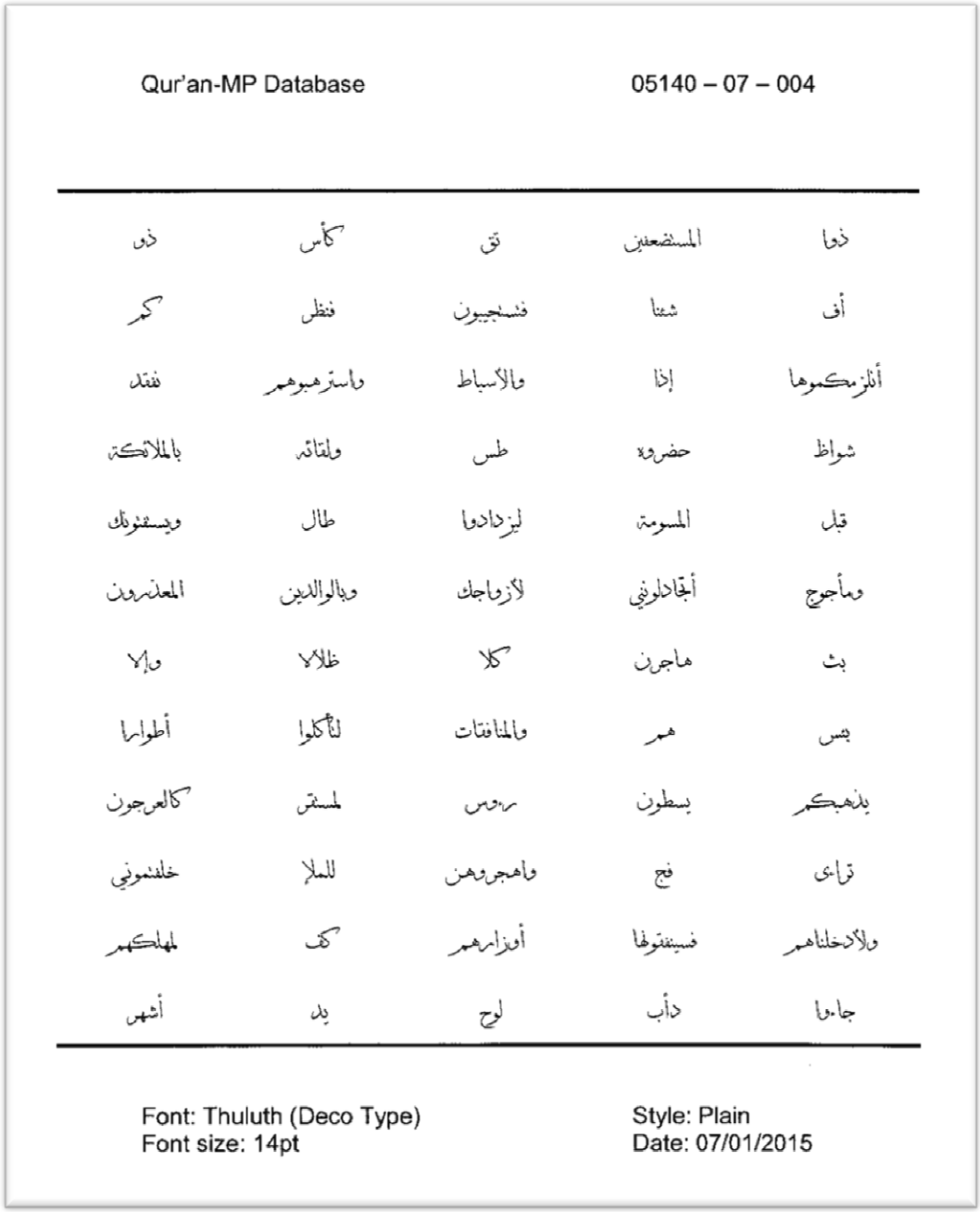


Figure 3.5: Form layout

Each form will have a unique code-number with three zones separated by dashes (e.g. 05140 – 07 – 004). This code-number comprise from left to right: the first zone includes data type one digit (Table 3.1) followed by four digits font information, they include font type in the first digit (Table 3.2), the next two digits are font size and the last digit is font

style (Table 3.3). For example, the code 03140 refers to real scanned data type, its printing font type is (3) Simplified Arabic font with size of 14 points in plain style (0). The second zone is two digits group-number, while the last zone is three digits page index in the group.

Table 3.1: Data types

Data type	Reference number
Scanned	Even digits including '0'
Synthetic	Odd digits

Table 3.2: Used Arabic fonts

Font name	Sample	Reference number
Andalus	رسم الخط العربي	1
Tahoma	رسم الخط العربي	2
Simplified Arabic	رسم الخط العربي	3
Traditional Arabic	رسم الخط العربي	4
Deco type Thuluth	رسم الخط العربي	5

Table 3.3: Font Style

Font style	Reference number
Plain	0
Bold	1
Italic	2
Bold and Italic	3

To simplify segmentation operations the text section organised in a table format. It helps the segmentation of line and word based on pixel location. The imaginary text table consists of five columns and twelve lines. Bearing in mind that word-shape is font dependent, table cells are sized enough to accommodate the largest word-shape in the corpus with less segmentation error. The text section can include up to the 60-word. The first word of the printed text is located on the top right in the text section to follow the reading and writing style of Arabic language. Font information at footer section includes font name, size, style, and printing date.

3.6.2 Data Preparation and Acquiring

Automatic generation of the filled forms is the first stage in the process. A Microsoft Word document generated contains structured form filled with text from the corpus. Each Word document is identified by its form code-number prefixed by font name separated from the first zone by dash. During the generation of Word documents another two text files are created, they are group and page reference list files for automatically word segmentation and labelling. Figure 3.6 shows samples of the

structure of these two files. The group reference file includes list of all generated forms at the same occasion. This list comprises form names, form code-number prefixed by database name "Quranic", followed by their contained number of lines and words, respectively. The form names are given to the scanned forms acquired by a scanner since the scanner assigns its own sequenced file names. The page reference file includes list of word-codes indicates their position in the corpus. The word-codes are ordered according to their occurrences in the forms, starting from top right of the first form to bottom left of the last form.

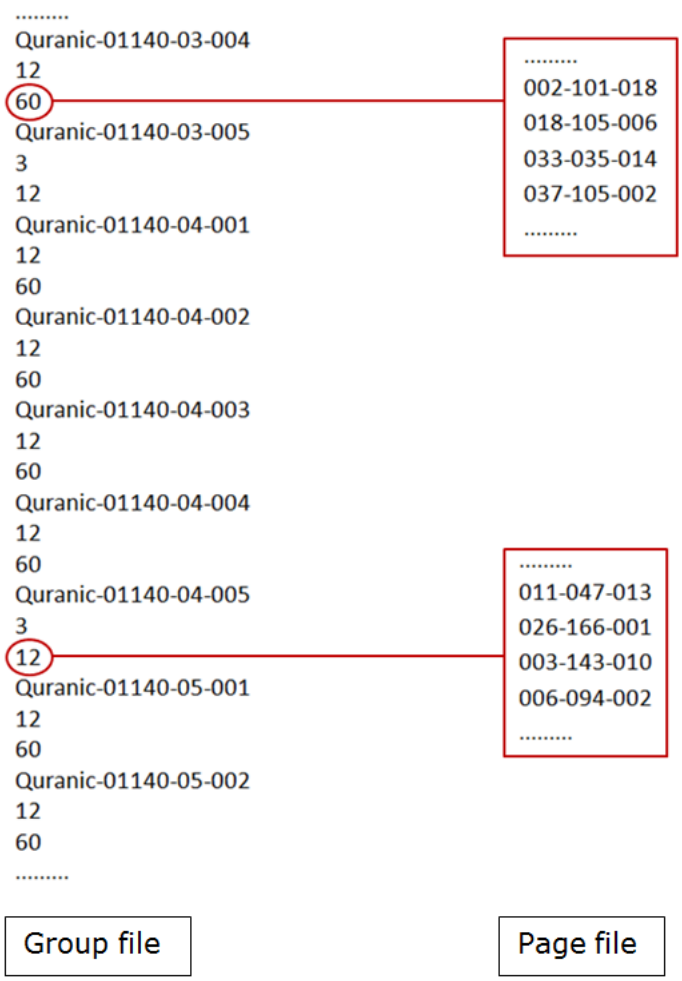


Figure 3.6: Group and Page reference files

Form creation passes by three stages: selection of words or sentences, document generation, and document to image conversion. The followings are applied for the two types of documents: training and testing purpose. Firstly, define number of groups; which refers to the frequency distribution of each word or sentence, and then select the words or sentences to be included. For both purposes, scatter the words over the table in the form text section, this is because most of the scanned documents can contain noises anywhere. The noise arises due to inefficient writing instrument, writing surfaces, or scanning machine. The binary images may have noises, include touching edges, broken strokes, filled loops, salt and pepper noises… etc. Scatter process is accompanied by group and page reference text-files creation.

In the synthetic image, the documents are converted to an image using dynamic programme simulates printing and scanning operations. In both ways scanned or synthetic, images are saved as binary (black and white) scale TIFF-format files with resolution of 300 DPI. To save storage and time, binarisation of the scanned image is done with the scanner hardware threshold. The really scanned forms have undergone two primarily stages of verification: the quality of printing and scanning. The final product is a scanned form having names as stated above.

3.6.3 Pre-processing and Word Segmentation

Pre-processing covers all those functions carried out to prepare original image to be suitable for later recognition stages. For off-line systems, pre-processing functions include: binarisation, noise filtering, skew detection and correction. These situations and others make it

difficult to analyse and process document images. Here, the aim of the pre-processing is mainly eliminating distortions at edges as a first step followed by skew detection and correction. Finally, document image is segmented into various zones like sections, text lines and words.

3.6.3.1 Skew Detection and Correction

In practice, these scanned documents can contain number of unavoidable and crucial problems; it can be noised, skewed, deformed.

in this context, presence of skew in scanned document images is a very common problem. The document image is skewed if it is not fed straight into the scanner either manually or automatically. Existence of a few degrees of skew within about three degrees is unavoidable [52]. This is feasible if the document is fed by a human operator. The automatic feeders may cause the document to rotate up to 20 degrees [52]. The skew of a document image called "global skew", where all text lines will have the same orientation, deviate from the true horizontal x-axis. Consequently, correcting the skew, orienting the text lines to be horizontal, is an important pre-processing step because it affects the efficiency of subsequent processing stages, such as segmentation and classification.

Skew correction is generally carried out by calculating the skew angle "θ" of the raw image and rotates it by "θ" in the opposite direction. A number of methods have been proposed for skew detection such as Projection Profile, Cross Correlation, and Hough Transform. In this work,

Hough transform, the widely used approach, is the adopted method for determining the skew angle.

The Hough transform technique detects lines, circles and other structures whenever their parametric equation is known. In the present context, it will be used for the detection of straight lines for skew angle determination. As the equation of any straight line in Cartesian space is:

Equation 3.1: Straight line equation

$$y = mx + b$$

The polar (also called normal) representation of straight lines is:

Equation 3.2: Straight line polar equation

$$\rho = x\cos\theta + y\sin\theta$$

Where ρ (rho) is the perpendicular distance of the line from the origin, and θ (theta) is the angle from the horizontal of the perpendicular line, Figure 3.7 illustrate this concept. In image analysis context, Hough transform maps each point in Cartesian image space (x, y) to a set of all straight lines going through that point in the (ρ, θ) Hough space.

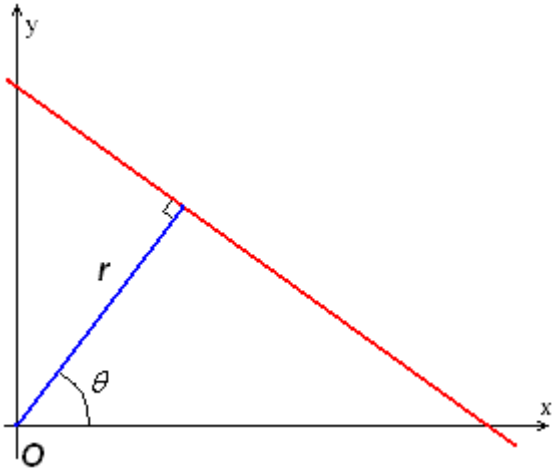


Figure 3.7: The representation of a line in the (x, y) space using (ρ, θ)

In general, to estimate the skew angle, the Hough-transform is applied on the image and the longest straight line will show the most accurate skew angle. The whole document image skew angle is calculated from the slope of one of the two separating lines. The line detection in a binary image can be summarized as follows:

1. Segment area enclosing pixels of one candidate separating lines; reducing the input data to process for low computational complexity

2. Apply an edge detection method to find all the edge points in the segmented area; In this work, the Canny method is applied for detecting boundaries of features within an image

3. Perform Hough transform on the detected edges for line detection. It maps all the data points in the image (x, y) into Hough space (p, θ)

4. The mapping result is the accumulator array element A(ρ, θ) represents the number of points lying on the corresponding line in the x-y plane

5. Detect best line candidates as local maxima in the accumulator cell array, the longest digital straight line, and its angle "θ" considered as the actual skew angle

In addition to the skew problem, the scanner sometimes presents distortions at the edge such as bounding box or lines. Before performing the skew angle detection, these edge distortions should be eliminated firstly. This can be done by pruning the image by specify the crop rectangle around the image with suitable margins, which can be defined experimentally.

3.6.3.2 Word Segmentation

The subsequent task to document-image skew correction is word segmentation. The imaginary text table plays a very important role in all levels of segmentation process; providing vertices pixel coordinates of text section and lines, and words. Accordingly, at different levels in the hierarchy, the crop rectangle around each component is defined and then it gets clipped. The clipped text section is as depicted in Figure 3.8, while Figure 3.9 shows a segmented line. Figure 3.10 illustrate word image, the prerequisite for the next step where each block should contain only one word.

Figure 3.8: Text section

Figure 3.9: Line segment

Figure 3.10: Segmented word using crop rectangle

The main objective of this stage is to find the "body" of the word in a binary image. To achieve this; firstly, complement the original word image to get binary-scale image having writing stroke in white pixels and background pixels are black. Secondly, the picture is "smeared" horizontally using Run-Length Smoothing Algorithm (RLSA). The RLSA is applied row-by-row to an image document. The principle of this smoothing algorithm is based on smearing a consecutive white pixels (represented by 1's) along the horizontal direction: i.e. the black space (represented by 0's) between them is filled with white pixels if their distance is within a predefined threshold.

Thirdly, find region boundaries of the largest white area in the smeared image. Finally, crop the word body in the original binary image based on the vertices of the smallest rectangle containing the region. Figure 3.11 displays the "body" of the word image as result of the tidy segmentation.

Figure 3.11: Tidy segmentation of the whole word body

The tidy whole word binary images are saved with resolution of 300 DPI in TIFF-format files. During all processing stages, images are verified at section, line, and word levels to ensure that there are no errors in segmentation process. The verification is crucial, because any random noise or smearing may make word segmentation imperfect. In the event of imperfect word tidy segmentation because of presence background noise, it's being eliminated manually and reapplies the Run-Length Smoothing Algorithm on it again.

The image name represents its attribute, and it is formatted, from left to right, as follows: one letter subset name, data type digit, font information part and the word position in the corpus. For example, the image file name a05140-01-002-164-006 indicates that the word-image is from subset (a), scanned, written in font number 5 in size 14 pts in plain style, related to group number 1, located in the corpus at chapter (2), verse number (164) word (6). The word-images are classified in group-subsets according to document type and font name.

3.6.4 Ground Truth

Ground truth (GT) refers to a number of information which describes attributes of each entry in the database. These ground truth annotation may including Number of words, PAWs, character sequence, font type, font size and so on. The GT data plays a vital role in recognition system development by providing information about the written text needed for text recognition. The availability of electronic corpus, facilitate automatic generation of GT files.

In this database each word image is accompanied by GT date in XML file format describing the image at word level. Figure 3.12 shows an example of an XML file at word level. The following GT information is available for each word image:

- Database name “Quranic_MP_Database”
- Lexicon word reference identifier
- Arabic word
- Number of PAWs
- Number of letters
- Word image file name
- Corpus name “The Holy Qura'n”
- Word identifier refers to its location in the corpus
- Writing instrument
- Binarisation
- Resolution
- Subset name
- Font identifier
- Font name
- Font size
- Font style

```
<?xml version="1.0" encoding="WINDOWS-1256"?>
- <Quranic_MP_Database>
    <LexiconWordID>3598</LexiconWordID>
    <Word>بنورهم</Word>
    <PAWs>3</PAWs>
    <Letters>6</Letters>
    <WordImage>a01140-01-002-017-012.tif</WordImage>
    <Corpus>The Holy Qura'n</Corpus>
    <CorpusWordID>002-017-012</CorpusWordID>
    <WritingInstrument>Machine</WritingInstrument>
    <Binarization>Scanner</Binarization>
    <Resolution>300 DPI</Resolution>
    <Subset>a0</Subset>
    <Group>01</Group>
    <FontID>1140</FontID>
    <Font>Andalus</Font>
    <Size>14</Size>
    <Style>Plain</Style>
</Quranic_MP_Database>
```

Figure 3.12: GT in XML file format for the word (بنورهم)

Another GT data file is provided for each subset folder that contains word-image samples. It comprises main information about all words in the subset and acting as lookup table for fast searching by one of the keywords: file name, word, or lexicon word-id. An example of XML file at subset level is given in Figure 3.13. The following GT information is available for each word image:

- Header information
    - Database name "Quranic_MP_Database"
    - Corpus name "The Holy Qura'n"
    - Writing instrument
    - Binarisation
    - Resolution
    - Subset name
    - Font identifier
    - Font name

    - Font size
    - Font style
- For each word image
    - Word image file name
    - Arabic word
    - Lexicon word reference identifier

The database files, images and GT, are stored in directory subset structure as depicted in Figure 3.14:

```
<?xml version="1.0" encoding="WINDOWS-1256"?>
- <Quranic_MP_Database>
  - <Head>
      <Corpus>The Holy Qura'n</Corpus>
      <WritingInstrument>Machine</WritingInstrument>
      <Binarization>Scanner</Binarization>
      <Resolution>300 DPI</Resolution>
      <Subset>a0</Subset>
      <FontID>1140</FontID>
      <Font>Andalus</Font>
      <Size>14</Size>
      <Style>Plain</Style>
    </Head>
  - <WordImage>
      <FileName>a01140-01-001-002-003.tif</FileName>
      <Word>رب</Word>
      <LexiconWordID>5529</LexiconWordID>
    </WordImage>
  - <WordImage>
      <FileName>a01140-01-002-004-011.tif</FileName>
      <Word>هم</Word>
      <LexiconWordID>10873</LexiconWordID>
    </WordImage>
  - <WordImage>
      <FileName>a01140-01-002-017-012.tif</FileName>
      <Word>يتورهم</Word>
      <LexiconWordID>3598</LexiconWordID>
    </WordImage>
  - <WordImage>
      <FileName>a01140-01-002-019-001.tif</FileName>
      <Word>أو</Word>
      <LexiconWordID>2755</LexiconWordID>
    </WordImage>
  - <WordImage>
      <FileName>a01140-01-002-025-024.tif</FileName>
      <Word>قيل</Word>
      <LexiconWordID>8236</LexiconWordID>
    </WordImage>
  - <WordImage>
      <FileName>a01140-01-002-025-030.tif</FileName>
      <Word>أزواج</Word>
      <LexiconWordID>667</LexiconWordID>
    </WordImage>
  - <WordImage>
      <FileName>a01140-01-002-030-004.tif</FileName>
      <Word>للملائكة</Word>
      <LexiconWordID>9256</LexiconWordID>
    </WordImage>
```

Figure 3.13: GT in XML file format for subset folder

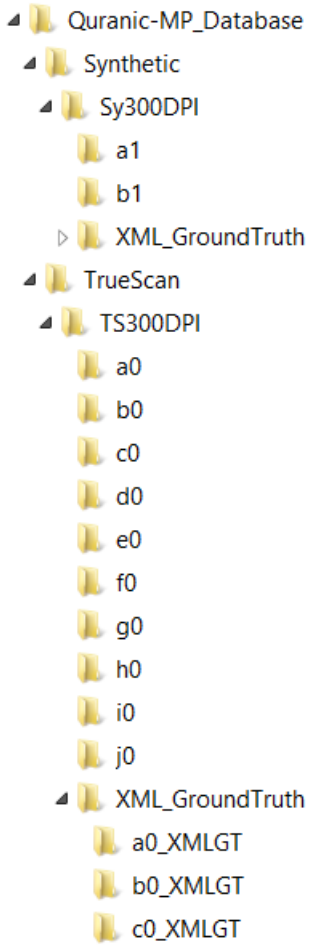


Figure 3.14: Database directory structure

3.7 Summary

Database implementation is a nontrivial task; it is not a simple collection of text images. This chapter presents a complete procedure to build a database considering the word as the major unit for text recognition. The procedure used to build machine printed text database, in general and Arabic script, in particular. The same procedure can be applied to construct handwritten words database with some modifications in the collection form.

Number of points influence database implementation is discussed. The constraints that should be satisfied to implement database fulfil this study requirements are described. Methodology of implementing of other related database is reviewed.

Text database implementation methodology described in details. Implementation methodology presents an automated system to create the database from a corpus. The fully computerized systems provide efficiency, accuracy, timelessness, security and economy. The automated system starts by form creating, and the following is word segmentation and labelling, and then ground truth generation.

Form layout designed in a way that simplifies line and word segmentation based on pixel location. Forms are designed using Microsoft Word and filled by words selected from The Holy Qur'an. The scanned forms are pre-processed for eliminating distortions at edges and skew detection and correction using Hough transform approach. The Run-Length Smoothing Algorithm used to help in cropping the word

body. File names of the tidy word binary images are following special format represents the word attributes. Finally, a ground truth files in XML format are supplied for each entry and subset folder in the database.

Note that all algorithms were developed using a variety of languages; MATLAB (R2009a/64-bit), Delphi 7, and Microsoft Visual Basic 2010.

CHAPTER 4

HMM/DCT HOLISTIC WHOLE WORD RECOGNISER

4.1 Introduction

Although other languages use Arabic letters, such as Persian and Urdu, Arabic character recognition has not reached the same level of maturity as other languages, especially English. This is attributed to a number of issues; lack of fundamental interaction between researchers in this field as well as deficiency of infrastructure supporting utilities, including Arabic text databases, electronic language corpora, and supporting staff; consequently each researcher has his own system and database with rare exceptions. Accordingly it is very difficult to give comparative results for the proposed methods due to the absence of standard benchmark databases. In addition to the former issues, the complexity of the Arabic script features demonstrates an additional challenge to build Arabic Character Recogniser; more details can be found in [11].

In this work, segmentation problem is avoided by considering the word as the major unit. The popular Block-based DCT transform applied to extracting word feature. The features of the entire word are fed to the recogniser to identify it without segmentation. The system is built on

HMMs, where each word was represented by a separate model. The applied HMMs to the task are Discrete 1D-HMMs built using the Hidden Markov Model Toolkit (HTK) [28]. Vector quantization is used to generate a discrete observation symbol density. In the recognition phase the recogniser produces N-best recognition word hypotheses lattice. A true scanned typewritten Arabic word image database for five different fonts is built for this research.

4.2 Thesis Methodology

The research objective is to develop a multi-font Arabic printed word recognition system for offline applications that accepts digital images of words (a sentence), computes features of each image, a word recognition process generates top-n candidate list for each word image (a word lattice); next, a procedure constructs all possible sentences from the word lattice; finally, constructed sentences are passed to linguistic post-processing stage to re-rank them according to their scores calculated using linguistic model. The outputs can be acquire top-n hypotheses from the scored sentences.

The technique to be applied for the design and implementation of the Arabic text recognition system is as follows:

- The corpus being used to experiment on is The Holy Qur'an
- Developing benchmark database
    a. Designing data collection form layout
    b. Microsoft Word creates document files contain filled form with words written in desired font properties
    c. For real scan images, print created documents and then scan them, while generate image for synthetic images

    d. Form images pre-processing and word segmentation
    e. Word images labelling and ground truth
- Word image global shape feature extraction
    a. Calculate the 2D-DCT coefficients of the image blocks, 8-by-8 pixels is the most commonly used block size
    b. After applying masking on each block, a number of coefficients are selected from each block and concatenated to construct a feature vector for the whole image
- Training phase
    a. Design of Discrete Hidden Markov Model (DHMM) models, n-state Left-to-Right topology, for each individual word from the training data set.
    b. Hidden Markov Model Toolkit (HTK) training tools estimates parameters for the HMM models using training words and their associated transcriptions [28]
- HTK recognition tools, also known as decoding tools, calculates the most likely words to the unknown word image, and produces the top-n hypotheses (a word lattice) from the lexicon for that word image
- construct n-gram SLM for the whole text of the corpus using HLM Toolkit provided by the HTK toolkit [28]
- Post-processing phase
    a. construct all possible sentences from the word lattice
    b. calculate the perplexity for each sentence in the test text using n-gram models
    c. re-rank the sentences according to their perplexity scores
- The system output is the list of the re-ranked sentences

4.3 Arabic Printed Datasets

One objective of this study is to establish a database to be used throughout this research work and to make it freely available for academic researches. The required database should have real scanned binarised samples of machine-printed Arabic words. Word image samples representing challenges of Arabic orthography to OCR technology; these challenges include, ligatures, overlaps, diacritics and style variation. All of these are well-known problems pose challenges to the recognition system.

Two databases were established:

- Quranic-252UW database for system validation and testing; it comprise 252 unique words (UW)
- Quranic-343UW database for provide evidence that the system developed is not predisposed to a particular words. it contains 343 unique words not presented in Quranic-252UW database

4.3.1 Word Set Selection Criteria:

The Arabic alphabet consists of 28 basic letters. Arabic writing is semi-cursive even when printed, and is written horizontally from right to left. Each word consists of one or more PAWs. The characters inside a PAW are normally connected to each other from the right side.

An Arabic text line can be considered as being composed of two baselines: an upper baseline and a lower one (see Figure 4.1). These two lines divide the word-image into three zones: 1) Upper zone: ascender

and upper diacritic points above the upper baseline; 2) Lower zone: descender and lower diacritic points which are under the lower baseline and 3) Middle zone: the main content of the word, the main body of the letters, is between the two baselines. Each letter has a part or all of its body in the middle zone, and, accordingly each text line has at least a middle zone.

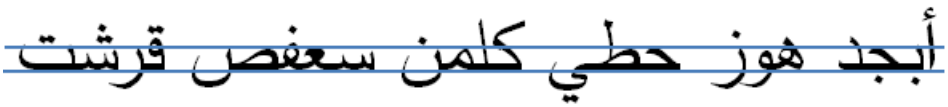


Figure 4.1: Upper and lower baselines of a text line (Simplified Arabic style of writing)

The word set has been chosen in order to capture, in different combination, the presence of ascenders, descenders and main body of the letters. Since more than half of the Arabic letters include dots in their shapes, dots have been taken into account in this work. Dots play an important role in the identification of characters when the characters' bodies have identical or similar shape. Dots can be one, two or three dots above or below the character's body. In addition to dots, Madda "~" and Hamza "ء" shapes are considered as well. Hamza may appear above or below the letter (e.g., "أ", "إ", "ؤ", "ئ") or enclosed within the letter main body ("ك") or isolated on the line ("ء"). The Madda may come instead of a Hamza on the Alif letter ("آ").

In this research, these words don't contain vowels and taTweel "elongation" character ("_"), but can have affixes and they are considered as a part of the word. Arabic affixes can be prefixes at the

beginning such as article "ال" and connector "و", suffixes at the end such as masculine plural "ون", and infixes take their position between two letters of the root to form the word radical such as "وا" positioned in between the root letters "فعل" to form a radical "فواعل". Multiple affixes can appear in a word as well as this combination is coherent.

Arabic writing uses many fonts and writing styles. Different Arabic fonts pose a dramatic change in the character's drawing and features with different font types. Arabic scripts are printed in fonts that show characters in different drawing and features such as ligature and overlapping, in addition to variations of character shapes which are font and position dependent.

4.3.2 Samples Distribution:

The availability of dataset for training and testing is a fundamental prerequisite for building a pattern recognition system. This data is not a simple collection of samples without consideration to classes' distribution, but it is necessary to consider the imbalanced class problem; classes with uneven distributions. This issue occurs when at least one of the classes having more sample than other classes in the training data [37] [39] [53]. In this case, the classifier would then be expected to perform best for classifying the frequently represented classes but very poor performance on the fewer representative classes.

To verify the performance of a classifier the data needs to be divided into disjoint sets; validation/training and testing sets. This demands

database maintains the balance of representations of different classes for training and testing procedures.

4.3.3 Statistical analysis of the Arabic Qur'an text corpus:

To select word-samples that can be included in this database, an authentic copy of the Arabic Qur'an text has been taken from (http://tanzil.info). The total number of words in the corpus text is 78,245 with 14,870 unique words. Table 4.1 shows frequency distribution of the characters and PAWs. Unique words frequency distribution shown in Table 4.2.

Table 4.1: Qur'an frequency distribution of the characters and PAWs

Characters	PAWs								Total
	1	2	3	4	5	6	7	8	
1	3	0	0	0	0	0	0	0	3
2	9514	2414	0	0	0	0	0	0	11928
3	6984	8571	915	0	0	0	0	0	16470
4	4830	9513	3094	257	0	0	0	0	17694
5	2241	4927	6182	993	61	0	0	0	14404
6	455	3064	4819	2057	349	11	0	0	10755
7	90	705	1498	1452	403	34	3	0	4185
8	16	324	864	679	428	51	0	1	2363
9	2	14	114	152	80	26	3	0	391
10	1	4	8	15	14	6	0	0	48
11	0	0	2	2	0	0	0	0	4
Total	24136	29536	17496	5607	1335	128	6	1	78245

Table 4.2: Qur'an unique words frequency distribution

Characters	Unique PAWs								Total
	1	2	3	4	5	6	7	8	
1	3	0	0	0	0	0	0	0	3
2	80	21	0	0	0	0	0	0	101
3	504	444	64	0	0	0	0	0	1012
4	1022	1478	616	47	0	0	0	0	3163
5	634	1719	1256	303	19	0	0	0	3931
6	268	1098	1472	597	108	7	0	0	3550
7	57	438	729	566	165	17	2	0	1974
8	13	150	278	254	133	31	0	1	860
9	2	14	66	91	48	17	3	0	241
10	1	3	6	12	7	3	0	0	32
11	0	0	1	2	0	0	0	0	3
Total	2584	5365	4488	1872	480	75	5	1	14870

4.3.4 Database implementation Process

The database used in this work is build up from words contained in the Qur'an text corpus. The Arabic words are printed in five fonts that are commonly used by the OCR researchers. Each token (word) is printed in the five Arabic fonts; Andalus, Simplified Arabic, Tahoma, Thuluth (Deco Type) and Traditional Arabic. The font size is 14 point in plain style. Table 4.3 lists samples of all used fonts.

Table 4.3: Samples of all used fonts typewritten in plain style 14 pt

Font name	Sample
Andalus	الأكثرية يميزون الخط العربي
Simplified Arabic	الأكثرية يميزون الخط العربي
Tahoma	الأكثرية يميزون الخط العربي
Thuluth (Deco Type)	الأكثرية يميزون الخط العربي
Traditional Arabic	الأكثرية يميزون الخط العربي

Heuristics are applied to select number of words. The combination of PAWs, ascenders, descenders and dots in addition to printing fonts guaranties a wide variability of images in the database. This variability also provides samples pose challenges to segmentation-based methods. These challenges include ligatures and overlapping and Arabic writing fonts. Another issue is the presence of dots that pose challenges to the recognition system; since there are letters have a same body (primary part) and distinguished only by the number and/or position of dots.

A comprehensive analysis and characterization of isolated handwritten Arabic letters are presented in [54]. Results for on- and off-line

handwriting recognition of isolated Arabic letters are reported, respectively, in references [54] and [55].

Word selection done according to selection criteria stated above considering two main standards defining word length; characters and PAWs count respects.

Word samples of the databases have been printed on a plain white A4 paper at specified resolution supported by the printer. Then, they were scanned as binary images (Black and White), as required by the feature extractor, with resolution set to 300 dpi. The image binarisation performed via the scanner hardware threshold to save storage and time. The scanned sheets are stored in a tiff file format. Sheet images pre-processing and word segmentation are done using MATLAB procedures. Table 4.4 illustrates samples of word-images printed in the five used fonts.

Table 4.4: Word-image samples for the five used fonts

Andalus	Tahoma	Simplified Arabic	Traditional Arabic	Thuluth
قد	قد	قد	قد	قد
ظمأ	ظمأ	ظمأ	ظمأ	ظمأ
كلمح	كلمح	كلمح	كلمح	كلمح
يصطفي	يصطفي	يصطفي	يصطفي	يصطفي
لمستقر	لمستقر	لمستقر	لمستقر	لمستقر
مطمئنين	مطمئنين	مطمئنين	مطمئنين	مطمئنين
للمسلمين	للمسلمين	للمسلمين	للمسلمين	للمسلمين
يستعجلونك	يستعجلونك	يستعجلونك	يستعجلونك	يستعجلونك
فسينفقونها	فسينفقونها	فسينفقونها	فسينفقونها	فسينفقونها
والمجاهدون	والمجاهدون	والمجاهدون	والمجاهدون	والمجاهدون
والذاكرين	والذاكرين	والذاكرين	والذاكرين	والذاكرين
لمردودون	لمردودون	لمردودون	لمردودون	لمردودون
ويدرءون	ويدرءون	ويدرءون	ويدرءون	ويدرءون
أوزارا	أوزارا	أوزارا	أوزارا	أوزارا
أزواج	أزواج	أزواج	أزواج	أزواج

4.3.4.1 Quranic-252UW database:

It is decided to select a set of word lengths (number of characters) have the same distribution, this leads to not consider the 1-character and 11-character word lengths. It is noteworthy that the number of samples per word character length is influenced by the 28-standard Arabic letters. Except 10-character length, where sample limitation exists, PAWs samples per each character length are selected to be equally (or nearly equally) sized. Table 4.5 shows distribution of the selected words according to the characters and PAWs lengths.

Table 4.5: Quranic-252UW database frequency distribution of the characters and PAWs

Characters	PAWs								Total
	1	2	3	4	5	6	7	8	
2	14	14	0	0	0	0	0	0	28
3	9	10	9	0	0	0	0	0	28
4	7	7	7	7	0	0	0	0	28
5	5	6	6	6	5	0	0	0	28
6	4	5	5	5	5	4	0	0	28
7	4	5	5	5	5	4	0	0	28
8	4	5	5	5	5	4	0	0	28
9	0	5	6	6	6	5	0	0	28
10	0	3	6	9	7	3	0	0	28
Total	47	60	49	43	33	20	0	0	252

The database is created using a lexicon includes 252 different words. For balancing requirements, each word has 25 samples at each font type. Therefore for each font type, there are total of 252 x 25 = 6,300 word samples. The database has 6,300 x 5 = 31,500 samples collected from real scanned documents.

Word samples of this database have been printed with resolution FastRes 1200 using a laser printer HP LaserJet M5035 Series MFP. The images are scanned on 300 dpi resolution using the same machine's scanner.

4.3.4.2 Quranic-343UW database:

Considering the unavailability of benchmarking databases for printed Arabic words that satisfy the requirements of this thesis; the Quranic-343UW database was developed. This database has importance to evaluate the developed system using unseen dataset which provide evidence that the system developed is not predisposed to a particular lexicon.

As can be seen in the Table 4.6, the number of words at each combination of word lengths, characters and PAWs, are equal to 9 unless there are not enough word samples in the corpus data. The words are randomly selected in order to get reliable results.

Table 4.6: Quranic-343UW database frequency distribution of the characters and PAWs

Characters	PAWs								Total
	1	2	3	4	5	6	7	8	
1	3	0	0	0	0	0	0	0	3
2	9	7	0	0	0	0	0	0	16
3	9	9	9	0	0	0	0	0	27
4	9	9	9	9	0	0	0	0	36
5	9	9	9	9	9	0	0	0	45
6	9	9	9	9	9	3	0	0	48
7	9	9	9	9	9	9	2	0	56
8	9	9	9	9	9	9	0	1	55
9	2	9	9	9	9	9	3	0	50
10	1	0	0	3	0	0	0	0	4
11	0	0	1	2	0	0	0	0	3
Total	69	70	64	59	45	30	5	1	343

The database contained 25,725 word instances for the five fonts, with a 343 word vocabulary. To maintain the balance of representations of different classes, each word has 15 samples at each font type. This means that each font has 343 x 15 = 5,145 samples collected from real scanned documents.

Word samples of this database have been printed at 600 dpi resolution with a laser printer, KONICA MINOLTA C364 Series PCL (bizhub C224e). The binary images were scanned with a resolution of 300 dpi using KONICA MINOLTA 423 Series PS (bizhub 363).

4.4 Proposed Recogniser Methodology

The objective is to develop a holistic whole word recognition system for offline applications that accepts Arabic word printed with five different fonts as listed in Table 4.3.

Methodology of training and recognition of the proposed recogniser is as shown in the block diagrams given in Figure 4.2. The system can be broken up into functional components: normalisation, feature extraction, and training and classification.

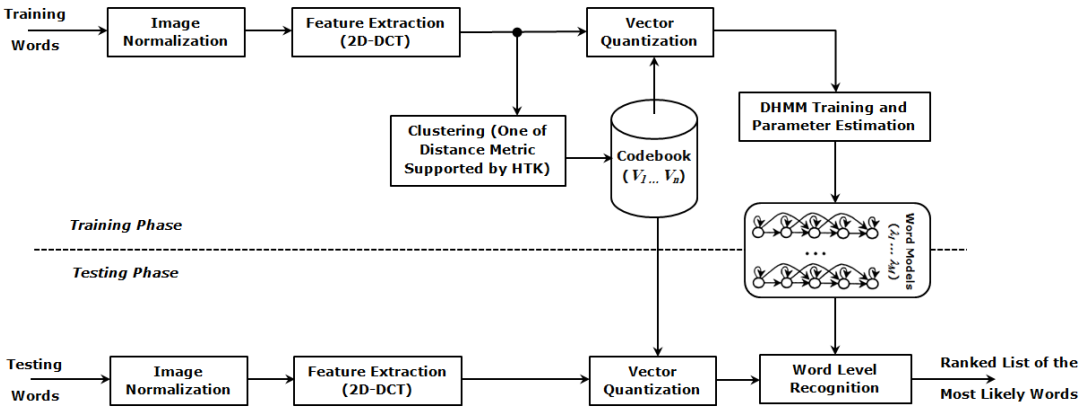


Figure 4.2: A block diagram of the proposed holistic whole word recogniser

The MATLAB (R2009a/64-bit) Image Processing Toolbox is used to generate the feature vectors. The HMM modules implementation is based on HTK toolkit, which was originally developed for speech recognition [28]. To speed up the algorithms and increase the efficiency, the overall system is implemented in Delphi 7 platform.

Normalisation: Before feature extraction, a number of normalisation operations are applied to a word image to convert images of various sizes into a single standard size. Normalisation operations are important in order to reduce the variability of the features to achieve high text recognition rates. Taking this into consideration, for the binary image, where the word is represented by black pixels and the background by white pixels, a number of steps that need to be taken to normalise the word image as shown in Figure 4.3. As can be seen from figure, the normalisation process comprises five main steps: 1) Flip the word image along the vertical axis to ensure consistency with the used techniques which are applied from left to right. 2) Image negating, where the word is represented by white text on black background. 3) Convert the binary image to a grey image. 4) For each font type, vertical and Horizontal compression or dilation for height and width normalisation, respectively, according to the calculated average width "W_{avg}" and average height "H_{avg}" of the training words. 5) Shrink the result image by a ratio "Q", this ratio adjusted systematically to the nearest ratio greater than or equal to "Q" to return image having rows and columns equal to integral multiples of "T". In addition to these steps, a Gaussian filter is applied to the image after each step to obtain a smooth grey-scale image.

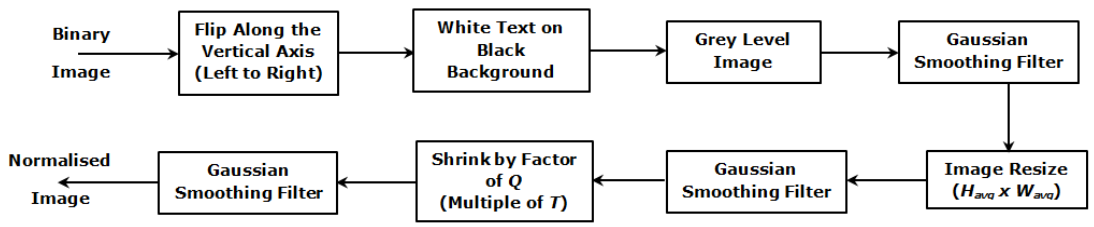


Figure 4.3: Block diagram of the applied word-image normalization

Feature extraction: Feature extraction means to extract the most significant features to represent the whole data with the least amount of elements which maximizes the recognition rate. The major goal of this stage is converting the word image to an appropriate sequential form suitable for the HMM recogniser module. A Block-Based DCT (BBDCT) feature extractor approach was used in this work. The proposed feature extractor system consists of three stages. In the first stage, the normalised image equally divided into non overlapped blocks of "T×T" dimension. In the next stage, the 2D-DCT is performed independently on the sub-image blocks to obtain the DCT coefficients. Finally, after applying the zigzag masking on each block as presented in section 2.3 and depicted in Figure 4.4, the first "r" coefficients are selected from each block to construct feature vector for the whole image.

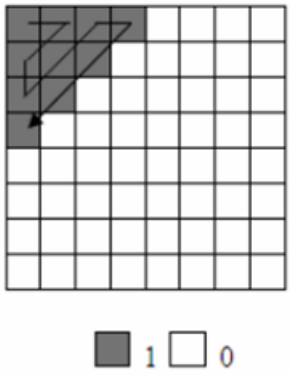


Figure 4.4: DCT coefficients selection using zigzag masking

DHMM recogniser: The recognition unit is the core of a complete OCR system. HMMs are one of the most widely and successfully used technique for Arabic character and word recognition problem. The designed system, in this work, is a closed vocabulary off–line machine-written word recognition based on Discrete 1D-HMMs. Each unique word

that appear in training data, word class (c), is modelled by a left-to-right HMM model (λ_c). Each word model has number of states and transitions between the states. The HTK is used for training and recognition phases. There are four main phases when constructing a text recognition system implemented using the HTK: data preparation, selection of a suitable HMM model structure, training and testing.

The goal of data preparation is to suit the needs of the HTK tool set. HTK has tools to prepare data for training and testing processes. After feature extraction stage in which the word image is transferred into a feature vector, these vectors should be written one by one into HTK format data files. For using a Discrete HMM with a discrete observation symbol density, rather than the continuous vectors, a vector quantisation (VQ) is used to map continuous density vectors into discrete symbols depending on the codebook index. The HQuant tool is directly applied to the feature vectors of the training subset, generating a VQ table (or so-called codebook) of predefined size. This tool supports three types of distance metric (simple Euclidean, full covariance Mahalanobis or diagonal covariance Mahalanobis) and two organisations of VQ codebook (simple linear table or binary tree). Once the codebook is built, HCopy tool convert feature-vector files (training and testing) to discrete files.

In the HMM model structure phase, a prototype HMM model is created for each training word image. A prototype model describes the topology of HMM models to be trained. The description includes number of states and the allowable transitions between states.

The next step is the model training. The parameters of the created prototype HMM models are re-estimated using the data from the discrete files. HERest is the main HTK training tool. It uses the Baum-Welch (Forward-Backward) algorithm to refine the parameters of the whole set of the HMMs simultaneously "embedded training".

Finally, the recognition is performed through the HTK command HVite. It is a general-purpose Viterbi word recogniser (Viterbi decoding) produces n-best word recognition list ranked depending on the probabilities of the word hypotheses.

4.5 Overfitting Problem

Overfitting is persistent problem in the machine learning community. The classifier may suffer from over-fitting issue caused by many factors, and the most important considerations must be taken to avoid this issue are:

a) Interdependency of training and testing data; using the same data for training and testing (the dependency) might over-train the classifier so that it learns the training data but is not able to generalize to other data not seen in the training phase. To overcome this problem, a different set of data called validation set is used during the learning

b) Size of training data; small sample sizes, may lead to risk of inaccuracy and overfitting, that is, parameters of the model are over-fitted to the particular training data. Cross-validation

method can help to combat overfitting when amount of data is limited

c) Noise free and synthetic image dataset; the main drawback of this approach is that it may bias a recogniser towards unnatural image styles. The resulting images can be physically or synthetically degraded by degradation modules

d) Dimensionality of the feature space; working on higher dimension the system starts to memorize and fail to classify new input data. Feature extraction is a special form of dimensionality reduction. The input data will be transformed into a reduced representation set of features. To ensuring a better generalization of the classifier, it is important to select type of feature extraction and the extracted features must be chosen carefully

e) Optimisation of the classifier parameters; the classifier has parameters that are provided by the user, such as number of HMM states and ANN layers, where the user starts with an initial value (size or number) and then he able to adjust these parameters incrementally to train the classifier. The larger the value, the more powerful the classifier. However, if one continues to increase the value, there is a point where the generalization gets worse. This is due to the fact that the classifier fall into over-fitting to the training data

f) Number of training iterations; if too many iterations or epochs are used, the system will tend to memorize the data instead of discovering the features. For this reason the re-estimation process is repeated until convergence is achieved or an upper limit on the iteration count is reached

All of these factors significantly impact recognition performance and leads to lower recognition rates due probably to over-fitting.

4.6 Experiment Parameters

The HMM classifier has number of parameters to be defined; including sub-block and mask sizes, HMMs word-model topology, the number of states and there transitions combined with their initial transition probabilities, and codebook features. Since these parameters can't be calculated mathematically, they are defined manually and a large number of experiments were conducted to find the most suitable combinations of these parameters that are adequate to all fonts and give high recognition rate (RR).

The word-images contained in the used database are already pre-processed binary images of single words. Horizontal and vertical image lengths are normalised according to the calculated averages (width and height) of the training words. Images are then down sampled to the nearest ratio greater than or equal to 1/2 of its original size to insure that the number of rows and columns are equal to integral multiples of "8". In all cases, a Gaussian filter with a standard deviation "σ" = 1 and a

kernel size of 3x3 is applied to the word image to obtain a smoothed grey level image.

In feature extraction phase, some of the coefficients are selected to construct a feature vector. The feature vectors size is font and training set dependent; it is defined systematically based on the dimension of the normalised training images, which influence the number of sub-blocks, and the mask size. At first, the BDCT is applied to the normalised image with sub-block size 8x8 in a non-overlapping manner. Then the zigzag masking method is used to select first number of coefficients from each sub-block to construct feature vector of each image. The mask size is found font dependent where each font has its own mask size as shown in Table 4.7.

Table 4.7: Font mask size

Font	Mask size
Andalus	8
Simplified Arabic	8
Tahoma	8
Thuluth DecoType	5
Traditional Arabic	10

The adopted HMMs topology is left-to-right model (or Bakis model) with start and end states. The used Bakis model allows transition to itself, the next state and the one after the next state. The exceptions are the first and the last states (an example is illustrated in Figure 4.5). The topology and the transitions between states are defined through the transition matrix of the HMM described in <TransP> subsection in prototype definition file as depicted on Figure 4.6. Zero probability values

indicate not allowed transition between states and the sum of each line of the matrix (initial transition probabilities) must be "1".

The optimal codebook size was chosen by experiments. The codebook features are: binary tree structured, generated by "diagonal covariance Mahalanobis" metric algorithm and its size is "32" for all the systems.

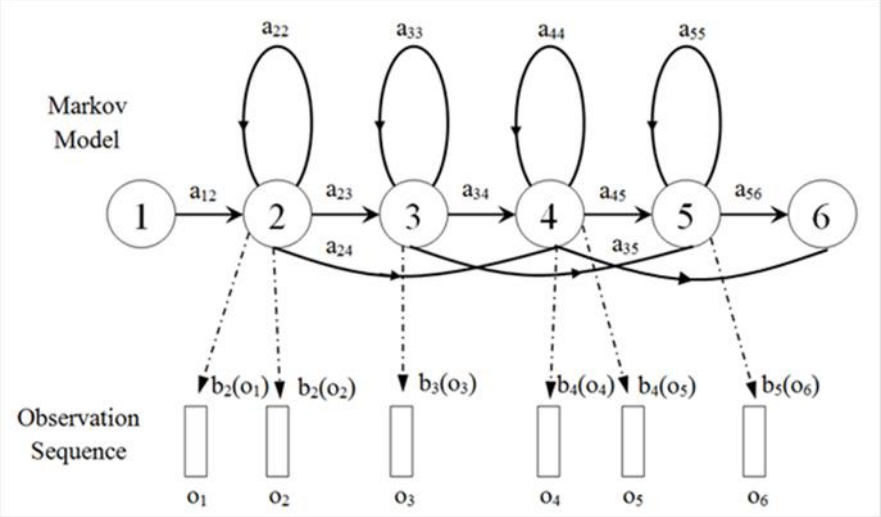


Figure 4.5: Left-to-right HMM model with 5-state (Bakis model)

```
      <TransP> 7
            0.0   1.0   0.0   0.0   0.0   0.0   0.0
            0.0   0.4   0.4   0.2   0.0   0.0   0.0
            0.0   0.0   0.5   0.5   0.0   0.0   0.0
            0.0   0.0   0.0   0.4   0.4   0.2   0.0
            0.0   0.0   0.0   0.0   0.5   0.5   0.0
            0.0   0.0   0.0   0.0   0.0   0.5   0.5
            0.0   0.0   0.0   0.0   0.0   0.0   0.0
<EndHMM>
```

Figure 4.6: Discrete HMM description file (prototype)

To avoid the over-fitting or over-learning problem, it has been suggested in HTK book number of steps that need to be taken when performing embedded training which executed by HERest. Investigate

the improvements in accuracy in each iteration by monitoring the overall log likelihood per frame provided by the HERest and stop training when no further improvement is obtained (may take an impossibly long time) or a chosen number of iterations have been performed (it has been advised 2 to 5 cycle). Empirically It has been found that doing more than "3" iterations lead to overfitting.

4.7 Experimental Results

The experiments were carried out on word samples of the 5 fonts in the created databases presented in section 4.3. The score of word recognition rate (WRR) is computed to compare the achieved results with other studies. The WRR was measured as the percent of the correctly classified words to the tested words as follows:

Equation 4.1: Word recognition rate (WRR)

$$WRR = \frac{Number\ of\ correct\ matches\ in\ test\ set}{Number\ of\ words\ in\ test\ set}$$

During the initial training of the system, it has been noted that no remarkable improvement can be achieved when number of training samples exceed "10" word-samples of each class. To avoid over-fitting, data was split into a training and test set. The test data is held out and not looked at during training.

For each database, the number of used samples for each unique word for each font-type in the experiments is 15–observation. Each dataset is divided to train and test sets, the distribution percentage are 67% for training and 33% for testing. Generally, all experiments (unless stated

otherwise) are conducted in the mono-font system where a different model is trained for each word font-type.

4.7.1 Performance Estimation

The proposed recogniser is evaluated with samples from the developed Quranic-252UW database. To evaluate the performance of the used classifier for each training dataset, a stratified k-fold cross-validation was performed for estimating test error. This prevents overfitting to a specific training set.

The k-fold cross-validation splits the training dataset into k disjoint subsets (roughly equal-sized parts) each with a class distribution approximating that of the original dataset. The recommended choices of k are 5 or 10. Overall estimated performance is reported by averaging WRR across the folds.

For each font-type dataset, the number of used samples in the validation experiments is 10–observation for each word in total of 10 x 252 = 2,520 word samples.

A set of two stratified five-fold cross-validation (2 x 5-fold cross-validation) has been performed. Each time 80% of the samples in the training set are used for training and the remaining 20% for validation. Five distinct sets/folds (a, b, c, d, e) were defined in the training dataset to be used for training and validating systems, four of them are used for training and one fold for validating the system. Each fold containing 504 word instances of 252 distinct words, where every word appears two

times (samples) in each fold. In general, 5-fold cross-validation was done for each font type, repeated for 5 trials, and average WRR was reported.

Table 4.8 shows the two founded systematically parameters which are font and training set dependent: the size of the normalised word-images and the feature vectors.

Table 4.9 to Table 4.13 show the performance estimation (top-1) on the validation set of the system using 5-fold cross-validation for each font type.

Table 4.8: Calculated parameters for validating mono-font Quranic-252UW database

Mono-Font 5-fold cross-validation (Quranic-252UW database)	Mask size	Dimension of normalised word-image		Feature size
		Width (Columns)	Height (Rows)	
Andalus	8	72	32	288
Tahoma	8	88	32	352
Simplified Arabic	8	72	32	288
Traditional Arabic	10	56	24	210
Deco type Thuluth	5	72	40	225

Table 4.9: Fivefold cross-validation WRR obtained for Andalus font

Experiment	Training folds	Testing fold	WRR %
1	a, b, c, d	e	99.01
2	a, b, c, e	d	97.82
3	a, b, d, e	c	97.62
4	a, c, d, e	b	98.41
5	b, c, d, e	a	98.21
	Average		98.21

Table 4.10: Fivefold cross-validation WRR obtained for Tahoma font

Experiment	Training folds	Testing fold	WRR %
1	a, b, c, d	e	97.62
2	a, b, c, e	d	97.62
3	a, b, d, e	c	99.21
4	a, c, d, e	b	97.02
5	b, c, d, e	a	97.42
	Average		97.78

Table 4.11: Fivefold cross-validation WRR obtained for Simplified Arabic font

Experiment	Training folds	Testing fold	WRR %
1	a, b, c, d	e	98.02
2	a, b, c, e	d	98.21
3	a, b, d, e	c	99.01
4	a, c, d, e	b	99.21
5	b, c, d, e	a	98.41
	Average		98.57

Table 4.12: Fivefold cross-validation WRR obtained for Traditional Arabic font

Experiment	Training folds	Testing fold	WRR %
1	a, b, c, d	e	97.02
2	a, b, c, e	d	97.22
3	a, b, d, e	c	95.83
4	a, c, d, e	b	96.23
5	b, c, d, e	a	96.83
	Average		96.63

Table 4.13: Fivefold cross-validation WRR obtained for Deco type Thuluth font

Experiment	Training folds	Testing fold	WRR %
1	a, b, c, d	e	98.41
2	a, b, c, e	d	98.81
3	a, b, d, e	c	98.02
4	a, c, d, e	b	97.62
5	b, c, d, e	a	98.41
	Average		98.25

Table 4.14: Cross-validation average WRR obtained for all font-types

Mono-Font	Average WRR %
Andalus	98.21
Tahoma	97.78
Simplified Arabic	98.57
Traditional Arabic	96.63
Deco type Thuluth	98.25

As it can be noticed from Table 4.14, the average results for all font-types are very similar, in the range 96.63% – 98.57%. The Simplified Arabic, Deco type Thuluth and Andalus fonts have the best performance (98.57%, 98.25% and 98.21%, respectively). In the other side, Traditional Arabic and Tahoma have the lowest performance (96.63% and 97.78%, respectively). The final result is very encouraging, and the proposed methodology exhibited promising results on tasks of typewritten Arabic word classification.

4.7.2 Classification accuracy

To prove the basic ideas presented in this thesis, the recognition of word images are carried out by two approaches: mono-font approach

and multi-font approach. Classification experiments are conducted on dataset selected from the Quranic-252UW database. Each font-type dataset contains 2,520 training samples (10 samples per word) and 1,260 test samples (5 samples per word).

The mono-font approach experiments have been executed by using codebook of size 32 and are referred to as "Mono-font-252UW-32CB". The acronyms UW and CB stand for Unique-Word and Code-Book, respectively. The multi-font approach experiments have been executed by using 32 and 64 codebook sizes and are referred to as "Multi-font-252UW-32CB" and "Multi-font-252UW-64CB", respectively.

4.7.2.1 Mono-font Classification (Mono-font-252UW-32CB)

This experiment is referred to as "Mono-font-252UW-32CB". The system was tested on a base of Quranic-252UW database. In this system, for each font style, a different model for each word is trained with codebook size of 32.

For each font-type dataset, the number of used word-samples in the training is 10–observation for each word-model. Given the test sample, its normalised word-images dimension and the feature vectors from all the 2,520 training samples are found. Combinations of dimension of normalised samples and size of feature vector founded for different fonts for Quranic-252UW database are summarised in Table 4.15.

Table 4.15: Calculated parameters for Mono-font-252UW

Mono-font-252UW	Mask size	Dimension of normalised word-image		Feature size
		Width (Columns)	*Height (Rows)*	
Andalus	8	72	32	288
Tahoma	8	88	32	352
Simplified Arabic	8	72	32	288
Traditional Arabic	10	56	24	210
Deco type Thuluth	5	72	40	225

Each font had 1,260 testing samples, for each unique word (252) five samples were used. The five samples which are used for testing purpose are not used in training part. Table 4.16 shows the overall recognition rates for different font styles. The system achieves an average recognition rate of 97.56%. The results show that the Andalus font gets slightly better recognition accuracy of 98.65% than other four fonts. The WRRs of the other 4 fonts were: 97.94% for Simplified Arabic, 97.62% for Tahoma, 96.83% for Traditional Arabic, and 96.75% for Thuluth.

Table 4.16: Mono-font-252UW-32CB experiment results (WRRs)

Mono-font-252UW-32CB	Top-1 %	Top-2 %	Top-3 %	Top-5 %	Top-7 %	Top-10 %	Top-15 %	Top-20 %	Top-25 %	Top-30 %
Andalus	98.65	99.44	99.52	99.60	99.68	99.76	99.76	99.76	99.84	99.84
Tahoma	97.62	98.73	98.97	99.44	99.60	99.68	99.68	99.68	99.68	99.76
Simplified Arabic	97.94	98.73	99.21	99.44	99.44	99.52	99.84	99.92	99.92	99.92
Traditional Arabic	96.83	98.49	99.37	99.68	99.76	99.84	99.92	99.92	99.92	99.92
Deco type Thuluth	96.75	98.81	99.21	99.52	99.52	99.68	99.76	99.76	99.76	99.84
Average	97.56	98.84	99.26	99.54	99.60	99.70	99.79	99.81	99.82	99.86

Table 4.17 depicts the classification accuracy (WRR in percentage) for validation/training and test sets. Experiments for the five fonts were carried out based on 2x5-fold cross validation and the average WRR was 97.89% corresponds to average WRR 97.56% reported for test set.

The results show that the proposed word recognition method is robustness and will generalize to an independent data set.

Table 4.17: Validation and test set performance (WRR%)

Mono-font	Mono-font 2x5-fold cross-validation (Average)	Mono-font-252UW-32CB (Top-1)
Andalus	98.21	98.65
Tahoma	97.78	97.62
Simplified Arabic	98.57	97.94
Traditional Arabic	96.63	96.83
Deco type Thuluth	98.25	96.75
Average	97.89	97.56

4.7.2.2 Multi-font Classification

In all of the earlier experiments, one font-type was used for each experiment at a time. Some set of the font-type word images were used for training and other set of word images from the same font-type were used for testing. In the current experiment a multi–font system is used.

A set of experiments were performed to test the system on multi-font models using the same mono-font classifier parameters with two exceptions: a) each word has only one HMM model in the system whatever the font-type of the word. b) Mask size of 8-element calculated by averaging the mask sizes in mono-font. Each word-model is trained using 10 samples from each font in total of 50 samples.

The classier was trained on a set of 12,600 words. The word-images were normalised to have a width 72 pixels and a height 32 pixels, and the extracted feature vector have size of 288 coefficients. The multi–font system is tested with testing sets from only a single font type at a time. The test set for each font-type is composed of 1,260 words, classifies under 252 classes, with 5 words for each class.

Multi–font system experiments were conducted to test the proposed method using two different codebook sizes, 32 and 64.

4.7.2.2.1 Multi-font-252UW-32CB

Table 4.18 summarises the WRRs obtained by the multi-font recognition system in top n cases, where n = 1, 2,…, 30. In top-1 position, the Simplified Arabic scored the highest WRRs 61.11% while Andalus had the lowest performance 48.97%. In the experiments, top-1, top-10 and top-30 results had 56.17%, 89.63% and 97.25% average accuracy, respectively.

Table 4.18: Multi-font-252UW-32CB experiment results (WRRs)

Multi-font-252UW-32CB	Top-1 %	Top-2 %	Top-3 %	Top-5 %	Top-7 %	Top-10 %	Top-15 %	Top-20 %	Top-25 %	Top-30 %
Andalus	48.97	62.78	71.19	78.57	83.25	87.70	91.51	93.89	95.40	96.83
Tahoma	54.76	67.54	76.59	83.25	86.59	89.52	93.57	94.92	96.59	97.54
Simplified Arabic	61.11	73.89	79.60	85.16	89.29	91.98	94.68	96.83	97.54	97.94
Traditional Arabic	61.03	73.33	79.76	85.40	88.33	90.56	93.57	95.71	96.51	96.98
Deco type Thuluth	55.00	65.79	72.54	79.76	83.97	88.41	92.14	94.76	95.95	96.98
Average	56.17	68.67	75.94	82.43	86.29	89.63	93.09	95.22	96.40	97.25

Figure 4.7 illustrates system performance with respect to minimum (Andalus), maximum (Simplified Arabic) and average WRRs and numbers of candidates. It is worth noting that the recognition rate increases

strongly at first (from top-1 to top-2), but the performance gain slows down for high "n". The Andalus font has the highest recognition rate jump from top-1 to top-2; where the WRR jumped from 48.97% to 62.78%. It is also noteworthy to mention that the system performance and stability was much better at higher top-n and is comparable to the results reported in the mono-font approach.

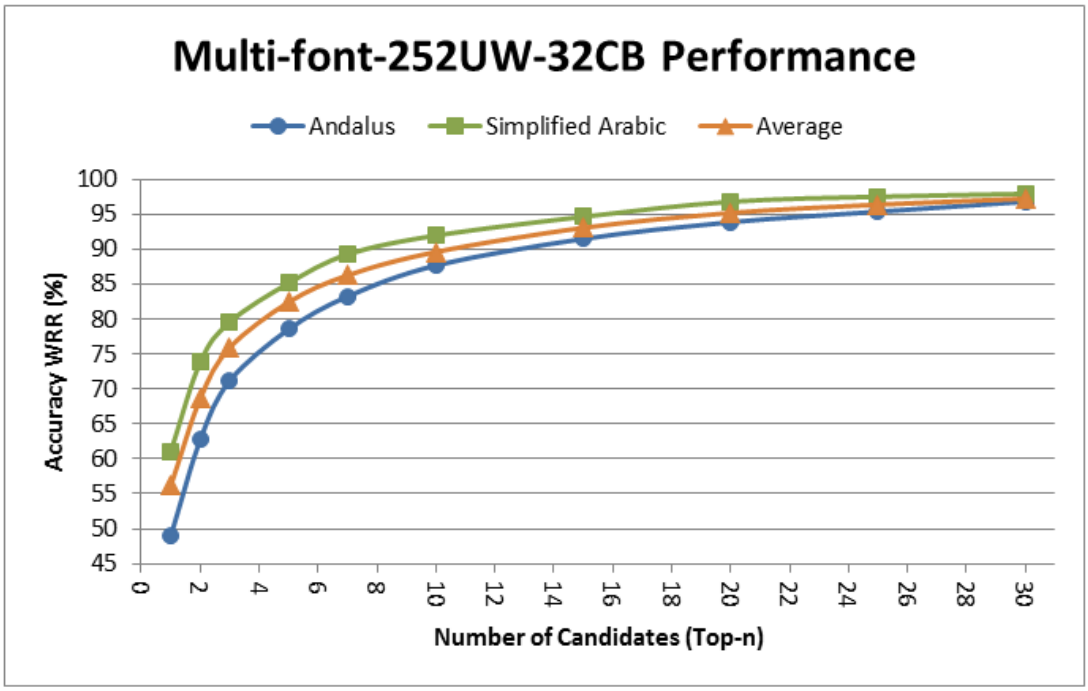


Figure 4.7: Multi-font-252UW-32CB Performance

4.7.2.2.2 Multi-font-252UW-64CB

The codebook size is a key factor that affects the recognition accuracy. For a given HMM model structure and feature set, it has been noticed that the system performance increased as codebook size increased until the system reaches its saturation point. Similar findings were reported by different authors including Khorsheed [35] and AlKhateeb et al. [39].

Font variation increases greatly with the use of multi-fonts (5 fonts), which adds to the complexity of the problem. The codebook size is specified empirically to cope with the large variance of all fonts' shapes in order to obtain the best performance. Table 4.19 shows the system performance as the codebook size increases for Andalus font. Andalus font was selected because it had the lowest performance according to the results reported in Table 4.18.

Table 4.19: Multi-font-252UW WRRs for Andalus using different codebook

Codebook size	Top-1 %	Top-2 %	Top-3 %	Top-5 %	Top-7 %	Top-10 %	Top-15 %	Top-20 %	Top-25 %	Top-30 %
32	48.97	62.78	71.19	78.57	83.25	87.7	91.51	93.89	95.4	96.83
64	61.98	73.25	79.92	87.3	90.32	93.1	95.79	97.38	97.94	98.49
128	62.94	76.03	81.59	86.59	89.52	92.54	94.92	96.75	97.62	98.25
256	57.38	68.41	74.21	81.9	85.87	89.84	94.05	96.19	97.06	97.62

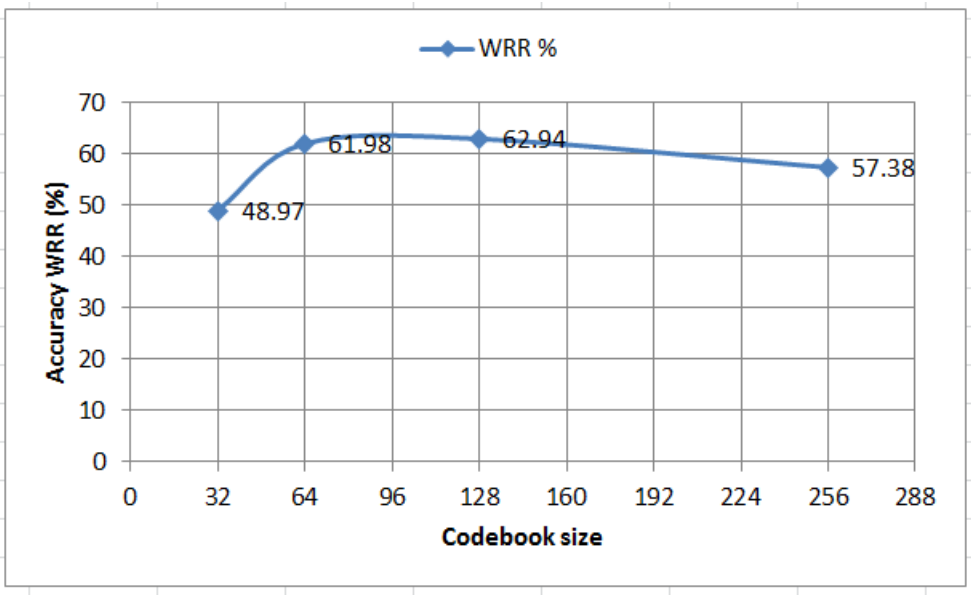


Figure 4.8: Multi-font-252UW for Andalus top-1, effect of varying the size of codebook on recognition accuracy

As shown in Figure 4.8, the system performance increases with increasing codebook size in each case. The figure shows a saturation of the performance when the codebook size reaches 128. It can be observed that the improvement (at top-1) was slightly better at codebook of size 128 (62.94%) than size 64 (61.98%). However, the computation time increases with the codebook size. As a result, an optimal codebook size is set to 64 based on a trade-off between computation time and enhance the recognition rate.

Table 4.20 summarises the results obtained by the multi-font recognition system uses 64 codebook size value. The results exhibit significant performance improvements compared to the results obtained based on the 32 codebook size value as reported earlier in subsection 4.7.2.2.1.

Table 4.20: Multi-font-252UW-64CB experiment results (WRRs)

Multi-font-252UW-64CB	Top-1 %	Top-2 %	Top-3 %	Top-5 %	Top-7 %	Top-10 %	Top-15 %	Top-20 %	Top-25 %	Top-30 %
Andalus	61.98	73.25	79.92	87.30	90.32	93.10	95.79	97.38	97.94	98.49
Tahoma	64.37	76.03	82.06	87.86	91.59	93.49	96.03	96.83	97.38	97.86
Simplified Arabic	72.14	81.90	86.43	90.87	93.41	95.71	97.46	98.25	98.57	98.89
Traditional Arabic	66.83	80.24	86.67	91.83	93.33	95.08	96.98	98.33	98.89	99.13
Deco type Thuluth	67.70	78.33	83.33	88.81	91.19	94.13	96.83	97.38	97.94	98.41
Average	66.60	77.95	83.68	89.33	91.97	94.30	96.62	97.63	98.14	98.56

A comparison between the results of the two approaches (Multi-font-252UW-32CB and Multi-font-252UW-64CB) is summarized in Table 4.21. The results show that the Multi-font-252UW-64CB gets better average recognition accuracy of 66.60% compared to the Multi-font-252UW-32CB‘s 56.17%.

Table 4.21: Comparing top-1 WRR% for Multi-font-252UW-32CB and Multi-font-252UW-64CB

Multi-font-252UW	Multi-font-252UW-32CB (Top-1)	Multi-font-252UW-64CB (Top-1)
Andalus	48.97	61.98
Tahoma	54.76	64.37
Simplified Arabic	61.11	72.14
Traditional Arabic	61.03	66.83
Deco type Thuluth	55.00	67.70
Average	56.17	66.60

4.7.3 Classifier stability test

One way of assessing the stability of a classifying algorithm is to compare obtained resulting recognition with modified versions of the input data. A classifier type is called stable when its behaviour changes only slightly when the training set is modified. The proposed recogniser is tested with created Quranic-343UW database containing 343 unique words not presented in Quranic-252UW database. Testing the stability of the classifier was carried out in the same way as in the previous section 4.7.2.

4.7.3.1 Mono-font Classification Stability

The calculated parameters and experimental results on Quranic-343UW dataset are displayed in Table 4.22 and Table 4.23 respectively. From the experimental results, it can be noted, in general, that the proposed algorithm is capable of recognising words with different and enlarged vocabulary (here the vocabulary was enlarged from 252 to 343) confirming its stability.

Table 4.22: Calculated parameters for Mono-font-343UW

Mono-font-343UW	Mask size	Dimension of normalised word-image		Feature size
		Width (Columns)	*Height (Rows)*	
Andalus	8	72	32	288
Tahoma	8	88	32	352
Simplified Arabic	8	64	32	256
Traditional Arabic	10	56	32	280
Deco type Thuluth	5	72	40	225

Table 4.23: Mono-font-343UW-32CB experiment results (WRRs)

Mono-font-343UW-32CB	Top-1 %	Top-2 %	Top-3 %	Top-5 %	Top-7 %	Top-10 %	Top-15 %	Top-20 %	Top-25 %	Top-30 %
Andalus	99.07	99.65	99.77	99.94	99.94	99.94	100.0	100.0	100.0	100.0
Tahoma	98.37	99.30	99.53	99.83	99.88	99.94	99.94	99.94	100.0	100.0
Simplified Arabic	99.30	99.71	99.77	99.88	99.88	99.94	99.94	99.94	99.94	99.94
Traditional Arabic	84.26	89.68	92.36	95.28	96.15	97.08	98.08	98.78	99.01	99.36
Deco type Thuluth	98.83	99.71	99.83	99.83	99.83	99.83	99.88	99.94	99.94	99.94
Average	95.97	97.61	98.25	98.95	99.14	99.35	99.57	99.72	99.78	99.85

These results show that 84.26% for Traditional Arabic font is the lowest percentages of the WRRs. The highest WRRs was 99.30% for the Simplified Arabic font. For the three other fonts, the WRRs were: 99.07% for Andalus, 98.83% for Thuluth, and 98.37% for Tahoma.

Comparing with results achieved using Quranic-252UW database (as shown in Table 4.16), the system shows stability at top-1, with the exception of the Traditional Arabic font. Naturally, the higher top-n performances are much more stable than the lower top-n.

The results of Table 4.24 demonstrate the top-1 WRRs obtained in mono-font approach by using the two versions of the databases, Quranic-252UW and Quranic-343UW. Although the lexicon size was enlarged from 252 to 343, the mono-font system still maintains a high recognition rate when evaluated on larger data sets. Furthermore, the recognition rates obtained (except Traditional Arabic font) are slightly higher than those achieved when using Quranic-252UW database.

Table 4.24: Comparing top-1 WRR% for Mono-font-252UW-32CB and Mono-font-343UW-32CB

Mono-font	Mono-font-252UW-32CB (Top-1)	Mono-font-343UW-32CB (Top-1)
Andalus	98.65	99.07
Tahoma	97.62	98.37
Simplified Arabic	97.94	99.30
Traditional Arabic	96.83	84.26
Deco type Thuluth	96.75	98.83
Average	97.56	95.97

4.7.3.2 Multi-font Classification Stability

The recognition system has the same configuration as the Multi-font system described in Section 4.7.2.2. In normalisation stage, word-images are normalised to 72 x 32 (width x height). Next, feature extraction phase, the feature vector (288 coefficients) for the entire image is then constructed from the first 8-elements coefficients.

The experiments were carried out by using 10 training and 5 test images per word for each font of 343 unique words. The classier was trained on a set of 1,7150 words; 343 (unique words) x 10 (samples) x 5

(fonts). Tests were carried out on samples of 1,715 word images for each font.

Planning for the experiment was done in the same way as the Multi-font classification experiment (subsection 4.7.2.2). Two experiments have been conducted on two different codebook sizes. The first experiment (Multi-font-343UW-32CB) used codebook size of 32, whereas the second experiment (Multi-font-343UW-32CB) used 64.

4.7.3.2.1 Multi-font-343UW-32CB

The results of the experiment are shown in Table 4.25. It is seen that the percentage of top-1 words correctly recognised ranges from 36.03% for Andalus to 55.16% for Simplified Arabic font. The average word recognition rates at top-1, top-10 and top-30 are 48.82%, 83.72% and 94.00%, respectively.

Table 4.25: Multi-font-343UW-32CB experiment results (WRRs)

Multi-font-343UW-32CB	Top-1 %	Top-2 %	Top-3 %	Top-5 %	Top-7 %	Top-10 %	Top-15 %	Top-20 %	Top-25 %	Top-30 %
Andalus	36.03	49.10	56.38	65.36	71.60	76.68	83.27	86.30	88.34	90.73
Tahoma	46.88	59.30	65.42	73.18	77.55	81.87	85.83	88.40	91.02	92.24
Simplified Arabic	54.58	66.76	72.89	79.94	83.67	87.76	91.60	93.64	94.87	95.92
Traditional Arabic	55.16	67.46	73.76	80.35	84.61	88.16	91.72	93.82	95.22	96.03
Deco type Thuluth	51.43	63.27	69.80	76.97	81.11	84.14	88.80	91.43	93.76	95.10
Average	48.82	61.18	67.65	75.16	79.71	83.72	88.24	90.72	92.64	94.00

4.7.3.2.2 Multi-font-343UW-64CB

The results are reported in Table 4.26, which shows that the multi-font system with a 64 codebook outperforms 32 codebook systems.

Average WRRs at top-1 for Multi-font-343UW-64CB is 65.24% compared to 48.82% for Multi-font-343UW-32CB.

Table 4.26: Multi-font-343UW-64CB experiment results (WRRs)

Multi-font-343UW-64CB	Top-1 %	Top-2 %	Top-3 %	Top-5 %	Top-7 %	Top-10 %	Top-15 %	Top-20 %	Top-25 %	Top-30 %
Andalus	57.67	70.20	76.91	83.50	87.93	91.49	94.46	95.86	96.56	97.20
Tahoma	60.58	72.59	78.37	85.25	88.75	91.66	94.58	95.92	96.97	97.67
Simplified Arabic	73.47	83.44	88.10	91.84	94.69	96.15	97.84	98.72	99.07	99.30
Traditional Arabic	64.49	75.04	79.77	86.53	90.61	92.77	95.92	97.43	98.19	98.78
Deco type Thuluth	69.97	79.77	84.90	89.74	91.84	94.46	96.09	97.08	97.73	98.60
Average	65.24	76.21	81.61	87.37	90.76	93.31	95.78	97.00	97.70	98.31

Comparison results of the Multi-font-252UW-64CB and the Multi-font-343UW-64CB experiments are given in Table 4.27. It can be seen that a similar performance was observed in different lexicon sizes, even with enlargement of size to about half (252 to 343 words)

Table 4.27: Comparing WRR% for Multi-font-252UW-64CB and Multi-font-343UW-64CB

Multi-font	Multi-font-252UW-64CB (Top-1)	Multi-font-343UW-64CB (Top-1)	Multi-font-252UW-64CB (Top-2)	Multi-font-343UW-64CB (Top-2)	Multi-font-252UW-64CB (Top-10)	Multi-font-343UW-64CB (Top-10)
Andalus	61.98	57.67	73.25	70.20	93.10	91.49
Tahoma	64.37	60.58	76.03	72.59	93.49	91.66
Simplified Arabic	72.14	73.47	81.90	83.44	95.71	96.15
Traditional Arabic	66.83	64.49	80.24	75.04	95.08	92.77
Deco type Thuluth	67.70	69.97	78.33	79.77	94.13	94.46
Average	66.60	65.24	77.95	76.21	94.30	93.31

4.8 Comparison with other authors

There is not much literature about Arabic printed word recognition system for offline applications. A fair comparison to other recognition systems in the literature is difficult due to the lack of availability of

common databases and the developed systems were tested under different circumstances.

Text recognition process is dependent upon number of factors and the recognition system results are influenced by these factors. These factors include database type (real scanned or synthetic for either machine-printed or screen-based OCR), lexicon size, font size and type, scanned paper quality, text image resolution, and holistic against analytical approach, and other factors.

- It is reported that the RR for the online systems is higher than the offline where temporal information is available [56] [57]. This information is unavailable in text images to be recognised, so it must be simulated. A common method is the sliding window technique; a segmentation free method transforms the 2-D text signal into a 1-D signal. This technique divides the word image into a sequence of windows (frames) and computes a feature vector for each frame as a function of time. Therefore, many researchers prefer to use sliding window method in order to get a better performance.

- The synthetic data has the advantage of not skewed and does not contain noise which is often seen in scanned documents [58], like salt-and-pepper noise and blurring. The scanned images are usually required to reduce noise as much as possible while retaining all of the image signal information. The presence of noise or losing in signal information will cause subsequent errors in recognition [59]; which are not avoidable in noise reduction

processing. The use of synthetic training dataset can lead to improved recognition performance. The generated data may not be as representative as real-world data and the recogniser may bias towards unnatural styles [60].

- In the analytical approaches, such as sliding window technique, each word is segmented into small classifiable element (a sequence of graphemes), these elements could be letters or pseudo-letters. Unfortunately, reliable classification is not always possible, even for machine-printed text. In response to the miss-classifying, the classified elements when assembled, would not lead to a real word having linguistic meaning. In [38] the samples are synthetic and the letter position is not considered in recognition phase, even though the recognition rate drops when word RR is considered rather than character RR.

- Unbalanced distribution of the classifying units, characters or words, affects its recognition accuracy [39]

- The recognition rates of combinations of different feature-types are better than using single feature type [61] [62] [63].

- Some researchers have done works on the impact of font sizes on the system performance [64] [65] [66]. As expected, they found that systems perform better on larger font size.

- Every holistic method uses a lexicon; the size of the lexicon is one of the most important parameters in holistic word recognition. As the number of classes increases, the probability of misclassification

becomes higher [67]. The system described in [34] confirms that as the size of lexicon increases, the recognition rate decreases.

- At the basic level, the text recogniser is capable of recognising isolated characters or words. Some of the approaches find only the top-1 hypothesis, while other approaches find the top-n hypothesis. For character/word recognition, the rate at which the correct character/word appears in the top-n possible alternatives can be high although the accuracy of the top candidate is low. To enhance the accuracy at the top-1, the use of higher level knowledge post-processing is required; in order to re-rank words or rescore word classes appeared in the top-n.

All of these factors provide significant information regards the results that have been reported by the authors in the literature review 2.5.

The used databases in the literature review and this thesis are summarised in Table 4.28.

Table 4.29 and Table 4.30 present a comparative summary of the results obtained by this study and those reported in literature section 2.52.5. All of the methods have their own superiorities and weaknesses. The achieved results show the efficiency of the developed system in this study compared with other rates found by other systems discussed in section 2.5.

Table 4.28: Summary of the used databases in the literature review and this thesis

Reference	Database	Fonts	Font sizes	Font style	data set image resolution	Used dataset size and unit	Training and test set sizes
Khorsheed (2004) [34]	His own, Word (appeared synthetic)	4 fonts	Random sizes ranging from 18 to 48pt	Not specified (appeared plain style)	Not specified	More than 1,700 Words	44%, 56%
Khorsheed (2007) [35]	His own, scanned text line	6 fonts	Not defined, while all line images were resized to 60 pixels	Not specified (appeared plain style)	Not specified	More than 15,000 lines (596,931 characters)	1,500 and 1,000 lines for each font
Al-Muhtaseb et al. (2008) [36]	His own, text line (appeared synthetic)	8 fonts	Not defined, however all text lines are normalised to 80 pixels height	not specified (appeared plain style)	Not specified	2,766 lines (224,109 characters) for each font	2,500 and 266 lines for each font
Slimane et al. (2010) [38]	APTI (screen-based OCR), synthetic word	10 fonts	24pts, while word heights are normalised to 45 pixels	Not specified (appeared plain style)	72 dot/inch	37,765 Words for each font	18,868 and 18,897 words for each font
This thesis	His own (Quranic-252UW), scanned word	5 fonts	14pt and words height are normalised to (32 -40), it is training font dependent	Plain style	300 dot/inch	3,780 words for each font	2,520 and 1,260 words for each font
	His own (Quranic-343UW), scanned word					5,145 words for each font	3,430 and 1,715 words for each font

Table 4.29: Mono-font comparative summary with the literature review

Reference (mono-font)	HMM model	Number of models	HMM model representation	Feature extraction technique	Type of features	Training and test set sizes	Correct recognition metric	Andalus	Tahoma	Simplified Arabic	Traditional Arabic	Thuluth
Khorsheed (2007) [35]	Character, 8 states, (mono-models)	60	Character	Statistical using sliding window	Three different type of features	1,500 and 1,000 lines for each font	Character	88.70	84.10	77.80	78.00	71.50
	Character, 8 states, (context-dependant tri-models)	9,393	Di- and tri-character					92.40	92.10	88.00	89.50	87.60
Al-Muhtaseb et al. (2008) [36]	Character, 5 or 7 states; font dependent	126	Character	Statistical using sliding window	One feature type	2,500 and 266 lines for each font	Character	96.83	99.57	99.55	98.81	98.02
This thesis	Word, 7 states, (252UW-32CB)	252	Word	Global transformation	One feature type	2,520 and 1,260 words for each font	Word	98.65	97.62	97.94	96.83	96.75
	Word, 7 states, (343UW-32CB)	343	Word			3,430 and 1,715 words for each font		99.07	98.37	99.30	84.26	98.83

Table 4.30: Multi-font comparative summary with the literature review

Reference (multi-font)	HMM model	Number of models	HMM model representation	Feature extraction technique	Type of features	Training and test set sizes	Correct recognition metric	Andalus	Tahoma	Simplified Arabic	Traditional Arabic	Thuluth
Khorsheed (2004) [34]	Word, 6 states	145	Word	Global transformation	One feature type	44%, 56%	Word	--	--	90.00	90.00	--
Slimane et al. (2010) [38]	Character, 5 states (Global)	64	Character position group	Various using sliding window	More than 7 different type of features	18,868 and 18,897 words for each font	Character	97.70	98.40	98.00	92.40	82.50
							Word	80.70	82.00	84.50	60.40	35.70
	Character, 5 states, (Cascading)						Character	99.20	98.90	98.30	95.00	91.70
							Word	94.20	85.8	86.50	71.60	52.30
This thesis	Word, 7 states, (252UW-64CB)	252	Word	Global transformation	One feature type	2,520 and 1,260 words for each font	Word	61.98	64.37	72.14	66.83	67.70
	Word, 7 states, (343UW-64CB)	343	Word			3,430 and 1,715 words for each font		57.67	60.58	73.47	64.49	69.97

4.8.1 Mono-font comparison

The performance in Table 4.29 is measured using two different metrics: character and word rate. Even though the other two approaches use sliding window technique and character metric, the recognition rates of the proposed approach are comparable to [36] and even better than [35].

The results show that the proposed system outperformed the Khorsheed [35] system accuracy, even in context-dependant tri-models, by over 6.27%. A more comparable result is that of Muhtaseb et al. [36]. The average word RR obtained by this project, in Mono-font-252UW-32CB experiments, is 97.56% and the average character RR obtained by [36] is 98.56%.

4.8.2 Multi-font comparison

The font-independent word recognition experiments are conducted using the same structure and parameters used for font-dependent word models, its codebook is 64 (experimentally chosen) and features mask size is 8-element calculated by averaging mono-font mask-sizes. The samples and size of training and testing of each font, and vocabulary size are the same as used in the mono-font experiments.

The results indicates that the proposed OCR scheme of this study is robust and stable. In Multi-font-252UW-64CB the accuracies obtained ranged from 61.98% to 72.14% at top-1 and from 93.10% to 95.71% at top-10. And the accuracies obtained in Multi-font-343UW-64CB are

ranged from 57.67% to 73.47% at top-1 and from 91.49% to 96.15% at top-10.

- The samples in Khorsheed [34] were rendered at random sizes ranging between 18pt and 48pt, which are larger than 14pt that used in this study. The highest WRR achieved by the current study is 73.47% for Simplified Arabic is relatively lower compared to the highest WRR (90.00%) attained by [34] for the same font. When the top-10 results are taken into account; the highest score in this thesis was also for Simplified Arabic in Multi-font-343UW-64CB experiments. The experiments are conducted with lexicon size of 343 words, and its top-10 accuracy had 96.15% compared to 98.00%; the highest accuracy at top-10 achieved by the reference [34] using lexicon size of 145 words.

- Slimane et al. [38] presented global and cascading multi-font systems for screen-based Arabic OCR. The global system is font independent system, while the cascading system is font dependent system. The cascading system is composed of two modules: a mono-font recogniser preceded by a font recogniser.

  The systems were evaluated using parts from the APTI database; synthetically-generated smoothed low-resolution word images, suitable for screen-based OCR systems. The train and test sets are word images rendered with font size 24pt in 10 fonts. The sliding window technique (segmentation free) is employed for extracting more than seven different type of features.

Most of these factors are evidences supporting the high character recognition rate, with exception to the complex fonts that demonstrate richer in overlaps, ligatures and flourishes. DecoType Thuluth and Traditional Arabic are two complex fonts. This indicates that the system is not robust and stable for all fonts. According to the authors, obtaining lower recognition accuracy can be attributed to the feature types which are not treating well these fonts.

The achieved results in global system ranged from 52.30% to 94.20%, and in cascading are ranged from 35.70% to 84.50%. Concerning the word-accuracy in font-independent systems; the global system underperforms the multi-font systems proposed in this thesis for the two complex fonts. In addition, the top-10 accuracy in the proposed multi-font systems outperforms the top-1 global system.

Since the cascading system is font recogniser flowed by font-dependent recogniser, and the calculated word accuracy is for correctly recognised font

If the font-dependent word recognition in the cascading system recogniser the correctly recognised fonts, in this case one can compare it with the mono-font systems developed un this study. The results show that developed systems in this study are outperform the cascading system.

The above discussion indicates that the proposed OCR schemes of this study; which returns a set of top-n candidate words with their scores, is very promising.

4.9 Summary and Conclusion

This work presents a new investigation in the holistic system for the offline recognition of machine-printed cursive words. Its feature extraction method employs block-based 2D-DCT. Experimental results showed that HTK/BDCT-based approaches achieve a similar performance to that of HTK/sliding-window and even better with less complexity. A word mono-model is the adopted technique where each HMM model represents a distinct word in the lexicon. Each word represented by features from one type that is DCT coefficients. Size of the feature vectors is font and training set dependent, where it is dependent on word normalised dimension. The extracted features are therefore language independent. The system is designed to produce word alternatives (Top-n), as required, providing a list of candidate words.

To prevent overfitting, validation on the training set is performed using the k-fold cross-validation technique. A stratified k-fold cross-validation (2 x 5-fold cross-validation) was performed (with codebook of size 32) for estimating tests error of the classifier in mono-font approach.

The proposed methodology experiments are carried out by two approaches: mono-font and multi-font approaches, evaluated on the two established real scanned databases, Quranic-252UW and Quranic-343UW. The mono-font experiments have been executed by using 32

and are referred to as “Mono-font-252UW-32CB” and “Mono-font-343UW-32CB”. The multi-font experiments are carried out by 32 and 64 codebook sizes and are referred to as “Multi-font-252UW-32CB”, “Multi -font-343UW-32CB”, “Multi-font-252UW-64CB”, and “Multi -font-343UW-64CB”. The mono-font results were very promising. However, in multi-font approach the 64 codebook results at top-10 were quite satisfactory.

Even though the proposed OCR method of this work exhibited promising results, future works should address important issue like improve the recognition accuracy by exploiting linguistic knowledge at the post-processing stage including, co-occurrence pattern of adjacent words, syntax and semantics.

CHAPTER 5

LINGUISTIC POST-PROCESSING FOR SENTENCE RECOGNITION

5.1 Introduction

Most of Arabic offline handwritten/printed recognition systems deal with single words even when context such as sentence or phrase is available. This is because of lack of infrastructure and supporting utilities that allow experiments on text recognition. Utilities such as electronic language corpora, data sets related to these corpora and language modelling tools supporting Arabic fonts are not prevalent.

Several serious attempts have been made to improve the performance of Arabic text recognisers. The improvement can take place at any combination of the following stages: pre-processing, feature extraction and representation, classifier model and training method, and post-processing procedure. The objective of this work is to investigate how to improve sentence/phrase recognition by applying linguistic knowledge at the post-processing stage. In particular, the linguistic post-processing is to improve reliability of the holistic word-based recogniser to handle larger vocabulary. The use of linguistic knowledge like frequencies of single words or word combinations, syntactic, semantic, and other properties can be incorporated specially when there are

alternative words. The exploitation of this higher-level knowledge is inspired from the human ability to infer and discriminate a word from many other possible words when reading (recognising) a text. Linguistic knowledge factors like word collocations, sentence structures, subject matter, and inference all take place in the human brain during reading [68].

This work uses text from The Holy Qur'an to enable use of Statistical Language Models (SLMs) to improve recognition rate. The Holy Qur'an has been used as the language base as it continues to be a frame of reference for the language. SLMs are conventional n-gram models of order 1 to 5. In this article, the role of the SLMs is to find the best paths through the recognised word lattice to select word combinations that most probably resemble the original sentence/phrase. The most probable is the one which has the lowest perplexity value. The perplexity calculation is a function of n-gram models. This procedure will re-rank the recogniser top-n results according to the perplexity value of each generated text.

The re-ranking is a post-processing technique that attempts to reposition the recogniser's top-n hypothesis output with the lowest word errors near the top of the list. It enhances the recognition accuracy through exploiting rich information/features set (such as linguistic knowledge), and operating at a global level (e.g. sentence). Since the re-ranking technique sees only the chosen top-n candidates, the small number of considered candidates is the determining factor of this methodology's effectiveness. The baseline system output is top-n

candidates with associated confidence scores that define an initial ranking, and then the re-ranking system may include these scores within the re-ranking model to re-rank these candidates.

5.2 Proposed Re-ranking Post-Processing System

This thesis presents a sentence recogniser which uses a lexicon derived from a known corpus. The recogniser classifies a sentence presented as word images in the input and produces a word graph. The candidates in the word graph are ranked according to their decoding scores. Generally, the valid candidate words for each word position in the recognised sentence can be at any rank in the graph. The most promising sentence could be among any combination of the candidates. Figure 5.1 illustrates an example of word lattice produced by a sentence classifier and the possible sentence construction paths. The HTK Toolkit provides the language modelling tools (HLM Toolkit) for building and testing the n-gram SLMs. The HLM Toolkit provides a tool for generating a perplexity metric to evaluate an n-gram LM using test text. The Perplexity evaluation tool calculates the perplexity for each sentence in the test text using n-gram models. The test text in this case will be the generated sentences from the word lattice. After perplexity calculation, the sentences are re-ranked according to their perplexity scores to be able to select top-n sentence candidates. This retrieves the most likely top-n sentences among the candidate sequences.

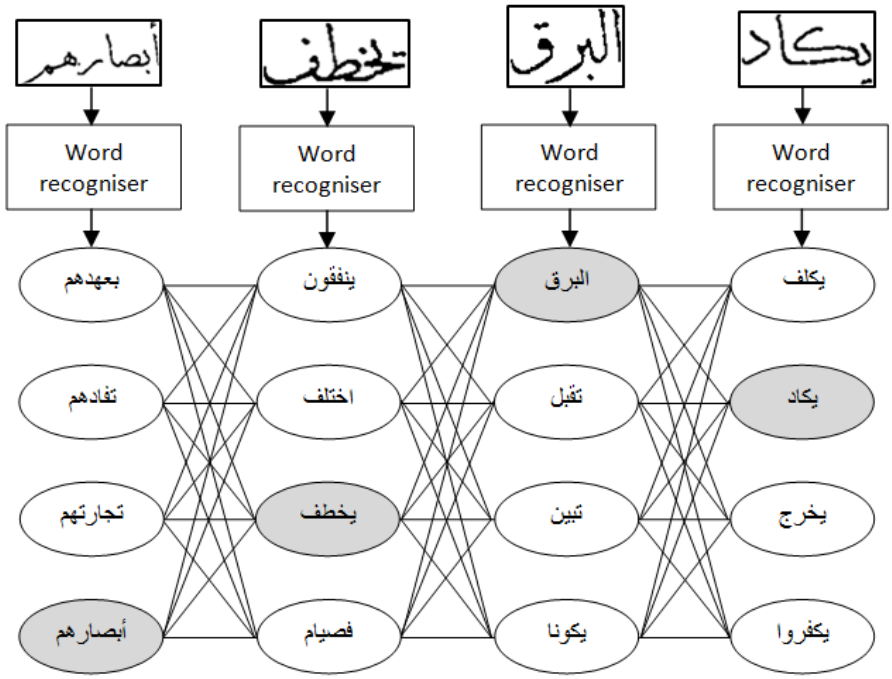


Figure 5.1: The word lattice produced for the sentence "يكاد البرق يخطف أبصارهم"

5.2.1 Word-based n-gram models

Statistical language modelling is crucial for a range of natural language processing applications, including optical character recognition, spelling and grammar checking, machine translation and information retrieval. The goal of a SLM is to create LMs that are able to capture the regularities of various phenomena of a natural language. These include syntax, semantic, morphology, and many more. Among the most important models are n-grams, Decision tree, and Context-free grammar. A brief review of major established SLM techniques can be found in [69]. The n-gram LMs are the most widely used LM as an essential tool in the area of OCR due to its simplicity and effectiveness

[70]. Moreover, there are publicly available toolkits, such as HTK and CMUSphinx[4], providing Language Modelling Toolkit.

Naturally, a word in any natural language is related to its either immediate or distant word neighbours. Therefore, n-gram LMs can be computed with different distances (d = 0, 1, 2, 3 …) on the corpus [71] [72], where $\mathrm{d} = 0$ corresponds to the conventional n-gram model known as the n-gram model. A phrase can be defined as a set of words ordered in a way to express the intended meaning to the sentence, and may not have any syntactic meaning, this is known as semantic. In this work the semantic (word n-gram) approach is used. At the word level, the n-gram LMs estimate the probability of the next word based on the last $\mathrm{n}-1$ preceding words. This can be interpreted as the n-gram modelling predicting the next word ($\mathrm{w_n}$) in a sentence depending on the last $\mathrm{n}-1$ words ($\mathrm{W_1^{n-1}}$). Considering language words, a sentence "S" composed of the words $\mathrm{w_1, w_2, \cdots, w_m}$ or $\mathrm{w_1^m}$, the word-based n-gram model is thus given by [28]:

Equation 5.1: Word-based n-gram model [28]

$$P(w_1, w_2, \ldots, w_m) \cong \prod_{i=1}^{m} P(w_i \mid w_{i-n+1}, w_{i-n+2}, \ldots, w_{i-1})$$

Where the conditional context component of the probability ($\mathrm{w_{i-n+1}, w_{i-n+2}, \ldots, w_{i-1}}$) is called history of word "i", and

[4] http://cmusphinx.sourceforge.net/wiki/

$P(w_i \mid w_{i-n+1}, w_{i-n+2},..., w_{i-1})$ is called maximum likelihood estimate, which can be expressed as follows:

Equation 5.2: maximum likelihood estimate [28]

$$P(w_i \mid w_{i-n+1}, w_{i-n+2},..., w_{i-1}) = \frac{C(w_{i-n+1}, w_{i-n+2},..., w_i)}{C(w_{i-n+1}, w_{i-n+2},..., w_{i-1})}$$

Where $C(.)$ stands for the count of a given word sequence in the training text (corpus).

Equation 5.2 calculates the maximum likelihood estimates for all possible word sequence events encountered in the training corpora and the sum of the probabilities of all events is "1", which means zero probabilities for unseen events. This approach leads to one issue, which is the LM should not assign zero probabilities to any non-existing event. Smoothing is a widely used technique to address this problem caused by un-encountered events. Various smoothing techniques have been developed. These include Back-off or Katz smoothing, Kneser - Ney smoothing and others, see [73] for review of available techniques.

5.2.2 Perplexity

Perplexity is one of the metric techniques for evaluating the quality of a given LM [69] [74], and is considered the most widely used technique. The quality of SLM is evaluated by calculating its perplexity with respect to some testing text. The perplexity can be devised from cross-entropy used in information theory. Given smoothed n-gram LM assigns probability $P(w_1, w_2, \cdots, w_m)$ to a word sequence $W = (w_1, w_2, \cdots, w_m)$ the cross-entropy H of this model on data W is defined as:

Equation 5.3: The cross-entropy H

$$H = -\frac{1}{m}\log_2 P(w_1, w_2, \ldots, w_m)$$

The perplexity, PP, of the test text, W, is related to cross-entropy, H, by the equation:

Equation 5.4: The perplexity PP

$$PP = 2^H$$

The Perplexity is measuring, on average, the number of possible successor words that can follow any given word in a test text with respect to the LM. Normally, the better the LM is the one which yields the lower perplexity. In this work, the better the text phrase, the lower the perplexity.

5.3 Sentence recognition system

In this section, a brief overview of the implemented system for Arabic sentence recognition will be present. The structure of the off-line sentence recognition system based on HMM as shown in Figure 1.4. It extends the holistic word recognition system developed in the previous chapter. The central component of the text recogniser is the word recogniser (WR). The input to the WR is a word signal and a lexicon. The output is a ranked list of the best "n" words in the lexicon which matches the word signal. Generally, the WR output often does not include the correct word as the top-1 choice. The post-processor is used to improve the raw WR output.

5.4 Linguistic post-processing system

After classifying all words within the input sentence the word lattice consisting of ranked top-n choices of valid candidate words for each word position as depicted in Figure 5.1. For the m-word position there are top-n candidates. This gives n^m potential paths through the lattice. These paths are representing the possible sentences with length of m-words. When the process is carried out for each path (using dynamic programme), the generated sentences are directed to the HLM Toolkit to calculate their perplexity. Finally, by using a dynamic programme the sentences are re-ranked in descending order according to their perplexity value and chose the desired top ranked ones.

5.5 Lexicon Dataset and Language Model Creation

Building a Phrase-based OCR consists of creating two major models, SLM and the lexicon driven word model recogniser. Training and testing the recogniser necessitates an underlying lexicon and the SLM requires perplexity calculation. The training and testing data (word images) are generated from the lexicon. The lexicon and LM are to be built from a corpus in plain texts format. As discussed in Sections 3.4 and 3.5, The Holy Qur'an (114-Surah, 6,348-Ayah, and 78,245-Kalimah) forms a robust NLP basis for researchers in Arabic language in general and complementing OCR.

5.5.1 Quranic-2279UW Database

To train the recogniser an even distribution word-image database is necessity. So far, no such database is openly accessible for Arabic language. However, the acquisition of a large amount of data using true scanned word images is labour-intensive, time expensive and error-prone process. Furthermore, it has to be noted that the main focus of this dissertation is how to employ the language context at post-processing stage to improve the recogniser accuracy. Thus, the synthetic database is one choice to evaluate performance. The availability of The Holy Qur'an text electronically eases automatic generation of synthetic data for various types of fonts in different sizes and styles.

A review of literature shows a number of implementations and use of synthetic Arabic database. Maergner and Pechwitz [58] presented an automatic technique to construct a synthetic database for Arabic printed text. The synthetic database used for training and testing OCR system and found it is suitable for the development or evaluation of Arabic OCR systems. In Slimane et al. [38] the synthetically generated database (called ATPI) is used to evaluate the Arabic OCR system. The ATPI database [8] composed synthetic Arabic printed word image. Another synthetically generated database presented in [10] called Multi-modal Arabic corpus (MMAC) suitable for the development of OCR and linguistics systems.

Constructing a synthetic database for recogniser lexicon can be accomplished by degrading the ideal text images. Firstly, a word processor is needed to create text documents automatically in a specified

combination of font, style and size. Secondly, a programme that converts the word processor document to an image file is required. To achieve these two steps a programme was developed: it writes each token to a Microsoft Word document, and subsequently converts this document to an image and saves it in a tiff image file format with 300DPI resolution. To simulate the characteristics of truly scanned document images the synthetic images must be augmented with simulated variations using a degradation model. For this purpose, a MATLAB degradation module is developed; it is composed of two parts: skewing and noising degradation defects, which are most frequent in actually scanned document images. A review of document image degradation modelling and their application in synthetic data generation can be found in [75].

Salt and pepper noise with an appropriate noise density in the interval [0.1, 0.05] is used in this work. Subsequently, images are de noised by using median filtering. These steps are to produce an image with degraded writing stroke only. Image rotation done in the range of ±2 degrees. Figure 5.2 depicts samples from true scanned and synthetic word images printed in Thuluth (Deco Type) font.

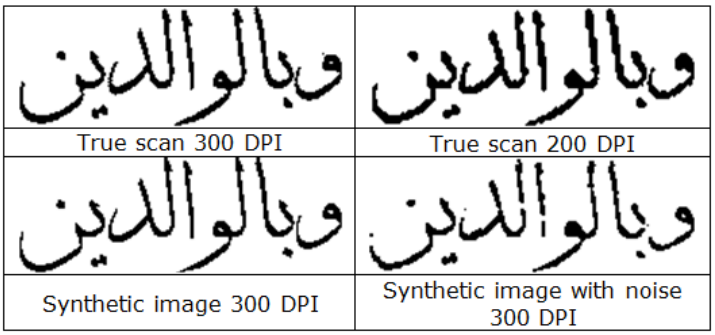


Figure 5.2: Word image sample from true scanned and synthetic word images printed in Thuluth (Deco Type) font

The developed synthetic word-sample database referred to as "Quranic-2279UW" contains the longest Surah in The Holy Qur'an. It is the second Surah in the index comprising 286-Ayah consisting of 6,140-Kalimah. The database lexicon size is 2,279 words; each word has 10 training samples, while the test set has 6,140 samples that are the number of words in the whole Surah. The text is printed in 14 point font size with plain style in five different Arabic fonts: Andalus, Simplified Arabic, Tahoma, Thuluth and Traditional Arabic. The noise and rotation are applied randomly for the first font samples and then for the identical word sample in all other fonts will have the same noise and rotation effects. All word images created in binary level are then stored with a resolution of 300 DPI in a tiff file format.

5.5.2 Language Model Creation

The statistics of the LM is calculated using a training corpus. The first stage in creating a LM is the pre-processing (text conditioning). First of all, the raw text(s) extracted from the source(s) have to be pre-processed by applying tokenization and normalization and sentence segmentation. Text tokenization is the process where the input text is analysed and split into single tokens including words, punctuation marks, and other types of tokens. Text normalization is text transformation into a single canonical form to make the data set more consistent. The normalization includes converting text to uni-case letters and removing diacritics. Moreover, it uses the pronunciation of the words and numbers rather than the orthographic symbols such as converting all numerical data into written form or defined class, for instance, NUM. In sentence

segmentation the input text is split into several sentences generally based on punctuation. The result of the pre-processing stage, tokenised and normalised segmented sentences, along with known n-gram, the statistical calculations will be conducted as follows: count the number of tokens in the corpus, or corpus size, maximum likelihood estimation, n-gram probabilities and smoothing probabilities.

For the purpose of the SLM the n-gram LMs over the word sequences of all the sentences in The Holy Qur'an are constructed. The n-gram modelling is built using the HLM toolkit provided by the HTK toolkit as described in [28]. The HLM toolkit provides generation of n-gram and class-based n-gram models. It is a separate set of tools for training and testing LMs. These tools are designed to run in style of the command-line prompt interface. The corpus text is stored in a file conditioned into a suitable format accepted by the HLM Toolkit. The corpus is tokenized and normalized by removing diacritics and elongation/tatweel character. Each line in the text file represents one sentence and embedded in the two separate tokens <S> and </S> start and end tokens respectively. The sentence here is referring to Ayah as a whole without consideration to the pause symbols within it.

The LMs are trained using n-gram size set to five, which enables generation of 1 to 5-gram LMs. The other main parameters are used in their default settings: n-gram cut-off is "1", minimum occurrence count for unigrams is "1", discounting type is Good-Turing since it supports Absolute discounting as well and "7" for discounting smoothing range for Good-Turing (Katz back-off) discounting. The tool LGPrep generates word

map file to store each new word encountered in the input text along with its allocated new id. It also counts n-grams and stores them in gram file(s). The LBuild command uses the resulting n-gram file(s) and their associated word map to build the LM. For more details on building LM see [28].

5.6 Experiments and results

The fully integrated recognition system, word recogniser integrated with linguistic knowledge post-processing, has been constructed and evaluated.

5.6.1 Word recogniser

The word recogniser was trained and evaluated using synthetic data set "Quranic-2279UW" created in section 5.5.1. The experiments for each font were carried out in the same way as in the Mono-font experiments described in section 4.7. For each font, the training set includes 22,790 word samples, while the test set includes 6,140 words. The testing samples are not included in the training sets. The word recogniser setup are as follows: HMM models are seven states Bakis model; codebook size is set to 32, non-overlapping sub-blocks having a size of 8x8, and the mask size is "5". Table 5.1 summarizes the top-10 recognition results before post-processing. The obtained result boundaries at top-1 are 73.97% and 58.49% for the Thuluth and Traditional Arabic font respectively.

Table 5.1: Accuracy of the Word Recognition Systems without Post-Processing

Font	Top-1 %	Top-2 %	Top-5 %	Top-10 %
Andalus	59.12	66.61	74.77	80.23
Simplified Arabic	70.75	77.96	84.90	88.71
Tahoma	73.86	81.69	88.24	91.76
Thuluth	73.97	81.12	88.31	91.87
Traditional Arabic	58.49	66.38	73.68	79.59
Average	67.24	74.75	81.98	86.43

5.6.2 Linguistic Post-processing Evaluation

For each font, before applying the post-processing the OCR output is divided into a series of phrases having a length of n-word where "n" is referring to the selected n-gram module. These phrases are created from the recogniser hypothesis outputs as described in section 5.4 and then they are evaluated in terms of perplexity based on the selected n-gram model. Phrase perplexity is calculated based on 1 to 5-gram LMs. Finally, the top-n phrases having the lowest perplexity are chosen. Here "n" is defined to be 10, the most frequently used by the researchers.

Table 5.2 through Table 5.8 show the accuracies of the recogniser on the test set before and after the re-ranking process using n-grams. Figure 5.3 to Figure 5.7 provide a comparison of the word recogniser versus the post-processor output. Figure 5.8 and Figure 5.9 illustrate the top-10 recognition accuracies after post-processing. Figure 5.8 illustrates the improvement (from top-1 to top-10) to top-1, while Figure 5.9 illustrates the worsening to out of the top-10 list.

The experimental results reveal a number of conclusions:

- The higher-order n-grams in general improve recognition rates, especially at the top-1 and when moving beyond bigram models. This tendency describes that the larger n-grams means a longer phrase that captures the wider context. This is to be expected and it is consistent with the conclusion reached by [71]
- The 1-gram and 2-gram are starting with worsening rather than improving the recognition rates, especially at the top-1 level
- The 1-gram result starts with a weak performance. Improvements tend to converge with the recogniser performance at the best 10 choice. Moreover, it has no worsening to out of the top-10 list
- There is strong correlation between the accuracy and the perplexity values. The higher OCR recognition rates the better the improvement. For a given phrase, the more correct words in its correct position, the lower the perplexity value. In turn the phrase with a lower perplexity value will have a higher probability to be at the top-1 choice
- Worsening to out of the top-10 list is too low, which means the LMs are promoting the alternatives to higher ranks rather than excluding them from the top-n list
- The rate of improvement of n-gram models beyond the bigram is very slow as they are moving towards larger top-n. Clearly, as the n-gram post-processing converges to top-10 its improvement rate reduces
- Beyond bigram model improvements are too close and getting tighter as n-gram order increases. This is also the case as they approach top-n. The improvement of the 4-gram model is a little lower than 5-gram model, and their improvement getting closer as they approach top-10

Table 5.2: Andalus Font Recognition Accuracy with Post-Processing

Andalus	OCR	1-gram	2-gram	3-gram	4-gram	5-gram
Top-1 %	59.12	30.86	46.99	63.55	67.93	69.45
Top-2 %	66.61	44.25	58.50	67.17	69.95	71.21
Top-3 %	70.46	52.51	64.23	69.17	71.25	72.44
Top-4 %	72.85	58.94	67.57	70.46	71.99	73.14
Top-5 %	74.77	64.01	70.08	71.32	72.64	73.84
Top-6 %	76.25	69.14	71.60	71.84	73.24	74.23
Top-7 %	77.70	73.45	73.11	72.48	73.68	74.46
Top-8 %	78.83	76.99	73.73	72.96	74.01	74.80
Top-9 %	79.53	79.14	74.71	73.37	74.30	75.07
Top-10 %	80.23	80.23	75.36	73.66	74.67	75.20
Out of the list %	19.77	19.77	24.64	26.34	25.33	24.80

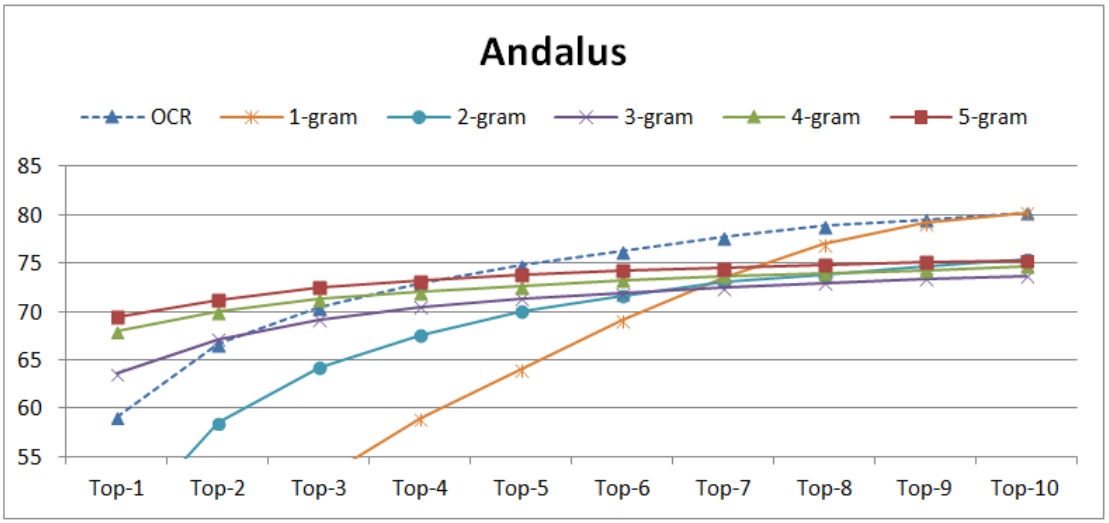


Figure 5.3: Andalus font recognition accuracy before and after the post-processing

Table 5.3: Simplified Arabic Font Recognition Accuracy with Post-Processing

Simplified Arabic	OCR	1-gram	2-gram	3-gram	4-gram	5-gram
Top-1 %	70.75	33.32	55.75	77.20	80.26	81.61
Top-2 %	77.96	47.87	68.75	80.99	82.07	83.01
Top-3 %	81.24	57.85	75.59	82.39	82.88	83.53
Top-4 %	83.53	65.15	79.09	82.93	83.37	84.09
Top-5 %	84.90	70.86	81.40	83.68	83.84	84.43
Top-6 %	86.09	76.47	82.98	84.06	84.15	84.71
Top-7 %	86.99	81.68	84.06	84.32	84.50	85.00
Top-8 %	87.75	85.49	84.53	84.66	84.79	85.21
Top-9 %	88.19	87.77	84.97	84.80	84.92	85.34
Top-10 %	88.71	88.71	85.60	84.97	85.02	85.52
Out of the list %	11.29	11.29	14.40	15.03	14.98	14.48

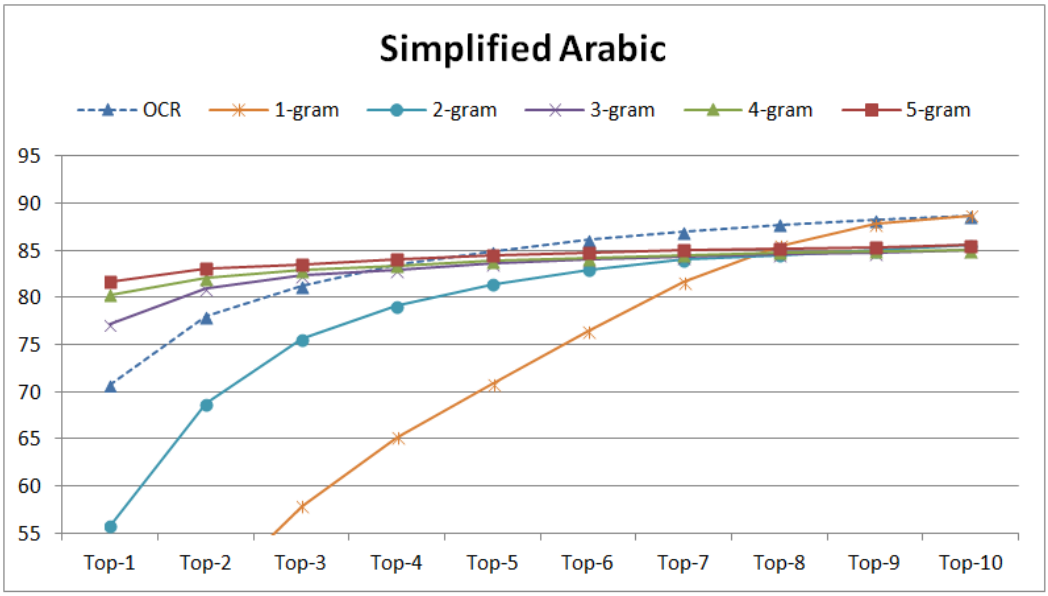


Figure 5.4: Simplified Arabic font recognition accuracy before and after the post-processing

Table 5.4: Tahoma Font Recognition Accuracy with Post-Processing

Tahoma	OCR	1-gram	2-gram	3-gram	4-gram	5-gram
Top-1 %	73.86	34.90	59.53	80.94	84.64	86.79
Top-2 %	81.69	50.33	72.75	84.69	86.87	88.14
Top-3 %	84.61	59.69	79.14	86.22	87.61	88.75
Top-4 %	87.02	67.38	83.00	87.17	87.87	88.97
Top-5 %	88.24	73.39	85.59	87.67	88.16	89.20
Top-6 %	89.23	78.99	86.91	87.96	88.39	89.38
Top-7 %	90.05	84.12	87.90	88.34	88.60	89.54
Top-8 %	90.77	88.40	88.57	88.58	88.70	89.67
Top-9 %	91.19	90.78	89.12	88.76	88.81	89.80
Top-10 %	91.76	91.76	89.53	88.89	88.96	89.89
Out of the list %	8.24	8.24	10.47	11.11	11.04	10.11

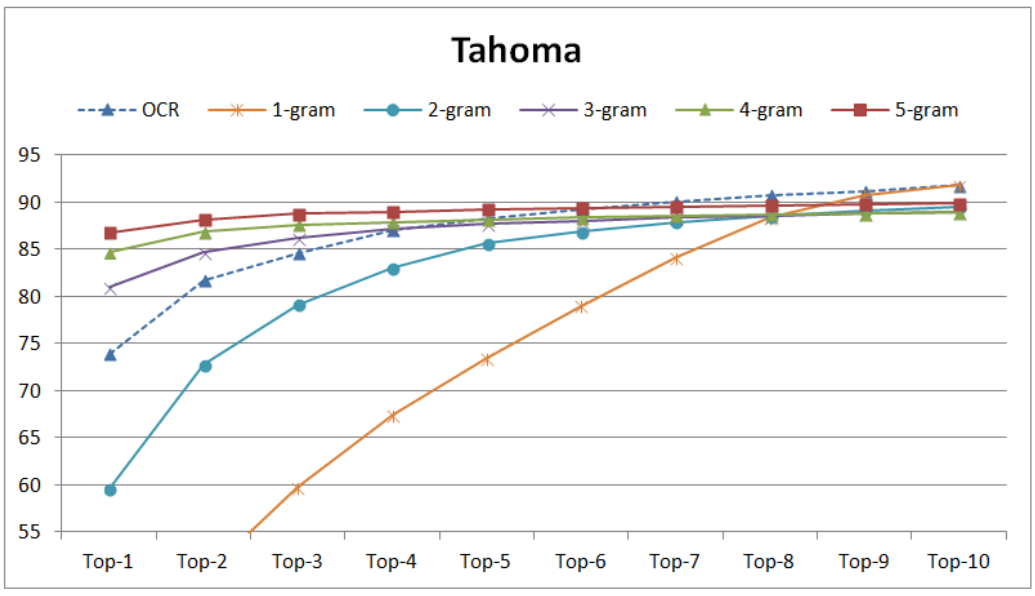


Figure 5.5: Tahoma font recognition accuracy before and after the post-processing

Table 5.5: Thuluth Font Recognition Accuracy with Post-Processing

Thuluth	OCR	1-gram	2-gram	3-gram	4-gram	5-gram
Top-1 %	73.97	38.42	59.07	82.20	86.37	86.40
Top-2 %	81.12	51.79	72.79	85.03	87.83	87.88
Top-3 %	84.66	60.10	80.55	86.48	88.40	88.55
Top-4 %	86.73	67.12	83.50	87.49	88.83	88.96
Top-5 %	88.31	73.70	85.91	87.85	89.01	89.30
Top-6 %	89.32	79.12	87.10	88.13	89.25	89.51
Top-7 %	90.24	84.41	88.08	88.42	89.40	89.64
Top-8 %	90.91	88.47	88.57	88.60	89.53	89.87
Top-9 %	91.38	90.81	89.09	88.78	89.64	90.00
Top-10 %	91.87	91.87	89.54	88.94	89.72	90.03
Out of the list %	8.13	8.13	10.46	11.06	10.28	9.97

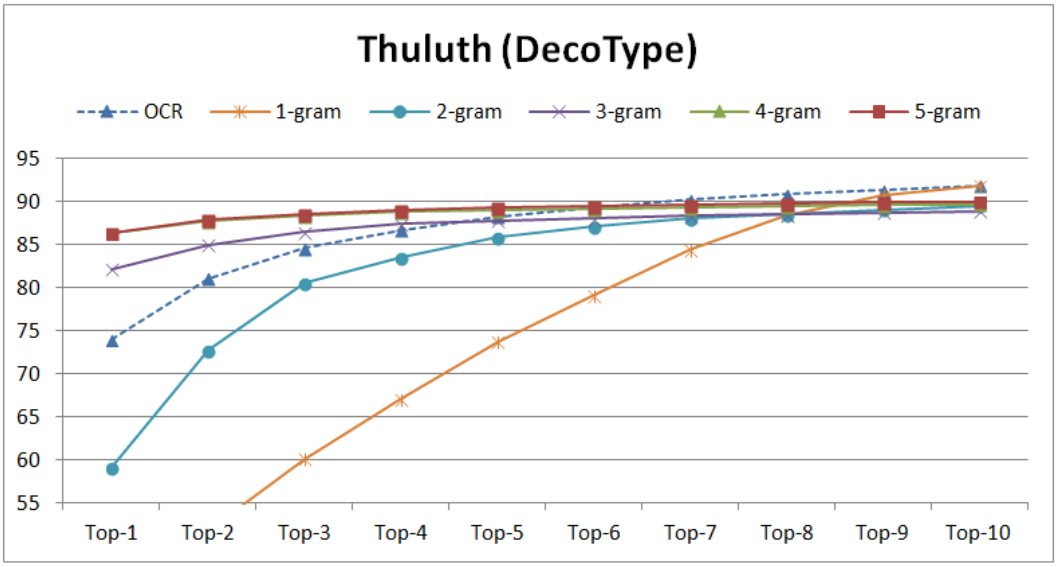


Figure 5.6: Thuluth font recognition accuracy before and after the post-processing

Table 5.6: Traditional Arabic Font Recognition Accuracy with Post-Processing

Traditional Arabic	OCR	1-gram	2-gram	3-gram	4-gram	5-gram
Top-1 %	58.49	29.09	44.43	62.10	65.26	67.52
Top-2 %	66.38	43.14	55.83	66.35	68.06	69.53
Top-3 %	69.69	51.56	63.47	68.99	69.59	70.83
Top-4 %	71.74	58.00	67.31	70.42	70.41	71.64
Top-5 %	73.68	63.52	69.66	71.45	71.03	72.17
Top-6 %	75.37	68.52	70.94	72.18	71.61	72.56
Top-7 %	76.84	73.08	72.46	72.83	71.97	72.92
Top-8 %	77.93	76.63	73.44	73.26	72.33	73.21
Top-9 %	78.91	78.84	74.12	73.73	72.87	73.53
Top-10 %	79.59	79.59	74.67	73.86	73.06	73.73
Out of the list %	20.41	20.41	25.33	26.14	26.94	26.27

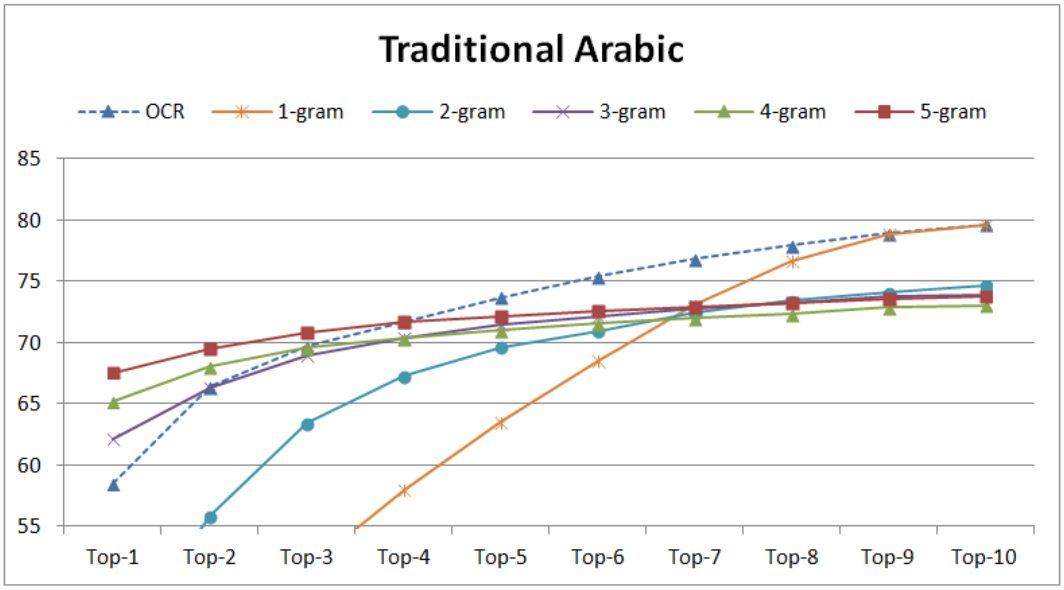


Figure 5.7: Traditional Arabic font recognition accuracy before and after the post-processing

Table 5.7: Font Recognition Accuracy (from Top-1 to Top-10)

Improvement to Top-1 after the Post-Processing

Font	1-gram %	2-gram %	3-gram %	4-gram %	5-gram %
Andalus	38.47	58.57	79.21	84.67	86.56
Simplified Arabic	37.56	62.84	87.02	90.47	92.00
Tahoma	38.04	64.87	88.21	92.24	94.59
Thuluth	41.82	64.30	89.47	94.01	94.04
Traditional Arabic	36.55	55.82	78.02	81.99	84.84

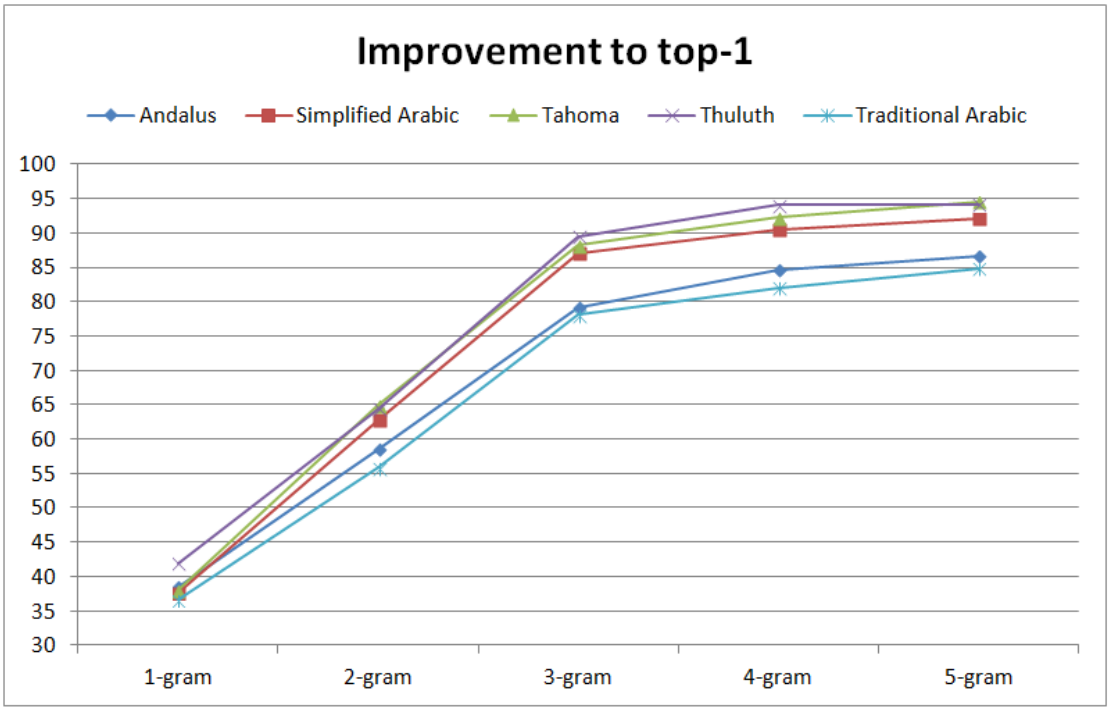


Figure 5.8: Fonts recognition accuracy improvement to top-1 after the post-processing

Table 5.8: Fonts Recognition Accuracy Worsening to Out of the Top-10 List after the Post-Processing

Font	OCR top-1 %	1-gram %	2-gram %	3-gram %	4-gram %	5-gram %
Andalus	59.12	0.00	6.07	8.18	6.92	6.27
Simplified Arabic	70.75	0.00	3.51	4.22	4.17	3.60
Tahoma	73.86	0.00	2.43	3.12	3.05	2.04
Thuluth	73.97	0.00	2.54	3.19	2.34	2.00
Traditional Arabic	58.49	0.00	6.18	7.20	8.21	7.37

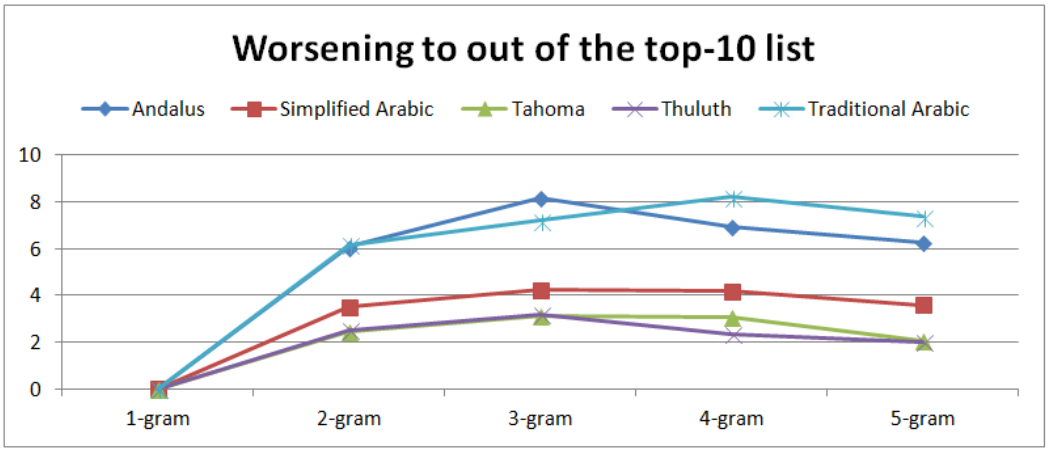


Figure 5.9: Font recognition accuracy worsening to out of the top-10 list after the post-processing

5.6.3 Linguistic post-processing Complexity

The linguistic post-processing execution time mainly depends on three factors: the complexity of constructing a word graph, the complexity of perplexity calculation and the complexity of phrase sorting. The

experimental results show that the post-processing improves the OCR at top-1 only when using 3-gram and above.

Table 5.9 presents a comparison between improved accuracies at top-1 choice. Table 5.10 shows the characteristics of the linguistic post-processing using OCR top-10; n-gram average accuracies and complexities. Table 5.11 demonstrates top-1 accuracy improvement between the selected n-grams. Table 5.12 shows the trade-off between complexity and accuracy for the selected n-grams.

Several observations can be made from the analysis of the results. First, the 5-gram speed is far slower than the others, on the other hand; its average accuracy is higher. Second, the 3-gram is faster than 4-gram and 5-gram by 4.59 and 32.86 times respectively, and 4-gram is faster than 5-gram by 7.17 times. Third, the 3-gram improvement compared with 4-gram and 5-gram is lower by 3.69% and 5.15% respectively, while the 5-gram is higher than 4-gram by 1.46%. Accordingly as a trade-off between complexity and accuracy the 4-gram word model is more suitable for The Holy Qur'an. This should not be surprising since the four-word sentence is the highest sentence length frequency distribution in The Holy Qur'an as presented in Figure 5.10.

Table 5.9: Top-1 Choice Accuracies Improvement Using N-Gram

Font	OCR top-1 accuracy %	n-gram top-1 accuracy %				
		1-gram	2-gram	3-gram	4-gram	5-gram
Andalus	59.12	30.86	46.99	63.55	67.93	69.45
Simplified Arabic	70.75	33.32	55.75	77.20	80.26	81.61
Tahoma	73.86	34.90	59.53	80.94	84.64	86.79
Thuluth	73.97	38.42	59.07	82.20	86.37	86.40
Traditional Arabic	58.49	29.09	44.43	62.10	65.26	67.52
Average	67.24	33.32	53.15	73.20	76.89	78.35

Table 5.10: n-gram Top-1 Average Accuracies and Complexities for All Fonts Using OCR Top-10 Outputs

n-gram	linguistic post-processing using OCR top-10		
	n-gram top-1 average accuracy %	Constructed phrases	Average execution time (hh:mm:ss)
3-gram	73.20	1000	00:23:13
4-gram	76.89	10000	01:46:27
5-gram	78.35	100000	12:42:54

Table 5.11: Top-1 Accuracy Improvement between the selected n-grams

Top-1 accuracy improvement	Minimum accuracy different %	Maximum accuracy different %	Average accuracy improvement %
4-gram over 3-gram	3.06	4.38	3.69
5-gram over 3-gram	4.20	5.9	5.16
5-gram over 4-gram	0.03	2.26	1.46

Table 5.12: Accuracy and Complexity Comparison between the selected n-grams

	Average time difference (n-m) (hh:mm:ss)		Average time speeding factor (m/n)		n-gram average accuracy difference % (n-m)	
m-gram / n-gram	3-gram	4-gram	4-gram	5-gram	3-gram	4-gram
3-gram			4.59	32.86		
4-gram	01:23:13			7.17	3.69	
5-gram	12:19:41	10:56:28			5.15	1.46

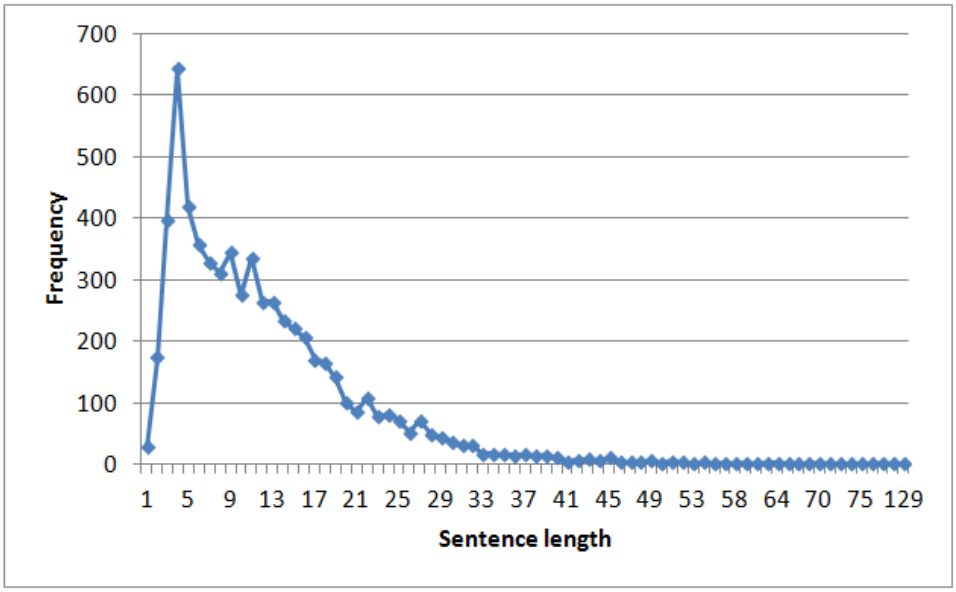


Figure 5.10: Sentence length frequency distribution in The Holy Qur'an

5.7 Comparison with Literature Review

The improvements achieved in this work are comparable to those reported in the literature review. It is very difficult to draw a direct comparison as these attempts are tested under varying conditions. Table 5.13 shows a comparative summary noting the different specific criteria reported.

Table 5.13: Comparison Summary between Works Reported in Literature Review

Reference	AlKhateeb et al. (2011) [39]	Prasad et al. (2008) [40]	Devlin et al. (2012) [41]		Al-Hajj et al. (2009) [42]	Farah et al. (2006) [43]		Current study
Re-ranking Technique	Structure-like features	word trigram	Word 5-gram	SMT-LM in combination with word 5-gram LM	fusion of three homogeneous HMM-based classifiers	parallel combination of three classifiers	Parallel combination of three classifiers followed by syntactical analyser	Word 5-gram and Perplexity
Rescoring Function	Confidence score and structure-like features	word trigram	Sentence Perplexity	SMT-LM Features	multilayer perceptron ANN decision maker	score summation statistical decision	Grammar base syntactical analyser	Sentence Perplexity
Recogniser testing data set	IFN/ENIT database v1.0p2 (town/village names)	newswire	Transcribed text		IFN/ENIT database (town/village names)	Self-constructed Arabic literal amounts database		Second Surah in The Holy Qur'an

Post-processing data set	IFN/ENIT database v1.0p2 (town/village names)	newswire	Arabic sets used to train OCR and SMT systems & Arabic GigaWord corpus	English GigaWord corpus in addition to 5-gram LM training set	IFN/ENIT database (town/village names)	None	Grammar describing the Arabic legal amounts	The Holy Qur'an
Top-n used in re-ranking	Top-10	not specified	Top-20		Top-10	Top-3	Top-9	Top-10
Top-1 Improvement	2.51%	not specified	0.67% WER	1.07% WER	3.36%	1.84%	3.84%	11.11%
Training set (Words)	19,724	More than 2.6 million	547.3 million	5 billion (SMT) + 547.3 million (LM)	19,724	- -	- -	78,245
Test set (Words)	6,735	not specified	22,261		6,735	- -	- -	6,140

Reference [41] is the most relevant work where the word 5-gram is used. Its LM is trained on around 550 million words and tested on 22,261 words. These figures are incomparable with this work where 78,245 training words and 6,140 testing words with vocabulary size of 14,870 words were used. The Current study outperformed the work reporting in [41]. This is due to two factors: vocabulary size and including the testing set in the LM training set (the LM is trained only using the in-domain corpus). The perplexity decreases if the vocabulary size is decreases [71] and when the LM training set includes the testing set [76]. Furthermore, the perplexity decreases when the size of LM training corpus and/or the number of recognised candidates increase [77]. For recognised scripts, increasing the number of candidates leads to a higher chance to capture

correct candidates in the top-n hypothesis list. However, training the LM using out-of-domain in addition to the in-domain training data set (Within-domain) may not improve the performance on an in-domain test set; since the characteristics (topic, style and other peculiarities) of the out-of-domain data do not match the in-domain corpus [78] this strategy may select irrelevant sentences.

5.8 Summary and Conclusion

The main contribution of this thesis is to investigate the improvement that can be achieved through language post-processing in Arabic OCR. The Data for the system is collected from The Holy Qur'an. The word recogniser is built on HTK Toolkit, and the developed database "Quranic-2279UW" is synthetic machine printed added artificial noise (salt and pepper noise, and rotation). The Quranic-2279UW database comprises word samples printed in five different Arabic fonts with font sizes of 14 points in a plain style. The baseline classifier been evaluated on 6,140 words making up 286 sentences.

The linguistic post-processing is based on n-gram SLM, and text goodness is measured by the perplexity. The n-gram training and perplexity calculations are completed using the HLM toolkit. The n-gram models are trained using the whole text of The Holy Qur'an. The n-gram models of order 1 to 5 are generated. The system is evaluated, and the average accuracy achieved at top-1 by the word recogniser is 67.24%. For a given n-gram, the linguistic post-processing is conducted on phrases consisting at maximum n words. The highest average accuracy improvement at top-1 is 78.35% by using 5-gram. As a compromise

between complexity and accuracy, it is found that 4-gram model is more suitable with average accuracy 76.89%.

The results of the post-processing experiments suggest several conclusions. First, the unigram and bigram models beginning with a significant decrease in the word accuracy rate, especially at top-1 while the others are improving it. This shows that, as n increases the word history becomes longer and words are presented with more consistent text context with lower perplexity than when presented in isolation. Second, the higher the recognition rate, the higher the improvement that can be attained, which confirms that the perplexity correlates with the accuracy rate. This explains that the conventional n-gram LM is sensitive to the content of the candidate list and required that the candidate list include more possible correct words. Third, the complexity of post-processing rapidly rises with increasing "n", as larger "n" means longer sentence, which leads to creating more sentence alternatives.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Introduction

The thesis describes method to holistic word recognising and use contextual information to improve performance of OCR system. The system inspired by the principle of human reading. According to psychological studies of reading, the humans treat the word as an indivisible entity and tend to read whole words at a time utilizing its overall shape. When people read, if the word occurs in a sentence, they use their knowledge of language (context). This knowledge can be used to identify a feasible word when he fails to predict degraded, missed or miss spelled words.

Given a sentence of Arabic text that has been segmented into words, a word recogniser generates a top-n candidate list for each word image; next, the top-n word lattice passed to a dynamic procedure to build all possible sentences with defined length; finally, using standard word n-gram SLMs, a phrase perplexity score calculation procedure re-ranks the generated sentences, and the most probable (top ranked) is the one which has the lowest perplexity value.

6.2 Conclusion

The problem of algorithmically transforming images of machine-printed Arabic phrases into machine-editable text was considered. A holistic word recognition using discrete HMM for machine-printed Arabic text has been implemented. It was assumed that words can be accurately detected and segmented and the recogniser receives as input an image of a single word. The classifier developed for images scanned in binary-scale with the resolution of 300 dpi.

In this project, the discrete HMM classifier and the linguistic post-processing were built on Hidden Markov Models Toolkit (HTK), which was originally developed for speech recognition. The HTK Toolkit is used for training and recognition, and the n-gram training and perplexity calculations are completed using the HLM Toolkit.

Since the unavailability of dataset related to a corpus that fits to this specific task, it is decided to build word-image Databases and SLMs that support this work. Although more data usually improve statistical systems, irrelevant additional data to testing dataset can decrease the performance. For this reason, it decided to choose a closed Arabic corpus namely The Holy Qur'an as a reference for databases and SLMs implementation.

Words are modelled as a whole without segmenting them into smaller units. A feature vector is extracted directly from a normalised image. This feature vector comprises global transformation (Block-based DCT) features. These features are presented to discrete HMM-based holistic

approaches classifier. The HMM-based classifier produces a word lattice with top-n ranked list. The proposed method is evaluated in experiments using offline machine-printed word database prepared by the author. The results for mono- and multi-font recognition achieved in this research compare favourably with other researchers. The comparison results are summarized in Tables Table 4.29 and Table 4.30.

The higher level information at the post-processing stage is responsible for selecting the right solution by using linguistic knowledge (standard n-gram SLMs) to re-rank the outputs. The candidates are re-ranked through exploiting phrase perplexity score. The results appear in Table 5.9 show that the method is effective in improving the accuracy.

The one type feature extraction technique (Block-based DCT) and the linguistic post-processing scheme (n-gram) make the developed system language independent OCR system and is not predisposed to a particular corpus. Moreover, the integrated post-processing module which exploits the contextual information is valid to any probabilistic baseline systems, which are naturally characterised as ranking tasks. These systems are including Automatic Speech Recognition, Statistical Machine Translation and Information Retrieval.

6.2.1 Databases

To validate/train the word recogniser a balanced databases of word-image has been constructed to ensure an even distribution of word samples. The same is true for test sets. In all cases there is no overlapping between the training and testing datasets.

A database collection form has been designed in a way that able to extract words based on pixel location. The collection form was prepared in Microsoft word document and can be used for machine printed and handwritten samples as well. A program code is used to fill collection form automatically. Also, image pre-processing and word segmentation are completed automatically using MATLAB routines.

The Arabic words appeared in the databases are typewritten in five fonts (Andalus, Simplified Arabic, Tahoma, Deco Type Thuluth and Traditional Arabic) having a size of 14 points in a plain style. The 14 pts is within the range from 10 to 16 pts that are commonly used in printed Arabic documents such as magazines [6]. The binary-level word images are saved in a tiff image file format with 300DPI resolution.

In this dissertation, two database types have been constructed; truly scanned and synthetic databases.

6.2.1.1 Truly scanned databases

For real scanned word images two real scanned databases, Quranic-252UW and Quranic-343UW, are established. The Quranic-252UW database comprises 31,500 samples representing 252 words, it is implemented for validating and testing the baseline system. The Quranic-343UW database contains 343 unique words in 25,725 images samples. The real scanned documents are the result of printing the filled forms on white plain papers, and then the printed papers are scanned in binary level.

6.2.1.2 Synthetic databases

Due to requirements for large amount of data for a reliable training and testing the sentence recognition systems, the artificial database for sentence word-images "Quranic-2279UW" is founded. It contains 28,930 word with a 2,279 word vocabulary. To synthetic word-images; a Microsoft Visual Basic module is built to convert the Microsoft word document to an image file.

The artificial data is simulating low resolution true scanning in two affects, noise and rotations, where their values are applied randomly. The noise is salt and pepper in the interval [0.1, 0.05] affecting the stroke only and the rotation in the range of ±2 degrees.

6.2.2 SLMs

In addition to word-image Databases, the n-gram models of order 1 to 5 are trained on The Holy Qur'an. The training took place after applying tokenization and normalization, and sentence segmentation on the corpus text. To enable the efficient use of the language models, smoothing technique for unseen n-grams, and count cut-offs are employed.

6.2.3 HMM/DCT Holistic Word Recogniser

A review of the published research confirms that recognition of printed Arabic word continues to present challenges. This is specially the case when segmentation is problematic. A word level recognition system is presented here that does not rely on any segmentation or require

baseline detection of ascenders and descenders. A Discrete Hidden Markov classifier along with a block-based Discrete Cosine Transform (DCT) is used to construct a novel holistic Arabic printed word recogniser. Each word in the lexicon is represented by a separate left-to-right HMM.

The system is applied on actual scanned word images with no overlap between the training and testing datasets. Word feature vectors are extracted using block-based DCT after applying the zigzag masking on each block and selecting the first "r" coefficients from each block. The sizes of the feature vector are font and training set dependent

A Hidden Markov Models Toolkit (HTK) is used to construct the recogniser. Since the HTK Toolkit does not support Arabic texts, therefore, a mapping scheme is developed to hold mapping between the Arabic words and their Latin representations.

Vector Quantisation is used to map each feature vector to the closest symbol in the codebook. The recognition of the unknown word is a word in the lexicon with the highest probability in the recognition result. The output of the system is multiple recognition hypotheses (N-best word lattice). The results for mono- and multi-fonts are encouraging when compared with other published research presented in literature section 2.5. A detailed comparison and analysis of the results were presented in chapter 4.

The classifier achieved WRRs in top-1 position on average of 97.56% and 66.60% for mono-font and multi-font models respectively. Naturally,

the Top-10 performances are much more stable and the average rising to 99.70% and 94.30% for mono- and multi-font, respectively.

6.2.4 Integrated Linguistic Post-processing

The post-processing stage is an important phase of any recognition system. It is enhancing the recognition accuracy by exploiting the rich context information which is not computable in the recogniser.

The aim is to improve the recognition results by re-ranking word candidates produced by a word level recogniser. It introduces a new method for improving Arabic sentence/phrase recognition by re-ranking its outputs through exploiting perplexity which is based on an n-gram statistical language model.

In fact, the post-processing task here is re-ranking not error detecting and correcting. The performance of the baseline system is reported and then compared against the accuracy with re-ranking post-processing using top-10 choices.

6.2.5 Sentence Recognition System

The sentence recognition system comprises word recognition and linguistic post-processing. The integrated system, word recogniser and linguistic post-processing, is based on the HTK Toolkit. The n-gram models of order 1 to 5 are created as well as a synthetic database for the lexicon. The synthetic database is used for training and recognition purposes by the HMM word recogniser. The recognition test evaluated using 286 sentences (Ayat) located in 6140 words.

For each font, before applying the n-gram post-processing on the top-m candidates of the baseline system output, phrases of n-words length assembled from m^n paths in the word lattice (each path represents a phrase). The length of phrases are n-gram dependent (1 to 5-gram). Considering the top-10 candidates received from the word recogniser, the linguistic post-processing calculates phrase perplexity using one of the n-gram LMs. The resulted phrased are re-ranked according to the perplexity values with the top-1 having the lowest perplexity.

The results are encouraging and confirm that lower perplexity correlates with higher accuracy. The achieved average accuracy at top-1 by the recogniser is 67.24% improved to 78.35% on average when using 5-gram post-processing. The complexity and accuracy of the system are evaluated and found that 4-gram is more suitable than 5-gram; it is much faster by average time speeding factor of 7.17 times with an average improvement of 76.89%.

6.3 Limitations

Some of the limitations of the research and evaluations presented in this dissertation are the following:

- The main reasons for baseline system errors are due to the nature of the DCT drawbacks. Although the DCT bypassed the segmentation stage, it is sensitive to the rotation as well as to the translation (shift) in the image [79]

- In database implementation; the segmentation technique of text section, lines, and words are exploiting vertices pixel

coordinates which necessitates writing each word in predefined position. The translation problem may arises and make the segmentation imperfect. This problem could be as a result of shifting the document image vertically or horizontally, or from the presence of any random noise at word tidy segmentation as shown in Figure 6.1

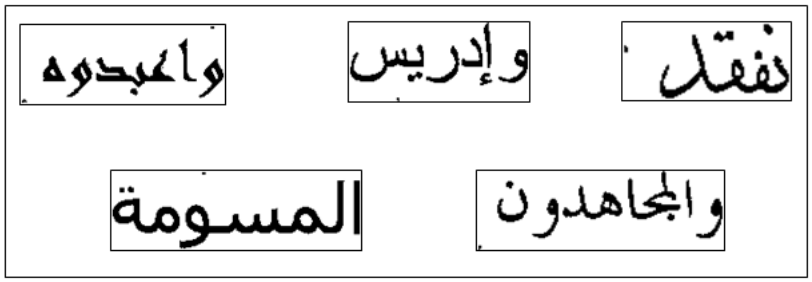


Figure 6.1: The translation problem due to word segmentation error

- During image acquiring document image rotation may occurred; it is when the text lines are not parallel to the true horizontal axis

- Other pre-processing operations on acquired image, such as filtration and morphological operation, may affect the topology of the object

- The word-based n-gram LM require rebuilding the LM when adding new words. Moreover, the number of possible n-grams increases exponentially with the size of the word vocabulary

- The order "n" is a key factor in n-gram language modelling limited by the HLM toolkit provided by the HTK toolkit

- According to the perplexity correlation with accuracy rate, the higher recognition rate is vital to achieve lower perplexity. This is showing that perplexity depending on LM and test text. For given LM, the perplexity affected by smoothing method, count cut-off, and the order of the n-gram. For test text, the candidate set size and the text length are influencing the perplexity and processing speed. Even though the linguistic post-processing improves the baseline system, it is worsening the candidates to lower ranks even to out of the list. This reflects the behaviour of the perplexity; lower perplexity does not necessarily lead to lesser error rates

- Arabic is highly inflected language as discussed in section 4.3.1. Words are derived from a root and pattern, combined with prefixes, suffixes and circumfixes. Different combinations of inflectional affixes, creating as many unique word forms. In the present work, words are simply taken without removing of its inflectional; which increase the rate of vocabulary growth. Furthermore, the vocabulary size will be increased by considering the presence of vowel diacritics and elongations in the words. Vocabulary growth problem is the main drawback of the holistic approach where its time complexity increases and a correct classification becomes difficult

- Finally, the classifier's performance has been evaluated, using only one font size (14 points) in a plain style for the five used

Arabic fonts. The words were printed using laser printers at high resolutions, at least 600 DPI, and scanned in binary-scale at 300 DPI resolutions

6.4 Thesis Contributions

The main contributions that are presented in this thesis are:

- Successfully employing holistic approach for word recognition and applying passage-level post-processing for context-dependent word correction. This show that the observation of the cognitive studies stated in section 1.4 is applicable

- The developed system is not predisposed to a particular corpus or language; since feature extraction technique is global transformation (Block based DCT) and the post-processing is statistical language model (word n-gram)

- Degraded text recognition is a difficult task. Integrating High-level knowledge sources can be used to infer and discriminate a degraded word among other possibilities

- The proposed method for machine-printed Arabic word has been successfully applied for mono- and multi-font

- New feature technique used for Arabic text; A Block-Based DCT (BBDCT) feature extractor approach

- One type DCT coefficients feature set is used

- New word-image normalisation approach for smaller feature vector size; images are normalised according to the calculated averages dimension, and then down sampled to the nearest ratio greater than or equal to 1/2 of its original size, and its dimension is multiples of "8"

- Feature vectors size defined systematically dependent on the font type and training set

- Developing a machine-printed generator to be used for creating a large number of synthetic word images

- Designing methods to deform images; by image skewing and noising degradation using salt and pepper noise

- The three implemented databases of Arabic word images will be made publicly available for the purpose of research to become a benchmark for printed Arabic words

- The synthetic database of Arabic word images; comprises all sentences (Ayah) of the longest Surah in The Holy Qur'an) is suitable for use in both OCR and OCR integrated linguistics post-processing systems development

- Automated methodology designed for real scanned word samples can be used for collecting and founding handwritten words database

- Using The Holy Qur'an help to advance the state-of-the-art in OCR research. Since it is closed corpus and it is considered by

linguists and grammarians as a reference; all of these help in prototype system design (recognition and linguistic) before generalising it

6.5 Future Work

The results achieved in this thesis motivate to enhance the performance of the proposed system and make the approach applicable on a wider range of tasks related to off-line text recognition application. Since text recognition is such a large subject, there is plenty of scope for future directions. The present work can be extended in the following directions:

- Finding remedy to the DCT drawbacks or use another feature technique that is invariance to rotation and translation
- Speed up the post-processing by reducing the number of sentences generated from the recogniser output lattice through applying rejection strategy
- Take into consideration the candidates' confidence probabilities in post-processing stage to minimise the worsening
- Enlarge the vocabulary size to examine the efficiency of the linguistic post-processing to improve reliability of the holistic word-based recogniser to handle larger vocabulary
- Apply cascading method used in [38] to develop multi-font systems using font identifier followed by mono-font word recognition

- Investigate the employing of other linguistic information in post-processing level such as semantic collocations, class-based n-gram, and Probabilistic context-free grammars

- In language model creation, take into consideration the pause symbols within Ayah which define a complete sentence

- Employ the idea of feed-back to improve the performance by automatically adjusting the system parameters; such as feature selection and the initial values of the state transition matrix

- Real-word error detection and correction to improve the recognition rate. The word errors resulted from the closed vocabulary word recogniser are real-word errors; this error occurs when a word is misspelled as another valid word. This could be handled by one of the following:

    - Since the proposed post-processing only sees the recogniser candidate list, it is necessary to incorporate linguistic knowledge by providing an access to the main corpus. This access is to help word prediction in a sentence by elaborating the linguistic knowledge such as word collocations, syntax and semantics

    - Employing the feedback to resolve the word ambiguity using different feature set and/or feature category like structural and statistical

- Make the created databases publicly available for the scientific community
- Expanding the benchmark databases with different types of words and paragraph related to electronic written corpora
- Extend the proposed system to work for different fonts, text sizes and styles (normal, bold, italic, bold italic) at different printing and image resolutions. These fonts including Arabic and other languages
- Build a handwritten benchmark database using the methodology described in chapter 3
- Apply the presented techniques for Arabic handwritten text of one writer; in the same manner as the mono-font approach. The Holy Qur'an text is available in handwritten books each one is written by a different writer in different styles such as Uthmani Scripts; this can help to initiate this point.
- A remedy to vocabulary growth problem by using Arabic word's morphology in the recognition phase is worth further investigation

To conclude, many issues are then still open and the problem of Arabic text recognition is still far from being solved.

BIBLIOGRAPHY

[1] S. Madhvanath and V. Govindaraju, "The role of holistic paradigms in handwritten word recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 23, pp. 149-164, 2001.

[2] C. J. Davis, "The spatial coding model of visual word identification," *Psychological review,* vol. 117, p. 713, 2010.

[3] M. Perea and E. Rosa, "Does "whole-word shape" play a role in visual word recognition?," *Perception & Psychophysics,* vol. 64, pp. 785-794, 2002.

[4] A. Belaïd and C. Choisy, "Human reading based strategies for off-line Arabic word recognition," in *Arabic and Chinese Handwriting Recognition*, ed: Springer, 2008, pp. 36-56.

[5] L. Heutte, A. Nosary, and T. Paquet, "A multiple agent architecture for handwritten text recognition," *Pattern recognition,* vol. 37, pp. 665-674, 2004.

[6] A. M. Elgammal and M. A. Ismail, "A graph-based segmentation and feature extraction framework for Arabic text recognition," in *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, 2001, pp. 622-626.

[7] M. T. Parvez and S. A. Mahmoud, "Offline Arabic handwritten text recognition: a survey," *ACM Computing Surveys (CSUR),* vol. 45, p. 23, 2013.

[8] F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi, and J. Hennebert, "A new arabic printed text image database and evaluation protocols," in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, 2009, pp. 946-950.

[9] A. G. Al-Hashim and S. A. Mahmoud, "Benchmark database and GUI environment for printed Arabic text recognition research," *WSEAS Trans. Inf. Sci. Appl,* vol. 4, pp. 587-597, 2010.

[10] A. AbdelRaouf, C. A. Higgins, T. Pridmore, and M. Khalil, "Building a multi-modal Arabic corpus (MMAC)," *International Journal on Document Analysis and Recognition (IJDAR),* vol. 13, pp. 285-302, 2010.

[11] A. M. AL-Shatnawi, S. AL-Salaimeh, F. H. AL-Zawaideh, and K. Omar, "Offline arabic text recognition–an overview," *World of Computer Science and Information Technology Journal (WCSIT),* vol. 1, pp. 184-192, 2011.

[12] M. Charfi, M. Kherallah, A. E. Baati, and A. M. Alimi, "A New Approach for Arabic Handwritten Postal Addresses Recognition," *arXiv preprint arXiv:1204.1678,* 2012.

[13] N. Azizi, N. Farah, and M. Sellami, "OFF-LINE HANDWRITTEN WORD RECOGNITION USING ENSEMBLE OF CLASSIFIER SELECTION AND FEATURES FUSION," *Journal of Theoretical & Applied Information Technology,* vol. 14, 2010.

[14] H. Xue and V. Govindaraju, "Hidden Markov models combining discrete symbols and continuous attributes in handwriting recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence,* pp. 458-462, 2006.

[15] S. Kanoun, A. Ennaji, Y. LeCourtier, and A. M. Alimi, "Script and nature differentiation for Arabic and Latin text images," in *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, 2002, pp. 309-313.

[16] I. SI Abuhaiba, "Arabic font recognition using decision trees built from common words," *CIT. Journal of computing and information technology,* vol. 13, pp. 211-224, 2005.

[17] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 31, pp. 855-868, 2009.

[18] J. H. AlKhateeb, J. Jiang, J. Ren, F. Khelifi, and S. S. Ipson, "Multiclass classification of unconstrained handwritten Arabic words using machine learning approaches," *Open Signal Processing Journal,* vol. 2, pp. 21-28, 2009.

[19] S. Bahgat, S. Ghomiemy, S. Aljahdali, and M. Alotaibi, "A proposed hybrid technique for recognizing Arabic characters," *International Journal of Advanced Research in Artificial Intelligence,* vol. 1, pp. 35-43, 2012.

[20] H. Hassen, "A Comparative study of Arabic handwritten characters invariant feature," *arXiv preprint arXiv:1211.1800,* 2012.

[21] M. Elzobi, A. Al-Hamadi, A. Saeed, and L. Dings, "Arabic handwriting recognition using Gabor wavelet transform and SVM,"

in *Signal Processing (ICSP), 2012 IEEE 11th International Conference on*, 2012, pp. 2154-2158.

[22] A. Lawgali, A. Bouridane, M. Angelova, and Z. Ghassemlooy, "Handwritten Arabic character recognition: Which feature extraction method?," *International Journal of Advanced Science and Technology,* vol. 34, pp. 1-8, 2011.

[23] N. Arica and F. T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 31, pp. 216-233, 2001.

[24] V. Maheshkar, S. Kamble, S. Agarwal, and V. K. Srivastava, "FEATURE IMAGE GENERATION USING LOW, MID AND HIGH FREQUENCY REGIONS FOR FACE RECOGNITION," *The International Journal of Multimedia & Its Applications,* vol. 4, p. 75, 2012.

[25] S. Dabbaghchian, M. P. Ghaemmaghami, and A. Aghagolzadeh, "Feature extraction using discrete cosine transform and discrimination power analysis with a face recognition technology," *Pattern Recognition,* vol. 43, pp. 1431-1440, 2010.

[26] G. Rajput and H. Anita, "Handwritten script recognition using DCT and wavelet features at block level," *IJCA, Special issue on RTIPPR (3),* pp. 158-163, 2010.

[27] J. H. AlKhateeb, J. Ren, J. Jiang, and S. Ipson, "A machine learning approach for classifying offline handwritten Arabic words," in *CyberWorlds, 2009. CW'09. International Conference on*, 2009, pp. 219-223.

[28] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason*, et al.*, "The HTK book (Revised for HTK version 3.4. 1)," *Cambridge University,* 2009.

[29] A. Elbaati, H. Boubaker, M. Kherallah, A. Ennaji, and A. M. Alimi, "Arabic handwriting recognition using restored stroke chronology," in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, 2009, pp. 411-415.

[30] M. Hamdani, H. El Abed, M. Kherallah, and A. M. Alimi, "Combining multiple HMMs using on-line and off-line features for off-line Arabic handwriting recognition," in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, 2009, pp. 201-205.

[31] H. E. Abed and V. Märgner, "Comparison of different preprocessing and feature extraction methods for offline recognition of handwritten arabicwords," in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, 2007, pp. 974-978.

[32] J. Kacur and G. Rozinaj, *Practical issues of building robust HMM models using HTK and SPHINX systems*: INTECH Open Access Publisher, 2008.

[33] L. M. Lorigo and V. Govindaraju, "Offline Arabic handwriting recognition: a survey," *IEEE Transactions on Pattern Analysis & Machine Intelligence,* pp. 712-724, 2006.

[34] M. Khorsheed, "A lexicon based system with multiple hmms to recognise typewritten and handwritten Arabic words," in *The 17th National Computer Conference, Madinah, Saudi Arabia*, 2004, pp. 5-8.

[35] M. S. Khorsheed, "Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK)," *Pattern Recognition Letters,* vol. 28, pp. 1563-1571, 2007.

[36] H. A. Al-Muhtaseb, S. A. Mahmoud, and R. S. Qahwaji, "Recognition of off-line printed Arabic text using Hidden Markov Models," *Signal Processing,* vol. 88, pp. 2902-2912, 2008.

[37] H. A. Al-Muhtaseb, S. A. Mahmoud, R. S. Qahwaji, M. Demiralp, N. Baykara, and N. Mastorakis, "A novel minimal Arabic script for preparing databases and benchmarks for Arabic text recognition research," in *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*, 2009.

[38] F. Slimane, S. Kanoun, A. M. Alimi, J. Hennebert, and R. Ingold, "Comparison of global and cascading recognition systems applied to multi-font arabic text," in *Proceedings of the 10th ACM symposium on Document engineering*, 2010, pp. 161-164.

[39] J. H. AlKhateeb, J. Ren, J. Jiang, and H. Al-Muhtaseb, "Offline handwritten Arabic cursive text recognition using Hidden Markov Models and re-ranking," *Pattern Recognition Letters,* vol. 32, pp. 1081-1088, 2011.

[40] R. Prasad, S. Saleem, M. Kamali, R. Meermeier, and P. Natarajan, "Improvements in hidden Markov model based Arabic OCR," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008, pp. 1-4.

[41] J. Devlin, M. Kamali, K. Subramanian, R. Prasad, and P. Natarajan, "Statistical machine translation as a language model for handwriting recognition," in *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, 2012, pp. 291-296.

[42] R. A.-H. Mohamad, L. Likforman-Sulem, and C. Mokbel, "Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 31, pp. 1165-1177, 2009.

[43] N. Farah, L. Souici, and M. Sellami, "Classifiers combination and syntax analysis for Arabic literal amount recognition," *Engineering Applications of Artificial Intelligence,* vol. 19, pp. 29-39, 2006.

[44] U.-V. Marti and H. Bunke, "A full English sentence database for off-line handwriting recognition," in *Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on*, 1999, pp. 705-708.

[45] T. Su, T. Zhang, and D. Guan, "HIT-MW dataset for offline Chinese handwritten text recognition," in *Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006.

[46] M. Ziaratban, K. Faez, and F. Bagheri, "FHT: An unconstraint Farsi handwritten text database," in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, 2009, pp. 281-285.

[47] S. A. Mahmoud, I. Ahmad, M. Alshayeb, and W. G. Al-Khatib, "A database for offline Arabic handwritten text recognition," in *Image Analysis and Recognition*, ed: Springer, 2011, pp. 397-406.

[48] A. Mezghani, S. Kanoun, M. Khemakhem, and H. E. Abed, "A database for arabic handwritten text image recognition and writer identification," in *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, 2012, pp. 399-402.

[49] A. Farghaly and K. Shaalan, "Arabic natural language processing: Challenges and solutions," *ACM Transactions on Asian Language Information Processing (TALIP),* vol. 8, p. 14, 2009.

[50] H. S. Al-Khalifa, M. M. Al-Yahya, A. Bahanshal, and I. Al-Odah, "SemQ: A proposed framework for representing semantic opposition in the Holy Quran using Semantic Web technologies," in *2009 International Conference on the Current Trends in Information Technology (CTIT)*, 2009, pp. 1-4.

[51] K. Dukes. (2009). *The Quranic Arabic Corpus*. Available: http://corpus.quran.com/

[52] M. S. Khorsheed, "Off-line Arabic character recognition–a review," *Pattern analysis & applications,* vol. 5, pp. 31-45, 2002.

[53] R. Longadge and S. Dongre, "Class Imbalance Problem in Data Mining Review," *arXiv preprint arXiv:1305.1707,* 2013.

[54] R. I. Zaghloul, E. F. AlRawashdeh, and D. M. Bader, "Multilevel Classifier in Recognition of Handwritten Arabic Characters," *Journal of Computer Science,* vol. 7, p. 512, 2011.

[55] M. Harouni, D. Mohamad, M. S. M. Rahim, S. M. Halawani, and M. Afzali, "Handwritten Arabic character recognition based on minimal geometric features," *International Journal of Machine Learning and Computing,* vol. 2, p. 578, 2012.

[56] B. Alsallakh and H. Safadi, "Arapen: an arabic online handwriting recognition system," in *2006 2nd International Conference on Information & amp; Communication Technologies*, 2006.

[57] R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 22, pp. 63-84, 2000.

[58] V. Märgner and M. Pechwitz, "Synthetic data for Arabic OCR system development," in *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, 2001, pp. 1159-1163.

[59] R. Kasturi, L. O'gorman, and V. Govindaraju, "Document image analysis: A primer," *Sadhana,* vol. 27, pp. 3-22, 2002.

[60] T. Varga and H. Bunke, "Effects of training set expansion in handwriting recognition using synthetic data," in *Proc. 11th Conf. of the Int. Graphonomics Society*, 2003, pp. 200-203.

[61] S. A. Mahmoud and S. M. Awaida, "Recognition of off-line handwritten Arabic (Indian) numerals using multi-scale features and support vector machines vs. hidden markov models," *The Arabian Journal for Science and Engineering,* vol. 34, pp. 429-444, 2009.

[62] N. Farah, T. Khadir, and M. Sellami, "Artificial neural network fusion: Application to Arabic words recognition," in *ESANN*, 2005, pp. 151-156.

[63] S. M. Awaidah and S. A. Mahmoud, "A multiple feature/resolution scheme to Arabic (Indian) numerals recognition using hidden Markov models," *Signal Processing,* vol. 89, pp. 1176-1184, 2009.

[64] O. Al-Jarrah, S. Al-Kiswany, B. Al-Gharaibeh, M. Fraiwan, and H. Khasawneh, "A new algorithm for arabic optical character recognition," *WSEAS Transactions on Information Science and Applications,* vol. 3, pp. 832-845, 2006.

[65] F. Einsele, R. Ingold, and J. Hennebert, "A language-independent, open-vocabulary system based on hmms for recognition of ultra low resolution words," in *Proceedings of the 2008 ACM symposium on Applied computing*, 2008, pp. 429-433.

[66] S. F. Rashid, F. Shafait, and T. M. Breuel, "An evaluation of HMM-based techniques for the recognition of screen rendered text," in *2011 International Conference on Document Analysis and Recognition*, 2011, pp. 1260-1264.

[67] A. L. Koerich, R. Sabourin, and C. Y. Suen, "Large vocabulary off-line handwriting recognition: A survey," *Pattern Analysis & Applications,* vol. 6, pp. 97-121, 2003.

[68] C. Y. Suen, "N-gram statistics for natural language understanding and text processing," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* pp. 164-172, 1979.

[69] R. Rosenfeld, "Two decades of statistical language modeling: Where do we go from here?," 2000.

[70] J. Gao, P. Nguyen, X. Li, C. Thrasher, M. Li, and K. Wang, "A Comparative Study of Bing Web N-gram Language Models for Web Search and Natural Language Processing," in *Web N-gram Workshop*, 2010, p. 16.

[71] K. Meftouh, M. Tayeb Laskri, and K. Smaili, "Modeling Arabic Language using statistical methods," *Arabian Journal for Science and Engineering,* vol. 35, p. 69, 2010.

[72] R. Rosenfeld, "A maximum entropy approach to adaptive statistical language modeling," 1996.

[73] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech & Language,* vol. 13, pp. 359-393, 1999.

[74] J. Domokos and G. Toderean, "Text conditioning and statistical language modeling aspects for Romanian language," *Acta*

Universitatis Sapientiae, Electrical and Mechanical Engineering, ISSN, pp. 2065-5916, 2009.

[75] H. S. Baird, "The state of the art of document image degradation modelling," in *Digital Document Processing*, ed: Springer, 2007, pp. 261-279.

[76] S. Nakagawa, E. Turmunkh, H. Kibishi, K. Ohta, Y. Fujii, M. Tsuchiya*, et al.*, "Development of large vocabulary continuous speech recognition system for Mongolian language," in *SLTU*, 2012, pp. 19-23.

[77] Y.-X. Li, C. L. Tan, and X. Ding, "A hybrid post-processing system for offline handwritten Chinese script recognition," *Pattern analysis and applications,* vol. 8, pp. 272-286, 2005.

[78] R. Iyer and M. Ostendorf, "Relevance weighting for combining multi-domain data for n-gram language modeling," *Computer Speech & Language,* vol. 13, pp. 267-282, 1999.

[79] S. A. Seyedin, M. F. A. Fauzi, and F. M. Anuar, "1-d dct shape feature for image retrieval," in *Signal and Image Processing Applications (ICSIPA), 2009 IEEE International Conference on*, 2009, pp. 431-436.