LUCIDAH

Ligative and Unligative Characters in a Dataset for Arabic Handwriting

Yousef Elarian[1]
Teaching and Learning Coaching
Interserve®
Madīnah, Saudi Arabia

Irfan Ahmad[2]
Information and Computer Science Department
King Fahd University of Petroleum & Minerals
Dhahran 31261, Saudi Arabia

Abdelmalek Zidouri[3]
Electrical Engineering Department
King Fahd University of Petroleum & Minerals
Dhahran 31261, Saudi Arabia

Wasfi G. Al-Khatib[4]
Information and Computer Science Department
King Fahd University of Petroleum & Minerals
Dhahran 31261, Saudi Arabia

***Abstract*—Arabic script is inherently cursive, even when machine-printed. When connected to other characters, some Arabic characters may be optionally written in compact aesthetic forms known as ligatures. It is useful to distinguish ligatures from ordinary characters for several applications, especially automatic text recognition. Datasets that do not annotate these ligatures may confuse the recognition system training. Some popular datasets manually annotate ligatures, but no dataset (prior to this work) took ligatures into consideration from the design phase. In this paper, a detailed study of Arabic ligatures and a design for a dataset that considers the representation of ligative and unligative characters are presented. Then, pilot data collection and recognition experiments are conducted on the presented dataset and on another popular dataset of handwritten Arabic words. These experiments show the benefit of annotating ligatures in datasets by reducing error-rates in character recognition tasks.**

***Keywords*—Arabic ligatures; automatic text recognition; handwriting datasets; Hidden Markov Models**

I. Introduction

Arabic script is widely used, not only for the Arabic language, but also for other languages like Urdu, Jawi, and Persian [1]. Despite such importance of the Arabic script, research in technologies that serve it still remains underdeveloped [2]. This is mainly attributed to the scarcity of specialized resources and algorithms that take the peculiarities of the Arabic script into consideration [2] [3].

Among the peculiarities of the Arabic script is that it is inherently (and diversely) cursive. Twenty-two out of the 28 Arabic letters must be connected to the letter that comes next to them in a word. The general form of connection is via short horizontal extensions, shown in the second row of Table I. Some characters, however, also accept to be connected in special compact forms known as *ligatures* [4] (or typographic ligatures [5]). Examples of Arabic ligatures are shown in the last row of Table I.

Most ligatures are optional, i.e., when certain sequences of characters (that we refer to as "ligatives") occur, they can be either written in the normal horizontal connection form or as a ligature, depending on the writer's choice.

Many character sequences are not ligative (or "unligative", as we refer to them); i.e., they do not have optional ligature forms. One particular family of ligatures ("لا" or "ـلا" family, with different Hamza ("ء") and Madda ("~") combinations, namely, "لأ", "ـلأ", "لآ", "ـلآ", "لإ", "ـلإ") is considered obligatory, i.e., its constituent LAM and ALEF characters cannot be connected in the common horizontal form. Table I shows different connections of a ligative, an unligative, and an obligatorily ligative examples.

Ligatures are often used in handwriting for their esthetic effects and their compactness. Their presence, however, can add complexities to tasks like Arabic character recognition and synthesis. Researchers in these and similar fields have attempted to handle some ligatures [6] [7] [8] [9], but a comprehensive and systematic list of ligatives was only available after the initial work presented by the authors in [4].

In addition to handwriting, Arabic ligatures are increasingly being incorporated in modern computer fonts to help make texts more compact, beautiful, and as an alternative method for paragraph justification [10] [11].

The "Arabic Presentation Forms-A" of the Unicode standard encodes more than 300 Arabic and Extended-Arabic ligatures [12]. We notice, however, that their set suffers from incompleteness and inconsistencies. For example, the "ئخ" ligature (as in "دائخ") is absent in Unicode despite the presence of similar ligatures such as "ئج", "ئح", "نج", "نخ", and "تخ" (cf. [4]).

To the best of our knowledge, Arabic ligatures were not systematically analyzed and studied before. In this paper, we present a detailed study of Arabic ligatures that lead to the design of LUCIDAH, a Ligative and Unligative Dataset for Arabic Handwriting. LUCIDAH is especially designed to be representative of both: ligatures and characters in their non-ligatured forms. It is also designed to be practically concise and natural; based on the guidelines in [4].

TABLE. I. EXAMPLES OF (A) A LIGATIVE, (B) AN UNLIGATIVE AND (C) AN OBLIGATORILY-LIGATIVE CHARACTERS

Mode of connection	**Ligative**	**Unligative**	**Obligatorily**
Unconnected Character-shapes	لـ ـمـ ـح	لـ ـس	لـ ـا
Non-Ligature Connections	لمح	لس	NA
Ligature Connections	لمح	NA	لا or لا

The rest of this paper is organized as follows. In Section II, background on Arabic characters in several typographic models is presented. Section III is devoted to the discussion of ligatures and their categorization. In Section IV, the design of LUCIDAH is described. In Section V, implementation issues such as the data collection forms, the demographic distribution of writers, and some scanning parameters are discussed. In Section VI, text recognition experiments are presented to highlight the impact of annotating ligatures on recognition error-rates. Finally, conclusions are presented in Section VII.

II. BACKGROUND ON THE ARABIC SCRIPT

The Arabic script consists of 28 alphabetic letters that are mapped to the 36 computer characters (shown and explained in details in the subsection of the Arabic Typographic Models). Most Arabic characters must connect to subsequent characters to compose words. A few Arabic characters, however, do not connect to subsequent characters (e.g., Letters "ا" and "ر" in Fig. 1). If any of the non-connecting characters occurs before another letter in a word, they cause the word to split into disconnected *Pieces of Arabic Words* (PAWs) [13] [14] [15] or *subwords* [16]. (In addition to that, there is one non-connecting character, "ء", that does not allow previous characters to connect to it, even when the previous character is a connecting one; hence, its occurrence also causes words to be split into PAWs).

A. Arabic Character-Shapes

Each Arabic character takes a "*character-shape*" based on its position in a PAW, as detailed in the following subsection. Arabic traditionally introduces four *character-shapes*, based on characters' context (i.e. connection with previous and next characters in a PAW). If a PAW consists of only one character, that character takes the *Alone* (A) character-shape. PAWs with more than one letter must start with a *Beginning* (B) character-shape and end with an *Ending* (E) character-shape. If the PAW consists of three or more letters, there would be characters between the Beginning and the Ending character-shapes. These must be *Middle* (M) character-shapes. Table II shows examples of (A), (B), (M), and (E) character-shapes of the letter "س".

Fig. 2 shows an example word consisting of three PAWs and annotates its character-shapes (as Alone (A), Beginning (B), Middle (M), and Ending (E)). Note that Arabic text is written from right to left; hence, the Beginning character-shape of an Arabic multi-lettered PAW appears at the right end of the PAW, whereas the Ending character-shape comes at the leftmost end of it.

Non-connecting letters do not connect, so they do not have Beginning (B) or Middle (M) character-shapes. In addition, there are two characters ("ة" and "ى") that linguistically can only occur at the end of a word, and hence are treated as non-connecting. Finally, as was mentioned earlier, there is one character, "ء" that does not accept connections from either side, and hence; it only has the Alone (A) character-shape.

B. Arabic Typographic Models

There are several Arabic typographic models. The traditional model introduces up to 4 character-shapes per letter (connecting letters having four shapes, non-connecting letters having two, and "ء" has one). The traditional model encompasses more than a hundred character-shapes, which can be a large number for some applications [17]. Hence, several other models were introduced to represent Arabic script with less numbers of shapes.

One of these reduction models is the 2-Shapes model [18]. The 2-Shapes model clusters the (B) and (M) shapes together and the (A) and (E) shapes together, whenever the differences are merely the connecting extension found preceding the (M) and (E) shape, as seen in Fig. 3 (left and middle parts).

The 1-Shape model introduces *root shapes*, or basic glyph parts that are present in all shapes of a character [17]. In addition to the common root shapes, many (A) and (E) character-shapes also have *tail*s. For example, the character "ص" in Fig. 3 (right) has the "صـ" root in all of its shapes, and the "ں" tail in its (A) and (E) shapes.

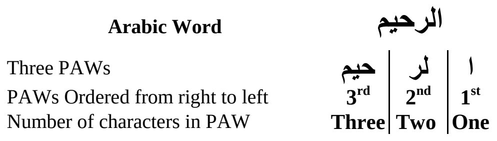


Fig. 1. An Arabic Word Divided into its PAWs.

TABLE. II. DIFFERENT CHARACTER-SHAPES OF THE ARABIC CHARACTER "س" BASED ON ITS CONNECTIONS TO PREVIOUS AND NEXT CHARACTERS

Shape	**Example Character-Shape**	**Example Word**	**Connected to previous character**	**Connected to next character**	**Size of PAW**[a]
(A)	س	درس	No	No	1
(B)	سـ	أسيف	No	Yes	2+
(M)	ـسـ	تسامح	Yes	Yes	3+
(E)	ـس	خس	Yes	No	2+

[a] The "+" sign after a number *X* in this column denotes "*X* or more characters".

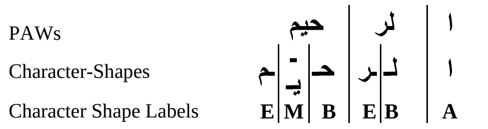


Fig. 2. An Arabic word divided into PAWs with character-shapes annotated.

4-Shapes Model				**2-Shape Model**		**1-Shape Model**
(A)	*(E)*	*(M)*	*(B)*	*(A)and (E)*	*(M)and (B)*	*(A), (E), (M), and (B)*
س	ـس	ـسـ	سـ	س ـس	سـ ـسـ	س ـس ـسـ سـ
ش	ـش	ـشـ	شـ	ش ـش	شـ ـشـ	ش ـش ـشـ شـ
ص	ـص	ـصـ	صـ	ص ـص	صـ ـصـ	ص ـص ـصـ صـ
ض	ـض	ـضـ	ضـ	ض ـض	ضـ ـضـ	ض ـض ـضـ ضـ
ط	ـط	ـطـ	طـ	ط ـط	طـ ـطـ	ط ـط ـطـ طـ
ظ	ـظ	ـظـ	ظـ	ظ ـظ	ظـ ـظـ	ظ ـظ ـظـ ظـ

Fig. 3. Examples of Character Groups in the 4-, 2-, and 1-Shape(s) Models.

Appendix I shows Arabic characters in different models. There are few characters that do not share the root shapes across some of their character shapes (such as “هـ” and “ك”) . These may not fully cluster their shapes in the 2- and 1-Shape model.

Table III shows examples of a character that would only fit in the 4-Shapes model (since its character-shapes do not share any common glyphs), another that fits in the 2-Shapes model, and a third character that can fit in any model including the 1-Shape model.

TABLE. III. EXAMPLES OF CHARACTER-SHAPES WITH THEIR SMALLEST (BEST) POSSIBLE SHAPE MODELS

Best Model Fit	(B)	(M)	(E)	(A)
4-Shapes	هـ	ـهـ	ـه	ه
2-Shapes	كـ	ـكـ	ـك	ك
1-Shape	صـ	ـصـ	ـص	ص

More reduction to the numbers of representative character-shapes of a dataset can be achieved via the *dot-less* (DL) model [19] [20]. The dot-less reduction model clusters Arabic character-shapes with identical main glyphs that differ only in upper or lower dots, upper or lower Hamza "ء", or upper Madda “ٓ”. In Fig. 4, we show the examples of Fig. 3 after DL reduction.

The dot-less model exploits resemblances among different letters whereas the Shapes model exploits resemblances among character-shapes. The two reductions can be combined. The groupings of character-shapes in DL reduction are shown in Appendix I.

The counts of character-shapes for different typographic models are displayed in Table IV. The table does not take into consideration counts of the secondary glyphs (dots, Hamza’s, Madda’s, extensions, and tails) nor the different allographs that some letters may have (e.g., the dent-less س (س)).

4-Shapes DL				2-Shapes DL		1-Shapes DL
(A)	*(E)*	*(M)*	*(B)*	*(A)and (E)*	*(M)and (B)*	*(A), (E), (M), and (B)*
س ش	ـس ـش	ـسـ ـشـ	سـ شـ	س ش	ـسـ سـ	سـ شـ and the tail ں
ص ض	ـص ـض	ـصـ ـضـ	صـ ضـ	ص ض	ـصـ ضـ	صـ ضـ and the tail ں
ظ	ـط ـظ	ـطـ ـظـ	طـ ظـ	ط ظ	ـطـ ظـ	ط ظ

Fig. 4. Examples of Character Groups in the 4-, 2-, and 1-Shape(s) DL Model.

TABLE. IV. COUNTS OF CHARACTER-SHAPES FOR DIFFERENT TYPOGRAPHIC MODELS

Model	(A)	(E)	(M)	(B)	Total
Traditional	36	35	23	23	117
Dot-Less 4-Shapes	19	18	11	11	59
2-Shapes	35 + 5[a]		23 + 3[b]		66
Dot-Less 2-Shapes	18 + 3[c]		11 + 2[d]		34
1-Shape	45				45
Dot-Less 1-Shape	24				24

[a] Extra (A) shapes correspond to ء, ع, غ, ه and ة.

[b] Extra (M) shapes correspond to ـعـ, ـغـ and ـهـ.

[c] Extra (A) shapes correspond to ء, ع and ه.

[d] Extra (M) shapes correspond to ـعـ and ـهـ.

In general, counts of glyphs should not be the only factor considered in deciding on a model, since reduction can impose more challenges in dealing with the secondary smaller glyphs.

III. CATEGORIZATION OF LIGATIVES AND LIGATURES

In this section, we attempt to categorize ligatures. First, we notice that there are some Arabic character-shapes that always allow their preceding and connecting characters to form ligatures [4]. We refer to such character-shapes as *Omni-ligative*. Omni-ligatives can be of either the Middle or the Ending shapes of a character (as they must be connected by their preceding characters).

The Omni-ligative character-shapes, using the dot-less model, are: “ـحـ”, “ـح”, “ـمـ”, “ـم”, “ـهـ”, “ـصـ”, “ـص”, “ـطـ”, and “ـط”; the last four being included after revising what was initially reported in [4]. Whether the “ي” character-shape is Omni-ligative or *Partio-ligative* can be debated (Partio-ligative is a term we coined from "partial" to indicate ligatives that are not Omni-ligatives). Here, we consider it a partio-ligative since a short concatenation stroke can be noticed when preceded by some characters making their combination more like an unligative.

The first five Omni-ligatives (viz. “ـحـ”, “ـح”, “ـمـ”, “ـم”, and “ـهـ”) are connected with a stroke that descends when writing (from the right to the left where the Omni-ligative character is to be connected from its top). The four next Omni-ligative characters get ligatured via strokes that ascend diagonally from their bottom left corner. We refer to these as “ascending Omni-ligatives”.

Descending ligatures can be further categorized according to their first-character into ligatures that involve inversion or flipping the first character down and others that connect from the top of a cusp without returning from the cusp to the baseline. The different categories of Omni-ligative connections are demonstrated in Table V.

Ligatures that result from partio-ligatives can also be categorized according to some shared patterns. For example, some involve top of the cusps connections, such as “سر”, “ـسر”, “صر”, “ـصر”, and “٧” (or the v-like allograph of “هـ”). Notice that characters with several allographs can have different ligativity with each allograph. Another example is sticking the circular base of “ک” and “ـک” with a highly ascending character like “ا” or “ل”.

TABLE. V. EXAMPLES OF ASCENDING AND DIFFERENT CATEGORIES OF DESCENDING CONNECTION STROKES IN OMNI-LIGATIVES

Connection Stroke	Ligatured	Not ligatured
Descending from right	جح	جح
Ascending from left	قط	قط
Descending from right with flipped down first-letter	بح	بح
Descending from the top of a cusp at the right	سح	سح

TABLE. VI. EXAMPLES OF DIFFERENT CATEGORIES OF CONNECTION STROKES IN PARTIO-LIGATIVES

Form	**From top of cusps**					**High link**	**"ک" and "کل"**	**Closed letter**	**Higher cusp**
Not ligatured	سي	سر	حبر	صر	نهر	ين	كا	حد	لبيتنا
Ligatured									

Some calligraphic styles (e.g., [11] [21]) add more restricted rules, but since these are not widely used in everyday handwriting, we do not include them in the ligatives list. The "حـ" and "ـحـ" character-shapes, for example, are sometimes made closed (ـصـ) especially before an ascending character (like "أ", "ـد" or "ـل"). Similarly, some writing styles [21] may encourage making one cusp taller than usual if more than three cusps occur in sequence. Examples of partio-ligative ligature categories are displayed in Table VI.

IV. DATASET DESIGN

Researchers have recognized the role of ligatures in Arabic datasets but lacked resources that systematically cover them. Research in text recognition [6], including popular Arabic handwriting datasets [7] [8], attempted to manually annotate ligatures. The absence of a comprehensive list of ligatures prior to [4] made recognizing ligatures difficult. In this section, we follow the guidelines and lists from [4] to design representative yet concise ligative and an unligative datasets.

Corpora and dataset design consider obtaining concise yet representative samples to reflect certain characteristics of a language. Representativeness in corpus and dataset design is defined as "the extent to which a sample includes the full range of variability in a population [22]". Sampling, when used in the context of corpora, refers to the process of selecting limited-sized yet representative texts that are expected to reproduce the characteristics of the underlying language. The sampling theory addresses important concerns such as the sampling unit and the sampling frame [23]. Moreover, we introduce the term *representing unit* to refer to the smallest writing unit that would assure representativeness and conciseness. The representing unit can be one (or more) word(s), a character, a character-shape, or even a pen stroke. By *representation criteria*, we refer to the set of units that need to be present in the dataset for it to be considered representative. The representation criteria can be, for example, to cover character-shapes under a certain reduction model (see Section 2). Representativeness under some reduction model can be assured either for each writer or collectively for sets of writers. We refer to the number of writers ensuring representation under some criteria as *representative forms.*

LUCIDAH design is mainly concerned with selecting adequate representing units, representation criteria, and representative forms to achieve an acceptable level of representativeness with concise and naturally written units. Typically, representativeness and conciseness; conciseness and naturalness; naturalness and representativeness may have conflicting requirements. In the presented dataset design, we attempt to balance all factors to reasonable levels via the exploitation of the powers of different reduction models for different parts of the dataset.

LUCIDAH has two main parts: the ligative and the unligative parts. For the representativeness of undistorted character-shapes and ligatures, LUCIDAH must contain, not only all the desired ligatures, but also all the desired character-shapes in unligative forms [4]. These parts are addressed by careful selection of texts containing characters with extra-care to select their preceding and following characters. The design of the ligative and unligative parts may have different requirements; and hence, each of these parts is discussed in a separate subsection, below.

A. Design of the Ligative Part

Ligatives, in general, can involve more than two characters, if one or more consecutive Middle character-shapes keep forming ligatives with their next characters. Sequences of characters are frequently referred to as *n*-grams. The problem of *n*-grams is that the counts of their combinations increase exponentially with the sizes of their constituent characters, *n*. This clearly makes collecting *n*-grams intractable. We decided to treat *n*-gram ligatives as (n - 1) connecting bigrams. Hence, we need to limit the representing unit of the ligative part of LUCIDAH to be within the bigram options only while hoping that bigrams would represent all *n*-grams with an acceptable level of naturalness.

A ligature may only occur if a character connects to its subsequent. Hence, bigram ligatures can be formed when either a B or an M character-shape connects to an M or an E character-shape. The four resulting combinations of connected bigrams (shown with examples in Table VII) are: B and E (denoted as *BE*, and also denotable as *A-bigram*; since it, as a whole, cannot be connected to neither previous nor next characters), B and M (denoted as *BM* and also denotable as *B-bigram*; since it cannot connect to a previous character but connects to a following character), M and M (denoted as *MM* and also denotable as *M-bigram*; since it has to be connected to both a previous and a next character), and M and E (denoted as *ME* and also denotable as *E-bigram*; since the whole ligature is not connected to a next but to a previous character).

Similarly, connected trigram can be expressed as BMM or a B-trigram, BME or an A-trigram, MMM or a M-trigram, and MME or an E-trigram. We counted the possible bigram combinations and bigram ligatives (including the "ascending Omni-ligatives") and these are displayed in Table VIII.

TABLE. VII. EXAMPLES OF A STANDALONE LIGATURE AND LIGATURES AT THE BEGINNING, MIDDLE, AND END, OF THEIR CORRESPONDING SUBWORD

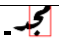
2nd **1st**	**(M)**	**(E)**
(B)	**(BM) or B-Ligative** 	**(BE) or A-Ligative**
(M)	**(MM) or M-Ligative**	**(ME) or E-Ligative**

TABLE. VIII. COUNTS OF LIGATIVE BIGRAMS WITH THE TOTAL NUMBER OF BIGRAMS COMBINATIONS FOR THE DIFFERENT TYPOGRAPHIC MODELS

Model	**(BE)**	**(ME)**	**(MM)**	**(BM)**	**Total**
Traditional	247 / **782**	255 / **782**	213 / **529**	212 / **529**	927 / **2622**
Dot-Less (DL)	56 / **198**	56 / **198**	58 / **121**	57 / **121**	227 / **638**
2-Shapes	281 / **884**		240 / **598**		521 / **1482**
(DL) 2-Shapes	65 / **234**		68 / **143**		133 / **377**
1-Shape	291 / **907**				291 / **907**
(DL) 1-Shape	70 / **245**				70 / **245**

Upon inspecting the counts of ligative bigrams under several models, we found that for a tractable size of a ligative part form, we should base the design on the minimum size possible, i.e. the combined dot-less with 1-Shape model. We chose the BE (or A-Ligatures) as representatives. The BE or A-Ligatures bigrams are standalone bigrams that do not come connected to other characters. Hence, they may be written alone with higher levels of naturalness than any of the other types. However, there are some ligatives that do not have standalone forms. These are inserted into short words in the dataset design using the 1-Shape Dot-less model. In addition, we prepared 12 different ligative forms and diversified the dot-less representatives in these so that each set of 12 forms collectively has a *representation criteria* of the 1-Shape model.

To summarize this part, the design of ligative part of the dataset uses standalone ligatures and short words as the representing unit. Each form alone represents the characters under the dot-less and 1-Shape combined criteria. Every 12 consecutive forms, however, can together achieve representativeness under the 1-Shape criteria (without the dot-less reduction).

B. Design of the Unligative Part

Pangrams are texts that contain all characters of an alphabet and lipograms are writings constrained to avoid certain sets of characters [24]. A representative unligative dataset should cover all character-shapes while avoiding ligatives. So, in a sense, the representativeness problem of the unligative dataset can be formulated as a search for pangram of all character-shapes with lipogram conditions to avoid ligatives.

The counts of necessary characters to satisfy the above conditions are significantly smaller than their ligative counterpart, as the bounds in [4] show. Hence, we could afford to use "sentences" as representing units, the "traditional model" as the representation criteria, and "each form" as a representative. These choices allowed us to increase the level of naturalness in the unligative part while maintaining it reasonably concise.

Under the above conditions, we aimed at finding a minimal set [25] of representative sentences for the unligative dataset. Several character-shape pangrams were created. However, the set of sentences in Fig. 5(a), along with the set of A-character-shapes in Fig. 5(b), were chosen for being the most concise. The separation of the eight character-shapes of Fig. 5(b) reduces the total number of required words by eight, since these shapes can only appear once in a word.

The text contains 43 words with 163 character-shapes, 6 of them combined into three instances of obligatory ligatures. Unfortunately, the two conditions of the unligative text cannot be fully fulfilled due to the need of the inclusion of Omni-ligatives, which cannot be guaranteed not to be written as ligatures by any sequence. Hence, we could only aim at avoiding partio-ligatives.

بلغ حاج أن أخاه ظمآن بوادي عوف طفق يسعى لإحضار ثلاث قرب زمزم تنجيه مع سطوع
وهيج الشمس حث عوض الشيخ نوح بصدد ذلك فأكرمه وصب وتكلف وقال للآت أعظم
ضبط سهيل وأشخاص لص الحي. غش راجح غثامة لذا جن بغيظ وانقض انتهت

(a)

خ س ش ط ظ غ ق ك

(b)

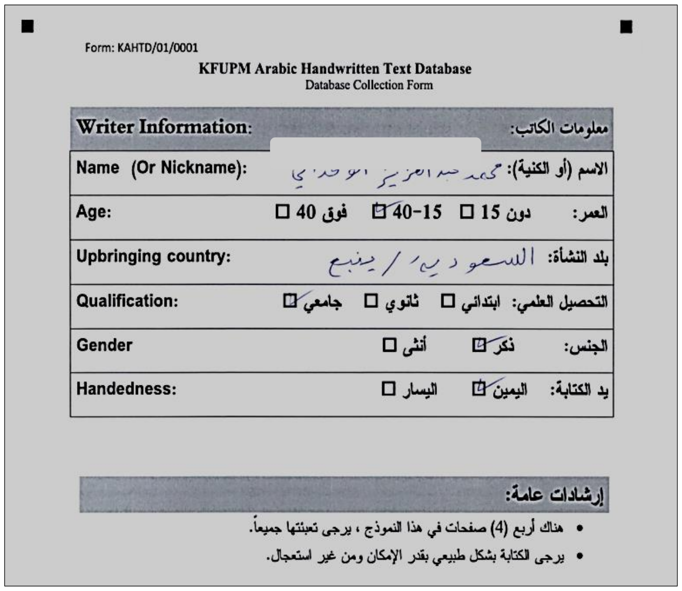
Fig. 5. (a) The Selected text for Collecting the Character-Shapes, and (b) The Complementary Set of Alone Character-Shapes.

V. DATASET IMPLEMENTATION AND COLLECTION

We have designed forms with ligative and unligative parts to collect handwritten samples. Every form is intended to be filled by a distinct and unique writer. The forms were printed, distributed, collected back and analyzed. After discarding incomplete forms, 450 of the collected forms were selected to be scanned at 300 DPI into TIFF colored images.

Parts of the ligative and unligative forms are shown in the figures below. Fig. 6 shows the part where the writers' information is acquired. Fig. 7 depicts the ligative parts grid, with the filling spaces being of equal and vast areas. Each set of 12 of these forms are similar but not identical as discussed earlier. Fig. 8 shows an example of the unligative paragraph along with the corresponding isolated characters.

On all form pages, three aligned filled squares are printed to ease automatic skew detection and correction. The boxes are printed in positions such that their centers of gravity form a right angle with the grids surrounding the content parts of the form. The corner that does not have an aligning box varies depending on the page-number within the form.



Form: KAHTD/01/0001

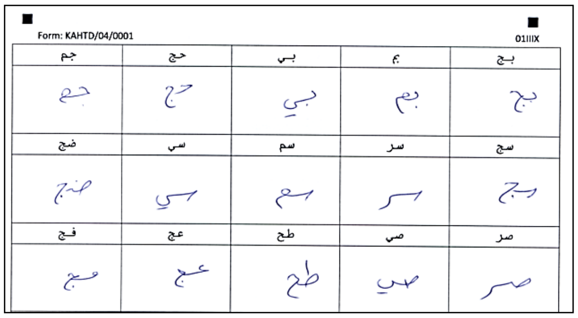
KFUPM Arabic Handwritten Text Database
Database Collection Form

Writer Information:		معلومات الكاتب:
Name (Or Nickname):	محمد عبد العزيز الوحداني	الاسم (أو الكنية):
Age:	☐ فوق 40 [✓ 15-40] ☐ دون 15	العمر:
Upbringing country:	السعودية / ينبع	بلد النشأة:
Qualification:	[✓ جامعي] ☐ ثانوي ☐ ابتدائي	التحصيل العلمي:
Gender	☐ أنثى [✓ ذكر]	الجنس:
Handedness:	☐ اليسار [✓ اليمين]	يد الكتابة:

إرشادات عامة:

- هناك أربع (4) صفحات في هذا النموذج ، يرجى تعبئتها جميعاً.
- يرجى الكتابة بشكل طبيعي بقدر الإمكان ومن غير استعجال.

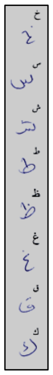
Fig. 6. A Scanned Sample of the Writer Information Collection form.



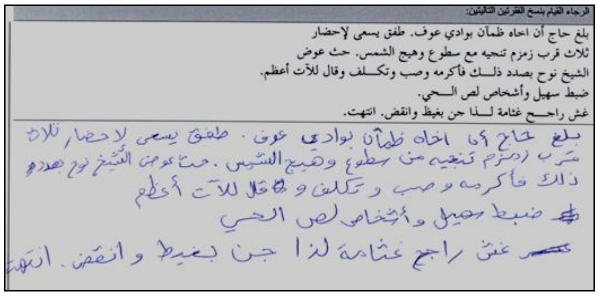
Form: KAHTD/04/0001 01IIIX

جم	حج	بي	بم	بج
ضج	سي	سم	سر	سج
فج	عج	طح	صي	صر

Fig. 7. Two Scanned forms of the Ligative Part of the Dataset.



خ
س
ش
ط
ظ
غ
ق
ك



الرجاء القيام بنسخ الفقرتين التاليتين:

بلغ حاج أن اخاه ظمآن بوادي عوف. طفق يسعى لإحضار
ثلاث قرب زمزم تنجيه مع سطوع وهيج الشمس. حث عوض
الشيخ نوح بصدد ذلــك فأكرمه وصب وتكــلف وقال للآت أعظم.
ضبط سهيل وأشخاص لص الـــحي.
غش راجـــح غثامة لـــذا جن بغيظ وانقض. انتهت.

(a) (b)

Fig. 8. A Scanned Sample of Filled (a) isolated Characters and (b) unligative Parts of a form.

Table IX displays a summary of our ultimate design choices related to the unit-selection and the coverage criteria of the different forms of the dataset. The representation criteria looked at the full range from the most restricted reduction-model (the combined dot-less 1-Shape model) to the traditional model. The representative form ranged from half a page paragraph to 12 pages to be filled by 12 writers. The representing units covered isolated characters, Standalone ligatures and words, and even full sentences and paragraphs.

In Table X, we show some statistical information on the regions, genders, writing-hands, and qualifications of the writers. We consider the following three regions: The Arab Peninsula: containing the Gulf countries, Yemen and Iraq, North Africa: containing Egypt, Sudan, and the countries of Northwest Africa, and Levant: containing Syria, Jordan, Palestine, and Lebanon.

TABLE. IX. UNITS AND COVERAGE CRITERIA OF THE DIFFERENT FORMS OF THE DATASET

Parts	**Representing Unit**	**Representation Criteria**	**Representative Forms**
Ligatures	Standalone ligatures and words	Combined dot-less and 1-Shape	1 form
		1-Shape ligatures	12 forms
Unligative text	Sentences	Traditional (all character-shapes)	1 form
Isolated characters	Isolated Characters		

TABLE. X. NUMBERS OF WRITERS IN THE COLLECTED DATASET PER REGION, GENDER, HANDEDNESS AND QUALIFICATIONS

Region	Arab Peninsula	417
	North Africa	22
	Levant	11
Gender	Male	398
	Female	52
Writing Hand	Right Hand	416
	Left Hand	34
Qualification	Intermediate School	4
	High School	386
	B.Sc. / BA	53
	M.Sc. / Ph.D.	7

VI. EXPERIMENTS AND RESULTS

In this section, we will present the text recognition experiments conducted to show the benefit of incorporating the knowledge of ligatures in dataset design for Arabic. First, we present the datasets used for the text recognition experiments. Then, we describe the text recognition tasks, the measures used for reporting the results, and the used text recognition system. This is followed by the description and discussion of the obtained results.

A. LUCIDAH Dataset

We used a part of the LUCIDAH dataset for training recognition. We selected 10 word images from the dataset which contain Omni-ligatives. Some writers wrote these character pairs as ligatures while others did not. A total of 594 word images were used for training and evaluation. Fig. 9 shows sample words from the LUCIDAH dataset.

B. IFN/ENIT Dataset

The IFN/ENIT dataset [8] consists of handwritten images of Tunisian towns and is one of the most popular publicly available datasets for Arabic text recognition research. The original dataset consists of 32,492 images divided into 5 sets (Sets *a*, *b*, *c*, *d*, and *e*). Later, 10,244 more images were added as Sets *f* and *s*. Fig. 10 shows some sample images from the IFN/ENIT dataset. For the experiments presented here, we will use the standard training-test partitions as reported in the literature (e.g., [26] [27]).

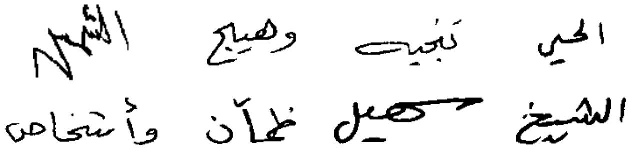


Fig. 9. Sample Text Images from the LUCIDAH Dataset.

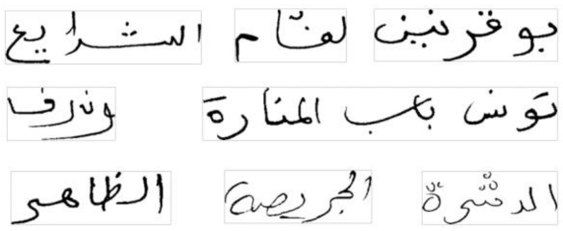


Fig. 10. Sample Text Images from the IFN/ENIT Dataset.

C. Text Recognition Task

We performed character-based recognition after training the system on the text images and two sets of ground-truths: one modeling ligatures and another not modeling them. We used character-based recognition without word dictionaries or language models to adequately show the effects of different modeling choices (ligatures as models versus models with no ligatures). Thus, for the text images, we predict the character sequence using only the character models training without using any n-grams or dictionaries. Accordingly, we report the result in terms of character error-rates (CER):

$$CER\ (\%) = \frac{S + I + D}{N} \times 100$$

where;

S is the error due to character substitution,

D is the error due to character deletion,

I is the error due to character insertion, and

N is the total number of characters in the evaluation set.

In addition to the CER, we also report the statistical significance of the recognition results. Statistical test for the difference of two proportions as presented in [28] is used to report the significance interval of the results at 95% confidence level. This helps us compare the results of two systems with high confidence.

D. Recognition System Description

In this section, we present the details of the system used for training and recognizing the word images. The present system is based on Hidden Markov Models (HMMs). We use the HTK tools [29] to build the recognition system. We, first, preprocess the text images by normalizing the height of the images to 96 pixels while keeping the aspect ratio unchanged for each image. This is followed by the feature extraction stage where sliding windows (of 8 pixels wide and the 96 pixels of height and overlap of 4 pixels) were used. Nine features, that are adapted from the work of Wienecke et al. [30] are extracted from the image segment under the window. In addition, we append 9 more derivative features for each window along the horizontal axis. Thus, the feature vector is of size 18.

To train the system, we build a separate model for each character-shape. A character-shape model consists of a continuous HMM with Bakis topology, thereby, allowing the possibility of skipping the consecutive state during state transitions. Hyper-parameters, like the number of states per model and the number of mixtures per states, were decided based on the performance of the recognizer on the development set.

A multi-step approach was followed for model initialization and training. As a first step, the models were initialized by the flat-start (uniform initialization) approach followed by iterative Baum-Welch training. Then, forced alignment was performed on the training data. The forced aligned data was, in turn, used to reinitialize the models using the Viterbi algorithm. Then, iterations of Baum-Welch training were performed on the reinitialized models. Finally, character recognition was performed using the Viterbi algorithm.

E. Text Recognition on LUCIDAH Dataset

In this section, we will present the experiments and results on the LUCIDAH dataset. As the used part of the dataset for the pilot study is small, the training and evaluation, we performed uniform initialization followed by few iterations of Baum-Welch training instead of the previously described multi-stage training.

The first experiments were conducted without models for ligature. Each character-shape was treated as a model. The training set contained 36 different character-shapes modelled into 36 HMMs. A total of 2,696 character-shape instances were available in the training set.

The second set of experiments was conducted utilizing the ligature models. The pilot dataset was manually investigated for the presence of ligatures. The ligatures were treated as separate models in this set of experiments. The training set encompassed 47 models (36 character-shapes and 11 ligature models). Table XI presents the key characteristics of the training set.

Table XII displays the character-based text recognition results for the two systems in terms of character error-rates (CER). Also, statistical significance at 95% confidence is reported. We can clearly see from the table that the system with the ligature models has significantly lower error-rates as compared to the system without ligature models, for a 95% confidence level. The results were encouraging, but since the pilot study only involved a small dataset, we decided to experiment with the IFN/ENIT dataset, too, as we discuss it in the following section.

F. Text Recognition using IFN/ENIT Dataset

In this section, we present the details of the experiments and the results obtained for character-based text recognition on the IFN/ENIT dataset. We used the standard *train-test* configurations as reported in the literature. System hyper-parameters were calibrated based on an evaluation set (Set *d*) with training sets *a, b, and c* (*i.e.* the *abc-d* configuration).

TABLE. XI. KEY MODEL STATISTICS ON LUCIDAH DATASET

	Number of HMMs	**Average number of samples per model**	**Median number of samples per model**
No Ligature Models	36	75	54
Ligature Models	**47**	**53**	**43**

TABLE. XII. CHARACTER RECOGNITION RATES ON THE LUCIDAH DATASET

System Description	**CER (%)**	**Statistical Significance**
System with no ligature models (baseline)	25.04	±1.40
System with ligature models	**21.52**	

TABLE. XIII. SUMMARY OF CHARACTER-BASED TEXT RECOGNITION RESULTS ON THE IFN/ENIT DATASET

System Description	**CERs (%)**			
	Train–Test Configurations			
	abc–d	**abcd–e**	**abcde–f**	**abcde–s**
System with no ligature models (baseline)	40.65	49.68	46.33	55.12
System with ligature models	**37.49**	**44.81**	**41.69**	**51.64**

Table XIII presents the summary of the text recognition results for the IFN/ENIT dataset using the system without ligatures (second row) and the system with ligatures (bottom row). Significance interval of the errors is ±0.35, ±0.38, ±0.32, and ±0.75 for evaluation sets *d*, *e*, *f*, and *s* respectively at 95% confidence level. We can see from the table that the system having ligature models outperforms the system having no ligature models.

Text recognition results on both datasets confirm that annotating ligatures in handwritten Arabic can enhance their recognition performance. Hence, considering ligatures in dataset design is important for text recognition research. The knowledge of the ligatures can help design datasets which can enable collection of ligature samples from the writers, in addition to its other applications.

VII. CONCLUSIONS

The Arabic script is widely used around the world. Arabic has an inherently cursive script where some character sequences can be replaced by more compact forms called ligatures. If ligatures are not distinctly annotated in datasets, their special forms may cause confusions for automatic text recognition systems. To ease the annotation of ligatures, we design and implement a ligative and unligative dataset for Arabic handwriting, LUCIDAH. Several design decisions taken to balance the representativeness, conciseness, and naturalness requirements of the dataset were presented in this paper. Pilot text recognition experiments were conducted on LUCIDAH and IFN/ENIT to show the significant benefits of annotating ligatures in reducing character recognition errors.

ACKNOWLEDGMENT

The authors thank King Fahd University of Petroleum and Minerals (KFUPM) and Interserve$^{®}$ for their support.

REFERENCES

[1] "Britannica Encyclopedia, Arabic Alphabet," [Online]. Available: https://www.britannica.com/topic/Arabic-alphabet. [Accessed July 2019].

[2] M. Shatnawi and S. Abdallah, "Improving Handwritten Arabic Character Recognition by Modeling Human Handwriting," ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 15, p. Artivle 3, November 2015.

[3] A. Al-Sallab, R. Baly, H. Hazem, S. B. Khaled, W. El-Hajj and G. Badaro, "AROMA: A Recursive Deep Learning Model for Opinion Mining in Arabic as a Low Resource Language," ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 16.4, no. 25, 2017.

[4] Y. Elarian, I. Ahmad, S. Awaida, W. Al-Khatib and A. Zidouri, "Arabic ligatures: Analysis and application in text recognition," in Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), Nancy, 2015.

[5] Wikipedia, "Typographic_Ligature," [Online]. Available: http://en.wikipedia.org/wiki/Typographic_ligature. [Accessed July 2019].

[6] J. Trenkle, A. Gillies, E. Erlandson, S. Schlosser and S. Cavin, "Advnces in Arabic Text Recognition," in in Symposium on Document Image Understanding Technology (SDIUT 2001), Columbia, Maryland, 2001.

[7] S. Schlosser, ERIM Arabic document database, Environmental Research Institue.

[8] M. Pechwitz, S. S. Maddouri, V. Märgner, N. Ellouze and H. Amiri, "IFN/ENIT - Database Of Handwritten Arabic Words," in CIFED, 2002.

[9] Y. Elarian, I. Ahmad, S. Awaida, W. Al-Khatib and A. Zidouri, "An Arabic Handwriting Synthesis System," Pattern Recognition, vol. 48, no. 3, pp. 849-861, 2015a.

[10] M. Elyaakoubi and A. Lazrek, "Justify Just or Just Justify," The Journal of Electronic Publishing, vol. 13, no. 1, 2010.

[11] M. I. Salama, "For the Aesthetics of Arabic Calligraphy," in International Conference for the Aestheitcs of Arabic Calligraphy, Alexandria, 2006.

[12] Unicode Consortium, "Arabic Presentation Forms-A," [Online]. Available: http://www.unicode.org/charts/PDF/U0600.pdf. [Accessed July 2019].

[13] Y. Elarian, A. Zidouri and W. Al-Khatib, "Ground-truth and Metric for the Evaluation of Arabic Handwritten Character Segmentation," in International Conference on Frontiers in Handwriting Recognition, Crete, 2014.

[14] I. Ahmad and G. Fink A., "Class-Based Contextual Modeling for Handwritten Arabic Text Recognition," in Frontiers in Handwriting Recognition (ICFHR), Shenzhen, 2016.

[15] Y. M. Alginahi, "A survey on Arabic Character Segmentation," International Journal on Character Analysis and Recognition (IJDAR), vol. 16, no. 2, pp. 105-126, 2013.

[16] M. Maliki, N. Al-Jawad and S. Jassim, "Sub-word based Arabic handwriting analysis for writer identification," in Proc. SPIE 8755, Mobile Multimedia/Image Processing, Security, and Applications , 2013.

[17] F. Menasri, N. Vincent, E. Augustin and M. Cheriet, "Shape-based Alphabet for Off-line Arabic Handwriting Recognition," in Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on (Volume:2), Parana, 2007.

[18] Y. Haralambous, "The traditional Arabic typecase extended to the Unicode set of glyphs," Electronic Publishing -- Origination, Dissemination and Design (EP-odd), vol. VOL. 8(2 & 3), no. June & September 1995, p. 111–123, 1995.

[19] I. Ahmad and G. A. Fink, "Multi-stage HMM based Arabic text recognition with rescoring," in 13th International Conference on Document Analysis and Recognition (ICDAR), Nancy, 2015.

[20] Y. Elarian and F. Idris, "A lexicon of Connected Components for Arabic Optical Text Recognition," in First International Workshop on Frontiers in Arabic Handwriting Recognition, Istanbul, 2010.

[21] M. Hashem, Artist, The Fundamentals of Arabic Calligraphy: a manuscript collecting the different calligraphy styles of Arabic الخطاط, محمد هاشم قواعد الخط العربي مجموعة خطية لأنواع الخطوط العربية. [Art]. Calligraphy, 1986.

[22] D. Biber, "Representativeness in Corpus Design," Literary and Linguistic Computing, vol. 8, no. 4, pp. 243-257, 1993.

[23] M. Tony, R. Xiao and Y. Tono, "Unit 2 Representativeness, balance and sampling," in Corpus-Based Language Studies, Routledge, 2006.

[24] D. D. Johnson, B. von Hoff Johnson and K. Schlichting, Logology: Word and language play, Guildford Press, 2004.

[25] H. A. Al-Muhtaseb, S. A. Mahmoud and a. R. S. Qahwaji, "A Novel Minimal Script for Arabic Text Recognition Databases and Benchmarks," International Journal of Circuits, Systems and Signal Processing, pp. 145-153, 2009.

[26] V. Margner and H. E. Abed, "Arabic Hnadwriting Recognitoin Competition," in Frontiers in Handwriting Recognition, 2010.

[27] H. El Abed, M. Kherallah, V. Margner and A. M. Alimi, "Online Arabic Handwriting Competition," International Journal on Document Analysis and Recognition (IJDAR), vol. 14(1), pp. 15-23, 2011.

[28] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," Neural computation , vol. 10.7, pp. 1895-1923, 1998.

[29] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. J. Odell, D. Ollason, D. Povey, V. Valtchev and P. Woodland, The HTK Book (for HTK Version 3.2. 1), Cambridge University Engineering Department, 2002.

[30] M. Wienecke, G. A. Fink, and G. Sagerer, "Toward automatic video-based whiteboard reading," International Journal of Document Analysis and Recognition , Vols. 7.2-3 , pp. 188-200, 2005.

Appendix

Appendix I: Groups of Arabic characters in the 4-, 2-, and 1-Shapes Normal (Fig. 11) and Dot-Less (Fig. 12) models. Highlighted character groups cannot be merged due to differences between their (B) and (M) and/or between their (A) and (E) shapes.

4-shapes model				2-shapes model				1-shape model			
(A)	(E)	(M)	(B)	(A)	(E)	(M)	(B)	(A)	(E)	(M)	(B)
ء				ء				ء			
ا	ـا			ا ـا				ا ـا			
أ	ـأ			أ ـأ				أ ـأ			
إ	ـإ			إ ـإ				إ ـإ			
آ	ـآ			آ ـآ				آ ـآ			
ب	ـب	ـبـ	بـ	ب ـب		بـ ـبـ		ب ـب ـبـ بـ			
ت	ـت	ـتـ	تـ	ت ـت		تـ ـتـ		ت ـت ـتـ تـ			
ث	ـث	ـثـ	ثـ	ث ـث		ثـ ـثـ		ث ـث ـثـ ثـ			
ج	ـج	ـجـ	جـ	ج ـج		جـ ـجـ		ج ـج ـجـ جـ			
ح	ـح	ـحـ	حـ	ح ـح		حـ ـحـ		ح ـح ـحـ حـ			
خ	ـخ	ـخـ	خـ	خ ـخ		خـ ـخـ		خ ـخ ـخـ خـ			
د	ـد			د ـد				د ـد			
ذ	ـذ			ذ ـذ				ذ ـذ			
ر	ـر			ر ـر				ر ـر			
ز	ـز			ز ـز				ز ـز			
س	ـس	ـسـ	سـ	س ـس		سـ ـسـ		س ـس ـسـ سـ			
ش	ـش	ـشـ	شـ	ش ـش		شـ ـشـ		ش ـش ـشـ شـ			
ص	ـص	ـصـ	صـ	ص ـص		صـ ـصـ		ص ـص ـصـ صـ			
ض	ـض	ـضـ	ضـ	ض ـض		ضـ ـضـ		ض ـض ـضـ ضـ			
ط	ـط	ـطـ	طـ	ط ـط		طـ ـطـ		ط ـط ـطـ طـ			
ظ	ـظ	ـظـ	ظـ	ظ ـظ		ظـ ـظـ		ظ ـظ ـظـ ظـ			
ع	ـع	ـعـ	عـ	ع	ـع ـعـ		عـ	ع	ـع ـعـ		عـ
غ	ـغ	ـغـ	غـ	غ	ـغ ـغـ		غـ	غ	ـغ ـغـ		غـ
ف	ـف	ـفـ	فـ	ف ـف		فـ ـفـ		ف ـف ـفـ فـ			
ق	ـق	ـقـ	قـ	ق ـق		قـ ـقـ		ق ـق ـقـ قـ			
ك	ـك	ـكـ	كـ	ك ـك		كـ ـكـ		ك ـك		كـ ـكـ	
ل	ـل	ـلـ	لـ	ل ـل		لـ ـلـ		ل ـل ـلـ لـ			
م	ـم	ـمـ	مـ	م ـم		مـ ـمـ		م ـم ـمـ مـ			
ن	ـن	ـنـ	نـ	ن ـن		نـ ـنـ		ن ـن ـنـ نـ			
ه	ـه	ـهـ	هـ	ه	ـه	ـهـ	هـ	ه	ـه	ـهـ	هـ
ة	ـة			ة	ـة			ة	ـة		
و	ـو			و ـو				و ـو			
ؤ	ـؤ			ؤ ـؤ				ؤ ـؤ			
ي	ـي	ـيـ	يـ	ي ـي		يـ ـيـ		ي ـي		يـ ـيـ	
ى	ـى			ى ـى				ى ـى			
ئ	ـئ	ـئـ	ئـ	ئ ـئ		ئـ ـئـ		ئ ـئ		ئـ ـئـ	

Fig. 11. Character Groups in the 4-, 2-, and 1-Shape(s) Models.

4-Shape Dot-less Model

(A)	*(E)*	*(M)*	*(B)*
ء			
ا أ إ آ	ـا ـأ ـإ ـآ		
ب ت ث	ـب ـت ـث	ـبـ ـتـ ـثـ ـنـ ـيـ ـئـ	بـ تـ ثـ نـ يـ ئـ
ن	ـن		
ي ئ ى	ـي ـئ ـى		
ج ح خ	ـج ـح ـخ	ـجـ ـحـ ـخـ	جـ حـ خـ
د ذ	ـد ـذ		
ر ز	ـر ـز		
س ش	ـس ـش	ـسـ ـشـ	سـ شـ
ص ض	ـص ـض	ـصـ ـضـ	صـ ضـ
ط ظ	ـط ـظ	ـطـ ـظـ	طـ ظـ
ع غ	ـع ـغ	ـعـ ـغـ	عـ غـ
ف	ـف	ـفـ ـقـ	فـ قـ
ق	ـق		
ك	ـك	ـكـ	كـ
ل	ـل	ـلـ	لـ
م	ـم	ـمـ	مـ
ه ة	ـه ـة	ـهـ	هـ
و ؤ	ـو ـؤ		

2-Shapes Dot-less Model

(A)	*(E)*	*(M)*	*(B)*
ء			
ـا ا ـأ أ ـإ إ ـآ آ			
ـب ب ـت ت ـث ث		بـ ـبـ تـ ـتـ ثـ ـثـ نـ ـنـ يـ ـيـ ئـ ـئـ	
ن ـن			
ـي ي ـئ ئ ـى ى			
ـج ج ـح ح ـخ خ		جـ ـجـ حـ ـحـ خـ ـخـ	
ـد د ـذ ذ			
ـر ر ـز ز			
ـس س ـش ش		سـ ـسـ شـ ـشـ	
ـص ص ـض ض		صـ ـصـ ضـ ـضـ	
ـط ط ـظ ظ		طـ ـطـ ظـ ـظـ	
ع غ	ـع ـغ	ـعـ ـغـ	عـ غـ
ـف ف		فـ ـفـ قـ ـقـ	
ـق ق			
ـك ك		كـ ـكـ	
ـل ل		لـ ـلـ	
ـم م		مـ ـمـ	
ه ة	ـه ـة	ـهـ	هـ
و ـو ؤ ـؤ			

hapes Dot-less Model

(A)	*(E)*	*(M)*	*(B)*
ء			
ا ـا أ ـأ إ ـإ آ ـآ			
ب ـب ت ـت ث ـث		بـ ـبـ تـ ـتـ ثـ ـثـ نـ ـنـ يـ ـيـ ئـ ـئـ	
ن ـن			
ي ـي ئ ـئ ى ـى			
جـ ـجـ حـ ـحـ خـ ـخـ ج ـج ح ـح خ ـخ			
د ـد ذ ـذ			
ر ـر ز ـز			
سـ ـسـ شـ ـشـ س ـس ش ـش			
صـ ـصـ ضـ ـضـ ص ـص ض ـض			
طـ ـطـ ظـ ـظـ ط ـط ظ ـظ			
ع غ	ـغـ ـغ ـعـ ـع		عـ غـ
ق ـق ف ـف			فـ ـفـ قـ ـقـ
ك ـك		كـ ـكـ	
ل ـل		لـ ـلـ	
م ـم		مـ ـمـ	
ه ة	ـه ـة	ـهـ	هـ
و ـو ؤ ـؤ			

Fig. 12. Character Groups in the 4-, 2-, and 1-Shape(s) Dotless Models.