Impact of Character Models Choice on Arabic Text Recognition Performance

Fouad SLIMANE[1,2], Rolf INGOLD[1], Slim KANOUN[2], Adel M. ALIMI[2], Jean HENNEBERT[1,3]
[1] *DIVA Group, Department of Informatics, University of Fribourg, Fribourg, Switzerland*
[2] *REsearch Group on Intelligent Machines (REGIM), ENIS, University of Sfax, Sfax, Tunisia*
[3] *Business Information Systems Institute, HES-SO // Wallis, Sierre, Switzerland*
Fouad.Slimane@unifr.ch, Rolf.Ingold@unifr.ch, Slim.Kanoun@enis.rnu.tn, Adel.Alimi@enis.rnu.tn, Jean.Hennebert@hevs.ch

Abstract

We analyze in this paper the impact of sub-models choice for automatic Arabic printed text recognition based on Hidden Markov Models (HMM). In our approach, sub-models correspond to characters shapes assembled to compose words models. One of the peculiarities of Arabic writing is to present various character shapes according to their position in the word. With 28 basic characters, there are over 120 different shapes. Ideally, there should be one sub-model for each different shape. However, some shapes are less frequent than others and, as training databases are finite, the learning process leads to less reliable models for the infrequent shapes. We show in this paper that an optimal set of models has then to be found looking for the trade-off between having more models capturing the intricacies of shapes and grouping the models of similar shapes with other. We propose in this paper different sets of sub-models that have been evaluated using the Arabic Printed Text Image (APTI) Database freely available for the scientific community.

1. Introduction

Research in automatic recognition of Arabic script dates back to the 70s. During the past two decades, the increase of the reliability of Arabic handwritten and printed text recognition was noticeable. The growing availability of evaluation databases [14] [10] and the organization of competitions [7] [8], have contributed to systematic comparisons of different recognition systems for the benefit of their improvement.

Most traditional approaches for Arabic printed text recognition are based on a priori segmentation of lines into words and characters. These solutions were based on systems developed for Latin and Chinese text recognition where the segmentation is possible by a natural separation of characters. This step is very difficult for Arabic due to its cursive or semi-cursive representation in its printed and handwritten form. Performing an a priori segmentation is even more difficult when images are on low resolution. To overcome this segmentation problem, some specific approaches have been proposed:

- Pseudo-global approach based on the concept of pseudo-word [9];
- Analytical approaches that integrate the morphological vocabulary structure in a post-processing step to validate words hypothesis [2] [12];
- Affixal approach using linguistic constraints on segments with prefix, suffix, infix and root rules [5];
- Neurolinguistic approach modeling linguistic concepts of vocabulary with neural networks [3].

More recently, stochastic approaches based on Hidden Markov Models (HMMs) have gained momentum thanks to their ability to perform an implicit segmentation while recognizing character shapes [1] [6] [9][13] [15] [17] [18]. These approaches are known in other areas like speech recognition and cursive Latin text recognition. Another advantage of HMMs is in the hierarchical approach of the modeling. Starting from sub-models corresponding to characters, word models and sentence models can be recomposed, allowing for the inclusion of so-called language models through dictionaries, finite-state grammars or stochastic grammars.

Further to this segmentation difficulty, another important peculiarity of Arabic script in comparison to

12th International Conference on Frontiers in Handwriting Recognition, ICFHR 2010. 2010. pp. 670-675

other languages is in the large variability of character shapes in the alphabet. First, the shapes vary depending on their position in the word and according to the used font. Secondly, the shapes can be generated with ligature or overlaps between characters such as in characters *Laam* and *Alif*. With 28 basic characters, there are over 120 different shapes, most of them slightly differing from the basis shape. From a pattern matching point of view, one model should be used for each different shape. However, in practice, training databases are finite and some shapes are much less frequent than others.

In the framework of HMM systems, we address in this paper the following question: is there an optimal set of HMM sub-models for Arabic recognition given a training database? In other words, does a trade-off exists between having more models capturing the intricacies of shapes and optimizing model training by grouping similar shapes with other. The aim of this paper is not to present a new Arabic recognition system but to present the impact of the choice of Arabic character shapes sub-models on system performance. We performed all our evaluations using the large Arabic Printed Text Image (APTI) Database freely available for the scientific community.

This paper is organized as follows. In Section 2, we present the most important characteristics of Arabic script. In Section 3, we describe our HMM based systems and the different clusters of character shapes we propose to evaluate. Section 4 is dedicated to the word images database we used for the evaluation. Finally, we discuss the obtained results in Section 5.

2. Characteristics of the Arabic script

With a quite large user base of about 300 million people worldwide, Arabic is very important in the culture of many people. Compared to printed Latin script, we can underline several important differences:

- Arabic is written from right to left;
- It is semi-cursive whether printed or handwritten. Each character has a connection point right and/or left linked on baseline;
- The concept of uppercase and lowercase in Arabic script does not exist;
- The Arabic alphabet is richer than its Latin equivalent. It contains 28 letters, most of which change shape according to their appearance at the beginning, middle or end of the word.

Table 1 shows the 28 Arabic letters with their different shapes according to their position in the word. Letters having just two kinds of appearances cannot be connected to the following letter, meaning that their "begin" shapes are simply their "isolated" shapes and their "middle" shapes are their "end" shapes. Taking into account the different shapes of Arabic characters, the different shapes increase from 28 to 100.

Table 1. Arabic letters

Letter label	Isolated	Begin	Middle	End
Alif	ا		ـا	
Baa	ب	بـ	ـبـ	ـب
Taaa	ت	تـ	ـتـ	ـت
Thaa	ث	ثـ	ـثـ	ـث
Jiim	ج	جـ	ـجـ	ـج
Haaa	ح	حـ	ـحـ	ـح
Xaa	خ	خـ	ـخـ	ـخ
Daal	د		ـد	
Thaal	ذ		ـذ	
Raa	ر		ـر	
Zaay	ز		ـز	
Siin	س	سـ	ـسـ	ـس
Shiin	ش	شـ	ـشـ	ـش
Saad	ص	صـ	ـصـ	ـص
Daad	ض	ضـ	ـضـ	ـض
Thaaa	ط	طـ	ـطـ	ـط
Taa	ظ	ظـ	ـظـ	ـظ
Ayn	ع	عـ	ـعـ	ـع
Ghayn	غ	غـ	ـغـ	ـغ
Faa	ف	فـ	ـفـ	ـف
Gaaf	ق	قـ	ـقـ	ـق
Kaaf	ك	كـ	ـكـ	ـك
Laam	ل	لـ	ـلـ	ـل
Miim	م	مـ	ـمـ	ـم
Nuun	ن	نـ	ـنـ	ـن
Haa	ه	هـ	ـهـ	ـه
Waaw	و		ـو	
Yaa	ي	يـ	ـيـ	ـي

In addition to this "positioning" variability, there are extra characters appearing as variations of some basic characters (see Table 2). The "ة (*TaaaClosed*)" is the same character "ت (*Taaa*)", but it can be used just in the end of Arabic names and cannot be used in verbs. Other characters are created by combination of "ء (*Hamza*)" – "ا (*Alif*)" or "ء (*Hamza*)" – "و (*Waaw*)". They are almost pronounced the same way but their use depends on their position in the word. Adding these characters to base classes, presented in Table 1, the different shapes number increases to 118.

Table 2. Additional Characters

Letter label	Isolated	Begin	Middle	End
HamzaAboveAlif	أ		أ	
TildAboveAlif	آ		آ	
HamzaUnderAlif	إ		إ	
HamzaAboveWaaw	ؤ		ؤ	
HamzaAboveAlifBroken	ئ	ئ	ئ	ئ
AlifBroken	ى			ى
Hamza	ء		ء	
TaaaClosed	ة			ة

Table 3 presents two examples of additional characters created with a succession of diacritic "ّ (*Chadda*)" and character "ن (*Noun*)" or "ي (*Yaa*)". The "(*Chadda*)" can also be combined with most other Arab characters, to create other shapes.

Table 3. Character shapes created by composition (*Yaa* or *Nuun* + *Chadda*) (*: characters not present in evaluation database)

Letter label	Isolated	Begin	Middle	End
YaaChadda	يّ	يّ	يّ	يّ
NuunChadda	نّ	نّ(*)	نّ(*)	نّ

Table 4 shows the variations of character "لا (*LaamAlif*)" created when the character Alif follows the character *Laam* in the word. With all these presented variation of Arabic character shapes, the total number go up to 134.

Table 4. Character shapes created by composition (*Alif*, *HamzaAboveAlif*, *HamzaUnderAlif* or *TildAboveAlif* + *Laam*)

Letter label	Isolated	Begin	Middle	End
LaamAlif	لا		لا	
LaamHamzaAboveAlif	لأ		لأ	
LaamHamzaUnderAlif	لإ		لإ	
LaamTildAboveAlif	لآ		لآ	

3. System developed

3.1. General characteristics

Our recognition system is based on HMMs (see for example [11] for an introduction). The system used in this paper has a similar architecture to the one presented in [13] and is inspired from the work in [4]. One of its main characteristics is to be open vocabulary, i.e. able to recognize any Arabic printed word. Currently, the system is constrained to a configuration (font, size, style), as described in Section 4, but our future work will aim to extend for multi-font, multi-size and multi-style Arabic printed text.

The system is developed using the toolkit HTK (Hidden Markov Models Toolkit)[19]. It works in two phases: learning and recognition. In both phases, we extract the same features. Each word image is transformed into binary image and a sequence X of feature vectors x_n is extracted from a narrow window of 8 pixels width shifted one pixel from right to left. Each feature vector x_n has 102 components: 51 features and the computation of so-called delta coefficients between adjacent vectors using the following formula:

$$\Delta x_n^j = x_{n+1}^j - x_{n-1}^j, \forall 1 < j < 51$$

$$\Delta x_n^j = x_n \quad where \quad n = 0 \quad or \quad n = N$$

For a full description of the features, see [13]. Among the features we use the number of connected black and white components, the gravity center, density, compactness, vertical and horizontal projection, baseline position, the number of relative extrema in the vertical projection, the number of relative extrema in the horizontal projection, etc.

Regarding the HMM topology, we use for all sub-models an equal length of 5 states. While it seems a priori sub-optimal against variable length topologies, we have shown in our previous work that using equal length of states gives consistently good performances [13][15].

During training time, the Expectation-Maximization (EM) algorithm is used to iteratively refines the component weights, means and variances to monotonically increase the likelihood of the training feature vectors [20]. In our experiments we used the EM algorithm to build the models by applying a simple binary splitting procedure to increase the number of Gaussian mixtures through the training procedure up to 512 mixtures.

At recognition time, an ergodic HMM including all sub-models is built and the best path in this model simply determines the winner sub-models sequence using the standard Viterbi decoding procedures available in HTK. Performances are evaluated in terms of word recognition rates using an unseen set of word images.

3.2. System Optimization

Compared to our previous work in [13], some optimizations of the system are introduced to consolidate its overall performance. First, a general optimization of various parameters of HMMs is done:

analysis window size, number of Gaussian mixture per state, number of training iterations, etc. Second, we introduce delta coefficients in the feature vector as explained above Finally, we perform some basic post-processing on the recognized sequence of character by including simple linguistic post-processing rules. Firstly, We keep just the name of sub-models and we remove the information about the position and secondly, since some Arabic characters can never followed (For example, *Alif Alif* or *Saad Siin*), we use this information to correct the minority of these errors generated by our system.

3.3. Arabic sub-models choice

The purpose of this work is to show that using different sets of sub-models has a large influence on system performance. We experimented with 10 systems trained and tested with the same data and using the same configuration as described above. The differences between these systems are in the quantity of sub-models used, ranging from 35 to 124.

Sys_120 Taking into account all Arabic character shapes presented in Tables 1, 2 and 3, we used a set of 120 sub-models. Such a set actually corresponds to the work presented in [1].

Sys_124 This set is obtained adding the 4 extra shapes of Table 4.

Sys_64 Starting from Sys_120, we grouped similar character shapes into 64 models according to the following rules: (1) beginning and middle shapes share the same model (2) end and isolated shapes share the same model. These rules apply for all characters with an exception for characters *Ayn* and *ghayn* where beginning, middle, end and isolated shapes are very different. This strategy of grouping is natural as beginning-middle and end-isolated character shapes are visually similar. Such grouping can also be found in related work [13].

Sys_38 We used here one single model for each characters of Table 1, 2 and 3, independently of their position.

Sys_68 Starting from Sys_64, we included 4 extra models for each characters of Table 4. Our motivation is here in the inherent difficulty to segment shapes of characters in Table 4, leading to frequently observed errors. For example, we consider *LaamAlif* as a new character shape to model the sequence *Laam* followed by *Alif*.

Sys_62 Starting from Sys_68 models, we operated some groupings of visually similar shapes. Two models are created with the following groups: {*HamzaAboveAlif_I, TildAboveAlif_I, Alif_I, HamzaUnderAlif_I}* and {*LaamHamzaAboveAlif_I, LaamTildAboveAlif_I, LaamHamzaUnderAlif_I, LaamAlif_I }*.

Sys_61 Starting from Sys_64, we operate the grouping of visually the following similar shapes {*HamzaAboveAlif_I, TildAboveAlif_I, Alif_I* and *HamzaUnderAlif_I*} into one shared model.

Sys_42 We used Sys_38 models and added 4 models for the character shapes presented in Table 4.

Sys_36 Using Sys_42 models, we operate the same grouping as for Sys_62.

Sys_35 Using Sys_38 models, we operate the same grouping as for Sys_61.

In table 5, we illustrate a result example of all systems.

Table 5. Example of word recognition with the various systems proposed

Image to recognize		الأهالي
	System output	*Arabic Transcription*
Sys_124	Alif_I LaamTildAboveAlif_I Haa_B Alif_E Laam_B Yaa_E	الأهـالي
Sys_120	Alif_I Laam_E TildAboveAlif_E Haa_B Alif_E Laam_B Yaa_E	الأهـالي
Sys_68	Alif_I LaamTildAboveAlif_I Haa_B Alif_I Laam_B Yaa_I	الأهـالي
Sys_64	Alif_I Laam_I TildAboveAlif_I Haa_B Alif_I Laam_B Yaa_I	الأهـالي
Sys_62	Alif_I LaamAlif_I Haa_B Alif_I Laam_B Yaa_I	الأهـالي
Sys_61	Alif_I Laam_I Alif_I Haa_B Alif_I Laam_B Yaa_I	الأهـالي
Sys_42	Alif LaamTildAboveAlif Haa Alif Laam Yaa	الأهـالي
Sys_38	Alif Laam TildAboveAlif Haa Alif Laam Yaa	الأهـالي
Sys_36	Alif LaamAlif Haa Alif Laam Yaa	الأهـالي
Sys_35	Alif Laam Alif Haa Alif Laam Yaa	الأهـالي

4. APTI database

To evaluate our systems, we used parts of the large Arabic Printed Text Image (APTI) Database [14]. APTI is freely available to the scientific community[1]. Images in APTI are synthetically created in low-resolution “72 dots/inch” with a lexicon of 113,284

[1] http://diuf.unifr.ch/diva/APTI/

different Arabic words, 10 fonts, 4 styles and 10 different sizes. It contains more than 45 million Arabic word images representing more than 250 million different character shapes. Each word image in APTI is fully described using an XML file containing ground truth information about the generation process and the sequence of characters. APTI is designed with the objective of analyzing the impact of multi-font, multi-size, multi-style variability on recognition systems with large quantities of data. The low-resolution nature of most of the data is interesting in the context of, for example, screen-based OCR.

APTI is built from texts taken from a variety of sources such as books, articles, web pages. Characters are then distributed according to real-life content with no focus on specific thematic. 120 labels were used in APTI to describe characters, taking into account their positions (beginning, middle, end, isolated). APTI is divided into 6 sets, 5 of which are freely available to the scientific community and one kept for future evaluations. The sets have been designed so that the number of words and representation of letters are very close from set to set (for more details about data dispersion, see [16]).

In our tests, we used the 5 available sets of APTI generated with font "*Arabic Transparent*", size "*24*" and style "*Plain*". 75,750 images (set 1, 2, 3 and 4) are used for the training phase and an additional 18,868 (set 5) different images were used for the test phase.

5. Experimental results

Results of the systems presented in Section 3 are summarized in Table 6,. All recognition rates are calculated using the character labels, without taking into account the positioning information. So, if the system recognizes *Alif_I* or *Alif_E*, it is automatically transformed in the label *Alif* to calculate the recognition rate.

Table 6. Result Systems

	Group 1				**Group 2**				**Group 3**	
System	**35**	**36**	**38**	**42**	**61**	**62**	**64**	**68**	**120**	**124**
Word RR	85.1	85.5	84.6	84.3	96.5	96.2	94.3	94.5	89.9	92.3
Character RR	99.2	99.2	99.2	99.1	99.7	99.7	99.7	99.7	99.6	99.7

Results have been dispatched in three groups in Table 6. Roughly speaking, Group 1 corresponds to systems where a single sub-model per character is used, independently of the variations of shapes due to the positioning of characters. Group 2 corresponds to using two sub-models per characters, one for the beginning-middle shapes and one for the end-isolated shapes. From Group 1 to Group 2, we basically double the number of sub-models. Group 3 corresponds to using as many sub-models as there are different shapes due to the positioning of characters. From Group 2 to Group 3, we double the number of sub-models for most characters.

Within a group, we perform some variations of the set of sub-models, including or not sub-models corresponding to characters of Table 4 or grouping similar shapes as explained, for example, for Sys_62 or Sys_61 in Section 3.2.

We can observe some interesting trends comparing results from one group to another. We see a significant increase of performance by doubling the amount of sub-models going from Group 1 to Group 2, gaining on average about 10% of recognition rate. It seems beneficial to model more finely the differences of shapes between beginning-middle shapes and end-isolated shapes. On the other hand, performances are decreasing of 2% to 6% going from Group 2 to Group 3. From a pattern recognition point of view, this result is counter-intuitive, as we should gain more precision of the modeling using more sub-models. However, we very probably see here the effect of having too few training data for less frequent representations of some character shapes. For example, character *HamzaAboveAlifBroken* in position end is represented with only 32 occurrences in the database. A visual inspection of the errors is actually supporting this statement where frequent errors are related to less frequent shapes in the training database.

Now if we compare results within a group, for example in Group 2, we observe that Sys_61 and Sys_62 are leading to the best performances. These systems are the one where we operate further grouping of similar shapes. The reason is also to be found in the frequency of occurrences of some character shapes which is apparently too low and where shared models lead to increased performances.

The inclusion of sub-models for characters in Table 4 leads to an increase of performance from Sys_120 to Sys_124. However, their inclusion in Group 1 and 2 is less convincing, leading to similar or slightly decreasing performances.

Overall, our best systems is Sys_61 in Group 2 with a word recognition rate of 96.5% and a character recognition rate of 99.7% .

6. Conclusion and future works

Results obtained by several research groups are showing that HMM based systems are well-suited for the recognition of Arabic printed text. Their main

advantage in the context of Arabic printed text is their ability to cope with the continuous nature of Arabic text where characters are connected to each other and where an a priori segmentation is difficult to realize. Another advantage of HMMs is in the hierarchical approach of the modeling where sub-models corresponding to characters are composed together to form word and sentence models.

In the framework of such HMM systems, we have addressed in this paper the question of the optimal set of sub-model to use in a given context. This question is related to the nature of Arabic characters that present varying shapes according to their positions in words. Ultimately, with an infinite set of training data, the answer would be to have one sub-model for each variation of characters. However, results of this paper are clearly showing that for a limited training set, there is a tradeoff between precision and reliability of sub-models. Precision is increased when using more sub-models to capture the different shapes of a given character. Reliability is decreased if not enough training data is available to train such sub-models.

The results of this paper are also encouraging future works in several directions. First, dynamic training scheme could be investigated where sub-models would be instantiated as soon as enough training data are available. Another potential direction would be to go for so-called adaptation training where sub-models trained on large quantity of data would be adapted to less frequent characters. For example, we could use a Maximum A Posteriori adaptation instead of the Expectation Maximization training used in this paper.

References

[1] H. A. Al-Muhtaseb, S. A. Mahmoud, and R. S. Qahwaji. Recognition of off-line printed Arabic text using Hidden Markov Models, *Signal Process*, 88(12):2902-2912, 2008.

[2] A. Amin and S. Al-Fedaghi. Machine recognition of printed arabic text utilizing natural language morphology. *Int. J. Man-Mach. Stud.*, 35(6):769–788, 1991.

[3] I. Ben Cheikh, A. Belaid, and A. Kacem. A novel approach for the recognition of a wide arabic handwritten word lexicon. *ICPR*, pages 1–4, December 2008.

[4] F. Einsele, R. Ingold, and J. Hennebert. A language independent, open-vocabulary system based omn hmms for recognition of ultra low resolution words. *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 429–433, 2008.

[5] S. Kanoun, A. M. Alimi, and Y. Lecourtier. Affixal approach for Arabic decomposable vocabulary recognition: A validation on printed word in only one font. *ICDAR*, pages 1025–1029, 2005.

[6] M. S. Khorsheed. Offline recognition of omnifont Arabic text using the hmm toolkit (htk). *PRL*, 28(12):1563–1571, 2007.

[7] V. Margner and H. El Abed. Icdar 2009 arabic handwriting recognition competition. *ICDAR*, pages 1383–1387, 2009.

[8] V. Margner and H. El Abed. Icdar 2009 arabic handwriting recognition competition. *ICDAR*, pages 1383–1387, 2009.

[9] H. Miled and N. Essoukri Ben Amara. Planar markov modeling for Arabic writing recognition: Advancement state. *ICDAR*, 0:0069, 2001.

[10] M. Pechwitz, S. Snoussi Maddouri, V. Margner, N. Ellouze, and H. Amiri. Ifn/enit - database of handwritten arabic words. *In Proc. of CIFED*, pages 129–136, 2002.

[11] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice Hall PTR, United States, 1993.

[12] T. Sari and M. Sellami. Morpho-lexical analysis for correcting ocr-generated arabic words (molex). *IWFHR,* pages 461-466, 2002.

[13] F. Slimane, R. Ingold, A. M. Alimi, and J. Hennebert. Duration models for arabic text recognition using hidden markov models. *CIMCA*, 0:838–843, 2008.

[14] F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi, and J. Hennebert. A new arabic printed text image database and evaluation protocols. *ICDAR*, pages 946–950, July 2009.

[15] F. Slimane, S. Kanoun, A. M. Alimi, J. Hennebert, and R. Ingold. Modèles de markov cachés et modèle de longueur pour la reconnaissance de l'écriture arabe basse résolution. *MajecSTIC*, 2009.

[16] F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi, and J. Hennebert. Database and evaluation protocols for Arabic printed text recognition. *Internal Publication, DIUF, University of Fribourg, Switzerland*, 2009.

[17] M. S. Khorsheed. Recognising handwritten Arabic manuscripts using a single hidden Markov model, *PRL*, v.24 n.14, October 2003, pp.2235-2242.

[18] M.C. Fehri. Reconnaissance de textes arabes mutifonte à l'aide d'une approche hybride neuro-markoviennes. Thèse de doctorat, Université des sciences, des techniques et de médecine de Tunis II, Tunisie, 1999.

[19] S. Young, G. Evermann, D. Kershaw, D. Moore, J. Odell, D. Ollason, V. Valtchev, P. Woodland. The HTK Book, *Cambridge University Engineering Dept.*, 2001.

[20] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Royal Statistical Society*, 39(1):1–38, 1977.