Comparison of Global and Cascading Recognition Systems Applied to Multi-font Arabic Text

Fouad Slimane[1,2], Slim Kanoun[2], Adel M. Alimi[2], Jean Hennebert[1,3] and Rolf Ingold[1]
[1]DIVA: Document, Image and Voice Analysis research group, Department of Informatics
University of Fribourg (unifr), Bd de Pérolles 90, CH-1700 Fribourg, Switzerland
[2]REGIM: REsearch Group on Intelligent Machines
University of Sfax, National School of Engineers (ENIS), BP 1173, Sfax, 3038, Tunisia
[3]Business Information System Institute, HES-SO // Wallis, Sierre,Switzerland
Fouad.Slimane@unifr.ch, Slim.Kanoun@yahoo.fr, Adel.Alimi@ieee.org,
Jean.Hennebert@hevs.ch, Rolf.Ingold@unifr.ch

ABSTRACT

A known difficulty of Arabic text recognition is in the large variability of printed representation from one font to the other. In this paper, we present a comparative study between two strategies for the recognition of multi-font Arabic text. The first strategy is to use a *global* recognition system working independently on all the fonts. The second strategy is to use a so-called *cascade* built from a font identification system followed by font-dependent systems. In order to reach a fair comparison, the feature extraction and the modeling algorithms based on HMMs are kept as similar as possible between both approaches. The evaluation is carried out on the large and publicly available APTI (Arabic Printed Text Image) database with 10 different fonts. The results are showing a clear advantage of performance for the cascading approach. However, the cascading system is more costly in terms of cpu and memory.

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: Text processing; I.7.5 [**Document Capture**]: Optical character recognition (OCR)

General Terms

Measurement, Performance, Theory

Keywords

APTI, font recognition, text recognition, GMM, HMM

1. INTRODUCTION

Over the past ten years, the performances of Arabic recognition systems have been improving significantly. The growing availability of benchmarking databases [14, 11, 4] and the organization of competitions [9, 8, 1], have contributed to systematic comparisons of various strategies for the benefit of their improvement.

Some related work focus on specific and/or limited Arabic vocabulary such as the recognition of handwritten Arabic cheques [3], the recognition of handwritten Tunisian town/village names [9] or the recognition of printed decomposable word [5]. Often, these approaches are based on an a priori segmentation of lines into words and characters or fragment of characters. A priori segmentation of Arabic text is very difficult due to its cursive or semi-cursive representation which exist in handwritten as well as in printed text. With such approaches, the segmentation is not only adding extra complexity, it is also introducing errors in early stages of the recognition system. To handle the difficulties of a priori segmenting the Arabic script, several researchers have proposed to use Hidden Markov Models (HMMs) [2, 7, 10, 12, 6] able to achieve segmentation and character recognition in a continued way. 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 or stochastic grammars. Finally, HMMs, through their emission probability density functions, are also robust in front of variabilities of the observations, which are in the case of Arabic, quite numerous.

An important peculiarity of Arabic script in comparison to other languages is indeed in the large variability of character shapes of the alphabet. Firstly, the shapes vary depending on their position in the word. Secondly, the shapes can be generated with ligature or characters overlaps such as for *Laam* "ﻟ" and *Alif* "ﺍ": *LaamAlif* "ﻻ". Thirdly, shapes, ligatures and overlap vary according to the font. From 28 basis characters, there are over 120 different shapes, most of them differing slightly from the corresponding basis shape. Another important source of variability in Arabic text is in the font-to-font variability. Ten of the mostly used fonts are illustrated in Figure 2, showing large differences in character shapes, ligatures and overlaps. Further to these intrinsic variabilities, the image generation and acquisition procedures will also add noise and variabilities to the signals. In this paper, we focus on images that are generated for screen display, web images or pdf rendering where the noise is mainly coming from the downsampling grid alignment and anti-aliasing filter. Such images are found in applications

DocEng2010, September 21–24, 2010, Manchester, United Kingdom.

such as *screen-based OCR* nowadays available for latin languages, but not yet covered by Arabic OCR.

We propose in this paper to present a comparative study between two strategies for the recognition of multi-font Arabic text. The first strategy is to use a *global* recognition system working independently on all the fonts. The second strategy is to use a so-called *cascade* built as a two steps system cascading font identification and mono-font text recognition. in other words, we convert the multi-font problem into several mono-font recognition problems.

Our off-line multi-font Arabic printed text OCR works in open vocabulary mode. By open vocabulary, we refer here to a system that can recognize any arbitrary word written in Arabic script, without limitations on the vocabulary size. The system is built and evaluated using the widely used Hidden Markov Model Toolkit (HTK) [18].

The paper is organized as follows. In section 2, we present the different steps and systems used to develop the proposed screen based OCR. Section 3 is dedicated to the word images database we used for the evaluation and results, and are followed by conclusions and future works.

2. SYSTEM DESCRIPTION

As illustrated in Figure 1, we compare two systems. The first one is a cascading system working in three steps: feature extraction, font recognition and word recognition using font-dependent models. The second one is a global multi-font system working in two steps: feature extraction and word recognition using font-independent models. Both systems share the same feature extraction frontend.

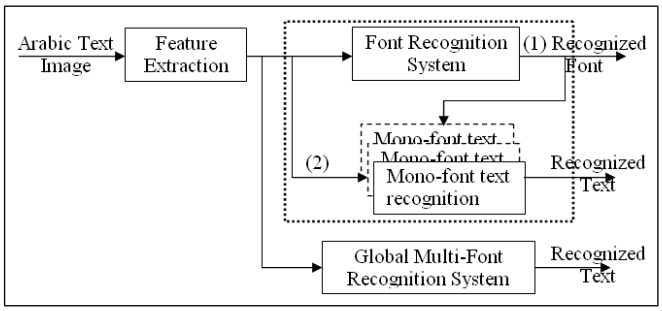

Figure 1: Global and cascading screen-based OCR systems.

2.1 Pre-processing and Feature Extraction

Each word image is normalized in gray-level into a rectangle with fixed height and then transformed into a sequence of feature vectors computed from a narrow analysis window, sliding from right to left on the word image. In our settings, at each step the uniform analysis window is shifted by 1 pixel. We performed several tests to determine the optimal size of the window and the normalized height to maximize the recognition rate, the best parameters we obtained are 45 pixels for the height and 8 pixels for the window width.

For a full description of the features used in our approach, we refer to [16]. We extract features such as 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. A feature vector x is extracted from each analysis window. Since no segmentation into letters is made, the word image is transformed into a sequence X of N feature vectors x_n. Each feature vector x_n has 102 components including 51 basis features concatenated with 51 so-called delta coefficients computed as a linear difference of the basis features in adjacent windows. The delta are computed in a similar way as in speech recognition, to include larger contextual information in an analysis window.

2.2 Global HMM Based Recognition System

Our HMM sub-models correspond to arabic characters and a selected set of their corresponding variations [15]. Similar character shapes are grouped into 64 models according to the following rules: (1) beginning and middle shapes share the same model, (2) isolated and end shapes share the same model. These rules apply for all characters with an exception for characters *Ayn* "ع" and *ghayn* "غ" 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 [12].

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 [12, 17]. In our settings, each state of the HMM computes the emission probability of features with a mixture of 512 Gaussians. In our previous experiments, this quantity has proven to be adequate considering the size of the training database and the complexity of probability density functions to estimate.

We used the Hidden Markov model Toolkit (HTK) to realize our evaluation [18]. HTK is a set of command line executables used for initializing, modifying, training and testing HMMs [12].

In the training phase, all files from the train set are used for the initialization of HMM sub-models, using HTK tool HCompV. For each word image of the training set, the corresponding sub-models are connected together to form a right-left HMM. The Expectation-Maximization (EM) algorithm is then used to iteratively refine the component weights, means and variances to monotonically increase the likelihood of the training feature vectors. In our experiments, we applied a simple binary splitting procedure to increase the number of Gaussian mixtures through the training procedure up to 512 mixtures per HMM states.

At recognition time, an ergodic HMM is composed using all sub-models. All transitions from one sub-model to the other are permitted. This approach allows recognizing potentially any word in an open vocabulary fashion. Another advantage of the ergodic topology is in its relatively lightweight memory and cpu footprint, when compared to more heavyweight approaches based on finite-state or stochastic grammars. The disadvantage of this procedure is that the system can propose invalid words as recognition results. However, simple table-lookup based post-processing can remove such invalid words. The recognition is done by computing the best state sequence in the HMM using a Viterbi procedure implemented with the HTK tool HVite. Performances are evaluated with the HTK HResult tool that output word and character recognition rates using an unseen set of word images.

10th ACM Symposium on Document Engineering (DocEng2010). 2010. pp. 161-164

2.3 Cascading Recognition System

The *cascading* system has two steps: (1) font identification using a GMM-based system and (2) word recognition using mono-font HMM-based system (see Figure 1).

2.3.1 GMM Based Font Recognition System

The proposed font recognition system uses Gaussian Mixture Models (GMMs) to estimate the likelihoods of the different fonts. Ten of the mostly used Arabic fonts present in the APTI benchmarking database are used in our settings (see Section 3.1). In the GMM approach, each font is represented by a single mixture of Gaussians. A GMM, which is actually a single state HMM, allows estimating the likelihood of a sequence of feature vectors assuming their independence.

All the training data available for a given font are pooled and used to estimate the emission probability functions of the font model. An expectation maximisation procedure coupled with a binary splitting procedure is also used to iteratively train the models and increase the number of Gaussians. At recognition time, the GMMs are fed in parallel by the features extracted from the image. The GMM issuing the highest likelihood score is elected and determines the font hypothesis. The parameters of the font recognition system have been tuned and benchmarked as explained in more details in [16].

2.3.2 Mono-font Text Recognition System

The mono-font recognition systems have the same configuration as the global HMM system described in Section 2.2. The only difference is in the training data that are pooled considering individual fonts. Ten font dependent systems are obtained through the training procedure. At testing time, the system used for recognition is selected according to the recognized font hypothesis.

3. EVALUATION

To evaluate our system, we used some parts of the large APTI database [14].

3.1 APTI Database

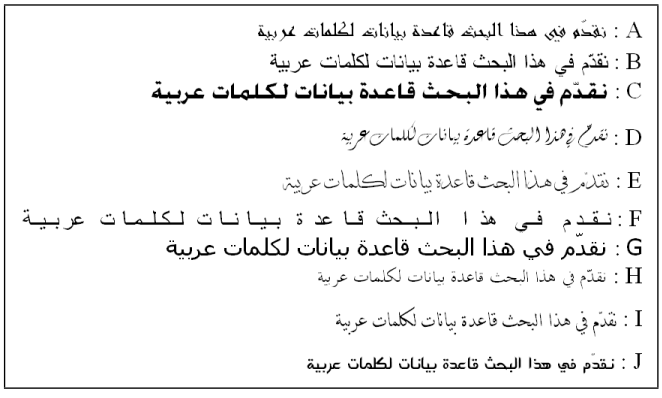

Figure 2: Fonts available in the APTI database.

Available from July 2009, APTI is freely distributed to the scientific community for benchmarking purposes[1]. At the time of writing this paper, eight research groups have started using the APTI database. The APTI database was created in low-resolution "72 dot/inch" with a lexicon of 113,284 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 the APTI database is fully described using an XML file containing ground truth information about the sequence of characters as well as information about its generation. All Arabic letters have been adequately represented in the database. 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. The sets have been designed so that the number of words and representations of letters are very close from set to set (for more details about data dispersion we refer to [13]).

As illustrated in Figure 2, ten fonts are available in APTI: (A) Andalus, (B) Arabic Transparent, (C) AdvertisingBold, (D) Diwani Letter, (E) DecoType Thuluth, (F) Simplified Arabic, (G) Tahoma, (H) Traditional Arabic, (I) DecoType Naskh and (J) M Unicode Sara.

Font recognition system is trained with 1000 word images for each font. So with 10 fonts and 1 font sizes(24), 10,000 word images were used in the training phase. In our tests for the text recognition system, we used 18'897 (set1) word images for each font and size. With 10 fonts and 1 font size, 1'889'700 word images were used in the training phase of global and font dependent systems and an additional 1'886'800 (set5) different word images were used for the test phase.

3.2 Results

All results are obtained with the font size 24 and the ten fonts available in APTI. All recognition rates are calculated using character labels, without taking into account the positioning information. So, if the system recognizes Alif_I (I: Isolate position) or Alif_E (E: End position), it is automatically transformed in the label Alif to calculate the recognition rate.

Table 1: Font recognition results

Font	*RR*	*Font*	*RR*
Advertising Bold	**98.3**	*Andalus*	**99.4**
Arabic Transparent	**97.7**	*M Unicode Sara*	**99.0**
Tahoma	**98.7**	*Simplified Arabic*	**96.6**
Traditional Arabic	**96.1**	*DecoType Naskh*	**92.6**
DecoType Thuluth	**95.0**	*Diwani Letter*	**94.0**
Mean RR		**96.7**	

We first report in Table 1 the font recognition results. Detailed results are provided for all fonts, showing good performance for most fonts and an average recognition rate (RR) of 96.7%. As *Arabic Transparent* and *Simplified Arabic* are very similar fonts, with a single difference in the inter-character horizontal elongation, we opted in our experiments to merge the fonts in a unique model. Most of the errors of the font recognition are observed on short words (typically less than 4 characters), which is rather normal. Better results could potentially be obtained in the case of larger length inputs such as lines or block of texts.

We report in Table 2 the word recognition results for the mono-font system and in Table 3 for the global system.

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

10th ACM Symposium on Document Engineering (DocEng2010). 2010. pp. 161-164