An OCR System for Arabic Calligraphy Documents

Hassanin Al-Barhamtoshy*1, Kamal Jambi1, Hany Ahmed1, Shaimaa Mohamed1, Mohsen Rashwan2, Sherif Abdou3

1Computing and Information Technology, King Abdulaziz University, KAU Jeddah, Saudi Arabia
2Faculty of Engineering, Cairo University, Egypt
3Computing and Information Science, Cairo University, Egypt
*Corresponding author E-mail: hassanin, kjambi}@kau.edu.sa

Abstract

This paper introduces to get good accuracy for Arabic OCR results for old documents and calligraphy documents. While our developed system has provided accurate results for modern Arabic documents, when we used that system for old Arabic documents, we got a steep degradation in performance, (around 25% accuracy compared with 85% for modern Arabic documents). Market of Arabic OCR for old documents is large and it deserves higher attention even more than the modern documents, which in many cases are already distributed in digitized format. Therefore, in this paper, we address the challenges of Arabic OCR for old documents. Firstly, we eliminated the word segmentation step and run the OCR process on the complete line. With this modification, we managed to avoid large number of segmentation errors on the word level but had to change our recognition approach to be a dictionary based. The second modification, we changed the used features to histogram gravity based one. This type of feature provided much better performance especially in the challenging cases of old documents such as low quality printing effects, wavy baselines and heavy noisy documents. Third, we used a hybrid model that integrate Neural Networks with HMM to provide better discrimination between the shapes of Arabic ligatures. The paper starts with fast review for the baseline system and the added enhancements of such OCR. Then introduce to the new developed OCR efforts for Arabic old documents and calligraphy documents.

Keywords: Use about five key words or phrases in alphabetical order, Separated by Semicolon.

1. Introduction

Arabic OCR is a very challenging application, due to writing nature: letter connectivity, cursive, context shape, ligatures, diacritics, dots, and it has many varieties in font names and styles. Moreover, the Arabic alphabet contains 28 letters. Each letter can be written between two and four shapes [1], depending on the position of that letter in the word. Most of Arabic fonts include complex ligatures shapes that consist of “consecutive letters of variable lengths as well as inter- and intra-word spaces”. The “problem of diacritical points can change the meaning of the word or sub-word” [1]. Fig. 1 shows sample of the challenges in Arabic scripts. Due to these reasons the current performance of state of art Arabic OCR systems is much lagging compared with the performance of other Latin based OCR systems. In this project, we targeted the advancing of Arabic OCR systems to achieve a practical level of accuracy (in the range 90-95%) with Omni-font capabilities to process unseen fonts.

In OCR system, the page is segmented into lines; each line is segmented to words. Each word is modeled as a sequence of ligature HMM models. The core system is built around the idea of the apparent analogy between the Arabic font/type-written OCR problem and the Automatic Speech Recognition (ASR) problem. Just as ASR systems use HMM to simultaneously segment/recognize the connected phonemes in continuous speech [2, 3], our OCR is using HMM to simultaneously segment/recognize the connected graphemes in connected Arabic script. Figure 2 illustrates the idea.

![Fig. 1: Arabic Script OCR Challenges](image-url)
As many of the used fonts for cursive Arabic scripts, extensively use ligatures where two or more letters are joined as a single glyph, which complicates the character level segmentation. Fig. 3 shows some samples of Arabic compound ligatures.

The holistic recognition dealing with the whole word as a particular inseparable component. An overall feature vector is extracted from the input word, which is then used to classify it against a stored dictionary of words [4]. The main challenge was developing a holistic Arabic OCR system that is computed to be efficient for a large vocabulary size. To achieve this objective, we developed a lexicon reduction approach that cluster the similar shape words to reduce the word recognition time to make that approach a practical one. That system used a hybrid of several holistic features that combine global word level Discrete Cosine Transform (DCT) based features and locale block based 4B-DCT features. Using this type of features, the system managed to achieve Omni-font performance with font and size independence. Fig. 4 shows the architecture of the developed Holistic Arabic OCR system.

The developed holistic OCR system consists of two modules. The first one is the training module where the holistic features are extracted from the training set of the word images. The extracted features are used to build set of clusters of the similar word shapes. The generated words’ clusters and their extracted features represent the knowledge base that is used in the recognition phase. The second module is the recognition module. In that module after applying the preprocessing operations on the input image, the detected text blocks are segmented to lines and words. The features are extracted for each word image. Then the word cluster, or best-n clusters that have the minimum Euclidean distance with the test image vector [5], is assigned.

1.1. Complete line recognition system

While our first approach, we opted to segment the line images to words then use word level recognition system. The main advantage of this approach is its efficiency since the recognition process is localized on word level also it provides an open vocabulary system since the system can compose any word using its ligature sequence [7]. The disadvantage of such approach is the word segmentation errors that can happen due to the overlapping that occurs in many Arabic fonts, which is the hardest challenge in cursive languages [8, 9]. Fig. 5 shows samples of segmentation errors from the collected data sets.

To avoid such type of errors we complete line recognition approach. This change lead to some major modifications in our decoding techniques.
2. New proposed model for Arabic OCR

When we evaluated the performance of our developed Arabic OCR features, on old Arabic documents, calligraphy documents, early printed documents or modern Arabic documents with high level of noise we got large degradation in the system performance. The error analysis for the OCR results of these documents revealed on several reasons for this steep degradation in performance, which is:

- The regularity in the shape evolution among the segments that represent the different states of an Arabic ligature is not always sustained in old documents.
- The wavy like baselines, which is common in the old Arabic documents would result in the deviation of the features values from there expected ones even after applying some remediation using the de-skewing techniques.
- The noisy images would result in complete damage in the features values.

Accordingly, for these reasons, we started to investigate the usage of histogram-based features as shown in Fig. 6.

![Histogram based Features with polar segments](image)

Fig. 6: Histogram based Features with polar segments

To estimate this features vector we use the following algorithm:

1. Apply a sliding window of 11 pixels in width along the writing direction.
2. Calculate the Center of Gravity “CoG”; Fig. 6 (a).
3. Divide the frame in 8 equally segments from CoG.
4. Count the number of foreground pixels in each segment (0-45, 45-90, 90-135, 135-180, 180-235, 235-270, 270-325, 325-360) as shown in Fig. 6(b).
5. Shifting 1-pixel along the writing direction.
6. Normalize the count by the total number of pixels in the whole frame which provides continues features in range between 0 and 1.

2.1. Hybrid HMM-DNN model

The characteristic of Arabic old documents presents a higher requirement in the used modeling. Models with high capacity are needed to model the diversity in the types of documents. Recently Deep Belief Networks (DBNs) with several hidden layers were proposed for acoustic modeling in speech recognition because they have a higher modeling capacity per parameter than GMMs [10]. The recent surge of interest with these effective models have resulted from the introduction of a pre-training stage, which is an unsupervised technique that can lead for an effective initialization of the DNN network parameters. Then a following step of supervised learning tunes the network parameters to optimize its discrimination capability. One of the effective approaches is the combination of the discrimination power of DNN and the dynamic timing modeling power of HMM in a hybrid DNN-HMM model. The context-dependent (CD)-DNN-HMM model that has been proposed for Large Vocabulary Automatic Speech Recognition (LVASR) has resulted in cutting word error rates by up to one third on the interesting conversational speech transcript jobs when compared to the state of art discriminatively trained conventional CD-GMM-HMM systems [11].

A DNN is simply a multi-layer perceptron with large number of hidden layers, that how it got the name of deep networks. The main challenge in training this type of networks is finding efficient training strategies that can avoid falling in poor local optimum that usually results with the complicated nonlinear error surface due to the increased number of hidden layers. A common training to deal with this local optimal challenge is to initialize the DNN parameters greedily and generatively. This can be achieved by treating each pair of layers in the network as a restricted Boltzmann machine (RBM) as a pre-training step before doing the whole network training of all the layers. This learning strategy enables the DNN training to start with well initialized weights and makes the global network training a feasible process.

An RBM is bipartite graph with two layers, the first one is considered the visible layer and second one is the hidden layer. The elements in the visible layer are only connected to the elements in the hidden layer. The visible layer elements are typically represented using Bernoulli or using Gaussian distributions while the hidden layer elements are commonly represented using Bernoulli distributions. This type of Gaussian–Bernoulli RBMs converts the real-valued stochastic variables (such as the OCR histogram features) to binary stochastic variables that can be processed further using the Bernoulli–Bernoulli RBMs [12].

Assumed the typical parameters y, the combined distribution p(v, h) over the visible elements v and hidden units h in the RBMs can be defined as:

\[
\exp(- \frac{(v, h, \lambda)}{Z(\lambda)})
\]

(1)

where \(E(v, h; \lambda)\) is an energy function and \(Z(\lambda)\) is the normalizing term. For Bernoulli RBMs, we have:
\( h(\cdot, \cdot) = -\sum_{i=1}^{h} \sum_{j=1}^{\cdot} h_{i,j} - \sum_{i=1}^{h} \sum_{j=1}^{\cdot} h_{j,i} \)  \( (2) \)
\( (\cdot, \cdot) = \sum \sum h \exp(- (\cdot, \cdot)) \)  \( (3) \)

The parameters \( \{W; a; b\} \) include the symmetric interaction between the units \((W_{ij})\) and the bias terms \((a_{i}, b_{j})\). On the other hand, for Gaussian-Bernoulli RBMs, we have:
\( (\cdot, \cdot) = \sum \sum h \exp(- (\cdot, \cdot)) \)
\( (\cdot) = \int \sum h \exp(- (\cdot, \cdot)) \)
\( (4) \) \( (5) \)

The pre-training step targets the maximization of the log-likelihood of the data, \( \log P(v; \cdot) \). Differentiating this log-likelihood with respect to the parameters results in a form that consists of the expectation over the data distribution minus the expectation over the model distribution. The exact computation of the first part, which is the expectation over the model, is intractable. A contrastive divergence based approach such as a one-step Gibbs sampling can be used to approximate the computation of the gradient of the log-likelihood probability. To build a DBN we stack several Bernoulli type RBMs on top of a one layer of type Gaussian-Bernoulli RBM. To learn the whole structure in a layer-by-layer manner the hidden activities of one RBM can be considered as the input data to a higher level RBM.

After the pre-training step of the DBN, a softmax layer is added on top of it and is trained using the classical back propagation approach. Given the model parameters fine-tuned over a pre-defined label set \( V = \{l_{1}; l_{2}; \ldots; l_{V}\} \), the DBN posterior gram for a feature vector frame \( x_{i} \) can be computed as
\( = \{ (\cdot; l_{1}); (\cdot; l_{2}); \ldots; (\cdot; l_{V}) \} \)
\( (\cdot; l_{i}) = \sum \sum h \exp(- (\cdot, \cdot)) \)
\( (6) \)

Fig. 7 illustrates the architecture of the used DNN-HMMs model for our Arabic OCR system.

3. The proposed integrated model

The model use the HMM structure to model the state transitions and DNN to model the state emissions. Two types of models where investigated in our system, the context independent ligatures based DNN-HMM and a context-dependent tri-ligature DNN-HMM model.

For context, independent based DNN-HMM we used DNN with output layer consisting from 963 nodes, that represents the 3 HMM states for each one of the used 321 ligatures. For the context, dependent based DNN-HMM we used DNN with output layer consisting from 3000 nodes that represents the clusters of the tied tri-ligature states. We used a clustering decision tree based on linguistic questions to cluster the Tri-ligature HMM model states. For the hidden layers, we experimented several DNN structures that ranged from 3 to 5 layers with each layer consisting form large number of nodes in the range 2000 to 4000 nodes.

The impeded Viterbi algorithm is used to train the DNN-HMM model. A ligature level decoder was integrated in the training process. The objective function is to find the determinant of the ligature sequence \( Ph \) that maximize:
\( h = \arg \max_{h} (h) = \arg \max_{h} (\cdot | h) (h | \cdot) \)
\( (7) \)

Where \( p(Ph) \) is the ligature level Language Model (LM) and \( (\cdot | h) \) is the OCR model probability estimated by:
\( (\cdot | h) = \sum (\cdot | h) (\cdot | h) \)
\( \equiv \max (0) \prod_{\cdot | h} = 1 \prod_{\cdot | h} = 0 (\cdot | h) \)
\( (8) \)

Where \(-1\) is the HMM sate transition probability and \((\cdot | h)\) is the observation probability that can estimated by:
\[
( | ) = ( | ) \cdot ( ) / ( ) / ( )
\]

Where \(( | )\) is the state posterior probability that is estimated from the DNN, \(( )\) is the prior probability of each state estimated from the training data, and \(( )\) is independent from the ligature sequence and can be ignored?

The main steps of the training procedure for the DNN-HMM model are summarized as following:

1- Train the best HMM model (either CI or CD) using the full training dataset.
2- Convert that HMM model to DNN-HMM model by borrowing the state transitions information (and state tying for the CD model).
3- Pre-train each layer in the DNN bottom-up layer by layer using the unlabelled training data.
4- Use the trained HMM to generate required labels for the training data frames up to the state level.
5- Use these acquired labels to train the last layer of the DNN using the back-propagation approach and the impeded Viterbi approach.
6- Use the trained DNN-HMM to generate new labels for the training data.
7- Repeat steps 5-6 for several iterations till the model converge.

3.1. Experimental Testing and Results

We compared the performance of Year-1 system, the 2-D HMM based system as reported in, and year-2 system which is hybrid HMM-DNN with histogram features and use complete line recognition with 3-gram language model as described in this report.

We used two test sets, the first one is 50 pages selected randomly from modern Arabic documents. The second test set was 50 pages selected from old and historical Arabic documents.

3.2. Modern Documents Results

The results in table (1) show that the new version of the system provides absolute 4% gain over the previous version. The gain can be contributed to the new histogram features, the enhanced HMM-DNN modeling and the complete line recognition to avoid word level segmentation errors. This accuracy improvement was on the price of the system speed. The complete line recognition has increased the search space greatly compared to the single word recognition in the old version. This change in recognition approach has slowed the recognition rate form 1 second / page in the old version to 20 seconds / page in the new version. With recent availability of large cloud computing facilities, the processing time for OCR systems can be considered a secondary issue, as you always can add hardware to compensate for the page recognition time delay, in comparison to the system accuracy, which is the more critical issue. Especially for Arabic OCR systems that have been lagging for a long time the performance of equivalent systems of Latin based languages. Therefore, any gain in the accuracy for an Arabic OCR system is for sure the most critical issue.

<table>
<thead>
<tr>
<th>Arabic OCR V.1</th>
<th>Arabic OCR V.2</th>
<th>Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>85.94%</td>
<td>89.99%</td>
<td>4.05%</td>
</tr>
</tbody>
</table>

Table 2 displays sample of recognition results for modern Arabic documents using the old and new Arabic OCR systems. The reference text is displayed to the right and the recognition results are displayed to the left. The highlighted lines in yellow color are the ones that include errors. The error words are highlighted in grey. The white lines are the ones that are recognized totally correct.

3.3. Calligraphy/ Historical Documents Results

The results in table 3 shows that the new version of the system provides absolute 62% absolute gain over the previous version. This was a huge gain, and it provides a practical level of performance. As seen in these results, the previous version of the system failed to achieve any practical performance for the old documents. The same degraded performance was achieved also for the most common commercial Arabic OCR systems such as Sakhr and Novo-Dynamics.
The large improvement of the new system can be contributed to 3 main reasons:

1- Using complete line segmentation, eliminated most of segmentation errors, especially with the words overlap which is very frequent in old Arabic documents.

2- The used histogram based features which proved to be robust against the low printing quality, the dominant noise effect and the wide variance in ligature shapes that represent the main features of old Arabic documents.

3- The discrimination power of the HMM-DNN models, which provided better modeling for the large number of Arabic ligatures that are common phenomena in Arabic old documents.

Table 4 displays the recognition results for the old and new versions of the system for some old Arabic documents.

4. Conclusions

In spite of the many efforts that have been devoted in the last two decades towards achieving Arabic OCR technologies with practical level of performance, the current state of art of Arabic OCR system is far lagging the equivalent technologies for Latin languages. In this paper we targeted the research and development of Arabic OCR that can provide practical performance for digitizing Arabic documents. Therefore, we worked on enhancing the performance of all the components of Arabic OCR system starting with document preprocessing and denoising, text detection, line segmentation, features extraction and the recognition engine. We introduced some new techniques and enhanced others. Also we addressed the challenges of old Arabic documents and managed to improve the OCR accuracy of such type of documents above the level to make them searchable documents. We evaluated the performance of the developed system against the best performing Arabic OCR systems that are commercially available and our system showed significant improve in OCR accuracy, around 10% absolute raise in the recognition accuracy.

Acknowledgment

This project was funded by the National Plan for Science, Technology and Innovation (MAARIFAH) – King Abdulaziz City for Science and Technology – the Kingdom of Saudi Arabia – award number (11-INF-1997-03). The authors also, thanks Science and Technology Unit, King Abdulaziz University for technical support.

References


