Two-Tier Approach for Arabic Offline Handwriting Recognition

Ahmad Abdulkader

Two-Tier Approach for Arabic Offline Handwriting Recognition

Ahmad AbdulKader
Microsoft Research
ahmadab@microsoft.com

Abstract

In this paper we present a novel approach for the recognition of offline Arabic handwritten text that is motivated by the Arabic letters' conditional joining rules. A lexicon of Arabic words can be expressed in terms of a new alphabet of PAWs (Part of Arabic Word). PAWs can be expressed in terms of letters. The recognition problem is decomposed into two problems that are solved simultaneously. To find the best matching word for an input image, a Two-Tier Beam search is performed. In Tier one the search is constrained by a letter to PAW lexicon. In Tier two, the search is constrained by a PAW to word lexicon. Directing the searches is a Neural Net based PAW recognizer.

Experiments conducted on the standard IFN/ENIT database [7] of handwritten Tunisian town names show word error rates of about 11%. This result is comparable to the results of the commonly used HMM based approaches.

Keywords: Offline Arabic handwriting recognition, Neural Networks, IFN/ENIT, Beam Search.

1. Introduction

The recognition of handwritten text in images, commonly known as offline handwriting recognition, is still a challenging task. Significant work still remains to be done before large scale commercially viable systems can be built. This is more so for Arabic (and other non-Latin scripts in general) than Latin scripts where less research effort has been put into solving the problem.

Most research in Arabic offline recognition has been directed to numeral and single character recognition [2]. Few examples exist where the offline recognition of Arabic words problem is addressed [6]. The availability of standard publicly available databases of handwritten Arabic text images like IFN/INIT database has encouraged more research in this area [6] [10].

For Latin scripts, HMM (Hidden Markov Model) based approaches have dominated the space of offline cursive word recognition [11] [1]. In a typical setup, a lexicon is provided to constrain the output of the recognizer. An HMM is then built for every word in the lexicon and the corresponding likelihood (probability of data being generated by the model) is computed. The most likely interpretation is then postulated to be the correct one.

In the few reported approaches to Arabic recognition, approaches very similar to the ones used in Latin were used [6]. Some attempts were made to modify the preprocessing and feature extraction phases to accommodate the different nature of the Arabic writing script. However, the author is not aware of any attempts to this date to exploit the unique properties of Arabic script for recognition purposes.

In this work, we will present an approach that exploits a key (yet often ignored) property of the Arabic writing script in building a recognition system. This property is basically the set of conditional joining rules that govern how Arabic letters are connected in cursive writing. In Section 2, we show how this property leads to the emergence of PAWs and how our approach exploits these to build a two-tier recognition system. In Section 3, we describe our recognition system in details. Section 4 reports the experimental results conducted on the publicly available IFN/ENIT database of handwritten Tunisian town names and how these compare to the results reported using alternative approaches.

A system built based on the approach described in this paper was submitted as an entry to the ICDAR05 Arabic word recognition competition [8]. The system was evaluated as the second best system on a blind test set and the best system on the non-blind test set. The author's remarks on the competition and the effect of the inconsistency between the training and test set distribution are provided at the end of the paper.

2. Exploiting the Arabic Writing System

Arabic (*arabī*) is the 5th most widely spoken language in the world. It is spoken by close to 300 million speakers mostly living in North Africa and South West Asia. It is the largest member of the Semitic branch of the Afro-Asiatic language family.

Arabic script has a distinct writing system that is significantly different from the commonly known Latin or Han based writing systems. Below is a brief

description of the writing system and how one of its unique properties has been exploited to build an offline word recognition system.

2.1. The Arabic writing system

The Arabic alphabet is written from right to left and is composed of 28 basic letters. Adaptations of the script for other languages such as Persian and Urdu have additional letters. There is no difference between written and printed letters; the writing is UNICASE (i.e. the concept of upper and lower case letters does not exist).

The Arabic script is cursive, and all primary letters have conditional forms for their glyphs, depending on whether they are at the beginning, middle or end of a word. Up to four distinct forms (initial, medial, final or isolated) of a letter might be exhibited [5].

However, only six letters (د ذ ر ز و) have either an isolated or a final form and do not have initial or medial forms. If followed by another letter, these six letters do not join with it, and so the next letter can only have its initial or isolated form despite not being the initial letter of a word. This rule applies to numerals and non-Arabic letters as well. This property is often referred to as ***conditional joining***. Figure 1 shows an illustration of this property.

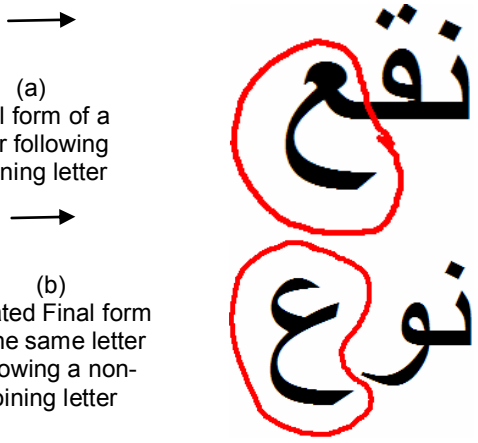

Figure 1. An illustration of the conditional joining property in Arabic script.

The conditional joining property leads to the emergence of PAWs (Part of Arabic Word). A PAW is a sequence of Arabic letters that are joined together with no exceptions. Given an Arabic word, it can be deterministically segmented into one or more PAWs.

It is worth noting that an Arabic writer must strictly abide by the conditional joining rule. Otherwise, the handwriting might be deemed unreadable. However, due to sloppiness in writing or image acquisition conditions, PAWs might end up being physically connected in an image. We empirically estimate that this happens in less than 5% of the overall PAW population. In Section 3.4, we will explain our approach for handling these cases.

2.2. A two-tier approach

Given the conditional joining property of the Arabic writing script, words can be looked at as being composed of a sequence of PAWs. In other words PAWs can be considered an alternative alphabet. The unique number of PAWs constituting a word lexicon grows sub-linearly with the number of words in the lexicon. Figure 2 shows how the number of unique PAWs grows with the size of an Arabic lexicon.

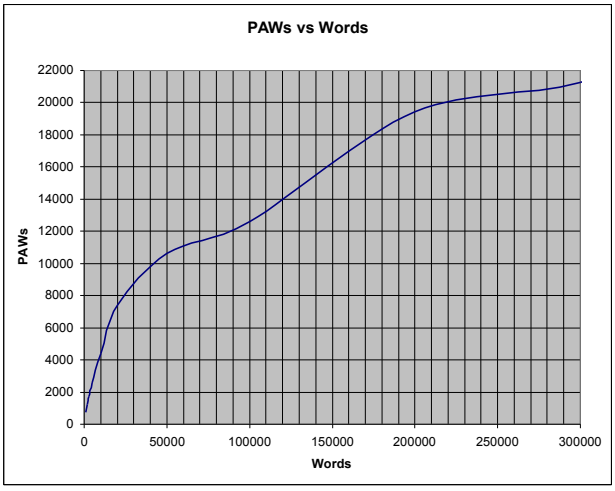

Figure 2. The number of unique PAWs in a lexicon grows sub-linearly with the number of words.

A lexicon of Arabic words can then be decomposed into two lexica. The first is a PAW to letter lexicon which lists all the unique PAWs and their spelling in terms of the letter alphabet. The second is a word to PAW lexicon that lists all the unique words and their spelling in terms of the PAW alphabet.

Consequently, the problem of finding the best matching lexicon entry to an image can be decomposed into two intertwined problems that are solved simultaneously. The first problem is finding the best possible mapping from characters to PAWs constrained by the first lexicon. The second problem is finding the best possible mapping from PAWs to words constrained by the second lexicon.

This two-tier approach has a number of useful properties. One property is that since lexicons constrain the outputs of the recognition process, a number of character recognition errors can be fixed in the PAW recognition phase. Figure 3 shows an example of this type of potential recognition errors. It is unlikely in this example that the second letter "ص" would have been proposed by a character recognizer given how poorly it is written.

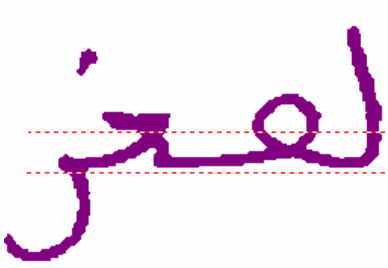

Figure 3. An example image of the لصغر PAW that is confusable with لعخر which is a valid lexicon PAW.

Another property is that PAWs end up having their own prior probabilities that can be utilized by the PAW recognizer to favor more frequently occurring PAWs. These prior probabilities can be looked at as a linguistic n-gram character model that drives the recognition process.

3. The recognition system

A block diagram of the two-tier recognition system is shown in Figure 4. In the following sections we will describe the preprocessing, normalization, segmentation, recognition and search steps in detail.

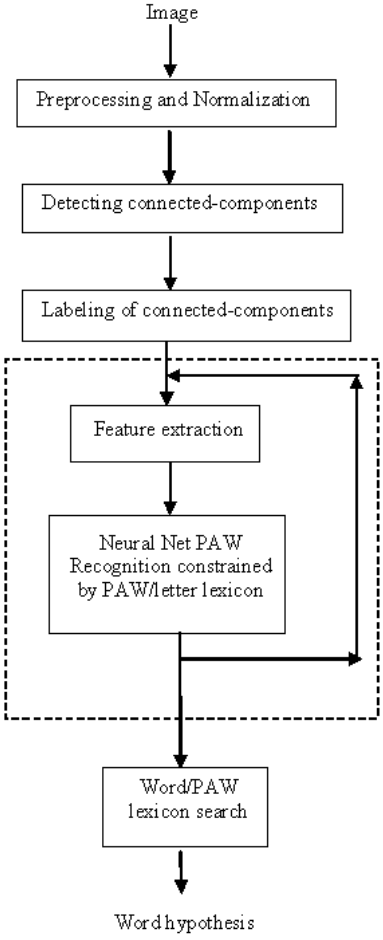

Figure 4. A block diagram of the recognition system

3.1. Preprocessing, normalization and segmentation

The images in the IFN/ENIT database had already passed through the basic processing of image binarization, cropping, word segmentation and noise reduction; we have skipped these phases in our experiments. The very first step of processing is the detection of connected-components. Connected-components whose width and height are below a certain threshold (the choice of which is not critical) are obtained. This step acts as an additional noise reduction step. Connected-components are then sorted from right to left based on their rightmost point. This allows the search algorithm to sequence through the connected-components in an order that is close to the writing order.

Connected-components are then labeled as 'primary' and 'secondary'. This labeling is performed by detecting relative horizontal overlaps between connected-components and applying some safe thresholds on connected-component sizes. Each secondary connected-component has to be associated to a primary one. No secondary component can exist alone. Figure 5 shows a color coded labeling of connected-components in an image of a word.

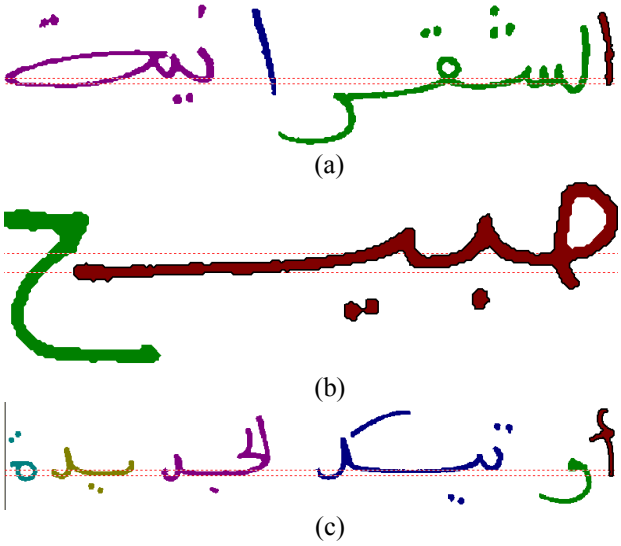

Figure 5. Three examples of color coded grouped connected-components. (a) A case where each connected-component group is an actual PAW. (b) A case where a PAW was split into two connected-component groups. (c) A case where two PAWs were joined in one connected-component group (purple color).

In 5(a) each connected-component group corresponds to exactly one PAW. We have empirically determined that this case represents around 65% of the overall population of words. Figure 5(b) shows a case where the two connected component groups correspond

to one PAW (i.e the over-segmentation case). Over-segmentation represents around 30% or the word population. Figure 5(c) shows the case where the purple connected component-group is actually two touching PAWs. This case is not inherently handled by the proposed approach. It constitutes around 5% of the cases. We will explain in section 3.4 how it was handled. As such, a fundamental assumption of the following steps of the system is that PAWs can only occur on connected-component group boundaries.

3.2. The Neural Net PAW recognizer

The IFN/ENIT database has a lexicon of 946 Tunisian town names. The number of unique PAWs in this word lexicon is 762. Although the training database might not necessarily have at least one sample of each valid word, it turns out that there is at least one sample present of every valid PAW.

Because of this, we decided to use a Neural Network based classifier to recognize PAWs. As the size of the word lexicon gets bigger and the number of valid PAWs grows, it might not be practical to directly use a Neural Network classifier for recognizing PAWs.

In our experiments we build two Neural Net PAW classifiers. The first classifier is a convolutional Neural Network. Convolutional Neural Networks [9] has been reported to have the best accuracy on offline handwritten digits. In this type of networks, the input image is scaled to fit a fixed size grid while maintaining its aspect ratio. Since the number of letters in a PAW can vary from 1 to 8, the grid aspect ratio has to be wide enough to accommodate the widest possible PAW while still maintaining its distinctness. The second classifier is based on features extracted from the directional codes of the connected-components constituting the PAW. Each of these two classifiers has 762 outputs and was trained with training sets that reflect the prior distributions of PAWs in the word lexicon.

3.3. Beam search

As mentioned above, the word lexicon can be decomposed into two lexica: A letter to PAW lexicon and a PAW to word lexicon. The letter to PAW lexicon is used to constrain the output of the PAW recognizer as mentioned above. The PAW to word recognizer is used to constrain the search for the best matching word.

Beam search an algorithm that is an extension to the best-first search. Like best-first search, it uses a heuristic function to evaluate the promise of each node it examines. Beam search, however, only unfolds the first *m* most promising nodes at each depth, where *m* is a fixed number, the "beam width". It is very commonly used in speech recognition [3].

The Beam search is used to find the best matching word to an image using the output of PAW recognizer as a search heuristic. The search sequences through the connected-components groups and considers either starting a new PAW or adding the group to the existing PAW. The list of possible PAWs together with their corresponding posterior probabilities produced by the PAW recognizer is retained. Different connected-component group to PAW mappings are kept in a lattice of possible segmentations. After sequencing through all the groups, the best possible segmentation is evaluated and chosen to be the winning hypothesis.

For practical reasons and to make sure that the segmentation possibilities in the lattice do not explode, two heuristics are used. First, the maximum number of connected-component groups per PAW is capped at 4. This number has been determined empirically based on the training data. Second, at every step in the lattice, segmentation possibilities that have a probability that is lower than the most probable segmentation by a certain threshold are pruned. This means that theoretically, the Beam search might not produce the most probable segmentation. However, this rarely happens in practice.

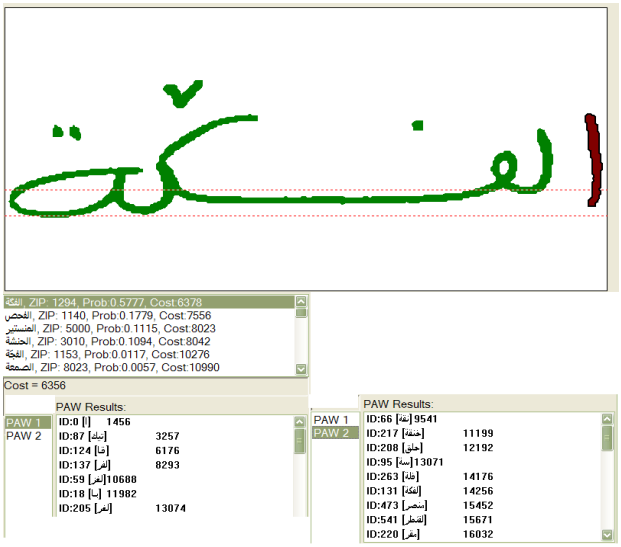

Figure 6. An recognition example showing the word recognition results in the top list and the PAW recognition results in the lower list boxes.

Figure 6 shows an example image, the final recognition results and the PAW recognition results of the two connected-component groups. Note that although the second PAW was misrecognized, the overall word was correctly recognized.

3.4. Handling exceptions

As pointed out earlier the under-segmentation case was empirically determined to constitute around 5% of the words. To handle the under-segmentation case, where more then one PAW end up being segmented as one connected-component group a final step in the process was added. The final step is triggered if the probability of the winning segmentation path in the lattice is lower than a certain threshold. This was found to be strong evidence that under-segmentation occurred. When triggered, a Viterbi search is performed on the

individual PAW recognition results of the connected-component groups. In this search the edit distance between the each of the PAW to Word lexicon and the recognition results are computed. Both PAW insertions and deletions are allowed with a penalty associated with each.

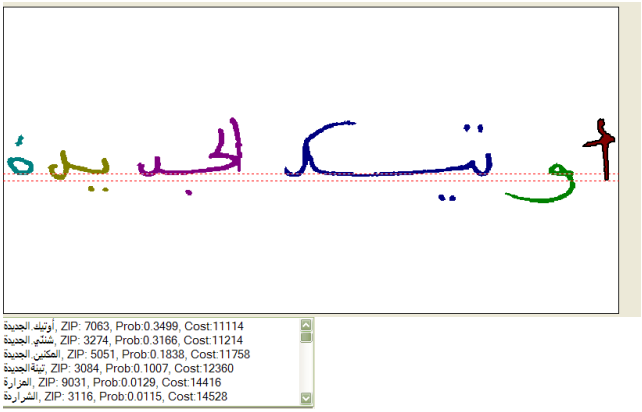

Figure 7. A recognition example of an under-segmented image. The Viterbi search that is triggered when the best Beam result is lower than a certain threshold produced the correct answer.

4. Experiments

4.1. The data set

Experiments were conducted on the publicly available IFN/ENIT database [7]. The database is split into four sets A, B, C & D. The 4 sets contain 26,459 images of segmented Tunisian town names handwritten by 411 unique writers. The total number of PAWs in the set is 115,585. For each image the ground truth information is available. The number of unique word labels is 946, and the number of unique PAW labels is 762.

Sets A, B & C were used for training and validation. Set D was used for evaluation. Set D has 6735 words handwritten by 104 unique writers none of which contributed any the words in sets A, B or C.

A widely agreed upon rule of thumb in building recognition systems is to ensure that recognizers are evaluated on a distribution similar to that of the training set. Since the 4 sets roughly have the same writer demographics, word distribution and consequently PAW distribution, this rule was upheld in our experiments.

4.2. The training process

One problem that was encountered during implementing the recognition system was getting data to train the PAW recognizer. As such, the database has word level ground truth information and does not have PAW level ground truth information. To solve this problem, we followed a bootstrapping technique similar to the bootstrapping from incomplete data in the well known Expectation-Maximization *EM* setting [4].

As mentioned in Section 3.1, our connected-component segmentation and grouping algorithm results in three different types of segmentation. For the first type, which we call *exact segmentation,* each of the resulting connected-component groups corresponds to one and exactly one PAW. Empirically, it was determined that *exact segmentation* cases constitute 65% of the total word population.

For each training sample, the number and the identity of the PAWs that constitute the sample's word label can be computed. To bootstrap the training process, a conjecture is made that for every sample in the training set where the number of connected-component groups is equal to the number of label PAWs, the identity of a specific PAW corresponds to the ground truth label of the connected-component group at the same position. This conjecture holds almost all the time. There are rare cases where PAW over-segmentation and under-segmentation occur an equal number of times in a word which results in breaking the *exact segmentation conjecture*.

As a first step, the PAW recognizers are trained on all the training samples that satisfy the *exact segmentation conjecture*, which is 65% of the training data. In subsequent steps, by using the ground truth word label and its corresponding PAWs, the PAW recognizer that was trained in the previous step is used to segment connected-component groups into PAWs. This is done by running the same exact algorithm described in Section 3 with only one entry in the word lexicon: the ground truth. This could only work for exactly segmented and over-segmented words. And so, the under-segmented words, which constitute 5% of the training set, are excluded from the training process. The training step is analogous to maximization step in EM, while the PAW re-segmentation phase is analogous to the expectation step. This sequence (training, re-segmentation) was repeated 3 times until no significant change in the accuracy of the PAW recognizer was observed.

4.3. PAW recognition results

The results of the two individual PAW recognizers and their combined results are shown in Table 1.

Table 1. The error rates of the individual PAW recognizer and the combined PAW recognizer on set D of the IFN/ENIT database.

Recognizer	*Top 1 Errors*	*Top 10 Errors*
Convolutional Net	44.86%	16.34%
Directional Codes	36.94%	13.27%
Combined Classifier	25.34%	10.09%

4.4. Word recognition results

Table 2 shows the error rates for the overall word recognizer as measured on set D of the IFN/INIT database. It also shows the results broken down by the type of segmentation encountered in the image.

Table 2. The error rates of overall word recognizer.

Data subset	*Top 1 Errors*	*Top 10 Errors*
All data	11.06%	4.99%
Exact Segmentation	7.11%	1.67%
Over-Segmented	13.33%	4.39%
Under-Segmented	36.03%	36.03%

5. Conclusion

In this paper we have presented a novel approach to the recognition of lexicon constrained Arabic handwritten words. The approach exploits the conditional joining of letters property in Arabic writing script to decompose the problem into two problems that are solved simultaneously. Using a Neural Network based PAW recognizer a two-tier Beam search is performed to find the best matching word to the input image. Word error rates of around 11% were achieved on the publicly available IFN/ENIT database. These results are comparable to the results reported on the same set using an alternative HMM based approach [6].

5.1. The ICDAR05 competition

The same results were also reported as part of the ICDAR05 Arabic handwritten word recognition competition report. A system that implements the presented approach was ranked as the second best entry on the blind-test (whose results are not reported here since the author has no access to it) and the best entry on the non-blind test set (set D).

It is worth noting that the blind set had a different distribution of words than all published sets A, B, C & D of the database. This in turn resulted in an unexpected PAW prior distribution. This might explain that the error rate reported on the blind set is significantly higher than the non-blind set. The author is of the opinion that the competing recognizers should have been evaluated on a distribution similar to that of the training set.

6. Acknowledgment

The author would like to thank the developers of the IFN/INIT database for making it possible to evaluate different Arabic handwritten word recognition systems in an objective manner and increasing interest in Arabic handwriting recognition.

7. References

[1] A. Vinciarelli, J. Luettin. "*Off-Line Cursive Script Recognition Based on Continuous Density HMM*" International Workshop on Frontiers in Handwriting Recognition, IWFHR 2000.

[2] B. Al-Badr and S. A. Mohmond. "*Survey and bibliography of Arabic optical text recognition*". Signal Processing, 41:49–77, 1995.

[3] H. Ney, D. Mergel, A. Noll, and A. Paesler. "*Data driven search organization for continuous speech recognition. IEEE Transactions on Signal Processing*", 40(2):272--281, February 1992.

[4] J.A. Bilmes, *"A gentle tutorial of the EM algorithm and its applications to parameter estimation for Gaussian mixture and hidden Markov models"*, Technical Report TR-97-021, International Computer Science Institute, Berkeley, California, 1998.

[5] K. Versteegh, "*The Arabic Language*", Edinburgh University Press, 1997.

[6] M. Pechwitz & V. Maergner, "*HMM based approach for hand- written Arabic word recognition using the IFN/ENIT database*", Proc. 7th Int. Conf. on Document Analysis and Recognition, Edinburgh, Scotland, 2003.

[7] M. Pechwitz, S. S. Maddouri, V. Maergner, N. Ellouze, and H. Amiri. "*IFN/ENIT - database of handwritten Arabic words*". In *Proc. of CIFED 2002*, pages 129–136.

[8] V. Margner, M. Pechwitz, H. E. Abed. "*ICDAR 2005 Arabic handwriting recognition competition*". Eighth International Conference on Document Analysis and Recognition, 2005. Proceedings. page(s): 70- 74 Vol. 1

[9] P. Simard, D. Steinkraus, J. C. Platt. "*Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis*". ICDAR 2003: 958-962.

[10] R.A. Haraty and H.M. El-Zabadani, "*Hawwaz: An Offline Arabic Handwriting Recognition System*", International Journal of Computers and Applications - 2005.

[11] T. Steinherz, E. Rivlin, N. Intrator, "*Off-Line Cursive Script Word Recognition A Survey*". International Journal on Document Analysis and Recognition -1999, 2:90-110.