Class-Based Contextual Modeling for Handwritten Arabic Text Recognition

Irfan Ahmad [1, 2]
[1] Information and Computer Science Department
KFUPM, Dhahran Saudi Arabia
irfanics@kfupm.edu.sa

Gernot A. Fink [2]
[2] Department of Computer Science
TU Dortmund University, Germany
gernot.fink@tu-dortmund.de

Abstract — **In this paper we will present our investigations related to contextual modeling for HMM-based handwritten Arabic text recognition. We will, first, discuss the justifications and the need for contextual modeling for handwritten Arabic text recognition. Next, we will discuss the issues related to contextual modeling for Arabic text recognition. Finally, we will present our novel class-based contextual modeling for HMM-based handwritten Arabic text recognition. Experiment results on word recognition tasks show improvements in word recognition rates when compared to using standard contextual HMMs. Moreover, the recognizers are significantly more compact as compared to the standard contextual HMM systems.**

Keywords — Contextutal modeling, Handwritten text recognition, Hidden Markov models, Arabic text recognition, Class-based contextual modeling.

I. Introduction And Related Work

Handwritten Arabic text recognition is a challenging task. Although techniques related to deep learning has gained recent attention and popularity from the researchers, hidden Markov models (HMMs) are still a popular choice due to its benefits like simple and robust training, no need for explicit segmentation of text line images, and the ease of integrating the n-grams as language models. HMMs also provide mechanisms to model and cope with contextual variability affecting a character due to its neighboring characters. Contextual HMMs have been used successfully in speech recognition and significant improvements in recognition results have been reported (e.g. [1], [2]). In fact, its use in speech recognition is a standard procedure. However, the justifications of using contextual HMMs in text recognition and the benefits of using them over the non-contextual HMMs have not been clearly established (cf. e.g. [3], [4]).

To setup contextual HMMs, first the non-contextual models (commonly termed as monophones in speech recognition) are initialized and trained and all the different contextual forms (generally the left and the right contexts are considered leading to what is, commonly, termed as triphones in speech recognition) are generated using the training transcriptions and initialized from the corresponding non-contextual forms of the models. Using contextual models greatly increases the number of HMMs in the recognition system and this can lead to inadequate training for each of the contextual form. This concern is addressed by performing some form of parameter sharing between the contextual models. The most common approach is to perform state tying of the different contextual forms of the corresponding non-contextual HMM.

There are two main approaches for state tying, i.e., the bottom-up data driven approach and the top-down decision tree based approach. In the data driven approach, the corresponding states of the contextual forms are tied if the inter-state distance is within a threshold. Appropriate distance measure is selected and the threshold value for state clustering is normally set empirically. For the decision tree based clustering approach, the corresponding states of all the contextual forms are initially pooled together and are then successively split based on questions, each splitting the group into two next level nodes until all the questions have been used. These questions are set by the experts having the domain knowledge of the script/language.

Contextual HMMs for handwritten Arabic text recognition with decision tree clustering were presented in [5]–[7]. The use of contextual HMMs with state clustering was reported in [8]. In [9], the authors presented decision tree based clustering for contextual HMMs where the clustering questions were based only on a character's core shape. In [10], the authors presented contextual modeling using HMMs for Arabic text recognition where a few contextual forms were manually selected to be modeled. These were mainly characters with descenders which potentially lead to overlaps with the neighboring characters. A total of only 44 contextual forms were added to the original model set. It seems that state clustering was not performed which is understandable given that only a few contextual models were added. In [11], the authors used contextual models for Arabic text recognition. A slight improvement was reported over the use of context independent modeling. One possible reason for not so large improvement in recognition performance, as stated in the paper, was that the use of Arabic character shapes as HMMs already captures most of the context and hence additional contextual modeling, with the implication of addition of many more models, may not be very helpful. The authors of the present paper presented contextual HMMs in conjunction with sub-character HMMs [12]. The use of contextual sub-characters including the connector models showed significant improvements in recognition rates. Data driven clustering was used to tie similar states between contextual forms of a sub-character while preserving the state sequences.

The rest of the paper is organized as follows: In Section II, we will discuss the justification and the need for contextual modeling for handwritten Arabic text recognition. Next, we will discuss the issues related to contextual modeling for Arabic text recognition in Section III. In Section IV, we will discuss class-based contextual modeling for Arabic text recognition. In Section V, we will present the experimental results and the discussions. Finally, conclusions will be presented in Section VI.

II. The need of contextual modeling for handwritten Arabic text recognition

Contextual variations in Arabic text can be visualized and understood at multiple levels. As each character in Arabic can take different shapes based on its position, this is the first contextual level that need to be modeled. Fig. 1 shows different Arabic words where the encircled glyphs in every row represent the same character. It can be clearly seen from the figure that the characters have significant variations due to their position in a word. The most common way to accommodate these contextual variations between different character shapes is by treating each character shape as a separate model. An alternative approach is to model a character, instead of character shape, as an HMM and use contextual HMM modeling to capture the shape-based variations as was presented by Prasad et al. for printed Arabic text recognition [4].

The second level of contextual variations is at the character-shape level. Even the character shapes show visual variations due to a number of reasons. As characters in Arabic script are connected to their neighboring characters in a word (a more correct term will be Parts of Arabic Word (PAW) instead of word as some characters do not connect to other characters in front of them), some stroke variations do occur when connecting a character shape to the next character shape in a word. Some variations are simply due to the different handwriting styles, but some variations seem to be due to a character’s neighboring characters. A prominent example of this phenomenon is the occurrence of character pairs that are treated as special ligatures like *lām-alif* (لا). A solution to address this is by modeling these ligatures as separate models (cf. [13]). But, the problem is that, some of these character pairs do not always appear in ligature forms and, thus, it is not always possible to model these character pairs as special models. Fig. 2 illustrates example character sequences written as special ligatures as well as in normal forms.

Another reason for variations at the character-shape level is a result of using the sliding window technique for feature extraction. Some characters partially overlap with other characters even though they might not, necessarily, be connected. These overlap get captured within the sliding window passing over a character and, as a result, effects the features computed for the character. Fig. 3 illustrates contextual variation at character-shape level due to the neighboring characters. Each row marks a character shape in a specific color to illustrate the variations in visual appearances due to its neighboring characters. In order to account for these variations, the most common approach is to model each character shape as a separate model and do contextual HMM modeling at the character-shape level, i.e., a contextual HMM represents a character shape in the context of its neighboring character shapes.

III. The issues associated with contextual modeling for Arabic text recognition

Because of the fact that variations at character-shape level, due to the neighboring character shapes, exists in handwritten Arabic texts, it becomes important to model them for better recognition performances. Contextual HMMs are, thus, a natural choice for Arabic text recognition. However, using tri-character (tri-character-shape in this case) HMMs for contextual modeling comes with their own issues, especially for Arabic script. The concern is related to the high number of contextual models that results from converting the mono-character-shape HMMs to tri-character-shape HMMs. Using character shapes as HMMs instead of character HMMs already led to a four-fold increase in the number of HMMs. Now, converting these character-shape HMMs into the contextual forms leads to a further increase in the number of HMMs. One can easily end up having thousands of HMMs. Having a huge number of HMMs leads to the problem of insufficient training data. Moreover, some low occurring character-shapes will have even lower number of its different contextual forms in the training data. This leads to inadequate model training. To alleviate this problem, some form of clustering is performed. The two most common approaches are the data-driven clustering and the decision-tree clustering. Both these techniques have been used for Arabic text recognition, as was presented in Section I. However, model clustering is applied after training the contextual forms and, thus, if the training was not adequate, the clustering will not be optimal. Thus, although there are strong justifications for using contextual HMMs for Arabic text recognition, its potential has not been greatly achieved.

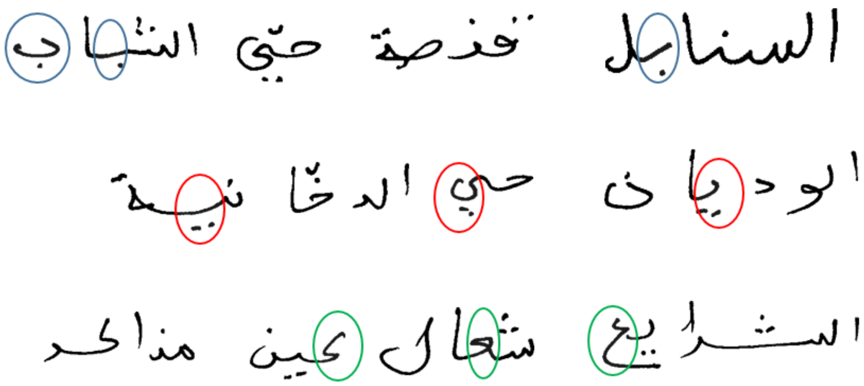
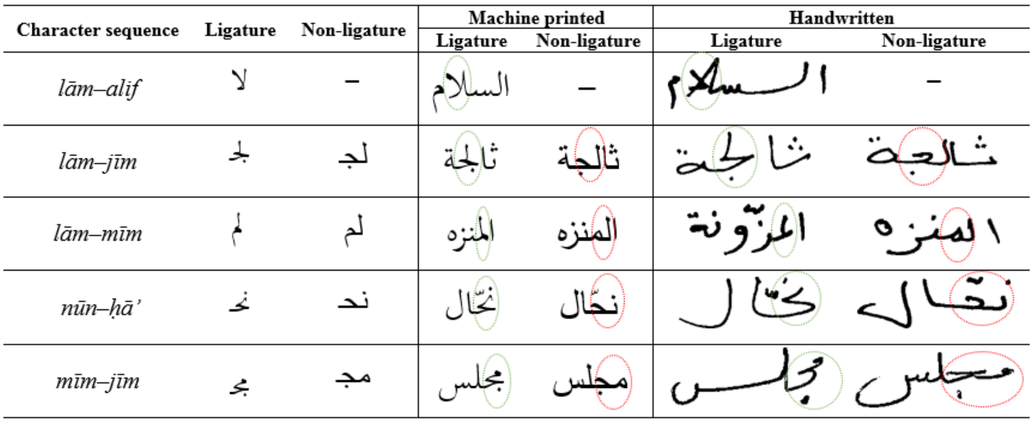


Fig. 1. Handwritten Arabic texts illustrating the variation in characters’ appearance. Glyphs encircled by a color in each row represent the same character.



Character sequence	Ligature	Non-ligature	Machine printed		Handwritten	
			Ligature	Non-ligature	Ligature	Non-ligature
lām–alif	لا	–	السلام	–	السلام	–
lām–jīm	لج	لجـ	ثالجة	ثالجة	ثالجة	ثالجة
lām–mīm	لم	لم	المنزه	المنزه	المزونة	المنزه
nūn–ḥā'	نخ	نخـ	نخال	نخال	نخال	نخال
mīm–jīm	مج	مجـ	مجلس	مجلس	مجلس	مجلس

Fig. 2. Example character sequences, their ligature and non-ligature forms with examples from machine printed and handwritten texts.

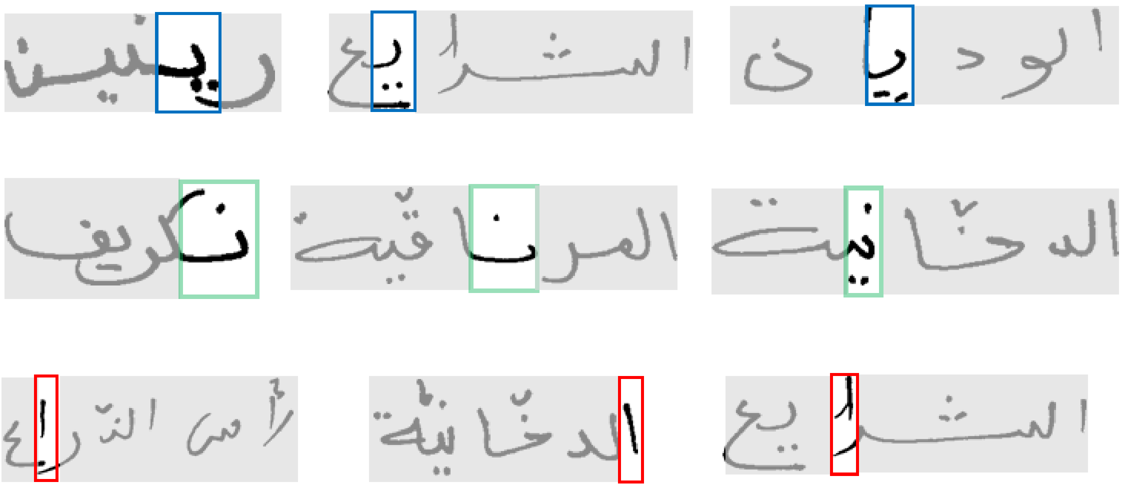


Fig. 3. Figure illustrating the effect of neighboring characters on character shapes. Each row shows instances of a specific character shape (enclosed within a colored-edge rectangle) and the variations in the visual appearances due to the neighboring characters.

IV. Class-Based Contextual Modeling for Arabic Text Recognition

As we have presented above, one of the main problems when using contextual HMMs for Arabic text recognition is the inadequate training of the resulting high number of tri-character models. We also observed in [12] that, by using sub-character modeling approach and particularly the connector models, we can reduce the number of unique tri-characters. These observations led us to investigate class-based contextual modeling for Arabic text recognition. The core idea was to limit the number of unique tri-characters by not modeling every tri-character as a separate model, but, instead, by grouping the characters in the left and the right context into classes where characters in each class have similar effects on its neighboring characters.

The idea of class-based contextual modeling for handwritten text recognition is not new. A similar idea was presented by Fink and Plötz [3] for offline recognition of handwritten text in Roman script. The authors grouped the characters appearing in the left and the right context into six different categories: characters occupying core area, characters with ascenders, characters with descenders, characters with ascenders and descenders, numerals, and the upper case characters. The groups were identical for both the left and the right context. Authors reported improvement in recognition results, as compared to the baseline context-independent system, when using this approach. At the same time, the standard approach to contextual HMMs resulted in poorer results when compared to the baseline system.

We grouped the Arabic characters into seven different classes for the left context, i.e., for character appearing after a given character, and four different classes for the right context, i.e., for the characters appearing before a given character. This grouping is subjective and was based on studying the characters' behavior in the context of other characters. Table I lists the character classes for the left context and Table II list the character classes for the right context. It should be noted that the number of classes as well as the specific characters in a given class is different for the two contexts. This is because the characters in Arabic have different influences on their left and right neighboring characters. For example, the character *rā'* (ر) has a prominent descender which affects the characters after it, i.e., to its left, but does not affect the characters before it, i.e., to its right. Thus, instead of modeling and training each unique tri-character form of an Arabic character shape, we only model and train the different contextual classes for a given character shape. Fig. 4 outlines the key steps involved in training the class-based contextual models and performing word recognition using them.

TABLE I: LIST OF CHARACTER CLASSES FOR THE LEFT CONTEXT.

Left contexts	
Class	**Example character shapes**
Ascenders	ـا ـل ـلـ
Descenders	ج ح خ ع غ
Core	ـب ـبـ ـت ـتـ ـث ـثـ ـد ـر
Loop	ـص ـض ـط ـظ ـف ـقـ ـمـ
Angular	ـجـ ـحـ ـخـ ـعـ ـغـ
Kaaf	ـك
Space	ا ب ت ن ل و د ر

TABLE II: LIST OF CHARACTER CLASSES FOR THE RIGHT CONTEXT.

Right contexts	
Class	**Example character shapes**
Ascenders	لـ ـلـ ـطـ ـظـ
Descenders	ـر ر و ـو د ـد
Core	بـ ـبـ تـ ـتـ ثـ ـثـ ـجـ ـحـ ـخـ ـعـ
Space	ا ـا ـل ـب ـت ـث ـج ـح ـخ ـع ـغ ـص ـض

Data preparation:

1. Training annotation:
   - The training set annotations are converted into tri-character forms.
   - The tri-character annotations are modified such that the left and the right contexts are mapped to the respective classes.
   - A list of unique class-based tri-characters is generated from the modified annotations.
2. Dictionary:
   - Words are defined in terms of class-based tri-characters.

Model Training:

1. The non-contextual models are initialized and trained using the standards procedures.
2. Each class-based tri-character model is initialized as a replica of its corresponding mono-character model which was trained in step–1.
3. The state-transition matrices of all the class-based tri-character models of a character are tied.
4. The class-based tri-character models are iteratively trained using the standard Baum-Welch training algorithm.
5. State level data-driven clustering is performed to tie similar states across different contextual forms of a character while preserving the state sequence. The thresholds are selected based on the recognition performance on the development set. The distance measure used to cluster similar states is as follows [14]:

$$d(x,y) = -\frac{1}{M}\sum_{m=1}^{M} log\left[b_y(\mu_{xm})\right] + log\left[b_x(\mu_{ym})\right]$$

   where M is the number of mixture components, μ_{xm} is the mean vector for the m^{th} mixture component of state x, and by(o) is the probability of generating observation o by state y which is given by:

$$b_y(o) = \sum_{m=1}^{M} c_{ym}\, N\left(o|\mu_{ym}, \Sigma_{ym}\right),$$

   where c_{ym} is the weight of the m^{th} mixture component of state y, and $N(o|\mu,\Sigma)$ is a multivariate Gaussian with mean vector μ and covariance matrix $\sum$.

6. The state-tied tri-character models are iteratively trained using the standard Baum-Welch training algorithm.

Decoding:

1. The class-based tri-character models along with the modified dictionary are used for recognizing the words. The standard Viterbi algorithm is used for decoding.

Fig. 4. Key steps involved in training the class-based contextual HMMs and text recognition utilizing the class-based contextual HMMs.

V. EXPERIMENTS AND RESULTS

In this section we will present the experiments we conducted and the results obtained for word recognition tasks using the IFN/ENIT database of handwritten Arabic words.

A. *Recognition task*

Our task is offline handwritten Arabic word recognition using the IFN/ENIT database [15]. The database consists of handwritten word images of names of Tunisian cities and towns divided into seven sets *a* to *f*, and *s*. The lexicon size is 937 names, but some names have two or more variations which are mainly due to the ligature models and the optional diacritics like *Shadda*. We experimented with the standard *train–test* configurations reported in the literature including the competitions using the IFN/ENIT database.

B. *Experimental details and discussions*

Our recognizer is a continuous HMM system built using the HTK tools [14]. Character shapes were used as basic modeling units, i.e., mono-models. There are a total of 178 unique character shapes in the IFN/ENIT dataset. Each of these is modeled as a separate HMM. We replaced few infrequent models having the diacritic *Shadda* over them with models representing the same character shapes without the *Shadda* over them. This led a reduced model set having a total of 157 models in our baseline system. Text images were preprocessed before feature extraction to correct the text baselines. We extracted nine geometrical features (the average number of ink pixels, the number of black-white transitions, the distance of the upper contour, the lower contour, and the center-of-gravity of the ink-pixels from the writing line, the orientation of the upper contour, the lower contour, and the center-of-gravity of the ink pixels) from each frame sliding over the word images. These features are adapted from [16] and [17]. We appended nine derivative features to the original features so the dimension of the feature vector is 18. Bakis topology was used for all the HMMs. Model length adaptation was performed by first training the models with large number of states and then removing those states from a model which has very low self-transition probabilities (the *slipping states* as presented in [18]). Other system parameters like the number of mixtures in each state were optimally configured based on the experiment configuration *abc–d* used for validation, i.e., the system was trained using sets *a, b,* and *c* and evaluated on set *d*. As a first step, a uniform initialization was done using the training data. In the next step, information from forced alignment of the training data was used to initialize individual HMMs using Viterbi initialization. This was followed by a number of iterations of Baum-Welch training. Finally the word hypothesis was made using Viterbi decoding.

For the contextual HMM system, different tri-character forms for each character shape are generated using the training-set transcriptions. Next, each tri-character model is initialized as a replica of its corresponding mono-character model which was trained before. The state-transition matrices of all the tri-character models of a character shape are tied. Next, the tri-character models are iteratively trained using the standard Baum-Welch training algorithm. State level data-driven clustering is performed to tie similar states across different contextual forms of a character while preserving the state sequence. The thresholds are selected based on the recognition performance on the development set. Finally, the state-tied tri-character models are, again, iteratively trained using the standard Baum-Welch training algorithm before performing decoding on the evaluation set. The steps followed for contextual HMMs are similar to what was described by Young and Woodland for continuous speech recognition [1], [14].

For the class-based contextual HMM system, the system initialization, training, and decoding is performed as presented in Fig. 4. The evaluation results on the development set–*d* for the three systems, in addition to the key system statistics like total number of HMMs in the system as well as the total number of states in the system (using the training sets *a*–*c*), are presented in Table III. The total number of states in the system is more indicative as many HMMs will have their states tied to other HMMs in the system. The reduction in the number of HMMs after tying happens when all the corresponding states of two HMMs are tied together; thereby, merging the two HMMs as one. The recognition results are shown in terms of *Word Error Rate* (*WER*). It can be seen from the table that improvement in WER is reported for both the contextual HMM system, i.e., the standard contextual system and the class-based contextual system. However, the improvement in WER for the class-based contextual system is much higher as compared to the improvement obtained by using the standard contextual system. Moreover, there is significant reduction in the number of HMMs and the total number of states in the class-based contextual HMM system as compared to the standard contextual HMM system. In fact, the total number of states in the class-based contextual HMM system is only a fraction higher when compared to the non-contextual system (almost a 20% increase). This results in a compact contextual HMM system.

Table IV presents the recognition results on all the training–test configurations: *abc–d, abcd–e, abcde–f,* and *abcde–s*. It can be seen from the table that improvements are observed in all the experimental configurations for the class-based contextual HMM system when compared to the non-contextual HMM system (the baseline system) as well as the standard contextual HMM system. For one experiment configuration (*abcd–e)* the standard contextual HMM system shows a lower performance than the baseline system whereas the class-based contextual HMM system still shows a small improvement over the baseline system. More importantly, both the contextual systems perform significantly better than the baseline system on the set *s* which is, relatively, a difficult evaluation set. Again, the class-based contextual system performs better than the standard contextual system. A comparison of the recognition rates of our system with other state-of-the-art systems evaluated on the IFN/ENIT database is presented in Table V. From the table we can see that the recognition rates of our systems are comparable to the other top systems.

TABLE III. COMPARISON OF THE NUMBER OF HMM MODELS, THE TOTAL NUMBER OF STATES, AND THE WER'S FOR THE THREE DIFFERENT SYSTEMS USING THE IFN/ENIT TRAINING SETS *a,b*, AND *c* AND EVALUATION ON SET *d*.

System Description	***Number of HMMs***	***Total number of states***	***WER***
Character-shape HMM system	157	1534	4.01
Contextual HMM system	4575 (3212 after tying)	38512 (2767 after tying)	3.86
Class-based contextual HMM system	626 (359 after tying)	5775 (1845 after tying)	3.37

VI. CONCLUSIONS

Text recognition is an interesting, as well as a challenging, research area in the field of pattern recognition. HMMs are widely used classifier for text recognition. Contextual modeling is an important aspect of any HMM-based speech recognition system; however, its use in the domain of text recognition is not that universal. In this paper we presented a detailed discussion and investigation on the need and the use of contextual HMMs for Arabic text recognition. The justifications for the use of contextual HMMs for handwritten Arabic text recognition, in addition to the issues faced when setting a contextual HMM system for Arabic text recognition, were presented. Class-based contextual modeling for HMM-based handwritten Arabic text recognition was presented which alleviates some of the problems listed. Experiments conducted on IFN/ENIT database of handwritten Arabic text demonstrated the benefits of using class-based contextual HMM system over a standard contextual HMM system for Arabic text recognition.

TABLE IV: SUMMARY OF THE RESULTS (IN WER'S) FOR HANDWRIITEN TEXT RECOGNITION USING THE THREE DIFFERETN SYSTEMS.

System Description	***WER***			
	Evaluation Sets			
	d	***e***	***f***	***s***
Baseline system	4.01	8.47	9.87	17.74
Contextual HMMs	3.86	8.81	9.81	15.89
Class-based contextual HMMs	**3.37**	**8.27**	**9.66**	**15.38**

TABLE V: COMPARISON WITH OTHER STATE-OF-THE-ART SYSTEMS EVALUTED ON THE IFN/ENIT DATABASE.

Systems	***Train–Test Configurations***			
	abc–d	***abcd–e***	***abcde–f***	***abcde–s***
UPV-PRHLT [19]	4.80	6.10	7.80	15.38
RWTH-OCR [20], [21]	3.47	7.26	7.80	15.45
Azeem and Ahmed [22]	2.30	6.56	6.90	15.20
Ahmad et al. [12]	2.78	6.48	7.85	14.88
Giménez et al. [23]	4.70	6.10	7.80	15.38
Ahmad and Fink [24]	1.92	5.07	7.70	15.45
Stahlberg and Vogel [7]	2.40	6.10	6.80	11.50
Present work	3.37	8.27	9.66	15.38

ACKNOWLEDGMENT

The authors would like to acknowledge the support provided by King Fahd University of Petroleum and Minerals (KFUPM) in conducting this research.

REFERENCES

[1] S. J. Young and P. C. Woodland, "State clustering in hidden Markov model-based continuous speech recognition," *Comput. Speech Lang.*, vol. 8, no. 4, pp. 369–383, Oct. 1994.

[2] A. Kosmala, J. Rottland, and G. Rigoll, "Improved on-line handwriting recognition using context dependent hidden Markov models," in *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, 1997, vol. 2, pp. 641–644.

[3] G. Fink, G. Fink, T. Plotz, and T. Plotz, "On the Use of Context-Dependent Modeling Units for HMM-Based Offline Handwriting Recognition," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*, 2007, vol. 2, pp. 729–733.

[4] R. Prasad, S. Saleem, M. Kamali, R. Meermeier, and P. Natarajan, "Improvements in hidden Markov model based Arabic OCR," in *19th International Conference on Pattern Recognition*, 2008, pp. 1–4.

[5] M. Hamdani, A. E.-D. Mousa, and H. Ney, "Open Vocabulary Arabic Handwriting Recognition Using Morphological Decomposition," in *12th International Conference on Document Analysis and Recognition*, 2013, pp. 280–284.

[6] M. Hamdani, P. Doetsch, M. Kozielski, A. E.-D. Mousa, and H. Ney, "The RWTH Large Vocabulary Arabic Handwriting Recognition System," in *11th IAPR International Workshop on Document Analysis Systems*, 2014, pp. 111–115.

[7] F. Stahlberg and S. Vogel, "The QCRI Recognition System for Handwritten Arabic," in *18th International Conference on Image Analysis and Processing — ICIAP 2015*, 2015, pp. 276–286.

[8] H. Cao, P. Natarajan, X. Peng, K. Subramanian, D. Belanger, and N. Li, "Progress in the Raytheon BBN Arabic Offline Handwriting Recognition System," in *14th International Conference on Frontiers in Handwriting Recognition*, 2014, pp. 555–560.

[9] M. Hamdani, P. Doetsch, and H. Ney, "Improvement of Context Dependent Modeling for Arabic Handwriting Recognition," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, 2014, pp. 494–499.

[10] R. El-Hajj, C. Mokbel, and L. Likforman-Sulem, "Recognition of Arabic handwritten words using contextual character models," *Doc. Recognit. Retr. XV-SPIE*, 2008.

[11] P. Natarajan, R. Prasad, H. Cao, K. Subramanian, S. Saleem, D. Belanger, S. Vitaladevuni, M. Kamali, and E. MacRostie, "Arabic Text Recognition Using a Script-Independent Methodology: A Unified HMM-Based Approach for Machine-Printed and Handwritten Text," in *Guide to OCR for Arabic Scripts*, V. Märgner and H. El Abed, Eds. London: Springer London, 2012, pp. 485–505.

[12] I. Ahmad, G. A. Fink, and S. A. Mahmoud, "Improvements in Sub-character HMM Model Based Arabic Text Recognition," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, 2014, pp. 537–542.

[13] Y. S. Elarian, I. Ahmad, S. Awaida, W. G. Al-Khatib, and A. Zidouri, "Arabic Ligatures: Analysis and Application in Text Recognition," in *13th International Conference on Document Analysis and Recognition (ICDAR 2015)*, 2015, pp. 896–900.

[14] S. J. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, 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.

[15] M. Pechwitz, S. S. Maddouri, V. Märgner, N. Ellouze, and H. Amiri, "IFN/ENIT - Database of Handwritten Arabic Words," in *7th Colloque International Francophone sur l'Ecrit et le Document , CIFED 2002*, 2002, pp. 129–136.

[16] M. Wienecke, G. A. Fink, and G. Sagerer, "Toward automatic video-based whiteboard reading," *Int. J. Doc. Anal. Recognit.*, vol. 7, no. 2, pp. 188–200, 2005.

[17] U.-V. Marti and H. Bunke, "Handwritten sentence recognition," in *Proc. of 15th International Conference on Pattern Recognition. ICPR-2000*, 2000, pp. 463–466.

[18] Z. Jiang, X. Ding, L. Peng, and C. Liu, "Analyzing the information entropy of states to optimize the number of states in an HMM-based off-line handwritten Arabic word recognizer," in *21st International Conference on Pattern Recognition (ICPR), 2012*, 2012, pp. 697–700.

[19] V. Margner and H. El Abed, "ICFHR 2010 - Arabic Handwriting Recognition Competition," in *2010 12th International Conference on Frontiers in Handwriting Recognition*, 2010, pp. 709–714.

[20] P. Dreuw, D. Rybach, G. Heigold, and H. Ney, "RWTH OCR: A Large Vocabulary Optical Character Recognition System for Arabic Scripts," in *Guide to OCR for Arabic Scripts SE - 9*, V. Märgner and H. El Abed, Eds. Springer London, 2012, pp. 215–254.

[21] V. Margner and H. El Abed, "ICDAR 2011 - Arabic Handwriting Recognition Competition," in *2011 International Conference on Document Analysis and Recognition*, 2011, pp. 1444–1448.

[22] S. Azeem and H. Ahmed, "Effective technique for the recognition of offline Arabic handwritten words using hidden Markov models," *Int. J. Doc. Anal. Recognit.*, vol. 16, no. 4, pp. 399–412, 2013.

[23] A. Giménez, I. Khoury, J. Andrés-Ferrer, and A. Juan, "Handwriting word recognition using windowed Bernoulli HMMs," *Pattern Recognit. Lett.*, vol. 35, pp. 149–156, Jan. 2014.

[24] I. Ahmad and G. A. Fink, "Multi-stage HMM based Arabic text recognition with rescoring," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 751–755.