2014 14th International Conference on Frontiers in Handwriting Recognition

Improvement of Context Dependent Modeling For Arabic Handwriting Recognition

Mahdi Hamdani, Patrick Doetsch, Hermann Ney
Human Language Technology and Pattern Recognition Computer Science Department
RWTH Aachen University, 52056 Aachen, Germany

***Abstract*—This paper proposes the improvement of context dependent modeling for Arabic handwriting recognition. Since the number of parameters in context dependent models is huge, CART trees are used for state tying. This work is based on a new set of questions for the CART tree construction based on a "lossy mapping" categorization of the Arabic shapes. The used system is a combination of Hidden Markov Models and Recurrent Neural Networks using the hybrid approach. A comparison between a Neural network trained using the baseline labels and another one based on the CART tree labels is done. The experimental results show that the use of the CART labels for the Neural Network training beneficial. The lossy mapping based CART tree performed better than the baseline system. An absolute improvement of 2.9% in terms of Word Error Rate is performed on the test set of the OpenHaRT database.**

Keywords*—*Arabic Handwriting Recognition, Context Dependent Modeling, Hidden Markov Models, Recurrent Neural Networks

I. Introduction

The improvement of Arabic handwriting recognition systems is a more challenging task. This can be seen by the improvement of the results in the evaluations organized regularly in this field [1], [2]. The designed systems are more robust and the tasks are also more difficult with larger datasets and/or harder inputs [3].

Hidden Markov Models (HMMs) are one of the most successful techniques used for large vocabulary handwriting recognition [4]. Recurrent Neural Networks (RNNs) are also one of the techniques that ameliorated the systems for handwriting recognition [5]. HMMs/RNNs combination was also used for handwriting recognition [6].

HMMs are used generally in an analytical approach in which the basic model is the character. Handwritten characters are heavily influenced by their context. Therefore, the use of the contextual information is very important in modeling characters. Context Dependent Modeling is one of the standard components used in building speech recognition systems [7], [8]. This approach was also successful for handwriting recognition systems.

Fink and Plotz proposed in [9] the use of context dependent modeling for HMMs based off-line handwriting recognition. The paper discusses the utility of using such models for off-line handwriting recognition. The number of parameters to estimate is huge for the context dependent model. Therefore state tying using a data driven approach is proposed. The used method merges tri-character units in a data-driven manner on the level of individual HMM states by applying an agglomerative clustering procedure. The authors applied the proposed model for English handwriting recognition.

Prasad et al. proposed context dependent glyph modeling for Arabic printed text recognition in [10]. The glyphs are a transformation of the base-form characters transcripts based on the character shape. The glyph modeling is improved by context dependent modeling. The context dependent models are trained using decision tree based state tying. The decision tree is trained using predefined questions regarding the character shape.

Contextual and dynamic information is used in [11] for handwriting recognition. Feature extraction is performed using a dynamic approach using derivative features. A horizontal derivation of feature vectors in order to capture a wider temporal context at the frame level. Context dependent modeling is used with decision trees for state tying. The context dependent and independent systems are combined using a neural network. The proposed approach is validated using Latin and Arabic handwriting datasets.

Using successful approaches from speech recognition to improve handwriting recognition is very important but not sufficient. This paper proposes the improvement of context dependent modeling for Arabic handwriting recognition. The state tying is improved by using a more robust set of questions. Moreover the CART labels are used to train an RNN. We used a hybrid HMM/RNN system for the validation of the proposed modeling.

The rest of the paper is organized as follows. Section II will give an overview of the used recognition system. The context dependent modeling is then presented in Section III followed by the Arabic shapes classification in Section IV. Finally the Experimental results and the conclusion and future work are presented in the last two Sections.

II. System Overview

A. Appearance based Feature Extraction

The used features are simply the pixels values. The input images are first scaled to a fixed height. After that, a sliding window is applied following the direction of the handwriting with a maximum overlap, i.e the window is moved by 1 pixel. The center of gravity (COG) of the black pixels is calculated for each window. The window is re-positioned such that the COG will be in the center of the window. The pixel values resulting from the repositioned windows are used as features vectors. Figure 1 presents an example of the used sliding window.

2167-6445/14 $31.00 © 2014 IEEE
DOI 10.1109/ICFHR.2014.89

Fig. 1. Feature extraction using a sliding window: pixel values are extracted from a repositioned window such that the center of gravity is in the middle.

B. Visual Model

The baseline system is based on Gaussian Mixture Hidden Markov Models (GHMMs). The main goal is to search an unknown word sequence $w_1^N := w_1, \ldots, w_N$, for which the sequence of features $x_1^T := x_1, \ldots, x_T$ fits best to the trained models. We maximize the posterior probability $p(w_1^N|x_1^T)$ over all possible word sequences w_1^N with unknown number of words N. This is described by the Bayes' decision rule presented in Equation 1.

$$x_1^T \rightarrow \hat{w}_1^N(x_1^T) = \arg\max_{w_1^N} \left\{ p^\kappa(w_1^N) p(x_1^T|w_1^N) \right\} \qquad (1)$$

with κ being a scaling exponent of the language model.

The used model is a GHMM with a Bakis topology, i.e. each state has a transition to the two next states. Each Gaussian is shared between two successive states. This property guaranty that each Gaussian is visited at least once. Further, we do not train the state transition probabilities but use fixed time distortion penalties instead. The size of the Arabic characters are very different. The number of HMM states is therefore estimated using the so called Model Length Estimation (MLE) presented in [12]. The characters are divided into MLE labels with one Gaussian for each them. Each character have a predefined number of MLE labels which is dependent on the number of states. Statistics on the state occupancy of each character are first collected from an initial good alignment. The number of states is finally the median value of all the performed statistics.

C. HMM/RNN combination

We apply the HMM trained on the estimated MLE labels in a forced alignment mode to the training data in order to generate a concrete labeling of all frames extracted during feature extraction. Afterwards the labeled frames are fed into a long short-term memory recurrent neural network (LSTM-RNN) which is trained to estimate the posterior distribution over the MLE labels for each frame. Two combination schemes for the final HMM/LSTM-RNN system combinations are common. In contrast to the tandem approach, where the activations of the RNN are mapped to some feature vector that is used to train a new HMM recognition system, we directly use the posteriors estimated by the softmax output layer of the LSTM-RNN to simulate the emission probability distributions of a previously trained HMM according to Equation 2. This combination technique is referred to as hybrid approach.

$$p(x_t|s_t, w) = \frac{p(s_t, w|x_t) \cdot p(x_t)}{p(s_t, w)^\alpha} \qquad (2)$$

where α is the priori scaling factor. Note that we can drop the $p(x_t)$ term on the right hand side of Equation 2 when searching for the most likely word sequence. In Figure 2 the integration of the LSTM-RNN is illustrated as a flow network.

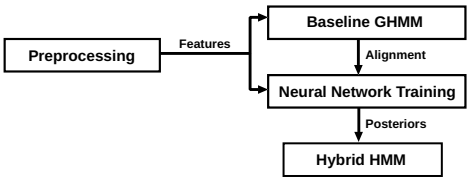

Fig. 2. Flow network of the hybrid HMM/LSTM-RNN combination approach. A baseline HMM creates a frame-wise labeling of the training data that is used to train the LSTM-RNN. During recognition the posterior estimates given by the output layer of the LSTM-RNN are used to simulate the emission probabilities of an HMM.

In our experiments results of the hybrid approach are competitive with the tandem approach with the advantage of avoiding the training of a new HMM. Therefore we opted for the use of the hybrid approach in this work. Decoding is performed in the traditional HMM framework. We chose this approach over CTC [5] because it enables us to include higher n-gram models for language modeling during decoding. The topology of the network consists of an input layer with one unit for each of the 35 components of the feature vector, three hidden layers with 500 LSTM memory cells in each layer and an output layer containing one unit per label or CART state.

D. Language Model

The n-gram LM is trained using text data collected from freely available newspapers and forums [13]. The number of running words for LM training is about 1 billion words. The LM training text is preprocessed before training in order to normalize the Arabic text. Indian digits , which are widely used in Arabic text, are mapped to Arabic (the lexicon contains only Arabic digits). The numbers are reversed including optional decimal points and then the digits of the numbers are separated by spaces. Punctuation and special characters are separated from the words. These two steps are important to reduce the noisy text which allows to have a better distribution of the probabilities.

The N most frequent words in the training corpus are used in the vocabulary for all the conducted experiments. The LM is a standard n-gram trained using the SRILM toolkit [14] with interpolated Kneser-Ney smoothing.

III. Context Dependent Modeling

The idea of context dependent modeling in the HMM framework is that the HMM will model a character within its context. The context is defined by the previous and next characters. This advantage of this type of modeling is very clear for handwriting. The writing style of the current stroke is always affected by its context: previous and future strokes. Figure 3 shows an example of the importance of context in Arabic handwriting.

Modeling characters within context cause practical problems for parameter estimation. The number of free parameters is huge if compared to the case of context independent modeling. State tying is essential to reduce the number of parameters

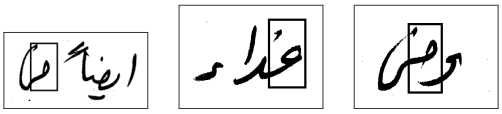

Fig. 3. Example of context dependency in Arabic handwriting: the character ف in its beginning form is written differently in different contexts by the same writer.

[15]. State tying aims to determine the states which share the same Gaussian distributions in the context of GHMMs.

There are multiple methods for state tying. The data driven method is based on state clustering. The main drawback of this method is that the tri-characters which are not seen in the training data are not tied to any model in the clusters. The solution is to use backing off models with simple generalization of the tri-characters to di-characters and mono-characters.

Decision trees are one of the most used methods in speech recognition systems. The objective is to tie the states which are similar. Decision trees are binary trees in which the inner nodes are tagged with questions and the leaves are tagged with class labels.

The construction of the CART tree is described with more details in [16]. The tri-characters states with similar emission probability distributions are tied together. A distance measure is used to define the similarity between the states. For calculating this distance, a single Gaussian distribution is estimated for every tri-character state using a baseline alignment of the data. The states are grouped into a root node AB. This node is then split using the question (out of the predefined set of questions) which gives the biggest likelihood improvement $LLI(A, B)$ for the child nodes A and B. The calculation of the likelihood improvement is presented in Equation 3.

$$\begin{aligned} LLI(A,B) &= LL(AB) - (LL(A) + LL(B)) \\ &= -\frac{1}{2}\left(n_A \sum_{d=1}^{D} log\left[\frac{\hat{\sigma}_{d,AB}}{\hat{\sigma}_{d,A}}\right]^2 + n_B \sum_{d=1}^{D} log\left[\frac{\hat{\sigma}_{d,AB}}{\hat{\sigma}_{d,B}}\right]^2\right) \end{aligned} \quad (3)$$

where n_X is the number of observations for node X, D is the dimensionality of the feature vector, and $\sigma_{d,X}$ the variance of component d of node X. The question is assigned to the tree node and the tri-character states are distributed over the two child nodes according to the question. This procedure is repeated until the predefined maximum number of leaves is reached. The resulting state tree is used for both training and recognition. An initial "good" segmentation should be used to calculate the likelihood improvement. The alignments generated by the context independent model are generally used.

The questions used in the CART construction concern the data to be classified using the tree. The questions are generally predefined using prior knowledge about the data. There are standard questions used in speech recognition systems based on phonetic properties (e.g. " Is the left context a vowel? "). The phonetic classes are predefined in the system. The questions used for CART trees generation in speech recognition systems are well defined and are almost standard for this field. Table I gives some examples of the phoneme classes used in the speech recognition systems.

TABLE I. EXAMPLES OF THE QUESTIONS USED FOR STATE TYING IN SPEECH RECOGNITION (ENGLISH)

Classes	**Phonemes Examples**
Vowels	ao aa iy uw eh ih uh ah ae
Diphthongs	ey ay ow aw oy
Semi vowels	y w
Liquids	l r

There is no standard set of questions used for CART trees construction in Arabic handwriting recognition. The adopted list of questions used in our previous work in [17] was based in a rough classification of the Arabic characters. The amelioration of this work can be done by the definition of questions based on a more robust classification. The next Section proposes the use of a well-known model to classify the Arabic characters based on their shapes.

IV. DECISION TREE QUESTIONS FOR ARABIC HANDWRITING

Arabic handwriting is cursive and the context has an influence on the way of writing. The position of a character as well as its context are important to define its shape. These basic information concerning Arabic handwriting style must be exploited to build a handwriting recognition system.

Generally, an Arabic character can be written in different ways depending on its position. We can find 1 to 4 variants for each character. Basically, the Arabic handwriting contains 28 letters. If we take into account position dependency we can reach more than 100 different forms. We have to say that some of these characters can be rare if using limited training data. Figure 4 presents some examples of characters written differently in different positions.

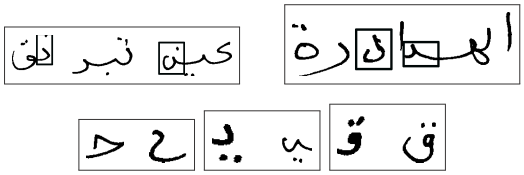

Fig. 4. Examples of the same characters in different positions in the word.

As presented in the figure, the character ن for example has a totally different shape in the positions start and end (left top of Figure 4). All the other images illustrate examples of some characters which have totally different shapes when the position of the character changes in the word. The information regarding the character position is generally unavailable for the Arabic handwriting databases. The IfN/ENIT database [18] contains additional information in the transcriptions regarding position and special shapes of characters like ligatures. This database is for small vocabulary recognition since it consists of images of Tunisian city names.

There are characters in the Arabic language which are never linked to the next context like أ, و, ر, etc. Therefore, the next character is written in its start form. There are 4 possible forms of the characters: "start", "middle", "end" and "alone" forms. The position dependent characters are tagged with B for begin, M for middle, E for end and A for alone.

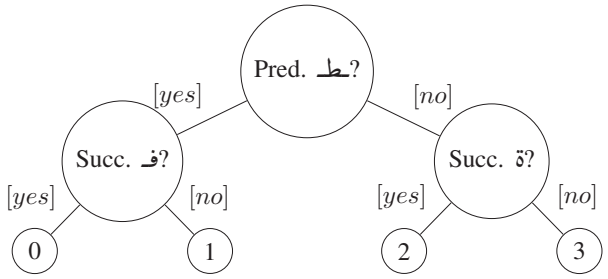

Fig. 5. Example of a CART tree: the root of the tree contains the question "is the predecessor character belongs to the class ـط?", if the response is "yes" the next question that comes is "is the successor character belongs to the class ـف?", if "yes" then the selected mixture index is "0".

The Arabic characters can be classified depending on their shapes. In [19] the Arabic shapes are divided using a "lossy mapping". The dots and the "hamzas" are deleted to classify the similar shapes together. The classification of the Arabic characters using the "lossy mapping" model is presented in Table II.

TABLE II. CLASSIFICATION OF THE ARABIC CHARACTERS USING THE "LOSSY MAPPING" MODEL

B	M	E	A
			ء
		ـا ـإ ـأ	ا إ أ
ثـ تـ بـ	شـ تـ بـ	ـش ـت ـب	ش ت ب
نـ	نـ	ـن	ن
ئـ يـ	ئـ يـ	ـى ـئ ـي	ى ئ ي
جـ خـ حـ	جـ خـ حـ	ـج ـخ ـح	ج خ ح
		ـذ ـد	ذ د
		ـذ ـر	ذ ر
شـ سـ	شـ سـ	ـش ـس	ش س
ضـ صـ	ضـ صـ	ـض ـص	ض ص
ظـ طـ	ظـ طـ	ـظ ـط	ظ ط
غـ عـ	غـ عـ	ـغ ـع	غ ع
فـ	فـ	ـف	ف
قـ	قـ	ـق	ق
كـ	كـ	ـك	ك
لـ	لـ	ـل	ل
مـ	مـ	ـم	م
هـ	هـ	ـه ـة	ه ة
		ـؤ ـو	ؤ و

Each cell of the Table II is used in the questions used in the CART tree construction. The colored characters are the representatives of the questions. The special characters like punctuations are divided into two major questions: is the character big or small?. The used corpus described in the next Section contains also Latin characters. Again two questions are added to know if the the neighboring character is written with capital or lowercase letters. More questions can be added for the Latin characters but the described system is Arabic and the number of Latin characters used in the system is not big. Figure 5 presents an example of the used CART tree.

As explained before, the questions which are selected in the final CART are dependent on the statistics collected from initial alignments. These questions should maximize the likelihood improvement LLI as presented in Equation 3. The leaves of the tree are simply the mixtures indexes of the GHMM.

V. EXPERIMENTAL RESULTS

A. Database

The OpenHaRT database is used in the validation step. This dataset is provided by the MADCAT[1] (Multilingual Automatic Document Classification Analysis and Translation) program within the context of the OpenHART evaluation[2]. The data consists of more than 40k handwritten pages with text chosen from web forums and newspapers. Table III gives statistics detailing the used data.

TABLE III. OPENHART DATASET STATISTICS

	Train set	Dev set	Test set
# of pages	42,148	470	633
# of paragraphs	182,879	1,832	3,144
# of words	4,361,056	48,832	77,628
# of characters	23,324,011	266,121	349,422

B. System Parameters

The recognition system is based on the RWTH Aachen University Open Source Speech Recognition Toolkit [20]. LSTM-RNN training was performed with a GPU based implementation using the Theano Python library [21].

Three different systems are compared in the experiments. The baseline system is the context independent recognition system using the MLE labels (we will call it "Context Independent"). Two systems based on the CART labels are also compared which are referred respectively as "Baseline CART" and "Lossy mapping CART" (this work). All systems are hybrid HMM/LSTM-RNN combinations. The LSTM-RNNs of the three systems were trained on roughly 90% of the training data (214.8M frames). About 10% (21.6M frames) of the training data was used as hold-out set in order to detect convergence of the LSTM-RNN training. The gradient descent training converged after 3-5 epochs using a fixed learning rate of 0.001 with momentum term. The parameters of the compared systems are presented in Table IV.

TABLE IV. COMPARISON OF TRAINING/RECOGNITION PARAMETERS FOR THE DIFFERENT SYSTEMS

	Context Independent	Baseline CART	Lossy mapping CART
# of labels	1680	5000	6000
# of densities	420,404	1,587,066	1,615,289
# of weights	11,829,698	15,153,018	16,154,018

The number of densities in the baseline HMM is obviously increasing if the number of labels increases. Similarly each additional label requires the corresponding unit in the output layer of an LSTM-RNN to be connected to the 500 units of the previous hidden layer which increases the number of parameters accordingly. The number of CART labels is not fully tuned for this work. However, former experiments using the old CART showed that a larger CART complexity did not lead to further improvements in recognition performance.

Recognition is done by a 400k vocabulary with a 4-gram closed vocabulary language model. This vocabulary gave a perplexity of 362 on the dev data. The Out-Of-Vocabulary

[1] http://www.itl.nist.gov/iad/mig/tests/madcat/index.html

[2] NIST Open Handwriting Recognition and Translation Evaluation (OpenHART 2013) http://www.nist.gov/itl/iad/mig/hart2013.cfm

(OOV) rate is 4.52% on the dev set and 7.65% on the test set. A language model scale κ of 10 gave best results in our experiments.

C. Results

The Word Error Rates (WER) and the Character Error Rates (CER) of the different systems are reported in Table V.

TABLE V. RESULTS ON THE OPENHART DATABASE (IN %)

	dev		test	
	WER	**CER**	**WER**	**CER**
Context Independent	19.1	8.9	22.8	11.9
Baseline CART	16.4	6.4	20.3	8.9
Lossy mapping CART	15.9	5.8	19.9	8.3

The parameters of LM scale, priori scale and time distortion penalties are optimized using the dev set. The results show that the system performance of the context dependent systems are better than the context independent one. The new CART improved the system performance from 16.4% to 15.9% in terms of WER for the dev set. The improvement is also shown in terms of CER from 6.4% to 5.8%.

When measuring the number of missclassified frames during HMM training on the designated hold-out set, we observed that the CART systems arrive at a frame error of 24.7% while the context independent system has a frame error that is about 1% smaller absolutely. However, in the final hybrid HMM/LSTM-RNN recognition we see that the posterior estimates over the highly differentiated labels of the CART models provide more information to the HMM during decoding. The results on the test set show a better improvement when using the lossy mapping CART. The WER goes down from 20.3% to 19.9% and the CER from 8.9% to 8.3%. An absolute improvement of 2.9% in terms of WER and 3.8% of CER is performed by using the context dependent labels and by using the lossy mapping CART. The new lossy mapping CART gave an absolute improvement of 0.4% in WER and 0.6% in CER if compared to the old CART.

D. Statistical Significance Test

The absolute improvements in terms of WER from the system based on the baseline CART to the lossy mapping cart system are 0.5% and 0.4% respectively for the dev and test sets. A statical significance test based on the work presented by Bisani and Ney in [22] is performed. The Probability Of Improvement (POI) is presented in Table VI.

TABLE VI. STATISTICAL SIGNIFICANCE TEST: COMPARISON BETWEEN THE SYSTEMS BASED ON THE BASELINE AND THE LOSSY MAPPING CARTS

	dev	**test**
ΔWER (in %)	0.5	0.4
POI (in %)	99.97	99.98

The POIs on the dev and test sets show that the improvements are statistically significant.

VI. CONCLUSION AND FUTURE WORK

This paper proposes the improvement of context dependent modeling for Arabic handwriting recognition. A new CART based on a lossy mapping technique is used to build the CART questions for Arabic. The dots and diacritics are deleted and the similar characters are grouped together in classes. The CART labels are used to train an LSTM-RNN to estimate the posterior probability of an HMM. The recognition system is an LSTM-RNN/HMM combination using the hybrid approach. The recognition system is based on a 4-gram LM and a 400k vocabulary for the recognition. The results of the system shows that context dependent modeling ameliorate the system performance. An absolute improvement of 2.9% in terms of WER and 3.8% of CER is shown for the test set. The use of the lossy mapping CART is also beneficial by improving the WER from 20.3% to 19.9% in terms of WER for the test set.

The presented system in this paper is a hybrid HMM/LSTM-RNN system without writer adaptation. Better results could be obtained with the tandem approach which can be combined with discriminative HMM training using the Minimum Phone Error (MPE) criterion and Constrained Maximum Likelihood Linear Regression (CMLLR) for writer adaptation as used in [17]. The OOV rates that are observed for the dev and test sets are relatively high. Open vocabulary approaches can be used in the future to improve the system performance (e.g. [13], [23]).

ACKNOWLEDGMENT

This work was partially supported by a Google Research Award and by the Quaero Program, funded by OSEO, French State agency for innovation. H. Ney was partially supported by a senior chair award from DIGITEO, a French research cluster in Ile-de-France.

REFERENCES

[1] A. Tong, M. Przybocki, V. Märgner, and H. E. Abed, "Nist 2013 open handwriting recognition and translation (openhart13) evaluation," in *International Workshop on Document Analysis Systems*, Tours à Loire Valley, France, Apr. 2014.

[2] V. Märgner and H. Abed, "ICDAR 2011 - Arabic handwriting recognition competition," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, Sept 2011, pp. 1444–1448.

[3] (2013, Jul.) Moyens automatisés de reconnaissance de documents écrits, the MAURDOR compain. [Online]. Available: www.maurdor-campaign.org

[4] A. L. Koerich, R. Sabourin, and C. Y. Suen, "Large vocabulary off-line handwriting recognition: A survey," *Pattern Analysis and Applications*, vol. 6, pp. 97–121, 2003.

[5] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. MIT Press, 2009, pp. 545–552.

[6] P. Dreuw, P. Doetsch, C. Plahl, and H. Ney, "Hierarchical hybrid MLP/HMM or rather MLP features for a discriminatively trained gaussian HMM: a comparison for offline handwriting recognition," in *IEEE International Conference on Image Processing*, Brussels, Belgium, Sep. 2011. [Online]. Available: http://www-i6.informatik.rwth-aachen.de/rwth-ocr/

[7] R. Schwartz, Y. Chow, O. Kimball, S. Roucos, M. Krasner, and J. Makhoul, "Context-dependent modeling for acoustic-phonetic recognition of continuous speech," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '85)*, vol. 10, 1985, pp. 1205–1208.

[8] K. F. Lee, "Context-dependent phonetic hidden markov models for speaker-independent continuous speech recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 38, no. 4, pp. 599–609, Apr 1990.

[9] G. Fink and T. Plotz, "On the use of context-dependent modeling units for hmm-based offline handwriting recognition," in *International Conference on Document Analysis and Recognition (ICDAR))*, vol. 2, sept. 2007, pp. 729 –733.

[10] R. Prasad, S. Saleem, M. Kamali, R. Meermeier, and P. Natarajan, "Improvements in hidden markov model based Arabic ocr," in *International Conference on Pattern Recognition (ICPR)*, 2008, pp. 1–4.

[11] A.-L. Bianne-Bernard, F. Menasri, R. Al-Hajj Mohamad, C. Mokbel, C. Kermorvant, and L. Likforman-Sulem, "Dynamic and contextual information in hmm modeling for handwritten word recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 2066–2080, Oct. 2011. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2011.22

[12] P. Dreuw, D. Rybach, C. Gollan, and H. Ney, "Writer adaptive training and writing variant model refinement for offline Arabic handwriting recognition," in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR)*, Barcelona, Spain, Jul. 2009.

[13] M. Hamdani, A. El-Desoky Mousa, and H. Ney, "Open vocabulary Arabic handwriting recognition using morphological decomposition," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 280–284.

[14] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *International Conference on Spoken Language Processing (ICSLP)*, vol. 2, Denver, Colorado, USA, Sep. 2002, pp. 901 – 904.

[15] K. Beulen, E. Bransch, and H. Ney, "State-tying for context dependent phoneme models," in *European Conference on Speech Communication and Technology*, vol. 3, Rhodes, Greece, Sep. 1997, pp. 1179–1182.

[16] K. Beulen and H. Ney, "Automatic question generation for decision tree based state tying," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seattle, WA, USA, May 1998, pp. 805–808.

[17] M. Hamdani, P. Doetsch, M. Kozielski, A. El-Desoky Mousa, and H. Ney, "The RWTH large vocabulary Arabic handwriting recognition system," in *International Workshop on Document Analysis Systems*, Tours à Loire Valley, France, Apr. 2014.

[18] M. Pechwitz, S. S. Maddouri, V. Märgner, N. Ellouze, and H. Amiri, "IfN/ENIT - database of handwritten Arabic words," in *In Proc. of CIFED 2002*, 2002, pp. 129–136.

[19] Y. Elarian and F. Idris, "A lexicon of connected components for Arabic optical text recognition," in *2010 - First International Workshop on Frontiers in Arabic Handwriting Recognition*, Istanbul, Turkey, Sep. 2011.

[20] D. Rybach, S. Hahn, P. Lehnen, D. Nolden, M. Sundermeyer, Z. Tüske, S. Wiesler, R. Schlüter, and H. Ney, "RASR - the RWTH Aachen university open source speech recognition toolkit," in *IEEE Automatic Speech Recognition and Understanding Workshop*, Hawaii, USA, Dec. 2011.

[21] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2010.

[22] M. Bisani and H. Ney, "Bootstrap estimates for confidence intervals in asr performance evaluation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Montreal, May 2004, pp. 409–412.

[23] M. Kozielski, D. Rybach, S. Hahn, R. Schlüter, and H. Ney, "Open vocabulary handwriting recognition using combined word-level and character-level language models," in *ICASSP*, 2013, pp. 8257–8261.