Open Vocabulary Arabic Handwriting Recognition Using Morphological Decomposition

Mahdi Hamdani[1], Amr El-Desoky Mousa[1], Hermann Ney[1,2]
[1] Human Language Technology and Pattern Recognition Group - RWTH Aachen University, Germany
[2] Spoken Language Processing Group, LIMSI CNRS, Paris, France
{hamdani, desoky, ney}@cs.rwth-aachen.de

***Abstract*—The use of Language Models (LMs) is a very important component in large and open vocabulary recognition systems. This paper presents an open-vocabulary approach for Arabic handwriting recognition. The proposed approach makes use of Arabic word decomposition based on morphological analysis. The vocabulary is a combination of words and subwords obtained by the decomposition process. Out Of Vocabulary (OOV) words can be recognized by combining different elements from the lexicon. The recognition system is based on Hidden Markov Models (HMMs) with position and context dependent character models. An n-gram LM trained on the decomposed text is used along with the HMMs during the search. The approach is evaluated using two Arabic handwriting datasets. The open vocabulary approach leads to a significant improvement in the system performance. Two different types experiments for two Arabic handwriting recognition tasks are conducted in this work. The proposed approach for open vocabulary allows to have an absolute improvement of up to 1% in the Word Error Rate (WER) for the constrained task and to keep the same performance of the baseline system for the unconstrained one.**

I. Introduction

Handwriting recognition is a challenging task with the growing difficulty of the tackled problems. This field started with relatively simple systems able to recognize single characters with very small vocabularies. The recognizers were ameliorated to deal with continuous handwriting in order to recognize isolated words and whole lines extracted from handwritten pages.

A survey of large vocabulary off-line handwriting recognition is presented in [1]. This paper describes the successful approaches which solve the problems induced by the vocabulary size growth. Search space organization and lexicon reduction and pruning approaches are discussed. The paper stressed the importance of using Language Models (LMs) for large vocabulary systems. The LMs significantly improve the system performance. Open vocabulary was also addressed in this survey as one of the main futures direction in the field of handwriting recognition.

The Out Of Vocabulary (OOV) problem can be solved using different techniques. Character LMs are one of the used approaches to overcome this problem. In [2], a lexicon free system is compared with a word-based lexicon for printed Arabic and English text recognition. A combination between the two techniques is also presented. The character LM is constrained with a unigram LM. Competitive results have been achieved by the hybrid system if compared to the word-based system.

In [3], an open vocabulary handwritten address recognition based on character LMs is presented. A dictionary free approach is proposed with the use of high order n-grams. The best found configuration is using a 7-grams LM. Nevertheless, the result is still worse than the dictionary-based approach.

The advantage of the character-based LMs is that the system is able to recognize any sequence of characters. However, the problem is that the confusion is very high. This fact explains that the dictionary-based or the combined approaches are more successful.

Open vocabulary Korean word recognition is proposed in [4]. The lexicon is automatically selected using a dynamic Bayesian network language model. The lexicon is built by collecting variable length character sequences from the raw texts. This probabilistic framework is used for automatic acquisition of linguistic units focusing on building high-order language models for open-vocabulary domains. The use of such a model for lexicon selection is argued by the complex morphology of the processed language.

Linguistic based decomposition of the text can also be used for open vocabulary handwriting recognition. The authors of [5] presented the affixial approach for printed Arabic text recognition. The proposed approach is aiming at categorizing the word hypotheses into prefix, suffix and radical (root and infix). This can be guaranteed by the the decomposable structure of the vocabulary. The recognition process is based on the explicit segmentation of the word. First, the eventual suffix and prefix are recognized with a reduced vocabulary. After that, the root can be predicted by eliminating the roots that are not matching with the classified prefix and suffix. One of the drawbacks of this approach is the use of an explicit segmentation which is possible in printed text but very difficult for handwriting.

This work tackle the problem of open vocabulary handwriting recognition using Hidden Markov Models (HMMs) combined with an n-gram LM in the search step. The proposed sub-lexical approach is based on the use of an LM trained on a text corpus decomposed by morphological analysis. The selected vocabulary allows to have mixed elements (words and sub-words). The OOVs are recognized by combining the different vocabulary elements. This approach was successful for open vocabulary speech recognition [6].

This paper is organized as follows. Section II gives some specificities of the Arabic language. An overview of the system is presented in Section III. The morphological analysis for language model training is describe in Section IV. Finally the

1520-5363/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDAR.2013.63

results are presented in Section V followed by the conclusions and future work.

II. Arabic Language Specificities

Arabic handwriting is cursive and the context has an influence on the way of writing. The position of a character in the word as well as its context (previous and next characters) are important to define its shape. Arabic handwriting contains also diacritics and special ligatures as presented in Figure 1. This basic information concerning Arabic handwriting style has to be carefully handled to build an Arabic handwriting recognition system.

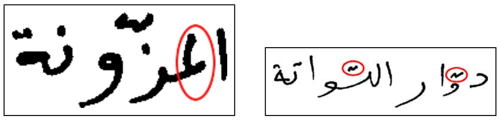

Fig. 1. Examples of Arabic Language Specificities (Left: Ligatures, Right: Diacritics)

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 character forms. We have to say that some of these characters can be rare if using limited training data.

III. System Description

The main goal of this paper is to analyze the improvement of system performance using the open vocabulary approach. The recognition system is not fully tuned with simple preprocessing and basic training of the classifier. We use a Hidden Markov Model (HMM) system based on the RWTH OCR [7]. Figure 2 presents the basic architecture of the recognition system.

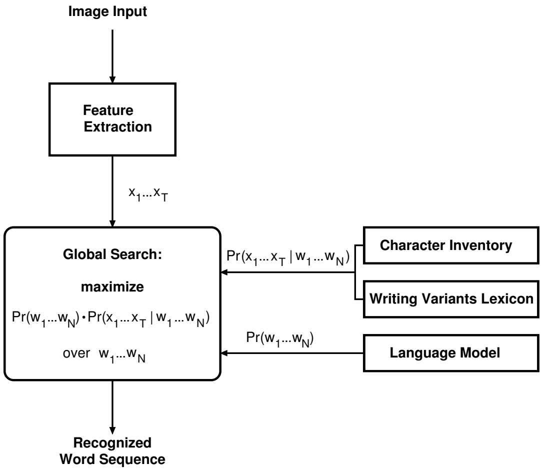

Fig. 2. RWTH OCR architecture

The first step in any pattern recognition system is the data preparation (or preprocessing) and the feature extraction. As we mentioned above the recognizer is based on HMMs (1-Dimensional), which have a limitation regarding image modeling (2-Dimensional). In fact, vertical image distortions have to be processed carefully. One way to deal with this problem is the vertical repositioning [8]. This is done by computing the center of gravity of a sliding window scanning the image from right to left (direction of writing). Afterwards the window is repositioned such that its center will be adjusted to the center of gravity.The features are pixel gray values extracted from a sliding window of size 9x30 with a maximum overlap (1 pixel shift). The 270 features are reduced to 35 using Principal Component Analysis (PCA). The used feature extraction technique is illustrated in Figure 3.

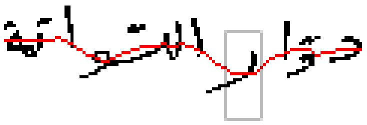

Fig. 3. Feature Extraction using repositioned sliding window

The task of handwriting recognition can be described as the classification of the word sequence $w_1^N = w_1, \ldots, w_N$ for which the sequence of features $x_1^T = x_1, \ldots, x_T$ is extracted. The posterior probability $p(w_1^N | x_1^T)$ is maximized over all possible word sequence w_1^N with N is the (unknown) number of words. Bayes decision rule is used to formulate the decision process as a mapping of the feature sequence x_1^T to the optimal word sequence via decision function $\hat{w}_1^N(x_1^T)$

$$x_1^T \rightarrow \hat{w}_1^N(x_1^T) = \arg\max_{w_1^N}\{p(w_1^N) \cdot p(x_1^T|w_1^N)\}$$

where $p(w_1^N)$ is an LM and $p(x_1^T|w_1^N)$ is the visual model. Here, the visual model is based on Hidden Markov Models in which the emission probabilities are modeled by Gaussian Mixture Models (GMM). The training procedure of the HMMs is based on the Expectation-Maximization (EM) and viterbi algorithms. The work on this paper presents results using the Maximum Likelihood (ML) criterion with globally pooled and diagonal covariance matrices. We opted for the use the analytical approach in which the HMMs are tri-characters ("triphones" in the speech recognition terminology). This approach is successful in the context of large as well as open vocabulary tasks.

The basic idea is to model characters within their context which is now a standard approach in speech recognition systems. The "triphones" are widely used in speech recognition technology. This technique is not yet used in all systems in Arabic handwriting recognition because of the lack of sufficiently big open access databases in this field. One of the problems in the context dependent modeling is that the number of possible triphones is huge, some triphones are not seen in the training. The number of parameters to estimate is very high wich can be overcome by state tying [9].

There are multiple proposed methods for state tying in the literature, the most successful is to use cart trees. A cart tree is a binary tree, the nodes are tagged with questions and the leaves are tagged with class labels. In our implementation questions are related to the shape of the character. The mixture label at the leaf identifies the mixture model for the triphone state.

The LM is a standard n-gram trained using the SRILM toolkit [10] with interpolated Kneser-Ney smoothing. The LM training text is first of all normalized using the following preprocessing steps. 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 steps allow to reduce noisy text and to have a better distribution of the probabilities.

IV. Morphological Analysis for Word Decomposition

A. MADA+TOKAN Tool

MADA+TOKAN 3.2 is a free toolkit which provides an extensive morphological and contextual analysis of raw Arabic text [11].

MADA (Morphological Analysis and Disambiguation for Arabic) is used for the derivation of all possible linguistic information of the words. This will eliminate any ambiguity surrounding the words using multiple features. MADA examines all possible analyses for each word and selects the analysis that matches the current context best. Support vector machine models are used to classify 19 distinct, weighted morphological features. The selected analyses contains complete information about diacritics, lexemes, glossary and morphology.

TOKAN, a general tokenizer for Arabic, is used to tokenize the disambiguated text generated by MADA. This is done with the help of a scheme provided to specify how the tokenization is done. TOKAN uses the morphological generation to recreate the word once different clitics are split off. This is done to guarantee the normalization of the word form and its consistency with other occurrences of that word.

B. Corpus Preparation and Vocabulary Selection

In this work, we used a tokenization scheme that allows to split the word into prefix, root and suffix. This splitting is taking into account all possible prefixes and suffixes in Arabic language which are supported by TOKAN. Table I provides the description of the used splitting scheme.

TABLE I. Tokenization scheme variables for the analysis of Arabic text

Type	Variable	Description
Prefix	QUES	The “question“ proclitic (e.g. أَ)
	CONJ	The ”conjunction” proclitic (و and فَ)
	PART	The “article“ proclitic
	FUT	The future marker clitic only (سـ)
	DART	The denite article only (أَل)
	NART	The negative articles only (لَا and مَا)
Radical	REST	The remainder of the word after the specied clitics have been separated
Suffix	PRON	Enclitics

For example, the word وسيكَاتبهَا will be decomposed to و (conjunction) + سـ (future marker clitic) + يكَاتب (rest) + هَا (suffix).

First of all, the words of the text corpus are decomposed using the MADA+TOKAN toolkit. The M most frequent full-words are left and the decomposed form of the remaining text is used. The final vocabulary is defined by selecting again the N most frequent words. The selected vocabulary contains new elements which are prefixes, suffixes and also new words. The prefixes and suffixes are tagged with a special marker ("+“). The recognition of unknown words is possible by the combination of the different vocabulary elements. In case of a prefix/suffix recognition, the marker is removed and the successive sub-words are combined.

V. Tests and Results

A. Data description

The presented approach is evaluated on two Arabic handwriting recognition tasks. The first dataset is provided by the MADCAT[1] (Multilingual Automatic Document Classification Analysis and Translation) program within the context of the OpenHART[2] evaluation. The data consists of more than $40k$ handwritten pages with text chosen from web forums and newspapers. Table II gives statistics detailing the used data.

TABLE II. OpenHART data statistics

	Train set	Dev set
# of pages	42,148	470
# of paragraphs	182,879	1,832
# of words	4,361,056	48,832
# of characters	23,324,011	266,121

Only a developement set is available for the OpenHART database. The evaluation of the system on the test set will be published in the OpenHART evaluation workshop.

KHATT is a freely available Arabic handwriting database [12]. The dataset consists of scanned handwritten pages with different writers, text and resolutions. Pages segmentation into lines is also provided to allow the direct evaluation of recognition systems without layout analysis. Table III presents some statistics of the used data from the KHATT database.

TABLE III. KHATT data statistics

	Train set	Dev set	Test set
# of pages	690	148	141
# of lines	9,475	1,902	1,997
# of words	129,826	26,142	26,449
# of characters	605,537	121,433	122757

B. Results on constrained task

The used data for the LM training is restricted to the officially available training text for the visual model. This task is intended to analyze the efficiency of the proposed approach in the case of lack of LM training data. The number of running words is 4 million words for the OpenHART task and $130k$

[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

for the KHATT task. The vocabulary in this task is simply all the words that appear in the training corpus.

Table IV presents the results on the OpenHART constrained task. The baseline full words system is compared with different configurations of the sub-lexical system. The number of full words kept in the training corpus is tuned. The best configuration on the OpenHART task is to keep $90k$ full words and decompose the others.

TABLE IV. RESULTS ON OPENHART DATA FOR THE CONSTRAINED TASK (WER IN %)

# of full words	Vocabulary size	Dev set
Baseline	100k	34.6
10k fw	42k	37.2
30k fw	60k	34.9
50k fw	72k	35.1
70k fw	83k	34.3
90k fw	94k	**34.1**
95k fw	97k	34.2

Table V gives a comparison of the results on the KHATT constrained task. The baseline full words system is compared with the sub-lexical approach. The results show that the proposed approach gives better results than the baseline system. An absolute improvement of 0.7% on the validation set and 0.9% on the test set are achieved.

TABLE V. RESULTS ON KHATT DATA FOR THE CONSTRAINED TASK (WER IN %)

	Vocabulary size	Dev set	Test set
Baseline	18k	34.1	33.6
5k fw	14k	33.5	32.7
10k fw	15k	**33.0**	**32.5**
15k fw	17k	33.1	32.5
17k fw	18k	33.5	33.0

An analysis of the number of affixes compared to the number of the full words for the OpenHART and KHATT constrained tasks is presented in Figure 4. The best result for the OpenHART task is obtained by keeping $90k$ full words which corresponds to a vocabulary of $94k$ entries with 14 prefixes and 13 suffixes. Figure 4 illustrates also the balance of the number of prefixes/suffixes relatively to the number of full words for the constrained KHATT task. The best result is obtained by keeping $10k$ of full words with a vocabulary size of $15k$ producing 27 prefixes and 13 suffixes. It is important to notice that the $15k$ full-words system has a slightly worse performance (0.1% worse in the validation set). The number of prefixes/suffixes is respectively 11 and 12 which is near to the best configuration in the OpenHART constrained task.

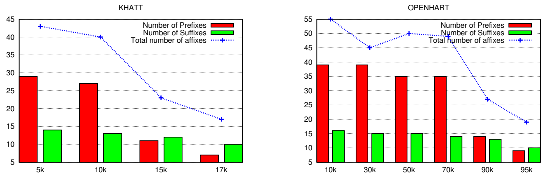

Fig. 4. Number of affixes for different configurations in the constrained task (right: OpenHART, left: KHATT)

C. Results on unconstrained task

The LM training data is not restricted to the training text of the used database. We selected data from in-domain text corpora collected from freely available newspapers and forums. The number of running words for LM training is around 1 billion words. The vocabulary of the unconstrained experiments is the $200k$ most frequent words in the training corpus. To avoid ambiguities, we have to notice that even if the number of full-words kept in the data is higher than the selected vocabulary size ($200k$), still some affixes are found in the final vocabulary due to their high frequency.

Table VI gives a comparison of the baseline system with different configuration of the open vocabulary system.

TABLE VI. RESULTS ON OPENHART DATA FOR THE UNCONSTRAINED TASK (WER IN %)

System	Dev set
Baseline	25.9
50k fw	27.5
100k fw	26.5
150k fw	26.0
200k fw	**25.9**
250k fw	26.0

The best configuration for the OpenHART unconstrained task is obtained by keeping $200k$ full words. The best system allows to have the same WER of the baseline system which is 25.9%. The results on the KHATT unconstrained task are presented in Table VII.

TABLE VII. RESULTS ON KHATT DATA FOR THE UNCONSTRAINED TASK (WER IN %)

System	Dev set	Test set
Baseline	27.8	26.8
50k fw	33.9	32.9
100k fw	28.7	27.6
150k fw	28.1	27.1
200k fw	**27.9**	**26.8**
250k fw	27.9	26.8

The best configuration for the sub-lexical approach is again by keeping $200k$ of full-words. The best system has the same performance of the baseline system for the test set but is 0.1% worse for the validation set. The $250k$ system has the same performance of the $200k$ system. A comparison between the number of affixes and the number of the kept full-words for the OpenHART and KHATT tasks is presented in Figure 5.

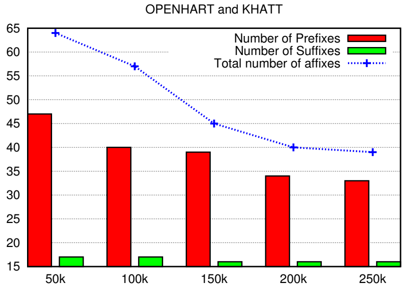

Fig. 5. Number of prefixes/suffixes compared with the number full words

The presented figure is valid for the two tasks (OpenHART and KHATT) since the corresponding selected vocabularies contain the same number of affixes with the same distribution. The best configuration in the two unconstrained tasks is using 34 prefixes and 16 suffixes.

The open vocabulary approach allows to have a significant improvement in the constrained task. The same performance is kept if comparing to the baseline system in the unconstrained task. This can be explained by the low OOV rate in the unconstrained task with a large vocabulary ($200k$). Table VIII gives an overview of the OOV rates for the different tasks comparing the baseline systems with the best configuration of the sub-words systems.

TABLE VIII. COMPARISON OF THE OOV RATE (%) FOR THE BASELINE AND THE SUB-WORDS (SW) BEST SYSTEMS

System		**OpenHART**	**KHATT**	
		Dev set	Dev set	Test set
Constrained	Baseline	8.29	11.41	11.40
	Best sw	5.7	10.41	10.58
Unconstrained	Baseline	3.50	4.18	3.51
	Best sw	2.08	2.28	1.86

There is an improvement in the OOV rate for all the experiments. The existence of an OOV rate which is greater than 0 is not contradictory with the open vocabulary concept. The sub-lexical approach can generate a high number of unknown words by combining the vocabulary elements.

The number of affixes in the unconstrained task is higher than in the constrained one. This can be explained by the decomposition of more data in the unconstrained task. This fact involves the repetition of more affixes which are selected in the final vocabulary.

VI. CONCLUSIONS AND FUTURE WORK

We presented in this paper an open vocabulary Arabic handwriting recognition system. The system is based on HMMs with n-gram LMs for the search step. Open vocabulary recognition is based on a sub-lexical approach. The text corpus for LM training is processed by a morphological analysis tool. The vocabulary is composed of words and sub-words after decomposition. Experiments are conducted using two Arabic handwriting datasets with two types of tasks. The LM training text for the constrained task is restricted to the official training data provided for the visual model training. Additional in-domain data is collected to train the LMs for the unconstrained task. An improvement of about 1% (WER) is achieved for the constrained task for the two datasets. The used approach allows to keep the same performance of the baseline system in the unconstrained task. The OOV rates are improved using the proposed approach.

The used recognition system is relatively simple using a Maximum Likelihood criterion (ML) for training the HMMs and a simple feature extraction method. The system performance can be improved using discriminative training with the Minimum Phone Error training criterion (MPE) for example. Neural Networks (NN) based feature extraction can be also integrated in the system. The improvement in the performance of the proposed approach can be more significant using a tuned recognizer. The combination of the proposed approach with a character based LM can be also investigated.

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. 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.

[2] I. Bazzi, R. Schwartz, and J. Makhoul, “An omnifont open-vocabulary ocr system for english and arabic,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 6, pp. 495–504, Jun. 1999. [Online]. Available: http://dx.doi.org/10.1109/34.771314

[3] A. Brakensiek, J. Rottland, and G. Rigoll, “Handwritten address recognition with open vocabulary using character n-grams,” in *Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*, ser. IWFHR '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 357–. [Online]. Available: http://dl.acm.org/citation.cfm?id=851040.856877

[4] S. Ryu and J. H. Kim, “Learning the lexicon from raw texts for open-vocabulary korean word recognition,” in *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, aug. 2003, pp. 202 – 206 vol.1.

[5] S. Kanoun, A. M. Alimi, and Y. Lecourtier, “Natural language morphology integration in off-line arabic optical text recognition,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 41, no. 2, pp. 579–590, 2011.

[6] A. El-Desoky Mousa, C. Gollan, D. Rybach, R. Schlüter, and H. Ney, “Investigating the use of morphological decomposition and diacritization for improving Arabic LVCSR,” in *Interspeech*, Brighton, UK, Sep. 2009, pp. 2679–2682.

[7] P. Dreuw, D. Rybach, G. Heigold, and H. Ney, *RWTH OCR: A Large Vocabulary Optical Character Recognition System for Arabic Scripts*. London, UK: Springer, Jul. 2012, ch. Part II: Recognition, pp. 215–254, iSBN 978-1-4471-4071-9. [Online]. Available: http://www-i6.informatik.rwth-aachen.de/rwth-ocr/

[8] P. Doetsch, M. Hamdani, A. Giménez, J. Andrés-Ferrer, A. Juan, and H. Ney, “Comparison of bernoulli and gaussian hmms using a vertical repositioning technique for off-line handwriting recognition,” Bari, Italy, Sep. 2012, pp. 3–7.

[9] 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.

[10] A. Stolcke, “SRILM - an extensible language modeling toolkit,” vol. 2, Denver, Colorado, USA, Sep. 2002, pp. 901 – 904.

[11] O. R. Nizar Habash and R. Roth, “Mada+tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization,” in *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, K. Choukri and B. Maegaard, Eds. Cairo, Egypt: The MEDAR Consortium, April 2009.

[12] S. A. Mahmoud, I. Ahmad, M. Alshayeb, W. G. Al-Khatib, M. T. Parvez, G. A. Fink, V. Märgner, and H. El Abed, “Khatt: Arabic offline handwritten text database,” in *Proc. International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Bari, Italy, Sep. 2012, pp. 447–452.