The UOB-Telecom ParisTech Arabic Handwriting Recognition and Translation Systems for the OpenHart 2013 Competition

Olivier Morillot[1], Cristina Oprean[1], Laurence Likforman-Sulem[1], Chafic Mokbel[2], Edgar Chammas[2], Emmanuèle Grosicki[3]

(1) Institut Mines-Telecom/Telecom ParisTech & CNRS LTCI, Paris, France
(2) University of Balamand, Balamand, El-Koura, Lebanon
(3) DGA Ingénierie des Projets, Bagneux, France

Abstract — **This article is a description of the two systems proposed for the recognition of Arabic handwritten text lines and for the automatic translation of text-line and sentence images into English text. The recognition systems are based on HMMs (Hidden Markov Models) and BLSTMs (bi-directional long short term memory) recurrent networks. Two SMT (Statistical Machine Translation) systems based on MOSES [1] were built for the evaluation system: one on text-line translation and one for sentence translation.**

Keywords—Arabic handwriting recognition; translation; text-line recognition; preprocessing; HMM; BLSTM;

I. Introduction

Handwriting recognition is a challenging task because of the inherent variability of character shapes. The most popular approaches for handwriting recognition are stochastic and neural networks, namely graphical models such as Hidden Markov Models (HMMs) and Recurrent Neural Networks (RNNs). HMMs can cope with non-linear distortions and offer a character-based representation of words: words models are obtained by the concatenation of compound character models which is convenient for enlarging the vocabulary. Neural Networks are improved by enriching them with recurrent connections and LSTM (Long Short Term Memory) blocks. This allows the recognition system to take into account the context of a word or a character within the text line. As a discriminative approach they are very efficient in terms of recognition accuracy.

A segmentation-free strategy for both approaches was chosen. This avoids the error-prone segmentation of text-lines into words, and words into characters. Thus, inputs of both recognizers are sequences of frames obtained by sliding a window along a text-line.

Systems were trained with a very restricted set of the training data: 16,000 text lines. Our best recognition system is the BLSTM with accuracy (*Accuracy=1-WER*) of 52 %.

We describe in the following all issues addressed during the OpenHaRT 2013 competition: data preprocessing, feature extraction, training with HMMs and BLSTMs, and description of dictionary and language model.

II. Preprocessing and Feature Extraction

The recognition performance depends on the quality of the images given as input to the system. Therefore it is necessary to reduce data variability. This variability concerns skew and slant angles, stroke width and height, background noise, etc. The preprocessing performed on an image facilitates the feature extraction step.

The database used for test and training is OpenHaRT 2013 [12], which contains handwritten Arabic documents. The documents are known to be difficult to recognize, as they come from different writers with different handwriting styles and various types of noise.

A. Preprocessing

The preprocessing methods are applied on a text-line image by respecting a defined order: building text-line images by using the coordinates of the word thumbnails, denoising, gray-level transform, deskew and deslant.

1) Building the text-line image:

As the tasks for the competition imply text-line recognition and translation, the first preprocessing step is text-line image construction. Given the coordinates for each word in the document, the text-line image is obtained by building a patchwork of word snippets. The advantage of this approach is the minimization of the background noise. An example of text-line images obtained by using text-line or word coordinates can be seen in fig. 1 (a) and fig. 1 (b).

2) Denoising and gray-level transform:

The text-line image obtained in the previous step is further processed to improve its quality. For removing the "salt and pepper" noise, a median filter is used. As the feature extraction module was designed for capturing information in gray-level images, the OpenHaRT database is transformed from binary to gray-level, as described in [2].

3) Deskew:

The recognition system is sensitive to the inclined writing. As we use horizontal sliding windows (see Section B), a slanted writing can induce a superposition of features belonging to different characters within the same vertical window. Therefore, a slant angle is globally determined in the text-line image, by maximizing a measure related to pixel densities in all the columns of the image as in [4]. This correction is then carried out by an affine transformation.

4) Deslant:

Slanted writing disrupts the baseline extraction from handwritten text images. As some of the features are based on pixel densities within the zones determined by the lower and upper baselines (see Section B), a slope correction should be applied. An original approach proposed by Morillot et al. in [3] was applied for baseline correction. It relies on a local estimation of the lower baseline, using a sliding window. The obtained line is further smoothed by using a Gaussian filter in order to remove discontinuities.

The image resulted after the preprocessing step is shown in fig.1 (c).

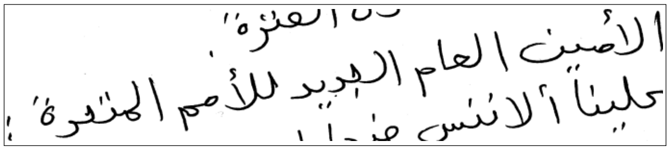

Fig. 1- (a) Text-line image extracted from a rectangle -box by using text-line coordinates

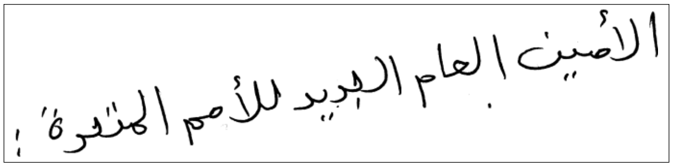

Fig. 1- (b) Text-line image extracted from a rectangle -box by using word coordinates

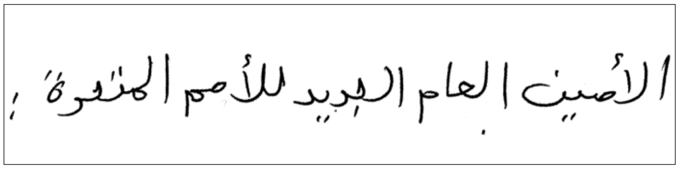

Fig. 1- (c) Preprocessed text-line image

B. *Feature extraction*

Feature extraction is carried out on preprocessed text-line images. The input of the system is a sequence of feature vectors extracted from a horizontal sliding window of fixed width. The size of the sliding windows is *w=9*. Each frame (sliding window) is divided into 20 cells. The height of the frame is the same as the height of the image provided at the input of the system. As the Arabic language is written from right to left, the shifting is performed in the same direction with *δ=3*. A set of 37 features (statistical, geometrical as described in [6] and directional features) is extracted for each frame, augmented with their first order derivative. The extracted features include:

- *2 features* representing background/foreground transitions;
- *12 features* for concavity configurations;
- *3 features* for the gravity center position – the first feature gives the position w.r.t. the baselines, the second one is the distance in number of pixels to the lower baseline, and the last one represents the difference between the gravity centers of two neighboring windows;
- *w=9 features* corresponding to the density of pixels in each column;
- *3 features* for the density of pixels within the window , above and below the baselines;
- and *8 directional features* corresponding to the histogram of gradients for 8 orientations from 0 to 7π/4, with a π/4 step.

In our python implementation, feature extraction takes approximately 3 minutes for each text-line image. Therefore, this step has been parallelized.

The sequence of vectors is modeled either by HMMs or by RNNs (recurrent neural networks). Their description is given in sections IV and V.

III. Dictionary and Language Model

Dictionary and language model (LM) were both built on the provided train database set (9693 documents). Only the 22000 most frequent words were used by our BLSTM [7] and HMM recognizers. This vocabulary restriction is undertaken in order to speed up decoding phases. A bigram language model is built from text-line train transcriptions with the defined vocabulary.

IV. BLSTM and HMM Recognizers

Both recognizers, BLSTMs and HMMs, work at the text-line level. The approach is segmentation-free since no explicit segmentation of the text-line into words, or words into characters, is performed.

A. *BLSTM recognizer*

Our principal recognition system is a BLSTM (Bidirectional Long Short-Term Memory) recurrent neural network as described in [5] and [7]. The BLSTM recognizer consists of the coupling of 2 recurrent neural networks. The value of an output unit at time step *t* is the linear combination of the outputs of the forward and backward hidden layers at this time step *t*. Hidden neural units are memory blocks called long short-term memory (LSTM). These units which include memory cells keep information through long time intervals and can be reset in an instant.

Forward and backward hidden layers are both made of 100 memory blocks. Output layer is made of 160 neurons, corresponding to the different characters (case sensitive), Arabic and Latin, numbers and punctuation marks. The BLSTM recognizer is trained with a gradient-based method. After each training epoch, the recognition error rate is evaluated on a validation set. If error rates do not improve for twenty epochs, network training is stopped. This strategy avoids data overfitting.

The BLSTM computes for each frame its corresponding outputs, each of them being associated to a character class. These outputs are normalized, providing for each character class, the posterior probability. Then a backward-forward token passing algorithm, referred to as CTC (Connectionist Temporal Classification) takes the posteriors as input and provides a sequence of words given the dictionary and the language model. We use the CTC implementation introduced in the works of Graves et al [5].

B. HMM recognizer

Hidden Markov Models (HMMs) have been widely used in handwritten recognition [8], [6]. HMMs are probabilistic automaton, the Markov Model, whose states are hidden and only observable by emitting an observation vector following probability distribution functions associated to the states. HMMs are often used in modeling the handwritten scripts by associating one HMM to each letter script or to a letter script in a specific context. In our system, the HMM has a Bakis topology: there is no transition from a state to a previous state while going from right to left, and one skip between states is allowed. This corresponds to the direction of writing in Arabic. It is worth noting that two different models have been used depending on the letter size. 5 states' models have been used for small size letters and punctuations while 8 states models were used for larger letters. The output probability distributions are Gaussians mixtures. Each state has an associated mixture with Gaussian probability distribution functions components. Mixtures with 32 Gaussian probability density functions were used in the experiments.

Context-independent and context-dependent letter models have been built [9]. For context-dependent models the left and right contexts of the letter are being considered. This modeling choice enables the processing of variable letter shapes in different contexts. However, contextual modeling has some limitations. The number of models becomes huge when context is considered. This has a direct impact on the resources needed during training and recognition. This includes the available training data to estimate the parameters of the different models. Several solutions exist to tackle this problem. In this work, the number of HMM parameters is reduced by using state-tying. A number of 181 context independent letter models have been used. This includes all variations of the Arabic letters but also some punctuations and Latin characters. All the derived context dependent models were estimated using state-tying and decision tree clustering.

The HTK toolkit has been used for both training and decoding [10]. Training has been performed in two stages. First only the words of the training data have been used as training set. Second, a small subset of the training lines has been used to reestimate the models parameters. For decoding, the estimated models have been used together with a bigram language model also estimated on the training data. The SRILM toolkit has been used in this estimation [11].

IV. MACHINE TRANSLATION

Statistical machine translation systems have shown great advances in the past years. The SMT system used in this evaluation is based on MOSES [1]. In order to translate a sequence of words, the conditional probability *Pr(e|f)* is maximized where *e* is a string in the native language (here English) that defines a translation of a string *f* in the foreign language (here Arabic). In other words, we look for the most probable native language sentence *e* provided the observed foreign sentence *f*. A probabilistic model is defined and its parameters are estimated and then used to calculate these probabilities. In the statistical decision framework, the best translation is identified as the most probable one, i.e.:

$$\tilde{e} = \arg\max_{e \in e^*} \Pr(e / f) = \arg\max_{e \in e^*} \Pr(f / e) \Pr(e)$$

As shown in this equation, maximizing the probability of the native language sequence of words given the sequence of words in the foreign language is equivalent to maximizing the product of the likelihood of the foreign language sentence given the native language sentence by the a priori probability of the native language sentence (Bayes rule). While the two forms are equal, the later one is more appropriate to the translation problem. This is mainly due to the a priori probability *Pr(e)* that can be used to constrain the grammatical/syntactical form in the native language while the grammatical/syntactical form of the foreign language is supposed to be correct. This a priori probability *Pr(e)* is usually known as the statistical language model in the native language.

The previous equation defines a production model for the foreign language string *f*. It is supposed that producing *f* is as a sequence of producing the correct native language sentence *e* following the language model distribution *Pr(e)* and then translating it to the foreign language sentence *f* through the forward channel described by the conditional distribution *Pr(f/e)*. This conditional distribution represents the conditional probability of a foreign language word given a native language word. The model to be used in order to define this distribution is not straightforward. First, one should decide if the model is restricted to words or to group of words. Second, it is important to decide how to get the pair of words or group of words. A parallel corpus is used for this purpose. In this case the machine translation will define the memory of the corpus.

Two SMT systems have been trained for the evaluation system. Both systems have been trained using the constrained training data provided by NIST. The first system used for the translation of the handwritten recognized text-lines has been trained and used on text-lines translation. The system was trained on clean data, i.e. the text-lines used for training are the transcribed text-lines and not the recognized ones. The second system was trained on sentences. This second system has been used for the translation of clean sentences provided in the third evaluation component (DTT) of the OpenHaRT database.

V. EXPERIMENTAL RESULTS

The BLSTM and HMM systems were trained on 16,000 text-line images (~11%) randomly chosen from the NIST database. The evaluation was done on 633 handwritten documents that contained 12,644 text-lines.

A. Text-line Recognition

The experimental conditions are provided in table I. Results provided by NIST for the text-line recognition task are shown in fig. 2 which summarizes the results obtained for all our systems. Our primary system BLSTM obtains the best recognition rate of 52%, compared to the HMM systems. This shows the interest of BLSTMs over HMMs in terms of recognition accuracy. Our HMM systems are either context-independent (CI c-hmm-1) or context-dependent (CD c-baseline-1, c-contextualhmm-1). For the best context-dependent system (c-baseline-1), character models were initialized on a word-based recognition system and trained on text-lines.

Our average performance is due to the fact that we trained our systems on only 11% of the given data (text-lines) for the training phase. Once the system was calibrated, we had only two weeks left to pre-process text-line images, extract features, train the system and decode the test data. For each of the systems a different sample of 11% of the training data was considered. The vocabulary was also limited to 22,000 or 30,000 words for the decoding phase. We believe that the system performance can be easily improved by considering the entire training database and combining classifiers.

TABLE I. DIR EXPERIMENTAL CONDITIONS

System ID	Method	Dict./LM size	Accuracy (1-WER)
p-blstm-1	BLSTM	22k words	0.5207
c-hmm-1	CI HMM	30k words	0.2140
c-contextualhmm-1	CD HMM	30k words	0.0651
c-baseline-1	CD HMM	22k words	0.2524

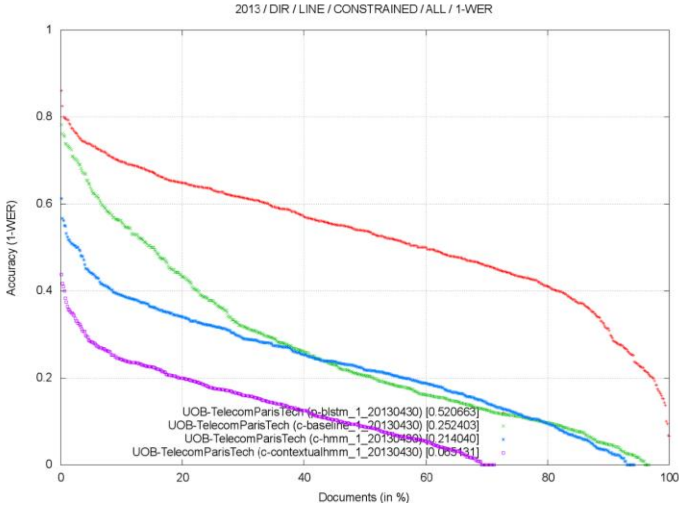

Fig. 2 - System performance on DIR (Document Image Recognition) task.

B. Translation

Translation results as computed by NIST are provided in fig. 3 and fig. 4 for the Document Image Translation (DIT) and the Document Text Translation (DTT) tasks respectively. 1-TER, BLEU and METEOR scores are used to present results. The results are shown for the constrained task, i.e. the whole system parameters are trained using the training set. The poor performance obtained in the DIT task compared to the DTT task are due to the poor recognition performance, especially that the DIT results correspond to the recognition by the HMM system. In addition, the performance in the DTT task also suffered from limiting the vocabulary to the 22,000 most frequent words. DIT and DIR experimental conditions are shown in table II.

TABLE II. DIT AND DTT EXPERIMENTAL CONDITIONS

System ID	Dict./LM size	Accuracy (1-TER)
DIT p-baseline-1	22k words	0.0815
DTT p-baseline-1	22k words	0.2131

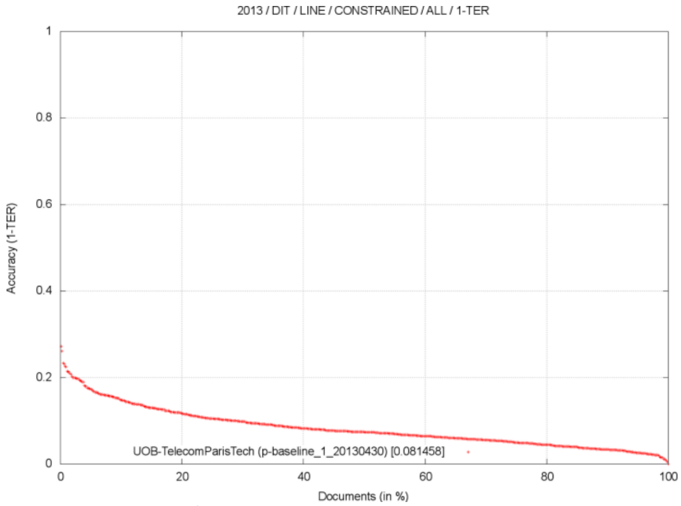

Fig. 3 - System performance on DIT (Document Image Translation) task.

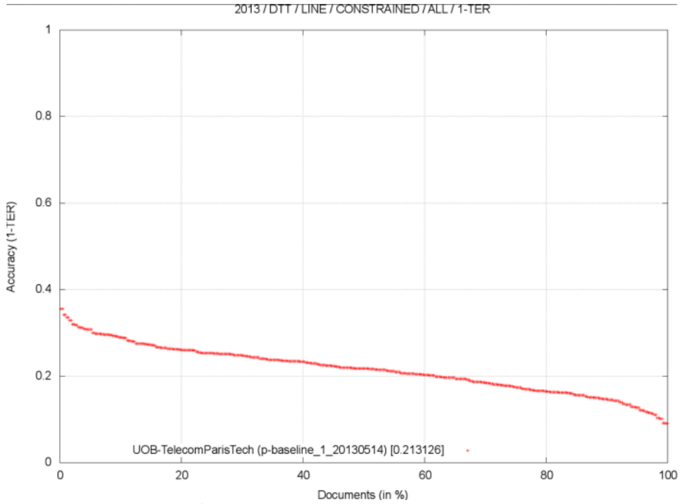

Fig. 4 - System performance on DTT (Document Text Translation) task.

VI. CONCLUSION

In this paper we have presented the recognition systems submitted by the University of Balamand (Lebanon) and Telecom ParisTech (France) for the OpenHaRT 2013 competition. We developed 2 recognition systems based on RNNs and HMMs, respectively, for the text-line recognition task and one HMM system for the translation task. Our best result with a recognition rate of 52% was obtained by using a single BLSTM recognizer trained on only 11% of the available data (145,000 text-lines). A first improvement of the system is to consider the entire database for training, instead of a small part. Combining the proposed recognizers will also improve performance. For the Document Image Translation task, we believe that training the translation system with the recognized text instead of the clean texts would also improve the performance. Using factored language models would also permit large improvement for the Document Text Translation.

REFERENCES

[1] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," ACL, Prague, Czech Republic, June 2007.

[2] C. Oprean, L. Likforman-Sulem, C. Mokbel, "Handwritten word preprocessing for database adaptation", in Document Recognition and Retrieval XX, San Francisco, 2013.

[3] O. Morillot, E. Grosicki, L. Likforman-Sulem. "Reconnaissance de courriers manuscrits par HMMs contextuels et modèle de langage", in CIFED, Bordeaux, 2012.

[4] A. Vinciarelli, J. Luettin: "A new normalization technique for cursive handwritten words", in Pattern Recognition Letters 22(9), 2001, pp. 1043-1050.

[5] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, A novel connectionist system for unconstrained handwriting recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence 31(5), 2009.

[6] R. El-Hajj, L. Likforman-Sulem, and C. Mokbel, "Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition", IEEE PAMI,Vol. 31, No 7, pp 1165-1177, 2009.

[7] O. Morillot, L. Likforman-Sulem, E. Grosicki,. "New baseline correction algorithm for text-line recognition with bidirectional recurrent neural networks", Journal of Electronic Imaging, 2013.

[8] R. Schwartz, C. LaPre, J. Makhoul, C. Raphael, Y. Zhao, "Language-Independent OCR Using a Continuous Speech Recognition System," ICPR, Vol. 3, p.99, 1996.

[9] A.-L. Bianne-Bernard, F. Menasri, R. El-Hajj, C. Mokbel, C. Kermorvant, and L. Likforman-Sulem, Dynamic and contextual information in HMM modeling for handwritten word recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No 10, pp. 2066-2080, 2011.

[10] S. Young, "The HTK Hidden Markov Model Toolkit: Design and philosophy," Department of Engineering, Cambridge University, UK, Tech. Rep. TR 153, 1993.

[11] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit", ICSLP, 2002.

[12] A. Tong, M. Przybocki, V. Maergner, and H. El Abed, "NIST 2013 Open Handwriting Recognition and Translation evaluation", Proceedings of the NIST 2013 Open Handwriting and Recognition Workshop, 2013, in press.