VOL. 13, NO. 22, NOVMBER 2018 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

A CORPUS FOR AMAZIGH TRANSCRIBED TO LATIN OCR SYSTEMS' EVALUATION

Khadija EL Gajoui[1], Fadoua Ataa Allah[2] and Mohammed Oumsis[3]
[1]Laboratory of Research in Informatics and Telecommunications, Faculty of Sciences, Rabat IT Center, Mohammed V University in Rabat, Morocco
[2]CEISIC, The Royal Institute of Amazigh Culture, Rabat, Morocco
[3]Department of Computer Science, School of Technology-Sale, Mohammed V University, Rabat, Morocco
E-Mail: khadija.gajoui@gmail.com

ABSTRACT

Corpora, initially created as resources for linguistic research, are attracting more and more the attention of machine learning researchers who are examining the potential of these corpora for training/ testing optical character recognition (OCR) systems. Following the last logic, this paper is concerned with research on OCR of printed historical and recent document written in Amazigh transcribed to Latin. It focuses, especially, on building a representative corpus dedicated to this language. In this paper, we describe the construction procedure of this corpus in tree levels, which are: line, word and character. Then we conduct a comparative evaluation of the corpus using an OCR system based on Long Short Term Memory approach. The comparison of the corpus is depending on the recognition rates and convergence in term of iteration number. Evaluation shows that the corpus level line gives the best result compared to the other levels with an error rate of 10.3%.

Keywords: optical character recognition, corpus construction, LSTM.

1. INTRODUCTION

Automatic transcription of printed documents into electronic format is an active field in research domains [1][2]. It largely depends on the success in character recognizing. In other hand, good recognition is based on a well done corpus. Standard corpora have become very important step in character recognition research. They are an essential requirement for the development and the evaluation of different character recognition methods. Actually, the conversion of the Amazigh cultural heritage embodied in its literary sources into an electronic version, and making its full texts available is atopic of great interest. Researches are produced aiming to overcome this problem [3]. Amazigh language can be represented in different forms [4]. One of these representations is the transcription of this language in Latin.

The main aim of this work is the construction of a corpus for Amazigh language transcribed in Latin. To the best of our knowledge, many corpora based Amazigh language have been developed [5][6], but none of them corresponding to this transcription.

The transcription of the Amazigh language in Latin has been used for decades [7]. Despite the variation of charsets used, this transcription appeared in historical books but also in recent books. Hence the interest of taking in consideration old and recent documents when creating the corpus dedicated for this language.

There are several difference points that can distinguish an historical book from another more recent. The effects of seniority on historical documents can be interpreted by: historical fonts including ligatures, characters displaced due to historical printing processes, character with fuzzy boundaries, paper quality, bleed through from the following page, blotches, etc. [8].

Another problem under study, in this work, is the level used in the corpus. Isolated character level, word level and line level are three different levels that can be present in a corpus, which the contextual knowledge differ from one level to another. This problem is discussed and treated in this paper.

To evaluate the constructed corpus, we used a new recognition algorithm based upon recurrent neural nets: Long Short Term Memory networks [9], usually just called "LSTMs". LSTM are a special kind of RNN, capable of learning order dependence in sequence prediction problems even in complex problem domains like machine translation, speech recognition, and more.

This work allows both to create a corpus for the Amazigh language transcribed in Latin as well as to compare the 3 corpus levels. The corpus built can be used in different works based on this language and the comparison can help in choosing corpus level in other OCR system.

In the remaining of the paper, we will describe our studied language, which is the Amazigh language transcribed in Latin, in section 2. In section 3, we will describe the procedure of the corpus construction. We will present, in section 4, the LSTM network, and, in section 5, the results of the evaluation. Then, we draw a conclusion and further works, in section 6.

2. AMAZIGH LANGUAGE

The Amazigh language, or Tamazight, is one of the oldest humankind languages. There are no official data on the number of Amazigh speakers, but the number of users is estimated to around thirty to forty million. The Amazigh language is present in a dozen of countries across the Maghreb-Sahel-Sahara. However, Morocco and Algeria are, by far, the two countries with the largest

Amazigh population. To transcribe Amazigh language, three writing systems are used [4]:

- Tifinagh, which is an authentic alphabet, attested in Libyan inscriptions since antiquity. Actually, it is the official script in Morocco since 2003.
- Arabic alphabet, used since the Arab arrival on the 6th century, especially in religion documents.
- The Latin alphabet, developed by colonial scholars at the end of the 19th century, and used later by national researchers.
- We are interested here in the transcription of the Amazigh language in Latin. An example of text written in Latin is shown in the following figure.

-« Imi ara d-iɛeddi weqḍar n taddert ṣbaḥ, dari-it-id, ad d-iɛeddi lɛinṣar, ɣur-k ak-id-walint tlawin! Melmi i tewḍeḍ s abrid ameqqran, aqḍar ad iṣub d akesar tama tazelmaṭ n webrid uzaɣar, ma d keč ad n-taliḍ d asawen tama tayeffust, ad yi-n-tafeḍ deg txerrubt izumal ik-ttrajuɣ. S yen ad nekcem ɣer tebḥirt-nneɣ, i wumi sawalen aɛerqub n Lqayed. Tthabin madden amḍiq-nni. Ula d abri-is ttkukrun a tawin, ma yella d baba, timeddiyin n wass kan id-yettɛedday, yessefqad ɣef tebḥirt-is. Kkes aɣbel i wul-ik ulac d acu ara tagadeḍ ».

Figure-1. Extract from 'Azal n tayri ' by AMARA.

No study has been done on this language except in our previous work [10][11]. It is a language based on Latin characters decorated with diacritics [11].

2.1 Charset

Several charsets version were used for the transcription of the Amazigh language in Latin over time. There are charsets witch are old and others newer.

For this study, we have collected a different charsets. Various type and age of books are covered.

The charsets that can be a generalization of all the charsets used for the transcription of the Amazigh language in Latin are listed in the following table:

Table-1. List of charsets used.

Book	Author	year
The Argan tree and its Tashelhiyt Berber lexicon[12]	Harry Stroomer	2008
Conte berber grivois du haut atlas[13]	Alphonse Leguil	2000
Choix de la version berbère du sud-ouest marocain [14]	Arsène Roux	1951
Manuel de Berbère Marocain (Dialecte Rifain) [15]	Léopold JUSTINARD	1926
Mots et choses berberes [16]	Emile Laoust	1920

2.2 Observations on this language

As we explained before, our studied language is based on Latin characters with diacritics. Diacritics are signs that come below, above, before or after the characters. In addition to these characters, the transcription also uses special characters [17].

In this study which revolves around this type of writing, we have used books from different categories and dates. Browsing these books allowed us to make some observations on this language. Among these observations, we can cite the frequent appearance of some character and words. The most frequent characters and sequence of characters are represented in the following table:

Table-2. An example of frequently used characters and words.

Frequent characters	Frequent words
Γ	Iγ
n	tt
i	is
w	ar
s	skarn
t	nns
a	id
	uk
	za
	ad

2.3 Punctuation & digits

The punctuation marks are present in the studied transcription as in all other languages. We can find the point, the virgule, the question mark, etc. But the most frequent punctuation marks that can be detected in the middle of a word are the hyphen '-' and '˘' as in 'iffoġ-d' (goes out), 'm˘ddĕn' (people) and 'iz˘nza-t' (sale it).

The digits used in this transcription are Arabic numerals in the Western system.

2.4 Unicode

Characters with diacritics are considered as special characters. Since many text editors can not read this type of character, we need to specify their Unicode. Unicode is another natural way for storing data. So, it can be processed by different editors, and subsequently be readily available to other researchers.

Here after a table showing the Unicode of some special characters used in the transcription.

Table-3. Unicode of some special characters used in the transcription.

Character	Unicode	Character	Unicode
Ă	0103	ů	016F
ḅ	1E05	ŭ	016D
ḍ	1E0D	ṛ	1E5B
ĕ	011B	ḷ	1E37
ġ	0121	š	0161
ḡ	1E21	ṡ	1E61
ḥ	1E25	ṭ	1E6D
ḫ	1E2B	ẓ	1E93
ŏ	014F		

3. THE CORPUS SIZE AND STRUCTURE

The purpose of this work is to build a corpus for the Amazigh language transcribed in Latin. There are two types of corpora, a linguistic corpus and a recognition corpus. A linguistic corpus is generally used for a linguistic purpose and is characterized by its large size that can reach tens of millions in some cases. Whereas, the goal of a recognition corpus is to train a system that will subsequently recognize a text. This type of corpus is characterized by the variety of examples in terms of style of writing and fonts but also enough size for a good training. In our work, we are interested in the last type of corpus. A good corpus implies good learning and therefore good recognition.

As we explained previously, to the best of our knowledge, no corpus dedicated to the Amazigh language transcribed in Latin exists. In this context, we were based on old documents and other more recent to build a corpus that takes into consideration both types of documents.

Since the type of characters used in the transcript are not compatible with all fonts by default. We have taken into consideration this criterion during the corpus construction.

3.1 Composition

In our case, a corpus is composed of a set of image files containing a text written in Amazigh transcribed in Latin. Each image is associated with a text file corresponding to the textual transcription of the image file. The image and its corresponding text file must have the same name so that the system can identify them.

As we mentioned earlier, the goal of this corpus is the training of a system in order to recognize recent documents as well as old documents. For this purpose, we divided our corpus into 2 parts.

The creation of the first part of this corpus is based on a text written in Amazigh transcribed in Latin. This text respects a set of criteria for example:

- the existence of the different charsets found in the Amazigh transcription;
- the presence of the most frequently confronted words mentioned at the top;
- each character in the charsets is present at least 5 times.

We applied, to this text, different sizes and different fonts suitable for this type of character. As example of the used fonts, we can mention: Arial, Cambric, Charissil, Tahoma, Calibri, Doulos.
From the resulted texts, with different size and font, we will generate a set of text images. Each text image corresponds to a text file with the transcription of the text on an image.

The second part of the corpus is based on books that bring together what is old and recent. Old books are characterized by fonts that are rare or sometimes no longer used. Type of paper and the effect of the scanner on old documents can also be a particularity for old books [11].

The images we will use for the construction of the 2nd part of the corpus are scanned images of printed books.

An example of book list used and the distribution over the different categories can be seen in the table bellow.

Table-4. Different book categories.

Category	Book
Linguistic	**Tirra - Aux origines de l'écriture au Maroc [7]**
Romance	Ijawwan n tayri [18]
Literature	Conte berber grivois du haut atlas [13]
Anthology	An anthology of Tashelhiyt Berber folktales [19]

The first step towards this part construction is images segmentation. A horizontal and vertical histogram is used to perform this segmentation. The pseudo-code of image text segmentation is presented as follows:

Procedure : Image text segmentation
Input : Image text **M**
Output : Images containing word or character **m**

1) Preprocessing of the image **M**
2) Apply vertical histogram **H**
3) Segmenting the Image **M** to **n** line image (**H**=0 correspond to the blank between lines)
4) For i=1,...,**n** do
5) ¦ Apply an horizontal histogram **h** to the image **Mi**
6) ¦ Segmenting into image words/ characters (depending on the succession of blanks where **h**=0) **mi**
7) Collect **mi** in **m**

Figure-2. The pseudo-code of image text segmentation.

After obtaining segmented images, we proceed to the conversion of these images into text in order to create the text files corresponding to each image.
To make this conversion, we can pass through one of these two methods:

- The first method is the easiest and obvious method. It consists of calling upon an optical character recognition system. In this context, we use the OCR system we developed previously, described in this article [11]. Since the system does not reach a recognition rate of 100%, and although this rate is quite high, it is imperative to conduct results verification after recognition as a manual post processing phase.
- The 2nd method is the manual method, where we have to browse the text images one by one, create the text file with the same name of the text image and place the corresponding text. This process is quite

difficult and requires time given the large number of images that exists in a corpus. However, this method is effective.

To build our corpus, we have mixed the two previously built parts. The size and partitioning of the corpus are detailed in the following paragraph.

3.2 Corpora sizes

Choosing the right corpus is always an important aspect of training and testing an OCR system.

Size and type are particular criteria for a corpus. They play an important role in the performance operation of the corpus. A corpus can be presented in one of three possible form types: line, word or character. The images composing the corpus can contain a line, a word or a character according to the type of corpus. The difference between the 3 types lies in the amount of information in a single image. An image with a line or word represents a succession of characters that can be frequently found. The character-level image does not contain information about the frequent succession of characters, but it does symbolize the significant basic unit of writing.

In this work, we will try to build 3 corpora for the Amazigh language transcribed in Latin. Each corpus corresponds to a specific type.

Line Corpus: This corpus is composed of images containing text lines. It is a mixture of the two parts mentioned above. The first part corresponding to the text images generated from the text created. For this part, we designed 2000 images of text lines with a text file transcription for each image.

The second part is based on the extraction of lines from books previously cited. For this purpose, we used different page samples from each book. We have extracted 5 to 10 pages per book. The page samples are varied: cover pages, summaries and different types of paragraphs. In this part, we used the horizontal histogram to extract the lines. 672 text images were obtained after the segmentation of the pages of different books and we created their transcription as a text file.

At the moment, this type of corpus consists of a total of 2672 image text. As we used document from different pages and police size, there are between 1 and 21 words per text line of on the average.

Word Corpus: In this corpus, we find images that contain one word each. As in the corpus line, we have also mixed the two parts explained before. We generated 4500 word images with their text files transcription. For the 2nd part, we used the horizontal histogram to extract the lines. Then, we used the vertical histogram to get words images. We have collected 1100 word images of which we have created the corresponding text file transcription.

In total, this corpus is composed of 5600 images of words. Each image is associated with a text file transcription.

We note that this type of corpus can be used as input for natural language processing (NLP) tasks.

Character corpus: The images contained in this corpus are images of isolated character. For the reasons that have been mentioned previously, the two parts were brought together in this corpus as well. The result for the 1st part was a set of 9800 character images with their transcription. In the 2nd part, as in the word corpus, we used a horizontal and vertical histogram to extract the characters from the books pages. The result being 2500 images each associated with its transcription as a text file. We have a total of 12,300 character images and their transcription.

4. LONG SHORT-TERM MEMORY (LSTM)

Neural networks models [20] have become increasingly popular for the language modeling task. In particular, recurrent neural networks are able to learn and generate time sequences and can take into account all of the predecessor words. These networks are available in several variants; the two main ones are Vanilla and LSTM. The problem of the recurrent networks is that they are difficult to train and therefore are unlikely to show the full potential of recurrent models. These problems are addressed by the Long Short-Term Memory neural network architecture.

An LSTM is a special kind of RNN architecture, capable of learning long-term dependencies.

4.1 Definition

The Long Short-Term Memory (LSTM) is a specific recurrent neural network architecture that was originally proposed by Hochreiter and Schmid Huber [9]. Nowadays, LSTM is became a widely used architecture due to its superior performance in modeling both short and long term dependencies in data more accurately than conventional RNNs[21].

The vanishing gradient is a persistent problem in the RNNs. The gradient of the error function gets scaled by a certain factor, whenever the neural network is propagated back through a unit. The gradient blows up or decays exponentially over time, according to the factor value, which can be greater or smaller than one. As a result, the gradient can either dominates the next weight adaptation step or effectively gets lost. This problem is called vanishing gradient problem [22].

The vanishing gradient problem is solved by LSTM. LSTM tries not to impose any bias towards recent observations and keeps constant error flowing back through time. It follows essentially the principle of the RNN architecture, with the difference that it implements a more elaborated internal processing unit called cell.

The robustness and peculiarity of LSTM allowed him to solve several task with a high level of difficulty. Among this task we can cite: Recognition of the temporal order of widely separated events in noisy input streams; Robust storage of high-precision real numbers across extended time intervals; Arithmetic operations on continuous input streams; Extraction of information conveyed by the temporal distance between events; Recognition of temporally extended patterns in noisy input sequences; Stable generation of precisely timed rhythms,

as well as smooth and non-smooth periodic trajectories [23].

LSTM is then able to clearly surpass the RNNs on different tasks in terms of reliability and speed.

4.2 Architecture

The deep architecture of LSTM is widely used today in sequence learning, and has shown great experimental power. However, this architecture is subject to many changes [24]. Therefore, many different LSTM variants and topologies have been developed motivated by an analysis of error flow in existing RNNs [25].

The layer in LSTM architecture is composed of memory cells, which are a set of blocks connected recurrently. These blocks are comparable to the memory chips in a digital computer. Each cell in the LSTM architecture contains three multiplicative units—the input, output and forget gates—that provide continuous analogues of write, read and reset operations for the cell. The gates are either entirely open ('O') or entirely closed ('—').

An example of LSTM cell with the gating units is presented in figure bellow.

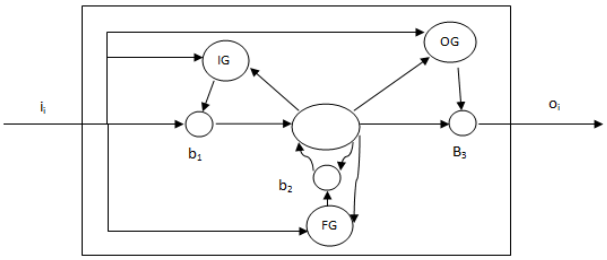

Figure-3. LSTM memory cell with gating units.

The LSTM unit adds several intermediate steps: First, we apply the activation function to i_i and we multiply the result by a factor b_1. Then, we multiply the inner activation value of the previous time step by the quantity b_2, and we add result of the multiplication due to the recurrent self-connection. Finally, we scale the result by b_3, and apply another activation function in order to find o_i. The small circles in the figure represent the factors b_1; b_2; $b_3 \in (0; 1)$. They are controlled by the circles IG, OG and FG corresponding to the input, output, and forget gate, respectively.

We can consider the traditional RNNs as a special case of LSTMs. If we set the input gate to all ones, the forget gate to all zeros (no memory), and the output gate to all ones.

The net can only interact with the cells via the gates.

The equations of the dynamics of this model are as follows:
Where:

$$i = \sigma(x_t U^i + s_{t-1} W^i)$$
$$f = \sigma(x_t U^f + s_{t-1} W^f)$$
$$o = \sigma(x_t U^o + s_{t-1} W^o)$$
$$g = tanh(x_t U^g + s_{t-1} W^g)$$
$$c_t = c_{t-1}{}^{o} f + g^{o} i$$
$$s_t = tanh(c_t)^{o} o$$
$$y = softmax(V s_t)$$

i: input gate indicates how much of the new information will be let through the memory cell.
f: forget gate is responsible for information. It should be thrown away from memory cell.
o: output gate indicates how much of the information will be passed to expose to the next time step.
g: self-recurrent, which is equal to standard RNN
c_t: internal memory of the memory cell
s_t: hidden state
y: final output

4.3 LSTM in OCR

As we explained earlier in this paper, LSTM has been used to overcome different problems. Among these problems we can find the optical characters recognition.

Research done on LSTM in the OCR field [26] has shown that this architecture has been able to overcome the problem of earlier neural networks to forget previously learned information, and has proven to be very successful in pattern recognition tasks such as handwriting recognition [27], even in the context of medieval manuscripts [28].

LSTM networks, has been employed to OCR modern as well historical documents. The excellent OCR results obtained prove that LSTM is also valid and powerful on this type of documents [28].

5. RESULTS AND DISCUSSIONS

After the creation of the three types of corpora corresponding to the Amazigh language transcribed in Latin, we will evaluate the corpora using LSTM approach. This later is known by its ability of recognition even in complex cases, as in our case where we process old documents. This evaluation allows testing the robustness and certainty of the constructed corpus. Another goal for this evaluation is the comparison between the three types of corpora. The comparison will be based on two criteria: the recognition rate and the convergence in terms of the iterations number.

5.1 Features

The phase that usually precedes the system train is the features extraction phase. In order to train the LSTM, we need to extract different features. The features we used are: gradients, singular points of the skeleton, the presence of holes, and unary-coded geometric information, such as location relative to the baseline and original aspect ratio and skew prior to skew correction.

5.2 Experimentations & results

We performed experiments on each corpus aside.

Corpus line: As explained in section 3, this corpus contains 2450 text images. To distinguish the training and test corpus, we took random samples in the

main corpus to form these two corpora. The training and test corpus consists respectively of 1960 and 490 text images.

In order to study the convergence in terms of iteration number, we configured the system to construct a model every 500 iterations. The error rates for each model related to this corpus are presented in the table bellow:

Table-5. Error rates for the line corpus.

Iteration	500	1000	2000	2500	3000	3500	4000	4500	5000	6000
%	65.8	28.1	17.7	15.3	13	12.1	11.2	10.8	10.5	10.3

Corpus word: This corpus contains 5600 images corresponding to images of words. We divided the corpus in training corpus and test one. The two corpora are composed respectively of 4480 and 1120 images.

After several experiments, we chose to configure the system to build a model every 1000 iterations. The error rates noted for every model are shown on the following table:

Table-6. Error rates for the word corpus.

Iteration	500	2000	4000	6000	8000	9000	10000	11000	11500	12000
%	100	100	90	76	59.6	52	45	41	39.4	34

Corpus character: this corpus is composed of images of isolated characters. It contains 12300 images. To form the corpus of training and test, we chose random examples of the main corpus in order to have 8000 images in the training corpus and 4300 images in the test corpus. The experiments performed for this corpus led us to choose a step of 10000 iterations before the construction of the model.

We note that for the case of the three types of corpus, the models are built on the basis of the previous models and not in an independent manner.

The results of the error rates for this corpus are shown in the following table:

Table-7. Error rates for the character corpus.

Iteration	**10000**	**20000**	**30000**	**40000**
%	100	97	96	86

5.3 Remarks & analysis

According to the experiments, the error rate of the corpus line decreases and arrives to a value of 10.3% after 6000 iterations and becomes stable. For the corpus level words the error rate follows a descending curve over time and notes a percentage of 34% from iteration 12000 where it stabilizes.

For the corpus character the error rate converges very slowly. After 40000 iterations it arrives at a very high error rate which is 86%. This error rate may be due to a problem of the corpus construction or an incompatibility with this type of system.We note that the time taken in each iteration delayed from one corpus to another. It is considerably longer in the case of the line corpus compared to the word corpus and the character corpus but this diversity can be neglected given the difference between the iteration numbers. It can be noticed that the line and word corpora give a good recognition rates with a difference in the convergence time where the convergence time of the line corpus is much faster than for the word corpus. However, the character corpus remains far from converging to a good error rate even after a large number of iterations.

The good results given by the line and words corpus can be a proof on the stability and the correctness of the constructed corpora.

From the results, we can easily see that the line corpus is the strongest compared to the other corpora. Therefore, line segmentation is the best. The advantage of the line corpus is due to the important amount of information in the line that has a succession of characters. Another strong point of this corpus in our case lies in the use of a system based on LSTM characterized by its strong memory.

Recognition rates noted for these experiments are significantly high, especially for the case of the corpus line. This result is proof of good construction and corpus performance. Since this corpus is the first of its kind for the Amazigh language transcribed in Latin, it can be a solid base on different works done on this language.

6. CONCLUSION AND FUTURE WORKS

In this paper, printed documents written in Amazigh transcribed in Latin are considered. This study aimed to develop a corpus dedicated to this language.

In the absence of a corpus that represents the studied language, the goal is to construct a corpus that will be a training base for an OCR system that will be able to transform this type of document into an electronic version with a considerable recognition rate.

First, we created 3 types of corpus. Each corpus corresponds to one of the tree levels, which are line, word and character. To evaluate these corpora, we used an LSTM-based OCR system known for its ability to learn in complicated conditions and also for its strong memory.

The results of this comparative evaluation show that the line level gives better results in a reduced time of convergence compared to the other levels.

Immediate future work includes plans to make this corpus available on the Internet. Before, we have to improve and feed the corpus by new examples especially for the part based on real documents Another perspective for this work is the improvement of the character corpus by searching for the failures and The correcting them. The corpus can serve as a basis for further research related to this language.

REFERENCES

[1] Eikvil L. 1993. OCR, Optical character recognition, norsk regnesentral.

[2] Vaidya M., Joshi Y. V. and Bhalerao M. 2017. Marathi numeral identification system in Devanagari script using discrete cosine transform, International Journal of Intelligent Engineering and Systems. 10(6): 78-86.

[3] El Ayachi *R*., Fakir M. and Bouikhalene B. 2011. Recognition of tifinaghe characters using dynamic programming & neural network, Recent advances in document recognition and understanding.

[4] Pouessel S. 2008. Écrire la langue berbère au royaume de Mohamed VI. Les enjeux politiques et identitaires du tifinagh au Maroc, Revue des mondes musulmans et de la méditerranée, (124), 219-239.

[5] Boulaknadel S. and Ataa Allah F. 2013. Building a standard amazigh corpus, Proc. of the Third International Conference on Intelligent Human Computer Interaction (IHCI 2011), Prague, Czech Republic, august, 2011. Springer Berlin Heidelberg. pp. 91-98.

[6] Bencharef O., Chihab Y. *et al*. 2015. Data set for tifinagh handwriting character recognition, Data in brief. 4: 11.

[7] Skounti A., Lemjidi A. and Nami M. 2003. Tirra aux origines de l'écriture au maroc, publications de l'institut royal de la culture amazigh, rabat.

[8] Springmann U., Najock D., Morgenroth H. *et al*. 2014. Ocr of historical printings of latin texts: problems, prospects, progress. Proc. of the First International Conference on Digital Access to Textual Cultural Heritage, ACM. pp. 71-75.

[9] Hochreiter S. and Schmidhuber J. 1997. Long short-term memory, Neural Computation, vol. 9, no 8, p. 1735-1780.

[10] El Gajoui K., Ataa Allah F. and Oumsis M. 2015. Training tesseract tool for Amazigh ocr, Proc. of the 15th International Conference on Applied Computer Science, P. 20-22.

[11] El Gajoui K., Ataa Allah F. and Oumsis M. 2015. Diacritical language ocr based on neural network: case of Amazigh language, Procedia computer science. 73: 298-305.

[12] Stroomer H. 2008. The argan tree and its tasheliyt berber lexicon, etudes et document berbères, université de leyde.

[13] Leguil A. 2000.Conte berber grivois du haut atlas, l'harmattan, paris.

[14] Roux A. 1951. Choix de version berbères parler du sud-ouest marocaine, France.

[15] Justinard L. V. 1926. Manuel de berbère marocain:(dialecte rifain), Geuthner.

[16] Laoust E. 1920. Mots et choses berbères, paris.

[17] El Gajoui K., Ataa Allah F. and Oumsis M. 2016. Recognition of amazigh language transcribed into latin based on polygonal approximation, International journal of circuits, systems and signal processing,vol 10, p. 297-305.

[18] Lasri A. 2008. Ijawwan n tayri, marrakech, imp ima.

[19] Stroomer H. 2001. An anthology of tashelhiyt berber folktales: (south Morocco), Köppe.

[20] Reddy G. T. and Khare N. 2017. Hybrid Firefly-Bat Optimized Fuzzy Artificial Neural Network Based Classifier for Diabetes Diagnosis, International Journal of Intelligent Engineering and Systems. 10(4): 18-27.

[21] Sak H., Senior A., and Beaufays F. 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling, proc. of Fifteenth Annual Conference of The International Speech Communication Association.

[22] Sundermeyer M., Schlüter R. and Ney H. 2012. Lstm neural networks for language modeling, proc. of

Thirteenth Annual Conference of the International Speech Communication Association.

[23] Gers F. A., Schmidhuber J. and Cummins F. 2000. Learning to forget: continual prediction with lstm, Neural computation. 12(10): 2451–2471.

[24] Schmidhuber J. 2015. Deep learning in neural networks: an overview, Neural networks, vol. 61, p. 85-117.

[25] Graves A. and Schmidhuber J. 2005.Framewise phoneme classification with bidirectional lstm and other neural network architectures, Neural networks. 18(5-6): 602-610.

[26] Springmann U. and Lüdeling A. 2016. Ocr of historical printings with an application to building diachronic corpora: a case study using the ridges herbal corpus, Arxiv preprint arxiv:1608.02153.

[27] Graves A., liwicki M., fernández S. *et al.*, 2009. a novel connectionist system for unconstrained handwriting recognition, IEEE transactions on pattern analysis and machine intelligence, Vol. 31 IEEE: 855-6.

[28] Fischer A., Wüthrich M., Liwicki M. *et al.* 2009. automatic transcription of handwritten medieval documents, Proc. of 15th International Conference on Virtual Systems and Multimedia, (VSMM'09), IEEE137- 42.