

The Bidirectional Long-Short-Term Memory Neural Network based Word Retrieval for Arabic Documents

Dr. Yusuf Perwej

*Department of Computer Science & Engg., Al Baha University
Al Baha, Kingdom of Saudi Arabia (KSA)
yusufperwej@gmail.com*

ABSTRACT

The reflow from Arabic document image collections is a challenging task. This is partly due to the insolubility of the Arabic script. Because of the peculiarity of the whole body of the Arabic words, namely connectivity between the characters, thereby the segmentation of An Arabic word is very arduous and also the variability of the handwritten styles and shapes as well as deceleration in the print therefore Arabic document repositories are not liable to indexation and reflow. In this paper, we are proposing an idea for reflow coherent Arabic document in response to an appropriate Arabic query word. We present a novel approach at reflow on an Arabic document image, using a Bidirectional Long-Short-Term Memory Neural Network. The designed to take relating to information into account, these networks can maintain Arabic word images that cannot be durable segmented into individual Arabic characters. The partitioning Arabic word, we easier the problem and receive elevated reflow rates. The proposed capable reflow scheme avoids unambiguous recognition of Arabic characters. An experimental evaluation on a dataset of Arabic word images conjunct from handwritten notebook show good precision even in the impendence of printing transformation and deceleration. The reflow Arabic word performance is comparison with baseline methods. These results encourage the development of real world systems for word reflow for Arabic documents.

Keywords: Arabic Script, Dynamic Time Warping (DTW), Word Spotting, Segmentation, Bidirectional Long-Short-Term Memory Neural Network (BLSTMNN), Euclidean Distance.

1 Introduction

Today scenarios there are two famous viewpoints to building search systems over document image collections firstly the optical character recognition (OCR) and secondly the word spotting. The optical character recognition has been applied to real life problems in many industries for a number of decades. But optical character recognition typically work well for distinctly documents and face severe arduous in character segmentation and recognition [1] in the presence of digression. Further, the robust optical character recognition is not yet available for many Arabic languages [2], which have an extended character set with a complex script layout. The word spotting approaches [3] overcome some of the drawbacks of optical character recognition. In word spotting segmentation at the Arabic word level is much more accurate than character/component segmentation since degradations typically do not impress the inter-word spaces [4]. The queries as well as the database images are transformed into a

sequence of feature vectors and compare is oftentimes carried out with the help of Dynamic Time Warping (DTW). We use Bidirectional Long-Short-Term Memory Neural Network for word spotting in this paper [5]. We extend these methods for word spotting in handwritten Arabic documents. Bidirectional Long-Short-Term Memory neural networks are recurrent networks with hidden layers consisting of long short-term memory [6] blocks. Bidirectional Long-Short-Term Memory neural network transform a sequence of feature vectors into a sequence of character class likelihood [7], and thereby building an mesne representation which is used for word spotting. In this paper, we demonstrate the well-turned use of a Bidirectional Long-Short-Term Memory neural network for printed Arabic documents. The main focus is on restitution from a collection of documents written in Arabic, a script for which robust and credible optical character recognition system do not exist. Bidirectional Long-Short-Term Memory neural networks are competent of, and matching of feature a vector which does not scale to huge number of data sets. We show our method on a couple of handwritten books in Arabic.

2 The Arabic Script

The Arabic script evolved from the Nabataean Aramaic script. It has been used since the 4th century AD, but the earliest document, an inscription in Arabic, Syriac and Greek dates from 512 AD [8]. The Arabic one of the six official languages of the United Nations, Arabic is a Semitic language spoken by between 300 and 400 million native speakers, and a further 250 million non-native speakers, in nearly twenty countries in the Middle East and North Africa. Arabic is quite a challenging language to learn for a number of reasons: it uses a number of sounds pronounced way back in the throat that can be tricky for speakers of English and other languages [9]. It is written with a cursive alphabet running from right to left in which the letters change shape depending on their position in a word [10]. The words consist of connected components or sub-words, and these are often called fragment of Arabic words in the literature. In Arabic script, there is no difference in the within word space and the between words space in figure 1.

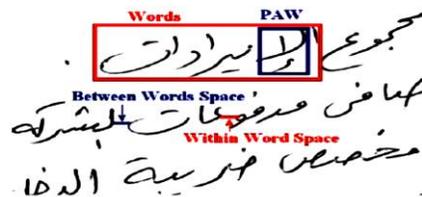


Figure 1: Arabic Script between Words Space

This lack of apparent boundaries between words, together with the fact that Arabic writing is naturally cursive and more voluntary than in other languages, make word spotting [11] in the Arabic language a challenging task in need of further research. Arabic word spotting approaches tend to segment documents into fragment of Arabic words rather than words, and then find ways to rebuild the words from the fragment of Arabic words. In this paper there may not be a sufficient number of examples of each class present in the training set. To avert incorrect restitution we are asiphon appropriate strategy. The Arabic words are mark out by a connecting top line, one of the famous initial steps in recognizing Arabic is to erase this line. This separates the vowel modifiers which span the top-zone from the middle zone. This is explained in figure 2.

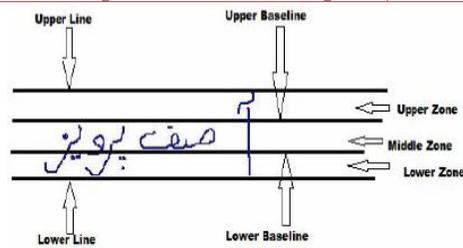


Figure 2: The Arabic word Script and its upper, middle, and lower zone

Similarly the lower zone is also erased. This mitigates the number of character classes to be recognized. We are interested in spotting words, or retrieving occasional Arabic documents analogous to a query. We take an intermediate route between exhaustive recognition and feature based Arabic word spotting in this work.

3 The Word Spotting

The word spotting is viewpoint that traditionally viewed as an image matching task between one or multiple query word-images and a set of candidate word-images in a database [12]. The goal is to greatly detract the amount of annotation work that has to be propitiate, by grouping all words into clusters. The each cluster contains words with the same remark. Once such a clustering of the data set exists, the number of words vested in a cluster can be used as an indication for determining the weightage of the word as a query term. The various systems have been compared in and showed that the best precision was obtained with the system based on Dynamic Time Warping an algorithm for measuring equality between two sequences which may vary in time or speed [13]. The word spotting can be classified in two type firstly the segmentation-based, and secondly the segmentation-free approaches. In segmentation-based approaches require each document image to be segmented at word level, taking advantage of the knowing of the structure of a document. In Segmentation-free approaches the image is divided into patches and the query word image is classified about each patch. Since all the regions are compared, these systems are computationally costly in terms of time [14]. In this paper, we are proposing word spotting based on a lately evolved recurrent neural network, termed bidirectional long short-term memory (BLSTM) neural network. The word spotting in the Arabic script dates back several years [15] when we unfortunately offered little work and solutions for handwritten Arabic script.

4 Dynamic Time Warping

In the year 1983, Joseph Kruskal and Mark Liberman [16] introduced a new technique to compare two curves (calculate the distance between them). The technique, that they called time warping, made it possible to make a matching between two curves that are subject not only to alteration by the usual additive random error but also to variations in speed from one portion to another. A guileless approach to comparing two time-variant feature vectors is to align one vector to the other, column by column, and computing the distance between the two. This viewpoint would perhaps get confused by the compression and expansion effects that “time” has on the features. The dynamic time warping takes care of this problem by compute the optimal time-warped alignment of the two feature vectors and the compute their distance. In figure 3, each vertical line connects a point in one time series to its correspondingly similar point in the other time series.

The lines have identical y-axis values, but have been separated so the vertical lines between them can be viewed more readily. If the two time series in figure 3 were identical, all of the lines would be straight vertical lines because no warping would be necessary to 'line up' the two time series. The warp path distance is the sum of the distances between the corresponding points. Thus, two time series that are identical except for localized stretching of the time axis (as in figure 1) will have warp path distances of zero [17]. The Dynamic time warping is an illustration of a dynamic programming problem. To compute the dynamic time warp, the first feature vector $f^{(1)} \in R^{(M,1)}$ is lined up against the second feature vector $f^{(2)} \in R^{(1,N)}$ and the $M \times N$ distance matrix $D(., .)$ is computed between every element in $f^{(1)}$ and $f^{(2)}$. The each $D(i, j)$ element is computed according to the following formula where $cost(f_i^{(1)}, f_j^{(2)})$ is the distance between the $f^{(1)}(i)$ and $f^{(2)}(j)$.

$$D(i,j) = cost(f_i^{(1)}, f_j^{(2)}) + \min \begin{cases} D(i, j-1) \\ D(i-1, j) \\ D(i-1, j-1) \end{cases}$$

The final DTW-distance is the value at $D(M,N)$ and the feature element to element warping-correspondence is found by tracing back from $D(M,N)$ to $D(0, 0)$. It is important to note that the Dynamic Time Warping constructs a matching path in a more complex way.

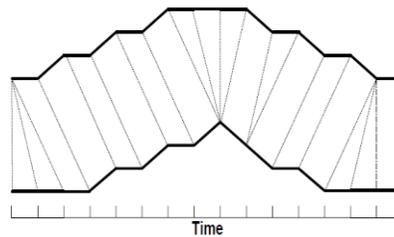


Figure 3: A Warping between Two Time Series

It is based on linear matching, but has four conditions or constraints (of which three are optional) that need to be satisfied. These conditions are called the continuity condition, the boundary condition, the monotonicity condition and the penup/pendown condition.

4.1 The Continuity Condition

The continuity condition decides how much the matching is allowed to differ from linear matching. This condition is the core of the Dynamic Time Warping and thus is not alternative. For some warping path $w = [w_1 w_2 \dots w_K]$ $w_k = (i, j)$ then $w_{k-1} = (i', j')$. Where $i - i' \leq 0$ and $j - j' \leq 0$. Thus, the allowable warping path is limited to adjacent cells. In figure 4 shows how this condition puts a constraint on the matching path. In this matching path, the matches corresponding to the crossed squares in the path are prohibited by this condition.

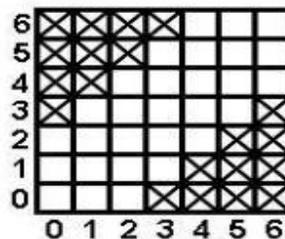


Figure 4: The Continuity Condition

4.2 The Boundary Condition

The boundary condition, if turned on, forces a match between the first points of the curves and a match between the last points of the curves. If this condition is turned on, the named points are matched, whether the continuity condition agrees or not and whether the points are nearest or not. The warping path must start D (1, 1) and end at D (M, N). In figure 5 shows how this condition puts a constraint on the matching path. The first points of the two curves and the last points of the two curves are forced to match.

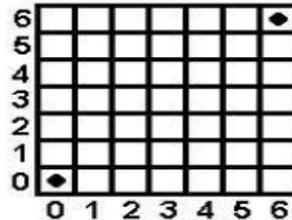


Figure 5: The Boundary Condition

4.3 The Monotonicity Condition

The monotonicity condition is a condition that inhibits the matching from “going back in time”. If at some point in the matching process it is decided that the i^{th} point of the first curve matches with the j^{th} point of the second curve, it is not possible for any point of the first curve with index $> i$ to match with a point of the second curve with index $< j^{\text{th}}$ and for any point on the first curve with index $< i^{\text{th}}$ to match with any point on the second curve with index $> j$. $w_k = (i, j)$ then $w_{k-1} = (i', j')$. Where $i - i' \geq 0$ and $j - j' \geq 0$. Thus, w is monotonically spaced in time. In figure 6 illustrates this condition. After the matching of the 5 shown point combinations, the matches corresponding to the crossed squares in the path are prohibited by this condition.

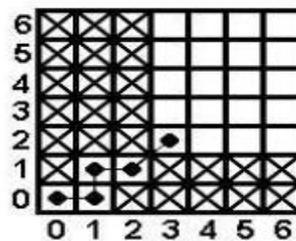


Figure 6: The Monotonicity Condition

4.4 The Penup/Pendown Condition

The penup/pendown condition ensures that pen up points of the first curve (points of which the z coordinate, which represents the pen pressure > 0) can only match with pen up points of the second curve, and that pen down points of the first curve (points of which the z coordinate ≤ 0) can only match with pen down points of the second curve. In figures 7 c and d show the matching of the allograph in figures a and b, with the stipulation turned off and on, respectively. As can be seen, the matching with the stipulation turned on is more natural than the matching with the stipulation turned off. Using the stipulation, the classifier can make a better distinction between allograph that have or do not have pen up points. The DTW is frequently used in handwriting recognition and also used in many other

disciplines including data mining, gesture recognition, speech recognition, robotics, manufacturing, and medicine [18].

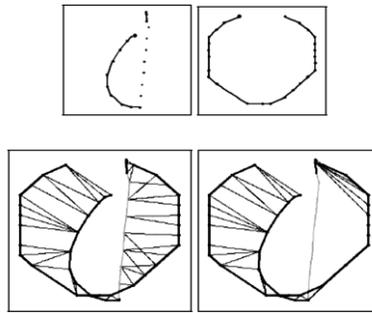


Figure 7: The Penup and Pendown Condition

5 Proposed Bidirectional Long Short Term Memory Neural Network Arabic Script Spotting System

The term Bi-directional computation means that two signal transformations, which are direct and inverse transformations exists and their performance is improved through their coupling effects. A Bidirectional neural networks model, it consists of two sub networks and can deal with two kinds of signal transformations Bi-directionally. The Bidirectional recurrent neural networks [19] are composed of two recurrent network layers, whereas the first one processes the sequence forwards and the second one processes it backwards. The both networks are connected to the same output layer, the Bidirectional net has reach to the total information about past and future sequence data points [20]. During training, the amount of contextual information that the network uses is learnt and does not have to be specified manually [21]. BLSTM neural networks make use of an arbitrary, Self- learned amount of past and future contextual information. Bidirectional networks can be applied whenever the sequence processing task is not truly online [22] in other words meaning the output is not in need after every input which makes them famous for speech recognition tasks where the output has to be present e.g. at the end of a sentence.

A drawback of conventional bidirectional RNN architectures is that the range of context that can actually be accessed is limited as the influence of a given input on the hidden layer either decays or blows up exponentially over time (vanishing gradient problem) [23]. An effective viewpoint to overcome the vanishing gradient problem is the Long Short-Term Memory architecture, which is able to store information in linear memory cells over a longer period of time and can learn the excellent amount of contextual information relevant for the classification task [24]. A Long Short-Term Memory layer is composed of recurrently connected memory blocks, each of which contains one or more recurrently connected memory cells, along with three multiplicative “gate” units firstly the input, secondly output, and thirdly forget gates. In figure 8 shows the architecture of a simple Long-Short-Term Memory block consisting of a single memory cell. The gates execution functions analogous to read, write, and reset operations. The cell input is multiplied by the activation of the input gate, the cell output by that of the output gate, and the previous cell values by the forget gate. Their influence of is to allow the network to store and retrieve information over long periods of time [25]. If example the input gate remains locked, the activation of the cell will not be overwritten by new inputs and can therefore be made available to

the net much later in the sequence by opening the output gate. This theory clear up the vanishing gradient problem and gives access to long range context information.

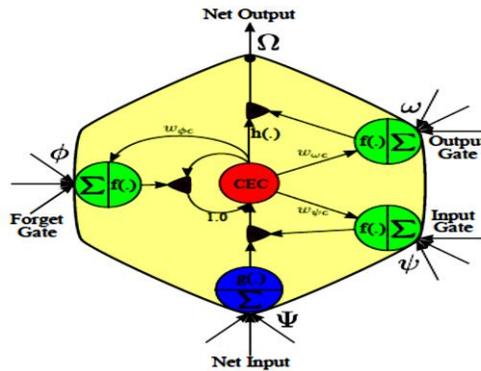


Figure 8: The LSTM Memory Block Consisting of One Memory Cell

The calculation of the activations in the forward pass of a single Long-Short-Term Memory cell is carried out in pursuance of the following procedure. The subscripts ψ , Φ , ω refer to the input gate, forget gate, and output gate, serially ψ and Ω to the net input and output of the cell. The c mentions to the memory cell and a_c to its state value. The output Ω of the single Long-Short-Term Memory cell is connected to other cells in the network.

1. The calculation of the activation $O_\psi(\tau)$ of the input gate with I containing all input units connected to the cell and H containing all hidden cells. The activation of the forget gate $O_\Phi(\tau) = f(a_\Phi(\tau))$ is computed analogously.

$$o_\psi(\tau) = f(a_\psi(\tau))$$

$$a_\psi(\tau) = \sum_{i \in I} w_{\psi i} x_i(\tau) + \sum_{h \in H} w_{\psi h} o_h(\tau - 1) + w_{\psi c} a_c(\tau - 1)$$

2. The calculation of the activation of the net input unit $O_\Psi(\tau) = g(a_\Psi(\tau))$, which receives current activations from the input units I and activations from the hidden units H one time step back.
3. The calculation of the cell state $a_c(\tau)$

$$a_c(\tau) = o_\Phi(\tau) a_c(\tau - 1) + o_\psi(\tau) o_\Psi(\tau)$$

4. The calculation of the activation of the output gate $O_\omega(\tau)$ similarly to the other gates, but using the current cell state $a_c(\tau)$

$$o_\omega(\tau) = f(a_\omega(\tau))$$

$$a_\omega(\tau) = \sum_{i \in I} w_{\omega i} x_i(\tau) + \sum_{h \in H} w_{\omega h} o_h(\tau - 1) + w_{\omega c} a_c(\tau)$$

5. The calculation of the cell output $O_\Omega(\tau)$

$$o_\Omega(\tau) = o_\omega(\tau) h(a_c(\tau))$$

The most probable sequence of Arabic characters is given by selecting the character with the highest likelihood for each step in the sequence. Latterly, this character sequence requirements to be shortened. We work with Arabic word images instead characters and lines. We are take an interest in retrieving documents based on queries which are in the first instance of Arabic words. In this paper first of all Arabic document images are segmented into words. The top line connects the entire Arabic word and increases the number of presumable shapes to be recognized, we segment the Arabic word by deleting top line. This process done with the help of a horizontal projection. We delete the top and bottom region and work with the midst region. This can result in sparing of the accuracy at the same recall. After that the garble the veracity labels sequence of the images to obtain the corresponding veracity after region it in the midst. In next phase deleting upper and lower region, each image is demonstrate as a sequence of feature vectors. No need require any further segmentation of the Arabic word images. They build the procedure powerful to the exemplary cuts and merges seen in the Arabic document images. In this paper our experiments, we use features extracted from vertical stripes of uniform length. We are extracting different feature firstly the upper and lower profiles measure the switch over of the top and bottom foreground pixel from the related Arabic word baselines. Secondly the printed background color transformation measures the number of transformation from printed to background and inverse.

The number of black pixels endows the information about the density of printed background in the vertical stripe. They input series of feature vectors are fed to the network for training all in all with their corresponding label sequences. If the system is trained after that utilization of the neural network output probabilities to return the most suitable label series for given test series. Finally, the entire Arabic database images are stored as the corresponding output label series returned by the neural network for Arabic word retrieval; these output labels are equality with the interpellation Arabic image label series. The combination of bidirectional networks and LSTM is called bidirectional LSTM, which has demonstrated wonderful performance in phoneme recognition keyword spotting, handwriting recognition and emotion recognition [26].

6 Experimental Results of Bidirectional Long Short Term Memory Neural Network Method for Arabic Script

In this section we present experiment performed to evaluate our method, and we show the results obtained and it is also comparing the performance against baseline prescript using Arabic script. We have performed two tests and evaluate their performance. In the first usage, we contemplate a two Arabic handwritten notebook in other words; it's also called AHN1 and AHN2 for testing and training. For second usage using trained AHN1 and AHN2 and test on another third Arabic handwritten notebook (AHN3) these Arabic notebooks are remark at the Arabic word level. In the first Arabic handwritten notebook (AHN1) contain 76 pages, 1640 lines, and 18304 Arabic words. Second Arabic handwritten notebook (AHN2) contains 91 pages, 2170 lines, and 22304 Arabic words as well as third Arabic handwritten notebook (AHN3) contain 113 pages, 2409 lines, and 24821 Arabic words, all three notebooks for using testing and training. In the first test, the Arabic document images were scanned and segmented into Arabic word images. In this paper, we are trained the network using the training set and the validation set to adjudicate the stopping criteria during the iterative training process.

Table 1: The Inside Lexicon and Outside Lexicon Queries

Queries	A	Mean Average Accuracy (MAA)
Inside Lexicon	93.85	85.79
Outside Lexicon	93.13	83.37

As Input every Arabic word image in the test set is given to the trained network. The network outputs a most probable sequence of labels for each Arabic image with the help of output possibility. We are utilizing 15% as the validation set and 70% of the images as the training set and the remaining 15% were used as a test set. In the next section count the edit distance from the query words for Arabic word retrieval. In this section first set contains words which already exist in the training set, it called inside lexicon and second set contains words which do not exist in the training set it called outside lexicon.

Table 2: A Comparison with the Euclidean Distance, DTW and BLSTM Model

Method	A	Mean Average Accuracy (MAA)
Euclidean Distance	79.43	72.57
Dynamic Time Warping	85.31	78.81
BLSTM Prescript	92.89	85.93

Here A represents the accuracy at a recall rate of 60%. Mean Average accuracy (MAA) is the mean of the area under the accuracy recall curve for all the queries. The results of inside lexicon and outside lexicon queries are shown aside in table 1. In another test using an Arabic handwritten notebook and trained the neural network on AHN1 and AHN2 and tested on AHN3. In the retrieval some operation performed firstly count their edit distances from the rest of the Arabic test images using their label sequence representations. Secondly, count the accuracy at a recall rate of 60%. We got MAA about 85.93% on this test. The appropriate results of proposed prescript are shown in Table 2. The Accuracy recall curve corresponding to the BLSTM is shown in Figure 9. We compare appropriate results with word spotting prescript with DTW in this case feature vectors are the homogeneous. An additional compare results using Euclidean distance.

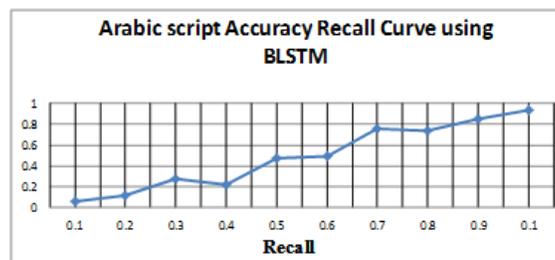


Figure 9: The Arabic Script Accuracy Recall Curve uses BLSTM

The Euclidean distance is the distance between two points in Euclidean space that one would measure with a ruler, and is given by the Pythagorean formula. Euclidean space was originally devised by the Greek mathematician Euclid around 300 B.C.E. to study the relationships between angles and distances. This system of geometry is still in use today. Euclidean geometry specifically applies to spaces of two and three dimensions. However, it can easily be generalized to higher order dimensions.

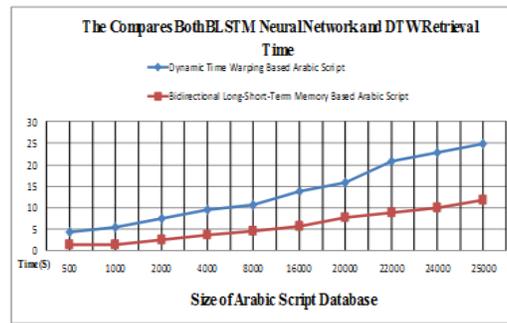


Figure 10: The Compares Both BLSTM Neural Network and DWT Retrieval Time

For Euclidean, all Arabic words are normalized to the homogeneous width. Its show the comparison of the BLSTM neural network rooted prescript with DTW distance and Euclidean distance rooted approach in the Table 2. It may be noticed that the proposed prescripts are better at retrieving episodic Arabic words. The plot in figure 10 compares the retrieval time for both Bidirectional Long-Short-Term Memory neural network prescript and Dynamic Time Warping rooted retrieval. It is distinctly notice that the BLSTM neural network rooted prescript is intensely than the DTW rooted matching and search process because DTW is the heavy computational burden required to find the optimal time alignment path.

7 Conclusion

In this paper, a word retrieval scheme for Arabic document is presented. Today scenario Arabic document recognition is exceedingly arduous to automate. The humankind being can diagnose variegated objects and make cognition out of volumetric amount of visual information, seemingly requiring very diminutive attempt. The emulate task execution by humankind to diagnose to the extent allowed by physical barricades will be extremely gainful for the system. The difficulty contains in the real world data handwritten Arabic alphabets, where handwritten Arabic characters are the input to the system, while in print characters will be goal output of the system. In this paper, we proposed a novel approach at reflow on an Arabic document image, using a Bidirectional Long-Short-Term Memory Neural Network (BLSTMNN). The designed to take relating to information into account, these networks can maintain Arabic word images that cannot be durable segmented into individual Arabic characters. The partitioning Arabic word, we easier the problem and receive elevated reflow rates. In this paper queries as well as the database images are transformed into a sequence of feature vectors and compare is oftentimes carried out with the help of Dynamic Time Warping (DTW). The favorable outcome originates from a well-balanced system. We outshine a sophisticated recognition technique for the retrieval task. On the contrary, by avoiding a complete recognition, and bypass the challenges posed by the Arabic script. In the future, we would like to evaluate different features of this network and also use a various language model and dictionary based post-processor to ameliorate the with more precision. We would also like to enlarge our work towards recognition of other various languages.

REFERENCES

- [1] S. V. Rice, G. Nagy, and T. A. Nartker, Optical Character Recognition: An Illustrated Guide to the Frontier. Kluwer, 1999.

- [2] H. Alamri, J. Sadri, C. Y. Suen, and N. Nobile. A novel comprehensive database for Arabic off-line handwriting recognition. In Proc. 11th Int. Conf. on Frontiers in Handwriting Recognition (ICFHR), pages 664–669, 2008.
- [3] T. M. Rath and R. Manmatha, “Word spotting for historical documents,” *IJDAR*, vol. 9, no. 2-4, pp. 139–152, 2007.
- [4] Pramod Sankar K., C. V. Jawahar and R. Manmatha, “Nearest Neighbor based Collection OCR,” in Proc. DAS, 2010.
- [5] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–80, 1997.
- [6] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke. A novel word spotting method based on recurrent neural networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(2):211–224, 2012.
- [7] F. Eyben, S. Bock, B. Schuller, and A. Graves, “Universal onset detection with bidirectional long short-term memory neural networks,” in *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR 2010)*, 2010.
- [8] M. Cheriet, M. Beldjehem. , “Visual Processing of Arabic Handwriting:Challenges and New Directions” , Summit on Arabic and Chinese handwriting (SACH’06), Washington-DC, USA, pp 129-136, 2006.
- [9] J. Dichy ,” On lemmatization in Arabic. A formal definition of the Arabic entries of multilingual lexical databases “ , ACL 39th Annual Meeting. Workshop on Arabic Language Processing; Status and Prospect. Toulouse, pp 23-30, 2001.
- [10] MA. Attia , T. Salakoski , F. Ginter , S. T. Pyysalo ,” Accommodating Multiword Expressions in an Arabic LFG Grammar “, In *Finland Springer-Verlag Berlin Heidelberg*, vol 4139, pp 87 – 98, 2006
- [11] R. Moghaddam and M. Cheriet. Application of multilevel classifier and clustering for automatic word spotting in historical document images. In Proc. 10th Int. Conf. on Document Analysis and Recognition (ICDAR), pages 511–515, 2009.
- [12] T. van der Zant, L. Schomaker, and K. Haak. Handwritten-word spotting using biologically inspired features. *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on*, 30(11):1945 –1957, Nov. 2008.
- [13] A. Kolz, J. Alspector, M. Augusteijn, R. Carlson, and G. V. Popescu, “A Line-oriented Approach to Word Spotting in Handwritten Documents,” *Pattern Analysis and Applications*, 2(3), pp. 153–168, 2000.
- [14] B. Gatos and I. Pratikakis, “Segmentation-free word spotting in historical printed documents,” in *International Conference on Document Analysis and Recognition*, July 2009, pp. 271– 275.

- [15] M. Burl and P. Perona, "Using Hierarchical Shape Models to Spot Keywords in Cursive handwriting," IEEECS Conference on Computer Vision and Pattern Recognition, pp. 535–540, June 1998.
- [16] Joseph B. Kruskal and Mark Liberman. The symmetric time-warping problem: from continuous to discrete. In David Sanko and Joseph B. Kruskal, editors, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparisons*. Addison-Wesley, Reading, Massachusetts, 1983.
- [17] C. S. Myers and L. R. Rabiner, "A Comparative Study of Several Dynamic Time-warping Algorithms for Connected Word Recognition," *The Bell System Technical Journal*, 60(7), pp. 1389- 1409, September 1981.
- [18] Toni M. Rath and R. Manmatha. Word image matching using dynamic time warping. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2, pages 521-527, Madison, WI, USA, 2003.
- [19] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45:2673–2681, November 1997.
- [20] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2673–2681, 1997.
- [21] Yusuf Perwej , Firoj Parwej, "A Neuroplasticity (Brain Plasticity) Approach to Use in Artificial Neural Network" for published in the *International Journal of Scientific & Engineering Research (IJSER)*, France , Vol.3, Issue 6, June 2012, Pages 1- 9, ISSN 2229 – 5518.
- [22] A. Graves, S. Fernandez, and J. Schmidhuber, "Bidirectional lstm networks for improved phoneme classification and recognition," in *Proc. of ICANN*, Warsaw, Poland, 2005, pp. 602–610.
- [23] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In S. C. Kremer and J. F. Kolen, editors, *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press, 2001.
- [24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [25] A. Graves and J. Schmidhuber. "Framewise phoneme classification with bidirectional LSTM and other neural network architectures", *Neural Networks*, 18(6):602– 610, 2005.
- [26] M. Wollmer, F. Eyben, J. Keshet, A. Graves, B. Schuller, and G. Rigoll, "Robust discriminative keyword spotting for emotionally colored spontaneous speech using bidirectional LSTM networks," in *Proc. of ICASSP*, Taipei, Taiwan, 2009.