A Sequence Learning Approach for Multiple Script Identification

Adnan Ul-Hasan*, Muhammad Zeshan Afzal*, Faisal Shafait†, Marcus Liwicki*‡, and Thomas M. Breuel*
*Department of Computer Science, University of Kaiserslautern, Kaiserslautern, Germany.
Email: {adnan,tmb}@cs.uni-kl.de, afzal@iupr.com
†NUST School of Electrical Engineering and Computer Science, Islamabad, Pakistan.
Email: faisal.shafait@seecs.edu.pk
‡DIVA Research Group, University of Fribourg, Switzerland.
Email: marcus.liwicki@unifr.ch

***Abstract*—In this paper, we present a novel methodology for multiple script identification using Long Short-Term Memory (LSTM) networks' sequence-learning capabilities. Our method is able to identify multiple scripts at text-line level, where two or more scripts are present in the same text-line. Unlike traditional techniques, where either shape features or bounding boxes of individual characters are extracted, the LSTM-based system learns a particular script in a supervised learning framework. Moreover, this system neither needs specific features nor other preprocessing steps other than text-line extraction and text-line normalization. The proposed method works on text-line level, where it identifies each character as belonging to a particular script. We have developed a database consisting of English and Greek script, and our system achieved a script recognition accuracy of 98.186% on this dataset.**

***Keywords*—Script-identification, LSTM networks, multilingual OCR**

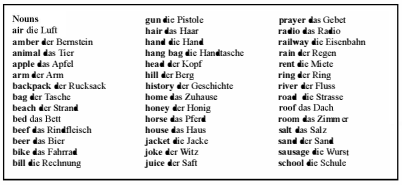
I. Introduction

Multilingual OCR is a difficult task and it presents several unique challenges that are otherwise are not encountered in unilingual OCR tasks. The traditional process for dealing with such documents for the purpose of OCR is to identify and separate the individual scripts. By doing so, the problem can be reduced to applying techniques for uni-lingual OCR.

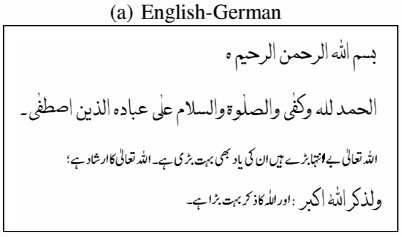
Diverse categories of multi-script documents exist, which can be broadly divided into three main categories:

1) the script is the same, albeit the languages are different, e.g., English-German (Latin Script) or Sanskrit and Hindi (Devanagari script). (Figure 1-a shows an example).
2) the character-set is the same, yet the scripts are different, e.g., Persian or Urdu documents in Nastaleeq or Naskh scripts (see Figure 1-b).
3) where both script and language are different (Figure 1-c), e.g., English/Greek or Latin/Arabic documents, etc.

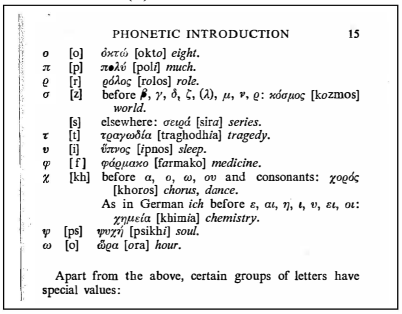
From the point of view of OCR, the most difficult task is to separate scripts of the first category where both languages are written with the same script. In our recent work [1], we have demonstrated that instead of identifying individual scripts for such documents, we can develop OCR models for a whole family of the given script(s). In this manner, we can avoid the harder task of distinguishing very similar languages. However, the identification of scripts is still important for other purposes.



(a) English-German



(b) Urdu-Arabic



(c) English-Greek

Figure 1. Types of commonly available multilingual/multiscript documents where: (a) the script is same (Latin) but the languages are different (German and English), (b) two languages share a common character-set but different scripts, (c) both the language and the script are different in a multilingual document.

This paper, however, addresses the challenge mentioned in the third category, i.e., identifying scripts in documents where both the language and the script are different. Our method is based on a recent variant of recurrent neural networks, *Long Short-Term Memory (LSTM)* networks. Unlike traditional approaches, where individual characters/symbols are segmented before recognition, LSTM networks learn multiple scripts on a single text-line level. Our methodology is more like a segmentation-free OCR approach, where sequnce-to-sequence mapping is done using discriminative learning.

978-1-4799-1805-8/15/$31.00 ©2015 IEEE

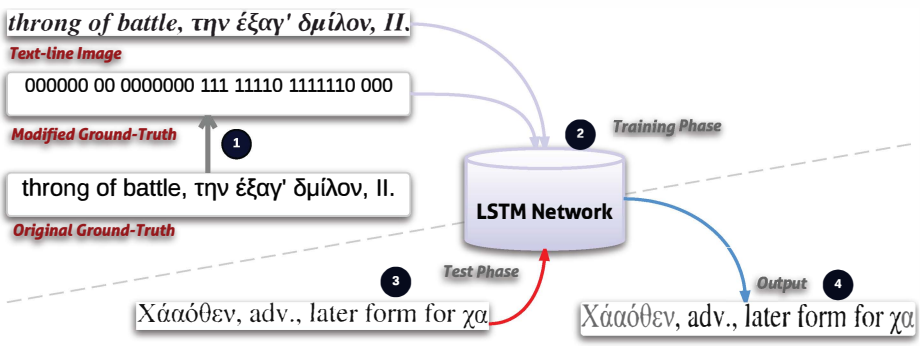


Figure 2. The pipeline of identifying multiple scripts using LSTM Networks. (1) The ground-truth information of a given text-line in a training set is modified to a string of class labels. (2) The LSTM network learns individual scripts using the modified ground-truth as target labels for the individual characters. (3) During the **test phase**, the test text-line image is segmented into individual scripts. Noteworthy, during the test phase, trained LSTM emits a sequence of class labels along with their approximate locations. (4) The location information is used to segment the input text-line image into respective script-images. In the shown sample image, the gray color shows the Greek script, while the black color shows the English (Latin) script.

This paper is further organized as follows. In the next section, a brief literature review on script identification is presented. Proposed script identification strategy is explained in Section II, while Section III describes the experimental evaluation. Discussion on results and error analysis, is given in Section IV and Section V concludes the paper with future directions.

A. Related Work

Pioneering work on multiple script identification was done by Spitz [2]. He used optical density distribution as a feature for the Han script and used typographic character codes for Latin-based script identification. In an another early work, Pal and Chaudhuri [3] developed a script identification system to recognize printed Roman, Chinese, Arabic, Devanagari, and Bangla text lines. They used different features to identify a particular script.

A comprehensive review on multiple script recognition was reported by Ghosh et al. in [4]. They divided the script recognition process into two categories, i.e., structure-base approach and appearance-based approach. He also gave comparative analysis of numerous systems. Appearance-based approaches are global in the sense that they extract text-blocks irrespective of document's script. They are usually faster and are applicable to a wide variety of scripts. Structure-based local approaches are useful where script identification is required at line-, word- or character-level. Both approaches require some kind of segmentation at the pre-processing stage.

Sharma et al. [5] proposed a methodology for script identification in video frames. They used three well-known features, namely, Zernike moments, Gabor and gradient features. SVM classifier was used to recognize a script based on these features. They evaluated their method on English, Hindi, and Bangla scripts. In order to overcome the problems associated with video frames, they used super-resolution and skeletonization as pre-processing steps.

Rani et al. [6] proposed a script identification method for pre-segmented English and Gurmukhi characters. They extracted Gabor features and gradient features from the characters and applied SVM for recognition.

Recently, Echi et al. [7] proposed a script identification technique for both printed and handwritten text for Arabic-Latin documents. They used many well-known features in addition to their own features to identify Arabic and English. They compared the performance of several classifiers including Naive Bayes, *k*-NN, Decision Trees, Multilayer perceptron (MLP) and Support vector machines (SVM) on selected features.

All of the above-mentioned methods rely on extraction of sophisticated features and image processing techniques along with various heuristics for script identification. This is because of the common belief that script-related features are essential for reliable script identification. Therefore, major effort has been spent by the research community to find and fine-tune script-related features. One big issue with all feature-based methodologies is the extensive use of heuristics to adapt methods to the task at hand (issue of adaptability).

Modern machine learning techniques, like deep belief networks and deep convolutional neural networks, are gaining popularity in the fields of computer vision and pattern recognition because they don't rely on sophisticated hand-crafted features; instead they extract features from the given data themselves, and still perform better than those methods based on hand-crafted features. Likewise, there has been many efforts in applying these techniques for many document analysis tasks including script-identification.

Rashid et al. [8] proposed a script identification method based on convolutional neural networks (CNN). The CNN was used to learn the shapes of individual connected components. No explicit features were extracted to estimate the shapes of these components. They tested their method on Latin/Greek, Latin/Arabic and Antiqua/Fraktur scripts. They, however, applied a post-processing step to assign small diacritics to corresponding base shapes.

Genzal et al. [9] proposed an HMM-based script identification system. They evaluated their system on 54 languages with 18 different scripts. On a private database, they reported an average script identification error rate of 1.73% for these scripts. The text-lines in their work were assigned a primary script if it had more than one script.

Both of the above-mentioned techniques [8], [9] are based on modern machine learning principle; however, the first work applies post-processing step to improve the accuracy, and in the second approach, carefully selected features were extracted and

also only the most frequently occurring script in a multi-script text-line was considered. On the other hand, our approach is very simple in that we did not extract any feature before the training, nor we applied any kind of post-processing on the output sequences. The details of our method are described in the following section.

II. Script Identification using LSTM Networks

The pipeline for the whole process is shown in Figure 2. We address the script identification challenge as a multi-class classification problem. All characters of a particular script are assigned a single target class-label, which LSTM learn in a supervised classification paradigm. We do not segment a text-line into individual characters; instead LSTM network learns scripts' shape in a sequence-to-sequence mapping mode. In simple words, we are training the LSTM network (see Section II-A for details) to learn a particular target class-label for all alphabets in a given language/script. We have used the open source OCRopus document analysis system [11] for our experiments. The OCRopus line recognizer is basically designed for transcription task, where the goal is to transcribe a given text-line. So, we have essentially adapted this OCR system to perform script recognition.

The modification we made to adapt the OCR system is that instead of giving the ground-truth information about the individual characters in a text-line, we provided a sequence of class-labels as the ground-truth at the training stage. To achieve this, the ground-truth information is modified so that it represents a two-class classification problem (see Figure 2). During the training, the modified ground-truth information is used as the target sequence for a given text-line image. LSTM network learns the characters' class association based on their own shape as well as the contextual information of surrounding shapes. In our experience, the LSTM line-recognizer first learns the *space* locations, and then try to learn the shapes of individual characters.

At the test stage, the test text-line images are given to the trained network and it emits class label for a each script in the text-line. The recognition is actually done at the character level, and the network produces a string of labels corresponding to each script as shown in Figure 4. The network also predicts the location of individual characters, which in turn can be used to separate the scripts.

Since the proposed approach for multiple script identification is designed to work on text-line images directly, it is not necessary to extract individual characters from the page or text-line images as done in most of traditional approaches. Only text-line extraction is required from a given full page. It is important to note that we do not identify a dominant script in a text-line if more than one scripts are present. In our approach, individual characters are identified as belonging to a particular script.

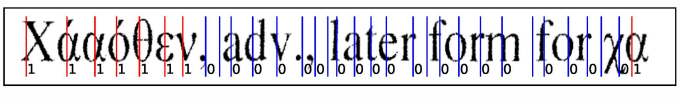


Figure 4. The output from the LSTM trained model for script identification is shown in vertical lines with associated class labels. The vertical line is at the approximate location of the character predicted by the LSTM line-recognizer. This information is then used to separate the two scripts.

Traditionally, some features are extracted from the input image to discriminate between characters of different scripts. LSTM networks, like other contemporary machine learning techniques, have shown to differentiate between many classes based on just raw pixel values [12]. Therefore, the raw pixel values have been used as the only features.

A. System Architecture

We used a 1D-LSTM architecture for the LSTM network. It is the same as described in [12]. In 1D-LSTM networks, a sliding window of 1-pixel width and of height equal to the image height traverses the input text-line image to convert it into a one dimensional sequence. The height of the image is termed as "depth" of the 1D sequence and each slice of the image thus obtained is called a "frame". It is necessary to make the depth of all images equal, so that the individual frames in each image are equal. This process of making heights equal is called "normalization", and it is essential for such types of networks. For this work, we use the same normalization method used in [13] and it is available in OCRopus system. This method of normalization is based on filter operations and affine transformation and can be used for many scripts.

The LSTM network consists of one input layer, one hidden layer containing 50 LSTM cells and one output layer. One such LSTM cell (simplified version) is shown in Figure 3. The advantage of such a neural architecture is that it avoids the problem of very small gradients multiplied with each other (the *vanishing gradient problem* [14]). The **forget gate** allows the network to "remember" long-term contextual information. CTC layer [10] is commonly used to align the target labels with the output activations. The use of this layer allows the LSTM networks to be trained on full sequences directly without the need to segment the input sequence. The LSTM network thus obtained can do both localization and recognition of individual characters simultaneously.

For 1D-LSTM training, a sliding window of 48×1 traverses over the input image and each frame is fed to the LSTM network for training. The network learns to classify each frame into one of the target classes (reject class included) based on the contextual information of that frame. We used the bidirectional mode of LSTM architectures, where there are basically two hidden layers, one traversing the input from right-to-left and the second from left-to-right. Both of these layers are connected to a single output layer.

III. Experimental Evaluation

A. Database

There is no standard ground-truthed database available for script-identification purpose. Therefore, we developed our own synthetic database consisting of text-line images from freely available English-Greek text. We used a line generation tool available in OCRopus for this purpose. This tool can generate any number of text-line images with given text file and font-files. We modified this utility to generate a ground-truth file for each transcription that suits our needs. Some of the degradation models described in [15] were also incorporated while generating text-line images. We generated $90,000$ text-line images in a variety of fonts (both serif and sans-serif). We

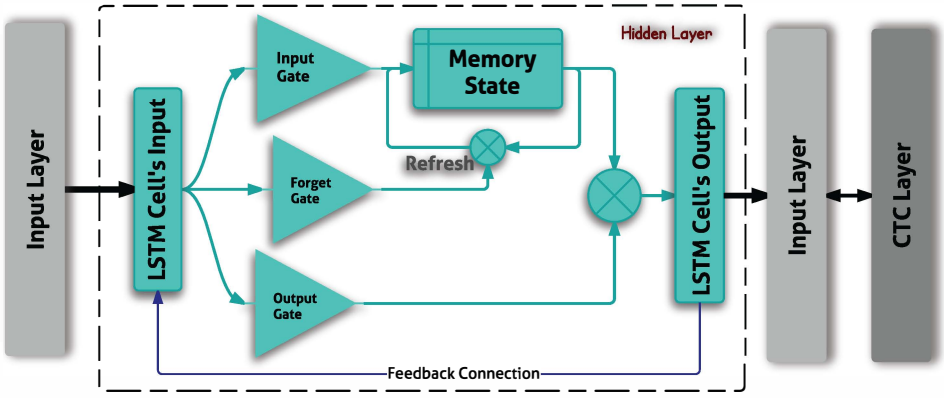


Figure 3. Simplified 1D-LSTM architecture. The Hidden layer consists of LSTM memory blocks. The input gate allows the **input** to be read, **output** gate allows outputs to be written and the **forget** gate allows retention of the information within the memory cell. The CTC layer aligns the output activations with the ground-truth sequence [10]. The input image is traversed by a $X \times 1$ (X being the height of the image) window to convert the 2D image into a 1D sequence.

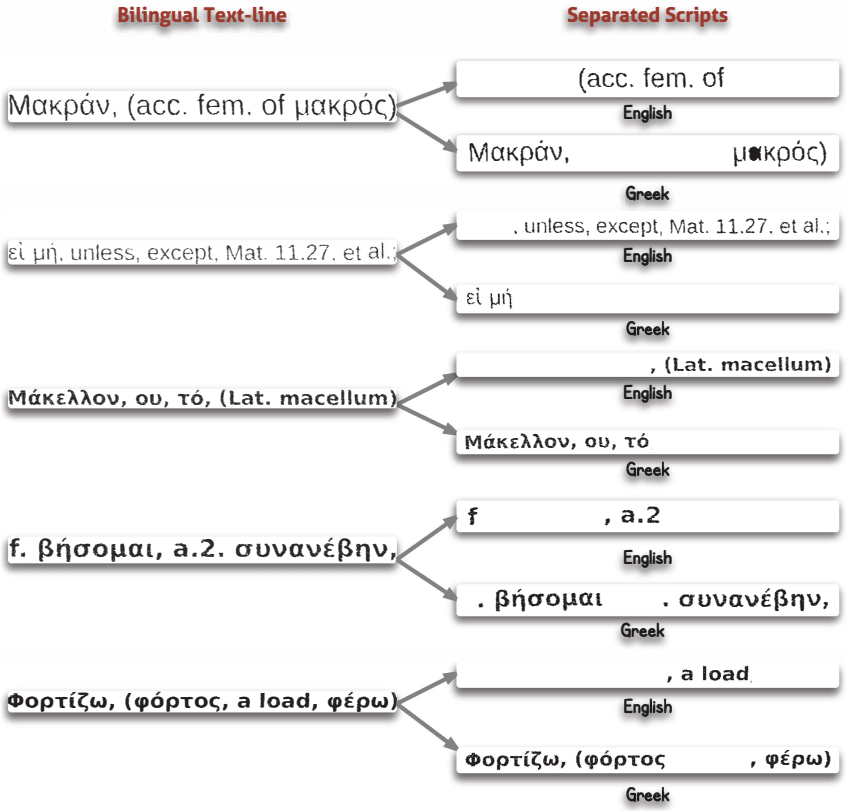


Figure 5. Samples of some text-lines separated by using LSTM-based script identification method. It is important to note that errors resulted from the punctuation marks including parentheses are not considered as these marks are generally script independent.

also generated a separate test database consisting of 9,500 text-line images from a different English-Greek text. This database is available free of cost and can be obtained from the authors.

B. Performance Metric

Since our training data is trained on text-line images with the ground-truth containing a sequence of corresponding class-labels, the trained models emits a sequence of class-labels in the form of a Unicode string for the given text-line at the test stage. Since we have adapted an OCR system for the script identification purpose, the output is like a transcription. Therefore, the performance is measured in terms of edit distance (or *Levenshtein Distance*).

IV. Results and Analysis

There were a total of 9,500 text-line images in our test dataset consisting of 289,013 characters. LSTM network trained for script-identification task as described in Section II yielded an accuracy of 98.19% (an error of 1.81%). Some sample test images are shown in Figure 5. Top 15 confusions are shown in Table I. It must be noted that we did not consider errors made by punctuations as they are generally script-independent.

Table I. TOP 15 CONFUSIONS FOR SCRIPT IDENTIFICATION TASK.

GT	No.	GT	No.	GT	No.
ν	749	ο	717	ε	552
ι	287	ρ	277	a	228
u	210	τ	174	x	149
φ	136	λ	135	η	128
ω	120	σ	120	γ	16

GT – Ground-truth,
No. – No. of occurrences

From Table I, one can see that the many top confusions are due to the similarity in shapes of characters between the two scripts, like 'v' with ν, 'o' with ο, 'i' with ι, 'p' with ρ, 'a' with α, 'u' with υ, 't' with τ and so on. These kind of errors will not be present in the case when characters of both scripts are significantly different, e.g., English and Hindi or Arabic and Chinese, etc. A bigram or trigram analysis could be done to further investigate the causes of such confusions.

Many confusions occur at the locations where the script changes from English to Greek or vice versa. For example, in Figure 4, the second last character χ is misclassified as 'X' because the preceding character is an 'r' (English). Contextual information at that time-step incorrectly makes the LSTM network classify this character as belonging to Latin script. Again, such errors will not occur in documents where characters' shape are very different.

Script identification has been reported by many researchers, but unavailability of a standard database hinders the performance comparison among reported methodologies. Moreover, the difference in performance metrics also obstruct a fair comparison.

The results also show the powerful learning capabilities of LSTM networks. LSTM networks were able to learn many variation in shapes for the same class, for example, it could recognize the class association of whole English and Greek scripts to a single class-label. This suggests us that these networks are very useful for large-scale classification problems where class members show large variations.

V. CONCLUSIONS

This paper introduces a new sequence-learning based method for identifying multiple scripts in multilingual documents. LSTM-based line recognizer has been trained to learn class associations of individual characters for a given bilingual text-line. Experimental evaluation on bilingual English-Greek data has shown encouraging results. This work can be extended further in numerous ways. Firstly, other bilingual documents such as English-Devanagari, Arabic-Latin or English-Chinese can take benefit directly from this approach. Secondly, this approach can be adopted for documents with more than two scripts. Another research direction could be that LSTM networks can be used to directly OCR the multilingual documents. They have demonstrated good competency to learn a lot of classes at the same time; this means that we can avoid the script identification step in some, if not all, multilingual documents.

ACKNOWLEDGEMENTS

This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project Kallimachos (01UG1415C).

REFERENCES

[1] A. Ul-Hasan and T. M. Breuel, "Can we Build Language Independent OCR using LSTM Networks?" in *International Workshop on Multilingual OCR*, Washington D.C., USA., Aug 2013, p. 9.

[2] A. L. Spitz, "Multilingual Document Recognition," in *Handbook of character Recognition and Document Image Analysis*, H. Bunke and P. S. P. Wang, Eds. World Scientific Publishing Company, 1997, pp. 259–284.

[3] U. Pal and B. B. Chaudhuri, "Identification of different script lines from multi-script documents," *Image Vision Computing*, vol. 20, no. 13-14, pp. 945–954, 2002.

[4] D. Ghosh, T. Dube, and A. P. Shivaprasad, "Script Recognition - A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2142–2161, 2010.

[5] N. Sharma, S. Chanda, U. Pal, and M. Blumenstein, "Word-wise script identification from video frames," in *ICDAR*, Washington D.C. USA, Aug 2013, pp. 867–871.

[6] R. Rani, R. Dhir, and G. S. Lehal, "Script Identification of Pre-segmented Multi-font Characters and Digits." in *ICDAR*, Washington D.C., USA., Aug. 2013, pp. 1150 – 1154.

[7] A. F. Echi, A. Saidani, and A. Belaid, "How to separate between Machine-Printed/Handwritten and Arabic/Latin Words?" *Electronic Letters on Computer Vision and Image Analysis*, vol. 13, no. 1, pp. 1–16, 2014.

[8] S. F. Rashid, F. Shafait, and T. M. Breuel, "Discriminative Learning For Script Recognition," in *ICIP*, Hong Kong, sep 2010, pp. 2145 – 2148.

[9] D. Genzel, A. C. Popat, R. Teunen, and Y. Fujii, "HMM-based Script Identification for OCR," in *International Workshop on Multilingual OCR*, Washington D.C., USA., Aug. 2013, p. 2.

[10] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks." in *ICML*, 2006, pp. 369–376.

[11] "OCRopus – Open Source Document Analysis and OCR System." [Online]. Available: https://github.com/tmbdev/ocropy

[12] T. M. Breuel, A. Ul-Hasan, M. Al Azawi, F. Shafait, "High Performance OCR for Printed English and Fraktur using LSTM Networks," in *ICDAR*, Washington D.C. USA, Aug 2013, pp. 683 – 687.

[13] M. R. Yousefi, M. R. Soheili, T. M. Breuel, and D. Stricker, "A Comparison of 1D and 2D LSTM Architectures for Recognition of Handwritten Arabic," in *DRR–XXI*, San Francisco, USA, 2015.

[14] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] H. S. Baird, "Document Image Defect Models," in *Structured Document Image Analysis*, H. S. Baird, H. Bunke, and K. Yamamoto, Eds. Springer-Verlag, 1992.