Script and Emphasis Detection using Recurrent Neural Networks

Christoph Gümmer
Leipzig University
Leipzig, Germany

Benjamin Kiessling
Université PSL
École Pratique des Hautes Études
Paris, France
benjamin.kiessling@psl.eu

Daniel Kinitz
Bibliotheca Arabica
Saxon Academy of Sciences
Leipzig, Germany
kinitz@saw-leipzig.de

Parivash Mashhadi
Leipzig University
Leipzig, Germany

I. Introduction

In Digital Humanities research documents containing multiple scripts and extensive text emphasis for semantic purposes are common, ranging from relatively simple parallel texts, to mixed Fraktur-Antiqua printing, dictionaries, and library catalogs. With the increased availability of Optical Character Recognition software at least in part accessible to the determined DH scholar robust script and text emphasis detection methods are of special importance for effective digitization of these works.

State of the art neural sequence-to-sequence models have largely supplanted older character-based methods for Optical Character Recognition. While neural methods have generally higher accuracy and decreased requirements on training data annotation depth, some earlier approaches, most notably the tesseract OCR engine [1], featured seamless classifier combination and common text emphasis detection. Neither are available in any freely licensed OCR package utilizing the advances of machine learning in the last decade.

A. Related work

Past approaches to segmentation-less multilingual OCR have focused on building combined models capable of recognizing multiple scripts [2]. Combined models are undesirable for multiple reasons. The irregularity of early modern printing and large number of typefaces result in character accuracy below 95% for mixed-font models even on mono-script texts [3] necessitating time consuming training data acquisition and retraining of these large models. In addition, reusing training data is regularly prevented by being embedded in other non-target scripts or typefaces and legal restrictions imposed by digitization agents..

A second direction labels OCR input images, most often lines, to be able to select appropriate monolingual recognition models.

The method described in [4] labeling whole lines using a recurrent neural network is inappropriate for many humanities texts because of extensive intra-line script switching. A recurrent neural script classifier based on overlapping sliding window profile feature sequences is shown in [5]. [6] published a conceptually simpler approach without feature extraction directly classifying character script using an LSTM network. A refined version of the latter method is the basis of our script detection system.

II. RNNs for Script and Emphasis Detection

A. Script Detection

The system treats script detection as a segmentation-less sequence classification problem, similar to text recognition. Instead of assigning a unique label to each code point or grapheme cluster we assign all code points of a particular script the same label, the network is trained to output the correct sequence of script labels using the CTC loss function [7]. It should be noted that CTC is on the face unsuitable for this task, as it includes no mechanism to ensure temporal alignment between graphemes in the input sequence and output activations; fortunately the LSTM network's activation are fairly close to their corresponding location in the input sequence. The output sequence is then used to split the line into single-script runs that can be classified with monolingual recognition models.

Script classes are ISO 15924 codes determined through each code point's Unicode script property[1] As there are graphemes that occur in multiple scripts, chiefly numerals and punctuation, we retain both the common and inherited properties. Merging these during post-processing based on their surrounding script increases robustness when classifying non-body text such as page numbers and tables, compared to fusing them beforehand. Bidirectional text is dealt with by rearranging the target sequence into display order using the Unicode BiDi algorithm before script assignment.

Apart from the mentioned merging step, two additional post-processing steps are performed. The first substitutes all individual runs of a line with the whole line if only a single script remains after common/inherited merging. The second stems from the observation that often only

[1]ISO 15924 includes separate identifiers for Antiqua and Fraktur texts and similarly visually distinct calligraphic hands for Syriac which are subsumed as Latin and Syriac in the Unicode database.

a subset of scripts the detection network is trained on occur in any document. A whitelist is added, merging runs of non-included scripts into the surrounding context after filtering for common confusions (Arabic-Syriac and Latin-Fraktur).

B. Emphasis Recognition

We evaluated two methods of encoding two common text emphasis methods for recognition by a RNN. Initially, italicized and text components with increased letter spacing were marked up with special start and stop markers and the model was trained to produce these markers. While the results of the training were promising, obtaining the amount of training data needed to reliably place both markers was infeasible for our target documents. Creating separate labels for italicized/spaced graphemes and training for these, remedied the marker placement issue with a sufficiently small amount of training data.

Separate alphabets for emphasized text components increase model size and execution time, tripling the size of the final fully connected layer. This large increase in possible output labels also seems to preclude fine-tuning base models by resizing the final linear projection of the network.

C. Architecture

Both the script detection and emphasis recognition share a common network architecture of bidirectional Long short-term memory RNN blocks trained with Connectionist Temporal Classification loss and single-sample stochastic gradient descent with momentum (learning rate:, momentum: 0.9). Early stopping is used to terminate training. The system is implemented as part of the kraken OCR engine.

The networks operate on binarized whole lines. Baselines and line height are normalized using a slightly modified version of the centerline normalizer implemented in the OCRopus system.

III. Preliminary Results

The script detection and emphasis recognition were evaluated as part of Bibliotheca Arabica which aims to gain new insights into Arabic literature from 1150 to 1850 CE by analysing the ways of production, transmission, and reception of texts. The basis of this research are ~500 library manuscript catalogs which are usually multilingual and employ structured text emphasis as semantic markup.

A. Dataset

We repurposed publicly available non-synthetic training data for recognition models to build a corpus of 76000 script-annotated line images containing Arabic, Cyrillic, polytonic Greek, Hebrew, Latin, Fraktur, and (western) Syriac text. The majority of text lines contain only a single non-common script although there are mixed lines for all scripts in the corpus. The exact distribution of code points is shown in table I. 760 randomly selected lines are separated from the corpus as a test set.

TABLE I
Script distribution in corpus

Script	Code Points	Lines
Arabic	916368	19757
Cyrillic	65524	1904
Greek	198324	5262
Hebrew	102575	2117
Fraktur	137332	3618
Latin	1570618	40809
Syriac	54092	1752
Inherited	30588	14236
Common	837931	75146

Emphasis recognition was evaluated on an English and romanized Arabic catalog using emphasis described in II-B with 350 transcribed lines. An additional 50 line transcriptions were used as a test set. Overall 220 lines contain some kind of text emphasis. It is representative of a large number of catalogs in purview of Bibliotheca Arabica.

B. Script Detection

The fully trained network yielded a character accuracy of 94.62% on the test set. Output for a French-Arabic bilingual sample page can be seen in 1. The misclassification of Eastern Arabic numerals as Latin text is caused by the transcription as Latin Arabic numerals in the ground truth.

C. Emphasis Recognition

The overall character accuracy of the network on the test set is 96.10%, with 96.38% on cursive and text with increased spacing.

References

[1] R. Smith, D. Antonova, and D.-S. Lee, "Adapting the tesseract open source ocr engine for multilingual ocr," in *Proceedings of the International Workshop on Multilingual OCR*, ACM, 2009, p. 1.

[2] A. Ul-Hasan and T. M. Breuel, "Can we build language-independent ocr using lstm networks?" In *Proceedings of the 4th International Workshop on Multilingual OCR*, ACM, 2013, p. 9.

[3] U. Springmann, F. Fink, and K. U. Schulz, "Automatic quality evaluation and (semi-) automatic improvement of mixed models for ocr on historical documents," *CoRR. URL: https://pdfs. semanticscholar. org/b8b7/c369a01164b289ed7c41ef33ce1b74e0fb1f. pdf*, 2016.

[4] Y. Fujii, K. Driesen, J. Baccash, A. Hurst, and A. C. Popat, "Sequence-to-label script identification for multilingual ocr," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, IEEE, vol. 1, 2017, pp. 161–168.

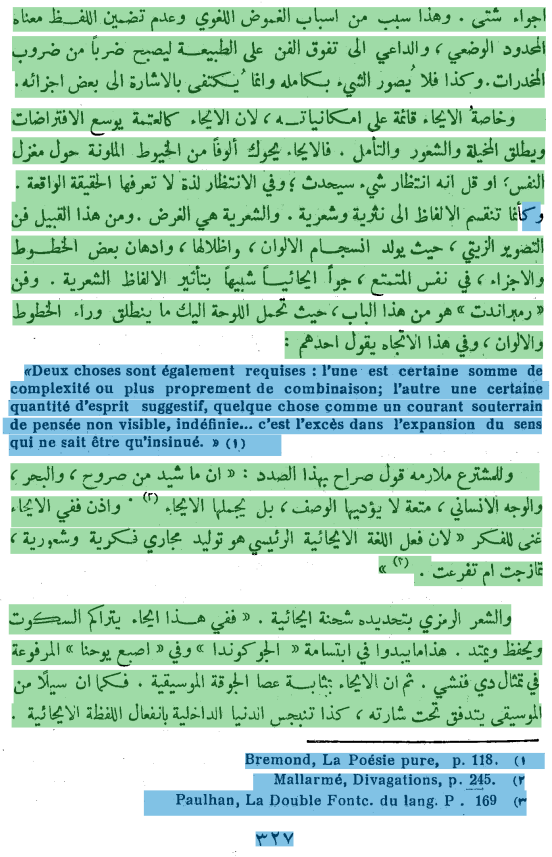

Fig. 1. Script recognition on French-Arabic sample page

[5] A. K. Singh and C. Jawahar, "Can rnns reliably separate script and language at word and line level?" In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, IEEE, 2015, pp. 976–980.

[6] A. Ul-Hasan, M. Z. Afzal, F. Shafait, M. Liwicki, and T. M. Breuel, "A sequence learning approach for multiple script identification," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, IEEE, 2015, pp. 1046–1050.

[7] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, ACM, 2006, pp. 369–376.