An Investigative Analysis of Different LSTM Libraries for Supervised and Unsupervised Architectures of OCR Training

Abstract—Optical Character Recognition (OCR) involves conversion of images of text into machine encoded editable text. Despite the wide research advancements in the field of OCR systems, the recognition capability of OCR systems on unseen or degraded historical documents is still questionable. The degradations in the document like torn pages, ink spread and blurred documents are major challenges especially in the old paper documents. Most of such degraded documents lack a generalized and reliable OCR system mainly because of the unavailability of ground-truth data and poor generalization capabilities of the OCR systems. Also manually transcribing the documents is a cumbersome task which also require certain language-specific expertise. This paper presents a feasibility study of different OCR architectures together with different preprocessing stages for a reliable OCR on such challenging documents. To this end, we evaluate various OCR settings on a dataset containing highly degraded historical German typewriter documents. This paper investigates various key aspects of OCR training such as the impact of incorporation of different LSTM libraries, grayscale or binarized data for training and training data size used on the subject dataset. In addition, difference in the effect of using completely manually transcribed data as compared to semi-corrected ground-truth data for anyOCR architecture of unsupervised OCR training have been analyzed on a small dataset. The anyOCR framework has shown promising results as an efficient OCR system which was evident with its comparison with other OCR systems. The various factors analyzed provided a feasible strategy for approaching the problem and evaluating highly challenging historical documents.

Index Terms—component, formatting, style, styling, insert

I. INTRODUCTION

Preserving and revisiting historical literatures is very important to understanding our roots. Most of these literatures are available as books and paper documents. The life-span and quality of these paper documents degrade over time. So there is a need for converting them into digital formats. In order that these documents can be easily accessible, it needs to be converted into editable text format.

Document Image Analysis (DIA) is the complete pipeline of transforming paper documents into the digital text format[1]. The core module in a DIA system is the optical character recognition (OCR)[2] module which is mainly responsible for text line recognition. OCR basically refers to the conversion of images of text into machine encoded editable text [2].

With advances in technology, OCR algorithms are now used for text recognition of far more complex and challenging scripts. A type of neural network called as Recurrent Neural Network (RNN) has been successfully applied to OCR, thanks to its ability to process sequential data better. RNNs essentially learns functions that map the entire history of previous inputs to each output [3]. However the range of context that can be accessed is limited as it suffers from the issue of vanishing gradient [11]. In order to overcome this issue, Long Short Term Memory(LSTM) is used. LSTM is a special type of RNN with multiplicative gates and feedback loops [4].

This paper investigates the supervised OCR training and unsupervised anyOCR training architectures with two different LSTM libraries - MDLSTM and BLSTM - for historical document images. Another aspect investigated was the use of either grayscale or binarized text lines images for training the LSTM architecture and examined if there is any significant difference in the recognized outputs. Likewise the quantity of training data used as input for the training does have effect on the sequence and context learning capabilities of the LSTM.

The rest of the paper is organized as follows. Section II briefly describes the recent OCR training architectures for sequence based learning. The dataset used in this paper and its preparation for training for different OCR architectures are defined in Section III under methodology. Section IV presents the performance evaluation results, which follows by conclusion in Section V.

II. RECENT OCR ARCHITECTURES

A. Methodologies in OCR

1) Segmentation-based OCR: Segmentation based OCR approach involves extraction of individual characters from text. Tesseract [5] is an example of an open source OCR which is based on this approach. Segmentation-based OCR systems can be broadly classified into two categories. The first one is based on template matching. The second class is based on over segmentation.

2) Segmentation-free OCR: Segmentation-free OCR approach [6] involves localization and recognition of characters at the text line level using sequence learning approach. This in turn helps to include contextual information.

B. Sequence Learning using LSTM

The LSTM architecture consists of a set of cyclically connected memory blocks. Each block contains one or more self-connected memory cells and three multiplicative gates the input, output and forget gates. The information can be stored and accessed for longer durations of time in the LSTM
memory cell with the help of these multiplicative gates [4]. This thereby avoids the vanishing gradient problem [7].

1) Bidirectional LSTM Networks (BLSTM): In order to have access to the future context BLSTM [7] networks are used. This is particularly used in the case of off-line data processing where the whole data sequence is available prior to processing. The data sequence is processed and scanned in two directions from left to right in the forward direction and also from right to left in the reverse direction. For this purpose two identical hidden layers are placed in the network which are processed in forward and backward direction. Both of these hidden layers are then connected to a single output layer to produce the output sequence.

2) One Dimensional LSTM (1D-LSTM): In a 1D-LSTM architecture [8], a window of width 1-pixel and height equal to the height of the image traverses through the text line image. This transforms the input sequence into a one-dimensional sequence. The height of the image is referred to as the depth of the 1D sequence and each column of the image thus obtained is called a frame. The resulting single-column frames are fed into the input layer of the LSTM network.

3) Multidimensional LSTM (MDLSTM): Multi-Dimensional LSTM (MDLSTM) [9] are networks capable of processing image along multiple dimensions. It usually scans the image from the four corners of the image. So in MDLSTM network with dimension n, each LSTM cell would have n self-connections and n forget gates. Hence at each time-step the network has reference to its contexts in all the n dimensions.

C. OCR Architectures

1) OCRopus: OCRopus [10] is an open source OCR framework which incorporates several OCR components. This is a complete framework for performing major operational steps in DIA. It has been designed to recognize multi-lingual and multi-script text. The architecture of the OCRopus system is depicted in Figure 2. It is comprised of four major components: preprocessing, layout analysis, text line recognition, and statistical language modeling. The OCRopus OCR training architecture is based on BLSTM model.

2) Tesseract: Tesseract [5] is an open source OCR engine that was developed by the HP research lab. In the architecture for Tesseract OCR system, firstly the connected components of the characters are analyzed. The outlines are gathered and stored together to form blobs. These blobs are then used to identify the text lines by using fixed pitch or proportional word algorithms. The text lines are segmented into words based on character spacing. This is done by fitting a base line and identifying the pitch or spacing of each character.

3) anyOCR with Tesseract: anyOCR [12] is a sequence learning OCR system. This is a framework for combining the segmentation based OCR approach and segmentation free OCR approach. This framework is used when the ground-truth data is not available or is minimally available. In this setting, a segmentation-based OCR is used to create the semi-corrected ground-truth data which is then used to train a segmentation-free OCR [12]. The anyOCR with Tesseract training architecture is shown in Figure 2.

4) anyOCR with Clustering: In this pipeline instead of using the Tesseract, a completely unsupervised clustering algorithm [14] is used as the segmentation based OCR. The pipeline of the architecture is as follows: the text lines are extracted [11] from the scanned document image. Then each text line is segmented into characters. The individual characters are stored as images. Then features from these character images are extracted. After this the characters are clustered into character clusters based on an unsupervised clustering algorithm knowns iterative k-means clustering algorithm [14]. After several iterations of character clustering performed, an average image of all characters in a cluster is obtained. If all the characters in a cluster are same then average image would be an average image of that character. However if the characters are different in a cluster then image would appear more blurred. The blurriness of an average image is measured and if its too high, then its reclustered. Next, a
anyOCR-Clustering Training Architecture: First text lines and unique symbols are extracted from the scanned data. These symbols are then clustered. After a language expert has annotated the resulting clusters, semi-correct ground-truth data for each text line is generated. In a second phase a LSTM-based OCR model is trained using the clustering output (using the OCRopus utility and RNNLIB architecture). The trained model can then be used to generate better ground-truth data iteratively.

language expert should manually identify and annotate each integral image using uni-codes of the characters [14]. Based on the uni-codes annotated and positional information of the character the corresponding output text files are obtained. These output text files serve as the semi-correct ground-truth data for the segmentation based OCR. The text line images and corresponding semi-corrected ground-truth are now used to train the LSTM model. The trained LSTM model produces the corrected OCR output. This can further be used again to generate the ground-truth which is more correct. This iterative procedure can be continued until reasonable improvement in the ground-truth data is obtained. The anyOCR with Clustering training architecture is shown in Figure 3.

For both anyOCR-Tesseract and anyOCR-Clustering training architectures, this paper makes use of two LSTM models: One is the OCRopus LSTM library which is a BLSTM [7] and the other LSTM library used is MDLSTM [9, 13] from RNNLIB architecture. The RNNLIB architecture includes both BLSTM and MDLSTM. The trained LSTM models can then be used again to predict the labels of the training data set. In this way the newly predicted labels in the second iteration can now be used as the new semi-corrected ground-truth data [12]. After enough improvement to the ground-truth data is achieved, this can be trained using a LSTM model. The trained LSTM model can now be used as the final OCR system for evaluating the test corpus.

III. METHODOLOGY

A. Dataset

1) Archiv der DDR-Opposition - Robert-Havemann-Gesellschaft 1950s - 1980s: This dataset consists of scanned copies of German document images with old German typewriter font. The scanned documents contain blurred and distorted text, strokes along the text and missing characters which cause major challenges to processing and recognizing the text. Some of the samples document images and cropped text line images are shown in Figure 4 and Figure 5, respectively. The training dataset prepared for training of OCR system includes a small dataset with 1000 text line images with manually transcribed ground-truth (referred here as TRAIN-1000) and a comparatively larger dataset of 10000 text line images without ground-truth (referred here as TRAIN-10000). A test set, along with target labels which was manually transcribed, of 500 text line images was used to evaluate the recognition accuracy of the trained OCR system (referred here as TEST-500).

2) Synthetic Dataset: The synthetic dataset is created by crawling the German news websites online. Around 20,000 text lines of data was crawled and saved as a text file. First the ttf-type fonts of Archiv der DDR-Opposition is forged. Afterwards, a utility from OCRopus is used to generate the synthetic data image set. The utility requires utf-8-encoded text-lines to generate the corresponding text-line images along with the forged ttf-type font files. The user can specify the parameter values or use the default values. There are many parameters that can be altered to make the synthetically generated text-line images to be closely resembling the scanned documents.

B. Data preparation using anyOCR-Tesseract pipeline

1) Data Preprocessing: An OCRopus utility is used to perform binarization. This is followed by the layout analysis stage where the layout of the binarized image is identified using the RAST layout analysis. An OCRopus utility is used for performing segmentation. The text lines are then saved in
2.1.4 Data preparation for RNNLIB

1) Data Preprocessing: AnyOCR-Clustering pipeline [14] follows similar steps as in the previously explained anyOCR-Tesseract pipeline. Unlike the former pipeline, current training methodology follows character-based clustering as the alternative segmentation based OCR approach.

C. Data preparation using anyOCR-Clustering pipeline

1) Data Preprocessing: Data preprocessing for anyOCR-Clustering pipeline [14] follows similar steps as in the previously explained anyOCR-Tesseract pipeline. Unlike the former pipeline, current training methodology follows character-based clustering as the alternative segmentation based OCR approach.

D. Data preparation for RNNLIB

1) RNLLIB Framework: RNLLIB [16] is a recurrent neural network library which are mainly used for sequence learning problems like handwriting and speech recognition. The data file format for all RNLLIB data files used for training, testing and validation needs to be in NETCDF format. NETCDF format is a binary file format which is mainly designed for handling large scientific datasets.

1) Data Preprocessing: The preprocessed text-line images obtained after performing binarization and text line segmentation are used as the training dataset for the RNLLIB LSTM architecture. The semi-corrected ground-truth for the text line images are acquired by using segmentation-based OCR system like Tesseract OCR and another by using the iterative k-means clustering. The erroneous outputs produced from these systems are used as the semi-correct ground-truth data for training the MDLSTM architecture using the RNLLIB deep learning framework.

3) Multi-Dimensional LSTM (MDLSTM): Multi-Dimensional LSTM (MDLSTM) [22, 30] are networks capable of processing image along multiple dimensions. It usually scans the image from the four corners of the image. This makes the architecture rotation and scale invariant. The architecture of the MDLSTM used in this thesis is shown below in Figure 6.

E. Training the OCR System

1) Semi-Corrected Data: Semi-corrected Dataset refers to the dataset which consists of erroneous ground-truth data obtained through segmentation based OCR in anyOCR pipeline. The training of this dataset is done using BLSTM architecture in the OCRopus and also using MDLSTM architecture in the RNLLIB framework. The tunable parameters in the network are set based on values mentioned in the previous section. After training the OCR model, the trained LSTM model is then used again to provide a second iteration of more correct ground-truth data. This iterative procedure can be continued for any number of iterations until the correctness of the ground-truth data to be trained for OCR system is improved considerably. RNLLIB MDLSTM is another LSTM architecture that is used to train the OCR system. The training dataset is comprised of 3 categories - one is a big dataset with binary images using anyOCR-Tesseract pipeline, another is a big dataset with grayscale image using anyOCR-Tesseract pipeline. These training dataset with corresponding erroneous ground-truth data are trained using the MDLSTM architecture. The training datasets and test dataset are converted into NETCDF format by saving them as .nc files. The system parameters used for the MDLSTM architecture are described in detail in the previous section. The obtained best model with respect to CTC- error and best label error can be applied again over the training set to achieve second iteration of improved ground-truth data. This can be continued iteratively until the overall recognition error rate is reduced considerably.

2) Fully-Labeled Data: Fully labeled dataset refers to the dataset comprising of ground-truth that is fully manually labeled (i.e. TRAIN-1000). The dataset comprises of 1000 text line images which are binarized. The training dataset
comprises of text line images from the German document corpus. This dataset is considered in order to understand the impact of the number of training examples in the efficiency of training. The text line training dataset images and their corresponding fully labeled ground-truth data are used for training the LSTM network. Here we have used the BLSTM architecture of OCRopus and MDLSTM architecture using the RNNLIB framework. The obtained OCR models are used to test the test data set comprising of 500 text line images. The synthetic data obtained from the German news websites were also trained using the OCRopus utility and obtained the corresponding model outputs.

IV. PERFORMANCE EVALUATION

A. Evaluation Metric

The evaluative performance of a trained OCR system is described in terms of Character Error Rate (CER). It is measured as an Levenshtein edit distance between output string sequence and the corresponding target string sequence. It can be defined as the ratio of the sum of insertions, deletions and substitutions to the total number of characters in the target output. It can be found as:

\[
\text{CER} = \frac{I + D + S}{\text{Total number of characters in the text line}} \times 100% \tag{1}
\]

where, \( I \), \( D \) and \( S \) are the number of insertions, deletions and substitutions, respectively.

B. Semi Corrected Data Evaluation Results

In the anyOCR-Tesseract pipeline, initially the trained German model of Tesseract OCR is used to produce the corresponding text output. Tesseract OCR yields a CER of 23.5% on the test dataset TEST-500. The output produced by Tesseract is then used as the semi-corrected ground-truth data for training the LSTM architectures. So this thus means that the LSTM based OCR systems have been trained with 23.5% erroneous ground-truth data on the TRAIN-10000 dataset. Two LSTM architectures have been used - BLSTM architecture from OCRopus framework and MDLSTM architecture using RNNLIB framework. During the first iteration, the trained BLSTM model yields a CER of 15.4% on the TEST-500 dataset while MDLSTM model yields a CER of 16.4% as can be seen in Table I. So, the trained MDLSTM and BLSTM models improves the test errors from 23.5% to 15.4% (a relative improvement of 35.0%) and 16.4% (a relative improvement of 30.2%), respectively.

During the second iteration, these trained LSTM models are used to produce improved ground-truth data. Now again the LSTM models are trained using the improved ground-truth data. The models thus trained gives a CER of 13.3% (a relative improvement of 13.6%) with MDLSTM based OCR model and a CER of 14.02% (a relative improvement of 14.5%) with BLSTM based OCR model. This iterative improvement of ground-truth can be continued but we have stopped at this stage. A pre-trained model of the OCRopus have been used to evaluate the test dataset and was found to achieve a CER of 22.5%. This was used in order to obtain a rough comparison of the improvement in the used models. The results of the four evaluations are listed in Table I.

Another aspect that was evaluated during the course of the paper is the impact of binarization of the image on the OCR recognition accuracy. For this purpose two training datasets were used. One set consisting of 10000 binary text line images and another consisting of the same 10000 gray-scale text line images. Initially the trained Tesseract OCR model is used to corresponding output text for the binary and grayscale images. It yields a CER of 23.5% for binary images and 28.9% for gray-scale images. These erroneous ground-truth data were used to train the MDLSTM from the RNNLIB framework. The trained LSTM models yielded a CER of 15.4% (a relative improvement 35%) for binary images and a CER of 17.3% (a relative improvement 41.0%) for grayscale images. The results of evaluation are provided in II.

C. Fully Labeled Data Evaluation Results

This section reports the results obtained by evaluating the performance of an OCR model trained on a small dataset comprising of 1000 text line images. Unlike the previous section, ground-truth data for the training data files are completely manually labeled. This is used to train the LSTM model. BLSTM and MDLSTM are the two LSTM architectures used to train the OCR system. The test set consists of 500 text line images which is the same test set used in the previous section. The MDLSTM and BLSTM yields a CER of 23.5% and 20.02% on the fully balled data. In order to have a comparison
between the recognition accuracy difference between fully labeled and semi-corrected data, semi-corrected ground-truth data are created for the same set of 1000 text line images using the anyOCR Tesseract pipeline. Trained LSTM models using semi-corrected ground-truth yields a CER of 28.9% for MDLSTM model and a CER of 25.3% for BLSTM model. The evaluation results can be seen in Table III.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Semi-Corrected GT</th>
<th>Fully labeled GT</th>
</tr>
</thead>
<tbody>
<tr>
<td>(RNNLIB-MDLSTM)</td>
<td>28.9%</td>
<td>23.5%</td>
</tr>
<tr>
<td>(OCRopus-BlSTM)</td>
<td>25.3%</td>
<td>20.02%</td>
</tr>
<tr>
<td>Tesseract</td>
<td></td>
<td>23.5%</td>
</tr>
<tr>
<td>OCRopus-pre-trainedmodel</td>
<td></td>
<td>22.5%</td>
</tr>
<tr>
<td>ABBYY</td>
<td>22.2%</td>
<td></td>
</tr>
</tbody>
</table>

**TABLE III**

| Comparison between anyOCR-Tesseract, anyOCR-Clustering, Tesseract, OCRopus-pre-trained-model and ABBYY. GT denotes ground-truth. The results obtained by the OCRopus-BlSTM yielded the best results, but notice that this system was trained with the correct GT information, while the anyOCR framework is trained with semi-correct GT information. However the difference in the CER(%) obtained on TEST-500 dataset is not very significant.

In order to understand the progression of training errors, initially trained Tesseract OCR was used to produce the corresponding output text files for the training dataset TRAIN-1000. It yielded a CER of 9.2% for Tesseract on the training dataset TRAIN-1000. Training of the LSTM models yields CER of 5.3% and 6.2%, for BLST and MDLSTM respectively, on the training dataset when using semi-corrected ground-truth data and CER of 2.99% and 3.5%, for BLST and MDLSTM respectively, on the training dataset when using fully labeled ground-truth data. The evaluation results are presented in Table IV.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Semi-Corrected GT</th>
<th>Fully labeled GT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tesseract</td>
<td>9.05%</td>
<td>-</td>
</tr>
<tr>
<td>(OCRopus-BlSTM)</td>
<td>5.3%</td>
<td>2.99%</td>
</tr>
<tr>
<td>(RNNLIB-MDLSTM)</td>
<td>6.2%</td>
<td>3.5%</td>
</tr>
</tbody>
</table>

**TABLE IV**

**ILLUSTRATING THE IMPROVEMENT IN TRAINING ERRORS (CER(%)) USING THE TRAIN-1000 DATASET AFTER USING THE ANYOCR FRAMEWORK. INITIAL SYSTEM WAS TRAINED ON ‘TRAIN-1000’ DATASET WITH 9.05% ERRONEOUS GROUND-TRUTH DATA.**

V. Conclusion

In this work we investigates the supervised OCR training and unsupervised anyOCR training frameworks with two different LSTM architectures - MDLSTM and BLSTM - for historical document images. The sequence learning capacity of the LSTM architectures was studied with respect to the very challenging old German document dataset. The anyOCR framework was shown to produce quite promising results on the studied dataset. The effectiveness of one LSTM architecture over the other when applied together with the various preprocessing stages of the anyOCR pipeline was analyzed and it was evident that there is no significant difference between the results obtained when using MDLSTM and BLSTM for training. The difference was a mere 1 percent which is rather insignificant to overall recognition capability of the OCR system. Another investigated aspect was the use of either grayscale and binarized text line images for training the LSTM architecture. It is observed that both the settings resulted in a significant CER improvements with the training module in anyOCR framework. However there was no significant advantage noticeable when using either formats of image data over the other while training. This is particularly conceivable for the dataset considered.

**REFERENCES**