Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 35–44

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

Offline HWR Accuracy Enhancement with Image Enhancement and Deep Learning Techniques

Aashu Kumar* Peeta Basa Pati

Department of Computer Science and Engineering, Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham, India

Abstract

Handwriting recognition (HWR) is the ability of a machine to recognize handwritten text present in images, scanned or photographed. While much of the work reported in this field deals with scanned images, recognizing handwritten content from camera-captured images is still a challenge. While researchers are focusing their efforts to improve on the recognition process of such characters, the current work focuses efforts outside the recognition process.

This work employs image enhancement techniques, noise removal, and binarization with adaptive thresholding & segmentation, to increase HWR accuracy. Additionally, the work of employs deep learning-based language models, such as BERT & BART, to further enhance the accuracy of recognized text. It is observed that HWR accuracy increases when smaller images such as word images are used as input to the HWR engine. On the post-processing front, BART is demonstrated to be superior in enhancing the accuracy of HWR-recognized text documents in a comparative study presented in this work. An accuracy improvement of 20+% is achieved with the BART model enhanced with transfer learning through a localized dataset to handle HWR-specific error patterns.

Keywords: Handwriting Recognition, Optical Character Recognition, Camera Captured Document Images, Language Models, Bidirectional Encoder Representation from Transformer (BERT), Bidirectional autoregressive transformers (BART).

1. Introduction

The goal of Handwriting recognition (HWR) technology is to recognize and digitize the handwritten text present in scanned or photographed images of paper. HWR a subset of character recognition technology that is used for content

*Corresponding Author.
Email Address: aashukumar0602@gmail.com

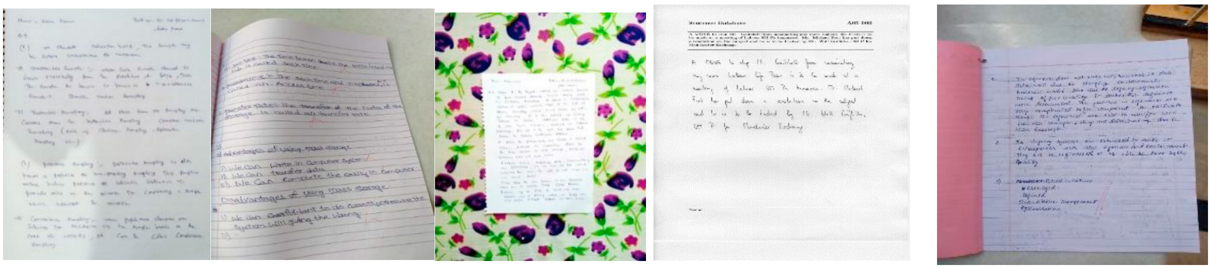
10.1016/j.procs.2022.12.399

digitalization from paper documents and digital slates. This digitalized text can subsequently be used for various downstream applications.

There are two types of HWR: online and offline. Online HWR deals with the recognition of handwritten text created on a computer, smartphone, or similar electronic devices. On the other hand, offline HWR deals with the recognition of handwritten text in scanned or camera captured documents. Offline handwritten character recognition is comparatively more difficult due to the variability of writing & handwriting quality of the writer. Offline HWR is also called as Intelligent Character Recognition (ICR) and these terminologies are synonymously used.

Numerous works have been carried out in ICR field. While a few of these works deal with enhancing the quality of input images, most researchers have focused on developing algorithms to handle the variability associated with the handwritten characters. These research outputs have been converted to consumable technologies as products and are available for general usage. However, these technologies assume imposition of various constraints in the quality of image acquisition such as imaging DPI, device of scanning or/and quality of paper used for writing.

Due to wide availability of low-cost devices such as smart phones, everyone has easy access to camera. Compared to scanner, camera is easy to use and handle and, therefore, camera has become a preferred mode of image acquisition. Document images acquired with camera contain a variety of distortions such as illumination variation, complex boundary, out of focus images, out of page objects and surface distortion (cylindrical, spherical and their complex combinations). Fig. 1 demonstrates these distortions with a few examples. Pre-processing of these images is employed to enhance their quality & improve the recognition accuracy. However, due to presence of a wide variety of such distortions, the output quality of a recognition engine is adversely affected and results in less than 100% accurate text.



(a) (b) (c) (d) (e)

Fig. 1: Examples of camera captured documents having different distortions: (a) illuminations variations, (b) uneven page surface leading to geometrical distortion, (c) out of page object images, (d) Samples of handwritten input documents obtained from IAM dataset, (e) AVV-220 dataset. IAM dataset contains scanned document from a flatbed scanner while AVV-220 images are taken with a handheld camera in a smartphone.

Research has already been carried out to effectively employ language models in correcting inaccuracies present in the text document. This work employs Natural Language Processing (NLP) based language models to enhance the text accuracy. Thus, this work is employs various pre-processing technique to enhance the input image quality to enhance the HWR accuracy. Further boost to the recognized text accuracy is provided by utilizing deep learning-based language models. In the remainder of this paper, Sec. 2 contains details on the prior work & Sec. 3 presents the system description. The results obtained from the experiments is discussed in section 4 while Sec. 5 concludes the study.

2. Literature Survey

Plamondon and Srihari [1] have proposed a method for automatic detection of handwritten text. In this method they have used pen- paper metaphor that mimics the process of electronic ink. To detect the shape and structure of the characters, they have employed self-organized feature maps (SOFM) technique. This technique depends on the structural rule-based models. Yang et al. [2] has proposed a long-short term methodology [LSTM] to identify the error type such as spelling or grammatical errors and its position in the text sentences. In this paper, the author has compared the performance of attention mechanism (encoder and decoder) against unidirectional recurrent neural network for correcting the errors present in a text document. They demonstrate that attention-based models perform better in correcting textual errors. Tan et al. [3] trained a nested Recurrent Neural Network (RNN) model with a large pseudo dataset generated with phonetic similarity. Hang et al. [4] have used a LSTM and RNN technique for predicting correct characters in an output text. In their work, they have shown that both LSTM and RNN models outperform the traditional lexicon-based algorithms. Carbonel & Anquitel [5] have proposed the idea of lexical approaches for handling text correction. They have used a string-matching algorithm with Edit distance metric for lexical post processing.

Wang et al. [6] have used sequence to sequence model with a replication mechanism for spelling correction. These methods require a large, labelled dataset for training. The limitation of this method is suboptimal performance when a small training dataset is used. Chen and Zesong [7] used a sequence-to-sequence model to handle grammatical error in the sentences. But the drawback with this model is that: it needs a lot of data to train and to enhance the performance of the error encoder. The spelling error dataset which are collected from real life scenarios are small in size. While creating synthetic dataset it may happen that the synthetically introduced errors are different from how human writers make mistake. An attempt has recently been made by Didenko and Shaptala [8] to develop a pre-trained bidirectional encoder representation from transformer (BERT) model. This model uses a GEC approach with a fully connected network to identify the errors present in the input sentence and subsequently correct the textual mistakes. In contrast, this model does not take into account the relationship between the errors and their types when determining the type of error. They demonstrate that the model has difficulties in detecting the errors and does not performed masking properly. Hu et al. [9] have proposed an improvised BERT model i.e., character-phonetic BERT model. An error correction and detection network based on BERT is represented by this model. They have used a detection model to predict an error probability of each word in the sentence. After that the error correction model has been used to replace error with correct word. They demonstrate that improvised BERT performance is better than simple BERT model.

For optical character recognition (OCR) post-processing, modified version of Viterbi Algorithm has been used for Stochastic Error Correction by Amengual and Vidal [10]. Cortes et al. [11] have proposed stochastic error correction for OCR post-processing. They have implemented a finite state algorithm to build an inference engine that identifies the grammatical mistakes in the sentences.

Thus, the surveyed literature may be divided to cover the following aspects: (i) image enhancement with corrections to distortions, (ii) recognition of characters (printed & handwritten), (iii) text spelling or grammar correction employing look-up or deep learning-based language models, and (iv) post-processing to recognized text to enhance the text output accuracy. The image enhancement methods employ various metrics to evaluate their efficacy, but none deal with recognition accuracy as a measure. Most text correction algorithms assume the text available as typed text with noise introduced at acquisition stage or later. The limited work dealing with OCR post-processing handle printed character documents only. To the best of knowledge of the authors, there is hardly any work that studies and reports impact of image enhancement techniques on recognition accuracy of camera captured handwritten documents. Similarly, the authors were unable to identify a research work which employed deep learning-based language models as postprocessing technique to increase accuracy of handwritten text recognition.

This work takes a hybrid approach to enhance the quality of recognition output from camera captured handwritten document images. Various pre-processing techniques such as binarization, noise removal, page layout segmentation and boundary detection are employed to enhance the input image quality. Additionally, post-processing techniques involving NLP based language models technique such as BERT and BART are applied to enhance the recognize text accuracy.

3. System Description

In proposed system, the images are enhanced in the pre-processing module before character recognition. This is employed to gain higher character recognition accuracy. The output text from the recognition engine is subsequently subjected to various post-processing steps to bring in further accuracy enhancement. Thus, the proposed system is a three-step process (refer Fig. 2): (i) pre-processing, (ii) character recognition with a 3rd party ICR engine, and (iii) post-processing for text accuracy improvement. Google read API is used as third party ICR engine used in this work.

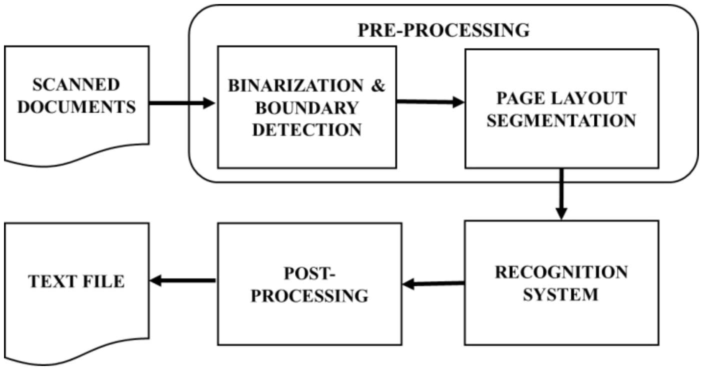


Fig. 2. Schematic diagram illustrating the flow of data through the 3-step proposed system.

The pre-processing steps include binarization, noise removal, boundary detection and page layout segmentation. The input to the pre-processing module is a document image & output is a segmented and binarized image. The binarized page image (output of pre-processing) is utilized by the recognition engine for character recognition. To handle various kinds of errors, present in recognized text, post-processing techniques are employed. These post-processing techniques rely on deep learning-based language models such as BERT & BART. The following subsections describe the datasets & the constituent modules employed in the proposed system and its experimentation.

3.1. Data Description

For the image pre-processing work, two separate datasets, namely, IAM dataset & AVV-40 dataset, are employed. The IAM dataset [12] consists of 529 images scanned at 300 DPI and stored in 8-bit grayscale format. The handwritten pages were collected from 529 people covering a wide variety of writing styles. Fig. 1. (d) shows an example IAM database image. It may be observed that the scanned image contains four distinct blocks: (1) header, (2) typewritten, (3) handwritten, and (4) footer. Two blocks are separated from each other by a horizontal line. AVV-40 dataset consists of 40 camera captured documents collected from answer scripts Amrita Vishwa Vidyapeetham (AVV) undergraduate students. These images are anonymized for student identity. All the images in AVV-40 are captured using the 12 mega pixel rear camera available in Motorola Moto-G5-Plus smart phone without any special constraints imposed. Fig. 1. (e) shows a sample AVV document image. Both datasets have images available in JPEG & PNG formats.

For post-processing module, four separate datasets are employed. One of the datasets is text dataset and two of the datasets are derived from the IAM dataset i.e., named as block and word dataset while the fourth one is derived from AVV-40 dataset. These four datasets are described below.

- C4_3M dataset – consists of 3 million pairs of incorrect and equivalent correct sentences [13]. This dataset is sourced from a public site and used as is without any refinement to the content. It is assumed that the control data, available as correct sentences, are manually verified for their accuracy.
- IAM Block dataset – consists of recognized text obtained from the IAM dataset images. The handwritten image block is recognized using Google Read service and control data is created by manual inspection & editing of these recognized texts.
- IAM Word dataset – consists of recognized text obtained from the IAM dataset images. The handwritten image block is segmented to words & recognized using Google Read service. Once the recognition is

completed, the text words are combined to form the text for the image block.

- AVV-40 dataset – consists of recognized text from the page images of AVV-220 dataset. Control data for these pages is created by manual inspection & editing of the recognized text.

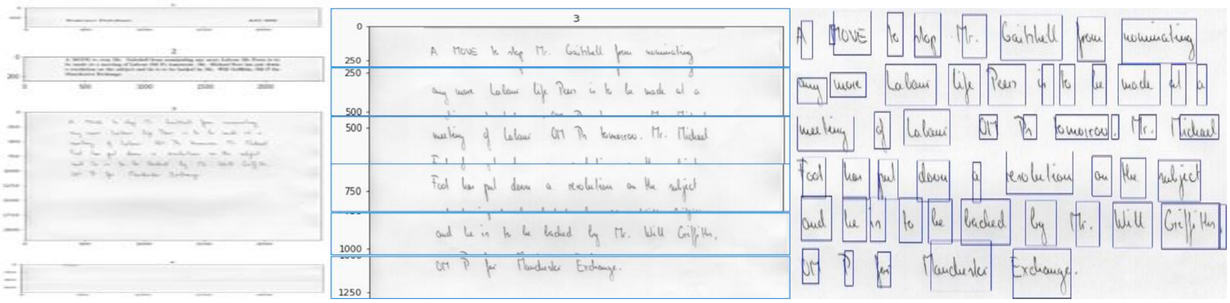
3.2. Pre-processing

Before passing the scanned documents into the recognition system, it is necessary to perform several pre-processing operations on noisy and distorted original documents. This pre-processing filters out the impurity from the document images. Binarization, noise removal, boundary detection and layout segmentation are some of the steps performed in the pre-processing module of the proposed system.

Images from IAM dataset (refer Fig. 1(d)) are scanned images; hence, boundary detection is not needed for them. A binarized image is generated as an output from the pre-processing block. The dividing of an image into several regions is called segmentation. The main benefit of segmentation is to breakdown the image into constituent parts for easier analysis. This work employs segmentation techniques to separate the printed block from handwritten text portion.

To study the effectiveness of these preprocessing steps, each scanned image from IAM dataset is segmented to 3 different levels. These segmented images are recognized with Google Read Engine and the efficiency of such recognition is evaluated. The results associated with this study is present in Table 1 in Results section. The three different levels of segmentation are: (i) block segmentation, (ii) line segmentation, and (iii) word segmentation. In the block segmentation, the page is divided into constituent text blocks. The block containing handwritten text, which is the block of interest for this study, is picked and further divided into lines of text with line segmentation technique. Each obtained line image is further divided into word images as word segmentation mechanism.

In this work only handwritten portion (block 3) of the IAM dataset images is important. This portion is extracted with block segmentation. Since the entire content of the documents in AVV-40 is handwritten, segmentation step is not carried out here.



(a) (b) (c)

Fig. 3: (a) output of page layout segmentation technique employed on original image shown in Fig. 1. (d). The original image contains printed text as well as handwritten text. Block 1 contain heading and serial no. of the scanned documents, block 2 and block 3 contains printed and handwritten texts, respectively, while block 4 contain signature of the writer, (b) Output of line segmentation technique employed on block 3 of page layout segmentation present in Fig. 3. (a), (c) output of word segmentation technique employed after line segmentation. The line segmentation technique has been employed on block 3 of page layout segmentation available in Fig.3. (a) The results of line and word segmentation has been merged and boundaries of segmented words are displayed on the block image.

3.2.1. Block segmentation

A closer study of the images from IAM dataset reveals that each image is separated into blocks by a horizontal line. This step exploits this structural feature to divide the image into blocks. The horizontal lines between sections are detected, by employing horizontal projection profile analysis, in the input binarized image to break it into segments.

Once the image is divided into blocks, the block 3 which is region of interest is picked and used for further processing. Fig. 3. (a) illustrates the working of this module with a sample input image and its output.

3.2.2. Line and Word Segmentation

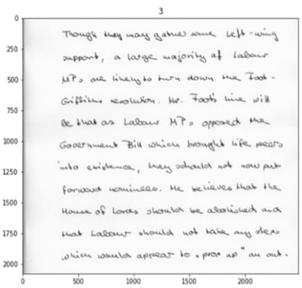
Line Segmentation is breaking up of handwritten document images into its constituent lines. Line segmentation has been performed by using horizontal projection. The binarized output image from block segmentation is taken as input for this step.

Word Segmentation breaks each line image of text into its constituent word images. The output of the line segmentation is used as input to this step where words are segmented based on vertical projection profile. Morphological operations are applied to connect the broken characters and help obtain a clean vertical projection profile for the segmentation of words to be obtained. A threshold is kept distinguishing between intercharacter spacing and the inter word spacing. The segmented line and word image is shown in Fig. 3. (b) and Fig. 3. (c) respectively.

3.3 Postprocessing

The previous section dealt with various pre-processing techniques employed in this work. Using these pre-processing techniques, a noisy input image is pre-processed, and an enhanced output image is generated. This enhanced output is used by the character recognition engine which recognizes the characters present in the document and generates a text string as output. Various factors adversely affect the accuracy of recognition, thus, resulting in less than 100% accurate output text. This section deals with various language specific deep learning NLP models employed in this work. Using these techniques, erroneous text, obtained from OCR engine as output, is enhanced to accurate text. Fig. 4. (a) presents example input handwritten document, its corresponding recognized text output (Fig. 4. (b)) as well as the ground truth or true data in fig. 4. (c).

Natural language models have been effectively used to discover error patterns present in text documents. These language models may be utilized to bring in correction to such erroneous text obtained from character recognition engines for input handwritten image blocks [14]. In this work, language models based on NLP technique are employed as post-processing steps to enhance an accuracy of a text obtained from OCR engine.



(a)

Through they may gather some Left-wing support,
a large majority of Labour are likely to
tuon down the Foot-Griffithe resolution. M
Fot line will be that as The Labour MPs opps od
the Goreenment nt bill which bought life peers Toot's
existence,they should not pot forward nomn ees.
he believes inees that the House of Lords should be
abolished and that Labour should not take ana any
appear to '' out-dated institution.

(b)

Though they may gather some Left-wing support,
a large majority of Labour 0M Ps are likely to
turn down the Foot-Griffiths resolution. Mr.
Foot's line will be that as Labour 0M Ps opposed
the Government Bill which brought life peers into
existence,they should not now put forward nominees.
He believes that the House of Lords should be
abolished and that Labour should not take any
steps which would appear to 'prop up" an out-dated
institution.

(c)

Fig. 4: (a) Sample segmented handwritten image block used as input to the character recognition engine, (b) the recognized text output from the engine, and (c) ground truth text corresponding to the segmented handwritten block shown in (a).

3.3.1. Bidirectional Encoder Representations from Transformers (BERT)

BERT is an NLP framework based on transformer technology. Transformers makes use of deep learning technology [15]. The BERT architecture is composed of an encoder-decoder network, by using a self-attention on encoder side and attention on the decoder side. BERT consists of three different kinds of embeddings such as Position Embedding, Segment Embeddings and Token Embeddings (refer Fig. 5). Position Embeddings is used to find the positional information of a word in the sentence. Segment Embeddings helps BERT to take multiple sentences as input. Token embeddings are learned from the Word Piece token vocabulary.

Decoder part of BERT model consists of three sublayers, namely, feed forward, encoder-decoder-attention and self-attention [16]. The feed forward network takes the representation as input and return the probability of all the words corresponding to mask token. The word which has highest probability will be the correct word in place of mask token. The encoder part predicts mask tokens for erroneous words in a sentence while the decoder part suggests potential corrections with a probability score. Token is subjected to a series of pre-processing activities such as, lower-case conversion, punctuation removal, and stop words removal. The Sentence Tokens, after pre-processing, are converted into a specific input format that is needed by BERT model. These modified Sentence Tokens are stored in an array and fed to the BERT model for encoding [17].

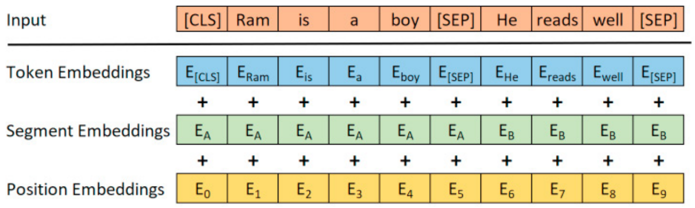


Fig. 5- Structural Breakdown of BERT input sequence.

3.3.2. Bidirectional autoregressive transformers (BART) Denoising Sequence to Sequence Model

BART is also a transformer-based model consisting of an encoder and a decoder block [18]. Pretrained BART model is taken and fine-tuned with two datasets (described below in this section). It is a denoising sequence-to-sequence based architecture. In this work two different BART models are used. First one is base BART model which is called generic BART. The second one is a fine-tunned BART model & is called as enhanced BART. Additionally, Marian-MT also known as mBART, is used in this work.

BART uses a neural machine translation (NMT) architecture which is based on a Transformer-based model. It uses a standard seq2seq/NMT with a bidirectional encoder (like BERT) and a left-to-right decoder (like GPT). This means the encoder's attention mask is fully visible, like BERT, and the decoder's attention mask is causal, like Generative Pre-trained Transformer (GPT only uses the decoder). In GPT tokens are predicted by seeing the leftward context only, so GPT cannot learn bidirectional representation [19].

In this work fine tunned BART model, consisting of 2 different modules, is used. The first part of BART is an encoder part which consists of 6 encoders. The encoder module of BART is same as that of BERT. The second part which is the decoder module, consists of 6 decoders, utilizes autoregressive technique. A bidirectional model is used to encode the erroneous text output and an autoregressive decoder is used to estimate the likelihood of the original document [20].

3.4. Bleu score for Accuracy Evaluation

Text to text comparison algorithm is used for evaluating the model’s performance. Bilingual evaluation understudy (BLEU) score is one of the metrics which is widely used for evaluating accuracies of NLP-translated text with respect to the ground truth sentences. BLEU score, valued between 0 to 1.

4. Results and Discussion

In this work, a study on efficacy of pre-processing technique is performed. Since the images used in this study deals with documents, instead of performing the outcome analysis with metrics such as SNR or other improvement scores, the text recognition accuracy is considered. The first study consists of the effects of segmentation on the recognition efficiency of the text present in document images. This study is carried out on the IAM dataset images, and the results are present in Table 1. Google Collab platform, basic and free version, using python as programming language has been utilized for carrying out the experiments reported in this paper.

Table 1. Effect of pre-processing & segmentation on handwritten text recognition accuracy in document images.

	ICR Accuracy (%)
Page Image	42.7%
Block Image	53.9%
Line Image	56.3%
Word Image	61.3%

Once the recognized text is received as output from the Google Read engine, it's compared against the ground truth data for accuracy evaluation with text-to-text comparison algorithm. Total 470 document images are used for this study. It may be observed from this table that the accuracy improves with segmentation of the document image into constituent components namely, blocks, lines and word segments. The best accuracy obtained is 61.3% which is almost 20% more than the recognition carried out by feeding the complete image to the recognition engine. This may be the effect of employing context specific segmentation logic which is not possible within the recognition engine block.

It was also observed that for a few images available in the IAM dataset, the segmentation could not be carried out effectively. Due to segmentation failure, the recognition output was also affected adversely. The above Table 1 contains the results of segmentation for documents where segmentation could be carried out effectively. Thus, it is logical to infer that while breaking down a document image to smaller blocks yield better recognition accuracy, such a strategy should be employed only in cases where proper segmentation is a certain possibility.

Employing language-based models for correcting textual errors has already been reported in literature (discussed in detail in Sec. 2). This work employs such models to study the effectiveness on available text data. For this study, a publicly available dataset, namely C4_3M dataset (ref. sec 3.1), is employed. The results associated with this study is present in Table 2, with column header "C4_3M dataset". Before employing the post-processing techniques, the base accuracy for the noisy text, available as 49% ICR accuracy, present in the dataset is evaluated for benchmarking.

Various language models, used for postprocessing technique, are: (i) Marian MT, (ii) BERT, and (iii) BART. These models are described in detail in Sec. 3.3. Marian MT requires training with our dataset while BERT and BART are available as pre-trained models. It may be noted, from the results in Table 2, that all these 3 models have improved the accuracy for C4_3M dataset from 49% to 59% & above. The maximum accuracy improvement, at 65.2%, is achieved with Marian MT. Since Marian MT is trained with the available dataset, it may be inferred that the dataset has certain attributes which may not be captured effectively with generic training. To validate this thought, pre-trained BART is finetuned with transfer learning on the C4_3M dataset. The experiment further enhances the accuracy to 68.3%, a 3% improvement over the generic model.

Encouraged by the observed accuracy improvements obtained through employing the various language models, experiments are conducted to check the same for handwritten ICR output text. These experiments were conducted for both block & word level text extracts for the IAM dataset. Table 2 presents the obtained improvements in columns titled "Block dataset" & "Word dataset", respectively. Benchmarking with plain ICR text accuracy is performed for comparison and accuracy improvement evaluation. Accuracy improvement to the order of 24% is observed for enhanced BART model which is consistent with the results observed for C4_3M dataset.

AVV-40 dataset contains images which are captured with a camera on a handheld device. These images contain higher levels of distortion and pose a challenge for segmentation. Furthermore, these documents contain only handwritten text in the entire page. Therefore, segmentation is avoided for these images and the entire page is taken as a block for ICR. The results of accuracy improvement for this dataset are present in Table 2, "AVV-40 dataset" column. Closer inspection of these results reveals the same pattern of improvement here. The enhanced BART improves the results from 39% to 56.5%, an increase of 17.5% (lower than observed for block & word images for IAM dataset). This may be due to the lower baseline accuracy for this dataset.

Table 2. Results of Post-Processing.

Metric (Bleu Score)	**C4_3M dataset**	**Block dataset**	**Word dataset**	**AVV-40 dataset**
ICR accuracy	49.0%	53.9%	61.3%	39.0%
Pre-trained BERT	59.2%	72.0%	76.4%	48.2%
Pre-trained BART	64.4%	73.2%	78.9%	51.5%
Marian MT	65.2%	75.4%	77.9%	54.4%
Enhanced BART	68.3%	77.3%	85.0%	56.5%
Consolidated Marian MT	67.2%	78.6%	82.3%	57.3%
Consolidated Enhanced BART	**70.6%**	**79.5%**	**86.4%**	**59.5%**

Further, the C4_3M, IAM dataset & Amrita training datasets are combined to form a combined training set. Study was conducted to evaluate the effectiveness of various models when learning is performed with the combined dataset. The accuracy improvement results for various datasets are tabulated in "Consolidated Marian MT" & "Consolidated Enhanced BART" rows of Table 2. A closer inspection of these results demonstrates that there is an improvement to the effectiveness of the models when trained with the combined training set. This is against the results obtained on the same models when trained with dataset specific training set. "Consolidated Marian MT" results are increased in range of 2 to 4% from the results available in Marian MT row, for various datasets. Similarly, "Consolidated Enhanced BART" improves the results by 2+% when compared to the results of "Enhanced BART". This validates the hypothesis that a combined training dataset pooled from multiple sources helps better generalization by the trained model.

5. Conclusion

This work employs pre-processing technique, noise removal, binarization and boundary detection, to improve the quality of input image to enhance the ICR accuracy. Additionally, this work proves that further improvement to recognized text may be brought in with post-processing techniques. It is observed that the recognition of word images generates better text output (61.3%) when compared to page (56.3%) or line images (42.7%). In the post-processing module, its proven that an enhanced BART model, finetuned with training data consolidated from recognition outputs from various types of handwritten images, provides the best improvement to the recognized text accuracy (86.4%) which is 20+% improvement over the ICR accuracy. Thus, looking at the obtained results from the various experiments, it may be inferred that a word level recognition of handwritten documents generates the best results when proper segmentation is possible. On the post-processing front, BART model finetuned with domain specific contextual dataset provides best accuracy improvement.

6. Acknowledgement

The authors express their gratitude to Amrita Vishwa Vidyapeetham for the infrastructure support. The authors also acknowledge the contributions of Dr. Beena B. M. and Dr. U. Kumaran, Department of CSE, Amrita Vishwa Vidyapeetham, Bengaluru campus, for their valuable inputs while reviewing the work.

References

[1] R. Plamondon and S. N. Srihari (2000) "Online and off-line handwriting recognition: a comprehensive survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 63-84, doi: 10.1109/34.824821.

[2] Yi yang, Peng jun Xie and Jun Tao (2017) “Alibaba at IJCNLP-2017 Task 1: Embeding grammatical feature into LSTM for Chinese Grammatical error diagnosis task”, Proceedings of the 8th International Joint Conference on Natural Language Processing, pp. 41-46.

[3] Min Tan, Dagang Chen, Zesong Li and Peng Wang (2020) “Spelling Error Correction with BERT based on Character-Phonetic”, IEEE 6th International Conference on Computer and Communications (ICCC), pp. 1146-1150, doi:10.1109/ICCC51575.2020.9345276.

[4] Yaung Hang, Xin Liu and Zhicho Shen (2018) “Spelling Error Correction Using a Nested RNN Model and Pseudo Training Data”, arXiv:1811.00238v1 [cs.CL].

[5] S. Carbonnel and E. Anquetil (2003) "Lexical post-processing optimization for handwritten word recognition," Proc. Seventh International Conference on Document Analysis and Recognition., pp. 477-481 vol.1, doi: 10.1109/ICDAR.2003.1227711.

[6] D.Wang, Y.Song and J.Li (2018) “A hybrid approach to automatic corpus generation for Chinese spelling check”, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2517-2527, doi: 10.18653/v1/D18-1273.

[7] Chen and Zesong (2016) “Sentence-Level Grammatical Error Identification as Sequence-to-Sequence Correction”, arXiv:1604.04677v1 [cs.CL] 16.

[8] Bohdan Didenko and Julia Shaptala (2019) “ Multi-headed Architecture Based on BERT for Grammatical Errors Correction”, Proc. Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 246–251, doi: 10.18653/v1/W19-4426.

[9] Yifei Hu, Xiaonan Jing, Youlim Ko and Julia Taylor Rayz (2021) “Misspelling Correction with Pre-trained Contextual Language Model”, arXiv:2101.03204v1 [cs.CL].

[10] J. C. Amengual and E. Vidal (1998) "Efficient error-correcting Viterbi parsing", in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 10, pp. 1109-1116, doi: 10.1109/34.722628.

[11] C. Perez-Cortes, J. C. Amengual, J. Arlandis and R. Llobet (2000) "Stochastic error-correcting parsing for OCR post-processing", Proceedings 15th International Conference on Pattern Recognition. ICPR-2000, pp. 405-408 vol.4, doi: 10.1109/ICPR.2000.902944.

[12] U. Marti and H. Bunke (2002) “The IAM-database: An English Sentence Database for Off-line Handwriting Recognition”, Int'l Journal on Document Analysis and Recognition, vol. 5, pp. 39 – 46, url: https://fki.tic.heia-fr.ch/databases/download-the-iam-handwriting-database.

[13] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li and Peter J. Liu (2019) “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”, url: https://www.tensorflow.org/datasets/catalog/c4.

[14] B. N. K. Sai and T. Sasikala (2019) "Object Detection and Count of Objects in Image using Tensor Flow Object Detection API", 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 542-546, doi: 10.1109/ICSSIT46314.2019.8987942.

[15] A. Shrivastava, J. Amudha, D. Gupta and K. Sharma (2019) "Deep Learning Model for Text Recognition in Images", 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-6, doi: 10.1109/ICCCNT45670.2019.8944593.

[16] D. R. Sharma and S. Bhaskaran (2019) "Categorization of Multilingual Text on Languages of Indic Script", 2019 Global Conference for Advancement in Technology (GCAT), pp. 1-4, doi: 10.1109/GCAT47503.2019.8978281.

[17] K. Yamuna, V. Shriamrut, D. Singh, V. Gopalasamy and V. Menon (2021) "Bert-based Braille Summarization of Long Documents," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1-6, doi: 10.1109/ICCCNT51525.2021.9579748.

[18] G. Joseph, Dr. Tripty Singh, Andan, H. M. and Aravind G (2015) “Registration Plate Recognition from Still Images and Videos in Real Time Conditions”, International Journal of Computer Science and Information Technologies, vol. 6, no. 3, pp. 2729-2734.

[19] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov and Luke Zettlemoyer (2019) “Denoising Sequence-to-Sequence Pre-training for Natural Language Generation”, Translation, and Comprehension”, arXiv:1910.13461v1 [cs.CL].

[20] S. Sooraj, Manjusha, K., M. Kumar, A., and Dr. Soman K. P (2018) “Deep learning based spell checker for Malayalam language”, Journal of Intelligent and Fuzzy Systems, vol. 34, pp. 1427-1434.