TrOCR Meets Language Models: An End-to-End Post-Correction Approach

Abstract. This study aims to enhance handwritten text recognition (HTR) performance and domain adaptability by combining an optical character recognition (OCR) model with a language model (LM) that serves as a corrector. This integration addresses three principal challenges: over-correction, which compromises text authenticity; poor domain adaptation; and the scarcity of annotated images. We explore the synergy between TrOCR, a state-of-the-art OCR model, and CharBERT, a BERT-based LM. A novel aspect of our research involves introducing common errors made by the recogniser into the LM, enabling it to consider these errors during correction, thereby improving overall performance. Our findings reveal that the hybrid TrOCR-CharBERT model effectively balances visual and linguistic information, preserving the authenticity of the original texts. Furthermore, the model is able to adapt to historical data even when the recogniser is trained solely on contemporary data, mitigating the need for a large number of annotated historical handwritten images.

Keywords: Handwritten Text Recognition · Domain Adaptation · Annotated Data Scarcity Mitigation

1 Introduction

OCR has become a key tool for digitising handwritten documents [31]. While OCR tasks for modern printed materials are typically straightforward, digitising historical texts or handwritten documents introduces complex challenges. Inadequate OCR can significantly affect downstream tasks such as text classification, named entity recognition [6], and information retrieval [14], leading to poor data utility.

One of the main challenges of OCR is the poor quality of images, which can include issues such as heterogeneous character heights, ink smears, and ink bleed-through [16]. Other challenges include the dynamics of languages and the lack of resources. Consequently, post-OCR correction is crucial to overcoming these limitations and enhancing the accuracy of digitised data [28]. Several post-OCR correction methods have been applied, showing significant improvements. Most of these methods function sequentially rather than in an end-to-end manner. However, allowing backpropagation to influence both the recogniser and the LM for correction concurrently can yield better results than training them separately [10]. This integrated approach enables the LM to leverage insights from its own processing and feedback from the recogniser, facilitating more accurate corrections.

This study seeks to assess the effectiveness of integrating a recogniser with an LM to enhance performance and to enable the composite model to adapt to historical data even when trained on modern data. We have selected TrOCR [18], a state-of-the-art (SOTA) OCR model known for its advanced text recognition capabilities, and CharBERT [24], a variant of the BERT [4] model with additional character-level processing. In addition, we propose a novel approach to integrate the common errors made by TrOCR into CharBERT. By leveraging these technologies, our approach substantially improves the accuracy and reliability of text digitisation processes. The model enhances accuracy and reliability and adapts to different time periods of English, even if the recogniser is trained solely on contemporary English. This approach significantly reduces the need for annotated OCR images.

To test domain adaptability, we train a CharBERT variant (referred to as CharBERT$_{\text{HISTORICAL}}$) on a historical dataset rather than the contemporary dataset originally used to pre-train CharBERT. We aim to validate the adaptability of different composite models with LMs trained on different domains of datasets. To test the effect of integrating common OCR errors into CharBERT, we introduce common errors made by TrOCR into the training process of CharBERT to enable it to learn to correct them. This variant of CharBERT is referred to as CharBERT$_{\mathcal{P}_{ij}}$.

The variants of CharBERT are trained with a substantially smaller amount of data than the original pre-trained CharBERT due to computational resources and time limitations. To ensure a fair comparison of the composite model with different variants of CharBERT, we trained a CharBERT using the same setup as the pre-trained CharBERT but with a smaller amount of data to serve as the baseline for other variants of CharBERT. This CharBERT will be referred to as CharBERT$_{\text{SMALL}}$.

2 Related Work

The OCR process includes several stages, such as preprocessing, segmentation, feature extraction, and recognition [11]. While traditional models typically handle these stages separately, modern OCR models operate on an end-to-end basis [8,26,2]. These end-to-end models streamline the text recognition process by integrating all these stages into a single, continuous workflow. This approach leverages advanced machine learning techniques, particularly deep learning.

Modern text recognition approaches employ convolutional neural networks (CNNs) [34] and long short-term memory networks [3] to enhance accuracy. Transformer models [33], originally developed for natural language processing tasks, have also been successfully adapted for OCR applications. Models such as TrOCR [18], which combines a Vision Transformer [5] with pre-trained LMs like RoBERTa [23], have demonstrated remarkable effectiveness in extracting text from images. Additionally, the incorporation of attention mechanisms in OCR models has enabled the network to focus on specific parts of an image sequentially, mimicking the human reading process. Examples include the Attentionbased Scene Text Recognizer [30] and STAR-Net [22], both of which have substantially enhanced OCR accuracy.

Post-OCR correction refers to the methods and processes used to correct errors in text after it has been converted from images (such as scanned documents or photos) to editable and searchable text data. This stage is crucial because OCR technology, despite its advancements, often makes mistakes due to various challenges such as poor image quality, complex layouts, unusual fonts, or difficult handwriting.

Post-OCR correction is an effective tool but comes with its own set of challenges, such as distorted outputs from OCR processes or domain-specific terminology within datasets. To address this, [12] proposed a RoBERTa model employing a self-supervised pre-training approach to predict masked sections of medical texts. The authors discuss and tackle the inherent challenges posed by the varying accuracy of OCR technology, especially in recognising texts that contain medical terminologies and are often scanned from physical documents where text may be skewed or obscured.

Other research proposed solving OCR error correction as a machine translation problem [1,25]. [25] employed two deep learning models: a word-based sequence-to-sequence model and a character-based model. Results show that character-based models, which allow for the correction of individual characters, handle words not seen during training more effectively. While word-based models struggle with unseen words, character-based models perform well across different datasets [10].

3 Data

3.1 Data for Composite Model Training

The data used in this study includes the George Washington (GW) handwritten dataset [7] and the Joseph Hooker (JH) handwritten dataset. These datasets serve as benchmarks for developing and evaluating handwriting recognition systems. We selected the GW and JH datasets because they represent different centuries of English and various topics, making them ideal for our experiments. Detailed information about these datasets is presented in Table 1.

With its extensive collection of 19th-century botanical writings and correspondence, the JH dataset provides a unique challenge for HTR technologies due to scientific terminology and personal handwriting styles. Similarly, the GW dataset, consisting of an array of 18th-century materials, including letters, diaries, and official documents, poses unique challenges due to the use of archaic words and phrases.

Image Processing In this study, images are initially resized to 384x384 pixels to comply with the input requirements of the pre-trained TrOCR model. Following resizing, the images undergo normalisation. The normalisation specifies the mean for each of the three colour channels (Red, Green, Blue), all set to 0.5. Together, resizing and normalising data ensure that no single pixel range overly influences the network due to its scale.

Metric	GW Dataset	JH Dataset
Text lines	656	6,916
Training data (lines)	329	5,532
Validation data (lines)	168	691
Test data (lines)	163	693
Tokens	4,850	38,831
Types	1,456	8,308
Unique characters	68	84
Average # of characters per line	40.23	28.45
Average # of tokens per line	7.39	5.62
Percentage of non-characters[1]	21%	22.4%

[1] Characters other than A-Z, a-z, 0-9

Table 1: GW and JH dataset statistics

3.2 Data for CharBERT Variants

The training process for $\text{CharBERT}_{\text{SMALL}}$ and $\text{CharBERT}_{\mathcal{P}_{ij}}$ involved randomly sampled sentences from Wikipedia (1.13 GB) with 167M words. The only difference is that the errors introduced in the inputs of $\text{CharBERT}_{\text{SMALL}}$ are random, while the errors introduced in $\text{CharBERT}_{\mathcal{P}_{ij}}$ follow the probability of OCR errors occurring in the OCR model outputs, which will be discussed shortly in Section 5. On the other hand, $\text{CharBERT}_{\text{HISTORICAL}}$ is trained on 637MB of literature from the 16th to 19th centuries [17,20,27,13,21,19,9].

$\text{CharBERT}_{\text{SMALL}}$ serves as a baseline for comparisons with two other implementations of CharBERT: $\text{CharBERT}_{\mathcal{P}_{ij}}$ – which incorporates common OCR errors into the model – and $\text{CharBERT}_{\text{HISTORICAL}}$ – which is trained on historical English (from the 16th and 19th centuries) rather than contemporary English.

4 Recogniser and Language Model

TrOCR [18] is a Transformer-based model that focuses on the text recognition part of the OCR task, converting images to text. It has been selected as our primary OCR model for text recognition due to its SOTA performance and its capability to adapt to new handwriting styles with little data (see, e. g., [32]). As an end-to-end model, TrOCR simplifies the processing pipeline by eliminating the need for separate image processing and feature extraction steps. This allows it to be easily fine-tuned in conjunction with LMs. TrOCR will serve as the baseline against the composite model of TrOCR combined with CharBERT in this study.

CharBERT [24] is an enhancement of BERT designed to address issues in byte-pair encoding (BPE) [29] used by pre-trained LMs like BERT and RoBERTa. CharBERT takes text as input during inference and outputs two representative embeddings: a token embedding and a character embedding.

CharBERT introduces two techniques during the pre-training stage: 1) employing a dual-channel architectural approach for both subword and character information and 2) utilising noisy language modelling (NLM) for unsupervised representation learning. The first technique processes and fuses subword and character-level information, ensuring a more robust representation in case of typos. The second technique involves introducing character-level noise into words, and training the model to correct these errors.

CharBERT will function as the corrector in our post-OCR correction system. It meets our requirements by featuring character-level processing and a BERT-like architecture. Additionally, given its pre-training tasks, CharBERT is particularly effective at correction tasks, making it highly suitable for our post-OCR correction needs. Moreover, we can easily integrate common OCR errors into the CharBERT pre-training task, NLM.

5 Composite Model Architecture

The composite model is designed to integrate TrOCR and CharBERT. During the inference phase, the decoder output is recycled back as input. Before this recycled input is fed back into the decoder, it undergoes correction and refinement by CharBERT. Consequently, CharBERT is positioned between the decoder input and the decoder stacks as illustrated in Figure 1. However, integrating these systems requires several adaptations to the models due to the following reasons: 1) The TrOCR decoder accepts token IDs as input, whereas CharBERT outputs embeddings; 2) CharBERT requires textual inputs, but the TrOCR decoder input is a tensor; 3) The embedding representations of TrOCR do not align with those of CharBERT; 4) The input to the TrOCR decoder is a single tensor, while CharBERT produces dual-channel outputs (token and character channel outputs). The following sections will discuss these adaptations in detail.

Adapted TrOCR The TrOCR decoder is specifically designed to accept token IDs as input, which are then mapped to embeddings. However, CharBERT outputs embeddings rather than token IDs, leading to a compatibility issue. To resolve this, we reposition the embedding layer from the TrOCR decoder to precede CharBERT. This adjustment ensures that token IDs are initially converted to the TrOCR embedding, which are then input into CharBERT for correction. Consequently, the adapted TrOCR can accept embeddings directly, bypassing the need for token IDs. This adaptation is shown in Figure 2

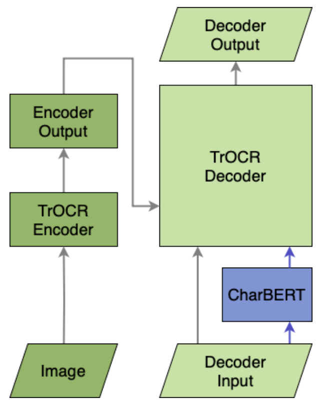

Fig. 1: Workflow in the composite model combining TrOCR and CharBERT.

Adapted CharBERT According to the modifications described in the adapted TrOCR, the TrOCR decoder input is now an embedding, which should be processed by CharBERT for correction. However, the original CharBERT only accepts text as input. Therefore, we have redesigned this revised model so that CharBERT no longer converts text into IDs and then into embeddings; instead, it receives pre-processed embeddings directly. This adaptation allows both token and character embeddings to be processed through their respective channels in CharBERT as shown in Figure 3.

Tensor Transformation Module Not only do the models represent the same text differently, but their embedding dimensions are also incompatible, further complicated by CharBERT's dual-channel embeddings. To overcome this issue, we designed an architecture referred to as the Tensor Transformation Module, as illustrated on the right side of Figure 4. The CNN and feedforward neural network (FFNN) layers not only align the dimensions between the tensors but also learn to map the contextual information from TrOCR embeddings to those of CharBERT.

In the first stage, the decoder input passes through a series of CNN layers, interspersed with LeakyReLU activation functions and batch normalisations, specifically designed to adjust the sequence dimension (dim=1). Subsequently, the output from the first stage is processed through FFNN layers in the second stage to modify the embedding dimension (dim=2).

The Tensor Transformation Module is specifically designed to convert the TrOCR decoder input into CharBERT token and character channel inputs. This module is also critical in the Tensor Combine Module.

Tensor Combine Module CharBERT produces two separate tensors – token and character representations – while the TrOCR decoder requires a single tensor input. To address this, we design the Tensor Combine Module to merge the two

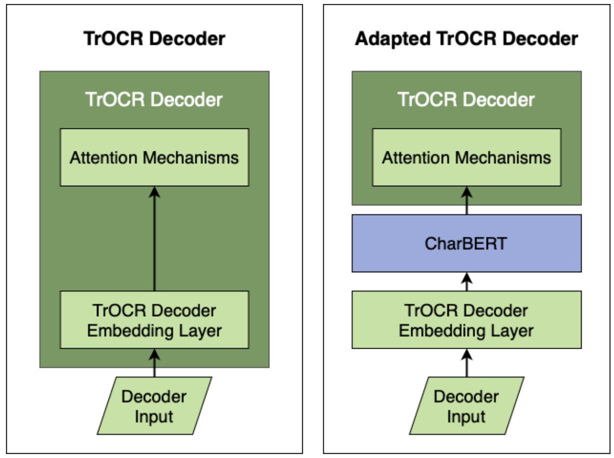

Fig. 2: Comparison of the TrOCR Decoder architecture.

output tensors from CharBERT into a single tensor. Additionally, a residual connection from the original TrOCR decoder embedding is added. This residual connection helps to reuse features from the original TrOCR decoder embedding and prevents gradient vanishing. The architecture of the Tensor Combine Module is shown on the left side of Figure 4.

Firstly, the two outputs from CharBERT undergo a transformation via the Tensor Transformation Module to match the original input size of the TrOCR decoder. Then, the Tensor Combine Module dynamically allocates attention weights to each word across the three input tensors. It incorporates linear layers as the attention network. This strategy is particularly effective for non-spatial input types such as text embedding.

Common Error Incorporation In our methodology, we enhance the training process by specifically targeting commonly misrecognised characters, such as "." and "," or "O" and "o," to reduce the likelihood of these errors in future recognitions. To achieve this, we begin by determining the transition probability $\mathcal{P}_{ij}$, where i represents the correct character that has been erroneously recognised as character j. We obtain $\mathcal{P}_{ij}$ by calculating the frequency of each character misrecognised by TrOCR on the GW and JH datasets. Then, according to this probability, we incorporate errors into the text during CharBERT NLM training.

Training CharBERT$_{\mathcal{P}_{ij}}$ To train CharBERT$_{\mathcal{P}_{ij}}$, we adhere to the training methods outlined in the original CharBERT paper, but with a smaller amount of data. In the original CharBERT training, the model is pre-trained by randomly adding, deleting, or swapping characters within the input text to simulate typical errors, thereby training CharBERT to correct them. For our specific application focusing on OCR corrections, we have modified this approach by replacing the

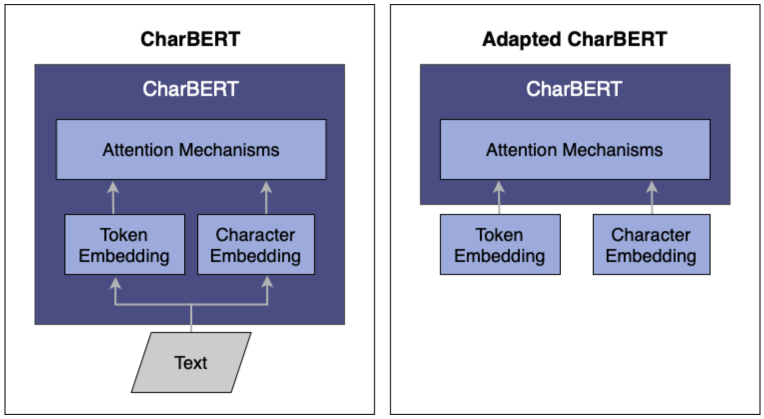

Fig. 3: Comparison of the CharBERT architecture

random swapping of characters with common misrecognised OCR errors according to the probability, $\mathcal{P}_{ij}$.

6 Experiments and Analysis

For the training of the composite model, we use Adam [15] as the optimiser and cross-entropy for loss computation. The learning rate is set to 1e−5 with a weight decay parameter of 1e−5. The composite model is trained with all the TrOCR parameters frozen. Each experiment utilises one A100 GPU with 80GB RAM.

In this study, CharBERT is initially trained on different datasets to learn fundamental language patterns. Subsequently, CharBERT is combined with the recogniser and trained on handwritten datasets. This approach enables the CharBERT to consider both its own knowledge and TrOCR's predictions when generating the output text, adjusting its predictions by considering TrOCR's decisions.

6.1 Baseline Model

We use the pre-trained handwritten large TrOCR as a baseline and evaluate on the GW and the JH datasets. Additionally, we use a fine-tuned version of TrOCR on each of these datasets for further comparison. The results in terms of word error rates (WER) and character error rates (CER) of this analysis are shown in Table 2.

Upon examining the GW dataset TrOCR outputs without fine-tuning, we see that TrOCR tends to over-correct the text. As illustrated below, TrOCR autocorrects "Expamples" (an original misspelling by George Washington) to "Examples," the correct form of the word. Additionally, it completes the truncated word "ar" as "arm," without presuming the word was inadvertently cut

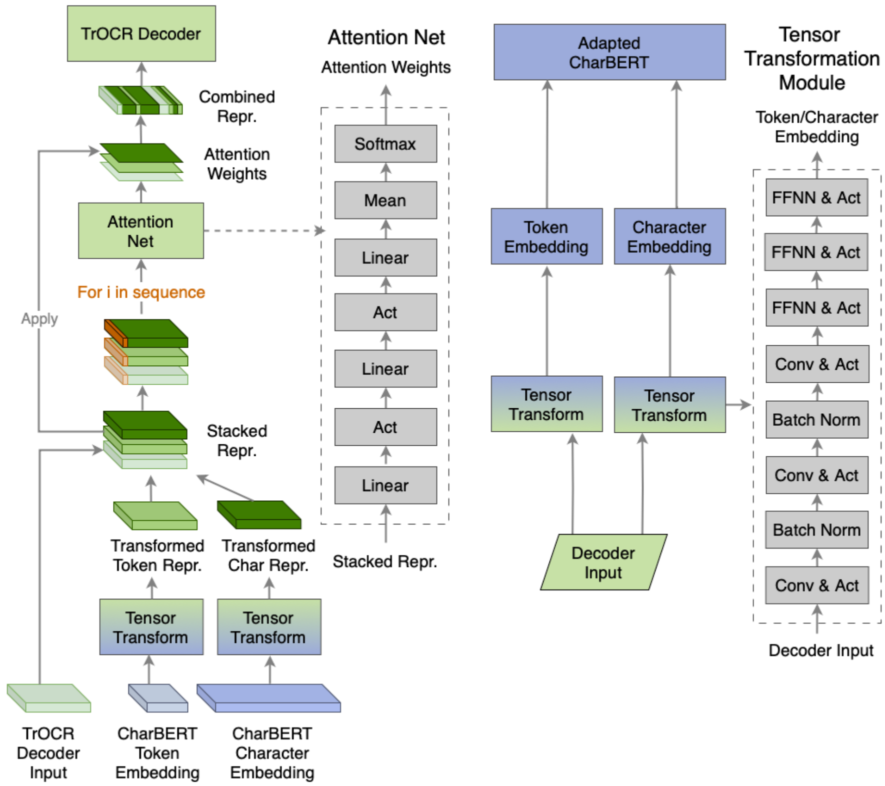

Fig. 4: Architecture of the Tensor Combine Module (left) and the Tensor Transformation Module (right).

short. A few images (fewer than 10) in the JH dataset contain printed rather than handwritten letters. TrOCR recognised "THE Camp." instead of the correct label "THE CAMP." Although TrOCR's handwritten model can recognise printed letters, it struggles with correct capitalisation.

```
Test on GW Dataset Without Fine-Tuning
label:  est occasion for Expamples, will be morally im
output: cut occasion for Examples, will be morally in-

Test on GW Dataset After Fine-Tuning
label:  that were expected in; and to wait the ar
output: That were expected in; and to wait the arm

Test on JH Dataset Without Fine-Tuning
label:  THE CAMP.
output: THE Camp.
```

Dataset	Fine-Tune	WER	CER
GW	False	37.76	15.40
GW	True	14.44	4.78
JH	False	91.31	58.57
JH	True	36.97	20.28

Table 2: Baseline model results in word and character error rates (WER, CER).

The tendency to over-correct is particularly noticeable at the end of sentences where the last word is truncated. TrOCR often attempts to complete these cut-off words, or it may substitute them with a different word that, while seemingly appropriate, is irrelevant to the original token image. In some cases, TrOCR even transforms the incomplete word into a non-existent word. Notably, this tendency to over-correct persists even after fine-tuning.

6.2 Composite Model (TrOCR-CharBERT)

The composite model significantly outperforms the baseline model and achieves more precise post-corrections. Unlike TrOCR alone, which may over-correct or erroneously complete words, this hybrid approach maintains the authenticity of the original images. For example, TrOCR-CharBERT correctly recognises "ar" and "Expamples" without over-correcting them as TrOCR does. By integrating CharBERT, the model leverages both visual information and linguistic knowledge, enabling it to make more informed decisions about when to amend the text and when to preserve the original input. This is also effective in the JH dataset, where the combined model correctly recognizes "THE CAMP.," accurately handling printed letters without the capitalization errors seen with TrOCR alone. The result of this analysis is shown in Table 3.

TrOCR-CharBERT substantially reduces the number of over-corrections, as shown in the Figure 5. This figure illustrates the percentage of outcomes for unfinished word scenarios within the GW dataset, comparing the fine-tuned TrOCR and the TrOCR-CharBERT. In the case of the GW testing dataset, which includes 30 labels[2] ending with unfinished words, the fine-tuned TrOCR model correctly transcribes only 5 unfinished words, whereas the TrOCR-CharBERT model correctly transcribes 10. The categories "Complete word," "Other word," and "Not a word" indicate whether the model attempted to complete the unfinished words, substituted them with a different word it deemed fit, or transformed them into non-words, respectively. The pie charts reveal that the integration with CharBERT significantly reduces the instances of attempting to complete words erroneously, demonstrating its ability to preserve text authenticity more accurately.

[2] The number of labels ending with unfinished words is counted by the author.

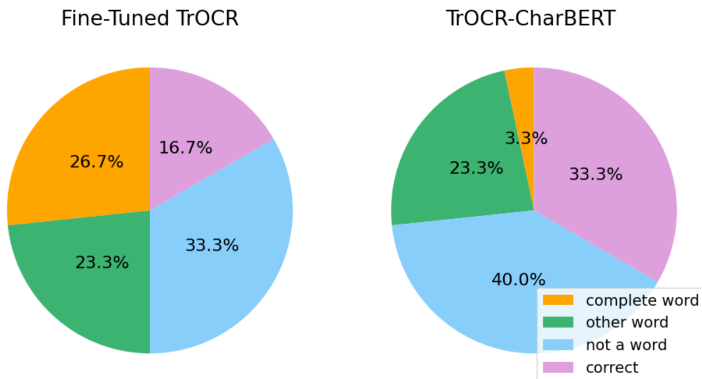

Fig. 5: Comparison of over-correction for TrOCR and TrOCR-CharBERT

Dataset	Model	WER	CER
GW	TrOCR-CharBERT	12.84	5.88
JH	TrOCR-CharBERT	35.30	21.33

Table 3: Results of TrOCR-CharBERT.

6.3 Validating Model Domain Adaptability

To assess the model's domain adaptability, we compare the performance of both TrOCR-CharBERT$_{\text{SMALL}}$ and TrOCR-CharBERT$_{\text{HISTORICAL}}$ to determine the extent of the composite model's ability to adapt to the domain-specific characteristics of different datasets. The result of this analysis is shown in Table 4.

TrOCR is trained on contemporary English and is frozen during the experiment. Despite this, there is still a performance boost when TrOCR-CharBERT$_{\text{HISTORICAL}}$ is applied to the GW dataset, which is not contemporary English. This indicates that the recogniser can be trained to recognise general English character glyphs, and the LM can adapt to different domains of image data by training on that specific domain corpora. This can greatly reduce the need for annotated OCR images.

6.4 TrOCR-CharBERT$_{\mathcal{P}_{ij}}$ Analysis

This analysis evaluates the positive effect of integrating common errors identified in TrOCR outputs into the training of CharBERT, referred to as CharBERT$_{\mathcal{P}_{ij}}$. The result of this analysis is shown in Table 5.

The results suggest that incorporating knowledge about common OCR mistakes into the model helps refine its predictions. This refinement is more pro-

Training Dataset	LM Training Data	WER	CER
GW	Contemporary English	13.88	6.51
GW	15^{th} – 18^{th} Century English	13.18	6.05

Table 4: Model Domain Adaptability Results

Training Dataset	Model	$\mathcal{P}_{ij}$	WER	CER
GW	TrOCR-CharBERT$_{\text{SMALL}}$	False	13.88	6.51
GW	TrOCR-CharBERT$_{\mathcal{P}_{ij}}$	True	12.94	6.03
JH	TrOCR-CharBERT$_{\text{SMALL}}$	False	34.42	21.60
JH	TrOCR-CharBERT$_{\mathcal{P}_{ij}}$	True	33.95	21.86

Table 5: TrOCR-CharBERT$_{\mathcal{P}_{ij}}$ Results

nounced in the GW dataset, indicating that the nature of errors in this dataset may be more systematically addressable. While the performance improvement is less marked for the JH dataset, there is still a reduction in WER from 34.42 to 33.95. Interestingly, the CER slightly increases from 21.60 to 21.86, indicating that while some errors are corrected, new ones may be introduced due to the complexity and variability in the JH dataset.

Integrating common OCR mistakes into the training process enhances model performance, particularly for more homogeneous datasets like GW. If we use CharBERT trained with more data, we can expect even better performance than the TrOCR-CharBERT model above.

7 Conclusion

Combining the recogniser with the LM allows the LM to access image information, correct words more accurately, and prevent over-correction. This helps preserve the authenticity of the texts in the images, which is crucial for OCR tasks. In addition, the composite model can adapt to different domains of data with only the LM trained on that specific domain of text. This significantly reduces the need for annotated historical text images. Furthermore, the composite structure allows us to integrate common OCR errors into the LM training process, significantly improving error rates by making it more aware of frequent recognition mistakes. Thus, integrating a recogniser and an LM remains a valid and promising approach, offering significant benefits worth further exploration. Future research should support multilingual scripts, and reduce the model's computational requirements to improve efficiency and applicability.

Disclosure of Interests. The authors declare that there are no conflicts of interest regarding the publication of this paper. This research received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The views and opinions expressed in this paper are those of the authors and do not necessarily reflect the official policy or position of any affiliated agency of the authors.

References

1. Chantal Amrhein and Simon Clematide. Supervised ocr error detection and correction using statistical and neural machine translation methods. *Journal for Language Technology and Computational Linguistics (JLCL)*, 33(1):49–76, 2018.
2. Birhanu Belay, Tewodros Habtegebrial, Million Meshesha, Marcus Liwicki, Gebeyehu Belay, and Didier Stricker. Amharic OCR: an end-to-end learning. *Applied Sciences*, 10(3):1117, 2020.
3. Thomas M Breuel, Adnan Ul-Hasan, Mayce Ali Al-Azawi, and Faisal Shafait. High-performance OCR for printed English and Fraktur using LSTM networks. In *2013 12th international conference on document analysis and recognition*, pages 683–687. IEEE, 2013.
4. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.
5. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.
6. Maud Ehrmann, Ahmed Hamdi, Elvys Linhares Pontes, Matteo Romanello, and Antoine Doucet. Named entity recognition and classification in historical documents: A survey. *ACM Computing Surveys*, 56(2):1–47, 2023.
7. Andreas Fischer, Andreas Keller, Volkmar Frinken, and Horst Bunke. Lexicon-free handwritten word spotting using character hmms. *Pattern recognition letters*, 33(7):934–942, 2012.
8. Jing Huang, Guan Pang, Rama Kovvuri, Mandy Toh, Kevin J Liang, Praveen Krishnan, Xi Yin, and Tal Hassner. A multiplexed network for end-to-end, multilingual OCR. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4547–4557, 2021.
9. Magnus Huber, Magnus Nissel, and Karin Puga. Old bailey corpus 2.0. hdl:11858/00-246C-0000-0023-8CFB-2, 2016. Licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.
10. Lei Kang, Pau Riba, Mauricio Villegas, Alicia Fornés, and Marçal Rusiñol. Candidate fusion: Integrating language modelling into a sequence-to-sequence handwritten word recognition architecture. *Pattern Recognition*, 112:107790, 2021.
11. Kanagarathinam Karthick, KB Ravindrakumar, R Francis, and S Ilankannan. Steps involved in text recognition and recent research in OCR; a study. *International Journal of Recent Technology and Engineering*, 8(1):2277–3878, 2019.
12. Srinidhi Karthikeyan, Alba G Seco de Herrera, Faiyaz Doctor, and Asim Mirza. An OCR post-correction approach using deep learning for processing medical reports. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(5):2574–2581, 2021.

13. Hannah Kermes, Stefania Degaetano-Ortlieb, Ashraf Khamis, Jörg Knappen, and Elke Teich. The royal society corpus: From uncharted data to corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1928–1931, 2016.
14. Kimmo Kettunen, Heikki Keskustalo, Sanna Kumpulainen, Tuula Pääkkönen, and Juha Rautiainen. OCR quality affects perceived usefulness of historical newspaper clippings – a user study. In *Proceedings of the 18th Italian Research Conference on Digital Libraries (IRCDL 2022)*, CEUR-WS.org, Padova, Italy, 2022. CEUR Workshop Proceedings.
15. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
16. Berat Kurar Barakat, Rafi Cohen, Ahmad Droby, Irina Rabaev, and Jihad El-Sana. Learning-free text line segmentation for historical handwritten documents. *Applied Sciences*, 10(22):8276, 2020.
17. Merja Kytö and Jonathan Culpeper. A corpus of english dialogues 1560-1760 (CED), 2006. Literary and Linguistic Data Service.
18. Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. TrOCR: Transformer-based optical character recognition with pre-trained models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13094–13102, 2023.
19. Literary and Linguistic Data Service. The lampeter corpus of Early Modern English tracts. Literary and Linguistic Data Service.
20. Literary and Linguistic Data Service. Pamphlets of the american revolution : [selections] / edited by bernard bailyn, 1994. Literary and Linguistic Data Service.
21. Literary and Linguistic Data Service. The english language of the north-west in the late modern english period: a corpus of late 18c prose, 2003. Literary and Linguistic Data Service.
22. Wei Liu, Chaofeng Chen, Kwan-Yee K Wong, Zhizhong Su, and Junyu Han. Star-net: a spatial attention residue network for scene text recognition. In *BMVC*, volume 2, page 7, 2016.
23. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
24. Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. CharBERT: Character-aware pre-trained language model. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 39–50, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.
25. Kareem Mokhtar, Syed Saqib Bukhari, and Andreas Dengel. OCR error correction: State-of-the-art vs an NMT-based approach. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 429–434. IEEE, 2018.
26. Clemens Neudecker, Konstantin Baierer, Maria Federbusch, Matthias Boenig, Kay-Michael Würzner, Volker Hartmann, and Elisa Herrmann. OCR-D: An end-to-end open source OCR framework for historical printed documents. In *Proceedings of the 3rd international conference on digital access to textual cultural heritage*, pages 53–58, 2019.
27. Terttu Nevalainen, Helena Raumolin-Brunberg, Jukka Keränen, Minna Nevala, Arja Nurmi, Minna Palander-Collin, Ann Taylor, Susan Pintzuk, and Anthony Warner. Parsed corpus of early english correspondence (PCEEC), 2006. Literary and Linguistic Data Service.

28. Thi Tuyet Hai Nguyen, Adam Jatowt, Mickael Coustaty, and Antoine Doucet. Survey of post-OCR processing approaches. *ACM Computing Surveys (CSUR)*, 54(6):1–37, 2021.
29. Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.
30. Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. ASTER: An attentional scene text recognizer with flexible rectification. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2035–2048, 2018.
31. Amarjot Singh, Ketan Bacchuwar, and Akshay Bhasin. A survey of OCR applications. *International Journal of Machine Learning and Computing*, 2(3):314, 2012.
32. Phillip Benjamin Ströbel, Tobias Hodel, Walter Boente, and Martin Volk. The adaptability of a transformer-based ocr model for historical documents. In *Document Analysis and Recognition – ICDAR 2023 Workshops: San José, CA, USA, August 24–26, 2023, Proceedings, Part I*, pages 34–48, 2023.
33. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
34. Haochen Zhang, Dong Liu, and Zhiwei Xiong. CNN-based text image super-resolution tailored for OCR. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2017.