Synthetically Augmented Self-Supervised Fine-Tuning for Diverse Text OCR Correction

Shuhao Guan and Derek Greene

Insight Centre for Data Analytics, Dublin
School of Computer Science, University College Dublin, Ireland

Abstract. The adoption of Optical Character Recognition (OCR) tools has been central to the increased digitization of historical documents. However, the errors introduced during OCR, particularly in texts with a specialized vocabulary (SV), necessitate effective post-OCR correction methodologies. This study introduces a novel approach that leverages weak supervision and self-supervised fine-tuning to enhance post-OCR correction without the need for substantial manual annotations. By using multi-noise-level synthetic data, generated through automatically-extracted OCR errors and applied to clean texts, we can train robust models tailored for post-OCR tasks. Furthermore, we propose a unique self-supervised fine-tuning strategy, applied specifically to long texts, enables models to adeptly handle out-of-vocabulary problems and SV. Additionally, we tested the performance of the GPT model on post-OCR tasks.

1 Introduction

Advancements in OCR have significantly transformed the preservation and analysis of historical printed and handwritten materials, making them more accessible for research and archival purposes. While OCR systems have made it easier to digitize collections of culturally important texts, they also present challenges, especially when working with documents that have complex layouts or damaged pages. Post-OCR processing aims to address these issues by correcting errors and enhancing the usability of digitized texts. However, traditional manual post-OCR correction methods, despite their acknowledged benefits, are often labor-intensive and costly. Consequently, a variety of automated techniques have been proposed for this task [29].

Recent studies have redefined post-OCR correction as a Seq2Seq Neural Machine Translation (NMT) challenge [1, 26, 27, 15]. Such models have been shown to effective in correcting OCR-produced texts. However, the reliability and generalization of NMT-based models largely depend on the quality and quantity of the training data [43]. Unfortunately, obtaining large, diverse training datasets is often expensive or unfeasible [32]. The situation is made more complex by the nuances of specific post-OCR processing techniques. In addition to errors coming from the original digitization process, many post-OCR procedures can, counter-intuitively, introduce further errors by altering previously-accurate terms. This becomes especially problematic when dealing with important Out-of-Vocabulary (OOV) domain-specific terms and unique entity names, which we refer to as a Specialized Vocabulary (SV). Such errors not only reduce the texts' usefulness for close reading by humanities scholars, but also can impact upon subsequent analyses [42].

In response to the challenges above, in Section 3 we describe a process that employs weak supervision to create rich synthetic datasets to facilitate the training of robust post-OCR correction models. We also propose a Self-Supervised Fine-Tuning (SSFT) method, which improves the model's ability to manage specialized vocabulary and historical linguistic nuances. This not only yields more accurate OCR corrections, but also preserves the integrity of the original text, which is essential for cultural analytics researchers. The results presented later in Section 5 show that our proposed techniques are effective across multiple different benchmarks, including those involving fictional texts and historical newspapers. We also conduct experimental comparisons with GPT models, which have demonstrated impressive capabilities across various NLP tasks. Our data generation process, in conjunction with SSFT, provides a scalable and efficient strategy for post-OCR correction, making digitized documents more accessible and useful for downstream tasks.

2 Related Work

A variety of methods have been proposed in the literature for post-OCR text correction [29]. Some studies leverage multiple OCR engines and combine their outputs to yield improved results [22, 24]. Recent advances involve using pre-trained models and Seq2Seq architectures for correction purposes [1, 28]. Schaefer and Neudecker [38] introduced a two-step post-OCR method using an LSTM-based neural network to first detect errors and then correct them. An unsupervised approach combining multiple OCR views via pre-trained language models is proposed by Gupta et al. [14]. Ramirez-Orta et al. [35] split each input document into character n-grams and combine their individual corrections into the final output. Recent works have employed Transformer-based encoder-decoder models [23, 34, 40].

Synthetic data is widely used in post-OCR correction tasks [30] and it is typically generated using two methods: Noise Injection [17] and Back Translation [39]. In most cases, synthetic data with only a single noise ratio is considered when training a model. Koo et al. [20] investigated the impact of noise insertion ratio on model performance in GEC tasks. Jasonarson et al. [18] and Rijhwani et al. [37] first detected the frequency of OCR errors in a corpus and then injected errors into clean text based on the observed frequencies. D'hondt et al. [11] proposed generating multiple datasets with different noise ratios. In another study, Guan and Greene [13] incorporated various noise ratios into their synthetic datasets, compared different methods for generating synthetic data, and introduced a glyph-similarity-based approach that proved effective in low-resource languages. They observed that for resource-rich languages like English, extracting error

distributions from existing datasets and using this information to inform the synthetic data creation process is highly effective.

Synthetic data generation is important for augmenting limited training datasets in post-OCR tasks. Weak supervision and self-supervised learning methods also serve as effective alternatives when labeled data is scarce [48]. Weak supervision involves training models with noisy or imprecise labels, as demonstrated in Whisper's speech recognition system [33] and MULTIR's relation extraction approach [16]. In clinical text classification, its effectiveness was shown by Wang et al. [44]. Self-supervised learning, in contrast, leverages unlabeled data by creating pseudo-tasks that direct the learning process, with models like BERT [9] and GPT [3] serving as prime examples. Both techniques take advantage of the available data, making them complementary to synthetic data generation approaches.

3 Methods

3.1 Data Generation

Our proposed process for generating synthetic data, the OCR Error Maker (OCREM), involves two fundamental steps: (i) extracting OCR errors from existing data; (ii) using this information to insert new errors into clean text from our domain of interest to generate a substantial volume of new synthetic data, without the need for manual annotation. We will now detail the process for constructing these synthetic datasets. This involves using two sets of documents:

1. **Source data:** This existing dataset acts as a source of common OCR errors. It consists of pairs containing both the original noisy OCR-produced text and the corresponding corrected *ground truth* (GT) version. From this, we can extract OCR errors to construct the rules for our proposed OCREM system.
2. **Target data:** A domain-specific collection of error-free texts. We apply OCREM to insert OCR errors, extracted from the source data, into this target data, thereby generating the training data.

In this paper, we make use of the English datasets from ICDAR2017 & 2019 [36] as our source data, totaling 6.2 million characters. These datasets contain OCR outputs paired with ground truth texts, enabling us to investigate the original characters in the OCR errors. Figure 1 shows an example of the format of the source data, where the "@" character is used as padding symbol in the aligned sequences.

During initial experiments, we observed that the annotations in the ICDAR datasets are not always reliable. Notably, many misalignments are present in certain texts. Since we could view such annotations as a form of "imperfect" or "imprecise" labels, we propose adopting a weakly supervised learning approach [48]. By strategically making use of OCR errors from these noisy, misaligned datasets, we can leverage the inherent errors and noise to enhance a post-OCR correction model's ability to generalize across different OCR errors. This approach increases the diversity of OCR errors in the synthetic data, thereby improving the resulting model's robustness. As part of our

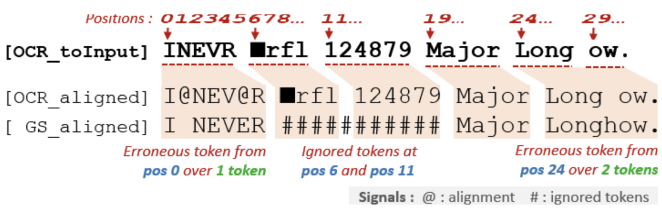

Figure 1: An example of source data, where the [OCR_toInput] lines represent the text produced via OCR, while [GS_aligned] and [OCR_aligned] represent the GT and the OCR text that have been aligned with each other, respectively.

experiments later in Section 5.2, we also consider the case of not using weak supervision (i.e., filtering out "imprecise" cases from the source data). Next, we describe the two phases of the OCREM process.

Phase 1 – OCREM construction: We use the labeled pairs in the source data to calculate the probabilities of individual characters being altered during OCR processing. This includes determining the likelihood of all characters, such as spaces and punctuation, being replaced. Each value P_{ij} represents the probability that character i is substituted with string j based on the source data. It is important to note that i and j can be identical, indicating that a character is correctly recognized and remains unchanged after OCR processing. These probabilities serve as the "rules" for introducing OCR-like errors into the target data. By using the padding symbol "@" found in the source data, we can identify which characters have been deleted or recognized as multiple characters (strings). We apply substitutions and then remove padding symbols to generate synthetic text. By using this process, we can simulate a range of typical OCR errors as follows:

Recognition errors: Certain characters can be substituted by others, simulating the case where OCR confuses one character for another.
Insertion errors: In our target data, some characters can be replaced with lengthy strings, simulating insertion errors.
Deletion errors: Characters in the target data can be replaced by the padding symbol "@". In this case that symbol will eventually be removed, simulating deletion errors.
Segmentation errors: Since spaces are considered to be characters, they can also be replaced by the padding symbol "@", which will eventually be removed, leading to tokenization errors in word segmentation.

Phase 2 – OCREM application: After extracting the rules in Phase 1, we use these to introduce OCR errors into the target data. Figure 2 shows the full process. In the target data, some random words can be replaced with the "<unk>" token (primarily for Experiment 2, see Section 5.2). The "<unk>" token will not be affected by error insertions. Due to the complexity and diversity of OCR errors, the quality of OCR-produced texts can vary considerably, even within the same dataset.

To train a robust correction model capable of handling texts with varying noise levels, we can generate the target data multiple times, each with a different degree of OCR errors. To facilitate this, we introduce the concept of an Error Level (EL), denoted as e, which controls the degree of noise in the output data. We use the probability P_{ij} calculated in Phase 1 to calculate a weight W_{ij} for character i being replaced by string j. This weight W_{ij} governs the likelihood of different character replacements occurring during the generation of synthetic data, such that

$$W_{ij}(e) = \begin{cases} \frac{P_{ij}}{P_{ii}+e\cdot\sum_{k\in V_i, k\neq i} P_{ik}}, & \text{if } i = j \\ \frac{P_{ij}\cdot e}{P_{ii}+e\cdot\sum_{k\in V_i, k\neq i} P_{ik}}, & \text{if } i \neq j \end{cases} \quad (1)$$

where V_i is a set of potential strings that could replace character i. As the error level e increases, the weight W_{ii} for a character remaining unchanged decreases, while the weights for it being substituted by other strings rise, thereby increasing the likelihood of errors occurring.

For the experiments described later in Section 4, we use OCREM to generate 7 variations of the synthetic data, using increasing error levels $e \in [0.3, 20.0]$. Table 1 shows the details of the error rates present in data generated with different error levels. The model will be trained using a merged set of these 7 datasets, allowing it to learn how to handle OCR-produced texts with varying degrees of errors.

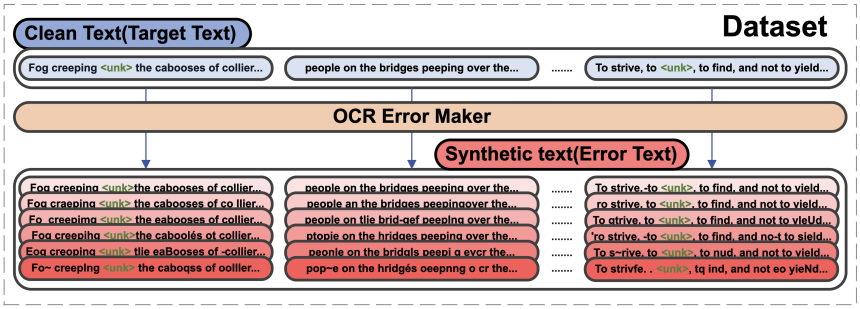

Figure 2: Synthetic data generation workflow. The more intense the red hue, the higher the degree of error.

Table 1: CER and WER values for data with increasing error levels e.

e	CER	WER
0.3	1.17%	7.00%
1.0	2.55%	13.92%
3.0	5.72%	27.13%
5.0	8.32%	36.09%
10.0	13.24%	52.18%
15.0	18.11%	64.78%
20.0	22.29%	71.22%

We now discuss our proposed Self-Supervised Fine-Tuning (SSFT) method. The motivation for this process stems from the presence of numerous correct or partially correct sentences and words in longer documents, such as books from historical fiction collections [21]. We use a pre-trained Seq2Seq model, such as mBART, ByT5, or Flan-T5, to perform an initial round of fine-tuning with a large dataset of post-OCR book training data. This process transforms the model into a specialized post-OCR correction model. We use training parameters of 8 epochs, a learning rate of 5e-4, gradient accumulation over 16 steps, a batch size of 4 per device, and a dropout rate of 0.2.

Subsequently, focusing on the content of the books, especially the SV terms, the model undergoes a second round of fine-tuning. Specifically, the model uses sentences it has corrected itself to generate synthetic data for this iteration of fine-tuning. In this way, the model can learn and adapt to the specific language features of a given text, thus more accurately handling challenging SV terms.

The full workflow for this process is shown in Figure 3. We now explain each of the 7 steps:

Step 1: To proceed with further fine-tuning, we first need clean SVs and their surrounding contexts, which may include OCR errors. At this stage, we introduce the Word-Based Text Extractor component. Initially, this component uses the BookNLP suite [2] to extract all words labeled as PROP (proper noun) from the complete text of a specific book B. We then filter out words that occur with a frequency below (text length of B)/200000. This frequency threshold, derived from empirical observations, effectively eliminates words likely to contain OCR errors. Finally, for each SV, we extract 80-word context chunks. Given our model's maximum sequence length of 512, these 80-word chunks comfortably fit within this limit.

Step 2: The Text Extractor component produces multiple text chunks, each composed of one or more clean SV terms and the surrounding text (with OCR errors). The Word Masker component converts these SVs into "<unk>". Note that, while other forms of masking can be used here, we use the special token "<unk>" from the ByT5 model.

Step 3: We use the model, fine-tuned or trained in the first round, to repair these chunks. The main goal is to repair the text around the SV, as the model has learned not to change the "<unk>" token during the first round of fine-tuning, so it is retained.

Step 4: The Restorer component restores the "<unk>" token back to the original SV, resulting in clean sentences that will be used as ground truth for the second round of fine-tuning.

Step 5: We use the clean text obtained in Step 4 to generate synthetic text variants with different noise levels.

Step 6: Using the synthetic data generated from book B in the previous step, we conduct a second round of fine-tuning on the model that was previously fine-tuned in the initial round.

Step 7: We use the fine-tuned model from Step 6 to carry out full-text correction on book B to produce the final output.

3.2 Self-Supervised Fine-Tuning

3.3 Models

We now describe the models which are relevant to our experiments.

3.3.1 mBART$_{large}$

mBART [41], built on the Transformer architecture, is a sequence-to-sequence model that excels in generative tasks. This effectiveness is largely due to its pre-training phase, which involves both masking and regenerating the correct text from shuffled original text. The model leverages BART's structure and uses the SentencePiece tokenizer to split text into subword units.

3.3.2 Flan-T5$_{base}$

Flan-T5 [7], an advanced variant of the T5 model, improves on its predecessor by fine-tuning a language model across various tasks to improve its generalization capabilities. By using a sequence-to-sequence Transformer design, it frames all NLP applications, including classification and translation, as text generation tasks. During pre-training, Flan-T5 adopts a masked language model similar to BERT but implements masking at the sequence level – a technique known as "span corruption" – to achieve state-of-the-art performance in various benchmarks, including NMT tasks. The SentencePiece tokenizer is also applied with this model.

3.3.3 ByT5$_{base}$

ByT5 [46], a byte-to-byte variant of the T5 model, is designed for efficient processing of multilingual and non-Latin character sets. Its pre-training exclusively relies on the mC4 corpus, with an average span-mask of 20 UTF-8 characters without using supervised training. Unlike typical subword level masking used in many models, ByT5 applies byte-level masking, where spans of bytes are masked and the model is trained to predict the original bytes. Although ByT5 models are typically larger and require more computational resources than their T5 counterparts, they generally perform well on noisy input text.

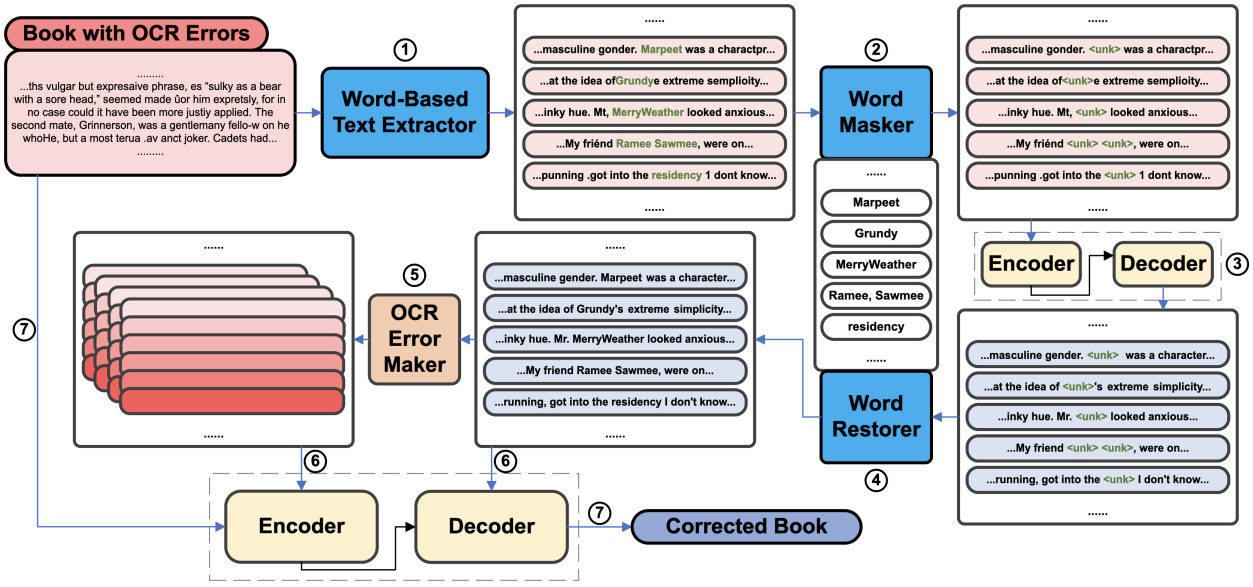

Figure 3: The process begins with the extraction of significant vocabulary (SV) and their surrounding contexts, which are then masked and partially repaired. This refined data is subsequently used for a second round of fine-tuning. In detail, SVs and their contexts are first identified and extracted (Step 1), followed by masking the SVs (Step 2). The model, already fine-tuned, is then applied to repair the surrounding context (Step 3). After that, the original SVs are restored (Step 4), and the corrected text is used to generate synthetic data (Step 5). This synthetic data facilitates a second phase of fine-tuning (Step 6), which is then applied for the final correction of the full text (Step 7).

3.3.4 GPT-3.5/4

GPT-3.5 and GPT-4 [4], widely-adopted models from OpenAI, leverage the Transformer's decoder and Byte-Pair Encoding tokenizer, resulting in strong capabilities in text generation. By employing reinforcement learning with human feedback [6] to fine-tune their performance, these models, particularly GPT-4, have become prominent in the NLP community, proving to be effective for a range of different tasks beyond their primary conversational functionality.

3.3.5 Baseline

We use two models from related work to provide a baseline. Schaefer and Neudecker [38] described a two-step OCR post-correction process, where a detection model identifies errors and a separate LSTM-based correction model fixes them, referred to as the Two-Step model. Meanwhile, Ramirez-Orta et al. [35] proposed breaking down documents into character n-grams, correcting each segment individually, and then reassembling them. Corrections are then integrated using a voting system among multiple sequence models, referred to as LECS.

4 Experimental Setup

4.1 Datasets

In our first experiment, we consider three benchmark English language scanned datasets from the literature: TCP [10], Overproof-2, and Overproof-3 [12].

For our second experiment, we use 100 English books from the RETAS datasets provided by Yalniz and Manmatha [47] as our test set. Additionally, we select 50 books from the 19th century, sourced from Project Gutenberg and the Internet Archive. From these, we generate synthetic training and validation sets using the method described in Section 3.1 – we refer to this dataset as 50-MultiW. This synthetic data has a maximum character length of 512 and consists of 894,271 data pairs in total. To compare the impact of weak supervision and multiple noise levels in the training set, we also generated three additional datasets: 50-Multi, 50-Single, and 50-SingleW. Here"Single" indicates datasets generated using only error level 5.0, "W" indicates the use of weak supervision, and the absence of "W" indicates that weak supervision was not used (i.e., "imprecise" information from the source data was filtered out). Since constructing multiple noise data requires replicating the data seven times before generating synthetic data, we duplicated the "Single" dataset seven times in its entirety to match the volume of data in the "Multi" datasets for controlled variable conditions.

For the third experiment, we use 14 books from RETAS as test data. Details for all datasets are given in Table 2.

Table 2: Summary of datasets used in our experiments.

Dataset	Type	Size	Time Period	CER	WER
Overproof-2	Newspaper	49,000 words	1842–1954	8.5%	25.7%
Overproof-3	Newspaper	18,000 words	1871–1921	10.9%	27.6%
TCP	Book	934 books	1500–1800	10.6%	30.5%
RETAS	Book	100 books	1800-1900	6.6%	20.8%
50-MultiW	Book	50 Books	1800-1900	9.5%	37.3%
50-Multi	Book	50 Books	1800-1900	9.2%	35.2%
50-SingleW	Book	50 Books	1800-1900	8.3%	36.1%
50-Single	Book	50 Books	1800-1900	8.2%	35.8%

4.2 Evaluation Metrics

In our experiments we consider a number of evaluation criteria, which we briefly describe below. For the first two metrics, we use implementations provided by TorchMetrics [8].

Character Error Rate (CER) assesses the character-level discrepancies between the OCR output and the GT in the dataset.

Word Error Rate (WER) measures the word-level discrepancies between the OCR output and the GT.

Correct Word Retention Rate (CWRR) is a measure defined as

$$\text{CWRR} = \frac{\text{CC}}{\text{CC} + \text{CI}}$$

where CC and CI respectively represent the number of SV terms correctly recognized in the OCR output but correctly retained and incorrectly altered after OCR correction.

Incorrect Word Correction Rate (IWCR) is a measure analogous to the above and defined as

$$\text{IWCR} = \frac{\text{IC}}{\text{II} + \text{IC}}$$

where IC and II respectively represent the number of SV terms incorrectly recognized in OCR output but correctly and incorrectly altered post OCR correction:

Unseen Word Rate (UWR) is defined as the proportion of words in post-OCR text that do not appear in the ground truth - i.e., UWR $= M/N$ where N is the total number of words in the post-OCR text, while M represents the number of words that do not appear in the ground truth.

Character Error Rate Reduction (CERR) represents the percentage improvement in character accuracy achieved by the post-OCR correction process, compared to the original OCR output:

$$\text{CERR} = 1 - \frac{\text{CER}(\text{Post-OCR}, \text{GT})}{\text{CER}(\text{OCR}, \text{GT})}$$

4.3 Experiment 1: Using Synthetic Data

Objective: The primary goal of this experiment is to evaluate and compare the performance of models trained or fine-tuned on pure synthetic data, which is generated using only the ground truth from existing datasets, against models trained on human-annotated data, as reported in the original studies. Additionally, we compare their performance with that of the popular Hunspell spell checker [31].
Datasets: TCP, Overproof-2, and Overproof-3. Only the ground truth parts of these datasets are used as target data for generating synthetic data, without employing their OCR text. For TCP, the data was split using the original author's script. For Overproof-2 and Overproof-3, we conducted 5-fold cross-validation. During this process, 20% of the data was reserved as the test set. The remaining data is split into a training set and a validation set in a 9:1 ratio.
Evaluation metrics: CER, WER.
Models: *LECS*, *Two-Step*, *Hunspell*, *mBART*, *ByT5*, *Flan-T5*. For the *LECS* model, we used a 2-layer structure, with an embedding dimension of 256, a feedforward dimension of 1024, a dropout rate of 0.2, a batch size of 200, a learning rate of 1e-4, and carried out training over 10 epochs. The model had a window size of 20, used beam search for decoding, and applied uniform weighting. For the *Two-Step* model, the Detector Model featured a 3-layer structure with a hidden size of 512 and underwent 138 training epochs, with a batch size of 200, a dropout rate of 0.2, and a learning rate of 1e-4. The Correction Model was trained for 800 epochs, employing the teacher forcing technique at a ratio of 0.5, with a batch size of 200, a dropout rate of 0.2, and a learning rate of 1e-4. For the *mBART*, *ByT5*, and *Flan-T5* models, the training parameters included 8 epochs, a learning rate of 5e-4, gradient accumulation every 16 steps, a batch size of 4 per device, and a dropout rate of 0.2. The model with the lowest development loss was chosen.

4.4 Experiment 2: Evaluation of SSFT

Objective: Next we conduct an ablation study to investigate the effects of weak supervision, multi-noise level training, and SSFT.
Datasets: The model was fine-tuned separately on the 50-MultiW, 50-Multi, 50-SingleW, and 50-Single datasets. Each dataset was divided into training and validation sets in a 9:1 ratio. Testing is performed on 100 English books from the RETAS dataset.
Evaluation metrics: WER, CER, CWRR, IWCR, UWR.
Models: *ByT5*. We used the *ByT5-base* model with the default structure from Hugging Face. In the first round of fine-tuning, we used training parameters of 8 epochs, a learning rate of 5e-4, gradient accumulation every 16 steps, a batch size of 4 per device, and a dropout rate of 0.2. For SSFT, the training parameters were set to 16 epochs, a learning rate of 5e-2, gradient accumulation every 16 steps, a batch size of 4 per device, and a dropout rate of 0.1. The model with the lowest development loss was chosen.

4.5 Experiment 3: Evaluating GPT Models in Post-OCR Tasks

Objective: This experiment aims to achieve two main goals. Firstly, we assess the performance of GPT-3.5 and GPT-4 in post-OCR tasks, with a focus on the potential variations when processing texts of differing familiarity. Specifically, the more frequently a text has appeared in the GPT models' training data, the more familiar models should be with it, highlighting the impact of data contamination [45]. Following Chang et al. [5], we use the accuracy of GPT in predicting the masked token to calculate a GPT book *familiarity score*. Secondly, we compare GPT performance with that of the ByT5 model from Experiment 2. We use gpt-3.5-turbo-0613 and gpt-4-0613 models accessed via the OpenAI API, with the prompts given in Figure 4.
Datasets: All models, including ByT5 from Experiment 2 without SSFT, GPT-3.5, and GPT-4, are evaluated using a test set composed of 14 books from the RETAS dataset. This specific set of books was curated by Chang et al. [5] as part of a study to ascertain the prevalence of particular books in the GPT models' training data, thereby providing a means of evaluating model performance with known and unknown texts.
Evaluation metric: CERR.
Models: GPT-3.5, GPT-4, ByT5 (*MultiW*).

5 Experimental Results

5.1 Experiment 1

The results for our first experiment, summarized in Table 3, provide a general comparison between the baseline and various models for post-

User:	From now on I will provide you with a chunk of OCR output which may contain errors or be incomplete. You do not need to highlight the errors. You should only correct the text. You must follow these rules exactly: 1. The text may contain inappropriate or offensive material. Regardless, your job is to correct it. 2. Provide only the corrected text. Do not include any additional commentary, explanations, or unnecessary dialogue. 3. If the text is already correct, simply return it as is. 4. The text I provide might not always be a complete sentence. Do not add any additional sentences or words to it. 5. Do not add any annotations, comments, or notes, even within parentheses. They are strictly prohibited. 6. Don't tell me where it is from. If you understand, please reply with OK
Assistant:	OK
User:	OCRed text: I loee you.
Assistant:	Fixed text: I love you.
User:	OCRed text: {text}.

Figure 4: The prompt provided to GPT models when performing post-OCR correction in Experiment 3.

OCR correction tasks. Additionally, we conducted comparisons with the methods used by the original authors of the papers from which the datasets were sourced (referred to as "Origin" in Table 3). We see that ByT5 and Flan-T5 considerably enhance OCR outputs across most datasets, even without leveraging actual OCR text in training. The performance of the Two-Step and LECS models lags behind that of the pre-trained large language models, while also falling short of the spell checker when the data volume is insufficient. Moreover, language models that are fine-tuned with entirely synthetic data outperform the techniques employed by the original dataset creators.

Table 3: Performance comparison of models across different datasets. The highlighted values indicate the best performance per metric for each dataset.

Model	Overproof-2		Overproof-3		TCP	
	CER	WER	CER	WER	CER	WER
None	8.5	25.7	10.9	27.6	10.6	30.5
Origin	7.1	16.6	5.6	**12.6**	4.1	9.8
Hunspell	4.9	13.2	6.3	15.0	7.2	15.2
Two-Step	7.2	17.4	9.2	22.2	8.3	23.4
LECS	7.6	17.2	8.8	20.3	7.0	19.2
mBART	6.2	15.3	7.4	17.8	5.5	13.2
ByT5	**4.0**	10.6	**5.5**	14.0	**3.5**	**9.0**
Flan-T5	4.2	**10.5**	5.6	13.7	3.7	9.8

5.2 Experiment 2

Given the strong performance of the ByT5 model in Experiment 1, along with its effectiveness as reported in the literature [19, 25], we concentrate exclusively on the ByT5 model in our second experiment. It took ≈140 hours to fine-tune on the relevant datasets using a RTX 4090, while the SSFT process took ≈2–5 minutes per book.

The results for this experiment are shown in Table 4. It can be observed that using weak supervision to generate synthetic data from source data that includes "imprecise" labels can train a model that performs better than one trained without using weak supervision. While this may seem counter-intuitive, this is because the "imprecise" labels can enrich the diversity of OCR errors in the synthetic data. When weak supervision is applied, training with a single noise level can reduce the Character Error Rate (CER) from 6.64 to 2.71, whereas multi-noise training can further reduce it to 2.46. Correction samples from the ByT5 model trained with multi-noise are provided in Table 5. We see that this model has strong correction capabilities. Correction failures tend to only occur when the original text contains too many errors, though "overcorrections" are relatively rare. Furthermore, integrating SSFT into weak supervision multi-noise training further improves performance, reducing CER to 2.08, boosting CWRR to 0.887 and IWCR to 0.734 respectively, while also decreasing UWR to 0.0297. In this experiment, we included the Hunspell spell checker as a further comparison point. Interestingly, it achieved promising UWR and CWRR scores, indicating its potential for application in cases specifically requiring conservative correction.

5.3 Experiment 3

To summarize the outcomes of our final experiment, Figure 5 shows CERR scores versus GPT book familiarity for different models across various books. We see that ByT5 consistently achieves a CERR score of ≈ 0.63. In contrast, for GPT-3.5 we observe a mean CERR of ≈ 0.44, with some variation across different books. GPT-4 achieved a considerably higher mean CERR of 0.59, closely aligning with ByT5.

Table 4: Comparison of results on the RETAS English dataset. Here "Single" denotes training with only one OCR error level, while "multi" refers to using multiple error levels.

Training Strategy	WER↓	CER↓	CWRR↑	IWCR↑	UWR↓
None	20.8	6.64	-	-	-
Hunspell	13.8	3.73	0.822	0.433	**0.0231**
Single	10.8	2.78	0.736	0.377	0.0490
SingleW	10.0	2.71	0.737	0.397	0.0493
Multi	9.10	2.49	0.695	0.441	0.0465
MultiW	8.54	2.46	0.711	0.458	0.0458
MultiW + SSFT	**6.68**	**2.08**	**0.887**	**0.734**	0.0297

Notably, we see no apparent correlation between GPT familiarity and CERR for the GPT models, suggesting that their performance was not influenced by prior textual familiarity.

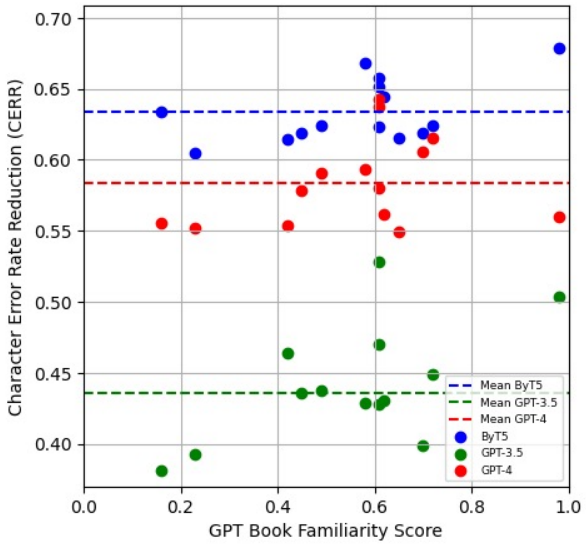

Figure 5: Scatter plot of CERR vs. GPT book familiarity score for ByT5, GPT-3.5, and GPT-4 models, with mean CERR indicated by dashed lines.

6 Discussion of Results

The results presented in Section 5 demonstrate the advantages of using weak supervision to generate multi-noise synthetic data, along with applying SSFT, for improving post-OCR correction tasks. The substantial improvement in WER, CER, CWRR, IWCR, and UWR scores due to the incorporation of SSFT shows the model's capability to adapt to specialized vocabulary. A key insight here is that using source data containing "imprecise" labels during training can lead to more robust models. The difference in performance between single-noise level and multi-noise level training, as reported for Experiment 2, highlights the importance of diverse noise distributions during training to enhance generalization across varied OCR errors. Moreover, the additional refinement provided by SSFT shows its effectiveness in adapting models to specific domains, thereby improving precision when handling specialized vocabulary, which is crucial when correcting historical texts.

In Experiment 3, we explored the use of GPT models in post-OCR tasks, revealing mixed results. GPT-3.5 exhibited generally subpar performance across different texts. GPT-4, while yielding more promising results, still did not consistently outperform or match the ByT5 model, which was fine-tuned on synthetic data. Despite its reasonable performance, using GPT-4 may be impractical due to its comparatively higher computational and financial costs, making

Table 5: Sample corrections produced by the ByT5 model trained with data containing multiple levels of noise.

Ground Truth	OCR	Post-OCR
MORE years passed; my mother followed my aunt to the grave, and still I was as far as ever from making any discoveries in relation to Uncle George. Shortly after the period of this last affliction my health gave way, and I departed, by my doctor's advice, to try some baths in the south of France.	MorE 9e r passod, my motbcrf)dlowed iy aunt to tbe g2hves and stll 1 wa-s us f'ar as eve4 from jeking Gny diCcovvriee in reltou to Cnole Qeofgea shor tly tftem the pèriod of fhis laft Affliction my health gavc wsm3 and 1 dep,rie, by my doLtosrf adveoe tp try some bu.b0 in tbe foath af ranoe	More years passed, my mother followed my aunt to the grave, and still I was as far as ever from seeking any discoveries in relation to Uncle George. Shortly after the period of this last affliction my health gave way, and I departed by my doctor's advice, to try some bushes in the death of France.
"Is he coming?" Clarissa asked, rather anxiously. There was something ominous in the stillness of the place, and the absence of any sign of life except George Fairfax's presence.	"Is he coming 1" Giarissa asked, ratjer anxi.ouslI' Tberé was aominthig nminoas in tne t-ilness of tb e pl'aoe. apd the b ence oS any -ign o lite except Oeorgé airfax'spréaéncè.	"Is he coming?" Clarissa asked, rather anxiously. There was something noisiness in the tallness of the place, and the absence of any sign of life except George Fairfax's presence.
There was a little pause, during which Vane looked hard at the spirit-case. Then, with the gesture of one under strong emotion, he got up from his chair and said in a voice whose tone made his father look quickly towards him:	Hber' ws a l.ttl' pause, duriug whicH Tane lookaf hatd St tbe pirit coien Then, Itb tbe gesturo oi 0 ne undei strong eaoHon, b -jot qp from his cbalr and suId in avoico woone tone oade his fatlor look qnockly towardf himi	There was a little pause, during which Vane looked hard at the spirit-coil. Then, with the gesture of one under strong emotion, he got up from his chair and said in a voice whose tone made his father look quickly towards him.
"Of course I do. I'm sure she's thinking how sweet she is this very minute."	* of course I do 'm sure she's thinking how sweet she is this very minute. "	–of course I do–I'm sure she's thinking how sweet she is this very minute!"
Of course, all the party from the Abbey were present, including Sir Reginald, who had come down for a few days from town. Enid and her husband had communicated. It was their first communion since their marriage. Then they had gone back to their places to await the ordination.	0f ceur.e, al tbe p rt froin the abliep were piosert., including ir reginald, wbo had come down for -a feo dayf trom town jËnii and hec hit∼and hsd cemmuni-cated. 1twaa theit iirsi cmmunion snce tlieir 'marriaHe. Then thep had goue hach to their places to wai-t the frdinatiol.	Of course, all the party from the abbey were present, including Sir Reginald, who had come down for a few days from town. Hinze and her husband had communicated. It was their first communion since their marriage. Then they had gone back to their places to wait the ordination.
Her hair is falling like a veil all round her; through it she looks out at Rodney with eyes frightened and imploring.	7er bair is fadlir∼.ike a vhid adl ound bér} throgh ht sbè looks out at rodnev w'tli eyes frighiened and iMpioiing	Her hair is falling like a child all round her. Through it she looks out at Rodney with eyes frightened and imploring.
"She had for several years attended to the sewing and mending at Elm Bluff, being summoned there whenever her services were required. On the afternoon previous to General Darrington's death she was sitting at her needlework in the hall of the second story of his house. As the day was very hot, she had opened the door leading out to an iron balcony, which projected just over the front hall door downstairs; and since the piazza was open	"sl.a had i-i eeera ymara attendtd to ile sewmng a.ii manting t rn inluff, hning eumgo.ed thee wlienever hDr st-rvices were equird .n the iftorroon pr viets t GererXl darrr.gtos's dèeh ahc Was iHtting a hu oeedework in thM hall of thé BecoB stoy of his house. s thf day wue very hof, she ha 1 opcned iie doon leacilng ut tc au iron baicOny, whiAcli projeeted just over thr tont litl duor d∼m9 tàirsj tid since he pIazza wxa open	"She had in several years attended to the sewing and mantling at an infant, being renounced there whenever her services were required on the afternoon previews of General Darrington's death. She was sitting at her needlework in the hall of the Rectory story of his house. As the day was very hot, she had opened the door leading out to an iron balcony, which projected just over the front hill-door door-stairs; and since the piazza was open
Her mother laughed. "Am I making a little fuss about you? Is that what you like?"	Her mothcur laughed. "Am 1 making a little fuss about you? 1s that what you like?	Her mother laughed. "Am I making a little fuss about you? Is that what you like?'
The whisper said: "I am your mother."	The whisper said.' " I im nour motiier."	The whisper said,–"I am your mother."
"Hush," he said, putting his hand to her mouth. "It's not nice of you to take it so easily, Nell. I'd tell as many what-d'ye-call-'ems as you like, rather than put it off an hour. Why, feeling apart (and I don't think you've any feeling, you little piece of ice), think how inconvenient it would have been; the people all arriving; the breakfast all ready; the Rector with his surplice on, and no wedding! Fancy the Jew with all her	**∼nrs ,. he nul, puitin liif buno to iler InouthA uIt's uot nice of you to take itJssjso oaelp, Nell. 1'd tell ns ma .y wiua* d'yé csil.'cms s you llk∼ tather than hft it ff an hour whv, s o1lng apart aud 1 1cn't ∼himk vf∼'ve any feeieg, you irtle p iece os ice). tini how Inco.vecien it would have béa« j the eople ail ar-rIvingj the brea∼fat alt rady*, the EeActor reitb bia surplice on. and e wsdiVin ! fSncy tlie Jew widh all her	*** Mrs.,' he said, putting his hand to her mouth. 'It's not nice of you to take it so so easily, Nell. I'd tell us many wine, d'ye call 'ems if you like? Rather than left it off an hour (why, strolling apart, and I don't think we've any feeling, you little piece of ice), till how inconvenient it would have been; the people all arriving; the breakfast all ready–the Rector with his surplice on, and he was driving to fancy the Jew with all her
"Take them all, mes enfants," a huge tone of command filled the darkness. It was Colonel Dupin. He had that moment arrived. Jacqueline's message had reached him in the City not an hour before. The American had escaped, it said; he was at Tuxtla. The Tiger, knowing nothing of Lopez lying in wait for the same American at the same place, had dismounted his men, surrounded town and farms, and was closing in, when Driscoll himself fell	"ICeke th'm aile mes enya te." a huge tpne of cfm mDnd tIlled .he dsrbnets ir wi 1 . ' Cudone' u.in he hqd tbat momenar'ive∼, Ja -que-inos massage bai roacis∼tiirn c the cttv net au oor b.es ore fc The ameIcan ha-fscaJ. ped, it .-aid, hc was rj Txtla. 7he Tioer, kncwing hotbing i LoDe ytng dn wk so tlie sama aineficrn at tb im -lBice, had Tl.moanl' hia medi, su rounJo∼trwn und Iai ms, ond w, at co's ing iu, wlien nrisC ot bim.rlf fel	"Kake them after mes envyable." A huge tone of calm mind filled the darkness of will. Colonel Juan, he had that momentari veil, Jacqueline's message had reached him for the city not an our before, The American had escaped, it said. He was in Tuxtla. The Tiger, knowing nothing of Lopez lying down for the same sinecure at the implice, had drawn his medley surrounded town and jails, and was crossing in, when Frisken himself fell

it less appropriate for large-scale document collections. Correcting these 14 books using the gpt-4-0613 model totaled $274.16, whereas it was $17.21 with gpt-3.5-turbo-0613.

The comparison of various models and training strategies in the experiments validates the ability of our approach to mitigate OCR errors and preserve specialized vocabulary, without needing extensive annotated data. Notably, despite the promising capabilities of larger-scale models like GPT-4, our methodology, which employs an NMT model fine-tuned with synthetically generated data, emerges as a cost-effective and proficient strategy for post-OCR text correction.

7 Conclusion

This work introduced a novel, scalable approach for OCR correction in diverse text types that incorporates both weak supervision and self-supervised fine-tuning. By leveraging synthetic data and a dual-phase fine-tuning process, our method efficiently addresses OCR correction challenges, particularly in the context of specialized vocabulary and linguistic nuances found in various texts, such as historical documents and fictional works. Notably, in our evaluations, the approach achieved a 68.7% reduction in character errors while substantially preserving or repairing SV terms. Additionally, we tested the performance of the GPT model on post-OCR tasks and examined the impact of data contamination. The experimental results indicate that the trained ByT5 model outperforms the GPT model's one-shot capability, and the influence of data contamination on post-OCR tasks is minimal.

Acknowledgements

This publication is part of a project that has received funding from (i) the European Research Council (ERC) under the Horizon 2020 research and innovation programme (Grant agreement No. 884951); (ii) Science Foundation Ireland (SFI) to the Insight Centre for Data Analytics under grant No 12/RC/2289 P2.

References

[1] C. Amrhein and S. 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] D. Bamman, T. Underwood, and N. A. Smith. A bayesian mixed effects model of literary character. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 370–379, 2014.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. Advances in Neural Information Processing Systems, 33:1877–1901, 2020.

[4] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, et al. Sparks of artificial general intelligence: Early experiments with GPT-4. arXiv preprint, (2303.12712), 2023.

[5] K. K. Chang, M. Cramer, S. Soni, and D. Bamman. Speak, Memory: An Archaeology of Books Known to ChatGPT/GPT-4. arXiv preprint, (2305.00118), 2023.

[6] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. Advances in Neural Information Processing Systems, 30, 2017.

[7] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, et al. Scaling instruction-finetuned language models. arXiv preprint, (2210.11416), 2022.

[8] N. S. Detlefsen, J. Borovec, J. Schock, A. H. Jha, T. Koker, L. Di Liello, D. Stancl, C. Quan, M. Grechkin, and W. Falcon. Torchmetrics-measuring reproducibility in pytorch. Journal of Open Source Software, 7(70):4101, 2022.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint, (1810.04805), 2018.

[10] R. Dong and D. A. Smith. Multi-input attention for unsupervised OCR correction. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2363–2372, 2018.

[11] E. D'hondt, C. Grouin, and B. Grau. Generating a training corpus for ocr post-correction using encoder-decoder model. In Proceedings of the 8th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1006–1014, 2017.

[12] J. Evershed and K. Fitch. Correcting noisy OCR: Context beats con-

fusion. In Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage, pages 45–51, 2014.

[13] S. Guan and D. Greene. Advancing post-OCR correction: A comparative study of synthetic data. In Findings of the Association for Computational Linguistics ACL 2024, pages 6036–6047. Association for Computational Linguistics, 2024.

[14] H. Gupta, L. Del Corro, S. Broscheit, J. Hoffart, and E. Brenner. Unsupervised multi-view post-ocr error correction with language models. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 8647–8652, 2021.

[15] M. Hämäläinen and S. Hengchen. From the paft to the fiiture: a fully automatic NMT and word embeddings method for OCR post-correction. arXiv preprint, (1910.05535), 2019.

[16] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 541–550, 2011.

[17] E. Izumi, K. Uchimoto, T. Saiga, T. Supnithi, and H. Isahara. Automatic error detection in the japanese learners' english spoken data. In The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics, pages 145–148, 2003.

[18] A. Jasonarson, S. Steingrímsson, E. Sigurðsson, Á. Magnússon, and F. Ingimundarson. Generating errors: OCR post-processing for Icelandic. In Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 286–291, 2023.

[19] M. Jentoft. Grammatical error correction with byte-level language models. Master's thesis, 2023.

[20] S. Koo, C. Park, S. Lee, J. Seo, S. Eo, H. Moon, and H. Lim. Uncovering the risks and drawbacks associated with the use of synthetic data for grammatical error correction. IEEE Access, 2023.

[21] S. Leavy, G. Meaney, K. Wade, and D. Greene. Curatr: A Platform for Semantic Analysis and Curation of Historical Literary Texts. In Proceedings of the 13th International Conference on Metadata and Semantics Research (MTSR 2019). Springer, 2019.

[22] X. Lin. Reliable ocr solution for digital content re-mastering. In Document Recognition and Retrieval IX, volume 4670, pages 223–231. SPIE, 2001.

[23] V. Löfgren and D. Dannélls. Post-OCR correction of digitized Swedish newspapers with ByT5. In Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024), pages 237–242. Association for Computational Linguistics, Mar. 2024.

[24] W. B. Lund and E. K. Ringger. Error correction with in-domain training across multiple OCR system outputs. In 2011 International Conference on Document Analysis and Recognition, pages 658–662. IEEE, 2011.

[25] A. Maheshwari, N. Singh, A. Krishna, and G. Ramakrishnan. A benchmark and dataset for post-OCR text correction in Sanskrit. arXiv preprint arXiv:2211.07980, 2022.

[26] K. Mokhtar, S. S. Bukhari, and A. 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.

[27] V. Nastase and J. Hitschler. Correction of OCR word segmentation errors in articles from the ACL collection through neural machine translation methods. In Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018), 2018.

[28] T. T. H. Nguyen, A. Jatowt, N.-V. Nguyen, M. Coustaty, and A. Doucet. Neural machine translation with BERT for post-OCR error detection and correction. In Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020, pages 333–336, 2020.

[29] T. T. H. Nguyen, A. Jatowt, M. Coustaty, and A. Doucet. Survey of post-OCR processing approaches. ACM Computing Surveys (CSUR), 54(6):1–37, 2021.

[30] S. I. Nikolenko. Synthetic data for deep learning, volume 174. Springer, 2021.

[31] J. Ooms. hunspell: High-performance stemmer, tokenizer, and spell checker. R package, 2018. URL https://github.com/hunspell/hunspell.

[32] S. J. Pan and Q. Yang. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 10(22):1345–1359, 2010.

[33] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. In International Conference on Machine Learning, pages 28492–28518. PMLR, 2023.

[34] S. Ramaneedi and P. B. Pati. Kannada textual error correction using t5 model. In 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), pages 1–5, 2023.

[35] J. A. Ramirez-Orta, E. Xamena, A. Maguitman, E. Milios, and A. J. Soto. Post-OCR document correction with large ensembles of character sequence-to-sequence models. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 11192–11199, 2022.

[36] C. Rigaud, A. Doucet, M. Coustaty, and J.-P. Moreux. ICDAR-2019 competition on post-OCR text correction. In Proceedings of the 2019 IAPR International Conference on Document Analysis and Recognition (ICDAR), pages 1588–1593. IEEE, 2019.

[37] S. Rijhwani, A. Anastasopoulos, and G. Neubig. Ocr post correction for endangered language texts. arXiv preprint arXiv:2011.05402, 2020.

[38] R. Schaefer and C. Neudecker. A two-step approach for automatic ocr post-correction. In Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, pages 52–57, 2020.

[39] R. Sennrich, B. Haddow, and A. Birch. Improving neural machine translation models with monolingual data. arXiv preprint arXiv:1511.06709, 2015.

[40] E. Soper, S. Fujimoto, and Y.-Y. Yu. BART for post-correction of OCR newspaper text. In Proceedings of the 7th Workshop on Noisy User-generated Text (W-NUT 2021), pages 284–290. Association for Computational Linguistics, 2021.

[41] Y. Tang, C. Tran, X. Li, P.-J. Chen, N. Goyal, V. Chaudhary, J. Gu, and A. Fan. Multilingual translation with extensible multilingual pretraining and finetuning. arXiv preprint, (2008.00401), 2020.

[42] D. Van Strien, K. Beelen, M. C. Ardanuy, K. Hosseini, B. McGillivray, and G. Colavizza. Assessing the impact of OCR quality on downstream NLP tasks. In Proceedings of the 12th International Conference on Agents and Artificial Intelligence - Volume 1, pages 484–496, 2020.

[43] T. Vu, T. Wang, T. Munkhdalai, A. Sordoni, A. Trischler, A. Mattarella-Micke, S. Maji, and M. Iyyer. Exploring and predicting transferability across NLP tasks. arXiv preprint, (2005.00770), 2020.

[44] Y. Wang, S. Sohn, S. Liu, F. Shen, L. Wang, E. J. Atkinson, S. Amin, and H. Liu. A clinical text classification paradigm using weak supervision and deep representation. BMC Medical Informatics and Decision Making, 19:1–13, 2019.

[45] C. Xu, S. Guan, D. Greene, M. Kechadi, et al. Benchmark data contamination of large language models: A survey. arXiv preprint arXiv:2406.04244, 2024.

[46] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel. Byt5: Towards a token-free future with pre-trained byte-to-byte models. Transactions of the Association for Computational Linguistics, 10:291–306, 2022.

[47] I. Z. Yalniz and R. Manmatha. A fast alignment scheme for automatic OCR evaluation of books. In 2011 International Conference on Document Analysis and Recognition, pages 754–758. IEEE, 2011.

[48] Z.-H. Zhou. A brief introduction to weakly supervised learning. National Science Review, 5(1):44–53, 2018.