BART for Post-Correction of OCR Newspaper Text

Elizabeth Soper[1]
University at Buffalo, SUNY
esoper@buffalo.edu

Stanley Fujimoto
Ancestry.com
sfujimoto@ancestry.com

Yen-Yun Yu
Work Done at Ancestry.com
yen-yun.yu@mailfence.com

Abstract

Optical character recognition (OCR) from newspaper page images is susceptible to noise due to degradation of old documents and variation in typesetting. In this report, we present a novel approach to OCR post-correction. We cast error correction as a translation task, and fine-tune BART, a transformer-based sequence-to-sequence language model pretrained to denoise corrupted text. We are the first to use sentence-level transformer models for OCR post-correction, and our best model achieves a 29.4% improvement in character accuracy over the original noisy OCR text. Our results demonstrate the utility of pretrained language models for dealing with noisy text.

1 Introduction

Newspapers capture daily life from a moment in history. Their contents may not ever be published in history books, but they are rife with stories about everyday people, and connecting people to these stories empowers journeys of personal discovery. Connecting people to interesting and relevant articles within a newspaper is often challenging because newspaper pages are stored as scanned images. Computer vision models have been developed to automatically separate pages into individual articles (via object recognition) and convert each article from image to text (via OCR). Still, because of the degradation of old documents and variation in typesetting, the resulting text often contains errors, which can cause problems for downstream applications of the data.

The field of Natural Language Processing (NLP) has undergone a rapid shift in the last several years due to the popularization of transformers as a new, powerful tool for language modeling (Vaswani et al., 2017). Many pretrained models like BERT (Devlin et al., 2019) and GPT (Radford et al., 2019) have advanced the state-of-the-art on numerous NLP tasks, from text classification to question answering and translation. The key to the success of these models is their flexibility; pretraining on generic tasks like masked language modelling or next sentence prediction gives models generalized language knowledge, which then allows them to be easily adapted to more specific tasks (Brown et al., 2020). While pretrained language models are highly adaptable to new tasks, they have been shown to be very sensitive to noise. Models like BERT break down easily in the presence of irregular spellings or errors in text (Kumar et al., 2020). This work shows that in fact this type of pretrained language model can actually be used to correct noise.

In this work, we test the utility of transformer models to automatically correct noisy text generated by OCR. Specifically, we fine-tune BART (Lewis et al., 2020) on the ICDAR 2017 Post-OCR Correction dataset (Chiron et al., 2017). Unlike most previous approaches, fine-tuning BART does not require character-level alignment of the training data. Our results indicate that the generalized knowledge attained by BART during pretraining is enough to perform well at OCR text correction with just a small amount of task-specific fine-tuning. Our best model improves the character accuracy of the evaluation set by 29.4%. Correcting OCR errors is a challenging task, but significantly impacts performance on any downstream use of the data.

The following report is organized as follows: in Section 2 we describe previous work on OCR post-correction, in Section 3 we outline the BART model architecture and present our methodology for fine-tuning the model, in Section 4 we report the results of our experiments, and finally we conclude in Section 5.

[1]Work completed during an internship at Ancestry.com

Proceedings of the 2021 EMNLP Workshop W-NUT: The Seventh Workshop on Noisy User-generated Text, pages 284–290

2 Previous Work

OCR post-correction has been an important problem since the inception of OCR technology. Traditional approaches to OCR post-correction relied on n-gram or dictionary-based techniques (Kukich, 1992). Recently, neural methods have become more popular. The International Conference on Document Analysis and Recognition (ICDAR) has held two competitions for OCR post-correction. The first, in 2017, was dominated by statistical and neural machine translation approaches (Chiron et al., 2017). The winner of the competition was Amrhein and Clematide (2018), who still hold the current state-of-the-art on the ICDAR 2017 benchmark. They use an ensemble of character-based statistical and neural machine translation models. The second ICDAR competition for OCR post-correction was held in 2019, after the introduction of transformer-based language models. The winning team, from Clova AI, used BERT embeddings as input to train CNN classifier, then character-level sequence-to-sequence (biLSTM) for correction (Rigaud et al., 2019).

Since then, there has been continued interest in applying pretrained language models to the task of text correction. Nguyen et al. (2020) use BERT embeddings to train an error detection network, and then apply character-level NMT for correction. BERT has also been used in the correction of errors from other sources; Zhang et al. (2020) correct errors generated by human typos. They use pretrained BERT embeddings as input to a biGRU to detect errors, and then fine-tune BERT on a masked language modeling task to correct the sentence, where the errors detected in the first step are soft-masked. Hu et al. (2020) use BERT embeddings plus edit distance between the errors and candidate replacements to correct pre-identified errors.

The above studies demonstrate the potential for pretrained language models in text correction. The present study builds on previous work by testing the potential of this type of model to handle text correction on their own in a single step, without additional infrastructure. We were interested in whether the generalized language knowledge acquired during pretraining could be directly transferred to OCR post-correction with a small amount of task-specific fine-tuning. To test this, we use BART, a sequence-to-sequence transformer model pretrained on text denoising, and fine-tune it on the ICDAR 2017 dataset (Chiron et al., 2017). This approach is novel in that error detection and correction are handled together in one step, and the model uses full sentences rather than performing character-level translation. To our knowledge, this is the first time that BART has been applied to the task of OCR post-correction. Our results suggest that BART can in fact be satisfactorily tuned as a stand-alone text correction model.

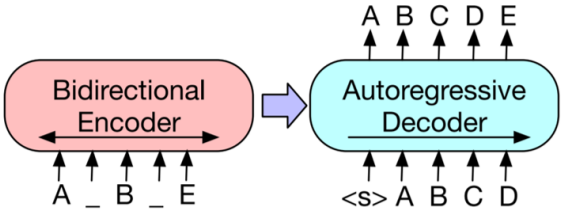

Figure 1: The BART architecture (taken from Lewis et al. 2020). Corrupted text is input to a bidirectional encoder (left), and then the likelihood of the original text is calculated by an auto-regressive decoder (right).

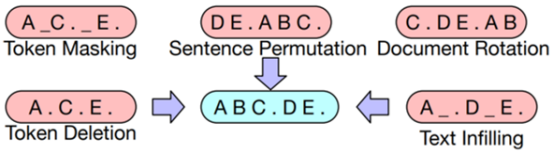

Figure 2: Transformations used for noising text in pretraining (taken from Lewis et al. 2020).

3 Methods

Next we outline the original BART model, followed by the methods used to fine-tune it for OCR post-correction.

3.1 BART model

Bidirectional and Auto-regressive Transformers (BART) is a sequence-to-sequence language model pretrained on a variety of denoising tasks to acquire general knowledge about how language works from large amounts of training data (Lewis et al., 2020). The BART architecture is shown in Figure 1; it consists of a bidirectional encoder (as in BERT: Devlin et al. 2019) plus an auto-regressive decoder (as in GPT: Radford et al. 2019). BART has achieved new state-of-the-art results on several NLP benchmarks, including dialogue response generation, question answering, and summarization.

One of the key differences between BART and other popular pretrained language models is its training objective. Where most language models are trained on a masked token prediction or next

OCR Error Type	Example	Corresponding BART Pretraining Objective
Over-segmentation	indecent → in decent	Text Infilling
Under-segmentation	and just → andjust	Token Deletion
Misrecognized Character	into → ipto	–
Missing Character	what → hat	Token Deletion
Hallucination	(empty string) → %_a_q$	Text Infilling

Table 1: Mapping of common OCR error types to BART's pretraining objectives.

token generation task, BART is trained to reconstruct text which has been corrupted in a variety of ways. BART's pretraining objective can be seen as a generalization of masked language modelling: in addition to token masking, input may also undergo a combination of token deletion, text infilling, sentence permutation, and document rotation. The set of text noising transformations used in BART pretraining is illustrated in Figure 2. By using a broader range of methods for corrupting input text, BART becomes more robust to noise. Adding noise to training data has been shown to improve model performance across many domains, even outside of NLP. In computer vision, for example, augmenting image data with color jitter or random erasing during training improves models for image classification and object detection (Shorten and Khoshgoftaar, 2019).

For our purpose of correcting errors in OCR-generated text, pretraining BART on various types of noisy data is particularly advantageous. First, because of the similarity between the types of text corruption seen during pretraining and the corruption introduced by OCR, BART theoretically could be fine-tuned for OCR-specific error types with a relatively small set of examples compared with state-of-the-art. Table 1 shows the most common types of errors introduced by OCR, and maps each to the closest corresponding BART pretraining objective. In particular, BART's token deletion and text infilling tasks are suited to OCR correction, as OCR frequently either misses characters or injects spurious characters. Notably, however, BART's pretraining involved deletion or insertion of entire tokens, whereas in OCR data it's more common to for individual characters to be deleted or inserted, rather than entire words.

Another important feature of BART which makes it particularly suitable for OCR correction is that *input* to the encoder need not be aligned with the decoder *output*. This is ideal for dealing with errors that result in a different number of tokens between the source and target texts, such as over- or under-segmentation, missing characters, or hallucinated characters, which are very frequent in OCR text. Dealing with the alignment of noisy and gold-standard texts is a non-trivial issue in creating OCR correction models, especially when creating training data. By using BART, we bypass the alignment problem.

3.2 Dataset

BART's pretraining corpus includes all of English Wikipedia plus the BookCorpus, which represents a wide range of genres. To fine-tune BART for OCR post-correction, we use the ICDAR 2017 Post-OCR Correction Dataset (Chiron et al., 2017). The data is a mixture of historical newspaper and monograph texts, ranging in date from 1744 to 1911. The dataset includes French and English language texts, but as BART was pretrained on English data only, French texts were removed for fine-tuning. The remaining English data contains 38,975 training sentences (27,414 monograph, 11,561 periodical) and 7,759 evaluation sentences (3,966 monograph, 3,793 periodical).

3.3 Fine-tuning Methods

We use Huggingface's transformers package for fine-tuning (Wolf et al., 2020). To prepare the data for input to the model, we split each article into sentences and removed the special alignment characters. Each sentence is tokenized with BART's tokenizer, which uses byte-level Byte Pair Encoding. The tokenized data is passed to the model one full sentence at a time.

The 'bart-base' checkpoint is our starting point for fine-tuning. The model is trained using batch size 6, AdamW optimizer, and cross-entropy loss between the model output and the target text to update weights. 3 epochs of training took approximately 2hr to train on a Tesla V100 SXM2 GPU

Error Type	Source	Prediction	Target
Over-segmentation	...before the **follow ing** morning...	...before the **follow- ing** morning...	...before the **follow- ing** morning...
Under-segmentation	Two Closes of Rich **oldSwarth** LAND, adjoining each other...	Two Closes of Rich **old Swarth** LAND, adjoining each other...	Two Closes of Rich **old Swarth** LAND, adjoining each other...
Misrecognized character	We sailed from Kalaniita Bay, **ar.d** soon we made the coast	We sailed from Kalaniita Bay, **and** soon we made the coast	We sailed from Kalamita Bay, **and** soon we made the coast
Missing character	**er** a good deal of argument, the facts were agreed to be turned **ipto** a special case	**After** a good deal of argument, the facts were agreed to be turned **into** a special case	**After** a good deal of argument, the facts were agreed to be turned **into** a special case
Hallucination	**248 THE FAMOUS niSTORY How** Fryer Bacon burnt his books of Magick	Fryer Bacon burnt his books of Magick	Fryer Bacon burnt his books of Magick

Table 2: Example results for different error types.

with 16GB RAM.

4 Results

Next we discuss the results of fine-tuning BART for OCR post-correction. Our model can handle a wide variety of error types, and improves the overall text accuracy by 29.4% on the evaluation set.

4.1 Qualitative Analysis

There are five main types of errors produced by OCR: (1) over-segmentation, (2) under-segmentation, (3) misrecognized character, (4) missing character, (5) hallucination. After fine-tuning, BART can recognize and correct all five types of errors.

Table 2 shows examples of how our model handles each type of error. For over-segmentation errors, our model corrects errors by adding a hyphen to the initial word fragment, rather than deleting the space and writing as a single word. This reflects the way the target text was formatted. These errors are typically due to words being broken across lines to fit within the margins.

We also found that the proposed model is less consistent at recognizing under-segmentation errors, most likely because it is less common in the ICDAR data, as well as the fact that deleting spaces has no direct parallel in BART's pretraining; the token deletion seen during pretraining always consisted of multi-character tokens.

For misrecognized character errors, the model consistently corrects the misspelling of common words, but struggles to correct misspelled proper names. Since proper names generally have very idiosyncratic spellings, they are much harder to predict, so the fact that our model struggles to correct unusual names is not a surprise.

In the example of an error involving missing characters, our model successfully recovers the missing character to complete the first word of the sentence. Note that in this example, the model also identifies and corrects a misrecognized character later in the sentence. This demonstrates the model's ability to automatically correct an arbitrary number of errors in a given input, without additional input information about the number or location of errors.

Last, in the hallucination error example, where there are characters which have no equivalent in the target text, our model correctly deletes the spurious characters. Hallucinated characters in the dataset are often non-alphabetic symbols (as shown in Table 1), which may make learning to identify such errors easier, but as shown in the example in Table 2, the model is able to recognized hallucinated alphabetic characters as well.

Approach	Monograph	Periodical	Overall
Char-SMT/NMT (Amrhein and Clematide, 2018)	**43**	**37**	**40**
CLAM (Chiron et al., 2017)	29	22	26
EFP (Chiron et al., 2017)	13	0	7
MMDT (Schulz and Kuhn, 2017)	20	0	10
WFST-PostOCR (Chiron et al., 2017)	28	0	14
Coustaty et al, 2018 (Coustaty et al., 2018)	30	10	20
Nguyen et al, 2020 (Nguyen et al., 2020)	36	27	31
BART w/ no fine-tuning	-7	-9	-8
Fine-tuned BART	32	23	29

Table 3: Model comparison on the English ICDAR 2017 dataset. Percent improvement is reported for the Monograph and Periodical sections, as well the overall improvement.

4.2 Quantitative Analysis

Performance on OCR post-correction is standardly measured by percent improvement between the source text and model prediction. This is calculated as shown in Equation (1), where $dist(\cdot)$ is the Levenshtein distance between two strings, s is the noisy source text, p is the predicted text from the model, and t is the target text.

$$\%_{improvement} := \frac{dist(s,t) - dist(p,t)}{len(t)} \quad (1)$$

Off-the-shelf BART with no fine-tuning does poorly at correcting errors, resulting in a 7.6% *decrease* in character accuracy. Our best fine-tuned model, on the other hand, achieved 29.4% improvement in text accuracy on the evaluation set, with 32.2% improvement on monographs and 23.1% improvement on periodicals. The disparity in performance between monographs and periodicals can be seen across other previous approaches, and is likely due to the imbalance in the training data, and possibly to the higher difficulty of the periodical genre.

The 37 point improvement in performance resulting from fine-tuning is remarkable given the relatively small size of the task-specific training set. The poor performance of BART before fine-tuning is somewhat surprising, given the similarity between the pretraining and fine-tuning tasks; one possible explanation is that during pretraining entire tokens were masked, inserted, or deleted, whereas during fine-tuning the errors were generally at the individual character level. Thus, while pretraining was not enough to yield strong results on OCR correction, the model could adapt quickly to the task because the only difference was the average span size of corrupted text.

A comparison of fine-tuned BART with previous approaches is shown in Table 3. For the sake of direct comparison, only approaches which reported evaluation metrics on the ICDAR 2017 dataset are included in the table. Fine-tuning BART on different datasets would allow for direct comparisons with other approaches (e.g. the Chinese typo dataset from Zhang et al. 2020); we leave this to future work. Overall, while the Amrhein and Clematide (2018) model outperforms ours, our approach is arguably *simpler*, as it requires no alignment information to train and performs both detection and correction in a single step. Furthermore, several additional techniques, like artificially injecting additional noise into training data, and hyperparameter optimization, could likely improve BART's performance even further.

5 Conclusion

In this work, we explored the use of BART for automatic correction of noisy text data generated by OCR. Our best model achieves 29.4% improvement in text accuracy. The resulting improvement in OCR text quality impacts all subsequent applications of the data. The approach described above is not specific to errors introduced by OCR; the same methods could in principle be applied to correct any type of noisy data, from the output of handwriting recognition models, to human-generated typos. Future work might test the utility of fine-tuning BART on a wider range of noisy text types, including fine-tuning mBART for correction of non-English data, as well as comparing BART with other sequence-to-sequence language models on text correction. Error correction is a critical first step for any NLP task when working with noisy data sources, and thus remains an important problem for the field.

Acknowledgements

In addition to the listed authors, Suraj Subraveti, Michael Brodie, Brent Carter, and Craig Whatcott also made important contributions to the work described in this report. We also thank the anonymous reviewers for their helpful feedback.

References

Chantal Amrhein and Simon Clematide. 2018. 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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel HerbertVoss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, , and Dario Amodei. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Guillaume Chiron, Antoine Doucet, Mickaël Coustaty, and Jean-Philippe Moreux. 2017. ICDAR 2017 competition on post-OCR text correction. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1423–1428. IEEE.

Mickaël Coustaty, Antoine Doucet, Adam Jatowt, Nhu-Van Nguyen, et al. 2018. Adaptive edit-distance and regression approach for post-OCR text correction. In *International Conference on Asian Digital Libraries*, pages 278–289. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *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, Minneapolis, Minnesota. Association for Computational Linguistics.

Yifei Hu, Xiaonan Jing, Youlim Ko, and Julia Taylor Rayz. 2020. Misspelling correction with pre-trained contextual language model. In *2020 IEEE 19th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pages 144–149. IEEE.

Karen Kukich. 1992. Techniques for automatically correcting words in text. *ACM Computing Surveys (CSUR)*, 24(4):377–439.

Ankit Kumar, Piyush Makhija, and Anuj Gupta. 2020. Noisy text data: Achilles' heel of BERT. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 16–21. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880. Association for Computational Linguistics.

Thi Tuyet Hai Nguyen, Adam Jatowt, Nhu-Van Nguyen, Mickael Coustaty, and Antoine Doucet. 2020. 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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *Technical Report*.

Christophe Rigaud, Antoine Doucet, Mickaël Coustaty, and Jean-Philippe Moreux. 2019. ICDAR 2019 competition on post-OCR text correction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1588–1593. IEEE.

Sarah Schulz and Jonas Kuhn. 2017. Multi-modular domain-tailored OCR post-correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2716–2726, Copenhagen, Denmark. Association for Computational Linguistics.

Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling error correction with soft-masked BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 882–890. Association for Computational Linguistics.