

Automatic Collation for Diversifying Corpora: Commonly Copied Texts as Distant Supervision for Handwritten Text Recognition

David A. Smith¹, Jacob Murel¹, Jonathan Parkes Allen² and Matthew Thomas Miller²

¹*Khoury College of Computer Sciences, Northeastern University, Boston MA, U.S.A.*

²*Roshan Institute for Persian Studies, University of Maryland, College Park MD, U.S.A.*

Abstract

Handwritten text recognition (HTR) has enabled many researchers to gather textual evidence from the human record. One common training paradigm for HTR is to identify an individual manuscript or coherent collection and to transcribe enough data to achieve acceptable performance on that collection. To build generalized models for Arabic-script manuscripts, perhaps one of the largest textual traditions in the pre-modern world, we need an approach that can improve its accuracy on unseen manuscripts and hands without linear growth in the amount of manually annotated data. We propose Automatic Collation for Diversifying Corpora (ACDC), taking advantage of the existence of multiple manuscripts of popular texts. Starting from an initial HTR model, ACDC automatically detects matching passages of popular texts in noisy HTR output and selects high-quality lines for retraining HTR without any manually annotated data. We demonstrate the effectiveness of this approach to distant supervision by annotating a test set drawn from a diverse collection of 59 Arabic-script manuscripts and a training set of 81 manuscripts of popular texts embedded within a larger corpus. After a few rounds of ACDC retraining, character accuracy rates on the test set increased by 19.6% absolute percentage, while a supervised model trained on manually annotated data from the same collection increased accuracy by 15.9%. We analyze the variation in ACDC’s performance across books and languages and discuss further applications to collating manuscript families.

Keywords

handwritten text recognition, collation, manuscripts


1. Introduction


Within the past decade, widely-available handwritten text recognition (HTR) tools have enabled many disciplines to investigate a wide range of handwritten documentary sources from the human record [1]. Most of the current generation of HTR systems are trained at the line level to optimize connectionist temporal classification (CTC) loss [2]. This frees users from having to annotate individual words or characters to produce training data; instead, they simply transcribe the plain text of each line. Creating training data still remains a bottleneck for HTR. Nockels et al. [1] describe the community around the Transkribus HTR system as “a

CHR 2023: Computational Humanities Research Conference, December 6 – 8, 2023, Paris, France

✉ dasmith@ccs.neu.edu (D. A. Smith); j.murel@northeastern.edu (J. Murel); jallen22@umd.edu (J. P. Allen); mtmiller@umd.edu (M. T. Miller)

ORCID [0000-0002-6636-6940](https://orcid.org/0000-0002-6636-6940) (D. A. Smith)

 © 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

‘bottom-up’ mass digitization movement, made up of hundreds of simultaneous projects driven by motivated researchers” creating their own training data and models. Put another way, an important use case for HTR is to help those “motivated researchers,” who know which documents they wish to transcribe, annotate enough data so that they can train a model to transcribe the rest.

We contend that there is room for a complementary training paradigm for HTR. In some projects, we are confronted with large collections of documents in a diversity of languages, hands, genres, and time periods. If we do not know *a priori* which documents will be interesting, it may be hard to allocate efficiently the time it takes to produce HTR training data for the whole collection. Instead, we propose a *distant supervision* approach to training HTR that takes advantage of the structure of many larger collections. **Automatic Collation for Diversifying Corpora** (ACDC) starts by having users assemble digital editions of texts they believe will be widely copied in the collection. Starting from an initial, imperfect model, ACDC proceeds by:

1. running initial HTR segmentation and transcription models on a diverse manuscript collection (§3);
2. aligning passages in this HTR output with passages in the reference digital editions (§4.1);
3. selecting manuscript lines with their page-image coordinate information and their corresponding text from the reference editions (§4.2); and
4. retraining the HTR model on the selected lines.

Once a new HTR model is trained, it can be used to re-transcribe the manuscript collection and run this process again (Figure 1).

Distant supervision has been employed at the paragraph level in HTR [3] and, using state-of-the-art vision transformers, for training joint segmentation and transcription models [4]. These systems, however, still assume that we have a “diplomatic”, ground-truth transcription of a particular manuscript paragraph or page. ACDC instead infers which matching passages between a noisy HTR transcript and a reference digital edition are close enough to use for training and which might contain variant readings.

In this paper, we demonstrate ACDC’s effectiveness by applying it to a diverse collection of Arabic-script manuscripts (§2). We annotate a test set drawn from a diverse collection of 59 Arabic-script manuscripts and a training set of 81 manuscripts of popular texts embedded within a larger corpus. After a few rounds of ACDC retraining, character accuracy rates on the test set increased by 19.6% absolute percentage, while a supervised model trained on manually annotated data from the same collection increased accuracy by 15.9% (§5). We analyze the variation in ACDC’s performance across books and languages and discuss further applications to collating manuscript families (§6). We have released our code, annotated data, and trained models under open-source licenses.¹

¹See repositories of code (https://github.com/OpenITI/acdc_train), test data (https://github.com/OpenITI/aocp_ms_eval), annotated lines from the training set to compare to unsupervised ACDC (https://github.com/OpenITI/arabic_ms_data), and trained models and evaluation data (https://github.com/OpenITI/acdc_results).

Table 1
Manuscript test data by language

language	Arabic	Ottoman Turkish	Persian	mixed	total
manuscripts	41	8	4	6	59
transcribed lines	1088	337	126	153	1704

Table 2
Manuscript families for training

author	title	language	MSS	transcribed lines
al-Jazūlī	Dalā'il al-khayrāt	Arabic	28	1602
Fīrūzābādī	al-Qāmūs al-muḥīṭ	Arabic	13	1254
Ḥāfiẓ	Divān	Persian	11	981
Sa'dī	Gulistān	Persian	17	1865
Taftāzānī	Sharḥ al-'Aqā'id al-Nasafīya	Arabic	12	1140
			81	6842

of 29 lines per book is not usually enough to train book-specific model with acceptably high accuracy. Unlike some other evaluations [8] on Arabic-script manuscripts, however, this paper focuses on the use-case where the training set and test sets come from different manuscripts and different hands.

To test the ACDC method’s effectiveness at producing HTR training data, we selected five texts, three in Arabic and two in Persian, for which we could find digital transcriptions and a reasonable number of digital editions (Table 2). We downloaded 81 sets of page images of these five texts. To perform error analysis and to be able to compare ACDC to supervised training, we transcribed 6842 lines in total from these 81 manuscripts. None of these manual transcriptions were used for training ACDC. During training, we also added 50 additional “distractor” manuscripts to evaluate the alignment process. None of these 131 manuscripts overlap with the 59 manuscripts used for testing. Furthermore, none of the 59 test manuscripts are copies of the five widely-copied works we use for training.

All of the manuscript images used here have been released by the libraries digitizing them into the public domain. We release the layout analysis and transcribed lines under an open-source license.

3. HTR Training and Testing

We employ the Kraken HTR system [9] for training and testing layout analysis and transcription models due to its support for right-to-left scripts and the curved baselines common in manuscripts. As with other current HTR systems, Kraken first uses a *segmentation model* to detect regions and lines in a page image; it then separately passes each extracted line image to a *transcription model* to produce text output. The ACDC method described here could be easily adapted to other line-oriented OCR systems.

The experiments in this paper start with segmentation and transcription models trained on annotations produced for Arabic and Persian printed books by the Open Islamicate Texts Initiative [10, 11]. While the layout of books and manuscripts is of course very different, we keep this print-trained segmentation model fixed for all experiments to focus on improvements in text alignment and transcription models. (See §6 for further discussion of layout analysis.)

We use *character accuracy rate* (CAR) to evaluate the effectiveness of transcription models. This metric computes the (Levenshtein) edit distance between the reference transcription and the model output and divides by the number of characters in the reference. The resulting character error rate is then subtracted from one, i.e. $CAR = 1 - \text{edit}(\text{reference}, \text{hypothesis}) / \#(\text{reference chars})$. We remove Arabic short vowel marks and merge variant forms of the letters *kaf* and *yah* in both the reference and hypothesis before comparing them. In addition to CAR, we also measure the *Arabic character accuracy rate* by removing spaces, punctuation, and other non-Arabic characters from the reference and hypothesis before comparing them. As we discuss in §4, we ignore non-letter characters when aligning noisy HTR output with digital editions. Arabic CAR is thus a helpful diagnostic for relating transcription accuracy on these letters to the amount of training data the ACDC method is able to extract. When summarizing these evaluation metrics across a test set, we take the average of the CAR for each book. This “macro averaging” ensures that books with more transcribed lines do not receive undue weight in the final evaluation.

We train transcription models with Kraken on pairs of manuscript line images and reference transcriptions. As with similar line-level HTR systems, Kraken minimizes connectionist temporal classification (CTC) loss [2] with respect to the weights of a convolutional plus recurrent neural network. For supervised training, both the boundaries of the lines within the page image and the transcriptions were produced manually as discussed above (Table 2). For ACDC training with distant supervision, the boundaries of the lines were produced by the print-trained segmentation model and the reference transcriptions were inferred by the collation process (§4). We trained transcription models both from scratch, i.e., with random initialization of all weights, and by *fine tuning* the existing print-trained model. In our experiments, fine-tuning always proved more effective on both validation and test data. For each training set, we randomly hold out 10% of the lines as validation data to perform early stopping and model selection. We use a constant learning rate of 10^{-4} recommended by Kraken for manuscript training and perform early stopping when the best CAR on validation data has not improved for ten iterations.

The print-trained model we use as a starting point for our experiments achieves a (macro-averaged) 60.5% CAR on the test set. As shown in Figure 2, this average combines clusters of books with CAR in the high 60s and above and books with CAR in the mid 50s and below. Fine-tuning this print model on the 6842 manually transcribed lines from our training set achieves an average CAR of 76.4%. While the print model transcribed 35 test books with CAR less than 60%, the supervised model performs below 60% on only six test books. Even the supervised model is trained with no overlap between the training books and test books. Its accuracy is therefore below what we would expect from the common HTR paradigm where training pages are drawn from the same book as test pages.

4. Collating Noisy HTR with Digital Editions

The ACDC method starts with the output from an initial HTR model—here, the print-trained model. It then aligns this HTR output with a collection of reference texts to see if any parts of the HTR output are sufficiently close to some passage in a reference text. In this section, we describe the inference process for collating noisy HTR output with reference texts or the HTR output on other manuscripts. We then analyze this collation output to select lines for retraining HTR models.

The proposed approach is another step in increasingly distant supervision for training HTR. Kraken, like other HTR systems that are trained to minimize the CTC loss between a reference transcription of a line and model predictions, already performs a character alignment for each line [2]. This process enables us to forgo annotating each character’s position on the page image and instead simply annotate a whole line with the desired sequence of characters. Chammas et al. [3] proposed collecting reference transcriptions at the paragraph level and using the best Levenshtein alignment with HTR output to split this reference into lines.

Coquenot et al. [4] proposed collecting full-page transcriptions and learning a page-level reading order. In this paper, we propose a *corpus-level* approach to alignment: rather than deciding ahead of time which lines or paragraphs or pages we should transcribe, we collect reference texts that we believe will overlap with significant number of manuscripts in our corpus (Table 2). When preparing input for ACDC, we do *not* need to specify which reference texts correspond to which manuscripts—let alone to which pages or lines.

This corpus-level approach, however, makes the alignment problem more difficult in two ways. First, there is the search problem of matching lines in HTR output with passages in arbitrarily long reference texts (§4.1). Second, we need to infer which line-level alignments are of high enough quality to use as training data (§4.2).

4.1. HMM Alignment Models

Unlike previous approaches to distant supervision for HTR, we cannot use Levenshtein alignment (i.e., Needleman-Wunsch) [3] since the page or other portion of a MS we happen to have may not cover the whole texts of the reference edition we are trying to align it to; moreover, a given MS page may contain material, such as commentary or other notes, extraneous to the

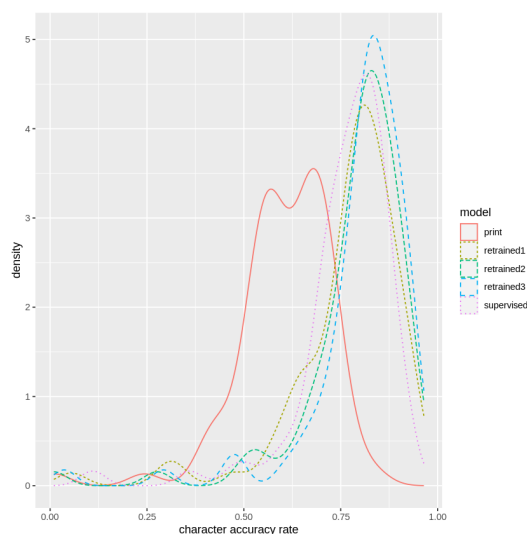


Figure 2: Distribution of CAR of the model trained on printed text compared to fully supervised training and to the first three iterations of ACDC training with distant supervision.

main text (Figure 7). Previous work on HTR, by contrast, has employed “diplomatic” transcriptions those manuscripts selected for the training set. Unlike previous work on text reuse detection [12, 13], we do not use Smith-Waterman alignment due to the problem of differences in reading order among different manuscripts and editions. Due to either differences in layout or errors in layout analysis, two versions of a text with the same material might present the same material in different sequence.

We propose, therefore, to use a more generalized finite-state approach to alignment based on hidden Markov models (HMMs) [14]. The observations are characters of (the HTR transcript of) the manuscript we are trying to align to a digital edition or another manuscript, and the hidden states are positions in these other witnesses. For any position in the target manuscript, the hidden state is a “read head” that specifies what source we might be copying from. Unlike Levenshtein or Smith-Waterman alignments, it is possible to move this read head backwards or forwards an arbitrary distance in the source. That does not mean that all jumps in position are equally likely, however.

As in other HMMs, we need to specify a *transition* distribution that assigns probabilities to shifts in position of the read head and an [emission] distribution that specifies what characters we are likely to observe in the target text when reading from a given location in the source. To compute $p(t_i|s_i)p(s_i|s_{i-1})$, the probability of generating the i th target character given the source position that generated the $i-1$ st, we consider that the source state can continue generating text in its current position with probability γ or it can move the read head anywhere in the source text with probability $1 - \gamma$. We then compute a probabilistic version of Levenshtein distance with parameter α , the probability that a character will be copied unchanged from the source to the target. The remaining $1 - \alpha$ probability is divided uniformly among all other possible edits, i.e., substitutions, insertions, and deletions. The probability that we will stop generating target text is $(\alpha + 1)/2$. We also include a pruning parameter g , the length of the allowable gap between target characters that are copied unchanged from the source. For the experiments in this paper, we let $\alpha = 0.8$, the average character accuracy rate in previous experiments on Arabic-script HTR. We let $\gamma = 0.998$ and $g = 600$. Since this HMM is a generative model of the target text, it is possible to reestimate α and γ from unlabeled data using expectation maximization. We leave that for future work since, as we shall see, we are able to recover sufficient high-quality aligned data at these parameter settings.

As with other edit-distance computations, the time and space complexity of inference with this HMM grows as the *product* of the lengths of the source and target texts. In common with other approaches to text-reuse analysis, therefore, we prune the search space by constraining the alignment at positions where we find sufficiently long matches between source and target. Unlike other text-reuse approaches that tokenize the input into words [12, 13], possibly lemmatizing or taking advantage of thesauri and other lexical resources [15], our alignment operates at the character level. The lower character accuracy rate for Arabic-script HTR makes matching even single words between two documents. Even more seriously, word-segmentation errors are especially common Arabic-script manuscripts: the space character is one of the most commonly inserted or deleted characters in our experiments. Instead of word n -gram features for pruning, therefore, we use subsequences of n characters in the alphanumeric Unicode class, thus ignoring both combining diacritics (e.g., short vowel marks in Arabic), whitespace, and punctuation. In preliminary experiments measuring the match rate between HTR output and

digital editions (see §4.2) without access to manual manuscript transcriptions, we set $n = 7$ and required $m = 5$ such subsequences to match before aligning a source and target passage. When performing a full collation of all manuscript pages against all other pages, this pruning results in running the HMM alignment on only 2% of possible pairs of pages. Other character-based methods instead apply alignment algorithms on all possible pairs, resulting in orders of magnitude more computational cost in order to maximize recall [16].²

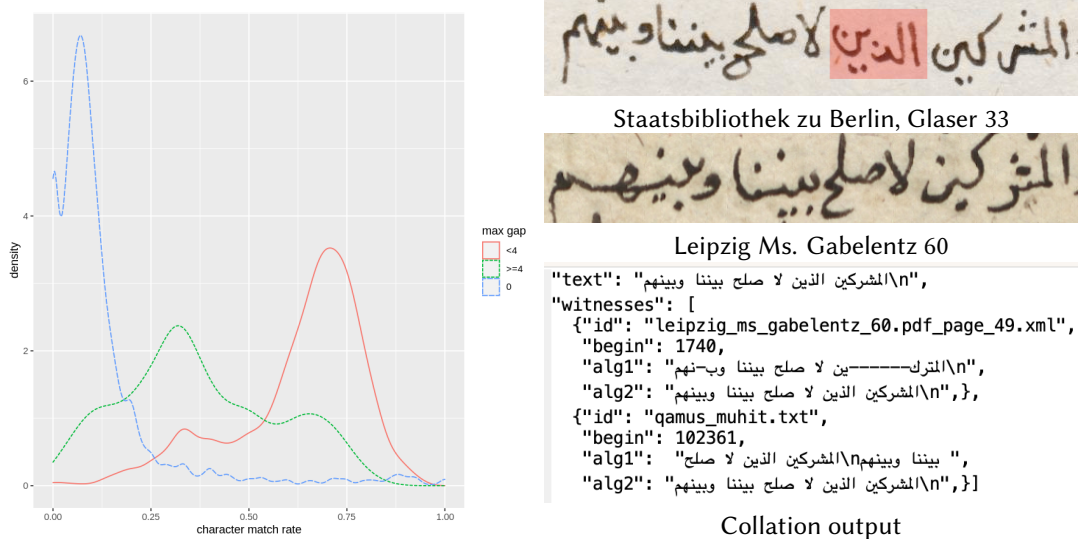
4.2. Scoring Candidate Lines for Training

Once HTR transcripts have been aligned with a collection of digital editions or with the HTR of other manuscripts, the output is organized with each manuscript being treated as the “target” text in turn. For each line of the target text, the alignment shows zero or more passages from other texts as witnesses. In the fragment of JSON output in Figure 3b, for example, one line of a Berlin manuscript is shown as the target text with one passage spanning two lines from a digital edition of Fīrūzābādī *al-Qāmūs al-muḥīṭ* and another passage from a Leipzig manuscript as witnesses. The digital edition matches the target Berlin manuscript perfectly, and so we can with high confidence use this transcription, along with a line image extracted by the print-trained segmentation model, as additional training data for HTR.

Not all lines, of course, match perfectly; moreover, it seems likely that manuscripts with mistakes in their transcription by the current HTR model might benefit more from additional training data. Differences between texts, however, can arise for two different reasons. First, as we saw above (§3), the output of the initial print-trained HTR model will match the diplomatic transcription in our evaluation set only 60.5% of the time. Second, the manuscripts we are transcribing with HTR, and the digital editions we are aligning to, may include variants *included by their writers or editors*. In the Figure 3b example, we can see that the Leipzig manuscript omits a word included in both the Berlin manuscript and the digital edition. It would therefore be dangerous to use the digital edition as ground truth for the image of the Leipzig manuscript.

To separate these two sources of variation, we analyze both the *match rate* (the proportion of characters in the digital edition that are exactly copied in the HTR transcript) and the pattern of *gaps* (insertions or deletions) in the alignment between them. Due to errors in the print-trained segmentation model, many lines are not fully or correctly identified (§6). We therefore exclude lines under five characters long (about one word, to exclude fragmentary lines) and those with a gap at the initial or final position in the alignment. We analyze the remaining lines by their match rate and their *max gap*, i.e., the length of the maximum number of contiguous insertions or deletions. Figure 3a shows that lines with a $\text{max gap} \geq 4$ mostly have a match rate below 50%. A significant cluster of these lines with longer gaps still has a match rate above 50%, as with the Leipzig MS example in Figure 3b. Perhaps surprisingly, lines with a max gap of zero, i.e., no insertions or deletions at all, tend to have a much lower match rate. Upon inspection, these tend to be lines with low accuracy in between other lines with much better accuracy where the HMM found a higher probability alignment by substituting a series of non-matching characters. There are a small number, as in the Berlin MS example, with zero gaps and high match rate. Finally, lines with a max gap between 1 and 3 inclusive had a match rate of mostly more than

²Version 2 of `passim` (<https://github.com/dasmiq/passim>) implements this model as `seriatim`.



(a) Match rate between HTR and edition (b) Collating a MS with an edition and another MS

Figure 3: Analyzing HTR collations: (a) Even with the baseline print HTR, a significant number of lines have more than 50% characters matching between HTR and a digital edition. (b) The collation between two manuscripts and a digital edition shows that one word present in the Berlin MS and the digital edition (highlighted in red) not present in the Leipzig MS (indicated by hyphens in the collation output).

50%. For our experiments, therefore, we selected those lines with a max gap less than 4 and a match rate greater than 50%. In experiments with a fixed number of lines for training, we selected them in descending order of match rate.

5. Experiments with ACDC Training

We have now described the components of the ACDC method:

1. compiling a corpus of manuscript page images that we believe to have some overlap with a collection of reference editions (§2);
2. running initial HTR segmentation and transcription models (§3) on this corpus;
3. aligning passages in this HTR output with passages in the reference texts (§4.1);
4. selecting manuscript lines with their page-image coordinate information and their corresponding text from the reference editions (§4.2); and
5. retraining the HTR model on the selected lines.

After executing these steps, we can *iterate* the process, returning to step 2 and using the re-trained HTR model to re-transcribe the training manuscripts. As noted above, this paper focuses on HTR transcription and does not retrain the segmentation model. When training HTR models, we can choose to train from scratch, i.e., from a random initialization of model parameters, or to start training from an existing model. As noted in §3, the latter always led to

Table 3

Mean character accuracy rate of the initial print-trained models compared to fully supervised training and to the first three iterations of ACDC training with distant supervision. Also shown are the CAR on manually transcribed lines of the training manuscripts and CAR when trained on a set of matched lines from each manuscript.

model	MS training lines	training CAR%	test CAR%
print	0	56.6	60.5
ACDC retrained 1	40,983	73.6	76.7
ACDC retrained 2	105,004	77.4	78.8
ACDC retrained 3	177,383	79.1	80.1
supervised	6,842	95.2	76.4
matched lines: ACDC retrained 1	2,786	68.8	72.2
matched lines: supervised	2,786	82.6	73.8

better validation and test accuracy in our experiments. Further search of the space of training hyperparameters might lead to gains, but we did not pursue the investigation.

Table 3 shows the average accuracy on the annotated lines of the training set and the test set of the initial print-trained model, the fully supervised model, and the first three iterations of using ACDC to retrain the HTR model without any access to transcribed manuscript data. On average, three iterations of ACDC training improved over the initial model’s CAR by 19.6% absolute, more than 3 percentage points above the performance of the supervised model.

Figure 2 shows the distribution of CAR over different books in the test set for each of these models. As noted above, the initial print-trained model exhibits two discernible clusters of books that perform above and below 60% CAR. Both ACDC and supervised training greatly reduce the number of poorly performing books and concentrate CAR more tightly at a higher level.

The range of accuracies achieved by the initial model across different books means that not all books are equally well represented in the training data ACDC extracts on this first (or later) iterations. As discussed in §4.2, we select lines with a short maximum gap length and match rate above 50%. In Figure 4a, we observe that higher Arabic CAR on training data for books, particularly when above 50%, leads to higher yields, i.e., a high proportion of a book’s lines extracted for training. We compute CAR on Arabic characters alone, excluding spaces and punctuation, because spaces and punctuation are also excluded when finding matching passages during the alignment process. Note that these evaluations on training data are not used during ACDC training or for model selection. The observations for the same book’s accuracy under the initial print-trained model and the first ACDC-trained model are linked by lines to show the direction of change. Figure 4b shows the next step in the process: higher numbers of training lines extracted for a book unsurprisingly lead to higher Arabic CAR when evaluated on that book.

We also examine the variation in accuracy at the level of individual test books. In Figure 5a, we see that almost all test books have higher accuracy after ACDC training than with the initial print-trained model, i.e., their points are above the diagonal. Each point is coded with the first

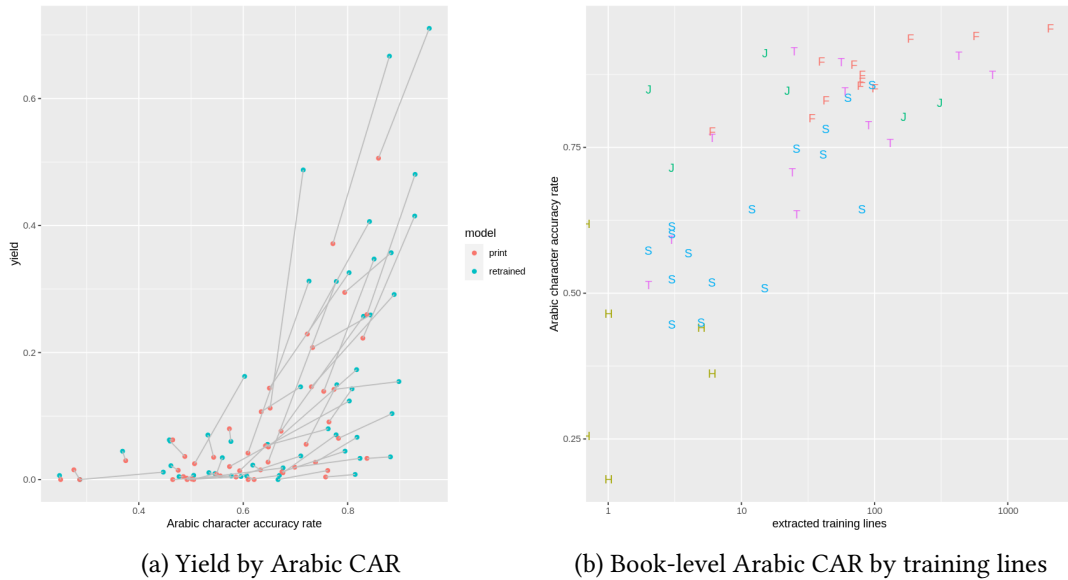


Figure 4: Relationship between Arabic character accuracy rate on training data and the amount of training data extracted for each book. In (a), the initial print-trained and first ACDC-trained models drive higher yields in extracting new training lines. Although these are book-level measurements, we see much higher yields with Arabic CAR above 50%, which is the line-level match rate threshold. In (b), we see the relationship at the book level between lines used for retraining the final ACDC model and Arabic CAR on the training set. Each book is marked with the first letter of the author’s name: Jazūlī, Fīrūzābādī, Ḥāfiẓ, Sa’dī, or Taftāzānī (Table 2).

letter of the book’s language. Books whose accuracy increased the most were in Arabic. This is less surprising considering the results in Figure 4b, where much more training data was extracted by ACDC for Arabic books. The exceptions to this consistent improvement were one Persian book and two Arabic documentary texts with hands very different from the book hands in the training set. Considering this result from another angle, Figure 5b shows that ACDC achieved better results than supervised training for most books, with the exception of the aforementioned documentary texts and a cluster of some of the Persian, Ottoman Turkish, and mixed-language books.

We ran an additional experiment to remove the difference in language coverage between the supervised and ACDC training sets. Selecting the same number of lines from each training book from among both the manually transcribed data and the lines extracted by the first iteration of ACDC leaves us with 2786 training lines. The bottom of Table 3 shows that the differences between the digital editions used by ACDC and the transcriptions produced manually for these manuscripts still results in a difference in performance between these models even when training data in exactly the same proportions is used, albeit smaller than the difference between the full training runs. The remaining differences between the learning curves of these training methods may be the result of ACDC training using the print-trained layout model to identify line images, while supervised training uses manually corrected line images. Even if ACDC could exactly recover the transcription of a line, a layout model’s cutting off some letters

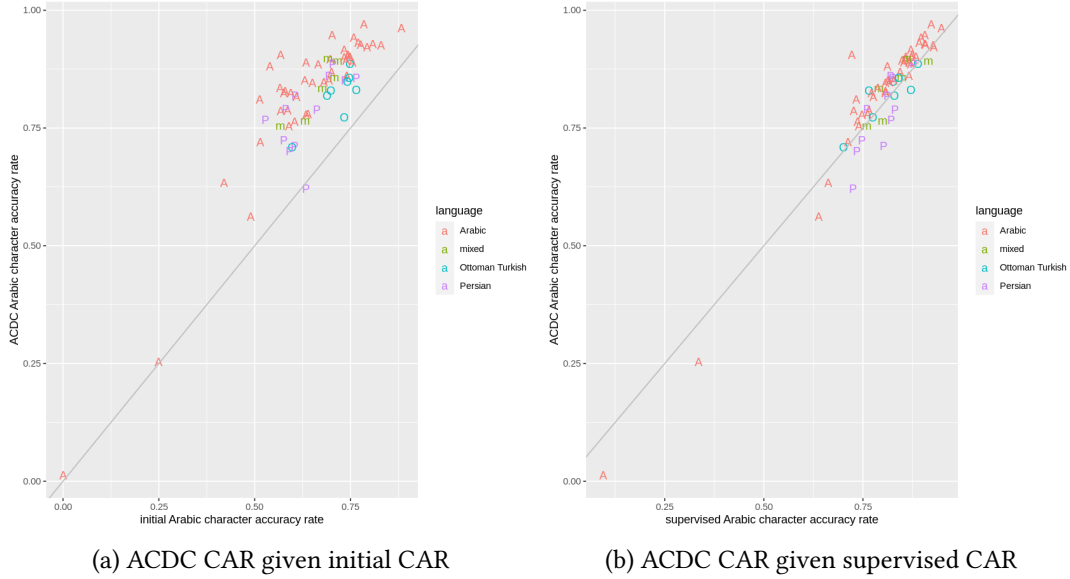


Figure 5: Arabic CAR on test data. In (a), we see that almost all books achieve gains over the initial print model in ACDC training, with the exception of two Arabic documentary texts in a very different hand and one Persian text. In (b), we see that ACDC achieves better results than supervised training for most books, with the exception of the Arabic documentary texts and a cluster of Persian, Ottoman Turkish, and mixed-language books.

in that line or erroneously including others would inhibit accurate training. Future work could aim both to evaluate the effectiveness of training on line images with erroneous boundaries and to bootstrap better layout models.

6. Discussion

The experiments in §5 show that Automatic Collation for Diversifying Corpora (ACDC) is a promising approach to improving HTR systems on diverse manuscript collections without additional annotated data. All that is required is that the manuscript collection have a sufficient number of widely-copied texts so that we can align their noisy HTR transcripts with clean digital editions. This may not be the case for many documentary archives with unique manuscript letters, for instance. Some archives of official documents, however, may include enough duplicated material for ACDC to work. Distant supervision will not in the near future, we expect, replace supervised training for projects where a researcher can identify ahead of time those documents or hands of interest and curate a training set for them. In any case, we reiterate that ACDC does not assume that the *test* set will have manuscripts that overlap with existing digital editions.

We also note that the impressive gains shown by ACDC were made despite working with a page segmentation model trained on printed texts. This model can often fail spectacularly (Figure 7). Even on that page, with a large amount of unrecognized marginalia, ACDC was able to extract one line. The majority of the training data extracted by ACDC from the training

manuscripts was from pages where the print layout model worked surprisingly well (Figure 6), despite some errors with features like rubrication or words written larger than others on the same line. We are hopeful, therefore, that a similar distant supervision approach can be employed to improve segmentation models by identifying and perhaps normalizing outputs on pages like these.

As the number of manuscripts with digitized page images grows, we expect that broad-coverage methods like ACDC will complement task-specific training sets. Beyond training HTR, we also expect that the collation methods developed here will be useful in producing multi-text editions (Figure 3b), as well as using evidence from multiple manuscripts to model the text-transmission process.

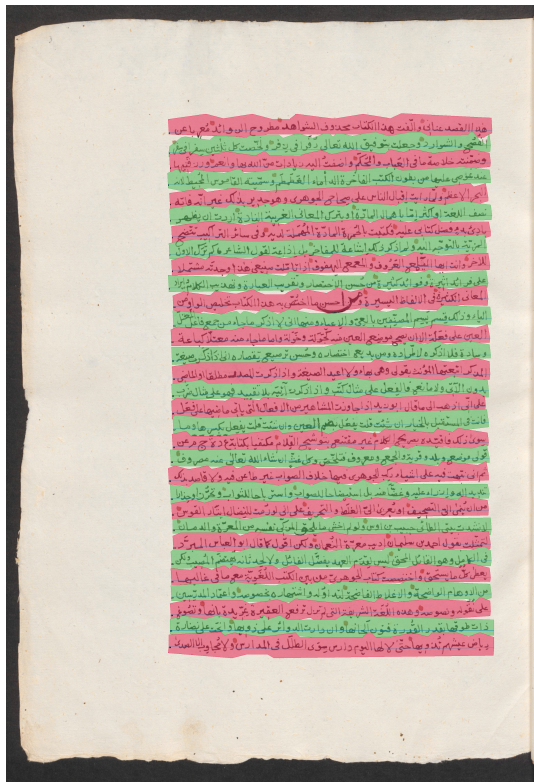
Acknowledgments

The authors would like to thank their collaborators on the Open Islamicate Texts Initiative—in particular, John Mullan, Lorenz Nigst, and Alejandro Toselli—for help annotating data and training the print models. This work was supported in part by a National Endowment for the Humanities Digital Humanities Advancement Grant (HAA-277203-21) and the Andrew W. Mellon Foundation’s Scholarly Communications and Information Technology program. Any views, findings, conclusions, or recommendations expressed do not necessarily reflect those of the NEH or Mellon.

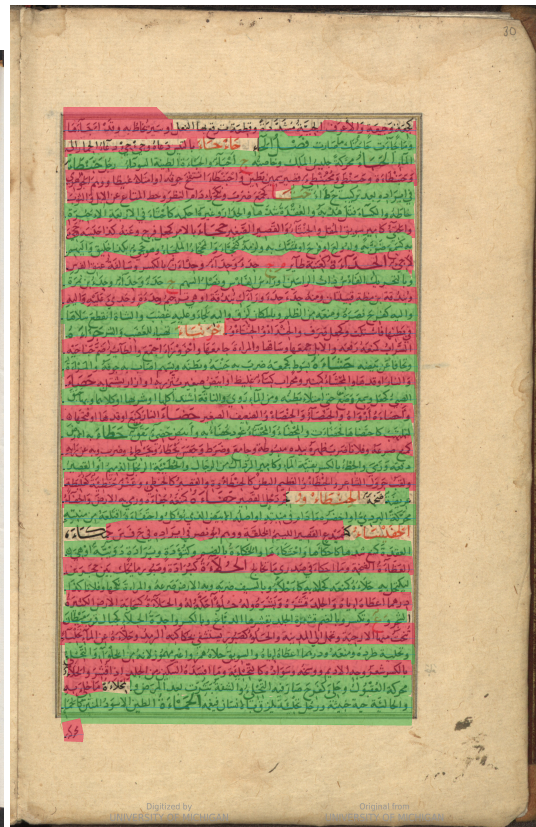
References

- [1] J. Nockels, P. Gooding, S. Ames, M. Terras, Understanding the application of handwritten text recognition technology in heritage contexts: a systematic review of *Transkribus* in published research, *Archival Science* 22 (2022) 367–392. doi:10.1007/s10502-022-09397-0.
- [2] A. Graves, S. Fernández, F. Gomez, J. Schmidhuber, Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks, in: *Proceedings of the International Conference on Machine Learning (ICML)*, 2006, pp. 369–376. doi:10.1145/1143844.1143891.
- [3] E. Chammas, C. Mokbel, L. Likforman-Sulem, Handwriting recognition of historical documents with few labeled data, in: *International Workshop on Document Analysis Systems (DAS)*, 2018, pp. 43–48. doi:10.1109/DAS.2018.15.
- [4] D. Coquenot, C. Chatelain, T. Paquet, DAN: a Segmentation-free Document Attention Network for Handwritten Document Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023) 1–17. doi:10.1109/TPAMI.2023.3235826, arXiv:2203.12273 [cs].
- [5] A. Bausi, P. G. Borbone, F. Briquelet-Chatonnet, P. Buzi, J. Gippert, C. Macé, M. Maniaci, Z. Melissakis, L. E. Parodi, W. Witakowski (Eds.), *Comparative Oriental Manuscript Studies: An Introduction*, COMSt, Comparative Oriental Manuscript Studies, Hamburg, Germany, 2015.

- [6] C. Ernst, In search of Sufi manuscripts, in: *The Eleventh Islamic Manuscript Conference: Sufism and Islamic Manuscript Culture*, 2016.
- [7] M. Romanov, Algorithmic analysis of medieval Arabic biographical collections, *Speculum* 92 (2017) 226–246.
- [8] A. Keinan-Schoonbaert, Results of the RASM2019 competition on recognition of historical Arabic scientific manuscripts, 2019. URL: <https://blogs.bl.uk/digital-scholarship/2019/09/rasm2019-results.html>.
- [9] B. Kiessling, Kraken: A universal text recognizer for the humanities, in: *Digital Humanities (DH)*, 2019.
- [10] M. Romanov, M. T. Miller, S. B. Savant, B. Kiessling, Important new developments in arabographic optical character recognition (ocr), 2017. [arXiv:1703.09550](https://arxiv.org/abs/1703.09550).
- [11] M. T. Miller, M. G. Romanov, S. B. Savant, Digitizing the Textual Heritage of the Premodern Islamicate World: Principles and Plans, *International Journal of Middle East Studies* 50 (2018) 103–109. URL: https://www.cambridge.org/core/product/identifier/S0020743817000964/type/journal_article. doi:10.1017/S0020743817000964.
- [12] D. A. Smith, R. Cordell, E. M. Dillon, N. Stramp, J. Wilkerson, Detecting and modeling local text reuse, in: *ACM-IEEE Joint Conference on Digital Libraries (JCDL)*, 2014.
- [13] K. Funk, L. A. Mullen, The Spine of American Law: Digital Text Analysis and U.S. Legal Practice, *The American Historical Review* 123 (2018) 132–164. doi:10.1093/ahr/123.1.132.
- [14] S. Vogel, H. Ney, C. Tillmann, HMM-based word alignment in statistical translation, in: *Proceedings of the 16th Conference on Computational Linguistics (COLING)*, volume 2, 1996, p. 836. doi:10.3115/993268.993313.
- [15] M. Moritz, A. Wiederhold, B. Pavlek, Y. Bizzoni, M. Büchler, Non-literal text reuse in historical texts: An approach to identify reuse transformations and its application to Bible reuse, in: *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, 2016.
- [16] P. Paju, H. Rantala, H. Salmi, Towards an ontology and epistemology of text reuse, in: E. Bunout, M. Ehrmann, F. Clavert (Eds.), *Digitised Newspapers – A New Eldorado for Historians?: Reflections on Tools, Methods and Epistemology*, De Gruyter, 2022.



(a) Staatsbibliothek zu Berlin, Glaser 33



(b) Staatsbibliothek zu Berlin, Glaser 133

Figure 6: These two manuscripts from Berlin contain copies of (6a) Firuzbadi *al-Qamūs al-muḥīṭ* and (6b) al-Jazūli *Dalā'il al-khayrāt*. The output of the line-extraction model trained on printed books is displayed using alternating bands of color overlaid on the page images. The left-hand image is close to perfect; on the right-hand page, the rubricated text has confused the model, although the catchword in the lower left has been caught. [Public domain, Staatsbibliothek zu Berlin – PK]



(a) Print-trained line detection

(b) Selected well-aligned line

Figure 7: Library of Congress PK6450 .G2 1593, one of the manuscripts used for alignment, contains Sa'di's *Gulistān*, along with extensive marginalia. Applying the print-trained line extraction model used throughout this paper fails to detect much of the text in both the body and the margins of the page (a). The output of the line-extraction model is displayed with alternating color overlays to increase contrast. On this page, the ACDC process extracts a single line from the body of the text (b). The marginalia from this copy are not found in our electronic edition and so would not be aligned in any case. [Public domain. Library of Congress, African and Middle East Division, Near East Section Persian Manuscript Collection]