Systematic Textual Availability of Manuscripts*

Hadar Miller,[1] Samuel Londner,[2] Tsvi Kuflik,[1]
Daria Vasyutinsky Shapira,[2] Nachum Dershowitz,[2] Moshe Lavee[1]
[1]Haifa University [2]Tel Aviv University

Abstract

The digital era has made millions of manuscript images in Hebrew available to all. However, despite major advancements in handwritten text recognition over the past decade, an efficient pipeline for large scale and accurate conversion of these manuscripts into useful machine-readable form is still sorely lacking.

We propose a pipeline that significantly improves recognition models for automatic transcription of Hebrew manuscripts. Transfer learning is used to fine-tune pretrained models. For post-recognition correction, it leverages text reuse, a common phenomenon in medieval manuscripts, and state-of-the-art large language models for medieval Hebrew.

The framework successfully handles noisy transcriptions and consistently suggests alternate, better readings. Initial results show that word level accuracy increased by 10% for new readings proposed by text-reuse detection. Moreover, the character level accuracy improved by 18% by fine-tuning models on the first few pages of each manuscript.

1 Introduction

The survival rate of medieval Hebrew manuscripts is much lower than that of Latin or Arabic texts. Thus, the extant Hebrew manuscripts—spread out in libraries and private collections worldwide—are a precious asset of historical, cultural and intellectual heritage.

The digital era has brought a renaissance to the study of ancient and medieval manuscripts, heretofore available for examination only to limited scholarly circles working at circumscribed locations. Recent advancements in digitization have made images of most of the surviving Hebrew manuscripts accessible now from every computer, notably through the Ktiv project of the National Library of Israel (National Library of Israel, 2021). On the order of one hundred thousand manuscripts—comprising more than ten million images—are expected with the completion of the Ktiv project.

Unfortunately, despite major progress in optical character recognition (OCR), an efficient framework for large-scale and accurate conversion of these manuscripts into a machine-readable form remains lacking. The complexity of the materials and the poor quality of many of the items constitute a major hindrance on the way to full textual accessibility.

With the rising prominence of artificial neural networks (ANN) and their application to handwritten text recognition (HTR), the accuracy of the automatic processes is continuously improving (AlKendi et al., 2024). The Tikkun Sofrim project (Kuflik et al., 2019; Wecker et al., 2022) designed and tested an ANN based, automatic transcription pipeline for Hebrew manuscripts. The project leveraged the open-source tool kraken (Kiessling, 2019), off-the-shelf methods for automatic page segmentation, layout analysis, and line segmentation and developed a tailored crowdsourcing platform to validate and correct automatic transcriptions (Kiessling, 2019). This led to the development of eScriptorium (Kiessling et al., 2019), a virtual research environment, enabling scholars to create a full-fledged transcription. However, kraken is designed to train a specific LSTM neural network model for each manuscript. This requires large efforts preparing labeled data for training the model for each manuscript. To dramatically reduce the quantity of manual annotation effort needed to create training

*Supported in part by the Israeli Ministry of Science and Technology (#3-17516), the Tel Aviv University Center for AI and Data Science, and the European Research Council (MiDRASH, Project No. 101071829). Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authorities can be held responsible for them.

sets for handwritten Hebrew text recognition, we employ a bank of pretrained models as an ensemble of models in parallel, combining their results. Moreover, when minimal labeling of manuscripts is available, we can use transfer learning to refine the accuracy of the pretrained models.

The crowdsourcing efforts needed for transcription validation and correction are labor intensive. We aim to increase the pipeline efficiency by dramatically reducing transcription error rate using post-recognition correction algorithms. The most effective method at our disposal for automatically improving transcriptions is the use of sequence alignment methods to line up the imperfectly deciphered texts with the same or with other compositions in existing corpora or previously transcribed manuscripts of similar texts (Miller et al., 2025). This approach was suggested in (Zhicharevich, 2011; Villegas et al., 2016) and by others. An early work on aligning OCR text with ground-truth (GT) transcriptions is (Rice et al., 1994). High-performance sequence alignment algorithms have long been used. Existing text alignment tools, however, generally assume accurate transcriptions, rather than error-riddled post-OCR texts. We propose a text-reuse detection framework designed for medieval Hebrew language, which utilizes fuzzy search on the inverted index, followed by an approximate alignment algorithm to handle noisy OCR. We combine this with the use of various state-of-the-art Hebrew language models to propose new and better readings.

2 Background and Related Work

2.1 Handwritten Text Recognition

We use off-the-shelf methods for automatic page segmentation, layout analysis, and line segmentation. Machine-learning based systems have seen wide use recently for these tasks, the majority using combinations of CNNs and LSTMs. State-of-the-art methods have been implemented in kraken and eScriptorium for mixed models in various scripts, including Hebrew, and for a wide range of manuscript types.

The best transcription results for such manuscripts are achieved by combinations of CNNs and BLSTMs (Dutta et al., 2018; Kahle et al., 2017; Kiessling, 2019). HTR efforts for medieval Hebrew manuscripts include (Kiessling et al., 2019; Kuflik et al., 2019; Kurar Barakat et al., 2019). The Sofer Mahir project digitized twenty large manuscripts of early rabbinic compositions.[1] In the Tikkoun Sofrim project (Kuflik et al., 2019; Wecker et al., 2022), crowdsourcing and machine learning were used to correct errors of the automatic transcriptions of several large manuscripts of medieval exegetical literature. Character error rates (CER) of 2–3% were typically attained for manuscripts with homogeneous layout and script but only around 9% in the presence of complications.

Today, given an undeciphered manuscript, we can achieve the best possible reading by use of the latest available bank of recognition models and algorithms. Aggregation and selection algorithms need to learn how to select the best automatic transcription model or combination of models for each specific manuscript (Kiessling, 2019; Reul et al., 2019). Letting OCR engines vote on readings has been done since at least the early 1990s (Handley and Hickey, 1991). Varying parameters of the input images (resolution, size, contrast) for each page can also have an impact, and image enhancement prior to OCR is commonplace. An attempt to apply this for Arabic was reported in (Kissos and Dershowitz, 2017); automatically choosing the most successful among a variety of image enhancements was found to yield twice the improvement of lexical post-OCR correction.

2.2 Transfer Learning

Manuscript handwriting styles being highly dependent on time, place, and individual scribes' predilections, improving over state-of-the-art models by leveraging transfer learning is an obvious choice. Models pretrained over a large corpus are fine-tuned on the first few annotated pages of a manuscript so as to help decipher the rest of the manuscript. In this way, the representation learned over a *source* dataset can be refined to solve the *target* task, namely transcribing documents of a smaller, disjoint dataset (Goodfellow et al., 2016). Recent research (Aradillas et al., 2021; Jaramillo et al., 2018) shows that the optimal method to improve accuracy is to fine-tune the parameters of the whole recognition model, while the first layer can be frozen without any meaningful performance degradation. In (Granet et al., 2018), the authors successfully apply transfer learning to historical handwritten Italian titles of plays.

[1] https://sofermahir.hypotheses.org.

2.3 NLP-Based Correction

Post-recognition error correction based on NLP techniques is a well-researched field. Pretrained language models of various kinds have been used to correct and refine OCR and HTR (Kukich, 1992; Zenkel et al., 2017), as well as optimized dedicated neural networks (Ghosh and Kristensson, 2017; Suissa et al., 2020). This approach can be further improved by adding a classifier and a weighted confusion matrix (Kissos and Dershowitz, 2016). In (Mahpod and Keller, 2018), an end-to-end jointly trained neural network for transcription and correction is proposed. In (Hannun, 2017), the author surveys the different HTR/OCR decoding strategies, and suggests incorporating a language model scoring by multiplying the OCR model's log-odds score matrix at decoding with corresponding conditional probabilities (or analog scores) derived from a language model (LM).

State-of-the-art pretrained transformer-based contextual language models such as BERT (Devlin et al., 2019) have been successfully used to detect and correct OCR errors (Nguyen et al., 2020) in English. Modern subword-level transformers-based language models are attractive to tackle post-recognition correction in morphologically rich languages such as Rabbinic Hebrew. Their advantage over character-level LMs has been demonstrated to be significant for semantic tasks (Keren et al., 2022). Similarly, classical word or phrase-based LMs have been shown to have lower accuracy when dealing with MRL (Amram et al., 2018; Seker et al., 2022).

BERT-like language models can be used to compute pseudo-perplexity, which has been shown to be an effective metric for scoring sentences for linguistic acceptability (Salazar et al., 2020; Lau et al., 2020), and thus for scoring and ranking candidate transcriptions. This measure is sensitive to the number of tokens, and in effect is biased towards longer sequences. Several normalizations have been proposed in (Lau et al., 2020), and following preliminary experiments we adopt their averaging normalization method, *MeanLP*. However, we normalize by the number of LM tokens, not by the number of words, as the averaging dimension is the token axis.

2.4 Text Reuse–Based Correction

Text-reuse detection algorithms are used to locate the content of a manuscript within a library of reference texts (Büchler et al., 2014), followed by alignment of the text against the most similar known text (Altschul et al., 1990; Hakala et al., 2019). Detected reused texts can be used to tackle potential failures of the automatic transcription (Zhicharevich, 2011).

Text-reuse detection. Manuscripts comprise human knowledge to be transmitted to others. The written transmission of information relied on various forms of intertextuality, whereby texts were either copied entirely (verbatim or in paraphrase) or were borrowed partially to inspire new ideas. This leads to the phenomenon of many witnesses available for a single segment of text. Thus, the likelihood that several witnesses have already been converted into a machine-readable form increases. For example, a manuscript segment could be matched with fragments quoted in later works, or appearing in dialog with other authors (Klein et al., 2014; Och and Ney, 2003; Smith et al., 2013), or made use of in the context of spreading and amplifying ideas and opinions (Smith et al., 2013; Wilkerson et al., 2015).

Text reuse engages the attention of humanities scholars when considering ancient manuscripts for a wide variety of languages, such as Greek (Lee, 2007). Most of the studies so far have focused on exploring the potential of information technology to automate text-reuse detection in a specific domain. Syntactic text-reuse detection frameworks rely on sequence alignment, which in turn requires aligning noisy OCR outputs, such as aligning dissimilar words or aligning multiple words in one to a single word in another. For our purposes, we propose a framework that handles noisy HTR, thanks to which even a gibberish-looking transcribed sentence can be accurately matched to reuses in other corpora (cf. Zhicharevich, 2011). See (Miller et al., 2025) for details.

Text alignment. Many alignment tools (e.g. Clough et al., 2002; Smith et al., 2015) assume accurate transcriptions. Brill et al. (2020) designed an alignment tool that aligns semantically similar words using word embeddings, but it cannot handle word boundary errors typical of OCR outputs. BLAST, designed for biological sequence matching, works well even when OCR errors exceed 50% (Vesanto et al., 2017). Miller et al. (2025) proposed an alignment tool—used here—for Hebrew capable of addressing word boundary errors, spelling mistakes, and aligning acronyms and synonyms.

2.5 Language of Corpus

Practical text-reuse detection and alignment challenges stem from the language of our interest. Hebrew is an orthographically and morphological complex language (Itai and Wintner, 2008). The number of valid inflected forms in Hebrew is 70 times larger than in English (HaCohen-Kerner et al., 2011). And there is no orthographic standard in Hebrew. More specifically *matres lectionis* are optional; a word may include it in one manuscript while it will be absent in another. We cannot know if a discrepancy is due to poor recognition or to an actual textual variant. Morphological analysis has been implemented in the text-reuse detection framework (Siegal and Shmidman, 2018) to convert the tokens into base form. Acronyms are ubiquitous in written Hebrew. There are 17,000 different abbreviations in rabbinic literature, 35% of which are ambiguous (HaCohen-Kerner et al., 2004), which challenges the alignment process. Furthermore, a Hebrew sentence can be written in multiple permutations while preserving meaning; therefore reuses may take on different forms, which may be scored by a framework like (Brill et al., 2020; Smith et al., 2014; Colavizza et al., 2014).

A few "encoder-only" modern Hebrew LMs have been proposed: HeBERT (Chriqui and Yahav, 2022), AlephBert (Seker et al., 2022), and AlephBertGimmel (Gueta et al., 2022). However, the Wikipedia-based dataset used to train them differs significantly in orthography and grammar from the old Hebrew used in manuscripts. One encoder-only LM for Rabbinic Hebrew is available, viz. BEREL (Shmidman et al., 2022), trained on 220 million words of this chronolect. Courtesy of the developers, we were provided three pre-publication variants, dubbed versions 1.0, 1.5 and 2.0. BEREL v1.0 is the model outlined in (Shmidman et al., 2022). BEREL v2.0 includes a number of improvements, including better tokenization of input samples, a larger source corpus, and supports sequences of up to 512 tokens. Whereas these two models have been trained on full sentences, BEREL v1.5 has been trained on partial sentences. More recently, the same authors introduced a large-scale generative causal (autoregressive) language model tailored for Rabbinic Hebrew called DictaLM (Shmidman et al., 2023), based on a decoder-only transformer architecture. This decoder-only transformer model is trained on a balanced corpus consisting of both Modern and Rabbinic Hebrew texts.

2.6 Combined Systems

The KITAB (Savant, 2016) and Open Islamicate Texts Initiative projects (OpenITI) (Allen et al., 2022), for Arabic and other manuscripts, have similar goals. Similar techniques are therefore appropriate.

3 Methodology

We designed a transcription pipeline that extends the one in (Kuflik et al., 2019), comprising the following steps:

1. First, manuscript images are needed. We rely on Ktiv, which is the midst of the process of digitizing the entire extant Hebrew manuscript corpus.
2. The next step is transcription of the text in the manuscript. We use the trained models of kraken to first segment and then transcribe the text appearing in the images.
3. Both text-reuse detection and large language models are then applied to propose corrections to several pages of the specific manuscript.
4. Based on that, experts correct any remaining transcription errors in those pages. The advanced user interface of (Kiessling et al., 2019) is used for this.
5. The recognition model is fine-tuned based on that ground truth.
6. The refined model is applied to the complete manuscript.
7. Experts or crowd-sourcing may be employed to correct any remaining errors.
8. The text-reuse detection framework kicks in again to map all interconnections between the manuscript and other documents in the corpus.
9. Finally, the outputs are delivered to humanities researchers.

4 Automatic Transcription

4.1 Handwritten Text Recognition

The automatic generation of transcribed text is achieved by the combination and integration of a variety of state-of-the-art algorithms. Core HTR is performed by the segmentation and recognition models trained on crowdsourced datasets in the Sofer Mahir effort (Stökl Ben Ezra et al., 2021). Accuracy is boosted by automatically selecting the most appropriate model, either via a semi-automatic recommendation system or by unsupervised analysis of graphical features. By manually labeling

the first pages of the manuscript and fine-tuning the models' parameters, one can further improve performance of the recognition models on specific manuscripts.[2]

4.2 Text Reuse–Based Corrections

We leverage text reuse and run the HTR data through a text-reuse detection framework which finds repetition pairs in the corpus and then align them based on a sequence alignment algorithm and propose a new and better reading for the HTR. Frameworks for short reuse detection first split large texts into small parts and try to detect reuses for each, commonly, n-gram over a sliding window (Foltỳnek et al., 2019). However, kraken automatically segments the manuscript into rows. Therefore, we utilized rows as our (varying-size, non-overlapping) sliding windows.

In the remainder of this section, we describe the text-reuse detection framework. It is tailored to Hebrew, on the one hand; on the other hand, it handles the expected noisy recognition inputs.

Preprocessing. We used the Sefaria digital corpus (Sefaria, Inc., 2021) as reference library. The digital texts are preprocessed, removing special characters from the data as in (Klein et al., 2014). Next we generate a positional inverted index (concordance) for fast candidate retrieval. In addition, a lexicon is created with an entry for each word in the corpus, holding the inflected word as it appears in the corpus as well as its base form extracted by a morphological analyzer (More et al., 2019). Each entry is enriched with the frequency of its appearance in the corpus.

Candidate retrieval. For each manuscript line, we execute a fuzzy search against the inverted index. For each token in the input line, we seek orthographically close tokens to allow for transcription errors as well as Hebrew's orthographic variability. We end up with a list of candidates suspected to have a text-reuse relation with the tested row.

Scoring candidates. The next step is to score the similarity between the tested line and each of the candidates. First we need to extract from each candidate a maximal segment pair, the most similar piece of text from the candidate with identical length to the tested line (Altschul et al., 1990). Then the similarity score between the two and the input line is measured by edit distance (Levenshtein, 1966). At this stage, we also measure the similarity between the candidate and the previous and following rows of the manuscript. We boost the candidate's score relative to the similarity with the neighboring rows. The intuition here is that the longer a passage is shared between documents the higher the probability of a text reuse relation between them. We employ predefined similarity thresholds for the decision to move the candidate forward to the alignment stage, an approach used by most text-reuse detection frameworks (Foltỳnek et al., 2019).

Fuzzy alignment. This stage aims to align all candidates against the tested row. Tokens with different orthography, abbreviations, and even synonyms are also detected and aligned. A score is assigned for each token's alignment measuring the framework's confidence in the match.

Alignment stage starts with a "traditional" sequence alignment, which aligns tokens that share the same orthography (Altschul et al., 1990). Their alignment score is set to 1. Tokens differing in orthography take the edit distance ratio between them as the score. Next we try to detect missing spaces. Word separation varies widely in manuscripts. That in turn occasionally causes recognition to merge two words into a single one (missing the space in between) or to wrongly detect a space and split one word into two. The framework will split or merge tokens according to the missing spaces and reduce the score relatively. Lastly, we try to align non-identical tokens and assign a score accordingly. Aligned synonyms, acronyms, or abbreviations share the confidence level of their surrounding tokens. If a token is not in the lexicon, the score is boosted.

Proposing readings. The final step is to choose the best reading for each token. Here we use majority vote between all available readings for each token. In this step only alignments that exceed a predefined threshold are included in the voting process. Preliminary results shows that our framework reduced the word error rate (WER) by 10%. The texts generated by the automatic transcription reached 81% of word level accuracy, while the new reading proposed by our text reuse framework boosted the accuracy to 91%.

4.3 NLP-Based Correction

We consider three approaches to language-based correction: (1) spellcheck, (2) pseudo-ensemble, and (3) shallow fusion.

[2]The original models are available on kraken's Zenodo archive, https://zenodo.org/communities/ocr_models/records.

Spellcheck. Given an input text potentially containing errors due to OCR inaccuracies, the algorithm attempts to correct the text by utilizing the predictive capabilities of a masked language model, namely BEREL.[3]

The algorithm is parameterized by:

• k, representing the number of top candidates to consider during the mask-filling process.

• θ_{rel} and θ_{abs}, relative and absolute Levenshtein distance (LD) thresholds, respectively, used to filter out implausible corrections.

• θ_{BERT}, an initial score threshold, for accepting or rejecting a correction based on the model's prediction score.

• A switch, whether to use regular LD or weighted LD (meaning that frequent recognition confusions, caused by graphical similarity, are assigned a lower weight), thus facilitating their correction by the algorithm.

The algorithm follows the following steps:

1. *Preprocessing*: The input text undergoes preprocessing to replace certain special characters and manage line breaks.
2. *Word Masking*: At each word position i in the input text, the word is masked using the tokenizer's mask token (usually "[MASK]").
3. *Model Prediction*:
(a) The masked text is passed through the language model.
(b) The algorithm retrieves the top k predictions for the masked token based on the logits from the "model".
4. *Correction Decision*:
(a) If the original word (prior to masking) is within the top k predictions, it is retained.
(b) Otherwise, a decision is made based on the LD between the original word w and each candidate c: If $\text{LD}(c, w) \leq \theta_{\text{abs}}$ and $\text{LD}_{\text{abs}}(c)/|w| < \theta_{\text{rel}}$, the candidate is deemed plausible.
(c) Among the plausible candidates, if the top candidate's score exceeds θ_{BERT}, it replaces the original word. If not, the original word is left intact, but alternatives are noted for potential review.
5. *Threshold Update*:
(a) The score associated with accepted predictions is stored.
(b) θ_{BERT} is updated based on the mean of these accepted scores, allowing the algorithm to dynamically adapt its confidence threshold.

The corrected text is returned. Additionally, for each line in the input, potential alternatives are provided by the system for manual review.

Pseudo-ensemble. Given an OCR model that generates output sequences, our objective is to generate alternative readings and to rank them to yield an enhanced prediction.

We generate many alternative readings using connectionist temporal classification (CTC) beam search, and select the best output using LM scoring. (This general method bears some similarity with test time augmentation. However, typically test-time augmentation is applied to the input, whereas we apply the transformations on the model's output.) This design choice was influenced by compute and latency constraints.

We evaluate two scoring algorithms, both based on perplexity, with or without normalization. We compare three versions of BEREL: v1, v1.5, and v2. Overall, this results in six scoring methods based on BEREL.

In the course of CTC decoding one can use beam search—that is, accumulate iteratively at each decoding step multiple highest-scoring possible outputs. After creating candidates at each step (the previous possible outputs, concatenated with any new token), only the b most probable outputs are kept. We can leverage this technique to generate multiple recognition candidates of a line. These b candidates, called the "beam width", can be considered the "best guesses" of the model. We then score every candidate using an LM, and return the candidate with the highest score.

Overall, the parameters of this algorithm are quite limited:

1. The number of candidates to generate, which we fix to be equal to the beam width. (In theory, the number of candidates can be any number. However, choosing a number smaller than the beam width means generating candidates but not evaluating them; choosing a higher number means adding candidates which differ only in the last token.)
2. The specific scoring model, of which, as mentioned, there are six.

Shallow fusion. In this third technique, we combine LM scores into the CTC decoding at inference. In practice, when decoding through the logit matrix, if the new character is a space we add to the logit the score given by a LM. This approach is also called in the literature "prefix beam search decoding with language model", and is similar to (Hannun

[3] https://huggingface.co/dicta-il/BEREL_2.0.

HTR model	Levenshtein thresholds	Scoring method	Original CAR	Improved CAR	Change CAR
Base	1 ; 0.6	BEREL v2	83.2	83.7	0.5
Fine-tuned			96.1	96.1	0.0

Table 1: Spellcheck—character accuracy change on Genève 146 holdout test set.

et al., 2014). It can be applied to the beam search decoding algorithm.

The algorithm is parameterized by:

- The language model, which may be BEREL v1, v1.5, or v2.
- The scoring method.
- The weight of the score to be added.

We consider the same scoring methods as in the previous case.

5 Experimental Setup

We perform first-pass HTR using both base and fine-tuned HTR models, in order to examine how the proposed methods can improve the standard measures, word accuracy rate (WAR) and character accuracy rate (CAR). Our main metrics are the changes in the accuracy rates, which means that we seek to have the highest possible positive change in WAR and CAR.

5.1 Model Choice and Fine-Tuning

Our experiments indicate that character accuracy can be boosted by around 18% by fine-tuning the recognition models over three labeled pages (see Figure 1). The particular choice of the source model does not seem to impact performance, nor adding more labeled data. We note that the same technique can be applied to segmentation models.

5.2 Post-correction Results

We performed tests on the manuscript Genève Comites Latentes 146 (or "Genève 146") (Bibliothèque de Genève), which contains a rabbinic homiletic work from late antiquity, *Midrash Tanḥuma*, in an Oriental Hebrew script of the 14th century. We determined the optimal parameters for spellcheck and pseudo-ensemble using exhaustive grid search, and for shallow fusion using random search. The parameter search was performed on a validation set. The results over a held-out test set are given in Tables 1–4. An example of a spacing correction and of a correct letter replacement are given in Figures 5; a misguided word split (albeit minor) is shown in Figure 6.

HTR model	Levenshtein thresholds	Scoring method	Original WAR	Improved WAR	Change WAR
Base	1 ; 0.6	BEREL v1.5	52.8	55.4	2.6
Fine-tuned		BEREL v2	88.3	88.3	0.0

Table 2: Spellcheck—word accuracy change on Genève 146 holdout test set.

HTR model	Number of candidates	Scoring method	Original CAR|WAR	Improved CAR|WAR	Change CAR|WAR
Base	50	BEREL v1.5	83.2 | 52.8	83.6 | 54.6	0.3 | 1.8
Fine-tuned		BEREL v3	96.1 | 88.3	96.1 | 89.3	<0.1 | 1.0

Table 3: Pseudo-ensemble—character and word accuracy changes on Genève 146 holdout test set.

HTR model	Parameters α ; size	Scoring method	Original CAR	Affected CAR	Change CAR
Base	5 ; 10	BEREL v2	85.1	83.4	−1.7
Fine-tuned			95.5	91.9	−3.6

Table 4: Shallow fusion—character accuracy change on Genève 146 holdout test set.

GT	ותרץ את גלגלתו ואם יהיה גבור שאין בכל הגיבורים
HTR	ותרץ את גלגלתו ואם יהיהגבור שאין בכל הגיבורים
PE	ותרץ את גלגלתו ואם יהיה גבור שאין בכל הגיבורים
GT	גדול על כל הארץ ובני אדם למטן על כן יהיו דבריך
HTR	גדול כל הארץ ובני אדם למטן על כן יהיו דבריך
PE	גדול על כל הארץ ובני אדם למטן על כן יהיו דבריך

Table 5: Examples of correct modifications using pseudo-ensemble on Genève 146.

In all cases, it is more challenging to improve on the fine-tuned recognition model's output. A possible interpretation would be that since the recognition model learns some linguistic features, its output has the appearance of acceptability, which fools the various logics. In the case of pseudo-ensemble, we note that the word accuracy changes by a much higher proportion than character accuracy. This indicates that the corrections effected by the LM have a light touch, mostly correcting words with few erroneous letters. Changing only such critical characters may weakly impact character accuracy but will strongly improve WAR.

An interesting case is that of Figure 2. The manuscript originally contained a spelling error, together with an interlinear correction. Recognition captures the original incorrect reading, missing the correction. The pseudo-ensemble correction replaces a letter in the word with the one between the lines, which gives the correct semantics, but not the correct reading. Actual spelling mistakes in medieval manuscripts are not uncommon, and are often left uncorrected in the text.

Shallow fusion demonstrated efficacy exclusively

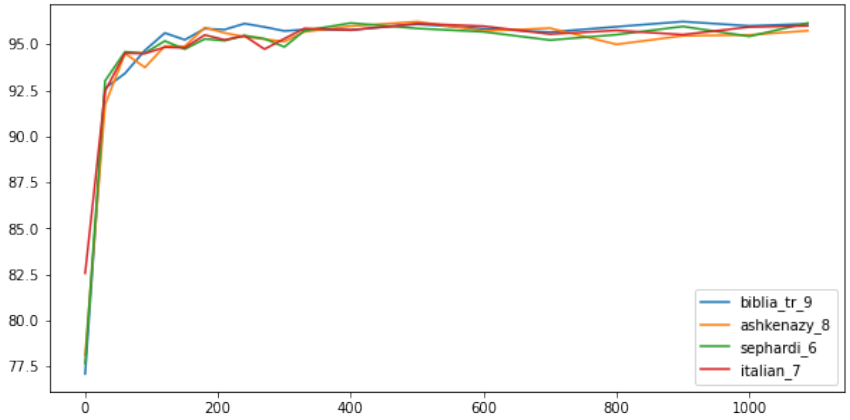

Figure 1: Character accuracy achieved by transfer learning, as a function of additional labeled lines used for fine-tuning. Models courtesy (Stökl Ben Ezra et al., 2021).

GT	להעיד ואם לאו אני מעיד שני והוא עד ומנין שה'ב'ה' נקרא
HTR	העיד ואם לאו אני מערד שני והוא עד ומנין שה'ב'ה' נקרא
PE	העיד ואם לאו אני מערד שני והוא עד ומנין שה' ב'ה' נקרא

Table 6: Example of incorrect modifications using pseudo-ensemble on Genève 146.

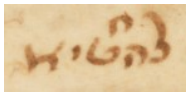

Figure 2: HTR read להטיא, ignoring the interlinear ח; pseudo-ensemble changed it to לחטיא, which has the same meaning as the correct להחטיא.

on the training dataset for a particular configuration of parameters, as indicated by a slight enhancement in CAR. However, this methodology lacked sufficient robustness and failed to generalize effectively to the holdout test dataset. This experiment may, accordingly, be deemed unsuccessful, and we hypothesize that the failure can be attributed to the utilization of a bidirectional critical model for scoring outputs. In contrast, we posit that the appropriate approach necessitates the use of conditional scores generated by a conditional generative model, such as DictaLM for Rabbinic Hebrew (Shmidman et al., 2023, 2024). Moreover, the limited context (parts of lines) available to the model at the rescoring stage may have impaired its capabilities.

5.3 Expert Proofreading

Following automatic transcription, a semi-automatic component allows experts to proof-read uncertain results. As detailed above, suspect results and possible corrections are suggested by the automatic components. This integration of a machine-aided person-in-the-loop allows for the efficient allocation of human expertise and effort.

Method model	Original WAR	Improved WAR	Change WAR
Text reuse	70.4	74.7	4.3
Spellcheck		72.8	2.4
SC then TR		75.3	4.9
TR then SC		75.2	4.8

Table 7: Word accuracy changes on Vatican 44. Text reuse compared with spellcheck, and text reuse combined with language model corrections.

5.4 Combining Methods

Our assessment of the text reuse algorithm was conducted on Vatican 44 (Biblioteca Apostolica Vaticana), a 14th-century Midrash compilation. The first two rows of Table 7 present the enhancements in word accuracy rate achieved through text reuse, in contrast to the spellcheck method applied to the identical dataset. The baseline for comparison is established by the base BibliA HTR model, which was not fine-tuned. The last two rows consider the permutations of the two approaches, specifically evaluating the sequence of implementation for text reuse (TR) and spellcheck (SC). Overall, leveraging text reuse resulted in more corrections that did language modeling alone. Combining the two gave the best of both worlds.[4]

6 Conclusions

The pipeline proposed here aims to improve the accessibility of historical manuscripts in a machine

[4]Our methods, models, and results are archived at https://gitlab.com/millerhadar/textreusefortranscription, https://gitlab.com/millerhadar/soferllmcorrection, and https://github.com/anutkk/sofer-stam.

readable form. Text-reuse detection, as a post-processing component, substantially improves the overall transcription, though it can easily introduce errors. The immediate gains are twofold: (1) The method minimizes the expert manual labor required to validate and correct the transcription, which in turn is utilized to fine-tune the models and improve accuracy. (2) The accuracy level reached automatically might be acceptable for use as is, without a manual pass. Given the flexibility of contemporary search engines, we expect that even imperfect text will significantly improve the accessibility of texts and images, a boon to both scholars and the wider public.

The efficiency of the pipeline we designed depends on the type of the text. (a) Manuscripts of familiar works only demand identification of the work and alignment of the entire work with the manuscript text, expected to be produced. Work on aligning text with images includes (Cohen et al., 2015; Ben-Shalom et al., 2017). (b) Manuscripts of an anthological nature demand further scrutiny, identifying the most probable source of each paragraph. (c) Compilations will benefit less from the search for textual parallels. It may be expected that with additional fine-tuning of the reference library and with better text-reuse thresholds and language models, the accuracy of the post-processing could be increased further.

The work described herein is continuing within the framework of the large-scale MiDRASH ERC Synergy project (Vasyutinsky-Shapira et al., 2024), led by Daniel Stökl Ben Ezra, Judith Olszowy-Schlanger, Nachum Dershowitz, and Avi Shmidman, in coöperation with Moshe Lavee and the National Library of Israel. Using the Ktiv manuscripts as its starting point, it aims to make the contents of preprint Hebrew-character (Hebrew, Aramaic, Judeo-Arabic, etc.) manuscripts accessible, with a primary focus on biblical, exegetical, and liturgical manuscripts. Model selection, post-OCR correction, and model refinement will be automated. Linguistic and paleographic analyses will also be performed.

References

Wissam AlKendi, Franck Gechter, Laurent Heyberger, and Christophe Guyeux. 2024. Advancements and challenges in handwritten text recognition: A comprehensive survey. *Journal of Imaging*, 10(1):18.

Jonathan Parkes Allen, Matthew Thomas Miller, John Mullan, and David Smith. 2022. Digitizing the Islamicate written traditions: History, state of the field, and best practices for open-source Arabic-script OCR. White paper AOCP Phase I White Paper v. 1.1, Open-ITI.

Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. 1990. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410.

Adam Amram, Anat Ben David, and Reut Tsarfaty. 2018. Representations and architectures in neural sentiment analysis for morphologically rich languages: A case study from Modern Hebrew. In *Proc. of the 27th International Conference on Computational Linguistics*, pages 2242–2252.

José Carlos Aradillas, Juan José Murillo-Fuentes, and Pablo M. Olmos. 2021. Boosting offline handwritten text recognition in historical documents with few labeled lines. *IEEE Access*, 9:76674–76688.

Adiel Ben-Shalom, Adi Silberpfennig, Nachum Dershowitz, Lior Wolf, and Yaacov Choueka. 2017. Querying Hebrew texts via word spotting. In *World Congress of Jewish Studies*, Jerusalem, Israel.

Biblioteca Apostolica Vaticana. Midrash Tanḥuma. Ms. Vat.ebr.44.pt.1.

Bibliothèque de Genève. Midrash Tanhuma (Leviticus-Numbers-Deuteronomy). Ms. Comites Latentes 146.

Oran Brill, Moshe Koppel, and Avi Shmidman. 2020. FAST: Fast and accurate synoptic texts. *Digital Scholarship in the Humanities*, 35(2):254–264.

Marco Büchler, Philip R. Burns, Martin Müller, Emily Franzini, and Greta Franzini. 2014. Towards a historical text re-use detection. In *Text Mining*, pages 221–238. Springer.

Avihay Chriqui and Inbal Yahav. 2022. HeBERT and HebEMO: A Hebrew BERT model and a tool for polarity analysis and emotion recognition. *INFORMS Journal on Data Science*, 1(1):81–95.

Paul Clough, Robert Gaizauskas, Scott S. L. Piao, and Yorick Wilks. 2002. METER: MEasuring TExt Reuse. In *Proceedings of 40th Anniversary Meeting for the Association for Computational Linguistics*, pages 152–159.

Rafi Cohen, Irina Rabaev, Jihad El-Sana, Klara Kedem, and Itshak Dinstein. 2015. Aligning transcript of historical documents using energy minimization. In *Proc. 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 266–270. IEEE.

Giovanni Colavizza, Mario Infelise, and Frédéric Kaplan. 2014. Mapping the early modern news flow: An enquiry by robust text reuse detection. In *International Conference on Social Informatics*, pages 244–253. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 4171–4186.

Kartik Dutta, Praveen Krishnan, Minesh Mathew, and C. V. Jawahar. 2018. Improving CNN-RNN hybrid networks for handwriting recognition. In *16th International Conference on Frontiers in Handwriting Recognition*, pages 80–85. IEEE.

Tomáš Foltỳnek, Norman Meuschke, and Bela Gipp. 2019. Academic plagiarism detection: A systematic literature review. *ACM Computing Surveys*, 52(6):1–42.

Shaona Ghosh and Per Ola Kristensson. 2017. Neural networks for text correction and completion in keyboard decoding. *arXiv preprint arXiv:1709.06429*.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Adeline Granet, Emmanuel Morin, Harold Mouchère, Solen Quiniou, and Christian Viard-Gaudin. 2018. Transfer learning for handwriting recognition on historical documents. In *7th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*.

Eylon Gueta, Avi Shmidman, Shaltiel Shmidman, Cheyn Shmuel Shmidman, Joshua Guedalia, Moshe Koppel, Dan Bareket, Amit Seker, and Reut Tsarfaty. 2022. Large pre-trained models with extra-large vocabularies: A contrastive analysis of Hebrew BERT models and a new one to outperform them all. *arXiv preprint arXiv:2211.15199*.

Yaakov HaCohen-Kerner, Ariel Kass, and Ariel Peretz. 2004. Baseline methods for automatic disambiguation of abbreviations in Jewish law documents. In *Advances in Natural Language Processing*, pages 58–69. Springer.

Yaakov HaCohen-Kerner, Nadav Schweitzer, and Dror Mughaz. 2011. Automatically identifying citations in Hebrew-Aramaic documents. *Cybernetics and Systems: An International Journal*, 42(3):180–197.

Kai Hakala, Aleksi Vesanto, Niko Miekka, Tapio Salakoski, and Filip Ginter. 2019. Leveraging text repetitions and denoising autoencoders in OCR post-correction. *arXiv preprint arXiv:1906.10907*.

John C. Handley and Thomas B. Hickey. 1991. Merging optical character recognition outputs for improved accuracy. In *RIAO '91: Intelligent Text and Image Handling*, pages 160–174, Paris. Le Centre de Hautes Etudes Internationales d'Informatique Documentaire.

Awni Hannun. 2017. Sequence modeling with CTC. *Distill*. https://distill.pub/2017/ctc.

Awni Y. Hannun, Andrew L. Maas, Daniel Jurafsky, and Andrew Y. Ng. 2014. First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs. *arXiv preprint arXiv:1408.2873*.

Alon Itai and Shuly Wintner. 2008. Language resources for Hebrew. *Language Resources and Evaluation*, 42(1):75–98.

José Carlos Aradillas Jaramillo, Juan José Murillo-Fuentes, and Pablo M. Olmos. 2018. Boosting handwriting text recognition in small databases with transfer learning. In *Proc. 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 429–434. IEEE.

Philip Kahle, Sebastian Colutto, Günter Hackl, and Günter Mühlberger. 2017. Transkribus – A service platform for transcription, recognition and retrieval of historical documents. In *Proc. 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 4, pages 19–24. IEEE.

Omri Keren, Tal Avinari, Reut Tsarfaty, and Omer Levy. 2022. Breaking character: Are subwords good enough for MRLs after all? *arXiv preprint arXiv:2204.04748*.

Benjamin Kiessling. 2019. Kraken – an universal text recognizer for the humanities. In *Digital Humanities (DH2019)*.

Benjamin Kiessling, Robin Tissot, Peter Stokes, and Daniel Stökl Ben Ezra. 2019. eScriptorium: An open source platform for historical document analysis. In *Proc. International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 19–24. IEEE.

Ido Kissos and Nachum Dershowitz. 2016. OCR error correction using character correction and feature-based word classification. In *Proc. 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 198–203. IEEE.

Ido Kissos and Nachum Dershowitz. 2017. Image and text correction using language models. In *Proc. 1st International Workshop on Arabic Script Analysis and Recognition*, pages 158–162. IEEE.

Benjamin Eliot Klein, Nachum Dershowitz, Lior Wolf, Orna Almogi, and Dorji Wangchuk. 2014. Finding inexact quotations within a Tibetan Buddhist corpus. In *Digital Humanities*, pages 486–488.

Tsvi Kuflik, Moshe Lavee, Daniel Stoekl Ben Ezra, Avigail Ohali, Vered Raziel-Kretzmer, Uri Schor, Alan Wecker, Elena Lolli, and Pauline Signoret. 2019. Tikkoun Sofrim combining HTR and crowdsourcing for automated transcription of Hebrew medieval manuscripts. In *Digital Humanities (DH2019)*.

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

Berat Kurar Barakat, Jihad El-Sana, and Irina Rabaev. 2019. The Pinkas dataset. In *Proc. International Conference on Document Analysis and Recognition (ICDAR)*, pages 732–737. IEEE.

Jey Han Lau, Carlos Armendariz, Shalom Lappin, Matthew Purver, and Chang Shu. 2020. How furiously can colorless green ideas sleep? Sentence acceptability in context. *Transactions of the Association for Computational Linguistics*, 8:296–310.

John S. Y. Lee. 2007. A computational model of text reuse in ancient literary texts. In *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 472–479.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710. In Russian, translated into English.

Shahar Mahpod and Yosi Keller. 2018. Auto-ML deep learning for Rashi scripts OCR. *CoRR*, abs/1811.01290.

Hadar Miller, Tsvi Kuflik, and Moshe Lavee. 2025. Text alignment in the service of text reuse detection. *Applied Sciences*, 15(6).

Amir More, Amit Seker, Victoria Basmova, and Reut Tsarfaty. 2019. Joint transition-based models for morpho-syntactic parsing: Parsing strategies for MRLs and a case study from Modern Hebrew. *Transactions of the Association for Computational Linguistics*, 7:33–48.

National Library of Israel. 2021. Digitized Hebrew manuscripts.

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 *Proc. of the ACM/IEEE Joint Conference on Digital Libraries*, pages 333–336.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Christian Reul, Dennis Christ, Alexander Hartelt, Nico Balbach, Maximilian Wehner, Uwe Springmann, Christoph Wick, Christine Grundig, Andreas Büttner, and Frank Puppe. 2019. OCR4all—An open-source tool providing a (semi-)automatic OCR workflow for historical printings. *Applied Sciences*, 9(22).

Stephen V. Rice, Junichi Kanai, and Thomas A. Nartker. 1994. An algorithm for matching OCR-generated text strings. In *Document Image Analysis*, pages 263–272. World Scientific.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*.

Sarah Bowen Savant. 2016. The history of Arabic books in the digital age. *British Academy Review*, 28.

Sefaria, Inc. 2021. A living library of Torah texts online.

Amit Seker, Elron Bandel, Dan Bareket, Idan Brusilovsky, Refael Greenfeld, and Reut Tsarfaty. 2022. AlephBERT: Language model pre-training and evaluation from sub-word to sentence level. In *Proc. of the 60th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 46–56.

Avi Shmidman, Joshua Guedalia, Shaltiel Shmidman, Cheyn Shmuel Shmidman, Eli Handel, and Moshe Koppel. 2022. Introducing BEREL: BERT embeddings for rabbinic-encoded language. *arXiv preprint arXiv:2208.01875*.

Shaltiel Shmidman, Avi Shmidman, Amir D. N. Cohen, and Moshe Koppel. 2024. Adapting LLMs to Hebrew: Unveiling DictaLM 2.0 with enhanced vocabulary and instruction capabilities. *arXiv preprint arXiv:2407.07080*.

Shaltiel Shmidman, Avi Shmidman, Amir David Nissan Cohen, and Moshe Koppel. 2023. Introducing DictaLM – A large generative language model for Modern Hebrew. *Preprint*, arXiv:2309.14568.

Michal Bar-Asher Siegal and Avi Shmidman. 2018. Reconstruction of the Mekhilta Deuteronomy using philological and computational tools. *Journal of Ancient Judaism*, 9(1):2–25.

David A. Smith, Ryan Cordel, Elizabeth Maddock Dillon, Nick Stramp, and John Wilkerson. 2014. Detecting and modeling local text reuse. In *IEEE/ACM Joint Conference on Digital Libraries*, pages 183–192.

David A. Smith, Ryan Cordell, and Elizabeth Maddock Dillon. 2013. Infectious texts: Modeling text reuse in nineteenth-century newspapers. In *IEEE International Conference on Big Data*, pages 86–94.

David A. Smith, Ryan Cordell, and Abby Mullen. 2015. Computational methods for uncovering reprinted texts in antebellum newspapers. *American Literary History*, 27(3):E1–E15.

Daniel Stökl Ben Ezra, Bronson Brown-DeVost, Pawel Jablonski, Hayim Lapin, Benjamin Kiessling, and Elena Lolli. 2021. BiblIA – a general model for medieval Hebrew manuscripts and an open annotated dataset. In *The 6th International Workshop on Historical Document Imaging and Processing*, pages 61–66.

Omri Suissa, Avshalom Elmalech, and Maayan Zhitomirsky-Geffet. 2020. Optimizing the neural network training for OCR error correction of historical Hebrew texts. In *iConference 2020 Proceedings*. iSchools.

Daria Vasyutinsky-Shapira, Berat Kurar-Barakat, Sharva Gogawale, Mohammad Suliman, and Nachum Dershowitz. 2024. MiDRASH – A project for computational analysis of medieval Hebrew

manuscripts. In *EUROGRAPHICS Workshop on Graphics and Cultural Heritage*.

Aleksi Vesanto, Asko Nivala, Heli Rantala, Tapio Salakoski, Hannu Salmi, and Filip Ginter. 2017. Applying BLAST to text reuse detection in Finnish newspapers and journals, 1771–1910. In *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language*, pages 54–58, Gothenburg. LiU Electronic Press.

Mauricio Villegas, Alejandro H. Toselli, Verónica Romero, and Enrique Vidal. 2016. Exploiting existing modern transcripts for historical handwritten text recognition. In *Proc. 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 66–71. IEEE.

Alan J. Wecker, Vered Raziel-Kertzmer, Daniel Stökl Ben Ezra, Moshe Lavee, Tsvi Kuflik, Dror Elovits, Moshe Schorr, Uri Schor, and Pawel Jablonski. 2022. Tikkoun Sofrim: Making ancient manuscripts digitally accessible: The case of Midrash Tanhuma. *ACM Journal of Computation and Cultural Heritage*, 15(2).

John Wilkerson, David Smith, and Nicholas Stramp. 2015. Tracing the flow of policy ideas in legislatures: A text reuse approach. *American Journal of Political Science*, 59(4):943–956.

Thomas Zenkel, Ramon Sanabria, Florian Metze, Jan Niehues, Matthias Sperber, Sebastian Stüker, and Alex Waibel. 2017. Comparison of decoding strategies for CTC acoustic models. *arXiv preprint arXiv:1708.04469*.

Alex Zhicharevich. 2011. Tools to aid OCR of Hebrew character manuscripts. Master's thesis, The Blavatnik School of Computer Science, Tel Aviv University, February.