Version 5 of the Kraken ATR Engine for the Humanities

Benjamin Kiessling

Version 5 of the Kraken ATR Engine for the Humanities

Benjamin Kiessling[1,2,3][0000−0001−9543−7827]

[1] ALMAnaCH, Centre Inria de Paris, France
[2] École Pratique des Hautes Études, Université PSL, Paris, France
[3] UMR 8546 CNRS-Université PSL (ENS-EPHE) - Aoroc - Paris, France
benjamin.kiessling@ephe.psl.eu

Abstract. Automatic text recognition for contemporary and historical printed or handwritten works has become a crucial tool in the inventory of many humanities scholars employing digital methods, but few retrodigitization software packages are adapted to this environment with its unparalleled diversity, domain-specific conventions, and low-resource settings.
We examine how the design principles behind kraken, a freely-licensed ATR engine optimized for use on historical and non-Latin script documents, pertain to specific use cases encountered in the humanities and allow it to be easily adapted to process even highly non-conventional material. In addition, we shine a light on the significant functional additions, performance enhancements, and quality of life improvements added in its recent version 5 stable releases. Among these are advanced reading order support including trainable reading order, unsupervised recognition pretraining, and a new public model repository with enhanced metadata. The latest release of the software can be found at https://kraken.re.

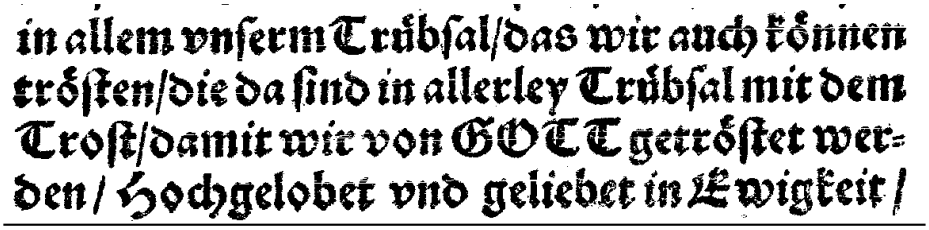
Keywords: Handwritten Text Recognition · Layout Analysis · Reading Order · Open Source Software

1 Introduction

Automatic text recognition[4] has emerged as a pivotal tool in the digital age, diffused widely in all kinds of applications bridging the gap between the analog and digital, facilitating the conversion of writing, no matter handwritten or machine-printed, in a variety of languages and writing systems that has never been seen before. Naturally, the humanities and in particular the historical sciences have not been untouched by these developments.

Nevertheless, the use of ATR technology in these environments poses particular challenges that are often insufficiently addressed by methods that have been designed for contemporary material written in major languages following

[4] Automatic text recognition (ATR) is used as an umbrella term comprising optical character recognition (OCR) and handwritten text recognition (HTR).



Editorial choices	Transcription
Retaining ligatures, distinction between long (ſ) and round (s), no normalization of u and v	in allem vnſerm Trůbſal/ das wir auch kõnnen tröſten/ die da ſind in allerley Trůbſal mit dem Troſt/damit wir von GOTT getröſtet wer- den/ Hochgelobet vnd geliebet in Ewigkeit/
Modernization of u-v, s-forms, umlauts, and hyphenation, dissolution of ligatures	in allem unserm Trübsal/ das wir auch können trösten/ die da sind in allerley Trübsal mit dem Trost/damit wir von GOTT getröstet wer- den/ Hochgelobet und geliebet in Ewigkeit/

Fig. 1. Examples of editorial choices resulting in different transcriptions for an early modern German print. Interventions are underlined.

modern conventions in terms of layout, typography and calligraphy, or orthography. The reasons for these deficiencies are numerous but can be grouped into two large categories: assumptions on *how a text functions* and assumptions on *how it should be transcribed*. The first category is rather self-evident. Modern writing practices have become highly standardized, even across languages, with a near-global understanding that writing happens on rectilinear lines, in a single direction, with clear rules on how text is segmented into paragraphs, sentences, clauses, and words, and fixed orthographic inventories. Arising both from a lack of awareness and a desire to boost the quality of recognition results cheaply, it is therefore common practice to implicitly or explicitly incorporate these norms when designing data models and methods. Unfortunately, historical and minority, non-Western writing tends to exhibit less uniformity and frequently contravenes the expectations with which these systems have been built. Degradations from these heuristics range from mild, such as a language model overcorrecting historical language use towards modern norms, to rendering outputs completely unusable, for example applying layout analysis methods detecting lines as horizontal bounding boxes on vertical, e.g., Mongolian writing.

The second category of deficiencies often arises from an overly simplified view of the transcription process. In the computer science domain, transcription, the conversion of analog writing into a digital representation of a text, is usually treated like a lossless and, importantly, neutral mapping between the glyphs on the page and their digital representation, which means that for any particular text on a page image an ATR system should produce a universally accepted single correct digital transcription. This position has wide-ranging impact on the design of text recognition methods, in particular when it comes to training and

fine-tuning schemes, but clashes with the conception prevalent in the humanities of transcription as an editorial process where the text on a page is transformed with an ensemble of choices into a transcription that serves a particular scientific purpose. This notion is not particularly intuitive to computer scientists but can easily be demonstrated (see figure 1). While there have been some efforts to establish common transcription guidelines for particular fields, such as CATMuS in the medieval sciences [**?**], transcriptions are fundamentally not interchangeable. A single document can validly be transcribed in a multitude of ways and humanities users of an ATR system will expect that system to produce output with editorial choices adapted to their particular scientific investigation.

The combination of these two properties means that off-the-shelf ATR systems are in many aspects unsuitable to retrodigitization work in the humanities. While for larger individual research projects the path of ad-hoc solutions developed from scratch or cobbled together from research implementations adapted to specific material is viable, this approach is laborious and leads to a significant duplication of effort as research outcomes such as datasets and software artifacts are unlikely to be reusable in other contexts.

Therefore, ATR software specifically tailored towards the requirements of humanities research is a prerequisite towards the effective utilization of this technology in this domain. Further, it allows not only the unidirectional diffusion of computer vision research into the hands of humanities scholars but is also a channel for computer vision researchers to identify research needs, assemble interchangeable datasets for low-resource material collaboratively, and incorporate novel methods in a familiar framework.

The kraken software is such an ATR package. Freely available under an Apache license, it is designed to be as language- and script-independent as possible. Nevertheless, it concedes through a modular architecture which allows the easy replacement of functional blocks that there will always be some material that the default methods struggle to process effectively. It is widely used in the humanities to retrodigitize machine-printed and handwritten text documents in dozens of writing systems sourced from all periods of history. The most recent 5th stable release includes a number of advances that add new functionality such as trainable reading order and unsupervised pretraining, improves recognition performance, and aids in model selection and distribution.

2 Related Work

Retrodigitization of historical and non-Western handwritten and printed text is an active and established research field with a significant body of literature. Due to the pipeline nature of a typical ATR workflow where a chain of functional blocks (preprocessing, layout analysis, recognition, postprocessing) is required to transform an input image into a digital text, this section is split into two parts: one describing research of component parts of this processing pipeline and the other integrative work aimed at producing end-to-end systems from these functional blocks.

2.1 Historical and non-Western ATR Methods

Conventional ATR research is often focused on modern material written in majority languages using the Latin script, but a substantial body of literature on historical and non-Western retrodigitization exists. Unfortunately, these methods frequently take a very narrow view of the research problem and are optimized for a single language, script, or historical period. As this holds in particular for functions that fall outside the core layout analysis and text recognition steps, e.g., postcorrection with linguistic modelling, this section will address only these two.

Layout analysis consists of multiple sub-tasks, some of which like line or region detection have been extensively studied while others like reading order determination have largely been ignored by the research community. The most common general purpose text line detection paradigm in the historical ATR domain follows the baseline detection scheme initially proposed in its modern form with a comprehensive Latin-script dataset and evaluation method in [**?**]. Following the overall trend of computer vision research, methods based on deep neural networks have risen in popularity over the last couple of years. Frequently, these are built on U-Nets[**?**] semantic segmentation backbones followed by postprocessing with various conventional computer vision techniques. Examples include [**?**,**?**], but other approaches have been evaluated as well such as recurrent convolutional neural networks [**?**], generative adversarial networks [**?**], and end-to-end detection[**?**]. While many methods are designed exclusively for baseline detection, the semantic segmentation approach lends itself to region segmentation and combined baseline and region segmentation methods have been proposed [**?**,**?**].

Text recognition for handwriting and machine-printed documents is a heavily researched subject with hundreds of methods proposed over the decades and has reached a state of maturity and robustness where algorithms do not require adaptation to the specificities of historical material. Since the creation of the connectionist temporal classification (CTC) loss [**?**], which allows alignment-free sequence modelling dispensing with the need to segment to the character level, line-wise text recognition where a sequence of characters is predicted from a whole line image at once has become the near universal paradigm. In this general framework, the most common class of methods uses hybrid convolution and recurrent neural networks. Dozens of variations of this basic construction exist; [**?**,**?**] are mentioned as representative examples, with only minor differences in accuracy. Multiple methods [**?**,**?**,**?**] incorporating transformer-style networks using autoregressive prediction dispensing with both recurrent layers and CTC-based training have been described in recent literature but have as of yet not seen widespread use in practical applications. Lastly, end-to-end methods that operate on whole page images without image segmentation[**?**,**?**,**?**] are starting to appear. While the promise of truly segmentation-free ATR is enticing, these methods require enormous training datasets both for initial training and fine-tuning which make their use on historical material impractical.

2.2 Historical and non-Western ATR Software

A few software packages integrating all necessary functions for an end-to-end ATR pipeline aimed at historical material exist, albeit their scope and target audience is often limited. Frequently, these packages do not only provide solely ATR functionality but as they are intended to be used autonomously by scholars with minimal technical expertise and training also incorporate ancillary functions such as training data annotation, editing, evaluation, publication, crowd-sourcing, etc., in an ergonomic web environment. While this is an obvious pathway to providing practical tools quickly, it comes with a lack of flexibility as it is difficult to reconcile user experience and exposing the full parametrization of a typical ATR workflow.

The OCR-D [**?**] project is a collaborative effort of German institutions in order to prepare a retrodigitization system to produce a retrospective national library of early modern, 16th-18th century CE, writing from the German-speaking countries. The approach of the consortium has been to create a workflow engine that allows the construction of individualized pipelines from functional blocks following pre-defined data interchange norms. Intended to be used primarily in archival settings, the software requires only minimal manual intervention, integrates into existing library infrastructure, and contains modules that aid in semi-automatic model selection for larger retrodigitization campaigns.

PERO OCR [**?**] is a freely-licensed OCR pipeline designed for the processing of historical printed documents developed as part of the PERO project. Layout analysis, text recognition, and language models to retrodigitize historical Czech newspapers scanned from low-quality microfilm are made available by the authors. A basic web-frontend for quality assurance and manual transcription correction is provided as well as an export into standard XML formats.

eScriptorium[**?**] is a digital text production pipeline for machine-printed and handwritten texts that utilizes kraken's ATR functionality and packages it in a platform that intends to be ergonomic enough to be used by humanities scholars without extensive machine learning or document analysis expertise while also allowing the processing of as diverse material as possible. Freely-licensed, supporting common standards in the retrodigitization community such as ALTO, PageXML, and METS, and exposing ATR's individual steps both in the interface and through an API, it is easily adapted and extended for novel use cases.

OCR4All[**?**] is a semi-automatic OCR workflow provided through a web-based platform which is specifically optimized for the retrodigitization of (Latin-script) historical prints. Built upon the Larex[**?**], the kraken layout analysis system, and the Calamari[**?**] recognizer, it allows users to retrodigitize machine-printed documents with minimal training.

Transkribus[**?**] models the entire ATR workflow including field and table recognition. As a proprietary solution that is locked behind a web-based platform, it suffers from the usual drawbacks presented by these for good scientific practices, from the impossibility of adaptation to specific use cases, a lack of ownership of artifacts produced with data entered on the platform, privacy concerns

when treating sensitive data, to questions of reproducibility and compliance with funders' open science mandates.

2.3 Conclusion

Methods, software, and platforms intended to solve specific tasks in humanities ATR are plentiful. Nevertheless, systems that are both complete, i.e., capture the entire processing chain from page image scans to a digital facsimile, and general enough to treat material originating from a wide range of writing cultures are much rarer. Both the ongoing creation of ad-hoc solutions for ATR on a research project level and the constant flow of literature proposing methods for particular collections of written material indicate that software such as kraken is fulfilling an unmet need.

3 Software Design

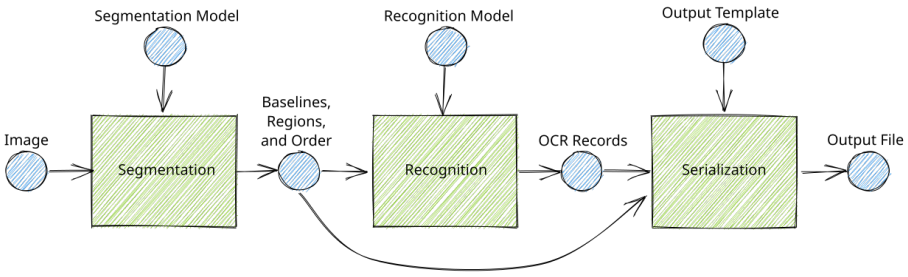


Fig. 2. The typical kraken inference workflow and its functional blocks (green) and artifacts (blue).

Kraken is an ATR engine written in Python utilizing standard scientific computing libraries such as PyTorch, SciPy, and Shapely to implement its computer vision functionality. Intended to be used both as a library to be selectively or wholly included as part of a larger application and a standalone software, it offers a stable API and a command line interface that expose the individual functional blocks of the ATR workflow. The overall structure of the workflow implemented in the software is shown in figure 2 and follows the established schema of text line and region detection and ordering followed by text recognition of individual lines. To address the two principal obstacles to ATR use in the historical sciences identified above, kraken is built on methods that are specifically designed to be as script- and language-agnostic as possible while at the same time adopting a modular architecture that allows easy replacement of functional blocks when these methods are found to be insufficient for particular types of documents.

Further, segmentation and recognition methods are chosen with the assumption that the ability to rapidly adapt for new layout analysis taxonomies and transcription guidelines with small training datasets is of utmost importance. As such, training data inefficient methods such as postcorrection with language modelling or transformer-based text recognizers are avoided as it is unlikely that they can be fine-tuned with the computational and data resources available in typical humanities research.

In order to make kraken as versatile as possible, great care has been taken to eliminate heuristics that rely on preconceived notions on how documents to be processed function and to make near-universal *sensible defaults* disengageable. A prime example is how the recognizer treats Unicode processing. As is common practice in text recognition methods, the recognizer predicts for an input line image a sequence of labels which are then mapped one by one into Unicode code points. Because the recurrent neural networks used for recognition have limited capacity to model long dependencies, these sequences of mapped code points are not emitted in the logical reading order of the text in a line but the display order[5], which requires them to be reshuffled with the Unicode BiDi algorithm before serialization. While these steps are adequate for the majority of text that one might digitize with an ATR system, there are use cases where deviation is required, for example when treating very large logo-ideographic writing systems such as Egyptian hieroglyphs or Chinese, where decomposition of a single code point into multiple labels boosts recognition accuracy, transliteration of a right-to-left script into a left-to-right script, e.g., Yiddish or Ottoman Turkish into Latin script, or when trying to recognize one of the more than hundred scripts that remain to be encoded by the Unicode consortium. Kraken allows arbitrary codecs mapping between labels and Unicode code point space in a many-to-many fashion supporting simultaneously decompositions, aggregations, and custom encoding schemes in the private use area in addition to enabling and disabling the BiDi algorithm, empowering scholars to modify and override processing steps that interfere with their desired transcription.

3.1 Data Interchange, Archival Formats, and Serialization

Essential to the integration of kraken into larger workflows is support for data interchange and archival file formats widespread in the retrodigitization domain while also permitting custom serialization formats. As the output of an ATR workflow is fundamentally a facsimile, even when truncated at earlier steps in the processing pipeline such as layout analysis, any reasonable output format is

[5] Unicode encodes code points in the order a human would read them (logical order), e.g., leftmost character first for a left-to-right script such as Latin, rightmost character for right-to-left scripts such as Hebrew and Arabic with special treatment of characters with soft directionality such as numerals and punctuation. The BiDi algorithm maps code point sequences containing left-to-right, right-to-left, or a mixture of both between the logical order they are stored in and the display order, i.e., the order from left-to-right they appear in when rendered for display.

likely to be similarly structured, which makes it possible to provide for a diverse set of standards with templates rendering the same basic record objects returned by the API. A number of templates for three open and commonly used ATR output formats are distributed as part of kraken: PageXML[**?**], ALTO[**?**], and hOCR[**?**], in addition to a basic rendering of the proprietary abbyyXML format. Further, external custom templates written in the Jinja templating language can be loaded by users transparently to render the results of any functional block into other text-based serialization formats, e.g., TEI.

3.2 Computer Vision Modules

The key computer vision techniques of interest implemented in kraken are the layout analysis and text recognition modules, as they are essential for the core function of the ATR system. Additional methods, such as a trainable reading order algorithm and unsupervised pretraining of the text recognizer, provide optional functionality and are intended for use in specific cases, such as when particular recognition needs arise or when manual annotation of training data is not feasible.

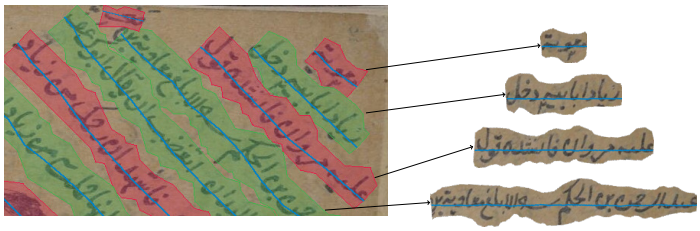


Fig. 3. Baseline (blue) + bounding polygon (red and green) line representation and normalization of line images for recognizer input.

Layout Analysis The layout analysis functional block is made up of a trainable joint text line and region detection system based on a semantic segmentation and postprocessing approach described in [**?**]. The text lines detected by the module are modelled as directed baselines and bounding polygons that are used to distinguish between line and non-line content. Regions are detected as arbitrarily shaped polygons. Both baselines and regions are detected jointly with a semantic segmentation neural network performing multi-label classification, which permits user-defined line and region typologies. In contrast to most other baseline detection algorithms which assume all lines on a document page are upright, line orientation is determined through boxes of auxiliary classes located at the end of each baseline, one class each for the start and end of the line, and is therefore learnable. A peculiarity of the method is that as each pixel can be assigned multiple labels, it can easily be used to detect overlapping regions (and baselines) with diverse semantics while conventional semantic segmentation-based region detectors are limited to non-overlapping entities.

The baseline paradigm is quite widespread among writing systems, but a number of exceptions such as Hebrew are written hanging from toplines instead or like Chinese are not aligned on a single line at all. Despite the possibility to approximate baselines for most of these, in order to offer a data model that conforms more closely to the expectations of users proficient in those scripts, the layout analysis module accommodates these choices through configuration switches and metadata deposited in the segmentation model files.

Reading Order One of the additions of the version 5 release is a trainable reading order functionality that supplements the default basic heuristic based on a simple topological ordering algorithm.

Table 1. Default text recognizer architecture

	Layer Type	Hyperparameters
0	Convolution	3×13 kernel size, 32 filters
1	Dropout	2D, $p = 0.1$
2	Maxpool	2×2 kernel size
3	Convolution	3×13 kernel size, 32 filters
4	Dropout	2D, $p = 0.1$
5	Maxpool	2×2 kernel size
6	Convolution	3×9 kernel size, 64 filters
7	Dropout	2D, $p = 0.1$
8	Maxpool	2×2 kernel size
9	Conv	3×9 kernel size, 64 filters
10	Dropout	2D, $p = 0.1$
11	LSTM	bidirectional, hidden size 200
12	Dropout	2D, $p = 0.1$
13	LSTM	bidirectional, hidden size 200
14	Dropout	2D, $p = 0.1$
15	LSTM	bidirectional, hidden size 200
16	Dropout	1D, $p = 0.5$
17	FC layer	variable size

The method is based upon a learnt pairwise order-relation operator [**?**] that predicts an approximate probabilistic adjacency matrix which is decoded into a directed path with a simple greedy algorithm. While this approach has limitations that make it unsuitable for generalized reading order models, in particular the inputs to the order-relation operator being features derived solely from line classification and position disconnected from any visual evidence, its radically simple construction makes it very efficient both in training data and computational requirements. Pairwise line sampling and the shallow neural network employed to predict the order-relation make it possible to train specialized reading order models with less than 10 pages in under 5 minutes on commodity hardware.

Reading order determination is closely linked to layout analysis; it is performed as part of the layout analysis step in the API and command line interface, and reading order models are trained on a fixed typology contained in the training data that is not easily interchangeable between different layout analysis models and their potential differences in class semantics[6]. To prevent accidental

[6] A controlled vocabulary to describe functional elements of historical page layouts exists with SegmOnto[**?**], but it is frequently modified to suit the needs of individual projects.

mismatch between these two types of models, reading order models are trained separately from layout analysis models but need to be added in an explicit step to the latter to create a combined model file that can then be utilized by users for layout analysis and reading order determination simultaneously.

Text Recognition Arguably the heart of any ATR software, the text recognizer transforms images of text into sequences of numerical labels that are subsequently mapped into Unicode code points. In line with the most common text recognition paradigm today, kraken utilizes a line-wise recognizer based on a neural network trained with CTC loss which predicts texts one line at a time, avoiding both the isolation of individual characters in the layout analysis step, an almost insurmountable task for many connected scripts such as Arabic, and avoiding the training data inefficiency of current end-to-end whole page recognition methods. The default architecture of the network is a hybrid CNN-LSTM (see table 1) made up of an stack of convolutional layers designed to extract local visual features followed by three LSTM layers to capture contextual information.

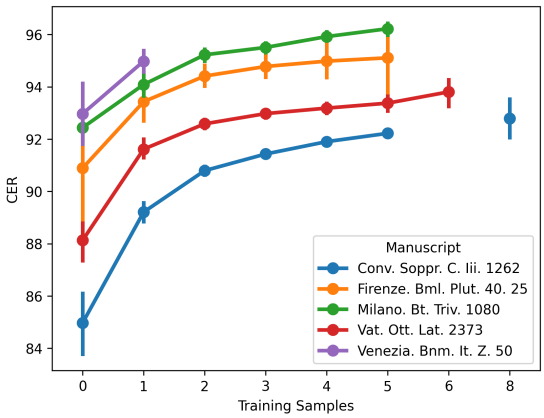


Fig. 4. Fine-tuning characteristics for a number of Italian manuscripts of the text recognizer from the generalized medieval CATMuS 1.5 base model[**?**]. Each model was trained on n randomly sampled page(s) (x-axis) and tested against the remainder of the manuscript. Error bars indicate standard deviation on the test set. Graph by Thibault Clérice produced with data from the Naples Dante Project.

An important part of the recognizer's accuracy is how text line images are prepared for recognition. Input line images are usually scaled to a fixed height before ingestion by the network, making it fairly sensitive towards large changes in relative scale of the text introduced by line rotation and curvature. Handwritten text naturally exhibits these deviations from perfect rectilinear writing, often in ways that make global rectification with deskewing and dewarping algorithms impossible. A benefit of the baseline and bounding polygon data model implemented in the layout analysis block is that individual text lines can be dewarped locally by applying a piecewise affine transform on a tesselation of baseline segments and bounding polygon in order to map the former into the plane (figure 3). Albeit requiring computational effort roughly equal to a forward pass of the recognizer itself, this mechanism allows recognition of arbitrarily shaped text lines at fixed scale which

substantially reduces training data requirements in comparison to unrectified, variable scale recognition.

The neural network architecture of the recognizer can be specified with the Variable Graph Specification Language (VGSL), a simple domain-specific language which allows configuring neural networks in a flexible and modular way. It allows users to define the structure of a neural network by specifying components such as layers, connections, and basic hyperparameters like image scaling and color modes in a graph-based format, making it easier to experiment with different architectures and configurations. While the provided default configuration is quite versatile and provides competitive results from work-specific models trained on modestly sized datasets of around a thousand lines to large generalized models trained capable of multi-language recognition such as [**?**], through VGSL model architecture size can be optimized for each particular application.

Unsupervised Pretraining While existing text recognition models can be adapted to new material with minimal effort (see figure 4) and even training a recognition model from scratch requires only modest amounts of training data which can be further reduced by scaling the model size with the built-in VGSL specification language, those are not always possible when a lack of base models or insufficient resources for annotation interfere. In kraken 5, a self-supervised pretraining method has been added, a reimplementation of [**?**], that uses a reconstruction loss on randomly masked out parts of unlabelled line images to learn visual representations in the convolutional and recurrent encoder layers. The pretrained encoder is then fine-tuned in conventional supervised fashion, albeit with much reduced training data requirements, with CTC loss by adding a randomly initialized linear projection layer.

Some quality metrics obtained on real-world historical data and the academic IAM dataset [**?**] in a variety of training configurations are shown in table 2.

3.3 Model Repository

Retrodigitization in the historical sciences often operates in settings with limited access to technical expertise, computational infrastructure, and labor to create substantial training datasets. As a result, the effective use of ATR in such environments depends heavily on the sharing and reuse of existing datasets and model weights to overcome the barriers of constrained resources by building on pre-existing work without the need for costly new data collection. Providing appropriate infrastructures to enable sharing of datasets and models not only enhances the effectiveness of research but also ensures that the individuals who contributed to creating these digital artifacts receive proper recognition for their work, fostering a culture of openness and collaboration, in addition to fulfilling open science requirements mandated by many funding bodies in a meaningful way.

While there exists a common database of openly accessible training datasets with HTR-United [**?**], the same is not the case for trained models. Some ATR software, including kraken, have conceived ad-hoc solutions to distribute training

Table 2. Character error rates on a selection of public models available on the model repository, the IAM dataset, and a comparison of single manuscript hand models trained with very small datasets from scratch and with weights obtained through unsupervised pretraining. Absolute improvements for the pretraining case are marked with ↓.

		CER
Public models[1]	Classical Chinese prints and manuscripts [**?**]	1.04
	Hebrew manuscript [**?**]	2.76
	Medieval Latin-script [**?**]	4.94
	Modern German manuscript [**?**]	5.42
	Modern French manuscript [**?**]	7.64
	Modern Spanish manuscript [**?**]	8.50
	Modern Latin manuscript [**?**]	8.58
IAM[2]	Scratch	5.12
	Fine-tuned[3]	3.91
No pretraining[4]	50 lines	60.85
	100 lines	26.70
	200 lines	6.20
Unsupervised pretraining	50 lines	58.94 (↓1.91)
	100 lines	18.30 (↓8.40)
	200 lines	5.40 (↓0.80)

[1] Metrics as reported by model authors.
[2] The model was trained on a conversion of the original IAM database XML files into ALTO. Train-val-test splits are page-wise and custom.
[3] Fine-tuned on the all language version of [**?**].
[4] Lines randomly sampled from a transcription of a French-language manuscript BnF Ms.5103 with 300 line fixed test set.

artifacts, often on general purpose model repositories. These setups are plagued by lack of consistent metadata, are specific to particular software packages, and frequently do not allow discovery and filtering of models, making them unsuitable for all but the most basic tasks.

With the current release, kraken's ad-hoc model repository implementation has been replaced with a module compliant with the HTRMoPo standard [**?**]. Inspired by the HTR-United schema, the new metadata schema includes both adequate information to aid in automatic selection of arbitrary training artifacts and a recommendation for a model card format to communicate information such as data selection policies or transcription guidelines that are difficult to express with machine-readable restricted vocabularies. Hosting on the Zenodo research data infrastructure ensures sustainable long-term archival, discoverability, and reproducibility by leveraging its robust metadata management and persistent digital object identifiers (DOI). The use of standard protocols such as OAI-PMH by the repository makes the model collection not just a silo limited to the ATR community but facilitates seamless integration with universal indexing services

and research aggregators. The new implementation also supports versioning of models through concept DOIs, which allow grouping multiple deposits under a single persistent identifier, an important feature as training is often an iterative process and models tend to be updated over time as the datasets they have been trained on are expanded.

The repository currently contains 48 different models covering 18 different writing systems, ranging from recognition models trained for a particular hand to large generalized models that achieve competitive results across multiple languages, styles, and historical epochs.

4 Conclusion and Outlook

I have presented the principal difficulties posed by historical material and scholarly practices in the historical sciences to automatic text recognition and how kraken aims to resolve them through a combination of software design choices and careful selection of methods matched to the scale of work in this domain. From the widespread use of the software spanning across dozens of languages, writing systems, and epochs, it is clear that our approach is sound and adapted to the constraints and requirements of research in the humanities.

While the system works generally well on a diverse set of material, some types of writing are insufficiently treated by it. This is largely due to deficiencies in the layout analysis module, which are a direct cause of misrecognition as the line-based recognizer cannot process line content that has not been correctly isolated in the text line detection step. A frequent error source is the heuristic computing the bounding polygon, which at times fails to correctly include diacritics or can even fail completely on some material with highly varying text size in lines or text embedded in decoration.

A new recognizer that does not require explicit line masking and extraction and can perform recognition directly from multi-modal line positional features, i.e., baselines or bounding boxes, and whole page image features instead is going to be added in the near future. The primary benefit of this approach is that it permits the recognizer to learn which parts of the page have to be taken into account to predict the text in a particular text line autonomously, which dispenses with the need for bounding polygons but also eliminates the strong coupling between layout analysis implementations and recognizer that has been observed in practice [**?**]. This method allows for the first time effective training of text recognition models from data produced with diverse software stacks, leading to a much more free mixing of methods and data sources in an ATR workflow.

Acknowledgments. Funded by the European Union (ERC, MiDRASH, Project No. 101071829). Views and opinions expressed are however those of the author(s) 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 authority can be held responsible for them. This work has received funding through the Investissements d'Avenir program of the Agence Nationale de la Recherche with reference ANR-21-ESRE-0005 (Biblissima+).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ares Oliveira, S., Seguin, B., Kaplan, F.: dhsegment: A generic deep-learning approach for document segmentation. In: 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 7–12 (2018). https://doi.org/10.1109/ICFHR-2018.2018.00011
2. Baierer, K.: hOCR - OCR Workflow and Output embedded in HTML (2020), https://kba.github.io/hocr-spec/1.2/
3. Bluche, T., Messina, R.: Gated convolutional recurrent neural networks for multilingual handwriting recognition. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 01, pp. 646–651 (2017). https://doi.org/10.1109/ICDAR.2017.111
4. Brisson, C., Constant, F., Bui, M.: Chinese historical documents automatic transcription (chat) models (Sep 2023). https://doi.org/10.5281/zenodo.8383732, https://doi.org/10.5281/zenodo.8383732
5. Castro, D., Bezerra, B.L.D., Zanchettin, C.: An end-to-end approach for handwriting recognition: From handwritten text lines to complete pages. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 264–273 (6 2024)
6. Chagué, A., Clérice, T.: "I'm here to fight for ground truth": HTR-United, a solution towards a common for HTR training data. In: Digital Humanities 2023: Collaboration as Opportunity. Alliance of Digital Humanities Organizations and University of Graz, Graz, Austria (Jul 2023), https://inria.hal.science/hal-04094233
7. Clérice, T., Pinche, A., Vlachou-Efstathiou, M., Chagué, A., Camps, J.B., Levenson, M.G., Brisville-Fertin, O., Boschetti, F., Fischer, F., Gervers, M., Boutreux, A., Manton, A., Gabay, S., O'Connor, P., Haverals, W., Kestemont, M., Vandyck, C., Kiessling, B.: CATMuS Medieval: A large scale cross-century dataset in latin scripts for handwritten text recognition and beyond. In: ICDAR 2024: International Conference on Document Analysis and Recognition, Athens, Greece, September 6-11, 2024 (2024), under review
8. Coquenet, D., Chatelain, C., Paquet, T.: Dan: A segmentation-free document attention network for handwritten document recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(7), 8227–8243 (2023). https://doi.org/10.1109/TPAMI.2023.3235826
9. Fujitake, M.: Dtrocr: Decoder-only transformer for optical character recognition. In: 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 8010–8020 (2024). https://doi.org/10.1109/WACV57701.2024.00784
10. Gabay, S.: Fondue-gd (Dec 2024). https://doi.org/10.5281/zenodo.14399779, https://doi.org/10.5281/zenodo.14399779
11. Gabay, S., Pinche, Christensen, K., Camps, J.B., Carboni, N.: SegmOnto, A Controlled Vocabulary to Describe the Layout of Pages (2023), https://segmonto.github.io/

12. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning. p. 369–376. ICML '06, Association for Computing Machinery, New York, NY, USA (2006). https://doi.org/10.1145/1143844.1143891, https://doi.org/10.1145/1143844.1143891
13. Gruning, T., Labahn, R., Diem, M., Kleber, F., Fiel, S.: READ-BAD: A New Dataset and Evaluation Scheme for Baseline Detection in Archival Documents . In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). pp. 351–356. IEEE Computer Society, Los Alamitos, CA, USA (Apr 2018). https://doi.org/10.1109/DAS.2018.38, https://doi.ieeecomputersociety.org/10.1109/DAS.2018.38
14. Grüning, T., Leifert, G., Strauß, T., Michael, J., Labahn, R.: A two-stage method for text line detection in historical documents. International Journal on Document Analysis and Recognition (IJDAR) **22**(3), 285–302 (9 2019). https://doi.org/10.1007/s10032-019-00332-1, https://doi.org/10.1007/s10032-019-00332-1
15. Kang, L., Riba, P., Rusiñol, M., Fornés, A., Villegas, M.: Pay attention to what you read: Non-recurrent handwritten text-line recognition. Pattern Recognition **129**, 108766 (2022). https://doi.org/https://doi.org/10.1016/j.patcog.2022.108766, https://www.sciencedirect.com/science/article/pii/S0031320322002473
16. Kiessling, B.: A Modular Region and Text Line Layout Analysis System. In: 17th International Conference on Frontiers in Handwriting Recognition, ICFHR 2020, Dortmund, Germany, September 8-10, 2020. pp. 313–318. IEEE (2020). https://doi.org/10.1109/ICFHR2020.2020.00064, https://hal.science/hal-04442992
17. Kiessling, B.: CurT: End-to-End Text Line Detection in Historical Documents with Transformers. In: Frontiers in Handwriting Recognition: 18th International Conference, ICFHR 2022, Hyderabad, India, December 4–7, 2022. Lecture Notes in Computer Science, vol. 13639, pp. 34–48. Springer International Publishing, Hyderabad, India (Dec 2022). https://doi.org/10.1007/978-3-031-21648-0_3, https://hal.science/hal-04036249
18. Kiessling, B.: The HTRMoPo schema for repositories of HTR/OCR models (2025), https://github.com/mittagessen/htrmopo
19. Kiessling, B., Tissot, R., Stokes, P.A., Ezra, D.S.B.: eScriptorium: An Open Source Platform for Historical Document Analysis. In: 2nd International Workshop on Open Services and Tools for Document Analysis, OST@ICDAR 2019, Sydney, Australia, September 22-25, 2019. p. 19. IEEE (2019). https://doi.org/10.1109/ICDARW.2019.10032
20. Klut, S., van Koert, R., Sluijter, R.: Laypa: A novel framework for applying segmentation networks to historical documents. In: Proceedings of the 7th International Workshop on Historical Document Imaging and Processing. p. 67–72. HIP '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3604951.3605520, https://doi.org/10.1145/3604951.3605520
21. Kodym, O., Hradiš, M.: Page layout analysis system for unconstrained historic documents. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) Document Analysis and Recognition – ICDAR 2021. pp. 492–506. Springer International Publishing, Cham (2021)

22. Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., Wei, F.: Trocr: Transformer-based optical character recognition with pre-trained models. Proceedings of the AAAI Conference on Artificial Intelligence **37**(11), 13094–13102 (6 2023). https://doi.org/10.1609/aaai.v37i11.26538, https://ojs.aaai.org/index.php/AAAI/article/view/26538
23. Marti, U.V., Bunke, H.: The iam-database: an english sentence database for offline handwriting recognition. International Journal on Document Analysis and Recognition **5**(1), 39–46 (2002)
24. Neudecker, C., Baierer, K., Federbusch, M., Boenig, M., Würzner, K.M., Hartmann, V., Herrmann, E.: Ocr-d: An end-to-end open source ocr framework for historical printed documents. In: Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage. p. 53–58. DATeCH2019, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3322905.3322917, https://doi.org/10.1145/3322905.3322917
25. Pinche, A.: Generic HTR Models for Medieval Manuscripts. The CREMMALab Project. Journal of Data Mining and Digital Humanities **Historical Documents and automatic text recognition** (Jun 2023), https://hal.science/hal-03837519
26. Pinche, A., Clérice, T., Chagué, A., Camps, J.B., Vlachou-Efstathiou, M., Gille Levenson, M., Brisville-Fertin, O., Boschetti, F., Fischer, F., Gervers, M., Boutreux, A., Manton, A., Gabay, S.: Catmus medieval (Jul 2024). https://doi.org/10.5281/zenodo.12743230, https://doi.org/10.5281/zenodo.12743230
27. Pletschacher, S., Antonacopoulos, A.: The page (page analysis and ground-truth elements) format framework. In: 2010 20th International Conference on Pattern Recognition. pp. 257–260 (2010). https://doi.org/10.1109/ICPR.2010.72
28. Project PERO: (2021), https://pero-ocr.fit.vutbr.cz/
29. Puigcerver, J.: Are multidimensional recurrent layers really necessary for handwritten text recognition? In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 01, pp. 67–72 (2017). https://doi.org/10.1109/ICDAR.2017.20
30. Quirós, L., Vidal, E.: Reading order detection on handwritten documents. Neural Comput. Appl. **34**(12), 9593–9611 (Jun 2022). https://doi.org/10.1007/s00521-022-06948-5, https://doi.org/10.1007/s00521-022-06948-5
31. Quirós, L.: Multi-task handwritten document layout analysis (2018), https://arxiv.org/abs/1806.08852
32. READ-COOP: Transkribus (2025), https://transkribus.org
33. Reul, C., Christ, D., Hartelt, A., Balbach, N., Wehner, M., Springmann, U., Wick, C., Grundig, C., Büttner, A., Puppe, F.: Ocr4all—an open-source tool providing a (semi-)automatic ocr workflow for historical printings. Applied Sciences **9**(22) (2019). https://doi.org/10.3390/app9224853, https://www.mdpi.com/2076-3417/9/22/4853
34. Reul, C., Springmann, U., Puppe, F.: Larex: A semi-automatic open-source tool for layout analysis and region extraction on early printed books. In: Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage. p. 137–142. DATeCH2017, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3078081.3078097, https://doi.org/10.1145/3078081.3078097

35. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241. Springer International Publishing, Cham (2015)
36. Singh, S.S., Karayev, S.: Full page handwriting recognition via image to sequence extraction. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) Document Analysis and Recognition – ICDAR 2021. pp. 55–69. Springer International Publishing, Cham (2021)
37. Stoekl Ben Ezra, D., Brown-DeVost, B., Jablonski, P., Lapin, H., Kiessling, B., Lolli, E.: Biblia - a general model for medieval hebrew manuscripts and an open annotated dataset. In: Proceedings of the 6th International Workshop on Historical Document Imaging and Processing. p. 61–66. HIP '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3476887.3476896, https://doi.org/10.1145/3476887.3476896
38. The ALTO editorial board: ALTO Technical Metadata for Layout and Text Objects (2023), https://www.loc.gov/standards/alto/
39. Vogler, N., Allen, J.P., Miller, M.T., Berg-Kirkpatrick, T.: Lacuna reconstruction: Self-supervised pre-training for low-resource historical document transcription. arXiv preprint arXiv:2112.08692 (2021)
40. Wick, C., Reul, C., Puppe, F.: Calamari - A High-Performance Tensorflow-based Deep Learning Package for Optical Character Recognition. Digital Humanities Quarterly **14**(1) (2020)
41. Wödlinger, M., Sablatnig, R.: Text baseline recognition using a recurrent convolutional neural network. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 4673–4679 (2021). https://doi.org/10.1109/ICPR48806.2021.9412624