12 Unlocking Swedish historical material through OCR and HTR

Dana Dannélls
University of Gothenburg

Robin Kurtz
National Library of Sweden

Erik Lenas
Swedish National Archives

Viktoria Löfgren
Swedish National Archives

This chapter presents some of the efforts made by three national institutions and the challenges each institution encountered while advancing Optical Character Recognition (OCR) and Handwritten Text Recognition (HTR) technologies for Swedish historical material. It introduces the resources, models, and tools that can be used to refine computational approaches for improving OCR and HTR processing, which, in turn, could enhance text- and data-driven research.

1 *Introduction*

The technology that allows libraries and cultural institutions to convert analog data into digital format, enabling searches by character, word, or phrase, is called Optical Character Recognition (OCR) for printed material and handwritten text recognition (HTR) for handwritten material.

To date, all digitized collections in Sweden have been processed using OCR/HTR models trained on materials with varying layouts and quality. As a result, the digitization process has introduced noisy data into the material. Enhancing these models and, consequently, improving the quality of the processed material relies heavily on the availability of diverse, high-quality Swedish language resources, such as corpora, annotated datasets, lexicons, computational tools, and language models. Language resources play a crucial role in the analysis and interpretation of linguistic information in printed or written form (Baron & Rayson 2009, Rigaud et al. 2019,

HUM INFRA Dana Dannélls, Robin Kurtz, Erik Lenas & Viktoria Löfgren. 2025. Unlocking Swedish historical material through OCR and HTR. In Gerlof Bouma, Dana Dannélls, Dimitrios Kokkinakis & Elena Volodina (eds.), *Huminfra handbook: Empowering digital and experimental humanities* (NEALT Proceedings Series 59), 405–431. University of Tartu Library. DOI: 10.58009/aere-perennius0181

Ehrmann et al. 2022) Although not as large as English, Swedish has a considerable amount of advanced language technology resources and tools.[1] Particularly over the last decade, advances in transformer techniques and fine-tuning methods —together with open-source implementations for pre-training general language models — have made cutting-edge Swedish large language models and large-scale datasets available.

Owing to these rapid developments and recognizing the advantages they have for improving the results of OCR and HTR processes, a workshop was held at Gothenburg University in the spring of 2024 to discuss the challenges that are associated with the latest techniques. The workshop brought together representatives from Språkbanken Text, KBLab, the datalab of the National Library of Sweden (Kungliga biblioteket – KB) and the AI lab at the National Archives of Sweden (Riksarkivet).

The workshop aimed to showcase the recently developed Swedish tools and resources and to discuss the challenges associated with them. Participants also explored strategies to address these challenges in the ongoing work on OCR and HTR for historical Swedish documents. In this paper, we summarize the tools and datasets that have been presented at the workshop. To set the stage, we first provide a short overview of the groups that participated in the workshop.

1.1 *Språkbanken Text*

Språkbanken Text is a leading research infrastructure for Swedish,[2] and a national center for the collection and refinement of language resources, primarily Swedish text corpora and lexicon resources. These resources are freely available to the general public for research, teaching, and other purposes, either through inclusive platforms such as Korp (Borin et al. 2025) and Strix (Forsberg et al. 2025).

Part of the infrastructure work is to develop computational tools for linguistic analysis to enrich the resources with linguistic data equipped from structured lexical resources. With rich linguistic resources, new possibilities for data-driven humanities research become feasible, from large-scale diachronic studies to fine-grained discourse analysis.

Most of the resources at Språkbanken Text are texts, some of them are historical resources from 1600 and onwards. They have often been added to the repository after being scanned and digitized elsewhere without any corrections of the erroneous character sequences or words in the text, which decreases the quality of the linguistic analysis, and in turn, restricts the

1 Swedish is the 5th largest Wikipedia by article count according to meta Wikipedia.
2 https://spraakbanken.gu.se/en

information retrieved from the text. Consequently, improving post-OCR text quality has centered on leveraging rich lexical resources (Dannélls et al. 2021a) and on the ongoing development of novel methods. The latest method presented by Löfgren & Dannélls (2024) has been added as a plugin to Mink,[3] a data platform that allows users to upload language data to a local computer and access them via Strix,[4] a document-centered search platform.

1.2 *KBLab — The National Library of Sweden*

The National Library of Sweden[5] collects, preserves, and gives access to almost everything that is published in Sweden, including books, newspapers, radio, TV, and more. Its collections currently hold over 18 million items, which are continuously digitized to make them more accessible.

KBLab[6] was founded in 2019 as the library's *datalab* to give researchers the possibility to do large-scale quantitative research, to curate data maintained by the National Library, and to train models on the data maintained by the library to be used by academia, governmental organizations, and the industry (Börjeson et al. 2024).

A new initiative at the National Library, KB, aims to leverage Artificial Intelligence (AI) and Machine Learning (ML) to enhance the accessibility of its collections, using, for example, speech-to-text models to make the enormous radio collections easily searchable via text, or text-vision models to enable searching for specific images based on descriptions.[7] With this in mind it is especially important to improve the OCR quality for newspapers for services such as tidningar.kb.se.

1.3 *The National Archives*

The Swedish National Archives[8] has collected and preserved the archives and data from government authorities since the early 17th century. Large-scale digitization is an ever ongoing project, and as of now, the National Archives has scanned about 270 million document images, which represents about 5 percent of the entire archives.

A large proportion of these documents are handwritten. The National Archives started working with HTR in 2019. Today, this work is continued

3 https://spraakbanken.gu.se/mink/
4 https://spraakbanken.gu.se/strix/
5 https://www.kb.se/
6 https://www.kb.se/in-english/research-collaboration/kblab.html
7 A demo for searching through a collection of postcards can be accessed here https://lab.kb.se/bildsok/
8 https://riksarkivet.se

by the newly established AI lab, which aims to scale up the extraction of searchable text from scanned documents through HTR and OCR. This makes the archive materials available for data-driven research in a way that was previously impossible. The AI lab also works with creating databases and search indexes for frequently requested documents using HTR and OCR, which streamlines the work of archivists, and makes case-handling much more effective.

The challenges of large-scale HTR/OCR projects are not only theoretical, but to a large extent also practical. The National Archives are currently investing in hardware, infrastructure and software development for scaling up the successful transformation of scanned documents to digital text or structured databases.

2 *Research problem and relevance to Digital / Experimental Humanities*

Historical digitized collections are large – on the order of a hundred million pages – and unique cultural heritage datasets containing information that rarely is conveyed in other media types. For humanities scholars, this material holds significant value, offering insight into societal contexts and historical descriptions. The original images underlying the collections are characterized by low document quality: complex layouts, a mixture of typefaces, headings in various sizes, images, enriched with unique typographical features, illustrations, and decorative elements. The challenges exhibited by different types of material are illustrated in Figure 1. To add to the complexity, because documents have been scanned using different systems with diverse ML techniques, leading to varying quality results over the years. In turn, the poor quality of the processed material makes it inaccessible to humanities scholars, who are unable to locate what they need because the vast range of sources and materials available for research remains out of reach for digital humanities work.

3 *OCR and HTR workflow*

The input to an OCR/HTR process is an image or a set of images. The output of the process is the automatically extracted text, specifically, the extracted characters, words and sentences, represented in the image.

Several commercial and open-source OCR and HTR software are offered on the market, with the most prominent ones being Abbyy FineReader,[9]

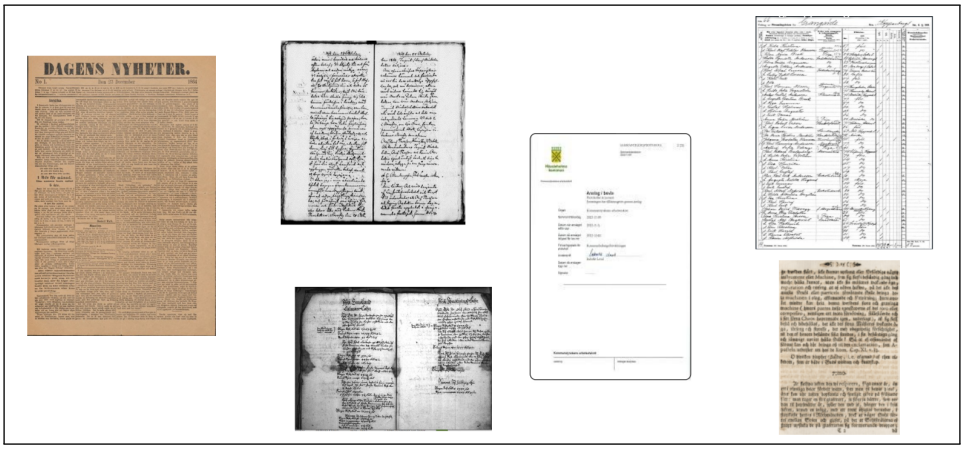


Figure 1: Historical digitized documents with varying complexities

Transkribus,[10] Tesseract,[11] Calamari,[12] and OCRopus.[13] Despite this, almost all libraries in Sweden are still using the commercial Abbyy FineReader and recently Transkribus as their core for OCR/HTR processing. Each software tool has its own built-in intelligence, which is applied slightly differently across various digitization workflows. Due to advances in digitization techniques, OCR results from pages processed a decade ago often differ from those processed more recently (Holley 2009). This is partially because older software versions were used in earlier digitization efforts.

A typical OCR/HTR process consists of four to five sequential steps, as shown in Figure 2: (1) Data/image processing, including binarization and normalization operations such as correcting skew lines and removing noise; (2) Segmentation following the image layout structure; (3) Feature extraction for simplifying the image data by detecting colours, objects and pixel points within the image object bounding box; (4) Character recognition using a language model; and (5) Post-correction of the output results from the language model. Over the years, researchers faced challenges in obtaining accurate results in each step, but thanks to advances in AI methods, some of the steps are no longer a challenge. However, since each step depends on the previous one, error propagation remains. In the following, we will describe each of the steps separately.

9 https://help.abbyy.com/en-us/
10 https://www.transkribus.org/
11 https://github.com/tesseract-ocr/tesseract
12 https://github.com/Calamari-OCR/calamari
13 https://github.com/ocropus

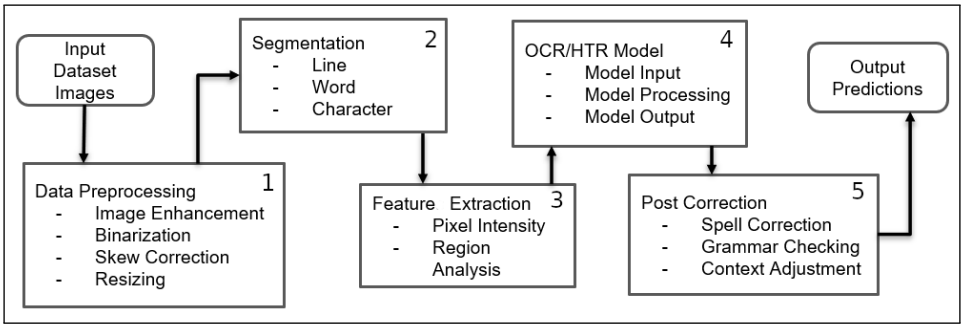


Figure 2: Stages of OCR/HTR processing

3.1 *Input images/dataset*

To understand the complexity of the task, consider the various formats of the scanned document provided as the input image to the OCR/HTR engine, as can been seen in Figure 1. Documents are ranging from single-column documents to multi-column layouts, different line lengths, and even complex table formats. Unfortunately, there is no single model capable of handling this diversity; instead, in the next preprocessing step each model must be trained to accommodate a specific input format.

3.2 *Data preprocessing*

The goal of this step is to improve the quality of the source document by addressing the damage caused by paper aging, usage, and wear and tear, as well as removing noise from faded characters, ink splodges, and bleed–through.

Depending on the quality of the source, an appropriate strategy, covering everything from simple binarization to advanced deep-learning techniques, must be chosen. The research field that is dedicated to solving these particular issues, is known as document analysis systems (Sfikas & Retsinas 2024). Notably, since 2020, the field has made significant breakthroughs thanks to advancements in transformer-based architectures (Wolf et al. 2020), which have made OCR and HTR systems less sensitive to the quality of the source.

3.3 *Layout analysis*

Layout analysis or segmentation, is the identification of boundaries between columns, lines, words, and characters. Things that affect the results of this step are small margins between lines and columns, or mixed fonts on the same page. The method for segmenting an image or a document

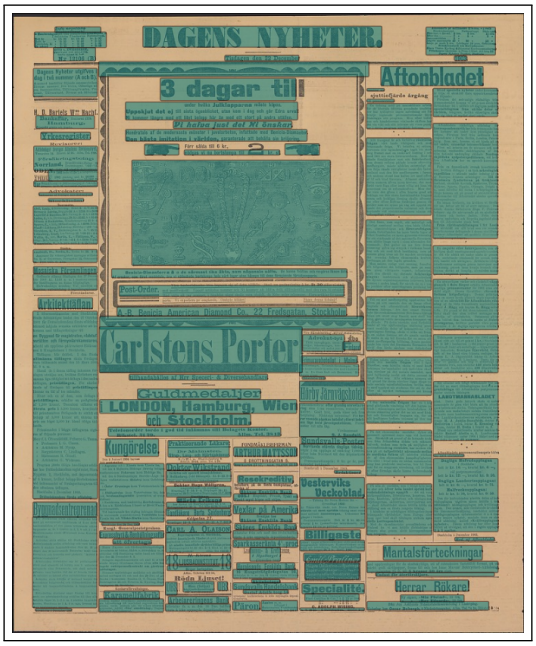


Figure 3: A newspaper page marked with identified bounding boxes

depends heavily on whether the document is structured, unstructured or semi-structured. For example, a regular book with one block of text per page containing a header and the page number is much easier to segment than a multi-column layout with images or headlines covering multiple columns as in Figure 3.

Similar to the previous step, this process can be entirely language-independent and is often examined within research focusing on document-level analysis (Sfikas & Retsinas 2024).

3.4 *Feature extraction*

The feature extraction stage is applied to capture the characteristics of characters and symbols in the image. It involves analyzing and extracting either numerical or structural attributes that describe the shape, structure, and distribution of pixels in characters. For example, a numerical feature extraction of the character <B> can result in a distribution of pixels as shown in Figure 4, and have a horizontal and a vertical feature vector that summarize the distribution that is: [4 2 2 4 2 2 4] [7 3 3 3 4].

1	**1**	**1**	**1**	0
1	0	0	0	**1**
1	0	0	0	**1**
1	**1**	**1**	**1**	0
1	0	0	0	**1**
1	0	0	0	**1**
1	**1**	**1**	**1**	0

Figure 4: The distribution of pixels of "B"

Off-the-shelf OCR and HTR systems apply different methods to train character models for recognizing mixed type faces (i.e. Fraktur, Schwabacher) and language specific characters, such as 'ä', 'ö', and 'å'. Usually, these systems do not have enough data to train character models for historical text, and consequently, they struggle to recognize uncommon characters such as rare ligatures because these characters were absent in the training material. Another problematic aspect is typically caused by limited predictability and generality, difficulty in distinguishing between similar characters and sensibility to translation Therefore, accurate character models are crucial for better system performance.

3.5 *OCR and HTR models*

OCR and HTR models are trained to recognize characters and words and classify them accordingly. The training is done with the help of a set of examples, where both the internal value (the image) and the target value (the text) are given. Based on the examples, the system makes various predictions from unseen images.

When training OCR and HTR models, the most important ingredient is the data that the model will be trained and tested/evaluated on. Training can be done on (semi-)automatically annotated data, which is data that was automatically annotated but additionally proofread by human annotators or annotated data generated by some model with high confidence. Evaluation generally involves human annotators who read through the source documents and label relevant information in different parts of the document. As this process is usually very costly, annotated datasets are rare and small. Different techniques and approaches to increasing the amount of data for training and evaluating OCR and HTR models have been proposed, for an overview and further reading see Agarwal & Anastasopoulos (2024).

3.6 *Post-correction*

Post-correction refers to the process applied to OCR or HTR output after the text has been scanned and processed. It is commonly used to correct residual or newly introduced processing errors. A straightforward approach to post-correction involves word replacement, in which edit-distance methods are applied to generate a list of candidate words, and the erroneous word is replaced with the closest match from a dictionary. This method can be further refined by incorporating word frequency data from large corpora to weight the probability of each candidate word, thereby increasing the likelihood of selecting the most accurate replacement.

Another approach to post-correction is context-based, meaning that the text sequence is not predicted greedily character by character, but is instead constructed by predicting the most probable token based on its preceding or surrounding context (Nguyen et al. 2021).

The post-correction step is typically integrated into a complete OCR/HTR system, but it can also be extended by incorporating an additional model to correct any remaining errors. In recent work on Swedish OCR for historical text, Löfgren & Dannélls (2024) implemented a character-based post-OCR method that significantly reduced both character and word error rates.

3.7 *Output predictions*

The output text from OCR/HTR is generally structured, often encoded in XML or ALTO files, and follows the hierarchical layout identified by the model during layout analysis (see Section 3.3). For multi-page documents, the output includes separate sections for each page, which in turn contain nested elements such as text blocks, columns, lines, or smaller units. Each element in this hierarchy typically includes metadata about its position in the document—such as page number—or its exact coordinates on the page (bounding box). It also specifies the type of content it represents, such as text, image, or table. While XML is a common format for this structured output, specialized formats like hOCR are also frequently used.

3.8 *Evaluation*

Evaluation is not part of the core OCR/HTR processing workflow, but it is a crucial step in assessing the performance of any system. To evaluate OCR and HTR systems, their output is compared to a reference—i.e., the correct text. Two standard metrics are commonly used to quantify errors: Character Error Rate (CER) and Word Error Rate (WER).

CER and WER are calculated by determining the number of edits—substitutions (S), deletions (D), and insertions (I)—needed to transform the output to the reference, by the number of characters/words in the reference (N):[14]

$$\text{CER/WER} = \frac{S + D + I}{N}.$$

While WER gives us a good estimate about the final performance of the OCR/HTR system, CER helps to get insight into specific character errors, such as regularly missing diacritics.

4 *Data and models for OCR/HTR processing*

OCR and HTR models for converting image to text are trained primarily on language data. To improve the accuracy of these models more data and in particular, reference data is required.[15] When the models have been trained and applied to new data, the results have to be evaluated against manually corrected versions of the data.

4.1 *Språkbanken Text*

As already mentioned, Språkbanken Text collects, annotates, and preserves language data, in particular written text. In this section we describe the datasets that are available for training and evaluating Swedish OCR models. All data has been released under CC BY license.[16]

Svensk Fraktur

The data is a collection of 199 predominantly Fraktur,[17] printed between approximately 1626 and 1816, that was collected within the project "A free cloud service for OCR"; a collaboration between Språkbanken Text and the Gothenburg University Library (Borin et al. 2016). The transcriptions were manually annotated with typographic information such as font changes and ligatures by a company that specializes in double-keying processes.[18]

14 Note that the error rate is not necessarily between one and zero.

15 Reference data or gold standard or ground-truth data – all terms are used interchangeably – is the correct version of the automatically processed data, manually curated, ideally by more than one human annotator.

16 https://spraakbanken.gu.se/resurser?s=OCR&language=All

17 Fraktur is a type of script used for writing European languages that originated in the 12th century.

18 Double-keying is a verification method to ensure accuracy, it involves entering the same data twice – usually by two different people and then comparing the two entries.

Table 1: An overview of the Swedish datasets with manual transcriptions

Dataset	OCR system	Time period	Characters
Svensk Fraktur	OCRopus	1626–1816	282 432
Then swänska Argus	OCRopus	1732–1734	259 468
Literature	Abbyy	1836–2001	7 266 866
Newspapers	Tesseract	1818–2018	6 956 748
	Abbyy		6 928 171
	Abbyy-Tesseract		6 922 088

Then Swänska Argus

The second source of blackletter data is *Then Swänska Argus*, which was prepared within the same project as *Swedish fraktur* (Borin et al. 2016). This dataset consists of 25 issues of *Then swänska Argus*, a periodical by Olof von Dalin published between 1732 and 1734 (Språkbanken Text 2020). It is significantly older than the newspaper texts, but the language is casual and simple for its time, and it is often cited as the first example of Late Modern Swedish (*yngre nysvenska*).

The transcription is based on a transcription made at Uppsala University, with some modification by Borin et al. (2016). The original *Then swänska Argus* has been republished at least once, and the transcription does not appear to be directly tied to a specific printing. As a result, the transcription does not contain any line breaks. In order to align it with the OCR output, line breaks were inserted in the transcription. The line breaks were inserted such that the sum of the CER of each individual line was minimized.

Literature

The literature dataset consists of 79 titles of Swedish literature printed between 1836 and 2001. In total, it contains about seven million characters, making it roughly the same size and from the same period as the newspapers (see Table 1). The OCR quality is generally much higher than the newspapers', most likely because of higher paper quality and simpler page layout. The data was provided as XML files by the Swedish Literature Bank, *Litteraturbanken*.[19]

19 https://litteraturbanken.se/

Table 2: The 10 most common character errors made by the single model (left) and after voting using an ensemble of 5 models (right).

Single model				Conf. voted models			
GT	Pred.	Count	Percentage	GT	Pred.	Count	Percentage
ä	a	124	4.06	ä	a	118	5.43
ö	o	95	3.11	å	ä	58	2.67
å	ä	67	2.19	ö	o	52	2.39
__	_	53	1.73	″	″	46	2.12
u	n	45	1.47	å	a	34	1.56
ä	å	39	1.28	n	u	31	1.43
″	″	38	1.27	ſ	f	29	1.33
f	ſ	37	1.21	ä	å	22	1.01
i	l	33	1.08	r	t	22	1.01
				f	ſ	22	1.01

GT: Ground Truth, the true character; Pred.: the predicted character; Count: the number of occurrences; Percentage: the percentage of the total number of errors. Note that GT "__" and Pred. "_" means a space has been deleted

Newspapers

The newspaper collection is partially a free digital reference resource featuring content from newspapers published between 1818 and 2018 (Dannélls et al. 2019). It is a subset of the largest newspaper collection at KB called Kubhist (see Section 4.2). A total of 200 digitized newspapers were selected from Kubhist, with one newspaper chosen per year. These newspapers were carefully curated to represent common variations in layout and typography. From each newspaper, the second and fourth pages were selected, resulting in a collection of 400 pages, each was segmented down to the paragraph level. Similar to Svensk Fraktur, this dataset was manually transcribed by an external agency which specializes in double-keying. Additionally, manual annotations were conducted at both the document and paragraph levels by two undergraduate students with backgrounds in language technology and linguistics. Due to copyright limitations, only a subset of the collection, spanning up to 1906 is freely available.

Reference data enables the analysis of OCR models' performance and helps to identify what types of errors OCR systems tend to make. Brandt Skelbye & Dannélls (2021) ran a qualitative analysis of the OCR errors in a selection of newspaper editions from 1818–1848, and listed the 10 most common character errors made by OCR models trained on Swedish historical data in Calamari (see Table 2).

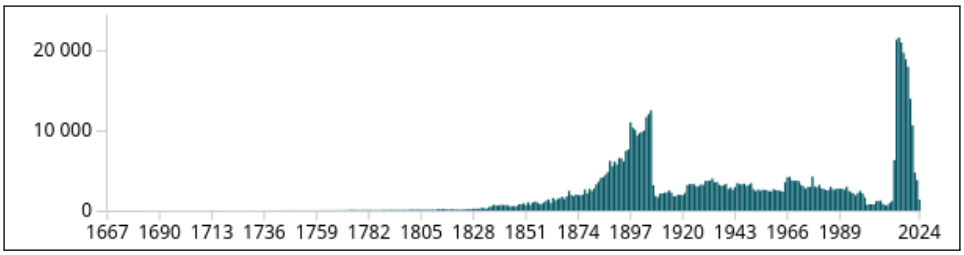


Figure 5: Misleading frequencies for the search term *örn* on tidningar.kb.se

Among the most common character errors, we find a mixture of loss of diacritics and ligatures such as for *'a' 'ä' 'å'*, or the mixing up of single letters. However, OCR errors are not limited to single characters. Another common error occurs in certain letter combinations such as *rn*, which are identified as *m* or vice versa. As such, searching for articles about *örn* 'eagles' in Swedish newspapers (see Figure 5) becomes next to impossible due to mix-ups with the more frequent *om* 'if', showcasing the need for better OCR systems for Swedish.

4.2 *National Library of Sweden*

The by far largest collection of digitized data at the National Library of Sweden is its newspaper collection, ranging from 1645 to today. The printed versions are delivered to the national library where they are digitized, and OCR processed with a commercial OCR system based on Abbyy FineReader. While KBLab has redone the OCR processing for some smaller collections, such as the proceedings of the Swedish parliament with a more recent version of Tesseract to increase the OCR quality, overhauling the complete newspaper collection is out of scope due to its sheer size (nearly 40 million pages currently available via tidningar.kb.se).

Another challenge with the digitization of newspapers lies in their layout. Extracting a specific article from one page requires finding all text-boxes belonging to the article, and determining their order. While newspaper layouts typically follow a left-to-right and up-down reading order, it is not always clear whether the text wraps around a picture or whether the picture serves as a barrier between articles, where the article actually starts and ends, and finally whether some sections belong to a different article, are an extension, or are something else such as advertisements (Rekathati 2021).

Any improvement on the OCR quality itself greatly increases this collection's value and avoids the need to apply fixes on the data as in Malmsten et al. (2020). While it is important for a cultural heritage institution to preserve

Table 3: Performance of the two-OCR engine and the two post-OCR correction models based on byT5

Model	CER (%)	WER (%)
Two-OCR	2.93	12.05
Löfgren & Dannélls (2024)	1.92	7.41
KBLab	1.57	6.23

the layout of newspapers, it is preferable for training language models and the analysis of the collection in general to be able to extract complete articles.

Post-OCR correction

The first step in improving the OCR quality of KB's newspaper collections was to follow Löfgren & Dannélls (2024) and train a new post-OCR correction model using a similar set-up. This model uses only newspaper data and is trained on a Nvidia DGX A100 with 8 GPUs, for much longer (75 iterations over the training data compared to 3), using a different learning rate scheduler, optimizer, and other related hyperparameters. The previous model by Löfgren & Dannélls (2024) reduced the WER by 39% compared to the output of the two-OCR engine by Dannélls et al. (2021b). The new model's improved capabilities manage to further reduce the WER by another 16%.

Table 3 shows the performance of the two-OCR model (Dannélls et al. 2021b), the post-OCR correction model based on byT5 (Xue et al. 2022) by Löfgren & Dannélls (2024), and the updated version trained at KBLab on the annotated newspaper dataset described in Section 4.1 using the same train-test split.

This relatively simple post processing step, can greatly enhance the usefulness of the data making for example search and word frequencies much more reliable. One caveat of this model is that it is trained on errors for older versions of Abbyy FineReader and Tesseract; this means that current versions of OCR engines potentially make new mistakes that are not learned by the post-OCR correction model. To further improve performance, we plan to update the training data (i) with automatic annotations by newer models, (ii) by employing the crowd-sourced annotations by Project Runeberg,[20] (iii) as well as input images generated from text.

20 https://runeberg.org/

4.3 *The National Archives*

The National Archives regularly publishes open datasets on its website[21] for use in different contexts. An especially valuable data set for training HTR models is *Göteborgs poliskammare*. It is the outcome of the crowdsourcing project described by Dick et al. (2024), in which 25 000 pages of 19th century handwritten police records were transcribed with the help of volunteers. The volunteers assisted with both creating an initial training set for an HTR model, and correcting the transcriptions made by the fine-tuned model. In total, this project yielded around 300 000 lines with high-quality transcriptions.

In addition to this data, the AI lab has collected datasets from within the National Archives and in collaboration with other institutions. The AI lab's latest HTR model, called *The Swedish Lion*[22] is trained on over one million lines of historical Swedish text from the years 1600–1900. Unfortunately, the training set cannot be published by the National Archives in its entirety, but roughly half the training set (566 000 lines) is available for download via the data and model sharing platform Huggingface.

Starting in 2025, HTR transcriptions produced by the Swedish Lion will be regularly published on the National Archives' website to be read alongside the original images. The transcriptions will be also available to download as plain text and Alto XML for further analysis.

5 *Research / use scenarios*

By releasing ground-truth data and OCR/HTR models, we seek to support better studies and user applications such as:

- Track linguistic and semantic change over time
- Improve search results in historical documents
- Identify statistical and semantic patterns
- Run analyses in large amounts of text and generate new research questions
- Translate historical texts into modern language for increased accessibility
- Integrate digital documents into chat applications

In the next section we present two use cases that describe how the user can utilize our tools to access text from a digital or analog image or document.

21 https://riksarkivet.se/psidata/

22 Available as an open-source model and as a model within the HTR platform Transkribus.

Figure 6: Search results for *gata* in Kalmar newspaper from 1872-06-15 in KB's online service

6 *Case studies*

After providing all the background information about OCR/HTR technology, the data and models available let us now look at two case studies that describe how the user can work with material that requires digitization, OCR or HTR processing.

6.1 *Case study 1: Identifying place names in newspaper collections*

In this use case, we assume the researcher aims to identify all place names in a set of newspaper collections available through KB's digital archive.[23] The collections have already been scanned, OCR processed, and made accessible online (see also Chapter X in this volume).

23 KB's digitized newspapers archive: https://tidningar.kb.se/

However, if the researcher's interest instead lies in a collection of printed novels that have not yet been digitized, the process begins differently. In such cases, the researcher must contact the digitization services at their university library,[24] requesting the material to be scanned and OCR processed. Most Swedish university libraries rely on the Abbyy FineReader OCR engine for this task. Although the libraries typically run the most recent model, update frequencies vary, meaning that OCR results can differ between institutions. Moreover, the quality of the source material—including print clarity, layout, and preservation state—has a major impact on OCR accuracy.

Once digitized, the next step is to identify place names. Unfortunately, KB's digital archive does not currently support named-entity search. As a workaround, researchers can narrow their search to terms that commonly appear in Swedish street names. For example, in Swedish, many street names end with *gata* 'street'. When we search for this term the search results in one single hit (see Figure 6). Upon closer inspection, however, we find that *gata* actually appears six times on the page, with three occurrences containing OCR errors: *Fiskaregatan, Storgatan, Norra Långgatan* (*OCR: Norra Länggatan*), *Storgatan, Nya Hamngatan* (*OCR: Nya Hanillgatail*), and *Westerlånggatan* (*OCR: Westcrlänggatani*). These discrepancies likely stem from either limitations in the search interface or errors introduced during OCR processing.

To perform a more refined analysis, researchers can use Strix, Språkbanken Text's document-centric platform for text exploration. Strix can be executed via Mink, Språkbanken Text's data processing platform, which allows users to upload their own collections, for example, digitized newspaper data, and run analytical workflows on the data.

There are three main ways to obtain data for analysis:

1. Request digitized files directly from KB, though access to copyrighted materials may be restricted.[25]

2. Download the image files and have them digitized by a university library.

3. Perform local OCR processing using Tesseract. For documents written in Fraktur, we recommend using Språkbanken Text's character model, specifically optimized for historical Swedish print (see Section 7).[26]

24 Many Swedish university libraries offer free digitization services, see for example Gothenburg University Library https://www.ub.gu.se/sv/tjanster-och-stod/digitala-kopior-ur-samlingarna

25 Access to copyrighted material may be restricted by KB.

26 In the future, Tesseract processing will be directly integrated into Mink

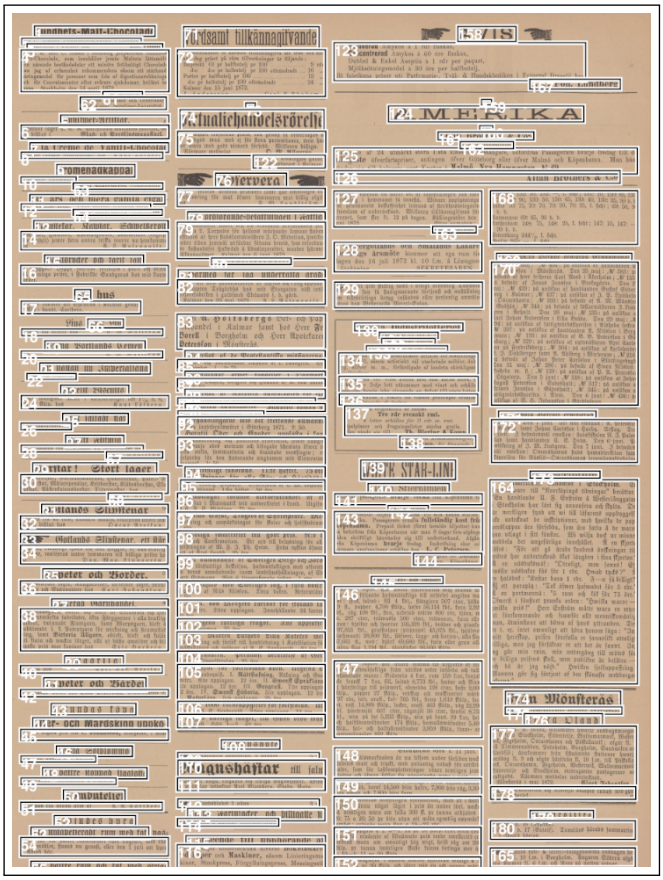


Figure 7: Segmentation results of Kalmar newspaper from 1872-06-15

The OCR process typically produces XML files that describe the document's page layout, including segmentation and annotation data. These files can be visualized to reconstruct how the OCR engine segmented the page, see Figure 7.[27] In this example, segments were numbered to indicate the order in which the OCR engine processed them (Dannélls et al. 2021b). It is important to note that this logical reading order is not always intuitive, and mis-segmentation often contributes to OCR errors.

Alongside XML output, OCR systems also generate plain text files containing recognized text (see Figure 8). Here, one of the text segments from a Kalmar newspaper page dated 1872-06-15 is shown side by side with its OCR

27 The University of Salford (Manchester) offers a set of tools for layout visualization and evaluation https://www.primaresearch.org/tools/PerformanceEvaluation

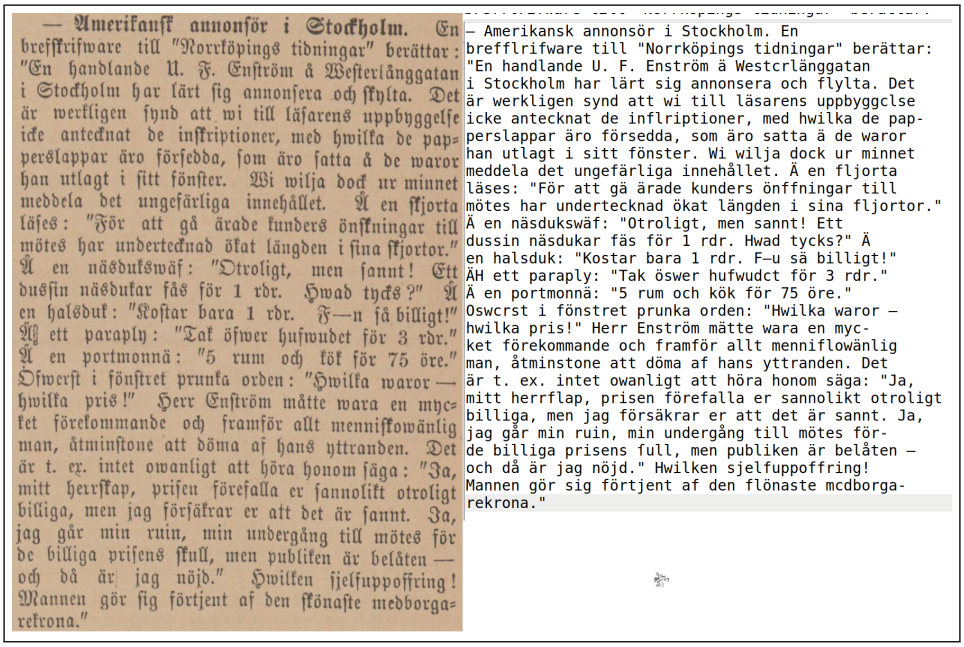


Figure 8: OCR transcription of segment 164 in Kalmar newspaper dated 1872-06-15. The image is shown on the left and the OCR output in plain text appears on the right

transcription. Before performing further searches on the text, it is advisable to identify and visualize OCR errors. To do so, we uploaded the OCR text files into Mink.

After uploading, the text is processed using Språkbanken Text's annotation pipeline, which includes post-OCR correction (Löfgren & Dannélls 2024). Once this step is done, Strix can launched locally to closely examine OCR errors and search through the processed material. Figure 9 shows how the results appear in Strix: on the left, the original OCR text; on the right, the corrected version after post-processing. Words highlighted in yellow mark differences between the two versions. For instance, the post-processing pipeline successfully corrected the street name *Westerlånggatan*, although some OCR errors remain unresolved.

This case study thus demonstrates a complete workflow—from identifying a research question (street names in newspapers), through OCR acquisition and quality control, to post-correction and close reading—using the integrated tools available via Språkbanken Text.

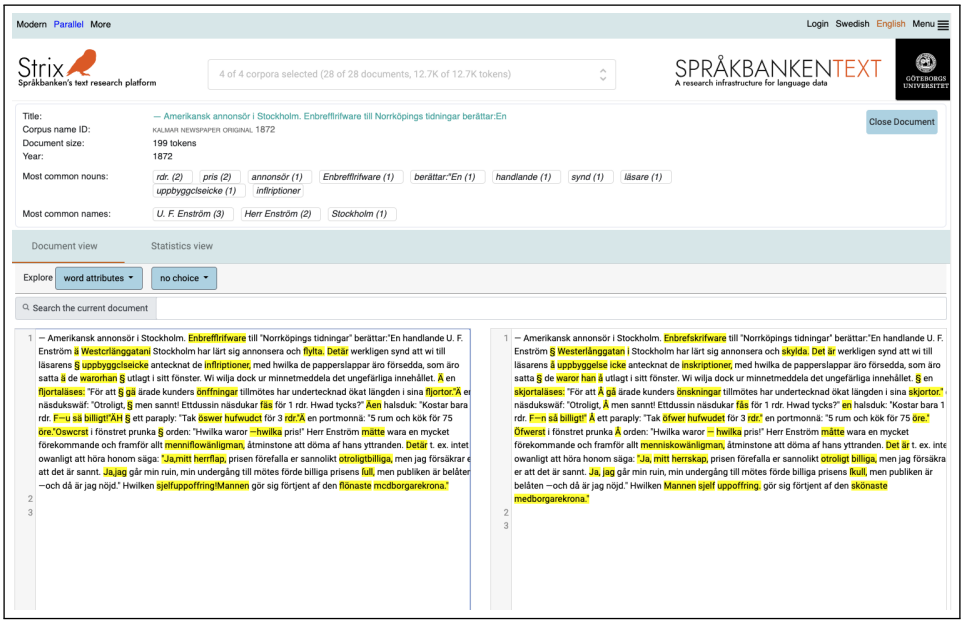
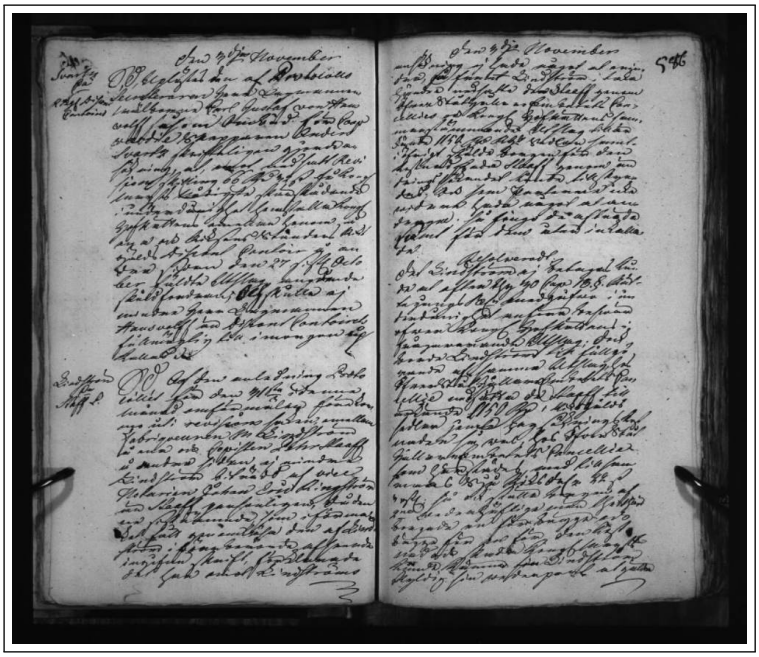


Figure 9: Post-OCR correction of segment 164 in Strix. OCR results before post-processing appear on the left and Post-OCR on the right

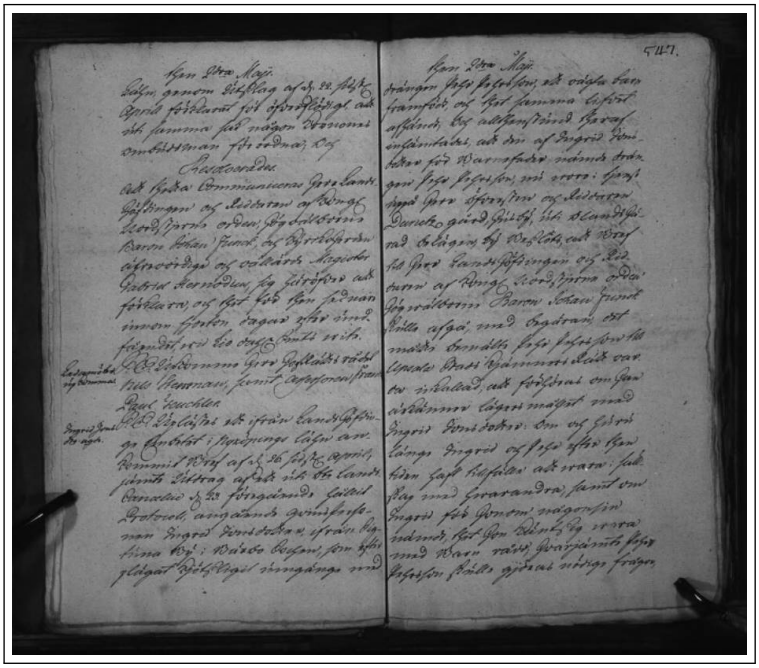
6.2 *Case study 2: Transcribing the archive of Svea Court of Appeal*

As the first step in the National Archives' effort in large-scale HTR, an ongoing project at the Archives aims to automatically transcribe the archive of Svea Court of Appeal (Swedish: *Svea Hovrätt*). The scanned archive contains over a million scanned images of handwritten court records from the 18th and early 19th centuries. Figure 10 shows two typical examples of images from the archive: free running text, in different hands, laid out in two-page spreads with occasional marginalia.

Maybe the first thing that comes to mind is developing the actual HTR model. For this you usually employ some type of neural network. The experience at the Swedish National Archives suggests that transformer architectures operating on a text line level give the best results (Li et al. 2023), even though interesting research exists on doing the text recognition on entire document images instead of text line (Kim et al. 2022, Wei et al. 2024). But these models, even though versions that are pre-trained on large quantities of handwritten text are freely available, do not work well out-of-the-box on historical Swedish handwriting. Thus, the first step, and often the most time-consuming one, is to create the training data for the models.



a.



b.

Figure 10: Examples of scanned documents from the archive of Svea Court of Appeal

Since the HTR model operates on the text line level, the document image must be segmented into lines. In this case, we segment each page into regions and then into lines using instance segmentation models.

When the models are trained and evaluated, they need to be run in an effective way and chained together in a pipeline, and this pipeline should export the results in an appropriate output format for downstream use. If you only have one set of models that should be used on a specific material, then writing this pipeline is pretty straight-forward. But if you want a level of generality that allows you to use any type of models, for instance trained with Hugging Face transformers, mmlabs or native pytorch, and use any kind of segmentation scheme you want, with customized evaluation schemes and several possible output formats, along with customizable preprocessing and postprocessing steps, then you need something like an entire HTR package, written specifically for implementing any type of HTR/OCR project effectively on any type of compute infrastructure. This project, at the Swedish National Archives, is called HTRflow, and is designed to achieve just this generality. The National Archives designed HTRflow with Svea Hovrätt as their use-case, but their aim was to maintain it, improve it, and use it as their inference-engine for all our HTR/OCR projects in the foreseeable future, hopefully aided by the open-source community.

The last stage in any HTR/OCR project is presenting the text to the end users, which in the case of Svea Hovrätt are researchers and the interested general public. For this a team within the IT-department wrote a customization of Universal Viewer 4, which allowed the transcribed text to be visible next to the document images, and by integrating the image viewer with The National Archives's search engine it is also possible to search through the text and get the results highlighted in the actual image.

7 *Practical information about resources, models and tools*

All the resources, models, and tools for Swedish presented here are freely available under the CC BY 4.0 license.

Resources

1. UB Fraktur a selection of 199 pages from 1626–1816 https://spraakbanken.gu.se/en/resources/svensk-fraktur-1626-1816.
2. A selection of digitized versions of Swedish newspapers from 1818 to 1870 https://spraakbanken.gu.se/resurser/svenska-tidningar-1818-1870

3. A selection of digitized versions of Swedish newspapers from 1871 to 1906 https://spraakbanken.gu.se/resurser/svenska-tidningar-1871-1906
4. Open source datasets https://riksarkivet.se/psidata
5. A dataset for training HTR models *Göteborgs poliskammare* https://riksarkivet.se/psidata/goteborgs-poliskammare
6. Ground truth HTR datasets https://huggingface.co/Riksarkivet

Models

1. OCR Post-correction model https://huggingface.co/KBLab/swedish-ocr-correction
   Training code of the model https://github.com/kb-labb/post-ocr-correction/tree/main
2. Swedish language OCR models trained with open source OCR software Calamari https://github.com/mskelb/OCR_SB
3. Trained Tesseract Fraktur model for Swedish http://demo.spraakdata.gu.se/ocr/models/clean_natural_140420-00024000.pyrnn.gz
4. HTR model based on the TrOCR architecture https://huggingface.co/Riksarkivet/trocr-base-handwritten-hist-swe-2

Tools

1. Mink https://spraakbanken.gu.se/mink
2. Strix https://spraakbanken.gu.se/strix
3. HTRflow https://huggingface.co/blog/Gabriel/htrflow

Acknowledgements

The research presented here has been enabled by the Swedish national research infrastructure Nationella språkbanken, funded jointly by the Swedish Research Council (2018–2024, contract 2017-00626) and the 10 participating partner institutions. We would like to acknowledge the Swedish national research infrastructure Huminfra, funded for the years 2022-2024 and 2005-2028, contracts 2021-00176 and 2023-00171 respectively, and the participating partner institutions.

References

Agarwal, Milind & Antonios Anastasopoulos. 2024. A concise survey of OCR for low-resource languages. In Manuel Mager, Abteen Ebrahimi, Shruti Rijhwani, Arturo Oncevay, Luis Chiruzzo, Robert Pugh & Katharina von der Wense (eds.), *Proceedings of the 4th workshop on natural language processing for indigenous languages of the Americas (AmericasNLP 2024)*, 88–102. Mexico City, Mexico: Association for Computational Linguistics. DOI: 10.18653/v1/2024.americasnlp-1.10.

Baron, Alistair & Paul Rayson. 2009. Automatic standardization of texts containing spelling variation: How much training data do you need? In Michaela Mahlberg, Victoria González-Díaz & Catherine Smith (eds.), *Proceedings of the corpus linguistics conference (CL2009)*. University of Liverpool, UK: Lancaster E-Prints UK. https://eprints.lancs.ac.uk/id/eprint/42529/.

Borin, Lars, Gerlof Bouma & Dana Dannélls. 2016. *A free cloud service for OCR / En fri molntjänst för OCR*. Tech. rep. University of Gothenburg, Gothenburg.

Borin, Lars, Markus Forsberg, Martin Hammarstedt, Louise Holmer & Arild Matsson. 2025. Korp: Språkbanken's word research platform. In Kristian Blensenius Dana Dannélls & Lars Borin (eds.), *Sixty years of Swedish computational lexicography*, 175–193. Berlin: De Gruyter.

Börjeson, Love, Chris Haffenden, Martin Malmsten, Fredrik Klingwall, Emma Rende, Robin Kurtz, Faton Rekathati, Hillevi Hägglöf & Justyna Sikora. 2024. Transfiguring the library as digital research infrastructure: Making KBLab at the national library of Sweden. *College & Research Libraries* 85(4).

Brandt Skelbye, Molly & Dana Dannélls. 2021. OCR processing of Swedish historical newspapers using deep hybrid CNN–LSTM networks. In Ruslan Mitkov & Galia Angelova (eds.), *Proceedings of the international conference on recent advances in natural language processing (RANLP)*. Held Online: INCOMA Ltd. https://aclanthology.org/2021.ranlp-1.23.

Dannélls, Dana, Lars Björk, Ove Dirdal & Torsten Johansson. 2021a. A two-OCR engine method for digitized Swedish newspapers. In *Selected papers from the CLARIN annual conference 2020, Linköping electronic conference proceedings 180*. Linköping: Linköping University Electronic Press.

Dannélls, Dana, Lars Björk, Ove Dirdal & Torsten Johansson. 2021b. A two-OCR engine method for digitized Swedish newspapers. In *Selected papers from the CLARIN annual conference 2020, linköping electronic conference proceedings 180*. Linköping: Linköping Electronic Conference Proceedings.

Dannélls, Dana, Lars Björk & Torsten Johansson. 2019. Evaluation and refinement of an enhanced OCR process for mass digitisation. *Digital Humanities in the Nordic and Baltic Countries Publications* 2(1). 112–123. DOI: 10.5617/dhnbpub.11085.

Dick, Kasperowski, Johansson Karl-Magnus & Karsvall Olof. 2024. Temporalities and values in an epistemic culture: Citizen humanities, local knowledge, and AI-supported transcription of archives. *Archives & Manuscripts* 51(2). 3–22. DOI: 10.37683/asa.v51.10937.

Ehrmann, Maud, Matteo Romanello, Antoine Doucet & Simon Clematide. 2022. Introducing the hipe 2022 shared task: Named entity recognition and linking in multilingual historical documents. In Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørvåg & Vinay Setty (eds.), *Advances in information retrieval*, 347–354. Cham: Springer International Publishing.

Forsberg, Markus, Dana Dannélls, Lars Borin & Aleksandrs Berdicevskis. 2025. Background: Språkbanken text. In *Sixty years of Swedish computational lexicography*, 161–173. Berlin: De Gruyter.

Holley, Rose. 2009. How Good Can It Get? Analysing and Improving OCR Accuracy in Large Scale Historic Newspaper Digitisation Programs. *D-Lib Magazine* 15(3/4). https://www.dlib.org/dlib/march09/holley/03holley.html.

Kim, Geewook, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han & Seunghyun Park. 2022. Ocr-free document understanding transformer. In *Computer vision – eccv 2022: 17th european conference, proceedings, part xxviii*, 498–517. Tel Aviv, Israel: Springer-Verlag. DOI: 10.1007/978-3-031-19815-1_29.

Li, Minghao, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li & Furu Wei. 2023. TrOCR: Transformer-based optical character recognition with pre-trained models. In *Proceedings of the thirty-seventh aaai conference on artificial intelligence and thirty-fifth conference on innovative applications of artificial intelligence and thirteenth symposium on educational advances in artificial intelligence* (AAAI'23/IAAI'23/EAAI'23), 13094–13102. AAAI Press. DOI: 10.1609/aaai.v37i11.26538.

Löfgren, Viktoria & Dana Dannélls. 2024. Post-OCR correction of digitized Swedish newspapers with ByT5. In Yuri Bizzoni, Stefania Degaetano-Ortlieb, Anna Kazantseva & Stan Szpakowicz (eds.), *Proceedings of the 8th joint sighum workshop on computational linguistics for cultural heritage, social sciences, humanities and literature* (*latech-clfl 2024*), 237–242. St. Julians, Malta: Association for Computational Linguistics. https://aclanthology.org/2024.latechclfl-1.23.

Malmsten, Martin, Love Börjeson & Chris Haffenden. 2020. *Playing with words at the National Library of Sweden – making a Swedish BERT*. https://arxiv.org/abs/2007.01658.

Nguyen, Thi, Adam Jatowt, Mickaël Coustaty & Antoine Doucet. 2021. Survey of post-OCR processing approaches. *ACM Computing Surveys* 54. 1–37. DOI: 10.1145/3453476.

Rekathati, Faton. 2021. *The KBLab blog: a multimodal approach to advertisement classification in digitized newspapers*. https://kb-labb.github.io/posts/2021-03-28-ad-classification/.

Rigaud, Christophe, Antoine Doucet, Mickaël Coustaty & Jean-Philippe Moreux. 2019. ICDAR 2019 competition on Post-OCR text correction. In *2019 international conference on document analysis and recognition (ICDAR)*, 1588–1593. DOI: 10.1109/ICDAR.2019.00255.

Sfikas, Giorgos & George Retsinas (eds.). 2024. *Document Analysis Systems - 16th IAPR International Workshop, DAS 2024 Proceedings*. Vol. 14994. (Lecture Notes in Computer Science). Springer. DOI: 10.1007/978-3-031-70442-0.

Språkbanken Text. 2020. *Dalin: Then Swänska Argus 1732-1734*. DOI: 10.23695/9z65-nv18.

Wei, Haoran, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, Chunrui Han & Xiangyu Zhang. 2024. *General OCR theory: towards OCR-2.0 via a unified end-to-end model*. https://openreview.net/forum?id=3LOcwfB4JX.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest & Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the conference on empirical methods in natural language processing: System demonstrations*, 38–45. Online: Association for Computational Linguistics. DOI: 10.18653/v1/2020.emnlp-demos.6.

Xue, Linting, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts & Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics* 10. 291–306. DOI: 10.1162/tacl_a_00461.

List of abbreviations

CER	Character Error Rate
GT	Ground Truth
HTR	Handwritten Text Recognition
KB	Kungliga biblioteket (National Library of Sweden)
NLP	Natural Language Processing
OCR	Optical Character Recognition
WER	Word Error Rate

Corresponding author

Dana Dannélls
Språkbanken Text
Department of Swedish,
Multilingualism, Language
Technology
University of Gothenburg
dana.dannells@svenska.gu.se