Comparative analysis of optical character recognition methods for Sámi texts from the National Library of Norway

Tita Enstad[1], **Trond Trosterud**[2], **Marie Iversdatter Røsok**[1], **Yngvil Beyer**[1], **Marie Roald**[1]

[1]National Library of Norway
[2]The Arctic University of Norway

tita.enstad@nb.no

Abstract

Optical Character Recognition (OCR) is crucial to the National Library of Norway's (NLN) digitisation process as it converts scanned documents into machine-readable text. However, for the Sámi documents in NLN's collection, the OCR accuracy is insufficient. Given that OCR quality affects downstream processes, evaluating and improving OCR for text written in Sámi languages is necessary to make these resources accessible. To address this need, this work fine-tunes and evaluates three established OCR approaches, Transkribus, Tesseract and TrOCR, for transcribing Sámi texts from NLN's collection. Our results show that Transkribus and TrOCR outperform Tesseract on this task, while Tesseract achieves superior performance on an out-of-domain dataset. Furthermore, we show that fine-tuning pre-trained models and supplementing manual annotations with machine annotations and synthetic text images can yield accurate OCR for Sámi languages, even with a moderate amount of manually annotated data.

1 Introduction

Optical Character Recognition (OCR) converts scanned documents into machine-readable text, which is crucial for making digitised materials available for search and analysis. For the National Library of Norway (NLN), the OCR output, among others, facilitates search for the online library (*Nettbiblioteket*[1]) and underpins analysis tools like the DH-Lab toolbox (Birkenes et al., 2023). However, while OCR quality is high for most Norwegian documents, it falls short for Sámi documents. The resulting text is insufficient for both search and for use in research or as a basis for language technology.

NLN has material in five Sámi languages: North Sámi, South Sámi, Lule Sámi, Inari Sámi and Skolt Sámi. Thus, developing an accurate OCR model for Sámi texts is important for NLN's mission to store and disseminate the materials in the library collection. Furthermore, for languages with limited resources, like Sámi languages, it is vital that the available resources are accessible to be searched and used for research. This paper describes a twofold contribution towards this goal:

1. Developing an OCR model for Sámi languages that improves the transcription accuracy of Sámi text in NLN's collection.
2. Comparing different OCR approaches in terms of transcribing smaller languages such as languages in the Sámi family.

2 Background

2.1 Sámi languages in the National Library of Norway's collection

Of the around 650 000 books and 4.6 million newspaper issues in NLN's digitised collection, about 3000 and 4500 are classified as Sámi, respectively. The classification generally means that the texts are written in Sámi, though some may just address Sámi-related topics.

With more than 20 000 speakers North Sámi is the most widely spoken Sámi language in Norway, Sweden and Finland, and it makes up the largest part of the Sámi collection at NLN. The other Sámi languages in NLN's collection all have less than 500 speakers. South and Lule Sámi are spoken in Norway and Sweden, and the collection contains a good amount of South and Lule Sámi books. Skolt Sámi, previously spoken in Norway and Russia, is now mainly spoken in Finland,

[1] https://www.nb.no/search

along with Inari Sámi, which has only ever been spoken in Finland. There is much less material in these languages in the collection ($<$ 20 books in total).

All five languages have standardised orthographies that were made or revised in the 1970s, 80s or 90s (Laakso and Skribnik, 2022; Olthuis et al., 2013; Magga, 1994), but the collection also includes earlier works that predate the standardised norms. To some extent these books contain non-standard letters or glyph-shapes and most words are spelled in ways differing from contemporary orthographies.

The Sámi written languages have letters not found in the Norwegian alphabet, but it varies from language to language which letters and how many. The alphabets have some letters in common, but none are identical. See Table 1 for an overview of these characters.

North	South	Lule	Inari	Skolt
Áá		Áá	Áá	
			Ââ	Ââ
			Ää	Ää
	Ïï			
				Õõ
	Öö			
Čč			Čč	Čč
Đđ			Đđ	Đđ
Ŋŋ		Ŋŋ	Ŋŋ	Ŋŋ
Šš			Šš	Šš
Ŧŧ				
Žž			Žž	Žž
				Ʒʒ
				Ǥǥ
				Ǧǧ
				Ǩǩ
				Ǯǯ
				ʹ
				,
				ˌ

Table 1: Overview of non-Norwegian characters used in the contemporary orthographies of the Sámi languages in the collection

2.2 Related work

While early OCR approaches often relied on hand-crafted image features combined with shape- and text-analysis (Smith, 2007), modern solutions use deep learning based models to learn informative features from the data itself. In particular, developments like convolutional neural networks (CNNs), bidirectional long-short-term-memory (LSTMs) (Hochreiter and Schmidhuber, 1997) and the Connectionist Temporal Classification (CTC) loss (Graves et al., 2006) has yielded state-of-the-art results (Shi et al., 2016; Puigcerver, 2017; van Koert et al., 2024; Tarride et al., 2024). Recently, transformer-based machine learning advancements have led to transformer-based OCR models such as TrOCR (Li et al., 2023).

OCR pipelines have also been developed for collections of digitised documents: Tesseract (Smith, 2007) is an open-source OCR framework for line segmentation and text recognition which includes pre-trained OCR models for several languages[2] and training scripts for training and fine-tuning on custom data. Since 2018, Tesseract has also supported LSTMs.

Another example is Transkribus, a proprietary platform for the recognition of printed and handwritten documents with a built-in interface for (semi-)manual transcription. The platform supports layout analysis and text recognition, using pre-existing or custom-trained models. The text recognition models are based on PyLaia (Puigcerver, 2017; Tarride et al., 2024), which uses a combination of CNNs and bidirectional LSTMs. Transcriptions can be exported, though models are restricted to use within the platform.

A recent advancement is transformers-based OCR. TrOCR is a state-of-the-art text recognition model that combines powerful transformer models for vision and language (Li et al., 2023). Specifically, TrOCR combines the “encoder” of a vision transformer (ViT) (Dosovitskiy et al., 2021), with the language generating “decoder” of a robustly optimised Bidirectional encoder representations from transformers approach (RoBERTa) model (Liu et al., 2020). TrOCR is specialised for text recognition, and will not perform ancillary tasks, like layout analysis. Moreover, while TrOCR is shown capable of outperforming Transkribus and Tesseract (Ströbel et al., 2023; Li et al., 2023), it is still a relatively recent algorithm, and there is still a need to assess its accuracy for low-resource languages.

OCR quality greatly impacts downstream processes (Lopresti, 2008; Järvelin et al., 2016; Evershed and Fitch, 2014). Consequently, parts of

[2] but none for the Sámi languages

a digitised collection with challenges like unusual fonts, bad scan quality or text in a low-resource language, will be less accessible. Several works have, thus, focused on improving OCR quality for texts with such challenges by e.g. using an ensemble of image preprocessing transforms (Koistinen et al., 2017), comparing various OCR- or handwritten text recognition (HTR)-models for smaller languages (Maarand et al., 2022; Memon et al., 2020; Tafti et al., 2016; Koistinen et al., 2017; Heliński et al., 2012) or post-correcting outputs (Poncelas et al., 2020; Duong et al., 2021).

OCR for low-resource languages is particularly challenging. Not only is there much less labelled data for training, but this problem is exacerbated further by potential changes in orthographies. Rijhwani et al. (2023) showed that including OCR in a semi-automatic annotation suite can aid annotation – even for a low-resource language such as Kwak'wala, where automatic annotation is difficult. Similarly, Yaseen and Hassani (2024) trained a Tesseract-based OCR system for Kurdish, another low-resource language. Agarwal and Anastasopoulos (2024) presented a concise survey of OCR for low-resource languages with a focus on Indigenous Languages of the Americas. Finally, Partanen and Rießler (2019) presented an OCR model for the Unified Northern Alphabet, used in the Soviet Union between 1931 and 1937 for Northern Minority languages (which includes Kildin Sámi).

3 Methods

3.1 Data

The main source for the data used in this work is NLN's digitised collection. Our goal was to create an OCR model for all languages in the collection, rather than one for each language, as this would allow for the most efficient integration into NLN's digitisation pipeline. However, we realised early that including Skolt Sámi would be difficult because of the three apostrophe characters that indicate pronunciation. This makes transcription difficult without a certain level of language proficiency. Thus, we proceeded with North, South, Lule and Inari Sámi.

In addition to data from NLN, we also used text-data data from the GiellaLT corpora[3] as basis for synthetic text images and data from the Divvun & Giellatekno fork of tesstrain[4] as basis for an out-of-domain (OOD) test set.

		South	North	Lule	Inari
Docs	GT	5	3	2	3
	Pred	265	1810	235	0
	Val	2	8	2	3
	Test	4	7	4	5
Lines	GT	208	5572	81	280
	Pred	7082	70413	6781	0
	Synth	76971	76949	76970	76497
	Val	53	1837	36	109
	Test	195	353	137	163
	OOD	0	122	0	0

Table 2: Distribution of documents and lines in each of the Sámi languages in the different datasets. GT, Val and Test refer to the data splits of the manually annotated data. Pred is the automatically annotated dataset, Synth is the synthetic dataset (natural language text but generated images) and OOD is the OOD Giellatekno test set.

Training data

We trained OCR models using manually transcribed data, machine transcribed data, and synthetic data[5]. See Table 2 for an overview.

Manually transcribed data We used Transkribus[6] (Kahle et al., 2017) to create the training data from the images of scanned pages. We used the platform's layout analysis, manually adjusting the results where necessary, then applied text recognition to the documents. Initially, we used a standard model provided by Transkribus. As we progressively corrected the recognised text, we trained new models, which were applied to recognise text in new documents, which we manually corrected to create the manually transcribed data.

Following this procedure, we transcribed 58 Sámi book and newspaper pages to create a manually transcribed training set, henceforth referred to as *Ground Truth Sámi* (GT-Sámi).

[3] https://giellalt.github.io/

[4] https://github.com/divvungiellatekno/tesstrain/tree/main/training-data/nor_sme-ground-truth

[5] As these texts contain copyrighted materials, the transcribed data sets can not be shared openly.

[6] We used the Transkribus Expert Client v1.28.0 and https://app.transkribus.org v4.0.0.150

Additionally, we already had 82 pages with 2998 manually transcribed Norwegian text lines (produced similarly as for GT-Sámi) that we included as training data. We refer to this data as *Ground Truth Norwegian* (GT-Nor).

Synthetic data To add more annotated Sámi text, we created synthetic data, which we refer to as the *Synthetic Sámi* dataset (Synth-Sámi). We used the SIKOR Sámi text corpus (SIKOR, 2021) as a basis of well-formed Sámi text, and generated images for the text lines (adding an uppercase version for $\simeq 10\,\%$ of the lines), using `CorpusTools`[7] to parse the XML files in the `converted`-directory of the `corpus-sma`[8], `corpus-sme`[9], `corpus-smj`[10] and `corpus-smn`[11] repositories. The images were created with Pillow[12] and Augraphy (Groleau et al., 2023), with variation in fonts and colours, and a varying degree of imperfections and noise added, resulting in 307 387 lines[13].

Automatically transcribed data As mentioned earlier, we trained Transkribus models incrementally while annotating data. Eventually, our Transkribus model[14] performed well on North, South and Lule Sámi, and we decided to automatically transcribe a larger amount of Sámi text with this model. We extracted page 30 from North, South and Lule Sámi books in NLN's collection and transcribed them automatically, which resulted in 2380 pages forming the *Predicted Sámi* (Pred-Sámi) dataset. This boosted the amount of data, but naturally, the transcriptions may not be correct.

Validation data

To evaluate during training and to select the best performing models for each architecture, we created a validation dataset. This dataset consists of 25 pages manually transcribed following the procedure described for GT-Sámi. Lines were selected from different books than the GT-Sámi training data while keeping a similar language distribution.

Test data

To compare the OCR approaches we used two test sets: one from NLN's collection and one from Divvun & Giellatekno's tesstrain data.

NLN test data As a goal of this work was to improve the transcriptions of Sámi documents in NLN's collection, we created a test set based on current transcriptions (baseline) of 21 pages from 18 books and 2 newspapers provided by NLN[15]. NLN stores these transcriptions as Analyzed Layout and Text Object-Extensible Markup Language (ALTO-XML) files with line segmentations and transcriptions. By matching the ALTO-XML transcriptions with manually annotated data, we created a test-set containing 848 text-lines.

Giellatekno test data The Giellatekno test data *nor-sme* was made for evaluating OCR reading of dictionares. It consists of 122 lines of dictionary data, thus text both in Norwegian and (contemporary) North Sámi. The dataset is available on Giellatekno's GitHub[16] We refer to this dataset as the OOD Giellatekno test set.

3.2 Evaluation metrics

Following previous work (Neudecker et al., 2021; Agarwal and Anastasopoulos, 2024), we used the character error rate (CER) and word error rate (WER) evaluation metrics. Specifically, we calculated collection level CER and WER (concatenating lines, with a space to separate them for WER) with Jiwer[17].

We also calculated an F_1 score for characters specific to the different Sámi languages, and an overall F_1 score for all non-Norwegian Sámi characters. The F_1 score is given by $\mathrm{F}_1 = 2\mathrm{TP}/(2\mathrm{TP} + \mathrm{FN} + \mathrm{FP})$, where TP, FP and FN is the number

[7] `https://github.com/divvun/CorpusTools`

[8] `https://github.com/giellalt/corpus-sma/`

[9] `https://github.com/giellalt/corpus-sme/`

[10] `https://github.com/giellalt/corpus-smj/`

[11] `https://github.com/giellalt/corpus-smn/`

[12] `https://python-pillow.org/` (Version 10.4.0)

[13] Code to generate synthetic data is on GitHub: `https://github.com/Sprakbanken/synthetic_text_images`

[14] Transkribus modelID 115833, publicly available in Transkribus

[15] We chose distinct books for the train, validation and test sets. However, due to few Inari Sámi books, 1 book is in both the train and test sets and 2 are in both the validation and test sets, but there is no page-overlap.

[16] `https://github.com/divvungiellatekno/tesstrain/tree/main/training-data/nor_sme-ground-truth`. We have corrected four transcriptions and used our corrected version of the test set which can be found on `https://github.com/MarieRoald/tesstrain/tree/fix-transcriptions/training-data/nor_sme-ground-truth`

[17] `https://github.com/jitsi/jiwer` (Version 3.0.4)

of true positives, false positives and false negatives, respectively. To measure TP, FP and FN in an OCR-setting, we only considered character counts, not location. Thus, for a given character, c, we set $\text{TP}_c = \min(n_c^{(\text{true})}, n_c^{(\text{pred})})$, $\text{FN} = \max(n_c^{(\text{true})} - n_c^{(\text{pred})}, 0)$ and $\text{FP} = \max(n_c^{(\text{pred})} - n_c^{(\text{true})}, 0)$, where $n_c^{(\text{true})}$ and $n_c^{(\text{pred})}$ are the number of c characters in the ground truth and predicted transcriptions, respectively. To compute an overall F_1, we combined the TP, FN, and FP across all lines and characters-of-interest.

To examine the types of errors our models made, we calculated the most common errors. Specifically, we used Stringalign (Moe and Roald, 2024), which implements optimal string alignment. Note that, in theory, multiple alignments can exist (e.g. if two letters are swapped), in which case Stringalign picks one.

3.3 Models and training

A goal of this work was evaluating different state-of-the-art OCR frameworks for Sámi text recognition. Specifically, we compared Transkribus, Tesseract and TrOCR. For each approach, we trained on several dataset combinations and chose the model based on mean(CER, WER) on the validation data for test-set evaluation.

Transkribus

We used Transkribus Expert for training Transkribus models[18]. We used standard parameters, but opted “Using exsisting line polygons for training”, and changed the batch size from 24 to 12[19]. We set 100 as maximum numbers of epochs, and 20 as early stopping. We used Transkribus print M1[20] as base model for 4 of the 5 models. All Transkribus models were run with the setting “Use language model”[21].

Tesseract

We used the official tesstrain repository[22] and Tesseract 5.4.1 for training. We experimented with both training models from scratch and fine-tuning existing models. During early experiments, we tried fine-tuning Norwegian, Finnish, and Estonian models using our Sámi dataset, and observed that the model with the Norwegian base adapted faster and performed better on our validation set. Thus, we continued training with the Norwegian base[23].

As tesstrain does not support dynamic learning rate and only exposes a few training hyperparameters to the user, we trained our models in 1-20 epoch increments, updating the learning rate until the model checkpoints no longer showed improvements on the validation set.

TrOCR

We used Huggingface Transformers (Wolf et al., 2020) to fit the TrOCR models, initialising with the parameters from the `microsoft/trocr-base-printed` repository. This model is pre-trained on both synthetic and printed text (Li et al., 2023). For fine-tuning, we had an initial learning rate of 10^{-6}, decreasing it by a constant amount for each iteration until it reached 10^{-7} at the final iteration. For models fine-tuned without Pred-Sámi, we trained for 200 epochs, evaluating and storing model parameters every fifth epoch. However, due to the data size and hardware limitations, models that included Pred-Sámi were only fine-tuned for 100 epochs, evaluating and storing model parameters every second epoch and selecting the checkpoint with the lowest validation CER.

Pre-training with synthetic data

We trained additional TrOCR and Tesseract models using synthetic data to assess the effect of adding such data[24]. After training all models without synthetic data, we retrained with the smallest amount of hand-annotated data (GT-Sámi) and best performing data combination, this time initialising with a model pre-trained on Synth-Sámi.

In particular, due to time and hardware limitations, we trained models on synthetic data in two stages inspired by the two-stage procedure in e.g (Li et al., 2023). For the first stage, we trained for five epochs on Synth-Sámi. For the second stage, we initialised with the best checkpoint from the

[18] `https://help.transkribus.org/model-setup-and-training`

[19] We changed this parameter after advice from the Transkribus team due to problems with the training stopping with `exitCode = 1`

[20] Transkribus ModelID 39995

[21] Which uses PyLaia’s n-gram model functionality to inform character predictions (Tarride et al., 2024).

[22] `https://github.com/tesseract-ocr/tesstrain` (Version 1.0.0, commit hash 45cacc5)

[23] `https://github.com/tesseract-ocr/tessdata_best/blob/main/nor.traineddata`

[24] We did not train Transkribus models with synthetic data as it does not support an easy way to train based on line images and because of its page-based pricing model.

w/o base	GT-Sámi	GT-Nor	Pred-Sámi	Synth base	Transkribus			Tesseract			TrOCR		
					CER	WER	mean	CER	WER	mean	CER	WER	mean
✓	✓				1.59	5.67	3.63	5.53	24.70	15.11			
	✓				1.28	4.34	2.81	2.05	9.84	5.95	1.98	9.29	5.64
	✓	✓			1.31	4.35	2.83	2.37	11.39	6.88	1.95	8.88	5.42
	✓		✓		1.48	4.02	2.75	1.85	8.17	5.01	1.28	5.00	3.14
	✓	✓	✓		**1.07**	**3.58**	**2.33**	1.81	7.96	4.89	1.32	5.14	3.23
	✓			✓				**1.78**	8.78	5.28	1.15	5.04	3.09
	✓		✓	✓							**1.08**	**4.29**	**2.69**
	✓	✓	✓	✓				1.79	**7.70**	**4.75**			

Table 3: CER, WER, and mean(CER, WER) on the validation set. The checkmarks indicate whether models were trained from scratch (i.e. not fine-tuning an existing base model) (first column) and what datasets were part of the training data

first stage (lowest CER) and continued training on real data.

4 Results

Code for training Tesseract and TrOCR models, creating synthetic data and more detailed dataset information is available through the supplement on GitHub[25].

4.1 NLN validation data

Transkribus models

As shown in Table 3, CER and WER decreased when we used the Transkribus Print M1 as the base model in addition to GT-Sámi. Hence, we continued to use the base model in the subsequent training. Supplementing GT-Sámi with GT-Nor did not improve performance, while supplementing with Pred-Sámi increased CER but decreased WER. However, adding both GT-Nor and Pred-Sámi led to the best-performing model on the validation set.

Tesseract models

From Table 3, we see that the model trained on GT-Sámi with a Norwegian base model greatly outperformed the corresponding model without a base model. We therefore continued training all Tesseract models from the Norwegian base model. Adding GT-Nor to the training data worsened the validation performance. However, adding Pred-Sámi to the training data improved validation performance, and adding both further improved the performance. Using Synth-Sámi also improved performance, and the model performed best in terms of mean(CER, WER) when all training datasets were used.

TrOCR models

For TrOCR, we observed that including GT-Nor in the training had a slight improvement when only training with GT-Sámi and no improvement when training with GT-Sámi and Pred-Sámi (see Table 3). Moreover, while including Pred-Sámi improved performance, pre-training with Synth-Sámi had a larger effect. The overall best-performing model was trained with both Synth-Sámi and Pred-Sámi in addition to GT-Sámi.

4.2 NLN test data

Table 4, shows that while Transkribus achieves a lower CER for most languages, it obtains a higher WER and a lower special character F_1-score compared to TrOCR. Tesseract performed worst on this dataset. However, all models greatly improve compared to the baseline, with the CER and WER being reduced by factors between 3.8 and 5.6.

The special character F_1-score in Table 4 shows that the baseline struggles with non-Norwegian Sámi characters. While the F_1 score does not take letter position into account, we also see the same pattern reflected in Table 5, which shows that seven of the ten most common mistakes for the baseline are replacing a non-Norwegian Sámi special character. In contrast, we see that our three

[25] `https://github.com/Sprakbanken/nodalida25_sami_ocr`

		Transkribus	Tesseract	TrOCR	Baseline
CER ↓ [%]	Overall	0.61	0.89	0.74	3.38
	South	0.33	1.09	0.33	2.05
	North	0.53	0.73	1.20	3.99
	Lule	0.34	0.26	0.66	2.46
	Inari	1.22	1.43	0.43	4.36
WER ↓ [%]	Overall	3.19	4.65	2.96	18.71
	South	2.42	7.45	2.33	15.98
	North	1.66	2.90	3.41	20.08
	Lule	3.27	1.84	3.47	13.27
	Inari	6.18	7.13	2.40	22.62
Sámi letter F_1 ↑ [%]	Overall	96.03	93.81	96.97	52.54
	South	90.24	83.02	93.92	24.52
	North	98.57	97.13	97.27	55.85
	Lule	97.91	97.88	97.06	51.75
	Inari	94.70	93.22	98.84	68.61

Table 4: CER, WER and Sámi letter F1 on NLN test data. The score for each language and overall score across languages are listed. Transkribus, Tesseract and TrOCR refer to the best performing model on the validation set for each model type. Baseline is the current OCR output in the online library. The downward arrows indicate that a low score is better, while the upward arrow indicates that a high score is better.

models make fewer mistakes, and their ten most common mistakes are less systematically replacing distinctive Sámi characters and include, e.g. insertions and deletions.

4.3 Giellatekno test data

In contrast to the NLN test data, the Tesseract model performed the best on the OOD test data from Giellatekno for all metrics (see Table 6). Transkribus was worst in terms of CER and WER, while TrOCR was worst in terms of the F_1 score.

In Table 7, we see the most common errors on the Giellatekno test set. The Transkribus model seems to have a tendency to add punctuation marks, and mistake the letter ø for e. All models fail to transcribe ü (of which there are only two in the Giellatekno test set). This is not surprising, as the letter rarely appears in the training data [26].

5 Discussion and conclusions

From Tables 3 and 4, we observe a jump in performance for the test set compared to the validation set. This increase is expected, as the test set annotations are of higher quality (more accurate line segmentations).

We see that applying a two-stage training using synthetic data for the first stage always improved the results. As such, if manual annotations are limited, the addition of synthetic data is worth considering. Furthermore, while the Pred-Sámi improved performance, its effect was less than including synthetic data. It would, thus, be interesting to investigate if further training on Synth-Sámi could eliminate the effect of Pred-Sámi. Finally, we note that including GT-Nor had a minimal effect when combined with Pred-Sámi. This finding, combined with the effect of pre-trained base models, suggests that language-independent features are already learned by the base models and highlights the value of language-specific data for fine-tuning on low-resource languages.

Unfortunately, as this work focuses on low-resource languages, few digitised texts exist. There is, therefore, a slight overlap between the books (but not pages) in the test set and the validation and training sets for Inari Sámi which could bias our results for the Inari Sámi language. Still, Inari Sámi obtained the worst CER and WER for Transkribus and the worst CER and second worst WER for Tesseract. Despite low amount of Inari Sámi, we included it in our analysis as there is an overlap between this alphabet and the North

[26] The letter ü appears 59 times in Synth-Sámi, 9 times in Pred-Sámi and 5 times in GT-Nor.

Transkribus				Tesseract				TrOCR				Baseline			
Error	n_e	n_m	n_c	Error	n_e	n_m	n_c	Error	n_e	n_m	n_c	Error	n_e	n_m	n_c
‘â’→‘á’	16	35	287	‘ï’→‘i’	24	27	160	‘Á’→‘A’	9	11	28	‘á’→‘å’	313	418	1136
‘â’→‘a’	14	35	287	‘â’→‘á’	22	29	287	‘’ → ‘l’	7	–	–	‘ï’→‘i’	137	139	160
‘Á’→‘A’	9	10	28	‘đ’→‘d’	12	14	173	‘Š’→‘S’	6	6	6	‘â’→‘å’	103	180	287
‘/’ → ‘ ’	9	9	10	‘Á’→‘A’	10	11	28	‘’ → ‘i’	5	–	–	‘–’→‘-’	75	77	82
‘i’ → ‘ï’	7	13	3299	‘’ →‘d’	8	–	–	‘’ → ‘ ’	4	–	–	‘š’→‘s’	72	95	215
‘đ’→‘d’	7	11	173	‘’ →‘á’	7	–	–	‘i’ → ‘ï’	4	21	3299	‘đ’→‘d’	48	61	173
‘š’→ ‘’	6	6	215	‘’ →‘i’	7	–	–	‘á’→‘å’	4	14	1136	‘á’→‘a’	46	418	1136
‘ä’→‘á’	5	6	150	‘s’→‘S’	7	8	1509	‘Č’→‘C’	4	4	8	‘â’→‘á’	30	180	287
‘ï’ → ‘i’	5	5	160	‘â’→‘å’	6	29	287	‘á’→‘a’	3	14	1136	‘â’→‘ä’	26	180	287
‘’ →‘-’	4	–	–	‘.’→ ‘’	5	6	509	‘a’→‘u’	3	8	3247	‘č’→‘c’	26	62	163

‘a’→‘b’: model transcribed “a” as “b”
‘a’→ ‘’ : model incorrectly deleted “a”
‘’ →‘b’: model incorrectly inserted “b”

n_e: Error count
n_m: Misses of the character left of →
n_c: Occurrences of the character left of →

Table 5: Top ten most common errors on the NLN test data. Transkribus, Tesseract and TrOCR refers to the best performing model on the validation set for each model type. Baseline is the current OCR output in the online library.

	Transkribus	Tesseract	TrOCR
CER ↓ [%]	0.70	0.12	0.43
WER ↓ [%]	5.85	1.02	3.31
F1 ↑ [%]	100.00	100.00	98.33

Table 6: CER, WER and Sámi letter F_1 on the OOD Giellatekno test set. The downwards arrows indicate that a low score is better, while the upwards arrow indicates that a high score is better.

Sámi alphabet, and our OCR models could improve upon NLN’s transcription for Inari Sámi.

All models improved considerably compared to the baseline and are good candidates for a re-OCR process. If transcription accuracy is the main focus, then Transkribus appears to perform the best. However, while Tesseract achieved the worst performance for the NLN test set, it performed the best on the OOD Giellatekno test set. Tesseract also has other benefits: it is available as open-source software and requires less compute than a TrOCR model.

While language-specific annotations are valuable, they are demanding to create, particularly for low-resource languages without good base models for semi-automatic annotations. However, our results show that by fine-tuning pre-trained models and augmenting manually annotated data with machine-annotated data and synthetic text images, we can achieve accurate OCR for Sámi languages, even with modest amounts of manual annotations.

6 Further work

As NLN’s collection includes works predating the standardised Sámi orthographies, a more accurate evaluation of the OCR could be gained by examining performance across different time periods. Moreover, training specialised models to transcribe non-standard letters or glyph-shapes could enable more detailed down-stream studies of changes in orthographies. Another gap is training OCR for other Sámi languages, such as Skolt Sámi.

Given that our results show that initialising on a dataset of synthetic text images was beneficial, it is worth exploring further. The models in this work are only trained on synthetic data for five epochs, indicating that potential improvements could be made by training on synthetic data for longer, i.e. until convergence. Moreover, creating a larger synthetic dataset with greater variation of text, fonts and augmentations (e.g. additional scanning augmentations or simulating non-standard orthographies), could improve the results further.

As this study focuses on the text recognition step of the OCR pipeline and compares three models, future research should explore additional OCR components and models. E.g. examining the ef-

Transkribus				Tesseract				TrOCR			
Error	n_e	n_m	n_c	Error	n_e	n_m	n_c	Error	n_e	n_m	n_c
‘’ → ‘.’	12	–	–	‘ü’→ ‘i’	1	2	2	‘ü’ → ‘ï’	2	2	2
‘ø’→‘e’	4	5	13	‘ü’→ ‘u’	1	2	2	‘’ → ‘,’	1	–	–
‘’ → ‘,’	2	–	–	‘t’ → ‘f’	1	1	220	‘t’ → ‘l’	1	2	220
‘ü’→‘u’	2	2	2	‘n’→‘m’	1	1	164	‘te’→‘s’	1	2	28
‘’ →‘k’	1	–	–					‘l’ → ‘’	1	1	169
‘ø’→‘o’	1	5	13					‘o’ →‘n’	1	1	149
‘c’→ ‘’	1	1	23					‘m’→‘n’	1	1	69
								‘c’ →‘e’	1	1	23
								‘-’ →‘–’	1	1	18
								‘ŋ’ →‘ž’	1	1	9
								‘=’ →‘2’	1	1	4
								‘x’ →‘s’	1	1	2

‘a’→‘b’: model transcribed “a” as “b” n_e: Error count
‘a’→ ‘’ : model incorrectly deleted “a” n_m: Misses of the character left of →
‘’ →‘b’: model incorrectly inserted “b” n_c: Occurrences of the character left of →

Table 7: Top ten most common errors on the OOD Giellatekno test data. Transkribus, Tesseract and TrOCR refers to the best performing model on the validation set for each model type.

fect of different line segmentation models and assessing if performance can be improved by finetuning the line segmentation or using end-to-end models. Additionally, extending the range of models examined — to include tools such as PyLaia (Puigcerver, 2017; Tarride et al., 2024) (which is part of Transkribus' pipeline), Loghi (van Koert et al., 2024), GOT-OCR (Wei et al., 2024) or larger TrOCR models — could yield improvements. Lastly, including post processing, e.g. with tools from GiellaLT (Pirinen et al., 2023), could improve OCR quality.

Acknowledgments

We would like to thank Arne Martinus Lindstad for his contributions to the annotation of the data and valuable feedback.

References

Milind Agarwal and Antonios Anastasopoulos. 2024. A concise survey of OCR for low-resource languages. In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP)*, pages 88–102. Association for Computational Linguistics.

Magnus Breder Birkenes, Lars Johnsen, and Andre Kåsen. 2023. NB DH-LAB: A corpus infrastructure for social sciences and humanities computing. In *CLARIN Annual Conference Proceedings, 2023*, pages 30–34, Leuven, Belgium. CLARIN.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Quan Duong, Mika Hämäläinen, and Simon Hengchen. 2021. An unsupervised method for OCR post-correction and spelling normalisation for Finnish. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 240–248, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

John Evershed and Kent Fitch. 2014. Correcting noisy OCR: Context beats confusion. In *DATeCH '14: Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, DATeCH '14, pages 45–51, New York, NY, USA. Association for Computing Machinery.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 369–376, New York, NY, USA. Association for Computing Machinery.

Alexander Groleau, Kok Wei Chee, Stefan Larson, Samay Maini, and Jonathan Boarman. 2023. Augraphy: A data augmentation library for document images. In *Document Analysis and Recognition - ICDAR 2023: 17th International Conference, San José, CA, USA, August 21–26, 2023, Proceedings, Part III*, page 384–401, San José, CA, USA. Springer Nature Switzerland.

Marcin Heliński, Miłosz Kmieciak, and Tomasz Parkoła. 2012. Report on the comparison of Tesseract and ABBYY FineReader OCR engines.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Anni Järvelin, Heikki Keskustalo, Eero Sormunen, Miamaria Saastamoinen, and Kimmo Kettunen. 2016. Information retrieval from historical newspaper collections in highly inflectional languages: A query expansion approach. *Journal of the Association for Information Science and Technology*, 67(12):2928–2946.

Philip Kahle, Sebastian Colutto, Günter Hackl, and Günter Mühlberger. 2017. Transkribus - a service platform for transcription, recognition and retrieval of historical documents. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 04, pages 19–24, Kyoto, Japan. IEEE.

Rutger van Koert, Stefan Klut, Tim Koornstra, Martijn Maas, and Luke Peters. 2024. Loghi: An end-to-end framework for making historical documents machine-readable. In *Document Analysis and Recognition – ICDAR 2024 Workshops*, pages 73–88, Athens, Greece. Springer Nature Switzerland.

Mika Koistinen, Kimmo Kettunen, and Tuula Pääkkönen. 2017. Improving optical character recognition of Finnish historical newspapers with a combination of fraktur & antiqua models and image preprocessing. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 277–283, Gothenburg, Sweden. Association for Computational Linguistics.

Johanna Laakso and Elena Skribnik. 2022. Graphization and orthographies of Uralic minority languages. In *The Oxford Guide to the Uralic Languages*, pages 91–100. Oxford University Press.

Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. 2023. TrOCR: Transformer-based optical character recognition with pre-trained models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37 No. 11, pages 13094–13102.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. RoBERTa: A robustly optimized BERT pretraining approach. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Daniel Lopresti. 2008. Optical character recognition errors and their effects on natural language processing. In *Proceedings of the second workshop on Analytics for Noisy Unstructured Text Data*, pages 9–16, New York, NY, United States. Association for Computing Machinery.

Martin Maarand, Yngvil Beyer, Andre Kåsen, Knut T. Fosseide, and Christopher Kermorvant. 2022. A comprehensive comparison of open-source libraries for handwritten text recognition in norwegian. In *Document Analysis Systems: 15th IAPR International Workshop, DAS 2022*, pages 399–413, La Rochelle, France. Springer International Publishing.

Ole Henrik Magga. 1994. Hvordan den nyeste nordsamiske rettskrivningen ble til. In *Festskrift til Ørnulv Vorren*. Tromsø Museum, Universitetet i Tromsø, Tromsø, Norway.

Jamshed Memon, Maira Sami, Rizwan Ahmed Khan, and Mueen Uddin. 2020. Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR). *IEEE access*, 8:142642–142668.

Yngve Mardal Moe and Marie Roald. 2024. Stringalign, version `5499dc8`, [Software]. `https://github.com/yngvem/stringalign`.

Clemens Neudecker, Konstantin Baierer, Mike Gerber, Christian Clausner, Apostolos Antonacopoulos, and Stefan Pletschacher. 2021. A survey of ocr evaluation tools and metrics. In *Proceedings of the 6th International Workshop on Historical Document Imaging and Processing*, HIP '21, page 13–18, New York, NY, USA. Association for Computing Machinery.

Marja-Liisa Olthuis, Suvi Kivelä, and Tove Skutnabb-Kangas. 2013. *Revitalising indigenous languages: How to recreate a lost generation*. Multilingual matters, Bristol, UK.

Niko Partanen and Michael Rießler. 2019. An OCR system for the unified northern alphabet. In *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*, pages 77–89, Tartu, Estonia. Association for Computational Linguistics.

Flammie Pirinen, Sjur Moshagen, and Katri Hiovain-Asikainen. 2023. GiellaLT — a stable infrastructure for Nordic minority languages and beyond. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 643–649, Tórshavn, Faroe Islands. University of Tartu Library.

Alberto Poncelas, Mohammad Aboomar, Jan Buts, James Hadley, and Andy Way. 2020. A tool for facilitating OCR postediting in historical documents. In *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 47–51, Marseille, France. European Language Resources Association (ELRA).

Joan Puigcerver. 2017. Are multidimensional recurrent layers really necessary for handwritten text recognition? In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 67–72, Kyoto, Japan. IEEE.

Shruti Rijhwani, Daisy Rosenblum, Michayla King, Antonios Anastasopoulos, and Graham Neubig. 2023. User-centric evaluation of OCR systems for kwak'wala. In *Proceedings of the Sixth Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 19–29, Remote. Association for Computational Linguistics.

Baoguang Shi, Xiang Bai, and Cong Yao. 2016. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304.

SIKOR. 2021. SIKOR UiT the Arctic University of Norway and the Norwegian Saami Parliament's Saami text collection, version 01.12.2021 [data set]. `http://gtweb.uit.no/korp`.

Ray Smith. 2007. An overview of the Tesseract OCR engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633, Los Alamitos, CA, USA. IEEE, IEEE Computer Society.

Phillip Benjamin Ströbel, Tobias Hodel, Walter Boente, and Martin Volk. 2023. The adaptability of a transformer-based ocr model for historical documents. In *Document Analysis and Recognition – ICDAR 2023 Workshops*, pages 34–48, San José, CA, USA. Springer Nature Switzerland.

Ahmad P Tafti, Ahmadreza Baghaie, Mehdi Assefi, Hamid R Arabnia, Zeyun Yu, and Peggy Peissig. 2016. OCR as a service: an experimental evaluation of Google Docs OCR, Tesseract, ABBYY FineReader, and Transym. In *Advances in Visual Computing 12th International Symposium, ISVC 2016, December 12-14, 2016, Proceedings, Part I 12*, pages 735–746, Las Vegas, NV, USA. Springer.

Solène Tarride, Yoann Schneider, Marie Generali-Lince, Mélodie Boillet, Bastien Abadie, and Christopher Kermorvant. 2024. Improving automatic text recognition with language models in the PyLaia open-source library. In *Document Analysis and Recognition - ICDAR 2024*, pages 387–404. Springer Nature Switzerland.

Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, Chunrui Han, and Xiangyu Zhang. 2024. General OCR theory: Towards OCR-2.0 via a unified end-to-end model. `https://arxiv.org/abs/2409.01704`.

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

Blnd Yaseen and Hossein Hassani. 2024. Making Old Kurdish publications processable by augmenting available optical character recognition engines. `https://arxiv.org/abs/2404.06101`.