OCR Post Correction for Endangered Language Texts

Shruti Rijhwani,[1] **Antonios Anastasopoulos,**[2,†] **Graham Neubig**[1]
[1]Language Technologies Institute, Carnegie Mellon University
[2]Department of Computer Science, George Mason University
{srijhwan,gneubig}@cs.cmu.edu, antonis@gmu.edu

Abstract

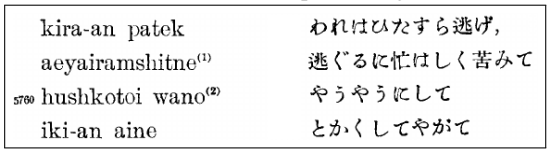
There is little to no data available to build natural language processing models for most endangered languages. However, textual data in these languages often exists in formats that are not machine-readable, such as paper books and scanned images. In this work, we address the task of extracting text from these resources. We create a benchmark dataset of transcriptions for scanned books in three critically endangered languages and present a systematic analysis of how general-purpose OCR tools are not robust to the data-scarce setting of endangered languages. We develop an OCR post-correction method tailored to ease training in this data-scarce setting, reducing the recognition error rate by 34% on average across the three languages.[1]

1 Introduction

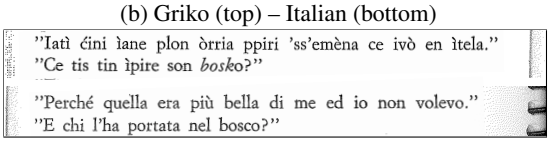
Natural language processing (NLP) systems exist for a small fraction of the world's over 6,000 living languages, the primary reason being the lack of resources required to train and evaluate models. Technological advances are concentrated on languages that have readily available data, and most other languages are left behind (Joshi et al., 2020). This is particularly notable in the case of endangered languages, i.e., languages that are in danger of becoming extinct due to dwindling numbers of native speakers and the younger generations shifting to using other languages. For most endangered languages, finding *any* data at all is challenging.

In many cases, natural language text in these languages does exist. However, it is locked away in formats that are not machine-readable — paper books, scanned images, and unstructured web pages. These include books from local publishing houses within the communities that speak endangered languages, such as educational or cultural materials. Additionally, linguists documenting these languages also create data such as word lists and interlinear glosses, often in the form of handwritten notes. Examples from such scanned documents are shown in Figure 1. Digitizing the textual data from these sources will not only enable NLP for endangered languages but also aid linguistic documentation, preservation, and accessibility efforts.

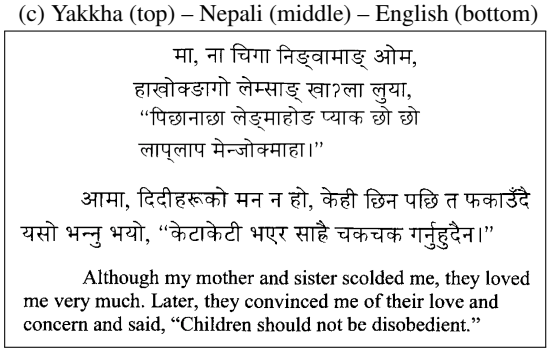
(a) Ainu (left) – Japanese (right)



(b) Griko (top) – Italian (bottom)



(c) Yakkha (top) – Nepali (middle) – English (bottom)



(d) Handwritten Shangaji – typed English glosses

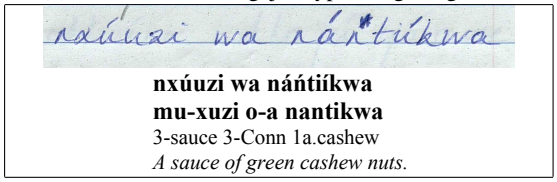


Figure 1: Examples of scanned documents in endangered languages accompanied by translations from the same scanned book (a, b, c) or linguistic archive (d).

†: Work done at Carnegie Mellon University.

[1]Code and data are available at https://shrutirij.github.io/ocr-el/.

In this work, we create a benchmark dataset and propose a suite of methods to extract data from these resources, focusing on scanned images of paper books containing endangered language text. Typically, this sort of digitization requires an optical character recognition (OCR) system. However, the large amounts of textual data and transcribed images needed to train state-of-the-art OCR models from scratch are unavailable in the endangered language setting. Instead, we focus on *post-correcting* the output of an off-the-shelf OCR tool that can handle a variety of scripts. We show that targeted methods for post-correction can significantly improve performance on endangered languages.

Although OCR post-correction is relatively well-studied, most existing methods rely on considerable resources in the target language, including a substantial amount of textual data to train a language model (Schnober et al., 2016; Dong and Smith, 2018; Rigaud et al., 2019) or to create synthetic data (Krishna et al., 2018). While readily available for high-resource languages, these resources are severely limited in endangered languages, preventing the direct application of existing post-correction methods in our setting.

As an alternative, we present a method that compounds on previous models for OCR post-correction, making three improvements tailored to the data-scarce setting. First, we use a **multi-source model** to incorporate information from the high-resource translations that commonly appear in endangered language books. These translations are usually in the *lingua franca* of the region (e.g., Figure 1 (a,b,c)) or the documentary linguist's primary language (e.g., Figure 1 (d) from Devos (2019)). Next, we introduce **structural biases** to ease learning from small amounts of data. Finally, we add **pretraining methods** to utilize the little unannotated data that exists in endangered languages.

We summarize our main contributions as follows:

- A benchmark dataset for OCR post-correction on three critically endangered languages: Ainu, Griko, and Yakkha.
- A systematic analysis of a general-purpose OCR system, demonstrating that it is not robust to the data-scarce setting of endangered languages.
- An OCR post-correction method that adapts the standard neural encoder-decoder framework to the highly under-resourced endangered language setting, reducing both the character error rate and the word error rate by 34% over a state-of-the-art general-purpose OCR system.

2 Problem Setting

In this section, we first define the task of OCR post-correction and introduce how we incorporate translations into the correction model. Next, we discuss the sources from which we obtain scanned documents containing endangered language texts.

2.1 Formulation

Optical Character Recognition OCR tools are trained to find the best transcription corresponding to the text in an image. The system typically consists of a recognition model that produces candidate text sequences conditioned on the input image and a language model that determines the probability of these sequences in the target language. We use a general-purpose OCR system (detailed in Section 4) to produce a *first pass transcription* of the endangered language text in the image. Let this be a sequence of characters $\boldsymbol{x} = [x_1, \ldots, x_N]$.

OCR post-correction The goal of post-correction is to reduce recognition errors in the first pass transcription — often caused by low quality scanning, physical deterioration of the paper book, or diverse layouts and typefaces (Dong and Smith, 2018). The focus of our work is on using post-correction to counterbalance the lack of OCR training data in the target endangered languages. The correction model takes $\boldsymbol{x}$ as input and produces the *final transcription* of the endangered language document, a sequence of characters $\boldsymbol{y} = [y_1, \ldots, y_K]$.

$$\boldsymbol{y} = \arg\max_{\boldsymbol{y'}} p_{\text{corr}}(\boldsymbol{y'}|\boldsymbol{x})$$

Incorporating translations We use information from high-resource translations of the endangered language text. These translations are commonly found within the same paper book or linguistic archive (e.g., Figure 1). We use an existing OCR system to obtain a transcription of the scanned translation, a sequence of characters $\boldsymbol{t} = [t_1, \ldots, t_M]$. This is used to condition the model:

$$\boldsymbol{y} = \arg\max_{\boldsymbol{y'}} p_{\text{corr}}(\boldsymbol{y'}|\boldsymbol{x}, \boldsymbol{t})$$

2.2 Endangered Language Documents

We explore online archives to determine how many scanned documents in endangered languages exist

as potential sources for data extraction (as of this writing, October 2020).

The Internet Archive,[2] a general-purpose archive of web content, has scanned books labeled with the language of their content. We find 11,674 books labeled with languages classified as "endangered" by UNESCO. Additionally, we find that endangered language linguistic archives contain thousands of documents in PDF format — the Archive of the Indigenous Languages of Latin America (AILLA)[3] contains ≈10,000 such documents and the Endangered Languages Archive (ELAR)[4] has ≈7,000.

How common are translations? As described in the introduction, endangered language documents often contain a translation into another (usually high-resource) language. While it is difficult to estimate the number of documents with translations, multilingual documents represent the majority in the archives we examined; AILLA contains 4,383 PDFs with bilingual text and 1,246 PDFs with trilingual text, while ELAR contains ≈5,000 multilingual documents. The structure of translations in these documents is varied, from dictionaries and interlinear glosses to scanned multilingual books.

3 Benchmark Dataset

From the sources described above, we select documents from three critically endangered languages[5] for annotation — Ainu, Griko, and Yakkha. These languages were chosen in an effort to create a geographically, typologically, and orthographically diverse benchmark. We focus this initial study on scanned images of printed books as opposed to handwritten notes, which are a relatively more challenging domain for OCR.

We manually transcribed the text corresponding to the endangered language content. The text corresponding to the translations is not manually transcribed. We also aligned the endangered language text to the OCR output on the translations, per the formulation in Section 2.1. We describe the annotated documents below and example images from our dataset are in Figure 1 (a), (b), (c).

Ainu is a severely endangered indigenous language from northern Japan, typically considered a language isolate. In our dataset, we use a book of Ainu epic poetry (*yukara*), with the "Kutune Shirka" yukara (Kindaichi, 1931) in Ainu transcribed in Latin script.[6] Each page in the book has a two-column structure — the left column has the Ainu text, and the right has its Japanese translation already aligned at the line-level, removing the need for manual alignment (see Figure 1 (a)). The book has 338 pages in total. Given the effort involved in annotation, we transcribe the Ainu text from 33 pages, totaling 816 lines.

Griko is an endangered Greek dialect spoken in southern Italy. The language uses a combination of the Latin alphabet and the Greek alphabet as its writing system. The document we use is a book of Griko folk tales compiled by Stomeo (1980). The book is structured such that in each fold of two pages, the left page has Griko text, and the right page has the corresponding translation in Italian. Of the 175 pages in the book, we annotate the Griko text from 33 pages and manually align it at the sentence-level to the Italian translation. This results in 807 annotated Griko sentences.

Yakkha is an endangered Sino-Tibetan language spoken in Nepal. It uses the Devanagari writing system. We use scanned images of three children's books, each of which has a story written in Yakkha along with its translation in Nepali and English (Schackow, 2012). We manually transcribe the Yakkha text from all three books. We also align the Yakkha text to both the Nepali and the English OCR at the sentence level with the help of an existing Yakkha dictionary (Schackow, 2015). In total, we have 159 annotated Yakkha sentences.

4 OCR Systems: Promises and Pitfalls

As briefly alluded to in the introduction, training an OCR model for each endangered language is challenging, given the limited available data. Instead, we use the general-purpose OCR system from the Google Vision AI toolkit[7] to get the first pass OCR transcription on our data.

The Google Vision OCR system (Fujii et al., 2017; Ingle et al., 2019) is highly performant and supports 60 major languages in 29 scripts. It can transcribe a wide range of higher resource languages with high accuracy, ideal for our proposed method of incorporating high-resource translations

[2] https://archive.org/

[3] https://ailla.utexas.org

[4] https://elar.soas.ac.uk/

[5] UNESCO defines critically endangered languages as those where the youngest speakers are grandparents and older, and they speak the language partially and infrequently.

[6] Some transcriptions of Ainu also use the Katakana script. See Howell (1951) for a discussion on Ainu folklore.

[7] https://cloud.google.com/vision

Language	CER	WER
Ainu	1.34	6.27
Griko	3.27	15.63
Yakkha	8.90	31.64

Table 1: Character error rate and word error rate with the Google Vision OCR system on our dataset.

into the post-correction model. Moreover, it is particularly well-suited to our task because it provides script-specific OCR models in addition to language-specific ones. Per-script models are more robust to unknown languages because they are trained on data from multiple languages and can act as a general character recognizer without relying on a single language's model. Since most endangered languages adopt standard scripts (often from the region's dominant language) as their writing systems, the per-script recognition models can provide a stable starting point for post-correction.

The metrics we use for evaluating performance are character error rate (CER) and word error rate (WER), representing the ratio of erroneous characters or words in the OCR prediction to the total number in the annotated transcription. More details are in Section 6. The CER and WER using the Google Vision OCR on our dataset are in Table 1.

4.1 OCR Performance

Across the three languages, the error rates indicate that we have a first pass transcription that is of reasonable quality, giving our post-correction method a reliable starting point. We note the particularly low CER for the Ainu data, reflecting previous work that has evaluated the Google Vision system to have strong performance on typed Latin script documents (Fujii et al., 2017). However, there remains considerable room for improvement in both CER and WER for all three languages.

Next, we look at the edit distance between the predicted and the gold transcriptions, in terms of insertion, deletion, and replacement operations. Replacement accounts for over 84% of the errors in the Griko and Ainu datasets, and 55% overall. This pattern is expected in the OCR task, as the recognition model uses the image to make predictions and is more likely to confuse a character's shape for another than to hallucinate or erase pixels. However, we observe that the errors in the Yakkha dataset do not follow this pattern. Instead, 87% of the errors for Yakkha occur because of deleted characters.

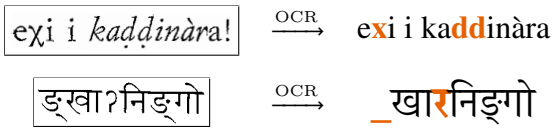


Figure 2: Examples of errors in Griko (top) and Yakkha (bottom) when using the Google Vision OCR.

4.2 Types of Errors

To better understand the challenges posed by the endangered language setting, we manually inspect all the errors made by the OCR system. While some errors are commonly seen in the OCR task, such as misidentified punctuation or incorrect word boundaries, 85% of the total errors occur due to specific characteristics of endangered languages that general-purpose OCR systems do not account for. Broadly, they can be categorized into two types, examples of which are shown in Figure 2:

- **Mixed scripts** The existing scripts that most endangered languages adopt as writing systems are often not ideal for comprehensively representing the language. For example, the Devanagari script does not have a grapheme for the glottal stop — as a solution, printed texts in the Yakkha language use the IPA symbol 'ʔ' (Schackow, 2015). Similarly, both Greek and Latin characters are used to write Griko. The Google Vision OCR is trained to detect script at the line-level and is not equipped to handle multiple scripts within a single word. As seen in Figure 2, the system does not recognize the Greek character χ in Griko and the IPA symbol ʔ in Yakkha. Mixed scripts cause 11% of the OCR errors.
- **Uncommon characters and diacritics** Endangered languages often use graphemes and diacritics that are part of the standard script but are not commonly seen in high-resource languages. Since these are likely rare in the OCR system's training data, they are frequently misidentified, accounting for 74% of the errors. In Figure 2, we see that the OCR system substitutes the uncommon diacritic ḍ in Griko. The system also deletes the Yakkha character ङ्, which is a 'half form' alphabet that is infrequent in several other Devanagari script languages (such as Hindi).

5 OCR Post-Correction Model

In this section, we describe our proposed OCR post-correction model. The base architecture of the model is a multi-source sequence-to-sequence

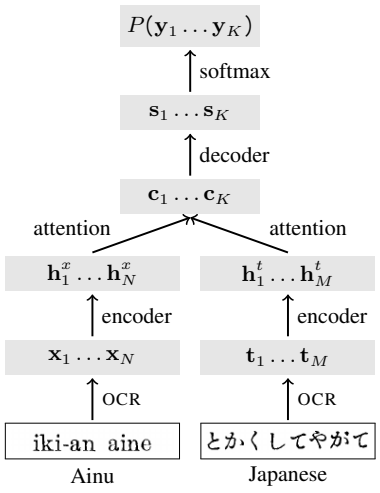


Figure 3: The proposed multi-source architecture with the encoder for an endangered language segment (left) and an encoder for the translated segment (right). The input to the encoders is the first pass OCR over the scanned images of each segment. For example, the OCR on the scanned images of some Ainu text (left) and its Japanese translation (right).

framework (Zoph and Knight, 2016; Libovický and Helcl, 2017) that uses an LSTM encoder-decoder model with attention (Bahdanau et al., 2015). We propose improvements to training and modeling for the multi-source architecture, specifically tailored to ease learning in data-scarce settings.

5.1 Multi-source Architecture

Our post-correction formulation takes as input the first pass OCR of the endangered language segment $\boldsymbol{x}$ and the OCR of the translated segment $\boldsymbol{t}$, to predict an error-free endangered language text $\boldsymbol{y}$. The model architecture is shown in Figure 3.

The model consists of two encoders — one that encodes $\boldsymbol{x}$ and one that encodes $\boldsymbol{t}$. Each encoder is a character-level bidirectional LSTM (Hochreiter and Schmidhuber, 1997) and transforms the input sequence of characters to a sequence of hidden state vectors: $\mathbf{h}^x$ for the endangered language text and $\mathbf{h}^t$ for the translation.

The model uses an attention mechanism during the decoding process to use information from the encoder hidden states. We compute the attention weights over each of the two encoders independently. At the decoding time step k:

$$e^x_{k,i} = \mathbf{v}^x \tanh\left(\mathbf{W}^x_1 \mathbf{s}_{k-1} + \mathbf{W}^x_2 \mathbf{h}^x_i\right) \quad (1)$$

$$\boldsymbol{\alpha}^x_k = \text{softmax}\left(\mathbf{e}^x_k\right)$$

$$\mathbf{c}^x_k = \left[\Sigma_i \alpha^x_{k,i} \mathbf{h}^x_i\right]$$

where $\mathbf{s}_{k-1}$ is the decoder state of the previous time step and $\mathbf{v}^x$, $\mathbf{W}^x_1$ and $\mathbf{W}^x_2$ are trainable parameters. The encoder hidden states $\mathbf{h}^x$ are weighted by the attention distribution $\boldsymbol{\alpha}^x_k$ to produce the context vector $\mathbf{c}^x_k$. We follow a similar procedure for the second encoder to produce $\mathbf{c}^t_k$. We concatenate the context vectors to combine attention from both sources (Zoph and Knight, 2016):

$$\mathbf{c}_k = \left[\mathbf{c}^x_k; \mathbf{c}^t_k\right]$$

$\mathbf{c}_k$ is used by the decoder LSTM to compute the next hidden state $\mathbf{s}_k$ and subsequently, the probability distribution for predicting the next character $\mathbf{y}_k$ of the target sequence $\boldsymbol{y}$:

$$\mathbf{s}_k = \text{lstm}\left(\mathbf{s}_{k-1}, \mathbf{c}_k, \mathbf{y}_{k-1}\right) \quad (2)$$

$$P\left(\mathbf{y}_k\right) = \text{softmax}\left(\mathbf{W}\mathbf{s}_k + \mathbf{b}\right) \quad (3)$$

Training and Inference The model is trained for each language with the cross-entropy loss ($\mathcal{L}_{\text{ce}}$) on the small amount of transcribed data we have. Beam search is used at inference time.

5.2 Model and Training Improvements

With the minimal annotated data we have, it is challenging for the neural network to learn a good distribution over the target characters. We propose a set of adaptations to the base architecture that improves the post-correction performance without additional annotation. The adaptations are based on characteristics of the OCR task itself and the performance of the upstream OCR tool (Section 4).

Diagonal attention loss As seen in Section 4, substitution errors are more frequent in the OCR task than insertions or deletions; consequently, we expect the source and target to have similar lengths. Moreover, post-correction is a monotonic sequence-to-sequence task, and reordering rarely occurs (Schnober et al., 2016). Hence, we expect attention weights to be higher at characters close to the diagonal for the endangered language encoder.

We modify the model such that all the elements in the attention vector that are not within j steps (we use $j = 3$) of the current time step k are added to the training loss, thereby encouraging elements away from the diagonal to have lower values. The diagonal loss summed over all time steps for a training instance, where N is the length of $\boldsymbol{x}$, is:

$$\mathcal{L}_{\text{diag}} = \sum_k \left(\sum_{i=1}^{k-j} \alpha^x_{k,i} + \sum_{i=k+j}^{N} \alpha^x_{k,i} \right)$$

Copy mechanism Table 1 indicates that the first pass OCR predicts a majority of the characters accurately. In this scenario, enabling the model to directly copy characters from the first pass OCR rather than generate a character at each time step might lead to better performance (Gu et al., 2016).

We incorporate the copy mechanism proposed in See et al. (2017). The mechanism computes a "generation probability" at each time step k, which is used to choose between *generating* a character (Equation 3) or *copying* a character from the source text by sampling from the attention distribution $\boldsymbol{\alpha}^x_k$.

Coverage Given the monotonicity of the post-correction task, the model should not attend to the same character repeatedly. However, repetition is a common problem for neural encoder-decoder models (Mi et al., 2016; Tu et al., 2016). To account for this problem, we adapt the coverage mechanism from See et al. (2017), which keeps track of the attention distribution over past time steps in a coverage vector. For time step k, the coverage vector would be $\mathbf{g}_k = \sum_{k'=0}^{k-1} \boldsymbol{\alpha}^x_{k'}$.

$\mathbf{g}_k$ is used as an extra input to the attention mechanism, ensuring that future attention decisions take the weights from previous time steps into account. Equation 1, with learnable parameter $\mathbf{w}_g$, becomes:

$$e^x_{k,i} = \mathbf{v}^x \tanh\left(\mathbf{W}^x_1 \mathbf{s}_{k-1} + \mathbf{W}^x_2 \mathbf{h}^x_i + \mathbf{w}_g g_{k,i}\right)$$

$\mathbf{g}_k$ is also used to penalize attending to the same locations repeatedly with a coverage loss. The coverage loss summed over all time steps k is:

$$\mathcal{L}_{\text{cov}} = \sum_k \sum_{i=1}^{n} \min\left(\alpha^x_{k,i}, g_{k,i}\right)$$

Therefore, with our model adaptations, the loss for a single training instance:

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \mathcal{L}_{\text{diag}} + \mathcal{L}_{\text{cov}} \tag{4}$$

5.3 Utilizing Untranscribed Data

As discussed in Section 3, given the effort involved, we transcribe only a subset of the pages in each scanned book. Nonetheless, we leverage the remaining unannotated pages for pretraining our model. We use the upstream OCR tool to get a first pass transcription on all the unannotated pages.

We then create "pseudo-target" transcriptions for the endangered language text as described below:

- **Denoising rules** Using a small fraction of the available annotated pages, we compute the edit distance operations between the first pass OCR and the gold transcription. The operations of each type (insertion, deletion, and replacement) are counted for each character and divided by the number of times that character appears in the first pass OCR. This forms a probability of how often the operation should be applied to that specific character.

  We use these probabilities to form rules, such as $p(\text{replace d with ḍ}) = 0.4$ for Griko and $p(\text{replace ? with ʔ}) = 0.7$ for Yakkha. These rules are applied to remove errors from, or "denoise", the first pass OCR transcription.

- **Sentence alignment** We use Yet Another Sentence Aligner (Lamraoui and Langlais, 2013) for unsupervised alignment of the endangered language and translation on the unannotated pages.

Given the aligned first pass OCR for the endangered language text and the translation along with the pseudo-target text, $\boldsymbol{x}$, $\boldsymbol{t}$ and $\hat{\boldsymbol{y}}$ respectively, the pretraining steps, in order, are:

- **Pretraining the encoders** We pretrain both the forward and backward LSTMs of each encoder with a character-level language model objective: the endangered language encoder on $\boldsymbol{x}$ and the translation encoder on $\boldsymbol{t}$.

- **Pretraining the decoder** The decoder is pretrained on the pseudo-target $\hat{\boldsymbol{y}}$ with a character-level language model objective.

- **Pretraining the seq-to-seq model** The model is pretrained with $\boldsymbol{x}$ and $\boldsymbol{t}$ as the sources and the pseudo-target $\hat{\boldsymbol{y}}$ as the target transcription, using the post-correction loss function $\mathcal{L}$ as defined in Equation 4.

6 Experiments

This section discusses our experimental setup and the post-correction performance on the three endangered languages on our dataset.

6.1 Experimental Setup

Data Splits We perform 10-fold cross-validation for all experimental settings because of the small size of the datasets. For each language, we divide the transcribed data into 11 segments — we use one segment for creating the *denoising rules* described in the previous section and the remaining ten as the

Model	Character Error Rate						Word Error Rate					
	Ainu		Griko		Yakkha		Ainu		Griko		Yakkha	
	Multi	Single	Multi	Single	Multi	Single	Multi	Single	Multi	Single	Multi	Single
FP-OCR	–	1.34	–	3.27	–	8.90	–	6.27	–	15.63	–	31.64
BASE	1.56	1.41	6.78	5.95	70.39	71.71	8.56	7.88	15.13	13.67	98.15	99.10
COPY	2.04	1.99	2.54	2.28	14.77	12.30	9.48	8.57	9.33	8.90	30.36	27.81
OURS	0.92	**0.80**	**1.66**	1.70	**7.75**	8.44	5.75	**5.19**	**7.46**	7.51	**20.95**	21.33

Table 2: Our method improves performance over all baselines (10-fold cross-validation averaged over five randomly seeded runs). We present multi- and single-source variants and **highlight** the best model for each language.

folds for cross-validation. In each cross-validation fold, eight segments are used for training, one for validation and one for testing.

We divide the dataset at the page-level for the Ainu and Griko documents, resulting in 11 segments of three pages each. For the Yakkha documents, we divide at the paragraph-level, due to the small size of the dataset. We have 33 paragraphs across the three books in our dataset, resulting in 11 segments that contain three paragraphs each. The multi-source results for Yakkha reported in Table 2 use the English translations. Results with Nepali are similar and are included in Appendix A.

Metrics We use two metrics for evaluating our systems: character error rate (CER) and word error rate (WER). Both metrics are based on edit distance and are standard for evaluating OCR and OCR post-correction (Berg-Kirkpatrick et al., 2013; Schulz and Kuhn, 2017). CER is the edit distance between the predicted and the gold transcriptions of the document, divided by the total number of characters in the gold transcription. WER is similar but is calculated at the word level.

Methods In our experiments, we compare the performance of our proposed method with the first pass OCR and with two systems from recent work in OCR post-correction. All the post-correction methods have two variants – the single-source model with only the endangered language encoder and the multi-source model that additionally uses the high-resource translation encoder.

- FP-OCR: The first pass transcription obtained from the Google Vision OCR system.
- BASE: This system is the base sequence-to-sequence architecture described in Section 5.1. Both the single-source and multi-source variants of this system are used for English OCR post-correction in Dong and Smith (2018).
- COPY: This system is the base architecture with a copy mechanism as described in Section 5.2. The single-source variant of this model is used for OCR post-correction on Romanized Sanskrit in Krishna et al. (2018).[8]
- OURS: The model with all the adaptations proposed in Section 5.2 and Section 5.3.

Implementation The post-correction models are implemented using the DyNet neural network toolkit (Neubig et al., 2017), and all reported results are the average of five training runs with different random seeds. We assume knowledge of the entire alphabet of the endangered language for all the methods, which is straightforward to obtain for most languages. The decoder's vocabulary contains all these characters, irrespective of their presence in the training data, with corresponding randomly-initialized character embeddings.

6.2 Main Results

Table 2 shows the performance of the baselines and our proposed method for each language. Overall, our method results in an improved CER and WER over existing methods across all three languages.

The BASE system does not improve the recognition rate over the first pass transcription, apart from a small decrease in the Griko WER. The performance on Yakkha, particularly, is significantly worse than FP-OCR: likely because the data size of Yakkha is much smaller than that of Griko and Ainu, and the model is unable to learn a reasonable distribution. However, on adding a copy mechanism to the base model in the COPY system, the performance is notably better for both Griko and Yakkha. This indicates that adaptations to the base model that cater to specific characteristics of the

[8] Although Krishna et al. (2018) use BPE tokenization, preliminary experiments showed that character-level models result in much better performance on our dataset, likely due to the limited data available for training the BPE model.

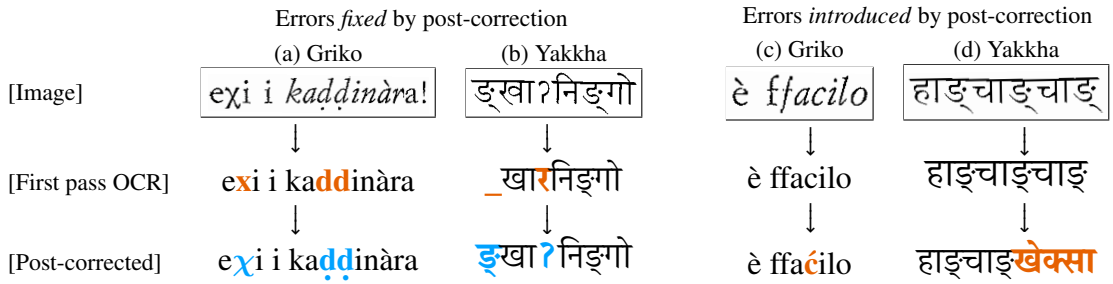


Figure 4: Our model fixes many mixed script and uncommon diacritics errors such as (a) and (b). In rare cases, it "over-corrects" the first pass OCR transcription, introducing errors such as (c) and (d).

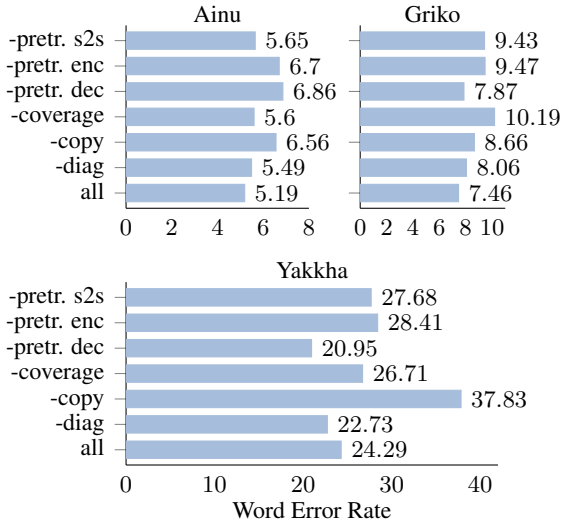


Figure 5: WER with model component ablations on the best model setting in Table 2. "all" includes all the adaptations we propose. Each ablation removes a single component from the "all" model, e.g. "-pretr. s2s" removes the seq-to-seq model pretraining.

post-correction task can alleviate some of the challenges of learning from small amounts of data.

The single-source and the multi-source variants of our proposed method improve over the baselines, demonstrating that our proposed model adaptations can improve recognition even without translations. We see that using the high-resource translations results in better post-correction performance for Griko and Yakkha, but the single-source model achieves better accuracy for Ainu. We attribute this to two factors: the very low error rate of the first pass transcription for Ainu and the relatively high error rate (based on manual inspection) of the OCR on the Japanese translation. Despite being a high-resource language, OCR is difficult due to the complexity of Japanese characters and low scan quality. The noise resulting from the Japanese OCR errors likely hurts the multi-source model.

6.3 Ablation Studies

Next, we study the effect of our proposed adaptations and evaluate their benefit to the performance of each language. Figure 5 shows the word error rate with models that remove one adaptation from the model with all the adaptations ("all").

For Ainu and Griko, removing any single component increases the WER, with the complete ("all") method performing the best. There is little variance in the Ainu ablations, likely due to the high-quality first pass transcription.

Our proposed adaptations add the most benefit for Yakkha, which has the fewest training data and relatively less accurate first pass OCR. The copy mechanism is crucial for good performance, but removing the decoder pretraining ("pretr. dec") leads to the best scores among all the ablations. The denoising rules used to create the pseudo-target data for Yakkha are likely not accurate since they are derived from only three paragraphs of annotated data. Consequently, using it to pretrain the decoder leads to a poor language model.

6.4 Error Analysis

We systematically inspect all the recognition errors in the output of our post-correction model to determine the sources of improvement with respect to the first pass OCR. We also examine the types of errors introduced by the post-correction process.

We observe a *91% reduction* in the number of errors due to mixed scripts and a *58% reduction* in the errors due to uncommon characters and diacritics (as defined in Section 4). Examples of these are shown in Figure 4 (a) and (b): mixed script errors such as the χ character in Griko and the glottal stop ʔ in Yakkha are successfully corrected by the model. The model is also able to correct uncommon character errors like ḍ in Griko and ङ in Yakkha.

Examples of errors introduced by the model are shown in Figure 4 (c) and (d). Example (c) is in Griko, where the model incorrectly adds a diacritic to a character. We attribute this to the fact that the first pass OCR does not recognize diacritics well; hence, the model learns to add diacritics frequently while generating the output. Example (d) is in Yakkha. The model inserts several incorrect characters, and can likely be attributed to the lack of a good language model due to the relatively smaller amount of training data we have in Yakkha.

7 Related Work

Post-correction for OCR is well-studied for high-resource languages. Early approaches include lexical methods and weighted finite-state methods (see Schulz and Kuhn (2017) for an overview). Recent work has primarily focused on using neural sequence-to-sequence models. Hämäläinen and Hengchen (2019) use a BiLSTM encoder-decoder with attention for historical English post-correction. Similar to our base model, Dong and Smith (2018) use a multi-source model to combine the first pass OCR from duplicate documents in English.

There has been little work on lower-resourced languages. Kolak and Resnik (2005) present a probabilistic edit distance based post-correction model applied to Cebuano and Igbo, and Krishna et al. (2018) show improvements on Romanized Sanksrit OCR by adding a copy mechanism to a neural sequence-to-sequence model.

Multi-source encoder-decoder models have been used for various tasks including machine translation (Zoph and Knight, 2016; Libovický and Helcl, 2017) and morphological inflection (Kann et al., 2017; Anastasopoulos and Neubig, 2019). Perhaps most relevant to our work is the multi-source model presented by Anastasopoulos and Chiang (2018), which uses high-resource translations to improve speech transcription of lower-resourced languages.

Finally, Bustamante et al. (2020) construct corpora for four endangered languages from text-based PDFs using rule-based heuristics. Data creation from such unstructured text files is an important research direction, complementing our method of extracting data from scanned images.

8 Conclusion

This work presents a first step towards extracting textual data in endangered languages from scanned images of paper books. We create a benchmark dataset with transcribed images in three endangered languages: Ainu, Griko, and Yakkha. We propose an OCR post-correction method that facilitates learning from small amounts of data, which results in a 34% average relative error reduction in both the character and word recognition rates.

As future work, we plan to investigate the effect of using other available data for the three languages (for example, word lists collected by documentary linguists or the additional Griko folk tales collected by Anastasopoulos et al. (2018)).

Additionally, it would be valuable to examine whether our method can improve the OCR on high-resource languages, which typically have much better recognition rates in the first pass transcription than the endangered languages in our dataset.

Further, we note our use of the Google Vision OCR system to obtain the first pass OCR for our experiments, primarily because it provides script-specific models as opposed to other general-purpose OCR systems that rely on language-specific models (as discussed in Section 4). Future work that focuses on overcoming the challenges of applying language-specific models to endangered language texts would be needed to confirm our method's applicability to post-correcting the first pass transcriptions from different OCR systems.

Lastly, given the annotation effort involved, this paper explores only a small fraction of the endangered language data available in linguistic and general-purpose archives. Future work will focus on large-scale digitization of scanned documents, aiming to expand our OCR benchmark on as many endangered languages as possible, in the hope of both easing linguistic documentation and preservation efforts and collecting enough data for NLP system development in under-represented languages.

Acknowledgements

We thank David Chiang, Walter Scheirer, and William Theisen for initial discussions on the project, the University of Notre Dame Library for the scanned "Kutune Shirka" Ainu-Japanese book, and Josep Quer for the scanned Griko folktales book. We also thank Taylor Berg-Kirkpatrick, Shuyan Zhou, Zi-Yi Dou, Yansen Wang, Zhen Fan, and Deepak Gopinath for feedback on the paper.

This material is based upon work supported in part by the National Science Foundation under Grant No. 1761548. Shruti Rijhwani is supported by a Bloomberg Data Science Ph.D. Fellowship.

References

Antonios Anastasopoulos and David Chiang. 2018. Leveraging translations for speech transcription in low-resource settings. In *Proc. INTERSPEECH*.

Antonios Anastasopoulos, Marika Lekakou, Josep Quer, Eleni Zimianiti, Justin DeBenedetto, and David Chiang. 2018. Part-of-speech tagging on an endangered language: a parallel Griko-Italian resource. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2529–2539, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Antonios Anastasopoulos and Graham Neubig. 2019. Pushing the limits of low-resource morphological inflection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 984–996, Hong Kong, China. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Taylor Berg-Kirkpatrick, Greg Durrett, and Dan Klein. 2013. Unsupervised transcription of historical documents. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 207–217, Sofia, Bulgaria. Association for Computational Linguistics.

Gina Bustamante, Arturo Oncevay, and Roberto Zariquiey. 2020. No data to crawl? monolingual corpus creation from PDF files of truly low-resource languages in Peru. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2914–2923, Marseille, France. European Language Resources Association.

Maud Devos. 2019. Shangaji. a maka or swahili language of mozambique. grammar, texts and wordlist. `https://elar.soas.ac.uk/Collection/MPI1029699`. Accessed: 2020-02-02.

Rui Dong and David Smith. 2018. Multi-input attention for unsupervised OCR correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2363–2372, Melbourne, Australia. Association for Computational Linguistics.

Yasuhisa Fujii, Karel Driesen, Jonathan Baccash, Ash Hurst, and Ashok C Popat. 2017. Sequence-to-label script identification for multilingual ocr. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 161–168. IEEE.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Mika Hämäläinen and Simon Hengchen. 2019. From the paft to the fiiture: a fully automatic NMT and word embeddings method for OCR post-correction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 431–436, Varna, Bulgaria. INCOMA Ltd.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Richard W Howell. 1951. The classification and description of ainu folklore. *The Journal of American Folklore*, 64(254):361–369.

R Reeve Ingle, Yasuhisa Fujii, Thomas Deselaers, Jonathan Baccash, and Ashok C Popat. 2019. A scalable handwritten text recognition system. *arXiv preprint arXiv:1904.09150*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2017. Neural multi-source morphological reinflection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 514–524, Valencia, Spain. Association for Computational Linguistics.

Kyōsuke Kindaichi. 1931. *Ainu Jojishi Yūkara no Kenkyū [Research on Ainu Epic Yukar]*. Tōkyō: Tōkyō Bunko.

Okan Kolak and Philip Resnik. 2005. OCR post-processing for low density languages. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 867–874, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Amrith Krishna, Bodhisattwa P. Majumder, Rajesh Bhat, and Pawan Goyal. 2018. Upcycle your OCR: Reusing OCRs for post-OCR text correction in Romanised Sanskrit. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 345–355, Brussels, Belgium. Association for Computational Linguistics.

Fethi Lamraoui and Philippe Langlais. 2013. Yet another fast, robust and open source sentence aligner. time to reconsider sentence alignment? In *XIV Machine Translation Summit*, Nice, France.

Jindřich Libovický and Jindřich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 196–202, Vancouver, Canada. Association for Computational Linguistics.

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage embedding models for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 955–960, Austin, Texas. Association for Computational Linguistics.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, Paul Michel, Yusuke Oda, Matthew Richardson, Naomi Saphra, Swabha Swayamdipta, and Pengcheng Yin. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

C. Rigaud, A. Doucet, M. Coustaty, and J. Moreux. 2019. ICDAR 2019 competition on post-OCR text correction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1588–1593.

Diana Schackow. 2012. Documentation and grammatical description of yakkha, nepal. `https://elar.soas.ac.uk/Collection/MPI186180`. Accessed: 2020-02-02.

Diana Schackow. 2015. *A grammar of Yakkha*. Language Science Press.

Carsten Schnober, Steffen Eger, Erik-Lân Do Dinh, and Iryna Gurevych. 2016. Still not there? comparing traditional sequence-to-sequence models to encoder-decoder neural networks on monotone string translation tasks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1703–1714, Osaka, Japan. The COLING 2016 Organizing Committee.

Sarah Schulz and Jonas Kuhn. 2017. Multi-modular domain-tailored OCR post-correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2716–2726, Copenhagen, Denmark. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Paolo Stomeo. 1980. *Racconti greci inediti di Sternatía*. La nuova Ellade, s.I.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany. Association for Computational Linguistics.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 30–34, San Diego, California. Association for Computational Linguistics.

A Appendix

A.1 Implementation Details

The hyperparameters used are:

- Character embedding size = 128
- Number of LSTM layers = 1
- Hidden state size of the LSTM = 256
- Attention size = 256
- Beam size = 4
- For the diagonal loss, $j = 3$
- Minibatch size for training = 1
- Maximum number of epochs = 150
- Patience for early stopping = 10 epochs
- Pretraining epochs for encoder/decoder = 10
- Pretraining epochs for seq-to-seq model = 5

We use the same values of the hyperparameters for each language and all the systems. We select the best model with early stopping on the character error rate of the validation set.

A.2 Additional Experimental Results

Performance on Yakkha with Nepali Table 3 shows the performance for the Yakkha dataset when using Nepali as the high-resource translation input to the multisource model. The performance is similar to those of the experiments using the English translations, as presented in Table 2.

Standard deviation on the main results Table 4 and Table 5 show the character error rate and word error rate respectively including the standard deviation over five randomly seeded runs, corresponding to the systems described in Table 2.

Model	CER	WER
FP-OCR	8.90	31.64
BASE	70.89	100.00
COPY	11.60	26.74
OURS	7.95	20.83

Table 3: Character error rate (CER) and word error rate (WER) for the Yakkha dataset with the multisource model that uses the OCR on Nepali as the high-resource translation. The table shows the mean over five random runs.

(a) Ainu

Model	Multi	Single
FP-OCR	–	1.34
BASE	1.56 ± 0.23	1.41 ± 0.16
COPY	2.04 ± 0.62	1.99 ± 0.41
OURS	0.92 ± 0.05	**0.80** ± 0.07

(b) Griko

Model	Multi	Single
FP-OCR	–	3.27
BASE	6.78 ± 0.62	5.95 ± 0.52
COPY	2.54 ± 0.31	2.28 ± 0.28
OURS	**1.66** ± 0.03	1.70 ± 0.21

(c) Yakkha

Model	Multi	Single
FP-OCR	–	8.90
BASE	70.39 ± 0.49	71.71 ± 0.71
COPY	14.77 ± 0.97	12.30 ± 2.39
OURS	**7.75** ± 0.46	8.44 ± 0.90

Table 4: Mean and standard deviation of the character error rate with 10-fold cross-validation over five random seeds. The results presented are the same as Table 2 with the added information of standard deviation. The best models for each language are **highlighted**.

(a) Ainu

Model	Multi	Single
FP-OCR	–	6.27
BASE	8.56 ± 1.01	7.88 ± 0.64
COPY	9.48 ± 3.07	8.57 ± 1.45
OURS	5.75 ± 0.24	**5.19** ± 0.31

(b) Griko

Model	Multi	Single
FP-OCR	–	15.63
BASE	15.13 ± 0.99	13.67 ± 1.17
COPY	9.33 ± 0.49	8.90 ± 0.51
OURS	**7.46** ± 0.09	7.51 ± 0.31

(c) Yakkha

Model	Multi	Single
FP-OCR	–	31.64
BASE	98.15 ± 1.55	99.10 ± 2.20
COPY	30.36 ± 1.39	27.81 ± 1.65
OURS	**20.95** ± 1.04	21.33 ± 0.53

Table 5: Mean and standard deviation of the word error rate with 10-fold cross-validation over five random seeds. The results presented are the same as Table 2 with the added information of standard deviation. The best models for each language are **highlighted**.