Mobilizing the Archive: Bitext Extraction at Scale from Archival Documents for Extremely Low-resource Languages

Chutong Meng
George Mason University
cmeng2@gmu.edu

Alec J. Bateman
Barron Associates
bateman@barron-associates.com

Antonios Anastasopoulos
George Mason University
antonis@gmu.edu

arXiv:2610.40233v1 [cs.CL] 29 Oct 2026

Abstract

While mining parallel sentences from web pages has proven effective for low-resource machine translation, the majority of the world's languages have little to no web presence. What they *do* have, though, are resources that are the result of centuries of efforts by linguists, such as grammars, dictionaries, and other documentation materials. We propose extracting translation examples from such linguistic resources, which offer parallel data for languages under-represented online and could potentially scale to thousands more languages. We develop a pipeline to achieve this, leveraging an Optical Character Recognition (OCR) model as well as a vision language model (VLM), to ensure identified content to be aligned with visual layout information. We demonstrate the utility of the extracted bitexts for bootstrapping translation systems across 20 languages, both by training machine translation models and by applying in-context learning for large language models. For in-context learning, we observe improvements of +15.78 and +6.94 chrF2++ over zero-shot translation for into low- and high-resource languages, respectively.[1]

1 Introduction

Modern machine translation (MT) models are typically trained on tens to hundreds of millions of *parallel sentences* (also termed as *bitexts*). These data are obtained by aligning parallel documents (Koehn, 2005) or mining from large-scale monolingual web data (Schwenk et al., 2021b; NLLB Team et al., 2022). However, most of the world's languages lack sufficient web data and, consequently, parallel sentences (Joshi et al., 2020), making data scarcity a challenge for underrepresented languages.

Large language models (LLMs) offer a potential alternative through in-context learning, whereby models can adapt to new tasks from a small number of examples provided in context (Brown et al., 2020; Wei et al., 2022). Recent work has used various linguistic resources, such as grammar book excerpts, dictionaries, morphological analyzers, and interlinear glossed text (IGT), to instruct LLMs to translate *unseen* low-resource languages (Tanzer et al., 2024; Zhang et al., 2024; Ramos et al., 2025). Among these resources, bitext examples appear to provide the largest gains (Aycock et al., 2025), motivating our focus on extracting parallel examples.

Linguistic resource collections can also provide substantially broader language coverage than web-based bitext mining. For example, the DReaM corpus (Virk et al., 2020) contains grammar books, language overviews, and dictionaries covering more than 4,000 languages, whereas large-scale bitext mining typically supports only around 200 languages reliably (NLLB Team et al., 2022). Although linguistic resources contain fewer parallel sentences, often only a few thousand, we show they can still lead to substantial improvements when used as in-context examples for MT (cf., Fig. 1).

Extracting structured data from heterogeneous linguistic resources at scale, however, is challenging. Prior work either relies on resources with predefined machine-readable structure (Hus and Anastasopoulos, 2024; Tanzer et al., 2024), or requires human effort to identify and extract relevant information based on textual and visual formatting cues (Merx et al., 2024; Aycock et al., 2025). Such approaches are difficult to generalize across previously unseen resources with diverse layouts.

Therefore, in this work, we develop an (almost) fully automatic pipeline for extracting bitexts from heterogeneous linguistic resources in PDF format from the DReaM corpus. Our pipeline consists of two stages. First, we apply optical character recognition (OCR) to recover text. Second, we use a vision language model (VLM) to identify parallel sentences using both the OCR output and page

[1] Our code and extracted data are publicly available at https://github.com/mct10/mobilizing_the_archive.

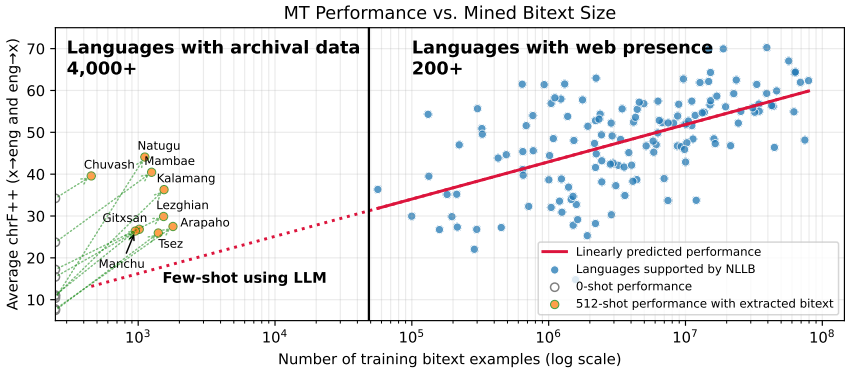


Figure 1: We illustrate the rough relationship between mined bitext size and MT performance, using `nllb-200-3.3B` as an example. Archival linguistic resources could extend bitext coverage from around 200 languages to thousands more, while substantially improving few-shot translation quality toward NLLB-level performance.

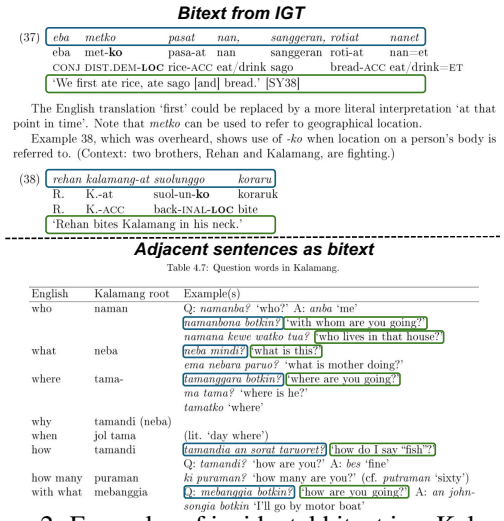


Figure 2: Examples of incidental bitext in a Kalamang book (Visser, 2016). Blue boxes indicate Kalamang sentences, and green boxes indicate English sentences.

image. Visual information is important because linguistic resources often encode parallelism through visual structure that is lost in plain text, such as IGT, adjacent translations, and parallel columns. Figure 2 shows an example of such incidental bitext. Applying our pipeline to 20 languages, we extract an average of 2,578 bitext examples per language.

2 Related Work

We review how previous work extracts structured data from linguistic resources. Appendix A provides additional discussion of existing bitext alignment methods.

2.1 Linguistic Resources Based MT

Recent work has shown that grammar books and other linguistic resources can improve MT for *unseen* languages when used with LLMs (Tanzer et al., 2024). A key challenge, however, is converting heterogeneous and poorly structured archival materials into usable structured data.

Several studies leverage structured or semi-structured machine readable resources. Tanzer et al. (2024) use LaTeX and plaintext grammar books together with well-formatted parallel sentences from *Dictionaria*, while Aycock et al. (2025) extract parallel examples based on text formatting.

OCR is frequently used for archival materials. Both Zhang et al. (2024) and Hus and Anastasopoulos (2024) use OCRed grammar books. Merx et al. (2024) OCR Mambai documents, then extract dictionaries based on text formatting and identify parallel sentences using Hunalign (Varga et al., 2007).

Manual intervention is sometimes needed. Zhang et al. (2025), for example, manually extract grammar rules and parallel examples from a grammar book. In some cases, extraction details are unspecified. Pei et al. (2025) extract parallel sentences from a dictionary without describing the procedure.

Unlike this line of prior work which primarily studies how different linguistic resources and prompting strategies impact MT performance, we focus on resource construction itself. We propose to extract useful data from archival linguistic resources with minimal manual intervention.

Most closely related to our work, Setiawan et al. (2026b) combine OCR with an LLM to extract parallel sentences and lexicons from PDF grammar books, but provide limited methodological detail and evaluate only one language. Setiawan et al. (2026a) adopt a two-stage approach to leverage OCR and VLM for dictionary entry extraction, but do not consider bitext extraction or downstream MT evaluation.

LRL name	LRL code	HRL	NLLB support	Existing data	
				Data source	Script
Arapaho	arp	eng	No	SIGMORPHON 2023	Latin
Chuvash	chv	eng	No	FLORES, BOUQuET	Cyrillic
Tsez	ddo	eng	No	SIGMORPHON 2023	Latin
Dogri	dgo	eng	No	FLORES, BOUQuET	Devanagari
Gitxsan	git	eng	No	SIGMORPHON 2023	Latin
Kalamang	kgv	eng	No	Tanzer et al. (2024)	Latin
Lezghian	lez	eng	No	SIGMORPHON 2023	Cyrillic
Mambae	mgm	eng	No	Merx et al. (2024)	Latin
Manchu	mnc	eng	No	Zhang et al. (2024)	Latin
Natügu	ntu	eng	No	SIGMORPHON 2023	Latin
Uspanteco	usp	spa	No	SIGMORPHON 2023	Latin
Southwestern Dinka	dik	eng	Yes	FLORES, BOUQuET	Latin
Paraguayan Guaraní	gug	eng	Yes	FLORES, BOUQuET	Latin
Iloko	ilo	eng	Yes	FLORES, BOUQuET	Latin
Southern Jinghpaw	kac	eng	Yes	FLORES, BOUQuET	Latin
Kabuverdianu	kea	eng	Yes	FLORES, BOUQuET	Latin
Kimbundu	kmb	por	Yes	FLORES, BOUQuET	Latin
Mizo	lus	eng	Yes	FLORES	Latin
Minangkabau	min	eng	Yes	FLORES	Latin
Wolof	wol	eng	Yes	FLORES, BOUQuET	Latin

Table 1: Summary of languages and their resource status, separated by NLLB support. Kimbundu is paired with Portuguese, for which our collection contains substantially more books, while Uspanteco is paired with Spanish to match its available test set. BOUQuET is only used as a secondary evaluation set where available.

3 Languages and Status

We select extremely under-resourced languages studied in prior work, which allows us to evaluate our method using existing test sets. Table 1 summarizes the languages and their resource status.

Our experimental design distinguishes between languages with and without web presence. Languages supported by NLLB (NLLB Team et al., 2022) have at least some, if not abundant, monolingual text online, making it likely that frontier LLMs have encountered them during pre-training. In contrast, other languages can be treated as approximating languages that are unlikely to have been seen in LLM pre-training. The latter group of languages allows us to simulate a setting in which a model is asked to translate a previously unseen language and the only available translation examples are extracted from archival resources.

4 Method

We extract and clean up bitexts from multilingual archival PDFs in three steps. First, we apply multilingual OCR to recover text (Section 4.1). Second, we use a VLM to identify bitext based on visual cues (Section 4.2). Finally, we clean and filter the resulting alignments (Section 4.3).

Model	CER (%)↓
PaddleOCR-VL	34.13
HunyuanOCR	11.37
dots.ocr	7.88
dots.ocr with retry	**5.66**

Table 2: CER for three OCR models on the synthesized FLORES PDFs.

4.1 OCR

Recently developed neural OCR models outperform traditional systems in accuracy and human-readability (Poznanski et al., 2025), but their multilingual performance had not been comprehensively benchmarked when we began this work. We therefore evaluate 3 multilingual models, dots.ocr (Li et al., 2025), HunyuanOCR (Hunyuan Vision Team et al., 2025), and PaddleOCR-VL (Cui et al., 2025) on synthetic multilingual PDFs.

Inspired by Ignat et al. (2022), we render FLORES (NLLB Team et al., 2022) devtest sentences into PDFs using playwright.[2] FLORES covers over 200 languages across 31 scripts. We report macro-averaged character error rate (CER) across languages in Table 2, with per-script results in Figure 3 in Appendix B.

dots.ocr performs best and supports the broadest range of scripts, consistent with recent multilingual OCR benchmarks (Kargaran et al., 2026;

[2] https://github.com/microsoft/playwright

Xu et al., 2026). However, it occasionally produces endless repetitions. Following Poznanski et al. (2025), we retry inference with a higher temperature when generation exceeds the length limit, reducing CER from 7.88% to 5.66%.

4.2 Vision Language Model

Incidental bitexts often exhibit certain *visual properties*, such as IGT or adjacent sentences in different languages (Figure 2). We therefore use a VLM that takes a page image and its OCR text as input and identifies candidate bitext pairs based on visual cues. Providing OCR text (1) avoids requiring the VLM to perform OCR, (2) leverages specialized OCR models that perform better on low-resource languages, and (3) enables the VLM to understand the content directly based on text.

We assume that each book primarily contains two languages: a low-resource language (LRL) described in a high-resource language (HRL). The LRL is obtained from the resource metadata. We identify the HRL by applying language identification (langID) to each book, counting sentences per language, and selecting the most frequent language. Since we target specific language directions (e.g., `arp-eng`), we retain only books matching the target pair. The VLM is expected to have substantial knowledge of the HRL and treat the remaining text as the LRL.

We use `Qwen3.5-27B-FP8` (Qwen Team, 2026a) as the backbone VLM, served using `vllm` (Kwon et al., 2023). Larger models performed substantially better in preliminary experiments, but computational constraints prevented us from using an even larger model. Prompts are provided in Appendix C.

4.3 Postprocessing

Raw extracted sentence pairs are noisy. For example, IGT may contain linguistic annotations, dictionaries may provide word-level rather than sentence-level alignments, and the VLM may produce incorrect alignments. We therefore follow Koehn et al. (2018, 2019) and apply language-agnostic cleaning and filtering (Section 4.3.1). We additionally reconcile orthographic mismatches between archival resources and existing data using language-specific normalization (Section 4.3.2).

We do not consider such postprocessing as a disadvantage, as conventional bitext mining pipelines likewise require extensive cleaning and filtering.

4.3.1 Language-agnostic Cleaning and Filtering

We apply the following language-agnostic cleaning and filtering steps:

- Remove IGT annotation symbols (e.g., `-`, `=`, and `ø`) from LRL sentences.
- Remove pairs where either sentence contains only digits or punctuation.
- Remove pairs where one sentence is more than twice as long as the other in characters.
- Remove pairs where neither sentence contains more than two space-separated words.
- Remove pairs with source-target Levenshtein similarity ratio above 0.9.
- Apply langID to both sentences. The HRL sentence must be identified as the HRL, while the LRL sentence must not be.

4.3.2 Language-specific Orthography Normalization

Orthographic mismatch, particularly in diacritics and script, is a major challenge in the extremely low-resource settings we are studying. Oftentimes, this is because of genuine diachronic changes in the orthography preferred by language communities and other stakeholders (e.g. documentary linguists) at different times. For all languages, we apply minor corrections for common OCR errors, identified through manual inspection of sampled book pages, plus the following language-specific normalization.

arp, ilo, kac, mgm, mnc We remove diacritics.

chv We transliterate the Latin script used in our resources to the Cyrillic script used by FLORES, following an online document.[3]

dgo Our resources primarily use the International Phonetic Alphabet (IPA), whereas FLORES uses Devanagari. As we could not find existing conversion schemes, we let the coding agent Codex derive rules by comparing FLORES dev sentences with text from the archival resources.

git We remove diacritics. The SIGMORPHON data additionally uses Combining Low Line (U+0332), e.g., K̲ (K followed by U+0332), which our OCR model fails to recognize. We do not restore it.

[3] https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/835784/ROMANIZATION_OF_CHUVASH.pdf

Lang	BLASER 2.0-QE↑	
	Extracted	Test / FLORES
dik	2.935	**3.281**
gug	3.375	**3.732**
ilo	3.356	**4.274**
kac	3.129	**3.570**
kea	3.293	**3.965**
kmb	2.470	**3.344**
lus	2.953	**3.696**
min	3.245	**4.138**
wol	3.123	**3.511**
Avg.	3.098	**3.723**

Table 3: QE scores for supported languages on extracted and FLORES bitexts. **Bold** indicates better performance. FLORES bitexts are of higher quality.

lez We convert the Latin script used in our resources to the Cyrillic script used by SIGMORPHON, following Haspelmath (1993).

ntu We convert the older orthography used in archival resources to the modern SIGMORPHON orthography following an online resource.[4]

Table 10 in Appendix D summarizes these procedures. While the reader may be disappointed to see that such manual efforts are still needed in an otherwise fully automatic pipeline, these rules primarily reconcile archival orthographic conventions with modern usage and evaluation standards, rather than fixing bitext extraction itself. We also note that such manual verification processes might be needed regardless, to reconcile archival material orthographies with current community practices.

4.4 Intrinsic Evaluation

We evaluate bitext quality using BLASER 2.0-QE (Dale and Costa-jussà, 2024), due to its broad language coverage. For supported languages, we score each sentence pair and report average QE scores for the extracted bitext and corresponding test set in Table 3. Scores for unsupported languages are unreliable and reported separately in Table 12 in the Appendix F.1.

Results The extracted bitexts achieve an average score of 3.098, compared with 3.723 for the test set. Although this gap is substantial, a score of 3 indicates that the sentences are *generally equivalent* in meaning, with only minor differences permitted under the XSTS scale (Licht et al., 2022) predicted by BLASER 2.0-QE.

[4] https://en.wikipedia.org/wiki/Nat%C3%BCgu_language#Orthography

Lang	#Extracted	Existing Datasets		
		#Train	#Dev	#Test
Without NLLB support				
arp	1,796	36,123	4,892	4,892
chv	454	–	997 / 504	1,012 / 854
ddo	1,403	3,547	445	445
dgo	137	–	498 / 504	499 / 854
git	1,018	31	42	37
kgv	1,545	375	–	100
lez	1,533	634	88	87
mgm	1,253	1,187	–	50
mnc	953	3	–	102
ntu	1,118	789	99	99
usp	1,007	9,116	232	633
With NLLB support				
dik	3,238	–	997 / 504	1,012 / 854
gug	910	–	997 / 504	1,012 / 854
ilo	14,263	–	997 / 504	1,012 / 854
kac	817	–	997 / 504	1,012 / 854
kea	1,483	–	997 / 504	1,012 / 854
kmb	13,067	–	997 / 504	1,012 / 854
lus	2,477	–	997	1,012
min	459	–	997	1,012
wol	2,632	–	997 / 504	1,012 / 854

Table 4: Number of sentences in the extracted and existing datasets. For FLORES, the devtest set is used as the test set. In cells with two numbers, the left one corresponds to FLORES and the right one to BOUQuET.

5 MT Experiments for Extrinsic Evaluation

We further assess the quality of extracted bitexts through few-shot LLM translation and MT model fine-tuning.

5.1 Training and Evaluation Data

As in Table 1, 11 languages are evaluated by FLORES,[5] 6 are evaluated using the SIGMORPHON 2023 Interlinear Glossing shared task data (Ginn et al., 2023),[6] and the remaining 3 rely on test sentences curated in prior work. We additionally use BOUQuET (Andrews et al., 2025) as a *secondary* evaluation set for the 9 supported languages.

We detokenize the SIGMORPHON test sentences using Moses (Koehn et al., 2007), while leaving other test sets unchanged.

To prevent train-test overlap, we remove extracted sentence pairs whose source or target has Levenshtein similarity ratio above 0.9 with any dev or test sentence.

Table 4 reports the number of extracted sentences after filtering and the sizes of existing datasets.

[5] https://huggingface.co/datasets/openlanguagedata/flores_plus

[6] https://github.com/sigmorphon/2023glossingST

5.2 Experiment Setup

Few-shot experiments We use `Qwen3.6-27B` (Qwen Team, 2026b) (hereafter `Qwen`). Additional results with the smaller `Llama-3.1-8B` (Grattafiori et al., 2024) are reported in Appendix F.2. Prompt templates are provided in Appendix C.

For example retrieval, we score candidate sentences with the input sentence by chrF2++ (Popović, 2017) and select the highest-scoring pairs. Prior work finds that retrieval method has limited impact on the peak MT performance (Setiawan et al., 2026b). An ablation study in Appendix F.3, though, shows that 512 examples perform best.

Fine-tuning experiments We fine-tune `nllb-200-distilled-1.3B` (hereafter `nllb`) using LoRA adapters (Hu et al., 2022). Due to the tiny datasets, we train one multilingual model on all extracted data using temperature-based sampling. More details are in Appendix G.

Baselines For languages supported by NLLB, we compare the fine-tuned `nllb` against the out-of-the-box model. For unsupported LRL-to-HRL translaion, we use the proxy source language codes in Table 7.[7] For all languages, we compare few-shot translation using `Qwen` against zero-shot.

Evaluation Metrics To evaluate, we lowercase hypotheses and references and remove punctuation except apostrophes. We report spBLEU[8] with the flores200 tokenizer (NLLB Team et al., 2022) and chrF2++,[9] both computed with SacreBLEU (Post, 2018). We compute statistical significance using paired bootstrap resampling (Koehn, 2004).[10]

6 Results and Analysis

We present main results in Section 6.1, analyze the limitations of traditional sentence alignment methods in Section 6.2, discuss the limitations of archival resource utility in Section 6.3, and explore broader applicability in Section 6.4.

6.1 Main Results

Tables 5 and 6 report chrF2++ scores for translation in and out of LRLs respectively. Corresponding spBLEU results are in Appendix Tables 14 and 15. Appendix F.5 reports results on BOUQuET.

Lang	Qwen		nllb	
	0 shot	512 shot	OOB	fine-tuned
Without NLLB support				
arp	3.86	*32.90 ↑29.04	–	9.96
chv	20.99	*30.49 ↑ 9.50	–	5.14
ddo	0.17	*24.24 ↑24.07	–	1.93
dgo	21.18	*20.68 ↓ 0.50	–	4.12
git	5.50	*23.16 ↑17.66	–	3.56
kgv	8.38	*39.23 ↑30.85	–	7.60
lez	9.58	*25.26 ↑15.68	–	3.97
mgm	17.34	*37.91 ↑20.57	–	18.78
mnc	3.92	*27.65 ↑23.73	–	3.03
ntu	5.51	*48.76 ↑43.25	–	9.39
usp	6.51	*28.09 ↑21.58	–	7.15
Avg.	*9.36*	*30.76* ↑21.40	–	*6.78*
With NLLB support				
dik	6.36	*16.89 ↑10.53	19.81	*11.28 ↓ 8.53
gug	14.90	*26.05 ↑11.15	37.27	*31.73 ↓ 5.54
ilo	42.39	*46.76 ↑ 4.37	52.56	*27.83 ↓24.73
kac	4.01	*25.91 ↑21.90	38.56	*26.03 ↓12.53
kea	29.50	*40.26 ↑10.76	45.33	*39.37 ↓ 5.96
kmb	12.33	*16.19 ↑ 3.86	24.96	*10.46 ↓14.50
lus	15.37	*24.56 ↑ 9.19	36.70	*8.73 ↓27.97
min	39.36	*43.40 ↑ 4.04	49.50	*26.02 ↓23.48
wol	22.62	*26.93 ↑ 4.31	25.72	*9.01 ↓16.71
Avg.	*20.76*	*29.66* ↑ 8.90	*36.71*	*21.16* ↓15.55
Avg.	14.49	30.27 ↑15.78	–	13.25

Table 5: Extracted data improves few-shot translation performance *into LRLs*. Results are reported in chrF2++(↑). OOB denotes out-of-the-box performance. *p-value < 0.05.

Extracted bitexts substantially improve `Qwen` performance for all languages except `dgo`, with average gains of 15.78 and 6.94 chrF2++ for translation into LRLs and HRLs, respectively. The lack of improvement for `dgo` is likely due to its automatically derived orthography normalization rules.

Fine-tuning `nllb` does not improve translation into LRLs, likely because of the limited training data (Koehn and Knowles, 2017). However, it improves translation into HRLs by 3.59 chrF2++ on average. Overall, we show that the extracted bitexts provide useful supervision even at this small scale.

6.2 A Failed Attempt: Bitext Extraction with Existing Sentence Alignment Methods

Given OCRed text, we also attempted to mine bitext using established sentence alignment methods. However, they perform poorly.

Sentence Embedding Evaluation State-of-the-art sentence alignment methods such as Global Mining (Schwenk et al., 2021b) and Vecalign (Thompson and Koehn, 2019) depend on

[7] In some cases, proxy source and target codes are identical (e.g., `eng_Latn` for both sides of `arp-eng`). This has negligible impact, as alternative proxies yield similar results.

[8] nrefs:1|case:lc|eff:no|tok:flores200|smooth:exp|version:2.6.0

[9] nrefs:1|case:lc|eff:yes|nc:6|nw:2|space:no|version:2.6.0

[10] bs:1000|seed:12345

Lang	Qwen		nllb	
	0 shot	512 shot	OOB	fine-tuned
Without NLLB support†				
arp	11.67	*22.09 ↑10.42	8.62	*12.59 ↑ 3.97
chv	47.46	*48.66 ↑ 1.20	19.72	*24.47 ↑ 4.75
ddo	14.76	*27.64 ↑12.88	10.27	*16.55 ↑ 6.28
dgo	47.52	*50.50 ↑ 2.98	42.86	*39.53 ↓ 3.33
git	25.39	*30.49 ↑ 5.10	15.42	*24.87 ↑ 9.45
kgv	13.30	*33.38 ↑20.08	11.89	*18.04 ↑ 6.15
lez	24.85	*34.51 ↑ 9.66	9.64	*20.51 ↑10.87
mgm	30.11	*42.99 ↑12.88	20.61	*30.30 ↑ 9.69
mnc	18.27	*24.85 ↑ 6.58	8.51	*12.33 ↑ 3.82
ntu	15.23	*40.01 ↑24.78	9.44	*38.73 ↑29.29
usp	11.74	*26.05 ↑14.31	9.98	*16.07 ↑ 6.09
Avg.	*23.66*	*34.65* ↑10.99	*15.18*	*23.09* ↑ 7.91
With NLLB support				
dik	23.94	*25.39 ↑ 1.45	29.49	*28.99 ↓ 0.50
gug	42.50	*44.06 ↑ 1.56	47.96	*45.19 ↓ 2.77
ilo	56.33	*58.02 ↑ 1.69	60.14	*58.58 ↓ 1.56
kac	24.24	*26.83 ↑ 2.59	38.29	*37.36 ↓ 0.93
kea	64.93	*69.59 ↑ 4.66	67.64	67.19 ↓ 0.45
kmb	24.96	*25.42 ↑ 0.46	30.62	*29.89 ↓ 0.73
lus	34.05	*35.83 ↑ 1.78	41.21	*39.09 ↓ 2.12
min	58.25	*60.40 ↑ 2.15	60.18	*56.66 ↓ 3.52
wol	38.14	*39.75 ↑ 1.61	39.21	*36.73 ↓ 2.48
Avg.	*40.82*	*42.81* ↑ 1.99	*46.08*	*44.41* ↓ 1.67
Avg.	31.38	38.32 ↑ 6.94	29.09	32.68 ↑ 3.59

Table 6: Extracted data improves few-shot translation performance *into HRLs*. Results are reported in chrF2++(↑). †Proxy source language codes are used. *p-value < 0.05.

high-quality sentence embeddings. We therefore evaluate embedding quality for our low-resource languages using xSIM (multilingual similarity search).[11] Given aligned bitext, xSIM embeds both source and target sentences into a shared space and retrieves the best target for each source sentence using a margin score (Artetxe and Schwenk, 2019a), where x and y are sentence embeddings and $\text{NN}_k(\cdot)$ finds the k nearest neighbors:

$$\text{sim}(x, y) = \frac{\cos(x, y)}{\sum_{z \in \text{NN}_k(x)} \frac{\cos(x,z)}{2k} + \sum_{z \in \text{NN}_k(y)} \frac{\cos(y,z)}{2k}} \quad (1)$$

We report alignment error against gold alignments. We evaluate LASER2 (Heffernan et al., 2022), LABSE (Feng et al., 2022), and SONAR embeddings (Duquenne et al., 2023) on test sets listed in Table 1. We use LASER2 rather than LASER3 because the latter is language-specific. And for languages unsupported by SONAR, we use the closest supported language code.

[11] https://github.com/facebookresearch/LASER/blob/main/tasks/xsim/README.md

Lang	LASER2 xsim↓	LABSE xsim↓	SONAR lang code	SONAR xsim↓
Without NLLB support†				
arp	96.89	**94.60**	eng_Latn	95.34
chv	76.19	33.70	tat_Cyrl	**33.40**
ddo	97.08	**90.34**	eng_Latn	90.79
dgo	40.48	0.80	hin_Deva	**0.20**
git	62.16	**43.24**	eng_Latn	54.05
kgv	84.00	61.00	ind_Latn	**52.00**
lez	87.36	**51.72**	rus_Cyrl	77.01
mgm	46.00	**28.00**	ind_Latn	34.00
mnc	**93.14**	95.10	khk_Cyrl	**93.14**
ntu	80.81	**52.53**	ind_Latn	70.71
usp	96.37	95.58	spa_Latn	**94.79**
With NLLB support				
dik	79.45	38.04	dik_Latn	**9.58**
gug	71.74	17.09	grn_Latn	**0.10**
ilo	61.96	6.62	ilo_Latn	**0.00**
kac	91.50	40.32	kac_Latn	**0.20**
kea	22.04	0.49	kea_Latn	**0.00**
kmb	74.01	39.62	kmb_Latn	**0.89**
lus	87.15	23.22	lus_Latn	**0.49**
min	42.59	1.19	min_Latn	**0.00**
wol	76.38	26.19	wol_Latn	**0.69**
Avg.	73.37	41.97		**35.37**

Table 7: xSIM error rate (%) on test sets. SONAR achieves the lowest error rate. †Proxy SONAR language codes are used.

Alignment Method	Precision	Recall	F1
Methods without prior knowledge			
Global Mining	0.0610	0.0549	0.0578
Local Mining	0.0851	0.2473	0.1266
Vecalign	0.2776	0.4560	0.3451
Hunalign w/o dict.	0.2435	0.2582	0.2507
VLM (our method)	**1.0000**	**0.7657**	**0.8673**
Methods with prior knowledge			
Hunalign w/ dict.	0.5876	0.6264	0.6064

Table 8: Alignment performance of different methods. VLM achieves the best performance. Its imperfect recall is due to pages dominated by word-level alignments, where sentence-level alignments are overlooked.

Table 7 reports xSIM error rates. Although SONAR performs best overall, error rates remain high for unseen languages such as Mambai (34%) and Kalamang (52%), limiting the effectiveness of embedding-based alignment.

Sentence Alignment Evaluation We evaluate alignment methods on mgm-eng, which has a relatively low SONAR xSIM error rate. Alignment performance on languages with higher xSIM error rates is expected to be worse.

We first segment OCR text into sentences, and use a langID model to separate two languages in the book. We then apply embedding based *Global Mining*, *Local Mining*, *Vecalign*, and non-embedding

Lang	Into LRL		Into HRL	
	Gold	Book	Gold	Book
Without NLLB support				
arp	48.32	*32.90 ↓15.42	39.30	*22.09 ↓17.21
chv	33.77	*30.49 ↓ 3.28	49.96	*48.66 ↓ 1.30
ddo	43.59	*24.24 ↓19.35	42.38	*27.64 ↓14.74
dgo	34.52	*20.68 ↓13.84	54.62	*50.50 ↓ 4.12
git	21.28	23.16 ↑ 1.88	26.08	*30.49 ↑ 4.41
kgv	38.97	39.23 ↑ 0.26	34.56	33.38 ↓ 1.18
lez	32.13	*25.26 ↓ 6.87	41.42	*34.51 ↓ 6.91
mgm	33.44	*37.91 ↑ 4.47	42.08	42.99 ↑ 0.91
†mnc	7.84	*27.65 ↑19.81	20.67	*24.85 ↑ 4.18
ntu	47.79	48.76 ↑ 0.97	39.00	40.01 ↑ 1.01
usp	42.15	*28.09 ↓14.06	42.21	*26.05 ↓16.16
With NLLB support				
dik	23.98	*16.89 ↓ 7.09	27.09	*25.39 ↓ 1.70
gug	34.84	*26.05 ↓ 8.79	45.50	*44.06 ↓ 1.44
ilo	51.30	*46.76 ↓ 4.54	59.09	*58.02 ↓ 1.07
kac	39.84	*25.91 ↓13.93	29.89	*26.83 ↓ 3.06
kea	57.46	*40.26 ↓17.20	71.50	*69.59 ↓ 1.91
kmb	33.62	*16.19 ↓17.43	30.32	*25.42 ↓ 4.90
lus	37.17	*24.56 ↓12.61	38.54	*35.83 ↓ 2.71
min	51.12	*43.40 ↓ 7.72	62.94	*60.40 ↓ 2.54
wol	32.07	*26.93 ↓ 5.14	42.06	*39.75 ↓ 2.31
Avg.	38.81	30.40 ↓ 8.41	43.08	38.99 ↓ 4.09

Table 9: chrF2++ scores for 512-shot translation with Qwen. Extracted bitexts are narrowing the gap to gold in-domain bitexts. †Excluded from the average as it has only 3 gold examples. *p-value < 0.05.

based *Hunalign* to OCRed book content page by page. Further details are provided in Appendix E.

We manually align 182 sentence pairs from randomly sampled pages of a Mambae manual (Hull, 2001) as groundtruths and report precision, recall, and F1 in Table 8. Without an auxiliary dictionary, traditional alignment methods perform poorly, while VLM achieves highest F1. Beyond poor embedding quality, text preprocessing introduces substantial errors. Sentence segmentation can fail when multiple parallel pairs occur on the same line or when OCR splits a single sentence across multiple lines. LangID errors can cause sentences in the same language to be incorrectly aligned.

VLM-based alignment avoids these failure modes by directly accessing images. It can infer sentence boundaries and correspondences from visual structure without separate sentence splitting, langID, or auxiliary dictionaries, producing substantially better alignment.

6.3 Using In-domain Parallel Sentences for Prompting

Section 6.1 assumes no additional parallel examples and uses only potentially out-of-domain extracted bitexts. As a skyline, we use available in-domain training or dev set sentences to quantify the gap between in-domain and extracted examples.

Table 9 compares few-shot MT performance with book examples and gold in-domain examples. We use FLORES dev set sentences when available and training set sentences otherwise.

Compared with book examples, in-domain examples improve the average chrF2++ from 30.40 to 38.81 for translation into LRLs and by 4.09 chrF2++ for into HRLs. For most languages, in-domain examples substantially outperform book examples, likely due to their higher quality (Table 3). Interestingly, book examples perform better in git, mgm, mnc, and ntu, possibly because more book examples are available and some books overlap with those used to construct the test sets. It suggests that extracting bitexts from books is still a feasible way towards narrowing the gap in LRL translation when gold bitexts are unavailable.

6.4 Broader Applicability: CreoleMT

To examine broader applicability, we apply our pipeline to 27 of the lowest-resource Creole languages in Kreyòl-MT (Robinson et al., 2024). Unlike our main experiments, we use only language-agnostic cleaning and filtering, without language-specific orthographic normalization. We evaluate few-shot experiments using Qwen on the public Kreyòl-MT test set using the same evaluation metrics. Results are reported in Table 17 in Appendix F.6.

Extracted bitexts substantially improve translation into Creole languages from 16.87 to 35.71 chrF2++ on average, and translation into English from 34.54 to 48.79 chrF2++. These results suggest that the pipeline generalizes to a broader set of low-resource languages even without language-specific human intervention.

7 Conclusion

We introduce a vision language model based pipeline for extracting parallel sentences from archival linguistic resources by leveraging their visual structure. Across more than 20 languages, few-shot translation with an LLM using the extracted bitext substantially improves low-resource MT over zero-shot translation, demonstrating the value of archival linguistic resources as a source of parallel data.

Limitations

Orthography mismatches remain challenging for archival materials. Proper normalization still requires manually designed language-specific rules, while OCR systems can struggle with diacritics and visually similar characters encoded as different Unicode code points (e.g., Latin and Cyrillic a).

Not all archival books are equally suitable sources of parallel data. Many of the books in our collection were published decades ago and may use historical or outdated orthographic conventions that differ from those used in contemporary evaluation data. Moreover, given the heterogeneous nature of archival linguistic resources, some books primarily cover topics such as phonology or grammatical descriptions and may contain relatively little parallel text useful for machine translation. Careful resource selection is therefore important for obtaining high-quality and relevant examples. Future advances in OCR and agentic systems might help automate these steps and further reduce manual intervention.

Acknowledgments

This material is partially supported by the National Science Foundation under CAREER award 2439202 and partially based upon work supported by the ASA(ALT) SBIR CcoE under Contract No. W51701-25-C-0060. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ASA(ALT) SBIR CcoE. We acknowledge the support by resources provided by the Office of Research Computing at George Mason University (https://orc.gmu.edu) and funded in part by grants from the NSF (Award Number 2018631).

References

Pierre Andrews, Mikel Artetxe, Mariano Coria Meglioli, Marta R. Costa-jussà, Joe Chuang, David Dale, Mark Duppenthaler, Nathanial Paul Ekberg, Cynthia Gao, Daniel Edward Licht, Jean Maillard, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Eduardo Sánchez, Ioannis Tsiamas, Arina Turkatenko, Albert Ventayol-Boada, and Shireen Yates. 2025. BOUQuET : dataset, benchmark and open initiative for universal quality evaluation in translation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27515–27535, Suzhou, China. Association for Computational Linguistics.

Pierre Andrews, Guillaume Wenzek, Kevin Heffernan, Onur Çelebi, Anna Sun, Ammar Kamran, Yingzhe Guo, Alexandre Mourachko, Holger Schwenk, and Angela Fan. 2022. stopes - modular machine translation pipelines. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 258–265, Abu Dhabi, UAE. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019a. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019b. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Seth Aycock, David Stap, Di Wu, Christof Monz, and Khalil Sima'an. 2025. Can LLMs really learn to translate a low-resource language from one grammar book? In *The Thirteenth International Conference on Learning Representations*.

Peter F. Brown, Jennifer C. Lai, and Robert L. Mercer. 1991. Aligning sentences in parallel corpora. In *29th Annual Meeting of the Association for Computational Linguistics*, pages 169–176, Berkeley, California, USA. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Everlyn Asiko Chimoto and Bruce A. Bassett. 2022. Very low resource sentence alignment: Luhya and Swahili. In *Proceedings of the Fifth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2022)*, pages 1–8, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, Jiaxuan Liu, Xueqing Wang, Changda Zhou, Hongen Liu, Manhui Lin, Yue Zhang, Yubo Zhang, Handong Zheng, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. 2025. Paddleocr-vl: Boosting multilingual document parsing via a 0.9b ultra-compact vision-language model. *Preprint*, arXiv:2510.14528.

David Dale and Marta R. Costa-jussà. 2024. BLASER 2.0: a metric for evaluation and quality estimation

of massively multilingual speech and text translation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16075–16085, Miami, Florida, USA. Association for Computational Linguistics.

Paul-Ambroise Duquenne, Holger Schwenk, and Benoît Sagot. 2023. Sonar: Sentence-level multimodal and language-agnostic representations. *Preprint*, arXiv:2308.11466.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

William A. Gale and Kenneth W. Church. 1993. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102.

Michael Ginn, Sarah Moeller, Alexis Palmer, Anna Stacey, Garrett Nicolai, Mans Hulden, and Miikka Silfverberg. 2023. Findings of the SIGMORPHON 2023 shared task on interlinear glossing. In *Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 186–201, Toronto, Canada. Association for Computational Linguistics.

Luís Gomes and Gabriel Pereira Lopes. 2016. First steps towards coverage-based sentence alignment. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2228–2231, Portorož, Slovenia. European Language Resources Association (ELRA).

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Martin Haspelmath. 1993. *A Grammar of Lezgian*. De Gruyter Mouton, Berlin, Boston.

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext mining using distilled sentence representations for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2101–2112, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Geoffrey Hull. 2001. *Mambai Language Manual (Ainaro dialect)*. Dili: Sebastião Aparício da Silva Project.

Hunyuan Vision Team, Pengyuan Lyu, Xingyu Wan, Gengluo Li, Shangpin Peng, Weinong Wang, Liang Wu, Huawen Shen, Yu Zhou, Canhui Tang, Qi Yang, Qiming Peng, Bin Luo, Hower Yang, Xinsong Zhang, Jinnian Zhang, Houwen Peng, Hongming Yang, Senhao Xie, and 13 others. 2025. Hunyuanocr technical report. *Preprint*, arXiv:2511.19575.

Jonathan Hus and Antonios Anastasopoulos. 2024. Back to school: Translation using grammar books. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20207–20219, Miami, Florida, USA. Association for Computational Linguistics.

Oana Ignat, Jean Maillard, Vishrav Chaudhary, and Francisco Guzmán. 2022. OCR improves machine translation for low-resource languages. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1164–1174, Dublin, Ireland. Association for Computational Linguistics.

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.

Amir Hossein Kargaran, Nafiseh Nikeghbal, Jana Diesner, François Yvon, and Hinrich Schütze. 2026. Glotocr bench: Ocr models still struggle beyond a handful of unicode scripts. *Preprint*, arXiv:2604.12978.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In

Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the WMT 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739, Belgium, Brussels. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Yumeng Li, Guang Yang, Hao Liu, Bowen Wang, and Colin Zhang. 2025. dots.ocr: Multilingual document layout parsing in a single vision-language model. *Preprint*, arXiv:2512.02498.

Daniel Licht, Cynthia Gao, Janice Lam, Francisco Guzman, Mona Diab, and Philipp Koehn. 2022. Consistent human evaluation of machine translation across language pairs. In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 309–321, Orlando, USA. Association for Machine Translation in the Americas.

Raphaël Merx, Aso Mahmudi, Katrina Langford, Leo Alberto de Araujo, and Ekaterina Vylomova. 2024. Low-resource machine translation through retrieval-augmented LLM prompting: A study on the Mambai language. In *Proceedings of the 2nd Workshop on Resources and Technologies for Indigenous, Endangered and Lesser-resourced Languages in Eurasia (EURALI) @ LREC-COLING 2024*, pages 1–11, Torino, Italia. ELRA and ICCL.

Robert C. Moore. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 135–144, Tiburon, USA. Springer.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Renhao Pei, Yihong Liu, Peiqin Lin, François Yvon, and Hinrich Schuetze. 2025. Understanding in-context machine translation for low-resource languages: A case study on Manchu. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8767–8788, Vienna, Austria. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Jake Poznanski, Aman Rangapur, Jon Borchardt, Jason Dunkelberger, Christopher Wilhelm, Kyle Lo, and Luca Soldaini. 2025. olmOCR: Unlocking trillions of tokens in PDFs with vision language models. In *Championing Open-source DEvelopment in ML Workshop @ ICML25*.

Qwen Team. 2026a. Qwen3.5: Towards native multimodal agents.

Qwen Team. 2026b. Qwen3.6-27B: Flagship-level coding in a 27B dense model.

Rita Ramos, Everlyn Asiko Chimoto, Maartje Ter Hoeve, and Natalie Schluter. 2025. GrammaMT: Improving machine translation with grammar-informed in-context learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29920–29940, Vienna, Austria. Association for Computational Linguistics.

Nathaniel R. Robinson, Raj Dabre, Ammon Shurtz, Rasul Dent, Onenamiyi Onesi, Claire Bizon Monroc, Loïc Grobol, Hasan Muhammad, Ashi Garg, Naome A. Etori, Vijay Murari Tiyyala, Olanrewaju Samuel, Matthew Dean Stutzman, Bismarck Bamfo Odoom, Sanjeev Khudanpur, Stephen D. Richardson, and Kenton Murray. 2024. Kreyòl-MT: Building MT for Latin American, Caribbean and colonial African creole languages. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3083–3110, Mexico City, Mexico. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021a. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021b. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online. Association for Computational Linguistics.

Rico Sennrich and Martin Volk. 2010. MT-based sentence alignment for OCR-generated parallel texts. In *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas: Research Papers*, Denver, Colorado, USA. Association for Machine Translation in the Americas.

David Setiawan, Temuulen Khishigsuren, Milind Agarwal, Pagnarith Pit, Aso Mahmudi, and Ekaterina Vylomova. 2026a. Mudidi: A two-stage framework for multilingual dictionary digitization with language models. *Preprint*, arXiv:2606.09435.

David Samuel Setiawan, Raphaël Merx, and Jey Han Lau. 2026b. Context volume drives performance: Tackling domain shift in extremely low-resource translation via rag. *Preprint*, arXiv:2601.09982.

Steve Sloto, Brian Thompson, Huda Khayrallah, Tobias Domhan, Thamme Gowda, and Philipp Koehn. 2023. Findings of the WMT 2023 shared task on parallel data curation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 95–102, Singapore. Association for Computational Linguistics.

Garrett Tanzer, Mirac Suzgun, Eline Visser, Dan Jurafsky, and Luke Melas-Kyriazi. 2024. A benchmark for learning to translate a new language from one grammar book. In *The Twelfth International Conference on Learning Representations*.

Brian Thompson and Philipp Koehn. 2019. Vecalign: Improved sentence alignment in linear time and space. 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 1342–1348, Hong Kong, China. Association for Computational Linguistics.

Dániel Varga, Péter Halácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2007. Parallel corpora for medium density languages. *Amsterdam Studies In The Theory And History Of Linguistic Science Series 4*, 292:247.

Shafqat Mumtaz Virk, Harald Hammarström, Markus Forsberg, and Søren Wichmann. 2020. The DReaM corpus: A multilingual annotated corpus of grammars for the world's languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 878–884, Marseille, France. European Language Resources Association.

Eline Visser. 2016. A grammar sketch of kalamang with a focus on phonetics and phonology.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Long Xu, Binghong Wu, TingHao YU, Hao Feng, Zhenyu Huang, Haoqing Jiang, Yunhao Wang, Shuo Huang, and feng zhang. 2026. MORE: A multilingual document parsing benchmark and evaluation. In *Forty-third International Conference on Machine Learning*.

Chen Zhang, Jiuheng Lin, Xiao Liu, Zekai Zhang, and Yansong Feng. 2025. Read it in two steps: Translating extremely low-resource languages with code-augmented grammar books. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3977–3997, Vienna, Austria. Association for Computational Linguistics.

Kexun Zhang, Yee Choi, Zhenqiao Song, Taiqi He, William Yang Wang, and Lei Li. 2024. Hire a linguist!: Learning endangered languages in LLMs with in-context linguistic descriptions. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15654–15669, Bangkok, Thailand. Association for Computational Linguistics.

A Related Work on Bitext Alignment

Bitext alignment algorithms have been popular to create sentence-aligned bitext for training MT systems. They can be broadly divided into bitext mining and sentence alignment.

Bitext mining (Schwenk et al., 2021a,b; NLLB Team et al., 2022) relies on language-agnostic multilingual sentence embedding models, such as LaBSE (Feng et al., 2022), LASER (Artetxe and Schwenk, 2019b; Heffernan et al., 2022), and SONAR (Duquenne et al., 2023), which map sentences of similar meanings to proximate points in a shared, fixed-size embedding space, regardless of the language. In the mining process, documents collected from web are first split into sentences, and language identification (langID) is applied to group sentences by language. Sentence embeddings are then computed for each sentence. Margin-based embedding similarities (Artetxe and Schwenk, 2019a) are computed for each pair of sentences, with the ones with highest similarities are used as bitext. The resulting bitext are often noise and require extensive filtering to improve quality (Koehn et al., 2019; Sloto et al., 2023).

Another line of work, sentence alignment, assumes the availability of parallel documents. A pair of parallel documents contain the same content but differ in language. Sentence alignment methods typically use dynamic programming with scoring functions based on sentence lengths (Brown et al., 1991; Gale and Church, 1993) or translation (Moore, 2002; Varga et al., 2007; Sennrich and Volk, 2010; Gomes and Lopes, 2016). The state-of-the art method Vecalign (Thompson and Koehn, 2019) relies on sentence embeddings.

Neither approach is well suited to extracting bitext from grammar books. Unlike parallel documents, grammar books contain substantial descriptive text surrounding example sentences, making document-level sentence alignment inappropriate. Bitext mining is more applicable, but relies on multilingual sentence representations that may generalize poorly to unseen and extremely low-resource languages (Artetxe and Schwenk, 2019b; Chimoto and Bassett, 2022)

B Script-level OCR Results

We report script-level CER in Figure 3. `dots.ocr`, without the retry strategy, already clearly outperforms `PaddleOCR-VL` and `HunyuanOCR`. However, we observe that decoding with the default temperature of 0.1 sometimes degenerate repetition until the maximum generation length is reached. Therefore, following Poznanski et al. (2025), we retry decoding with progressively higher temperatures of 0.3, 0.5, and 0.8, whenever the generated sequence reaches the maximum token limit. This retry strategy substantially improves recognition performance for the `Hang`, `Hant`, and `Hebr` scripts.

C Prompts

Figure 4 shows the VLM prompt used for bitext extraction. Figure 5 and Figure 6 show the system and user prompts used in the few-shot experiments, respectively.

D Orthography Normalization for Main Results

We summarize the orthography normalization strategies in Table 10. The most salient challenge is the inconsistent usage of diacritics between the archival books and existing datasets. We manually reconcile some of these inconsistencies. However, the normalization is not exhaustive, and mismatches in diacritic usage still remain.

D.1 Normalizing Special Letters for both Example and Test Sentences

Since OCR systems often fail to recognize diacritics and special letters, such as the Combining Low Line character (U+0332), we examine whether these OCR errors substantially affect downstream MT performance. Specifically, we remove diacritics and the Combining Low Line character from both the training and test sets and re-evaluate our few-shot MT experiments for the affected languages: `ddo`, `dik`, `git`, `gug`, `kmb`, `wol`. Our motivation is text will remain intelligible even when diacritics are omitted.

Table 11 compares performance before and after this additional normalization. The results are very similar across the two settings, suggesting that errors involving these characters have only a limited impact on MT performance in our experiments.

E Applying Existing Alignment Methods

Preprocessing Sentence alignment methods assume two documents containing the same content in different languages. Linguistic resources such as grammar books violate this assumption, so we split the OCR text into two language-specific documents. We segment the text into sentences using a sentence splitter,[12] and then apply the langID model from NLLB Team et al. (2022) to each sentence. Because the langID model does not support all low-resource languages, we only trust predictions of high-resource languages. For `mgm-eng`, for example, sentences classified as `eng` are assigned to English, while all other sentences are assigned to `mgm`. We then split the sentences into separate files for each language.

Sentence Mining We consider two variants of sentence mining: *Global Mining*, applied separately to each book, and *Local Mining*, applied separately to each page. For each pair of source and target sentences, we compute the margin score (Eq. 1) and keep the highest-scoring pairs. Our implementation is based on `stopes` (Andrews et al., 2022).

Vecalign We apply Vecalign separately to each page, using the original implementation.[13] For simplicity, we disable many-to-many alignments and allow only 1-1 alignments.

[12] https://github.com/mediacloud/sentence-splitter

[13] https://github.com/thompsonb/vecalign

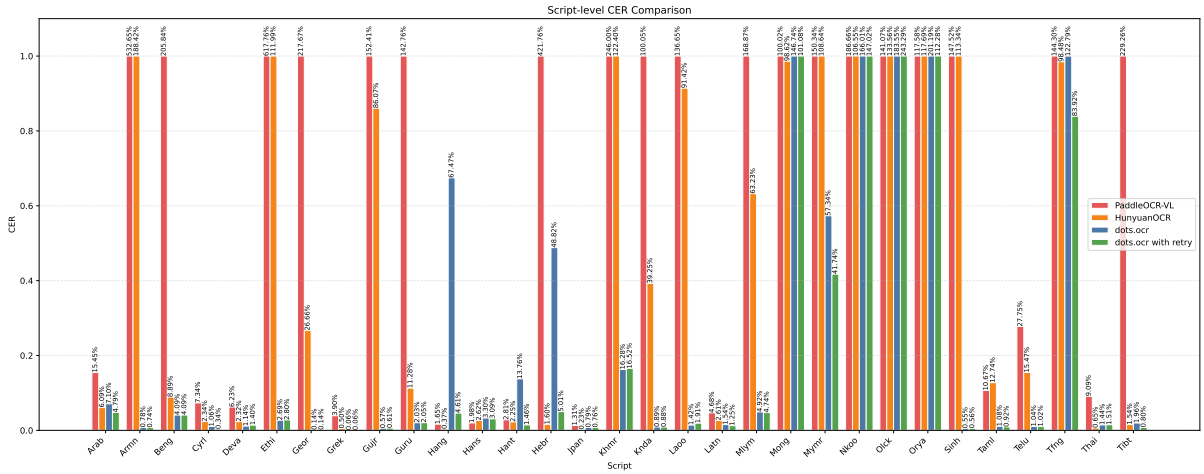


Figure 3: Script-level CER for `dots.ocr`, `HunyuanOCR`, and `PaddleOCR-VL`. CER values exceeding 100% indicate that the corresponding script is not supported by the model.

Hunalign Due to the poor generalization of sentence embeddings, we also consider non-embedding-based Hunalign (Varga et al., 2007), previously used in Merx et al. (2024) for Mambai alignment. Hunalign relies on a dictionary and sentence lengths. We reuse the dictionary entries from Merx et al. (2024), which were extract based on text formatting. But we note that dictionaries are often not readily available in practice. We apply Hunalign separately to each page.

For Sentence Mining and Vecalign, we tune the score thresholds to maximize F1.

F Additional Results

F.1 Intrinsic Evaluation for Unseen Languages

We report BLASER 2.0-QE scores for unsupported languages in Table 12. The quality gap between the extracted examples and the test examples is relatively small. However, because BLASER 2.0-QE may itself be unreliable for these unsupported languages, we treat these scores only as a rough reference and do not use them as a primary evaluation metric.

F.2 Few-shot Experiments using Llama

We provide additional few-shot results on the 20 main languages using a Llama model (`meta-llama/Llama-3.1-8B` (Grattafiori et al., 2024)).

Table 13 reports the results. We observe improvements for most translation directions, with average gains of 7.20 and 2.33 chrF2++ for translation in LRL and HRL, respectively. These results suggest that the extracted examples provide substantial benefits even when used with a considerably smaller, less powerful model.

F.3 Optimize the Number of Bitext Examples

We vary the number of bitext examples from 2 to up to 512 to examine how few-shot MT performance changes with the number of examples. We evaluate `Llama-3.1-8B` and `Qwen3.6-27B` on the `arp-eng` dev set.

As shown in Figure 7, performance of `Llama-3.1-8B` saturates quickly and plateaus at approximately 8 examples, whereas performance of `Qwen3.6-27B` continues to improve. Therefore, we perform 512-shot experiment for `Qwen3.6-27B` and 8-shot for `Llama-3.1-8B`. We do not evaluate more than 512 examples because gains diminish while memory demands and inference time become prohibitive.

F.4 spBLEU Scores for the Main Results

We report spBLEU scores for translation into LRLs and HRLs in Table 14 and Table 15, respectively, for the 20 main languages.

The conclusions are similar to those in the main results. Few-shot MT yields substantial gains compared to zero-shot, with average improvements of 8.96 and 5.07 spBLEU for translation into LRLs and HRLs, respectively. Fine-tuning `nllb` hurts its translation performance on languages it originally supports, but improves translation into HRLs for newly supported languages.

Lang	Books	Existing Data	Normalization
arp	use diacritics (e.g., é, í, ó)	no diacritics	remove diacritics
chv	Latin script	Cyrillic script	script conversion
ddo	noisy OCR (e.g., ā, ç, ġ)	diacritics (e.g., ä, č, š)	correct OCR errors
dgo	IPA	Devanagari script	Codex-proposed conversion rules
dik	diacritics and noisy OCR (e.g., à, ç, ē)	diacritics (e.g., í, ó, ú)	correct OCR errors
git	use diacritics	no diacritics; use U+0332	remove diacritics
gug	diacritics (e.g., ç, ë, ô)	diacritics (e.g., ö, û, ü)	correct OCR errors
ilo	diacritics (e.g., ã, ç, í)	few diacritics	remove diacritics
kac	diacritics (e.g., ă, ė, ū)	few diacritics	remove diacritics
kea	diacritics largely consistent		none
kgv	orthographies consistent		none
kmb	diacritics (e.g., ā, ġ, ň)	fewer diacritics (e.g., ã, ñ, ó)	correct OCR errors
lez	Latin script	Cyrillic script	script conversion
lus	diacritics (e.g., á, ì, ě)	fewer diacritics (e.g., á, é, î)	normalize diacritics
mgm	diacritics (e.g., í, ó, ú)	few diacritics	remove diacritics
min	orthographies consistent		none
mnc	diacritics (e.g., ī, š, ū)	no diacritics	remove diacritics from books
ntu	diacritics (e.g., ä, ë, ö)	no diacritics	normalize diacritics
usp	diacritics largely consistent		none
wol	diacritics (e.g., å, ṇ, ū)	diacritics (e.g., î, ô, ü)	normalize diacritics

Table 10: Orthographic differences between the archival books and the existing dataset, along with the normalization applied to reconcile them. Differences caused by OCR errors are also included.

F.5 Evaluation Results on BOUQuET

We compare 512-shot translation using Qwen3.6-27B against its 0-shot performance on the BOUQuET test set. For the topline setting, we use in-context examples drawn from the BOUQuET dev set.

Table 16 reports chrF2++ scores for translation into LRLs and HRLs. Using parallel sentences from books as in-context examples yields substantial improvements, with average gains of 9.33 and 3.31 chrF2++ for translation into LRLs and HRLs, respectively.

However, a substantial gap remains relative to the topline setting, which is 10.02 chrF2++ for translation into LRLs and 3.34 chrF2++ for translation into HRLs.

F.6 Evaluation Results on Kreyòl-MT

We report chrF2++ scores for Qwen 512-shot translation experiments in Table 17. We observe significant improvements for nearly all translation directions, with the exceptions of eng→bah and eng→gul.

G Details of NLLB fine-tuning

We fine-tune `nllb-200-distilled-1.3B` using LoRA adapters with rank 16 appied to all linear modules. We use a batch size of 32 with 8 gradient accumulation steps and a learning rate of 1×10^{-4}. The learning rate is linearly warmed up over the first 10% training steps.

For each language not originally supported by NLLB, we add a new special language token.

To upsample lower-resource languages, we use temperature-based sampling. The sampling probability for language i is defined as

$$p_i = \frac{N_i^{1/T}}{\sum_j N_j^{1/T}}$$

where N_i denotes the number of training sentence pairs for language i, T is the temperature, and p_i is the probability of sampling a sentence pair from language i at each training step. We set $T = 5$.

Lang	Standard Norm.	Further Norm.
Into-LRL translation		
ddo	24.24	25.86 ↑ 1.62
dik	16.89	19.92 ↑ 3.03
git	23.16	25.14 ↑ 1.98
gug	26.05	24.63 ↓ 1.42
kmb	16.19	16.52 ↑ 0.33
wol	26.93	26.63 ↓ 0.30
Into-HRL translation		
ddo	27.64	26.99 ↓ 0.65
dik	25.39	25.66 ↑ 0.27
git	30.49	29.09 ↓ 1.40
gug	44.06	42.32 ↓ 1.74
kmb	25.42	24.40 ↓ 1.02
wol	39.75	38.31 ↓ 1.44

Table 11: 512 shot chrF2++ scores with Qwen. Standard Norm. applies only the normalization steps in Section 4.3 to the extracted bitext, while Further Norm. additionally removes diacritics from both the extracted and test sentences.

Lang	BLASER 2.0-QE↑	
	Extracted	Test
arp	2.798	2.755
chv	2.804	2.792
ddo	2.721	2.466
dgo	2.851	3.683
git	2.867	2.820
kgv	2.667	2.769
lez	2.702	2.556
mgm	2.941	3.048
mnc	2.558	2.514
ntu	2.406	2.468
usp	2.148	2.458
Avg.	2.678	2.757

Table 12: BLASER 2.0-QE scores for unsupported languages.

We train for 7,500 steps, corresponding to approximately 5 epochs, and use the final checkpoint for evaluation.

H Computation Resources

All experiments are conducted on a single A100-80GB GPU. OCR takes from a few minutes to several hours per book, depending on the number of pages and the script. VLM-based bitext extraction takes approximately 20 minutes per book. MT experiments take between 30 minutes and 36 hours per translation direction, depending on the number of input sentences and bitext examples.

Lang	Into LRL		Into HRL	
	0 shot	8 shot	0 shot	8 shot
Without NLLB support				
arp	5.02	*23.39 ↑18.37	11.88	*16.72 ↑ 4.84
chv	15.56	*18.29 ↑ 2.73	33.42	*33.94 ↑ 0.52
ddo	0.60	*13.46 ↑12.86	16.19	*19.56 ↑ 3.37
dgo	13.08	*16.59 ↑ 3.51	36.84	36.60 ↓ 0.24
git	7.18	*11.47 ↑ 4.29	23.96	25.57 ↑ 1.61
kgv	12.46	*25.53 ↑13.07	14.36	*22.76 ↑ 8.40
lez	4.33	*16.08 ↑11.75	18.87	19.82 ↑ 0.95
mgm	12.51	*23.85 ↑11.34	26.43	*30.19 ↑ 3.76
mnc	0.00	*17.17 ↑17.17	12.81	*15.88 ↑ 3.07
ntu	7.31	*30.53 ↑23.22	15.26	*28.17 ↑12.91
usp	9.57	*17.49 ↑ 7.92	11.73	*17.70 ↑ 5.97
With NLLB support				
dik	9.98	*11.75 ↑ 1.77	22.19	*21.66 ↓ 0.53
gug	16.56	*18.17 ↑ 1.61	30.12	*30.61 ↑ 0.49
ilo	33.26	*34.64 ↑ 1.38	44.25	44.14 ↓ 0.11
kac	6.07	*18.87 ↑12.80	22.30	22.50 ↑ 0.20
kea	24.92	*28.29 ↑ 3.37	51.52	*53.13 ↑ 1.61
kmb	13.09	*12.21 ↓ 0.88	20.20	20.48 ↑ 0.28
lus	25.56	*22.00 ↓ 3.56	31.43	*30.94 ↓ 0.49
min	34.78	34.83 ↑ 0.05	48.30	48.49 ↑ 0.19
wol	14.08	*15.37 ↑ 1.29	24.88	24.81 ↓ 0.07
Avg.	13.30	20.50 ↑ 7.20	25.85	28.18 ↑ 2.33

Table 13: chrF2++ (↑) scores for few-shot MT using Llama-3.1-8B. *p-value < 0.05.

Lang	Qwen		nllb	
	0 shot	512 shot	OOB	fine-tuned
Without NLLB support				
arp	0.14	*16.73 ↑16.59	–	1.00
chv	6.69	*20.85 ↑14.16	–	0.42
ddo	0.00	*11.83 ↑11.83	–	0.08
dgo	4.21	4.10 ↓ 0.11	–	0.24
git	0.07	*6.48 ↑ 6.41	–	0.08
kgv	0.23	*18.59 ↑18.36	–	0.66
lez	0.74	*14.92 ↑14.18	–	0.17
mgm	2.16	*10.81 ↑ 8.65	–	1.98
mnc	0.17	*7.37 ↑ 7.20	–	0.04
ntu	0.36	*34.69 ↑34.33	–	1.10
usp	0.07	*10.28 ↑10.21	–	0.31
Avg.	*1.35*	*14.24* ↑12.89	–	*0.55*
With NLLB support				
dik	0.70	*4.40 ↑ 3.70	3.18	*1.66 ↓ 1.52
gug	1.83	*6.66 ↑ 4.83	17.12	*13.31 ↓ 3.81
ilo	15.19	*19.44 ↑ 4.25	27.03	*9.39 ↓17.64
kac	0.76	*6.72 ↑ 5.96	14.92	*7.64 ↓ 7.28
kea	5.64	*13.64 ↑ 8.00	23.69	*18.09 ↓ 5.60
kmb	1.71	1.60 ↓ 0.11	4.31	*0.53 ↓ 3.78
lus	1.60	*5.23 ↑ 3.63	13.28	*0.85 ↓12.43
min	12.15	*16.50 ↑ 4.35	24.05	*8.66 ↓15.39
wol	4.72	*7.64 ↑ 2.92	6.53	*1.60 ↓ 4.93
Avg.	*4.92*	*9.09* ↑ 4.17	*14.90*	*6.86* ↓ 8.04
Avg.	2.96	11.92 ↑ 8.96	–	3.39

Table 14: spBLEU scores for translation *into LRL*. *p-value < 0.05.

This document may contain parallel sentences in `source_lang` and `target_lang`. Extract all paired `source_lang-target_lang` translations.

Return a JSON list in the following format:
```json
[ " src_code": "<source_lang sentence>", "tgt_code": "<target_lang sentence>", "confidence_reasoning": "Briefly justify why this is a translation and not an exercise"]
```

Extraction rules:
- Extract a pair only when alignment is visually or structurally indicated (e.g., adjacency, parallel columns, keywords like "translation").
- Understand the content first. If sentences appear only as independent exercises or examples without explicit structural cues linking them, do not extract them.
- Sentences appearing in exercises, drills, tests, or prompts (e.g., instructions asking the reader to translate or say the equivalent) are NOT translation pairs and must be ignored.
- Do not hallucinate. If more than one plausible alignment exists for a sentence, do not extract it.
- If numbering is used (e.g., 1., (2), 3)), numbers must match exactly between `source_lang` and `target_lang`. If numbering is inconsistent, incomplete, or not strictly one-to-one, do not extract.
- Paired sentences must preserve punctuation structure (e.g., ?, !, quotes, colon usage). Mismatched punctuation indicates non-parallel text and the pair should be rejected.
- When encountering interlinear glossed text, extract only the original sentence and the translation line, and ignore the morpheme-by-morpheme gloss line.
- Each extracted pair must contain exactly one `source_lang` sentence and exactly one `target_lang` sentence.
- Extract only complete sentences.
- Do not translate, paraphrase, or modify sentences.
- Enforce strict sequential alignment. Sentence pairs must follow the same order in both languages. Do not skip, reorder, or cross-match sentences.
- If no valid parallel sentence pairs are found, return [].
- If an extracted pair is determined invalid, include "INVALID" in "confidence_reasoning".

You are given both the document image and its OCR text. Use both sources jointly. If they conflict, prioritize the OCR text. Do not invent or infer missing text.
OCR text:
`OCR_TEXT`

Figure 4: Instruction template used for bitext extraction.

You are a professional translator. Do not say you do not speak the language. Do not say you do not have enough information. You must provide a translation. Do not include explanations, comments, or extra text. The entire response must be a single line containing only the translation. When provided with example translations, use them as reference to help ensure correct translation.

Figure 5: System prompt template used for few-shot experiments.

Translate the following sentence from `SRC_LANG` to `TGT_LANG`: `SRC_SENT`
Below are some example translations that might be useful:
`SRC_LANG`: `SRC_SENT_1`
`TGT_LANG`: `TGT_SENT_1`

`SRC_LANG`: `SRC_SENT_2`
`TGT_LANG`: `TGT_SENT_2`
...

Now, please provide the translation.
`SRC_LANG`: `TGT_SENT`
`TGT_LANG`:

Figure 6: User prompt template used for few-shot experiments.

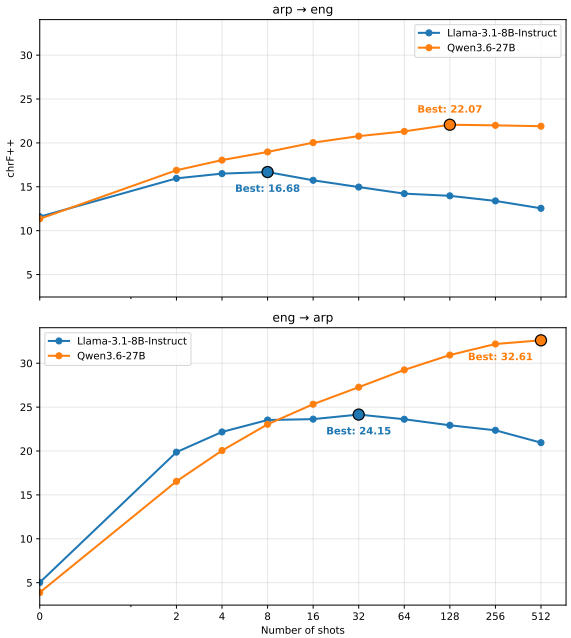
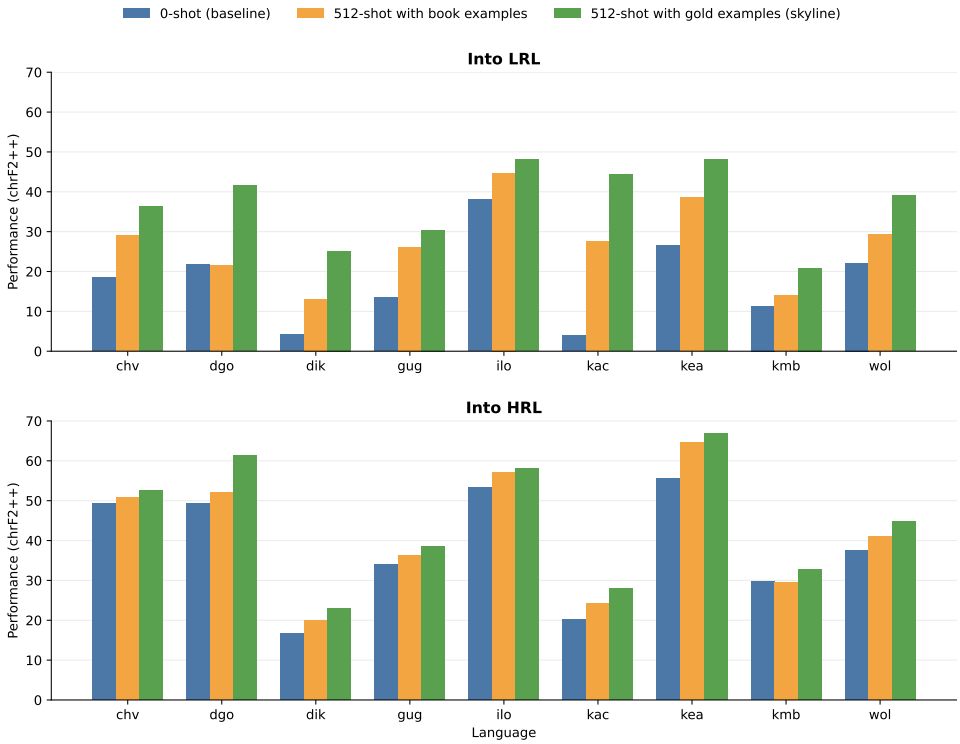


Figure 7: Few-shot MT experiments on `arp` dev set with varying numbers of parallel examples. Performance of `Qwen3.6-27B` continues to improve as more examples are provided, whereas `Llama-3.1-8B-instruct` plateaus quickly.

Lang	Qwen		nllb	
	0 shot	512 shot	OOB	fine-tuned
Without NLLB support†				
arp	2.37	*7.98 ↑ 5.61	0.67	*1.57 ↑ 0.90
chv	25.00	*26.23 ↑ 1.23	2.76	*5.10 ↑ 2.34
ddo	0.31	*8.35 ↑ 8.04	0.20	*1.47 ↑ 1.27
dgo	24.53	*27.76 ↑ 3.23	19.39	*14.09 ↓ 5.30
git	7.78	*10.87 ↑ 3.09	1.40	*6.17 ↑ 4.77
kgv	3.54	*20.58 ↑17.04	1.27	*7.32 ↑ 6.05
lez	6.41	*10.97 ↑ 4.56	0.50	*2.55 ↑ 2.05
mgm	11.00	*24.78 ↑13.78	3.66	*13.29 ↑ 9.63
mnc	2.38	*5.43 ↑ 3.05	0.10	0.56 ↑ 0.46
ntu	2.92	*21.42 ↑18.50	0.58	*18.25 ↑17.67
usp	0.67	*7.05 ↑ 6.38	0.34	*2.21 ↑ 1.87
Avg.	*7.90*	*15.58* ↑ 7.68	*2.81*	*6.60* ↑ 3.79
With NLLB support				
dik	9.50	*10.61 ↑ 1.11	14.14	*11.70 ↓ 2.44
gug	23.13	*24.83 ↑ 1.70	28.41	*24.17 ↓ 4.24
ilo	36.60	*38.64 ↑ 2.04	41.72	*38.66 ↓ 3.06
kac	7.88	*8.51 ↑ 0.63	18.26	*15.27 ↓ 2.99
kea	46.58	*51.98 ↑ 5.40	50.41	*48.20 ↓ 2.21
kmb	9.10	9.33 ↑ 0.23	13.78	*11.06 ↓ 2.72
lus	16.00	*17.01 ↑ 1.01	22.29	*18.75 ↓ 3.54
min	38.20	*41.07 ↑ 2.87	41.17	*35.77 ↓ 5.40
wol	18.79	*20.56 ↑ 1.77	20.92	*16.44 ↓ 4.48
Avg.	*22.86*	*24.73* ↑ 1.87	*27.90*	*24.45* ↓ 3.45
Avg.	14.63	19.70 ↑ 5.07	14.10	14.63 ↑ 0.53

Table 15: spBLEU scores for translation *into HRL*. †Proxy source language codes are used. *p-value < 0.05.



Lang	Into LRL			Into HRL		
	0 shot	512 shot		0 shot	512 shot	
		Book	Gold		Book	Gold
Without NLLB support						
chv	18.43	*29.03 ↑10.60	*36.30 ↑ 7.27	49.41	*50.76 ↑ 1.35	*52.45 ↑ 1.69
dgo	21.79	*21.44 ↓ 0.35	*41.62 ↑20.18	49.34	*52.11 ↑ 2.77	*61.37 ↑ 9.26
With NLLB support						
dik	4.15	*13.07 ↑ 8.92	*25.10 ↑12.03	16.68	*19.98 ↑ 3.30	*23.00 ↑ 3.02
gug	13.43	*26.10 ↑12.67	*30.36 ↑ 4.26	33.90	*36.28 ↑ 2.38	*38.41 ↑ 2.13
ilo	38.05	*44.77 ↑ 6.72	*48.16 ↑ 3.39	53.27	*57.00 ↑ 3.73	*57.98 ↑ 0.98
kac	3.93	*27.50 ↑23.57	*44.33 ↑16.83	20.14	*24.27 ↑ 4.13	*27.87 ↑ 3.60
kea	26.50	*38.61 ↑12.11	*48.12 ↑ 9.51	55.68	*64.61 ↑ 8.93	*66.97 ↑ 2.36
kmb	11.34	*13.96 ↑ 2.62	*20.87 ↑ 6.91	29.83	29.41 ↓ 0.42	*32.79 ↑ 3.38
wol	22.15	*29.28 ↑ 7.13	*39.03 ↑ 9.75	37.48	*41.03 ↑ 3.55	*44.74 ↑ 3.71
Avg.	17.75	27.08 ↑ 9.33	37.10 ↑10.02	38.41	41.72 ↑ 3.31	45.06 ↑ 3.34

Table 16: chrF2++ (↑) results for `Qwen3.6-27B` on the BOUQuET test set. 0-shot performance serves as the baseline for comparison with 512-shot using book examples. The 512-shot setting with gold in-domain examples serves as the topline for comparison with 512-shot prompting using book examples. *p-value < 0.05. On the top is a visualization of the results.

Lang	Into Creole		Into English	
	0 shot	512 shot	0 shot	512 shot
aoa	9.64	*23.35 ↑13.71	10.90	*23.62 ↑12.72
bah	11.77	*7.61 ↓ 4.16	51.31	51.89 ↑ 0.58
brc	10.52	*39.29 ↑28.77	15.23	*44.71 ↑29.48
bzj	25.12	*28.25 ↑ 3.13	38.73	*48.14 ↑ 9.41
bzk	13.10	*36.45 ↑23.35	57.61	*62.41 ↑ 4.80
cri	11.05	*23.27 ↑12.22	9.68	*20.96 ↑11.28
dcr	10.91	*34.42 ↑23.51	28.93	*56.06 ↑27.13
fab	9.51	*22.82 ↑13.31	10.07	*23.67 ↑13.60
fng	12.65	*52.28 ↑39.63	17.76	*31.73 ↑13.97
fpe	17.16	*31.82 ↑14.66	27.71	*57.80 ↑30.09
gcr	22.86	*48.71 ↑25.85	46.71	*54.16 ↑ 7.45
gpe	13.83	*16.02 ↑ 2.19	40.62	*55.44 ↑14.82
gul	38.44	*32.74 ↓ 5.70	67.30	68.10 ↑ 0.80
gyn	15.73	*41.12 ↑25.39	34.94	*45.73 ↑10.79
icr	9.90	*53.45 ↑43.55	30.10	*64.71 ↑34.61
jam	25.09	*41.20 ↑16.11	84.90	*88.82 ↑ 3.92
kri	21.62	*45.67 ↑24.05	38.64	*47.39 ↑ 8.75
ktu	13.11	*50.84 ↑37.73	27.78	*43.98 ↑16.20
mue	10.26	*18.55 ↑ 8.29	16.40	*24.56 ↑ 8.16
pov	18.12	*35.52 ↑17.40	33.06	*49.19 ↑16.13
pre	10.95	*51.38 ↑40.43	14.40	*57.31 ↑42.91
rcf	19.10	*23.99 ↑ 4.89	41.46	*46.42 ↑ 4.96
srm	9.21	*22.37 ↑13.16	13.07	*24.85 ↑11.78
srn	21.22	*37.01 ↑15.79	39.08	* 51.03 ↑11.95
svc	18.43	*44.67 ↑26.24	33.39	*59.44 ↑26.05
tpi	34.90	*49.84 ↑14.94	49.35	*51.66 ↑ 2.31
wes	21.24	*51.50 ↑30.26	53.48	*63.66 ↑10.18
Avg.	16.87	35.71 ↑18.84	34.54	48.79 ↑14.25

Table 17: chrF2++ scores with Qwen on the Kreyòl-MT public test set. *p-value < 0.05.