Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Cross-Lingual Transfer Learning and Autonomous Data Bootstrapping for VLM-Based Ottoman Turkish Handwritten Text Recognition

Gökhan Usta

gusta@itu.edu.tr

Istanbul Technical University

Oğuz Alpoğlu

Fatih Günaydın

Istanbul Medeniyet University

Research Article

Keywords: Ottoman Turkish, Handwritten Text Recognition, Vision-Language Models, Historical Document Analysis, Cross-Lingual Transfer

Posted Date: July 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10418926/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

Historical Ottoman Turkish manuscripts remain severely under-resourced for automated transcription, compounded by cursive right-to-left Arabic-script writing, non-phonetic orthography, and a hybrid Turkish-Arabic-Persian vocabulary. This paper presents an end-to-end Vision-Language Model (VLM) framework for Ottoman handwritten text recognition (HTR), built by further LoRA (Low-Rank Adaptation) fine-tuning an Arabic-English handwriting VLM (itself derived from Qwen2.5-VL-3B-Instruct). We develop two variants: Azra 1, fine-tuned on a mixed corpus of 1,306 human-verified Osmanlica.com lines and 2,000 page-level Council of Higher Education (YÖK) thesis pages containing Latin-to-Arabic reverse-transliterated ground truth; and Azra 2, which utilizes 23,072 lines autonomously bootstrapped exclusively from the YÖK page corpus, with no overlap with Azra 1's curated data, by employing Azra 1 to draft transcriptions that are then difflib-filtered against the page-level reference text, requiring no human annotation. On an in-distribution Riqa test set, Azra 1 attains a Character Error Rate (CER) of 0.208, a 69.2% relative reduction over the base model, outperforming the commercial Osmanlica.com platform. On an out-of-distribution, predominantly Naskh benchmark (OpenITI MAKHZAN), Azra 2's larger, more diverse bootstrapped set instead yields comparable or superior generalization (CER 0.360 vs. 0.357, chrF + + 36.74 vs. 32.38). A character-level error analysis traces systematic confusions to specific stages of the reverse-transliteration pipeline rather than generic recognition noise, offering guidance for improving autonomous data generation in low-resource historical scripts. Our results show that autonomous bootstrapping can approach curated fine-tuning performance while cutting annotation costs, and that cross-lingual transfer from Arabic handwriting models is an effective foundation for Ottoman Turkish HTR.

1. Introduction

The Ottoman Empire's vast archives, spanning several centuries and continents, contain millions of documents that serve as the collective memory of the era. These materials include official records, literary works, periodicals, and personal letters, making their preservation and efficient retrieval of paramount significance to historians and researchers [1, 2]. However, despite the accelerating pace of digitization, the manual transcription of this massive corpus is highly labor-intensive and largely inaccessible to the general public. This accessibility barrier is exacerbated by the fact that the Ottoman script was abandoned in 1928 following the Turkish alphabet reform, resulting in a scarcity of experts capable of reading these historical texts today [3]. Furthermore, the disproportionate focus of modern HTR models on Latin-based languages leaves a significant gap in the transcription of Arabic-script archives, a disparity that deeply constrains historical research and limits the accessibility of post-Ottoman cultural heritage [13].

Automated Optical Character Recognition (OCR) and Handwritten Text Recognition (HTR) systems present a promising solution, yet Ottoman Turkish poses unique computational challenges. The script is inherently cursive, written from right to left, and features letters that change their shapes depending on their position within a word whether initial, medial, final, or isolated [2, 4]. Furthermore, the language is a

complex synthesis of Turkish syntax combined with extensive Arabic and Persian vocabulary, meaning that accurate recognition heavily relies on contextual and linguistic knowledge. The frequent use of diacritical marks, the omission of vowels, and the dense vertical stacking of characters (especially in specific calligraphic styles like Riqa) severely hinder the performance of standard OCR engines [5]. Moreover, the transliteration process is further complicated by the non-phonetic nature of the Ottoman script compared to phonetic modern Turkish. Because vowels are frequently omitted and a single Ottoman character can correspond to multiple Turkish phonemes (e.g., '' mapping to 'v', 'o', 'ö', 'u', or 'ü'), resolving such homographic ambiguity requires complex morphological parsing, contextual analysis, and dictionary-based synthesis [15].

While significant strides have been made in recognizing printed Ottoman texts [2, 6, 7], extending these successes to handwritten documents remains an open problem. Handwriting variations among different scribes, ink degradation, and complex historical orthography contribute to high error rates in traditional models [5]. Consequently, the noise generated by imperfect OCR and HTR systems poses a major bottleneck for downstream Natural Language Processing (NLP) tasks. While modern Turkish NLP has experienced rapid advancements, historical Ottoman texts remain severely under-resourced by comparison. Foundational historical NLP resources such as the OTA-BOUN dependency treebank and the HisTR named entity recognition dataset fundamentally rely on the availability of highly accurate, machine-readable transcriptions [12], making accurate automated transcription all the more critical.

To bridge this gap, this study proposes a novel end-to-end framework leveraging Vision-Language Models (VLMs) to transcribe historical Ottoman Turkish documents. Specifically, we adapt the Qwen2.5-VL-3B-Instruct architecture to process the mixed-script and highly cursive nature of historical documents. Our primary contributions to the literature are summarized as follows:

- We developed two distinct VLM-based models fine-tuned from sherif1313/Arabic-English-handwritten-OCR-v3 (Sherif's model) [20]: Azra 1, fine-tuned on a mixed corpus of curated line-level Riqa samples and page-level reverse-transliterated Latin-script YÖK thesis data; and Azra 2, trained exclusively via a fully autonomous data bootstrapping pipeline that requires no human annotation.
- We investigate cross-lingual transfer learning from Arabic to Ottoman Turkish under a low-resource regime, empirically justifying our choice of an Arabic-English handwriting base model through zero-shot ablation and a character-level error analysis demonstrating that Arabic-script priors transfer effectively but also introduce script-specific orthographic confusions (e.g., /// and /).
- We introduce this autonomous bootstrapping methodology in detail, combining an initial draft model, LLM-based reverse-transliteration, and difflib-based page-line alignment to generate high-fidelity training pairs without direct human verification.
- We benchmark both models on an open-access, mixed-style (Naskh, Ta'liq, Riqa) Ottoman dataset (MAKHZAN) alongside our in-distribution Riqa test set, providing a systematic comparison of in-distribution versus out-of-distribution performance that reveals the limitations of style-specific overfitting and the benefits of diverse training corpora.

- We provide a comprehensive evaluation against the existing commercial Osmanlica.com OCR system and zero-shot vision-language models, situating our results relative to figures reported in the literature for prior Ottoman OCR/HTR architectures.

2. Related Work

In recent years, the digitization of historical texts has spurred a transition from traditional heuristic methods to deep learning architectures. For Arabic-based scripts, Convolutional Neural Networks (CNNs) have been widely employed to capture intricate handwriting strokes, local textures, and stylistic cues.

2.1 Deep Learning in Ottoman OCR and HTR

Subsequent early deep learning approaches for Ottoman document recognition focused heavily on printed texts. Researchers have successfully utilized end-to-end Convolutional Recurrent Neural Networks (CRNNs), combining CNN feature extractors, Long Short-Term Memory (LSTM) units, and Connectionist Temporal Classification (CTC) layers [1, 6]. Systems like the Osmanlica.com hybrid model, trained on datasets of synthetic and original Naskh printed documents, achieved high character recognition accuracies [1, 6], and their architectures have been incorporated into comprehensive end-to-end conversion pipelines mapping documents directly to modern Turkish [2, 8]. We include this Osmanlica.com system as a direct experimental baseline in Section 4. Other recent studies have bypassed multi-stage transliteration entirely, employing Word Beam Search decoders alongside CNN-BiLSTM networks for single-stage Ottoman-to-Turkish transcription [7]. This single-stage approach directly outputs Latin-based modern Turkish from Arabic-script images, circumventing the cascading errors typical of separate OCR and transliteration pipelines. Meanwhile, for pipelines requiring explicit post-OCR transliteration, recent dictionary-based frameworks utilizing traditional NLP methods such as morphological parsing and spelling normalization have proven highly effective, reducing the normalized Word Error Rate (WER) to 6.31% [15].

Addressing handwritten Ottoman documents has required different strategies due to the lack of standardized typefaces. Object detection frameworks, such as YOLO and Faster R-CNN, have been adapted to detect and recognize individual characters in historical manuscripts, offering spatial awareness of character locations [4]. Additionally, the complex variations caused by delayed strokes (e.g., dots and crosses) in cursive scripts have been explored using Hidden Markov Models (HMMs) and Bidirectional LSTMs, highlighting the importance of canonical stroke ordering in sequence recognition tasks [9]. To mitigate the pervasive data scarcity inherent in historical cursive scripts, synthetic data generation and transfer learning have also proven highly effective [14]. Recently, CRNN models specifically trained on the challenging Riqa font have established strong baselines (e.g., 12.77% CER), though they continue to struggle with heavy degradation and overlapping ligatures [5]. To tackle the challenges of low-resourced historical languages, researchers have also explored fine-tuning vision-encoder models, such as TrOCR, on multilingual Arabic-script datasets (including Arabic, Persian, and

Ottoman Turkish), demonstrating that multilingual joint training can effectively mitigate data sparsity [13]. Overcoming these recognition hurdles is increasingly urgent, as the recent introduction of large-scale historical corpora, like the 11-million-token Ottoman Text Corpus (OTC), highlights that robust HTR pipelines are an essential prerequisite for performing advanced computational linguistics and text analysis on Ottoman documents [12].

2.2 Platform and Tool Evaluations

Several studies have evaluated the performance of off-the-shelf and web-based OCR tools on Ottoman texts. Comparisons across platforms reveal that while general-purpose tools (such as ABBYY FineReader, Tesseract, and Google Docs) perform adequately on clean, printed Naskh text, custom-trained deep learning models significantly outperform them [1, 6]. The Riqa script characterized by rapid, cursive strokes and omitted or connected diacritical dots poses a severe challenge to these standard models [5]. To generate robust training data from inaccurate historical transcripts, researchers have also successfully utilized automated alignment techniques via string edit distances and HMMs [10]. Furthermore, platforms like Transkribus have been utilized for Ottoman periodicals, underscoring the difficulties in mapping historical Ottoman orthography to modern Latin-script transcriptions without a one-to-one correspondence [3].

2.3 Vision-Language Models for Historical Documents

The advent of Large Vision-Language Models (VLMs) has introduced new paradigms for document understanding. Recent open-weight models, such as CHURRO (fine-tuned from Qwen 2.5 VL), have demonstrated state-of-the-art performance in page-level historical text recognition across dozens of languages and scripts, outperforming both traditional OCR pipelines and closed-weight commercial VLMs [11]. These architectures bypass complex segmentation steps and leverage extensive pre-trained language representations to handle degraded texts, historical spellings, and complex layouts [11]. Building upon this momentum, our work customizes the Qwen 2.5 VL architecture specifically for the nuances of Ottoman Turkish handwriting.

To efficiently adapt these massive architectures to specific low-resource domains, Parameter-Efficient Fine-Tuning (PEFT) methods like Low-Rank Adaptation (LoRA) have become indispensable. By injecting trainable rank decomposition matrices into the Transformer layers while freezing the pre-trained weights, LoRA significantly reduces the computational burden and memory footprint of fine-tuning without sacrificing model quality, making it highly effective for specialized linguistic adaptations [16]. Furthermore, cross-lingual and cross-script transfer has recently benefited from transliteration and romanization strategies. Utilizing romanized text as an interface or leveraging transliteration in prompting has been shown to enhance the cross-lingual alignment of LLMs, unlocking capabilities for low-resource languages written in non-Latin scripts and mitigating token fertility issues [17]. Our approach builds on this paradigm by employing an LLM (Gemini) for reverse-transliteration, mapping modern Latin-script transcriptions back to their historical Arabic-script representations to synthesize training pairs. Most closely related to our work, CHURRO [11] fine-tunes the Qwen2.5-VL backbone for

historical text recognition across dozens of scripts at the page level. Our approach differs by specializing exclusively in Ottoman Turkish at the line-level granularity via explicit segmentation, enabling a fine-grained, script-specific error analysis of orthographic confusions.

Despite these diverse advancements across HTR architectures and VLM adaptations, a critical gap remains in the literature. First, no prior work has successfully applied an autonomous bootstrapping pipeline that converts page-level training data into line-level training pairs specifically for Ottoman HTR without requiring additional human annotation at the line level, although self-training strategies have been utilized in broader handwritten word recognition contexts [18]. Second, there is a lack of controlled experimental comparisons evaluating the impact of manually curated versus autonomously generated data sources on the fine-tuning of VLMs. Finally, the capacity of these models for cross-style generalization particularly between highly distinct calligraphic traditions such as Riqa and Naskh has not been systematically investigated. This study addresses these gaps by presenting a comprehensive framework for Ottoman document transcription.

3. Proposed Framework and Dataset Construction

This section presents the end-to-end pipeline developed for Ottoman Turkish HTR, addressing three interrelated challenges: (i) selecting and correcting an appropriate base VLM architecture, (ii) curating a high-quality seed dataset from existing Ottoman Turkish resources, and (iii) scaling training data autonomously through a novel reverse-transliteration and bootstrapping methodology. The overall workflow is illustrated in Fig. 1.

3.1. Base Model

The foundation of our framework rests on Qwen2.5-VL-3B-Instruct, a 3-billion parameter vision-language model (VLM) selected for its native multi-script support, efficient inference footprint, and strong zero-shot OCR capabilities across diverse scripts [42]. Unlike conventional encoder–decoder HTR architectures that require script-specific tokenizers and are constrained to fixed-vocabulary character sets, VLMs operate with a unified text generation paradigm that inherently accommodates the mixed-script nature of Ottoman Turkish documents where Arabic, Persian, and Turkish morphological elements co-occur.

We adopt the LoRA-fine-tuned variant published on HuggingFace as sherif1313/Arabic-English-handwritten-OCR-v3 (hereafter referred to as the Base Model, or Sherif's model), which was originally fine-tuned for modern Arabic and English handwriting recognition. This model serves as our starting point because its LoRA adapters already encode fundamental Arabic-script visual features (right-to-left ligatures, diacritical marks, connected letter forms) that overlap substantially with Ottoman Turkish orthography. During initial fine-tuning, we observed output instability. We identified that ensuring weight tying between the input embeddings and output projection layer (tie_word_embeddings=True) was strictly necessary for proper convergence.

The first training dataset, designated Azra 1, was constructed from two complementary sources, both targeting the Riqa calligraphic style prevalent in late Ottoman administrative and literary manuscripts. The primary source consisted of 1,575 line-level image-text pairs downloaded from the Osmanlıca.com platform. This dataset was annotated using the web-based Hokkabaz tool by Ottoman experts over a 2-year period, combining semi-automatic line segmentation with manual polygon correction. From this curated corpus, 157 lines were held out as an in-distribution test set (of which 112 valid lines were used for final evaluation, excluding 45 samples where the baseline commercial system failed to produce any output), with the remaining 1,418 lines used for training.

This curated corpus was supplemented by a substantial volume of additional training data derived from 19 Master's theses archived in Turkey's Council of Higher Education (YÖK) National Thesis Center, comprising transcriptions of Ottoman şer'iyye sicilleri (court registers) and population registers spanning the 17th to early 20th centuries [23–41]. These theses contain students' Latin-alphabet transliterations of original Ottoman Turkish Riqa manuscripts, and to convert these transliterations back into Arabic script, we employed Google's Gemini 3.1 Flash Lite language model via API. Approximately 2,000 pages of Latin-script text drawn from these 19 sources were processed through this pipeline, generating page-level Arabic-script ground truth. The combined Azra 1 training set thus provides a diverse foundation of human-verified line-level annotations and LLM-generated page-level annotations, all within the Riqa calligraphic tradition.

3.2. Autonomous Data Bootstrapping: The Azra 2 Pipeline

The Azra 2 pipeline introduces a fully autonomous data bootstrapping methodology that eliminates human annotation from the training data generation process. This design allows us to test whether a VLM fine-tuned on autonomously bootstrapped data can achieve competitive or even superior performance compared to models trained on manually curated datasets. Full-page manuscript images from the YÖK thesis corpus, comprising more than 2,000 pages of Riqa-style documents, were first processed through the DBNet-based line segmentation model Hedrax/AHDoc [21], which was pre-trained on Arabic and Hebrew historical document layouts and segmented each manuscript page into individual text line bounding boxes, yielding 112,714 candidate line images from the source corpus. Each segmented line image was then passed through Azra 1 model hosted on the vLLM inference engine, which enabled high-throughput batched text generation employing a repetition penalty of 1.15 to mitigate hallucinatory loops and produced a draft Arabic-script transcription for each line. Simultaneously, the corresponding page-level Arabic-script text, served as a reference text against which a dynamic sliding-window alignment algorithm based on Python's difflib.SequenceMatcher was applied.

This algorithm scanned a moving window of up to 40 words, calculating character-level similarity ratios while accounting for varying predicted word counts, tolerating lengths from three fewer to five more words relative to the prediction; a preliminary garbage filter additionally eliminated extremely short outputs of fewer than ten characters and repetitive bureaucratic formulaic phrases before alignment was attempted. To ensure high data quality without human intervention, we enforced a strict similarity

threshold whereby only image–text pairs achieving an alignment ratio of 0.60 or higher against the generated page-level reference text were retained; this rigorous filtering resulted in the rejection of 87,080 candidate lines, an approximately 77% rejection rate, while the pipeline successfully aligned and preserved 25,634 high-confidence line-level image–text pairs.

These 25,634 samples were partitioned into three non-overlapping sets: 23,072 samples for training, 1,281 samples for validation, and 1,281 samples held out as an independent test set, fully accounting for all retained pairs. The base model, sherif1313, was then fine-tuned on this filtered dataset using LoRA, producing the model designated Azra 2. It is important to note that Azra 2 is trained on the bootstrapped data and does not incorporate the Osmanlica.com corpus used in Azra 1. This design choice isolates the training data source curated versus autonomously bootstrapped as the primary experimental variable. While this comparison is intentional, it is not conducted under fully identical conditions: optimal convergence for each dataset naturally required distinct hyperparameter configurations (e.g., LoRA rank 16 vs. 32, learning rate 1e-5 vs. 3e-5, per-device batch size 1 vs. 14), as detailed in Table 1. The observed performance differences should therefore be interpreted as reflecting the combined influence of training data quality and scale, rather than data source alone.

3.3. Independent Cross-Style Benchmark: The OpenITI MAKHZAN Test Set

We constructed an independent out-of-distribution test set from the OpenITI MAKHZAN collection [22] to evaluate the generalization capability of our models beyond the Riqa script style on which they were trained, a corpus maintained by the Open Islamicate Texts Initiative comprising digitized manuscripts that span multiple Islamic calligraphic traditions. Our test set consists of 142 line-level image–text pairs drawn from the MAKHZAN repository, of which approximately 95% are in the Naskh calligraphic style, with the remainder representing other historical Ottoman and Arabic scripts; this composition deliberately introduces a domain shift, as Naskh differs from Riqa in letter proportions, baseline regularity, and the degree of cursive connection between characters, making the MAKHZAN test set a rigorous benchmark for assessing whether models trained on Riqa-style data can generalize to unseen calligraphic traditions. All 142 test samples are accompanied by human-verified Arabic-script ground truth transcriptions from the OpenITI project, ensuring evaluation integrity independent of our own annotation pipeline.

3.4. Training Protocols

In all fine-tuning experiments, Low-Rank Adaptation (LoRA) was applied to the Qwen2.5-VL-3B-Instruct backbone, which comprises 3 billion parameters. LoRA facilitates parameter-efficient adaptation by introducing trainable low-rank decomposition matrices into selected linear layers, while the pre-trained weights remain frozen. Although both Azra 1 and Azra 2 share the same base architecture and incorporate the critical tie_word_embeddings correction detailed in Section 3.1, they differ in their LoRA

configurations, training infrastructure, and dataset characteristics. A comprehensive comparison of the training hyperparameters is provided in Table 1.

Table 1
Complete training hyperparameters for Azra 1 and Azra 2.

Parameter	Azra 1	Azra 2
LoRA Rank (r)	16	32
LoRA Alpha (α)	32	64
Target Modules	q/k/v/o_proj, gate/up/down_proj, qkv, fc1, fc2	q/k/v/o_proj, gate/up/down_proj, qkv, fc1, fc2, proj
Trainable Parameters	39.8M (1.05%)	82.2M (2.14%)
Learning Rate	1×10^{-5}	3×10^{-5}
Gradient Accumulation Steps	16	2
Actual Epochs Completed	5 (1,070 steps)	~ 4 (3,750 steps)
Hardware	NVIDIA RTX PRO 6000	NVIDIA RTX PRO 6000

LoRA adapters were injected into both the language model's multi-head attention layers (q_proj, k_proj, v_proj, o_proj) and feed-forward layers (gate_proj, up_proj, down_proj), as well as the vision encoder's fused QKV attention layer and MLP blocks (fc1, fc2). This hybrid targeting strategy enables cross-modal adaptation, allowing the model to jointly refine its visual feature extraction and text generation pathways. Azra 2 additionally includes the visual-language connector projection layer (proj) in its LoRA targets, increasing its trainable parameter count from 39.8M (1.05%) to 82.2M (2.14%) of the total 3B model parameters. The train/validation split was performed at the document level, assigning complete manuscript pages and their derived line segments exclusively to one partition, so that near-duplicate lines originating from the same page cannot appear in both training and validation sets. Any residual cross-partition similarity was further mitigated by the difflib-based deduplication applied upstream in the bootstrapping pipeline. Training was scheduled for 5 epochs; however, early stopping was triggered at approximately the fourth epoch (3,750 out of 4,120 steps). The validation CER on Azra 2's held-out split exhibited a monotonic decrease from 0.353 at step 125 to 0.179 at step 3,250 before plateauing, confirming successful convergence without severe overfitting.

To optimize learning efficiency, both training configurations utilized a custom data collator designed to mask prompt tokens during the cross-entropy loss computation. Specifically, all tokens preceding the assistant's response including the system prompt, image tokens, and instruction text were masked during the cross-entropy loss calculation, ensuring the model does not expend capacity on reconstructing the prompt. This ensures that the loss is computed exclusively on the ground truth transcription tokens, preventing the model from expending capacity on reconstructing the fixed

instruction template and concentrating the learning signal entirely on the core HTR task. All experiments were conducted on a workstation equipped with NVIDIA RTX 6000 Ada GPUs. Both models were fine-tuned without gradient checkpointing due to sufficient VRAM. All training procedures were executed in bfloat16 mixed-precision, utilizing TF32-enabled matrix multiplications to maximize hardware efficiency.

4. Experimental Results

This section presents results for four systems Base Model (sherif1313), Azra 1 (curated fine-tuning), Azra 2 (bootstrapped fine-tuning), and Osmanlica.com across the 112-sample in-distribution Riqa test set and the 142-sample out-of-distribution MAKHZAN test set.

4.1. In-Distribution Performance on Riqa Script

Table 2 presents the performance of all four systems on the Riqa test set drawn from Osmanlica.com. This evaluation demonstrates the effect of the architectural correction and fine-tuning on performance, and provides the in-distribution baseline against which cross-style generalization will be assessed.

Table 2
In-Distribution Performance on Riqa Script

System	CER (↓)	WER (↓)	chrF++ (↑)
Base Model	0.6768	1.1278	20.95
Azra 1	**0.2083**	0.6689	52.84
Azra 2	0.4104	0.9011	32.02
Osmanlica.com	0.3603	0.6356	46.35

Several key findings emerge from these results. Azra 1 achieves a 69.2% relative CER reduction over the Base Model (0.2083 vs. 0.6768), consistent with the expected effect of the architectural correction. The architectural correction (tie_word_embeddings fix) combined with domain-specific fine-tuning on Ottoman Turkish data transforms the model from near-unusable (67.7% CER) to practically functional (20.8% CER). The chrF++ (Character F-score++) [19] improvement from 20.95 to 52.84 (+ 152.2%) similarly underscores the magnitude of this gain.

4.2. Out-of-Distribution Generalization: Cross-Style Performance

Table 3 presents results on the 142-sample OpenITI MAKHZAN test set, which is composed predominantly of Naskh-style manuscripts, a calligraphic tradition that was not represented in any model's training data.

Table 3
Out-of-Distribution Performance on the OpenITI MAKHZAN Benchmark

System	CER (↓)	WER (↓)	chrF++ (↑)
Base Model	0.3945	0.9208	37.07
Azra 1	0.3574	0.9625	32.38
Azra 2	0.3601	0.9113	36.74
Osmanlica.com	0.5101	0.8425	28.89

This evaluation examines how Azra 1 and Azra 2 generalize to out-of-distribution data, testing whether Azra 2's larger and more diverse bootstrapped training set confers greater robustness than Azra 1's curated but narrower training corpus.

Table 4
Cross-Style Performance Summary Riqa vs. MAKHZAN

System	Riqa CER	Riqa WER	Riqa chrF++	MAKHZAN CER	MAKHZAN WER	MAKHZAN chrF++
Base Model	0.6768	1.1278	20.95	0.3945	0.9208	37.07
Azra 1	0.2083	0.6689	52.84	0.3574	0.9625	32.38
Azra 2	0.4104	0.9011	32.02	0.3601	0.9113	36.74
Osmanlica.com	0.3603	0.6356	46.35	0.5101	0.8425	28.89

4.3. Error Analysis and Linguistic Biases

A fine-grained error analysis reveals systematic character-level biases that trace directly to the data generation methodology, particularly the Gemini reverse-transliteration stage. We performed a full character-level Levenshtein alignment between each system's output and the ground truth on both test sets, tabulating the exact frequency of eight targeted substitutions involving the three visually similar *ya* variants (ﻯ ,ﻱ ,ﻲ) and the Kef/Sağır Kef pair (ﻙ/ﻚ). Figure 5 reports these counts for the Riqa and MAKHZAN test sets, respectively, across all four systems: the Base Model (sherif1313), Azra 1, Azra 2, and the Osmanlica.com commercial platform.

5. Discussion

5.1. The Impact of Autonomous Bootstrapping vs. Curated Training Data

The Base Model exhibits catastrophic failure modes on Riqa: 6 of 112 samples are statistical outliers (CER > 0.7337), with the worst cases producing entirely Latin-script output (CER = 13.85) or repetitive

hallucinated text a fallback to its English-language training distribution when confronted with unfamiliar Ottoman visual patterns. Azra 2 underperforms Azra 1 in-distribution, which is expected: Azra 2's training data was generated entirely through autonomous bootstrapping, with no exposure to the Osmanlica.com corpus the test set is drawn from.

The comparison with Osmanlica.com highlights a robustness gap orthogonal to CER: of 157 submitted test images, 45 (28.7%) returned empty outputs, since its detection-then-recognition pipeline fails entirely when the detector misses text. All three VLM-based systems, by contrast, produced output for every image, processing it holistically rather than through a separate detection stage. Restricting comparison to Osmanlica.com's 112 non-empty outputs, its CER (0.3603) is still well behind Azra 1 (0.2083, a 42.2% relative gap) though it achieves the best WER of any system (0.6356), suggesting dictionary-based post-processing or re-scoring that corrects whole words without improving character-level precision.

5.2. Limits of Cross-Script Generalization

The ranking established on Riqa does not hold on MAKHZAN. Azra 1's CER rises from 0.2083 to 0.3574 (a 71.6% relative degradation), and it posts the worst WER (0.9625) and lowest chrF++ (32.38) of the three VLMs consistent with its Riqa performance reflecting domain overfitting rather than general HTR competence. Azra 2 generalizes markedly better: a comparable CER to Azra 1 (0.3601 vs. 0.3574) but a better WER (0.9113 vs. 0.9625) and chrF++ (36.74 vs. 32.38), suggesting its larger, more heterogeneous bootstrapped corpus has a regularizing effect that limits the style-specific overfitting seen in Azra 1.

The Base Model is unexpectedly competitive here (CER 0.3945, within ~ 10% of both fine-tuned models), implying its diverse Arabic-handwriting pre-training offers a style-agnostic foundation that partially compensates outside Riqa though it remains unstable, with the most outliers (23 samples, CER > 0.7363) and the single worst case (CER = 1.875) of any system. Osmanlica.com fares worst overall (CER 0.5101, chrF + + 28.89), reflecting its narrow training focus on Ottoman administrative Riqa documents. Table 4 makes the broader point explicit: CER convergence across Base Model, Azra 1, and Azra 2 into a narrow band (0.357–0.395) on MAKHZAN, combined with Azra 2 beating Azra 1 on WER and chrF + + despite trailing it substantially on Riqa, shows that in-distribution ranking is not a reliable predictor of out-of-distribution ranking particularly for word- and character-n-gram-level metrics.

5.3. Tracing Linguistic Biases to the Reverse-Transliteration Pipeline

These results indicate the primary bottleneck in autonomous data generation is not data volume but the linguistic fidelity of the reverse-transliteration step, not generic HTR noise.

Ya-variant confusions. The three visually similar ya forms Ottoman/Persian ﯼ, standard Arabic ي, and alif maqsura ى differ only in sub-linear dot placement. On Riqa, Gemini's reverse-transliteration favors standard Arabic ي over Ottoman ﯼ (its training data being predominantly modern Arabic), propagating into Azra 2's predictions (153 :ﯼ→ي ;361 :ﯼ→ى). The Base Model shows the same directional bias at

comparable magnitude (228, 98), indicating the tendency predates Gemini and is already present in sherif1313's Arabic-handwriting pretraining Azra 2's fine-tuning reinforces rather than creates it. On MAKHZAN, the pattern reverses: Azra 1, trained partly on Osmanlica.com's Ottoman-□ convention, substitutes (258) □→□ and (130) □→□; Osmanlica.com shows the identical mirrored pattern at comparable magnitude (158, 75), confirming both systems share a convention that fails to transfer to the Naskh-dominant corpus. Azra 2 shows no residual bias on the main ya-variant substitutions (0 for both □→□ and □→□) on MAKHZAN, and has a small distinct confusion between the two non-Ottoman variants (27 :□→□) alongside a □→□ substitution count of 15, reflecting its lack of exposure to the Ottoman convention.

Kef/Sağır Kef. Ottoman's archaic □ has no modern Arabic/Persian equivalent; Gemini typically maps Latin "ng"/"ñ" to standard □ instead. This produces a clean split: □→□ over-correction appears almost exclusively in Azra 2 (39 Riqa, 15 MAKHZAN) and is negligible elsewhere, while □→□ under-production appears in Azra 1 (13), the Base Model (7), and Osmanlica.com (11) but is essentially absent from Azra 2 consistent with the bootstrapping pipeline correcting toward historical orthography where the other systems do not.

Script-level failures. Independent of these character-variant confusions, the Base Model shows broader script-level instability: frequent Latin-character insertions on Riqa (a: 314, n: 173, e: 169, r: 124, f: 113) and Arabic-diacritic over-generation on MAKHZAN (fatha: 119, sukun: 42, kasra: 39, damma: 37), reflecting its lack of domain calibration.

Together, the convergence of Azra 1 and Osmanlica.com on an identical mirrored ya-pattern, and the isolation of the □↔□ over-correction to Azra 2, confirm that these confusions trace to specific pipeline stages rather than generic recognition noise. Improving the LLM's awareness of Ottoman-specific conventions particularly the □/□/□ and □/□ distinctions would likely yield disproportionate downstream gains. Rule-based find-and-replace correction is not feasible given the mixed Arabic/Persian/Ottoman spelling conventions coexisting within the same text, so we opted not to enforce full standardization in the autonomous pipeline.

5.4. Limitations

Several limitations should be noted. First, because Azra 2's ground truth derives entirely from Gemini's reverse-transliteration, any systematic errors introduced at that stage propagate directly into the bootstrapped training corpus. Second, the pipeline depends on a proprietary LLM (Gemini 3.1 Flash Lite), introducing a recurring API cost and a dependency on a system whose behavior may change over time. Third, this study evaluated CRNN+BiLSTM + CTC-based (Osmanlica.com) and VLM-based architectures only; OCR-specific, layout-aware VLMs (e.g., Document AI) should be tested in future work. Fourth, our evaluation spans only two calligraphic traditions (Riqa and Naskh) over comparatively modest test sets (112 and 142 lines, respectively); broader coverage across additional Ottoman scripts and larger benchmarks would strengthen the generality of our generalization claims. Finally, because Azra 1 and Azra 2 required distinct hyperparameter configurations to reach convergence on their respective

datasets (Table 1), the observed performance gap cannot be attributed to data source alone, and a fully controlled ablation isolating data scale, diversity, and hyperparameters independently remains an open direction for future work.

6. Conclusion and Future Work

This paper presented an end-to-end VLM framework for Ottoman Turkish handwritten text recognition, comparing curated and autonomously bootstrapped fine-tuning strategies. The results show a clear trade-off. Azra 1, trained on curated data, achieves the best in-distribution performance but generalizes poorly to an unseen calligraphic style (Naskh). Azra 2, trained entirely on autonomously bootstrapped data, performs worse in-distribution but generalizes more consistently, suggesting that larger and more heterogeneous training data improves robustness to domain shift. Both models substantially outperform the base model; relative to the commercial Osmanlica.com system, Azra 1 outperforms it on both benchmarks, while Azra 2 outperforms it on the out-of-distribution MAKHZAN set but trails it on the in-distribution Riqa set. The error analysis further shows that most remaining errors are not random recognition noise, but are concentrated in a small number of orthographic confusions (ya-variants and kef/sağır kef) that originate in the reverse-transliteration stage of the pipeline. This indicates that these errors can likely be reduced by improving that specific stage, without requiring additional labeled data or retraining the recognition model itself.

Future work should extend this evaluation to additional Ottoman calligraphic styles, such as Ta'lîq, Dîvânî, and Siyakat, and should directly address the identified orthographic confusions, for example through improved transliteration prompts or a post-hoc normalization step. More broadly, the autonomous bootstrapping approach used here is not specific to Ottoman Turkish. It may be applicable to other historical languages written in scripts that have diverged from their modern forms, particularly in cases where expert annotators are scarce relative to the amount of available archival material.

Declarations

Author Contribution

• Gökhan Usta: Conceptualization, Methodology, Software, Data Curation, Formal Analysis, Writing – Original Draft.• Oğuz Alpoğlu: Validation, Data Curation, Writing – Review & Editing.• Fatih Günaydın: Investigation, Data Curation, Writing – Review & Editing.

Acknowledgement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability

The datasets generated and analyzed during the current study, along with the source code for the models, are openly available in the project's GitHub repository: https://github.com/gusta-bib/ottoman-htr

References

1. Dölek, İ., Kurt, A.: A deep learning model for Ottoman OCR. Concurrency Comput. Pract. Exp. **34**, e6937 (2022). https://doi.org/10.1002/cpe.6937
2. Dölek, İ., Kurt, A.: Osmanlıcadan Türkçeye uçtan uca aktarım. J Smart Syst Res 3:1–10. (2022). Available: https://izlik.org/JA43PT44UY
3. Kirmizialtin, S., Wrisley, D.J.: Automated transcription of non-Latin script periodicals: a case study in the Ottoman Turkish print archive. DHQ 16(2). (2022). http://www.digitalhumanities.org/dhq/vol/16/2/000577/000577.html
4. Demir, A.A., Özkaya, U.: An object detection-based character recognition method for Ottoman handwritten documents. Int. J. Doc. Anal. Recognit. **29**, 111–128 (2026). https://doi.org/10.1007/s10032-025-00529-7
5. Dölek, İ., Kurt, A.: Ottoman HTR: recognition of the Ottoman Riqa font using deep learning models. In: 2025 Innovations in Intelligent Systems and Applications Conference (ASYU), pp 1–6 (2025)
6. Dölek, İ., Kurt, A.: Ottoman OCR: printed Naskh font. In: 2021 International Conference on Innovations in Intelligent Systems and Applications (INISTA), Kocaeli, Turkey, pp 1–5. (2021). https://doi.org/10.1109/INISTA52262.2021.9548616
7. Tasdemir, E.F.B., Tandoğan, Z., Akansu, S.D., Kızılırmak, F., Sen, M.U., Akcan, A., Kuru, M., Yanikoglu, B.: Automatic transcription of Ottoman documents using deep learning. In: Document Analysis Systems: 16th IAPR International Workshop, DAS 2024, Athens, Greece, pp 422–435. (2024). https://doi.org/10.1007/978-3-031-70442-0_26
8. Uzan, S., Dölek, İ.: Osmanlıca-Türkçe uçtan uca aktarım sistemi-Osmanlica.com projesinin Türk dili ve edebiyatı alanına katkıları. Fırat Üniversitesi Sosyal Bilimler Dergisi. **35**, 907–924 (2025). https://doi.org/10.18069/firatsbed.1570398
9. Tasdemir, E.F.B., Yanikoglu, B.: A comparative study of delayed stroke handling approaches in online handwriting. Int. J. Doc. Anal. Recognit. **22**, 15–28 (2019). https://doi.org/10.1007/s10032-018-0313-2
10. Romero-Gómez, V., Toselli, A.H., Bosch, V., Sánchez, J.A., Vidal, E.: Automatic alignment of handwritten images and transcripts for training handwritten text recognition systems. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), Vienna, Austria, pp 328–333. (2018). https://doi.org/10.1109/DAS.2018.41
11. Semnani, S.J., Zhang, H., He, X., Tekgurler, M., Lam, M.: CHURRO: making history readable with an open-weight large vision-language model for high-accuracy, low-cost historical text recognition. In:

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Suzhou, China, pp 34777–34824. (2025). https://doi.org/10.18653/v1/2025.emnlp-main.1763

12. Özateş, Ş.B., Tıraş, T.E., Adak, E.E., Doğan, B., Karagöz, F.B., Genç, E.E., Bilgin Taşdemir EF: Building foundations for natural language processing of historical Turkish: resources and models. arXiv:2501.04828 (2026)
13. Broadwell, P., Patel, U., Tekgürler, M.: Multilingual handwritten text recognition (HTR) models for large-scale processing of archival documents in low-resourced Arabic-script languages. SSRN. (2025). https://doi.org/10.2139/ssrn.5190984
14. Ahmadian, A., Fouladi, K., Araabi, B.N.: Model-based Persian calligraphy synthesis via learning to transfer templates to personal styles. Int. J. Doc. Anal. Recognit. **23**, 183–203 (2020). https://doi.org/10.1007/s10032-020-00353-1
15. Dölek, İ., Kurt, A.: The Ottoman-Turkish transliteration using traditional NLP techniques. Research Square (preprint). (2025). https://doi.org/10.21203/rs.3.rs-5735281/v1
16. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Chen, W.: LoRA: low-rank adaptation of large language models. arXiv:2106.09685 (2021)
17. Dabre, J.J., Gala, R.M.A., Jayakumar, J., Puduppully, T., Kunchukuttan, R.: A RomanSetu: efficiently unlocking multilingual capabilities of large language models via romanization. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand, pp 15593–15615. (2024). https://doi.org/10.18653/v1/2024.acl-long.833
18. Wolf, F., Fink, G.A.: Self-training for handwritten word recognition and retrieval. Int. J. Doc. Anal. Recognit. **27**, 225–244 (2024). https://doi.org/10.1007/s10032-024-00484-9
19. Popovic, M.: chrF++: words helping character n-grams. In: Proceedings of the Second Conference on Machine Translation, Copenhagen, Denmark, pp 612–618. (2017). https://doi.org/10.18653/v1/W17-4770
20. sherif1313: Arabic-English-handwritten-OCR-v3. HuggingFace Model Repository. https://huggingface.co/sherif1313/Arabic-English-handwritten-OCR-v3. (2025). Accessed 2025
21. Waly, A., Tarek, B., Feteha, A., Yehia, R., Amr, G., Fares, A.: Arabic Handwritten Document OCR Solution with Binarization and Adaptive Scale Fusion Detection. In: 2024 6th Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, pp 316–319. (2024). https://doi.org/10.1109/NILES63360.2024.10753216
22. Romanov, M., et al.: OpenITI: a machine-readable corpus of Islamicate texts. (2019). https://openiti.org/
23. Akkaya, T.: 18/1 numaralı Develi şer'iyye sicili (H. 1314/M. 1896–1897) transkripsiyon ve değerlendirme [Transcription and evaluation of Develi court register no. 18/1 (AH 1314/AD 1896–1897)]. Dissertation, Erciyes University, Kayseri, Turkey (2015)
24. Güneş, N.: 11 numaralı Develi şer'iyye sicili (H. 1320-1/M. 1903-4) transkripsiyon ve değerlendirme [Transcription and evaluation of Develi court register no. 11 (AH 1320-21/AD 1903–1904)]. Dissertation, Erciyes University, Kayseri, Turkey (2013)

25. Arık Kaygısız, A.: I numaralı Develi şer'iyye sicili (H. 1311/M. 1893 - H. 1313/M. 1895) transkripsiyon ve değerlendirme [Transcription and evaluation of Develi court register no. I (AH 1311–1313/AD 1893–1895)]. Dissertation, Erciyes University, Kayseri, Turkey (2006)
26. Atalay, Ç.: 24 numaralı Kırşehir şer'iyye sicili (H.1328–1332)(M.1910–1914)(R.1326–1329) transkripsiyon ve değerlendirme [Transcription and evaluation of Kırşehir court register no. 24 (AH 1328–1332/AD 1910–1914)]. Dissertation, Erciyes University, Kayseri, Turkey (2007)
27. Çelik, F.: (H. 1143–1145)(M. 1730–1733) tarihli Çankırı şer'iyye sicilinin transkripsiyon ve değerlendirmesi [Transcription and evaluation of the Çankırı court register dated AH 1143–1145/AD 1730–1733]. Dissertation, Niğde Ömer Halisdemir University, Niğde, Turkey (2019)
28. Salep, M.: 9/1 numaralı Develi şer'iyye sicili'nin (H. 1317–1318/M. 1899–1901) transkripsiyon ve değerlendirilmesi [Transcription and evaluation of Develi court register no. 9/1 (AH 1317–1318/AD 1899–1901)]. Dissertation, Erciyes University, Kayseri, Turkey (2008)
29. Kalkan, G.: Oltu sancağı müslim nüfus defterinin transkripsiyon ve değerlendirmesi [H.1251(M.1835/1836)] [Transcription and evaluation of the Oltu sanjak Muslim population register (AH 1251/AD 1835–1836)]. Dissertation, Amasya University, Amasya, Turkey (2022)
30. Barış, G.: 2 no'lu (H. 1299–1304/M. 1883–1888) Yozgat şer'iyye sicilinin transkripsiyonu ve değerlendirilmesi [Transcription and evaluation of Yozgat court register no. 2 (AH 1299–1304/AD 1883–1888)]. Dissertation, Erciyes University, Kayseri, Turkey (2007)
31. Öztürk, M.: 216/2 numaralı Urfa şer'iyye sicili'nin transkripsiyon ve değerlendirmesi (H. 1315–1317/M. 1897–1899) [Transcription and evaluation of Urfa court register no. 216/2 (AH 1315–1317/AD 1897–1899)]. Dissertation, Harran University, Şanlıurfa, Turkey (2020)
32. Bahadır, H.: 5 no'lu Amasya şer'iyye sicili'nin transkripsiyonu ve değerlendirilmesi [Transcription and evaluation of Amasya court register no. 5]. Dissertation, Amasya University, Amasya, Turkey (2019)
33. Çora, E.O.: 22 numaralı Kırşehir şer'iyye sicili'nin (H.1327–1328/M. 1909–1910) transkripsiyonu ve değerlendirilmesi [Transcription and evaluation of Kırşehir court register no. 22 (AH 1327–1328/AD 1909–1910)]. Dissertation, Bitlis Eren University, Bitlis, Turkey (2023)
34. Zeytinci, M.: Manisa kazasının 167 numaralı şer'iye sicili'nin transkripsiyon ve değerlendirilmesi (H. 1111–1113/M. 1700–1702) [Transcription and evaluation of Manisa district court register no. 167 (AH 1111–1113/AD 1700–1702)]. Dissertation, Ege University, İzmir, Turkey (2024)
35. Sağlam Öztürk, Ş.: 67 numaralı Ayıntab şer'iyye sicili'nin transkripsiyonu ve değerlendirmesi (H. 1128–1129/M. 1716–1717 S. 429–603) [Transcription and evaluation of Ayntab court register no. 67 (AH 1128–1129/AD 1716–1717, pp. 429–603)]. Dissertation, Gaziantep University, Gaziantep, Turkey (2020)
36. Karagöz, S.: 1 no'lu Ayıntab şer'îyye sicili defteri transkripsiyonu [Transcription of Ayntab court register no. 1]. Dissertation, Muğla University, Muğla, Turkey (2012)
37. Uçan, Y.E., 1342/M: 46 numaralı Develi şer'iyye sicili (H. 1332/M. 1914 - H. 1924) transkripsiyon ve değerlendirme [Transcription and evaluation of Develi court register no. 46 (AH 1332–1342/AD 1914–1924)]. Dissertation, Erciyes University, Kayseri, Turkey (2006)

38. Yıldız, Ş.: 11 numaralı Sivas şer'iyye sicil defteri (değerlendirme-metin) [Sivas court register no. 11 (evaluation and text)]. Dissertation, Sivas Cumhuriyet University, Sivas, Turkey (2025)
39. Kaya, E.: 38 numaralı Kütahya şer'iyye sicili'nin transkripsiyonu ve değerlendirmesi [Transcription and evaluation of Kütahya court register no. 38]. Dissertation, Kütahya Dumlupınar University, Kütahya, Turkey (2024)
40. Nefes, H.: 1443 numaralı Giresun şer'iyye sicil defterinin transkripsiyonu ve değerlendirilmesi [Transcription and evaluation of Giresun court register no. 1443]. Dissertation, Ondokuz Mayıs University, Samsun, Turkey (2023)
41. Bekteş, Y.: 1420 numaralı Giresun şer'iyye sicil defterinin transkripsiyonu ve değerlendirilmesi [Transcription and evaluation of Giresun court register no. 1420]. Dissertation, Ondokuz Mayıs University, Samsun, Turkey (2018)
42. Yang, A., Yang, B., Zhang, B., et al.: Qwen2.5 technical report. arXiv:2412.15115 (2025)

Figures

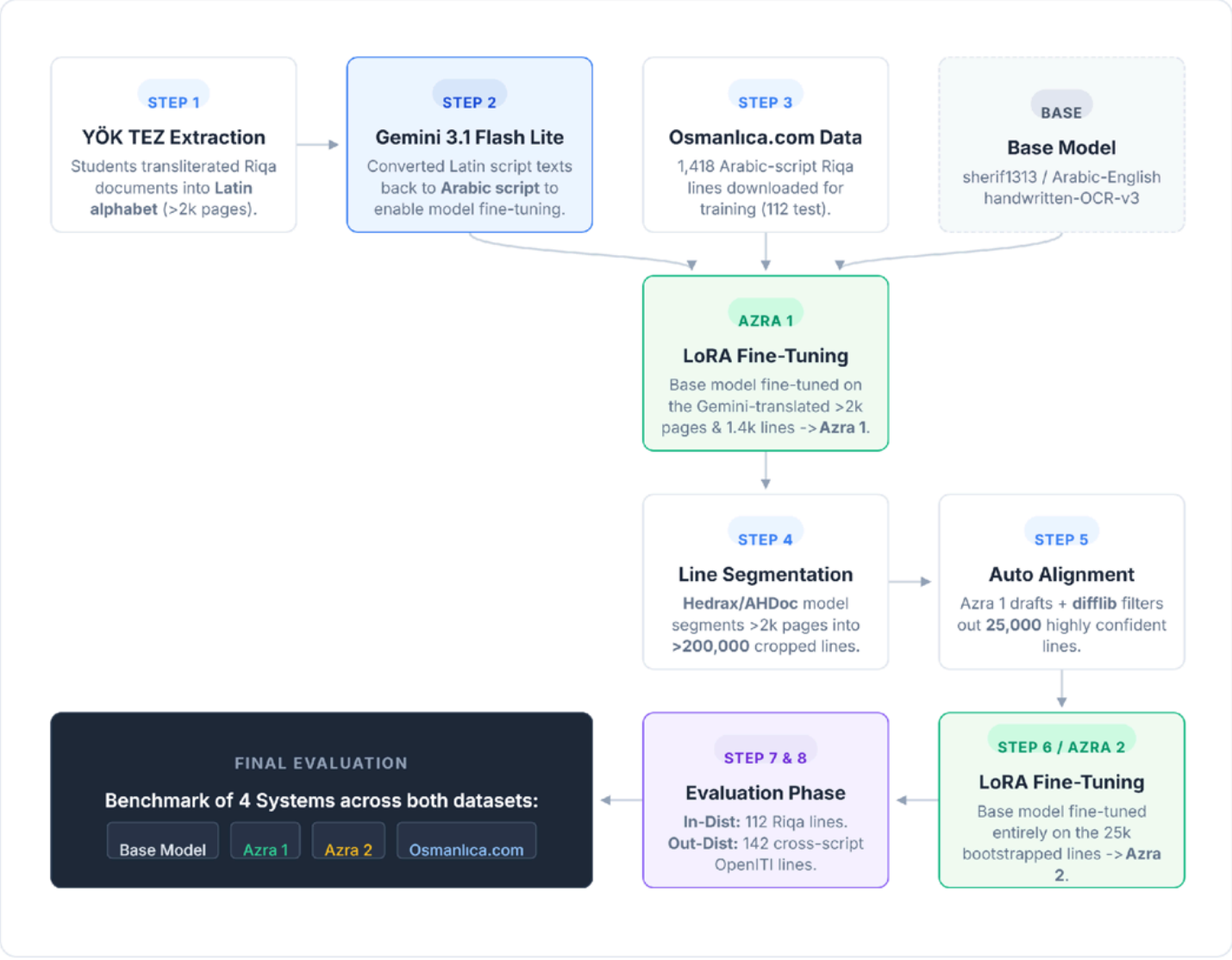


Figure 1

End-to-end pipeline for autonomous Ottoman Turkish HTR dataset construction, model fine-tuning (Azra 1, Azra 2), and evaluation.

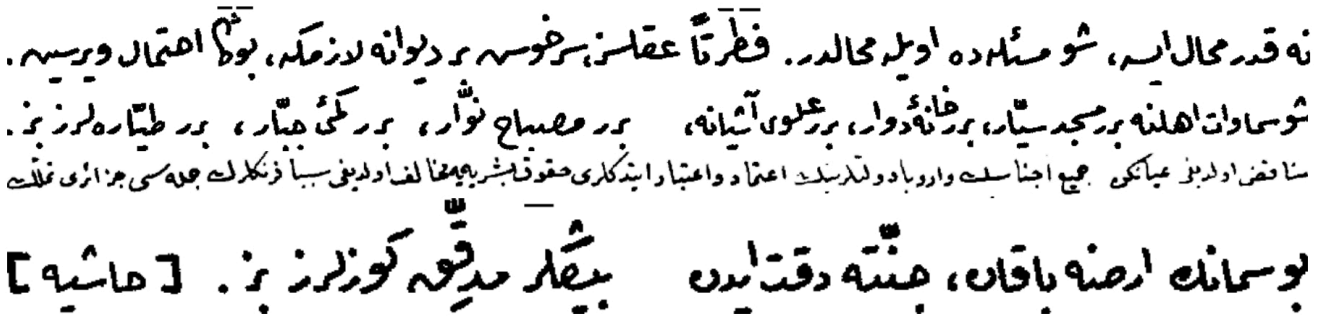


Figure 2

Representative line-level image samples from the Osmanlica.com Riqa-style dataset used in Azra 1 training.

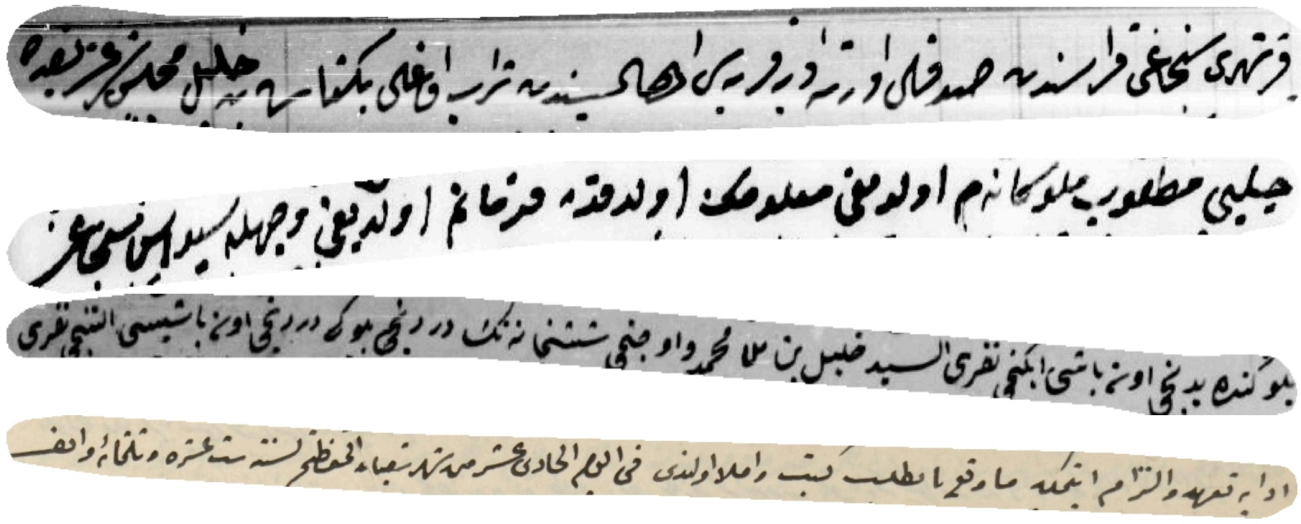


Figure 3

Representative line-level image samples from the autonomously bootstrapped Azra 2 training corpus, generated via the YOLO segmentation and difflib alignment pipeline.



Figure 4

Representative line-level image samples from the OpenITI MAKHZAN benchmark test set, comprising Naskh and other calligraphic styles not seen during training.

(a) Riqa Test Set

Substitution	Azra 1	Azra 2	Base Model	Osmanlıca.com
ي ⟶ ى *Ottoman → Arabic ya*	13	361	228	—
ى ⟶ ى *Ottoman → Alif Maqsura*	—	153	98	—
ڤ ⟶ ك *Kaf → Sağır Kef*	—	39	—	—
ك ⟶ ڤ *Sağır Kef → Kaf*	13	—	—	11

(b) OpenITI MAKHZAN Test Set

Substitution	Azra 1	Azra 2	Base Model	Osmanlıca.com
ى ⟶ ي *Arabic → Ottoman ya*	258	—	—	158
ى ⟶ ى *Alif Maqsura → Ottoman*	130	—	—	75
ى ⟶ ي *Arabic → Alif Maqsura*	—	27	—	—
ي ⟶ ى *Alif Maqsura → Arabic*	—	—	23	—
ڤ ⟶ ك *Kaf → Sağır Kef*	—	15	—	—

Figure 5

Character-Level Error Analysis of Model Substitution Patterns by Test Set