

Breaking the Script Barrier: Automatic De-Romanization of Maghribi Arabizi to Arabic Script in Social Media

Abdessamad Nafissi

Independent Researcher, Tampa, FL, USA
abdessamadnfs@gmail.com

Abstract. Maghribi social-media users frequently write dialectal Arabic in Arabizi, a non-standard Romanized script that mixes Latin letters and numerals. This paper presents an empirical benchmark for converting Moroccan Arabizi to Arabic script; it does not propose a new model architecture. The benchmark uses model-generated Arabizi paired with its Arabic-script source material, rather than naturally authored Arabizi. Four systems are compared on a frozen 805-passage test split: an ordered rule mapper, a maximum likelihood estimation (MLE) word mapper with regex fallback, a character-level word long short-term memory (LSTM) encoder-decoder, and a pretrained full-passage mT5-small Transformer. The Transformer obtains the highest SacreBLEU (36.75) and position-wise token agreement (21.54%), while MLE obtains the strongest chrF (69.60), character error rate (15.17%), and word error rate (42.48%). Thus, the modern baseline does not dominate all measures: 78 Transformer outputs do not emit an end-of-sequence token within the configured limit and sometimes repeat text. Two native-speaker audits examine 300 test references and 100 sampled training alignments; a separate review of 50 LSTM errors distinguishes genuine errors from acceptable variants. The results show that lexical coverage remains highly competitive, that evaluation conclusions depend on the metric, and that alignment noise and valid dialectal variation require explicit human analysis.

Keywords: Maghribi Arabizi · De-Romanization · Dialectal Arabic
· Transliteration · mT5 · Human Evaluation

1 Introduction

Social media has accelerated the use of dialectal Arabic online. In the Maghreb, users write Moroccan, Algerian, and Tunisian varieties in Arabizi, an informal Romanized script that uses Latin letters and numerals to approximate Arabic phonology, such as 7 for ح and 3 for ع . This practice supports informal communication but creates a barrier for Arabic natural language processing (NLP) systems that expect native Arabic script [2]. The experiments in this paper evaluate Moroccan Darija only; they do not establish performance for Algerian or Tunisian varieties.

The task studied here is *de-romanization*: converting an Arabizi string into Arabic script while preserving its dialectal lexical content. It differs from romanization, which maps native script to Latin script; from orthographic normalization, which standardizes variants within a script; and from translation into Modern Standard Arabic (MSA), which may change lexical and grammatical form. The distinction matters because multiple Arabic-script spellings may be acceptable for the same Darija input and because full-passage transliteration may benefit from context [8, 9].

The contribution is an audited experimental study rather than a novel algorithm. It (i) compares transparent rule and lexical baselines with both an LSTM and a contemporary pretrained Transformer; (ii) reports SacreBLEU, chrF, character error rate (CER), word error rate (WER), n-gram precision, and a legacy position-wise token measure; (iii) quantifies MLE out-of-vocabulary (OOV) behavior and the regex fallback; and (iv) adds independent human checks of test-reference validity and training-pair alignment, plus a qualitative review of LSTM errors. This scope avoids treating one word-level LSTM as representative of neural de-romanization in general.

2 Dataset and Quality Control

2.1 Corpus Source and Splits

The parallel data come from the UBC-NLP NileChat Arabizi-Morocco resource distributed on Hugging Face as `UBC-NLP/nilechat-arabizi-mor`. According to its dataset card, Arabic-script Moroccan material from fineweb-edu-Morocco and LHV-Morocco was converted to Arabizi with CohereLabs’ Command R+ (August 2024 release); the Arabizi is therefore generated rather than naturally typed by social-media users. The creators request citation of the NileChat paper [7], and the dataset is released under the Creative Commons Attribution-NonCommercial 4.0 license. The fixed experimental export contains 6,436 training, 804 validation, and 805 held-out test passage pairs. The archived artifacts verify that the test export was excluded from model fitting. The recovered preprocessing notebook documents two successive random splits with seed 42, assigning 80% of the cleaned corpus to training and dividing the remainder equally between validation and test. Reconstructing this procedure reproduced all archived partitions exactly in identifiers, source text, target text, and row order. The source revision, upstream subset-selection procedure, original corpus-deduplication operation, and complete history of test-informed rule development were not preserved, so stronger historical claims about an “untouched” test set are avoided. Before partitioning, the recovered code removes URLs, reduces character runs of three or more to two, replaces selected punctuation with spaces, and filters out Arabizi containing Arabic-block characters, empty sides, and word-count ratios outside [0.5, 2.0]. In the saved run, these row filters removed none of the 8,045 already-cleaned input rows. The original upstream retrieval and corpus-deduplication code are absent from the recovered notebook.

Exact SHA-256 identifiers for all three frozen splits are stored in the released `run_manifest.json`.

After whitespace normalization, exact cross-split overlap was checked for identifiers, source strings, target strings, and source–target pairs, both before and after the experiment’s normalization. No exact overlaps were found. A separate lowercased whitespace-token trigram term frequency–inverse document frequency screen, capped at 200,000 features and excluding exact matches, found no train/validation/test candidates at cosine similarity ≥ 0.85 . This heuristic screen can miss paraphrases and spelling variants. A post-hoc identifier check also found 28 normalized prefixes shared by train and test, affecting 38 test rows, after removing each identifier’s final underscore-delimited component. The upstream meaning of these prefixes could not be reconstructed. These rows are neither exact text duplicates nor above-threshold trigram matches, but source–document group independence is therefore unverified.

2.2 Alignment Constraint

Both word-level learned systems require source and target passages to have equal whitespace-token counts before positional token pairing. Only 692 of 6,436 training passages (10.75%) meet this condition; 5,744 are skipped for both MLE and LSTM pair extraction but remain in the corpus. The 692 passages yield 178,353 aligned token occurrences. Length filtering and pair deduplication leave 79,909 unique LSTM training pairs. Validation produces 12,165 unique pairs from 91 eligible passages. All 805 test passages remain in final evaluation; diagnostic test–pair extraction does not filter the benchmark.

Equal token counts do not guarantee linguistic alignment. To test this assumption, two native Moroccan Arabic speakers independently labeled 100 training token pairs drawn from 89 of the 692 eligible passages. Sampling was proportional to token occurrence. The annotators completed their work without seeing each other’s labels. Both labeled 75 pairs aligned and 25 misaligned, with 100% observed agreement and Cohen’s $\kappa = 1.00$. This audit concerns positional training pairs, not sentence-level reference validity. The 25 sampled misalignments include segmentation and target-content mismatches, and the sample should not be extrapolated as a corpus-wide error rate. No training item was changed after this audit.

2.3 Test-Reference Validation

Two native Moroccan Arabic speakers separately reviewed a random sample of 300 held-out test pairs. The rubric labeled each pair valid, inaccurate, or uncertain according to whether the Arabizi input and Arabic-script target formed a contextually appropriate de-romanization pair. Both accepted all 300; no pair was removed or corrected. Observed agreement is therefore 100%, but Cohen’s κ is undefined because the observed marginals imply chance agreement of 1 and hence a zero denominator. The validation is a targeted quality check, not evidence that every possible spelling has a single gold form.

2.4 Normalization

The executed pipeline uses custom whitespace cleanup, lowercasing of Arabizi where required by a model, and limited token normalization for aligned qualitative analysis. Arabic spelling, punctuation, and diacritics are not normalized in the main metrics.

3 Systems

3.1 Ordered Rule Mapper

The deterministic baseline applies regex substitutions in a fixed order. Multi-character patterns such as *kh*, *gh*, *ch*, *ou*, and *ee* precede single-character replacements. Numerals include $7 \rightarrow \text{ﺉ}$, $3 \rightarrow \text{ﻉ}$, $9 \rightarrow \text{ﻖ}$, $5 \rightarrow \text{ﻑ}$, and $2 \rightarrow \text{ﺀ}$. The system is transparent and deterministic; characters without a rule pass through unchanged. It cannot resolve lexical or contextual ambiguity.

3.2 MLE with Regex Fallback

The maximum likelihood estimation (MLE) system counts positionally aligned training tokens and stores the most frequent Arabic target for each Arabizi type. For an unseen test token, it invokes the same regex mapper. For an Arabizi token w_{az} and candidate Arabic token w_{ar} ,

$$P(w_{ar} | w_{az}) = \frac{C(w_{az}, w_{ar})}{\sum_{w' \in V_{ar}} C(w_{az}, w')}, \quad \hat{w}_{ar} = \arg \max_{w_{ar}} P(w_{ar} | w_{az}). \quad (1)$$

Here, $C(w_{az}, w_{ar})$ is the observed pair count and V_{ar} is the Arabic target vocabulary associated with the source type. The learned dictionary contains 46,901 Arabizi types. This baseline captures frequent lexical mappings but has no sentence context and inherits noise from positional pairing, a known concern for dialectal OOV processing [3].

3.3 Character-Level LSTM

The TensorFlow/Keras sequence-to-sequence (Seq2Seq) system is a word-level character model with 263,566 parameters. A masked long short-term memory (LSTM) encoder consumes one-hot Arabizi characters, and an LSTM decoder generates Arabic symbols under teacher forcing between explicit start and end symbols. It uses 79 encoder symbols, 142 decoder symbols, fixed encoded lengths of 23 and 22 (the decoder length includes boundary symbols), latent dimension 128, batch size 128, and RMSprop with initial learning rate 10^{-3} . Training permits at most 50 epochs, saves the best validation checkpoint, and uses early stopping with patience 5 and best-weight restoration. Temporal binary weights mask target padding. Greedy inference maps unknown input characters to an

unknown symbol, clips longer inputs to 23 positions, and generates at most 22 decoder steps. Words are decoded independently, so this system is neural but not sentence-contextual.

The decoder minimizes categorical cross-entropy,

$$\mathcal{L}_B = -\frac{1}{BT_{\max}} \sum_{n=1}^B \sum_{t=1}^{T_{\max}} m_{n,t} \sum_{c=1}^{|\mathcal{C}_{ar}|} y_{n,t,c} \log \hat{y}_{n,t,c}. \quad (2)$$

Here B is the current batch size, $T_{\max}$ is the fixed decoder dimension, $m_{n,t}$ is 1 for a valid next-symbol target (including the end symbol) and 0 for padding, $\mathcal{C}_{ar}$ is the decoder symbol vocabulary, $y_{n,t,c}$ is the one-hot gold indicator, and $\hat{y}_{n,t,c}$ is the predicted probability. This is the reduction used by the saved Keras model with temporal sample weights; it does not renormalize each word by its individual length. Across the 50 recorded epochs, training loss decreases from 0.894 to 0.349 and validation loss from 0.816 to 0.331 without visible divergence. However, 4,388 of 12,165 validation word pairs (36.07%) also occur in training, so validation loss is not an unseen-word generalization estimate. Original LSTM training hardware and runtime were not recorded.

3.4 Pretrained Full-Passage mT5

The contemporary baseline fine-tunes `google/mt5-small`, a multilingual text-to-text Transformer pretrained on 101 languages [16]. Unlike MLE and the LSTM, it consumes complete passages and uses all 6,436 training and 804 validation pairs without positional word-count filtering. The pinned model revision is `73fb5dbe4756edadc8fbe8c769b0a109493acf7a`; it has 300,176,768 parameters. Fine-tuning uses three epochs, seed 42, Adafactor, an initial learning rate of 10^{-3} with linear decay, microbatch size 1, gradient accumulation over 16 microbatches, bfloat16 precision, and gradient checkpointing. The pinned slow tokenizer is loaded from the same revision. Neither inputs nor references are truncated. Greedy decoding uses a frozen limit of 2,048 new tokens.

Training on an NVIDIA L4 (22.0 GiB) takes 4,625 seconds (77.1 minutes), peaks at 8.31 GiB allocated memory, and reaches best validation loss 0.554 at epoch 3. The preserved log evaluates at epoch boundaries (steps 403, 806, and 1,209) and selects the final checkpoint by validation loss. The environment records PyTorch 2.11.0+cu128, Transformers 5.8.1, Accelerate 1.13.0, and SentencePiece 0.2.1. Generating the 805 test passages takes 8,719 seconds (10.83 seconds per passage). This comparison is deliberately practical, not architecture-controlled: mT5 benefits from external pretraining, sentence context, subword modeling, and more passage-level training examples than the two word-aligned learned systems.

4 Evaluation

All reported primary scores use the same 805 frozen test passages after collapsing runs of whitespace. SacreBLEU uses case-sensitive international tokenization and

exponential smoothing [11, 14]. Its reproducibility signature is `nrefs:1|case:mixed|eff:no|tok:intl|smooth:exp|version:2.6.0`. chrF uses character order 6, $\beta = 2$, and excludes whitespace [13]. CER is corpus-level Levenshtein character edits divided by total reference characters, retaining normalized spaces as Unicode code points; WER is word-sequence Levenshtein edits divided by total reference words. CER and WER are lower-is-better and may exceed 100% when insertions are extensive.

The legacy position-wise exact-match score, renamed *position-wise token agreement* (PosEM), counts identical whitespace tokens at the same index and divides by the number of reference tokens. It is retained for comparability with the submitted experiment, but it is not standard word accuracy: a single insertion shifts later matches, while extra trailing prediction tokens are not penalized. This explains why PosEM can be low even when BLEU is substantially higher; BLEU gives partial credit for matching n-grams found elsewhere in the passage.

A post-hoc paired bootstrap analysis [10] resamples complete test passages with replacement 10,000 times (seed 42), using identical sampled indices for all four systems. It reports percentile 95% confidence intervals for BLEU and for the mT5-minus-MLE difference. The focal BLEU comparison also uses SacreBLEU’s paired bootstrap significance test with the same resample count and seed. All 805 passages, including decoding failures, are retained. The intervals are conditional on the saved models and an independent-passage resampling assumption; they do not quantify training-seed variability or account for unverified source-document grouping.

5 Results

5.1 Blind-Test Benchmark

Table 1 gives the primary results. mT5 has the highest observed BLEU and PosEM, but MLE has the highest chrF and the lowest CER and WER. The finding is therefore metric-dependent. In particular, the LSTM improves over the regex baseline in BLEU and chrF but has worse CER, so it is not uniformly better. The BLEU 95% intervals are [34.84, 38.71] for mT5 and [34.80, 36.08] for MLE. Their observed difference is 1.31 points, with a paired difference interval of $[-0.55, 3.24]$ and paired-bootstrap $p = 0.074$. This focal comparison does not establish a BLEU advantage at the nominal 0.05 level; it also does not demonstrate equivalence. The aggregate reproducibility record includes intervals for all four systems.

Table 1: Primary results on all 805 blind-test passages. BLEU, chrF, and PosEM are higher-is-better; CER and WER are lower-is-better.

System	BLEU	chrF	CER (%)	WER (%)	PosEM (%)
Rule mapper	7.24	46.73	24.50	72.50	8.69
MLE + regex	35.44	69.60	15.17	42.48	16.76
Word LSTM	9.62	51.98	44.94	69.00	9.42
mT5-small	36.75	68.86	42.86	64.03	21.54

Individual BLEU precisions in Fig. 1 add detail but do not by themselves prove contextual reasoning. MLE has the strongest unigram and bigram precision, while mT5 is strongest at trigrams and four-grams. The flatter mT5 curve is consistent with better preservation of longer spans, but external pretraining and access to full passages are confounds. The LSTM decodes words independently and does not use sentence context.

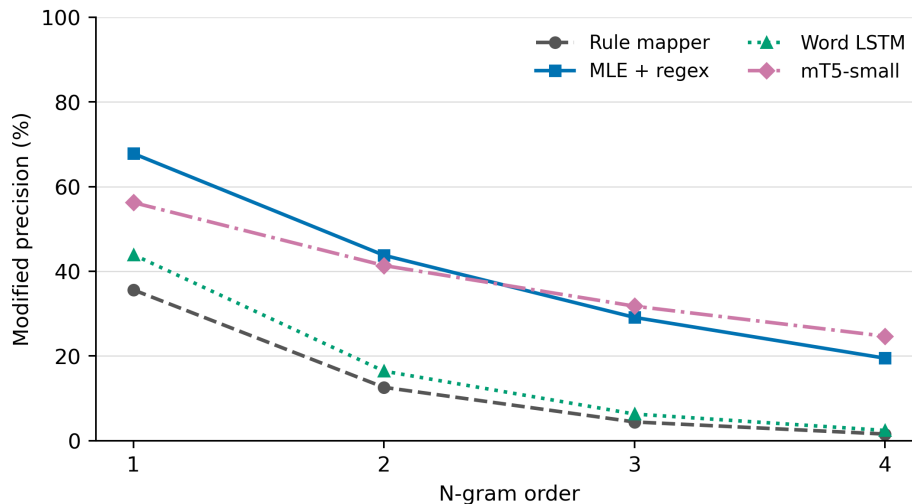


Fig. 1: Modified n-gram precision on all 805 blind-test passages. Markers and line styles distinguish systems in grayscale. These corpus overlap curves are descriptive, not a controlled test of contextual reasoning.

5.2 MLE Coverage and Fallback

On the test set, 47,327 of 225,581 token occurrences (20.98%) are absent from the MLE dictionary. At the type level, 34,893 of 51,904 types (67.23%) are unseen, reflecting the long tail of spelling variants. Table 2 isolates the regex fallback by

replacing it with copying an unseen lowercased token unchanged. The fallback raises BLEU by 3.20 and reduces CER by 16.93 absolute points. The submitted statistical result should therefore be understood as *MLE with regex fallback*, not a pure lookup table.

Table 2: MLE unseen-token ablation on all blind-test passages.

Unseen-token policy	BLEU	chrF	CER (%)	WER (%)	PosEM (%)
Regex fallback	35.44	69.60	15.17	42.48	16.76
Copy Arabizi token	32.24	54.66	32.10	45.20	16.03

5.3 Transformer Generation Stability

All 805 mT5 outputs are included in the primary row of Table 1. Seventy-eight outputs (9.69%) do not emit an end-of-sequence token before the fixed 2,048-token generation limit; many contain long repetitions. Table 3 separates these groups only as a post-hoc diagnostic. It must not replace the primary benchmark or be used to retune decoding on the test set.

Table 3: Post-hoc mT5 generation diagnostic; the all-row result remains primary.

Subset	<i>n</i>	BLEU	chrF	CER (%)	WER (%)
All outputs	805	36.75	68.86	42.86	64.03
Reached EOS	727	46.11	74.06	23.20	42.88
No EOS	78	9.67	38.99	191.60	224.94

5.4 Human-Validated Subset

Table 4 evaluates all four systems on the 300 independently checked references. The same mixed ranking holds: mT5 has the highest BLEU and PosEM, while MLE has lower error rates. This suggests that the full-test conclusion is not driven by obviously invalid reference pairs.

Table 4: Results on the 300-row human-validated test subset.

System	BLEU	chrF	CER (%)	WER (%)	PosEM (%)
Rule mapper	7.34	46.76	24.55	72.40	8.84
MLE + regex	35.58	69.74	15.14	42.31	16.73
Word LSTM	9.43	51.75	45.63	69.22	9.32
mT5-small	37.20	69.47	40.25	61.51	20.89

6 Human and Linguistic Analysis

6.1 Reviewed LSTM Errors

The 50 apparent LSTM errors were drawn from the 300-row human-validated subset. Predicted and reference token sequences were lightly normalized for punctuation, Arabic diacritics, alef variants, and alef maqsura, then aligned with `SequenceMatcher`. Unequal-length replacement spans were excluded from the review pool to reduce false pairings caused by word-position shifts; 50 remaining cases were sampled with random seed 42. An initial human review classified 42 as true model errors, 6 as acceptable variants or non-errors, and 2 as uncertain. The reviewer could assign multiple contributing labels. For summary reporting, one primary label was selected by the fixed priority order lexical substitution, phonetic ambiguity, numerical transliteration, clitic segmentation, loanword/code-switching, punctuation, alignment shift, source noise, gold-reference issue, no error, and other. This reduction yields 22 lexical substitutions, 19 phonetic ambiguities, and one source-noise case among the 42 true errors; these are derived primary categories rather than mutually exclusive labels originally selected by the reviewer.

For camera-ready adjudication, two reviewers re-examined the disputed case $w7a \rightarrow \text{والحال}$ (reference واخا) with its full passage context and agreed that it should be labeled uncertain rather than acceptable. They assigned *other* as the primary type and source noise as a contributing factor: in their rationale, 7 encodes ح, whereas the contextual form requires the خ sound, commonly encoded by 5. The final counts are therefore 42 true model errors, 5 acceptable variants or non-errors, and 3 uncertain cases. The small, error-conditioned sample cannot estimate corpus accuracy or the population distribution of error types.

Table 5: Final human judgments for 50 sampled apparent LSTM errors after full-context adjudication.

Judgment	Count
True model error	42
Acceptable variant / not a model error	5
Uncertain	3

6.2 Phenomenon-Oriented Interpretation

Numeral substitutions are often locally correct: for example, 3 commonly represents the Arabic consonant *ayn*, while 7 commonly represents the voiceless pharyngeal consonant *ha*. Errors can remain in the surrounding vowels or lexical choice, as in *l3asr*, where producing العاصر instead of العصر shows that mapping the numeral is necessary but insufficient. Dates and times create a separate problem because their digits should be preserved rather than treated as Arabizi letters. This analysis is qualitative; it does not estimate phenomenon-specific error rates.

Clitics such as *w-*, *l-*, *b-*, and *f-* attach productively and complicate both whitespace alignment and exact evaluation. A predicted clitic may be linguistically plausible even when the reference uses another segmentation. Short forms also show lexical ambiguity: the LSTM sometimes generates a fluent but unrelated frequent word, which accounts for the largest human error category. French or Spanish names and loanwords introduce code-switching choices between copying, Arabic-script transliteration, and lexical normalization. Finally, noisy or underspecified Arabizi may admit several plausible spellings. These cases explain why BLEU, edit distance, and human judgment answer different questions; none alone is a complete measure of dialectal adequacy.

7 Related Work and Positioning

Directly related work includes automatic transliteration of Romanized dialectal Arabic and Arabizi detection and conversion [1, 5]. Their resources and evaluation protocols differ from the generated Moroccan benchmark used here, so their reported scores are not directly comparable. Arabic romanization, script conversion, and dialectal resource development also include bibliographic romanization, dialect lexicons, and morphological tools [6, 8, 12]. Cross-script studies show that romanization can assist transfer but can also erase useful distinctions [4, 15]. Recent full-sentence transliteration work demonstrates the value of combining lexical models, language models, and pretrained sequence-to-sequence systems while carefully separating contextual from non-contextual gains [9]. mT5 provides a multilingual pretrained text-to-text architecture suitable for the stronger baseline added here [16].

The originality of this work lies in its focused Maghribi Arabizi benchmark and audit protocol: four modeling tiers, explicit OOV fallback analysis, cross-split overlap checks, multiple automatic metrics, and human checks of both test references and the equal-count alignment assumption. It does not claim a new architecture or that the tested mT5 configuration represents the ceiling of Transformer-based de-romanization.

8 Conclusion

This study established an audited four-system benchmark for Moroccan Arabizi de-romanization. No system dominates every evaluation dimension. Under a con-

figuration that jointly changes architecture, parameter count, subword tokenization, external pretraining, passage context, and access to training passages, mT5 achieves the highest observed BLEU, position-wise token agreement, and longer n-gram precision. However, the paired bootstrap analysis does not establish its 1.31-point BLEU lead over MLE at the 0.05 level. The MLE mapper with regex fallback instead produces the best chrF, CER, and WER, indicating stronger character- and word-level form fidelity on this benchmark. Seventy-eight mT5 outputs do not emit EOS within the configured limit; their repetitions sharply increase edit distance despite the model’s higher corpus BLEU.

The comparison therefore reveals complementary strengths rather than a single universal winner. The rule mapper offers transparency but limited lexical knowledge; MLE exploits observed lexical regularities and an effective fallback; the word-level LSTM learns local character correspondences but is constrained by noisy positional alignment; and mT5 achieves stronger longer-span overlap while its generation failures contribute large edit penalties. The ranking changes with the metric because the systems fail differently. The human audits show that positional alignment quality, source noise, and valid dialectal spelling variation are central experimental variables. These findings define the benchmark’s value without treating the generated Arabizi resource as a substitute for evaluation on naturally authored text or the automatic ranking as evidence of human preference.

Limitations

The source-side Arabizi was generated from Arabic-script Moroccan material, so transfer to naturally authored Arabizi, code-switching, and real-user spelling distributions remains untested. The full-passage Transformer is not an architecture-only comparison with the two word-level learned systems because it also differs in pretraining, scale, tokenization, context, and training-data eligibility; only one fine-tuning seed is reported. The exact original mT5 training script was not preserved: the archived resumable runner differs from the successful run in evaluation/checkpoint frequency, although the executed configuration and log are retained. The 2,048-token generation limit was fixed before inference, yet 9.69% of outputs do not emit EOS within it. MLE and the LSTM are trained from only 10.75% of passages and inherit positional alignment noise. The 100-pair alignment audit is occurrence-weighted and not a corpus prevalence estimate; the 50-case LSTM error review is small and error-conditioned, and its disputed source-noise case required full-context consensus adjudication. An exploratory blinded mT5–MLE output review was completed on 50 random passages and 10 additional decoding-failure cases, but is not used to validate the automatic ranking: whole-passage acceptability labels were nearly constant, preferences differed between reviewers, and two acceptability judgments remained unresolved. Its aggregate results are retained in the audit record; no human-preference advantage is claimed. The 300-row reference audit contains no negative labels, making Cohen’s κ undefined. Upstream subset construction and source-group semantics

could not be reconstructed, and shared identifier prefixes mean source-document independence is unverified. The near-duplicate search is heuristic, and BLEU, chrF, CER, WER, and PosEM remain reference-sensitive when several dialectal spellings are acceptable.

Ethical Considerations and AI-Use Disclosure

The resource contains social-media-style language that may include noisy, personal, or sensitive material. The experiments do not attempt to identify users, infer personal attributes, or profile individuals. Any release of derived data or model outputs should follow the source license and minimize personally identifying content. Native-speaker annotations were used for linguistic quality control and are reported in aggregate.

Generative AI tools were used for code debugging, editorial feedback, and manuscript revision support. The author verified the experimental claims, reported values, citations, and methodological descriptions against the executed notebooks, saved outputs, and annotation files.

Data and Code Availability. Code, split identifiers, file hashes, aggregate annotation summaries, and aggregate evaluation artifacts are maintained in a private GitHub repository:

`https://github.com/A-Nafissi/Arabizi-into-Arabic-Script`

Repository access is available on request from the author using the email address listed above. Full passage text and raw annotation workbooks are not redistributed. Access to the source corpus remains governed by the upstream Hugging Face access conditions and CC BY-NC 4.0 license.

Disclosure of Interests. The author has no competing interests to declare that are relevant to the content of this article.

References

- [1] Al-Badrashiny, M., Eskander, R., Habash, N., Rambow, O.: Automatic transliteration of romanized dialectal arabic. In: Proceedings of the Eighteenth Conference on Computational Natural Language Learning. pp. 30–38. Association for Computational Linguistics, Ann Arbor, Michigan (2014). <https://doi.org/10.3115/v1/W14-1604>
- [2] Alayba, A.M.: Arabic natural language processing (NLP): A comprehensive review of challenges, techniques, and emerging trends. *Computers* **14**(11), 497 (2025). <https://doi.org/10.3390/computers14110497>
- [3] Aminian, M., Ghoneim, M., Diab, M.: Handling OOV words in dialectal arabic to english machine translation. In: Proceedings of the EMNLP’2014 Workshop on Language Technology for Closely Related Languages and Language Variants. pp. 99–108. Association for Computational Linguistics, Doha, Qatar (2014). <https://doi.org/10.3115/v1/W14-4213>

- [4] Amrhein, C., Sennrich, R.: On romanization for model transfer between scripts in neural machine translation. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 2461–2469. Association for Computational Linguistics (2020). <https://doi.org/10.18653/v1/2020.findings-emnlp.223>
- [5] Darwish, K.: Arabizi detection and conversion to arabic. In: Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing. pp. 217–224. Association for Computational Linguistics, Doha, Qatar (2014), <https://aclanthology.org/W14-3629/>
- [6] Diab, M., Al-Badrashiny, M., Aminian, M., Attia, M., Elfardy, H., Habash, N., Hawwari, A., Salloum, W., Dasigi, P., Eskander, R.: Tharwa: A large scale dialectal arabic - standard arabic - english lexicon. In: Proceedings of the Language Resources and Evaluation Conference. pp. 3782–3789. European Language Resources Association (ELRA) (2014), <https://aclanthology.org/L14-1115/>
- [7] El Mekki, A., Atou, H., Nacar, O., Shehata, S., Abdul-Mageed, M.: NileChat: Towards linguistically diverse and culturally aware LLMs for local communities. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. pp. 10967–10991. Association for Computational Linguistics (2025). <https://doi.org/10.18653/v1/2025.emnlp-main.556>
- [8] Eryani, F., Habash, N.: Automatic romanization of arabic bibliographic records. In: Proceedings of the Sixth Arabic Natural Language Processing Workshop. pp. 213–218. Association for Computational Linguistics (2021), <https://aclanthology.org/2021.wanlp-1.23/>
- [9] Kirov, C., Johny, C., Katanova, A., Gutkin, A., Roark, B.: Context-aware transliteration of romanized south asian languages. *Computational Linguistics* **50**(2), 475–534 (2024). https://doi.org/10.1162/coli_a_00510
- [10] Koehn, P.: Statistical significance tests for machine translation evaluation. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. pp. 388–395. Association for Computational Linguistics, Barcelona, Spain (2004), <https://aclanthology.org/W04-3250/>
- [11] Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: A method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. pp. 311–318. Association for Computational Linguistics (2002). <https://doi.org/10.3115/1073083.1073135>
- [12] Pasha, A., Al-Badrashiny, M., Diab, M., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., Roth, R.: MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In: Proceedings of the Language Resources and Evaluation Conference. pp. 1094–1101. European Language Resources Association (ELRA) (2014), <https://aclanthology.org/L14-1479/>
- [13] Popović, M.: chrF: Character n-gram F-score for automatic MT evaluation. In: Proceedings of the Tenth Workshop on Statistical Machine

- Translation. pp. 392–395. Association for Computational Linguistics (2015). <https://doi.org/10.18653/v1/W15-3049>
- [14] Post, M.: A call for clarity in reporting BLEU scores. In: Proceedings of the Third Conference on Machine Translation: Research Papers. pp. 186–191. Association for Computational Linguistics, Brussels, Belgium (2018). <https://doi.org/10.18653/v1/W18-6319>
 - [15] Purkayastha, S., Ruder, S., Pfeiffer, J., Gurevych, I., Vulić, I.: Romanization-based large-scale adaptation of multilingual language models. In: Findings of the Association for Computational Linguistics: EMNLP 2023. pp. 7996–8005. Association for Computational Linguistics (2023). <https://doi.org/10.18653/v1/2023.findings-emnlp.538>
 - [16] Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C.: mT5: A massively multilingual pre-trained text-to-text transformer. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 483–498. Association for Computational Linguistics, Online (2021). <https://doi.org/10.18653/v1/2021.naacl-main.41>