

Research Article

# A Sequence-to-Sequence Deep Learning Framework for Arabic Root Extraction and Stemming

Zaid Rajih Mohammed  <sup>1\*</sup>

<sup>1</sup>Faculty of Medical Sciences, Jabir ibn Hayyan University for Medical and Pharmaceutical Sciences, Najaf, Iraq.

\*Corresponding author: [zaid.rajah@jmu.edu.iq](mailto:zaid.rajah@jmu.edu.iq)

## Article Info

**Keywords:** Arabic NLP, Root extraction, Stemming, Morphological analysis, Sequence-to-sequence, Attention, BiLSTM, Transformer.

**Received:** 05.09.2026;

**Accepted:** 20.10.2026;

**Published:** 28.10.2026



© 2026 by the author's. The terms and conditions of the Creative Commons Attribution (CC BY) license apply to this open access article.

## Abstract

Arabic morphology is non-concatenative: words are built by interdigitating a consonantal root with a vocalic pattern and then attaching clitics, so the root letters are rarely contiguous and weak letters (و, ي, ل) and hamza change or disappear. Rule-based root extractors such as Khoja and ISRI therefore fail on irregular, weak, doubled and foreign words and on out-of-vocabulary forms. We formulate root extraction as character-level sequence generation,  $F(\text{word}) \rightarrow \text{root}$ , and compare three encoder-decoder architectures (BiLSTM with Bahdanau attention, BiGRU with Luong attention, and a Transformer) with four baselines (ISRI, Tashaphyne, Khoja and a character CRF over edit tags) on identical test items. The main benchmark is built from the Lexicon of the Modern Arabic Language (Kaggle release, 61,190 word-root pairs, 5,706 roots); a second benchmark of 45,249 inflected, clitic-bearing tokens with sentence context is derived from the Quranic Arabic Corpus. On unseen word forms the best model (BiLSTM + Bahdanau) reaches 89.43% root accuracy and 96.84% character F1 on the modern-Arabic benchmark, against 70.49% for Khoja and 77.23% for the CRF, and 95.33% on the Quranic benchmark (Khoja 65.07%); every difference from a baseline is significant (exact McNemar test), whereas the three neural architectures do not differ significantly. Gains are largest on defective (naqis), doubly weak (lafif) and doubled roots. On roots never seen in training the models still lead on modern Arabic but fall behind affix rules on clitic-bearing Quranic tokens, and a model trained on dictionary forms alone transfers poorly to inflected text unless clitic-bearing examples are added. We also report an ablation study and a stemming (lemma) task, and release the benchmark splits and code.

## 1. Introduction

Most Arabic words derive from a root of three (occasionally four) consonants that carries a core meaning, combined with a morphological pattern and optional proclitics (conjunctions, prepositions, the article) and enclitics (pronouns). The word 'المستخدمون' and the users', for example, contains the root م - د - خ only as three scattered letters. Reducing words to roots groups hundreds of surface forms into one index term and is used in information retrieval, text classification, sentiment analysis and machine translation for Arabic [1–3]. Root extraction differs from light stemming, which only strips affixes, and from lemmatisation, which returns a dictionary headword; it is the most aggressive and most language-specific of the three.

### 1.1. Research gap

Rule-based extractors [4–6] remove affixes and match the remainder against pattern and root lists. They break down when a root letter is itself an affix letter, when weak letters and hamza alternate (قال from ل – و – ق), when a doubled root is contracted (مدّ from د – د – م), for broken plurals, and for words absent from their lists. Recent neural work Table 1 targets stems, lemmas or segmentation rather than roots [7–10], evaluates mostly on classical text or treebank lemmas, and rarely compares attention mechanisms or reports performance on roots never seen during training. LLM studies further show that large pre-trained models do not reliably encode root–pattern structure [11].

### 1.2. Main contributions

1. A character-level sequence-to-sequence formulation of Arabic root extraction, with three encoder–decoder architectures (BiLSTM + Bahdanau, BiGRU + Luong, Transformer) trained and compared under identical conditions.
2. An empirical evaluation against rule-based (ISRI, Tashaphyne, Khoja) and statistical (CRF) baselines on the same test items, with significance tests, per-root-class accuracy, unseen-root and cross-corpus tests, an ablation study and a stemming task.
3. Two reproducible benchmarks: a modern-Arabic word→root benchmark built from the Lexicon of the Modern Arabic Language (CC BY 4.0) and a context-bearing Quranic benchmark aligned with orthographic text, both with leakage-free splits.

## 2. Related Work

### 2.1. Traditional and rule-based approaches

Khoja and Garside [4] strip the longest matching prefix and suffix and match the stem against pattern templates and a root dictionary. The ISRI stemmer [5] extracts roots without a root dictionary, and Tashaphyne [6] offers a configurable light stemmer with root extraction. Light stemmers [12] and the hybrid light stemmer of Namly and Bouzoubaa [13] remove affixes only, while Farasa [14] segments and stems with a ranking SVM. A recent survey [1] notes that the lack of a unified list of roots, patterns and affixes still limits rule-based extraction.

### 2.2. Machine-learning approaches

Statistical methods cast stemming as classification or sequence labelling. Conditional random fields [15] label each character, and support-vector and naive-Bayes classifiers choose among candidate stems. Akhmetov et al. [16] learn stemming rules from corpora, reporting 27.1% test accuracy, and Maneetah [17] shows that rule-based stemming errors can hurt a downstream classifier. Lexicon-lookup tools such as Alma [18] are fast and accurate on in-vocabulary words but cannot analyse unseen forms.

### 2.3. Deep learning in morphological analysis

Masdar [19] introduced an LSTM encoder–decoder for Arabic stems without attention. Char Stemmer [7] uses a character BiLSTM seq2seq model on 260,000 classical words and reports 93.88% stemming accuracy. For lemmatisation, AraBART [8] reaches 94.71% (Farasa 97.32%), a BLSTM with morphological filtering and an HMM layer improves accuracy by nine points [9], and Saeed and Habash [20] find classification over lemma–POS–gloss tags more robust than seq2seq generation, which can hallucinate implausible forms. Deep recurrent segmenters reach 98.33% F1 on ATB [10]. Pre-trained encoders (AraBERT [21], MARBERT [22]) operate on sub-words and do not expose roots, and LLM evaluations [11], [23] show weak root–pattern generalisation. None of these works extracts roots with an attention-based character seq2seq model on modern Arabic with an unseen-root test.

**Table 1:** Recent related works (numbers as reported by the authors; not directly comparable across datasets)

| Work                          | Year | Model               | Target     | Data                     | Result            |
|-------------------------------|------|---------------------|------------|--------------------------|-------------------|
| Alaswaad & Minaei-Bidgoli [7] | 2026 | Char BiLSTM seq2seq | Stem       | Classical, 260k          | 93.88% acc.       |
| Afartass et al. [8]           | 2026 | AraBART             | Lemma      | PADT + Farasa            | 94.71% acc.       |
| Belayachi & Mazroui [9]       | 2026 | BLSTM + HMM         | Lemma      | MSA corpus               | +9 pts            |
| Abdelrehem et al. [10]        | 2026 | BiGRU tagging       | Segment.   | ATBv3                    | 98.33% F1         |
| Saeed & Habash [20]           | 2025 | LPG classification  | Lemma      | Multi-genre              | —                 |
| Jarrar et al. [18]            | 2024 | Lexicon lookup      | Lemma/root | ATB                      | 87.8% F1 (lemma)  |
| Akhmetov et al. [16]          | 2026 | Rule learning       | Stem/lemma | —                        | 27.1% acc.        |
| <b>Proposed</b>               | 2026 | BiLSTM              | Root       | Modern lexicon + Quranic | 89.43% acc. (LEX) |

## 3. Proposed Methodology

### 3.1. Problem formulation

Given a normalised word  $w = (w_1, \dots, w_n)$  over the Arabic alphabet, the task is to generate its root  $r = (r_1, \dots, r_m)$ , usually  $m = 3$  or  $4$ . A model with parameters  $\theta$  factorises

$$P(r | w; \theta) = \prod_t P(r_t | r_1, \dots, r_{t-1}, w; \theta)$$

and is trained by minimising the negative log-likelihood of the gold roots with teacher forcing. At test time the root is decoded with beam search ( $k = 5$ ). Because generation is free, the model can restore letters that are not visible in the word (the  $\text{و}$  of  $\text{قال} \rightarrow \text{قول}$ , the second  $\text{د}$  of  $\text{مد} \rightarrow \text{مدد}$ ), which extraction-by-deletion methods cannot do. For the context experiment the input becomes (previous word || word || next word) with separator symbols Figure 1.

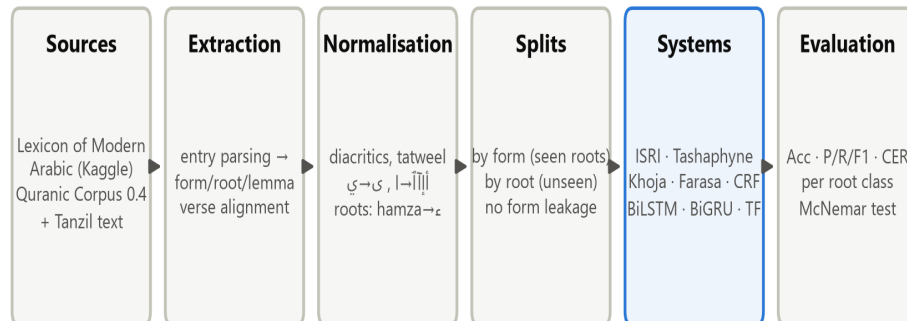


**Figure 1:** Processing pipeline: sources, extraction, normalisation, leakage-free splits, systems and evaluation

### 3.2. Datasets and preprocessing

**LEX (main benchmark).** The Lexicon of the Modern Arabic Language [24] is distributed on Kaggle (tahaalselwii/arabic-lexicons, CC BY 4.0) as 32,075 entries, each labelled with its root. From every entry we extract the headword and the forms the lexicon itself prints with it: spelling variants, plurals, feminine and singular forms for nouns; present tense, verbal nouns, active and passive participles and imperatives for verbs. No form is generated. Notes in parentheses and complement words are removed; for sound roots the root letters appear in order inside 99.80% of the extracted forms, which bounds the extraction noise. The result is 61,190 word–root pairs, 57,553 unique forms and 5,706 roots, including loanwords whose ‘root’ is their consonant skeleton (e.g. *ارستقراطية*).

**QAC (second benchmark).** The Quranic Arabic Corpus 0.4 [25] annotates each of its 77,429 tokens with morphological segments and, for derived words, the root. Its Uthmani script cannot be converted letter by letter into standard spelling, so we align each token with the Tanzil ‘simple-clean’ orthographic text [26] by (chapter, verse, word) and drop verses whose word counts differ (468 of 6236 verses). Tokens without a root (particles, pronouns, most proper nouns) are excluded, leaving 45,249 tokens (13,183 forms, 1,606 roots). Unlike LEX, these tokens keep their clitics (و, ف, ب, ال, pronoun suffixes) and have left and right neighbours.

**Normalisation.** Diacritics and tatweel are removed, ٱ→ي, ٱ→ا are mapped to ا and ي to ي. In roots every hamza carrier is written ء so that root letters remain distinct. **Splits.** Items are split 80/10/10 by normalised form, so no form occurs in two parts (seen-root split); a second split groups by root, so test roots never occur in training (unseen-root split). Leakage checks give zero shared forms and zero shared roots respectively. After normalisation 5.01% of LEX forms and 0.38% of QAC forms have more than one root; we therefore also report lenient accuracy (the prediction is any root the same form takes in the corpus) As show in table 2 Figure 2.

**Table 2:** Benchmark statistics

| Corpus | Items | Forms | Roots | Lemmas | Ambig. (%) | Train/Dev/Test  |
|--------|-------|-------|-------|--------|------------|-----------------|
| LEX    | 61190 | 57553 | 5706  | 20116  | 5.01       | 48990/6091/6109 |
| QAC    | 45249 | 13183 | 1606  | 3961   | 0.38       | 36975/4180/4094 |

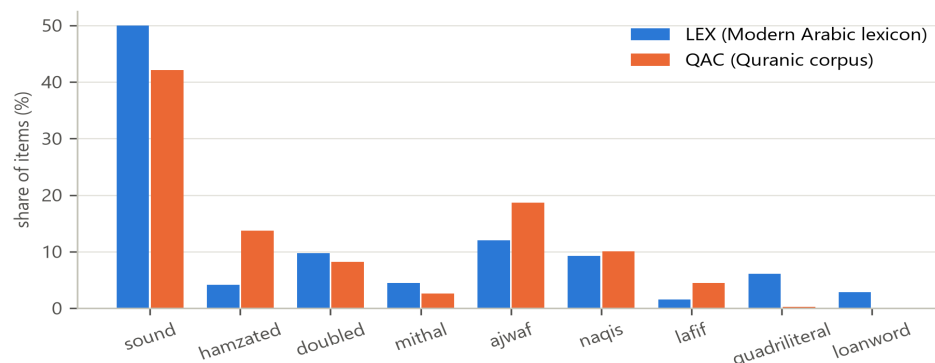


**Figure 2:** Distribution of root classes (sound, hamzated, doubled, weak-initial mithal, weak-medial ajwaf, weak-final naqis, doubly weak laff, quadrilateral, loanword) in both benchmarks

### 3.3. Model architecture

The models follow the encoder–decoder paradigm [27]. Embedding layer. Each character is mapped to a 128-dimensional vector (separate source and target embeddings) Figure 3 Encoder. A two-layer bidirectional LSTM [28] (or GRU [29]) with 256 units per direction reads the word; the final forward and backward states are projected to initialise the decoder. Attention. At each decoding step  $t$  the decoder state  $s_t$  scores every encoder state  $h_i$ ; Bahdanau attention [30] uses an additive score and feeds the context into the recurrent input, Luong attention [31] uses a bilinear score after the recurrent step with input feeding:

$$\text{Bahdanau: } e_{ti} = v^T \tan h(W_h h_i + W_s s_{t-1})$$

$$\text{Luong: } e_{ti} = s_t^T W_a h_i$$

$$\alpha_{ti} = \text{softmax}_i(e_{ti}), \quad c_t = \sum_i \alpha_{ti} h_i$$

Decoder. A two-layer unidirectional LSTM/GRU produces one root letter per step from the previous letter, its state and  $c$ , until the end symbol. The Transformer [32] variant uses three encoder and three decoder layers (model width 256, four heads, feed-forward 512) with learned positional embeddings. All models are small (3.1–4.3 M parameters) and train on a 4 GB GPU.

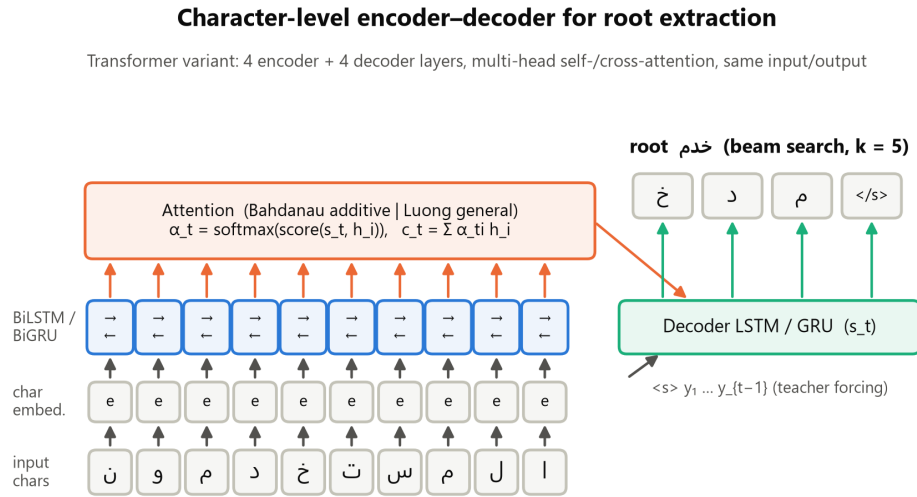


**Figure 3:** Character-level encoder–decoder with attention; the input word is read right-to-left as written

### 3.4. Baseline models

Rule-based: ISRI [5] (NLTK), Tashaphyne 0.3.6 [6] and Khoja [4] (the command-line release by M. Saad). Statistical: a linear-chain CRF [15] trained on our data. Because the root is not a subsequence of the word for weak and doubled roots, we align word and root by minimum edit cost and tag each letter keep, drop, replace-by-x, or keep-and-insert-x; the tag sequence deterministically rebuilds the root, and the tag inventory learned from training covers 99.97% of LEX and 99.98% of QAC test items. Features are the letter, its  $\pm 3$  neighbours, bigrams and trigrams, position from both ends, word length and the first/last two and three letters. Farasa [14] returns stems, so it is scored against lemmas in the stemming experiment only. A Masdar-style model without attention [19] is covered by our no-attention ablation.

### 3.5. Training details

Adam [33] (learning rate  $10^{-3}$  for recurrent models,  $5 \times 10^{-4}$  for the Transformer), batch size 256, dropout 0.2, gradient clipping at 1.0, learning rate halved when development accuracy stalls for two epochs, early stopping after six epochs without improvement (at most 60). Every neural configuration is trained with three random seeds and we report mean  $\pm$  standard deviation. Experiments ran with PyTorch 2.14 on an NVIDIA GTX 1650 SUPER (4 GB).

## 4. Experimental Setup and Results

### 4.1. Evaluation metrics

Accuracy is the share of words whose whole root is generated exactly. Precision, recall and F1 are computed on root letters as multisets (micro-averaged), so a root with one wrong letter earns partial credit. The character error rate is  $\text{CER} = \sum \text{Levenshtein}(\hat{r}, r) / \sum |r|$ . Differences between two systems on the same test items are tested with the exact McNemar test [34].

## 4.2. Main results

**Table 3:** Root extraction on LEX, seen-root split (test forms unseen). Neural rows: mean  $\pm$  s.d. over three seeds

| System         | Acc. (%)         | Len. (%)         | P (%)            | R (%)            | F1 (%)           | CER                 | w/s   |
|----------------|------------------|------------------|------------------|------------------|------------------|---------------------|-------|
| ISRI           | 62.43            | 64.69            | 88.09            | 88.43            | 88.26            | 0.1967              | 3913  |
| Tashaphyne     | 67.64            | 71.17            | 91.09            | 88.49            | 89.77            | 0.1635              | 16019 |
| Khoja          | 70.49            | 74.02            | 89.47            | 91.02            | 90.24            | 0.1593              | 20400 |
| CRF            | 77.23            | 81.01            | 95.88            | 93.20            | 94.52            | 0.0920              |       |
| <b>BiLSTM</b>  | 89.43 $\pm$ 0.30 | 94.83 $\pm$ 0.32 | 97.03 $\pm$ 0.13 | 96.65 $\pm$ 0.09 | 96.84 $\pm$ 0.11 | 0.0479 $\pm$ 0.0012 | 4429  |
| <b>BiGRU</b>   | 89.35 $\pm$ 0.15 | 94.74 $\pm$ 0.13 | 97.04 $\pm$ 0.06 | 96.58 $\pm$ 0.10 | 96.81 $\pm$ 0.08 | 0.0496 $\pm$ 0.0008 | 5737  |
| <b>Transf.</b> | 88.83 $\pm$ 0.47 | 94.19 $\pm$ 0.53 | 96.78 $\pm$ 0.15 | 96.44 $\pm$ 0.16 | 96.61 $\pm$ 0.15 | 0.0527 $\pm$ 0.0022 | 384   |

**Table 4:** Root extraction on QAC, seen-root split (inflected tokens with clitics). Neural rows: mean  $\pm$  s.d. over three seeds

| System         | Acc. (%)         | Len. (%)         | P (%)            | R (%)            | F1 (%)           | CER                 | w/s   |
|----------------|------------------|------------------|------------------|------------------|------------------|---------------------|-------|
| ISRI           | 54.91            | 55.01            | 84.42            | 85.30            | 84.86            | 0.2487              | 2741  |
| Tashaphyne     | 59.97            | 60.26            | 86.61            | 86.20            | 86.41            | 0.2226              | 12171 |
| Khoja          | 65.07            | 65.44            | 86.32            | 86.81            | 86.57            | 0.2174              | 14531 |
| CRF            | 74.40            | 74.77            | 95.11            | 93.65            | 94.37            | 0.1038              |       |
| <b>BiLSTM</b>  | 95.33 $\pm$ 0.25 | 95.70 $\pm$ 0.25 | 98.55 $\pm$ 0.08 | 98.54 $\pm$ 0.07 | 98.55 $\pm$ 0.08 | 0.0245 $\pm$ 0.0009 | 2783  |
| <b>BiGRU</b>   | 95.29 $\pm$ 0.81 | 95.68 $\pm$ 0.83 | 98.47 $\pm$ 0.30 | 98.46 $\pm$ 0.30 | 98.47 $\pm$ 0.30 | 0.0251 $\pm$ 0.0047 | 3132  |
| <b>Transf.</b> | 94.48 $\pm$ 0.68 | 94.85 $\pm$ 0.68 | 98.24 $\pm$ 0.15 | 98.23 $\pm$ 0.14 | 98.23 $\pm$ 0.14 | 0.0305 $\pm$ 0.0044 | 668   |

On LEX the best model, BiLSTM + Bahdanau, extracts 89.43% of roots exactly, 18.94 points above Khoja and 12.20 above the CRF, with character F1 96.84% and CER 0.0479 as show in Table 3.

On QAC, where tokens carry clitics, the best model reaches 95.33% against 65.07% for Khoja and 74.40% for the CRF as show in Table 4. The recurrent models with attention are within a few tenths of a point of each other; the Transformer converges more slowly and ends slightly lower, and none of the differences between the neural models is significant Table 5, Figure 4. Figure 5 shows the development curves.

**Table 5:** Exact McNemar test between the BiLSTM + Bahdanau model (seed 1) and each system

| Corpus | Comparison    | BiLSTM only | Other only | p         |
|--------|---------------|-------------|------------|-----------|
| LEX    | vs ISRI       | 1734        | 104        | < 1e-300  |
| LEX    | vs Tashaphyne | 1456        | 144        | 2.92e-273 |
| LEX    | vs Khoja      | 1334        | 196        | 4.13e-208 |
| LEX    | vs CRF        | 863         | 137        | 2.52e-129 |
| LEX    | vs BiGRU      | 181         | 199        | 3.83e-01  |
| LEX    | vs Transf.    | 197         | 204        | 7.65e-01  |
| QAC    | vs ISRI       | 1681        | 32         | < 1e-300  |
| QAC    | vs Tashaphyne | 1508        | 66         | < 1e-300  |
| QAC    | vs Khoja      | 1284        | 51         | 1.71e-309 |
| QAC    | vs CRF        | 891         | 40         | 3.46e-210 |
| QAC    | vs BiGRU      | 56          | 71         | 2.14e-01  |
| QAC    | vs Transf.    | 90          | 69         | 1.12e-01  |

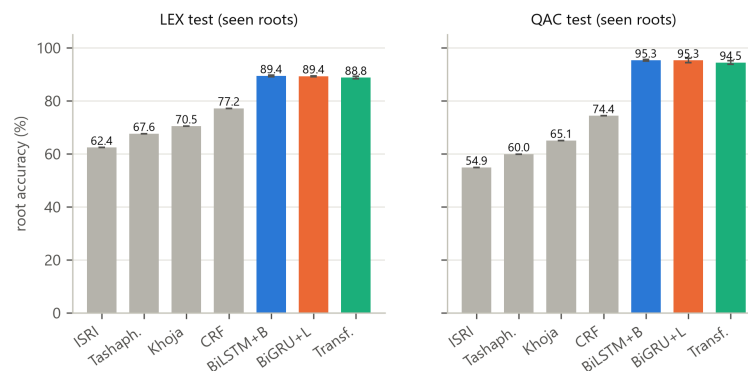


**Figure 4:** Root accuracy of all systems on the LEX and QAC test sets (error bars: s.d. over seeds)

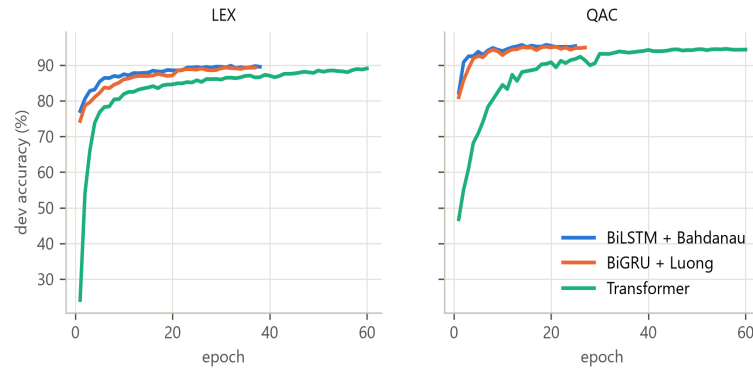


Figure 5: Development accuracy per epoch (seed 1)

### 4.3. Roots never seen in training

Table 6: LEX, unseen-root split: every test root is absent from training

| System         | Acc. (%)   | Len. (%)   | P (%)      | R (%)      | F1 (%)     | CER           |
|----------------|------------|------------|------------|------------|------------|---------------|
| ISRI           | 64.93      | 66.79      | 88.85      | 89.09      | 88.97      | 0.1860        |
| Tashaphyne     | 67.83      | 71.32      | 91.21      | 88.57      | 89.87      | 0.1645        |
| Khoja          | 69.50      | 72.64      | 89.42      | 90.53      | 89.97      | 0.1605        |
| CRF            | 67.15      | 74.30      | 93.30      | 90.61      | 91.94      | 0.1335        |
| <b>BiLSTM</b>  | 78.10±0.52 | 84.04±0.76 | 94.20±0.30 | 92.67±0.44 | 93.43±0.14 | 0.1010±0.0029 |
| <b>BiGRU</b>   | 77.82±0.40 | 83.29±1.10 | 94.18±0.07 | 92.53±0.62 | 93.35±0.31 | 0.1048±0.0028 |
| <b>Transf.</b> | 77.83±0.15 | 83.87±0.17 | 93.94±0.19 | 92.63±0.27 | 93.28±0.06 | 0.1037±0.0023 |

Table 7: QAC, unseen-root split

| System         | Acc. (%)    | Len. (%)    | P (%)      | R (%)      | F1 (%)     | CER           |
|----------------|-------------|-------------|------------|------------|------------|---------------|
| ISRI           | 61.09       | 61.11       | 86.66      | 87.40      | 87.03      | 0.2086        |
| Tashaphyne     | 70.45       | 70.60       | 89.92      | 89.84      | 89.88      | 0.1731        |
| Khoja          | 69.49       | 69.52       | 88.45      | 88.09      | 88.27      | 0.1974        |
| CRF            | 49.15       | 49.65       | 90.88      | 84.16      | 87.39      | 0.2176        |
| <b>BiLSTM</b>  | 62.69±1.77  | 62.98±1.65  | 87.52±0.53 | 87.40±0.52 | 87.46±0.52 | 0.1804±0.0089 |
| <b>BiGRU</b>   | 60.49±6.69  | 60.86±6.78  | 87.17±2.38 | 87.06±2.38 | 87.12±2.38 | 0.2124±0.0414 |
| <b>Transf.</b> | 34.61±11.64 | 35.05±11.63 | 83.20±0.79 | 83.14±0.78 | 83.17±0.79 | 0.3702±0.0857 |

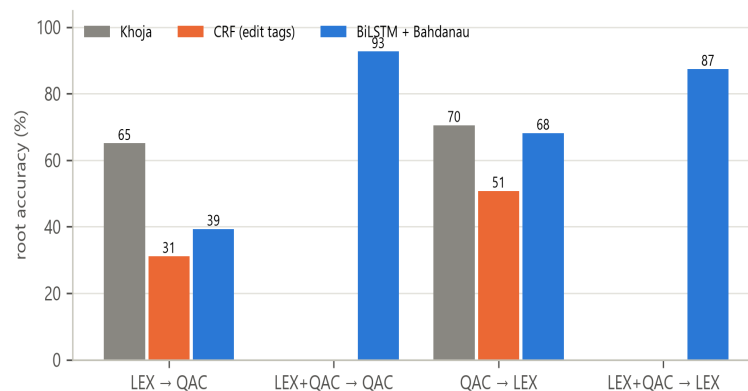
When the root itself is new, the models can no longer draw on a memorised root inventory. On LEX the best model still reaches 78.10%, against 69.50% for the best rule-based system (Khoja) and 67.15% for the CRF. On QAC, however, the best model (BiLSTM + Bahdanau, 62.69%) falls below Tashaphyne (70.45%): with clitics attached and only about 1,300 training roots, the models cannot generalise to new roots as well as affix rules do as show in Table 7. Rule-based extraction therefore remains competitive for genuinely unknown roots in clitic-rich text, and a hybrid that falls back to rules for low-confidence outputs is a natural extension. The Transformer is unstable in this low-data setting (standard deviation 11.64 points across seeds), whereas the recurrent models are not.

### 4.4. Cross-corpus evaluation and combined training

A model trained only on LEX dictionary forms reaches 39.31% on QAC tokens: the lexicon contains no clitics, so the model has never seen و, ف, ب, ل or pronoun suffixes around a stem, whereas Khoja strips them by rule. Training on both benchmarks Table 8, Figure 6 gives 87.39% on LEX and 92.66% on QAC with a single model, i.e. the data-driven model needs clitic-bearing examples, not more rules.

**Table 8:** Training on one benchmark and testing on the other; LEX+QAC is trained on both (test forms of both removed)

| Train→Test    | System     | Acc. (%)   | Len. (%)   | P (%)      | R (%)      | F1 (%)     | CER           |
|---------------|------------|------------|------------|------------|------------|------------|---------------|
| LEX → QAC     | ISRI       | 54.91      | 55.01      | 84.42      | 85.30      | 84.86      | 0.2487        |
| LEX → QAC     | Tashaphyne | 59.97      | 60.26      | 86.61      | 86.20      | 86.41      | 0.2226        |
| LEX → QAC     | Khoja      | 65.07      | 65.44      | 86.32      | 86.81      | 86.57      | 0.2174        |
| LEX → QAC     | CRF        | 31.12      | 31.29      | 73.25      | 86.96      | 79.52      | 0.4070        |
| LEX → QAC     | BiLSTM     | 39.31±1.19 | 39.68±1.15 | 72.59±0.77 | 88.25±0.15 | 79.65±0.44 | 0.4258±0.0116 |
| LEX → QAC     | BiGRU      | 39.30±0.27 | 39.68±0.28 | 71.86±1.05 | 88.06±0.56 | 79.14±0.78 | 0.4335±0.0123 |
| LEX → QAC     | Transf.    | 38.48±0.93 | 38.78±0.99 | 71.96±0.13 | 89.41±0.37 | 79.74±0.22 | 0.4318±0.0015 |
| QAC → LEX     | ISRI       | 62.43      | 64.69      | 88.09      | 88.43      | 88.26      | 0.1967        |
| QAC → LEX     | Tashaphyne | 67.64      | 71.17      | 91.09      | 88.49      | 89.77      | 0.1635        |
| QAC → LEX     | Khoja      | 70.49      | 74.02      | 89.47      | 91.02      | 90.24      | 0.1593        |
| QAC → LEX     | CRF        | 50.76      | 53.07      | 91.21      | 84.64      | 87.80      | 0.2118        |
| QAC → LEX     | BiLSTM     | 68.15±0.37 | 72.44±0.24 | 91.78±0.10 | 87.56±0.10 | 89.62±0.10 | 0.1661±0.0015 |
| QAC → LEX     | BiGRU      | 67.15±0.94 | 71.30±1.06 | 91.31±0.29 | 87.29±0.23 | 89.25±0.25 | 0.1748±0.0047 |
| QAC → LEX     | Transf.    | 64.01±2.18 | 68.09±2.19 | 90.56±0.39 | 86.45±0.38 | 88.46±0.39 | 0.1975±0.0146 |
| LEX+QAC → LEX | BiLSTM     | 87.39±1.20 | 92.60±1.44 | 96.54±0.22 | 95.83±0.55 | 96.18±0.37 | 0.0585±0.0051 |
| LEX+QAC → QAC | BiLSTM     | 92.66±0.48 | 93.09±0.47 | 97.49±0.23 | 97.76±0.19 | 97.63±0.21 | 0.0424±0.0028 |
| LEX+QAC → LEX | BiGRU      | 87.51      | 92.88      | 96.66      | 95.73      | 96.19      | 0.0593        |
| LEX+QAC → QAC | BiGRU      | 93.16      | 93.53      | 97.57      | 97.93      | 97.75      | 0.0384        |
| LEX+QAC → LEX | Transf.    | 86.87      | 92.11      | 96.33      | 95.37      | 95.85      | 0.0652        |
| LEX+QAC → QAC | Transf.    | 91.52      | 91.91      | 97.00      | 97.42      | 97.21      | 0.0466        |

**Figure 6:** Cross-corpus accuracy and the effect of combined training

#### 4.5. Context versus single-word input

Only 1.50% of QAC tokens belong to forms with more than one root, so the isolated-word ceiling on the test set is 99.66%. Adding the neighbouring words changes accuracy by +0.51 points for BiLSTM, + 0.34 for BiGRU and -1.00 for the Transformer as show in Table 9. The recurrent gains are small, one to two standard deviations across seeds, and the Transformer, which has no recurrence to weight the centre word, is distracted by the longer input. Isolated-word extraction is therefore sufficient for Quranic text; context is kept as an option for undiacritised modern homographs Figure 7.

**Table 9:** QAC: word only versus word with its left and right neighbours

| System                       | Acc. (%)   | Len. (%)   | P (%)      | R (%)      | F1 (%)     | CER           |
|------------------------------|------------|------------|------------|------------|------------|---------------|
| <b>BiLSTM (word only)</b>    | 95.33±0.25 | 95.70±0.25 | 98.55±0.08 | 98.54±0.07 | 98.55±0.08 | 0.0245±0.0009 |
| <b>BiLSTM (+ context)</b>    | 95.85±0.35 | 96.21±0.32 | 98.65±0.11 | 98.62±0.11 | 98.64±0.11 | 0.0222±0.0025 |
| <b>BiGRU (word only)</b>     | 95.29±0.81 | 95.68±0.83 | 98.47±0.30 | 98.46±0.30 | 98.47±0.30 | 0.0251±0.0047 |
| <b>BiGRU (+ context)</b>     | 95.64±0.57 | 96.00±0.63 | 98.60±0.21 | 98.57±0.22 | 98.59±0.21 | 0.0233±0.0032 |
| <b>Transf. (word only)</b>   | 94.48±0.68 | 94.85±0.68 | 98.24±0.15 | 98.23±0.14 | 98.23±0.14 | 0.0305±0.0044 |
| <b>Transf. (+ context)</b>   | 93.48±0.82 | 93.82±0.83 | 97.78±0.28 | 97.76±0.27 | 97.77±0.27 | 0.0363±0.0043 |
| Isolated-word ceiling (test) | 99.66      |            |            |            |            |               |

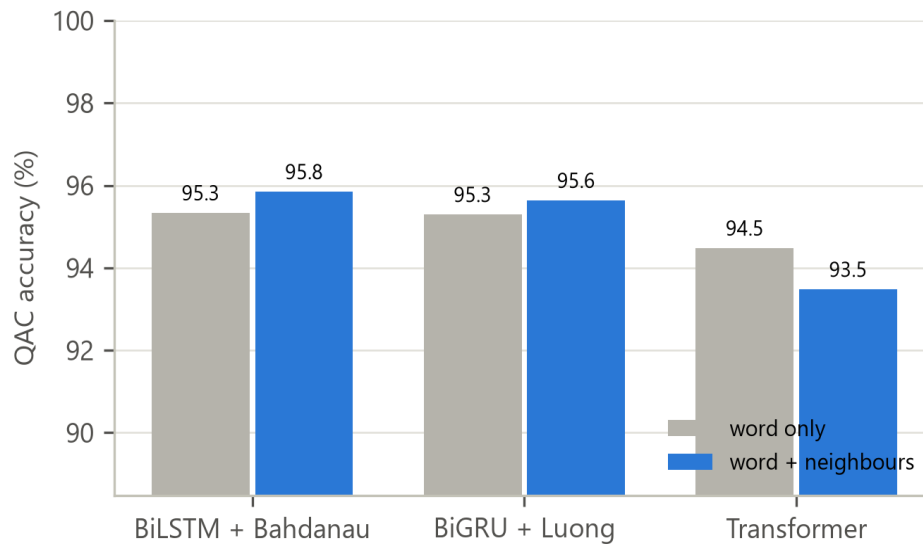


**Figure 7:** Effect of sentence context on QAC

#### 4.6. Ablation study

**Table 10:** Ablation of BiLSTM + Bahdanau on LEX (full model: three seeds)

| Variant                | Acc. (%)   | F1 (%)     | CER           | Params | Epochs |
|------------------------|------------|------------|---------------|--------|--------|
| Full model (beam 5)    | 89.43±0.30 | 96.84±0.11 | 0.0479±0.0012 | 4.32M  | 44     |
| Greedy decoding        | 89.41±0.29 | 96.83±0.11 | 0.0479±0.0015 |        |        |
| No attention (3 seeds) | 89.69±0.32 | 96.96±0.09 | 0.0484±0.0015 | 3.57M  | 40     |
| Hidden 128             | 89.10      | 96.73      | 0.0492        | 1.20M  | 46     |
| Hidden 512             | 89.72      | 96.95      | 0.0463        | 16.35M | 33     |
| 1 layer                | 89.43      | 96.86      | 0.0483        | 2.21M  | 34     |
| 3 layers               | 89.31      | 96.83      | 0.0479        | 6.42M  | 38     |
| Embedding 64           | 89.21      | 96.79      | 0.0488        | 4.11M  | 38     |
| No dropout             | 89.64      | 96.90      | 0.0474        | 4.32M  | 38     |
| 25% training data      | 83.73      | 95.13      | 0.0745        | 4.32M  | 38     |
| 50% training data      | 86.61      | 96.01      | 0.0604        | 4.32M  | 40     |

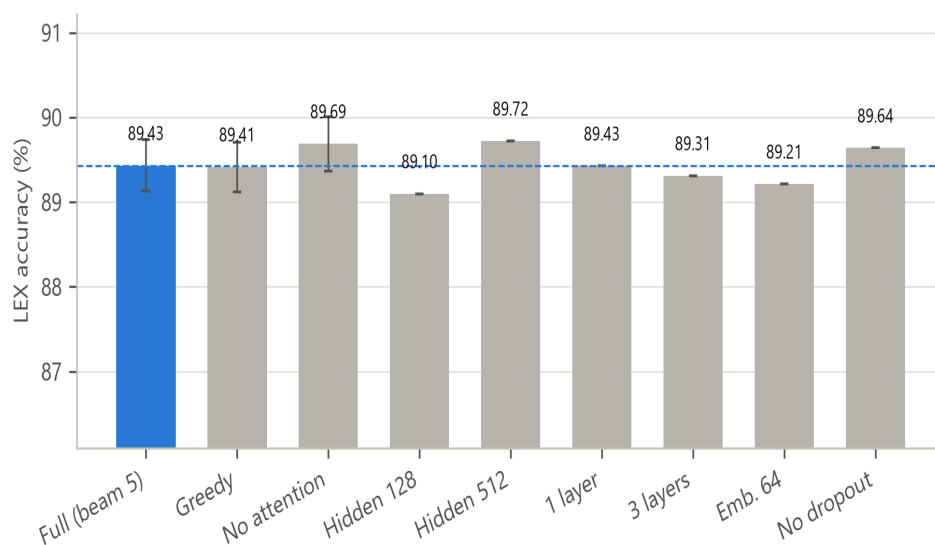


**Figure 8:** Ablation results on LEX

Removing attention does not lower accuracy (89.69% without versus 89.43% with attention, three seeds each); roots are at most a few letters long and the final encoder state already summarises the word, so on this task attention serves interpretability Figure 11 rather than accuracy.

Greedy decoding gives 89.41%, practically equal to beam search, so beam search is not needed in deployment. Capacity matters little: 128 to 512 hidden units give 89.10% to 89.72%, and one to three layers give 89.43% to 89.31%. Halving and quartering the training data gives 86.61% and 83.73% Figure 9: data, not capacity, is the limiting factor Table 10, Figure 8.

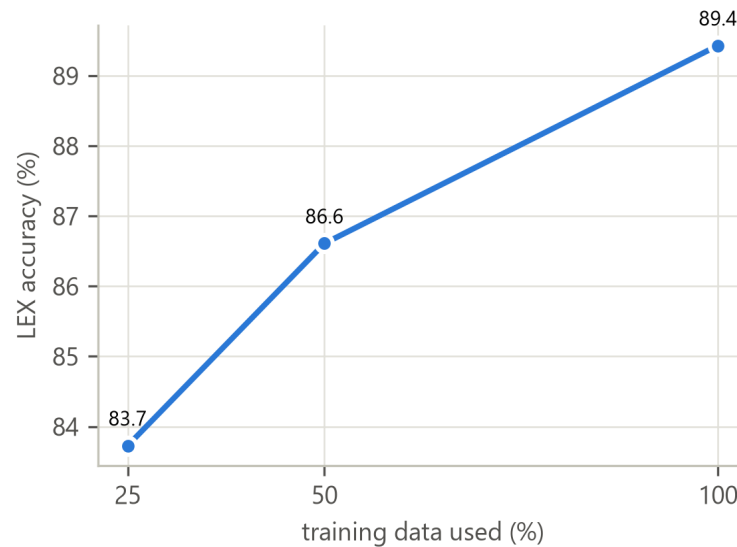


**Figure 9:** Accuracy as a function of the amount of training data

#### 4.7. Stemming (lemma) task

The same architecture, trained with lemmas as targets, serves as a dictionary-form stemmer; Table 11 compares it with Farasa and the Tashaphyne light stem on the same items. The models generate the exact headword for 77.61% of test forms, against 38.63% for Farasa and 23.36% for Tashaphyne. The gap overstates the difference, because the two tools strip affixes to a stem rather than restoring the dictionary form, but the character F1 scores, which give partial credit, show the same ordering. Lemma generation is harder than root extraction (about 12 points lower) because the model must also choose the vowel-bearing pattern of the headword.

**Table 11:** Generating the lexicon headword (lemma) instead of the root on LEX (neural rows: one seed). Farasa and Tashaphyne return stems, which follow different conventions, so their scores are indicative only

| System                | Acc. (%) | F1 (%) | CER    |
|-----------------------|----------|--------|--------|
| Farasa stemmer        | 38.63    | 88.08  | 0.2247 |
| Tashaphyne light stem | 23.36    | 84.88  | 0.2849 |
| <b>BiLSTM</b>         | 77.13    | 95.46  | 0.0841 |
| <b>BiGRU</b>          | 77.61    | 95.59  | 0.0818 |
| <b>Transf.</b>        | 77.56    | 95.63  | 0.0840 |

#### 4.8. Error analysis

Word length separates the systems further Table 12 Khoja degrades on long LEX words (34.29% for nine letters or more), whereas BiLSTM + Bahdanau stays at 88.57% Table 13. The Transformer matches the recurrent model on short and medium words but falls behind on the longest ones (77.14% on LEX, 89.19% on QAC, against 97.30% for BiLSTM); these bins are small ( $n < 40$ ), so the difference is indicative only.

**Table 12:** Accuracy (%) by root class on the seen-root test sets;  $n$  = test items per class; neural model: seed 1

| Corpus | System | sound | hamz. | doubl. | mithal | ajwaf | naqis | laff  | quadr. | loan  |
|--------|--------|-------|-------|--------|--------|-------|-------|-------|--------|-------|
| LEX    | n      | 3081  | 239   | 638    | 263    | 699   | 547   | 100   | 373    | 164   |
| LEX    | Khoja  | 90.36 | 64.02 | 54.55  | 63.50  | 60.37 | 24.31 | 49.00 | 53.89  | 28.66 |
| LEX    | CRF    | 91.24 | 64.44 | 66.93  | 73.00  | 67.95 | 49.54 | 55.00 | 77.75  | 24.39 |
| LEX    | BiLSTM | 97.21 | 77.41 | 88.71  | 84.79  | 80.11 | 77.51 | 85.00 | 81.50  | 62.20 |
| QAC    | n      | 1848  | 351   | 297    | 90     | 884   | 435   | 177   | 12     |       |
| QAC    | Khoja  | 86.74 | 70.37 | 47.47  | 67.78  | 51.47 | 15.17 | 50.28 | 16.67  |       |
| QAC    | CRF    | 85.55 | 59.54 | 64.31  | 58.89  | 73.19 | 53.56 | 68.93 | 83.33  |       |
| QAC    | BiLSTM | 98.27 | 95.73 | 86.53  | 88.89  | 93.55 | 90.57 | 99.44 | 91.67  |       |

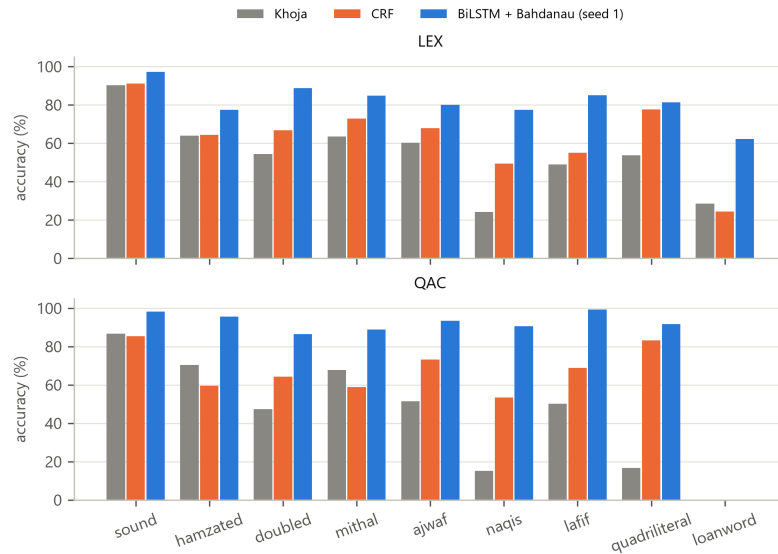


**Figure 10:** Accuracy by root class for Khoja, the CRF and the best neural model

**Table 13:** Accuracy (%) by word length in letters (seen-root test sets; neural models: seed 1)

| Corpus | System  | ≤4             | 5–6            | 7–8           | ≥9           |
|--------|---------|----------------|----------------|---------------|--------------|
| LEX    | Khoja   | 67.59 (n=2613) | 74.04 (n=3151) | 62.90 (n=310) | 34.29 (n=35) |
| LEX    | CRF     | 73.13 (n=2613) | 81.31 (n=3151) | 75.16 (n=310) | 34.29 (n=35) |
| LEX    | BiLSTM  | 84.73 (n=2613) | 92.29 (n=3151) | 93.87 (n=310) | 88.57 (n=35) |
| LEX    | Transf. | 84.65 (n=2613) | 92.95 (n=3151) | 91.29 (n=310) | 77.14 (n=35) |
| QAC    | Khoja   | 54.83 (n=1554) | 73.92 (n=1890) | 63.30 (n=613) | 72.97 (n=37) |
| QAC    | CRF     | 83.91 (n=1554) | 70.16 (n=1890) | 63.46 (n=613) | 72.97 (n=37) |
| QAC    | BiLSTM  | 93.18 (n=1554) | 96.24 (n=1890) | 96.90 (n=613) | 97.30 (n=37) |
| QAC    | Transf. | 93.89 (n=1554) | 95.34 (n=1890) | 94.94 (n=613) | 89.19 (n=37) |

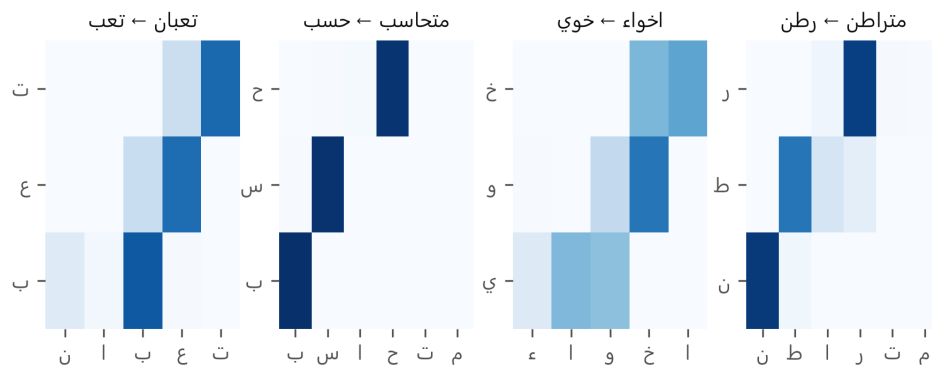
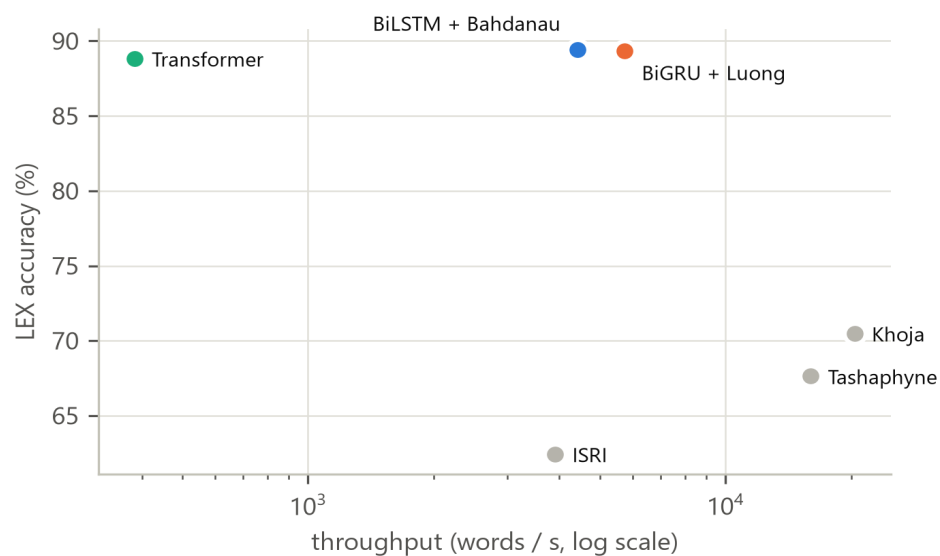
**Table 14:** Error types on the test sets. ‘Invented’ = the predicted root does not occur in the training/development root inventory

| System     | Test items | Errors | Invented (%) | Len. mismatch (%) |
|------------|------------|--------|--------------|-------------------|
| LEX BiLSTM | 6109       | 665    | 23.3         | 26.2              |
| LEX Khoja  | 6109       | 1803   | 44.8         | 34.7              |
| LEX CRF    | 6109       | 1391   | 71.8         | 69.0              |
| QAC BiLSTM | 4094       | 197    | 43.7         | 0.5               |
| QAC Khoja  | 4094       | 1430   | 73.3         | 32.9              |
| QAC CRF    | 4094       | 1048   | 92.7         | 77.8              |

The neural model makes 665 errors on 6109 LEX test words, Khoja 1803. Only 23.3% of the model’s errors are roots outside the known inventory (Khoja 44.8%); most errors are real but wrong roots, typically a weak letter confused with another (ل/ي in ajwaf and naqis roots), hamza versus alif at the first position, or a doubled root read as a weak one Table 14. The hallucination risk noted for seq2seq lemmatisers [20] is therefore limited here and could be removed entirely by constraining the beam to known roots Table 15. Loanwords remain the weakest class because their long consonant skeletons are rare in training; defective (naqis) roots show the largest gain over rules, because the model restores the final weak letter that Khoja cannot recover. Figure 11 shows attention maps: each root letter attends to its position in the word, skipping pattern and affix letters.

**Table 15:** Examples of errors of the neural model on LEX

| Category      | Word       | Gold     | Predicted | Pred. root |
|---------------|------------|----------|-----------|------------|
| loanword      | اجر        | ءاجر     | جري       | real root  |
| loanword      | اجرة       | ءاجر     | ءجر       | real root  |
| hamzated      | ابط        | ءبط      | بطط       | real root  |
| hamzated      | اتن        | ءتن      | ءنن       | real root  |
| naqis         | مات        | ءتي      | موت       | real root  |
| hamzated      | اجر        | ءجر      | جري       | real root  |
| hamzated      | خذ         | ءخذ      | خذي       | real root  |
| loanword      | ارستقراطية | ءرستقراط | ءرستقراطي | invented   |
| quadriliteral | ارنية      | ءرنب     | رنب       | real root  |
| quadriliteral | ارنيات     | ءرنب     | رنب       | real root  |
| naqis         | تاسيا      | ءسو      | ءسي       | real root  |
| naqis         | اسيانه     | ءسي      | ءسيانه    | invented   |
| loanword      | الحقوانة   | ءحقوان   | حقو       | real root  |
| quadriliteral | يوقلم      | ءقلم     | قلم       | real root  |

**Figure 11:** Attention weights of BiLSTM + Bahdanau for four test words (input read right-to-left)**Figure 12:** Accuracy versus throughput (words per second, GPU for neural models, CPU for rules)

## 5. Discussion

Interpretation. A small character model learns the root–pattern mapping of modern Arabic well enough to outperform hand-written rules by a wide margin on unseen forms of known roots, and it does so most clearly where rules are weakest: defective (naqis), doubly weak (lafif) and doubled roots. The advantage shrinks on unseen roots, is lost to affix rules on unseen roots in clitic-bearing Quranic text, and disappears across corpora when the training data lack clitics; combined training restores it. This locates the remaining difficulty in data coverage rather than in the architecture.

Complexity versus accuracy. Rule-based extractors process thousands to tens of thousands of words per second on a CPU and need no training data; the recurrent models run at a few thousand words per second on a modest GPU Figure 12 and require annotated pairs. Bahdanau and Luong attention perform alike, attention itself adds no accuracy, and the Transformer brings no gain on these short inputs, trails on the longest words Table 13 and is unstable when training roots are few.

Context versus isolation. Because root ambiguity after normalisation is rare in both benchmarks, isolated-word input is nearly sufficient; context would matter more for undiacritised homographs in running modern text and dialects.

Limitations. LEX pairs come from dictionary entries, so the modern benchmark lacks clitics and running-text frequencies; QAC is Classical Arabic; 7.5% of Quranic verses were dropped by the alignment; loanword ‘roots’ follow the lexicon’s convention; the models do not beat rules on unseen Quranic roots; ablation variants other than no-attention, the lemma task and the BiGRU/Transformer rows of combined training use one seed. Farasa and Tashaphyne return stems, so their lemma scores depend on conventions.

## 6. Conclusion and Future Work

We framed Arabic root extraction as character-level sequence generation and showed on two new, leakage-free benchmarks that attention-based encoder–decoders reach 89.43% (modern Arabic) and 95.33% (Quranic) root accuracy, far above Khoja, ISRI, Tashaphyne and a CRF on the same items, with the largest gains on defective (naqis), doubly weak (lafif) and doubled roots. For roots absent from training, rules remain competitive on clitic-bearing text. Future work will extend the training data with clitic-bearing modern text and dialectal Arabic, constrain decoding to a root inventory, fall back to rules for low-confidence outputs, combine the model with pre-trained contextual encoders and large language models for ambiguous words, and deploy it as a real-time normaliser for Arabic search engines.

### Article Information

**Ethical Approval:** This study was approved by the Scientific Research Ethics Committee at the College of Medical Sciences / Jaber Ibn Hayyan University of Medical and Pharmaceutical Sciences under approval number 8, dated 1/4/2026.

**Informed Consent:** Written informed consent was obtained from all participants before their inclusion in the study.

**Data Availability:** The datasets generated and/or analyzed during the current study are available from the corresponding author upon reasonable request.

**Conflict of Interest:** The author declares that they have no conflict of interest.

**Funding:** No external funding was received for this study.

**AI Use Disclosure:** No generative artificial intelligence tools were used in the preparation of this manuscript.

## References

- [1] N. J. Thalji. A survey of Arabic root extraction algorithms. *Iraqi J. Sci.*, 65(4):2304–2321, 2024. URL <https://doi.org/10.24996/ijs.2024.65.4.43>.
- [2] Y. Farhan, M. Tareq, B. Shannaq, O. Ali, S. Almaqbali, and V. P. Sriram. MORSE-QE: A morphology-aware, embedding-driven framework with root extraction for Arabic dialectal query expansion. *Informatica*, 49(20), 2026. URL <https://doi.org/10.31449/inf.v49i20.9628>.
- [3] H. M. F. Saleh, A. Salah, M. Mahdi, and H. Elfiqui. An empirical evaluation of raw text, light stemming, and root extraction for Arabic plagiarism detection. *J. Comput. Commun.*, 5(2):20–29, 2026. URL <https://doi.org/10.21608/jocc.2026.467541.1112>.
- [4] S. Khoja and R. Garside. Stemming Arabic Text. Technical report, Computing Department, Lancaster University, Lancaster, U.K., 1999.
- [5] K. Taghva, R. Elkhoury, and J. Coombs. Arabic stemming without a root dictionary. In *Proc. ITCC*, pages 152–157, 2005. URL <https://doi.org/10.1109/ITCC.2005.90>.
- [6] T. Zerrouki. Tashaphyne: A Python package for Arabic light stemming. *J. Open Source Softw.*, 9(93):6063, 2024. URL <https://doi.org/10.21105/joss.06063>.
- [7] A. Alaswaad and B. Minaei-Bidgoli. An intelligent deep learning for Arabic stemming and morphological classification. *Bull. Electr. Eng. Inform.*, 15(4):3748–3761, 2026.
- [8] S. Afartass, F. Ataa Allah, and K. Minaoui. AraBART-based Arabic lemmatization. *WSEAS Trans. Inf. Sci. Appl.*, 23:1, 2026. URL <https://doi.org/10.37394/23209.2026.23.1>.
- [9] S. Belayachi and A. Mazroui. A hybrid deep neural network and morphological knowledge to enhance Arabic lemmatization. *Int. J. Adv. Soft Comput. Appl.*, 18(1):227–241, 2026. URL <https://doi.org/10.15849/ijasca.v18i1.33>.
- [10] N. Abdelrehem, Y. Hifny, and H. Shamardan. Enhancing Arabic morphological segmentation with deep recurrent neural architectures. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 25(10):87, 2026. URL <https://doi.org/10.1145/3846165>.
- [11] Y. Y. Alakeel, C. Qwaider, H. Aldarmaki, and S. Alqahtani. Morphemes without borders: Evaluating root–pattern morphology in Arabic tokenizers and LLMs. In *Proc. LREC*, pages 11787–11799, 2026. URL <https://doi.org/10.63317/3bk9j7feheoq>.

- [12] L. S. Larkey, L. Ballesteros, and M. E. Connell. Light stemming for Arabic information retrieval. In A. Soudi, A. van den Bosch, and G. Neumann, editors, *Arabic Computational Morphology*, pages 221–243. Springer, Dordrecht, 2007.
- [13] D. Namly and K. Bouzoubaa. An innovative Arabic light stemmer developed using a hybrid approach. *Int. J. Electr. Comput. Eng.*, 15(2):2356–2363, 2025. URL <https://doi.org/10.11591/ijece.v15i2.pp2356-2363>.
- [14] A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak. Farasa: A fast and furious segmenter for Arabic. In *Proc. NAACL Demos*, pages 11–16, 2016. URL <https://doi.org/10.18653/v1/N16-3003>.
- [15] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*, pages 282–289, 2001.
- [16] I. Akhmetov, B. I. M. Al-Raba’a, A. Gelbukh, R. Mussabayev, and A. Krassovitskiy. Revisiting Arabic morphology: A machine-learning-based approach to stemming and root character permutations with a publicly available open-source implementation. *Comput. Syst.*, 30(1), 2026. URL <https://doi.org/10.13053/cys-30-1-6322>.
- [17] Y. Maneetah. A comparative analysis of stemming strategies in Arabic broken plural classification using logistic regression. *Tobruk Univ. J. Eng. Sci.*, 6(2):166–176, 2025. URL <https://doi.org/10.64516/kmkhyp28>.
- [18] M. Jarrar, D. Akra, and T. Hammouda. Alma: Fast lemmatizer and POS tagger for Arabic. *Procedia Comput. Sci.*, 244:378–387, 2024. URL <https://doi.org/10.1016/j.procs.2024.10.212>.
- [19] M. M. Fouad, A. Mahany, and I. Katib. Masdar: A novel sequence-to-sequence deep learning model for Arabic stemming. In *Proc. IntelliSys, AISC*, pages 363–373. Springer, 2019. URL [https://doi.org/10.1007/978-3-030-29513-4\\_26](https://doi.org/10.1007/978-3-030-29513-4_26).
- [20] M. Saeed and N. Habash. Lemmatization as a classification task: Results from Arabic across multiple genres. In *Proc. EMNLP*, pages 30002–30017, 2025. URL <https://doi.org/10.18653/v1/2025.emnlp-main.1525>.
- [21] W. Antoun, F. Baly, and H. Hajj. AraBERT: Transformer-based model for Arabic language understanding. In *Proc. OSACT Workshop, LREC*, pages 9–15, 2020.
- [22] M. Abdul-Mageed, A. Elmadany, and E. M. B. Nagoudi. ARBERT MARBERT: Deep bidirectional transformers for Arabic. In *Proc. ACL-IJCNLP*, pages 7088–7105, 2021. URL <https://doi.org/10.18653/v1/2021.acl-long.551>.
- [23] M. Reda, S. Khalifa, R. Marzouk, and N. Habash. YallaMorph: A benchmark for evaluating Arabic morphological generation in large language models. arXiv preprint, 2026. URL <https://doi.org/10.48550/arXiv.2609.10153>.
- [24] A. M. Omar. *Mu’jam al-Lugha al-‘Arabiyya al-Mu’asira (Lexicon of the Modern Arabic Language)*. Alam al-Kutub, Cairo, 2008. Digital release: T. Alsawi, “Arabic Lexicons,” Kaggle, 2024, licensed under CC BY 4.0.
- [25] K. Dukes and N. Habash. Morphological annotation of Quranic Arabic. In *Proc. LREC*, 2010.
- [26] Tanzil Project. Quran text (simple-clean). 2026. URL <https://tanzil.net>.
- [27] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Proc. NeurIPS*, pages 3104–3112, 2014.
- [28] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997. URL <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [29] K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proc. EMNLP*, pages 1724–1734, 2014. URL <https://doi.org/10.3115/v1/D14-1179>.
- [30] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*, 2015.
- [31] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proc. EMNLP*, pages 1412–1421, 2015. URL <https://doi.org/10.18653/v1/D15-1166>.
- [32] A. Vaswani et al. Attention is all you need. In *Proc. NeurIPS*, pages 5998–6008, 2017.
- [33] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015.
- [34] Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2): 153–157, 1947. URL <https://doi.org/10.1007/BF02295996>.