Learning Stress in Arabic Low-Resource Settings

Abdelrahim 'Abed' Qaddoumi[1] Salam Khalifa[1,2] Jordan Kodner[1]
Jeffrey Heinz[1] Owen Rambow[1]
[1]Institute for Advanced Computational Science and Department of Linguistics,
Stony Brook University
[2]Computational Approaches to Modeling Language (CAMeL) Lab, NYU Abu Dhabi
{first.last}@stonybrook.edu

Abstract

We predict lexical stress in Arabic varieties from syllable structure, modeling stress assignment as generation: given an unstressed input, the system outputs a stress-marked word. We compare four approaches: a grammar induction algorithm (BUFIA), a transformer-based neural network, a rule-based method derived from linguistic literature, and a frequency baseline. The models are evaluated across several low-resource settings by varying the training data size by words, structural type, and syllable count. BUFIA outperforms the neural network, especially when data are scarce. This points to grammar induction as an interpretable and sample-efficient approach for learning stress.

1 Introduction

Lexical stress is the emphasis of one syllable in a word. Stress may be contrastive, as in many American English verb-noun pairs (recórd (verb) vs. récord (noun), prodúce (verb) vs. próduce (noun)), or may be completely predicted from phonological structure, as is the case for the Arabic varieties discussed in this paper. Closely related language varieties may vary in their stress patterns, as is the case for Arabic, and for English as well. For example, British English may the stress *weekend* second syllable, whereas American English stresses the first. Southern varieties of American English may employ initial stress in words like *police*, where other varieties stress the final syllable.

In addition to its roles in linguistics *per se*, stress can enhance engineering applications like automatic speech recognition and text-to-speech systems (Wang and Seneff; Ludusan et al., 2012; Zou et al., 2021; Geneva et al., 2023). Building on these applications, research sometimess focuses on automatic stress prediction Dou et al. (2009); van Esch et al. (2016).

In linguistics, stress in Arabic has arguably received the most attention among languages outside English (Watson, 2011). This attention is due to Arabic varieties providing an excellent environment for investigating stress. All varieties exhibit stress and share similarities such that stress location is a function of both syllable shape and position. Yet, they differ in the distribution of syllable types, the leftmost extent of stress (third or fourth syllable from the right),[1] and the degree to which lexical information beyond phonology may affect stress (Watson, 2011).

Computationally, Arabic presents an interesting case because of its diglossia (i.e., the use of two varieties, one for spoken and another for written). Modern Standard Arabic (MSA), the written variety, has good resources, while most spoken varieties have few. This resource imbalance creates a challenge for practical modeling in data-scarce settings. However, since Arabic regional varieties are related and vary in predictable ways, they provide ample ground for cross-variety transfer techniques. Still, computational research for Arabic stress has been limited.

In this paper, we study stress prediction in three Arabic varieties: Egyptian, Sudanese, and Jordanian. As Arabic stress is strongly constrained by syllable structure and is determined within a limited window near the right edge of the word, we represent each word as a sequence of CV syllables (e.g., <كاتب> [ˈkaa.tib] 'writer' is represented as `cvv.cvc`). This allows us to generate the stressed version of the word by capitalizing the stressed syllable (i.e., `CVV.cvc`).

We compare four approaches: (1) BUFIA is a grammar induction approach that learns constraints and compiles them into a finite-state acceptor/transducer (Chandlee et al., 2019), (2) Neural transducer, a transformer-based character-level

[1]We use "left" to refer to the start of words and "right" to reference the end, keeping in line with most literature on stress. Our data consists of left-to-right IPA transcriptions rather than right-to-left Arabic script.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 262–279

encoder–decoder (Wu et al., 2021), (3) a strong baseline rule-based implementation derived from the literature in descriptive linguistics on each variety's stress system, and (4) Majority baseline predicting the most frequent stress position.

We evaluate all approaches in low-resource conditions by varying the size of the training data across three distinct measures: random word-based sampling, structural-type sampling (where the training set is constructed by progressively including larger proportions of syllable-structure types available in the language), and length-based sampling (where words of increasing syllable count expand the training dataset). This evaluation design enables us to test whether a model can generalize to unseen structural types and longer words, moving beyond a limited number of words encountered during training.

Our results indicate that BUFIA is a viable modeling strategy for phonological generalizations when data are limited, and interpretability is desirable. The paper makes the following contributions: it creates stress-marked datasets for Jordanian and Sudanese, compares BUFIA and a neural transducer in multiple low-resource settings, and develops a method for testing generalization using linguistic data.

The remainder of the paper is organized as follows: §2 reviews relevant linguistic background; §3 describes the grammar induction algorithm (BUFIA); §4 presents the datasets; §5 and §6 detail the experimental setup; §7 reports results; and §8 provides error analyses and discussion.

2 Linguistic Background

2.1 Computational Stress Learning

Computational learning of stress has a long history in linguistics. Dresher and Kaye (1990) frame learning stress as a setting in which a small number of metrical parameters are set in response to phonological cues. Gupta and Touretzky (1991) explore whether a perceptron can be trained to predict stress, demonstrating one of the earliest statistical approaches to learning stress. Goldsmith (1994) approach the study of accent systems using dynamic computations. In the early 2000s, with the rise of Optimality theory, researchers studied the problem through the lens of constraints and their rankings (e.g., (Tesar and Smolensky, 2000)).

Hayes and Wilson (2008) uses weighted constraints in a Maximum Entropy (MaxEnt) phonotactic learner to learn stress, among other phonological phenomena. Heinz (2009) argue that using finite-state machines and simple learners that attest to stress systems can be characterized by local properties and learn from their local patterns. Recently, many studies have examined stress using statistical approaches such as neural networks (Prickett and Pater, 2025) and MaxEnt (Lee et al., 2025). While our work shares the same problem as we are trying to learn stress, we differ in the goal of the research, as we are trying to use stress as an application to put a lens on the difference between two modeling approaches, one being neural networks and the second being grammar induction algorithms.

2.2 Arabic Varieties

While often treated as a single language, modern Arabic is actually a collection of regional varieties that differ in their phonological patterns. Our study focuses on three varieties: the Ammani variety of Jordanian Arabic, the Khartoum variety of Sudanese Arabic, and the Cairene variety of Egyptian Arabic. While this choice was constrained by data availability, the three varieties provide good coverage of Arabic stress patterns. We use the existing resources when available for the varieties described in Section 4. Data was checked by the first and second authors, who are native speakers of Ammani and Sudanese respectively.

Jordanian Arabic (JOR), spoken by around 12 million people in Jordan (World Bank, 2024), falls within the higher level grouping of Levantine varieties (Eberhard et al., 2025). Jordanian, in turn, is comprised of sedentary and Bedouin varieties (Palva, 1984). The former may be further divided into Ammani (spoken in the capital region), Northern Jordanian, Southern Jordanian, and Aqaba (Herin et al., 2021).

Within Egyptian Arabic (EGY), which has about 83 million speakers (Eberhard et al., 2025), we focus on Cairene, the urban variety of the national capital. This variety, with approximately 20 million native speakers, is unusual among regional varieties for its prominence in the media, which makes it widely understood across the Arab world (Khalil, 2011).

Sudanese Arabic is spoken by about 41 million people in Sudan, South Sudan, and Eritrea (Eberhard et al., 2025). It is usually grouped with Egyptian Arabic in the Egypto-Sudanic or Nile Valley category (Versteegh, 2014). According to Gasim

(1965), there is a difference between the sedentary dialects along the Nile and the pastoralist dialects in western Sudan. Most phonological research focuses on the prestige variety, known as Khartoum Arabic or Sudanese Standard Arabic (Mustapha, 1982; Manfredi, 2015).

2.3 Stress and Syllables

Stress is an abstraction that captures a constellation of distinct physiological phenomena. Different languages employ different combinations of these phenomena to indicate stress, including but not limited to loudness, pitch, and duration (Ladefoged and Johnson, 2014).

Stress predictability differs by language: it can be lexically specified, in which case speakers memorize the stress pattern for each word, or phonologically predictable from syllable weight or position. It can also be a mixture of lexical and phonological features. Finnish is a language with phonological stress (Suomi et al., 2009), whereas Tagalog has contrastive lexical stress (e.g., words differ only in stress) (Schachter and Otanes, 1983), and according to Vendelin (2010) Spanish stress is a hybrid pattern as it is lexically contrastive but constrained to a right-edge window with a preference for penultimate stress (Gordon and van der Hulst, 2020).

In systems with predictable stress, stress can be assigned mechanically (Hayes, 1995). Arabic varieties' stress patterns are phonological, although they differ in the details of stress assignment. Stress placement is usually computable by looking at the heaviest syllable from the rightmost edge of the word (assuming a left-to-right transcription for simplicity) (Watson, 2002; Broselow, 2017).

Stress may be mechanically assigned to specific syllables in a word according to their syllable weight. To explain this, we provide a brief review of syllable structure. See (Goldsmith, 2011) for a more thorough introduction. We assume the standard decomposition of a syllable into onset-nucleus-coda. Syllables must contain a nucleus, which is usually a vowel (`v`), and they may include some number of consonants (`c`) in the onset before the nucleus and/or in the coda after the nucleus. A long vowel is represented by `vv`.

Languages differ in which combinations they permit. All Arabic varieties require that each syllable contain an onset, so the minimal licit syllable is `cv` (Broselow, 2017). The number of segments in a syllable's nucleus and coda determines its weight (Broselow et al., 1997). Syllables with only a short vowel in the nucleus and a `c` in the onset, `cv`, are light, those with a long vowel or a short vowel and a single coda consonant are heavy (`cvv`, `cvc`), and those with a long vowel and coda or multiple coda consonants are super-heavy (`cvvc`, `cvcc`, `cvvcc`, etc.). Some varieties may treat word-final syllables differently from other positions, for example, treating a word-final `cvc` syllable as light (Hayes, 1995).[2]

There are two further complexities of stress which our system does not account for. First, our representations for Arabic do not include secondary stress. It has been argued that Arabic lacks secondary stress (Al-Jarrah, 2002), and it is not included in our datasets, but see Becker (2022). Second, stress may interact with phrase- and sentence-level prosodic patterns, but this is not applicable to our model, which runs on word lists.

3 BUFIA

This section describes the Bottom Up Factor Inference Algorithm (BUFIA; Chandlee et al., 2019; Rawski, 2021; Swanson et al., 2026). This algorithm identifies a finite set of constraints that are satisfied by the training data.

Words are represented as mathematical structures containing substructures (Rogers and Lambert, 2019). Words are well-formed only if all of the substructures they contain are well-formed. Conversely, the presence of any ill-formed substructure in a word means the word itself is ill-formed. Consequently, a constraint is just an ill-formed structure. For example, consider the formal language $0(ba)^*0$, which represents words as strings of a, b with word boundaries explicitly marked with 0. A grammar for this language can be given by the finite set of ill-formed structures $0a, aa, bb, b0$. As constraints, these structures pick out the language whose words do not begin with a, do not contain aa substrings, do not contain bb substrings, and do not end with b.

BUFIA works with any representational schema for words, provided the schema is fixed in advance. For instance, instead of representing b as an atomic symbol, it could be defined as a vector of feature values such as [+stress, +consonant, −vowel]. In this way, a constraint banning adja-

[2] For more information about the allowed syllable structure, see Appendix Table 8, which lists each variety's licit syllable structures sorted by their weights.

cent stress in words could be given as the substructure [+stress][+stress].

BUFIA is designed to find the structures *absent* in data. It can do this because the structure containment relation partially orders the space of logically possible constraints. It conducts a breadth-first traversal of this space from the bottom up and checks whether the structures it encounters are present in the training data. If not, it adds them to the grammar it eventually outputs. It then also knows that all structural constraints that contain the found one need not be checked. BUFIA proceeds in this way until a stopping condition has been met. In this paper, the stopping condition was set to limit the length to a maximum of four segments. Chandlee et al. (2019) proves that BUFIA returns a grammar whose language is (1) the smallest one in the class of languages consistent with the data, and (2) contains the most general constraints of any other grammar in the class satisfying (1).

Appendix Table 6 shows an example of the constraints for one Jordanian BUFIA result. The table is split into different sections. For example, we have [][+wb] and [+wb][], which means that word boundaries only occur at word edges. Then, we have [+vowel][+vowel][+vowel], meaning there can not be three vowels in one sequence.

4 Data

The Egyptian data comes from the Egyptian Colloquial Arabic Lexicon (Kilany et al., 2002). The lexicon is drawn from 140 CALLHOME telephone conversations of native speakers of Cairene Egyptian Arabic. Each of its 51,202 data points provides a surface (spoken) form, its length in syllables, and the stress pattern indicating which syllable is stressed. Since the actual syllabification is not provided, use Simple Syllabify tool (Kodner, 2016), which identifies syllable nuclei and then inserts syllable boundaries following the Maximum Onset Principle (Goldsmith, 2011). The syllabifier, which we evaluate in (Qaddoumi et al., 2026), performed well across several Arabic varieties, achieving accuracies of 99% on EGY, 98% on JOR, and 97%+ on other available varieties.

We collected the Jordanian data by scraping all entries for South Levantine Arabic on Wiktionary.[3] This initially yielded around 2,782 entries, which included IPA transcriptions indicating both syllabification and stress, which we then filtered through deduplication, which ultimately left us with 2,082 words.

We collected the Sudanese data by extracting all the examples used in (Hamid, 1984). The data was collected and verified by the second author. There were 756 entries in total after we removed phrases that have two stresses or more, since the rules of a phrase's stress are different from lexical stress. In addition to this, we remove any words missing stress annotation.

4.1 Preprocessing

We preprocess the data to remove errors and inconsistencies introduced by the original annotation and to prepare it for BUFIA and the neural network. Wiktionary in particular, while an overall reliable source, is crowdsourced, and is subject to a degree of noise (Sakunkoo and Sakunkoo, 2025). We remove duplicates and entries with transcriptions containing only one or two characters, since they are trivial. We further standardize transcriptions as follows: Long vowels are represented as a repeated character, for example, <كاتب> [ka:.tib] 'writer', will be [kaa.tib]. Geminates are represented by repeating the same character, for example, <كتّب> [kat:ab] 'he made x write', will be represented as [kat.tab].

When an entry provides multiple IPA transcriptions, we retain each distinct transcription as a separate surface form for that variety, for example, <شُوكُولَاتَة> [ʃu.kuu.laa.ta] or [ʃu.ku.laa.ta] 'chocolate'. For the cases where one segment is presented as optional in a single transcription, we always include the optional segment, as in the following example, where <اسمر> /(ɪ)smar/ 'dark' is treated as [ɪs.mar].

All superscripts such as 'ˤ' and subscripts such as '̪' are removed as well, since they are not themselves segments and they correspond to annotations or to phonetic distinctions which do not affect Arabic stress (Association, 1999), for example, the tie bar '͡' and pharyngeal diacritics 'ˤ' in <آية الله> /ʔaa.ja.tu͡lˤ.lˤaah/ 'sign of God'. We also remove [ ] and / /.

5 Methodology: Problem Formulations

General Problem Formulation Our task is to predict which syllable in a word receives primary stress. The input is a phonetic transcription where the vowels are automatically replaced to `v` and the consonants to `c`. The usage of CV sequences is mo-

[3] https://www.wiktionary.org/

tivated by the fact that stress assignment in Arabic and other languages is mainly governed by syllable weight, which is determined by syllable structure in terms of vowels and consonants rather than the particular identities of these segments.

We define the problem as: given a word **x** consisting of **n** syllables each represented as a sequence of characters `c` (consonants) and `v` (vowels), a **generator** should output the word with the stressed syllable marked. We adopt the convention of indicating the stressed syllable with upper case `C` and `V`. For example, given the input $\mathbf{x}_g =$ `c v c v v c`, the output should be $\mathbf{y} =$ `c v C V V C`.

To test the capabilities of BUFIA and NN, we evaluate the models under three sampling paradigms. Within each same random seed and the specific sampling paradigms the models are evaluated on identical splits. The paradigms differ in how the training, dev, and test set are constructed and each is designed to test a different kind of generalization, thus it will give us a deeper insight into the nature of generalization. However, since the dev and test sets differ for the three sampling paradigms, the results from different sampling paradigms cannot be compared. Indeed, we are not trying to compare the three paradigms; rather, we are using them to perform different comparisons of the two learning algorithms we are investigating (BUFIA and neural learning).

5.1 Data Splits

Across all sampling paradigms we split the data described in Section 4 following a 80/10/10 train/dev/test split. Then we test both models on the full size of the dev and test datasets. These splits are done 5 times with different seeds.

Details about training set sizes for all sampling strategies are provided in Table 9 and Appendix Tables 10, 11, 12, 13, 14, and 15.

Sampling Paradigm 1: Random word sampling (data-quantity scaling). This paradigm measures how well models perform as we scale the training data quantity. The data is randomly picked without any constraints. Test and dev sets are drawn first, at random, fixed for each seed, and each is 10% of the total data. We then construct seven *nested* training subsets at approximately 1% of the training data, then 3%, 6%, 13%, 25%, 50%, and finally 100% of the training partition. For exact numbers per dataset, check Table 9.

SC	#CVTs	Test OOV	Dev OOV
≤2	34	51.16%	41.67%
≤4	80	20.93%	16.67%

Table 1: The percentage of CV templates represented in the test and dev but not included in the train set for the sampling by syllable counts (Paradigm 2). The SC column indicates the syllable counts in training subset. The column #CVTs is the number of CV structures present in training set, while Test and Dev OOV indicate the percentage of CV structures not present in those splits which are absent from the training set. These numbers are for seed 21 for Jordanian just to illustrate the distribution of data.

Sampling Paradigm 2: Sampling words by their syllable count (generalization to unseen lengths). This sampling approach is sensitive to words' length syllables, in order to evaluate models' ability to generalize to unseen lengths.

To generate the splits for this experiment, words are sorted by their lengths in syllables. For example, the full JOR dataset contains 328 monosyllabic, 1,356 disyllabic, 382 trisyllabic words, and 15 words with four syllables. Due to the limited number of four-syllable words in Jordanian, we decided to group them with trisyllabic words.

Sampling then proceeds as follows: First, we ensure that the test and dev datasets sample all lengths by sampling one word of each length for both partitions. Second, sampling continues until the full dataset is exhausted, ensuring an approximately 80/10/10 split.

The training set is then subsampled to produce learning curves as for Paradigm 1. However, subsampling is sensitive to words' syllable count. We first sample all monosyllabic and disyllabic words. We then create a larger superset by sampling word trisyllabic words, and so on, until finally the entire training data is included. These subsamples create scenarios where dev and test contain words with syllable lengths unattested in training.

The training set is again split by syllable count. We start with sampling words made of 1 or 2 syllables counts. Then in the next step, we sample from 3 syllables increasing syllable counts until it reaches as many as we can obtain from the training data. Note that there might be syllable counts that are present in test and dev but that do not exist in the training. In Table1, we can see an example for the out-of-vocabulary distribution for seed 21 for Jordanian.

Size (%)	#CVTs	Test OOV	Dev OOV
1	7	93.75%	93.75%
3	7	93.75%	93.75%
6	8	92.5%	92.5%
13	13	87.5%	87.5%
25	20	81.25%	81.25%
50	42	60%	58.75%
100	58	41.25%	41.25%

Table 2: The percentage of CV templates represented in the test and dev but not included in the train set for the Sampling by CV structures. The size column represent the training subset size, #CVTs is the number of CV structures present in training set, and Test and Dev OOV represent the percentage of CV structures not present in training set. These numbers are for seed 21 for Jordanian just to illustrate the distribution of data.

Sampling Paradigm 3: Sampling words by their CV structures (generalization to unseen templates). This paradigm evaluates models' ability to generalize to unseen syllabic structures. We define a unique *structure type* as a pattern of `c` and `v` segments, ignoring stress. For example, `cv.cvc`, `cvc.cvc`, and `cvc.cvvc` are three distinct structure types. While `CV.cv.cv`, `cv.CV.cv`, and `cv.cv.CV` are one structure type.

To generate the data splits for this paradigm, we begin by sorting words by their structure types. For example, in JOR there are 271 words with the `cv.cvc` structure type, 265 with the `cvc.cvc` structure type, 129 with the `cvc.cvvc` structure type, and so on

Parallel to the sampling for Paradigm 2, we ensure that dev and test contain at least one instance of every structure type present in the data set before regular sampling. One caveat to this constraint is that a structure type that only occurs once in the full data set will only occur in dev or test for a given random seed, and not both.

The training data is once again subsampled, this time in a way that is sensitive to syllable types. We start by selecting a number of structure types that generate approximately 1% of the whole dataset, then 3%, and so on, until we reach the full size of the training dataset. This number varies; for example, we could sample 17 entries with 3 structure types in the first step, or 129 entries with 1 structure type. The number of samples depends on the 1% threshold applied to the entire dataset.

For example, if we have 1500 words in training dataset and we are sampling and we select a structure type with 129 entries, which is larger than 1% of the training dataset, we stop sampling and start creating the next split which is 3%. Now, since we already have 129 words in the training data then sampled structure type will stop the sampling and move to the next step, since 129 is larger than 45. We keep repeating this until we have the 7 steps. For the exact number of entries, and structure types, check Appendix Tables 10, 11, and 12

6 Methodology: Models and Baselines

Models and Baselines We evaluate two models on generation: BUFIA (Section 6.2) and a neural transducer (Section 6.1). We have two baselines, the majority baseline of picking the most frequent stress position in each language, and a rule-based implementation derived from the descriptive literature on each variety's stress system. The rule-based baseline is a serious baseline, as much scholarly work has addressed this issue (Section 6.3).

6.1 Neural Network (NN)

Our neural model is the character–level transformer presented in (Wu et al., 2021), which has been widely strong baseline for morphological inflection, historical text normalization, grapheme-to-phoneme conversion, and transliteration tasks. The model is a standard encoder–decoder with multi-head self-attention. This is trained using the syllable structure representation adopted throughout this paper (i.e., given input `c v c v v c`, the model should generate `c v c v v c`).The parameters of the model are described in Table 7.

6.2 BUFIA-Derived Finite State Machine

We used an off-the-shelf C++ implementation of BUFIA.[4] For easier processing, we transform the constraints generated by BUFIA into a constraint format suitable for using The Language Toolkit (LTK) (Lambert, 2024) to generate language-equivalent finite state acceptors (FSAs). The generated FSAs can be used directly to evaluate the accuracy of the induced grammar. The evaluation is done based on a generation task in which the FSA receives an unstressed input and outputs the stressed version.

The FSAs generated from LTK are unweighted. This causes an issue when there is apparent optionality in the data. For example, if the data has a list

[4] https://github.com/pterodactylogan/bufia

of 100 inputs of the form `cv.cv`, with 99 having stress pattern `CV.cv` and one `cv.CV`, then both inputs will be accepted because they are valid paths. We therefore add weights to the FSA to account for the distributions seen in the training set. The idea is to count the total number of transitions in a valid path using the Viterbi algorithm.

Once we have the total count for all segments present in a valid path in the training dataset, we start assigning weights state by state. We do this by going from the start state, find the outgoing arcs, evaluate the total count of outgoing arcs, and compute a smoothed probability using $p = (count + \alpha)/(total\_count + \alpha * m)$, where $\alpha = 0.5$, and m is the number of outgoing arcs. Then we set the cost to $c = -log(p)$.

For the generator FST, the state machine generates all toggle versions and picks the lowest-cost path if there are multiple paths. This means the input `cv.cv.cv` generates three inputs `CV.cv.cv`, `cv.CV.cv`, and `cv.cv.CV`. Thus, each input generates multiple paths with different costs. The FSA "outputs" the path with the lowest cost and the other paths are invalid.

6.3 Linguistic Knowledge: Stress Assignment in JOR, SUD and EGY

This experiment implemented the rules mentioned for Jordanian, Sudanese, and Egyptian stress assignment.

We use linguistic literature to develop a sophisticated baseline in addition to the most frequent position baseline. To implement the sophisticated baseline and test it, we need an abstract word representation that captures only syllable weight and stress, since that is how linguists predict Arabic stress. Unstressed syllables are represented as *l* (light), *h* (heavy), and *x* (super-heavy). Stressed syllables are capitalized *L*, *H*, and *X*. For example, Jordanian Arabic مكاتب [ma.ˈkaa.tib] 'offices' contains the syllable structures `cv.CVV.cvc`, corresponding to the weight representation [*l H h*]. The data in Appendix Table 8 contains all the syllable structures present in our data and their weights.

In JOR, stress and vowel length together are contrastive, yielding near-minimal pairs like مكاتيب [ma.ka.ˈtiib] 'letters' and مكاتب [ma.ˈkaa.tib] 'offices' derived from the same root. JOR stress has been subject to several theoretical treatments in terms of traditional ordered rule phonology and metrical phonology (Al-Sughayer, 1990; AbuAbbas, 2003; Al-Wer, 2007). We implemented the following rules from the cited work:

(1) **Jordanian Stress Assignment**

- Stress the rightmost heavy or super-heavy syllable if it is final, penultimate, or antepenultimate.
- Stress the antepenultimate syllable otherwise.

Stress assignment in Cairene Egyptian Arabic is generally predictable and has been the subject of several treatments in different frameworks (Broselow, 1976; McCarthy, 1979; Watson, 2002). Following Becker (2022), we summarize stress assignment in 2. Stress is predictable if one of the last three syllables is heavy or the word consists of one to four light syllables.

(2) **Cairene Stress Assignment**

- If the final syllable is heavy or super-heavy, assign final stress.
- If the final syllable is light and the penult is heavy, assign penultimate stress.
- If the final two syllables are light and the antepenult is heavy, assign the penultimate stress.
- If the word is disyllabic and both syllables are light, assign penultimate stress.
- If the word is a monomorpheme, assign final stress.
- Else, stress the antepenult in trisyllabic or longer words.

Similar to JOR and EGY, stress in SUD is very predictable and has been studied by several researchers, including (Abdel-Khalig, 2014; Ali, 2017; Dickins, 2007). We implemented the slightly modified following rules based on (Hamid, 1984):

(3) **Sudanese Stress Assignment**

- Stress the heavy or super-heavy syllable if it is the only one.
- If there is more than one heavy or super-heavy syllable, stress the rightmost one.
- Otherwise, stress the first syllable

6.4 Baseline

We use a standard simple baseline based on the most frequent stress pattern. The most frequent stress index in Jordanian Arabic is the first syllable, with 72.7%. The most frequent stress index in Egyptian Arabic is the second syllable with 55.0%.

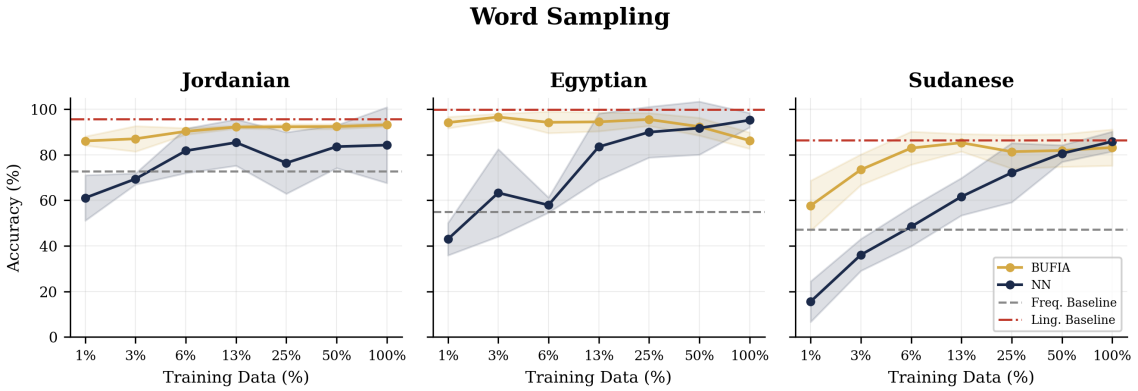

Figure 1: BUFIA compared to NN in word sampling (Sampling Paradigm 1) experiment in three languages.

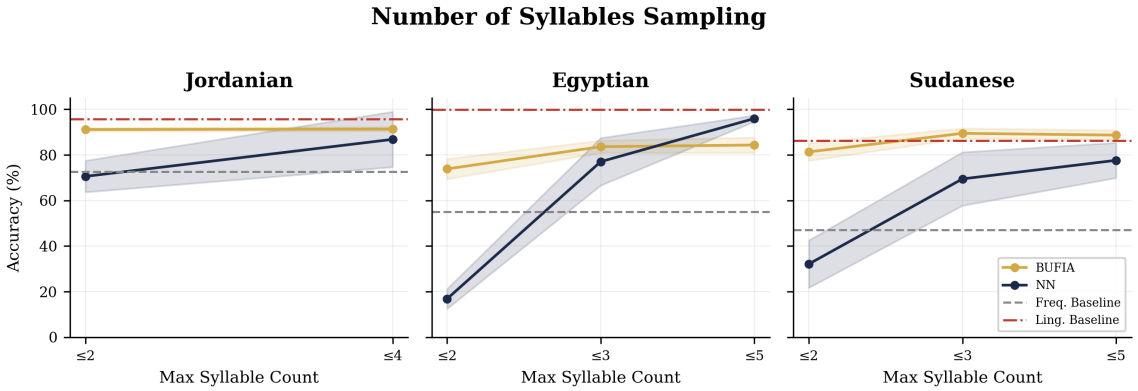

Figure 2: BUFIA compared to NN in the number of syllables sampling experiment (Sampling Paradigm 2) for three languages. The reason why JOR is missing 3 in the graph is because there are only 15 words with 4 syllables in all of the dataset, so we decided to combine them in one step.

The most frequent stress index in Sudanese Arabic is the first syllable with 47.1%. Neither the frequency baseline nor the linguistic rules are trained on our data; therefore, they have no learning curve.

7 Results

The rules derived from the linguistic literature (Section 6.3) achieve very high performance on all three dialects: 99.8% for Egyptian Arabic, in 95.7% for Jordanian Arabic, and 86.3% for Sudanese Arabic.It is worth noting that the linguistic baseline outperformed both models under most conditions. This fact shows that the Arabic stress systems are rule-governed and well studied linguistically.

We can see from Figures 1, 2, and 3 that across all three sampling strategies, BUFIA performs better than the neural network in most settings, especially when training size is minimal. However, for both BUFIA and the neural network, accuracy generally improves with increased training data, and the performance gap between them shrinks. For Egyptian Arabic, we see in Figures 2 and 3 that the neural model outperforms BUFIA at the largest training sizes.

A few other trends stand out. First, the variance between runs of BUFIA is much lower than for the neural network, indicating a relative robustness to the random nature of the data splits. Second, sampling by structure (Sampling Paradigm 3; Figure 3) proves the most challenging for both models, but especially for the neural network, which only surpasses the frequency baseline on Jordanian and Egyptian, and even then, only on the largest training size. Third, we see that BUFIA's performance begins high, and plateaus or even decreases over the course of the two other sampling strategies. The reason for the decrease is that BUFIA is very sensitive to noise and optionality.

8 Error Analysis

In this section, we discuss the errors the different models make while predicting stress. The errors in both NN and BUFIA at the 100% rate are er-

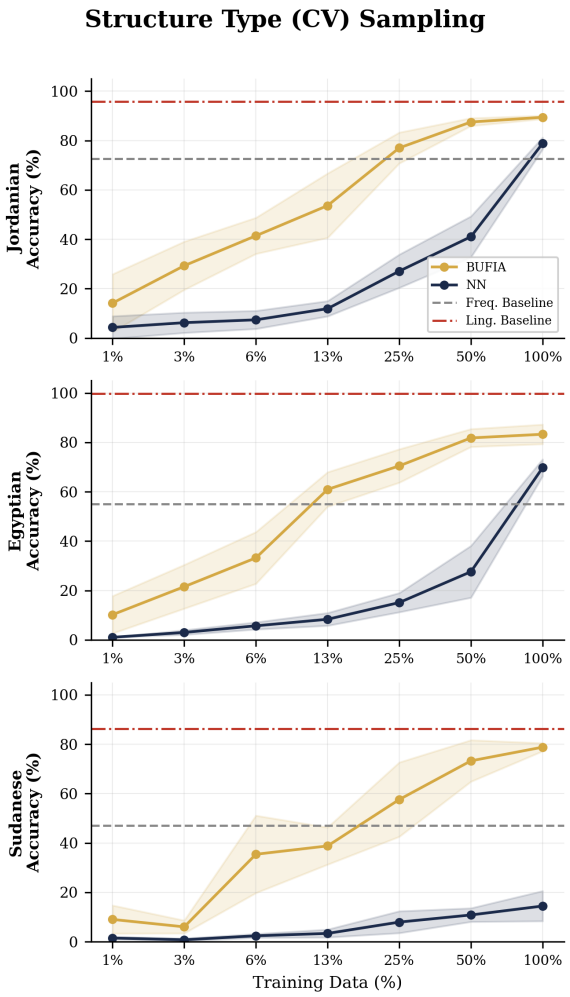

Figure 3: BUFIA compared to NN in structure type (CV) sampling experiment (Sampling Paradigm 3) in three languages.

rors due to exceptions, for example in some cases clitics do not count in the phonemic word but for this setup it will still show as a syllable.Another example, a structure like `cvc.cv.cv` has cases of stress such as `CVC.cv.cv` and `cvc.CV.cv` with the latter being the majority, thus the model gets the `CVC.cv.cv` incorrectly. This is also the case for linguistic rules and it is why the rule-based analysis does not achieve 100% accuracy.

8.1 BUFIA Error Analysis

The errors are due to the training length restriction. BUFIA was trained only on a window of 4 characters due to its exponential big-O time complexity (Chandlee et al., 2019). Thus, it had mistakes such as `CVV.cvvc` instead of `cvv.CVVC` and `CVC.ccvv.cv` instead of `cvc.CCVV.cv`, showing that BUFIA has some issues with 4-segment syllables.

8.2 NN Error Analysis

The types of mistakes that appear in the neural network are changes in syllable-structure identity. For example, outputting `CCCV.cvc` instead of `CCVC.cvc`, where the model swapped `VC` to `CV` in the first syllable. In another case, it outputted `cv.cv.CVC.cv` instead of `cv.cvc.CVV.cv` where a `V` was changed to `C`. There were a few extreme cases of this degenerate performance,for example, in one data point when sampling from structure types, one of the outputs was `cvc.CV.cv.cv.c v.cv.cv.cv.cv.cv.cv.cv.cv.cv.cv.cv. cv.cv.v.cv.cv.cv.cv.cv.cv.cv.cv.cv.cv .cv.cv.cv.cv.cv.cv.cv.CVVVVVVVVV.cv.cv` instead of `cv.cvc.cv.CVC.cv.cv`. This type of errors are the consequence of treating this as a generation rather than labeling problem.

9 Conclusion

We study the problem of predicting which syllable in a word is stressed in Egyptian and Jordanian Arabic from their unstressed surface forms. We use both Egyptian and Jordanian to simulate the low-resource setting. We compare state-of-the-art character-level neural transformer models with a grammar-induction algorithm (BUFIA), a frequency baseline, and a more sophisticated handcrafted-rule baseline developed by linguists. We simulate a learning curve by incrementally increasing the amount of data. There are three settings for that increase: randomly picking, sampling based on syllable structure type, and sampling based on the number of syllables. Our results suggest that a grammar induction algorithm can be a highly effective alternative to other approaches in low-resource settings. It also shows that the character-based model benefits from increased data size and increased data representation "complexity". Finally, linguistically crafted rules derived from prior research in the linguistic literature performed well in predicting the stress phenomenon in real-world data.

In future work, we will investigate classification approaches (both classifying whether a representation with stress is correct, and which syllable is stressed). This approach may mitigate some of the hallucination errors of the neural network. However, we still expect the BUFIA approach to outperform the NN at smaller training sizes.

Limitations

We have investigated only three Arabic varieties. The goal of this paper is to pave the way for a broader investigation of a wider range of varieties and to demonstrate that BUFIA can be used for this purpose. We chose the varieties because data were available and because there is scholarly work on these varieties. Extending our work to more varieties is future work. We use the IPA transcription as a starting point, but it is not always available. We will also try to extend the window size for BUFIA to more than four segments.

References

Ali Abdel-Khalig. 2014. *Syllabification and phrasing in three dialects of Sudanese Arabic*. Ph.D. thesis, University of Toronto.

Khaled Hasan AbuAbbas. 2003. *Topics in the phonology of Jordanian Arabic: An optimality theory perspective*. University of Kansas.

Rasheed Saleem Al-Jarrah. 2002. *An optimality-theoretic analysis of stress in the English of native Arabic speakers*. Ball State University.

Khalil Ibrahim Al-Sughayer. 1990. *Aspects of comparative Jordanian and modern standard Arabic phonology*. Michigan State University.

Enam Al-Wer. 2007. Jordanian arabic (amman). *Encyclopedia of Arabic language and linguistics*, 2:505–517.

Abdel-Khalig Ali. 2017. Prosodic domains of syllabification in sudanese arabic. *Perspectives on Arabic Linguistics XXIX*, pages 33–54.

International Phonetic Association. 1999. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press.

Michael Becker. 2022. Cairene arabic stress is local. *Radical: A Journal of Phonology*, 4:211–247.

Ellen Broselow. 2017. Syllable structure in the dialects of arabic. *The Routledge handbook of Arabic linguistics*, pages 32–47.

Ellen Broselow, Su-I Chen, and Marie Huffman. 1997. Syllable weight: convergence of phonology and phonetics. *Phonology*, 14(1):47–82.

Ellen I Broselow. 1976. *The phonology of Egyptian Arabic*. University of Massachusetts Amherst.

Jane Chandlee, Remi Eyraud, Jeffrey Heinz, Adam Jardine, and Jonathan Rawski. 2019. Learning with partially ordered representations. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 91–101, Toronto, Canada. Association for Computational Linguistics.

James Dickins. 2007. *Sudanese Arabic: Phonematics and syllable structure*, volume 38. Otto Harrassowitz Verlag.

Qing Dou, Shane Bergsma, Sittichai Jiampojamarn, and Grzegorz Kondrak. 2009. A ranking approach to stress prediction for letter-to-phoneme conversion. In *Proceedings of the Joint Conference of the 47th annual meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 118–126.

Elan Dresher and Jonathan Kaye. 1990. A computational learning model for metrical phonology. *Cognition*, 34:137–195.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig, editors. 2025. *Ethnologue: Languages of the World*, twenty-eighth edition. SIL International, Dallas, Texas.

Awn al-Sharif Gasim. 1965. Some aspects of sudanese colloquial arabic. *Sudan notes and records*, 46:40–49.

Diana Geneva, Georgi Shopov, Kostadin Garov, Maria Todorova, Stefan Gerdjikov, and Stoyan Mihov. 2023. Accentor: An explicit lexical stress model for tts systems. In *Proc. Interspeech 2023*, pages 4848–4852.

John Goldsmith. 1994. A dynamic computational theory of accent systems. In Jennifer Cole and Charles Kisseberth, editors, *Perspectives in Phonology*, pages 1–28. Stanford: Center for the Study of Language and Information.

John Goldsmith. 2011. The syllable. *The handbook of phonological theory*, pages 164–196.

Matthew K Gordon and Harry van der Hulst. 2020. Word-stress systems.

Prahlad Gupta and David Touretzky. 1991. What a perceptron reveals about metrical phonology. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pages 334–339.

Abdel Halim M Hamid. 1984. *A descriptive analysis of Sudanese colloquial Arabic phonology*. University of Illinois at Urbana-Champaign.

Bruce Hayes. 1995. *Metrical stress theory: Principles and case studies*. University of Chicago Press.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic inquiry*, 39(3):379–440.

Jeffrey Heinz. 2009. On the role of locality in learning stress patterns. *Phonology*, 26(2):303–351.

Bruno Herin, Igor Younes, Enam Al-Wer, and Youssef Al-Sirour. 2021. The classification of bedouin arabic: Insights from northern jordan. *Languages*, 7(1):1.

Saussan Khalil. 2011. Talk like an Egyptian: Egyptian Arabic as an option for teaching communicative spoken Arabic.

Hanaa Kilany, Hassan Gadalla, Howaida Arram, Ashraf Yacoub, Alaa El-Habashi, and Cynthia McLemore. 2002. Egyptian colloquial arabic lexicon. https://catalog.ldc.upenn.edu/LDC99L22. LDC Catalog No. LDC99L22.

Jordan Kodner. 2016. Simple Syllabify.

P Ladefoged and K Johnson. 2014. A course in phonetics, 7th edn. cengage learning. *Inc., Boston*.

Dakotah Lambert. 2024. System description: A theorem-prover for subregular systems: The language toolkit and its interpreter, plebby. In *International Symposium on Functional and Logic Programming*, pages 311–328. Springer.

Seung Suk Lee, Joe Pater, and Brandon Prickett. 2025. Representing and learning stress in a maxent framework. In *Proceedings of the Annual Meetings on Phonology*, volume 1. University of Massachusetts Amherst Libraries.

Bogdan Ludusan, Stefan Ziegler, and Guillaume Gravier. 2012. Integrating stress information in large vocabulary continuous speech recognition. In *INTERSPEECH-Annual Conference of the International Speech Communication Association*, page na.

Stefano Manfredi. 2015. Ethnolinguistic identity and language revitalization among the laggorí in the nuba mountains. *Multidimensional Change in Sudan (1989–2011): Reshaping Livelihoods, Conflicts and Identities*, page 281.

John J McCarthy. 1979. On stress and syllabification. *Linguistic inquiry*, 10(3):443–465.

Abdel Rahman Mustapha. 1982. *La phonologie de l'arabe soudanais (phonématique et accentuation, Tome 1)*. Ph.D. thesis, PhD thesis: Paris: Université de la Sorbonne Nouvelle.

Heikki Palva. 1984. A general classification for the arabic dialects spoken in palestine and transjordan. *Studia Orientalia Electronica*, 55:357–376.

Brandon Prickett and Joe Pater. 2025. Learning and generalizing stress patterns with a sequence-to-sequence neural network. *Linguistics Vanguard*, (0).

Abdelrahim Qaddoumi, Jordan Kodner, Salam Khalifa, Ellen Broselow, and Owen Rambow. 2026. Syllable structures across Arabic varieties. In *Proceedings of the 13th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 250–260, Rabat, Morocco. Association for Computational Linguistics.

Jonathan Rawski. 2021. *Structure and Learning in Natural Language*. Ph.D. thesis, Stony Brook University.

James Rogers and Dakotah Lambert. 2019. Some classes of sets of structures definable without quantifiers. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 63–77, Toronto, Canada. Association for Computational Linguistics.

Jonathan Sakunkoo and Annabella Sakunkoo. 2025. Lost and found: Computational quality assurance of crowdsourced knowledge on morphological defectivity in wiktionary. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 998–1003.

Paul Schachter and Fe T Otanes. 1983. *Tagalog reference grammar*. Univ of California Press.

Kari Suomi, Juhani Toivanen, and Riikka Ylitalo. 2009. *Finnish sound structure: Phonetics, phonology, phonotactics and prosody*. University of Oulu.

Logan Swanson, Jeffrey Heinz, and Jon Rawski. 2026. Phonotactic learning with structure, not statistics. *Linguistic Inquiry*. In press.

Bruce Tesar and Paul Smolensky. 2000. *Learnability in Optimalty Theory*. MIT Press, Cambridge, MA.

Daan van Esch, Mason Chua, and Kanishka Rao. 2016. Predicting pronunciations with syllabification and stress with recurrent neural networks. In *INTERSPEECH*, pages 2841–2845.

Inga Vendelin. 2010. Perception of predictable stress: A cross-linguistic investigation. *Journal of Phonetics*.

Kees Versteegh. 2014. *Arabic language*. Edinburgh University Press.

Chao Wang and Stephanie Seneff. Lexical stress modeling for improved speech recognition of spontaneous telephone speech in the jupiter domain.

Janet CE Watson. 2002. *The phonology and morphology of Arabic*. Oxford University Press, USA.

Janet CE Watson. 2011. Word stress in arabic.

World Bank. 2024. Population, total — jordan. Accessed 2025-10-05.

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2021. Applying the transformer to character-level transduction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1901–1907.

Yuxiang Zou, Shichao Liu, Xiang Yin, Haopeng Lin, Chunfeng Wang, Haoyu Zhang, and Zejun Ma. 2021. Fine-grained prosody modeling in neural speech synthesis using tobi representation. In *Interspeech*, pages 3146–3150.

A Appendix

LDC symbol	Phonetic description	IPA symbol
C	voiceless glottal stop	ʔ
b	voiced bilabial stop	b
t	voiceless dental stop	t
g	voiced velar stop	g
H	voiceless pharyngeal fricative	ħ
x	voiceless velar fricative	x
d	voiced dental stop	d
r	voiced alveolar flap	ɾ
z	voiced alveolar fricative	z
$	voiceless alveopalatal fricative	ʃ
S	voiceless alveolar velarized fricative	sˤ
D	voiced dental velarized stop	dˤ
T	voiceless dental velarized stop	tˤ
Z	voiced velarized interdental fricative	ðˤ
c	voiced pharyngeal fricative	ʕ
G	voiced uvular fricative	ɣ
f	voiceless labio-dental fricative	f
q	voiceless glottal stop	ʔ
Q	voiceless pharyngeal stop	—
k	voiceless velar stop	k
l	voiced alveolar lateral	l
m	voiced bilabial nasal	m
n	voiced alveolar nasal	n
h	voiceless glottal fricative	h
w	voiced bilabial continuant	w
y	voiced palatal continuant	j
v	voiced labio-dental fricative	v
j	voiced alveopalatal affricate	dʒ

Table 3: Phonology table of the LDC Arabic lexicon and its corresponding IPA — consonants and glides

LDC symbol	Phonetic description	IPA symbol
@	low front unrounded vowel	ɑ
a	low back unrounded vowel	a
i	high front unrounded vowel	i
u	high back rounded vowel	u
%	long @	ɑɑ / ɑː
A	long a	aa / aː
I	long i	ii / iː
O	long back mid rounded vowel	oo / oː
U	long u	uu / uː
E	long front mid unrounded vowel	ee / eː
ay	front upgliding diphthong	aj
aw	back upgliding diphthong	aw

Table 4: Vowels and diphthongs

	c	**v**	**C**	**V**	**#**	**0**
stress	−	−	+	+	0	0
consonant	+	−	+	−	0	0
vowel	−	+	−	+	0	0
wb	0	0	0	0	+	0
sb	0	0	0	0	0	+

Table 5: BUFIA feature matrix: c, s = unstressed consonant/vowel; C, V = stressed counterparts; wb = word boundary; sb = syllable boundary. Symbols: $+$ present, $-$ absent, 0 not applicable.

#	Factor 1	Factor 2	Factor 3	Factor 4
1	[+sb]	—	—	—
2	[-stress]	[+stress,+vowel]	—	—
3	[+stress]	[-stress,+vowel]	—	—
4	[+wb]	[+vowel]	—	—
5	[+wb]	[+wb]	—	—
6	[ ]	[+wb]	[ ]	—
7	[-stress]	[+stress]	[-stress]	—
8	[-stress]	[+stress]	[+consonant]	—
9	[-stress]	[+stress]	[+wb]	—
10	[+stress]	[-stress]	[+stress]	—
11	[+stress]	[-stress]	[+consonant]	—
12	[+stress]	[-stress]	[+wb]	—
13	[+consonant]	[+consonant]	[+stress,+consonant]	—
14	[+consonant]	[-stress,+consonant]	[+consonant]	—
15	[+consonant]	[-stress,+consonant]	[+wb]	—
16	[+vowel]	[+vowel]	[+vowel]	—
17	[+vowel]	[-stress,+vowel]	[+stress]	—
18	[+vowel]	[-stress,+vowel]	[+wb]	—
19	[+wb]	[-stress]	[+stress]	—
20	[+wb]	[+stress]	[-stress]	—
21	[+stress,+vowel]	[ ]	[+stress,+vowel]	—
22	[-stress]	[ ]	[+stress]	[+wb]
23	[+stress]	[ ]	[-stress]	[+stress]
24	[+stress]	[ ]	[+consonant]	[+stress,+vowel]
25	[+stress]	[+consonant]	[+stress,+vowel]	[-stress]
26	[+stress]	[+consonant]	[+stress,+vowel]	[+wb]
27	[+vowel]	[ ]	[+consonant]	[+stress,+consonant]
28	[+vowel]	[ ]	[-stress,+consonant]	[+consonant]
29	[+vowel]	[ ]	[+stress,+vowel]	[-stress]
30	[+wb]	[ ]	[-stress]	[+wb]
31	[+wb]	[ ]	[+consonant]	[+consonant]
32	[+wb]	[ ]	[+consonant]	[+wb]
33	[+wb]	[ ]	[+vowel]	[-stress,+vowel]
34	[-stress,+consonant]	[ ]	[+vowel]	[+wb]
35	[-stress,+consonant]	[ ]	[+vowel]	[-stress,+vowel]

Table 6: BUFIA-learned phonotactic constraints with k=4 and n=3 for Jordanian stress. `wb` = word boundary. `sb` = syllable boundary. Empty `[ ]` indicates that any segment is allowed.

Hyperparameter	Value
Batch size	400
Max steps	20,000
Optimizer	Adam (β_1=0.9, β_2=0.98)
Learning rate	0.001 (min LR 1×10^{-5})
Momentum	0.9
Error stop	10^{-8}
LR scheduler	Warmup inverse square; warmup steps = 4000
Label smoothing	0.1
Gradient clipping (max_norm)	1.0
Max input length (`max_seq_len`)	128
Max decode length (`max_decode_len`)	128
Dropout	0.3
Embedding Dimension	256
Decode strategy	Greedy
Beam size	5
Best model selection	By accuracy (`bestacc=True`)
Shuffle	True
Other flags	`cleanup_anyway=True`, `saveall=False`

Table 7: Training and decoding hyperparameters

	Syllable	Jordanian	Egyptian	Sudanese
l/L	CV	[sa.ʔa.lu] “they asked”	[ka.tab] “he wrote”	[ka.tab] “he wrote”
h/H	CVV	[saa.ʔil] “questioner”	[kaa.tib] “writer”	[kaa.tib] “writer”
	CVC*	[sam.ne] “ghee”	[mak.tab] “office”	[mak.tab] “office”
	CCVC	[ʃraħ.ha] “explain it fem”	–	–
x/X	CVCC	[sadd] “he blocked”	–	[bank] “bank”
	CVVC	[faat] “he entered”	[ki.taab] “book”	[baab] “door”
	CCVVC	[klaab] “dogs”	–	–
	CCVCC	[mʃadd] “fastener”	–	–
	CVVCC	[dʒaadd] “serious ms. sg.”	–	[koobs] “electric socket”

Table 8: Example of licit syllable structures in Jordanian, Sudanese and Egyptian. We note that both CVCC and CVVC can occur only in the word-final position for Egyptian (McCarthy, 1979). * CVC’s weight is light if it is in word final position because the last C in CVC is extrametrical (Hayes, 1995). Syllables said to not exist in the variety in the literature are marked with a ‘-’.

Step	Seed	Train	Dev	Test
1	21	26	-	-
	42	26	-	-
	82	26	-	-
	99	26	-	-
	101	26	-	-
2	21	53	-	-
	42	53	-	-
	82	53	-	-
	99	53	-	-
	101	53	-	-
3	21	104	-	-
	42	104	-	-
	82	104	-	-
	99	104	-	-
	101	104	-	-
4	21	216	-	-
	42	216	-	-
	82	216	-	-
	99	216	-	-
	101	216	-	-
5	21	416	-	-
	42	416	-	-
	82	416	-	-
	99	416	-	-
	101	416	-	-
6	21	832	-	-
	42	832	-	-
	82	832	-	-
	99	832	-	-
	101	832	-	-
7	21	1664	209	209
	42	1664	209	209
	82	1664	209	209
	99	1664	209	209
	101	1664	209	209

(a) Jordanian

Step	Seed	Train	Dev	Test
1	21	24	-	-
	42	24	-	-
	82	24	-	-
	99	24	-	-
	101	24	-	-
2	21	48	-	-
	42	48	-	-
	82	48	-	-
	99	48	-	-
	101	48	-	-
3	21	95	-	-
	42	95	-	-
	82	95	-	-
	99	95	-	-
	101	95	-	-
4	21	196	-	-
	42	196	-	-
	82	196	-	-
	99	196	-	-
	101	196	-	-
5	21	378	-	-
	42	378	-	-
	82	378	-	-
	99	378	-	-
	101	378	-	-
6	21	757	-	-
	42	757	-	-
	82	757	-	-
	99	757	-	-
	101	757	-	-
7	21	1514	190	190
	42	1514	190	190
	82	1514	190	190
	99	1514	190	190
	101	1514	190	190

(b) Egyptian

Step	Seed	Train	Dev	Test
1	21	9	-	-
	42	9	-	-
	82	9	-	-
	99	9	-	-
	101	9	-	-
2	21	19	-	-
	42	19	-	-
	82	19	-	-
	99	19	-	-
	101	19	-	-
3	21	38	-	-
	42	38	-	-
	82	38	-	-
	99	38	-	-
	101	38	-	-
4	21	78	-	-
	42	78	-	-
	82	78	-	-
	99	78	-	-
	101	78	-	-
5	21	151	-	-
	42	151	-	-
	82	151	-	-
	99	151	-	-
	101	151	-	-
6	21	302	-	-
	42	302	-	-
	82	302	-	-
	99	302	-	-
	101	302	-	-
7	21	604	76	76
	42	604	76	76
	82	604	76	76
	99	604	76	76
	101	604	76	76

(c) Sudanese

Table 9: Word sampling dataset splits for Jordanian, Egyptian, and Sudanese Arabic.

Step	Seed	Train	Types	Dev	Types	Test	Types
1	42	17	3	-	-	-	-
	82	264	1	-	-	-	-
	99	28	2	-	-	-	-
	101	129	1	-	-	-	-
2	21	65	4	-	-	-	-
	82	269	2	-	-	-	-
	99	61	3	-	-	-	-
	101	157	2	-	-	-	-
3	21	147	5	-	-	-	-
	42	154	6	-	-	-	-
	82	270	3	-	-	-	-
	99	131	6	-	-	-	-
	101	158	3	-	-	-	-
4	21	282	8	-	-	-	-
	42	226	9	-	-	-	-
	82	357	6	-	-	-	-
	99	222	8	-	-	-	-
	101	248	6	-	-	-	-
5	21	648	12	-	-	-	-
	42	500	11	-	-	-	-
	82	501	11	-	-	-	-
	99	501	21	-	-	-	-
	101	598	14	-	-	-	-
6	21	853	30	-	-	-	-
	42	847	24	-	-	-	-
	82	1050	23	-	-	-	-
	99	1065	27	-	-	-	-
	101	899	22	-	-	-	-
7	21	1691	44	208	76	208	76
	42	1691	42	208	76	208	76
	82	1691	42	208	76	208	76
	99	1691	44	208	76	208	76
	101	1691	43	208	76	208	76

Table 10: Jordanian – Structure Type (CVS)

Step	Seed	Train	Types	Dev	Types	Test	Types
1	42	26	1	-	-	-	-
	82	31	1	-	-	-	-
	101	55	1	-	-	-	-
2	21	132	3	-	-	-	-
	82	48	2	-	-	-	-
	99	106	3	-	-	-	-
	101	209	2	-	-	-	-
3	21	179	4	-	-	-	-
	42	143	5	-	-	-	-
	82	140	4	-	-	-	-
	99	142	4	-	-	-	-
	101	275	4	-	-	-	-
4	21	344	8	-	-	-	-
	42	202	8	-	-	-	-
	82	225	8	-	-	-	-
	99	217	9	-	-	-	-
	101	409	8	-	-	-	-
5	21	431	14	-	-	-	-
	42	408	15	-	-	-	-
	82	456	14	-	-	-	-
	99	397	15	-	-	-	-
	101	594	15	-	-	-	-
6	21	783	30	-	-	-	-
	42	826	33	-	-	-	-
	82	778	28	-	-	-	-
	99	817	29	-	-	-	-
	101	851	29	-	-	-	-
7	21	1553	56	189	108	189	108
	42	1547	60	189	108	189	108
	82	1555	54	189	108	189	108
	99	1549	58	189	108	189	108
	101	1549	57	189	108	189	108

Table 11: Egyptian – Structure Type (CVS)

Step	Seed	Train	Types	Dev Types		Test	Types
1	21	23	1	-	-	-	-
	42	29	2	-	-	-	-
	99	134	1	-	-	-	-
	101	19	3	-	-	-	-
2	21	34	2	-	-	-	-
	42	47	4	-	-	-	-
	82	136	2	-	-	-	-
	99	136	2	-	-	-	-
3	21	54	4	-	-	-	-
	82	159	3	-	-	-	-
	99	157	3	-	-	-	-
	101	50	7	-	-	-	-
4	21	102	7	-	-	-	-
	42	132	5	-	-	-	-
	82	248	5	-	-	-	-
	99	162	5	-	-	-	-
	101	101	11	-	-	-	-
5	21	155	10	-	-	-	-
	42	170	9	-	-	-	-
	82	297	9	-	-	-	-
	99	217	9	-	-	-	-
	101	165	16	-	-	-	-
6	21	375	25	-	-	-	-
	42	436	24	-	-	-	-
	82	408	18	-	-	-	-
	99	386	18	-	-	-	-
	101	339	23	-	-	-	-
7	21	617	36	83	83	83	83
	42	617	36	83	83	83	83
	82	617	36	83	83	83	83
	99	617	36	83	83	83	83
	101	617	36	83	83	83	83

Table 12: Sudanese – Structure Type (CVS)

Step	Seed	Train	Types	Dev	Types	Test	Types
1	21	260	1	-	-	-	-
2	21	1348	2	-	-	-	-
	42	1350	2	-	-	-	-
	82	1358	2	-	-	-	-
	99	1362	2	-	-	-	-
	101	1358	2	-	-	-	-
3	21	1666	4	208	4	208	4
	42	1666	4	208	4	208	4
	82	1666	4	208	4	208	4
	99	1666	4	208	4	208	4
	101	1666	4	208	4	208	4

Table 13: Jordanian – Number of Syllables

Step	Seed	Train	Types	Dev	Types	Test	Types
1	21	575	2	-	-	-	-
	42	613	2	-	-	-	-
	82	605	2	-	-	-	-
	99	611	2	-	-	-	-
	101	627	2	-	-	-	-
2	21	1325	3	-	-	-	-
	42	1331	3	-	-	-	-
	82	1334	3	-	-	-	-
	99	1345	3	-	-	-	-
	101	1327	3	-	-	-	-
3	21	1517	5	189	7	189	7
	42	1517	5	189	7	189	7
	82	1516	5	189	7	189	7
	99	1517	5	189	7	189	7
	101	1516	5	189	7	189	7

Table 14: Egyptian – Number of Syllables

Step	Seed	Train	Types	Dev	Types	Test	Types
1	21	310	2	-	-	-	-
	42	299	2	-	-	-	-
	82	296	2	-	-	-	-
	99	310	2	-	-	-	-
	101	308	2	-	-	-	-
2	21	543	3	-	-	-	-
	42	542	3	-	-	-	-
	82	538	3	-	-	-	-
	99	547	3	-	-	-	-
	101	544	3	-	-	-	-
3	21	604	5	76	6	76	6
	42	604	5	76	6	76	6
	82	604	5	76	6	76	6
	99	604	5	76	6	76	6
	101	604	5	76	6	76	6

Table 15: Sudanese – Number of Syllables

		Jordanian (JOR)		**Egyptian (EGY)**		**Sudanese (SUD)**	
Seed	**% Data**	**NN**	**BUFIA**	**NN**	**BUFIA**	**NN**	**BUFIA**
21	1%	44.02	85.65	33.16	94.21	7.89	47.37
21	3%	67.46	77.03	77.89	97.89	34.21	73.68
21	6%	89.00	89.00	59.47	96.84	36.84	71.05
21	13%	93.30	92.34	93.68	97.89	61.84	82.89
21	25%	50.72	91.39	89.47	97.89	77.63	76.32
21	50%	91.87	91.39	68.42	94.21	81.58	81.58
21	100%	90.91	92.82	94.21	79.47	89.47	86.84
42	1%	67.46	83.73	39.47	90.00	18.42	57.89
42	3%	71.29	88.52	62.11	97.37	27.63	67.11
42	6%	83.73	88.52	62.11	98.42	47.37	84.21
42	13%	86.60	91.39	86.32	98.42	55.26	88.16
42	25%	75.60	90.91	97.37	98.42	73.68	88.16
42	50%	71.29	92.34	96.84	96.32	82.89	84.21
42	100%	94.74	93.78	89.47	88.42	86.84	88.16
82	1%	66.03	87.56	42.11	97.37	2.63	68.42
82	3%	73.21	86.60	91.05	96.84	42.11	67.11
82	6%	90.43	91.39	55.79	96.84	42.11	78.95
82	13%	91.87	91.87	95.79	86.84	50.00	78.95
82	25%	88.04	91.87	97.37	90.53	47.37	82.89
82	50%	88.04	91.39	98.42	91.05	80.26	81.58
82	100%	89.47	92.34	97.37	89.47	78.95	78.95
99	1%	71.77	84.21	45.26	95.79	26.32	43.42
99	3%	67.94	88.52	42.63	96.84	30.26	73.68
99	6%	63.16	92.34	52.63	85.26	55.26	89.47
99	13%	65.55	93.30	86.32	94.74	72.37	89.47
99	25%	81.82	93.78	96.84	95.79	85.53	69.74
99	50%	73.21	94.26	97.89	85.26	84.21	69.74
99	100%	51.20	94.26	98.42	85.79	82.89	69.74
101	1%	55.98	89.00	54.74	93.16	22.37	71.05
101	3%	66.99	94.26	42.63	93.68	46.05	85.53
101	6%	82.30	89.95	59.47	93.68	60.53	90.79
101	13%	89.47	91.87	55.26	94.21	68.42	86.84
101	25%	85.65	93.30	68.42	94.74	76.32	89.47
101	50%	93.30	92.34	96.84	94.74	73.68	92.11
101	100%	94.74	92.34	96.32	87.89	90.79	92.11

Table 16: Full results for the Word Sampling experiment. Accuracy (%) for NN and BUFIA across 5 seeds and 3 languages. Columns are grouped by language: Jordanian (JOR), Egyptian (EGY), and Sudanese (SUD).

		Jordanian (JOR)		Egyptian (EGY)		Sudanese (SUD)	
Seed	**% Data**	**NN**	**BUFIA**	**NN**	**BUFIA**	**NN**	**BUFIA**
21	1%	–	–	–	–	1.20	7.23
21	3%	4.81	28.85	3.17	34.92	0.00	2.41
21	6%	9.62	38.94	4.76	48.15	1.20	55.42
21	13%	12.98	60.10	10.58	64.55	3.61	49.40
21	25%	25.00	71.63	21.69	67.72	14.46	73.49
21	50%	28.85	88.94	43.92	85.71	14.46	77.11
21	100%	82.21	89.42	64.02	84.66	9.64	75.90
42	1%	0.96	10.10	1.06	19.58	1.20	10.84
42	3%	–	–	–	–	–	–
42	6%	5.29	40.38	4.76	36.51	2.41	39.76
42	13%	8.17	76.44	6.88	61.38	1.20	39.76
42	25%	20.19	76.92	14.29	62.96	2.41	51.81
42	50%	38.46	88.94	35.98	81.48	6.02	73.49
42	100%	78.37	88.46	69.31	85.71	12.05	79.52
82	1%	12.02	33.65	1.06	10.05	–	–
82	3%	12.98	31.25	1.59	10.58	1.20	8.43
82	6%	13.46	31.25	4.23	26.46	2.41	9.64
82	13%	17.31	43.75	5.29	47.62	6.02	36.14
82	25%	22.60	68.75	11.64	72.49	9.64	72.29
82	50%	37.98	87.50	19.58	80.95	10.84	74.70
82	100%	75.00	90.38	70.90	77.25	12.05	80.72
99	1%	0.96	2.40	–	–	1.20	1.20
99	3%	1.92	14.90	3.17	17.99	1.20	7.23
99	6%	3.37	42.79	6.88	17.46	2.41	27.71
99	13%	10.58	42.79	6.88	63.49	2.41	26.51
99	25%	39.42	85.10	11.11	66.67	3.61	32.53
99	50%	47.12	84.62	16.93	75.66	12.05	57.83
99	100%	80.29	89.90	74.60	80.42	26.51	78.31
101	1%	3.37	10.58	1.06	1.06	2.41	16.87
101	3%	5.29	42.31	4.23	22.75	–	–
101	6%	5.29	53.85	7.94	37.57	3.61	44.58
101	13%	10.58	45.19	12.17	67.72	3.61	42.17
101	25%	28.37	82.69	16.93	82.54	9.64	57.83
101	50%	52.88	87.50	21.69	85.19	10.84	83.13
101	100%	78.37	88.94	69.84	88.36	12.05	79.52

Table 17: Full results for the Structure Type (CV) Sampling experiment. Accuracy (%) for NN and BUFIA across 5 seeds and 3 languages. Columns are grouped by language: Jordanian (JOR), Egyptian (EGY), and Sudanese (SUD). Missing cells indicate that the split was not available.

Language	Seed	Max Syll.	NN Acc	BUFIA Acc
Jordanian (JOR)	21	≤2	70.19	89.90
	21	≤4	95.19	90.38
	42	≤2	77.40	91.35
	42	≤4	95.19	91.35
	82	≤2	75.48	90.87
	82	≤4	88.94	92.31
	99	≤2	57.69	91.35
	99	≤4	62.98	92.31
	101	≤2	72.12	92.31
	101	≤4	91.83	90.38
Egyptian (EGY)	21	≤2	13.23	70.37
	21	≤3	67.72	85.19
	21	≤5	96.30	83.07
	42	≤2	12.70	71.96
	42	≤3	85.19	79.37
	42	≤5	94.18	79.37
	82	≤2	14.81	73.02
	82	≤3	86.24	84.13
	82	≤5	97.88	83.60
	99	≤2	19.05	71.43
	99	≤3	61.38	82.01
	99	≤5	94.71	86.24
	101	≤2	24.34	82.54
	101	≤3	84.66	87.30
	101	≤5	96.83	89.42
Sudanese (SUD)	21	≤2	18.42	76.32
	21	≤3	75.00	86.84
	21	≤5	84.21	85.53
	42	≤2	28.95	86.84
	42	≤3	46.05	90.79
	42	≤5	78.95	92.11
	82	≤2	42.11	77.63
	82	≤3	75.00	86.84
	82	≤5	84.21	88.16
	99	≤2	25.00	82.89
	99	≤3	75.00	90.79
	99	≤5	77.63	89.47
	101	≤2	46.05	82.89
	101	≤3	76.32	92.11
	101	≤5	63.16	88.16

Table 18: Full results for the Number of Syllables Sampling experiment. Accuracy (%) for NN and BUFIA across 5 seeds, 3 languages, and data splits by maximum syllable count.