Picking Up Where the Linguist Left Off: Mapping Morphology to Phonology through Learning the Residuals

Salam Khalifa[†‡], Abed Qaddoumi[††], Ellen Broselow[†], and Owen Rambow[†‡]
[†]Department of Linguistics, and
[‡]Institute for Advanced Computational Science (IACS)
Stony Brook University
[††]Independent
{first.last}@stonybrook.edu

Abstract

Learning morphophonological mappings between the spoken form of a language and its underlying morphological structures is crucial for enriching resources for morphologically rich languages like Arabic. In this work, we focus on Egyptian Arabic as our case study and explore the integration of linguistic knowledge with a neural transformer model. Our approach involves learning to correct the residual errors from hand-crafted rules to predict the spoken form from a given underlying morphological representation. We demonstrate that using a minimal set of rules, we can effectively recover errors even in very low-resource settings.

1 Introduction

Modern Standard Arabic (MSA) and Classical Arabic are heavily studied and privileged with many resources. Arabic dialects, on the other hand, are considered to be low-resource. Some dialects, such as Egyptian and Gulf Arabic, have dedicated resources for morphological analysis but lack corresponding phonological representations. A paired representation of morphological analysis with full phonological representation is important for downstream tasks like text-to-speech (TTS) and dialect identification. Moreover, given a morphological representation in MSA as an output from a morphological analyzer, generating the phonological representation is straightforward using a handful of rules due to the relative transparency of the standard orthography in MSA. This is, however, challenging for dialects, which generally lack conventionalized consistent orthography, and in which the mapping from underlying to phonological forms may be less transparent. While grapheme-to-phoneme models seem suitable for this task, they require large amounts of data to train, which is not readily available for most Arabic dialects.

In Arabic, a large number of phonological processes are sensitive to morphological structure. It is, therefore, reasonable to map morphophonological properties and utilize them in generating phonological representations. Theoretical linguistics offers a rich literature discussing various frameworks to describe these processes efficiently. However, these frameworks are traditionally based on what is assumed to be a representative subset of the language data and are not tested against large corpora. Consequently, different analyses may diverge both in terms of the posited underlying representations and the rules or constraints that relate these representations to the spoken forms.

In this work, we explore the task of generating spoken forms of words from their underlying morphological representations for dialectal Arabic, specifically the Cairene variety of Egyptian Arabic. Our approach utilizes the expertise of a linguist to provide morphophonological rules that map a word's morphological representation to its spoken form. We then train a character-level neural transformer model to learn the residual errors, i.e., those errors that remain after applying the linguist's rules. Thus, by comparing the linguist's rule-based predictions and the actual spoken forms, this model serves as an additional correction step.

2 Related Work

Modern morphological analyzers for Arabic (Habash et al., 2012; Khalifa et al., 2017; Boudchiche et al., 2017; Taji et al., 2018; Khairallah et al., 2024) provide many features beyond morphosyntactic ones such as morphological tokenization, diacritization, and root and pattern information, but phonological representation is usually absent. Although there is some degree of phonotactic modeling, it is typically constrained by orthography. When fully diacritized, the standard orthography of MSA is phonologically transparent to a large degree, making phonological transcription of inflected words relatively simple, as demonstrated

Proceedings of The Second Arabic Natural Language Processing Conference, pages 258–264
August 16, 2024

by Biadsy et al. (2009a) and Habash et al. (2018). However, dialectal Arabic lacks conventionalized orthography and its phonological forms may be quite distant from the standard orthography.

Biadsy et al. (2009b) modeled phonotactics for dialectal Arabic for the dialect identification task; however, it was based on speech signals directly. Bouamor et al. (2018) provides phonological transcription for some core lexicon entries, although those were elicited from native speakers and are not modeled to generate phonological representations for new forms productively.

In our own recent work, we generate phonological forms by learning morphophonological mappings automatically (Khalifa et al., 2022, 2023); however, while explainable, our approach produces a large number of rules and so far has modest performance as an end-to-end system.

To the best of our knowledge, there is no effort focusing primarily on producing phonological representations from morphological ones for dialectal Arabic. In this work, we take inspiration from Zamani et al. (2018): we train a model to correct the predictions of hand-crafted rules.

3 Linguistic Background

Phonology is one of the most salient differences between MSA and dialectal Arabic. While most phonotactics for MSA can be derived through morphotactics on the level of orthography, the same is not necessarily true for dialects. Naturally occurring dialectal text is usually spelled spontaneously, and even in conventional spelling, it is undiacritized, and thus far less reflective of phonology compared to MSA.

In this work, we focus on modeling the mapping from morphology to phonology through morphophonological mappings, specifically at the phonemic level rather than the phonetic one. In addition to some morpheme-specific rules, most phonological processes in Egyptian Arabic are triggered by strict requirements on how segments are organized into syllables. One such requirement is that each syllable must begin with exactly one consonant; when an underlying representation begins with a vowel, that onset consonant is supplied either by resyllabification of the final consonant of a preceding word or by insertion of a glottal stop (hamza). A second requirement is that, except in phrase-final position, syllables may end with no more than one consonant; thus, when concatenation of morphemes creates a triconsonantal sequence, a vowel is inserted following the second consonant. A third requirement restricts long vowels to either high or low positions in the vocal tract and to stressed syllables; underlyingly long vowels are therefore shortened when unstressed and raised if underlyingly mid. Similarly, long vowels are restricted to open syllables except in word-final position. Finally, a word-internal short high vowel [i] is deleted when the consonants surrounding the vowel can be reassigned to neighboring syllables.

4 Data

In this work, we chose the Egyptian Colloquial Arabic Lexicon (**ECAL**; Kilany et al., 2002) to acquire the spoken forms (SF). ECAL also provides detailed morphological analysis for its entries but only in the form of a complex part-of-speech (POS) tag and has minimal segmentation. Since our goal is to map the morphological representation (MR) of the word, we need to generate a fine-grained tokenized MR. To do so, we use the CALIMA$_{EGY}$ morphological analyzer (Habash et al., 2012) with the analyzer engine from CamelTools (Obeid et al., 2020). We utilize the undiacritized orthographic form provided by ECAL and run it through CALIMA$_{EGY}$ and extract the tokenization from the best matching analysis based on the similarities between the POS tags and lemmas provided by both resources. CALIMA$_{EGY}$ uses a conventional orthographic representation for its entries that balances both phonology and etymology (Habash et al., 2018). We map the different morphs in the tokenized form from their orthographic form to their phonemic form. We note that this phonemic form is *hypothesized* based on certain assumptions about the *underlying representation* of the different morphs (Chomsky and Halle, 1968; Hyman, 2018). For example, word-final vowels are considered to be underlyingly short (Broselow, 1976; Abdel-Massih et al., 1979; Hamid, 1984; Haddad, 1984) although others have argued in favor of underlyingly long final vowels in various dialects (Abdo, 1969; Abu-Mansour, 1987; McCarthy, 2005). While we acknowledge the diverse hypotheses concerning underlying representations, our focus here is on comparing the performance of different systems, assuming a specific set of underlying representations. Since rules consider context from the underlying representation, adopting a different set of underlying representations would not necessarily affect the final predictions. In addition to these fundamental choices about the underlying representation, we perform other orthographic-specific changes to convert the orthography-oriented representation coming out of CALIMA$_{EGY}$ into a phonemic representation: a) long vowels that are represented orthographically with a short vowel and a glide are normalized into a single symbol: 'iy','uw' –> 'I','U'[1], b) all glottal stop shapes are normalized into a bare hamza, c) consonants with unconditional sound change (i.e., /q/, /j/, /ð/, and /θ/ ق, ج, ث, and ذ) preserve their etymological spelling as it is not central to morphophonological processes.

While converting the single morphs from orthography to phonology is relatively straightforward, generating the full spoken form is not due to the complex interactions as explained in §3.

The resulting dataset contains pairs of MR and SF. An example entry for the word /kita:bha/ 'her book' كتابها[2] below, where '#' represents word boundaries and '=' is the stem-suffix boundary:

(1) `MR: #kitAb=ha#     SF: #kitabha#`

We followed the splits for TRAIN/DEV/EVAL that were provided by ECAL.

5 Linguist Rules

Linguistic formalisms, such as phonological rules or constraints, provide an efficient framework for describing generalized linguistic phenomena. Trained, well-versed linguists develop formal grammars mapping posited underlying representation to spoken forms based on extensive study of collected data points representing spoken language.

In this work, we use a minimal set of morphophonological rewrite rules provided by Broselow, a linguist experienced in Egyptian Arabic phonology. Six rules in total were provided based on Broselow (1976, 2017), as well as syllabification and stress assignment operations. Because rules that concern the organization of segments into syllables may interact, they must be applied in order. The rules we use are as follows (in order):

1. **Definite Article /l/ Assimilation** (`sun_letters`) Similar to MSA, the /l/ in the definite article /(?)il/ assimilates to a following coronal consonant.

2. **First Vowel Deletion** (`FVD`) When a proclitic ends with a short vowel, and the stem starts with a short vowel, the first vowel deletes, $\mathtt{v_x}$`-`$\mathtt{v_y} \rightarrow \mathtt{v_y}$; where `-` is a prefix boundary.

3. **Vowel insertion** (`VI`) An epenthetic vowel is inserted to avoid a sequence of three consonants. `CCC` → `CCvC`. While some pronoun enclitics require a specific vowel, the default for insertion is /i/.

   Syllabification (`syllabify`) An operation to add syllable boundaries "`.`" with the condition that there are no onsetless syllables, i.e., a syllable must start with a consonant.

4. **High Vowel Deletion** (`HVD`) The deletion of /i/ when it falls in the following context `VCi.CV`; where `V` denotes a short or a long vowel.

5. **Long Vowel Shortening** (`VLS`) Any long vowels in closed syllables are shortened, unless the syllable is word-final. `VVCCV` → `VCCV`.

   Stress Assignment (`stress_assign`) An operation to assign stress according to Mitchell (1978).

6. **Unstressed Long Vowel Shortening** (`ULVS`) Any long vowel that remains unstressed must be shortened.

Figure 1 shows the process of applying the rules in order.

6 Approach: Learning the Residual

Preliminary analysis of the rule predictions on a sample of data has shown that while the rules cover a wide range of phenomena, they don't predict everything correctly. This is due to several factors: **a)** a mismatch between the expectation of the form of the MR and the automatically generated MR; **b)** less frequent phonological processes that were not addressed by the current set of rules; **c)** some of the MRs which we automatically generated from the ECAL corpus have non-trivial errors that require extensive manual adjustments. To correct for these cases, we train a character-level neural transformer to learn the residual between the rule-predicted SF and the gold SF.

There are two principal goals to this effort: **a)** To understand the coverage of the set of rules proposed by the linguist on naturally occurring data. **b)** To evaluate the efficacy of learning the residuals when presented with a small amount of training data, as most Arabic dialects are low-resource.

[1] We use the same romanization script that was used by the LDC for our data with minor modification for readability. Check the appendix for the full map.

[2] Arabic script spelling follows CODA (Habash et al., 2018).

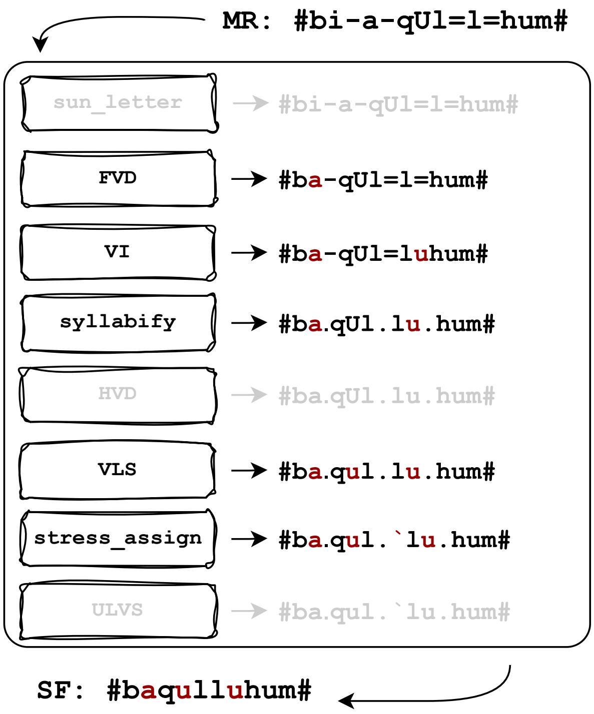

Figure 1: An example of applying the set of the six rules on the word /baʔulluhum/ 'I am telling them' باقول لهم. Grayed out rules indicate no effect on the input. Morpheme boundaries are removed before syllabification.

6.1 Experimental Setup

Data We partitioned TRAIN (12K words) using frequency-weighted sampling into smaller sets to evaluate the learning curve, starting from a tiny set of examples until the full set. The splits range from 100-1,000 examples with increments of 100, and then from 1,000 to full TRAIN with increments of 1,000. For evaluation, we extracted data points from DEV (5K) that were not seen in TRAIN, therefore creating OOV-DEV (2K). The respective EVAL and OOV-EVAL portions are reserved for further experiments.

Baselines

- **DONOTHING** Copy MR as is (without morpheme boundaries).
- **TRNF** Train the character-level neural transformer described in (Wu et al., 2021) using the grapheme-to-phoneme task setting to predict SF given MR.

Systems

- **RULES** Apply the set of linguist's rules described in §5. The rules are implemented using regular expressions. Note that the rules were provided independently of our corpus; they are based on the linguist's expertise.
- **TRNF-RES** Train the TRNF to predict SF from the (possibly wrong) SF predicted by RULES (RSF).

7 Evaluation and Discussion

We evaluate different configurations using OOV-DEV. For the first set of experiments, we evaluate the overall performance of all systems across the different learning curve training sizes. **TRNF** is trained on MR-SF pairs for each of the splits, while **TRNF-RES** is trained on RSF-SF pairs after applying **RULES** at each split. The accuracy across the different sizes is shown in Figure 2. We see that at the lowest setting (100-1,000), the rules alone perform best. However, within the same setup, **TRNF-RES** performs notably better than **TRNF**, which indicates the value of learning the residuals. One could argue that learning the residual may seem easier since the pairs are more similar. However, the significantly lower performance of **TRNF** compared to **DONOTHING** suggests that this family of models is not well equipped to learn to copy. As the data size increases, both neural models quickly catch up and the difference becomes negligible.

size	TRNF-RES						TRNF
	1	2	3	4	5	All	
100	6.7%	6.7%	7.4%	9.8%	9.6%	11.1%	4.6%
200	28.1%	31.3%	32.5%	43.8%	46.3%	49.0%	22.5%
300	42.4%	41.4%	44.7%	51.6%	56.6%	60.5%	32.6%
400	54.2%	54.2%	57.4%	63.3%	67.0%	71.5%	43.9%
500	62.1%	62.2%	61.2%	69.1%	72.6%	76.0%	49.7%
600	**64.8%**	65.9%	67.8%	72.2%	74.1%	76.3%	57.5%
700	**68.1%**	**69.5%**	**71.1%**	**74.4%**	74.7%	80.0%	59.4%
800	**69.3%**	**71.4%**	**72.6%**	**75.9%**	**78.2%**	76.8%	64.1%
900	**73.5%**	**73.7%**	**74.7%**	**78.6%**	77.5%	79.3%	65.9%
1,000	**74.5%**	**74.8%**	**75.3%**	**79.3%**	**78.2%**	81.6%	69.2%
Rules	64.3%	68.5%	70.5%	73.2%	78.2%	86.5%	
DoNth	61.1%						

Table 1: Accuracy on OOV-DEV for extremely low-resource settings. **TRNF-RES** is trained on the residuals using incremental sets of rules. Bolded numbers mean that the accuracy for the specific **TRNF-RES** is higher than its respective **RULES**.

The second set of experiments targets understanding the effects of using a limited number of rules. This setup is expected when we have a true low-resource scenario with a very limited number of examples and linguistic knowledge: does de-

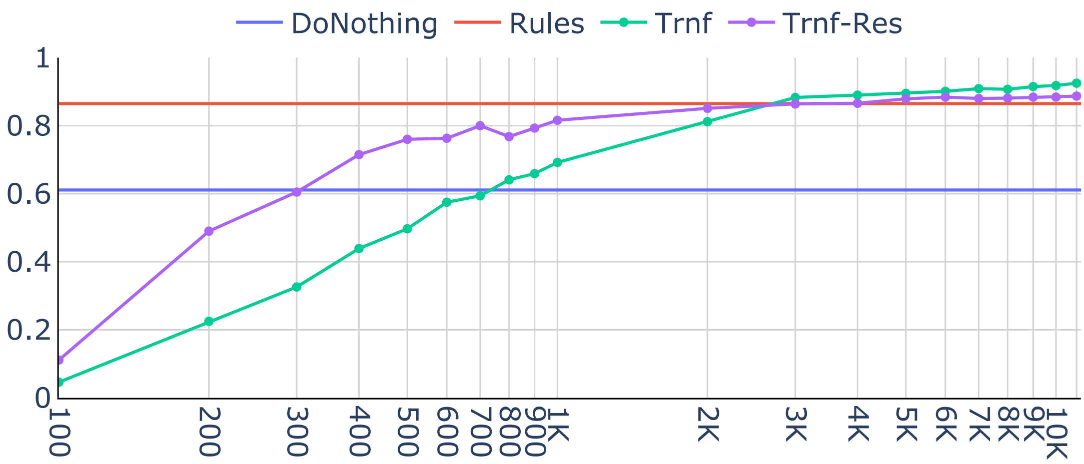

Figure 2: Accuracy on OOV-DEV for all baselines and systems across different sets of training sizes. The x-axis is in log scale. Note that **RULES** scores don't change; the rules are derived independently of the training data.

vising a small number of hand-crafted rules make a difference? We train **TRNF-RES** on the residuals of applying the rules incrementally, starting with the first rule and up to all six rules, following the order provided by the linguist. We focused on the lower training sizes as shown in Table 1. First, we observe that across the board, **TRNF** is consistently behind all **TRNF-RES** models, which suggests that in low-resource settings, any number of rules improves performance over a transformer alone. Second, we see that in the lowest half of the learning curve, rules alone outperform their **TRNF-RES** counterparts. However, starting at 600 examples, **TRNF-RES** transformers catch up to or outperform the rules which they have learned to correct.

7.1 Error Analysis

We conducted an error analysis for all baselines and systems at training size 900 where **TRNF-RES-4** outperformed both **TRNF** and **RULES-4** with accuracies 78.6%, 65.9%, and 73.2%, respectively. We sampled 200 entries from the OOV-DEV and compared them for all three systems. First, we analyzed cases where **TRNF-RES-4** succeeds and **TRNF** fails (18% of the time). The errors committed by **TRNF** fall primarily into three classes: i) adding gemination, ii) dropping consonants, and iii) erroneous vowel operations. Within the same set of data points, the cases where **RULES-4** failed were mostly due to the absence of the missing 6th rule, with only two cases where the first five rules didn't cover the processes correctly. Second, we examined the set where **RULES-4** succeeded and **TRNF-RES-4** failed (7% of the time). In this case, the errors committed by **TRNF-RES-4** were less severe than those of **TRNF**. They were mostly wrong vowel operations with the occasional added gemination. These analyses suggest that training to learn the residual makes the resulting residual model less prone to hallucination errors, and it successfully recovers cases which rules have missed.

8 Conclusion

We study the problem of generating the spoken form of words from their underlying morphological representation for Egyptian Arabic. Our approach uses state-of-the-art character-level neural transformer models to train a "residual" model which mitigates the shortcomings of using rules based on linguistic knowledge.[3] We find that a comprehensive set of hand-crafted linguistic rules is more accurate than any neural system (full or residual) in very low-resource settings (< 1,000 words of training data). However, when we do not have a full rule set, but we do have around 600 to 1,000 training examples, the residual model outperforms both the rules it has been trained to improve on, and a neural model trained without rules.

We aim to explore this effort in the context of child language acquisition, where children efficiently hypothesize and generalize rules from limited input. Furthermore, we will examine scenarios with limited data from one dialect and linguistic knowledge from a closely related dialect and analyze how learning the residuals can facilitate knowledge transfer.

[3] https://github.com/slkh/morphophono-res

Acknowledgements

We thank Robert Hoberman, Mohammadsadegh Ghalayani, Seyed Abolghasem Mirroshandel, and Jordan Kodner for their helpful discussion and efforts. The authors would like to thank Stony Brook Research Computing and Cyberinfrastructure, and the Institute for Advanced Computational Science at Stony Brook University for access to the SeaWulf computing system, made possible by grants from the National Science Foundation (#1531492 and Major Research Instrumentation award #2215987), with matching funds from Empire State Development's Division of Science, Technology and Innovation (NYSTAR) program (contract C210148). Rambow gratefully acknowledges support from the Institute for Advanced Computational Science at Stony Brook University.

References

Ernest T. Abdel-Massih, Zaki N. Abdel-Malek, and El-Said M. Badawi. 1979. *A Reference Grammar of Egyptian Arabic*. Georgetown University Press.

Daud Atiyeh Abdo. 1969. *Stress and Arabic phonology*. University of Illinois at Urbana-Champaign.

Mahasen Hasan Abu-Mansour. 1987. *A nonlinear analysis of Arabic syllabic phonology, with special reference to Makkan*. University of Florida.

Fadi Biadsy, Nizar Habash, and Julia Hirschberg. 2009a. Improving the Arabic Pronunciation Dictionary for Phone and Word Recognition with Linguistically-Based Pronunciation Rules. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 397–405, Boulder, Colorado.

Fadi Biadsy, Julia Hirschberg, and Nizar Habash. 2009b. Spoken Arabic Dialect Identification Using Phonotactic Modeling. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages (CASL)*, pages 53–61, Athens, Greece.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and Kemal Oflazer. 2018. The MADAR Arabic dialect corpus and lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Mohamed Boudchiche, Azzeddine Mazroui, Mohamed Ould Abdallahi Ould Bebah, Abdelhak Lakhouaja, and Abderrahim Boudlal. 2017. AlKhalil Morpho Sys 2: A robust Arabic morpho-syntactic analyzer. *Journal of King Saud University - Computer and Information Sciences*, 29(2):141–146.

Ellen Broselow. 1976. *The Phonology of Egyptian Arabic*. Ph.D. thesis. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2023-07-26.

Ellen Broselow. 2017. Syllable Structure in the Dialects of Arabic. *The Routledge handbook of Arabic linguistics*, pages 32–47.

Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. Harper & Row New York.

Nizar Habash, Ramy Eskander, and Abdelati Hawwari. 2012. A Morphological Analyzer for Egyptian Arabic. In *Proceedings of the Workshop of the Special Interest Group on Computational Morphology and Phonology (SIGMORPHON)*, pages 1–9, Montréal, Canada.

Nizar Habash, Salam Khalifa, Fadhl Eryani, Owen Rambow, Dana Abdulrahim, Alexander Erdmann, Reem Faraj, Wajdi Zaghouani, Houda Bouamor, Nasser Zalmout, Sara Hassan, Faisal Al shargi, Sakhar Alkhereyf, Basma Abdulkareem, Ramy Eskander, Mohammad Salameh, and Hind Saddiki. 2018. Unified Guidelines and Resources for Arabic Dialect Orthography. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Ghassan F Haddad. 1984. *Problems and issues in the phonology of Lebanese Arabic*. University of Illinois at Urbana-Champaign.

Abdel Halim Hamid. 1984. *A Descriptive Analysis of Sudanese Colloquial Arabic Phonology*. Ph.D. thesis. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2023-02-19.

Larry M. Hyman. 2018. Why underlying representations? *Journal of Linguistics*, 54(3):591–610.

Christian Khairallah, Salam Khalifa, Reham Marzouk, Mayar Nassar, and Nizar Habash. 2024. A large-scale open-source morphological analyzer for modern standard arabic. In *Proceedings of the joint LREC-COLING conference*.

Salam Khalifa, Sara Hassan, and Nizar Habash. 2017. A morphological analyzer for Gulf Arabic verbs. In *Proceedings of the Workshop for Arabic Natural Language Processing (WANLP)*, Valencia, Spain.

Salam Khalifa, Jordan Kodner, and Owen Rambow. 2022. Towards learning Arabic morphophonology. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Salam Khalifa, Sarah Payne, Jordan Kodner, Ellen Broselow, and Owen Rambow. 2023. A cautious generalization goes a long way: Learning morphophonological rules. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1793–1805, Toronto, Canada. Association for Computational Linguistics.

Hanaa Kilany, Hassan Gadalla, Howaida Arram, Ashraf Yacoub, Alaa El-Habashi, and Cynthia McLemore.

2002. Egyptian Colloquial Arabic Lexicon. LDC catalog number LDC99L22.

John J McCarthy. 2005. The length of stem-final vowels in colloquial arabic. *Perspectives on Arabic linguistics XVII–XVIII*, pages 1–26.

T.F. Mitchell. 1978. *An Introduction to Egyptian Colloquial Arabic*. Clarendon Press.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Dima Taji, Jamila El Gizuli, and Nizar Habash. 2018. An Arabic dependency treebank in the travel domain. In *Proceedings of the Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT)*, Miyazaki, Japan.

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, Online. Association for Computational Linguistics.

Mohammadzaman Zamani, H. Andrew Schwartz, Veronica Lynn, Salvatore Giorgi, and Niranjan Balasubramanian. 2018. Residualized factor adaptation for community social media prediction tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3560–3569, Brussels, Belgium. Association for Computational Linguistics.

A Transcription Map

Map: Arabic /IPA/ (LDC): أ ؤ إ ئ /ʔa/ (’), ب /b/ (b), ج /j/ (j), د /d/ (d), ه /h/ (h), و /w/ (w), ز /z/ (z), ح /ḥ/ (H), ط /tˤ/ (T), ي /y/ (y), ك /k/ (k), ل /l/ (l), م /m/ (m), ن /n/ (n), س /s/ (s), ع /ʕ/ (E), ف /f/ (f), ص /sˤ/ (S), ق /q/ (q), ر /r/ (r), ش /ʃ/ ($), ت /t/ (t), ة /-a(t)/ (a,at), ث /θ/ (v), خ /x/ (x), ذ /ð/ (*), ض /dˤ/ (D), غ /ɣ/ (g), ظ /ðˤ/ (Z), ـَ /a/ (a), ـُ /u/ (u), ـِ /i/ (i), ا ى /a:/ (A), و /u:/ (U), ي /i:/ (I).