Towards Learning Arabic Morphophonology

Salam Khalifa, Jordan Kodner, and Owen Rambow
Department of Linguistics, and
Institute for Advanced Computational Science (IACS)
Stony Brook University
{first.last}@stonybrook.edu

Abstract

One core challenge facing morphological inflection systems is capturing language-specific morphophonological changes. This is particularly true of languages like Arabic which are morphologically complex. In this paper, we learn explicit morphophonological rules from morphologically annotated Egyptian Arabic and corresponding surface forms. These rules are human-interpretable, capture known morphophonological phenomena in the language, and are generalizable to unseen forms.

1 Introduction

Much progress has been made in tasks such as morphological (re-)inflection and morphological analysis in recent years (e.g., Narasimhan et al., 2015; Kirov and Cotterell, 2018; Belth et al., 2021). However, low-resource languages still prove to be a significant challenge, despite growing interest, as the recent SIGMORPHON shared task reveals (Kodner et al., 2022; Kodner and Khalifa, 2022). Arabic dialects present a specific challenge in that there is an almost continual variation between dialects, mainly along the geographical dimension, and most dialects are low-resource. Cairene Arabic morphology is related to that of the dialects of (for example) Alexandria, Sohag, Aswan, and Khartoum. So from an NLP point of view, if we have knowledge of Cairene and need a morphological tool for one of the other dialects, we should be able to leverage our knowledge of Cairene. We propose to address the challenge by modeling morphophonological rules explicitly. Such rules provide an *explainable* representation of morphophonology. Once we have those rules, we can create NLP tools while leveraging rules from adjacent dialects. Having a standalone model of morphology, in terms of morphophonology or morphosyntax, improves the performance of many downstream NLP tasks such as machine translation (Sennrich and Haddow, 2016; Erdmann et al., 2019; Alhafni et al., 2020), speech synthesis (Halabi, 2016) and morphological disambiguation (Khalifa et al., 2020; Inoue et al., 2022). Morphological resources provide explicit linguistic knowledge that is not necessarily captured by learning models.

In this paper we present a preliminary study on automatically learning morphophonological rules for Cairene Egyptian Arabic (henceforth, EGY). We choose EGY because it is well-studied and has many resources. The learning process relies on the rule representation inspired by the notion of *phonological rules*, where a phonological alternation is explicitly represented via an input, an output, and the phonemic context. We evaluate our approach based the accuracy of the generated forms and the genralizability of the learned rules. Additionally, we describe the dataset preparation process as there is no suitable dataset for our task. To the best of our knowledge, this task of rule-learning to specifically model morphophonology has not been studied before in the context of Arabic NLP. This study will help us investigate to what degree can we learn explicit linguistic properties from simple representations.

2 Related Work

There have been many efforts on morphological modeling for Arabic. Precompiled tabular morphological analyzers (Buckwalter, 2002, 2004; Graff et al., 2009; Habash et al., 2012; Khalifa et al., 2017; Taji et al., 2018) became the standard in many Arabic NLP pipelines. While they provide rich morphological analysis, they are directly encoded into the lexicon and do not explicitly model descriptive linguistic phenomena such as morphophonological interactions. In contrast, earlier efforts that modeled morphology using finite-state technology (e.g., Beesley, 1998; Habash and Rambow, 2006) used explicit rules leveraging roots and patterns. However, they were manually built and were abstracted to a high degree. More re-

Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP), pages 295 - 301
December 8, 2022 ©2022 Association for Computational Linguistics

Dialect	Realization
Egyptian	kitabha
Sudanese	kitaaba
Hijazi	kitaabaha
Emirati	kitaabha

Table 1: Different realizations of the same underlying form /kitaab+haa/ 'her book' كتابها in four dialects.

cently, Habash et al. (2022) focused on modeling allomorphy through linguistically descriptive rules. However, the rules are manually created and do not model phonological representations. Other efforts adopting neural approaches to modeling morphological inflections (Wu et al., 2021; Dankers et al., 2021; Batsuren et al., 2022) perform well for many languages, however, those models do not provide insightful general rules or descriptions of linguistic phenomena. In this effort we take a generative view on morphophonology and we aim to learn morphophonological rules and apply them automatically.

3 Background

Morphophonology and Arabic Morphophonology is the study of the interaction between morphological and phonological processes. In particular, morphophonemic analysis aims at discovering the set of underlying forms and ordered rules that are consistent with the data it analyzes (Hayes, 2008).

Arabic morphophonology is especially interesting as its complex morphology is both templatic and concatenative. Morphophonological changes occur on the stem pattern and on stem and word boundaries. In the case of concatnative morphology, adding morphemes around the stem may trigger phonological changes. Most of these reinterpret the syllabic structure of utterances, and Arabic varieties may employ different processes to maintain such structures (Broselow, 2017). Table 1 shows how different varieties realize the same underlying representation: Egyptian, Sudanese, and Hijazi all employ different strategies to avoid a super-heavy syllable /-taab/, while Emirati permits it.

Rule Representation The transformation rules that we aim to extract are inspired by the *Sound Pattern of English* (SPE; Chomsky and Halle, 1968), where a hypothetical underlying representation (UR) is transformed into a surface form (SF) by the application of a series of rules. Below is an example of a phonological rule representing *r-dropping* in many dialects of British English, where r is dropped when it falls between a vowel and a syllable boundary $]_\sigma$.

$$\text{r} \rightarrow \emptyset \text{ / V \_\_\_ }]_\sigma$$
$$\text{UR} \rightarrow \text{SF / (context) \_\_\_ (context)}$$

Our work takes inspiration from the main three components of a rule, which are the UR, SF, and the context. The exact notion of rule, however, differs in order to make it machine-friendly. To this end, we take additional inspiration from two-level phonology (Antworth, 1991), which compresses stacks of SPE rules into a single UR and SF without intermediate steps.

4 Data

Our focus is on developing an *explainable* learning approach. Therefore, we control our experimental setup by having a few assumptions: a) we deal with whole words out of context, b) the data is in a broad phonetic transcription, c) SF is the word produced and UR is the morphologically segmented underlying representation, and d) the phonemic and morphemic inventories are assumed to be acquired beforehand (for example, by observing words in which the segmentation task is trivial).

Though EGY is resource-rich relative to many other varieties, there is no dataset that has been annotated for the task of morphophonological learning. To build such a dataset we need to create pairs of UR and SF to learn and evaluate morphophonological rules. In this work, we employ two existing resources created specifically for EGY: ECAL, and CALIMA_{EGY}.

4.1 Resources

The Egyptian Colloquial Arabic Lexicon (**ECAL**; Kilany et al., 2002) is a pronunciation dictionary primarily based on CALLHOME Egypt (Gadalla et al., 1997). Each entry in ECAL includes an orthographic, phonological, and morphological representation (Table 2(a)). Phonological forms represent SF. The orthography is undiacritized, and ECAL does not provide a full morphological segmentation. Therefore, we cannot use ECAL alone to extract URs, and we employ a separate resource in order to generate a hypothesized UR with morpheme boundaries.

CALIMA_{EGY} (Habash et al., 2012) is a morphological analyzer that generates a set of possible analyses for a given input token out of context. Each analysis includes a diacritized orthographic

(a)

ECA	Arabic	Pronunciation	lemma:morph
mafatIHu	مفاتيحه	m@f@tIHu	muftAH:noun+masc-inan-plural+gen-3rd-masc-sg

(b)

diac	lemma	BW	POS	gen	num	enc0
مَفاتِيحُه	مُفْتاح	POSS_PRON_3MS/هُ+NOUN/مَفاتِيح	noun	m	s	3ms_poss

Table 2: An example of a partial entry from ECAL in (a). An example of a partial entry from CALIMA$_{EGY}$ in (b).

form, morphological segmentation, and morphological features. We leverage the segmentation provided through CALIMA$_{EGY}$ as the starting point for a UR to the SF extracted from ECAL.

4.2 Dataset Creation

We generate a UR from CALIMA$_{EGY}$ for every SF extracted from ECAL. We use the CamelTools (Obeid et al., 2020) analyzer engine. We feed in the ECAL orthographic form to generate all the possible analyses. We then automatically choose the best matching analysis based on the orthography, lemma, part-of-speech (POS), and morphological features from both resources. Tables 2(a,b) show the necessary information used from both resources for the word /mafatiiħu/ 'his keys' مفاتيحه. Once the best analysis is chosen, the segmentation is extracted from the Buckwalter fine-grained POS tag (Buckwalter, 2002) generated as part of the CALIMA$_{EGY}$ analysis.

Forms are normalized to approximate UR forms. Only *stem-bound* morphophonological sound changes, i.e., entirely predictable changes, are normalized. These included changes such as unconditioned /q/ > /ʔ/ and the distinction between emphatic and non-emphatic vowels. Another aspect to take into consideration is the hypothesized underlying representation of the affixes and clitics. Some morphemes, such as the 2.fem.sg clitic /ik/, can have two forms, [ik] or [kii] depending on the last segment in the stem. In such cases, we remained faithful to the form provided by CALIMA$_{EGY}$ which is always consistent.

Finally, we enrich the segmentation provided by the analyzer, delimiting prefixes with -, suffixes with =, and word boundaries with #.[1] Table 3(a) shows an example of the final (UR,SF) pair. When generating the final set of UR and SF pairs, we only picked entries that belong to the open-class POS, i.e., verbs, nouns, and adjective.

[1]We do not make a distinction between affixes and clitics boundaries because we discovered that it does not significantly affect the learning process.

Data

(a)

UR	SF	
#mafAtIH=uh#	#mafatIHu#	مَفاتِيحُه

Locate alternations in UR

(b) #mafAtIH=uh#

Fit to smallest context

(c) afAtIH=uh#

Generalize

(d) aCACIC=uh#

Rule

(e)

LHS	RHS
aCACIC=uh# → aCaCICu#	

Table 3: This figure shows process of rule extraction starting from (UR,SF) pairs. An entry from the dataset in (a). In (b), different alternations are located through Levenshtein distance, morpheme boundary changes are in blue, and phonemic changes are in red to visualize the changes. We then reduce UR to the smallest context in (c). Followed by generalizing the stem consonants in (d). In (e) we see the final form of the rule.

4.3 Splits

ECAL was based on a continuous text corpus and indicates the occurrences of the entries in each of the three splits in the original corpus, namely, TRAIN, DEV, and EVAL. Because the splits are based on running text, words may re-occur in each of the splits. So in addition to DEV and EVAL, we also create DEV-OOV and EVAL-OOV by removing any overlap with TRAIN as shown in Table 4(a).

5 Learning Approach

We frame the learning problem as learning simple transformation rules that capture morphophonological interactions in a given dataset of (UR,SF) pairs.

5.1 Rule Extraction

We employ a simple rule learning mechanism that consists of extracting string transformations and converting those transformations into *rules* learned from TRAIN. In the first round, the string transformations are captured by calculating Levenshtein distance on every (UR,SF) to extract edit operations. Those edit operations are then used to locate the positions of where alternations are happening. Alternations to both the morpheme boundaries and phonemes are being considered. Levenshtein edits return whole word contexts. In order to improve generalizability, we choose a window of 2 around an alternation, if more than one alternation occur, then the window will be around the smallest substring that contains all alternations. And to further generalize, all consonants of the stem are replaced with a generic *C* character. The vowels and all other morphemes remain fully specified. Note that in the original notion of rules mentioned in §3, the rule always corresponds to a single change, so several rules may have to apply in sequence to yield the appropriate SF. However, in our adaptation, a single rule captures all changes simultaneously.

A rule in our definition consists of two components: the left-hand side (LHS) of the rule which represents the UR and context, and the right-hand side (RHS) of the rule which represents the SF. In case no change occurs, i.e., UR and SF are identical, in other words, the only alternations are deletions of morpheme boundaries, then the rule is reduced to UR→*copy*. Every rule has an accompanying frequency which is the number of (UR,SF) pair types that generated this rule. Table 3(e) shows an example of a rule. After rule extraction and generalization we ended up with 4,661 rules, which is 35.4% of the size of TRAIN.

5.2 Rule Selection

Since rules are specified by a limited context, it is often the case that more than one rule could apply to a given UR. Finding a rule that matches a given UR and produces the correct SF is not a trivial task. At this stage of our study, we employ a simple heuristic to select the most fitting rule. For a new UR, the longest and the most *specific* LHS is chosen. Specificity is determined by the least amount of unspecified consonants in the stem, i.e., the least number of *C*s. If the chosen LHS is found to participate in multiple rules, then the most frequent rule is applied.

(a)

	TRAIN	DEV	EVAL
All	13,170	5,180	6,974
OOV	–	2,189	2,271

(b)

	TRAIN	DEV	EVAL
All	90.1%	80.5%	82.3%
OOV	–	69.4%	68.9%

Table 4: Accuracy on each data split in (a). **All** represents all the types belonging to the respective splits as indicated in ECAL. **OOV** represents the same splits excluding types which also occur in TRAIN in (b).

6 Evaluation

To evaluate our current rule learning approach, we compute the accuracy of the generated SF for every UR, reported in Table 4(b). TRAIN accuracy is reported for the purpose of validating the generalizability of the rules. Performance is under 100% because of the rule abstraction process and rule selection heuristic. Even in TRAIN, there are words to which multiple rules can apply.

Two numbers are provided for both DEV and EVAL. The numbers in the **All** group indicate performance on the full splits. These indicate likely performance in future downstream tasks applied to running text, however, these contain words which were also present in TRAIN, so they are not in themselves a good indicator of our model's ability to generalize to unseen words. The out-of-vocabulary (**OOV**) numbers only report accuracy on types that were unseen during training. They retain most of their performance, indicating that the rules that our model learns do apply to new types.

7 Discussion and Error Analysis

The results discussed in §6 are good indicators of the generalizability of all the components of our approach, including rule representation, extraction, and selection heuristics. We performed a qualitative error analysis to further verify the generalizability and linguistic validity of the acquired rules.

Sources of Errors We investigated sources of errors in the SF production by comparing the rules the were selected with the ground-truth rules of the 31% of DEV-OOV forms that were incorrectly produced. The ground-truth rules were classified as either *in-vocabulary* rules (INV-rules) which exist in the acquired rule inventory or *out-of-vocabulary* rules (OOV-rules) which do not. Of the errors, 32% misproduced words had INV-rules. The selection heuristic is the driving source of this error: in the overwhelming majority of cases, the most specific LHS was selected. On the other hand, 68% of the errors had OOV-rules, which means that those rules were never seen before. We investigated 100 of those rules (30%). We found that all phenomena that those rules capture are in fact already captured in existing rules, but the context of the alternation is new, and therefore, the LHS is deemed unseen. This investigation emphasizes the crucial roles of the rule search heuristic and choice of the context.

Linguistic Phenomena To reaffirm the value of learning morphophonology through rules, we analyzed the top 60 (*non-copy*) most frequent rules. The most frequent rule in this sample had a frequency of 166 and the lowest was 15. We describe the captured phenomena in the following points:

- Word-final long vowel shortening.
- Assimilation of determiner-final /l/ to a stem-initial coronal. The "sun" and "moon" letter rule.
- Shortening of stem /aa/ in certain patterns.
- Epenthetic /u/ and /a/ to break *CCC* clusters.
- Deletion of stem-initial glottal stop after a prefix.
- Lengthening of the feminine suffix marker /a/ when it attaches to some pronominal suffixes in active participles.
- Deletion of word-final /h/ in the 3.masc.sg /-uh/.
- Deletion of /i/ in the active participles of the pattern /CACiC/ before a pronominal suffix.

Those findings mirror descriptive phonology for EGY (Abdel-Massih et al., 1979; Broselow, 2017). The small number of phenomena we found in the rules highlights once again the importance of determining the optimal context. This is a matter we are currently investigating.

8 Conclusion and Future work

In this paper we presented a morphophonological learning approach for Egyptian Arabic. The main goal was to learn morphophonological rules from pairs of underlying representations and surface forms. We achieved this goal with a production accuracy of 82% on the evaluation set and 68% on completely unseen tokens from the same set. Additionally, the linguistic phenomena captured through the rules align with the descriptive grammars of Egyptian Arabic. This effort also resulted in a new dataset designed specifically for this task. The dataset was generated by combining relevant information from a pronunciation lexicon and an orthography-based morphological analyzer.

In ongoing work, we continue to develop the crucial components of our rule learning approach. We are focusing on developing a more dynamic approach to determine the context of a change and the degree of phone abstraction. We will validate our approach by applying it to more dialects, including dialects with very scarce resources. Additionally, in low resource simulated settings, we plan to investigate the cognitive plausibility of the rules which will give insights to child acquisition of morphophonological phenomena.

Acknowledgements

This project is part of a larger effort taking place at the Department of Linguistics at Stony Brook, we would like to thank the members of the research group Ellen Broselow, Robert Hoberman, Jeff Heinz, and Daniel Greeson for their feedback. The authors of this paper were partially funded by a Stony Brook University research seed grant. Some experiments were performed on the SeaWulf HPC cluster maintained by RCC, and IACS at Stony Brook University and made possible by NSF grant #1531492.

References

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

Bashar Alhafni, Nizar Habash, and Houda Bouamor. 2020. Gender-aware reinflection using linguistically enhanced neural models. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 139–150, Barcelona, Spain (Online). Association for Computational Linguistics.

Evan L Antworth. 1991. Introduction to two-level phonology. *Notes on Linguistics*, 53:4–18.

Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinovic, Kyle Gorman, Zdeněk Žabokrtský, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, Fausto Giunchiglia, Ryan Cotterell, and Ekaterina Vylomova. 2022. The SIGMORPHON 2022 Shared Task on Morpheme Segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 103–116, Seattle, Washington. Association for Computational Linguistics.

Kenneth Beesley. 1998. Arabic morphology using only finite-state operations. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages (CASL)*, pages 50–7, Montereal.

Caleb A Belth, Sarah RB Payne, Deniz Beser, Jordan Kodner, and Charles Yang. 2021. The greedy and recursive search for morphological productivity. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 43.

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

Tim Buckwalter. 2002. Buckwalter Arabic morphological analyzer version 1.0. Linguistic Data Consortium (LDC) catalog number LDC2002L49, ISBN 1-58563-257-0.

Tim Buckwalter. 2004. Buckwalter Arabic Morphological Analyzer Version 2.0. LDC catalog number LDC2004L02, ISBN 1-58563-324-0.

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

Verna Dankers, Anna Langedijk, Kate McCurdy, Adina Williams, and Dieuwke Hupkes. 2021. Generalising to German plural noun classes, from the perspective of a recurrent neural network. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 94–108, Online. Association for Computational Linguistics.

Alexander Erdmann, Salam Khalifa, Mai Oudah, Nizar Habash, and Houda Bouamor. 2019. A Little Linguistics Goes a Long Way: Unsupervised Segmentation with Limited Language Specific Guidance. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 113–124, Florence, Italy. Association for Computational Linguistics.

Hassan Gadalla, Hanaa Kilany, Howaida Arram, Ashraf Yacoub, Alaa El-Habashi, Amr Shalaby, Krisjanis Karins, Everett Rowson, Robert MacIntyre, Paul Kingsbury, David Graff, and Cynthia McLemore. 1997. CALLHOME Egyptian Arabic transcripts LDC97T19. Web Download. Philadelphia: Linguistic Data Consortium.

David Graff, Mohamed Maamouri, Basma Bouziri, Sondos Krouna, Seth Kulick, and Tim Buckwalter. 2009. Standard Arabic Morphological Analyzer (SAMA) Version 3.1. Linguistic Data Consortium LDC2009E73.

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, Reham Marzouk, Christian Khairallah, and Salam Khalifa. 2022. Morphotactic modeling in an open-source multi-dialectal Arabic morphological analyzer and generator. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 92–102, Seattle, Washington. Association for Computational Linguistics.

Nizar Habash and Owen Rambow. 2006. MAGEAD: A morphological analyzer and generator for the Arabic dialects. In *Proceedings of the International Conference on Computational Linguistics and the Conference of the Association for Computational Linguistics (COLING-ACL)*, pages 681–688, Sydney, Australia.

Nawar Halabi. 2016. *Modern standard Arabic phonetics for speech synthesis*. Ph.D. thesis, University of Southampton.

Bruce Hayes. 2008. *Introductory Phonology*. Blackwell Textbooks in Linguistics. Wiley.

Go Inoue, Salam Khalifa, and Nizar Habash. 2022. Morphosyntactic Tagging with Pre-trained Language Models for Arabic and its Dialects. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1708–1719, Dublin, Ireland. Association for Computational Linguistics.

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, Nasser Zalmout, and Nizar Habash. 2020. Morphological Analysis and Disambiguation for Gulf Arabic: The Interplay between Resources and Methods. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3895–3904, Marseille, France. European Language Resources Association.

Hanaa Kilany, Hassan Gadalla, Howaida Arram, Ashraf Yacoub, Alaa El-Habashi, and Cynthia McLemore. 2002. Egyptian Colloquial Arabic Lexicon. LDC catalog number LDC99L22.

Christo Kirov and Ryan Cotterell. 2018. Recurrent neural networks in linguistic theory: Revisiting pinker and prince (1988) and the past tense debate. *Transactions of the Association for Computational Linguistics*, 6:651–665.

Jordan Kodner and Salam Khalifa. 2022. SIGMORPHON–UniMorph 2022 shared task 0: Modeling inflection in language acquisition. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 157–175, Seattle, Washington. Association for Computational Linguistics.

Jordan Kodner, Salam Khalifa, Khuyagbaatar Batsuren, Hossep Dolatian, Ryan Cotterell, Faruk Akkus, Antonios Anastasopoulos, Taras Andrushko, Aryaman Arora, Nona Atanalov, Gábor Bella, Elena Budianskaya, Yustinus Ghanggo Ate, Omer Goldman, David Guriel, Simon Guriel, Silvia Guriel-Agiashvili, Witold Kieraś, Andrew Krizhanovsky, Natalia Krizhanovsky, Igor Marchenko, Magdalena Markowska, Polina Mashkovtseva, Maria Nepomniashchaya, Daria Rodionova, Karina Scheifer, Alexandra Sorova, Anastasia Yemelina, Jeremiah Young, and Ekaterina Vylomova. 2022. SIGMORPHON–UniMorph 2022 shared task 0: Generalization and typologically diverse morphological inflection. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 176–203, Seattle, Washington. Association for Computational Linguistics.

Karthik Narasimhan, Regina Barzilay, and Tommi Jaakkola. 2015. An unsupervised method for uncovering morphological chains. *Transactions of the Association for Computational Linguistics*, 3:157–167.

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.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, volume 1, pages 83–91.

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.