DialG2P: Dialectal Grapheme-to-Phoneme. Arabic as a Case Study

Majd Hawasly[1], **Hamdy Mubarak**[1], **Ahmed Abdelali**[2], **Ahmed Ali**[2]
[1] Qatar Computing Research Institute, HBKU, Doha, Qatar
[2] Humain, Riyadh, Saudi Arabia
mhawasly@hbku.edu.qa

Abstract

Grapheme-to-phoneme (G2P) models are essential components in text-to-speech (TTS) and pronunciation assessment applications. While standard forms of languages have gained attention in that regard, dialectal speech, which often serves as the primary means of spoken communication for many communities, as it is the case for Arabic, has not received the same level of focus. In this paper, we introduce an end-to-end dialectal G2P for Egyptian Arabic, a dialect without standard orthography. Our novel architecture accomplishes three tasks: (i) restores short vowels of the diacritical marks for the dialectal text; (ii) maps certain characters that happen only in the spoken version of the dialectal Arabic to their dialect-specific character transcriptions; and finally (iii) converts the previous step output to the corresponding phoneme sequence. We benchmark G2P on a modular cascaded system, a large language model, and our multi-task end-to-end architecture.

1 Introduction

Acquiring accurate pronunciation is essential for both text-to-speech (TTS) and mispronunciation detection and diagnosis (MDD). Mapping graphemes (written symbols) to phonemes (spoken sounds) —- the grapheme–to–phoneme (G2P) task —-involves predicting the correct pronunciation of a word from its written form. This can be challenging due to inconsistencies between the written and spoken formats of a language (Bisani and Ney, 2008; Peters et al., 2017; Rao et al., 2015; Yao and Zweig, 2015). The G2P task is language-dependent and is affected by many language-specific factors, like the script, phonotactic constraints, and other orthographic factors (Frost and Katz, 1992; Li et al., 2022).

In TTS, the phonemizer is an important component in the front-end pipeline to convert text to phoneme sequence, which is used to train acoustic models that generate speech (Tan et al., 2021). Furthermore, in MDD, G2P is crucial for pronunciation assessment and scoring as it is needed to measure phoneme error rate (PER), to help language learners improve both perception and production of phonemes, and to develop awareness and tolerance for phoneme variations (Rogerson-Revell, 2021).

Bisani and Ney (2008) introduced joint-sequence models using a probabilistic framework that is applicable to G2P, used maximum approximation in training and n-best list for generation, along with confidence score for G2P. On the other hand, Sequence-to-sequence (Seq2Seq) has proven to be effective for machine translation tasks. Yao and Zweig (2015) deployed Seq2Seq in G2P and got a good boost in performance using bi-directional long short-term memory (BiLSTM) neural networks that use the same alignment information as machine translation (MT) approaches. While previous methods focused on well-resourced languages, Li et al. (2022) applied zero-shot learning to approximate G2P models for low-resource languages, building a language family tree to identify top-K nearest languages, to leverage their training sets. Their method was tested on over 600 unseen languages and outperformed baselines.

Arabic is typically written without diacritics (or short vowels). Diacritization (aka vowelization or diacritics restoration) is one of the major challenges in Arabic natural language processing (NLP) due to the complexity of Arabic morphology. The absence of diacritics causes ambiguity in morphological, phonological, syntactic, and semantic levels. Arabic can be divided into three main varieties, namely **Modern Standard Arabic (MSA)**: the language used in newspapers, books, and formal speeches;

Proceedings of The Third Arabic Natural Language Processing Conference, pages 466–471

Classical Arabic (CA): the language of historical books; and **Dialectal Arabic (DA)**: the spoken language in daily communications and is also widely used on social media. MSA is the official language in the 22 Arab countries, and there are 34 variations of the Arabic spoken dialects[1], that could be classified into five coarse-grained groups, namely: Egyptian, Levantine, Gulf , Maghrebi , and Iraqi (Cotterell and Callison-Burch, 2014), or per-country dialects (Mubarak and Darwish, 2014; Abdelali et al., 2021). Recent attempts to address diacritization in MSA and dialects with a neural architecture include (Elmallah et al., 2024).

Biadsy et al. (2009) investigated MSA G2P where they proposed linguistically motivated pronunciation rules cascaded with an automatic vowlizer to the written text, and their method showed superior performance in phoneme error rate. Motivated by that work, Ali et al. (2014) introduced Vowelization to Phonemes (V2P) pipeline with some changes to the original mapping, and released the first public Arabic pronunciation lexicon[2] which led to significant improvement in Arabic automatic speech recognition (ASR). Dialectal vowelization and phonemization were studied in (Harrat et al., 2013, 2014) using rule-based and statistical approaches applied to the Algiers dialect. Finally, Al-Haj et al. (2009) studied pronunciation modeling for Iraqi-Arabic using weights computed via forced alignment, which showed an improvement in the word error rate (WER).

We build on previous contributions and introduce an end-to-end model for G2P for dialectal Arabic that combines vowelization and phonemization together, along with dialectal support for various pronunciations. We assess our method on EGY. Unlike previous studies, which were tuned for specific dialects, our method and the techniques used here are generic enough and can be applied to any language or dialect with similar challenges. Our contributions are:

- We propose a new method that combines vowelization with dialect-specific special sounds;
- We evaluate a large language model (LLM) for the dialectal phoneme recognition task;
- We share the first testset that combines the diacritization and verbatim pronunciation of Egyptian tweets.

[1]Ethnologue: www.ethnologue.com/browse/names

[2]https://catalog.ldc.upenn.edu/LDC2017L01

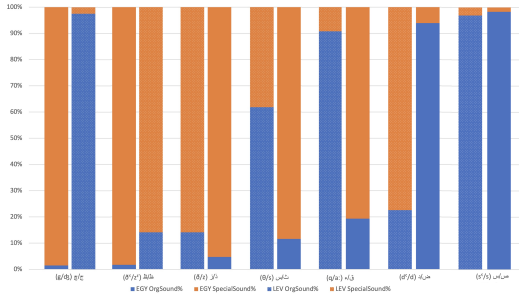

Figure 1: Distribution of use of special sounds in our data in the Egyptian (left bar) and Levantine (right bar) dialects. Blue shows the original sound of the character, while brown shows the modified special sound in the corresponding dialect.

2 Data

For EGY diacritization, we randomly selected 10,000 EGY tweets from QADI corpus (Abdelali et al., 2021). We gave clear guidelines to a native speaker (expert linguist) to fully diacritize the text, and provide the verbatim pronunciation according to the dialect spoken in Cairo, the capital. Here is an example of the output[3]: فِيهْ حَاجَة / حَاچَة دِلْوَقْتِي / دِلْوَءْتِي عَنْ اِلسَّوْرَةْ / اِلثَّوْرَةْ ([fiːh ħaːjah/ħaːgah dilwaqtiː/ dilwaʔtiː ʕan ilθawrah/ilsawrah], *There is something now about the revolution*). Some verbatim sounds can be written using the Arabic alphabet, e.g., changing ق (q) to ء (') as in قُلْت / ءُلْت (qult/'ult, *I said*). In addition, there are some sounds that are borrowed from other languages and do not exist in the original Arabic alphabet, namely چ ڤ پ (g, v, p) as in چوچل، ڤيتامين، سپراي (*Google, vitamin, spray*). We use the term "**Special Sounds**" to refer to all the changed sounds that exist in the Arabic alphabet or are borrowed.

We asked another native speaker from Syria familiar with other LEV dialects to diacritze and provide verbatim pronunciation for a random sample of 10,000 LEV tweets from QADI. We followed the same sound mappings in EGY and LEV dialects. Figure 1 shows differences in the usage of special sounds between EGY and LEV dialects. For example, pronouncing ج (j) in EGY as چ (g) is used dominantly (98.5%), and the rare case is the opposite as in جينات ([jiːnāt], *genes*). On the other hand, in LEV, we found the pronunciation of ج (j) as چ (g) is an exception (2.46%) in cases like جْرُوبْ / چْرُوبْ ([jruːb/gruːb], *group*). The most

[3]Format: written-word/spoken-word.

common special sound between EGY and LEV dialects is pronouncing ق (q) as ء (') with percentages equal to 80.56% and 77.37%, respectively[4].

2.1 Arabic Phoneme Prediction

Languages are often categorized along a spectrum ranging from "transparent" or "shallow" to "opaque" or "deep." In a transparent orthography, G2P mapping is consistent and direct. In an opaque orthography, this relationship is less predictable (Jiampojamarn and Kondrak, 2010; Kaplan and Kay, 1994). Arabic does have a relatively transparent alphabet in the sense that most letters correspond directly to specific sounds (Harrat et al., 2014).

While early work focused on Modern Standard Arabic, (Al-Ani, 1970) provided an early survey of Arabic phonemes and their acoustic mapping. This research was followed by further investigations using rule-based mapping of phonemes and graphemes. This work was typically performed on a small set of examples or limited datasets (Alghamdi et al., 2004; Al-Anzi and Abuzeina, 2017). Dictionaries of G2P were used as a tool for conversion. These resources were designed by linguists who often additionally covered dialectal variations (Harrat et al., 2014). Statistical approaches of language modeling were used for the transformation of written form of Arabic to its graphemic form; (Harrat et al., 2014) used SRILM (Stolcke, 2002) to build a model that mapped dialectal Arabic into grapheme representation.

3 Proposed Method

For dialectal G2P, we investigated seq2seq Transformer model using an attention mechanism. The transformer setup comprises an attention-based sequence-to-sequence transformer (Vaswani et al., 2017) followed by a 1-to-1 character-to-phoneme mapping. Figure 2 shows the system overview.

3.1 Data Pre-processing

The input text is preprocessed following the convention introduced in (Mubarak et al., 2019b) and (Mubarak et al., 2019a). A special sentence start token, repeated six times, and a special sentence end token also repeated six times, are added to the sentence. A sliding window of size 7 extracts lines of a fixed length of seven words/tokens. An example can be seen in Figure 2. The resulting lines are then tokenized into individual letters, and a special symbol is added for word separation.

3.2 Architecture

The transformer model has an encoder-decoder architecture with six layers, 512 hidden units, and 8 self-attention heads per layer. It is multi-task trained to predict the suitable diacritic mark per letter, and, based on context, to substitute certain letters with other letters or special characters added to the vocabulary to capture unique sounds that do not conform to standard Arabic pronunciation.

3.3 Post-processing and Phoneme Mapping

Due to the moving window, every word is presented to the transformer model seven times with different contexts. A simple majority voting mechanism is employed to choose a final representation of each letter in every word. Finally, the 1-to-1 character-to-phoneme mapping replaces the resulting characters with their corresponding phoneme sequences.

3.4 Training

We use a dataset of 10,000 manually-diacritized tweets in Egyptian dialectal Arabic, and a handcrafted rule set to substitute certain letters with alternative/special characters to capture their different dialectal pronunciation, extracted from the statistics in Figure 1. The data is randomly split into training, validation and testing sets with an 80-10-10 ratio. The transformer is trained for 300,000 steps with a batch size of 512 and LazyAdam optimizer (TensorFlow, 2019) to handle sparsity. We shall share the test split with the community.

3.5 Baselines

To benchmark DialG2P on the testing dataset of 1000 tweets, we introduce a number of baselines:

Transformer A similar transformer model that was trained on the single task of diacritization using the same data split.

GPT-4 We tested a zero-shot and a few-shot prompt on GPT-4 to only predict the diacritization. GPT-4 did not give good results in restoring the special sounds, so we used the default special sounds (defSS) as shown in Figure 1 to replace the sounds that are always changed in 80% of the cases. The few-shot prompt is:

`I will give you some tweets written in the Egyptian dialect, and their full diacritization. Input: <tweet text without diacritics>.`

[4]We release the diacritized tweet data from this work at https://github.com/qcri/DialG2P

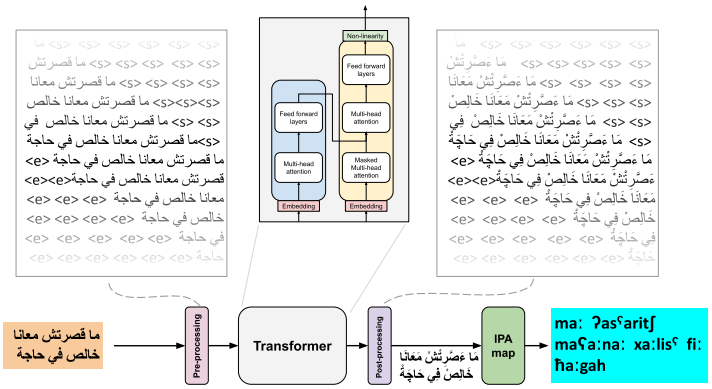

Figure 2: An overview of DialG2P approach.
Buckwalter transliteration of input: `mA qSrt$ mEAnA fy HAjp` , and output: `mA aSrt$ mEAnA fy HAgp`.

`Output: <fully diacritized tweet text>.`

`Now, diacritize this Egyptian Arabic tweet fully and write only the final diacritized tweet according to the Egyptian pronunciation: <input>`

Transformer cascade A cascade of the diacritization transformer `Transformer` and the special sound rule set used to generate the training data.

3.6 Metrics

We report 1) the standard **Word Error Rate** (`WER`) and 2) **Phoneme Error Rate** (`PER`). For analysis purposes, we also report 3) **Diacritic error rate** (`DER`): the number of diacritic marks that are different to the reference divided by their total number, and 4) **Character error rates** (`CER`): the number of different characters from the reference divided by the total number of characters.

3.7 Results

The experimental results for the proposed DialG2P model and various baselines on the Egyptian Arabic end-to-end G2P task are detailed in Table 1. The table provides a direct quantitative comparison of all tested models across critical metrics, offering a comprehensive view of performance at different granularities from word-level accuracy to character-level precision and diacritic restoration. DialG2P achieved a WER of 5.15%, PER of 1.71%, DER of 1.67%, and CER of 0.05%. These results place DialG2P nearly on par with the `Transformer cascade` model and ahead of `Transformer+defSS` baseline in WER, PER and CER. Notably, DialG2P achieved the lowest CER, indicating superior character-level accuracy in its output. However, there was a slight regression in the diacretization performance as compared to the specialized transformer, possibly indicating the reduced capacity. On the other hand, a capable LLM like GPT-4 struggles with the G2P task even when presented with 10 examples for in-context learning.

Model	*WER%*	*PER%*	*DER%*	*CER%*
`Transformer`	17.26	4.88	**1.62**	3.35
`Transformer+defSS`	6.32	2.02	**1.62**	0.41
`GPT-4 (0-shot)`	47.57	16.81	13.64	3.67
`GPT-4 (0-shot)+defSS`	40.71	14.27	13.64	0.69
`GPT-4 (10-shot)`	33.66	10.91	7.97	3.29
`GPT-4 (10-shot)+defSS`	25.14	8.23	7.97	0.32
`Transformer cascade`	**5.11**	**1.70**	**1.62**	0.09
`DialG2P`	5.15	1.71	1.67	**0.05**

Table 1: Word, phoneme, diacritic and character error rates for DialG2P and baselines.

4 Conclusions

The experiments highlight that dialectal G2P is a multi-faceted problem requiring solutions beyond standard diacritization. The successful integration of "special sound" handling, either through explicit rules or end-to-end mappings, is crucial to achieve high accuracy. The end-to-end multi-task approach of DialG2P offers a promising direction, demonstrating that complex dialectal phenomena can be effectively learned within a unified neural architecture, potentially simplifying the development compared to cascaded systems. While this study focused exclusively on Egyptian Arabic, the generic nature of the proposed technique suggests applicability to other dialects or languages with similar challenges. We plan to extend this work to other Arabic dialects.

Limitations

- This short paper focused on a single dialect (Egyptian Arabic) for its empirical evaluation.
- A single annotator was tasked with creating the training data for this work.
- A rule base was used to create the gold data with regard to character replacement.

References

Ahmed Abdelali, Hamdy Mubarak, Younes Samih, Sabit Hassan, and Kareem Darwish. 2021. QADI: Arabic dialect identification in the wild. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 1–10.

Salman H Al-Ani. 1970. *Arabic phonology: An acoustical and physiological Investigation*. The Hague, Mouton.

Fawaz S. Al-Anzi and Dia Abuzeina. 2017. The impact of phonological rules on Arabic speech recognition. *Int. J. Speech Technol.*, 20(3):715–723.

Hassan Al-Haj, Roger Hsiao, Ian Lane, Alan W Black, and Alex Waibel. 2009. Pronunciation modeling for dialectal Arabic speech recognition. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 525–528. IEEE.

Mansour Alghamdi, Husni Almuhtasib, and Mustafa Elshafei. 2004. Arabic phonological rules. *King Saud University Journal: Computer Sciences and Information*, 16:1–25.

Ahmed Ali, Yifan Zhang, Patrick Cardinal, Najim Dahak, Stephan Vogel, and James Glass. 2014. A complete KALDI recipe for building Arabic speech recognition systems. In *2014 IEEE spoken language technology workshop (SLT)*, pages 525–529. IEEE.

Fadi Biadsy, Nizar Habash, and Julia Hirschberg. 2009. Improving the Arabic pronunciation dictionary for phone and word recognition with linguistically-based pronunciation rules. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 397–405.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech communication*, 50(5):434–451.

Ryan Cotterell and Chris Callison-Burch. 2014. A multi-dialect, multi-genre corpus of informal written Arabic. In *LREC*, pages 241–245.

Muhammad Morsy Elmallah, Mahmoud Reda, Kareem Darwish, Abdelrahman El-Sheikh, Ashraf Hatim Elneima, Murtadha Aljubran, Nouf Alsaeed, Reem Mohammed, and Mohamed Al-Badrashiny. 2024. Arabic diacritization using morphologically informed character-level model. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1446–1454, Torino, Italia. ELRA and ICCL.

Ram Frost and Marian Katz. 1992. *Orthography, phonology, morphology and meaning*. Elsevier.

Salima Harrat, Mourad Abbas, Karima Meftouh, and Kamel Smaili. 2013. Diacritics restoration for Arabic dialects. In *INTERSPEECH 2013-14th Annual Conference of the International Speech Communication Association*.

Salima Harrat, Karima Meftouh, Mourad Abbas, and Kamel Smaïli. 2014. Grapheme to phoneme conversion-an Arabic dialect case. In *Spoken Language Technologies for Under-resourced Languages*.

Sittichai Jiampojamarn and Grzegorz Kondrak. 2010. Letter-phoneme alignment: An exploration. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 780–788, Uppsala, Sweden. Association for Computational Linguistics.

Ronald M. Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Xinjian Li, Florian Metze, David R Mortensen, Shinji Watanabe, and Alan W Black. 2022. Zero-shot learning for grapheme to phoneme conversion with language ensemble. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2106–2115.

Hamdy Mubarak, Ahmed Abdelali, Kareem Darwish, Mohamed Eldesouki, Younes Samih, and Hassan Sajjad. 2019a. A system for diacritizing four varieties of Arabic. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 217–222.

Hamdy Mubarak, Ahmed Abdelali, Hassan Sajjad, Younes Samih, and Kareem Darwish. 2019b. Highly effective Arabic diacritization using sequence to sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2390–2395.

Hamdy Mubarak and Kareem Darwish. 2014. Using twitter to collect a multi-dialectal corpus of Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 1–7.

Ben Peters, Jon Dehdari, and Josef van Genabith. 2017. Massively multilingual neural grapheme-to-phoneme conversion. *arXiv preprint arXiv:1708.01464*.

Kanishka Rao, Fuchun Peng, Haşim Sak, and Françoise Beaufays. 2015. Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4225–4229.

Pamela M Rogerson-Revell. 2021. Computer-assisted pronunciation training (capt): Current issues and future directions. *Relc Journal*, 52(1):189–205.

Andreas Stolcke. 2002. SRILM-an extensible language modeling toolkit. In *Seventh international conference on spoken language processing*.

Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. 2021. A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561*.

TensorFlow. 2019. Lazyadamoptimizer. https://github.com/tensorflow/tensorflow/blob/r1.13/tensorflow/contrib/opt/python/training/lazy_adam_optimizer.py. Accessed: 2025-07-06.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Kaisheng Yao and Geoffrey Zweig. 2015. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. *arXiv preprint arXiv:1506.00196*.