Gulf Arabic Diacritization: Guidelines, Initial Dataset, and Results

Nouf Alabbasi[1], **Mohamed AlBadrashiny**[2], **Maryam Aldahmani**[3], **Ahmed AlDhanhani**[4], **Abdullah Saleh Alhashmi**[4], **Fawaghy Alhashmi**[3], **Khalid Al Hashemi**[4], **Rama Alkhobbi**[5], **Shamma AlMaazmi**[6], **Mohammed Alyafeai**[7], **Mariam M. Alzaabi**[4], **Mohamed Alzaabi**[4], **Fatma Badri**[5], **Kareem Darwish**[2], **Ehab Mansour Diab**[8], **Muhammad Elmallah**[2], **Amira Elnashar**[9], **Ashraf Elneima**[2], **MHD Tameem Kabbani**[9], **Nour Rabih**[10], **Ahmad Saad**[11], **Ammar Mamoun Sousou**[5]

[1] New York University Abu Dhabi, UAE, [2] aiXplain Inc., CA, USA,
[3] United Arab Emirates University, AD, UAE, [4] Khalifa University, AD, UAE,
[5] Independent, [6] University of Sharjah, SH, UAE,
[7] Cyber Gate Defense, AD, UAE, [8] Invent Technology, AD, UAE,
[9] American University of Sharjah, SH, UAE, [10] King's College London, LDN, UK
[11] Mohammed Bin Rashid Space Centre, Dubai, UAE

naa475@nyu.edu, mohamed@aixplain.com, Maryam.aldahmani21@hotmail.com,
ahmed.aldhanhani@ku.ac.ae, nnh3@hotmail.com, 201800838@uaeu.ac.ae,
1000052995@ku.ac.ae, Ramaalkhobbi@gmail.com, shammaalmazmi@hotmail.com,
mohamed@cybergate.tech, 100044533@ku.ac.ae, m7mdz3abii@gmail.com,
fatmakbadri@gmail.com, kareem.darwish@aiXplain.com, ehab@invent-technology.net,
muhammad.elmallah@aiXplain.com, g00082075@aus.edu, ashraf.hatim@aixplain.com,
b00088948@aus.edu, noor_rabie@hotmail.com, ahmadateejuae@gmail.com, ammar13ma@gmail.com

Abstract

Arabic diacritic recovery is important for a variety of downstream tasks such as text-to-speech. In this paper, we introduce a new Gulf Arabic diacritization dataset composed of 19,850 words based on a subset of the Gumar corpus. We provide a comprehensive set of guidelines for diacritization to enable the diacritization of more data. We also report on diacritization results based on the new corpus using a word-based Hidden Markov Model and a character-based sequence to sequence model.

1 Introduction

Arabic has two types of vowels, namely long and short vowels. Although long vowels are explicitly written, short vowels, which take the form of diacritic marks, are typically omitted in written Arabic, and readers need to infer these diacritics to properly pronounce words. Thus, diacritic recovery, also referred to as diacritization, is important for downstream tasks such as text-to-speech and language learning. Most previous efforts pertaining to Arabic diacritic recovery have focused on Modern Standard Arabic (MSA) and Classical Arabic (CA). The focus on these Arabic varieties has been aided by the availability of large training corpora such as the Penn Arabic Treebank (Maamouri et al., 2004) and Tashkeela (Zerrouki and Balla, 2017) and relatively stable diacritization standards. There has been some efforts related to diacritizing different Arabic dialects such as Egyptian (Zalmout and Habash, 2020), Palestinian (Jarrar et al., 2017), Morrocan, and Tunisian (Mubarak et al., 2019). Many challenges face dialectal diacritization, mostly related to the availability of large consistent data. While MSA/CA corpora may be composed of millions of words, dialectal datasets have been capped at tens of thousands of words (Mubarak et al., 2019; Zalmout and Habash, 2020). Furthermore, diacritization of the same dialect may differ from town to town, complicating data standardization and consistency. For example, the word سخن (sxn[1] – hot) is diacritized in the Egyptian dialect as سُخُنْ (suxuno) in Alexandria and سُخْنْ (suxono) in Cairo. Variations in pronunciation of words are rather common within the same dialect in locales of close geographical proximity.

In this paper, we present a new public diacritized dataset for Gulf Arabic in accordance to the pronunciation of the city of Dubai in the United Arab Emirates (UAE). The dataset is a 19,850 words subset of the Gumar corpus (Khalifa et al., 2018), which is composed of roughly 200 thousand words from Emirati internet novels. To diacritize the cor-

[1] Buckwalter transliteration is used exclusively in the paper.

Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP), pages 356 - 360

pus, we conducted a workshop that included two senior computational linguists and 15 native speakers of the Emirati dialect to codify the diacritization guidelines and to actually diacritize the corpus. We split the corpus into training and test sentences, and we proceeded to build two different Emirati diacritizers using a word-based Hidden Markov Model (HMM) and a character-based sequence to sequence mapping architecture.

The contributions of this paper is as follows:

- We present a new dataset for Gulf Arabic diacritization based on the sub-dialect spoken in Dubai, UAE.
- We formalize guidelines for the diacritization of the dialect.
- We present initial results using 2 different diacritization models.

2 Related Work

Many approaches have been used for Arabic diacritization such as HMMs (Gal, 2002; Darwish et al., 2017), finite state transducers (Nelken and Shieber, 2005), character-based maximum entropy based classification (Zitouni et al., 2006), and a variety of deep learning approaches (Abandah et al., 2015; Belinkov and Glass, 2015; Mubarak et al., 2019; Rashwan et al., 2015). For MSA, most approaches tend to handle core word diacritics, which disambiguate a word in context, separately from case-endings, which typically appear at end of a word and specify the syntactic role of words. However, more recent approaches have resorted to guessing both types of diacritics jointly (Fadel et al., 2019; Mubarak et al., 2019) by either casting the problem as a character sequence labeling problem or as a character sequence to sequence (seq2seq) mapping respectively. Since the seq2seq models have a tendency to hallucinate, Mubarak et al. (2019) used a combination of limited context and voting to overcome the problem.

Unlike MSA, Arabic dialects generally omit case-endings and tend to apply *sukun* (o) on the last letter. Prior work on dialectal diacritization is rather scant. For dialectal Egyptian, Zalmout and Habash (2020) developed a morphological analyzer that also performs diacritization using sequence to sequence modeling. They reported a diacritization accuracy of 85.0%. For dialectal Gulf, Khalifa et al. (2017) developed a morphological analyzer for dialectal Gulf verbs, but diacritization was not the focus of their work. Khalifa et al. (2018) morphologically tagged an Emirati subset of the Gumar corpus including the diacritization of lemmas. However, mapping the diacritics from lemmas to words is non-trivial. For dialectal Palestinian, Jarrar et al. (2017) annotated a corpus of containing 43k words and diacritized all words. Abdelali et al. (2018); Darwish et al. (2018); Mubarak et al. (2019) used diacritized translations of the bible into dialectal Moroccan (151K words) and Tunisian (142K words) to train biLSTM over CRF, CRF only, and seq2seq models respectively for diacritizing both dialects. Of all three approaches, the seq2seq model led to the lowest word error rate (Moroccan: 1.4% and Tunisian: 2.5%).

3 Dataset

As mentioned earlier, we diacritized a subset of the Gumar corpus (Khalifa et al., 2018). The Gumar corpus is a collection of Internet novels composed of roughly 100 million words. A 200 thousand words subset of Gumar was in the Emirati dialect and was manually morphologically tagged. Though the lemmas were diacritized, their diacritization often did not correspond directly to the diacritization of words. Thus, we proceeded to diacritize a 19,850 word subset of the tagged Emirati portion of Gumar. We used the CODAfied version of the text, as opposed to the raw text, to have greater consistency in spelling. CODA, or Conventional Orthography for Dialectal Arabic, is an attempt at standardizing the spelling of different Arabic dialects (Habash et al., 2012).

For diacritization, we conducted a workshop that included two senior computational linguists and 15 native speakers of the Emirati dialect to codify the diacritization guidelines.

Diacritization Standards: After lengthy discussions, we settled on the following guidelines:

- All diacritization must be consistent with the accent spoken in Dubai, UAE.
- Leading *Hamza* in a closed set of words, such as ابو (>bw – father of) is not pronounced and hence undiacritized.
- Consecutive letters can have sukun, such as شِفْتُها ($ifotohA – I saw her)
- Words can start with *sukun*, such as بْيِلْعَب (boyiloEab – he plays). To ascertain if a word starts

with *sukun*, we use the *w* test, where the leading letter gets a *sukun* if it has a *sukun* when the letter *w* is added as a prefix.

- All words end with either *sukun*, which is assumed and subsequently dropped, or *shadda* (∼).
- In ambiguous cases, *kasra* (i) is prioritized over *fatha* (a), which is prioritized over *dammah* (u).
- *Sukun* over *Lam Alaqamrya* does not need to be explicitly put. Ex. القَمَر (Alqamar – the moon).
- The question word ش ($) always has a *dammah* (u).
- Coordinating conjunction letter و (w) in most cases has a *kasra* (i). Ex. وِقال (wiqAl – and he said).
- In ambiguous cases, plurality is prioritized over duality and that's because plurality occurs more, and the duality is a subset of the plurality.
- The singular masculine present tense marker ي (y) can only have *kasra* or *sukun*. Ex. يِلْعَب (yiloEab – he plays).
- Three letter past tense verbs are diacritized as فِعَل (fiEal), Except for verbs that start with ا (A). Ex. سِبَح (sibah – he swam).
- Some colors have specific diacritized forms, namely: حَمَر (H**a**m**a**r – red) and خَضَر (x**a**D**a**r – green).
- Default diacritics (*fatHa* followed by *alef*, *kasra* followed by *ya*, and *damma* followed by *wa*) are omitted.
- There is no need for a *kasra* for *hamza below alef* إ (<).
- *tanween fatha* (F) should come before the letter *alef* ا (A). Ex. طَبْعًا (TaboEFA – of course).
- For plural verbs that end with ﻭا (wA), the و (w) mostly has *sukun* and the letter before it has *fatha*. Ex. لِعْبَوْا (liEobawoA – they played).
- We used the MSA diacritics and did not introduce any new diacritic marks.

Diacritization Process: We used a three step diacritization process designed to increase speed and improve accuracy. The steps are as follows:

- We diacritized the most frequent 1,300 words in the annotated Emirati Gumar corpus out of context. Our intuition was that most words have either one diacritized form or one diacritized form that is more dominant, and the most common words would cover a large proportion of the text in the corpus. Some example words that we diacritized in this manner are: أَرْمِس (>aromis – I speak), غالْيَة (gAloyap – precisous), and ليش (ly$ – why)[2]. We used the word list to automatically diacritize the corpus.
- We split the native speakers in the workshop into 4 groups, and each group was responsible to diacritize a different subset of the corpus. The groups were instructed to work together and to resolve disagreements. Each group was given sentences that were roughly 5,000 words.
- A senior computational linguist who is well versed in the Gulf dialect performed two rounds of review over the work of all the groups with frequent consultations with members of the groups.

Table 1 shows three sample sentences after review. The newly diacritized portion of Gumar is 2,953 sentences, which is composed of 19,850 words. For subsequent experiments, we split the dataset into training and test splits. Table 2 shows the breakdown of the dataset.

4 Experiments

We trained two different diacritization models based on our new dataset. Prior to training the models, we tokenized all the text to separate all punctuation. The data did not have any emojis, URLs, or emails. The models were as follows:

HMM Model: As the name suggests, we used a Hidden Markov Model to find the best diacritization of words in context. We used KenLM[3] to train a word trigram language model and an in-house implementation of A-star search to ascertain the best path in the lattice.

Seq2seq Model: We re-implemented the setup that was suggested by Mubarak et al. (2019). The model used the RNN-based sequence to sequence model that is implemented in OpenNMT (Klein

[2]As can be seen from the example, we removed default diacritics

[3]https://github.com/kpu/kenlm

Sentence	Buckwalter transliteration	Translation
سيف : سَمَعْتيها شو قالَت لج ؟	syf : samaEotyhA $w qAlat lij	Saif: Did you hear what he she told you ?
وِقال خَليفَة : وِسَلْمِى عَليها بَعِد . . .	wiqAl xalyfap : wisalomaY EalyhA baEid	and Khalifa said: and Salma what about her.
جَزّوي : شو . . خَبْروني . .	jaz∼wy : $w ... xaborwny	Jazouy: what ... tell me.

Table 1: Example diacritized sentences.

split	Words	Sentences
Train	18,174	2,700
Test	1,676	253
Total	19,850	2,953

Table 2: Breakdown of diacritized dataset

Model	WER
HMM	6.7%
Seq2Seq	8.6%

Table 3: Diacritization results: Word Error Rate (WER)

et al., 2017), which is a neural machine translation toolkit, to translate undiacritized characters to diacritized characters. Since seq2seq models may hallucinate, we restricted contexts to 5 words instead of attempting to diacritize entire sentences and implemented voting across multiple contexts (Mubarak et al., 2019). The underlying model uses 2 unidirectional LSTM layers with 512 states and a dropout rate of 0.3. We also used 200 sentences from the training set as a validation set.

Table 3 shows the diacritization results of both models. As can be seen, the HMM model performed slightly better than the seq2seq model with 6.7% and 8.6% WER respectively. To understand the results, we proceeded to classify all the errors resulting from both approaches. We found that the most dominant errors for the HMM model were due to out of vocabulary words (OOVs), accounting for 73.3% of the errors. Given that we were using the CODAfied version of the Gumar corpus, we suspect that the OOV problem would be more pronounced for dialectal Gulf in the wild, where creative spellings would be more common. Conversely, hallucinations accounted for 34.6% of the errors for the seq2seq model. An example of hallucination is the word اُسْبوع (AusobwE – week) resulting in the misspelled version اَسْبِعوع (AasobiwE). We suspect that hallucination errors would be less pronounced if we had more training data. The results and error types seem to suggest that the dataset is relatively small, and more data is required to build more robust diacritizers. We hope that the newly annotated corpus with the associated diacritization standards can pave the way to building larger datasets.

5 Conclusion

In this paper we introduced a new dataset for Gulf Arabic diacritization based on the sub-dialect spoken in Dubai, UAE. The diacritization was based on formalized diacrtization guidelines that was developed by two senior computational linguists along with 15 native speakers, who were also instrumental in performing the actual diacritization. We plan to release the dataset publicly under an open source license. We also presented initial results using 2 different diacritization models. Though the dataset is relatively small (19,850 words), we were able to build two diacritization models that achieved less than 9% word error rate. We plan to expand the size of the corpus, particularly for non-CODAfied Gulf text. We hope that models trained on our data can help significantly speed up the diactization process.

6 Limitations

Some of the limitations include: 1) the corpus is based on one genre, namely Internet novels, that have limited linguistic diversity; 2) diacritization was done on the CODAfied subset of the Gumar corpus, while much naturally appearing text may not be CODA compliant; and 3) the dataset is relatively small and more data is required to train robust diacritization models (particularly deep learning models).

7 Ethics Statement

All the data that we annotated is in the public domain, and private data was used.

References

Gheith A Abandah, Alex Graves, Balkees Al-Shagoor, Alaa Arabiyat, Fuad Jamour, and Majid Al-Taee. 2015. Automatic diacritization of arabic text using recurrent neural networks. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(2):183–197.

Ahmed Abdelali, Mohammed Attia, Younes Samih, Kareem Darwish, and Hamdy Mubarak. 2018. Diacritization of maghrebi arabic sub-dialects. *arXiv preprint arXiv:1810.06619*.

Yonatan Belinkov and James Glass. 2015. Arabic diacritization with recurrent neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2281–2285, Lisbon, Portugal.

Kareem Darwish, Ahmed Abdelali, Hamdy Mubarak, Younes Samih, and Mohammed Attia. 2018. Diacritization of moroccan and tunisian arabic dialects: A crf approach. *OSACT*, 3:62.

Kareem Darwish, Hamdy Mubarak, and Ahmed Abdelali. 2017. Arabic diacritization: Stats, rules, and hacks. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 9–17.

Ali Fadel, Ibraheem Tuffaha, Mahmoud Al-Ayyoub, et al. 2019. Arabic text diacritization using deep neural networks. In *2019 2nd international conference on computer applications & information security (ICCAIS)*, pages 1–7. IEEE.

Ya'akov Gal. 2002. An hmm approach to vowel restoration in arabic and hebrew. In *Proceedings of the ACL-02 workshop on Computational approaches to Semitic languages*, pages 1–7. Association for Computational Linguistics.

Nizar Habash, Mona Diab, and Owen Rambow. 2012. Conventional orthography for dialectal Arabic. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Mustafa Jarrar, Nizar Habash, Faeq Alrimawi, Diyam Akra, and Nasser Zalmout. 2017. Curras: an annotated corpus for the palestinian arabic dialect. *Language Resources and Evaluation*, 51(3):745–775.

Salam Khalifa, Nizar Habash, Fadhl Eryani, Ossama Obeid, Dana Abdulrahim, and Meera Al Kaabi. 2018. A morphologically annotated corpus of emirati arabic. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

Salam Khalifa, Sara Hassan, and Nizar Habash. 2017. A morphological analyzer for gulf arabic verbs. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 35–45.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Mohammed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The penn arabic treebank: building a large-scale annotated arabic corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, pages 102—-109.

Hamdy Mubarak, Ahmed Abdelali, Kareem Darwish, Mohamed Eldesouki, Younes Samih, and Hassan Sajjad. 2019. 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.

Rani Nelken and Stuart M Shieber. 2005. Arabic diacritization using weighted finite-state transducers. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 79–86. Association for Computational Linguistics.

Mohsen Rashwan, Ahmad Al Sallab, M. Raafat, and Ahmed Rafea. 2015. Deep learning framework with confused sub-set resolution architecture for automatic arabic diacritization. In *IEEE Transactions on Audio, Speech, and Language Processing*, pages 505–516.

Nasser Zalmout and Nizar Habash. 2020. Joint diacritization, lemmatization, normalization, and fine-grained morphological tagging. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8297–8307.

Taha Zerrouki and Amar Balla. 2017. Tashkeela: Novel corpus of arabic vocalized texts, data for auto-diacritization systems. *Data in brief*, 11:147–151.

Imed Zitouni, Jeffrey S Sorensen, and Ruhi Sarikaya. 2006. Maximum entropy based restoration of arabic diacritics. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 577–584. Association for Computational Linguistics.