Data Augmentation for Speech-Based Diacritic Restoration

Sara Shatnawi[1], **Sawsan Alqahtani**[2], **Shady Shehata**[1], **Hanan Aldarmaki**[1]
[1]Mohamed bin Zayed University of Artificial Intelligence
[2]Princess Nourah Bint Abdulrahman University
[1]{sara.shatnawi;shady.shehata;hanan.aldarmaki}@mbzuai.ac.ae, [2]saalqhtani@pnu.edu.sa

Abstract

This paper describes a data augmentation technique for boosting the performance of speech-based diacritic restoration. Our experiments demonstrate the utility of this approach, resulting in improved generalization of all models across different test sets. In addition, we describe the first multi-modal diacritic restoration model, utilizing both speech and text as input modalities. This type of model can be used to diacritize speech transcripts. Unlike previous work that relies on an external ASR model, the proposed model is far more compact and efficient. While the multi-modal framework does not surpass the ASR-based model for this task, it offers a promising approach for improving the efficiency of speech-based diacritization, with a potential for improvement using data augmentation and other methods.

1 Introduction

The Arabic Language is one of the official international languages (al Yamin, 2023) ranking as the sixth language in world organizations, the third in the Organization of African Unity, and the first in the Islamic World (Bakalla, 2023). Arabic is the native language of more than 400 million people in Arabic countries and is spoken by more than 5 million speakers beyond the Arab world (Bakalla, 2023). The Arabic script consists of 28 basic alphabets, representing consonants and long vowels. As short vowels are not represented in the alphabet or the basic construction of words, diacritics can be employed to indicate them[1].These diacritics are dropped from almost all written text, except documents with pronunciation accuracy that cannot be compromised, such as religious texts or children's books. As a result, Arabic text corpora used for building machine learning models are typically undiacritized. However, studies have shown that including diacritics can lead to substantial improvements in various applications. For instance, Abbache et al. (2023) demonstrated that diacritized word embeddings significantly outperform non-diacritized versions. Similarly, diacritics enhance the quality of speech datasets used in tasks like Text-to-Speech (TTS) and Automatic Speech Recognition (ASR). By incorporating diacritics, these systems can generate more natural-sounding speech, improve pronunciation accuracy, and create more precise transcriptions (Aldarmaki and Ghannam, 2023).

Transcribed speech datasets used for building ASR models are typically undiacritized. As a result, building diacritized models requires a separate diacritization step, which can be performed on the transcripts using many popular text-based automatic diacritic restoration models like, ALISoft (URL, 2023), Farasa (QCRI, 2020), Mishkal (Zerrouki, 2020), and Camelira (Obeid et al., 2022). However, in a recent work, Shatnawi et al. (2024) discussed the degradation in performance of diacritic restoration models when applied to speech datasets, resulting in diacritic error rates ranging between 4-50%. They proposed a diacritic restoration framework that incorporates speech audio as another source of information. In particular, they utilize a pre-trained ASR model that is fine-tuned to predict diacritized transcripts as an additional source of input for diacritic restoration. This framework resulted in notable improvements in performance. However, the model relies on an ASR model as an extrinsic module, and therefore, its overall performance is constrained by the performance of the ASR model. In other words, the model proposed in Shatnawi et al. (2024) is not truly multi-modal. Another major limitation in this line of work is the scarcity of diacritized speech corpora that can be used for training both the ASR and diacritic restoration models.

[1] For instance, the undiacritized word البر can be diacritized as البَرّ meaning 'land', البِرّ meaning 'benevolence', and البُرّ meaning 'wheat'

Proceedings of The Second Arabic Natural Language Processing Conference, pages 160–169
August 16, 2024 ©2024 Association for Computational Linguistics

Given the potential advantages of integrating speech features into the model, coupled with the scarcity of such data, we explore a data augmentation strategy to boost the training data volume. In this approach, we generate randomly diacritized inputs and utilize a text-to-speech (TTS) model to synthesize speech. We hypothesize that diacritic restoration models overfit to word structure and other textual patterns, so the proposed approach aims to randomize the relationship between diacritics and the underlying text to avoid this kind of overfitting. The proposed data augmentation enables the model to learn from a wide range of diacritic variations regardless of the underlying word structure and textual context to help improve generalization by enforcing the recognition of acoustic over textual patterns. In addition, we propose a multi-modal variant of the model in Shatnawi et al. (2024), where speech is used directly as input rather than relying on an extrinsic ASR model.

Our experiments show that the proposed data augmentation method improves diacritic recognition performance across all test sets, validating the hypothesis that models typically overfit to the textual context. While the proposed multi-modal architecture improved over the text-only baselines, it still lags behind the ASR augmented model described in Shatnawi et al. (2024). Nevertheless, the model offers comparable performance while being more integrated and efficient, demonstrating the first implementation of a multi-modal diacritic restoration model.

2 Related Work

Speech-based Diacritic Restoration: The majority of diacritization models primarily operate on text-based approaches, utilizing either rule-based methodologies (Fashwan and Alansary, 2016; Alansary, 2018; Pasha et al., 2014) or statistical methods (Fadel et al., 2019; Hifny, 2012; Al-Thubaity et al., 2020). Integration of speech data into automatic diacritization processes has been minimally explored in prior research. The most recent relevant work is Shatnawi et al. (2024), in which they proposed a diacritization framework that incorporates both text and audio to diacritize speech data sets. Specifically, they utilized a pre-trained Automatic Speech Recognition (ASR) model, which was fine-tuned on diacritized Arabic speech data to produce diacritized transcripts of speech utterances. These transcripts were utilized as supplementary inputs for diacritic restoration models, consistently enhancing performance compared to text-only approaches. Their model's performance is notably influenced by the quality and nature of the training data used to fine-tune the ASR model, directly impacting diacritic restoration performance.

Data Augmentation (DA): The scarcity of labeled data for supervised learning often results in poor model's performance. To mitigate this issue, Data Augmentation (DA) methods are used to artificially expand the data by generating new examples from existing ones (Pellicer et al., 2023). The predominant DA techniques used in NLP include back-translation (Edunov et al., 2018; Yu et al., 2018; Xie et al., 2020), lexical substitution (e.g., synonym replacement, insertion, exchange, swap, and deletion) (Zhang et al., 2015; Wei and Zou, 2019), and paraphrasing (Iyyer et al., 2018; Kumar et al., 2020). Beyond unimodal tasks, Hao et al. (2023) recently introduced an innovative DA technique for generating semantically relevant image-text pairs in which they interpolate existing images and concatenate their corresponding text descriptions, enriching the training dataset and enhancing the performance of multimodal models.

Despite the wide application of data augmentation in many NLP tasks, its implementation in diacritization remains notably limited. To the best of our knowledge, there has been no previously applied approach for data augmentation to strengthen and diversify diacritization models.

Multi-Modal Performance: Multi-modal models offer a significant advantage by enriching feature representation both in depth and breadth, as evidenced in previous applications (Joshi et al., 2021; Albalawi et al., 2023). However, optimal performance is not guaranteed solely through this approach. The effectiveness of multi-modality depends on factors such as the nature of the data and the specific task being addressed. For instance, in a study by Albalawi et al. (2023), two models were introduced for rumor detection: a unimodal model relying solely on textual features and a multi-modal model incorporating both textual and fusion features. The performance of the multi-modal model failed to outperform that of the text-based model.

Original
وَالثَّانِي أَنَّهُ اللِّبَاسُ .وَالْعَيْشُ .وَالنِّعَمُ
وَهُوَ قَوْلُ ابْنِ عَبَّاسٍ رَضِيَ اللَّهُ عَنْهُمَا .وَالثَّالِثُ.
يُوَاسِ بِنَفْسِهِ مُوَاسَاةَ الْمُسَاعِفِ
Randomly Diacritized
وَالثَانِيٍّ أُنَهِ اللِبَاسِ. وَالعُيَشِ .وَالنِعْمٌّ.
وُهُوَ قِوَلٍ ابَنُ عَبَاسُ رُضَّيَ اللهُ عِنُهُمَا. وَالثَالْثٌ.
يِوَاسَ بِنَفَسُهِ مُوَّاسَّاةِ المُسَاعِفُ

Table 1: Randomly Generated Diacritized Examples.

Diacritic	Name	IPA	Word Position
سَ	Fatha	/a/	Any
سُ	Damma	/u/	Any
سِ	Kasrah	/i/	Any
سْ	Sukoon	∅	Any
سّ	Shaddah	:	Any
سٍ	Tanween Fath	/an/	End
سٌ	Tanween Damm	/un/	End
سً	Tanween Kasr	/in/	End

Table 2: Types of diacritics and their international phonetic alphabet representation (IPA). Table adapted from Mijlad and Younoussi (2019).

3 Data Augmentation for Speech-Based Diacritic Restoration

Building on previous research on data augmentation and multi-modal models in natural language processing, we propose a novel approach that extends these concepts to address the specific challenges of speech-based diacritization. To address the challenge of data scarcity and enhance generalization, we investigate a data augmentation technique to supplement training data by creating a synthetic speech dataset. The main objective of integrating a speech dataset in this augmentation process is to leverage pronunciation features, precisely mapping inputs to the diacritized form, rather than solely relying on textual context for disambiguation. In this context-agnostic approach, our emphasis is on accurately utilizing pronunciation to map letters to their diacritized forms.

We augment the datasets through two main steps: (1) randomly generating diacritics for each letter in the text, and (2) converting the diacritized text to audio using a Text-to-Soeech (TTS) model. We applied this data augmentation approach to our training set (discussed in Section 5.1), doubling the size of the dataset used for training the diacritic restoration model.

3.1 Random Generation of Diacritics

The process begins by adding random diacritics to each letter in the dataset, followed by the application of predefined rules to ensure the resulting text remains pronounceable. Developing these rules involves the application of heuristics to ensure that the resulting random diacritics can be pronounced by native Arabic speakers. Table 1 showcases some examples of the resulting randomly diacritized utterances.We extracted a sample from the dataset and conducted iterative manual analyses to identify patterns and characteristics of diacritized sentences that produce pronounceable words, thus informing the formulation of the heuristic rules. These rules fall into two main categories: replacement rules (Section 3.1.1) and deletion rules (Section 3.1.2))

3.1.1 Replacement Rules

Replacement rules involve substituting randomly generated, difficult-to-pronounce diacritics with new, randomly chosen ones, based on predefined rules derived from our manual analysis discussed earlier. Specifically, we replace the randomly generated diacritic if it meets one of the following conditions (See Table 2 for identifying Arabic diacritics):

- Sukoon or Shaddah if they appear at the first letter of the word (e.g., مْقعد);
- Tanween if it appears in any letter except the last letter in the word;
- One of the two Shaddahs appearing on two contiguous characters (e.g., مَقّعّد);
- One of the two Sukoons appearing on two contiguous characters (e.g., مَقْعْد);
- Any diacritic that is not Fatha or Damma appearing on Hamza on top (أ) at the beginning of a word (example of allowed variations, in this case, أَصبحَ or أُصبحَ);
- Any diacritic that is not Kasra appearing on Hamza below Alef (إ) such as the word إلى;

- Any diacritic that is not Fatha before the tied T (ة) (e.g, the Arabic word مَدرَسَة). Fatha is implicitly pronounced so we cannot replace it with any other diacritics;

- Any diacritic other than Fatha before the letter Alef of the following forms: (ى) or (ا);

- Stand-alone Shadda should be followed by another diacritic.

3.1.2 Deletion Rules

Deletion rules involve removing invalid diacritics altogether. Specifically, we delete the randomly generated diacritic if it meets one of the following conditions:

- All diacritics placed on the characters that are not in the Arabic alphabet;

- All diacritics applied to the following forms of Alef: Alef Madd (آ), Alef (ا), Maqsura (ى), and at the beginning of a word (Alef followed by the letter Lam) indicating the definiteness of a word (ال).

- Any additional diacritic for each letter, except if this additional diacritic accompanies Shaddah (i.e., each letter should have only one diacritic except in the case of Shaddah which can be followed by an additional diacritic).

Table 1 shows some examples of the resulting text, after applying the rules above. Note that while the words are no longer valid, they are all pronounceable.

3.2 Converting Diacritized Text to Audios

To train our model, we need both audio and their associated transcripts. In the first step described in the previous section, we obtained diacritized transcripts (randomly generated); now we need to generate the corresponding audio for each transcript. To that end, we employed the Google Cloud Text-to-Speech AI API[2]. We utilized a single voice, namely ar-XA-Wavenet-D, for this purpose. The generated audios are then integrated and utilized for training the diacritic restoration model.

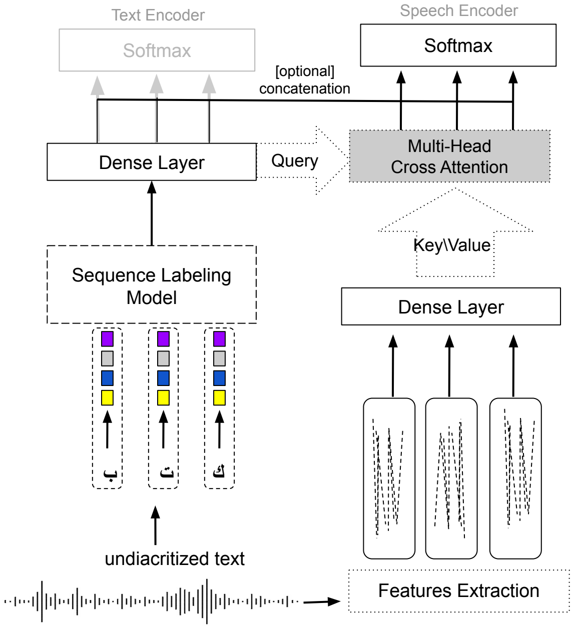

Figure 1: The proposed Multi-Modal diacritic restoration model takes a pair of undiacritized transcript (left side) and the corresponding speech features (right side), and a cross-attention mechanism to fuse the two modalities. The figure is modified from Shatnawi et al. (2024).

4 Multi-Modal Diacritic Restoration

The architecture for speech-based diacritic restoration proposed in Shatnawi et al. (2024) relies on an external ASR model (i.e. Whisper (Radford et al., 2023)) used to first process the audio and generate transcriptions. Although the model significantly reduced diacritic error rates, its reliance on an external system increases computational demands, in addition to the resources already required for processing ASR transcripts in the diacritic restoration model. To create a more efficient model, we propose to directly use the speech as an input to a mutli-modal architecture. We adopt the architecture proposed in Shatnawi et al. (2024) for the text side and the cross-attention mechanism, but we change the speech processing module. More concretely, the architecture (depicted in Figure 1) consists of a sequence labeling model to process the undiacritized text input and a speech feature extraction model. The two sides are then combined using a cross-attention mechanism, with the query vectors obtained from the text side and the key/value vectors from the speech side. The text sequence labeling model can be any sequence labeling architecture, such as Transformer or LSTM, as described in Shatnawi et al. (2024).

[2] cloud.google.com/text-to-speech?hl=en

For the speech feature extraction, we use a pre-trained self-supervised model to account for the small size of our training set. Our aim is to capitalize on the strengths of both modalities to enhance diacritic restoration accuracy and eliminate the dependency on an extrinsic ASR model for a more streamlined and efficient process. We used the self-supervised XLS-R model (Hsu et al., 2021), a large-scale multi-lingual pre-trained model for speech feature extraction. Trained on a vast dataset comprising 436K hours of unlabeled speech, XLS-R adopts the wav2vec 2.0 objective (Schneider et al., 2019) across 128 languages. This objective involves learning contextual representations of speech by predicting masked portions of the audio waveform. The model is typically fine-tuned for downstream tasks such as speech translation, ASR, or speech classification. We first fine-tuned the model for ASR using our speech training set, which is done by adding a linear layer with CTC loss. The fine-tuning ensures that the model's inner representations are already geared toward the representation of Arabic sounds.

5 Experimental Settings

5.1 Datasets

We replicate the experimental settings used in Shatnawi et al. (2024), including the data sets used for training and testing[3]. In particular, we use the Classical Arabic Text-To-Speech Corpus **ClArTTS** (Kulkarni et al., 2023). The ClArTTS dataset comprises audio recordings of classical Arabic speech accompanied by manually diacritized and verified transcripts. It consists of approximately 12 hours of recorded speech from a single male speaker, totaling around 10,000 relatively short utterances. Additionally, there is a separate subset of approximately 30 minutes reserved for testing purposes. For experiments involving text-only models, we utilize the cleaned Tashkeela Corpus, which encompasses a vast collection of Arabic texts predominantly from Classical Arabic literature and religious texts, along with a smaller portion in the Modern Standard Arabic variety. This corpus contains 2.3 million words distributed across 55,000 lines and is derived from the original Tashkeela corpus introduced by (Zerrouki and Balla, 2017). For Testing, we also use a small manual diacritic dataset, proposed for TTS purposes and derived from broadcast news (Baali et al., 2023). This dataset (dubbed QASR TTS)[4] contains one hour of speech by a male speaker and one hour of speech by a female speaker.

5.2 Model Setup

We used the Adam optimizer (Kingma and Ba, 2014) for learning and used 0.2 for the dropout rate[5]. The text-side of the multi-modal model is first pre-trained on the Tashkeela corpus. The complete multi-modal model is trained on the ClArTTS corpus only. For the XLS-R side, we used the version trained on 53 languages, which includes Arabic[6]. We fine-tuned it first using ASR with CTC loss on the ClArTTS corpus. Training is configured with hyper-parameters, including a learning rate of 3e-4 and 30 epochs, leveraging mixed precision (FP16) for faster training. After fine-tuning XLS-R, we integrated the model into the multimodal architecture up to layer 22, freezing the XLS-R parameters while training the multimodal diacritic restoration model[7]. Comparison of model parameters are shown in the following Table. Including the ASR or speech feature extractor in the calculation, the multi-modal model has around 40% of the parameters of the Text+ASR model. In both cases, the largest contributor to model size is the speech module.

Component	Text+ASR	Multi-Modal
Text-side encoder	∼700K†	∼700K†
ASR / Feature extractor	∼763M	∼315M
Speech-side encoder	∼700K†	0
Cross-attention	263808	1181312
Total (- speech module)	∼764M	∼316M

Table 3: Number of parameters for the proposed multi-modal vs. Text+ASR model from Shatnawi et al. (2024). † This estimate is based on a 2-layer BiLSTM sequence encoder; the number can vary depending on model configuration.

6 Results & Discussion

In this section, we first examine the performance of the proposed multi-modal diacritic restoration model, and then show the effect of our proposed data augmentation technique. As a baseline, we use

[3] The experiments are available at https://github.com/SaraShatnawi/Diacritization.git

[4] https://arabicspeech.org/qasr_tts

[5] We investigated different optimizers (Adam, SDG, RMSprop, and Adagrad) and different dropout values (0.1, 0.2, 0.3, and 0.001), and selected the best among them.

[6] https://huggingface.co/facebook/wav2vec2-large-xlsr-53

[7] Preliminary experiments indicated that later layers perform better for this task than early or intermediate layers.

		Including 'no diacritic'		**Excluding 'no diacritic'**	
Model	**Architecture**	**w. case ending**	**w.o case ending**	**w. case ending**	**w.o case ending**
Test Set: CLArTTS					
Text-only	Transformer	9.63	7.38	11.61	8.91
Text+ASR	Transformer	**3.63**	**2.71**	**4.07**	**3.17**
Multimodal	Transformer	8.10	6.23	9.30	7.27
Text-only	LSTM	4.93	3.55	5.86	4.30
Text+ASR	LSTM	**2.70**	**1.83**	**2.85**	**1.99**
Multimodal	LSTM	4.41	3.13	4.40	3.60
Test Set: Male Speaker from QASR TTS					
Text-only	Transformer	29.28	23.84	33.68	25.59
Text+ASR	Transformer	**21.69**	**14.44**	**24.55**	**14.16**
Multimodal	Transformer	30.11	25.65	33.59	28.21
Text-only	LSTM	20.67	12.63	23.49	12.04
Text+ASR	LSTM	**19.82**	**12.34**	**21.98**	**11.24**
Multimodal	LSTM	22.63	15.28	29.31	18.23
Test Set: Female Speaker from QASR TTS					
Text-only	Transformer	39.19	35.35	37.27	26.19
Text+ASR	Transformer	**35.69**	**30.72**	**31.84**	**18.85**
Multimodal	Transformer	38.28	34.42	37.17	25.69
Text-only	LSTM	35.33	29.53	31.45	17.05
Text+ASR	LSTM	**34.06**	**29.11**	**29.12**	**15.82**
Multimodal	LSTM	35.56	29.05	31.17	16.79

Table 4: Diacritic Error Rate (DER) % for Multimodal framework using fine-tune XLS-R on Transformer and LSTM Architecture, evaluated on CLArTTS, Male Speaker from QASR TTS, and Female Speaker from QASR TTS datasets. Note: All the models in this table are trained on the CLArTTS dataset and Tashkella.

the text-only and text+ASR models from Shatnawi et al. (2024) as it achieves the best performance reported on the test data (Refer to Shatnawi et al. (2024) for comparison with other diacritic restoration models).

6.1 Multi-Modal Performance

Table 4 shows the performance of the proposed multi-modal diacritic restoration model compared to the baselines. Using speech features generally improves the performance of Text-Only diacritic restoration, resulting in reduced diacritic error rates across. However, the multi-modal model lags behind the Text+ASR model, and sometimes even behind the Text-Only model. There are a couple of differences between the two models that could explain this difference, but one thing to highlight here is that the multi-modal model is more streamlined and efficient as it works without reliance on an external ASR model with the full pipeline of ASR inference. Yet, one of the potential drawbacks that led to sub-optimal performance is that XLS-R is a self-supervised model and has only been fine-tuned on the ClArTTS corpus; on the other hand, the Text+ASR uses the Whisper model, which is already optimized for ASR even before fine-tuning. In Table 5, we show ASR performance on the test set using our fine-tuned XLS-R vs. the Whisper model used in Shatnawi et al. (2024). Whisper reduces error rates almost by half, which plays a large role in the performance of Text+ASR model. This suggests a possible future development of the multi-modal model, where XLS-R can be optimized first on more general (un-diacritized) Arabic speech data and then fine-tuned on ClArTTS, to achieve performance on a par with Whisper, before integrating it with the multi-modal diacritic restoration model.

Dataset	w.o Diacritics		w. Diacritics	
	CER	WER	CER	WER
XLS-R model fine-tuned for ASR				
ClArTTS Test	4.43	28.79	4.40	34.0
QASR Female	25.59	73.0	37.52	96.12
QASR Male	23.32	67.52	29.6	84.36
Whisper model fine-tuned for ASR				
ClArTTS Test	2.20	8.02	2.90	14.43
QASR Female	11.6	36.9	27.5	87.3
QASR Male	11.1	36.4	21.06	72.4

Table 5: Character Error Rate (CER)% and Word Error Rate (WER) % using the Whisper and XLS-R model fine-tuned on ClArTTS training set

		Including ‘no diacritic’		Excluding ‘no diacritic’	
Model	**Frozen**	**w. case ending**	**w.o case ending**	**w. case ending**	**w.o case ending**
Test Set: CLArTTS					
Transformer	✓	8.29	6.37	9.22	7.14
Transformer	×	17.73	19.65	18.66	18.88
LSTM	✓	4.53	3.18	4.96	3.57
LSTM	×	8.45	7.00	10.06	8.38
Test Set: Male Speaker from QASR TTS					
Transformer	✓	31.39	24.08	36.46	26.20
Transformer	×	32.10	25.58	38.16	28.76
LSTM	✓	23.72	15.24	24.87	15.06
LSTM	×	24.78	18.03	26.90	16.39
Test Set: Female Speaker from QASR TTS					
Transformer	✓	41.99	37.83	43.28	32.12
Transformer	×	45.02	40.86	46.31	35.15
LSTM	✓	36.05	30.53	36.47	21.29
LSTM	×	38.89	33.64	37.79	23.31

Table 6: Diacritic Error Rate (DER) % for Multimodal framework using fine-tune XLS-R fine-tuned with text-encoder, evaluated on CLArTTS, Male Speaker from QASR TTS, and Female Speaker from QASR TTS datasets. Note: All the models in this table are trained on the CLArTTS dataset and Tashkella.

6.2 Effect of XLS-R Parameter Updates

The XLS-R model used in our experiments above is first fine-tuned for the task of ASR on ClArTTS. Once trained for ASR, we integrated the model in the multi-modal framework. While training the multi-modal diacritic restoration model, we experimented with either freezing the XLS-R parameters or continuing to update them. We found that freezing the parameters at this stage results in far better performance. The results with/without freezing the parameters are shown in Table 6. While this result seems rather counter-intuitive, a possible explanation is the small size of our training set compared to the complexity of the multi-modal model, leading to catastrophic forgetting of the useful features learned in the ASR fine-tuning stage.

6.3 Experiments with Data Augmentation

In this section, we describe experiments on the effect of data augmentation in speech-based diacritic restoration. In particular, we test the usefulness of the data augmentation method described in Section 3, where random diacritics are used for synthesizing speech. Our hypothesis is that this scheme can lead to better generalization from the speech features due to the reduction in the predictability of the diacritics from the textual context.

6.3.1 Experimental Settings

To create the synthetic audio from random diacritics, we used the ClArTTS text and applied the randomization rules. We use used Google TTS to synthesize the speech from these inputs[8]. This gave as twice the original train set size in total. We used this augmented data for training our Multi-Modal model, but we did not modify the underlying and frozen XLS-R model. For Text+ASR, we re-trained both the underlying ASR model (i.e. Whisper) and the Text+ASR diacritic restoration model.

6.3.2 Results

As seen in Table 7, this data augmentation method improved the diacritic restoration performance across the board. The benefits of data augmentation are particularly evident for models that start with higher DER, with absolute reductions ranging from around 1% to 5% are observed for all models. This underscores the importance of exposing the diacritic restoration model to a wider variety of diacritic variations to enable generalizations beyond what can be inferred from text alone. Furthermore, the improvements are more evident for the Multi-Modal model compared with Text+ASR, but the gap between them remains, where the ASR based model continues to outperform the multi-modal model. Importantly, though, without the data augmentation, the Multi-Modal model did not consistently improve over the Text-Only model; with the addition of roughly 10K utterances using the data augmentation method, the results are consistently better than the Text-Only model across all test sets.

[8] We tried several open-source and commericial TTS systems, but most of them do not produce speech that is faithful to the provided random diacritics. The best one at producing speech as specified was Google WaveNet model.

			Including ‘no diacritic’		Excluding ‘no diacritic’	
Model	**Architecture**	**Data Augmentation**	**w. case ending**	**w.o case ending**	**w. case ending**	**w.o case ending**
Multi-modal	**Test Set: CLArTTS**					
	Transformer	×	8.10	6.23	9.30	7.27
	Transformer	✓	5.09	4.13	6.48	6.82
	LSTM	×	4.41	3.13	4.40	3.60
	LSTM	✓	3.78	2.54	4.33	2.68
	Test Set: Male Speaker from QASR TTS					
	Transformer	×	30.11	25.65	33.59	28.21
	Transformer	✓	25.21	17.69	28.34	20.18
	LSTM	×	22.63	15.28	29.31	18.23
	LSTM	✓	20.15	13.64	22.13	18.26
	Test Set: Female Speaker from QASR TTS					
	Transformer	×	38.28	34.42	37.17	25.69
	Transformer	✓	36.31	32.07	37.11	31.46
	LSTM	×	35.56	29.05	31.17	16.79
	LSTM	✓	34.52	29.01	35.12	20.23
Text + ASR	**Test Set: CLArTTS**					
	Transformer	×	3.63	2.71	4.07	3.17
	Transformer	✓	3.19	2.64	3.89	2.73
	LSTM	×	2.70	1.83	2.85	1.99
	LSTM	✓	2.18	1.07	2.18	1.90
	Test Set: Male Speaker from QASR TTS					
	Transformer	×	21.69	14.44	24.55	14.16
	Transformer	✓	20.13	13.03	24.33	15.08
	LSTM	×	19.82	12.34	21.98	11.24
	LSTM	✓	17.11	9.63	20.27	10.37
	Test Set: Female Speaker from QASR TTS					
	Transformer	×	35.69	30.72	31.84	18.85
	Transformer	✓	33.23	29.18	34.13	32.95
	LSTM	×	34.06	29.11	29.12	15.82
	LSTM	✓	31.78	26.98	27.91	13.76

Table 7: Diacritic Error Rate (DER) % for Multimodal (Fine-tuned XLS-R) and Text+ASR model evaluated on CLArTTS, Male Speaker from QASR TTS, and Female Speaker from QASR TTS datasets. We compare the performance of the models with and without the proposed data augmentation method, showing improvements with data augmentation across the board.

7 Conclusions

In this paper, we introduced a novel approach for data augmentation that has consistently improved speech-based diacritic restoration models, resulting in enhanced performance on speech datasets compared to models that rely solely on text, and compared to models that do not use data augmentation. Additionally, we explored the potential of a multi-modal approach that does not rely on an extrinsic ASR system as previously proposed for this task. While the multi-modal approach provides a more efficient and streamlined process, and improves over text-only model when data augmentation is used, it did not reach the performance provided by the ASR based model. This work highlights a couple of factors that influence the performance of such models. First, diacritic restoration models are generally limited by the type of data used for training them, and most existing open-source data sets for diacritic restoration are based on Classical Arabic, leading to poor performance in Modern Standard Arabic. Applying data augmentation techniques, such as one proposed in this paper, can lead to better generalization. Second, adding speech features where speech data is available results in better performance than text-only models, as the speech contains the low level acoustic features that distinguish these diacritics. Finally, while speech-based diacritic restoration works, the performance highly depends on the underlying speech model used to extract speech features. Our experiments show that optimizing the performance of the models used for feature extraction is essential, and future work could enhance the performance of the multi-modal architecture by integrating features from models that are first fine-tuned and optimized for ASR.

Limitations

One of the limitations of the experiments described here is the large gap in ASR performance between the two models underlying the Multi-Modal and Text+ASR models. This gap in performance is due to the fact that Whisper is already pre-trained for ASR in Arabic, and fine-tuning it on the small ClArTTS results in significantly better ASR performance compared to fine-tuning the smaller XLS-R model from scratch on the same set. Unfortunately, Whisper could not be easily integrated in our multi-modal model directly for a fair comparison. A more thorough evaluation should start by optimizing the underlying models to nearly similar performance. One more difference between the ASR+Text and multi-modal model with data augmentation is that we only used data augmentation for updating the multi-modal model, but the underlying XLS-R was frozen with the weights from ClArTTS only. While the results still showed larger reductions in error rates for the multi-modal model, a better comparison would be made by testing the effect of data augmentation on the underlying speech model. Another limitation is that the synthetic speech used for data augmentation is all based on the same underlying text. Our rationale was to add redundancy in the underlying text but differences in diacritics to minimize the reliance on the text context. However, using more varied underlying texts from MSA could have been beneficial, but we did not experiment with such variations.

References

Mohamed Abbache, Ahmed Abbache, Jingwen Xu, Farid Meziane, and Xianbin Wen. 2023. The impact of arabic diacritization on word embeddings. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(6):1–30.

Abdulmohsen Al-Thubaity, Atheer Alkhalifa, Abdulrahman Almuhareb, and Waleed Alsanie. 2020. Arabic diacritization using bidirectional long short-term memory neural networks with conditional random fields. *IEEE Access*, 8:154984–154996.

Daud Lintang al Yamin. 2023. Bahasa arab sebagai identitas budaya islam dan pemersatu keberagaman suku. *Ta'limi| Journal of Arabic Education and Arabic Studies*, 2(1):73–86.

Sameh Alansary. 2018. Alserag: an automatic diacritization system for arabic. *Intelligent Natural Language Processing: Trends and Applications*, pages 523–543.

Rasha M Albalawi, Amani T Jamal, Alaa O Khadidos, and Areej M Alhothali. 2023. Multimodal arabic rumors detection. *IEEE Access*, pages 1–1.

Hanan Aldarmaki and Ahmad Ghannam. 2023. Diacritic recognition performance in arabic asr. *arXiv preprint arXiv:2302.14022*.

Massa Baali, Tomoki Hayashi, Hamdy Mubarak, Soumi Maiti, Shinji Watanabe, Wassim El-Hajj, and Ahmed Ali. 2023. Unsupervised data selection for tts: Using arabic broadcast news as a case study. *arXiv preprint arXiv:2301.09099*.

Muhammad Hasan Bakalla. 2023. *Arabic culture: through its language and literature*. Taylor & Francis.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Ali Fadel, Ibraheem Tuffaha, Bara' Al-Jawarneh, and Mahmoud Al-Ayyoub. 2019. Neural Arabic text diacritization: State of the art results and a novel approach for machine translation. In *Proceedings of the 6th Workshop on Asian Translation*, pages 215–225, Hong Kong, China. Association for Computational Linguistics.

Amany Fashwan and Sameh Alansary. 2016. A rule based method for adding case ending diacritics for modern standard arabic texts. In *16th International Conference on Language Engineering. The Egyptian Society of Language Engineering (ESOLE)*.

Xiaoshuai Hao, Yi Zhu, Srikar Appalaraju, Aston Zhang, Wanqian Zhang, Bo Li, and Mu Li. 2023. Mixgen: A new multi-modal data augmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 379–389.

Yasser Hifny. 2012. Smoothing techniques for arabic diacritics restoration. In *Proceedings of the 12th Conference Lang. Eng.(ESOLEC'12)*, 1, pages 6–12. Citeseer.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*.

Gargi Joshi, Rahee Walambe, and Ketan Kotecha. 2021. A review on explainability in multimodal deep neural nets. *IEEE Access*, 9:59800–59821.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ajinkya Kulkarni, Atharva Kulkarni, Sara Abedalmonem Mohammad Shatnawi, and Hanan Aldarmaki. 2023. Clartts: An open-source classical arabic text-to-speech corpus. *arXiv preprint arXiv:2303.00069*.

Ashutosh Kumar, Kabir Ahuja, Raghuram Vadapalli, and Partha Talukdar. 2020. Syntax-guided controlled generation of paraphrases. *Transactions of the Association for Computational Linguistics*, 8:330–345.

Ali Mijlad and Yacine El Younoussi. 2019. Arabic text diacritization: Overview and solution. In *Proceedings of the 4th International Conference on Smart City Applications*, SCA '19, New York, NY, USA. Association for Computing Machinery.

Ossama Obeid, Go Inoue, and Nizar Habash. 2022. Camelira: An arabic multi-dialect morphological disambiguator. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 319–326.

Arfath Pasha, Mohamed Al-Badrashiny, Mona T Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *Lrec*, volume 14, pages 1094–1101.

Lucas Francisco Amaral Orosco Pellicer, Taynan Maier Ferreira, and Anna Helena Reali Costa. 2023. Data augmentation techniques in natural language processing. *Applied Soft Computing*, 132:109803.

QCRI. 2020. Farasa api diacritization module. Accessed on October 12, 2022.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*.

Sara Shatnawi, Sawsan Alqahtani, and Hanan Aldarmaki. 2024. Automatic restoration of diacritics for speech data sets. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. Association for Computational Linguistics".

URL. 2023. Ali-soft. Accessed on October 12, 2023.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

Taha Zerrouki. 2020. Towards an open platform for arabic language processing.

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

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.