Arabic-Adapted One-Step Speech-to-Diacritized ASR: Evaluation and Error Analysis

Osamah Abduljalil[1], Dalal Ali[1], Razan Bajaman[1], Abdullah Alharbi[1,2]

[1]College of Computer and Information Sciences,
Imam Mohammad Ibn Saud Islamic University, Riyadh 11432, Saudi Arabia
[2]King Salman Global Academy for Arabic Language (KSAA), Riyadh, Saudi Arabia
{446016735, 446012611, 440022262}@sm.imamu.edu.sa
aialharbii@imamu.edu.sa, aialharbi@ksaa.gov.sa

Abstract

Arabic diacritics encode phonetic information essential for pronunciation, disambiguation, and downstream applications, yet most Arabic ASR systems generate undiacritized output. In this work, we study direct speech-to-diacritized-text recognition using a single-stage ASR pipeline that predicts diacritics jointly with Arabic letters, without text-based post-processing. We evaluate two Arabic-adapted ASR architectures—wav2vec 2.0 XLSR-53 and Whisper-base—under a unified experimental setup on the ClArTTS Classical Arabic dataset. Performance is assessed using surface and lexical WER/CER alongside diacritic error rate (DER) to disentangle base transcription accuracy from diacritic realization. Our results show that Arabic-adapted wav2vec 2.0 achieves substantially lower diacritic error rates than Whisper, indicating stronger exploitation of acoustic cues relevant to vowelization. We further analyze the effect of decoding strategy and provide a detailed breakdown of diacritic errors, highlighting challenges associated with short vowels and morphosyntactic markers. These findings underscore the importance of model architecture and Arabic-specific adaptation for accurate diacritized Arabic ASR.

1 Introduction

Arabic diacritics encode essential phonetic and linguistic information, including short vowels, consonant gemination, and case endings. These marks play a critical role in pronunciation, lexical disambiguation, and downstream applications such as text-to-speech synthesis, language learning, and linguistic analysis (Habash, 2010). Despite their importance, most Arabic text and speech corpora omit diacritics, leaving them implicit and forcing computational systems to infer them from context (Abed et al., 2019). Consequently, modern Arabic Automatic Speech Recognition (ASR) systems typically produce undiacritized output, limiting their suitability for applications that require fully vocalized text (Aldarmaki and Ghannam, 2023).

A common approach for obtaining diacritized Arabic text from speech is to apply text-based diacritization as a post-processing step to undiacritized ASR output (Petrică et al., 2014; Shatnawi et al., 2024). While convenient, this two-stage pipeline treats diacritization as a purely textual inference problem. Once speech is transcribed without diacritics, acoustic cues associated with vowel realization and consonant length are irreversibly discarded. Text-based diacritizers must then rely on lexical statistics and syntactic context, making them particularly sensitive to transcription errors, ambiguous word forms, and domain mismatch between spoken and written Arabic.

An alternative paradigm integrates diacritization directly into the ASR process by training models to generate diacritized text in a single step from speech. In this setting, diacritics are predicted alongside alphabetic characters, allowing the model to exploit phonetic evidence present in the acoustic signal. Recent work has shown that such direct ASR-based diacritization can outperform text-based post-processing in terms of diacritic accuracy and robustness, suggesting that diacritics are more naturally modeled as a speech-grounded phenomenon rather than reconstructed from text alone (Aldarmaki and Ghannam, 2023; Shatnawi et al., 2024).

While prior studies have compared ASR-based and text-based diacritization pipelines, an important question remains underexplored: *what type of ASR model is best suited for direct Arabic diacritization from speech?* Modern ASR systems are typically pretrained either on large multilingual corpora or on language-specific data. Multilingual pretraining offers scale and diversity, whereas language-specialized pretraining may better capture Arabic-specific phonology, morphology, and orthographic conventions. For Arabic, whose dia-

Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script (AbjadNLP 2026), pages 371–379

critic system closely reflects fine-grained phonetic distinctions, this trade-off is especially consequential.

In this work, we investigate whether Arabic-specialized ASR models or general multilingual ASR models are more effective for producing fully diacritized Arabic text directly from speech. We focus exclusively on a single-stage ASR pipeline that outputs diacritics as part of the transcription, thereby isolating the effect of model pretraining from confounding factors introduced by text-based post-processing. Using representative architectures from the wav2vec 2.0 and Whisper families, we compare multilingual pretrained models with Arabic-adapted counterparts under identical training and evaluation conditions. In addition, we analyze the impact of decoding strategy by comparing greedy decoding with beam search, examining whether delayed commitment during inference improves diacritic realization.

Unlike prior work that relies on multilingual or hybrid ASR+diacritization pipelines, we adopt a controlled single-stage setup that isolates acoustic modeling effects, enabling clearer attribution of gains to architecture and pretraining. Our contributions are threefold:

- A controlled comparison of Arabic-adapted and multilingual ASR models for direct diacritized transcription.
- An evaluation framework that separates diacritic accuracy from base transcription errors.
- An analysis of decoding strategies and their impact on diacritic prediction.

These contributions inform the design of ASR systems requiring accurate and phonetically faithful diacritized Arabic output.

2 Related Work

Previous studies on diacritized Arabic ASR have examined the impact of diacritics on recognition performance and proposed approaches that predict diacritics directly within ASR systems. (Abed et al., 2019) compared diacritized and nondiacritized Arabic ASR models across multiple corpus sizes and showed that incorporating diacritics increased WER by 0.59%–3.29% using GMM- and DNN-based architectures. Additionally, (Alsayadi et al., 2021) explored end-to-end deep learning approaches for diacritised Arabic ASR using CNN-LSTM, CTC, and attention-based models, where CNN-LSTM with attention outperformed conventional and joint CTC-attention systems in terms of WER and CER.

Aldarmaki and Ghannam (2023) studied direct Arabic ASR diacritization by fine-tuning pretrained ASR models, namely Whisper-medium and XLS-R, and evaluated diacritic recognition beyond WER/CER using diacritics *coverage* and *precision*. More recently, (Alrumiah and Al-Shargabi, 2023) proposed DNN-based models for diacritized Arabic ASR, converting Quranic and Classical Arabic speech directly into fully diacritized text, with RNN-CTC achieving the best performance.

Furthermore, (Alaqel and El Hindi, 2025b) proposed a lightweight encoder-only Transformer with Relative Positional Encoding and CTC training for direct diacritized Arabic ASR, achieving competitive performance while significantly reducing model size compared to large-scale models such as XLSR. Similarly, (Alaqel and El Hindi, 2025a) demonstrated the effectiveness of transfer learning for diacritized Arabic ASR by fine-tuning a multilingual XLSR model on diacritized Arabic data.

Complementing these model-focused studies, recent open-source effort have emphasized broader coverage across Arabic varieties. (Grigoryan et al., 2025) introduced open-source FastConformer-based ASR models, including a unified system for Modern Standard Arabic and Classical Arabic with support for diacritized output.

3 Methodology

This section describes the data, preprocessing pipeline, and speech recognition models used in our study. We focus on a single-stage ASR approach that directly generates fully diacritized Arabic text from speech, without any text-based post-processing.

3.1 Dataset

We employ the ClArTTS (Classical Arabic Text-to-Speech) dataset (Kulkarni et al., 2023), which provides paired Classical Arabic speech and fully diacritized text. It comprises roughly 12 hours of recordings from a single male speaker, with audio sourced from a LibriVox audiobook and subsequently segmented, transcribed, and annotated by hand. All audio is distributed as waveform files sampled at 44.1 kHz, and the accompanying transcripts are entirely diacritized at the character level. All audio samples were resampled to 16 kHz to

match the input requirements of the pretrained ASR models. While this corpus was initially created for text-to-speech research, in this study we use it in the opposite direction, for speech-to-diacritized-text recognition. The dataset creators supply a predefined partition. The dataset is divided into approximately 88% training, 10% validation, and 2% test sets, and these splits are used unchanged across all experiments. While ClArTTS was originally developed for text-to-speech research and consists of read speech from a single speaker, it is well suited to the goals of this study. The corpus provides fully diacritized transcripts with consistent orthographic and phonetic alignment, enabling controlled analysis of diacritic realization directly from acoustic evidence. This setting allows us to isolate the effects of model pretraining and decoding strategy on diacritization behavior without confounding factors introduced by speaker variability, background noise, or inconsistent annotation.

3.2 Speech Recognition Models

We employ end-to-end automatic speech recognition models to directly map Arabic speech to fully diacritized text in a single stage. Diacritics are treated as first-class output symbols and are predicted jointly with Arabic letters at the character level. No text-based diacritization or post-processing is applied.

We consider two representative ASR architectures that differ in both modeling objective and decoding behavior. The first is based on the wav2vec 2.0 model (Kulkarni et al., 2023), which uses a convolutional feature encoder followed by a transformer encoder and is trained with a Connectionist Temporal Classification (CTC) objective. The second follows the Whisper architecture, an encoder–decoder transformer trained in a sequence-to-sequence manner with autoregressive decoding.

For the wav2vec 2.0-based system, we use an Arabic-adapted XLSR-53 checkpoint (Grosman, 2021) that was pretrained on large-scale multilingual speech data and subsequently fine-tuned on Arabic speech corpora. This model is designed to capture Arabic-specific acoustic and phonetic characteristics while retaining the benefits of multilingual pretraining. The model is further fine-tuned in our experiments to predict fully diacritized Arabic text using a CTC objective.

For the sequence-to-sequence setting, we use an Arabic-adapted Whisper-base model [1] derived from the multilingual Whisper architecture. This checkpoint was fine-tuned on multiple Arabic speech datasets to improve alignment with Arabic acoustic patterns. In our experiments, the model is trained to generate fully diacritized Arabic text directly from speech inputs, treating diacritics as part of the output vocabulary.

We use the Whisper-base variant rather than Whisper-medium (used in prior work by Aldarmaki and Ghannam (2023)) due to computational constraints and the availability of suitable pretrained checkpoints. At the time of this study, we could not identify a publicly available Arabic-adapted Whisper-medium model that aligns with our research objectives of direct diacritized transcription from speech. The selected Whisper-base checkpoint provides the necessary Arabic-specific adaptation while remaining computationally feasible for our experimental setup. Despite the difference in model size, this choice enables meaningful comparison with the Arabic-adapted wav2vec 2.0 model and allows us to isolate the effects of architecture and decoding strategy on diacritization performance.

By evaluating these two architectures under a unified single-stage ASR formulation, we aim to analyze how model pretraining scope, training objective, and decoding strategy influence Arabic diacritic realization from speech.

4 Experimental Setup

This section describes how the experiments were conducted. We first establish baseline performance using pretrained models without task-specific adaptation, and then detail the fine-tuning procedure used to adapt each model for direct Arabic diacritization from speech. The purpose of this section is to make the experimental methodology explicit and reproducible.

4.1 Reference Baselines

To contextualize our results, we compare our findings with the multilingual ASR baselines reported by Aldarmaki and Ghannam (Aldarmaki and Ghannam, 2023), a comprehensive study on Arabic diacritic recognition using direct ASR-based approaches. Their work evaluates multilingual pretrained models, including Whisper and

[1] https://huggingface.co/YazanSalameh/Whisper-base-Arabic

wav2vec 2.0 XLS-R, within a single-stage ASR pipeline that directly generates diacritized Arabic text from speech, without text-based post-processing.

Our study adopts the same pipeline design, enabling a direct comparison with these multilingual baselines. While prior work demonstrates the effectiveness of direct ASR-based diacritization using general-purpose multilingual models, our comparison focuses on assessing whether Arabic-specific ASR adaptation yields additional gains in diacritic recognition accuracy. We intentionally do not include a two-stage ASR followed by text-based diacritization baseline. Our objective is to isolate how much diacritic information can be recovered directly from acoustic evidence alone. Introducing a text-based component would confound this analysis by incorporating linguistic priors unrelated to the speech signal.

4.2 Fine-Tuning Procedure

All experiments follow a unified single-stage ASR setup in which models are trained to generate fully diacritized Arabic text directly from speech. Fine-tuning is performed on the ClArTTS training split, with model selection based on performance on the validation split. The test split is used exclusively for final evaluation.

Both models are fine-tuned using the same training–validation–test partitions and the same text and audio preprocessing pipeline described in Section3. Training is intentionally limited to a small number of epochs due to the homogeneity and high quality of the dataset, reducing the risk of overfitting while enabling controlled comparison across architectures. After fine-tuning, we analyze the same speech sample used in the baseline evaluation to qualitatively assess the impact of model adaptation on diacritized transcription.

The table illustrates1 the effect of fine-tuning both Wav2Vec2 and Whisper for direct diacritized Arabic speech recognition. Prior to fine-tuning, Whisper generates output that is mostly without diacritics, indicating a limited capacity to restore short vowel markings, while Wav2Vec2 already retains a larger share of vowel diacritics, though with some substitution mistakes. After fine-tuning, both models show better diacritic consistency and fewer errors; nonetheless, Wav2Vec2 stays closer to the ground-truth transcription, especially in recovering short vowels and word-final diacritics. This case illustrates that fine-tuning improves diacritic modeling in both architectures, with Wav2Vec2 displaying heightened sensitivity to fine-grained acoustic cues that are crucial for Arabic diacritization.

The reference		"صَرَفْتُ النَّاسَ عَنْ بَالِي فَحَبْلُ وِدَادِهِمْ بَالِي وَحَبْلُ اللهِ مُعْتَصمي بِهِ عَلَّقْتُ آمَالِي"
Before fine-tuning	wav2vec	"صَرَفْتُ النَّاسَ عَنْ بَالِي فَحَبْلُ رِدَادِهِمْ بالِ وَحَبْلُ اللهِ مُعْتَصَمِ بِهِ عَلَّقْتُ آمَالِ"
	Whisper	"صرفت الناس عن بال فحبلوا إدادهم بال وحبل الله معتصم به علقت آمال "
After fine-tuning	wav2vec	"صَرَفْتُ النَّاسَ عَنْبَالِي فَحَبْلُ دَادِهِمْ بَالِي وَحَبْلُ اللهِ مُعْتَصَمِ بِهِ عَلَّقْتُ آمَالِ"
	Whisper	"صَرَفْتُ النَّاسَ عَنْ بَالْلْ فَحَبْلُ وِدَادِهِمْ بَالِي وَحَبْلُ اللهِ مُعْطَصَمِ بِهِ عَلَّقْتُ آمَالِ"

Figure 1: Comparison of diacritized outputs produced by Wav2Vec2 and Whisper before and after fine-tuning.

Training Configuration. Fine-tuning is performed using the AdamW optimizer with an initial learning rate of 1e−5. Models are trained with a batch size of 2 for 3 epochs. Gradient accumulation is set to 1.

For the wav2vec 2.0-based model, fine-tuning uses a CTC objective over the character-level vocabulary that includes Arabic letters, diacritics, the tatweel character, and spaces. For the Whisper-based model, training follows an autoregressive sequence-to-sequence objective, where the decoder generates the diacritized output sequence conditioned on the encoded speech representation.

Checkpoint Selection. Model checkpoints are selected based on the lowest validation loss after 3 epochs for wav2vec 2.0 and 3 epochs for Whisper. The selected checkpoints are then used for all evaluations reported in Section5.

Decoding Configuration. At inference time, all models are evaluated without any external language model or text-based post-processing to ensure that diacritic predictions rely solely on acoustic modeling and decoding behavior. For the Whisper-based system, we evaluate both greedy decoding and beam search decoding with a beam width of 5. For the wav2vec 2.0-based system, greedy decoding is used.

This decoding configuration reflects architectural differences between the two models. Whisper follows an encoder–decoder architecture with autoregressive sequence generation, for which beam search is a standard inference strategy that maintains multiple candidate hypotheses and enables sequence-level optimization. In contrast, the

wav2vec 2.0-based model is trained with a Connectionist Temporal Classification (CTC) objective, which assumes conditional independence between output labels given the acoustic input. While beam search can be applied to CTC models, it is typically most effective when combined with an external language model. To avoid introducing additional linguistic constraints and to maintain a controlled comparison focused on acoustic modeling, we therefore use greedy decoding for the wav2vec 2.0-based system.

4.3 Evaluation Metrics

Model performance is evaluated on the ClArTTS test split using three complementary error measures that disentangle base transcription accuracy from diacritic realization. All metrics are computed at both the word and character levels where applicable.

Surface Error Rates. Surface word error rate (WER_{surf}) and character error rate (CER_{surf}) are computed on the full output strings, treating Arabic letters and diacritics as part of the same symbol sequence. These metrics reflect the end-to-end quality of the diacritized ASR output.

Lexical Error Rates. Lexical WER (WER_{lex}) and CER (CER_{lex}) are computed after removing diacritics from both reference and hypothesis texts. These metrics quantify recognition accuracy of the underlying Arabic letter sequence independently of diacritic placement.

Diacritic Error Rate. Diacritic error rate (DER) measures errors in diacritic realization only. DER is computed by aligning hypothesis and reference sequences at the character level and counting substitutions, insertions, and deletions involving diacritic symbols, normalized by the total number of reference diacritics. This metric isolates diacritic prediction performance from base transcription accuracy.

5 Results

We evaluate each model on the ClArTTS test split ($N = 205$) and report three complementary performance measures: (i) **surface error rates** (WER_{surf}, CER_{surf}) that treat diacritics as part of the output string, reflecting end-to-end transcription quality; (ii) **lexical error rates** (WER_{lex}, CER_{lex}) that ignore diacritics to isolate the precision of base letter recognition; and (iii) **diacritic error rate (DER)**, which specifically measures the proportion of incorrectly predicted diacritics and serves as our primary metric for quality of diacritization.

5.1 Overall ASR Performance

Table 1 presents the performance of all evaluated systems. The Arabic-adapted wav2vec 2.0 model substantially outperforms both Whisper configurations across all metrics. With greedy decoding, wav2vec 2.0 achieves a DER of 4.1%, compared to 21.3% for Whisper with greedy decoding and 15.7% with beam search. This represents a 74% relative reduction in diacritic errors compared to the best Whisper configuration.

The performance advantage extends beyond diacritics: wav2vec 2.0 achieves a surface WER of 14.9% versus 37.7% for Whisper-Beam, and a lexical WER of 7.8% versus 19.8%. The consistent superiority across both diacritized and undiacritized metrics indicates that wav2vec 2.0 provides better acoustic-phonetic modeling for this Classical Arabic speech recognition task.

5.2 Effect of Decoding Strategy

For the Arabic-adapted Whisper model, beam search decoding provides consistent improvements over greedy decoding. DER decreases from 21.3% to 15.7%, while surface WER improves from 43.4% to 37.7%. Lexical error rates also improve, with WER decreasing from 22.7% to 19.8%, suggesting that beam search benefits both diacritic prediction and base character recognition.

However, even with beam search, Whisper's DER remains nearly four times higher than wav2vec 2.0's greedy decoding performance, indicating that model architecture and pretraining have a substantially larger impact than decoding strategy alone.

5.3 Diacritic Error Analysis

Table 2 breaks down errors by diacritic category. Short vowel diacritics (fatha, damma, kasra) account for the majority of errors across all systems. Among these, fatha errors are most frequent, with 65 errors for Whisper-Greedy, 55 for Whisper-Beam, and 36 for wav2vec-Greedy. This pattern reflects both the high frequency of fatha in Arabic text and the acoustic challenges in distinguishing short vowels.

Sukun errors are also substantial (38-39 for Whisper, 23 for wav2vec), representing roughly 18-20% of total errors for Whisper and 19% for

Model	Decoding	N_{used}	WER_{surf}	CER_{surf}	WER_{lex}	CER_{lex}	DER
Whisper	Greedy	205	43.4%	12.1%	22.7%	6.7%	21.3%
Whisper	Beam	205	37.7%	9.8%	19.8%	6.1%	15.7%
wav2vec	Greedy	205	14.9%	2.8%	7.8%	2.0%	4.1%

Table 1: Performance on the ClArTTS test set. Surface error rates include diacritics; lexical error rates measure base letter recognition only. DER (diacritic error rate) isolates diacritization accuracy. Lower values indicate better performance.

wav2vec. Interestingly, sukun is the only diacritic category where beam search does not reduce errors for Whisper, with error counts remaining stable at 38-39.

In contrast, shadda (gemination marker) produces relatively few errors (8-12 across systems) and shows minimal variation between decoding strategies. This stability suggests that consonant gemination provides more robust acoustic cues than vowel distinctions.

Tanween errors, while individually less frequent, are collectively notable (28-30 total errors for Whisper, 19 for wav2vec). These case-marking diacritics present unique challenges as they depend on morphosyntactic context beyond local phonetic information.

5.4 Comparison Across Error Dimensions

Examining the relationship between lexical and diacritic errors reveals different model behaviors. For wav2vec 2.0, the lexical CER of 2.0% and DER of 4.1% show that diacritic errors occur at roughly twice the rate of base letter errors. For Whisper-Beam, lexical CER is 6.1% while DER is 15.7%, indicating that diacritization is disproportionately more challenging diacritic errors occur at 2.6 times the rate of letter errors.

This pattern suggests that while diacritization is harder than base letter recognition for both models, the gap is more pronounced for Whisper. The wav2vec 2.0 model appears to leverage acoustic information more effectively for both base transcription and diacritic prediction.

6 Discussion

This section interprets our experimental findings in the context of the research questions posed in the introduction: How do Arabic-specialized ASR models compare to multilingual models for direct diacritization, and what role does decoding strategy play in diacritic prediction quality?

6.1 Arabic-Specialized vs. Multilingual Models

Our central finding is that Arabic-specialized pretraining provides substantial benefits for direct diacritization from speech. The Arabic-adapted wav2vec 2.0 model achieves a DER of 4.1%, representing a dramatic improvement over the Arabic-adapted Whisper's 15.7% (with beam search). To contextualize these results, we relate them to the multilingual baseline study of Aldarmaki and Ghannam (2023), which evaluated unmodified multilingual ASR models for Arabic diacritization. That study reports that multilingual Whisper models achieve DERs in the range of 18–22% on broadcast news data, while multilingual XLS-R models achieve DERs of 8–12% depending on model size and dataset. Although direct numerical comparison is complicated by differences in test conditions (our read Classical Arabic versus their broadcast Modern Standard Arabic), the observed performance trends are informative: the Arabic-adapted wav2vec 2.0 model substantially outperforms the multilingual XLS-R baselines they report, while the Arabic-adapted Whisper model performs comparably to or slightly better than multilingual Whisper baselines.

This comparison suggests that **Arabic-specific adaptation provides substantial gains for wav2vec-family models**, reducing diacritic errors by approximately 50–66% relative to multilingual baselines. In contrast, the benefit is less pronounced for Whisper, where Arabic adaptation yields more modest improvements.

The performance gap between the two Arabic-adapted models (4.1% vs. 15.7% DER) further indicates that **model architecture and pretraining methodology play a major role in diacritic prediction quality**, beyond language-specific adaptation alone.

6.2 Effect of Decoding Strategy

Beam search decoding provides consistent but moderate improvements for the Whisper model, re-

Diacritic	Whisper-Greedy	Whisper-Beam	wav2vec-Greedy
Fatha	65	55	36
Damma	33	29	24
Kasra	32	30	13
Sukun	38	39	23
Shadda	12	12	8
Tanween Fatha	5	7	4
Tanween Damma	10	11	5
Tanween Kasra	13	12	10
Total	**208**	**195**	**123**

Table 2: Diacritic errors by type across systems on the test set.

ducing DER from 21.3% to 15.7%—a 26% relative improvement. This gain demonstrates that inference-time search can partially compensate for model limitations by maintaining multiple hypotheses and selecting globally more coherent diacritic sequences.

The mechanism behind this improvement relates to the ambiguous nature of Arabic short vowels. In Arabic speech, vowels like fatha, damma, and kasra often have subtle acoustic realizations that are highly context-dependent and affected by coarticulation. Greedy decoding commits to the locally most probable diacritic at each time step, which can lead to cascading errors when acoustic evidence is weak or ambiguous. Beam search mitigates this by deferring commitment until broader context is available, allowing the model to favor globally consistent diacritic patterns even when local acoustic evidence is ambiguous.

However, the impact of decoding strategy remains limited: beam search improves Whisper's DER by 5.6 percentage points, while the gap between Whisper-Beam and wav2vec-Greedy remains 11.6 percentage points. This indicates that **acoustic modeling quality dominates over decoding strategy** in determining overall diacritization performance.

Notably, sukun errors (representing vowel absence) do not decrease with beam search, and in fact slightly increase from 38 to 39 errors. This stability suggests that sukun detection relies primarily on local acoustic cues (brief silence, abrupt consonant transitions) that are either captured or missed by the acoustic model, with limited benefit from global context during search. This finding aligns with phonetic theory: sukun marks the absence of a phoneme rather than the presence of one, making it fundamentally different from vowel diacritics in terms of acoustic evidence.

This pattern suggests that most diacritic errors arise from limitations in acoustic representation rather than search mistakes. When the correct vowel is not strongly encoded in the acoustic features, expanding the hypothesis space provides limited benefit. Consequently, improving acoustic modeling is likely to yield larger gains than increasing decoding complexity.

6.3 Linguistic Interpretation of Errors

The error distribution across diacritic types reveals linguistically meaningful patterns. Short vowel diacritics (fatha, damma, kasra) constitute approximately 63% of total errors for wav2vec and 67% for Whisper, reflecting both their frequency in Arabic and their acoustic ambiguity. Within this category, fatha is most error-prone (36-65 errors), likely because it represents the most common vowel /a/ and exhibits the greatest acoustic variability across phonetic contexts.

Among short vowels, kasra shows the most dramatic improvement with wav2vec (13 errors) compared to Whisper-Beam (30 errors), suggesting that the high front vowel /i/ benefits particularly from wav2vec's acoustic modeling. This may relate to kasra's formant characteristics, which provide more distinctive acoustic signatures that wav2vec's learned representations capture effectively.

Sukun errors (23-39) represent a substantial portion of mistakes and exhibit unique behavior: they remain stable across Whisper's decoding strategies and constitute a proportionally larger share of wav2vec's errors (19% of total) compared to short vowels. This pattern reflects the inherent difficulty of detecting vowel absence from acoustic signal, particularly in fluent speech where syllable boundaries are not always clearly marked. The acoustic cues for sukun—such as consonant clusters without intervening vowel formants—are subtle and easily confused with very short or reduced vowels.

Shadda errors are consistently low (8-12) and stable across conditions, validating the expectation that consonant gemination is acoustically salient.

Geminated consonants in Arabic have markedly longer duration and often greater articulatory force than their singleton counterparts, providing robust acoustic evidence that both models capture reliably. The fact that beam search does not reduce shadda errors (12 for both Whisper configurations) further confirms that these errors stem from acoustic modeling limitations rather than contextual ambiguity.

Tanween diacritics present a distinct challenge: they remain problematic across all systems (19-30 total errors) and show inconsistent response to beam search. Unlike other diacritics that primarily encode phonetic information, tanween marks grammatical case and indefiniteness—features that depend on syntactic context beyond the local acoustic signal. Moreover, case endings are often weakly articulated or omitted entirely in natural Arabic speech, especially in Classical Arabic recitation where pausal forms are common. This variability in production makes tanween inherently difficult to predict from acoustics alone, suggesting that morphosyntactic diacritics may require linguistic knowledge beyond acoustic modeling.

6.4 Implications for System Design

Our findings have practical implications for designing diacritized Arabic ASR systems:

Prioritize model architecture and pretraining. The 11.6 percentage point DER gap between wav2vec-Greedy and Whisper-Beam indicates that selecting the right model architecture provides far greater benefit than optimizing decoding strategy. For applications requiring diacritized output, investing in models with strong acoustic representation learning (like wav2vec 2.0) yields better returns than post-hoc decoding improvements.

Arabic-specific adaptation is beneficial but architecture-dependent. While Arabic adaptation substantially improves wav2vec 2.0 (reducing DER from reported 8-12% for multilingual XLS-R to 4.1% in our study), the benefits appear more modest for Whisper. This suggests that language-specific fine-tuning should be paired with architectures that can effectively exploit language-specific acoustic patterns.

Beam search provides modest but worthwhile gains. The 26% relative DER reduction from beam search represents a meaningful improvement that requires no model retraining—only increased inference computation. For production systems where accuracy justifies computational cost, beam search is a straightforward enhancement.

Consider hybrid approaches for grammatical diacritics. The persistent difficulty with tanween suggests that purely acoustic approaches may be insufficient for morphosyntactic diacritics. Future systems might benefit from hybrid architectures that combine acoustic ASR with lightweight grammatical analysis for context-dependent diacritics.

7 Limitations

This study evaluates direct speech-to-diacritized Arabic ASR in a controlled setting, which limits the generalizability of the findings. All experiments are conducted on the ClArTTS corpus, consisting of approximately 12 hours of read Classical Arabic from a single speaker. While this enables precise analysis of diacritic realization, it does not reflect realistic conditions such as multi-speaker, spontaneous, dialectal, or noisy speech, and performance may therefore differ in practice.

In addition, our comparison is restricted to two representative model families (wav2vec 2.0 and Whisper) with one checkpoint each, so the observed differences may partly reflect model scale or pretraining choices. We also focus exclusively on a single-stage pipeline and do not include a two-stage ASR+text-diacritization baseline, and we evaluate only greedy and beam search decoding. Despite these constraints, the controlled design allows us to isolate the effects of acoustic modeling and Arabic-specific adaptation on diacritic prediction.

8 Conclusion

In this work, we study direct Arabic diacritized transcription from speech using a single-stage ASR pipeline that predicts diacritics jointly with Arabic letters and avoids text-based post-processing. Our approach compares two Arabic-adapted ASR models, wav2vec 2.0 and Whisper, using a unified experimental setup, and examines greedy and beam search decoding for Whisper. On the ClArTTS test split, wav2vec 2.0 with greedy decoding achieves a lower diacritic error rate than Whisper under both decoding strategies, and also yields lower surface and lexical word error rates. Beam search consistently improves Whisper performance, but a clear performance gap remains between the two models. The study is limited to single-speaker read Classical Arabic data and a restricted set of architectures and decoding methods, and future work should consider more diverse speech conditions and broader model coverage.

References

Sa'ed Abed, Mohammad Alshayeji, and Sari Sultan. 2019. Diacritics effect on arabic speech recognition. *Arabian Journal for Science and Engineering*, 44(11):9043–9056.

Haifa Alaqel and Khalil El Hindi. 2025a. Improving diacritical arabic speech recognition: Transformer-based models with transfer learning and hybrid data augmentation. *Information*, 16(3).

Haifa Alaqel and Khalil El Hindi. 2025b. Lightweight end-to-end diacritical arabic speech recognition using ctc-transformer with relative positional encoding. *Mathematics*, 13(20).

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

Sarah S. Alrumiah and Amal A. Al-Shargabi. 2023. A deep diacritics-based recognition model for arabic speech: Quranic verses as case study. *IEEE Access*, 11:81348–81360.

Hamzah Alsayadi, Abdelaziz Abdelhamid, Islam Hegazy, and Zaki Fayed. 2021. Arabic speech recognition using end-to-end deep learning. *IET Signal Processing*, 15.

Lilit Grigoryan, Nikolay Karpov, Enas Albasiri, Vitaly Lavrukhin, and Boris Ginsburg. 2025. Open automatic speech recognition models for classical and modern standard arabic.

Jonatas Grosman. 2021. Fine-tuned XLSR-53 large model for speech recognition in Arabic. `https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-arabic`.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*. Morgan & Claypool Publishers.

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*.

Lucian Petrică, Horia Cucu, Andi Buzo, and Corneliu Burileanu. 2014. A robust diacritics restoration system using unreliable raw text data. *Spoken Language Technologies for Under-Resourced Languages*, pages 215–220.

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: Human Language Technologies (Volume 1: Long Papers)*, pages 4166–4176.