Abjad AI at KSAA-2026 Shared Task 2: Grouped Speech Conditioning for Arabic Diacritization

Naif Alharthi, Ahmad Ghannam, Faris Alasmary, Kholood Al Tabash, Shouq Sadah, Lahouari Ghouti
Abjad AI, Saudi Arabia
{nalharthi, aghannam, falasmary, kaltabash, ssadah, lghouti}@abjad.com.sa

Abstract

We describe Abjad AI's submission to KSAA-2026 Shared Task 2 on automatic diacritization of Arabic speech dictation. The task requires generating fully diacritized text given speech audio and an undiacritized transcript. Because text-only diacritization cannot resolve ambiguities that are recoverable from the acoustic signal, we propose conditioning a character-level encoder-only Transformer (CATT) on speech representations. We introduce grouped speech conditioning, which downsamples speech encoder features into a small set of pooled tokens concatenated to the text input, enabling efficient fusion without architectural changes to CATT. We train with a two-phase schedule that first freezes the text encoder, then fine-tunes the full model. Our best system, using Whisper-small features with grouping size 5, achieves a Diacritization Error Rate (DER) of 5.68 and a Word Error Rate (WER) of 11.85 under the official primary test setting. Notably, we find that Whisper-small consistently outperforms Whisper-large-v3, suggesting that compact speech representations better suit this fusion setting. This is an extended and revised version of our previous work (Ghannam et al., 2025).

Keywords: Arabic diacritization, multimodal learning, speech conditioning, Whisper, KSAA shared task

1. Introduction

Arabic diacritics (tashkīl) are typically omitted in everyday writing, which increases lexical ambiguity and negatively impacts applications such as text-to-speech, reading assistance, and educational tools. Automatic diacritization aims to restore missing diacritics, but purely text-based systems struggle in cases where multiple vowelizations are plausible without additional context. In dictation settings, audio provides strong phonetic cues that can resolve ambiguity and guide diacritic placement.

KSAA-2026 Shared Task 2 targets *speech-aware* diacritization: given an audio clip and its undiacritized transcript, systems must output fully diacritized Arabic. Our system continues the direction of our earlier CATT-Whisper work (Ghannam et al., 2025), with a focus on a lightweight fusion mechanism and a systematic sweep over speech backbones and speech-to-text conditioning strength.

Contributions. In this paper, we describe our system and report three main findings: (1) A simple concatenation-based fusion strategy, where speech features are average-pooled over consecutive groups of G frames and prepended to the text sequence, is sufficient to improve diacritization over a text-only baseline without requiring cross-attention or other expensive mechanisms. (2) Whisper-small (Radford et al., 2022) consistently outperforms both smaller and larger Whisper variants as well as Squeezeformer (Kim et al., 2022), suggesting diminishing returns from increased encoder capacity in this setting. (3) A two-stage training schedule, which freezes the text encoder for the first 10 epochs before unfreezing, stabilizes training, though we note that this choice was not ablated against alternative schedules.

2. Task and Data

The task input consists of speech audio and undiacritized transcript pairs; the output is the diacritized transcript. We follow the official shared-task split and evaluation setup. We apply minimal text normalization: removing non-Arabic characters and collapsing whitespace. We do not use data augmentation.

2.1. Related Work

Recent work on Arabic diacritic restoration (DR) has increasingly explored multimodal approaches that incorporate speech information to address ambiguities that cannot be resolved from text alone (Elgamal et al., 2024). Because Arabic orthography typically omits short vowels, many surface forms admit multiple valid vowelizations, which makes acoustic evidence particularly valuable in speech-driven applications such as dictation and automatic speech recognition (ASR).

Speech-aware DR uses acoustic features extracted from audio to improve diacritization accuracy. Previous work has shown that speech information is especially helpful in ASR and dictation tasks, where pronunciation cues can help select the correct vowelization, particularly for dialectal and spontaneous speech (Shatnawi et al., 2024a,b).

Proceedings of The 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks, pages 252–255
May 11, 2026.

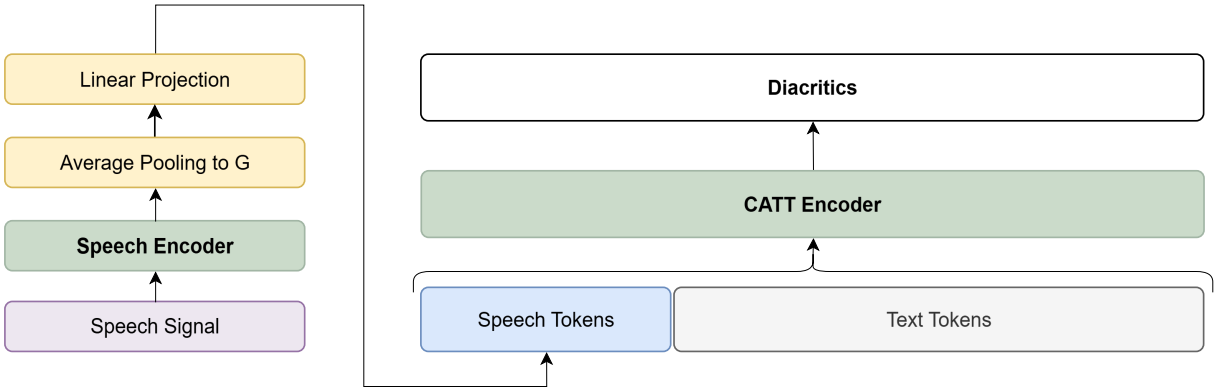

Figure 1: Overview of our grouped speech conditioning architecture for Arabic diacritization. The speech signal is encoded by a speech backbone (Whisper/Squeezeformer) to produce T frame-level vectors, which are average-pooled over consecutive groups of G frames. The pooled representations are then linearly projected to the CATT embedding dimension, concatenated with the text tokens, and fed into the CATT encoder to predict the output diacritics.

Multimodal fusion architectures have been proposed to combine speech and text representations for DR. Our prior work, CATT-Whisper, demonstrated that integrating a Whisper speech encoder with a character-level diacritization model improves performance by injecting phonetic information into the prediction process (Ghannam et al., 2025).

3. Methodology

In this work, we extend our previous CATT-Whisper model by improving the multimodal fusion between speech and text through a lightweight grouped speech conditioning mechanism. We also compare different speech encoders, including Whisper and Squeezeformer, to study the trade-offs between efficiency and diacritization performance in the KSAA dictation setting.

3.1. Text Backbone: CATT

We use CATT, a character-level encoder-only Transformer (char-BERT style) (Alasmary et al., 2024). Given a character sequence, CATT produces contextualized representations and a token-level classifier predicts diacritic tags per character. This architecture supports fine-grained diacritics and is compatible with long sequences.

3.2. Speech Backbone

We evaluate two speech encoders:

- **Whisper**: encoder representations from Whisper variants (base, small, large-v3) (Radford et al., 2022).
- **Squeezeformer**: an efficient Transformer-based ASR backbone (Kim et al., 2022).

3.3. Grouped Speech Conditioning

We use grouping to control how much speech information is injected while keeping compute costs low. We tune the grouping factor $G \in \{3, 5, 10, 15\}$, where G denotes the number of consecutive frame-level vectors averaged into one pooled speech token. Let $S \in \mathbb{R}^{T \times d_s}$ be frame-level speech representations and $X \in \mathbb{R}^{L \times d_t}$ be text embeddings. We downsample S by average-pooling every G frames, producing $\tilde{S} \in \mathbb{R}^{K \times d_s}$ where $K = T/G$. We then linearly project to the text dimension and concatenate:

$$Z = [\mathrm{Proj}(\tilde{S}); X] \in \mathbb{R}^{(K+L) \times d_t}.$$

The concatenated sequence Z is fed to the CATT encoder. A token-level classifier predicts diacritics for the text positions. Too small G retains a long speech-token sequence ($K = T/G$), which can inject excessive and potentially redundant acoustic context and increase compute due to the longer concatenated sequence $(K + L)$. Conversely, too large G over-compresses speech by averaging many frames into each token, reducing K and compute but potentially smoothing away short-vowel cues and other fine-grained phonetic information needed for accurate diacritization. We therefore tune G on the development set to balance acoustic detail against sequence length and efficiency.

4. Training Setup

We train with cross-entropy over diacritic labels using AdamW (Loshchilov and Hutter, 2019), batch size 32, and maximum length 1024. For all experiments, we initialize the CATT text backbone from a pretrained encoder-only char-BERT checkpoint,

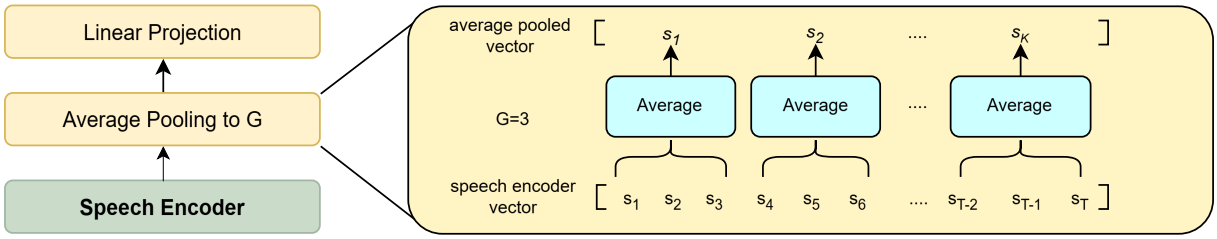

Figure 2: Grouped speech conditioning (G=3). Frame-level speech embeddings are grouped into chunks of three and average-pooled into pooled speech tokens ($s_1, s_2, \ldots, s_K$), then linearly projected to the CATT embedding space to condition the diacritization model.

while the speech encoder is initialized from the selected backbone (Whisper or Squeezeformer). We select hyperparameters on the development set using the official primary metric setting (WOCE, excluding no-diacritic), and we report the full metric grid in Table 1. We use a two-stage schedule:

1. **Freeze CATT** for the first 10 epochs, allowing the speech projection and fusion to adapt.
2. **Unfreeze CATT** and continue for 10 additional epochs.

Freezing the pretrained CATT backbone initially stabilizes training by preventing early updates from disrupting text representations. This warm-up lets the model learn a mapping from speech tokens to the CATT space for cross-modal conditioning. In the second stage, unfreezing CATT allows end-to-end adaptation, letting the text encoder specialize for dictation and better integrate acoustic cues for diacritics.

5. Evaluation Protocol

We use the same evaluation protocol used in the baseline, reporting WER and DER under: (i) **with case ending** vs. (ii) **without case ending**; and (iii) **including no-diacritic** vs. (iv) **excluding no-diacritic** [1]. Unless otherwise stated, we highlight the *without case ending* setting.

6. Results

6.1. Development Set: Backbone and Group Size Sweep

Table 1 summarizes the dev set sweep over speech backbones and grouping size G using the selected settings, excluding no-diacritic tokens and without case endings.

[1] https://github.com/rufaelfekadu/Diac

Backbone	Metric	Group size G			
		3	5	10	15
Sqf	DER	6.91	6.82	6.84	7.01
	WER	14.80	14.60	14.60	15.07
W-base	DER	6.92	6.64	6.56	6.42
	WER	14.80	14.32	14.08	13.80
W-small	DER	6.39	**6.18**	6.40	6.32
	WER	13.53	**13.29**	13.53	13.61
W-large	DER	6.75	6.50	6.56	6.61
	WER	14.68	14.16	14.12	14.28

Table 1: Development sweep (Sqf for Squeezeformer, W for Whisper).

6.2. Full Dev Metrics for the Best Model

	Incl		Excl	
Whisper-small, G=5	DER	WER	DER	WER
WCE	8.15	28.32	9.08	26.14
WOCE	4.62	15.76	6.18	13.29

Table 2: Full evaluation metrics for Whisper-small with $G = 5$ on the development set.

6.3. Shared Task Test Results

Participant	DER	WER
meshal	4.80	10.48
nadaadelmousa	5.15	11.06
naif_alharthi	**5.68**	**11.85**
nahian_abu	5.87	13.43
Hassan	6.86	14.32
omarnj10	25.27	31.07
astral_fate	25.03	49.07

Table 3: Official test leaderboard (Al Wazrah et al., 2026). Our values are bolded.

7. Discussion

We observe three consistent trends. First, **Whisper backbones outperform Squeezeformer** across all grouping factors, suggesting that Whisper's large-scale pretraining provides more transferable acoustic representations for dictation-style diacritization. Second, the **grouping factor G (frames averaged per pooled token)** offers an accuracy–efficiency knob: smaller G yields more pooled speech tokens ($K = T/G$) and longer sequences ($K + L$), which can add redundant context and increase compute, while larger G compresses speech more aggressively and may smooth away short-vowel cues. Empirically, Whisper-small peaks at $G{=}5$ on the dev set, indicating a good balance between acoustic detail and compression. Third, results under **excluding no-diacritic** provide a sharper signal for model selection than the including setting, since they focus on positions where diacritics matter. These findings align with prior speech-based diacritization work highlighting the value of acoustic cues and robustness techniques such as augmentation (Shatnawi et al., 2024a,b).

8. Conclusion

We presented Abjad AI's system for KSAA-2026 Shared Task 2: a character-level diacritization model conditioned on speech via grouped speech tokens. Whisper-small with $G{=}5$ performed best in our experiments.

9. Bibliographical References

Asma Al Wazrah, Waad Alshammari, Rawan Almatham, Raghad Al-Rasheed, Afrah Altamimi, Rufael Marew, Sawsan Alqahtani, Hanan Aldarmaki, Abdullah Alharbi, Abdulrahman Alshehri, Mohamed Assar, Amal Almazrua, and Abdulrahman AlOsaimy. 2026. Ksaa-2026 shared task on arabic speech dictation with automatic diacritization. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7)*.

Faris Alasmary, Orjuwan Zaafarani, and Ahmad Ghannam. 2024. CATT: Character-based Arabic tashkeel transformer. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 250–257, Bangkok, Thailand. Association for Computational Linguistics.

Salman Elgamal, Ossama Obeid, Mhd Kabbani, Go Inoue, and Nizar Habash. 2024. Arabic diacritics in the wild: Exploiting opportunities for improved diacritization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14815–14829, Bangkok, Thailand. Association for Computational Linguistics.

Ahmad Ghannam, Naif Alharthi, Faris Alasmary, Kholood Al Tabash, Shouq Sadah, and Lahouari Ghouti. 2025. Abjad AI at NADI 2025: CATT-whisper: Multimodal diacritic restoration using text and speech representations. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 757–761, Suzhou, China. Association for Computational Linguistics.

Sehoon Kim, Amir Gholami, Albert Shaw, Nicholas Lee, Karttikeya Mangalam, Jitendra Malik, Michael W. Mahoney, and Kurt Keutzer. 2022. Squeezeformer: An efficient transformer for automatic speech recognition. arXiv preprint arXiv:2206.00888.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. arXiv preprint arXiv:2212.04356.

Sara Shatnawi, Sawsan Alqahtani, and Hanan Aldarmaki. 2024a. Automatic restoration of diacritics for speech data sets. arXiv preprint arXiv:2311.10771.

Sara Shatnawi, Sawsan Alqahtani, Shady Shehata, and Hanan Aldarmaki. 2024b. Data augmentation for speech-based diacritic restoration. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 160–169, Bangkok, Thailand. Association for Computational Linguistics.