

Automatic Pronunciation Error Detection and Correction of the Holy Quran’s Learners Using Deep Learning

Abdullah Abdelfttah

Computer and Systems Engineering
Faculty of Engineering Ain Shams University
Cairo, Egypt
abdullahmlyossef@gmail.com

Mahmoud I. Khalil

Computer and Systems Engineering
Faculty of Engineering Ain Shams University
Cairo, Egypt
mahmoud.khalil@eng.asu.edu.eg

Hazem Abbas

Computer and Systems Engineering
Faculty of Engineering Ain Shams University
Cairo, Egypt
hazem.abbas@eng.asu.edu.eg

Abstract—Assessing spoken language is challenging, and quantifying pronunciation metrics for machine learning models is even harder. However, for the Holy Quran, this task is simplified by the rigorous recitation rules (tajweed) established by Muslim scholars, enabling highly effective assessment. Despite this advantage, the scarcity of high-quality annotated data remains a significant barrier.

In this work, we bridge these gaps by introducing: (1) A 98% automated pipeline to produce high-quality Quranic datasets – encompassing: Collection of recitations from expert reciters, Segmentation at pause points (waqf) using our fine-tuned wav2vec2-BERT model, Transcription of segments, Transcript verification via our novel Tasmeea algorithm; (2) 850+ hours of audio (300K annotated utterances); (3) A novel ASR-based approach for pronunciation error detection, utilizing our custom Quran Phonetic Script (QPS) to encode Tajweed rules (unlike the IPA standard for Modern Standard Arabic). QPS uses a two-level script: (Phoneme level): Encodes Arabic letters with short/long vowels. (Sifa level): Encodes articulation characteristics of every phoneme. We further include comprehensive modeling with our novel multi-level CTC Model which achieved 0.16% average Phoneme Error Rate (PER) on the testset. We release all code, data, and models as open-source: <https://obadx.github.io/prepare-quran-dataset/>

Index Terms—Mispronunciation Detection Model, Arabic Natural Language Processing, End-to-end Models

I Introduction

Assessing pronunciation is not a simple task [1], as it does not only rely on pronouncing phonemes correctly but also involves other factors like intonation, prosody, and stress. Does learning these mean one is done? No—other factors include fluency and completeness [1]. However, the Holy Quran presents unique characteristics: it is among the easiest spoken texts to learn despite containing special phonemes absent in other languages.

The pronunciation of the Holy Quran is governed by rigorously strict rules formally defined by ancient Muslim scholars since the 6th century. Despite their beauty and precision, these rules have not been comprehensively digitized (to our knowledge) for Quranic pronunciation assessment.

Although RDI pioneered computer-aided Quranic instruction [2], they neither disclosed their phoneticization process

nor released data/models. Consequently, new research must start from basics: defining phoneticization, data, and models. To bridge this gap, we introduce:

- **A Phonetizer:** Encodes *all* Tajweed rules and articulation attributes (*Sifat*) defined by classical scholars, except *Ishmam* (إشمام)
- **A 98% automated pipeline:** Generates highly accurate datasets from expert recitations
- **A dataset:** ~300K annotated utterances (890 hours)
- **Integration:** Our multi-level CTC model proves the Quranic phonetic script is learnable (0.16% average phoneme error rate)

The paper is organized as follows:

- **Related Work:** Expands on strengths/weaknesses of prior research
- **Quran Phonetic Script:** Introduces our two-level script: **phonemes** and **Sifat** (10 attributes → 11 total levels)
- **Data Pipeline:** Stages include:
 - 1) Digitized Quran script as foundation
 - 2) *Hafs* methodology criteria
 - 3) Expert recitation collection
 - 4) Segmentation at pause points (وقف)
 - 5) Segment transcription
 - 6) Validation via *Tasmee* (تسميع) algorithm
- **Modeling:** Demonstrates learnability of the phonetic script
- **Results:** Analysis of outcomes
- **Limitations & Future Work:** Next research directions
- **Conclusion:** Summary of contributions
- **Appendix:** Details on *Mushaf* attributes and algorithms

II Related Work

II-A Quran Pronunciation Datasets

We discuss the most important datasets here. everyayah¹ is the largest openly available dataset with 26 complete *Mushafs* segmented and annotated by Ayah by experts like Al Hossary

¹everyayah.com

and non-experts such as Fares Abbad. Qdat [3] contains 1509 utterances of single specific Ayahs labeled for three rules: Madd, Ghunna, and Ikhfaa. Although the scale is relatively small, it was widely adopted by the community [4], and [5] due to being open-source. The Tarteel v1 dataset [6] consists of 25K utterances with diacritics and no Tajweed rules. The latter is the Tarteel² private dataset, a massive 9K-hour collection annotated with diacritics without Tajweed rules. The most recent benchmark is IqraaEval [7], which presents a test set of 2.2 hours from 18 speakers, but uses Modern Standard Arabic (MSA) without Tajweed rules.

II-B Quran Pronunciation Models

To our knowledge, the first work addressing automated pronunciation assessment for the Holy Quran is RDI [2], which built a complete system for detecting pronunciation errors. The work does not specify which errors were included or excluded but mentions testing Qalqala, Idgham, and Iqlab rules. It also omits details on Quranic word phoneticization. Subsequent work continued with [8] and [9], using Deep Neural Networks (DNNs) to replace HMMs and improve the system. Many studies rely on modeling phoneme duration for duration-dependent rules like Madd and Ghunna, e.g., [10], [11], but use limited datasets and focus on specific verses rather than the entire Quran. Others concentrate on detecting specific rules like Qalqala [4] or Ghunna and Madd [5], [12]. However, most efforts except RDI work train on small-scale datasets from specific Quranic chapters.

At this point, Tarteel emerges; though lacking Tajweed rules, they built a robust ASR system for diacritized character detection. They developed a crowd-sourced dataset [6] of 25K utterances (68 hours), later extended via application users to 9K hours of private annotated data. The work most aligned with our vision of detecting all error types (including Tajweed and *Sifat*/articulation attributes) is [13]. Although it relies on HMMs and minimal data, it introduces a multi-level detection system: *Makhraj* (phoneme level) and Tajweed rules level.

II-C Pretrained Speech Encoders with Self-Supervised Learning (SSL)

Speech pretraining began early [14] but was constrained by the sequential nature of Recurrent Neural Networks (RNNs) [15]. The rise of Transformers [16] facilitated greater GPU parallelization, enabling large-scale pretraining. BERT [17] using Masked Language Modeling (MLM) introduced large unsupervised pretraining which has better results on downstream tasks. This soon extended to speech with wav2vec [18] and wav2vec2.0, which added product quantization [19]. Conformer later replaced vanilla Transformers for speech by integrating convolution [20]. Google’s Wav2Vec2-BERT [21] then applied MLM to speech. Finally, Facebook extended Wav2Vec2-BERT pretraining [22] to 4.5M hours (including 110K Arabic hours), ideal for low-resource language fine-tuning.

²tarteel.ai

III Quran Phonetic Script

We consider the Quran Phonetic Script to be the most valuable and important contribution of our work. By formalizing the assessment of Holy Quran pronunciation as an ASR problem represented through this script, we provide a comprehensive solution to the task.

III-A Motivation for Developing Quran Phonetic Script

Modern Standard Arabic (MSA) orthography cannot adequately represent Tajweed rules for error detection. For example, MSA cannot measure the precise length of Madd rules. Previous research (e.g., [23]) focused on single rules like Qalqalah. Our phonetic script addresses this limitation by capturing all Tajweed pronunciation errors except *Ishmam* (إشمام), which involves a visual mouth movement without audible output.

III-B Background

We based our script on classical Muslim scholarship rather than the International Phonetic Alphabet (IPA) for these reasons:

- 1) **Historical Precedence:** Muslim scholars from the 6th to 14th centuries rigorously defined Quranic errors centuries before modern phonetics emerged in the West.
- 2) **Scientific Foundation:** Scholars like Al-Khalil ibn Ahmad (6th century AH) systematically described articulations and attributes with remarkable accuracy comparable to modern phonetics [24].
- 3) **Pedagogical Relevance:** Learners’ errors align with classical definitions according to expert Quran teachers.

III-C Defining Mistakes in Quran Recitation

Following [25], Quran recitation errors fall into three categories:

- **Articulation Errors:** Incorrect pronunciation of phonemes
- **Attribute Errors:** Mistakes in letter characteristics (*Sifat al-Huruf*)
- **Tajweed Rule Errors:** Incorrect application of rules like Ghunnah, Madd, etc.

Our script comprehensively addresses all three aspects through two output levels:

- **Phonemes Level:** Represents letters, vowels, and Tajweed rules
- **Sifat Level:** Represents articulation attributes for each phoneme

Refer to tables: VI VII for Phonemes and Sifat levels.

Key Design Principles

- 1) **Madd Representation:**
 - Normal Madd appears as consecutive madd symbols (e.g., 4-beat Madd: (||||
 - Madd al-Leen represented with multiple waw/yaa symbols
- 2) **Emphatic Articulation:**

- Stressed Ghunnah (e.g., النون المشددة) as three consecutive noon symbols (ننن)
- Ikhfa represented as three consecutive noon_mokhfah (□□□) or meem_mokhfah (□□□)

3) Idgham Handling:

- Assimilation represented by doubling (e.g., يَعْملُ مَنْ → مَيِّيعَمَلْ)

(4) Special Cases:

- Sakin: No following symbol
- Imala: fatha_momala and alif_momala
- Rawm: dama_mokhtalasa marker

Example: In table I shows how our phonetizer works.

III-D Development Methodology

Our phonetization has two steps:

1) Imlaei to Uthmani Conversion

We selected Uthmani script as our foundation because:

- Contains specialized Tajweed diacritics (Madd, Tasheel, etc.)
- Preserves pause rules critical for recitation (e.g., stopping on رحمت)

In order to do that, we created an annotation UI to manually annotate misaligned words in both scripts. For example II, after that, we developed an algorithm that relies on the annotations to convert Imlaei to Uthmani.

2) Uthmani to Phonetic Script Conversion

We implemented the process through 26 sequential operations. Each operation contains one or more regular expressions, as shown in the Appendix VIII-A.

3) Extracting Sifat: Next, we extract the 10 attributes (Sifat) defined in Table VII, excluding Inhiraf (إحراف) as it describes the shidda/rakhawa spectrum, and Leen (اللين) as it was already handled through our Madd representation.

IV Data Preparation

To prepare the data, we first defined selection criteria. We aimed to collect recitations from the best reciters worldwide to serve as references for judging Quran learners. In our study, we considered only *Hafs* riwayat (رواية حفص) as it's the most popular recitation method globally. Recognizing that manual data annotation requires significant effort and time, we created a 98% automated pipeline for data collection. The steps are: (1) Choose a digitized Quran script as the project foundation. (2) Define criteria for *Hafs* methodology. (3) Collect expert recitations (4) Segment recitations at pause points (وقف) (5) Transcribe segments. (6) Validate data through *Tasmee* (تسميع) Algorithm. (7) Develop Quran Phonetic Script.

We define a *Moshaf* as a complete Quran recitation (chapters 1-114) by a specific reciter. Statistics are summarized in table III. We manually annotated 5400 samples out of 286,537 utterances, resulting for the automation ratio of 98%.

IV-A Choose a Digitized Version of the Holy Quran

The Quran has multiple digitized versions including Tanzil³ and King Fahd Complex⁴. We chose Tanzil because:

- It uses standard Unicode characters
- Contains both *Imlaei* and *Uthmani* versions
- Maintains high accuracy

We excluded KFGQPC due to its evolving/unstable nature compared to Tanzil.

IV-B Define Variant Criteria for Hafs

Hafs riwayat contains variants, e.g., *Madd Al-Munfasil* (مد المنفصل) can extend 2, 4, 5, or 6 beats. We rigorously defined these variants through the Qira'at literature [26], summarized in the following attributes in the Appendix section VIII-B.

IV-C Collect Expert Recitations

We collected recitations from 22 world-class reciters with premium audio quality, totaling **893 hours** pre-filtering.

Number of Moshaf Items	Number of Reciters
27	22
Total Hours	Total Size (GB)
893.1	48.48

Fig. 1. Database Collection Statistics

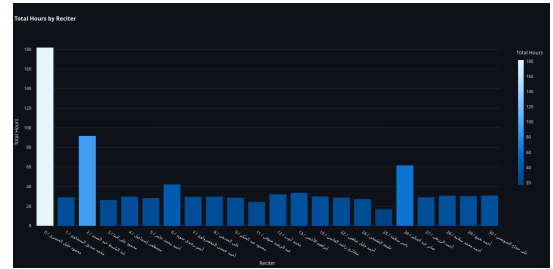


Fig. 2. Reciters Statistics

We developed a web GUI using Streamlit⁵ that:

- Downloads and extracts metadata for each track
- Organizes data by Moshaf (each chapter as "001.mp3")
- Annotates Moshaf attributes

IV-D Segment Recitations

Since Tajweed rules are affected by pauses (وقف), accurate segmentation is crucial. We initially tested open-source Voice Activity Detection (VAD) models including SileroVAD [27] and PyAnnotate [28]. Poor Quran-specific performance led us to develop a custom segmenter by fine-tuning Wav2Vec2-BERT [22] for frame-level classification.

³<https://tanzil.net>

⁴<https://qurancomplex.gov.sa>

⁵<https://streamlit.io/>

TABLE I
Examples of Uthmani to Phonetic Script Conversion with Sifat Attributes

Uthmani	Phonetic	H/J	S/R	T/T	Itb	Saf	Qal	Tik	Taf	Ist	Gho
أ	ء	jahr	shd	mrq	mnf	no	nql	nkr	ntf	nst	nmg
ب	ث	hams	shd	mrq	mnf	no	nql	nkr	ntf	nst	nmg
ج	خ	hams	rkh	mrq	mnf	no	nql	nkr	ntf	nst	nmg
د	ح	hams	rkh	mrq	mnf	no	nql	nkr	ntf	nst	nmg
هـ	ج	jahr	shd	mrq	mnf	no	nql	nkr	ntf	nst	nmg
و	□□□□□	jahr	rkh	mrq	mnf	no	nql	nkr	ntf	nst	nmg
ز	ننن	jahr	btw	mrq	mnf	no	nql	nkr	ntf	nst	mg
ح	□	jahr	rkh	mrq	mnf	no	nql	nkr	ntf	nst	nmg

Phonetization of word (تَجَوَّى)

Attribute Abbreviations:

H/J: Hams/Jahr S/R: Shidda/Rakhawa T/T: Tafkheem/Taqeeq Itb: Itbaq
Saf: Safeer Qal: Qalqla Tik: Tikraar Taf: Tafashie Ist: Istitala Gho: Ghonna

Value Abbreviations:

shd: shadeed rkh: rikhw btw: between mrq: moraqaq
mof: mofakham mnf: monfateh mtb: motbaq no: no safeer
nql: not_moqaalal nkr: not_mokarar ntf: not_motafashie
nst: not_mostateel nmg: not_maghnoon mg: maghnoon

TABLE II
Example of misalignment between Uthmani and Imlaey Scripts

Imlaey Script	Uthmani Script
لَمْ يَنْ يَأْ	يَنْتَوَّى

TABLE III
Dataset Statistics per Moshaf

Moshaf ID	Hours	Length
0.0	28.47721296	9133
0.1	40.31257093	10764
0.2	49.46541671	9971
0.3	37.18758118	12604
1.0	28.40784367	10939
2.0	51.04665234	9942
2.1	30.02847051	10394
3.0	25.19377593	10444
4.0	29.12333379	10994
5.0	28.01777693	11482
6.0	39.38568468	12435
7.0	28.25627201	9907
8.0	30.85935158	10330
9.0	27.95178738	10642
11.0	24.00685052	10363
12.0	33.42429862	9880
13.0	33.99108879	9377
19.0	30.11410843	11278
22.0	28.10947704	10332
24.0	28.51243509	9868
25.0	16.92910042	7922
26.0	30.44461112	11565
26.1	32.71190443	11850
27.0	28.05097968	11213
28.0	31.05318768	10535
29.0	27.78900316	11061
30.0	29.14366461	11312
Total	847.9944402	286537

IV-D1 Preparing Segmenter Data: We selected mosahf compatible with SileroVAD v4, using EveryAyah⁶ (pre-

⁶<https://everyayah.com/>

segmented by ayah) as ground truth. After tuning parameters per Moshaf:

- Threshold
- Minimum silence duration (merges segments)
- Minimum speech duration (discards short segments)
- Padding (added at segment boundaries)

IV-D1a Data Augmentation: Using the Audiomentations [29] library, we replicated SileroVAD's noise setup on 40% of samples, adding:

- TimeStretch (0.8x-1.5x) to simulate recitation speeds
- Sliding window truncation (1-second windows) for long samples instead of exclusion

IV-D2 Training Segmenter: We fine-tuned Wav2Vec2-BERT for frame classification (1 epoch):

Results of our segmenter on unseen mosahf in table IV-D2:

TABLE IV

Test results of the segmenter on unseen full mosahf. The result is validated by actual usage of the segmenter

Metric	Value
Test Loss	0.0277
Test Accuracy	0.9935
Test F1 Score	0.99476

IV-E Transcribe Segmented Parts

We employed Tarteel ASR [30] (Whisper fine-tuned on Quranic recitations [31]). To handle its 30-second limit, we used sliding window truncation (10-second windows), with verification in the next step.

IV-F Verification of Segmentation and Transcription

Segmentation Verification: Manual inspection of 50-75 random samples per Moshaf. Moshaf 25.0 was excluded due to poor segmentation.

Transcription Verification: *Tasmeea*-inspired algorithm: (1) Match segments to Quranic text. (2) Identify missing surah parts. (3) Manual correction.

Refer to the *Tasmeea* Algorithm in the Appendix 1

After matching, we catalogued missing Quranic portions per surah. Then correct transcription errors identified through the above process.

V Modeling The Quran Phonetic Script

Our Quran Phonetic script has two outputs: phonemes and sifat (which has 10 attributes). We modeled this as follows: Imagine you are given an input speech utterance and want to output transcripts in Arabic, English, French, and German simultaneously. We implemented this as a speech encoder with a linear layer for each language. Replacing languages with our 11 levels (phonemes and the 10 sifat), we obtain 11 parallel transcription levels. We chose CTC loss [32] without language model integration because we aim to capture what the user actually said, not what they intended to say. We name our architecture **Multi-level CTC**.

We compute the loss by averaging all CTC losses for the 11 levels, assigning a weight of 0.4 to the phonemes level as it has the largest vocabulary size (43) compared to other levels.

We fine-tuned Facebook’s Wav2Vec2-Bert [22] for a single epoch with a constant learning rate of $5e-5$ with 64 batch size. We applied augmentations identical to Silero VAD [27] using the audiomentations library [29], with additional augmentations: TimeStretch and GainTransition. We filtered out samples longer than 30 seconds not due to model limitations, but for efficient GPU utilization - sacrificing only 3k samples out of 250k training samples.

The training was done using an H200 GPU with 141 GB of GPU memory for 7 hours.

VI Results

We trained on all Mushaf provided above, reserving Mushaf 26.1 and 19.0 for testing. The evaluation results are presented in Table V. We notice that the Average Phoneme Error Rate is **0.16%**, which proves our idea that the Quran Phonetics Script can be learned.

We tested some actual samples with errors in Madd, Ghunna, Qalqala, and Tafkheem. The model was able to detect them even though it was not trained on recitations with pronunciation errors. Although this is not sufficient, we plan to annotate available datasets like [6] to conduct comprehensive evaluations.

We note that the PER across nearly all levels is balanced, except for the phoneme level, as it has a much larger vocabulary of 44 (with padding).

TABLE V
Test Results on moshaf 26.1 and 19.0

Metric	Value
loss	0.01162
per_phonemes	0.00543
per_hams_or_jahr	0.00117
per_shidda_or_rakhawa	0.00172
per_tafkheem_or_taqeeq	0.00167
per_itbaq	0.00092
per_safer	0.00132
per_qalqla	0.00085
per_tikraar	0.0009
per_tafashie	0.0016
per_istitala	0.0008
per_ghonna	0.0013
average_per	0.0016

VII Limitations and Future Work

Our primary limitation is that our dataset consists of golden recitations with no errors, limiting our ability to evaluate

Voice Activity Detection (VAD) Approaches

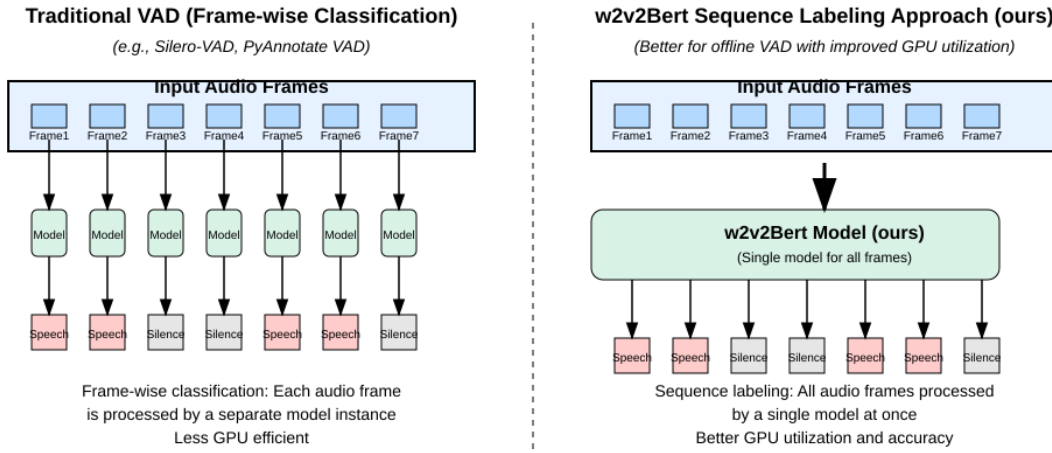


Fig. 3. VAD architecture vs. standard streaming models

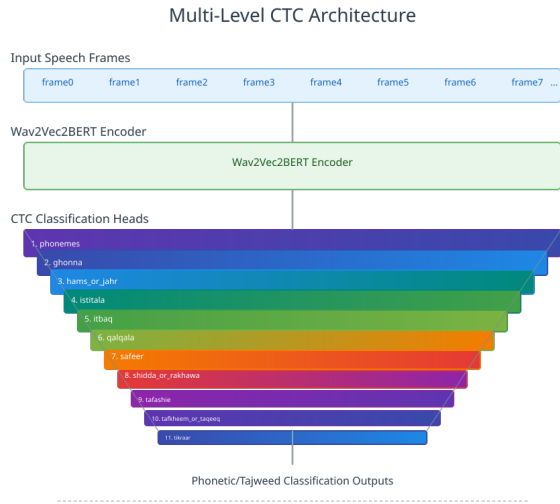


Fig. 4. Multi-level CTC loss Architecture composed of 11 Heads for every level and CTC loss for every level with weighted average loss

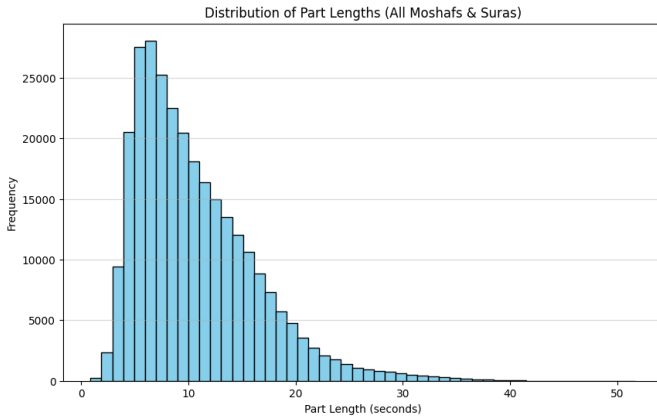


Fig. 5. Recitations lengths in seconds for the whole dataset

performance on real-world data. Although we tested on a few actual samples and successfully detected *madd*, *ghunnah*, and *qalqalah* errors, we need to develop a comprehensive dataset containing error-containing recitations transcribed with our Quran Phonetic Script.

A secondary limitation arises from attribute-specific articulation patterns: Certain attributes apply exclusively to individual letters, such as *Istitala* for (ض) and *Tikrar* for (ج). Consequently, we expect our model will be unable to capture instances of (ض) without *Istitala* or (ج) without *Tikrar*. This limitation similarly applies to Tajweed rules that occur less frequently in the Holy Quran, such as *Imala*, *Rawm*, and *Tasheel*.

VIII Conclusion

We present a novel approach for assessing pronunciation errors in Holy Quran learners through a multi-level Quran Phonetic Script that captures all pronunciation errors for

Hafs (except *Ishmam*, as it is a visual diacritic not orally produced). We provide 890 hours of annotated audio data with 300K samples, a 98%-efficient pipeline for generating similar datasets, and a novel multi-level CTC model. Achieving a 0.16% average phoneme error rate on unseen test data proves the learnability of the Quran Phonetic Script, fundamentally transforming Holy Quran pronunciation assessment methodology.

Acknowledgment

We express our profound gratitude to several individuals and organizations: Sheikh Ahmed Abdelsalam, Sheikh Mustafa Fathy, and Sheikh Mohamed Rabee for their invaluable guidance in understanding and representing Tajweed rules and learner mistakes. We also thank BA-HPC⁷ for providing access to high-performance computing resources and facilitating data processing. Special appreciation goes to Engineer Khaled Bahaa for his assistance with paying method for GPUs.

References

- [1] Y. E. Kheir, A. Ali, and S. A. Chowdhury, "Automatic pronunciation assessment—a review," *arXiv preprint arXiv:2310.13974*, 2023.
- [2] M. Sherif, A. Samir, A. Khalil, and R. Mohsen, "Enhancing usability of capl system for quran recitation learning," *INTERSPEECH*, 2007.
- [3] H. M. Osman, B. S. Mustafa, and Y. Faisal, "Qdat: a data set for reciting the quran," *International Journal on Islamic Applications in Computer Science And Technology*, vol. 9, no. 1, pp. 1–9, 2021.
- [4] D. Omran, S. Fawzi, and A. Kandil, "Automatic detection of some tajweed rules," in *2023 20th Learning and Technology Conference (LT)*, 2023, pp. 157–160.
- [5] D. Shaiakhmetov, G. Gimaletdinova, K. Momunov, and S. Cankurt, "Evaluation of the pronunciation of tajweed rules based on dnn as a step towards interactive recitation learning," *arXiv preprint arXiv:2503.23470*, 2025.
- [6] H. I. Khan, A. Abid, M. M. Moussa, and A. Abou-Allaban, "The tarteel dataset: crowd-sourced and labeled quranic recitation," 2021.
- [7] Y. E. Kheir, O. Ibrahim, A. Meghanani, N. Almarwani, H. O. Toyin, S. Alharbi, M. Alfadly, L. Alkanhal, I. Selim, S. Elbatal *et al.*, "Towards a unified benchmark for arabic pronunciation assessment: Quranic recitation as case study," *arXiv preprint arXiv:2506.07722*, 2025.
- [8] S. M. Abdou and M. Rashwan, "A computer aided pronunciation learning system for teaching the holy quran recitation rules," in *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 2014, pp. 543–550.
- [9] M. Al-Marri, H. Raafat, M. Abdallah, S. Abdou, and M. Rashwan, "Computer aided qur'an pronunciation using dnn," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 5, pp. 3257–3271, 2018.
- [10] A. Mohammed, M. S. B. Sunar, and M. S. H. Salam, "Recognition of holy quran recitation rules using phoneme duration," in *International Conference of Reliable Information and Communication Technology*. Springer, 2017, pp. 343–352.
- [11] A. M. A. Alqadasi, A. M. Zeki, M. S. Sunar, M. S. B. H. Salam, R. Abdulghafor, and N. A. Khaled, "Improving automatic forced alignment for phoneme segmentation in quranic recitation," *IEEE Access*, vol. 12, pp. 229–244, 2023.
- [12] Y. S. Alsahafi and M. Asad, "Empirical study on mispronunciation detection for tajweed rules during quran recitation," in *2024 6th International Conference on Computing and Informatics (ICCI)*, 2024, pp. 39–45.
- [13] B. Putra, B. T. Atmaja, and D. Prananto, "Developing speech recognition system for quranic verse recitation learning software," *IJID (International Journal on Informatics for Development)*, vol. 1, no. 2, pp. 1–8, 2012.
- [14] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

⁷<https://hpc.bibalex.org/>

[illegible]

Appendix

VIII-A Uthmani to Phonetic Conversion Operations

The 26 sequential phonetization operations:

- 1) **DisassembleHrofMoqatta** (تفكيك حروف مقطعة): Separates Quranic initials (e.g., الم, ال) into individual letters.

TABLE VI
Phoneme Set (43 Symbols)

Phoneme Name	Symbol
hamza	ء
baa	ب
taa	ت
thaa	ث
jeem	ج
haa_mohmala	ح
khaa	خ
daal	د
thaal	ذ
raa	ر
zay	ز
seen	س
sheen	ش
saad	ص
daad	ض
taa_mofakhama	ط
zaa_mofakhama	ظ
ayn	ع
ghyn	غ
faa	ف
qaf	ق
kaf	ك
lam	ل
meem	م
noon	ن
haa	ه
waw	و
yaa	ي
alif	ا
yaa_madd	آ
waw_madd	أ
fatha	َ
dama	ِ
kasra	ِ
fatha_momala	ْ
alif_momala	ـ
hamza_mosahala	ء
qlqla	ء
noon_mokhfah	ء
meem_mokhfah	ء
sakt	ء
dama_mokhtalasa	ء

- 2) **SpecialCases** (حالات خاصة): Handles special words like `يسط` that have different pronunciation forms defined in `MoshafAttributes`.
- 3) **BeginWithHamzatWasl** (البدء الوصل): Processes words starting with connecting hamza (◌ِ) and converts it to hamza (ء) with appropriate harakah for nouns and verbs.
- 4) **BeginWithSaken** (البدء بساكن): Manages words beginning with a consonant (sakin) like `لَيَطْعُ` as Arabic doesn't start utterances with consonants.
- 5) **ConvertAlifMaksora** (تحويل المقصورة): Converts `ى` in Uthmani script to either yaa (ي) or alif (ا) based on context.
- 6) **NormalizeHmazat** (توحيد الهمزات): Standardizes hamza

TABLE VII
Sifat Set (10 Attributes)

Sifat (English)	Sifat (Arabic)	Available Attributes (English)	Available Attributes (Arabic)
hams_or_jahr	الجهر أو الهمس	hams, jahr	جهر همس,
shidda_or_rakhawa	الرخاوة أو الشدة	shadeed, between, rikhw	رخو بين, بين شديد,
tafkheem_or_taqeeq	التزقيق أو التقخيم	mofakham, moraqaq	مرفق مفخم,
itbaq	الإطباق	monfateh, motbaq	مطبق منفتح,
safeer	الصفير	safeer, no_safeer	صفير لا صفير,
qalqla	القلقلة	moqalqal, not_moqalqal	مقلقل غير مقلقل,
tikraar	التكرار	mokarar, not_mokarar	مكرر غير مكرر,
tafashie	التقشي	motafashie, not_motafashie	متقشي غير متقشي,
istitala	الاستطالة	mostateel, not_mostateel	مستطيل غير مستطيل,
ghonna	الغنة	maghnoon, not_maghnoon	مغنون غير مغنون,

forms ء to ئ) و ا (أ)

- (7) **IthbatYaaYohie** (إثبات) ياء (يحيى) Handles words like يُحْيِي where two yaa letters occur - resolves conflicts when pausing on words with consecutive consonants (التقاء الساكنين) by adding another yaa at end.
- (8) **RemoveKasheeda** (إزالة الكشيدة) Deletes elongation marks (—) from text.
- (9) **RemoveHmzatWaslMiddle** (إزالة الوصل همزة) Removes connecting hamza (◌ْ) in non-initial positions.
- (10) **RemoveSkoonMostadeer** (حذف سكون فوق الحرف) Eliminates letters with circular sukoon diacritics (مستدير) like alif in جَمْعُوا.
- (11) **SkoonMostateel** (سكون مستطيل) Removes alif with elongated sukoon mid-word and adds it at the end during pauses (وقف).
- (12) **MaddAlewad** (مد العوض) Removes alif after tanween fatha mid-word and adds alif while removing tanween at pause positions (وقف).
- (13) **WawAlsallah** (واو الصلاة) Replaces letter waw (و) with small alif above combined with alif.
- (14) **EnlargeSmallLetters** (تكبير الحروف الصغيرة) Resizes miniature Arabic letters to standard proportions.
- (15) **CleanEnd** (تنظيف النهاية) Removes redundant diacritics and spaces at word endings.
- (16) **NormalizeTaa** (توحيد التاء) Converts ة (taa marbuta) to ت or ه based on context, and converts final ة to haa (ه).
- (17) **AddAlifIsMAllah** (إضافة اسم ألف الله) Inserts compensatory alif in derivatives of "الله".
- (18) **PrepareGhonnaIdghamIqlab** (تهيئة الإقلام الغنة والإقلاب) Preprocesses text for nasalization, assimilation, and conversion rules.
- (19) **ItiqaaAlsaknan** (التقاء الساكنين) Resolves consecutive consonants by inserting vowels.
- (20) **DeleteShaddaAtBeginning** (حذف الشدة البداية) Removes shadda (◌ْ◌ْ) from word-initial letters.
- (21) **Ghonna** (غنة) Applies nasalization during pronunciation of sakin noon and tanween.
- (22) **Tasheel** (تسهيل) Adds a letter representing alif with tasheel easing.
- (23) **Imala** (إمالة) Converts fatha with imala to fatha_momala phoneme and alif with imala to

alif_momala phoneme.

- (24) **Madd** (مد) Adds madd symbols for all madd types, inserting madd_alif (◌ِ◌ِ), madd_waw (◌ُ◌ُ) and madd_yaa (◌َ◌َ).
- (25) **Qalqla** (قلقلة) Adds echoing effect to ط, ق, ب, ج letters with sukoon.
- (26) **RemoveRasHaaAndShadda** (إزالة علامة الحاء رأس) Deletes sukoon diacritic marks.

Algorithm 1 Tasmeea Algorithm

Require: $text_segments = [s_1, s_2, \dots, s_n]$, $sura_idx$,
 $overlap_words = 6$, $window_words = 30$,
 $acceptance_ratio = 0.5$, flags for special phrases

Ensure: List of tuples ($match, ratio$) per segment

- 1: $aya \leftarrow 1$ {Start at first verse}
- 2: $penalty \leftarrow 0$
- 3: **for** each segment s_i in $text_segments$ **do**
- 4: $norm_text \leftarrow normalize(s_i)$ {Remove spaces/diacritics}
- 5: $min_win \leftarrow window_words - 10$, $max_win \leftarrow window_words + 10$
- 6: $start_range \leftarrow [-(overlap + penalty), (overlap + \max(window_words, max_win) + penalty)]$
- 7: **if** first segment **and** $include_istiaatha$ **then**
- 8: Check istiaatha special case
- 9: **else if** last segment **and** $include_sadaka$ **then**
- 10: Check sadaka special case
- 11: **end if**
- 12: $best_ratio \leftarrow 0$, $best_match \leftarrow null$
- 13: **for** each start position p in $start_range$ **do**
- 14: **for** each window size $w \in [min_win, max_win]$ **do**
- 15: $c \leftarrow \text{extract candidate at } (aya, p, w)$
- 16: $dist \leftarrow \text{edit_distance}(norm_text, c)$
- 17: $ratio \leftarrow 1 - \frac{\min(dist, |norm_text|)}{|norm_text|}$
- 18: **if** $ratio > best_ratio$ **or** ($ratio = best_ratio$ **and** $|p| < |best_start|$) **then**
- 19: update $best_ratio$, $best_match$, $best_start$, $best_window$
- 20: **end if**
- 21: **end for**
- 22: **end for**
- 23: **if** $best_ratio < acceptance_ratio$ **then**
- 24: output ($null, best_ratio$)
- 25: $penalty \leftarrow max_win$
- 26: $aya \leftarrow aya + 1$ {Default advance}
- 27: **else**
- 28: output ($best_match, best_ratio$)
- 29: $aya \leftarrow aya + best_start + best_window$
- 30: $penalty \leftarrow 0$
- 31: **end if**
- 32: **end for**
- 33: **Complexity:** $O(N \cdot W \cdot L^2)$ { N =segments, W =window size, L =segment length}

VIII-B Moshaf Attribute Definitions

- **rewaya** (الرواية)
 - Values: - hafs (حفص)
 - Default Value:
 - More Info: The type of the quran Rewaya.
- **recitation_speed** (سرعة التلاوة)
 - Values:
 - * mujawad (مجود)

- * above_murattal (فوق المرتل)
- * murattal (مرتل)
- * hadr (حدر)

- Default Value: murattal (مرتل)
- More Info: The recitation speed sorted from slowest to the fastest الأسرع إلى الأبطأ من مرتبة التلاوة سرعة

- **takbeer** (التكبير)

- Values:
 - * no_takbeer (لا تكبير)
 - * beginning_of_sharh (التكبير لأول الشرح أول من الناس)
 - * end_of_doha (التكبير لآخر الضحى آخر من الناس)
 - * general_takbeer (التكبير لإلا سورة كل أول التكبير)
- Default Value: no_takbeer (لا تكبير)
- More Info: The ways to add takbeer (الله أكبر after Istiaatha (استعاذة) and between end of the surah and beginning of the surah. no_takbeer: "لا تكبير" — No Takbeer (No proclamation of greatness, i.e., there is no Takbeer recitation) beginning_of_sharh: "الناس — Takbeer from the beginning of Surah Ash-Sharh to the beginning of Surah An-Nas end_of_dohaf: "التكبير — Takbeer from the end of Surah Ad-Duha to the end of Surah An-Nas general_takbeer: "التكبير — التوبة" إلا سورة كل أول Takbeer at the beginning of every Surah except Surah At-Tawbah

- **madd_monfasel_len** (مد المنفصل)

- Values:
 - * 2
 - * 3
 - * 4
 - * 5
- Default Value:
- More Info: The length of Mad Al Monfasel "مد المنفصل" for Hafs Rewaya.

- **madd_mottasel_len** (مد المتصل)

- Values:
 - * 4
 - * 5
 - * 6
- Default Value:
- More Info: The length of Mad Al Motasel "مد المتصل" for Hafs.

- **madd_mottasel_waqf** (وقف المتصل المد مقدار)

- Values:
 - * 4
 - * 5
 - * 6
- Default Value:
- More Info: The length of Madd Almotasel at pause for Hafs.. Example "السماء".

- **madd_aared_len** (مد العارض)

- Values:
 - * 2
 - * 4
 - * 6
- Default Value:
- More Info: The length of Mad Al Aared ”مد العارض للسكون”.
- **madd_alleen_len** (اللين) مد (مقدار
 - Values:
 - * 2
 - * 4
 - * 6
 - Default Value: None
 - More Info: The length of the Madd al-Leen when stopping at the end of a word (for a sakin waw or ya preceded by a letter with a fatha) should be less than or equal to the length of Madd al-'Arid (the temporary stretch due to stopping). **Default Value is equal to madd_aared_len.** القوف عن اللين مد مقدار يكون أن ويجب مفتوح) حرف وقبلها الساكنة والياء الساكنة (الواو العارض مع يساوي أو من أقل اللين مد مقدار
- **ghonna_lam_and_raa** (غنة) (الراء) و اللام
 - Values:
 - * ghonna (غنة)
 - * no_ghonna (لا غنة)
 - Default Value: no_ghonna (لا غنة)
 - More Info: The ghonna for merging (Idghaam) noon with Lam and Raa for Hafs.
- **meem_aal_imran** (وصلا) الله {الم تعالى: قوله في عمران آل (ميم)
 - Values:
 - * waqf (وقف)
 - * wasl_2 (مركبتين) ومدها الميم (فتح)
 - * wasl_6 (حركات) ستة ومدها الميم (فتح)
 - Default Value: waqf (وقف)
 - More Info: The ways to recite the word meem Aal Imran (الم الله) at connected recitation. waqf: Pause with a prolonged madd (elongation) of 6 harakat (beats). wasl_2 Pronounce ”meem” with fathah (a short ”a” sound) and stretch it for 2 harakat. wasl_6 Pronounce ”meem” with fathah and stretch it for 6 harakat.
- **madd_yaa_alayn_alharfy** (العين) الحرفي اللازم المد (مقدار
 - Values:
 - * 2
 - * 4
 - * 6
 - Default Value: 6
 - More Info: The length of Lzem Harfy of Yaa in letter Al-Ayen Madd الحرفي ”المد” Maryam ”الشورى” AlShura ”مريم”.
- **saken_before_hamz** (الهمز) قبل (الساكن)
 - Values:
 - * tahqeeq (تحقيق)
- * general_sakt (عام) (سكت)
 - * local_sakt (خاص) (سكت)
- Default Value: tahqeeq (تحقيق)
- More Info: The ways of Hafs for saken before hamz. ”The letter with sukoon before the hamzah And.”(ع) it has three forms: full articulation (tahqeeq), general pause (general_sakt), and specific pause (local_skat).
- **sakt_iwaja** (الكهف) في عوجا عند (السكت)
 - Values:
 - * sakt (سكت)
 - * waqf (وقف)
 - * idraj (إدراج)
 - Default Value: waqf (وقف)
 - More Info: The ways to recite the word ”عوجا” (Iwaja). sakt means slight pause. idraj means not sakt. waqf: means full pause, so we can not determine whether the reciter uses sakt or idraj (no sakt).
- **sakt_marqdena** (يس) في مرقدنا عند (السكت)
 - Values:
 - * sakt (سكت)
 - * waqf (وقف)
 - * idraj (إدراج)
 - Default Value: waqf (وقف)
 - More Info: The ways to recite the word ”مرقدنا” (Marqadana) in Surat Yassen. sakt means slight pause. idraj means not sakt. waqf: means full pause, so we can not determine whether the reciter uses sakt or idraj (no sakt).
- **sakt_man_raq** (القيامه) في راق من عند (السكت)
 - Values:
 - * sakt (سكت)
 - * waqf (وقف)
 - * idraj (إدراج)
 - Default Value: sakt (سكت)
 - More Info: The ways to recite the word ”راق” (Man Raq) in Surat Al Qiyama. sakt means slight pause. idraj means not sakt. waqf: means full pause, so we can not determine whether the reciter uses sakt or idraj (no sakt).
- **sakt_bal_ran** (المطففين) في ران بل عند (السكت)
 - Values:
 - * sakt (سكت)
 - * waqf (وقف)
 - * idraj (إدراج)
 - Default Value: sakt (سكت)
 - More Info: The ways to recite the word ”ران” (Bal Ran) in Surat Al Motaffin. sakt means slight pause. idraj means not sakt. waqf: means full pause, so we can not determine whether the reciter uses sakt or idraj (no sakt).
- **sakt_maleeyah** (بالحاقة) هلك {إماليه تعالى قوله (وجه)

- Values:
 - * sakt (سكت)
 - * waqf (وقف)
 - * idgham (إدغام)
- Default Value: waqf (وقف)
- More Info: The ways to recite the word {هَلِكْ} in Surah Al-Ahqaf. sakt means slight pause. idgham Assimilation of the letter 'Ha' (ه) into the letter 'Ha' (ه) with complete assimilation. waqf: means full pause, so we can not determine whether the reciter uses sakt or idgham.
- **between_anfal_and_tawba** (التوبة) الأنفال بين وجه
 - Values:
 - * waqf (وقف)
 - * sakt (سكت)
 - * wasl (وصل)
 - Default Value: waqf (وقف)
 - More Info: The ways to recite end of Surah Al-Anfal and beginning of Surah At-Tawbah.
- **noon_and_yaseen** (الإدغام) الإظهار عند النون في الإظهار والقلم {ن} و {يس} تعالى:
 - Values:
 - * izhar (إظهار)
 - * idgham (إدغام)
 - Default Value: izhar (إظهار)
 - More Info: Whether to merge noon of both: {يس} and {ن} with "idgham" or not "izhar".
- **yaa_ataan** (بالنمل) {أتان} تعالى قوله في وقفا وحذفها الياء (إثبات)
 - Values:
 - * wasl (وصل)
 - * hadhf (حذف)
 - * ithbat (إثبات)
 - Default Value: wasl (وصل)
 - More Info: The affirmation and omission of the letter 'Yaa' in the pause of the verse {أتاني} in Surah An-Naml. wasl: means connected recitation without pausing as (أتاني). hadhf: means deletion of letter (ي) at pause so recited as (أتان). ithbat: means confirmation reciting letter (ي) at pause as (أتاني).
- **start_with_ism** (الحجرات) سورة في {الاسم} بكلمة البدء وجه
 - Values:
 - * wasl (وصل)
 - * lism (لسم)
 - * alism (ألسم)
 - Default Value: wasl (وصل)
 - More Info: The ruling on starting with the word {الاسم} in Surah Al-Hujurat. lism Recited as (لسم) at the beginning. alism Recited as (ألسم). wasl: means completing recitation without pausing as normal, So Reciting is as (لسم). (بئس)
- **yabsut** (البقرة) ويبسط {يقبض} و {الله تعالى} قوله في والصاد (السين)
 - Values:
 - * seen (سين)
- * saad (صاد)
- Default Value: seen (سين)
- More Info: The ruling on pronouncing seen (س) or saad (ص) in the verse {ويبسط} {يقبض} و {الله تعالى} in Surah Al-Baqarah.
- **bastah** {بسطة} الخلق في {وزادكم تعالى}: قوله في والصاد (السين) بالأعراف
 - Values:
 - * seen (سين)
 - * saad (صاد)
 - Default Value: seen (سين)
 - More Info: The ruling on pronouncing seen (س) or saad (ص) in the verse {وزادكم} {بسطة} الخلق في Surah Al-A'raf.
- **almusaytirun** {المصيطرون} هم {أم تعالى} قوله في والصاد (السين) بالطور
 - Values:
 - * seen (سين)
 - * saad (صاد)
 - Default Value: saad (صاد)
 - More Info: The pronunciation of seen (س) or saad (ص) in the verse {المصيطرون} هم {أم} in Surah At-Tur.
- **bimusaytir** {بمصيطن} عليهم {لست تعالى}: قوله في والصاد (السين) بالغاشية
 - Values:
 - * seen (سين)
 - * saad (صاد)
 - Default Value: saad (صاد)
 - More Info: The pronunciation of seen (س) or saad (ص) in the verse {بمصيطن} عليهم {لست} in Surah Al-Ghashiyah.
- **tasheel_or_madd** {الذكرين} تعالى: قوله في الوصل (همزة) والنمل {بيونس} و {الله} يونس موضعي و {الآن} الأنعام بموضعي
 - Values:
 - * tasheel (تسهيل)
 - * madd (مد)
 - Default Value: madd (مد)
 - More Info: Tasheel of Madd "وَجْعَ" أو التسهيل "وَجْعَ" for 6 words in The Holy Quran: "ءائن". "ءالله". "ءالذكرين".
- **yalhath_dhalik** {ذلك} {يلهث تعالى}: قوله في وعدمه (الإدغام) بالأعراف
 - Values:
 - * izhar (إظهار)
 - * idgham (إدغام)
 - * waqf (وقف)
 - Default Value: idgham (إدغام)
 - More Info: The assimilation (idgham) and non-assimilation (izhar) in the verse {يلهث} {ذلك} in Surah Al-A'raf. waqf: means the reciter has paused on (يلهث)
- **irkab_maana** {بهور} {اركب تعالى}: قوله في والإظهار (الإدغام)
 - Values:

- * izhar (إظهار)
- * idgham (إدغام)
- * waqf (وقف)
- Default Value: idgham (إدغام)
- More Info: The assimilation and clear pronunciation in the verse {اركب معنا} in Surah Hud. This refers to the recitation rules concerning whether the letter "Noon" (ن) is assimilated into the following letter or pronounced clearly when reciting this specific verse. waqf: means the reciter has paused on (اركب)
- **noon_tamanna** {تأمننا لا تعالى قوله في الاختلاس} والروم (الإشمام) يوسف {على
 - Values:
 - * ishmam (إشمام)
 - * rawm (روم)
 - Default Value: ishmam (إشمام)
 - More Info: The nasalization (ishmam) or the slight drawing (rawm) in the verse {لا تأمننا لا يوسف}
- **harakat_daaf** {ضعف} تعالى قوله في ضم) أو (فتح الضاد حركة)
 - Values:
 - * fath (فتح)
 - * dam (ضم)
 - Default Value: fath (فتح)
 - More Info: The vowel movement of the letter 'Dhad' (ض) (whether with fath or dam) in the word {ضعف} in Surah Ar-Rum.
- **alif_salasila** {سلاسل} تعالى: قوله في وقفا وحذفها الألف (إثبات) الإنسان بسورة
 - Values:
 - * hadhf (حذف)
 - * ithbat (إثبات)
 - * wasl (وصل)
 - Default Value: wasl (وصل)
 - More Info: Affirmation and omission of the 'Alif' when pausing in the verse {سلاسل} in Surah Al-Insan. This refers to the recitation rule regarding whether the final "Alif" in the word "سلاسل" is pronounced (affirmed) or omitted when pausing (waqf) at this word during recitation in the specific verse from Surah Al-Insan. hadhf: means to remove alif (ا) during pause as (سلاسل) ithbat: means to recite alif (ا) during pause as (سلاسل) wasl means completing the recitation as normal without pausing, so recite it as (سلاسل) وأغلا
- **idgham_nakhluqum** أو ناقصا إدغاما الكاف في القاف (إدغام) بالمرسلات {نخلقكم} كاملا
 - Values:
 - * idgham_kamil (إدغام كامل)
 - * idgham_naqis (إدغام ناقص)
 - Default Value: idgham_kamil (إدغام كامل)
 - More Info: Assimilation of the letter 'Qaf' into the letter 'Kaf,' whether incomplete (idgham_naqis) or complete (idgham_kamil), in the verse {نخلقكم} in Surah Al-Mursalat.
- **raa_firq** {وصلا الشعراء في {فرق} راء في والترقيق (التفخيم)
 - Values:
 - * waqf (وقف)
 - * tafkheem (تفخيم)
 - * tarqeeq (ترقيق)
 - Default Value: tafkheem (تفخيم)
 - More Info: Emphasis and softening of the letter 'Ra' in the word {فرق} in Surah Ash-Shu'ara' when connected (wasl). This refers to the recitation rules concerning whether the letter "Ra" (ر) in the word "فرق" is pronounced with emphasis (tafkheem) or softening (tarqeeq) when reciting the specific verse from Surah Ash-Shu'ara' in connected speech. waqf: means pausing so we only have one way (tafkheem of Raa)
- **raa_alqitr** {وقفا سبأ في {القطر} راء في والترقيق (التفخيم)
 - Values:
 - * wasl (وصل)
 - * tafkheem (تفخيم)
 - * tarqeeq (ترقيق)
 - Default Value: wasl (وصل)
 - More Info: Emphasis and softening of the letter 'Ra' in the word {القطر} in Surah Saba' when pausing (waqf). This refers to the recitation rules regarding whether the letter "Ra" (ر) in the word "القطر" is pronounced with emphasis (tafkheem) or softening (tarqeeq) when pausing at this word in Surah Saba'. wasl: means not pausing so we only have one way (tarqeeq of Raa)
- **raa_misr** {وموضعي يونس في {مصر} راء في والترقيق (التفخيم)
 - Values:
 - * wasl (وصل)
 - * tafkheem (تفخيم)
 - * tarqeeq (ترقيق)
 - Default Value: wasl (وصل)
 - More Info: Emphasis and softening of the letter 'Ra' in the word {مصر} in Surah Yunus, and in the locations of Surah Yusuf and Surah Az-Zukhruf when pausing (waqf). This refers to the recitation rules regarding whether the letter "Ra" (ر) in the word "مصر" is pronounced with emphasis (tafkheem) or softening (tarqeeq) at the specific pauses in these Surahs. wasl: means not pausing so we only have one way (tafkheem of Raa)
- **raa_nudhur** {وقفا بالقمر {نذر} راء في والترقيق (التفخيم)
 - Values:
 - * wasl (وصل)
 - * tafkheem (تفخيم)
 - * tarqeeq (ترقيق)
 - Default Value: tafkheem (تفخيم)

- More Info: Emphasis and softening of the letter 'Ra' in the word {نذر} in Surah Al-Qamar when pausing (waqf). This refers to the recitation rules regarding whether the letter "Ra" (ر) in the word "نذر" is pronounced with emphasis (tafkheem) or softening (tarqeeq) when pausing at this word in Surah Al-Qamar. wasl: means not pausing so we only have one way (tarqeeq of Raa)
- **raa_yasr** بطه أسر { و أن بالفجر {يسر} راء في والترقيق (التخيم) وقفاً والدخان والحجر بهود وإفسر { والشعراء
 - Values:
 - * wasl (وصل)
 - * tafkheem (تخيم)
 - * tarqeeq (ترقيق)
 - Default Value: tarqeeq (ترقيق)
 - More Info: Emphasis and softening of the letter 'Ra' in the word {يسر} in Surah Al-Fajr when pausing (waqf). This refers to the recitation rules regarding whether the letter "Ra" (ر) in the word "يسر" is pronounced with emphasis (tafkheem) or softening (tarqeeq) when pausing at this word in Surah Al-Fajr. wasl: means not pausing so we only have one way (tarqeeq of Raa)
- **meem_mokhfah** مدغمة) أو مخفأة الميم (هل
 - Values:
 - * meem (ميم)
 - * ikhfah (إخفاء)
 - Default Value: ikhfah (إخفاء)
 - More Info: This is not a **standard** Hafs way but a disagreement between **scholars** in our century on how to **pronounce Ikhfa** for meem. Some **scholars** do full merging (إدغام) and the others open the **lips** a little bit (إخفاء). We did not want to add this, but some of the best reciters disagree about this.