



Quranic Speech Recognition: Studio vs. Phone Microphone

Comparative analysis of ASR performance, noise robustness, and architectural choices for the Tarsikh application

Author: Tarsikh Project

Benchmark: Quran-Lab (600 clips)

Model: Zipformer 65M INT8

DOI : [10.5281/zenodo.21770922](https://doi.org/10.5281/zenodo.21770922)

📌 Executive Summary

This document presents our study on ASR error modeling and details the creation of our phonetic confusion matrix in JSON format, designed to optimize real-time recitation tracking in the Tarsikh application. We explain the algorithmic methodology used to generate the global confusion matrix based on the official open-source benchmark [Quran-Lab](#). We also detail the engineering philosophy of the **Tarsikh** application, which relies on real-time phonetic alignment and manual user correction.

1. THE ROLE OF ASR IN TARSIKH: READING TRACKING VS. AUTO-CORRECTION

It is essential to clarify a technical reality: **no embedded (on-device, offline) ASR model is accurate enough today to perform automatic Tajweed correction without generating false positives.**

Standard ASR models (with an integrated Language Model) tend to "auto-correct" the reciter's hesitations and mask mistakes. In **Tarsikh**, ASR therefore does not act as an automatic corrector:

- **ASR acts as an aligner/tracker:** The model listens to the voice to scroll the text in real time, respond to pauses, and synchronize recitation progress.
- **Correction is manual:** Users themselves mark the verses or words where they hesitated or made a mistake. This choice guarantees zero false alarms and empowers the learner.

2. OFFICIAL BENCHMARK & MODEL PERFORMANCE

The results below are based on the official [Quranic ASR Benchmark](#) dataset (600 verified clips without data leakage).

MODEL	TYPE	MODE	WER (ERROR RATE)
FastConformer Quran	Offline	Heavy	4.13 %
🏆 Zipformer Phoneme (Tarsikh Model)	On-Device	● Streaming	5.83 %
Tarteel (Official)	Cloud API	Streaming	10.99 %
Whisper-large-v3	Offline	Heavy	11.65 %
Google Chirp 3	Cloud API	Streaming	13.56 %

The model used in Tarsikh, the [Zipformer Phoneme 65M](#), achieves the lowest Word Error Rate (WER) among the evaluated streaming architectures on this benchmark, outperforming tested commercial cloud solutions.

3. THE REAL-WORLD TEST: STUDIO VS. PHONE MICROPHONE

As highlighted by the benchmark creators: *“tlog is the one that matters for apps; studio numbers flatter every model.”*

For the Tarsikh application, we chose to generate a production confusion matrix based on deliberately degraded recordings (fan noise, echo, background radio). Since fricative consonants (خ, ث) are particularly sensitive to white noise, this approach ensures smooth tracking even under difficult acoustic conditions.

4. METHODOLOGY & ALIGNMENT ALGORITHM

To generate the global confusion matrix, we cross-reference the canonical phonetic transcription of the Quran with the AI predictions on the audio clips:

- **Official Transcription:** Reference target phonemes come directly from the canonical Hafs dictionary provided in the open-source repository [Muno459/zipformer_p-quran](#) (`ordered_quran_phonemes.json`).
- **Audio Processing & Inference:** The 200 smartphone-recorded audio clips (`tlog_holdout`) from the benchmark are converted to standard 16 kHz Mono WAV format (via FFmpeg) and processed by the Sherpa-ONNX Zipformer engine (professional studio recitations were intentionally excluded to match real-world usage).
- **Levenshtein Alignment:** The `alignSequences` function compares the target sequence with the ASR prediction to count matches, substitutions, and omissions (<OMITTED>):

```
// Dynamic programming alignment algorithm (Levenshtein)
function alignSequences(targetTokens, predictedTokens) {
  const m = targetTokens.length, n = predictedTokens.length;
  const dp = Array.from({ length: m + 1 }, () => Array(n + 1).fill(0));

  for (let i = 0; i <= m; i++) dp[i][0] = i;
  for (let j = 0; j <= n; j++) dp[0][j] = j;

  for (let i = 1; i <= m; i++) {
    for (let j = 1; j <= n; j++) {
      const cost = targetTokens[i - 1] === predictedTokens[j - 1] ? 0 : 1;
      dp[i][j] = Math.min(
        dp[i - 1][j] + 1,
        dp[i][j - 1] + 1,
        dp[i - 1][j - 1] + cost
      );
    }
  }

  let i = m, j = n;
  const alignments = [];
  while (i > 0 || j > 0) {
    if (i > 0 && j > 0 && dp[i][j] === dp[i - 1][j - 1] + (targetTokens[i - 1] === predictedTokens[j - 1] ? 0 : 1)) {
      alignments.push({ target: targetTokens[i - 1], predicted: predictedTokens[j - 1] });
      i--; j--;
    } else if (i > 0 && dp[i][j] === dp[i - 1][j] + 1) {
      alignments.push({ target: targetTokens[i - 1], predicted: "<OMITTED>" });
      i--;
    } else {
      alignments.push({ target: "<INSERTED>", predicted: predictedTokens[j - 1] });
      j--;
    }
  }
  return alignments.reverse();
}
```

5. GLOBAL CONFUSION MATRIX (FULL JSON)

⚠ Note on Usage in Tarsikh

The JSON below is the matrix calculated exclusively from real smartphone recordings (tlog_holdout) from the Hugging Face benchmark. **The production version embedded in the Tarsikh application uses its own matrix derived from an even more degraded benchmark (fan noise, echo)** to ensure maximum tolerance against ambient noise.

Full JSON dictionary:

```
{
  "و": {
    "ظ": 0, "ث": 0, "ف": 0, "ن": 0, "ه": 0, "ل": 0.01, "ا": 0, "ر": 0, "ي": 0.01, "م": 0.09, "": 0.86, "و": 1
  },
  "ا": {
    "": 0.92, "ر": 0.01, "ي": 0.05, "ف": 0, "ه": 0, "ع": 0, "ن": 0, "د": 0, "و": 0, "ب": 0, "ج": 0.01, "ث": 0, "ل": 0.01, "ا": 0, "ن": 0, "ء": 0
  },
  "ء": {
    "ر": 0, "ف": 0.01, "م": 0.01, "ا": 0.01, "ي": 0.01, "ت": 0.01, "و": 0.01, "ع": 0.02, "ه": 0, "ب": 0.06, "": 0.84, "ء": 1
  },
  "ن": {
    "ب": 0, "خ": 0, "ر": 0, "ل": 0.01, "ك": 0,
    "و": {
      "ر": 0, "ذ": 0, "ي": 0, "ث": 0, "م": 0, "": 0, "ق": 0, "ب": 0, "ث": 0, "ء": 0, "ل": 0.1, "": 0.86, "ن": 1
    },
    "ه": {
      "ا": 0, "ن": 0.01, "ل": 0, "ق": 0.01, "ء": 0.01, "ر": 0, "ي": 0.01, "ح": 0.01, "و": 0.08, "": 0.86, "ه": 1
    },
    "ث": {
      "ث": 0.89, "": 0.03, "ي": 0.06, "و": 0, "ا": 0, "و": 0, "ن": 0, "ب": 0.01, "ر": 0, "ج": 0, "م": 0, "ل": 0
    },
    "ر": {
      "ع": 0, "ث": 0, "ه": 0, "ح": 0, "": 0.01, "و": 0.01, "م": 0.01, "ا": 0.01, "ب": 0.59, "ث": 0.37, "": 1
    },
    "ل": {
      "ء": 0, "ا": 0, "ر": 0.01, "م": 0, "ج": 0, "ب": 0.02, "ن": 0, "ي": 0, "ذ": 0.04, "": 0.92, "ل": 1
    },
    "م": {
      "ه": 0, "ب": 0, "ن": 0, "ل": 0.01, "و": 0.01, "ب": 0, "ن": 0, "ر": 0.04, "": 0.93, "م": 1
    },
    "ا": {
      "ه": 0, "ث": 0, "س": 0, "ر": 0, "ن": 0, "ل": 0, "ء": 0, "ع": 0, "": 0.13, "": 0.85, "ا": 1
    },
    "ق": {
      "ظ": 0.01, "ك": 0.01, "ض": 0.01, "ت": 0.01, "و": 0.01, "ث": 0.01, "ظ": 0.03, "خ": 0.06, "": 0.01, "د": 0.82, "ق": 1
    },
    "س": 0.01, "ا": 1
  },
  "ع": {
    "ر": 0, "ه": 0.01, "ا": 0, "د": 0, "ل": 0.02, "ح": 0, "ت": 0.03, "ء": 0, "ر": 0.1, "": 0.8, "ع": 1
  },
  "ب": {
    "ر": 0, "ض": 0.04, "": 0, "ت": 0.01, "و": 0.93, "ب": 0.01, "م": 1
  },
  "ج": {
    "ث": 0.01, "ر": 0.01, "": 0.15, "": 0.01, "ت": 0.78, "ج": 0.05, "": 1
  },
  "د": {
    "ه": 0.01, "ذ": 0.01, "ج": 0.02, "ت": 0.01, "ن": 0.01, "ا": 0.01, "ف": 0.11, "": 0.81, "د": 0.01, "ن": 1
  },
  "ر": {
    "ر": 0.94, "": 0.04, "و": 0, "ه": 0, "ا": 0, "ث": 0, "ل": 0, "ن": 0, "م": 0, "ي": 0.01, "": 0, "ج": 1
  },
  "ي": {
    "ا": 0, "ل": 0, "ء": 0, "ر": 0, "ن": 0.01, "ب": 0, "ج": 0.05, "": 0.92, "ي": 1
  },
  "ك": {
    "ب": 0, "ء": 0, "ه": 0.05, "": 0, "ت": 0.01, "ق": 0.92, "ك": 1
  },
  "ذ": {
    "ه": 0, "د": 0, "م": 0.01, "ر": 0, "ن": 0.32, "": 0.66, "ذ": 1
  },
  "ث": {
    "س": 0, "ك": 0, "م": 0.01, "ج": 0, "ا": 0, "ر": 0, "ف": 0.05, "": 0.92, "ث": 1
  },
  "ر": {
    "م": 0.01, "ل": 0, "ن": 0, "و": 0, "ب": 0.01, "ث": 0, "ج": 0.01, "ع": 0, "ك": 0.12, "": 0.02, "ا": 0.82, "ر": 1
  },
  "ذ": {
    "ظ": 0.01, "ل": 0.01, "ر": 0.04, "د": 0.05, "": 0.9, "ذ": 1
  },
  "م": {
    "": 0.05, "": 0.05, "م": 0.9, "م": 1
  },
  "ن": {
    "": 0.01, "ث": 0.01, "ب": 0.01, "ل": 0.01, "ا": 0.01, "": 0, "ت": 0.02, "م": 0.01, "ن": 0.09, "": 0.84, "ن": 1
  },
  "س": {
    "": 0.01, "ض": 0.03, "": 0.01, "ف": 0.95, "س": 1
  },
  "ج": {
    "": 0.01, "و": 0.03, "ك": 0.01, "ز": 0.01, "س": 0.04, "ش": 0.07, "": 0.81, "ج": 1
  },
  "ه": {
    "": 0.01, "ر": 0.01, "ع": 0.02, "خ": 0.04, "": 0.85, "ح": 0.07, "ه": 1
  },
  "ص": {
    "": 0.8, "": 0.11, "": 0.09, "ص": 1
  },
  "خ": {
    "": 0.02, "ق": 0.02, "ز": 0.02, "ف": 0.07, "ح": 0.87, "خ": 1
  },
  "ف": {
    "": 0.01, "ه": 0.01, "س": 0.01, "ذ": 0.01, "و": 0.01, "ص": 0.08, "": 0.01, "ث": 0.01, "ق": 0.87, "ف": 1
  },
  "ظ": {
    "": 0.03, "ظ": 0.13, "": 0.03, "د": 0.81, "ظ": 1
  },
  "ز": {
    "": 0.04, "د": 0.04, "ج": 0.04, "": 0.04, "ب": 0.04, "ض": 0.04, "م": 0.04, "غ": 0.7, "ز": 1
  },
  "ش": {
    "": 0.02, "ح": 0.04, "": 0.02, "ص": 0.93, "ش": 1
  },
  "غ": {
    "": 0.05, "ب": 0.05, "": 0.05, "ق": 0.86, "غ": 1
  },
  "ث": {
    "": 0.03, "ص": 0.03, "ق": 0.03, "ص": 0.03, "ف": 0.06, "ت": 0.03, "ر": 0.79, "ث": 1
  },
  "ض": {
    "": 0.03, "": 0.03, "خ": 0.07, "ج": 0.03, "ك": 0.1, "د": 0.7, "ض": 0.03, "ظ": 1
  },
  "ز": {
    "": 0.04, "ل": 0.02, "ب": 0.02, "ص": 0.02, "ر": 0.02, "ي": 0.02, "ذ": 0.02, "": 0.82, "ز": 1
  }
}
```

💡 Matrix Reading Guide:

Each entry follows the structure: "Target_Phoneme" : { "Predicted_Phoneme" : Probability }

- **Main key (e.g., "ث"):** The actual Quranic phoneme that was supposed to be spoken.
- **Internal keys (e.g., "ت", "ث", "<OMITTED>"):** What the ASR model actually predicted.
- **Probability (e.g., 0.79 = 79%):** The observed relative frequency.

Interpretation example for "ث": { "0.06:"ت", 0.79:"ث", "<OMITTED>":0.03}:

When the reciter pronounces a ث (Tha), the AI correctly recognizes it **79%** of the time, confuses it with a ت (Ta) **6%** of the time, and omits it entirely (<OMITTED>) **3%** of the time.

6. OPEN-SOURCE RESOURCES & LINKS

To view or reproduce this work:

- **Benchmark Dataset:** [Quran-Lab/quranic-asr-benchmark](https://quran-lab.github.io/quranic-asr-benchmark/)
- **Streaming ASR Model:** [Muno459/zipformer_p-quran](https://github.com/Muno459/zipformer_p-quran)

HOW TO CITE THIS WORK

If you use this benchmark analysis, phonetic confusion matrix, or alignment methodology in your research, please cite this technical report as follows:

```
@techreport{tarsikh2026asr,
  title={Quranic Speech Recognition: Studio vs. Phone Microphone \& Phonetic Error Modeling for Tarsikh},
  author={Tarsikh Project},
  year={2026},
  institution={Tarsikh},
  doi={10.5281/zenodo.21770922},
  url={https://doi.org/10.5281/zenodo.21770922}
}
```