Supplementary Material

Overview

This supplementary document provides additional details that complement the main paper. As referenced in Section 2.2 of our work, it begins by presenting a comprehensive breakdown of the labeled datasets used for fine-tuning, with specific train and validation splits for Urdu, Persian, and Arabic. Furthermore, this document elaborates on our SentencePiece tokenization strategy, as discussed in Section 3.2 of the main paper. It provides details on its integration within the Fairseq framework and presents empirical results comparing its Word Error Rate (WER) performance against a character-based approach.

1 Labeled Data Distribution Across Datasets

Most existing studies evaluating Automatic Speech Recognition (ASR) models focus on assessing performance using a single dataset. For instance, models trained on the Common Voice dataset are typically evaluated only on the same dataset. However, when these models are tested on other datasets without fine-tuning, their performance often proves suboptimal, highlighting a lack of generalizability.

To address this limitation and improve the robustness of ASR models across diverse datasets, we adopted a multi-dataset approach for training and evaluation. For each target language, we combined a fraction of data from multiple datasets, including Common Voice, to create comprehensive training and testing splits. This approach ensures that models are exposed to a wider variety of speech patterns, accents, and recording conditions, thereby enhancing their generalizability.

Below, we provide the exact durations of the datasets used for each language, along with the specific train and test splits:

Urdu Dataset	**Train Duration (hours)**	**Validation Duration (hours)**
Common Voice	4.23	0.73
SME_news	7.48	1.28
Tiny Urdu Speech Corpus	5.41	0.95
Indic Voices	42.9	7.70

Table 1: Urdu dataset splits for fine-tuning.

Persian Dataset	**Train Duration (hours)**	**Validation Duration (hours)**
Common Voice	36.5	6.44
Persian Speech Corpus	2.28	0.21
My Audio Tiny	2.25	0.34
TTS Female	22.6	4.04
Moradi	0.83	0.13
ParsiGOO	3.56	0.64

Table 2: Persian dataset splits for fine-tuning.

Arabic Dataset	**Train Duration (hours)**	**Validation Duration (hours)**
Common Voice	27.5	4.89
SLR-108	8.5	1.53
MGB	37.0	5.11

Table 3: Arabic dataset splits for fine-tuning.

2 SentencePiece Tokenization

To improve the tokenization process in our speech recognition pipeline, we incorporated SentencePiece tokenization within the Fairseq framework. We first trained a Byte-Pair Encoding (BPE) SentencePiece model with a vocabulary size of 512, using the transcriptions from the training dataset. This vocabulary was subsequently utilized to initialize the Connectionist Temporal Classification (CTC) layer of the wav2vec model.

2.1 Integration of SentencePiece in Fairseq

In the default character-based tokenization used in Fairseq, sentences are split into individual characters. To integrate SentencePiece, we modified this process by first segmenting the sentence into words. Instead of directly splitting words at the character level, we applied SentencePiece encoding to tokenize each word into subword units. This approach retains meaningful subword structures while reducing the token sequence length.

During inference, the decoded tokens were grouped at the word level, and each set of tokens was converted back into corresponding words. Finally, the words were concatenated to reconstruct the complete transcription.

2.2 Impact on Word Error Rate (WER)

To evaluate the impact of SentencePiece tokenization, we fine-tuned our CP2 model using both the default character-based tokenization and the proposed SentencePiece-based approach. The WER comparison across Urdu, Persian, and Arabic is summarized in Table 4.

Model	**Urdu**	**Persian**	**Arabic**
CP2 (Character-based)	25.8	26.2	39.0
CP2 (SentencePiece-based)	20.6	17.1	32.9

Table 4: Word Error Rate (WER) comparison between character-based and SentencePiece-based tokenization.

The results demonstrate a significant reduction in WER across all three languages, highlighting the effectiveness of subword-level tokenization over character-based tokenization in CTC-based speech recognition.