# SeamlessM4T × FLEURS Evaluation Pipeline

## Overview

This code builds a complete evaluation pipeline for testing Meta’s SeamlessM4T-v2 model on the FLEURS dataset for multiple Indian languages. It performs speech and text translation tasks, saves intermediate results (text predictions and audio outputs), and computes standard translation/ASR metrics.

## Supported Languages

- Full tasks (all 4 tasks: S2TT, T2TT, S2ST, T2ST): Telugu (tel), Urdu (urd)  
- Partial tasks (text-only tasks: S2TT, T2TT): Tamil (tam), Odia (ory)

## Pipeline Tasks

For each parallel sentence pair in FLEURS, the following tasks are run:

* Speech-to-Text Translation (S2TT): Input Hindi speech (16kHz resampled) → Output target language text
* Text-to-Text Translation (T2TT): Input Hindi text → Output target language text
* Speech-to-Speech Translation (S2ST) (full tasks only): Input Hindi speech → Output target language speech (.wav), decoded with ASR for evaluation
* Text-to-Speech Translation (T2ST) (full tasks only): Input Hindi text → Output target language speech (.wav), decoded with ASR for evaluation

## Outputs

CSV Files per language containing:

* Source text (Hindi transcription)
* Reference target text
* Predictions from each task (S2TT, T2TT, S2ST-ASR, T2ST-ASR)

Audio Files saved in task-specific folders (input\_audios/, s2s\_outputs/, t2s\_outputs/).

## Metrics Computed

For each language, the following metrics are reported:

* S2TT & T2TT (Text outputs): SacreBLEU, chrF2++, Word Error Rate (WER)
* S2ST & T2ST (Speech outputs, evaluated via ASR): SacreBLEU, WER

## Execution Flow

1. Load source (Hindi) and target language test sets from FLEURS.
2. Align parallel sentences using id.
3. Run SeamlessM4T inference for the required tasks.
4. Save outputs (CSV + audio).
5. Compute and print metrics.

In short: Your code evaluates SeamlessM4T translations for Hindi → {Telugu, Urdu, Tamil, Odia}, runs 4 speech/text translation tasks, saves outputs, and computes BLEU, chrF2++, and WER metrics.