Task Assignment | AI Researcher Intern- Speech & Audio | Josh Talks

Question-1

Background
You are provided with ~10 hours of Hindi ASR training data (here) in the format shown below (audio + transcription metadata)

Important Note: The Url's mentioned above in the the question and further questions might not work, PFB the instructions for modifying the same

Instructions to access the data :this is the example of a new transcription URL https://storage.googleapis.com/upload_goai/967179/825780_transcription.json, the recording and metadata follows the same format, please modify the other URL's while processing the data

Dataset Schema Description

- **user_id** – Identifier for the speaker/user associated with the audio (anonymized).
- **recording_id** – Unique identifier for the specific audio recording within the dataset.
- **language** – Language label of the audio (e.g., "hi" for Hindi).
- **duration** – Duration of the audio recording (in seconds). Useful for filtering or batching.
- **rec_url_gcp** – URL link to the raw audio file stored on cloud (e.g., Google Cloud Storage). This is the main audio input for training/evaluation.
- **transcription_url** – URL to the ground-truth transcription text corresponding to the audio file. This is the label to be used for fine-tuning.
- **metadata_url** – URL to additional metadata about the recording (may include device type, noise level, accents, or collection conditions). Optional for training, but can help in analysis.

Your Task

a) Preprocess the dataset and share what you did to process the data and make it ready for training.
b) Fine-tune Whisper-small on this dataset and evaluate both the pretrained Whisper-small baseline and your fine-tuned model on the Hindi portion of the FLEURS test dataset.
c) Report the Word Error Rate (WER) in a structured table format. Here

Question-2

Background
In real-world conversational speech, speakers often produce **disfluencies** such as **fillers** ("uh", "umm"), repetitions ("I… I was saying"), **false starts**, **prolongations** ("sooooo"), and **hesitations**. These elements are natural in human communication but present challenges for speech recognition systems, spoken dialogue systems, and speech-to-speech translation.

For robust ASR and speech AI, it is critical to **identify and segment these disfluencies** so they can be modeled explicitly (for realism) or filtered out (for cleaner transcripts). This task evaluates your ability to work with speech data not only for recognition, but also for linguistic signal processing and dataset curation.

Your Task
You are provided with the same 10 hours of Hindi audio dataset (audio + transcription metadata). Your goal is to identify speech disfluencies from the audio and create a structured dataset of segmented disfluency clips.

Here is the list of target speech disfluencies (e.g., fillers like “uh”, “umm”, repetitions, false starts, prolongations, hesitations).

Using this list

a) Each recording has multiple segments or utterances on the basis of which they have been transcribed. Analyze the 10-hour dataset and detect which segments of each recording have occurrences of the target disfluencies.
b) For each segment where a disfluency is detected, clip the audio as per the time stamp of that segment from that complete recording.
c) Create a **structured sheet** (CSV/Google Sheet) that records each detected disfluency , link to the segmented audio (not of the full recording), and the recording id the segment belongs to.

Deliverables

a) Methodology Summary:
- How you detected the disfluencies. (Short answer)
- How you clipped the audio segment from the complete recording. (Short answer)
- Any preprocessing/normalization applied. (Short answer)

b) Output Dataset (Sheet Format)
Your output should contain each row which should represent one disfluency occurrence, with the schema as mentioned in the sheet format

c) Segmented Audio Files
Upload audio cuts corresponding to each row in the sheet (short clips containing only the disfluency occurrence segment).

Question-3

Background
In a subset of our Hindi conversational dataset, which was human transcribed we have identified **~1,77,000 unique** words (here). Some of these words are obvious spelling mistakes.

Our goal is to improve transcription accuracy of our dataset. One proposed approach is to separate these words into two groups:

- Words that have 100% accurate spelling
- Words that are incorrect because they contain spelling mistakes

The idea is that once we identify the words with errors, we can go back to the corresponding audio segments and selectively re-human transcribe only those segments, rather than redoing the entire dataset.

Keep in mind that in our transcription guidelines: **English words spoken in the conversation are transcribed in Devanagari script.** For example, “computer” spoken in English should appear as “कंप्यूटर.” In such cases, the Devanagari transcription counts as the correct spelling, not an error.

Your Task
Identify which of the 1,75,000 words are correctly spelled vs. incorrectly spelled. Share the approach you undertook to come to this conclusion.

Deliverables

a. Share the final number of unique correct spelled words in the dataset
b. A google sheet containing 2 columns, one with the the final list of unique words and second making them as ‘correct spelling’ and ‘incorrect spelling’

We are looking for your ability to combine **linguistic reasoning with practical data-cleaning strategies** that balance accuracy and efficiency.

Important Note: Instructions to access the data :this is the example of a new transcription URL https://storage.googleapis.com/upload_goai/967179/825780_transcription.json, the recording and metadata follows the same format, please modify the other URL's while processing the data

Question - 4) Here, you are given transcriptions from five ASR models for the same audio and a human reference, which may contain errors.

- Design an approach (theory + pseudocode/code) to:
- Construct a lattice that captures all valid transcription alternatives from the model outputs.
- Handle insertions, deletions, and substitutions in a way that does not unfairly penalize models when the reference is wrong.
- Decide when to trust model agreement over the reference.

Choose and justify the alignment unit (word / subword / phrase). Then compute WER for each model using lattice based transcription and model output

Your method should reduce WER for models that were unfairly penalized and keep it unchanged for the others.