A Solution to the Acoustic Track of PREPARE Challenge

Wei Dong
Ann Abor Algorithms Inc.
wdong@aaalgo.com

Yuanfang Guan
University of Michigan
gyuanfan@umich.edu

1. Introduction

1.1 Overview of Solution

Our submission tackles the task of detecting Alzheimer's disease and related dementias (AD/ADRD) from short audio recordings. We leverage OpenAI's pretrained Whisper model [1] as a powerful speech feature extractor, keeping only its encoder and replacing the original decoder with a custom classification head. This allows us to capture rich acoustic information from each audio snippet while focusing on the critical features relevant to cognitive decline.

To handle input of potentially variable lengths, we append a small number of trainable "prediction tokens" to each audio sequence before it passes through Whisper's encoder. These special tokens learn to selectively attend to relevant parts of the audio, pooling the most salient information for the downstream classifier. As a result, our model can adapt to different speaking styles, durations, and acoustic conditions without manually segmenting or normalizing the input.

The source code of our solution is shared on github.[1]

1.2 Dataset Background

The competition dataset includes short (≤30 second) spoken clips of individuals with one of three cognitive statuses: **Control**, **MCI** (mild cognitive impairment), or **ADRD** (advanced dementia diagnoses such as AD or primary progressive aphasia). Each sample is associated with demographic attributes (age, sex), and the raw audio. Our solution did not make use of the eGeMAPS-based acoustic features provided by the competition. The accuracy was measured with the multi-class **log-loss** metric.

[1] Source code of our solution: https://github.com/aaalgo/PREPARE/

1.3 Leaderboard Results

Our final submission achieved a **public leaderboard score** of **0.6697** and a **private leaderboard score** of **0.6610**, placing our team in **8th** position. These results highlight the effectiveness of repurposing a pretrained speech recognition model for a specialized classification task in the medical domain.

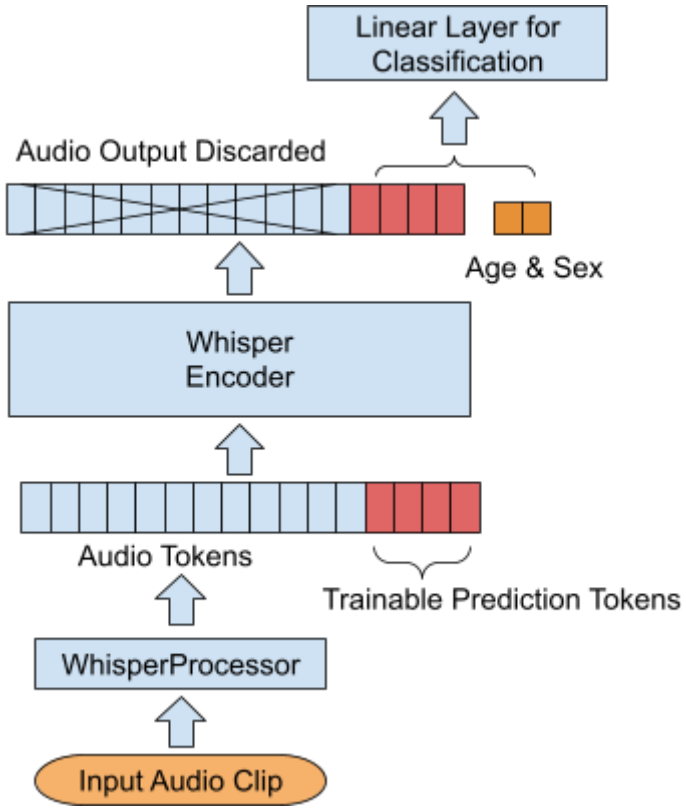

Figure 1. Model architecture illustrated. We make use of the Whisper encoder for feature extraction. The key modification is to append K trainable prediction tokens to the audio tokens extracted from the input audio clip by the WhisperProcessor, and only use the output of these prediction tokens for final classification.

2. Methodology

2.1 Model Architecture & Core Ideas

Whisper Encoder as a Feature Extractor

Whisper follows an **encoder-decoder architecture,** a.k.a. the **transformer**, where an **encoder** transforms the input audio (converted into a log-mel spectrogram) into a latent representation, and a **decoder** then uses cross-attention to generate text tokens in an autoregressive manner.

In Whisper's intended use case—speech recognition—the encoder distills acoustic information while the decoder generates the detected text.

For **dementia classification**, however, we do not need to generate text. Instead, we only require a powerful representation of the audio signal that captures relevant cognitive and acoustic cues. By **removing the decoder** and focusing solely on the **pretrained encoder**, we obtain a feature extractor that has already learned robust speech features across diverse data. These embeddings are then fed into a custom classification head to predict cognitive status, effectively repurposing Whisper's learned speech representations for a medical diagnostic task.

Prediction Tokens

To extract relevant features from an **input of potentially variable lengths**, we introduce a set of K (default: 4) **prediction tokens with trainable embedding**, which are appended to the encoder's input before it processes the sequence. In a typical Transformer-based encoder, there is a **one-to-one correspondence** between input tokens and output features: each input token passes through multiple layers of self-attention and feed-forward networks, resulting in a corresponding output embedding at the same position.

By appending these extra trainable tokens, we allow them to learn how to "query" the relevant segments of the audio. During the self-attention operation, the prediction tokens attend to all other input tokens (representing the audio frames), capturing salient acoustic patterns. After all the encoder layers, we obtain one output embedding for each input token, including these appended tokens.

Since our goal is to classify cognitive status, we only **retain the final output features** of the trainable prediction tokens. These output embeddings naturally pool crucial diagnostic information from the entire audio sequence, funneling it into a few concise vectors. Finally, a **classification head** (e.g., a dense layer) projects these pooled embeddings into probabilities for **Control**, **MCI**, and **ADRD** diagnoses.

2.2 Other Implementation Details

Usage of Demographic Features

We incorporate the patient's **age** and **sex** by concatenating these two scalars to the final encoder outputs of the prediction tokens.

Ensemble of Models

To enhance robustness, we train **10 independent models** with different random seeds and average their predicted probabilities at inference.

Audio Preprocessing

Resampling: Each raw audio clip (48 kHz, ≤30 seconds) is resampled to **16 kHz**, matching Whisper's training.

Mel-Spectrogram: The WhisperProcessor converts each resampled clip into a sequence of **3,000 frames**, each an 80-dimensional vector.

Offline Storage: These frame sequences are saved in a pickle file, which is then used for both training and inference, speeding up experimentation.

Data Augmentation

During training, each of the 3,000 frames is **scaled by a random factor in [0.95,1.05]**. This simple augmentation introduces slight variations in amplitude and helps the model learn more robust representations of speech signals.

Custom Forward Pass

Because the original Whisper code does not natively support injecting custom tokens, we implemented a **modified forward function** for the encoder. The prediction tokens are appended to the input embeddings, passed through the self-attention layers, and their final hidden states are then used for classification.

Model Versions

We evaluated various **multilingual Whisper** releases and determined that the **"medium"** model offered the best balance of accuracy and inference speed. Interestingly, the **"large"** variant showed lower performance for this particular dataset, possibly due to overfitting.

3. Alternative Experiments Without Accuracy Improvement

3.1 Demographic-Based Prediction Tokens

We hypothesized that encoding demographic information (age and sex) directly into prediction tokens might make the model more sensitive to potential risk factors. Instead of using trainable embedding vectors for each prediction token, we initialized them based on the demographic

data. We use a small trainable MLP network to convert the one-hot embedding of age and sex to the 80-dimensional vector that is compatible with the audio features. The idea was that these tokens would then learn to attend to portions of the audio most relevant for each demographic profile.

However, this more elaborate demographic-based token initialization did not improve overall accuracy or reduce log-loss. The simpler method—concatenating demographic features at the encoder output—proved just as effective.

3.2 Aggressive Augmentation

We also experimented with an aggressive form of data augmentation during training, where we randomly selected consecutive 2,000 frames out of the full 3,000-frame sequence. The rationale was that 20 out of the 30 seconds recording might be enough for making predictions. Because Whisper works with a fixed input shape, we are forced to use the input size of 2,000 at inference time. This allows us to do inference-time augmentation by applying a sliding window of 2,000 at a step size of 200 to the original time series and then average the predictions. Again, this did not bring accuracy improvements.

3.3 Speaker Detection

During the competition, one submission posted a strikingly low log-loss score (approximately 0.2xx), far outperforming the rest of the leaderboard (most of which hovered around 0.6xx). This unexpected jump in performance prompted speculation about potential **dataset exploitation**—specifically, whether some participants might be using **speaker identification** techniques to recognize individuals who appear in both the training and test sets.

To explore this possibility, we trained a **1D Convolutional Network** to identify speakers from the audio data. Our hypothesis was that, if the same speaker's voice truly appeared in both training and test sets, a robust speaker identification model could approach near-perfect classification on those overlap cases. However, our experimental results did **not** confirm any significant overlap or yield meaningful performance gains.

4. Other Comments

Our **attention with trainable prediction tokens** approach was based on our solution to another competition which was held in parallel. For the Youth Mental Health Narratives: Automated Abstraction challenge, we applied the same technique to the **Llama 3.2** model (in both 1G and 3G configurations) and secured an 8th place finish on the private leaderboard. For the present Alzheimer's challenge, we simply replaced the Llama backbone with the **Whisper** encoder, quickly achieving competitive performance with minimal additional effort. Although we invested substantial efforts exploring further enhancements, the improvements were modest. Overall, this experience highlights the remarkable **power of pretrained foundation models** when effectively adapted to new tasks.

While larger foundation models often provide improved accuracy, in this competition we found that **Whisper–Large** did not outperform the **Medium** variant. During training, the Large model converged quickly but began overfitting sooner than the Medium model, ultimately leading to lower overall accuracy.

Reference

[1] Radford, Alec, et al. "Robust speech recognition via large-scale weak supervision." International conference on machine learning. PMLR, 2023.