A Vocal-Cue Interpreter for Minimally Verbal Individuals

Executive Summary | Deep Learning Project | Erdős Institute, Summer 2024

Team: Monalisa Dutta | Sarasi Jayasekara | Rahul Krishna | Alessandro Malusà | Atharva Patil | Julian Rosen
GitHub: https://github.com/julianrosen/erdos_dl_recanvo_project

Premise

Motivation: Nonverbal vocalizations play an important role in communication,particularly for individuals with few spoken words. While caretakers of minimally-verbal individuals often learn to interpret nonverbal vocalizations, the vocalizations can be difficult to interpret by people unfamiliar with the individual.

The Dataset: The ReCANVo dataset consists of ~7k audio recordings of vocalizations from 8 non-verbal or minimally-verbal individuals (such as people with developmental disabilities). The recordings were taken in a real-world setting, in a number of long sessions held at different locations (later broken into clips), and were categorized on the spot by the speaker's caregiver based on context, non-verbal cues, and familiarity with the speaker. There are several predefined categories such as self talk, frustrated, delighted, request, etc., and caregivers could also specify custom categories.

Our Goal: To train a model per individual that accurately predicts labels and improves upon previous work.

KPIs: Our primary metric is unweighted F1 score, for comparability with the existing work.

Related Work: There have been a few publications on classifying vocalization in the ReCANVo dataset. Most relevant to our work is Transfer Learning with Real-World Nonverbal Vocalizations from Minimally Speaking Individuals.

Approach & Results

Initial Decisions: For experimentation, we select two participants with large and varied sets of observations (P01, P05) and train a collection of models for both. For each of these two participants we drop the recordings that correspond to labels that have fewer than 30 data points.

Our baseline model predictions have significantly higher accuracy on vocalizations coming from sessions that were represented in the training data, which we believe is due to the model picking up on background sounds from the session. This will hurt the model's ability to generalize. To mitigate this, we adopt several strategies including,

1. Creating the train-test split in such a way that all the data points for a few entire sessions are completely contained in the test set.
2. Experimenting with adding extra layers of ambient noise or removing ambient noise to confuse the potential session recognition of the model

Models & Techniques:

- For extracting features from audio data, we focus on two deep models (HuBERT and AST) each with pre-trained weights. We also use mel spectrograms directly as the feature extraction technique, for comparison.
- As classifiers, we use CNN-based neural networks (for mel spectrograms), some networks with fully-connected layers (for features coming from deep models), and we also try some traditional machine learning methods (logistic regression, tree-based classifiers) as a way of fine tuning additional layers on top of the layers of the pre-trained deep models.
- To add ambient noise to recordings we use clips from the DEMAND dataset and to remove ambient noise we use an encoder-decoder based speech enhancement model called denoiser. Noise addition/removal is used in some of our experiments, but not all.

- We train several different combinations of models of the form “Feature Extractor + Classifier” on the training data. For participants 01 and 05 respectively, the model with the best performance was HuBert with one dense layer added on top (trained with penalty) and HuBert with two dense layers added on top, with F1 scores of 0.793 and 0.627.

Final Results & Conclusions

- On the test sets for participants 01 and 05 respectively, the best performing model displayed F1 scores of 0.712 and 0.582, each of which were improvements on the earlier team’s results that inspired us.
- Among all the models we used for feature extraction on this dataset, HuBert performed better than the others.

Possible Further Developments

- Since the models trained on P01 (who had a significantly higher count of data points compared to P05) outperformed that of P05, it seems reasonable to expect that model performances may significantly improve with more recorded sessions for an individual.
- Explore whether using clips of ambient noise from the actual session recordings as added background might help improve performance of the noise addition method.
- Combine our “noise engineering” methods with the other architectures we considered — besides HuBert + logistic regression.
- Attempt classification by broader label classes, e.g., by sentiment (positive vs. negative) and energy level (high vs. low).
- Build a model that can be generally trained, then be fine tuned for each individual, and develop it into a front end application.