Speaker Counting Problem

Kanishk Yadav (Author)

Original AISG Model:
https://drive.google.com/drive/folders/1bQlwT9opUZKvO7j2onO3VsCm3tC5V-pB

The model was originally given to me by AISG. Their model was using pretrained WavLM model tweaked for the task of overlapped speaker counting. I further trained and performed inferencing on 4 different models on different datasets (and their combinations) explained below and populated the results (in slides).

The folder contains the following:

- **HTML** contains all the necessary html pages for learning how to inference and train the models. (OSD/final deployment package/docs/build/html)
- **Job Scripts** contains the NSCC job results for all the models I trained and performed inferencing on.
- **inference_pipeline.pbs** contains the pbs script for performing inferencing (makes changes in paths accordingly).
- **train_pipeline.pbs** contains the pbs script for performing training (make changes in paths accordingly).
- **Trained Models** contains the 4 trained models on the following datasets:

DIHARD3
DIHARD3 *followed by* Libri2Mix
Libri2Mix *followed by* DIHARD3
Libri2Mix (Original by AISG)

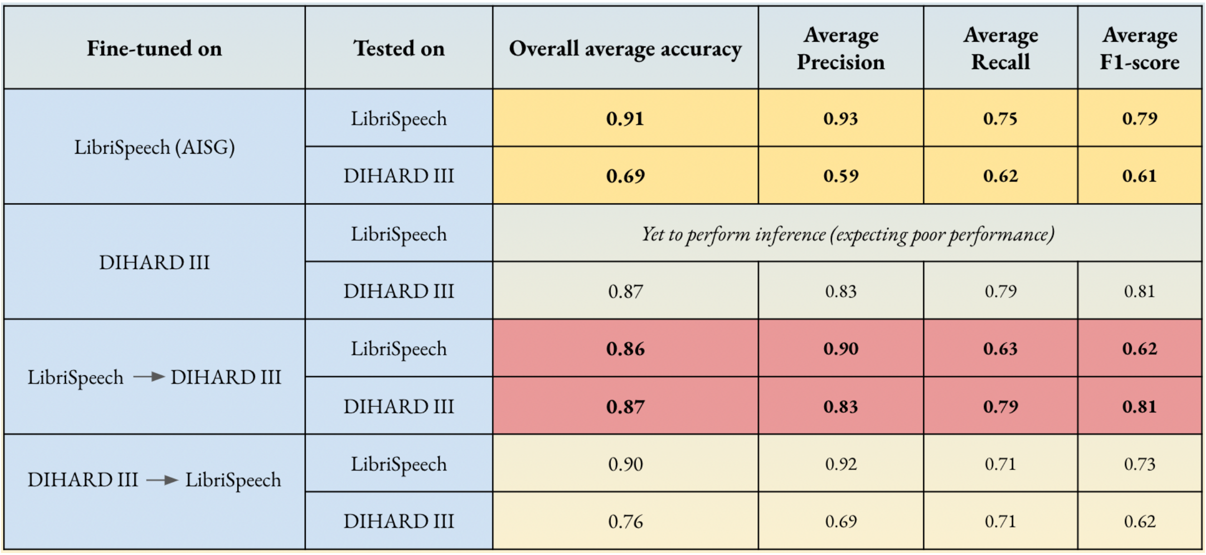
Yes, the DIHARD3 dataset had to be modified in a way (to make it suitable for the task of speaker counting with 0, 1, 2 speakers) that's similar to a way Libri2Mix was modified by AISG.

- **Results Reports** contains the results of inferencing performed on each of the 4 trained models in 6 different ways (this is a little confusing so feel free to ping me).

Trained on	Inferencing on
DIHARD3	DIHARD3
Libri2Mix	Libri2Mix
DIHARD3 followed by LibriMix	DIHARD3
DIHARD3 followed by LibriMix	LibriMix
LibriMix followed by DIHARD3	DIHARD3
LibriMix followed by DIHARD3	LibriMix

Keep in mind that for training, the training version of the respective dataset is used and similarly for testing (either testing or evaluation). Each of their folders contains a **info.txt** file for more explanation.

These were the final results:



Fine-tuned on	Tested on	Overall average accuracy	Average Precision	Average Recall	Average F1-score
LibriSpeech (AISG)	LibriSpeech	**0.91**	**0.93**	**0.75**	**0.79**
	DIHARD III	**0.69**	**0.59**	**0.62**	**0.61**
DIHARD III	LibriSpeech	*Yet to perform inference (expecting poor performance)*			
	DIHARD III	0.87	0.83	0.79	0.81
LibriSpeech → DIHARD III	LibriSpeech	**0.86**	**0.90**	**0.63**	**0.62**
	DIHARD III	**0.87**	**0.83**	**0.79**	**0.81**
DIHARD III → LibriSpeech	LibriSpeech	0.90	0.92	0.71	0.73
	DIHARD III	0.76	0.69	0.71	0.62

I felt that the third model (LS followed by DH3 gave the most balanced results for all scenarios and can hence be used as a baseline for further exploration).

- **klass2-conda-env.yaml** is the environment to be set-up.

- **Py Scripts** are python scripts written by me to pre-process dataset (pipelines given by AISG were a little resource intensive so I had to take other approach - the technique is the same)
- **src** contains the pipelines (most of them are similar to the ones in the code by AISG, but since I tweaked a few of them, I have shared them here) - (OSD/final deployment package/klass-osd-kedro-pipeline/klass-osd/src)
- **DH3-WavLM** contains the necessary scripts required to convert/transform the DIHARD3 dataset into a dataset that is recognizable for the speaker counting task.

Since the original package shared by AISG has all the steps required to install all the datasets and run pipelines, I am not including my entire repo in the package because the datasets are itself >20 GB in size. Almost everything that's necessary including the trained models has been attached in the package.

0.4 and 0.6 are the chunk size in seconds (400 ms and 600 ms)