AAI Project: Text-Independent Speaker Identification

Deadline: 31/12/2022

Speaker identification (SI) is a multiclass classification task of classifying the identity of an unknown voice in a set of speakers [1,2]. Based on whether the spoken text is constrained, SI systems can be categorized into text-dependent (TD) ones and text-independent (TI). The TI mode is more challenging, as the system should avoid the influence of the spoken content. By default, an SI system only classifies speakers that have been registered, although the system can be used to classify new speakers with some techniques. The default setting is adopted by this project, i.e., the classes in the test set are all included in the training set.

In this project, you are required to build a text-independent speaker identification system. Given a dataset of <audio, label> pairs, the model should learn to classify the speaker identity of the input speech. Then the model will be tested on another set of audio spoken by the same people. **The dataset for SI training is specified, and you are NOT allowed to use other labeled data.** However, you may use extra **unlabeled data** to train your model.

SI is a well-explored task, so there are various literatures for your reference [2,3,4,5]. Below are the descriptions of the dataset and submission requirements.

Dataset

Description

The dataset is a subset of LibriSpeech [6], derived from recordings of audiobooks. The dataset consists of 250 speakers. The number of training samples is 23172. For testing, there are 2657 standard samples and 2684 noisy samples. The noisy samples are made of clean samples and noise clips. There are 2000 noise clips used in total, all derived from the DNS Challenge [7] and given in the dataset. The structure of the data directory is illustrated below.

```
LibriSpeech-SI/
├── train/
│    ├── spk001/
│    │   ├── spk001_001.flac
│    │   ├── spk001_002.flac
│    │   └── ...
│    │
│    ├── ...
│    │
│    └── spk250/
│        ├── spk250_001.flac
│        ├── spk250_002.flac
│        └── ...
│
├── test/
│   ├── test0001.flac
│   ├── ...
```

```
│   └── test2657.flac
│
├── test-noisy/
│   ├── test2658.flac
│   ├── ...
│   └── test5341.flac
│
└── noise/
    ├── 00Yi2Lz3UWc.wav
    ├── 01i2tgcRaDA.wav
    └── ...
```

The *train* directory, which contains the training set, has two levels: speaker and sample. The directories within train represent different speakers, and their names should be used as labels. All the audio samples belonging to the same speaker are corresponding training samples. The *test* and *test-noisy* both contain testing samples. The *noise* directory contains noise clips that appear in *test-noisy* samples.

The *train* subset provided above is used for training and validation. You should manually separate the dataset into the training and validation subsets. Since the recordings of LibriSpeech is quite clean, you can achieve pretty high accuracy on the standard test set with proper techniques. This is why another noisy test set is provided. In the noisy samples, the Signal-to-Noise Ratio (SNR) is kept to 15, sometimes higher. To overcome the clean/noisy domain gap, you can use the noise clips to simulate noisy conditions during training.

Download

The dataset can be downloaded from Blackboard (7.0G). The access requires your SUSTech identity.

Submission

There are three items you should submit: 1) prediction results from your model, 2) the source code used for training and inference, and 3) a report. Below are the details.

Prediction Results

Two separate test sets are provided, each containing a bunch of audio files. After training your model, you need to generate speaker labels on the test set. The results should be stored in a *.txt* file, and the content should look like

```
test0001.flac spk049
test0002.flac spk201
...
test5341.flac spk029
```

In the example, each line contains the file name and the predicted speaker id that are separated by a space. You should submit this result file for the objective evaluation. **Note that the evaluation is mainly based on the standard test set, so please focus on the performance on clean samples first.**

Source Code

The whole process will involve data pre-processing, model training, and prediction. You will definitely have to pre-process the data on your own. You are encouraged to have your own model implementation, at least an interface combining open resources. All the code should be packed and submitted. Probably you will use Python, and you can attach a simple environment configuration file, e.g., requirement.txt.

Report

Several key elements should be included in your report: 1) model description, 2) experimental details, 3) the training records, 4) member contribution.

In the model description, you have to describe the model structure that you use. For example, the model can be a combination of CNNs, Transformer layers, and a pooling head. The computation process should be explained. For the experimental details, you should mention how you implement your model, what features your model takes, the open resources that you use, and the hyperparameters including the learning rate, model size, etc. As for the training records, you can provide the loss curves on the training and validation sets. If you improve your model gradually, you can give further explanations. Finally, the contribution of each group member should be mentioned.

References

[1] Jahangir, Rashid, et al. "Speaker identification through artificial intelligence techniques: A comprehensive review and research challenges." *Expert Systems with Applications* 171 (2021): 114591.

[2] An, Nguyen Nang, Nguyen Quang Thanh, and Yanbing Liu. "Deep CNNs with self-attention for speaker identification." *IEEE access* 7 (2019): 85327-85337.

[3] Qi, Minhui, et al. "Deep CNN with se block for speaker recognition." *2020 Information Communication Technologies Conference (ICTC)* . IEEE, 2020.

[4] Safari, Pooyan, Miquel India, and Javier Hernando. "Self-attention encoding and pooling for speaker recognition." *arXiv preprint arXiv:2008.01077* (2020).

[5] Yang, Shu-wen, et al. "Superb: Speech processing universal performance benchmark." *arXiv preprint arXiv:2105.01051* (2021).

[6] Panayotov, Vassil, et al. "Librispeech: an asr corpus based on public domain audio books." *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)* . IEEE, 2015.

[7]**https://dns4public.blob.core.windows.net/dns4archive/datasets_fullband/noise_fullband/datasets_fullband.noise_fullband.audioset_000.tar.bz2**