

This folder contains data that can be used in experiments with phoneme recognition in speech samples recorded in Polish. Acoustic data used here were extracted from CLARIN-PL speech corpus after rejecting speech samples where recorded sequence of words does not correspond strictly to the word sequence declared as the sample orthographic transcription. Three data sets are prepared which can be used for recognizer training, validation and testing. Data sets are located in three subfolders:

- `train` - contains data that should be used as the training set,
- `devel` - contains data that can be used for the recognizer validation,
- `test` - contain data that can be used for testing.

Datasets are stored in the format conformant with python numpy library, so they can be easily processed and reshaped if necessary in python code. The contents of `train/devel/test` folders are analogous. Below we describe contents of train folder. Remaining ones are organized in the same way.

Train folder contains the following data components:

- `train.mfcc.bin` – the file containing pairs (acoustic features, phoneme index) for all frames extracted from all speech samples assigned to train set. Acoustic features are here MFCC features (Mel Frequency Cepstral Coefficients). The features were extracted from audio files using HCopy program from HTK package. The extraction parameters are defined in `HCopy_MFCC.cnf` file located in `src` subfolder.
- `train.mfsc.bin` – the file containing pairs (acoustic features, phoneme index) for all frames extracted from all speech samples assigned to train set. Acoustic features are here MFSC features (Mel Frequency Spectral Coefficients). The features were extracted from audio files using HCopy program from HTK package. The extraction parameters are defined in `HCopy_MFSC.cnf` file located in `src` subfolder,
- `train.zip` – the archive file containing feature files as well as phone alignment files of all used speech samples. Speech samples are grouped by sessions (session is the set of samples recorded by the same speaker in constant acoustic conditions). The data from each session are stored in the individual subfolder (subfolder are named by numbers signed automatically by the program that converts CLARIN-PL data into loadable datasets). For each speech sample there are three files:
 - `<nnn>.mfsc` – binary file containing MFSC features created by Hcopy program,
 - `<nnn>.mfcc` – binary file containing MFCC features created by Hcopy program,
 - `<nnn>.out` – text file containing frame alignment to phones.

In the simple usage scenario the actual training and testing sets can be created using merely `*.mfcc.bin` or `*.mfsc.bin` files. Alignment files (`.out`) can be used if feature extraction method (or parametrization) is to be changed. Feature files (`<nnn>.mfsc` or `<nnn>.mfcc`) can be utilized in experiments with feature selection or if the per-frame feature vector should be extended with some new features.

`train.mfcc.bin` and `train.mfsc.bin` bin files contain data organized as per frame (feature, phoneme) vectors. In the phoneme recognition experiments, usually the feature vector (recognizer input) is the concatenation of the per-frame feature vector of the frame surrounding the one being currently recognized. Therefore, the actual training data should be created by merging feature vectors extracted in individual frames. It can be done using the script `ds_extract.bat` (Windows) or `ds_extract.bash` (Linux) located in `src` subfolder. The script should be called in the following way.

```
src/ds_extract <feature_type> <context_width> train_data_fraction>
<test_data_fraction>
```

where:

<feature_type> - one of mfsc or mfcc
<context_width> - single side context width in frames
<train_data_raction> - percentage of data in input bulk dataset to be used for training (0-100)
<test_data_raction> - percentage of data in input bulk dataset to be used for development and testing (0-100)

For example:

```
src/ds_extract mfsc 05 50 70
```

will create three ready to load datasets:

train.mfsc.50.05.bin

devel.mfsc.70.05.bin

test.mfsc.70.05.bin.

These files can be used in the simple experiment with phoneme recognition using DNN. The example of how to load the binary datasets and how to use them for DNN training and testing can be found in `nn_train_and_test.py` python program. In order to run the experiment with the prepared data files call the program as follows:

```
python src/nn_train_and_set.py train.mfsc.50.05.bin  
devel.mfsc.70.05.bin test.mfsc.70.05.bin -e 30
```