Emotion detection challenge

By Henri

Emotion detection

Emotions

- In our dataset we have 5 Emotions:
  - Anger **(A)**
  - Empathic **(E)**
  - Neutral **(N)**
  - Positive **(P)**
  - Rest **(R)**

Dataset origin

- FAU - AIBO corpus (made by University of Erlangen)
- Overall corpus has length of ~ 9h
- ~ 5h of Training data and ~ 4h of evalua
- Overall 1.1GB of raw wave data - 200MB of extracted LLD features
- 18216 utterances or “chunks”

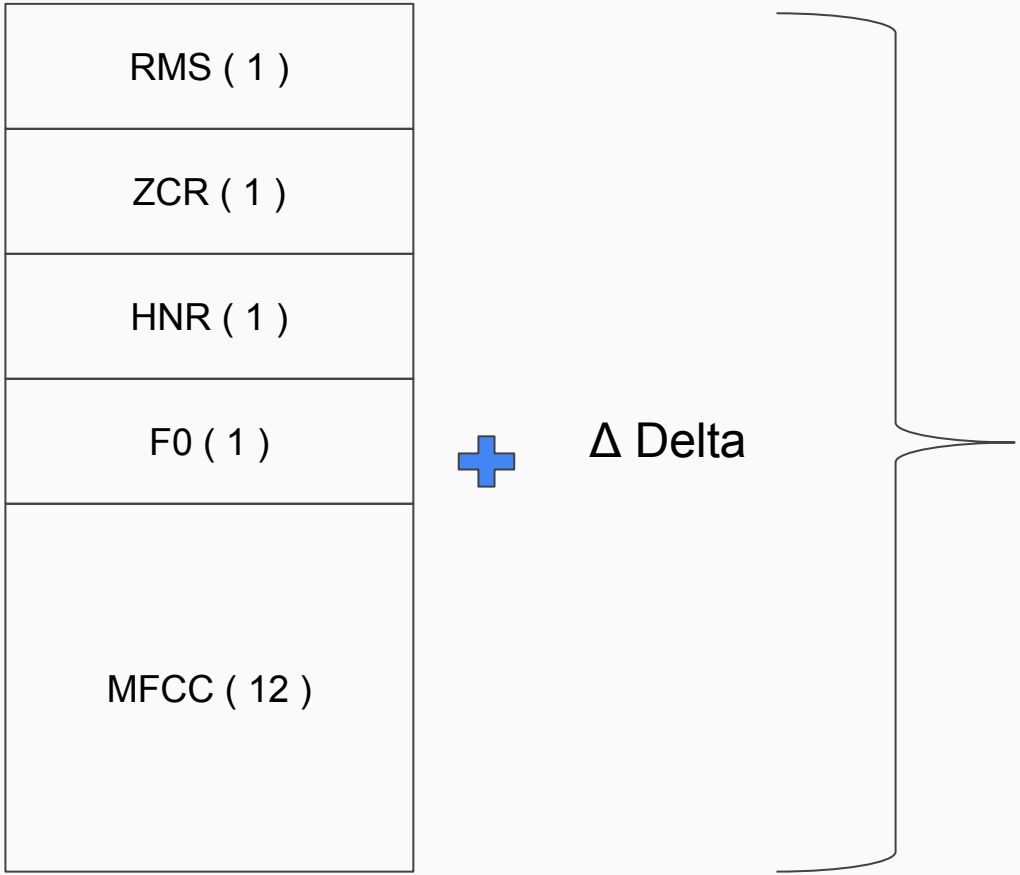
Dataset composition

	A	E	N	P	R	Sum
#train	881	2093	5590	674	721	9959
#test	611	1508	5377	215	546	8257
Sum	1492	3601	10967	889	1267	18216

Features

- The standard features that are used in this task are LLD (Low level descriptors)
- It’s basically a fusion of many ordinary features, such as MFCC, ZCR (zero crossing rate, F0-frequency (pitch), HNR (Harmonics to Noise), RMS (Root mean square))
- An utterance level feature is then calculated over all samples by computing 12 different reduction methods (moments, extremes, linear regression)

Feature composition



- 32 dimensional feature
- 12 reductions are applied on the given features (frame wise)
- Mean, variance, extreme (min,max), linear regression and so on
- Resulting feature is 32*12 (384) dimensional features per utterance

Additional features

- We additionally provide common - samplewise - features, such as MFCC and FBANK
- MFCC is 39 dimensional (13 static + Δ + Δ Δ)
- FBANK is 120 dimensional (40 static + Δ + Δ Δ)
- These are **not necessary** to complete the task, only if the given features are not enough (too few samples), you might take a look!

Features

- Features are provided in the CSV format
- Loading features is pretty easy:

```
python -c "import numpy as np;
a=np.loadtxt(open('data','rb'),delimiter=',');
```

Goal

- Train a successful classifier for emotion detection task
- Play with the parameters of this classifier to improve its performance
- Evaluate the performance on the test dataset!

Task (Compulsory)

- Participate in the “classifier sub challenge”
  - Train a classifier for the 5 class problem. (**A, E, N , P, R**)
  - Test your classifier on the given test data
  - Every classifier and every possible method is **allowed!** (including fusion classifiers or more sophisticated ones)
  - Results are weighted over the F-score of the unweighted accuracy and unweighted recall
- No feature extraction is necessary

Task (optional)

- If necessary, we also provide the raw wave files, to extract your own features.
- Additional points can be given if you participate in the two class challenge, which only discriminates into two classes (**NEG, IDL**)
- Extend the system by using the provided transcriptions (e.g. using HMM based classification)

Baseline

The baseline system uses a basic HMM which estimates probabilities. The maximum probability over 5 states is chosen to be the representative for the utterance/chunk.

	Precision	Recall
Baseline	29.6%	35.5%

Framework recommendations

- Deep learning:
  - MXNet (Python)
  - CNTK (New version has python backend)
  - Torch / Torchnet (Lua)
- Feature extraction (optional) :
  - HTK
  - Kaldi
  - Opensmile
  - Pymir (Python)
- Machine Learning:
  - Scikit-learn (python)
- Other recommendations:
  - For python, please use conda! Makes **installing** packages and environments in python super easy!

Data directories

- Two files on the ftp:
  - Features_labels_lld.zip - are the feature extracted files, **recommended**
  - Features_labels_raw_wav.zip - are the raw wavs, for **optional** feature extraction
- After downloading the required dataset, extract it using e.g unzip
- The data is generally formatted as:
  - <school>_<student>_<turn>_<chunk>.wav or <school>_<student>_<turn>_<chunk>.csv
- <school> is “Ohm” for training data, “Mont” for test data
- <student> is a unique students id
- <turn> is a consecutive number
- <chunk> is the current “emotion” within the current turn. Consecutive number

Labels

- Labels are found in labels/train for the respective task.
- Labelfile has the format:
- UTTERANCENAME LABEL CONFIDENCE
    - Utterancename is the same as the filename
    - Label is one of the five given classes
    - Confidence can be neglected for the challenge

Scoring

Final result is based on the average recall and average precision (since we have a 5 class problem)

A script can be found in the root directory of the extracted files called “score.py”, which uses scikit-learn to score.

E.g.:

```
./score.py myresults labels/test
```

Submission

Upload your result to the FTP!

The final result should be the produced output of your model in the format:

UTTERANCENAME PREDICTEDLABEL

The best model will be announced later!

Also provide a short **model description** (2 pages at least), using the templates from ICASSP or Interspeech

Have fun! Thanks! 谢谢