How to extract cmp for DV3 using merlin (eng)

노트북:	일archive		
만든 날짜:	2019-07-09 오전 11:05	**업데이트:**	2019-07-15 오후 2:42
작성자:	hash2430@gmail.com		

How to extract trimmed cmp for DV3 using merlin

Following assumes 's2' as project name for merlin and extracts features out of VCTK using 22050 sampling rate, 1024 fft size, 256 hopping size.

1. DV3 generates trimmed wav at 'VCTK-Corpus/wav_trim_22050'
2. Create directory 's2' at /egs/build_your_own_voice
3. run setup
    1. ./01_setup.sh s2
        1. 'conf/global_settings.cfg' file is generated
        2. Make some modification
            1. SamplingFreq = 22050

```
./01_setup.sh s2
```

4. extract acoustic features

    1. Description: Extracting SPTK features such as mgc, lf0, bap
    2. Shell script modification: original 03_prepare_acoustic_features.sh expects wav files to be right below the DB directory. However, VCTK has one more layer for each speaker. Thus you have to create new shell script for that.
        1. 03_vctk_prepare_acoustic_features.sh

```
mkdir -p database/feats
./03_vctk_prepare_acoustic_features.sh /home/administrator/Music/VCTK-
Corpus/wav_trim_22050 database/feats
```

    3. Result

        1. Time spent: 21min 57sec
        2. output:
            1. World creates bapd, f0, sp and SPTK generates bap, lf0, mgc.
            2. For our 'database/feats/' directory we only need bap, lf0, mgc, but I left all the other for debugging purpose.
4. Etc.
    1. For 22050 sampling rate, 1024 fft size, 256 samples hopping size, you have to fix some scripts.

```
1. Add the case of 22050 sampling rate at
"/home/administrator/projects/merlin/misc/scripts/vocoder/world/extract_features_for_merlin.py"
     nFFTHalf = 1024
     alpha = 0.65
# nFFTHalf is option for SPTK. unlike the name, SPTK takes the variable as fft
size.
# According to SPTK manual, that variable is taken as half of fft only when
extracting mgc directly from wav. Whereas we are extracting mgc from sp
indirectly.)

2. "/home/administrator/projects/merlin/tools/WORLD/test/analysis.cpp":
      'frame_period' variable changes from 5.0 to 11.609 (=(256/22050)*1000)  to
match 256 hopping size of DV3 for VCTK.
3. Any 'frame_period' variables of
/home/administrator/projects/merlin/tools/WORLD/test directory has to be changed
likewise.
4. Sometimes the variable is deifned in macro as "FRAMEPERIOD"
5. Hardcode the "frame_period" argument of "src/synthesis.cpp" Synthesis()
6. test/synth.cpp: modify FRAMEPERIOD
7. build the tool by running "merlin/tools/compile_tools.sh"
8. Verification: fft size is printed when extracting feature. See if it is 1024.
```

5. Create file_id_list.scp
    1. In original Merlin flow, file_id_list.scp is created when creating labels. In here end2end is assumed thus label generating steps are excluded.
    2. Thus I have separately generated shell script to generate file_id_list.
        1. gen_file_id_list.sh
6. prepare conf files

```
./04_prepare_conf_files.sh conf/global_settings.cfg
```

1. Result of shell script: conf/acoustic_s2.conf, conf/test_synth.conf is created
2. conf/acoustic_s2.conf modification

```
# sub-processes


NORMLAB  : False
MAKECMP  : True
NORMCMP  : True


TRAINDNN : False
DNNGEN   : False


GENWAV   : False
CALMCD   : False
```

7. Create cmp
    1. Comment the following parts from run_merlin.py

```
1. assert according to data split => This part is commented because we want all
features to be extracted using World here regardless of the split
    #print "file_id_list: {}\n".format(cfg.file_id_scp)
    #print "total_file_number={0}, train_file_number={1}, valid_file_number={2},
test_file_number={3}\n".format(total_file_number,
cfg.train_file_number,cfg.valid_file_number,cfg.test_file_number)
    #assert cfg.train_file_number+cfg.valid_file_number+cfg.test_file_number ==
total_file_number, 'check train, valid, test file number'
2. trimming using HTS label => This part is commented because I am trimming with
dB information when preprocessing occurs at DV3.
```

```
        # elif cfg.remove_silence_using_hts_labels:
        #     ## back off to previous method using HTS labels:
        #     remover = SilenceRemover(n_cmp = cfg.cmp_dim, silence_pattern =
cfg.silence_pattern, label_type=cfg.label_type, remove_frame_features =
cfg.add_frame_features, subphone_feats = cfg.subphone_feats)
        #     remover.remove_silence(nn_cmp_file_list, in_label_align_file_list,
nn_cmp_file_list) # save to itself
```

2.call run_merlin.py

```
python run_merlin.py
/home/administrator/projects/merlin/egs/build_your_own_voice/s2/conf/acoustic_s2.conf
```

8. Normalize cmp
    1. Above run_merlin.py step covers normalization as well.
9. Copy and synthesis cmp to verfiy if cmp is extracted correct
    1. test_synth_s2.conf modification

```
framelength: 1024

fw_alpha: 0.65


# sub-processes


NORMLAB  : False
MAKECMP: False
NORMCMP: False


TRAINDNN: False
DNNGEN   : False


GENWAV   : True
CALMCD: False
```

2. Copy mgc, bap, f0 from 'acoustic_model/data' into experiments/s2/test_synthesis/wav

3.Write filename without extention to 'experiments/s2/test_synthesis/test_id_list.scp'

```
p225_001
```

4. call run_merlin

```
python run_merlin.py
/home/administrator/projects/merlin/egs/build_your_own_voice/s2/conf/test_synth_s2.conf
```