Kaldi 실습

실습 목적

- 음성 인식에 필요한 지식 습득
- Kaldi 라이브러리를 이용한 음성 인식기 구축 과정 체험 및 실습
- 다양한 형태의 음성 인식기 구축 능력 습득

목차

1. Linux 및 Kaldi 설명

2. Mini-librispeech 데이터 설명

3. 데이터 전처리 및 특징 추출 실습

4. 음성 인식기 학습 및 디코딩

Linux 및 Kaldi 설명

Linux 기본 명령어

- ls
- cd
- cp
- mv
- rm

Linux 기본 명령어

- cat
- head
- tail
- find
- grep
- bash

Kaldi 소개

- Kaldi
  - Automatic Speech Recognition (ASR) Toolkit
    - 음성 데이터가 입력되었을 때, 해당 데이터에 대한 가장 높은 확률을 가지는 단어의 sequence를 자동으로 출력하는 Toolkit
    - 현재 가장 널리 쓰이는 음성인식 Toolkit
  - 특징
    - C++ 기반의 오픈소스 음성인식 툴킷
    - 다양한 neural network 기반 음향 모델 학습 지원
      - DBN, CNN, LSTM, RNN, end-to-end
    - 주요 corpus에 대한 튜토리얼 자료 제공
    - 리눅스 환경에서 설치하는 것을 권장

Mini-librispeech

Mini-librispeech

- Mini-librispeech
  - LibriSpeech corpus중 일부 2608 문장(67 화자)을 뽑아 만든 데이터셋
  - 사람이 직접 읽은 오디오북에서 추출한 문장들로 구성
  - 무료로 데이터 및 언어모델 이용 가능

Mini-librispeech

❑ 데이터 살펴보기

```
cd /opt/kaldi/egs/mini_librispeech/s5/corpus/LibriSpeech
```

```
root@a50c15ffed3d:/opt/kaldi/egs/mini_librispeech/s5/corpus/LibriSpeech# ls
CHAPTERS.TXT  LICENSE.TXT  README.TXT  SPEAKERS.TXT  dev-clean-2  train-clean-5
```

```
cd train-clean-5
ls
```

```
root@a50c15ffed3d:/opt/kaldi/egs/mini_librispeech/s5/corpus/LibriSpeech# cd train-clean-5
root@a50c15ffed3d:/opt/kaldi/egs/mini_librispeech/s5/corpus/LibriSpeech/train-clean-5# ls
1088  163   1867  19    2136  2416  3242  3526  3947  4640  5789  669   7312  7859
118   1737  1898  1970  226   32    332   3664  460   4680  6272  6848  7367  8629
```

❑ 음성 파일

```
flac -d 163-122947-0003.flac && mv 163-122947-0003.wav /opt/kaldi/share_kaldi/
```

❑ 대본 파일

```
head -n 10 163-122947.trans.txt
```

데이터 전처리 및 특징 추출 실습

Kaldi 음성 인식 실습

- 음성인식 모델 전체 과정을 한번에 처리할 수 있는 스크립트 제공

```
bash run.sh
```

  - 데이터 다운로드
  - GMM-HMM 모델 학습
  - DNN-HMM 모델 학습
  - 음성 디코딩

- 데이터 전 처리 부분만 직접 바꾸어 주면 다른 데이터셋에서도 적용 가능

데이터 셋 전처리

- egs 폴더에서 전처리 스크립트 제공

```
local/data_prep.sh ./corpus/LibriSpeech/train-clean-5 data/train_total
local/data_prep.sh ./corpus/LibriSpeech/dev-clean-2 data/test
```

- 전처리 폴더 확인

```
cd data/train_total
```

  - wav.scp

```
head -n 10 wav.scp
```

  - utt2spk

```
head -n 10 utt2spk
```

  - spk2utt

```
head -n 10 spk2utt
```

  - text

```
head -n 10 text
```

- 데이터 분할

```
utils/subset_data_dir_tr_cv.sh --cv-spk-percent 30 data/train_total data/train data/dev
```

발음 사전 준비

- egs 폴더에서 전처리 스크립트 제공

```
local/prepare_dict.sh --stage 3 data/local/lm data/local/lm data/local/dict
```

- 전처리 폴더 확인

```
cd data/local/dict
```

  - lexicon

```
head -n 30 lexicon.txt
```

  - 음소 구분

```
cat nonsilence_phones.txt
cat silence_phones.txt
cat optional_silence.txt
```

  - 음소 클러스터링

```
cat extra_questions.txt
```

발음 사전 모델 준비 (음소-단어)

- egs 폴더에서 전처리 스크립트 제공

```
utils/prepare_lang.sh data/local/dict "<UNK>" data/local/lang_tmp data/lang
```

- 전처리 폴더 확인

```
cd data/lang_nosp
```

  - 음소, 단어, 발음 사전에 없는 단어 (Out Of Vocabulary)

```
head -n 30 phones.txt
head -n 30 words.txt
cat oov.txt
```

  - Lexicon WFST(weighted finite state transducer)

```
fstprint --isymbols=phones.txt --osymbols=words.txt L.fst | head -n 40
```

언어 모델 준비 (단어-단어)

- arpa 형식의 trigram 모델 제공

```
gunzip -c data/local/lm/lm_tgsmall.arpa.gz | tail -n 100
```

- Grammar WFST(weighted finite state transducer) 생성

```
gunzip -c data/local/lm/lm_tgsmall.arpa.gz | \
arpa2fst --disambig-symbol=#0  --read-symbol-table=data/lang/words.txt - data/lang/G.fst
```

- Grammar WFST(weighted finite state transducer) 확인

```
fstprint --isymbols=words.txt --osymbols=words.txt G.fst | head -n 40
```

음성 특징 추출

- MFCC 특징 추출

```
steps/make_mfcc.sh --cmd run.pl data/train
steps/make_mfcc.sh --cmd run.pl data/dev
steps/make_mfcc.sh --cmd run.pl data/test
```

- 추출 결과 확인

```
copy-feats scp:data/train/feats.scp ark,t:- | head –n 100
```

- 화자별 normalization 값 계산

```
steps/compute_cmvn_stats.sh data/train
steps/compute_cmvn_stats.sh data/dev
steps/compute_cmvn_stats.sh data/test
```

- 계산 결과 확인

```
copy-feats scp:data/train/cmvn.scp ark,t:- | cat
```

음성 인식기 학습 및 디코딩

Monophone 기반 음성 인식 모델

- 음성 인식 모델 학습

```
steps/train_mono.sh --cmd run.pl data/train data/lang exp/mono
```

- 디코딩 그래프 생성

```
utils/mkgraph.sh data/lang exp/mono exp/mono/graph
```

- 음성 파일 디코딩

```
steps/decode.sh --nj 1 --cmd run.pl exp/mono/graph data/dev exp/mono/decode_dev
```

Triphone 기반 음성 인식 모델

- alignment 추출

```
steps/align_si.sh --cmd run.pl data/train data/lang exp/mono exp/mono_ali_train
```

- 음성 인식 모델 학습

```
steps/train_deltas.sh 2000 10000 data/train_500short data/lang_nosp exp/mono_ali_train exp/tri
```

- 디코딩 그래프 생성

```
utils/mkgraph.sh data/lang_nosp_test_tgsmall exp/tri exp/tri/graph_nosp_tgsmall
```

- 음성 파일 디코딩

```
steps/decode.sh --nj 2 --cmd run.pl exp/tri/graph_nosp_tgsmall data/dev exp/tri/decode_nosp_tgsmall_dev
```