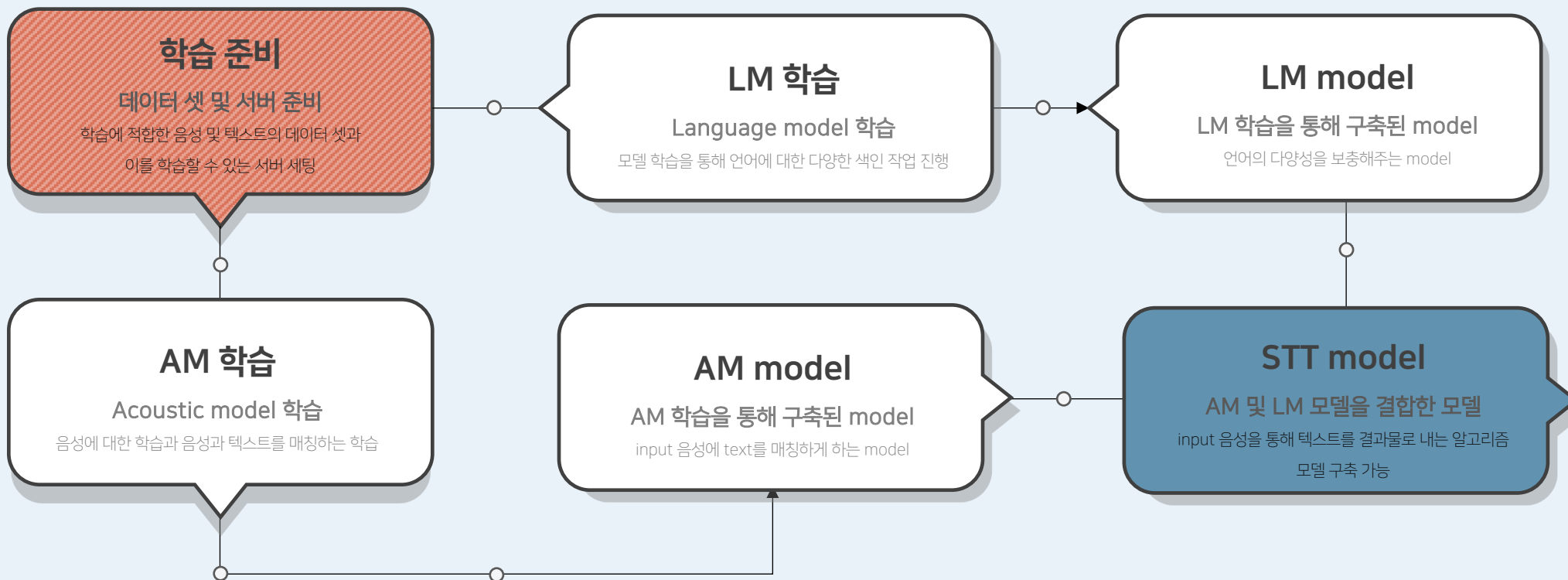


STT

STT

STT 모델 구축을 위한 전체 workflow



STT DATA

음성인식을 위한 데이터 전처리



기본 데이터 형태

텍스트와 음성 셋

> 음성 : 8k 혹은 16k wav 형태

> 텍스트 : UTF8 인코딩 사용



데이터 형태별 학습 방법

음성 길이에 따른 필요 GPU 및 ram

> 음성 : 8k 혹은 16k wav 형태

> 텍스트 : UTF8 인코딩 사용



GPU	ram (GB)	max duration (min)
GTX1080	8	0.5
GTX1080Ti	11	2
V100	32	11

참고 : <https://pms.maum.ai/confluence/pages/viewpage.action?pageId=15342283>



학습용 데이터 준비

가진 GPU와 ram에 맞는 길이의 음성 데이터 및 이에 매칭되는 텍스트 데이터 준비

> CNN STT의 경우 **통일성** 있는 기준을 갖고 데이터를 정제하면 됨

> data의 sample rate에 맞게 하지 않더라도, 학습 전처리 시 자동적으로 설정된 값에 맞게 전처리 진행

> 학습 전 baseline 데이터 준비 필요

STT - Docker

음성인식 학습을 위한 docker 사용



docker image 받기

최신의 w2l image

> url : <https://docker.maum.ai:8443/>

> 명령어 : docker pull docker.maum.ai:443/brain/w2l:1.1.0 (버전정보)

01

서버 정보 : 182.162.19.12

ID : minds

PW : msl1234!



docker run

docker run -it --gpus all -e LC_ALL=C.UTF-8 --name w2l-Train_yoong -v /data1:/data1 docker.maum.ai:443/brain/w2l:1.1.0 /bin/bash

> 해당 명령어는 gpu를 모두 사용하게 되어있음 (변경 필요시 : --gpus "device=1" 로 사용하고자 하는 gpu 지정 필요)

> --name : 만들고자 하는 container의 이름

> -v : local과 docker의 공유 폴더

※ local 폴더의 경우 상세 경로까지 지정 가능하며, docker 내에서는 local 경로가 아닌 docker의 경로를 사용해주어야 함

[GPU NUM - 12번]

LHR : 0

CHW : 1

CMJ : 2

SJY : 3

RJH : 4

LJW : 5

LSM : 6

LKM : 7

[GPU NUM - 29번]

YSM - 0번

HW - 1번

STT - LM

Language Model 학습



container 접속

`docker exec -it [container name] bash`

> attach로 docker에 접속할 경우 history가 남으니 attach로 접속하는 것도 좋은 방법입니다



1단계

`python make_lm_text.py --txt_path /data1/stt_10/TEST/data --output /data1/stt_10/TEST/lm_output`

> 경로 : `cd /root/wav2letter/recipes/korean/data`

> `--txt_path` : resources의 경로

> `--output` : 결과물(`lm_train.txt`)을 받을 경로 (생성해둔 `lm_output`의 경로)

→ 추가 가능 명령어

> `-l eng` : default = kor

> `-ue` : eos 사용시 필수 (→ 사용시 줄바꿈이 된 결과값을 받을 수 있음)

> `-ut` : word lm이 아닌 token lm 학습시 필수

STT - LM

Language Model 학습



2단계 (word lm)

```
/root/kenlm/build/bin/implz --discount_fallback -o 4 /data1/stt_10/TEST/lm_output /lm_train.txt /data1/stt_10/TEST/lm_output /lm.arpa
```

token lm

```
/root/kenlm/build/bin/implz -T /tmp -S 50G --discount_fallback --prune 0 0 0 0 0 1 1 1 2 3 -o 20 trie
```

```
</data1/stt_10/TEST/lm_output/lm_train.txt >/data1/stt_10/TEST/lm_output/lm.arpa
```

> < 1단계에서 생성된 lm_train.txt 경로

> > 생성될 (저장될) lm.arpa의 경로



3단계

```
python prepare_kenlm.py --dst /data1/stt_10/TEST/lm_output
```

> 경로 : cd /root/wav2letter/recipes/korean/data

> --dst : 결과물 받을 경로

→ 추가 가능 명령어

> -l eng : default = kor

> -ut : word lm이 아닌 token lm 학습시 필수

STT - AM

Acoustic Model 학습



config 수정

vi /data1/stt_10/TEST/train.cfg

> 공통 수정 사항

- runname, rundir, tokendir, archdir, train, valid
 - ※ 보통은 *_output 형태로 쓰기 때문에 그 앞만 경로를 수정해주면 됨
- > noise의 경우 필요시 사용하며, 미사용시 공백처리 (주석처리 불가능)

1. sample rate별

16k

- samplerate=16000
- minnoisesample=400
- maxnoisesample=10560000

8k

- samplerate=8000
- minnoisesample=200
- maxnoisesample=5280000

2. 언어별

한국어

- lexicon = 미사용
 - (자소단위까지 분리하지 않기 때문에 성능 영향 미미)
- uesnormalizer=true
- uesmecab=true
- ※ eos 사용시
 - eostoken=true
 - replable=0

영어

- lexicon=~am_output/am 경로
 - ※ lexicon = 데이터 색인 사전
- usenormalizer=false
- usemecab=false
- replabel=2

```
#Training config for Librispeech using Gated ConvNets
# Replace `[...]`, `[MODEL_DST]`, `[DATA_DST]` with appropriate paths
--runname=edulab_test
--rundir=/data1/stt_10/TEST/am_output/model
--tokensdir=/data1/stt_10/TEST/am_output/am
--archdir=/data1/stt_10/TEST/am_output
--train=/data1/stt_10/TEST/am_output/train.lst
--valid=/data1/stt_10/TEST/am_output/dev.lst
--noise=
--lexicon=
--arch=network.arch
--tokens=tokens.txt
--criterion=asg
--lr=0.0001
--lrcrit=0.000001
# --linseg=1
--linseg=0
--momentum=0.8
--maxgradnorm=0.2
--surround=1
--onorm=target
--sqnorm=true
--mfsc=true
--nthread=6
--batchsize=1
--transdiag=4
--netoptim=adam
--critoptim=adam
--adambeta1=0.9
--adambeta2=0.999
--optimepsilon=1e-8
--usenormalizer=true
--usemecab=true
--eostoken=true
--replabel=0
--samplerate=8000
--minnoisesample=100
--maxnoisesample=5280000
```

STT - AM

Acoustic Model 학습



1단계

```
python prepare_am.py --max_duration 60000 --train /data1/stt_10/TEST/data --output /data1/stt_10/TEST/am_output -s 8000 -ue
```

> 경로 : cd /root/wav2letter/recipes/korean/data

> 16k가 default로 설정되어 있음

> max_duration : 사용될 음성의 최대 길이 (ms 기준)

→ 600000 = 10분 / 60000 = 1분

→ 추가 가능 명령어

> -l eng : default = kor

> -ue : eos 사용시 필수 (→ 사용시 줄바꿈이 된 결과값을 받을 수 있음)

> -s 8000 : smaplerate가 8k일 경우 추가 필요



2단계

```
./Train fork /data1/stt_10/baseline/model/005_model_dev-lg.bin -enable_distributed false --logtostderr 1 --flagfile /data1/stt_10/TEST/train.cfg
```

> 경로 : cd /root/wav2letter/build

> fork 이후 경로 = baseline 경로

> --flagfile : train.cfg 경로

여타 에러 없이 Epoch 1 started! 가 나오면 학습 정상 진행

감사합니다