LangCon 2021

# 한국어 음성 인식

## KoSpeech 개발기부터 OpenSpeech 개발기까지

TUNiB 김수환


# Presenter Introduction

# Presenter Introduction



## 👤 Profile

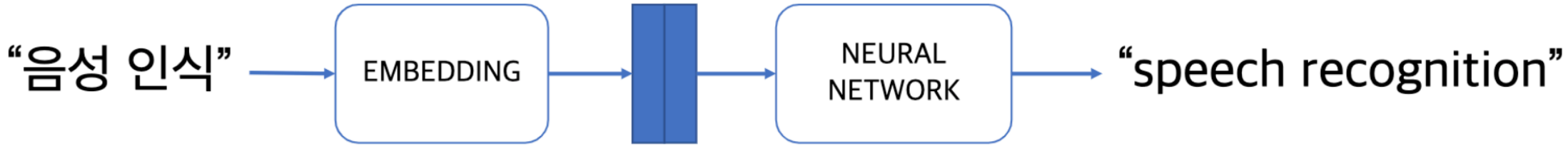
- ***Basic Information***
  - ***Name*** : Soohwan Kim
  - ***Nation*** : Republic of Korea
  - ***Birth*** : 1995.10.11
- ***Professional Information***
  - ***Job*** : AI Research Engineer
  - ***Company*** : TUNiB
- ***Speech Related Projects***
  - Pororo - Multilingual-TTS
  - Pororo - Automatic Speech Recognition
  - OpenSpeech
  - KoSpeech
- ***Social Information***
  - ***GitHub*** : https://github.com/sooftware
  - ***Blog*** : https://blog.naver.com/sooftware
  - ***Facebook*** : https://www.facebook.com/sooftware95/
  - ***Linked-In*** : https://www.linkedin.com/in/Soo-hwan/
  - ***E-mail*** : sh951011@gmail.com


# What is the Speech Recognition?

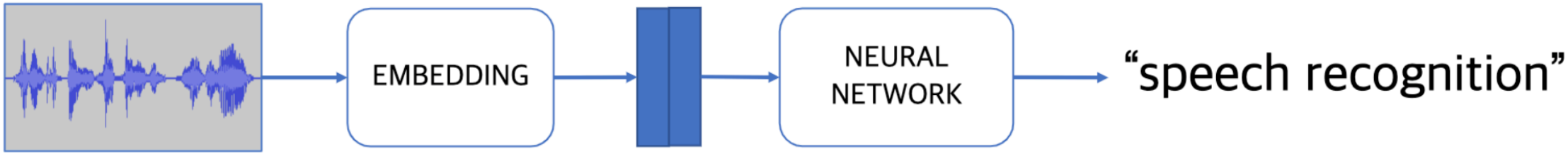
# What is the Speech Recognition?



- 기계 번역




- 음성 인식




사람의 음성으로부터 발화한 텍스트를 얻어내는 기술

# What is the Speech Recognition?

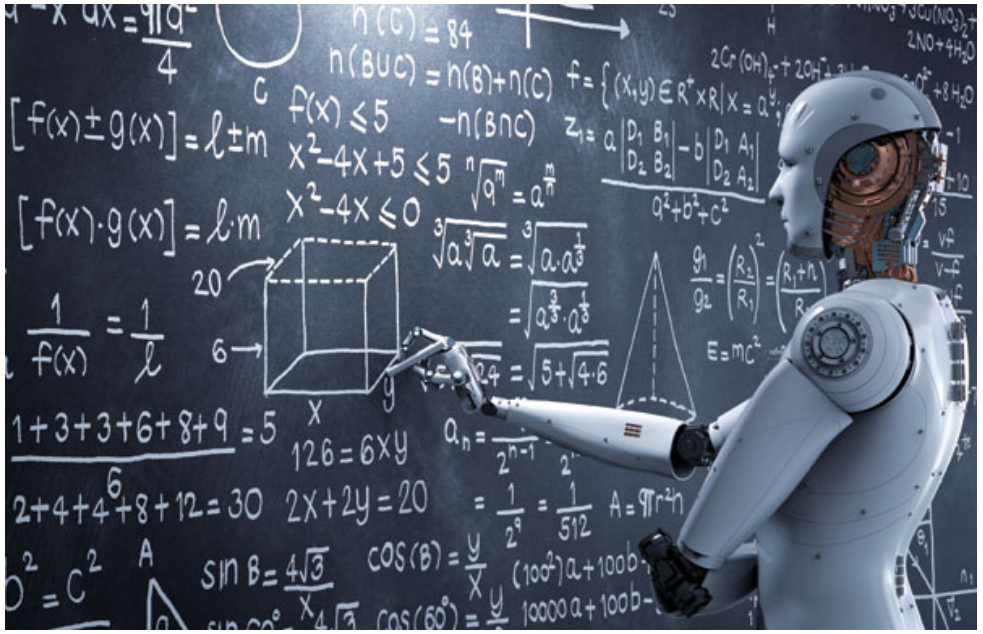



Bixby


이미 일상생활에서 익숙한 음성인식을 활용한 서비스들

# What is the Speech Recognition?





하지만 아직 갈 길이 많이 남은 음성 인식..

# 기능대화 음성인식



사용자가 어느 질문/발화를 할 지 어느정도 **예측 가능함**

**예측 가능한 범위에서 시나리오를 만들거나, 모델을 만들 수 있음**

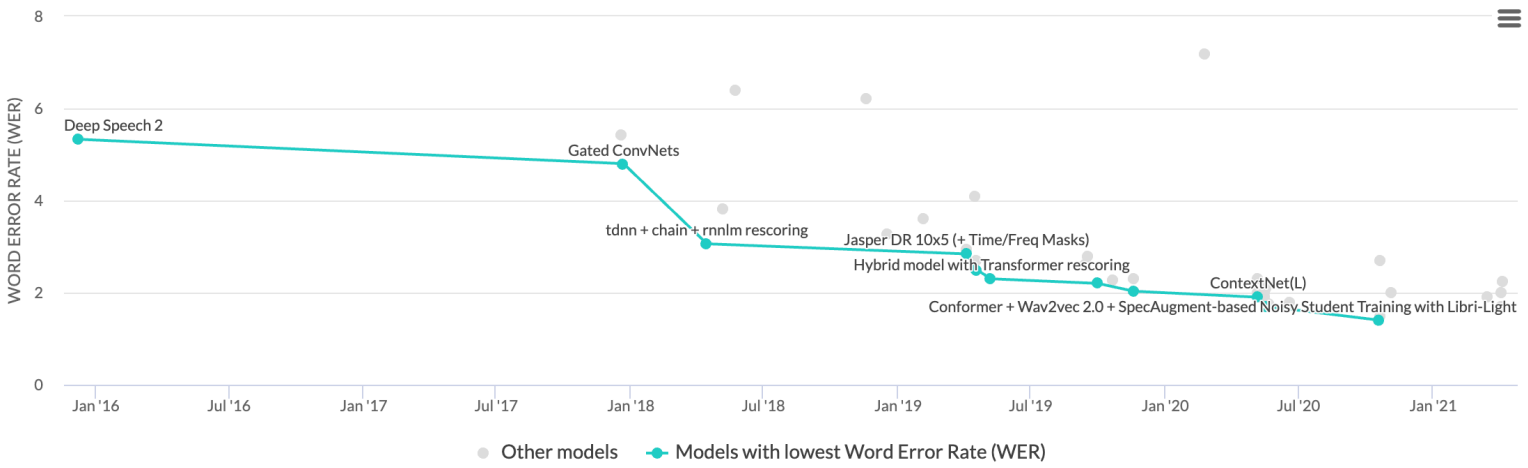
## 일반 음성인식의 기술적 난이도


Input을 예상할 수가 없음

무한한 주제, 무한한 문맥(Context), 노이즈 등에 따른 무한한 경우의 수가 만들어짐

# 대표적인 음성인식 데이터셋

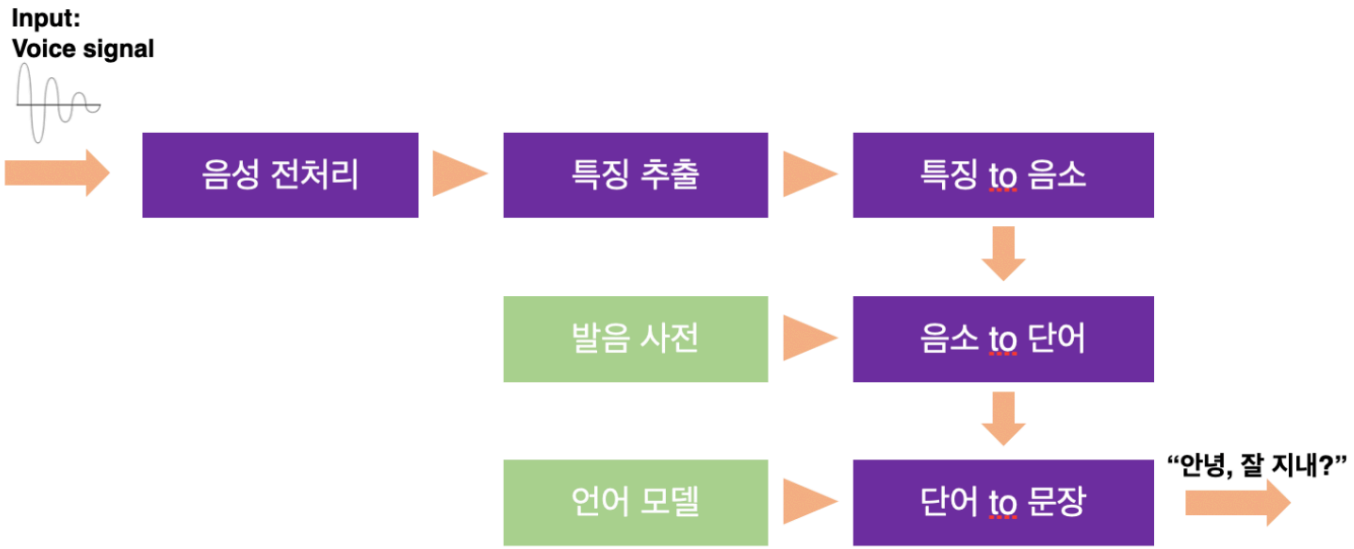
## LibriSpeech Error Rate




최근 딥러닝의 발전으로 상당한 개선이 이루어짐


# End-to-End Speech Recognition

## **Deep Learning Boom** (2012년) 이전의 음성인식 방법




출처: YouTube 강의 - "딥러닝과 음성인식"

**Deep Learning Boom** (2012년) 이후의 음성인식 방법

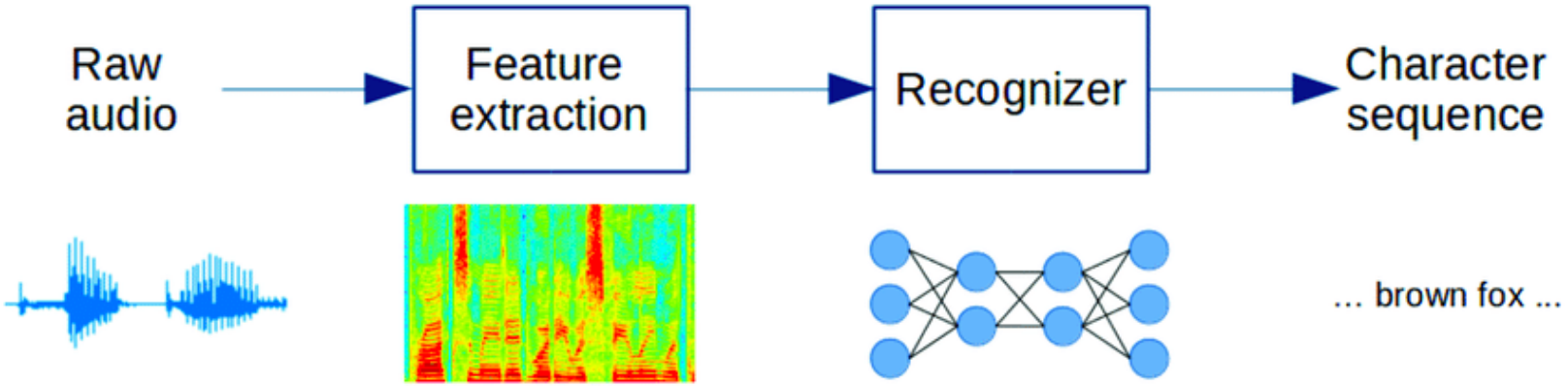
**End-to-End 학습**


출처 : YouTube 강의 “딥러닝과 음성인식”

그냥 통째로 Input, Output 넣어서

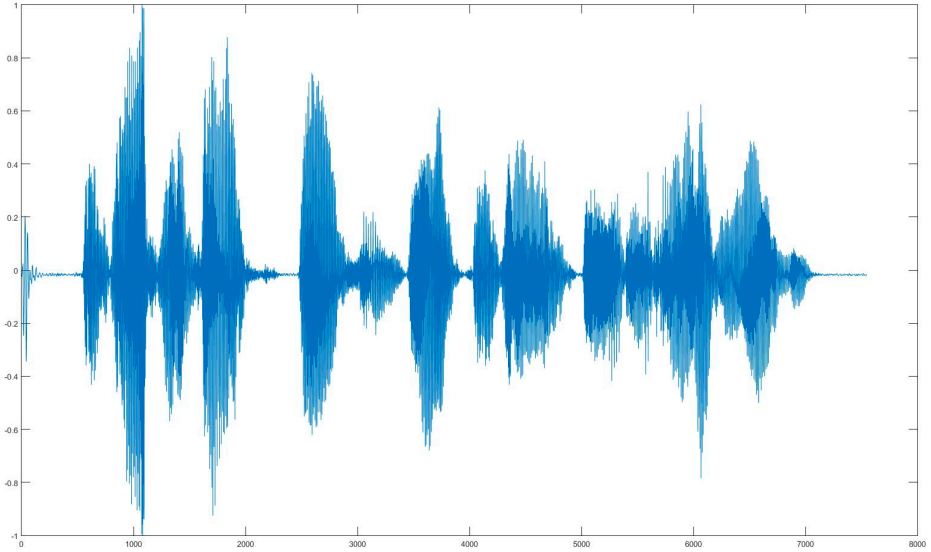
문법과 발음까지 **한꺼번에 모두 학습**시켜 버리자!

**End-to-End** Speech Recognition



# Audio File

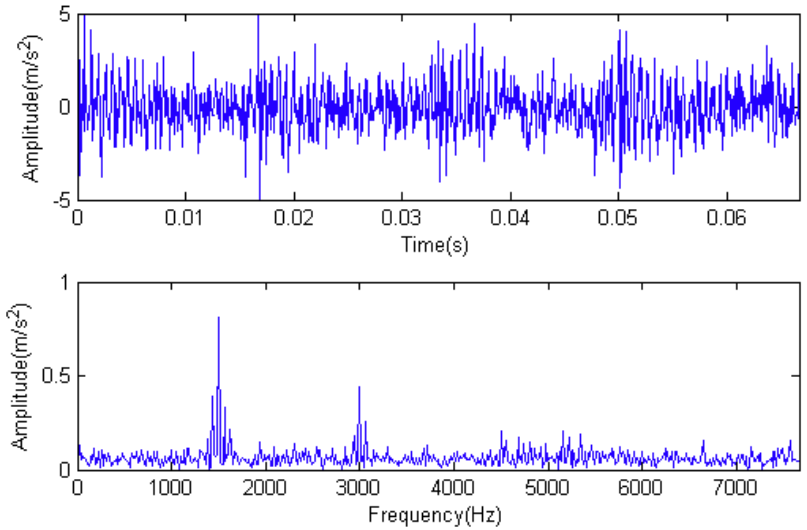
(wav, pcm, flac etc.)



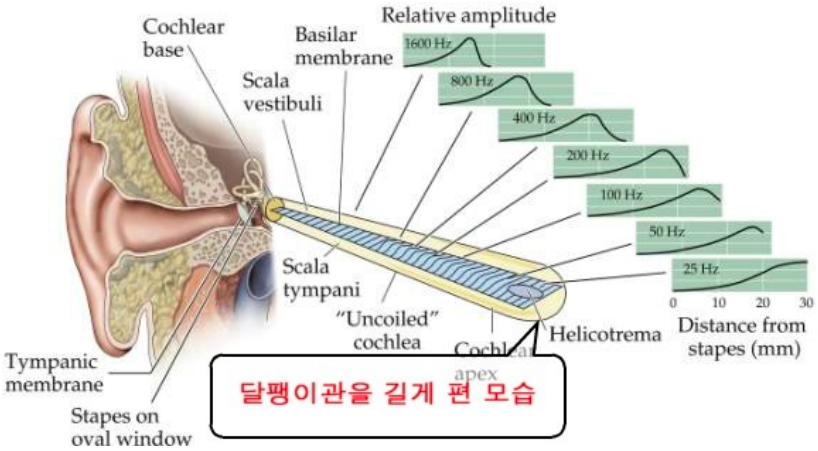

일반적으로 16,000Hz의 샘플링 레이트

Raw한 오디오 신호에서는 많은 정보를 얻을 수 없음

# Feature Extraction



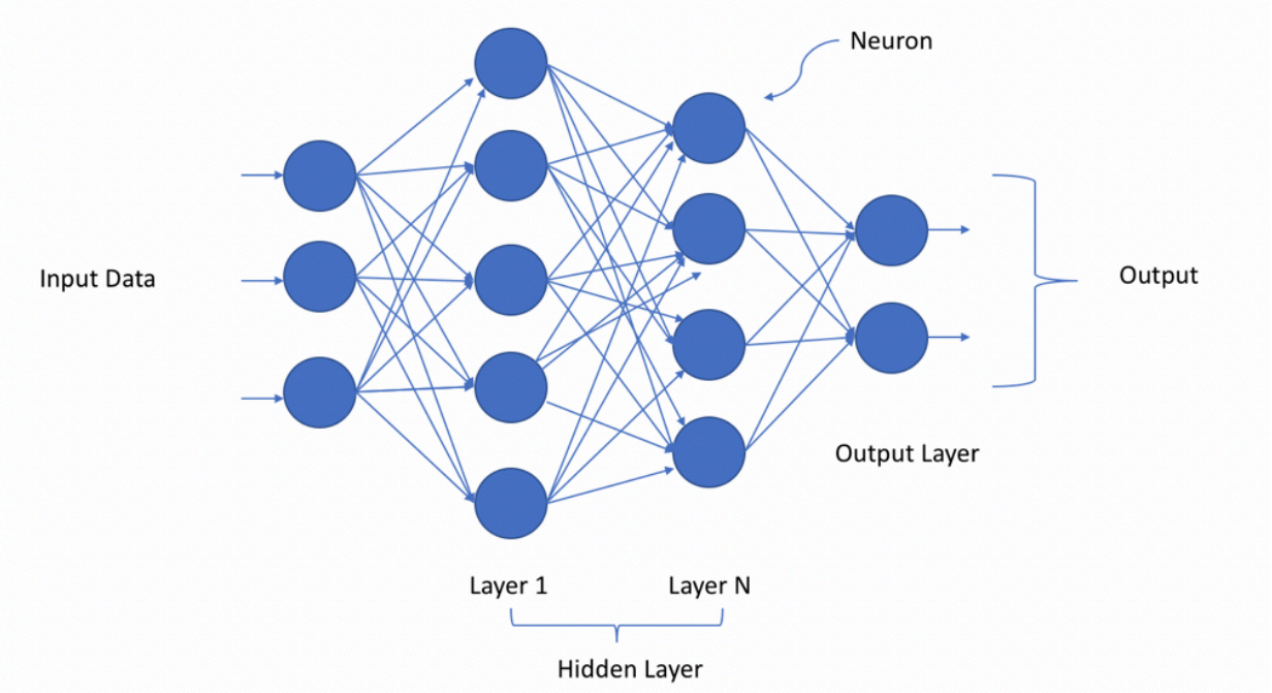

Short-Time Fourier transform
20ms 프레임 / 10ms 겹치도록



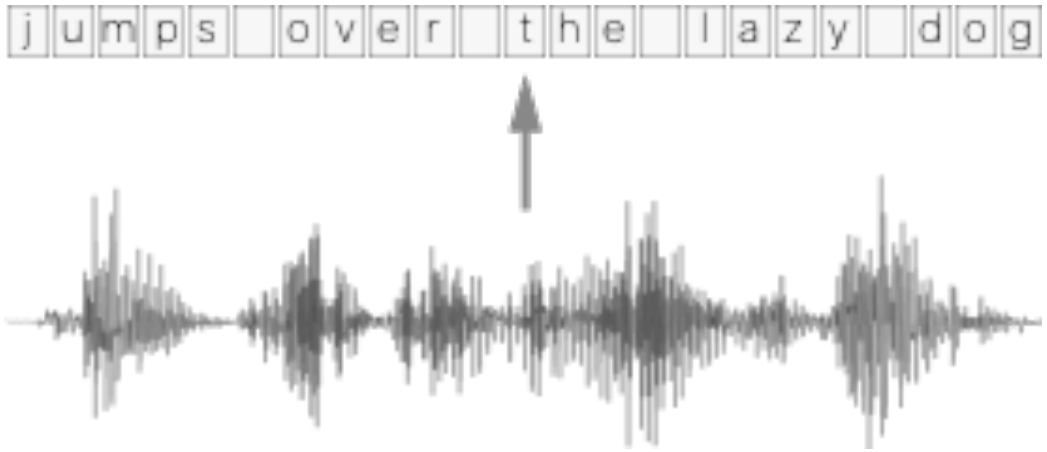

Mel Scale

• "Voice Recognition using MFCC Algorithm" (IJIRAE, 2014)

# Acoustic Modeling

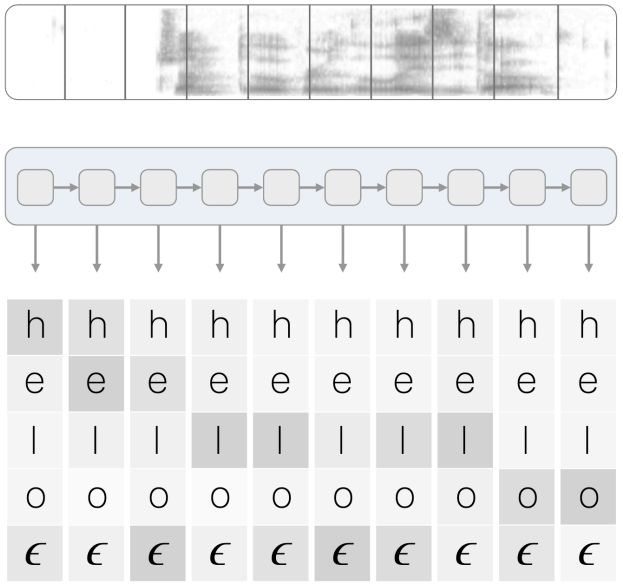


# Acoustic Modeling의 어려움

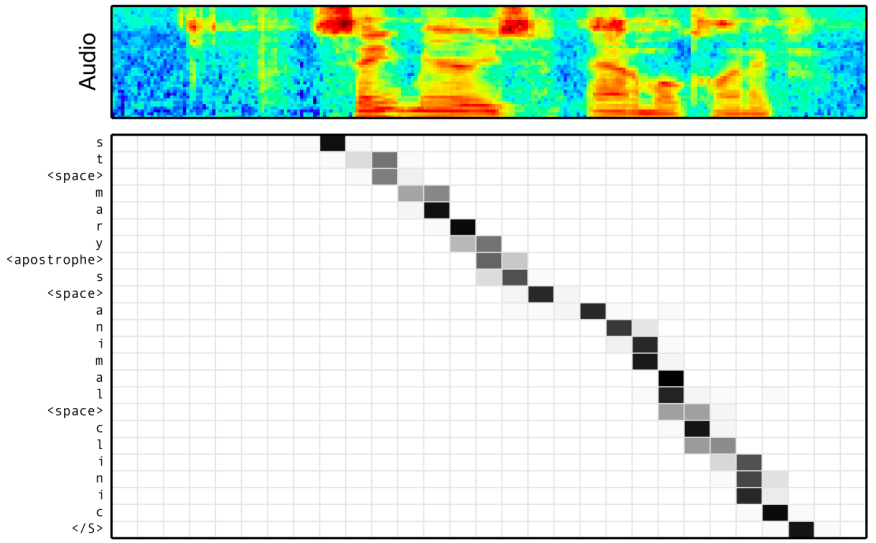



**Signal과 Text의 alignment를 알기 어려움**

# Acoustic Modeling

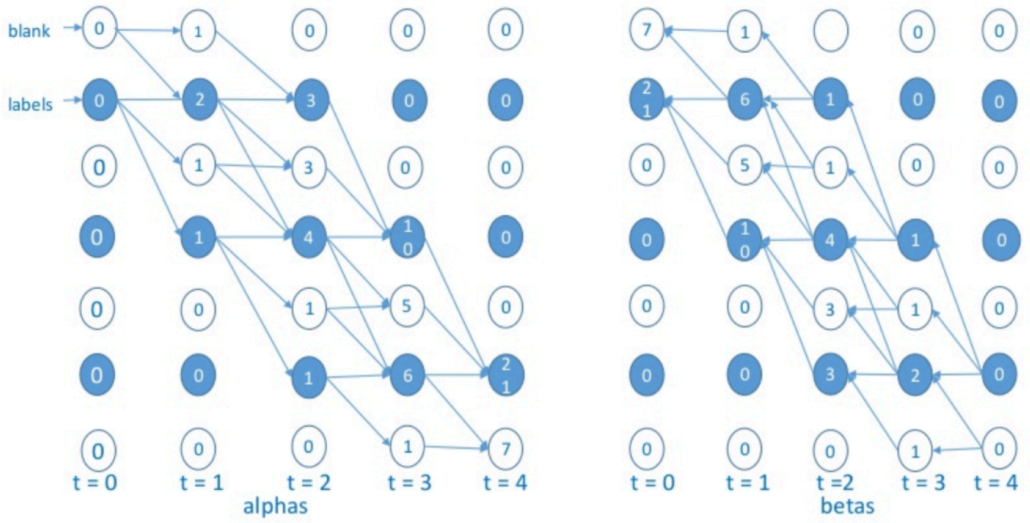



CTC




Attention

## Connectionist Temporal Classification




- CTC는 Loss !!
- 가능한 모든 경로에 대한 loss를 계산해서 합하는 방식

$$p(\mathbf{l}|\mathbf{x}) = \sum_{t=1}^{T} \sum_{s=1}^{|\mathbf{l}|} \frac{\alpha_t(s)\beta_t(s)}{y_{l_s}^t}.$$


- Suhas Pillai, “Intelligent Handwriting Recognition_MIL_presentation_v3_final”, Slideshare
- Alex Graves et al. “Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks”, ICML, 2006

## Connectionist Temporal Classification

P(_ _ T U _ _ _ _ N _ _ _ _ _ _ I I I I _ _ B B _ _ )

+

.

.

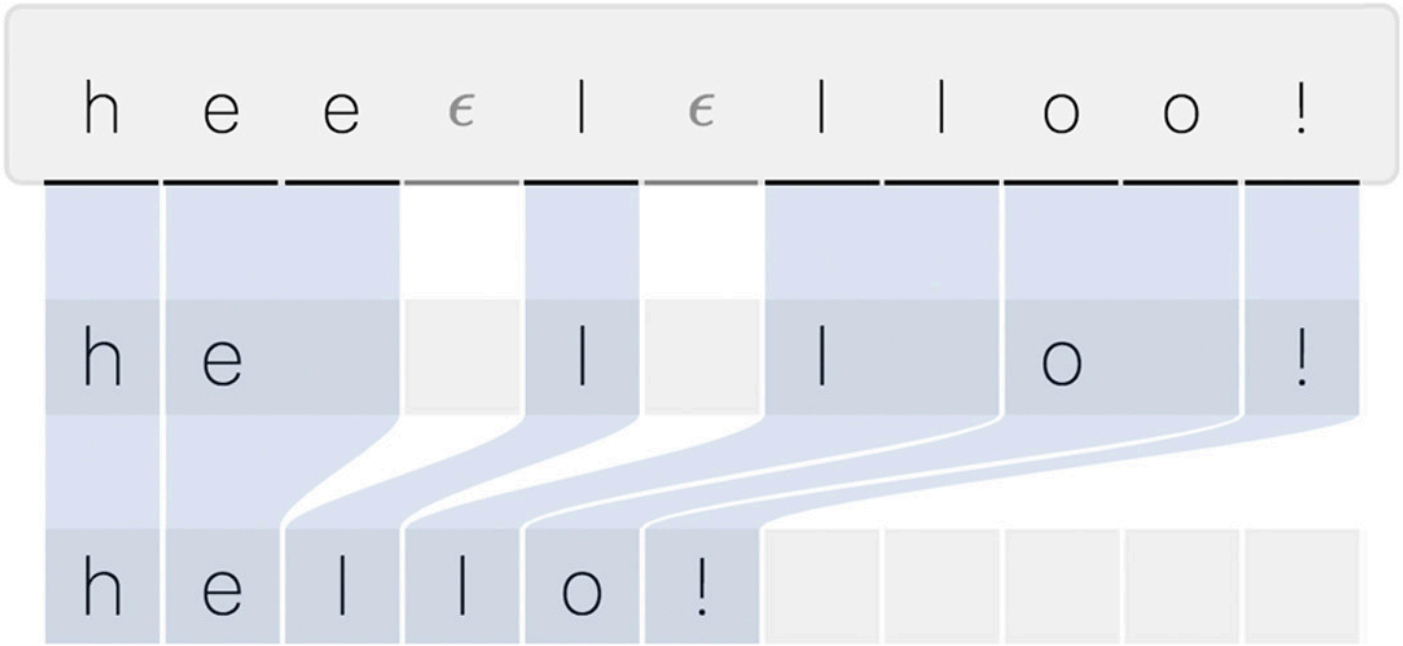
.

+

P(_ _ T _ _ U U _ _ N N N _ _ _ _ I I _ _ B _ _ )

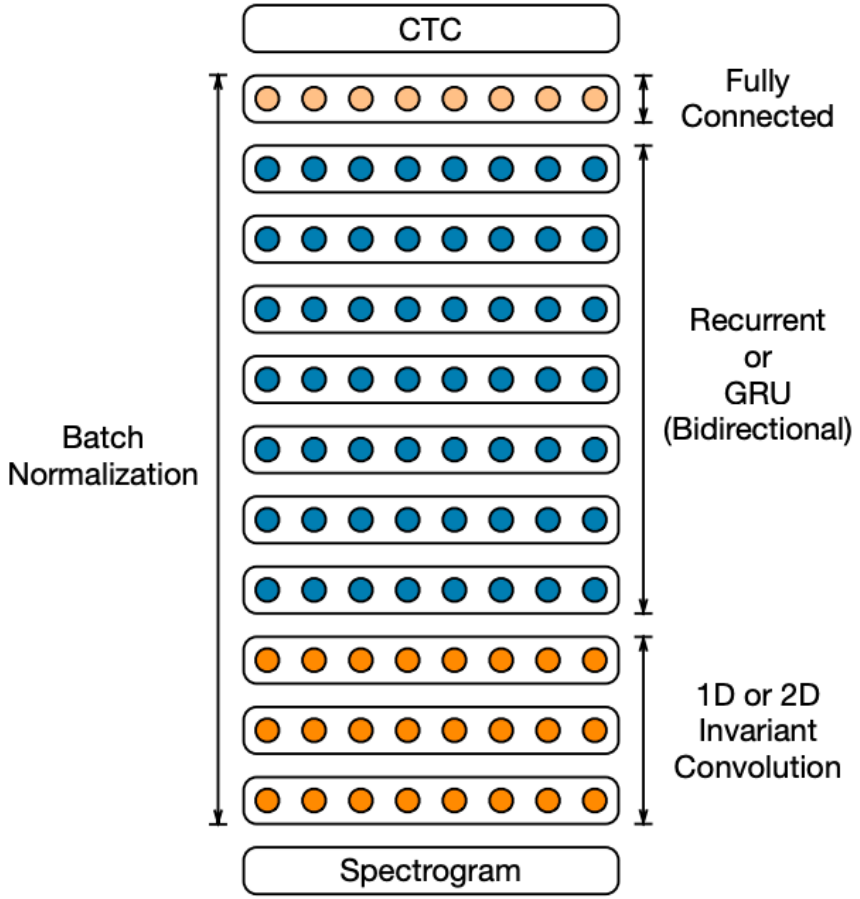
- _ (blank) : 클래스 간의 구분자 역할
- 복잡한 계산, **But 미분 가능 !!**
- Dynamic Programming

## Connectionist Temporal Classification



# End-to-End Speech Recognition



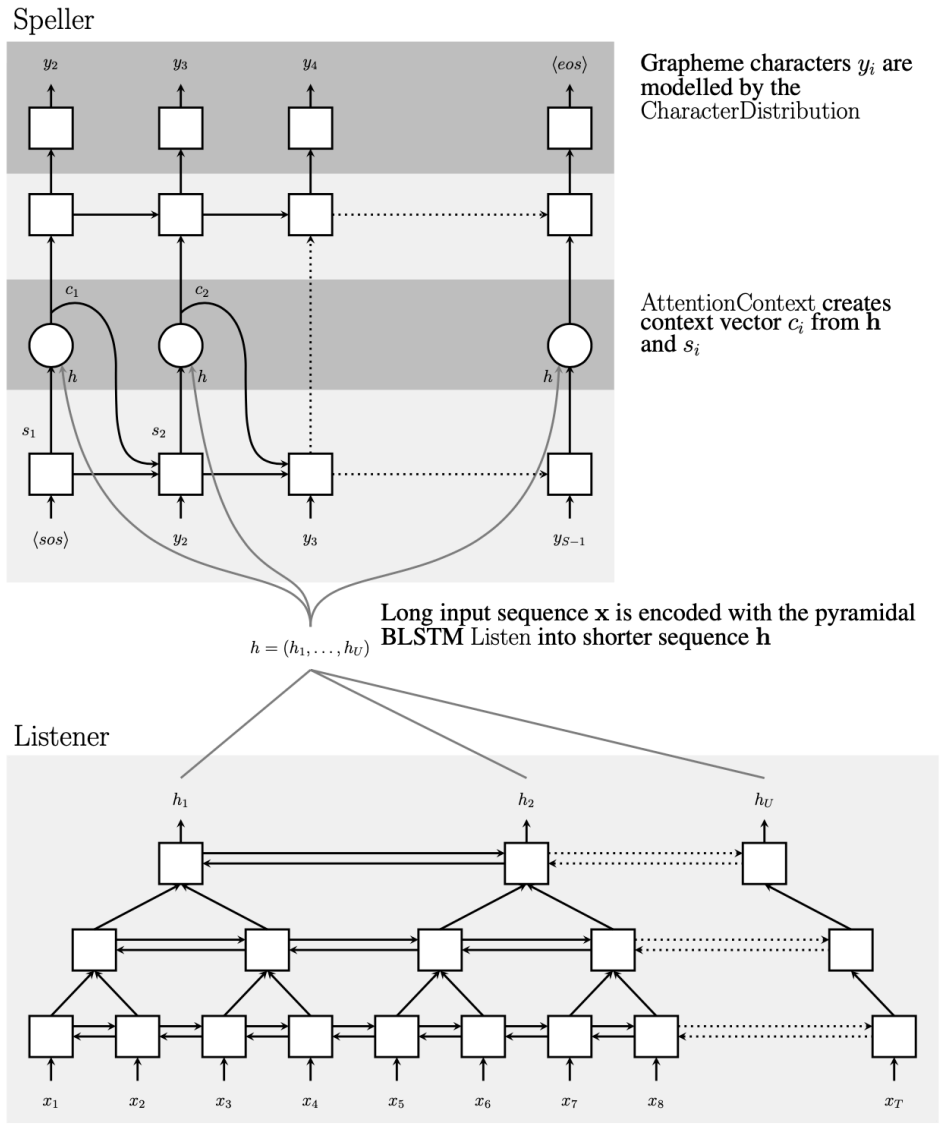



- Baidu 리서치에서 제안
- 대표적인 CTC 기반 모델
- CTC 기반이기 때문에 Auto-regressive 디코더가 없음

• "Deep Speech 2: End-to-End Speech Recognition in English and Mandarin" (ICML, 2016)

# End-to-End Speech Recognition

Language Conference

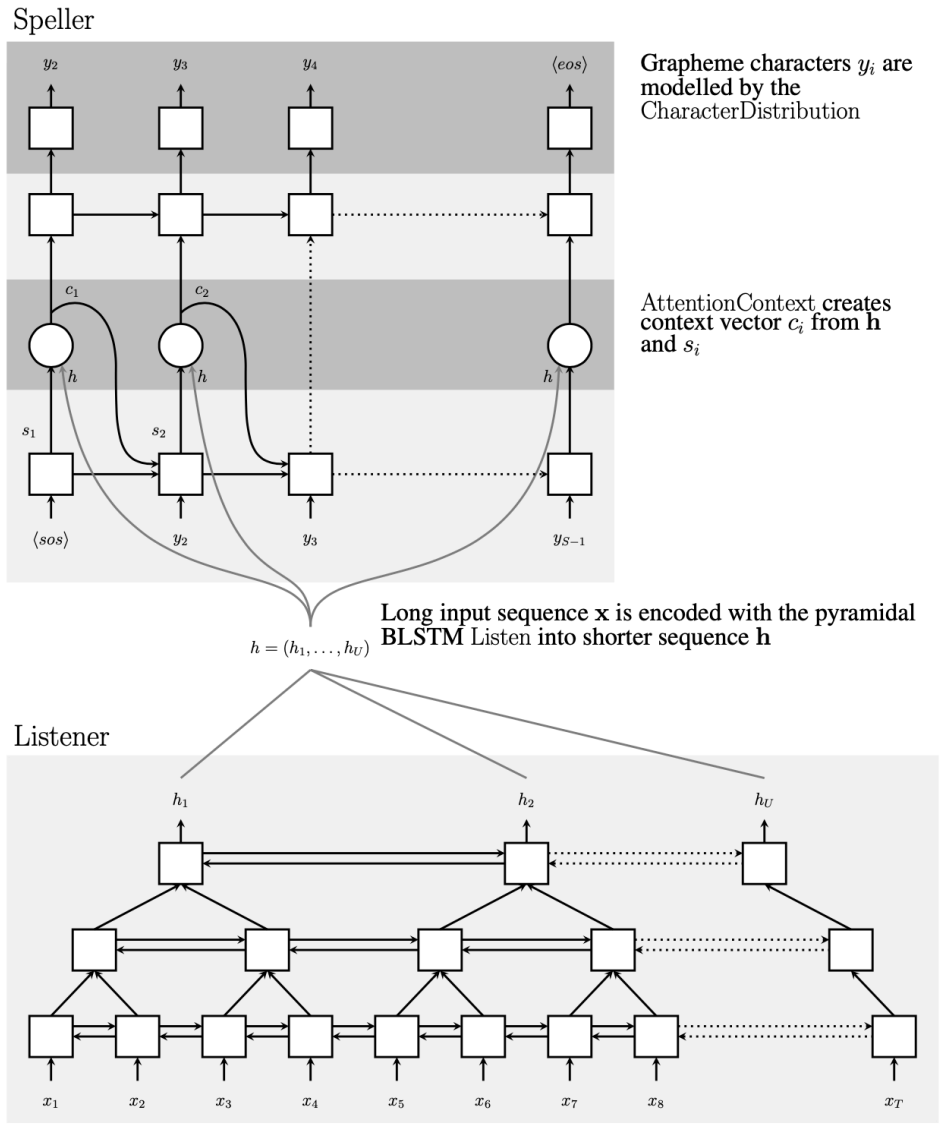


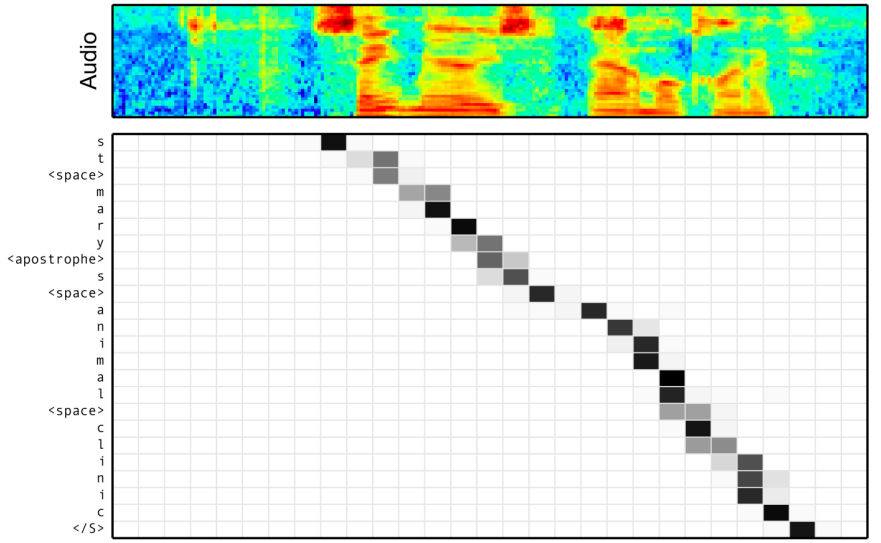

- 자연어처리 분야에서 제안된 RNN 기반 Seq2seq를 음성인식에 적용
- Cross-Entropy Loss Function 사용
- Listener: 인코더로 음성 신호의 피쳐를 High-level 피쳐로 변환해주는 역할
- Speller: Attention 기반 디코더. Listener의 피쳐를 받아서 글자를 뱉어주는 역할

• "Listen, Attend and Spell" (ICASSP, 2016)

# End-to-End Speech Recognition



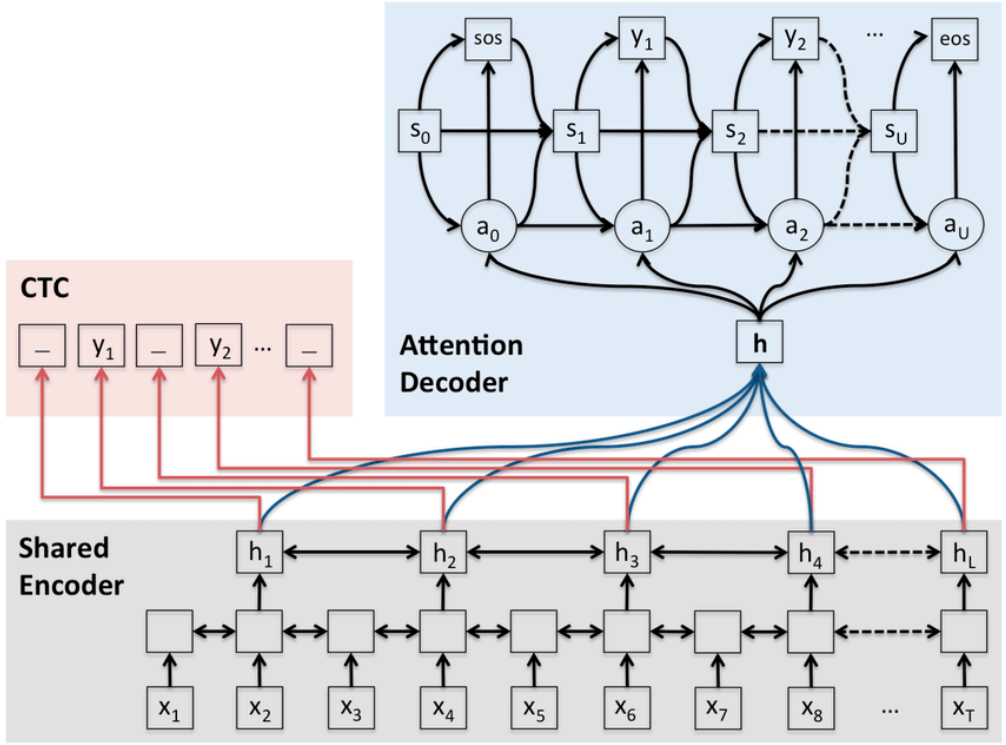



- 어텐션이 Signal과 Text의 alignment를 잡아줌
- 해당 논문 발표 이후 다양한 어텐션 매커니즘 등의 변형들을 적용한 모델들이 출현하며 음성인식 발전에 큰 기여

• "Listen, Attend and Spell" (ICASSP, 2016)

## Joint CTC-Attention




- CTC 방식과 LAS 방식을 합친 방식 제안
- 인코더에 CTC, Cross-Entropy(CE) Loss 모두 사용
- 인코더가 더 Robust해지고 더 빠르게 수렴하는 경향을 보임
- 인코더-디코더 아키텍처에 적용할 수 있기 때문에 널리 사용되고 있음

• "Joint CTC-Attention based End-to-End Speech Recognition using Multi-Task Learning" (ICASSP, 2017)

# Metric

Character Error Rate (CER)

**Edit Distance**


substitution = 2

deletion = 1

insertion = 1

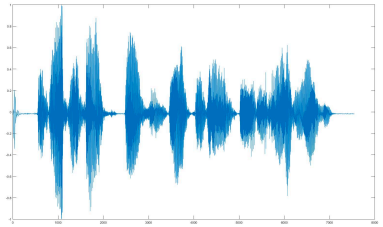
$$ED = \frac{2+1+1}{\text{length of Ref.}} = \frac{4}{8} = 0.5$$

https://en.wikipedia.org/wiki/Levenshtein_distance

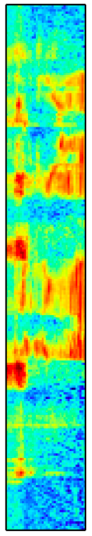
# End-to-End Speech Recognition

Language Conference

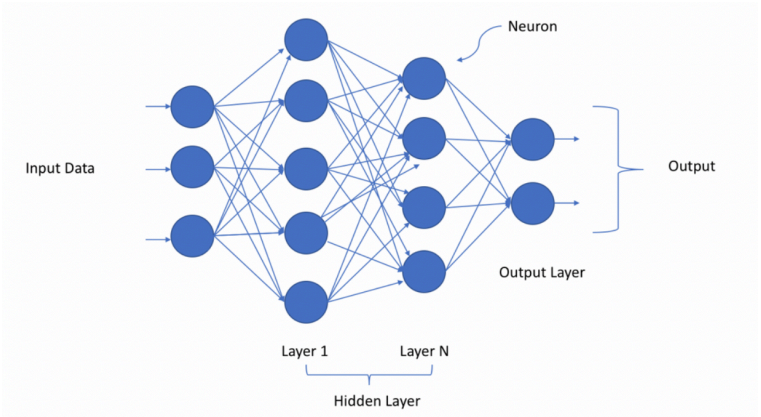
Raw Audio



Feature



Model




Transcript

아 뭐 뭔 소리야 그건 또


# Korean Speech Recognition

Open Access Article

# KsponSpeech: Korean Spontaneous Speech Corpus for Automatic Speech Recognition

by Jeong-Uk Bang, Seung Yun *, Seung-Hi Kim, Mu-Yeol Choi, Min-Kyu Lee, Yeo-Jeong Kim, Dong-Hyun Kim, Jun Park, Young-Jik Lee and Sang-Hun Kim

Artificial Intelligence Research Laboratory, Electronics and Telecommunications Research Institute (ETRI), 218 Gajeong-ro, Yuseong-gu, Daejeon 34129, Korea

* Author to whom correspondence should be addressed.

*Appl. Sci.* **2020**, *10*(19), 6936; https://doi.org/10.3390/app10196936

**Received: 27 August 2020 / Revised: 28 September 2020 / Accepted: 29 September 2020 / Published: 3 October 2020**

(This article belongs to the Section Computing and Artificial Intelligence)

- "KsponSpeech: Korean Spontaneous Speech Corpus for Automatic Speech Recognition" (MDPI, 2020)

## Data Analysis

(a) Dual transcription (orthography/pronunciation)
- 너 혹시 (컴퓨터/컴퓨타)에 대해 뭐 잘 알아?
- *Do you know a lot about (computers/computars)?*

(b) Filler word symbol ('/')
- 어/ 자세히 보면은 걔가 제일 요행을 바래.
- *Uh/, if you look closely, he wants luck the most.*

(c) Repeated word symbol ('+')
- 어/ 나+ 나는 작년에 제주도를 두 번이나 갔거든?
- *Uh/, I'm+, I went to Jeju twice last year.*

(d) Ambiguous pronunciation symbol ('*')
- 맞아. 그러니까* 드라마로도 나오고 영화로도 나오는 거지.
- *That's right. That's why* they are released as dramas and movies.*

(e) Non-speech event symbols ('b/':breath, 'l/':laughter, 'o/':overlapped utterance, 'n/':noise)
- 진짜 맛있어. l/ 내가 요즘에 가장 좋아하는 과자야. b/
- *It's really good. [laughter] That's one of my favorite snacks recently. [breath]*

(f) Numeric notation (numbers and units/pronunciation)
- 그리고 또 KFC는 이제 (9시/아홉 시) 지나면은 치킨이 원 플러스 원하니까.
- *And in KFC, if it's past (9 o'clock/nine o'clock), you can buy one and get one free.*

## Data Analysis

- Raw Data

```
"b/ 아/ 모+ 몬 소리야 (70%)/(칠 십 퍼센트) 확률이라니 n/"
```

- b/, n/, / .. 등의 잡음 레이블 삭제

```
"아/ 모+ 몬 소리야 (70%)/(칠 십 퍼센트) 확률이라니"
```

- 제공된 (철자전사)/(발음전사) 중 발음전사 사용

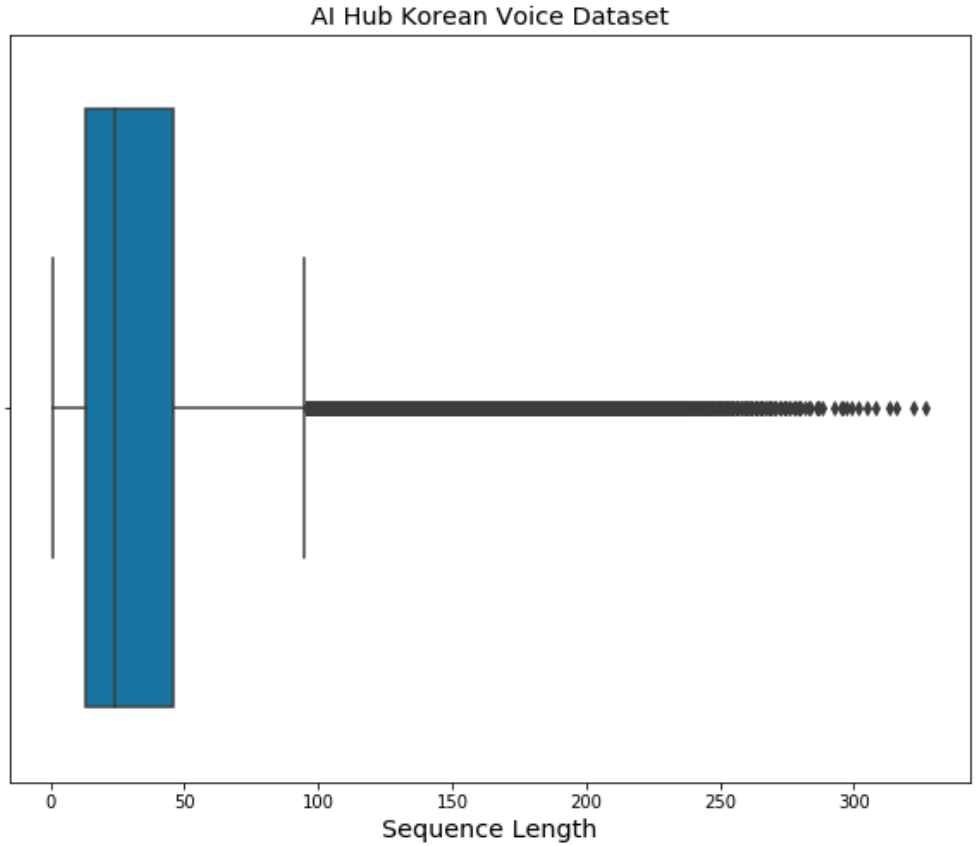
```
"아/ 모+ 몬 소리야 칠 십 퍼센트 확률이라니"
```

- 간투어 표현 등을 위해 사용된 '/', '*', '+' 등의 레이블 삭제

```
"아 모 몬 소리야 칠 십 퍼센트 확률이라니"
```

- Code: https://github.com/sooftware/ksponspeech

**Data Analysis**




KsponSpeech Box-plot (Text-Length)

## Data Analysis

| id | char | freq |
|---|---|---|
| 0 | | 5774462 |
| 1 | . | 640924 |
| 2 | 그 | 556373 |
| 3 | 이 | 509291 |
| 4 | 는 | 374559 |
| . | . | . |
| 2329 | 갭 | 1 |
| 2330 | 걈 | 1 |
| 2331 | 걕 | 1 |
| 2332 | 갗 | 1 |

| id | char | freq |
|---|---|---|
| 0 | | 5774462 |
| 1 | . | 640924 |
| 2 | 그 | 556373 |
| 3 | 이 | 509291 |
| 4 | 는 | 374559 |
| . | . | . |
| 2032 | 꼳 | 2 |
| 2033 | 꼄 | 2 |
| 2034 | 껖 | 2 |
| 2035 | 껀 | 2 |

- 등장하는 문자 분석
- 1번씩만 등장하는 300개의 문자 삭제
- Vocal Size ≈ 2,000

## Data Format

**오디오 경로 [TAB] 한글 전사 [TAB] ID 전사**

KsponSpeech_01\KsponSpeech_0001\KsponSpeech_000001.pcm [TAB] 아 몬 소리야 그건 또. [TAB] 9 4 727 4 174 34 28 4 6 101 4 128 5
KsponSpeech_01\KsponSpeech_0001\KsponSpeech_000003.pcm [TAB] 그래서 지호랑 계단 올라와서 막 위에 운동하는 기구 있대요. [TAB] 6 36 20 … 20 4
KsponSpeech_01\KsponSpeech_0001\KsponSpeech_000005.pcm [TAB] 그게 영 점 일 프로 가정의 아이들과 가정의 모습이야? [TAB] 6 23 4 154 … 368 7 28 16
…
…

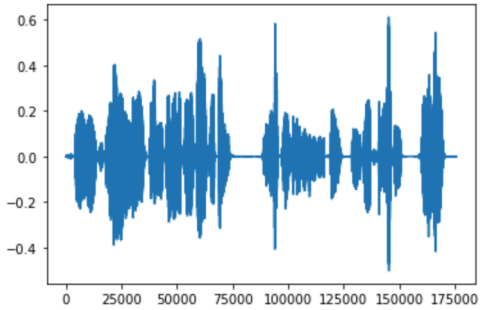
## Feature Extraction by librosa

- librosa를 통해 쉽게 오디오 로드 및 피쳐 추출 가능
- Mel-Spectrogram, MFCC 등의 다양한 피쳐 추출 가능

```
import librosa
import numpy as np
import matplotlib.pyplot as plt

signal, sample_rate = librosa.load('../557.wav', sr=16000)

plt.plot(signal)
plt.show()
```



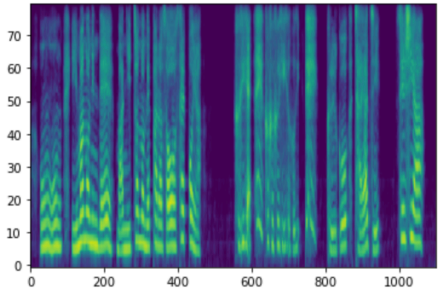

```
frame_length = 20.0
frame_shift = 10.0
n_mels = 80

n_fft = int(round(sample_rate * 0.001 * frame_length))
hop_length = int(round(sample_rate * 0.001 * frame_shift))

spectrogram = librosa.feature.melspectrogram(signal, sr=sample_rate, hop_length=hop_length, n_fft=n_fft, n_mels=n_mels)
spectrogram = librosa.power_to_db(spectrogram, ref=np.max)

plt.imshow(spectrogram, aspect='auto', origin='lower')
```

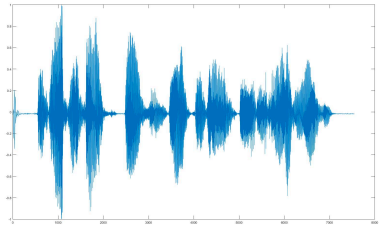
```
<matplotlib.image.AxesImage at 0x7fc7b02bad90>
```



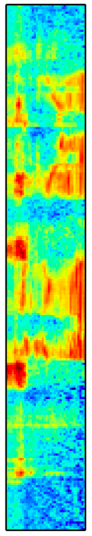
# Korean Speech Recognition

Language Conference

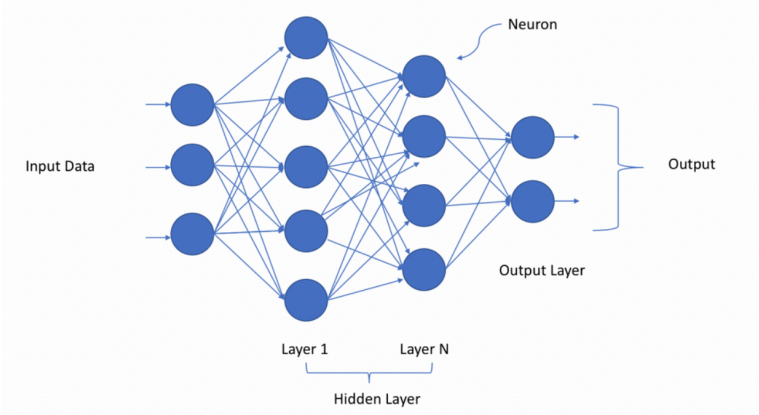
Raw Audio



Feature



Model




Transcript

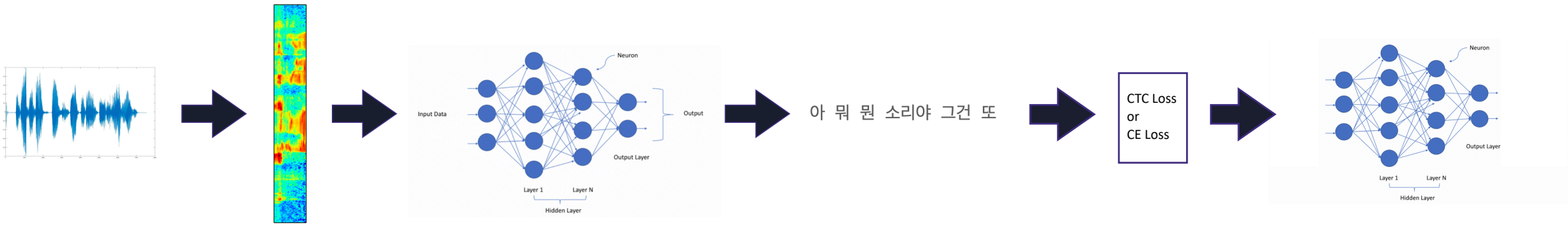
아 뭐 뭔 소리야 그건 또

하지만 작년 초만 하더라도 End-to-End 방식의 한국어 음성인식 오픈소스가 없던 상황


# KoSpeech & OpenSpeech

## Speech Recognition Training Pipeline

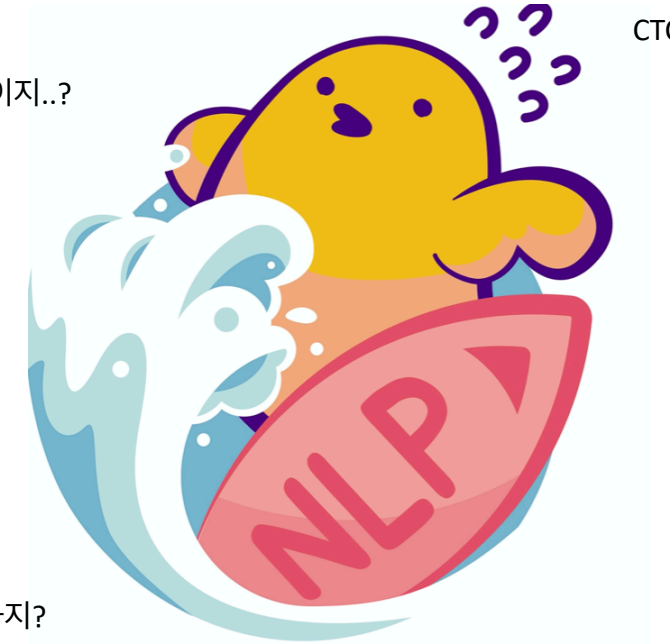
Raw Audio → Feature → Model → Transcript → Loss → Parameter Update




모델이 충분히 수렴할 때까지 반복

Loss가 왜 nan이지..?

CTC 기반 모델은 디코더가 어디갔지..?

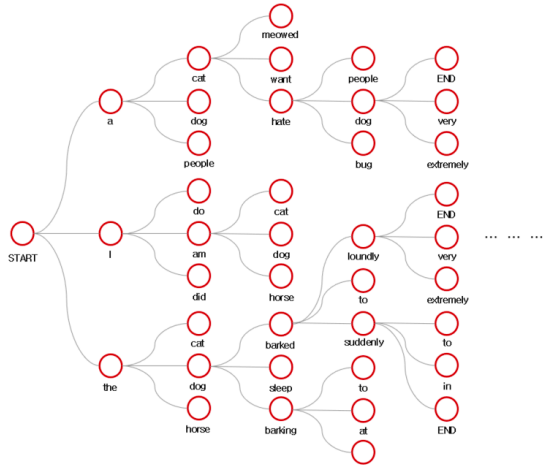



메모리가 자꾸 터지는데 어떡하지..

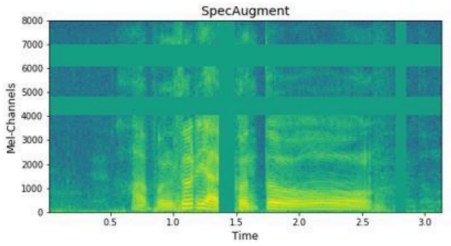
하이퍼 파라미터는 어떻게 정하지?

많은 삽질과 공부

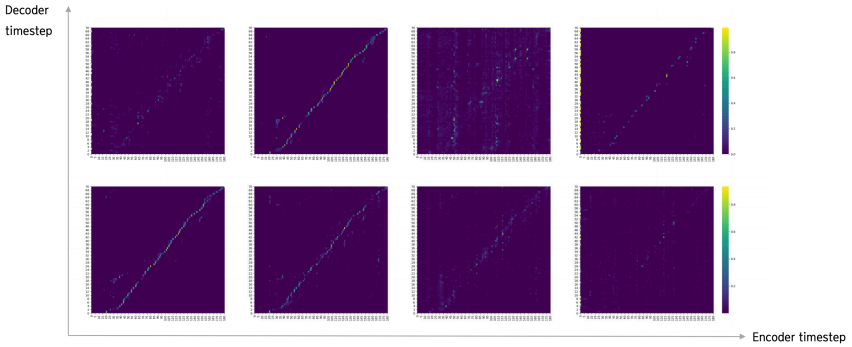
공부자료: https://github.com/sooftware/Speech-Recognition-Tutorial



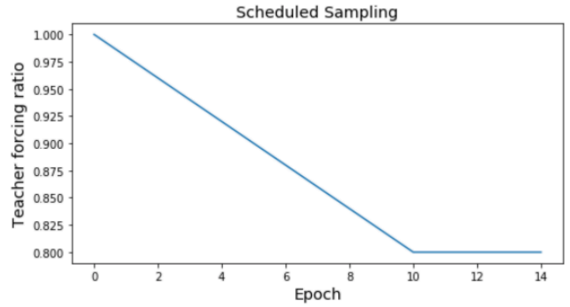

Beam Search



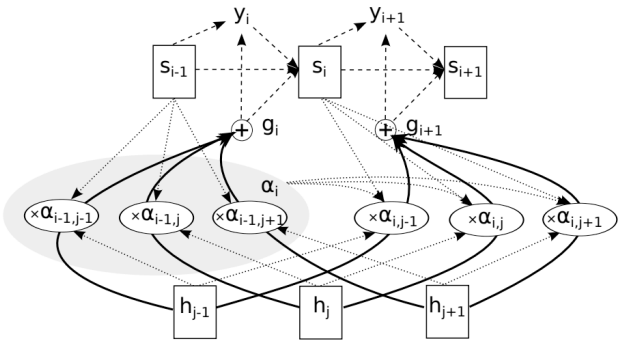

SpecAugment



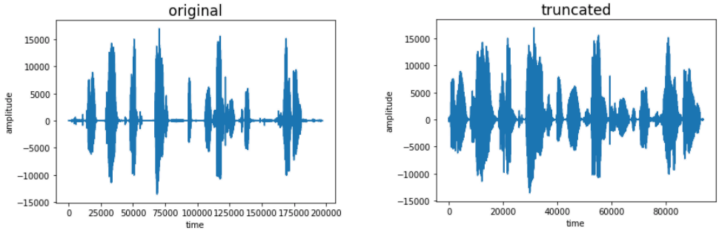

Multi-head Attention
&
Attention Map Visualization



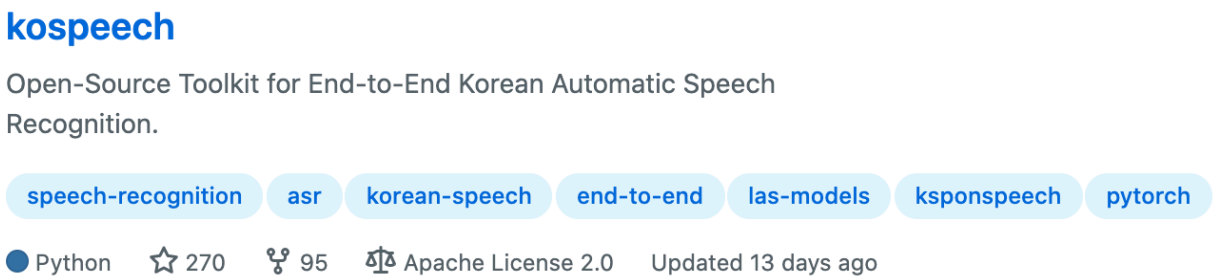

Scheduled Sampling




Location-aware Attention




Delete Silence




Spectrogram, Mel-Spectrogram, MFCC, Filter-Bank 피쳐 지원

LAS, Transformer, DeepSpeech2, Jasper, Conformer 등의 모델 지원

Character / Subword / Grapheme 전처리 및 학습 지원

Etc.

[GitHub] : https://github.com/sooftware/kospeech

[TECHNICAL-REPORT] : https://arxiv.org/abs/2009.03092

[PAPER] : https://www.sciencedirect.com/science/article/pii/S2665963821000026

# KoSpeech

**An Apache 2.0 ASR research library, built on PyTorch, for developing end-to-end speech recognition models.**

옵션이 많아지면서 사용자 입장에서 파악이 어려워짐

모델 등 모듈 추가 시 다른 코드들도 같이 수정을 해야함 (확장 속도가 더딤)

한국어만 지원

바닐라 파이토치 기반 (Multi-Node Training 등의 어려움)

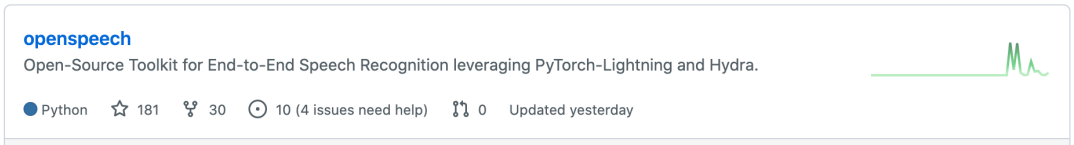

Facebook-Research에서 공개한 configuration 관리 프레임워크인 Hydra 적용 및 전체적인 구조 수정을 했지만,

기존에 잘 돌아가던 코드에서 예기치 못한 버그가 생기는 등의 어려움이 있었음

- https://github.com/facebookresearch/hydra


- 여러 언어 레시피 지원 (한국어, 영어, 중국어)
- PyTorch-Lightning 기반
- 모델 및 기타 모듈을 추가하기 쉬운 구조
- 더 많은 모델 지원
- 추상화 극대화

OPENSPEECH




- https://github.com/openspeech-team/openspeech

PyTorch Lightning

+


많은 configuration을 hierarchical하게 관리하면서 확장성이 큰 구조를 위해 PyTorch-Lightning + Hydra 적용
Fairseq에서 Hydra를 적용한 구조를 많이 참고

- Fairseq: https://github.com/pytorch/fairseq
- Openspeech' s Hydra: https://openspeech-team.github.io/openspeech/notes/hydra_configs.html

## Supported Recipe

KsponSpeech
1,000h

LibriSpeech
1,000h

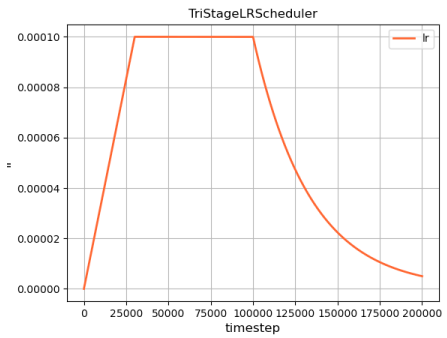
AISHELL-1
100h

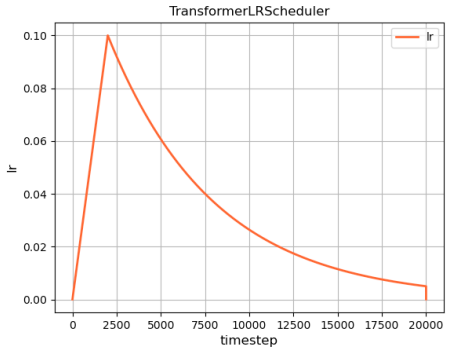
- "KsponSpeech: Korean Spontaneous Speech Corpus for Automatic Speech Recognition" (MDPI, 2020)
- "LibriSpeech: An ASR Corpus Based on Public Domain Audio Books" (ICASSP, 2015)
- "AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline" (O-COCOSDA, 2017)

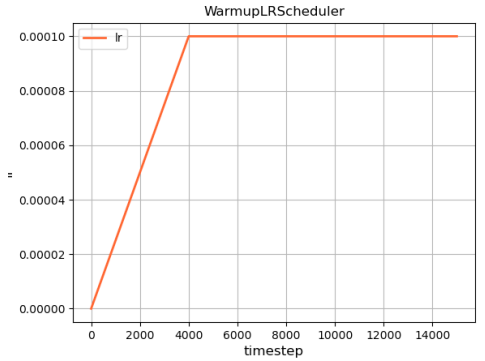
# Supported Models

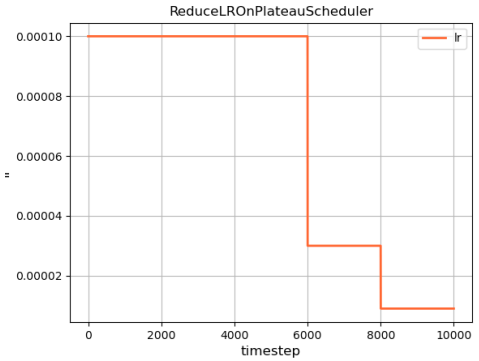
1. **DeepSpeech2** (from Baidu Research) released with paper Deep Speech 2: End-to-End Speech Recognition in English and Mandarin, by Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Hannun, Billy Jun, Patrick LeGresley, Libby Lin, Sharan Narang, Andrew Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang, Bo Xiao, Dani Yogatama, Jun Zhan, Zhenyao Zhu.
2. **RNN-Transducer** (from University of Toronto) released with paper Sequence Transduction with Recurrent Neural Networks, by Alex Graves.
3. **LSTM Language Model** (from RWTH Aachen University) released with paper LSTM Neural Networks for Language Modeling, by Martin Sundermeyer, Ralf Schluter, and Hermann Ney.
4. **Listen Attend Spell** (from Carnegie Mellon University and Google Brain) released with paper Listen, Attend and Spell, by William Chan, Navdeep Jaitly, Quoc V. Le, Oriol Vinyals.
5. **Location-aware attention based Listen Attend Spell** (from University of Wrocław and Jacobs University and Universite de Montreal) released with paper Attention-Based Models for Speech Recognition, by Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, Yoshua Bengio.
6. **Joint CTC-Attention based Listen Attend Spell** (from Mitsubishi Electric Research Laboratories and Carnegie Mellon University) released with paper Joint CTC-Attention based End-to-End Speech Recognition using Multi-task Learning, by Suyoun Kim, Takaaki Hori, Shinji Watanabe.
7. **Deep CNN Encoder with Joint CTC-Attention Listen Attend Spell** (from Mitsubishi Electric Research Laboratories and Massachusetts Institute of Technology and Carnegie Mellon University) released with paper Advances in Joint CTC-Attention based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM, by Takaaki Hori, Shinji Watanabe, Yu Zhang, William Chan.
8. **Multi-head attention based Listen Attend Spell** (from Google) released with paper State-of-the-art Speech Recognition With Sequence-to-Sequence Models, by Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, Michiel Bacchiani.
9. **Speech-Transformer** (from University of Chinese Academy of Sciences and Institute of Automation and Chinese Academy of Sciences) released with paper Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition, by Linhao Dong; Shuang Xu; Bo Xu.
10. **VGG-Transformer** (from Facebook AI Research) released with paper Transformers with convolutional context for ASR, by Abdelrahman Mohamed, Dmytro Okhonko, Luke Zettlemoyer.
11. **Transformer with CTC** (from NTT Communication Science Laboratories, Waseda University, Center for Language and Speech Processing, Johns Hopkins University) released with paper Improving Transformer-based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration, by Shigeki Karita, Nelson Enrique Yalta Soplin, Shinji Watanabe, Marc Delcroix, Atsunori Ogawa, Tomohiro Nakatani.
12. **Joint CTC-Attention based Transformer**(from NTT Corporation) released with paper Self-Distillation for Improving CTC-Transformer-based ASR Systems, by Takafumi Moriya, Tsubasa Ochiai, Shigeki Karita, Hiroshi Sato, Tomohiro Tanaka, Takanori Ashihara, Ryo Masumura, Yusuke Shinohara, Marc Delcroix.
13. **Transformer Language Model** (from Amazon Web Services) released with paper Language Models with Transformers, by Chenguang Wang, Mu Li, Alexander J. Smola.
14. **Jasper** (from NVIDIA and New York University) released with paper Jasper: An End-to-End Convolutional Neural Acoustic Model, by Jason Li, Vitaly Lavrukhin, Boris Ginsburg, Ryan Leary, Oleksii Kuchaiev, Jonathan M. Cohen, Huyen Nguyen, Ravi Teja Gadde.
15. **QuartzNet** (from NVIDIA and Univ. of Illinois and Univ. of Saint Petersburg) released with paper QuartzNet: Deep Automatic Speech Recognition with 1D Time-Channel Separable Convolutions, by Samuel Kriman, Stanislav Beliaev, Boris Ginsburg, Jocelyn Huang, Oleksii Kuchaiev, Vitaly Lavrukhin, Ryan Leary, Jason Li, Yang Zhang.
16. **Transformer Transducer** (from Facebook AI) released with paper Transformer-Transducer: End-to-End Speech Recognition with Self-Attention, by Ching-Feng Yeh, Jay Mahadeokar, Kaustubh Kalgaonkar, Yongqiang Wang, Duc Le, Mahaveer Jain, Kjell Schubert, Christian Fuegen, Michael L. Seltzer.
17. **Conformer** (from Google) released with paper Conformer: Convolution-augmented Transformer for Speech Recognition, by Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, Ruoming Pang.
18. **Conformer with CTC** (from Northwestern Polytechnical University and University of Bordeaux and Johns Hopkins University and Human Dataware Lab and Kyoto University and NTT Corporation and Shanghai Jiao Tong University and Chinese Academy of Sciences) released with paper Recent Developments on ESPNET Toolkit Boosted by Conformer, by Pengcheng Guo, Florian Boyer, Xuankai Chang, Tomoki Hayashi, Yosuke Higuchi, Hirofumi Inaguma, Naoyuki Kamo, Chenda Li, Daniel Garcia-Romero, Jiatong Shi, Jing Shi, Shinji Watanabe, Kun Wei, Wangyou Zhang, Yuekai Zhang.
19. **Conformer with LSTM Decoder** (from IBM Research AI) released with paper On the limit of English conversational speech recognition, by Zoltán Tüske, George Saon, Brian Kingsbury.
20. **ContextNet** (from Google) released with paper ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context, by Wei Han, Zhengdong Zhang, Yu Zhang, Jiahui Yu, Chung-Cheng Chiu, James Qin, Anmol Gulati, Ruoming Pang, Yonghui Wu.

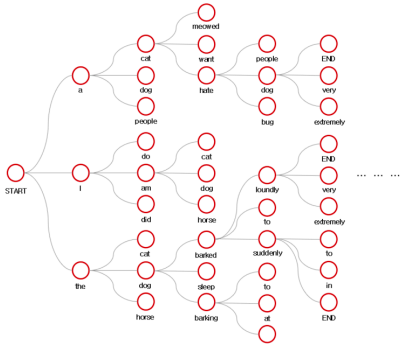
# Supported Modules

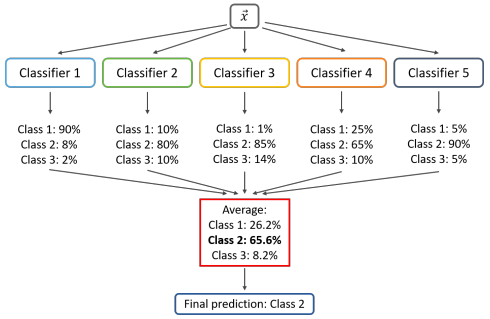


## Training Command

- Example1: Train the `conformer-lstm` model with `filter-bank` features on GPU.

```
$ python ./openspeech_cli/hydra_train.py \
    dataset=librispeech \
    dataset.dataset_download=True \
    dataset.dataset_path=$DATASET_PATH \
    dataset.manifest_file_path=$MANIFEST_FILE_PATH \
    tokenizer=libri_subword \
    model=conformer_lstm \
    audio=fbank \
    lr_scheduler=warmup_reduce_lr_on_plateau \
    trainer=gpu \
    criterion=cross_entropy
```

- Example2: Train the `listen-attend-spell` model with `mel-spectrogram` features On TPU:

```
$ python ./openspeech_cli/hydra_train.py \
    dataset=ksponspeech \
    dataset.dataset_path=$DATASET_PATH \
    dataset.manifest_file_path=$MANIFEST_FILE_PATH \
    dataset.test_dataset_path=$TEST_DATASET_PATH \
    dataset.test_manifest_dir=$TEST_MANIFEST_DIR \
    tokenizer=kspon_character \
    model=listen_attend_spell \
    audio=melspectrogram \
    lr_scheduler=warmup_reduce_lr_on_plateau \
    trainer=tpu \
    criterion=cross_entropy
```

## Add Custom Data Recipe (Korean)

1. 데이터 전처리 코드 작성

- Manifest file: 오디오 경로 [TAP] 한글 전사 [TAP] ID 전사
- Vocab file: 원하는 단위에 맞는 vocab:ID 파일 생성

2. LightningDataModule 정의

- Manifest 파일 파싱 및 train / valid / test 셋 분리 및 데이터 로더 정의
- 예제: https://github.com/openspeech-team/openspeech/blob/main/openspeech/datasets/ksponspeech/lit_data_module.py

Contribution is welcome !!


http://www.tunib.ai/


# End Of Document