A1003:
음성인식 및 기계학습

인공지능연구소, 복합지능연구실

박기영

강사 소개

- 인공지능연구소 복합지능연구실
- 1997~2003: 계산및신경망시스템연구실 석,박사
- 2003~2005: 삼성종합기술원 HCILab, Interaction Lab.
- 2005~: ETRI
    - 2007~: 음성처리연구실, 음성지능연구실, 복합지능연구실
    - 2007~2009: 신성장동력산업용 대용량대화형 분산 내장처리 음성인터페이스 기술개발
    - 2010~2014: 모바일 플랫폼 기반 대화모델 적용 자연어 음성인터페이스 기술개발
    - 2015~2018: 언어학습을 위한 자유발화형 음성대화처리 원천기술 개발
    - 2019~2021: 다중 화자간 대화 음성인식 기술 개발
    - 2019~2028: 준지도학습형 언어지능 원천기술 및 이에 기반한 외국인 지원용 한국어 튜터링 서비스 개발
    - 2022~2026: 다화자 동시처리를 위한 인공지능 기반 대화 모델링 기술 개발

강의내용: Overview

- 음성인식 이론
- 음성인식 실습
- 딥러닝 이론/실습 (Transformer)
- 딥러닝/리눅스 개발환경

강의내용: Overview

- 음성 이론
  - Sampling, FFT, Mel, HMM, GMM, n-gram, AM, LM, wFST, decoder, …
- 딥러닝 이론
  - RNN, LSTM, Transformer, …
- 딥러닝 실무
  - recipe, learning rate, batchsize, …
- 개발 실무
  - terminal, shell,
  - vi, git, tmux, anaconda, jupyter, docker, …

강의 목표

	음성 이론	딥러닝 이론	딥러닝 실무	개발 실무
강의내용	50	20	20	10
현업	20	20	30	30

강의일정

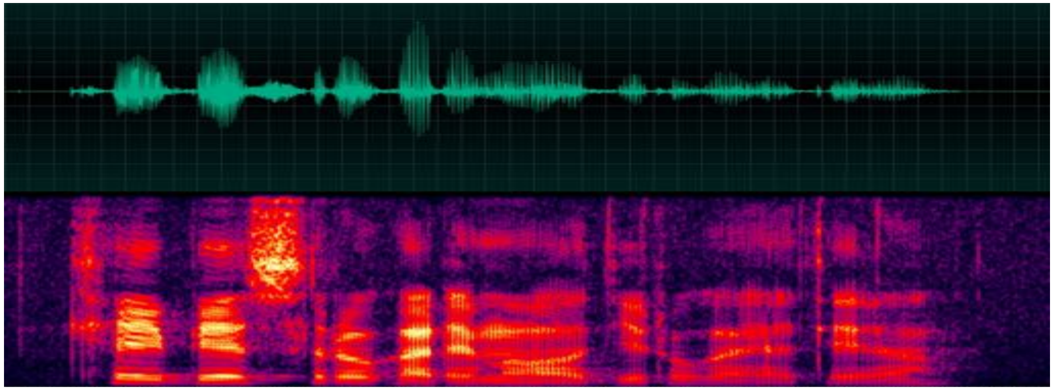
1일차	2일차	3일차
• 음성인식 개요 • (고전적) 음성인식 이론	• 딥러닝기반 (고전적) 음성인식 이론 • 종단형음성인식 개요	• 음성인식 평가 실습 • 성능 측정
• 실습환경소개 • 인식률 측정하기	• ESPNet 소개 • 종단형 음성인식 recipe 살펴보기	• 성능개선 방안 • 연구동향
• (고전적) 음성인식 이론 • 특징추출	• 트랜스포머 소개	• 연구동향 • OpenAI Whisper 실습
• 훈련DB 소개 • 특징추출 실습	• 음성인식 훈련 실습 • 훈련과정 분석 • Tensorboard	• 실습 마무리
Homework		

강의내용: 1일차

- What is Speech Recognition
- How to Evaluate Performance
- 실습
  - 실습환경 구성
  - WER 계산하기
- Feature Extraction
- 실습
  - Audio 들어보기, 멜스펙트럼 그려보기
- Q&A

what is speech recognition

- ASR(Automatic Speech Recognition), STT: Speech-to-text



  - Isolated, Connected, Continuous, Keyword Spotting
  - Speaker Dependent/Independent
- Difference with Image/Video Classification
  - Sequence Generation Problem

History of ASR

1950,60s

- Phonetic Recognizer
- 10 digit recognition
- DTW
- Idea of Continous ASR(CMU)

1970s

- IBM, Bell Lab, ...
- DARPA program
  - CMU Harpy:1,011 words vocab., FSN

1980s

- Connected words recognition (Fluently spoken)
- Template based → Statistical Methods
- HMM
- N-gram, Neural Nets.
- DARPA program
  - CMU SPHINX
  - BBN, SRI

1990s

- MCE, MMI
- DARPA programs
  - Natural Lanuage Recognition, ATIS, Broadcast news, Switchboard
- Robust ASR
- Applications

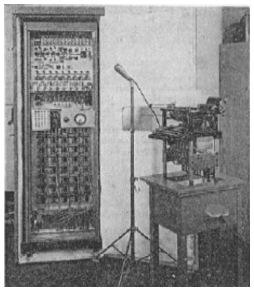
2000s

- Spontaneous speech
- Robust ASR
- Multimodal

50 Years of Progress in Speech and Speaker Recognition Research, ECTI Transactions On Computer And Information Technology, 2005

Applications

1956,
RCA Labs



1975,
1997,
Nuance

DRAGON
NaturallySpeaking

2012,
Google
Voice
Search

2011,
Apple
Siri

2014,
Amazon

2018,
Google
Duplex

1997,
삼성
애니콜

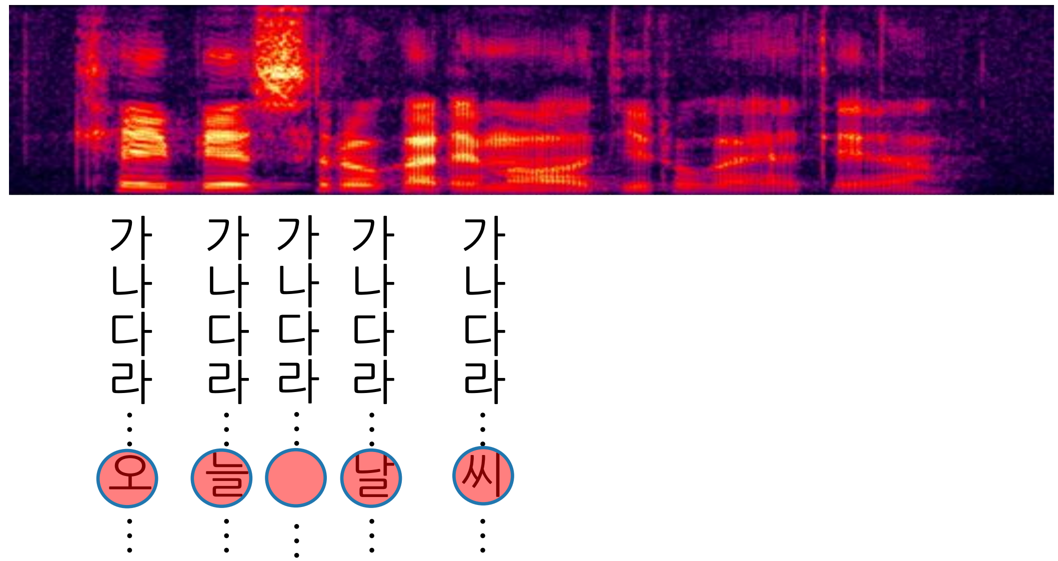
2008,
파인디지털

2012,
다음

2012~
ETRI

How It works

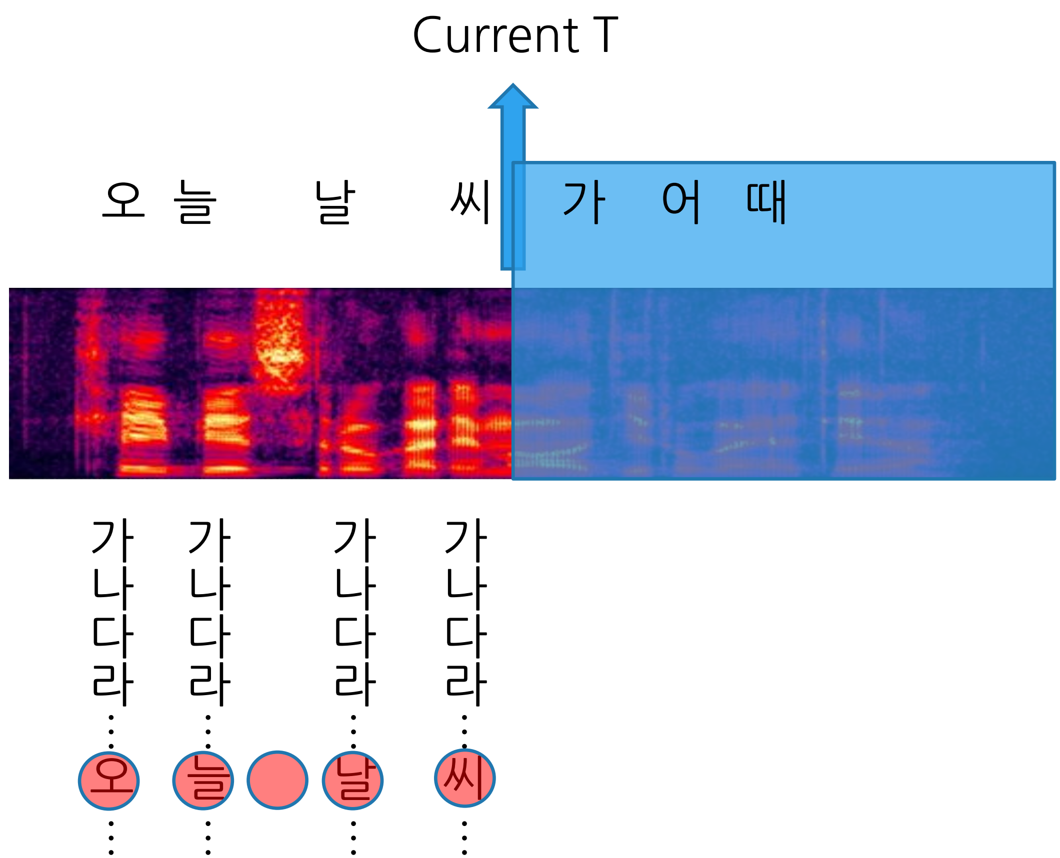
- W* = argmax P(W|X)
  - To Find Most Probable Word Sequence Given Input Signal/Feature



- Considerations
  - Boundary? Segmentation?
  - Output Units? Words, Characters, Phoneme, …
  - Classification Accuarcy? Unit Accuracy vs. Sentence Accuracy

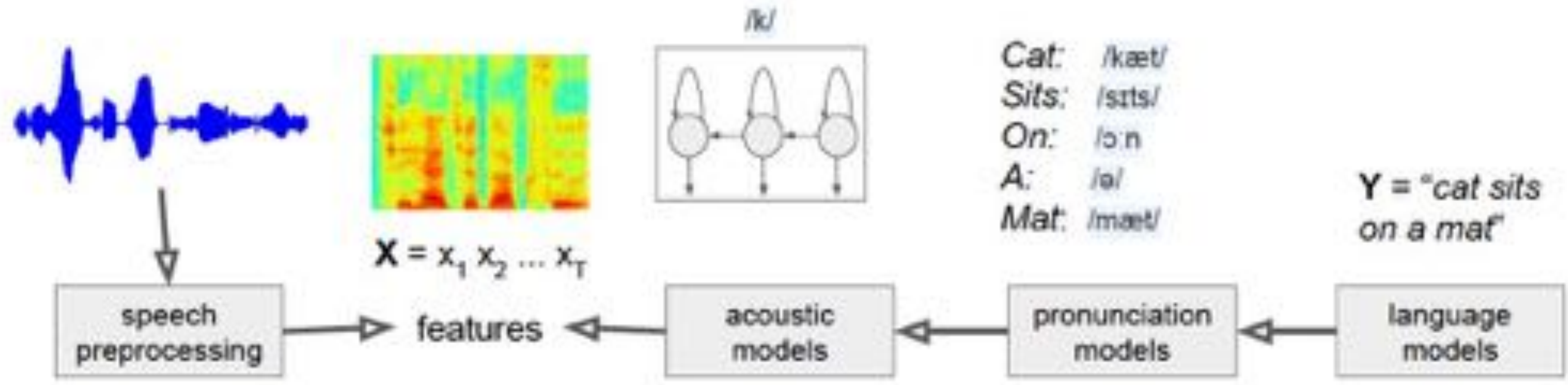
Context/Latency

- Batch or Streaming?



How It really works

- W* = argmax log P(W|X)
- = argmax log P(X|Q)P(Q|W)P(W)
- To Find Most Probable Sequence Among Plausible Words Sequences



https://heartbeat.fritz.ai/the-3-deep-learning-frameworks-for-end-to-end-speech-recognition-that-power-your-devices-37b891ddc380

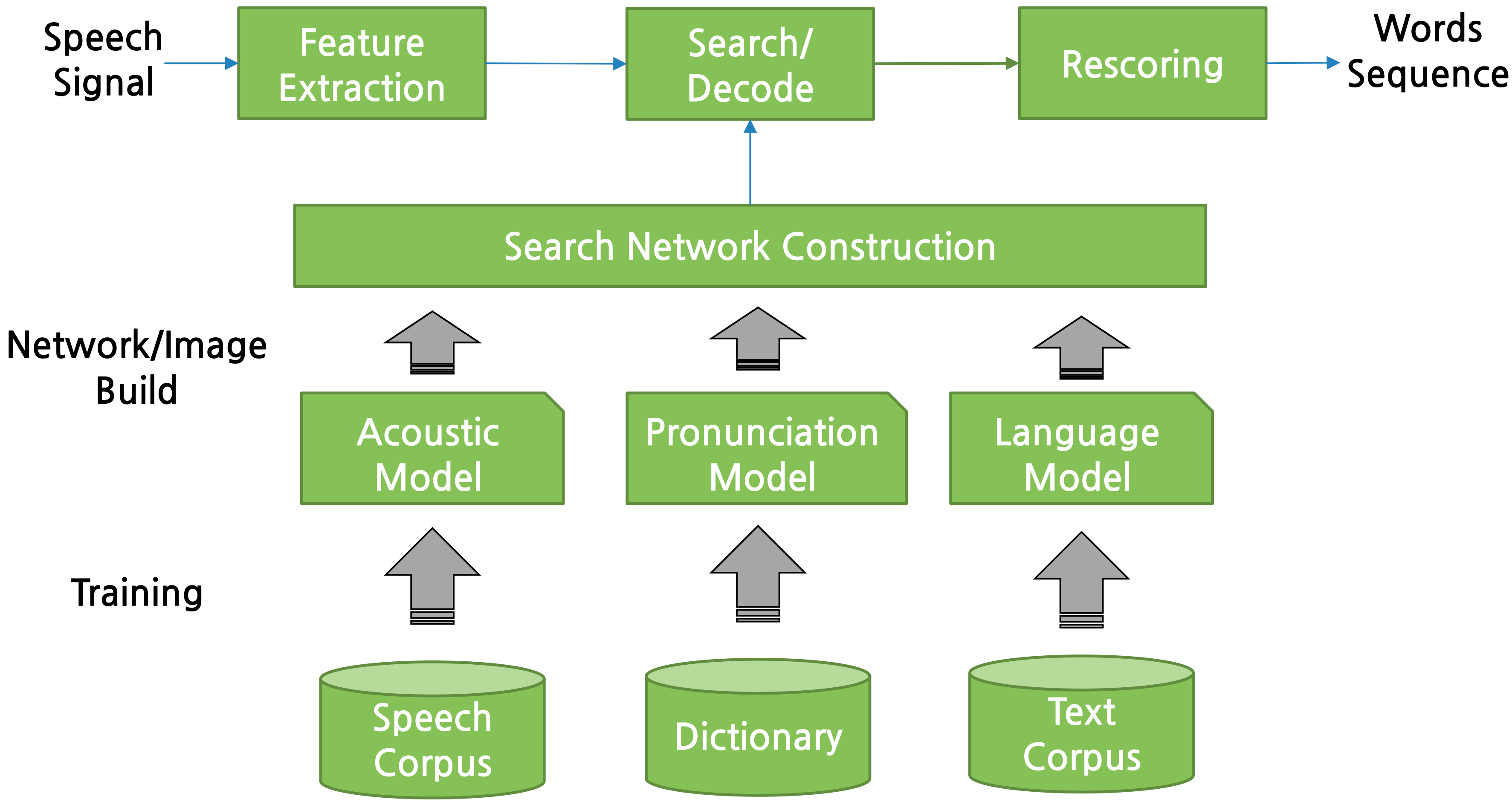
Guess who?

- Find A Criminal Among Suspects Given Evidence
- Criminal = argmax P(Suspect|Evidence)
- Criminal = argmax P(Evidence|Suspect)

  = argmax

  P(Evidence|Behavior)P(Behavior|Suspect)P(Suspect)

Structure of traditional asr



Evaluation Metric

- Types of Error
  - Substitution
  - Deletion
  - Insertion
- Error Rate (cab be > 1)
  - (S + D + I)/N
- Accuracy (can be < 0)
  - 1-(Error Rate)
- WER/CER/SER:
  - Word/Character/Sentence Error Rate

```
REF     : how is the weather  today
REC/HYP: how was the better   to day
```

In Words: WER = 100%, Acc=0%
- N= 5: how, is, the, weather, today
- S = 2
- D = 1
- I = 2

```
how is  the weather today
how was the better  to day
```

In Chars: CER = 25%, Acc=75%
- N= 20:

h,o,w,i,s,t,h,e,w,e,a,t,h,e,r,t,o,d,a,y
- S = 3
- D = 1
- I = 1

```
how is  the weather today
how was the better  to day
```

In Sentence: SER = 100%, Acc=0%
- N = 1
- S = 1

Quiz

- REF: 오늘 서울의 날씨가 어때
- REC: 음 오눌의 날씨 가 어때
- WER = ?

한국어

- REF: 오늘 서울의 날씨가 어때
- REC: 음 오눌의 날씨 가 어때
- WER=4/4 = 1.0 Acc=0.0
  - N=4, 오늘, 날씨가, 어때요
  - S = 2, D = 1, I = 2, WER=5/4
  - S = 3, I = 1, WER=4/4
- CER= 4/10 = 0.4, Acc=0.6
  - N = 10
  - S = 1
  - D = 2
  - I = 1

	오늘	서울의	날씨가		어때
음	오눌의		날씨	가	어때

오늘	서울의	날씨가		어때
음	오눌의	날씨	가	어때

	오	늘	서	울	의	날	씨	가	어	때
음	오	눌			의	날	씨	가	어	때

측정방법

- Edit distance 측정
  - https://en.wikipedia.org/wiki/Edit_distance
- 사용도구
  - HResults (HTK)
  - compute-wer (kaldi)
  - sclite (NIST, ESPnet)
- 예)
  - compute-wer ark:ref.txt ark:rec.txt

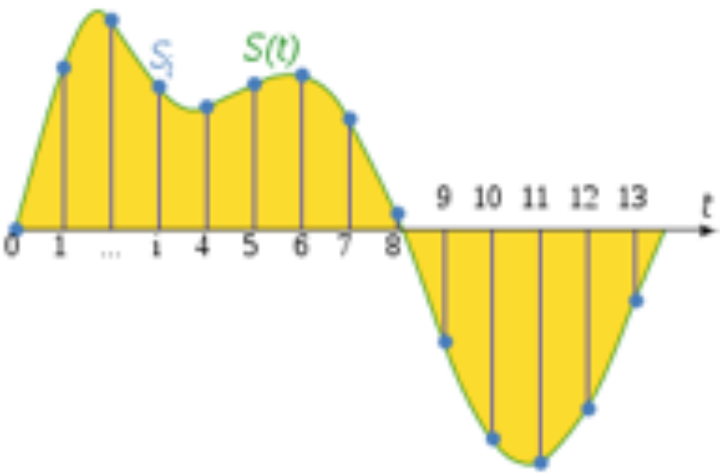
실습

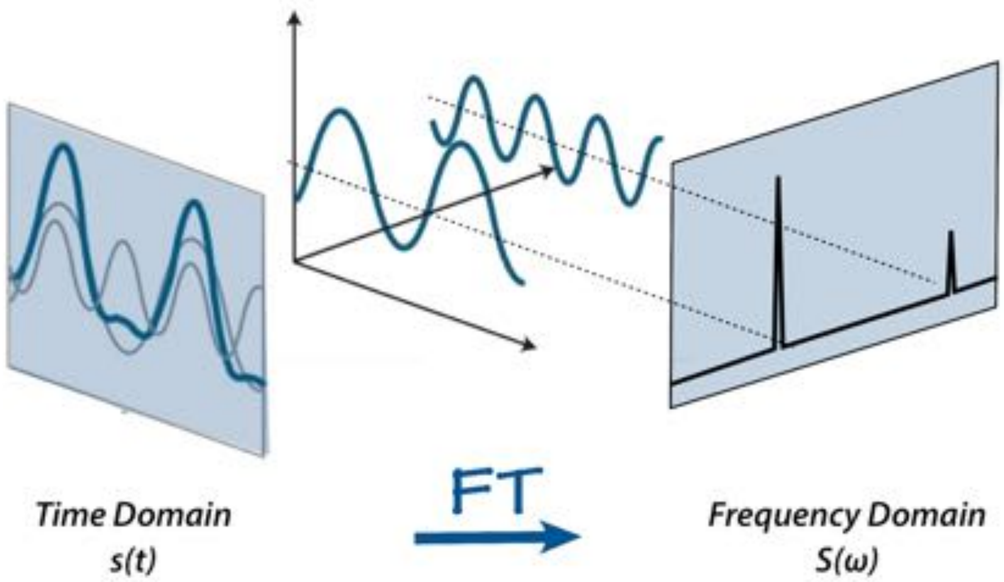
- 사용환경 설명/로그인/세션 생성
- VS Code/Jupyter/Python
- WER 측정

https://www.nvidia.com/ko-kr/data-center/dgx-a100/

feature extraction

sampling and Spectrum

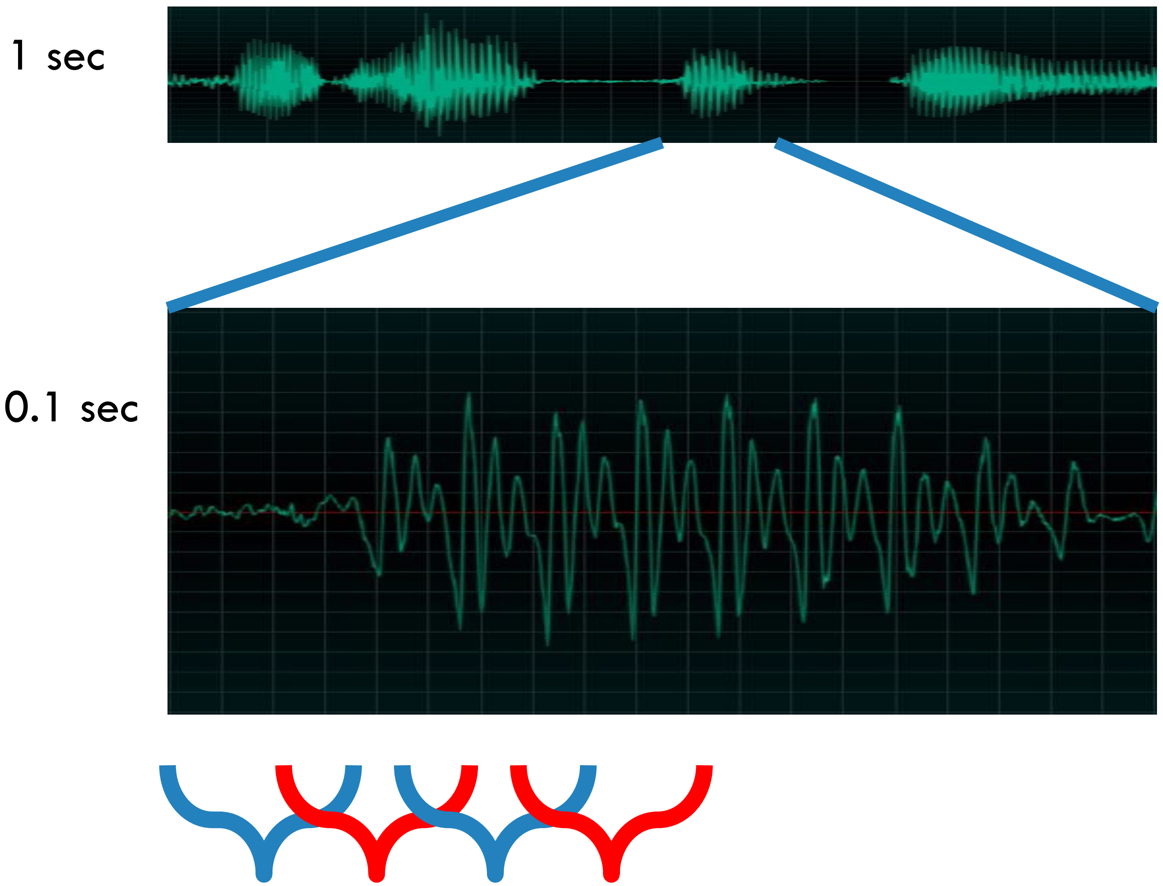




8kHz: Narrowband, 전화망
16kHz: Wideband
44.1/48kHz: High quality audio

https://en.wikipedia.org/wiki/Sampling_(signal_processing)
https://towardsdatascience.com/understanding-audio-data-fourier-transform-fft-spectrogram-and-speech-recognition-a4072d228520

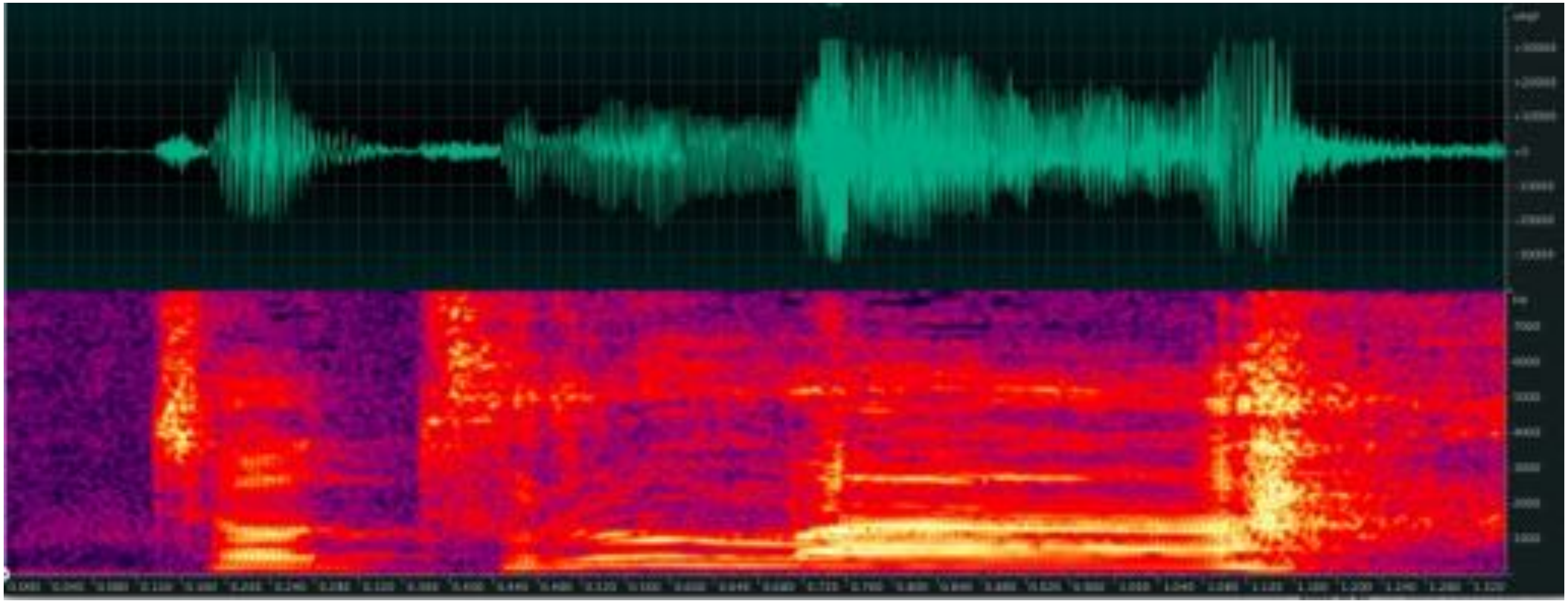
Frame-Wise Processing



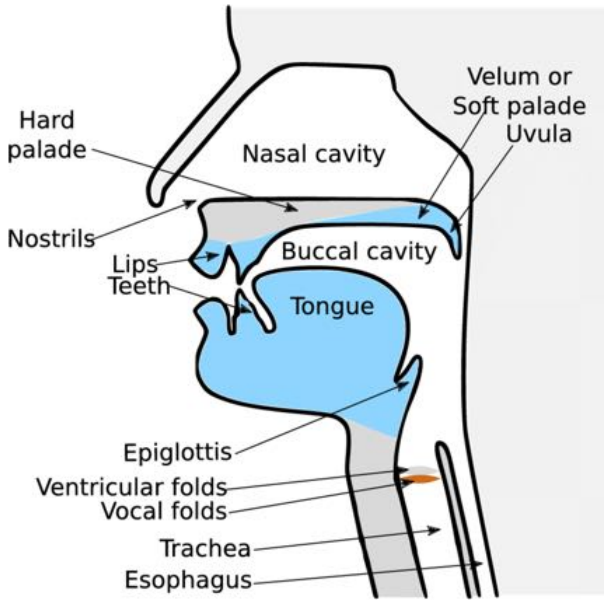
Windowing,
Window length,
FFT size,
Hop size

Spectrogram

- Series Of Spectrum
- Matlab, Python, Adobe Audition, Audacity, …
- Frame Shift, Overlap, Window Length, Windowing, FFT points

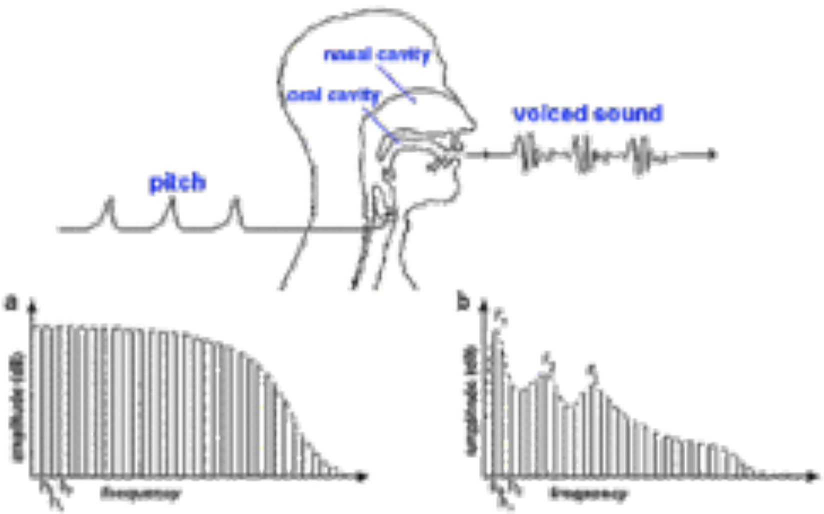


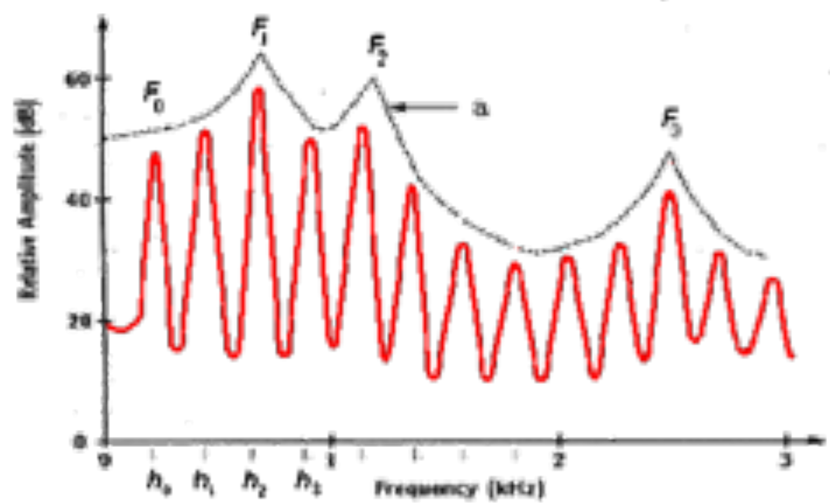
Voice Production



https://www.researchgate.net/publication/318814563_Analyzing_of_the_vocal_fold_dynamics_using_laryngeal_videos
https://www.youtube.com/watch?v=kfkFTw3sBXQ

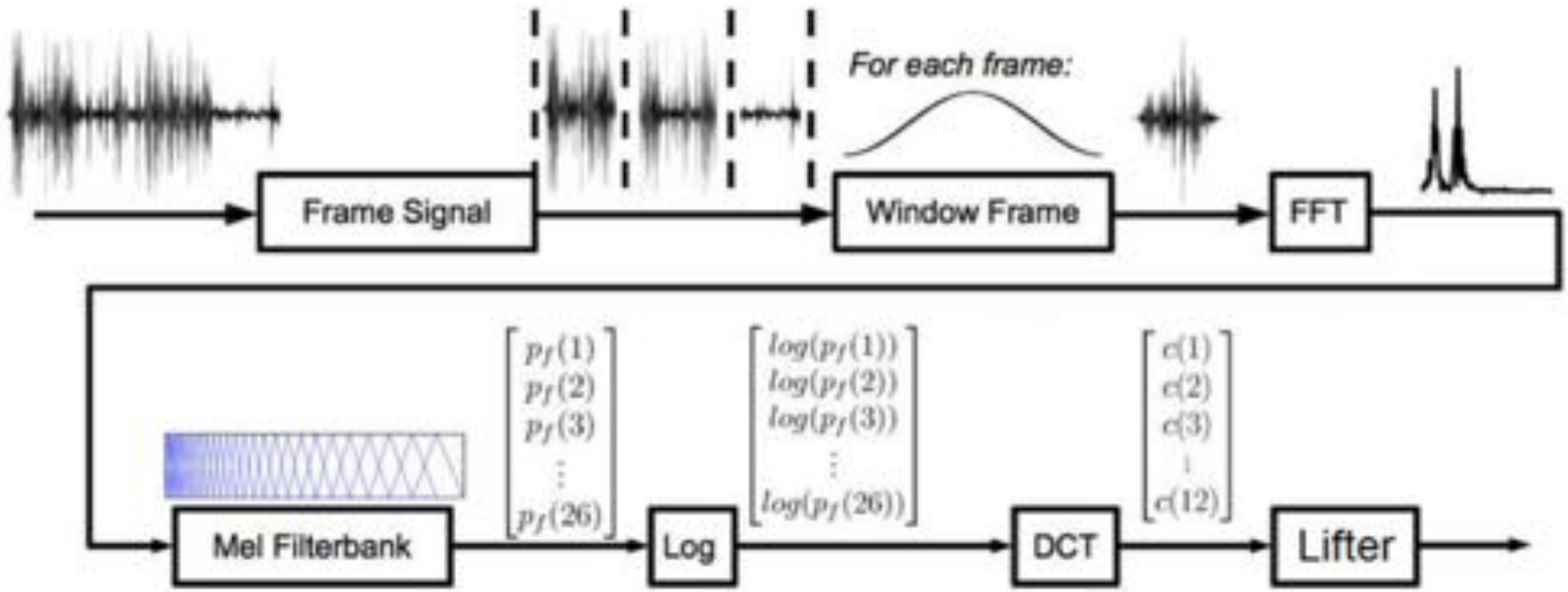
Pitch and Formant





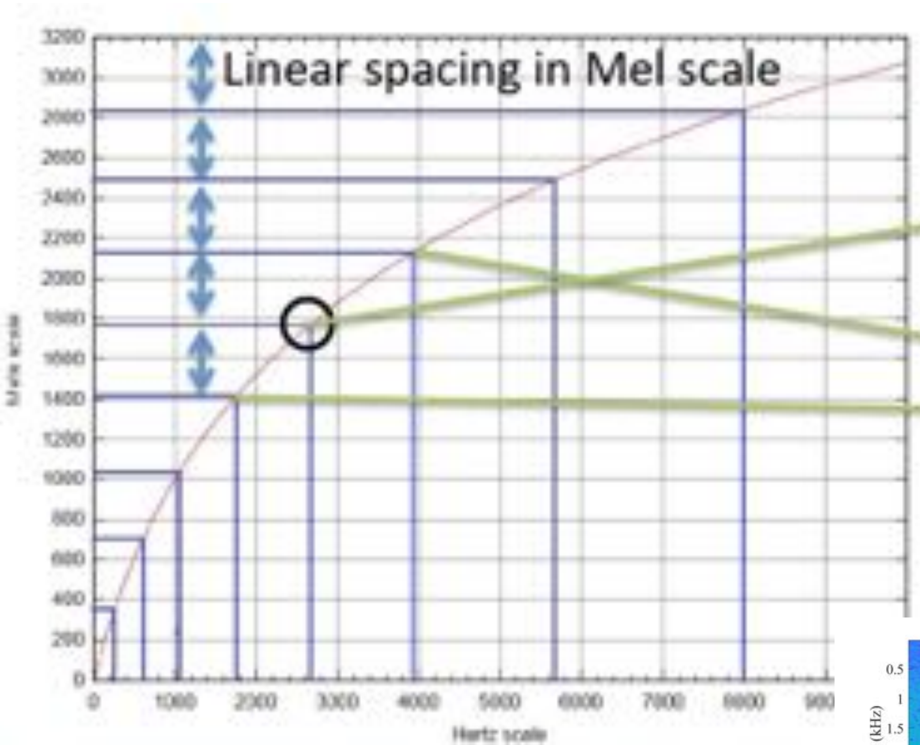
http://147.162.36.50/cochlea/cochleapages/theory/sndproc/sndcomm.htm

Feature extraction: MFCC

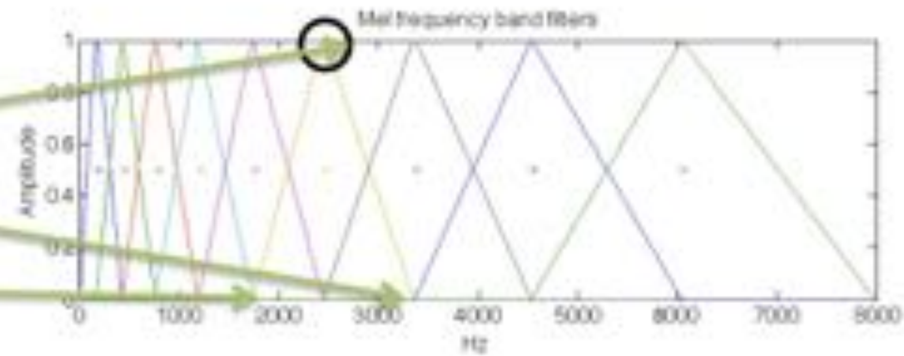


https://hyunlee103.tistory.com/46

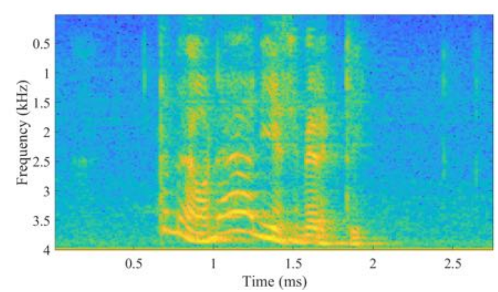
Mel Filterbank

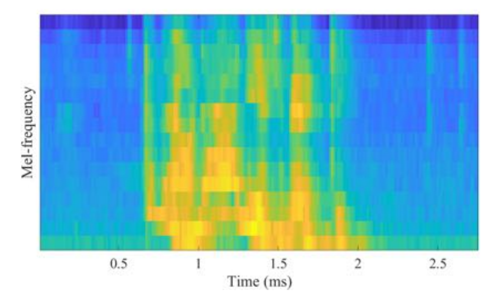


# of filters: 23 → 40 → 80+3



https://hyunlee103.tistory.com/46





https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC

CMVN

- Cepstral Mean Variance Normalization
  - Zero-mean Unit Variance
- CMS: Cepstral Mean Substraction
  - Per Utterance
  - 채널/화자 효과를 제거하고 발성의 특성만 남김
- For Deep Learning
  - Global CMVN
  - For better convergence

Homework

- 음성인식 평가용 테스트데이터 수집
- 본인 (또는 근처 아무나) 목소리를 녹음
  - 16kHz, wav (uncompressed), mono
  - (일단 녹음하고 확인해봅시다)
- 인식률이 되도록 좋게 or 나쁘게 나오도록
  - But don't be too evil… noise, yell, whisper…
- 10문장 정도, 1문장당 10초 정도.
- wav/text pair (wav.scp, text)
- Due: 3일차 시작 전까지

강의내용: 2일차

- Classical ASR
- Introduction to End-to-End ASR
- 실습
  - ESPNet 소개
  - 종단형 음성인식 recipe 살펴보기
- Transformer
- 실습
  - 한국어 1,000시간 훈련 DB를 이용한 훈련 시작

Classical ASR

HMM-based ASR

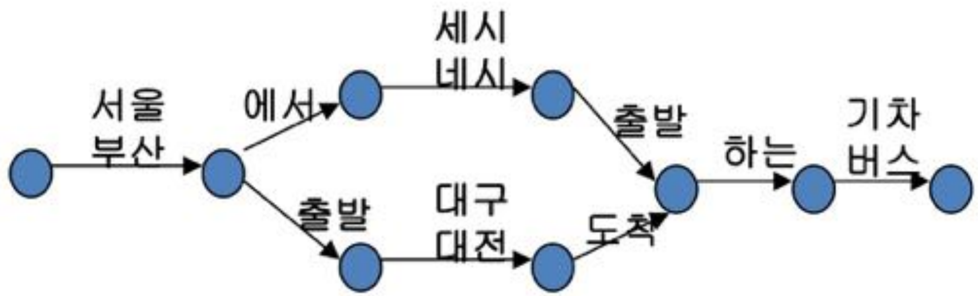
- How we call it?
  - Conventional
  - Traditional/Classical
  - Ancient
- Why?
  - 내부 동작을 이해하고 문제점 또는 성능 개선 방법을 찾기 위해서

era of hidden markov model

- Problem to Solve:
- $W* = \text{argmax} \log P(W|X)$
- $\quad = \text{argmax} \log P(X|Q)P(Q|W)P(W)$
- P(W): Language Model, $P(Wt|Wt\text{-}1, Wt\text{-}2, \cdots)$
- P(Q|Wt): Prononciation Model
- P(X|Q): Acoustic Model

Language Model

- 단어간의 연결 가능성을 이용하여 search space를 제한



- Deterministic Grammar
  - FSN (Finite State Network)
  - JSGF (Java Speech Grammar

$time = 세시|네시;
$city = 서울|부산|대구|대전;
$trans = 기차|버스;
sent-start $city (에서 $time 출발 |
출발 $city 도착) 하는 $trans sent-end

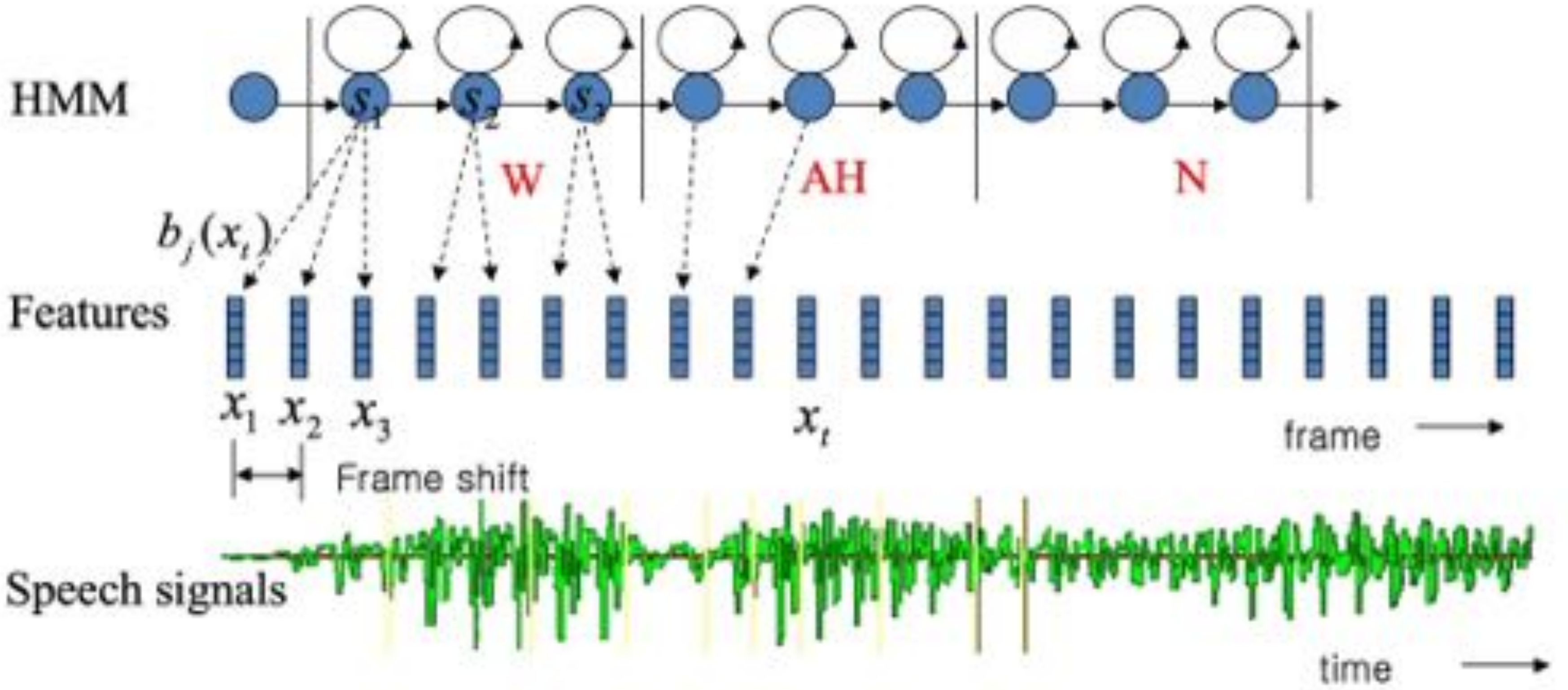
- Stochastic Grammar
  - N-gram

P(에서|서울)=0.2 P(세시|에서)=0.5
P(출발|세시)=1.0 P(하는|출발)=0.5
P(출발|서울)=0.5 P(도착|대구)=0.9
…

Prononciation Model

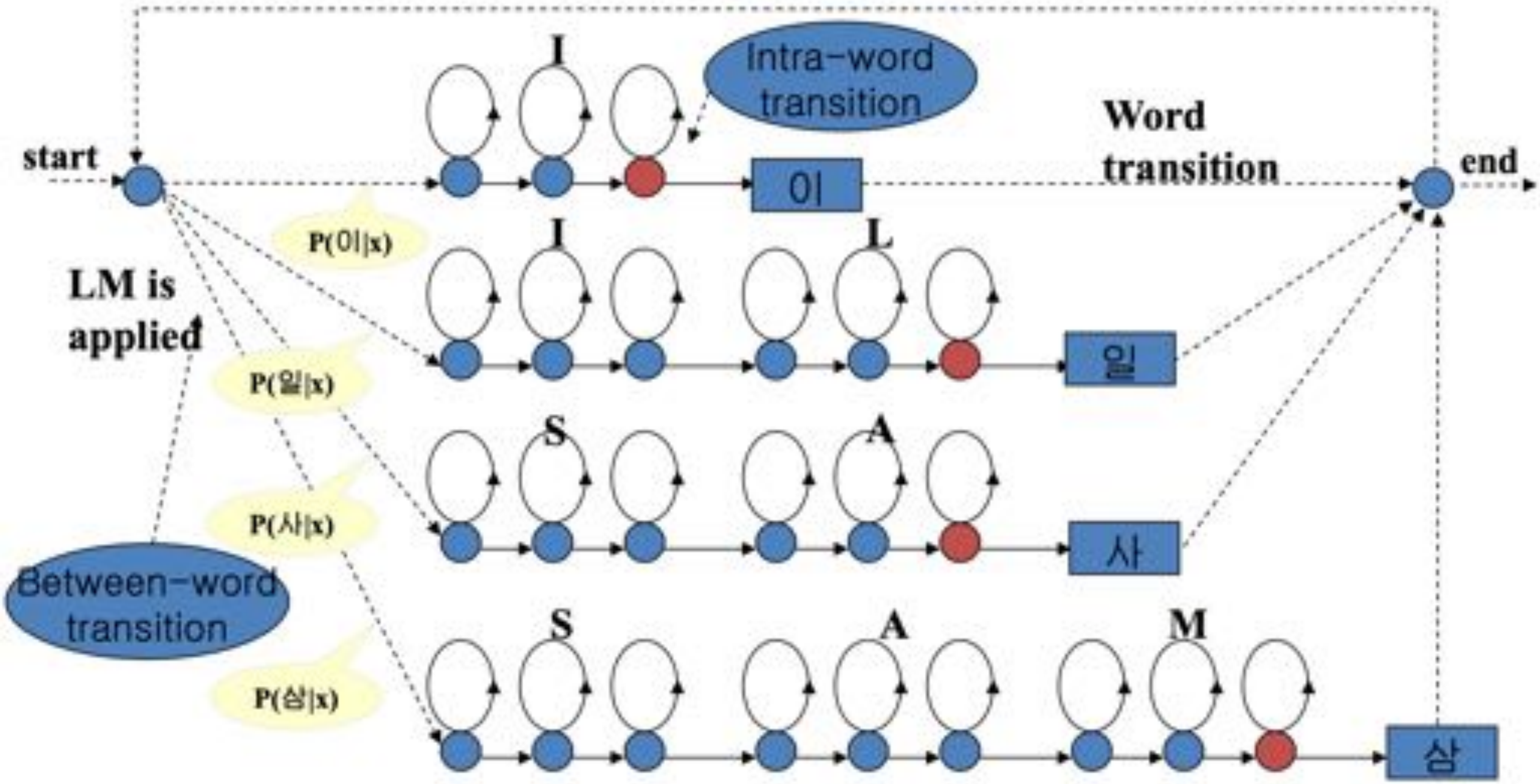
- How a word is pronunciated
- Very Langauge-Dependent and Requires Expert Knowledge
  - 대한민국: /d E h a xn m i xn g u xg/
  - 2NE1, 야탑역, 맨유
- Phoneset
  - 한국어: ETRI 46 phoneset
  - 영어: CMUDict(48), TIMIT(61) → CMU 39 phoneset
- Rule-based, Statistical Approach, Neural Approach

Acoustic Model



http://speech.cbnu.ac.kr/srhome/technology/index.html

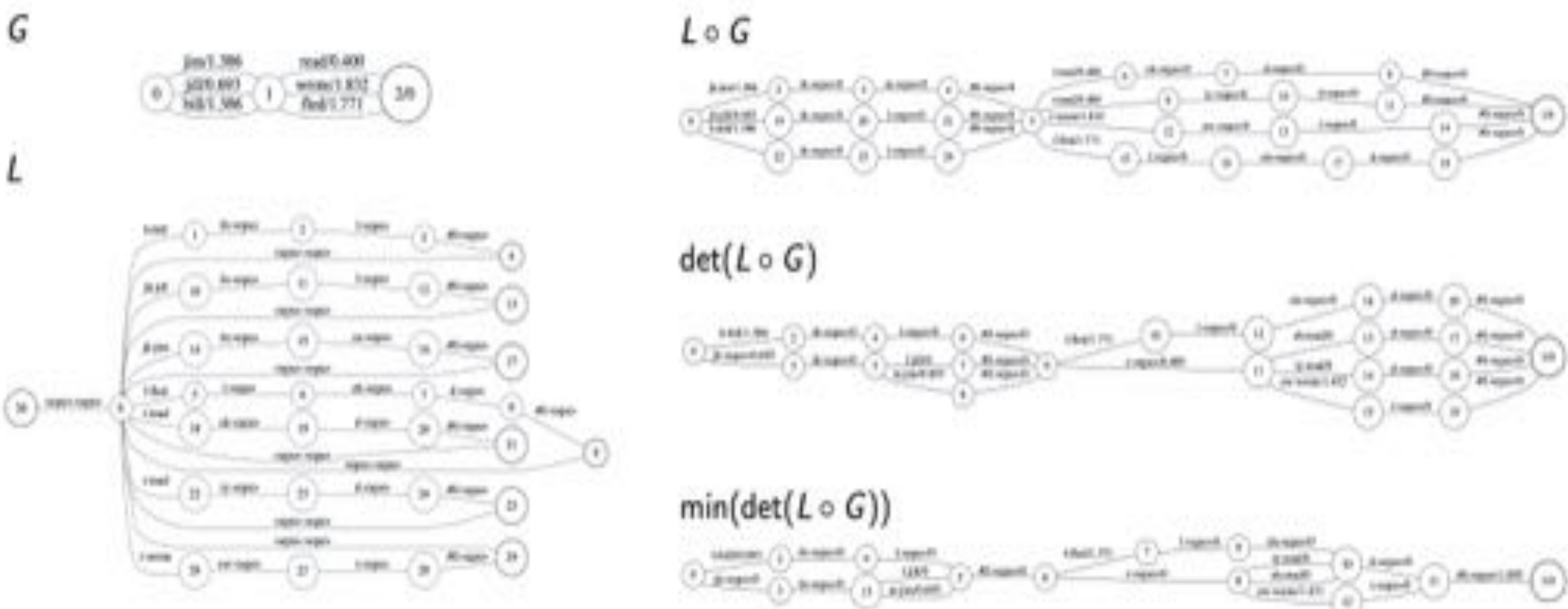
Search Network



http://speech.cbnu.ac.kr/srhome/technology/index.html

Search Network: wFST

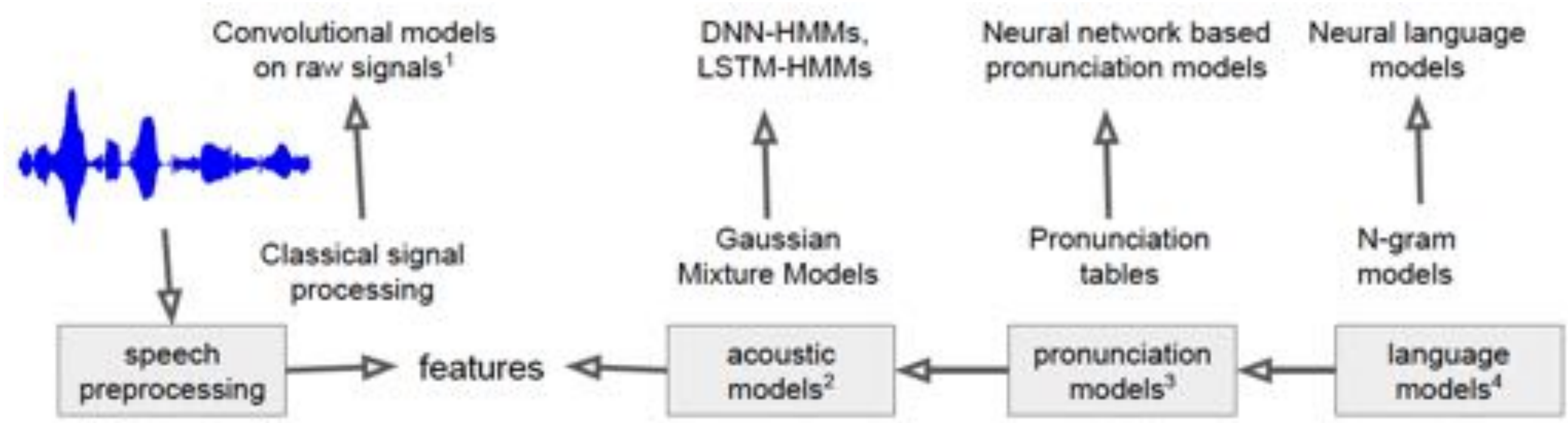
- Weighted Finite Statue Transducer



https://medium.com/@jonathan_hui/speech-recognition-weighted-finite-state-transducers-wfst-a4ece08a89b7

Deep Learning for ASR

Deep Learning for ASR



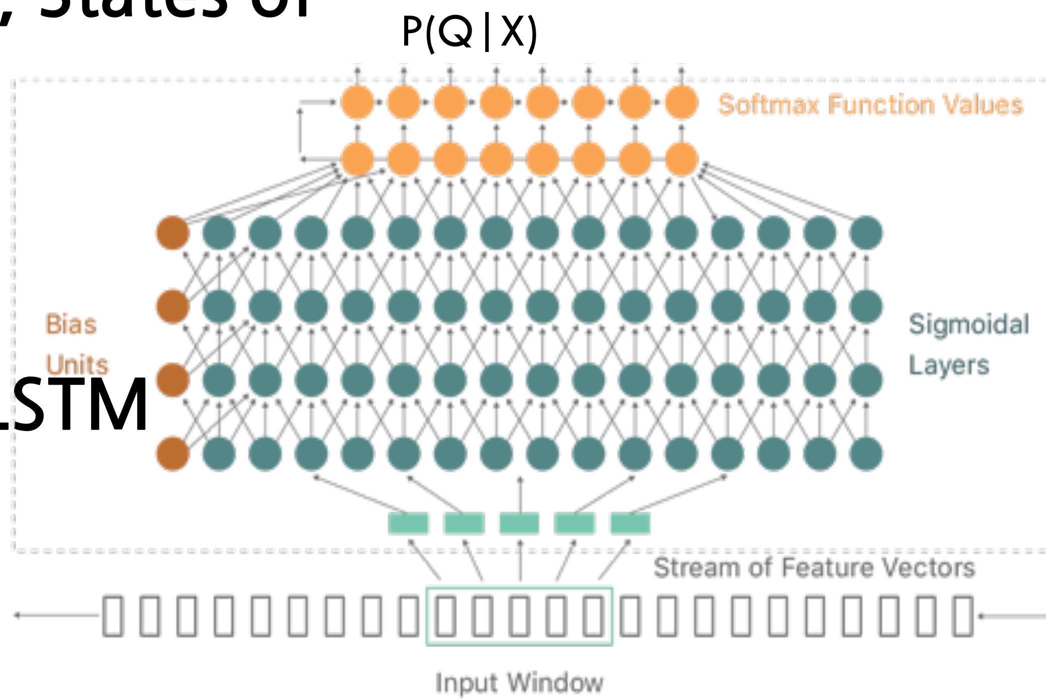
https://heartbeat.fritz.ai/the-3-deep-learning-frameworks-for-end-to-end-speech-recognition-that-power-your-devices-37b891ddc380

DNN-HMM (1)

- Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition, 2012, IEEE Trans. on Audio, Speech and Language Processing, Microsoft
- Large Vocabulary Continuous Speech Recognition With Context-dependent DBN-HMMS, 2011, ICASSP, Microsoft
- Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups, 2012, Hinton et al.

DNN-HMM (2)

- P(X|Q) = P(Q|X)P(X)/P(Q)
- Output Units: Senone, States of HMM(5k~20k)
- FC-DNN
- CNN
- RNN: GRU, LSTM, Bi-LSTM
- TDNN
- Longer Context Helps



TRAINING OF DNN-HMM

- Requires Frame-wise Label
- Forced-Alignment using Seed Model (Usually GMM-HMM Model)
  - Speech/Text Pair → g2p → State level alignment

- Kaldi Toolkit (2009~, JHU)
  - https://github.com/kaldi-asr/kaldi
- HTK Toolkit (1989~, Cambridge)
  - http://htk.eng.cam.ac.uk/
  - https://github.com/open-speech/HTK

end-to-end ASR

Contents

- How (Ancient) ASR Works: Recap
- How ASR Works: To-Be
- Introduction to
  - RNN, Attention, Encoder-Decoder, Word Embedding
  - Sequence-to-Sequence Model
- Transformer
- Transformer for ASR
- End-to-End ASR in Practice
- Q&A

Guess who?

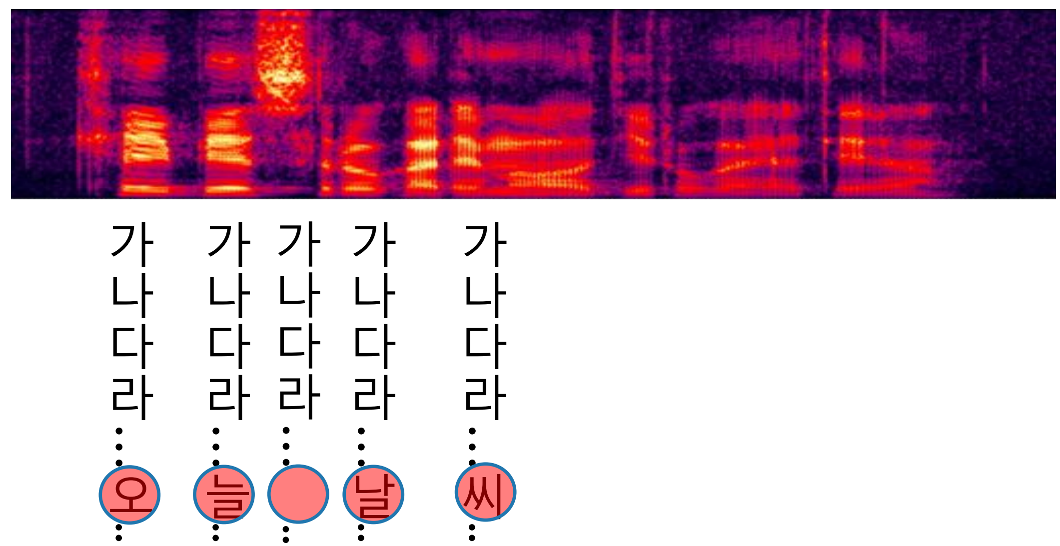
- Find A Criminal Among Suspects Given Evidence
- Criminal = argmax P(Suspect|Evidence)
- Criminal = argmax P(Evidence|Suspect)

  = argmax

  P(Evidence|Behavior)P(Behavior|Suspect)P(Suspect)

How It works: REVISITED

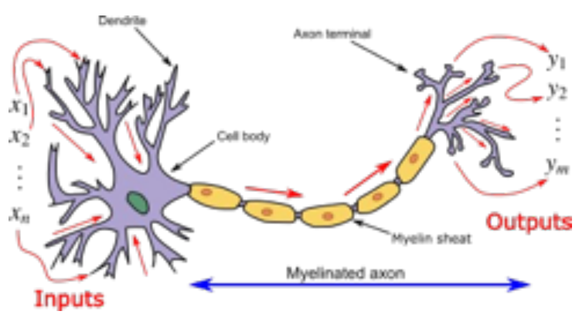
- W* = argmax P(W|X)
  - To Find Most Probable Word Sequence Given Input Signal/Feature



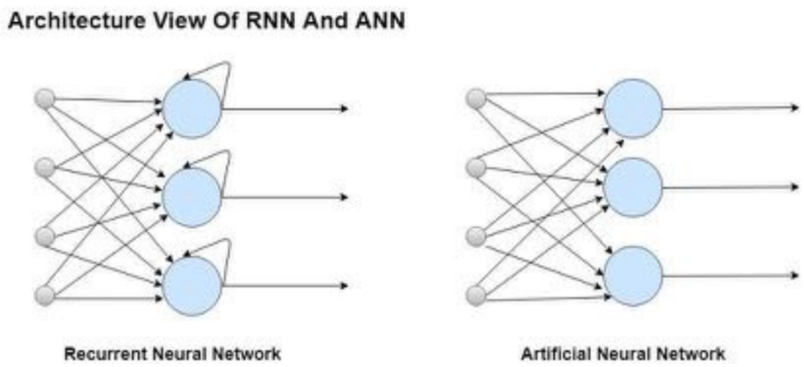
- Considerations
  - Boundary? Segmentation?
  - Output Units? Words, Characters, Phoneme, …
  - Classification Accuarcy? Unit Accuracy vs. Sentence Accuracy

RNN: Recurrent neural network

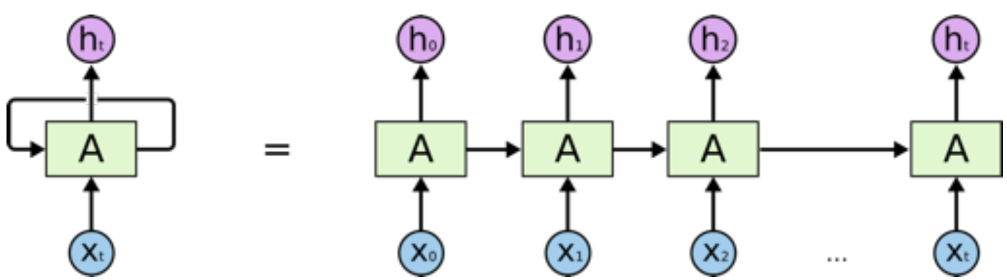
- Neural Networks
  - Mimic human brain: Neuron, Synapse



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



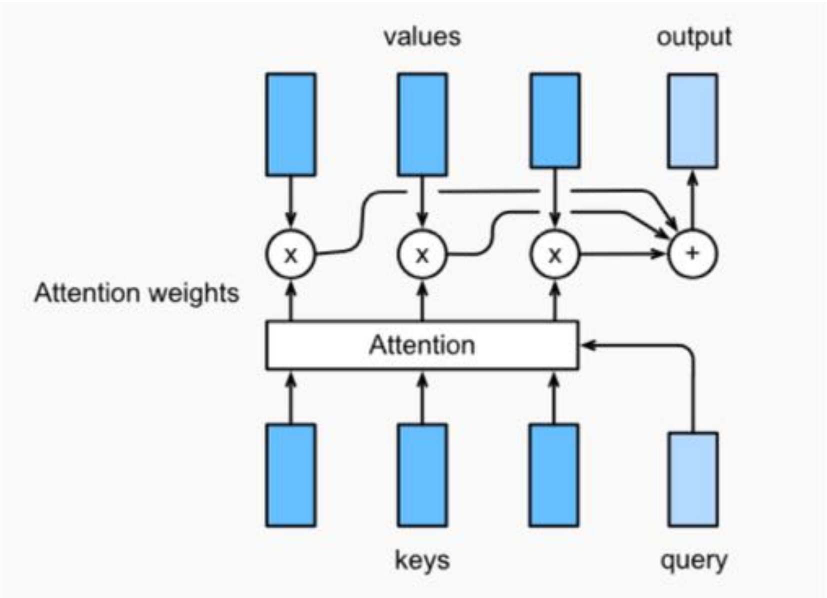
https://medium.com/datadriveninvestor/recurrent-neural-networks-in-deep-learning-part-1-df3c8c9198ba



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Attention

- Query, Key, Value
- Memory = Dictionary(Key,Value)
- Output = Weighted Sum of Value
- Weight = Similarity Between Query and Key



https://programming.vip/docs/5e4cadd75dc1d.html

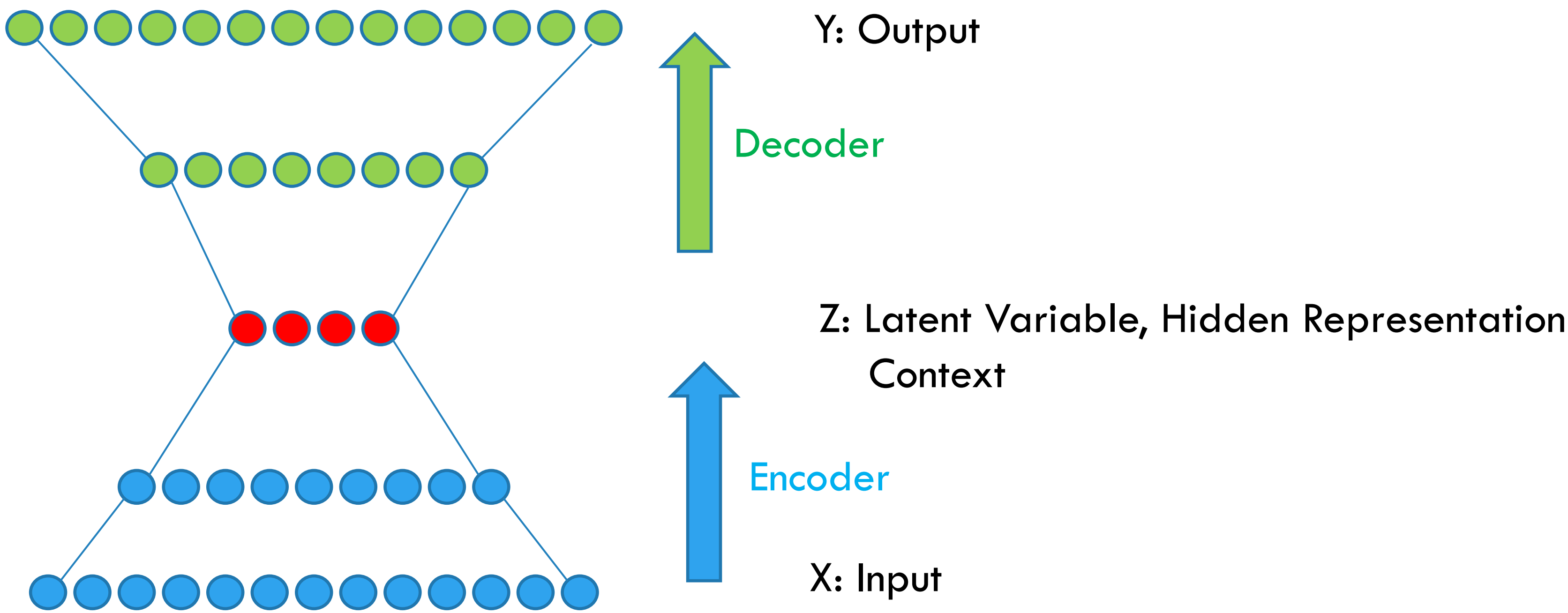
Word Embedding

- Word2Vec, …
- Sparse Representation vs. Dense Representation
- Preserve Meaning
  - 한국 - 서울 + 파리 = 프랑스
  - 어머니 - 아버지 + 여자 = 남자
  - 아버지 + 여자 = 어머니

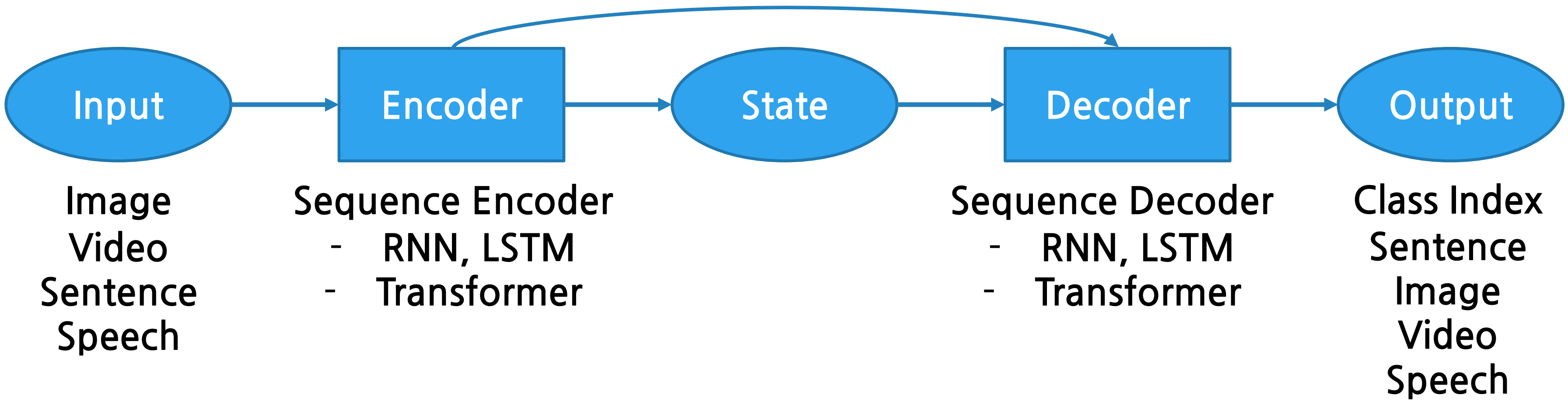
Index	Words	One-hot
1	aaron	000…00001
2	aback	000…00010
3	abacus	000…00100
…	…	…
15439	macaroni	00…010…0
	…	…
29500	zulu	1000…0000

encoder-decoder

- Auto Encoder



Encoder-Decoder for Sequence

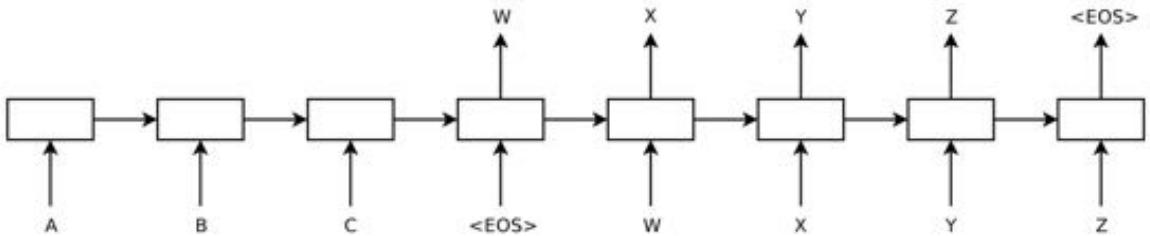


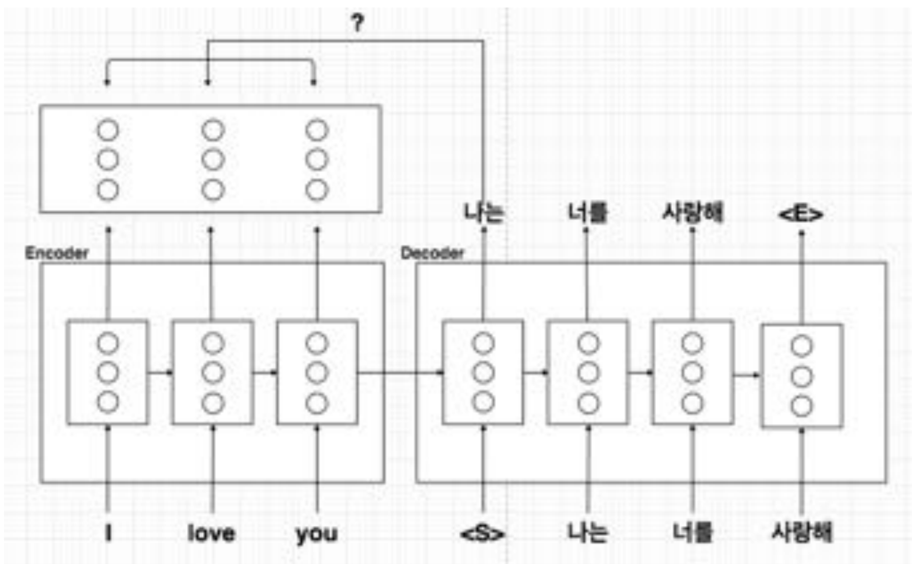
- Translation
- Image/Video Captioning
- Q&A, Document Summarization
- Speech
  - Recognition, Synthesis, Translation, Dialog System(Google Duplex, 2018)

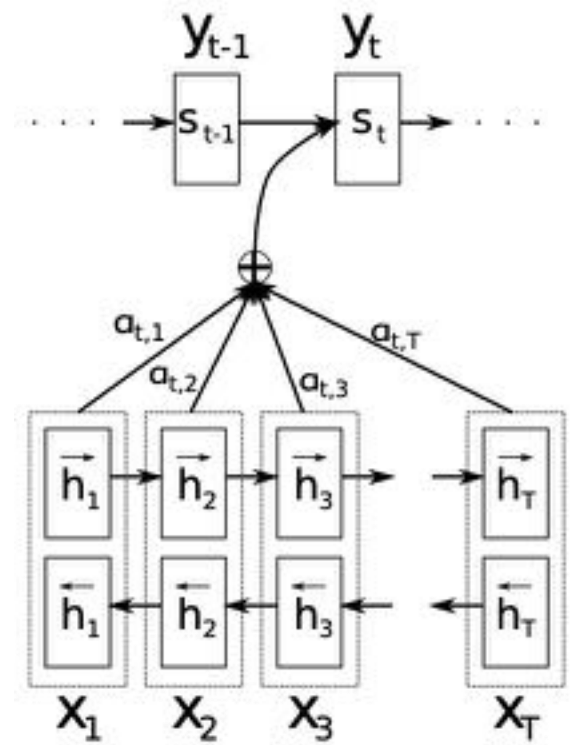
Era of Sequence-to-Sequence

- Natural Language Processing
- Sequence to Sequence Leanring with Neural Networks, NeurIPS, 2015
- Neural Machine Translation By Jointly Learning To Align And Translate, ICLR, 2016
- Attention Is All You Need, NuerIPS, 2017
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, ACL, 2019

Sequence to Sequence with Attention





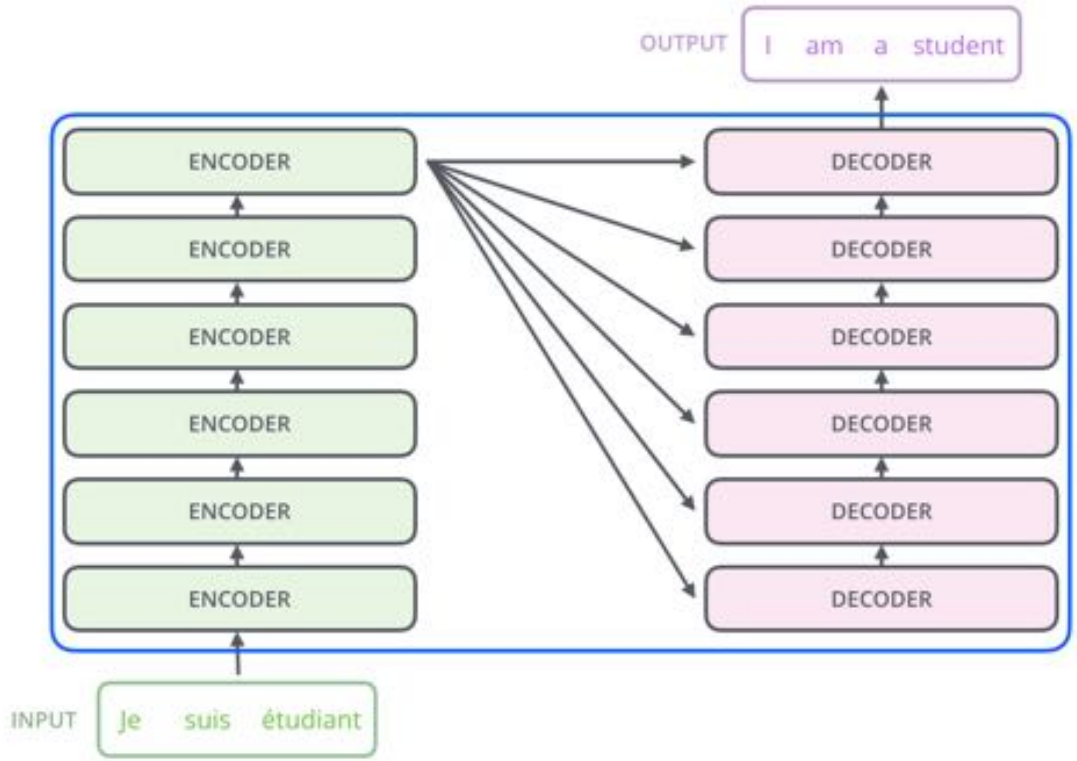


https://medium.com/platfarm어텐션-메커니즘과-transfomer-self-attention-842498fd3225

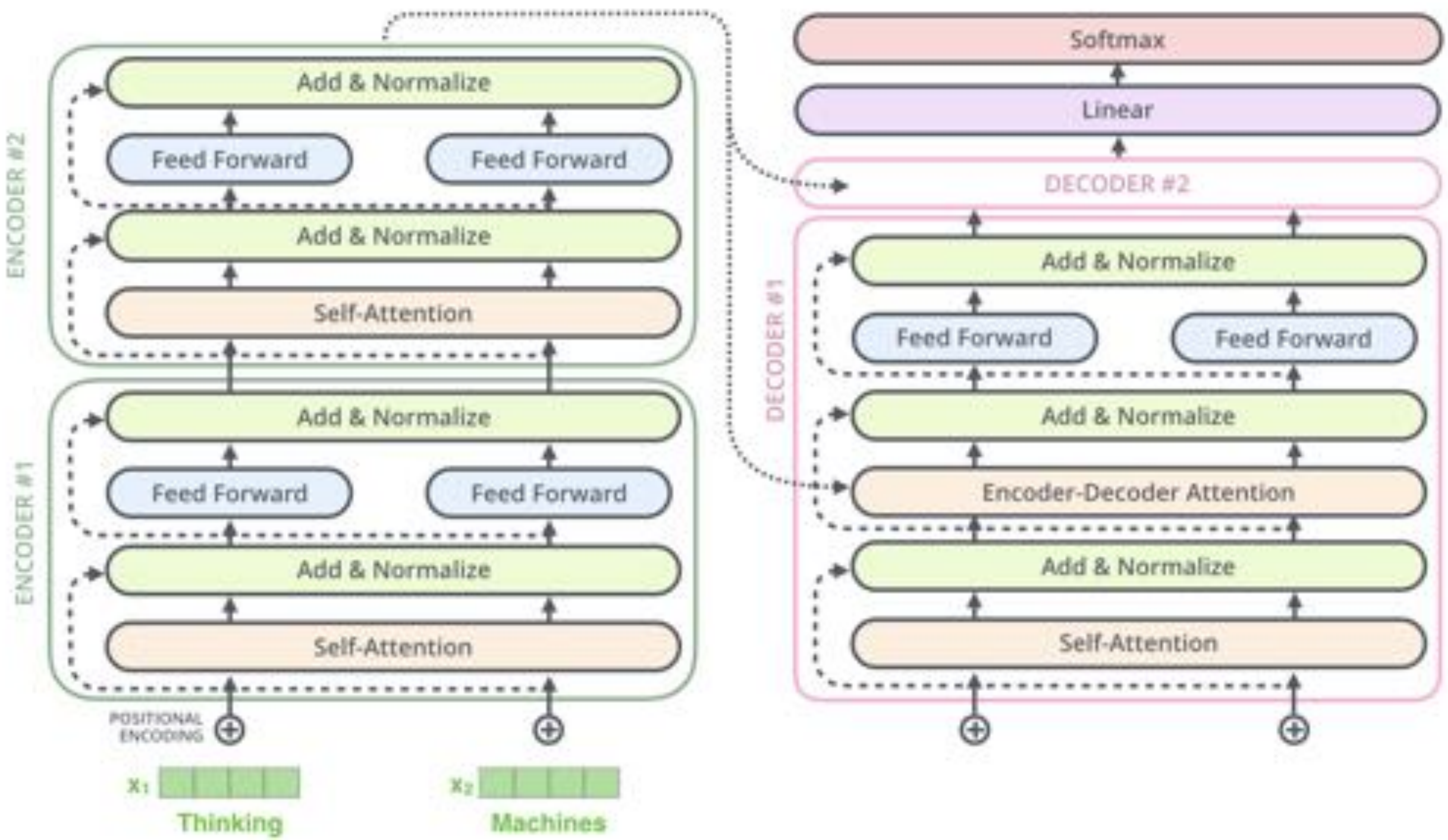
Transformer

Transformer: Overall STructure

- https://jalammar.github.io/illustrated-transformer/

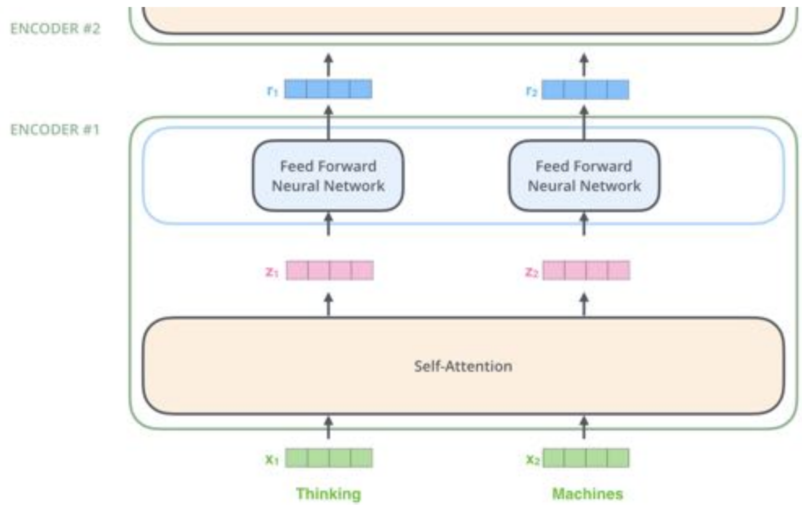


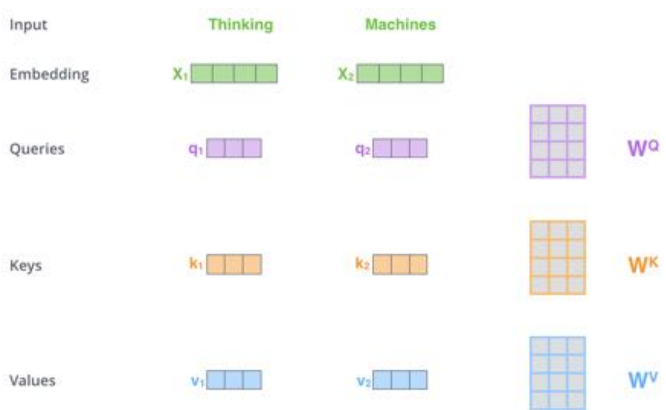
Transfomer: Detailed STructure

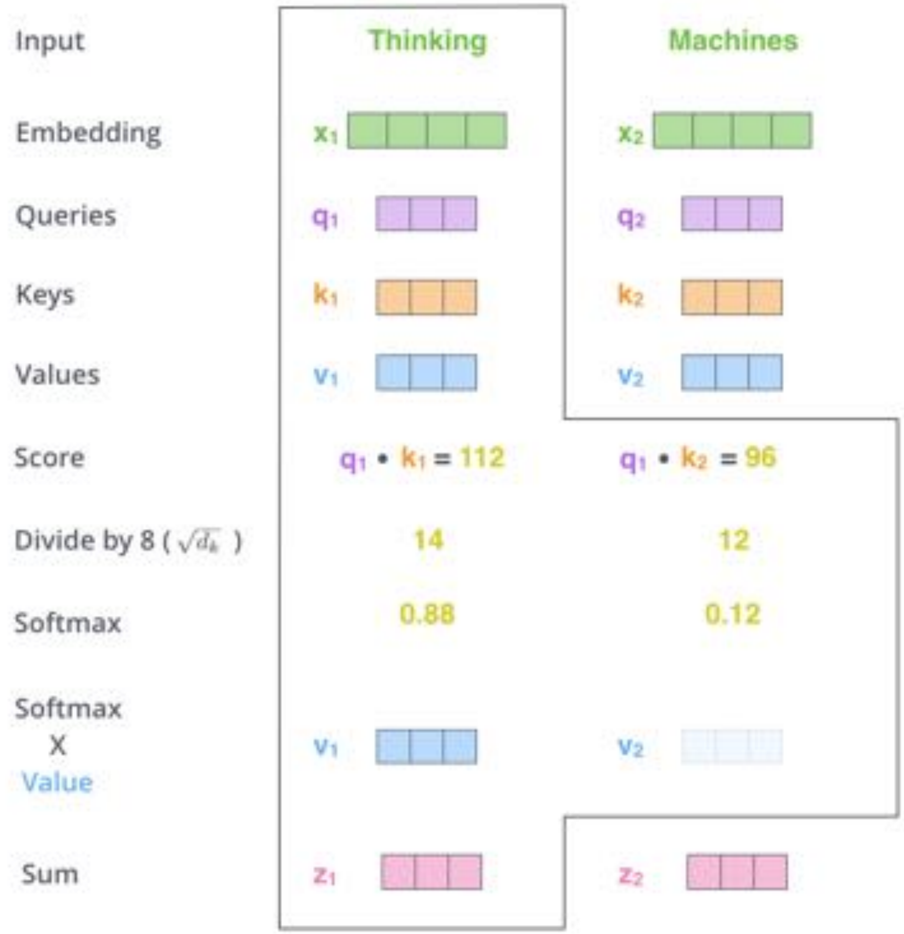


https://jalammar.github.io/illustrated-transformer/

Self Attention

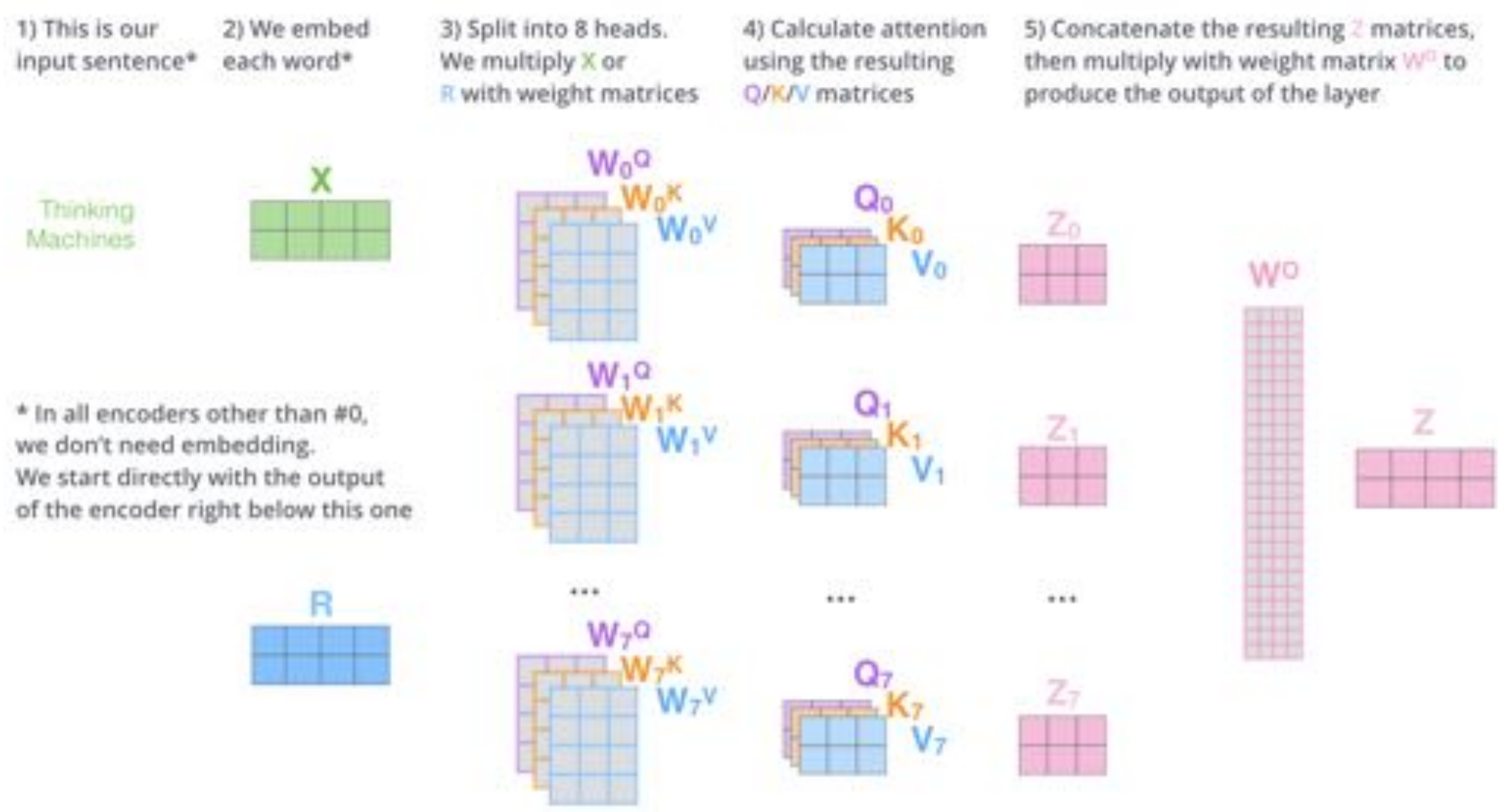






https://jalammar.github.io/illustrated-transformer/

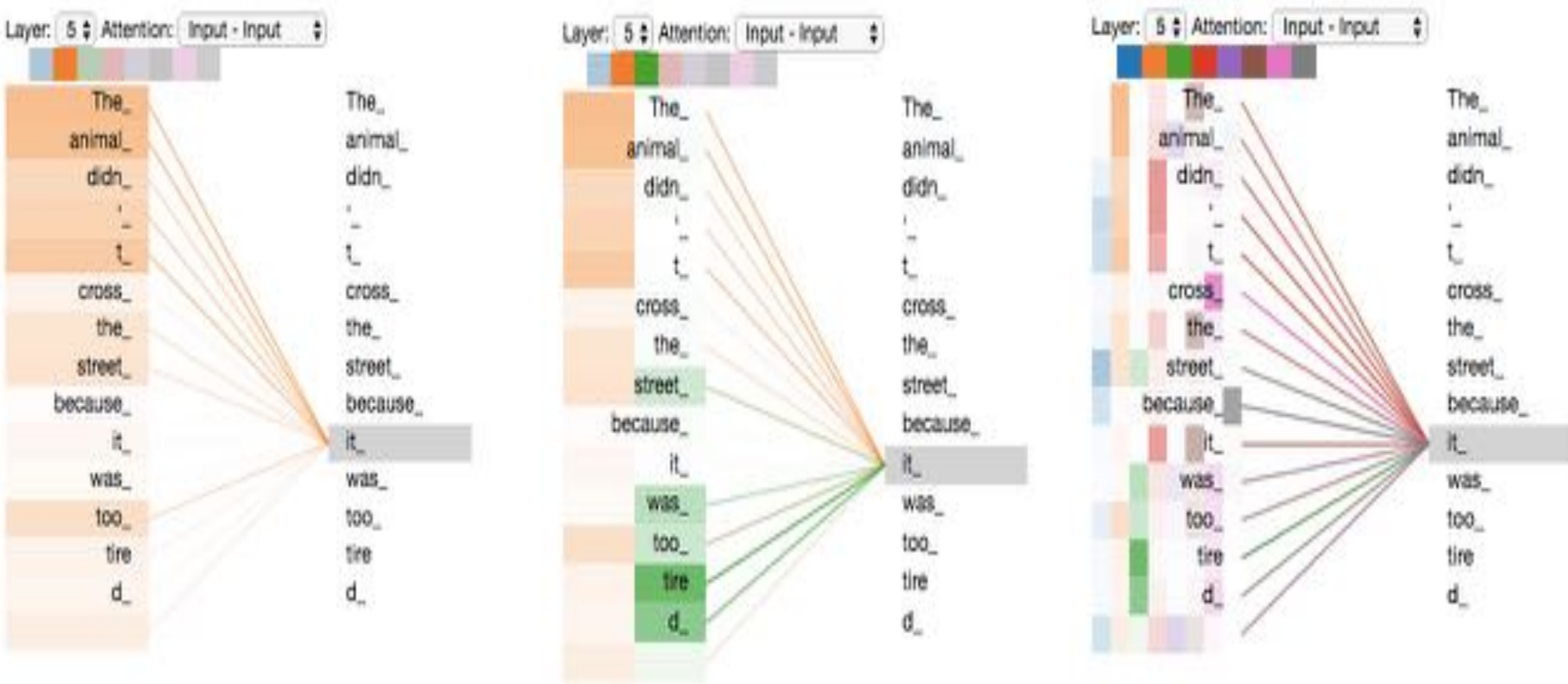
Multihead attention



https://jalammar.github.io/illustrated-transformer/

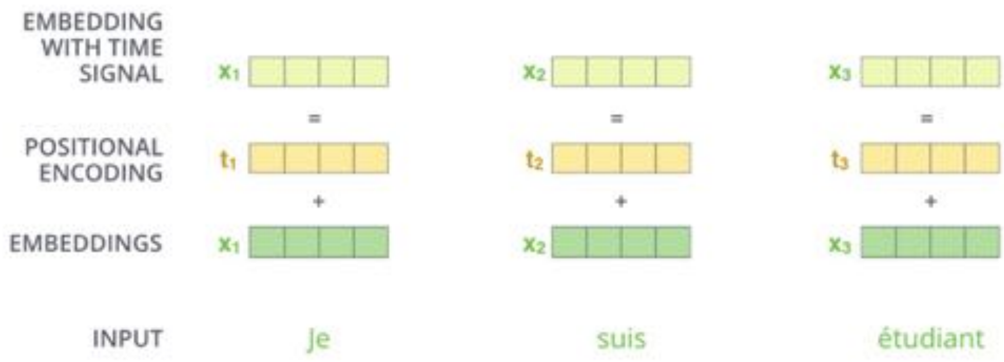
Effect of Self Attention

The animal didn't cross the street because it was too tired



Positional Encoding

- No Position Depenent Computation in Transfomer



0 :	0	0	0	0
1 :	0	0	0	1
2 :	0	0	1	0
3 :	0	0	1	1
4 :	0	1	0	0
5 :	0	1	0	1
6 :	0	1	1	0
7 :	0	1	1	1

- Absolute/Relative Position Encoding
  - Sinusoidal Positional Encoding

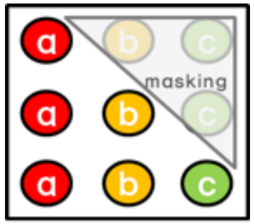
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

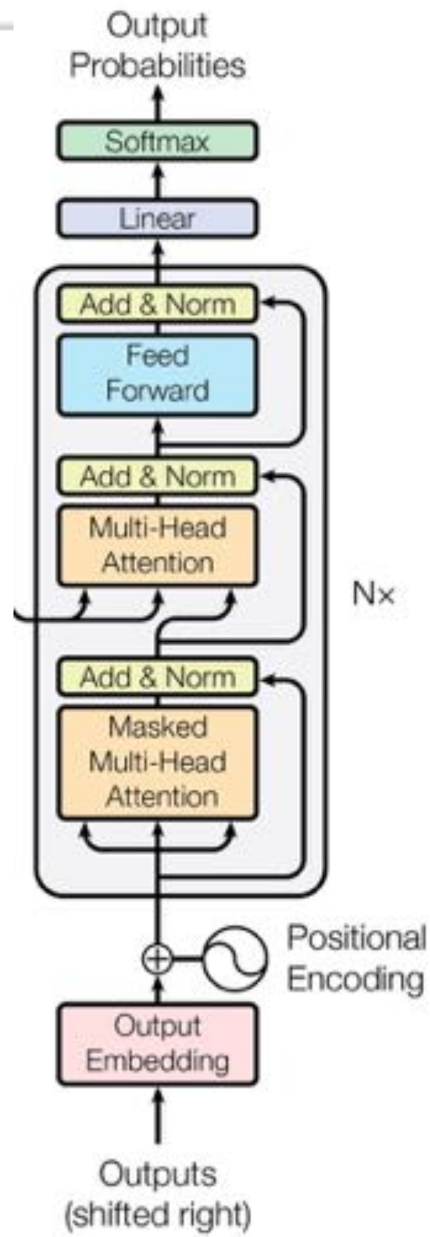
https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

Decoder

- Masked Multi-Head Self Attention

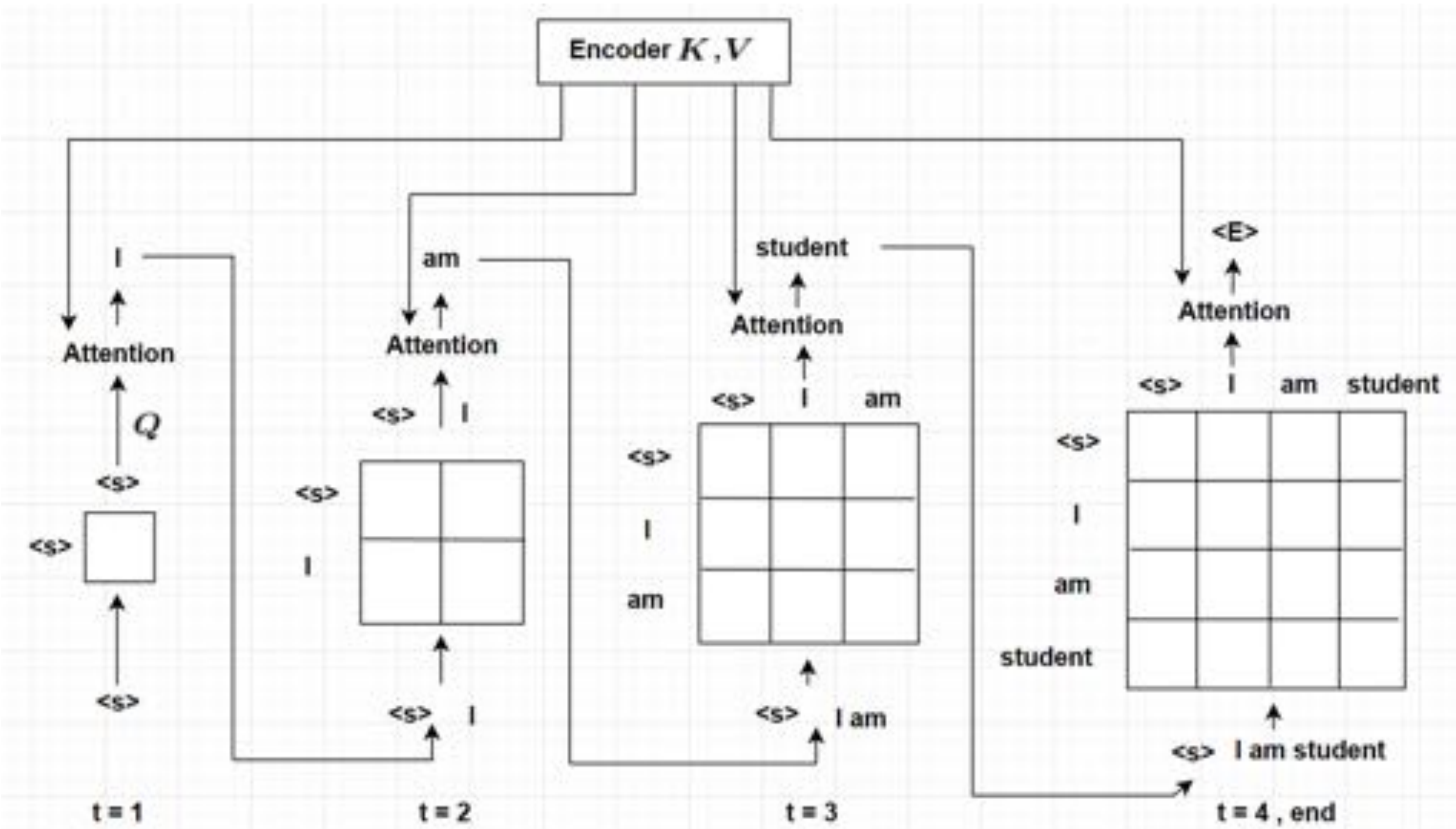


- Encoder-Decoder Attention
  - K, V from Encoder Last Layer
  - Q from Self Attention
- Beam Search



Attention Is All You Need, NuerIPS, 2017

Decoder in action



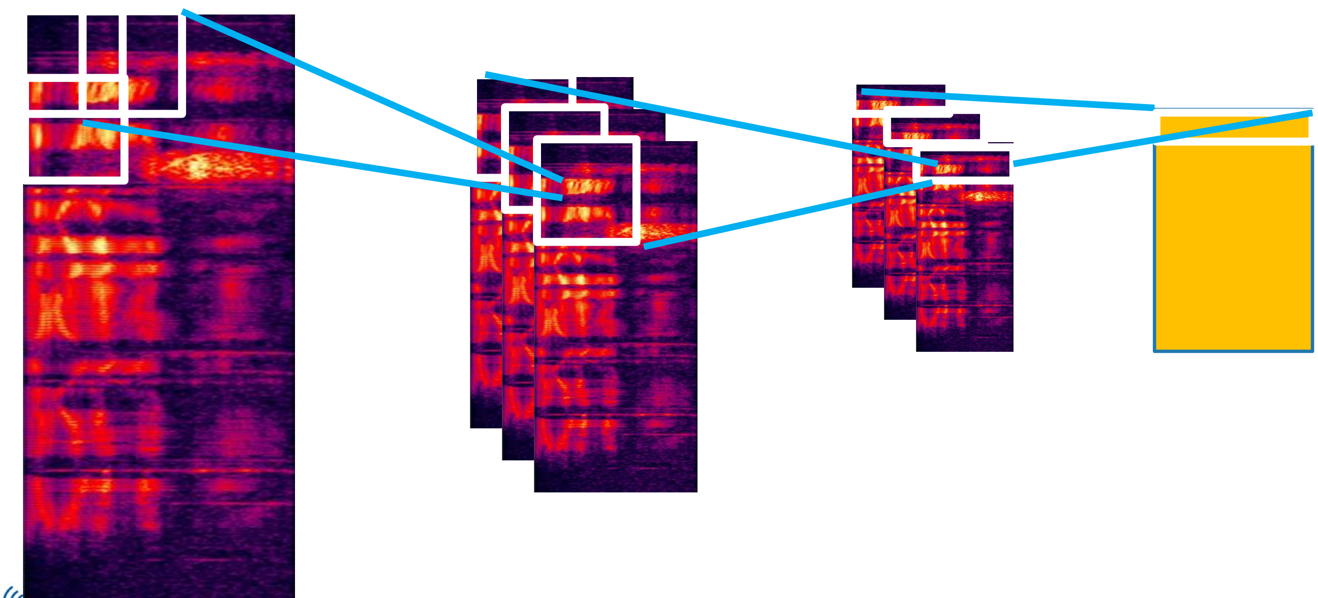
https://medium.com/platfarm/어텐션-메커니즘과-transfomer-self-attention-842498fd3225

End-to-end For ASR

- ESPNet: End-to-end Speech Processing Toolkit
  - ASR, TTS, Speech Translation
  - https://github.com/espnet/espnet
- CTC Hybrid*: Connectionist Temporal Classification
  - Multi-task training with CTC Criteiron
  - Increase Stability while Training
  - Hybrid CTC/Attention Architecture for End-to-End Speech Recognition, IEEE Journal of Selected Topics in Signal Processing, 2018
- Input Embedding

Input embedding

- TEXT: Input = Word: One-hot → Vector
- ASR: Input = MELFB: Vector → Vector
- 2x Conv2d layer, 3x3 kernel with stride=2
  - TxF → adim x T/2 x F/2 → adim x T/4 x F/4 → T/4 x adim



End-to-End ASR In Practice

- Output Units
  - 영어: Alphabet, BPE(Byte Pair Encoding), Word
  - 한국어: Char(음절~2500), BPE(~5000), 형태소분석기
- Relative Performance
  - WER/CER
  - 25% (GMM-HMM) → 15% (DNN-HMM) → 10% (LSTM-HMM)
  - 7% Transformer
- Limitation
  - Process Whole Sentence → Streaming ASR

강의내용: 3일차

- End-to-End ASR In Practice
- ASR 성능 개선 방안
- 실습
  - 훈련된 모델의 평가(공통평가셋)
  - 개인별 평가셋을 이용한 평가
- Recent trends in ASR
- 실습
  - Wrapup and Backup
  - Q&A

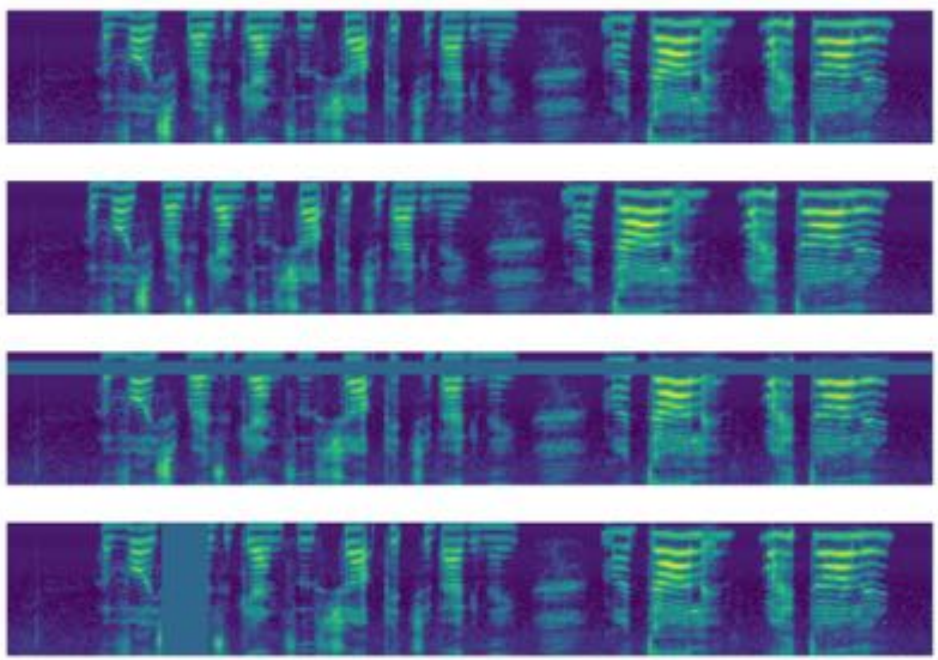
```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.51.06    Driver Version: 450.51.06    CUDA Version: 11.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  A100-SXM4-40GB      On   | 00000000:07:00.0 Off |                    0 |
| N/A   32C    P0    58W / 400W |      3MiB / 40537MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   1  A100-SXM4-40GB      On   | 00000000:0F:00.0 Off |                    0 |
| N/A   32C    P0    64W / 400W |  11550MiB / 40537MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   2  A100-SXM4-40GB      On   | 00000000:47:00.0 Off |                    0 |
| N/A   51C    P0   137W / 400W |  21953MiB / 40537MiB |     98%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   3  A100-SXM4-40GB      On   | 00000000:4E:00.0 Off |                    0 |
| N/A   51C    P0   184W / 400W |  33058MiB / 40537MiB |    100%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   4  A100-SXM4-40GB      On   | 00000000:87:00.0 Off |                    0 |
| N/A   57C    P0   245W / 400W |  33178MiB / 40537MiB |     99%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   5  A100-SXM4-40GB      On   | 00000000:90:00.0 Off |                    0 |
| N/A   55C    P0   225W / 400W |  33422MiB / 40537MiB |    100%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   6  A100-SXM4-40GB      On   | 00000000:B7:00.0 Off |                    0 |
| N/A   45C    P0    63W / 400W |      3MiB / 40537MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   7  A100-SXM4-40GB      On   | 00000000:BD:00.0 Off |                    0 |
| N/A   45C    P0    59W / 400W |      3MiB / 40537MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    1   N/A  N/A    232775      C   python3                         11547MiB |
|    2   N/A  N/A    174416      C   python3                         10723MiB |
|    2   N/A  N/A    231070      C   python3                         11227MiB |
|    3   N/A  N/A     37989      C   python3                         11107MiB |
|    3   N/A  N/A     67856      C   python3                         11227MiB |
|    3   N/A  N/A    235696      C   python3                         10721MiB |
|    4   N/A  N/A    209694      C   python3                         11227MiB |
|    4   N/A  N/A    233410      C   python3                         11227MiB |
|    4   N/A  N/A    240193      C   python3                         10721MiB |
|    5   N/A  N/A    199164      C   python3                         11227MiB |
|    5   N/A  N/A    242124      C   python3                         10069MiB |
|    5   N/A  N/A    244836      C   python3                         12123MiB |
+-----------------------------------------------------------------------------+
```

성능개선방안

- 데이터!
  - 실환경 데이터수집: 적응훈련/연결학습
  - 음향모델/언어모델?
- 데이터!!
  - 데이터 증강
  - SpecAug, Speed/Volume perturbation, Noise addition, Simulated data
- 모델 파라미터
  - Number of epoch
  - Number of paramerters: layers, dimension etc
  - Gradient scale: batchsize, learning rate etc
  - Robustness: dropout rate,

SpecAug

- SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition (2019)
- https://ai.googleblog.com/2019/04/specaugment-new-data-augmentation.html



input

time warp

frequency masking

time masking

Hyperparameters (1)

- Hyperparameter experiments on end-to-end automatic speech recognition (말소리와 음성과학, 2021)

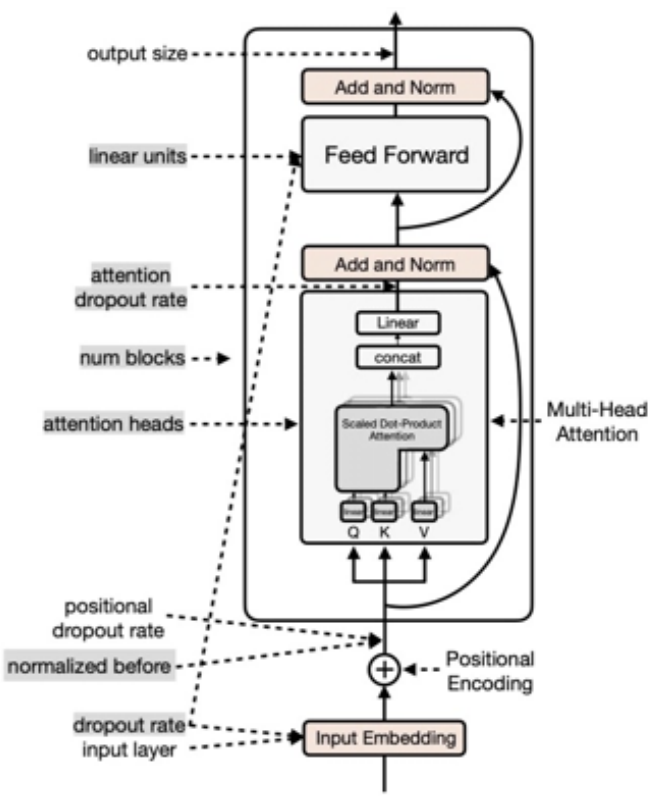


Table 1. The range of hyperparameters in the transformer encoder network

Hyperparameters	Values				
output size	**256**				
input layer	**2d conv**				
normalized before	**True**			false	
attention heads	1	**2**	4	8	
linear units	512	1,024	**2,048**	4,096	
num blocks	2	4	6	8	**12**
dropout rate	0.0	**0.1**	0.2	0.3	0.4
positional dropout rate	**0.1**				
attention dropout rate	**0.0**	0.1	0.2	0.3	0.4

Table 2. The range of transformer decoder network hyperparameters

Hyperparameters	Values				
attention heads	1	**2**	4	8	
linear units	512	1,024	**2,048**	4,096	
num blocks	2	4	**6**	8	12
dropout rate	0.0	**0.1**	0.2	0.3	0.4
positional dropout rate	**0.1**				
self attention dropout rate	**0.0**	0.1	0.2	0.3	0.4
src attention dropout rate	**0.0**				

Hyperparameters (2)

Table 3. The range of the model hyperparameters

Hyperparameters	Values				
batch type	**folded**				
batch size	**32**				
accum grad	**8**				
max epoch	**50**				
patience	**none**				
init	chainer	**xavier uniform**	xavier normal	kai-minguni-form	kaiming normal
optim	**adam**				
lr	**0.005**				
scheduler	**warmuplr**				
warmup steps	10,000	20,000	**30,000**	40,000	
keep nbest model	5	**10**	15	20	
ctc weight	0.0	0.1	0.2	**0.3**	0.4
lsm weight	0.0	**0.1**	0.2	0.3	0.4
length normalized loss	true		**false**		

Hyperparameters (3)

- Impact on WER (Example Only)

Table 4. WER from each hyperparameter model on WSJ dev93 and eval92

Hyperparameters		Values / WER				
MODEL	init	chainer	xavier uniform	xavier normal	kaiming uniform	kaiming normal
		42.0/35.1	**17.0/12.7**	17.3/14.0	17.7/13.1	17.6/13.4
	warmup steps	10,000	20,000	30,000	40,000	
		15.6/12.4	16.0/12.8	17.3/12.7	17.3/13.6	
	keep nbest model	5	10	15	20	
		17.0/12.8	17.3/12.7	16.9/13.0	16.9/13.4	
	ctc weight		0.1	0.2	0.3	0.4
			17.3/13.6	17.0/13.1	**17.3/12.7**	16.5/13.0
	lsm weight		0.1	0.2	0.3	0.4
			17.3/12.7	17.3/13.2	17.5/12.9	18.0/13.3
	length normalized loss	true	false			
		17.7/14.0	**17.3/12.7**			
ENCODER	attention heads	1	2	4	8	
		17.6/13.3	**16.7/13.0**	17.3/12.7	17.6/13.4	
	linear units	512	1,024	2,048	4,096	
		18.9/14.8	18.0/14.0	17.3/12.7	**16.3/12.3**	
	num blocks	2	4	6	8	12
		24.5/19.7	20.2/16.0	18.5/15.0	17.3/13.5	**17.3/12.7**
	dropout rate	0.0	0.1	0.2	0.3	0.4
		17.4/14.4	**17.3/12.7**	17.7/13.4	17.8/13.7	20.4/15.7
	attention dropout rate	0.0	0.1	0.2	0.3	0.4
		17.3/12.7	16.5/13.0	15.6/12.8	**15.8/12.6**	15.9/12.7
	normalized before	true	false			
		17.3/12.7	14.1			
DECODER	attention heads	1	2	4	8	
		17.3/13.0	17.1/12.9	**17.3/12.7**	17.6/12.8	
	linear units	512	1,024	2,048	4,096	
		17.7/13.5	17.5/13.4	17.2/12.7	**16.9/12.9**	
	num blocks	2	4	6	8	12
		19.7/16.0	17.2/13.6	17.3/12.7	**16.3/12.5**	16.3/12.5
	dropout rate	0.0	0.1	0.2	0.3	0.4
		16.5/13.3	**17.3/12.7**	16.9/13.8	16.3/13.6	17.5/13.5
	self attention dropout rate	0.0	0.1	0.2	0.3	0.4
		17.3/12.7	16.5/13.8	17.0/13.9	16.6/13.8	16.5/13.5

WER, word error rate.

Ongoing REsearches

- Semi/Un-Supervised Training
  - Training without labeled data
  - wav2vec 2.0, …
- Data augmentation
  - Generative models
- Transfer learning
  - Domain transfer
- Domain adaptation
- Streaming Transformer

wav2vec 2.0

- wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations (NeurIPS, 2020)
  - Constrative Loss

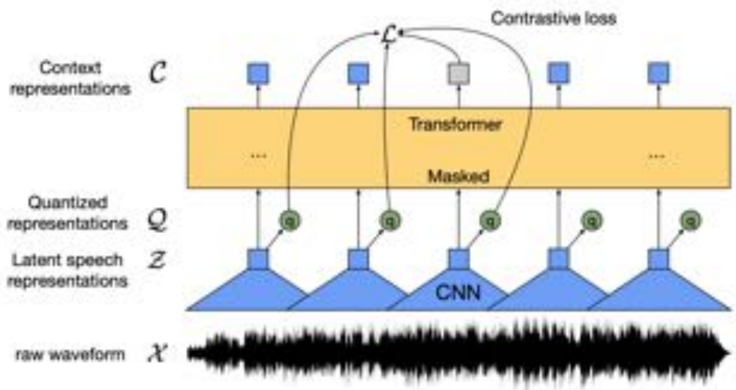


Figure 1: Illustration of our framework which jointly learns contextualized speech representations and an inventory of discretized speech units.

$$\mathcal{L} = \mathcal{L}_m + \alpha \mathcal{L}_d$$

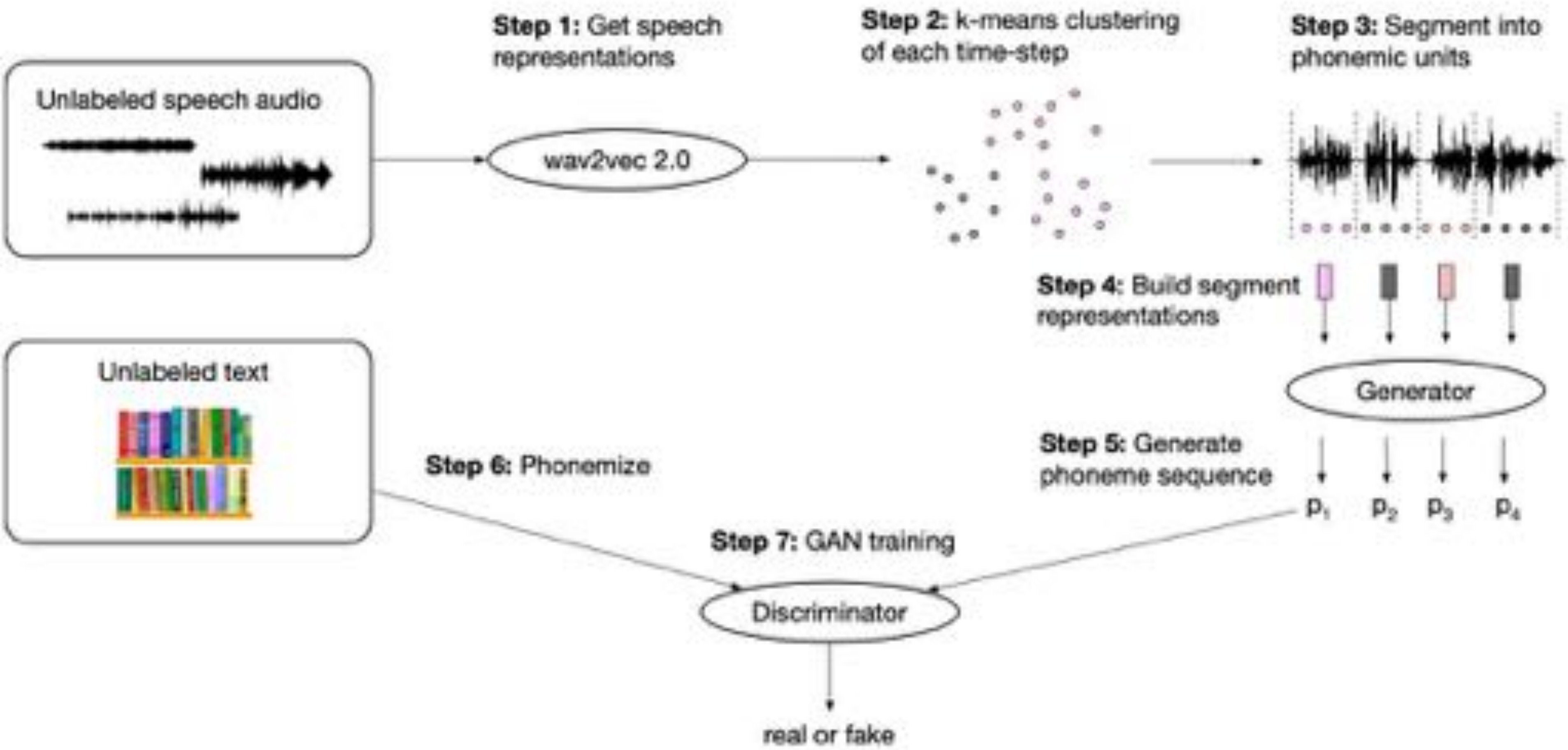
$$\mathcal{L}_m = -\log \frac{\exp(sim(\mathbf{c}_t, \mathbf{q}_t)/\kappa)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(sim(\mathbf{c}_t, \tilde{\mathbf{q}})/\kappa)}$$

$$\mathcal{L}_d = \frac{1}{GV} \sum_{g=1}^{G} -H(\bar{p}_g) = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \bar{p}_{g,v} \log \bar{p}_{g,v}$$

Unsupervised Speech Recognition

- NeurIPS, 2021, Alexei Baevski, Wei-Ning Hsu, Alexis CONNEAU, Michael Auli
- Motivation
  - 7,000 Languages > 125 STT Lang
  - Unsupervised NMT: Conneau et al 2018, …
  - A Framework for unsupervised learning of speech recognition

Architecture



Unsupervised Learning

- Objective
  - GAN loss
  - Gradient penalty
  - Segment smoothness penalty
  - Phoneme diversity loss

$$\min_{\mathcal{G}} \max_{\mathcal{C}} \mathop{\mathbb{E}}_{P^r \sim \mathcal{P}^r} [\log \mathcal{C}(P^r)] - \mathop{\mathbb{E}}_{S \sim \mathcal{S}} [\log (1 - \mathcal{C}(\mathcal{G}(S)))] - \lambda \mathcal{L}_{gp} + \gamma \mathcal{L}_{sp} + \eta \mathcal{L}_{pd}$$

$$\mathcal{L}_{gp} = \mathop{\mathbb{E}}_{\tilde{P} \sim \tilde{\mathcal{P}}} \left[\left(\| \nabla \mathcal{C}(\tilde{P}) \| - 1 \right)^2 \right]$$

$$\mathcal{L}_{sp} = \sum_{(p_t, p_{t+1}) \in \mathcal{G}(S)} \| p_t - p_{t+1} \|^2 \qquad \mathcal{L}_{pd} = \frac{1}{|B|} \sum_{S \in B} -H_{\mathcal{G}}(\mathcal{G}(S))$$

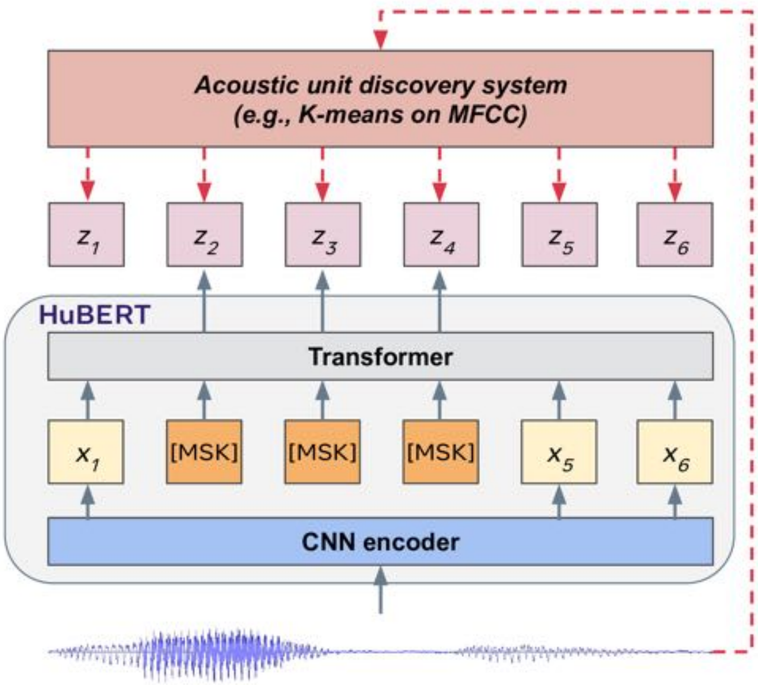
Experimental Results

- Comparison to Supervised Speech Recognition on Librispeech

Model	Unlabeled data	LM	dev clean	dev other	test clean	test other
960h - Supervised learning						
DeepSpeech 2 [Amodei et al., 2016]	-	5-gram	-	-	5.33	13.25
Fully Conv [Zeghidour et al., 2018]	-	ConvLM	3.08	9.94	3.26	10.47
TDNN+Kaldi [Xu et al., 2018]	-	4-gram	2.71	7.37	3.12	7.63
SpecAugment [Park et al., 2019]	-	RNN	-	-	2.5	5.8
ContextNet [Han et al., 2020]	-	LSTM	1.9	3.9	1.9	4.1
Conformer [Gulati et al., 2020]	-	LSTM	2.1	4.3	1.9	3.9
960h - Self and semi-supervised learning						
Transf. + PL [Synnaeve et al., 2020]	LL-60k	CLM+Transf.	2.00	3.65	2.09	4.11
IPL [Xu et al., 2020b]	LL-60k	4-gram+Transf.	1.85	3.26	2.10	4.01
NST [Park et al., 2020]	LL-60k	LSTM	1.6	3.4	1.7	3.4
wav2vec 2.0 [Baevski et al., 2020c]	LL-60k	Transf.	1.6	3.0	1.8	3.3
wav2vec 2.0 + NST [Zhang et al., 2020b]	LL-60k	LSTM	1.3	2.6	1.4	2.6
Unsupervised learning						
wav2vec-U LARGE	LL-60k	4-gram	13.3	15.1	13.8	18.0
wav2vec-U LARGE + ST	LL-60k	4-gram	3.4	6.0	3.8	6.5
	LL-60k	Transf.	3.2	5.5	3.4	5.9

Hu-BERT

- HuBERT: **Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units**
  - IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021



		BASE	LARGE	X-LARGE
CNN Encoder	strides	5, 2, 2, 2, 2, 2, 2		
	kernel width	10, 3, 3, 3, 3, 2, 2		
	channel	512		
Transformer	layer	12	24	48
	embedding dim.	768	1024	1280
	inner FFN dim.	3072	4096	5120
	layerdrop prob	0.05	0	0
	attention heads	8	16	16
Projection	dim.	256	768	1024
Num. of Params		95M	317M	964M

TABLE I: Model architecture summary for BASE, LARGE, and X-LARGE HuBERT models

Hu-BERT: Results

Model	Unlabeled Data	LM	dev-clean	dev-other	test-clean	test-other
		10-min labeled				
DiscreteBERT [51]	LS-960	4-gram	15.7	24.1	16.3	25.2
wav2vec 2.0 BASE [6]	LS-960	4-gram	8.9	15.7	9.1	15.6
wav2vec 2.0 LARGE [6]	LL-60k	4-gram	6.3	9.8	6.6	10.3
wav2vec 2.0 LARGE [6]	LL-60k	Transformer	4.6	7.9	4.8	8.2
HUBERT BASE	LS-960	4-gram	9.1	15.0	9.7	15.3
HUBERT LARGE	LL-60k	4-gram	6.1	9.4	6.6	10.1
HUBERT LARGE	LL-60k	Transformer	4.3	7.0	4.7	7.6
HUBERT X-LARGE	LL-60k	Transformer	4.4	6.1	4.6	6.8
		1-hour labeled				
DeCoAR 2.0 [50]	LS-960	4-gram	-	-	13.8	29.1
DiscreteBERT [51]	LS-960	4-gram	8.5	16.4	9.0	17.6
wav2vec 2.0 BASE [6]	LS-960	4-gram	5.0	10.8	5.5	11.3
wav2vec 2.0 LARGE [6]	LL-60k	Transformer	2.9	5.4	2.9	5.8
HUBERT BASE	LS-960	4-gram	5.6	10.9	6.1	11.3
HUBERT LARGE	LL-60k	Transformer	2.6	4.9	2.9	5.4
HUBERT X-LARGE	LL-60k	Transformer	2.6	4.2	2.8	4.8
		10-hour labeled				
SlimIPL [54]	LS-960	4-gram + Transformer	5.3	7.9	5.5	9.0
DeCoAR 2.0 [50]	LS-960	4-gram	-	-	5.4	13.3
DiscreteBERT [51]	LS-960	4-gram	5.3	13.2	5.9	14.1
wav2vec 2.0 BASE [6]	LS-960	4-gram	3.8	9.1	4.3	9.5
wav2vec 2.0 LARGE [6]	LL-60k	Transformer	2.4	4.8	2.6	4.9
HUBERT BASE	LS-960	4-gram	3.9	9.0	4.3	9.4
HUBERT LARGE	LL-60k	Transformer	2.2	4.3	2.4	4.6
HUBERT X-LARGE	LL-60k	Transformer	2.1	3.6	2.3	4.0
		100-hour labeled				
IPL [12]	LL-60k	4-gram + Transformer	3.19	6.14	3.72	7.11
SlimIPL [54]	LS-860	4-gram + Transformer	2.2	4.6	2.7	5.2
Noisy Student [61]	LS-860	LSTM	3.9	8.8	4.2	8.6
DeCoAR 2.0 [50]	LS-960	4-gram	-	-	5.0	12.1
DiscreteBERT [51]	LS-960	4-gram	4.0	10.9	4.5	12.1
wav2vec 2.0 BASE [6]	LS-960	4-gram	2.7	7.9	3.4	8.0
wav2vec 2.0 LARGE [6]	LL-60k	Transformer	1.9	4.0	2.0	4.0
HUBERT BASE	LS-960	4-gram	2.7	7.8	3.4	8.1
HUBERT LARGE	LL-60k	Transformer	1.8	3.7	2.1	3.9
HUBERT X-LARGE	LL-60k	Transformer	1.7	3.0	1.9	3.5

TABLE II: Results and comparison with the literature on low resource setups (10-min, 1-hour, 10-hour, and 100-hour of labeled data).

AV-HuBERT

- https://github.com/facebookresearch/av_hubert
- Learning Audio-Visual Speech Representation by Masked Multimodal Cluster Prediction, 2022
- Robust Self-Supervised Audio-Visual Speech Recognition

AV-HuBERT: Architecture

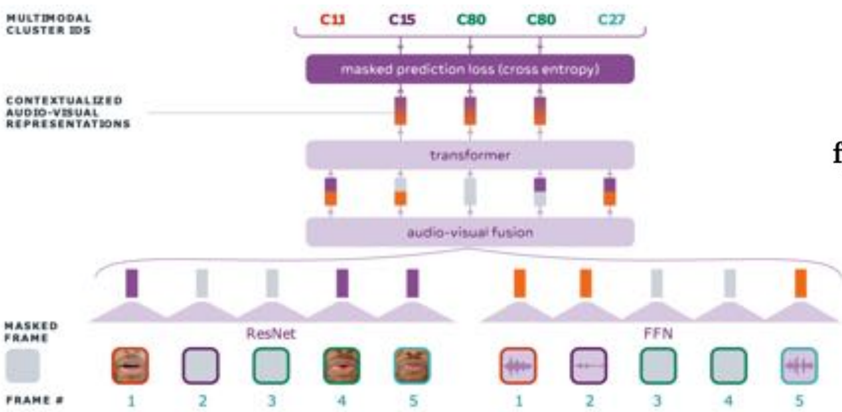


Figure 1: Illustration of AV-HuBERT. Masked prediction losses are only computed for the three middle frames, because at least one modality is masked for those frames. See section A for its comparison between single-modal and cross-modal visual HuBERT.

$$\mathbf{f}_t^{av} = \begin{cases} \texttt{concat}(\mathbf{f}_t^a, \mathbf{f}_t^v) & \text{with } p_m \\ \texttt{concat}(\mathbf{f}_t^a, \mathbf{0}) & \text{with } (1-p_m)p_a \\ \texttt{concat}(\mathbf{0}, \mathbf{f}_t^v) & \text{with } (1-p_m)(1-p_a) \end{cases}$$

AV-ASR

Figure 1: *AV-HuBERT for audio-visual speech recognition. X: mask; blue waveform: original audio; orange waveform: noise; C_n: audio-visual clusters. Dashed box: the pre-trained part*

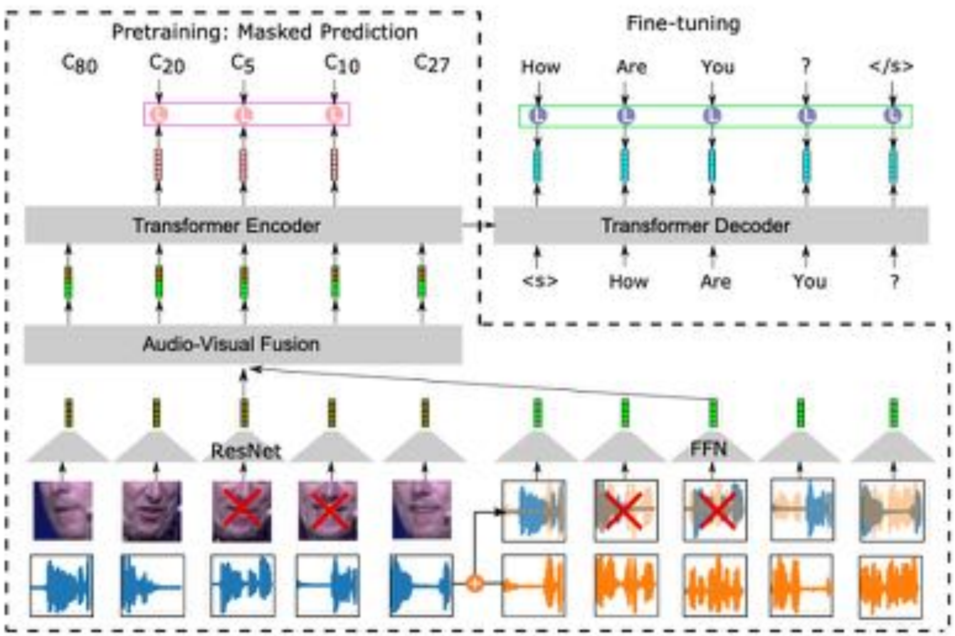
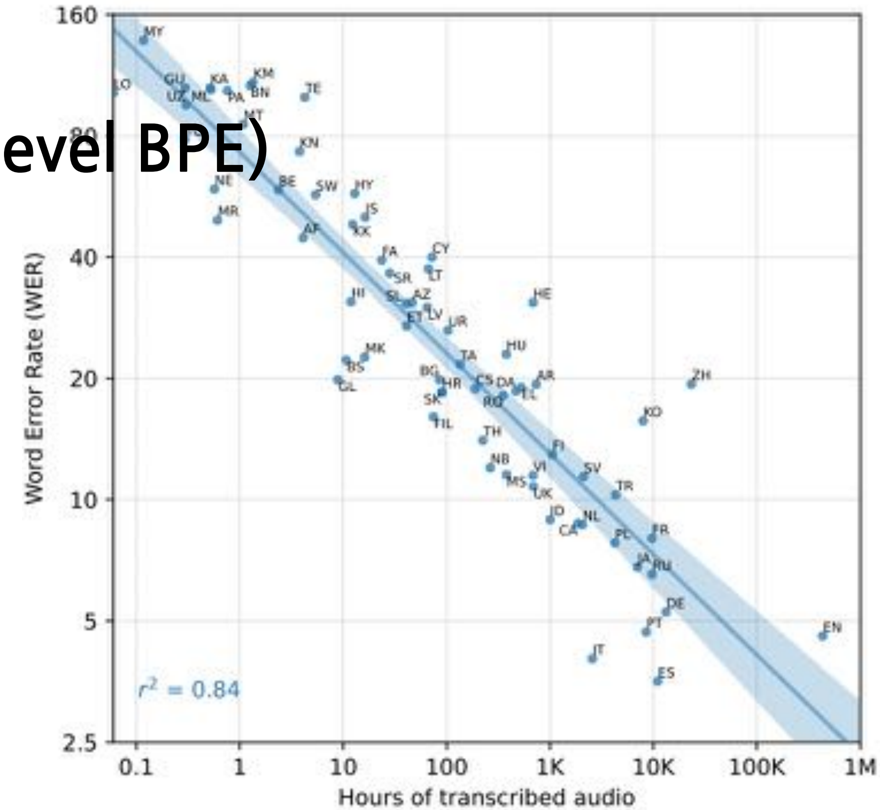


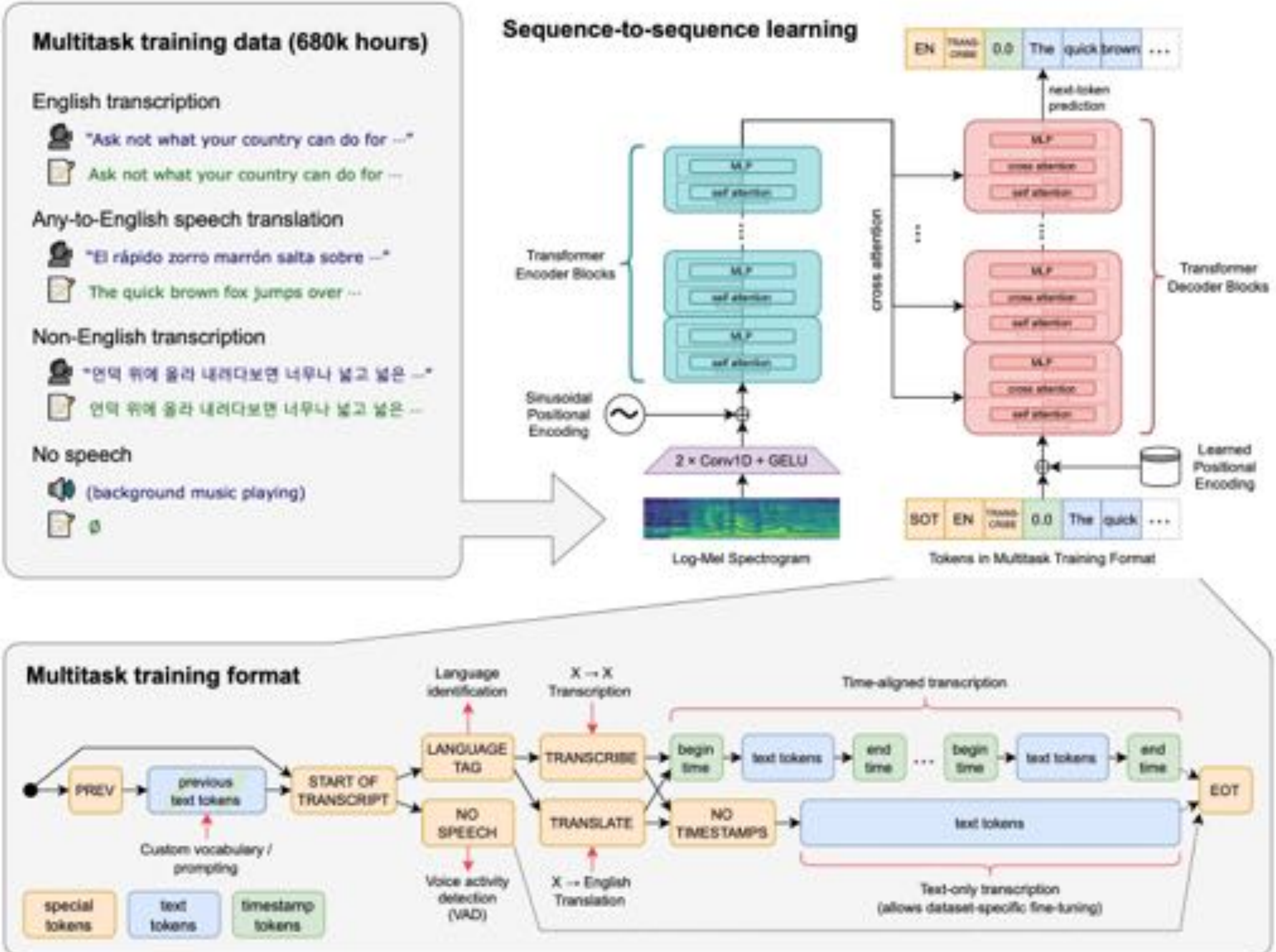
Table 3: *Comparison among models with different pre-training configurations and input modalities. C: clean audio, N: noisy audio. The N-WER is averaged over 4 noise types and 5 SNRs.*

Model Size	PT Type	FT Data	Audio-only		Audio-visual	
			C-WER	N-WER	C-WER	N-WER
(a). LARGE	None	30h	20.6	59.2	20.8	42.9
(b). LARGE	Clean	30h	4.3	39.8	3.3	9.3
(c). LARGE	Noisy	30h	3.8	28.7	3.3	7.8
(d). LARGE	None	433h	4.7	39.2	3.5	14.8
(e). LARGE	Clean	433h	1.5	29.1	1.4	6.9
(f). LARGE	Noisy	433h	1.6	25.8	1.4	5.8

OpenAI Whisper

- Robust Speech Recognition via Large-Scale Weak Supervision, 2022
- https://github.com/openai/whisper
- 680k audio
  - 117k = 96 languages (byte level BPE)
  - 125k = X → en
- Multitask Learning
  - Multi-lingual ASR
  - Translation
  - Language Identification





Multitask training data (680k hours)
English transcription
"Ask not what your country can do for ···"
Ask not what your country can do for ···
Any-to-English speech translation
"El rápido zorro marrón salta sobre ···"
The quick brown fox jumps over ···
Non-English transcription
"언덕 위에 올라 내려다보면 너무나 넓고 넓은 ···"
언덕 위에 올라 내려다보면 너무나 넓고 넓은 ···
No speech
(background music playing)
∅
Sequence-to-sequence learning
EN
TRANS-CRIBE
0.0
The
quick
brown
···
next-token prediction
MLP
cross attention
self attention
Transformer Encoder Blocks
Transformer Decoder Blocks
cross attention
Sinusoidal Positional Encoding
2 × Conv1D + GELU
Log-Mel Spectrogram
Learned Positional Encoding
SOT
Tokens in Multitask Training Format
Multitask training format
Language identification
X → X Transcription
Time-aligned transcription
PREV
previous text tokens
START OF TRANSCRIPT
LANGUAGE TAG
NO SPEECH
TRANSCRIBE
TRANSLATE
begin time
text tokens
end time
NO TIMESTAMPS
EOT
Custom vocabulary / prompting
Voice activity detection (VAD)
X → English Translation
Text-only transcription (allows dataset-specific fine-tuning)
special tokens
text tokens
timestamp tokens

Discussion and Q&A