Wav2vec 2.0

Data labeling의 한계

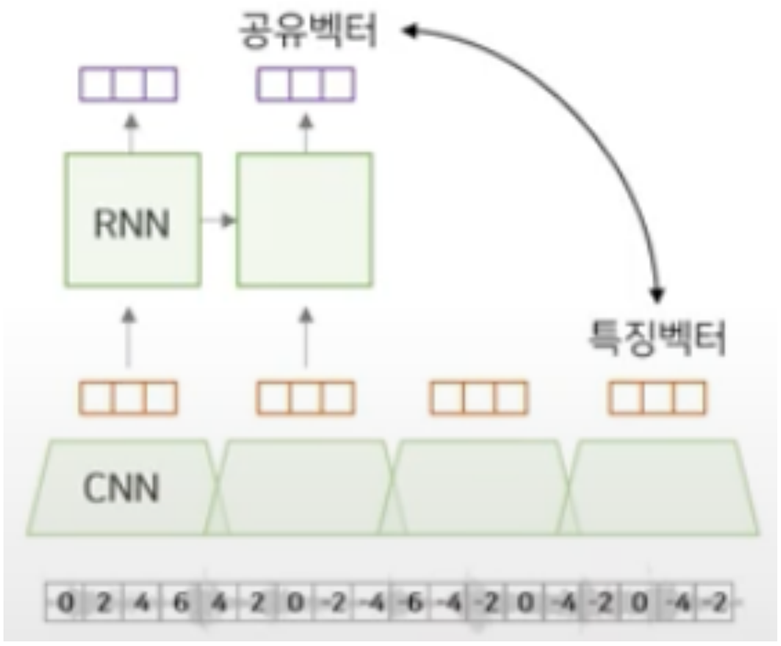
- Supervised learning : 고성능, but 데이터 수집의 한계
  - semi-supervised(사전모델에만 labeled data 사용), unsupervised learning(unlabeled data만 사용)
- Self-supervised: 자기 자신을 정답지 삼아 학습하는 방법
  - BERT: input sequence -> masking -> 주변 단어들 이용해 예측
  - (wav2vec 2.0 : 53,000시간 오디오 데이터로 self-supervised pre-training 진행, 이후 10분의 라벨링 데이터로 fine-tuning)

Self-supervised learning in ASR

- 음성 데이터만으로 음성의 특징을 잘 추출하는 것이 목표
  - 오디오를 가장 잘 대표(represent)할 수 있는 pre-trained model을 만들자.
- CPC -> wav2vec -> VQ-wav2vec -> wav2vec 2.0

CPC

- Contrastive Predictive Coding (CPC)
  - 데이터 내 공유정보를 추출하는 모델, 즉 pre-trained model을 목표로 함.
    - 공유정보 : 감성, 품사 등 데이터 상에서 어느 정도 공유되고 느리게 변하는 정보



Encoder (CNN) : 일정길이 음성을 특징벡터로 변환
Aggregator (RNN) : 특징벡터를 공유벡터로 변환

공유벡터는 이전시점-미래시점 특징벡터의 공유정보로 훈련된다.

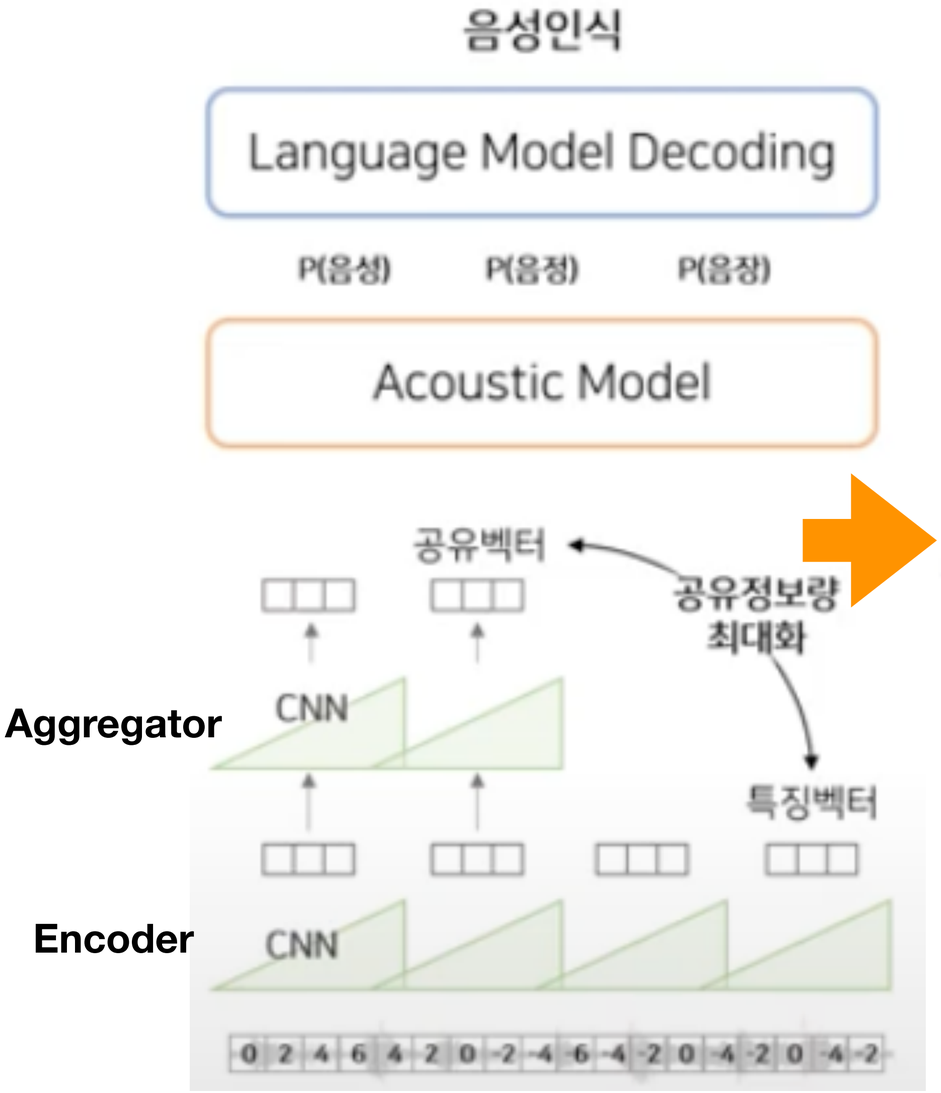
=> 추출된 feature vector를 활용해 각종 task를 수행한다.

장점) 음성 데이터만으로 음성 feature 추출 가능, 여러 분야서 적용 가능
단점) RNN 병렬처리 불가로 속도 문제, 음성 도메인에서 성능이 아쉬움

wav2vec

- CPC를 음성 분야에 적합하도록 적용한 논문.
- Input: raw audio / Output : general representation of audio
  - RNN to CNN => 병렬처리 가능, 속도 개선
  - 추출된 representation => AM, LM 성능향상에 사용
    - AM: 오디오 신호-음소 간 관계를 모델링하는 모델.
    - LM: 텍스트만 활용, 맥락을 고려해 특정 단어가 나올 확률을 도출하는 모델.
      - ex. p(음성학)= p(음) + p(성|음) + p(학|음,성)

wav2vec



Contrastive task : (주변정보로 복원된) 공유벡터와 특징벡터가 최대한 유사하도록 학습한다.

word2vec의 negative sampling 방식처럼, 앞으로 올 오디오 데이터를 예측

$$\mathcal{L}_k = -\sum_{i=1}^{T-k} \left(log\sigma(\mathbf{z}_{i+k}^{\top} h_k(c_i)) + \lambda \mathbb{E}_{\tilde{\mathbf{z}} \sim p_n} [log\sigma(-\tilde{\mathbf{z}}^{\top} h_k(c_i))] \right)$$

Z로부터 k개 뒤 sample이 positive sample일 확률

Z~(negative sample)이 negative sample일 확률

wav2vec

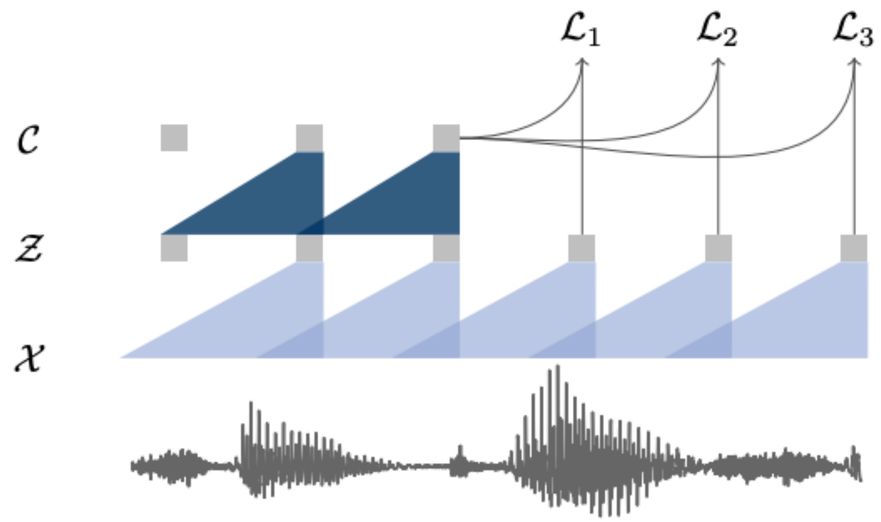
간단한 예시

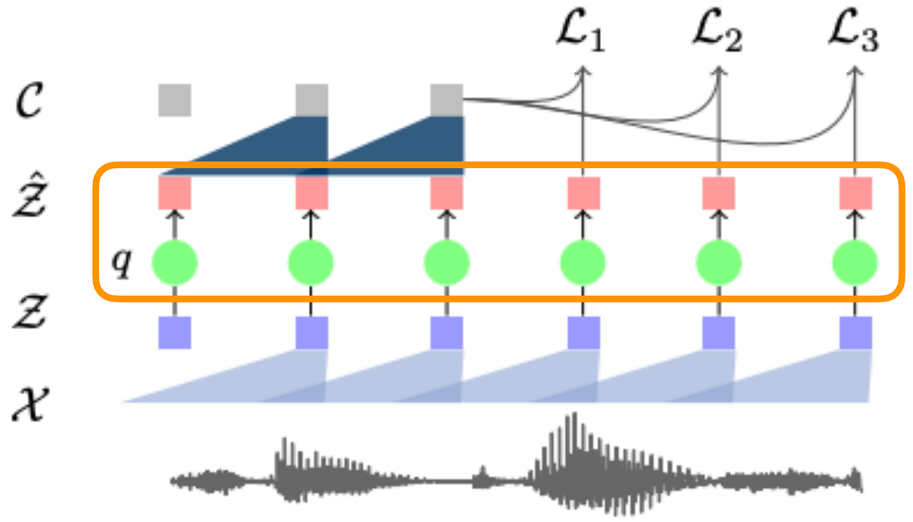
	w	p(음성)	p(음정)	p(음장)
Language Model	0.2	0.4	0.2	0.2
Acoustic Model	0.4	0.2	0.2	0.4
		0.16	0.12	**0.20**

Decoding : AM/LM으로부터 특정 단어의 도출 확률을 구한 뒤
가장 높은 확률의 단어를 선택한다.

VQ-wav2vec

- BERT의 MLM(masked language model, pre-training 단계)을 음성 분야에 적용
  - 발화된 텍스트 간 sequence 패턴 학습으로 ASR 성능 향상
  - BERT에서와 마찬가지로 음성 신호의 이산화 필요.
  - (encoder-aggregator 기본모델) + Vector Quantization 모듈 = VQ-wav2vec

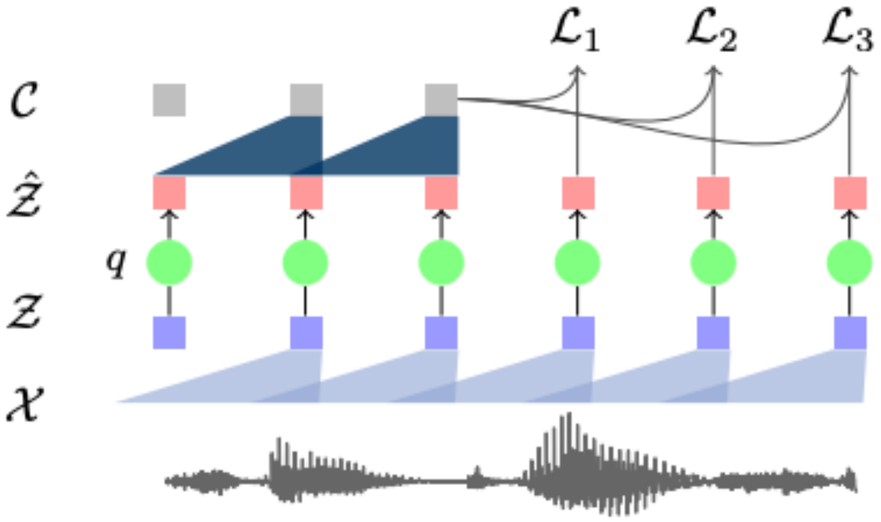


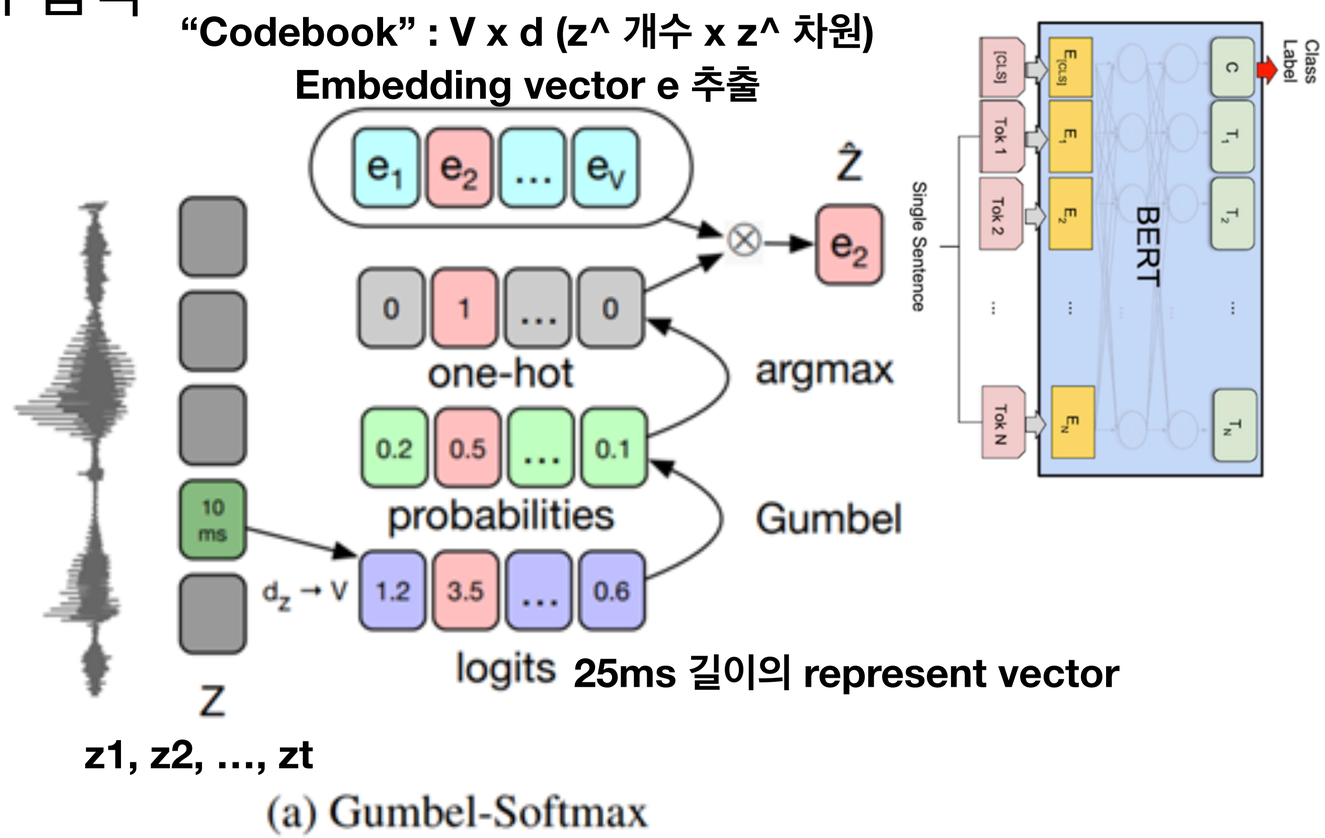


VQ-wav2vec

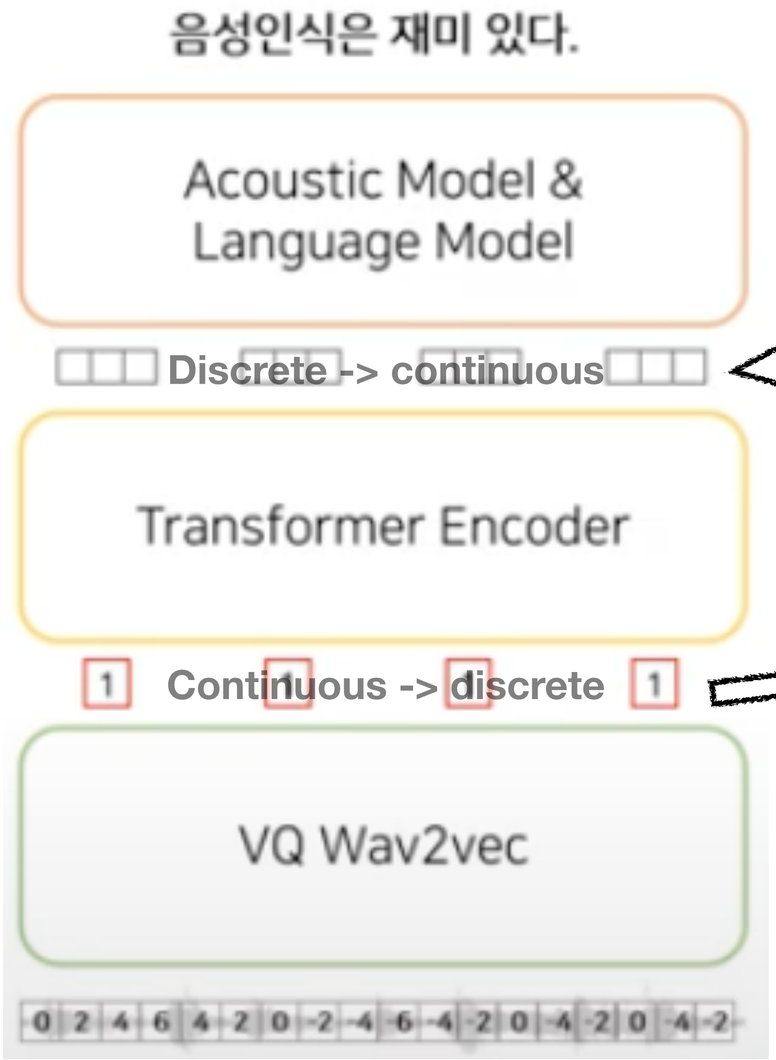
- VQ 모듈 : Gumbel-Softmax / k-means clustering 적용, representation을 이산화 (index를 나타냄).

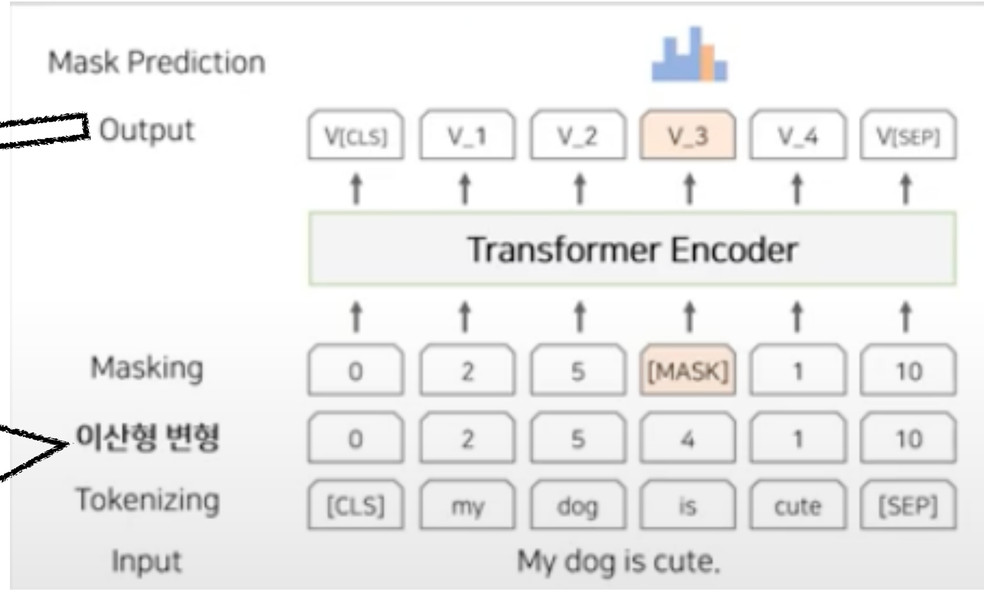
- 이산화 시스템을 통해 이후 BERT layer에 입력





VQ-wav2vec



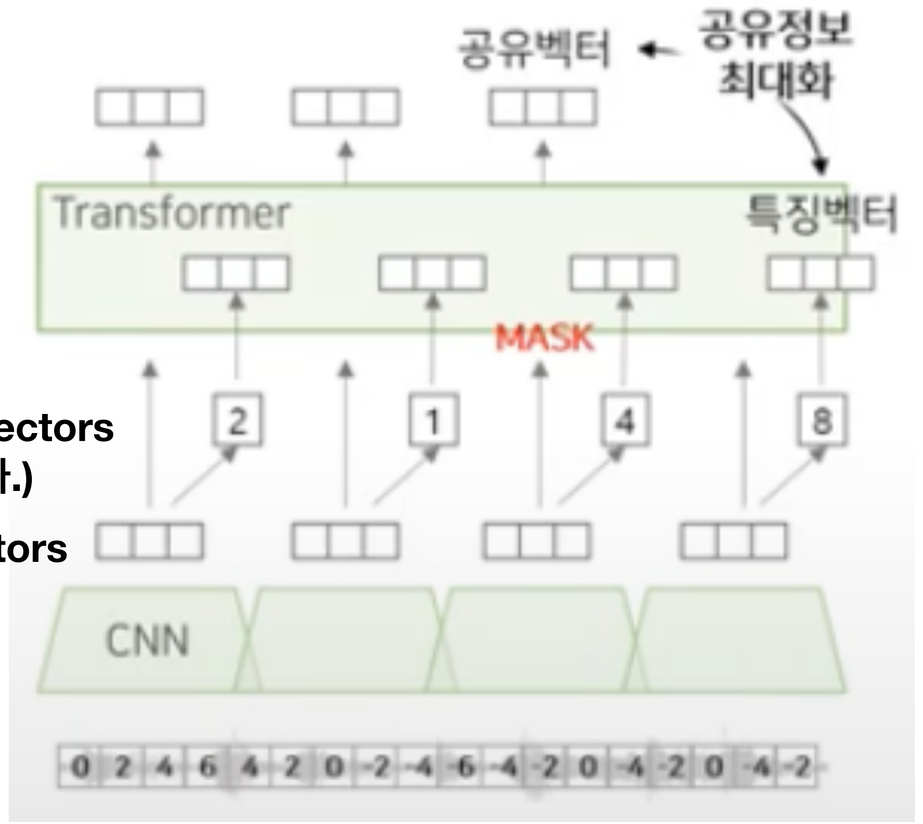


3가지 모듈을 따로

wav2vec 2.0

- Pre-train 단계에서 MLM+CPC 함께 적용
  - Aggregator: CNN => Transformer
  - 특징벡터는 codebook indexing 통해서가 아닌,

    transformer 통해 나오게 된다.



wav2vec 2.0 vs BERT

- BERT와 유사하게, 마스킹된 speech unit을 예측하고 스스로 훈련한다.
- 단, 음성 신호는 단어 등의 단위로 분할될 수 없는 연속 신호이다.
  - 따라서 25ms 길이의 represent vector z1, z2, …., zT를 이산화한 값이 마치 token input으로 여겨지게 된다.
- wav2vec에서는 NSP 없이 MLM만 사용한다.
- wav2vec encoder input은 매우 짧은 오디오 토큰 (ex. 10ms)이므로 임의로 masking 시 예측이 쉽다는 단점이 있다.
  - 따라서 오디오 토큰 여러개를 뭉쳐서 (ex. 10ms x 10개 = 100ms) masking을 진행한다.

- 출처

[1] wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations https://arxiv.org/abs/2006.11477

[2] vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations https://arxiv.org/abs/1910.05453

[3] wav2vec: Unsupervised Pre-training for Speech Recognition https://arxiv.org/abs/1904.05862

[4] [Paper Review] Vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations https://www.youtube.com/watch?v=mPtyfqWHs3s

[5] https://nongnongai.tistory.com/m/34

[6] https://kaen2891.tistory.com/82

[7] https://sylim2357.github.io/paper%20review/wav2vec/

[8] https://sylim2357.github.io/paper%20review/vq-wav2vec/