WHISPER 분석

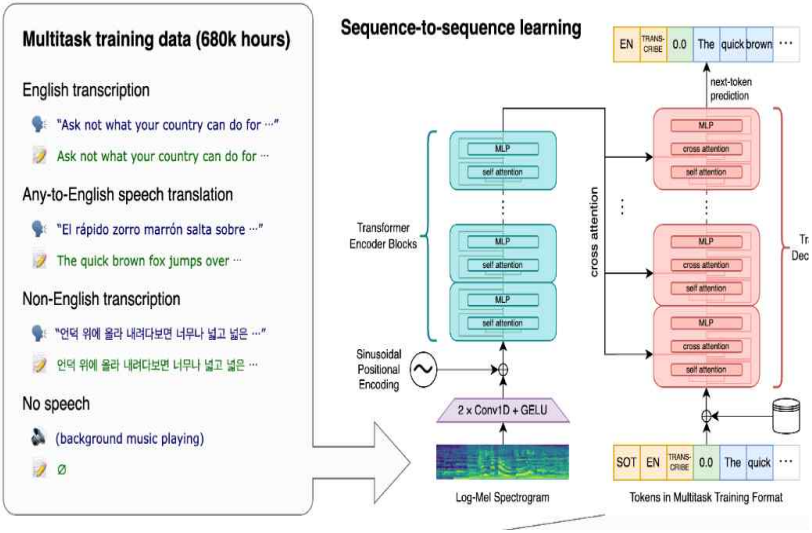
-Transformer 모델과 Attention 매커니즘 개념 기초

지도교수 : 김선만
발표자 : AINC 202158007 노진산
(2025.01.06)

Contents

WHISPER에 대한 간단한 개념 이해

자동적인 음성 인식 모델



트랜스포머 기반의 인코더-디코더 아키텍쳐 사용

-처리 과정 :

1. 음성 데이터를 Mel-Spectrogram으로 변환
2. 인코더 : 음성 특징 추출
3. 디코더 : 텍스트 생성 및 다양한 작업 수행

-응용 분야 :

자동 자막 생성
다국어 실시간 번역
음성 기반 인터페이스 개발
회의록 자동 작성

https://github.com/openai/whisper

Sequence To Sequence 학습 과정

WHISPER의 학습 구조

1. 입력 데이터 처리

-680,000시간 분량의 멀티태스크 학습 데이터 -사용
-영어 전사(transcription), 영어 번역, 비영어 전사 등 다양한 작업 포함
-배경 음악이나 무음 구간도 학습 데이터로 활용

2. 오디오 신호 변환

-입력된 오디오를 Log-Mel Spectrogram으로 변환
-2개의 Conv1D 레이어와 GELU 활성화 함수를 통해 특징 추출
-Sinusoidal Positional Encoding을 적용하여 시퀀스의 위치 정보 보존

3. 인코더 처리

-Transformer Encoder Blocks에서 처리
-각 블록은 MLP(Multi-Layer Perceptron)와 Self-Attention 레이어로 구성
-여러 개의 인코더 블록을 통해 깊은 특징 추출

4. 디코더 처리

-Transformer Decoder Blocks에서 텍스트 생성
-MLP, Cross-Attention, Self-Attention 레이어로 구성
-토큰 단위로 순차적 예측 수행

5. 출력 생성

-다음 토큰을 예측하는 방식으로 텍스트 생성
-특수 토큰(EN, TRANSCRIBE 등)을 통해 작업 유형 지정
-멀티태스크 학습을 통해 다양한 출력 형식 지원

Attention 매커니즘에 대한 이해

입력 시퀀스의 각 요소들 간의 관계성을 학습하는 핵심 메커니즘

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Query(Q) :
현재 처리하고자 하는 정보
"무엇을 찾고 있는가?" 나타냄

Key(K) :
참조할 수 있는 정보들의 집합
Query와의 연관성을 측정하는 기준

Value(V) :
실제 정보를 담고 있는 벡터
최종 출력에 반영될 내용

인코더에서의 Self-Attention

-음성 데이터의 시간적 특징 추출
-Transformer Encoder Blocks 내 MLP와 결합
-장거리 의존성 포착으로 음성 문맥 이해

디코더에서의 Attention

-Self-Attention: 생성된 텍스트 내 관계성 학습
-Cross-Attention: 음성-텍스트 간 매핑 수행
-순차적 토큰 생성으로 정확한 전사 실현

Attention 매커니즘의 계산 과정

Query, Key, Value를 이용하여 관련성을 계산하고 중요한 부분에 집중하는 방식

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Query(Q) :
현재 처리하고자 하는 정보, 찾고자 하는 정보를 담은 텐서
“무엇을 찾고 있는가?” 나타냄

Key(K) :
참조할 수 있는 정보들의 집합, 참조할 수 있는 정보들의 집합
Query와의 연관성을 측정하는 기준

Value(V) :
실제 정보를 담고 있는 벡터, 실제 정보를 담고 있는 벡터들
최종 출력에 반영될 내용

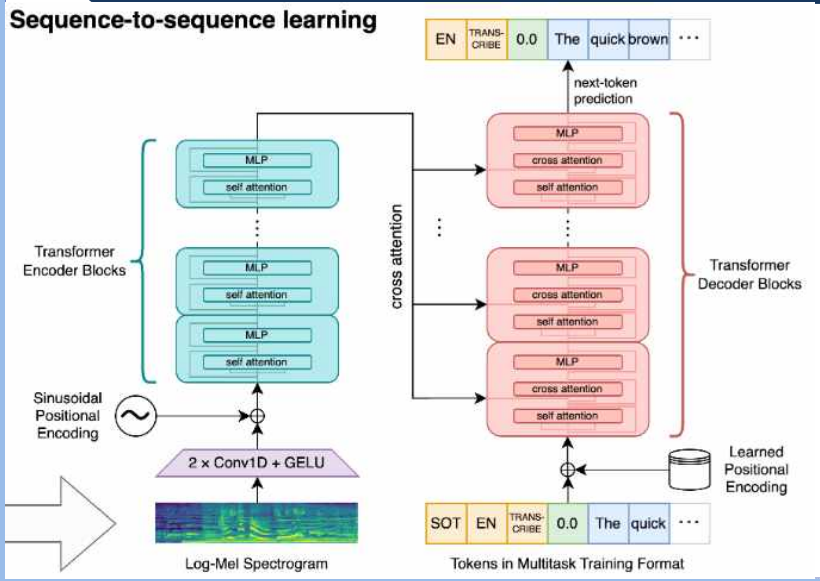
1.내적 계산
–Query와 각 Key 사이의 유사도를 내적으로 계산
–유사한 정보일수록 내적값이 크게 나옴

2.가중치 생성
–내적 결과에 지수함수를 적용하여 양수로 변환
–전체 합이 1이 되도록 정규화

3.최종 출력 생성
–계산된 가중치와 Value들의 가중합 계산
(Query와 유사한 Key에 해당하는 Value가 더 큰 비중을 차지)

Transformer 모델

트랜스포머 모델은 인코더-디코더 구조를 기반으로 하며, 시퀀스-투-시퀀스 학습을 수행



https://github.com/openai/whisper

–음성 데이터 변환

Mel–Spectrogram : 음성 신호를 시각적으로 표현하는 방법

–인코더 구조

Self–Attention : 입력 시퀀스 내의 모든 요소들 간의 관계를 학습, 각 요소가 다른 모든 요소와의 관련성을 계산

–디코더 구조

Cross–Attention :
Query : 디코더의 현재 상태에서 생성
Key, Value : 인코더의 최종 출력에서 생성
인코더의 정보를 디코더에서 출력 생성에서 활용

두 가지의 어텐션 매커니즘의 조합으로 효과적으로 시퀀스–투–스퀀스 학습을 수행할 수 있음

Transformer 모델 과 RNN, CNN의 차이점

RNN (Recurrent Neural Network)

–순차적 처리: 입력을 순서대로 하나씩 처리
–이전 상태의 정보를 hidden state로 전달
–장거리 의존성 학습의 어려움 존재
–병렬 처리가 불가능

CNN (Convolutional Neural Network)

–지역적 특징 추출에 특화
–컨볼루션 필터를 통한 특징 맵 생성
–고정된 크기의 수용 영역
–전체적인 문맥 파악이 상대적으로 어려움

트랜스포머 모델

–병렬 처리: 전체 시퀀스를 동시에 처리
–Self–attention을 통한 전역적 관계 학습
–위치 정보를 positional encoding으로 처리
–장거리 의존성 학습이 용이

–처리 속도

RNN: 순차적 처리로 인한 느린 속도

CNN: 병렬 처리 가능

트랜스포머: 완전한 병렬 처리로 가장 빠름

–메모리 효율성

RNN: 메모리 효율적

CNN: 중간 수준의 메모리 사용

트랜스포머: 큰 메모리 요구량

–학습 능력

RNN: 순차 데이터 학습에 적합하나 장거리 의존성 취약

CNN: 지역적 패턴 학습에 강점

트랜스포머: 전역적 관계 학습과 장거리 의존성 처리에 우수

감사합니다

Q&A

지도교수 : 김선만
발표자 : AINC 202158007노진산