Week5 - 분석 base

진행 상태	시작 전

25기 분석 인태영

Attentions

- Query: 현재 처리하고자 하는 vector
- Key: Query와 얼마나 비슷한지 알아보기 위해 유사도를 측정당하는 vector
- Value: Query와 Key 간의 유사도 측정을 통해 구해진 가중치가 적용되는 vector

Dot-product Attention

Query와 Key의 내적을 통해 단어 간 유사도를 산출하고, 이를 softmax 함수를 통해 확률 분포 형태로 변환한다. 이렇게 얻어진 가중치는 Value에 적용되어 가중합을 계산하며, 이 결과가 Attention Value가 된다.

Tranformer

Transformer는 먼저 입력 문장을 벡터로 바꾸는 임베딩 과정을 거친다. 각 토큰은 학습 가능한 행렬 W에서 해당하는 벡터를 찾아오는 방식으로 변환된다.

하지만 이것만으로는 단어의 순서 정보가 없기 때문에 Positional Encoding을 더해서 문맥과 순서를 동시에 반영할 수 있게 만든다.

인코더에서는 Self-Attention 메커니즘이 작동한다. 이를 통해 단어들이 서로 간의 관계를 스스로 계산한다. Scaled Dot-Product Attention으로 유사도를 안정적으로 구하고, Multi-Head Attention 구조를 사용해서 병렬로 여러 관점의 정보를 뽑아낸다. 각 head의 결과들을 연결한 다음 선형 변환을 거쳐 입력과 같은 크기의 벡터로 만든다.

디코더에서는. 미래의 단어를 미리 보면 안 되기 때문에 Masked Multi-Head Attention을 사용한다. 미래 위치의 점수를 -∞로 만들어서 softmax를 통과하면 0이 된다. 이를 통해 과

거 단어들만 보고 예측하게 만든다. 그리고 Position-wise Feed-Forward Network를 이용해 ReLU 활성화 함수와 두 번의 선형 변환을 통해 비선형성을 추가하면서도 차원은 그대로 유지한다.

그리고 학습을 안정시키기 위한 두 가지 기법이 있다. Residual Connection은 입력과 출력을 더해주는 방식으로 gradient vanishing 문제를 해결한다. Layer Normalization은 시퀀스 길이와 상관없이 정규화를 수행해서 모델이 더 빨리 수렴하도록 돕는다.

GPT

GPT는 라벨이 붙은 데이터가 부족하다는 문제를 해결하기 위해 **Semi-supervised Learning** 방식을 사용한다.

1. **Unsupervised Pre-training:** 이때 모델은 대규모 비라벨 데이터에 대해 다음 단어를 예측하는 언어 모델링을 수행한다. 예를 들어 문장이 "The weather is nice today."일 때, k=3의 윈도우 크기를 사용한다면 $P(\text{nice} \mid \text{the}, \text{weather}, \text{is})$와 $P(\text{today} \mid \text{weather}, \text{is}, \text{nice})$와 같은 확률을 최대화하도록 학습한다. 이렇게 함으로써 모델은 문맥적 패턴과 언어적 규칙을 학습하게 된다.
2. **Supervised Fine-tuning:** 여기서는 라벨이 있는 데이터셋 C를 사용한다. 입력 시퀀스 x1..xm이 주어지고, 그에 대응하는 정답 레이블 y를 바탕으로 지도 학습을 수행한다. 이 과정을 통해 모델은 사전 학습된 언어 능력을 바탕으로 특정 태스크(예: 분류, 질의응답 등)에 맞도록 조정된다.

즉, GPT는 **대규모 비라벨 데이터로 언어적 지식을 먼저 쌓은 후, 소규모 라벨 데이터로 세부적인 과제에 적합하게 다듬는 방식을 사용한다.**

BERT

BERT는 크게 unsupervised pre-training과 **supervised fine-tuning** 두 단계를 거쳐 학습된다.

1. Masked Language Model: 입력 문장의 15% 토큰을 무작위로 선택해 마스킹을 수행하는 방식이다. 이 중 80%는 [Mask]로 바꾸고, 10%는 무작위 단어로 교체하며, 나머지 10%는 원래 단어를 그대로 둔다. 모델은 이런 변형된 문장에서 가려진 단어를 맞히도록 학습되며, 이는 양방향 문맥 정보를 동시에 활용할 수 있다는 장점이 있다. 예를 들어 My dog is hairy라는 문장에서 *My dog is [MASK]*, *My dog is apple*, My dog is hairy와 같이 변형된 입력을 통해 학습이 진행된다.

2. Next Sentence Prediction(NSP): 두 문장이 주어졌을 때, 두 번째 문장이 실제로 원래 이어지는 문장인지 여부를 예측한다. 이를 위해 문장 구분 토큰을 사용하며, 이 학습은 이후 질의응답이나 자연어 추론과 같은 태스크에서 중요한 역할을 한다. 예를 들어 “BERT는 트랜스포머 인코더를 포함한다”라는 문장이 주어졌을 때, 올바른 후속 문장인지 여부를 예측하도록 학습한다.

이후 **Supervised Fine-tuning** 단계에서는 특정 태스크의 라벨이 달린 데이터셋을 사용해 BERT 전체를 end-to-end 방식으로 미세조정한다. 이 과정을 통해 사전학습으로 쌓은 언어적 이해를 실제 과제에 맞게 적용할 수 있게 된다.

BART

BART는 BERT와 GPT의 장점을 결합한 구조로, 인코더-디코더 기반의 트랜스포머 모델이다. **Denoising Autoencoder** 방식으로, 원래의 문장에 다양한 형태의 노이즈를 추가한 뒤 이를 복원하는 과정을 학습한다는 점이다. 즉, 입력에 손상을 가하고 이를 복원해내면서 문장 구조와 의미를 깊이 이해하도록 설계되어 있다.

BERT는 주어진 문장에서 일부 토큰을 가려놓고 맞추는 **Masked Language Model** 방식을 사용하고, GPT는 다음 단어를 예측하는 **Autoregressive** 방식을 사용한다. 반면 BART는 **Bidirectional Encoder**와 **Autoregressive Decoder**를 함께 사용하여 두 방식의 장점을 모두 활용한다. 인코더는 문장의 의미를 양방향으로 파악하고, 디코더는 문장을 순차적으로 생성하는 역할을 한다.

노이즈 기법으로는 **토큰 마스킹, 토큰 삭제, 문장 순열, 문서 회전, 텍스트 채우기** 등이 있으며, 이러한 변형된 입력을 다시 원래의 문장으로 복원하도록 학습한다. 이 과정에서 모델은 문장의 구조적 의미를 보존하면서도 다양한 변형에 강건한 표현을 학습할 수 있다.

최종적으로 인코더는 입력 전체의 의미를 가장 잘 담아내는 벡터를 만들어내고, 디코더는 이를 기반으로 원래의 문장이나 목표 출력 문장을 생성한다. 따라서 BART는 기계 번역, 요약, 질의응답 등 여러 자연어 생성·이해 작업에서 강력한 성능을 발휘한다.