Chapter 17 BERT를 중심으로

바라본 NLP와 LLM의 이모저모

2024 초급 LLM with AI application

이동준

BERT의 개념

BERT의 정의

- **Bidirectional Encoder** Representations from **Transformers**
- **BERT**는 Google에서 2018년에 발표한 (한때는 유망했던) **자연어 처리(NLP)** 모델
- **Transformer 아키텍처**를 기반으로 한 **pre-trained 언어 모델**
- GPT와 달리 BERT는 텍스트의 **양방향(Bidirectional)** 문맥을 학습
- 인코더 중심(입력 이해 중심)의 모델
- (개인적으로) 인사이트의 중요성을 설명하는 모델

NLP 및 LLM 발전 과정: Word2Vec 부터 MAMBA까지

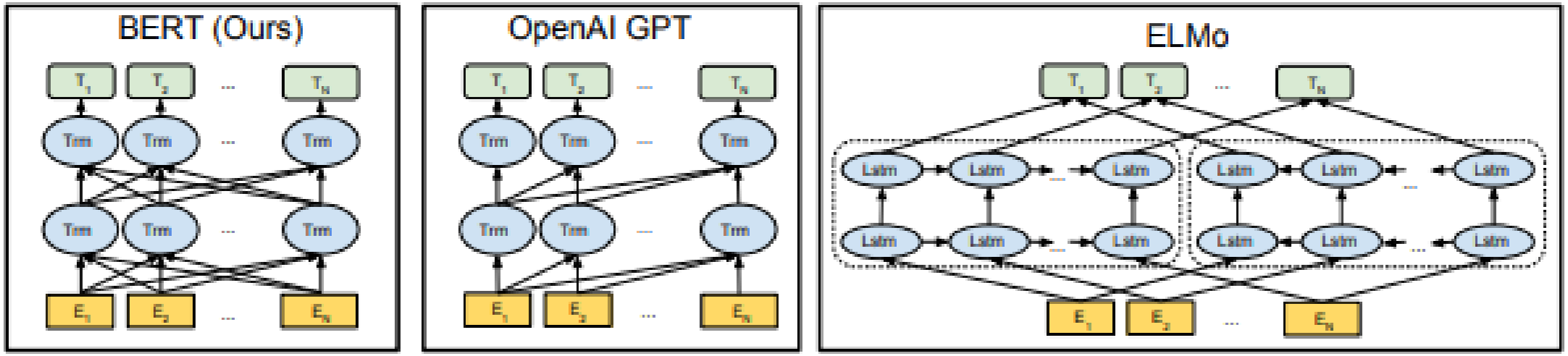
모델	핵심 발전	평가
CNN, RNN	초기 딥러닝 모델(CNN은 이미지, RNN은 텍스트)	RNN의 장기 기억 손실 문제
Word2Vec	단어를 고정된 벡터로 표현(정적 임베딩)	문맥 정보 부족
ELMo(2017)	LSTM 기반의 양방향 동적 임베딩 적용(BiLSTM).	병렬 처리의 어려움
GPT(2017)	Transformer 디코더 기반, 텍스트 생성 최적화.	바야흐로 Chat GPT의 시대
BERT(2018)	- Transformer 인코더 기반, 양방향 문맥 학습 - 사전 훈련 태스크 도입.	Transformer의 시대
MAMBA(2023)	- SSM, 선택 메커니즘 기반 아키텍처 - 최적의 시간 복잡도	?

BERT의 주요 특징

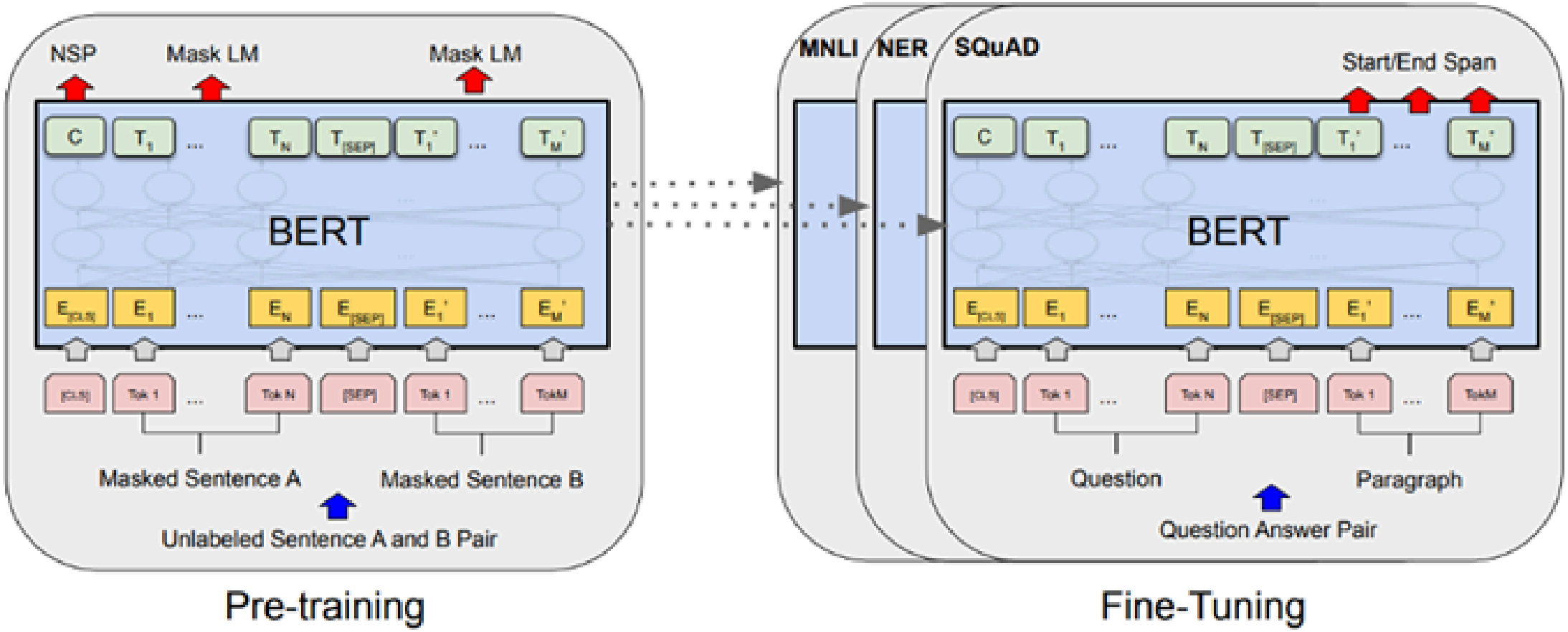
- BERT는 **Transformer의 인코더**를 사용하여 병렬 처리 가능
- Self-Attention 메커니즘으로 **양방향 문맥(Bidirectional Context)** 을 동시에 학습
- **변화점**:
  - i. **완전한 양방향 학습**:
    - ELMo는 순방향/역방향을 따로 학습하지만, BERT는 Self-Attention으로 한 번에 처리
  - ii. **사전 훈련 태스크 도입**:
    - Masked Language Model(MLM): 일부 단어를 마스킹하고 이를 예측
    - Next Sentence Prediction(NSP): 두 문장의 연결 관계를 학습
  - iii. **병렬 처리**:
    - Transformer 구조 덕분에 대규모 데이터 학습 속도 향상
- **한계**:
  - 굉장히 큰 모델 크기, 상대적으로 느린 추론 속도

ELMo, GPT, BERT

- BERT는 양방향성
- GPT는 단방향성
- ELMo는 독립적인 반대 단방향성



BERT의 학습 과정 개념도



BERT의 사전 학습 태스크: MLM과 NSP

1. Masked Language Model (MLM)

1.1 정의

- 문장에서 일부 단어를 **[MASK] 토큰**으로 대체한 다음 모델이 해당 단어를 예측하도록 학습
- 이를 통해 문장의 **양방향 문맥(Bidirectional Context)** 을 활용 가능

1.2 특징

- 양방향 문맥 활용: 이전 및 이후 단어를 모두 고려하여 학습
- 동적 임베딩 학습: 동일한 단어라도 문맥에 따라 다른 의미를 학습

1.3 예시

- 입력 문장: "The [MASK] sat on the mat."
- 모델 예측: "cat"

BERT의 사전 학습 태스크: MLM과 NSP

1.4 학습 과정

1. **입력 처리**:
   - 문장에서 단어의 15%를 무작위로 선택해 [MASK]로 대체
     - 예: "The cat sat on the mat" → "The cat [MASK] on the mat"
2. **예측 목표**:
   - [MASK] 토큰에 해당하는 단어를 예측
     - 출력: "sat"
3. **마스킹 규칙**:
   - 80%: [MASK]로 대체
   - 10%: 다른 랜덤 단어로 대체
   - 10%: 원래 단어를 그대로 유지
   - 이러한 규칙은 모델이 **단순히 [MASK] 토큰을 학습하는 것**을 방지
     (이런 건 애드혹이 아닐까요)

BERT의 사전 학습 태스크: MLM과 NSP

2. Next Sentence Prediction (NSP)

2.1 정의

- 두 문장이 주어졌을 때 **두 번째 문장이 첫 번째 문장의 다음에 오는 문장인지 여부**를 예측
- 문장 간 관계를 학습

2.2 특징

- 문장 간 관계 학습: NSP는 두 개 문장의 관계를 이해하는 데 집중

2.3 예시

- 문장 A: "The cat is on the mat."
- 문장 B: "It is watching outside."
  - 결과: `isNext`
- 문장 A: "The cat is on the mat."
- 문장 B: "The weather is nice today."
  - 결과: `notNext`

BERT의 사전 학습 태스크: MLM과 NSP

2.4 학습 과정

1. **입력 처리**:
   - 입력 문장 쌍을 생성
     - 50%: 실제로 연결된 문장 쌍
     - 50%: 무작위로 선택된 문장 쌍
2. **예측 목표**:
   - 첫 번째 문장(A)와 두 번째 문장(B)이 연결된 문장인지 아닌지를 예측
   - 출력 라벨:
     - `isNext` : 두 문장이 논리적으로 연결
     - `notNext` : 두 문장이 아무 연관 없음
3. **입력 형식**:
   - `[CLS]` 토큰: 첫 번째 문장
   - `[SEP]` 토큰: 문장 A와 문장 B를 구분

BERT의 파인 튜닝 태스크: MNLI, NER, SQuAD

1. MNLI (Multi-Genre Natural Language Inference)

- 두 번째 문장이 첫 번째 문장과 어떤 관계인지 분류
- 라벨은 **Entailment(포함)**, **Neutral(중립)**, **Contradiction(모순)**

2. NER (Named Entity Recognition)

- 텍스트에서 특정 개체(예: 사람, 장소, 날짜, 조직 등)를 식별하고 분류하는 태스크
- 텍스트의 각 토큰을 태깅화

3. SQuAD (Stanford Question Answering Dataset)

- 문서 내에서 사용자의 질문에 대한 답변을 추출하는 **질의응답(Q&A)** 태스크
- **예시**:
  - 입력 문서: "BERT는 2018년에 Google에서 발표된 모델이다."
  - 질문: "BERT는 언제 발표되었는가?"
  - 출력: "2018년"

Chapter 16 MAMBA. 새로운 아키텍처

2024 초급 LLM with AI application

이동준

MAMBA의 개념

기존 Transformer와 경쟁할지도 모르는(?) MAMBA의 정의

- Memory-Augmented Multi-modal Attention
- Selective State Space Model
- **State Space Model (SSM)** 과 **선택 메커니즘**을 결합한 모델
- 기존 모델의 **계산 효율성**, **메모리 사용량**, **확장성**을 크게 개선한 모델
- 기존 Transformer 모델이 가지는 계산 복잡성 문제를 해결하면서 RNN, CNN 방식을 응용한 모델
- 멀티모달에도 강한 능력을 갖추도록 설계된 모델

MAMBA의 특징

1) State Space Model (SSM)

- **장기 의존성(Long-term Dependency)** 학습에 특화된 모델.
- RNN의 순차적 계산 방식을 병렬화 가능하도록 개선.
- 긴 시퀀스에서의 정보 유지를 매우 효율적으로 처리.

2) 선택 메커니즘

- 중요도가 높은 데이터에만 주의를 집중하는 **Selective Attention** 메커니즘.
- **계산 복잡도**를 기존 Transformer의 $O(n^2)$에서 $O(n \log n)$으로 감소.
- 긴 시퀀스 처리 시, 불필요한 연산을 줄이고 메모리 사용량 절감.

MAMBA의 SSM (State Space Model) - A 행렬

- SSM (State Space Model) 은 긴 시퀀스를 효율적으로 처리하기 위해 RNN을 약간 응용한 모델
- **상태 전이(State Transition)** 와 **출력 생성(Output Generation)** 을 통해 시퀀스 데이터를 모델링

상태 전이(학습)	**출력 생성(추론)**
$h_t = A \cdot h_{t-1} + B \cdot x_t$	$y_t = C \cdot h_t + D \cdot x_t$
- h_t: 현재 상태.	- y_t: 현재 출력.
- A: **상태 전이 행렬.**	- C: 출력 변환 행렬.
- B: 입력 변환 행렬.	- D: 값 변환.
- h_{t-1}: 직전 상태.	- x_t: 현재 입력.

- RNN처럼 순차적으로 학습하지만 훨씬 성능이 좋으며, 효율적이면서도 장기 의존성 문제를 해결 가능

MAMBA의 선택 메커니즘 - B 행렬, C 행렬

- 전체 데이터를 처리하지 않고, 중요한 정보만 반영하여 긴 시퀀스를 **선택적으로 처리**하는 메커니즘
- 표는 같지만, B 행렬과 C 행렬에서 중요한 정보만 추출(랜덤, 일정 거리, 적절 선택 등)

상태 전이(학습)	출력 생성(추론)
$h_t = A \cdot h_{t-1} + B \cdot x_t$	$y_t = C \cdot h_t + D \cdot x_t$
- h_t: 현재 상태.	- y_t: 현재 출력.
- A: 상태 전이 행렬.	- **C: 출력 변환 행렬.**
- **B: 입력 변환 행렬.**	- D: 값 변환.
- h_{t-1}: 직전 상태.	- x_t: 현재 입력.

- Transformer에서 캐시를 이용해서 연산을 최적화했다면, MAMBA에서는 **선택 메커니즘**으로 최적화

Transformer와 MAMBA의 차이

구분	Transformer	MAMBA
주요 특징	Self-Attention 메커니즘으로 문맥 정보 학습	선택 메커니즘으로 효율적 문맥 정보 학습
계산 복잡도	입력 길이에 따라 계산량 급증	SSM 사용으로 계산복잡도 최적화
학습 효율성	대규모 데이터 학습 가능하지만 속도가 느림	**더 빠른 학습 및 추론 속도**
한계	계산 비용이 높아 긴 시퀀스 처리에 비효율적	특정 태스크에 최적화, 범용성은 다소 제한적

결론 - 나는 뭘 해먹고 살아야 하나(?)

바야흐로 AI의 시대

- Transformer 또한 오래지 않아 (MAMBA를 비롯한) 경쟁자에 의해 그 지위를 잃을 것으로 예상
- AI 영역은 지금처럼 서로 뒤집고 뒤집히는 관계가 지속할 것으로 예상
- **끝나지 않는 사람의 꿈**이 AI를 본질적으로 변화시키는 역동성
- 21세기 초반의 닷컴 버블와 유사해보이지만, AI 혁명은 닷컴 버블과 달리 **강력한 실체가 존재**
- AI에 탑승을 하는 것이 좋지 않을까....

들어주셔서 감사합니다.