손에 잡히는 언어모델 구현

언어모델 개발 튜토리얼

2023. 8. 23

류연희, 허기홍

KAIST Programming Systems Laboratory

배경 & 목표

- 프로그래밍**언어** 연구회
- 프로그래밍언어로 글을 읽고 쓰고 이해하는 두 가지 방법
  - 규칙 기반 (전통적): 파싱, 컴파일, 분석 등
  - 확률 기반 (장안의 화제): 언어 모델
- 목표:
  - 두 가지 방법을 적재적소에 활용하여 PL 문제 해결
  - 두 가지 방법의 장점을 결합한 새로운 프로그래밍 시스템 실현
- 진행: 류연희 (KAIST 박사과정, 안전한 프로그래밍 언어 모델 연구)
- 도움: 김태은, 김재호, 박종찬, 장수진 (KAIST)

목차

- 자료: https://github.com/prosyslab/sigpl23-tutorial

시간	내용
13:30 – 14:45	• 코드 언어 모델 소개 • Colab 을 이용한 실습 환경 설정 • Transformers 라이브러리 사용해보기 • 오픈소스 거대 언어 모델의 생성 기능 이용해보기
14:45 – 15:00	• 휴식
15:00 – 16:15	• Transformers 라이브러리를 이용한 코드 수정 Fine-tuning 실습

코드 언어모델

- 언어모델 기술을 이용하여 프로그램 소스 코드를 학습한 모델
- 전통적인 언어모델: 주어진 문자열 다음에 올 문자열을 예측하는 방법
  - 예시: N-gram 모델, GPT 모델
  - 예시: "가는 말이 고와야 오는 말도 □"에서 □는 무엇인가?
- 보다 넓은 의미: 문자열의 순서에 기반해서 자연어 문장의 의미를 이해하는 방법
  - 예시: BERT 모델, T5 모델
  - 예시: "가는 말이 고와야 □ 말도 곱다." 에서 □는 무엇인가?
- 최근의 좁은 의미: 트랜스포머 구조를 사용하여 학습된 언어모델

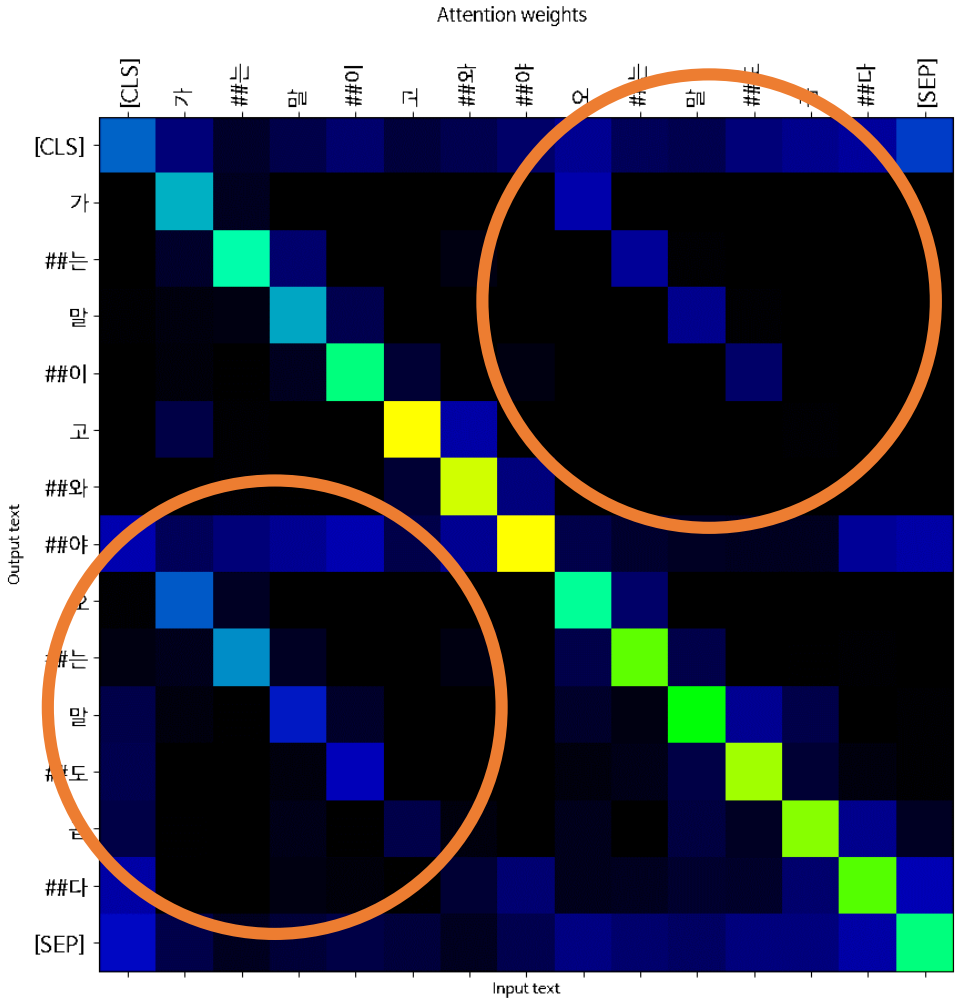
트랜스포머 구조

- 심층 신경망(DNN) 구조의 일종
  - CNN과 같이 입력의 길이가 고정
  - RNN과 같이 순차적 데이터의 의미를 이해
- Attention 구조를 여러층 적재하여 데이터를 심층적으로 이해
  - "Attention is all you need" (2017, Google)
- 인코더-디코더 구조로 구성
  - 인코더: 데이터 이해
  - 디코더: 데이터 생성

Self-Attention 예시

- Multilingual BERT, 마지막 레이어의 Attention

"가는 말이 고" 와 "오는 말도 곱" 사이의 Attention 점수가 높다

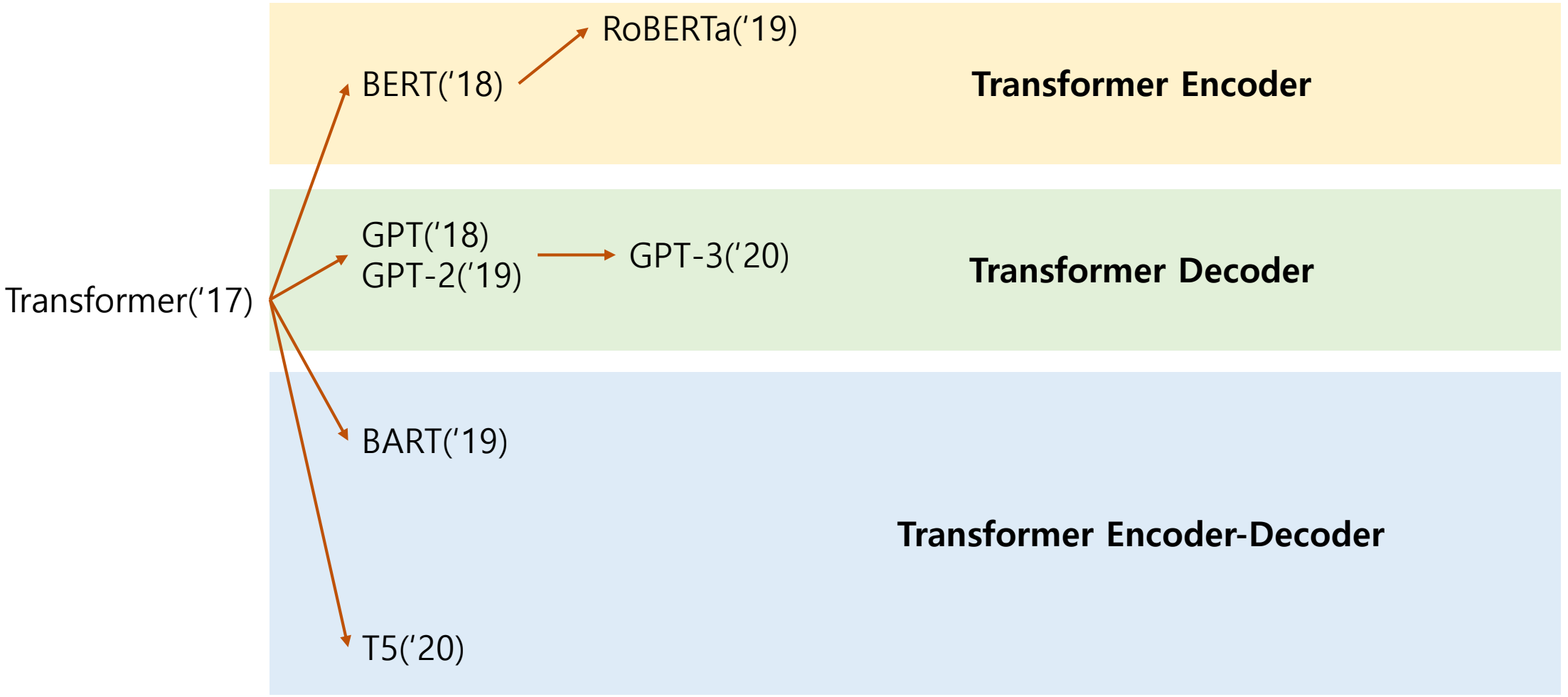


※ 주의: Attention 점수에 "사람이 이해할 수 있는 설명 능력이 있는가?"는 아직 연구와 논쟁 진행중

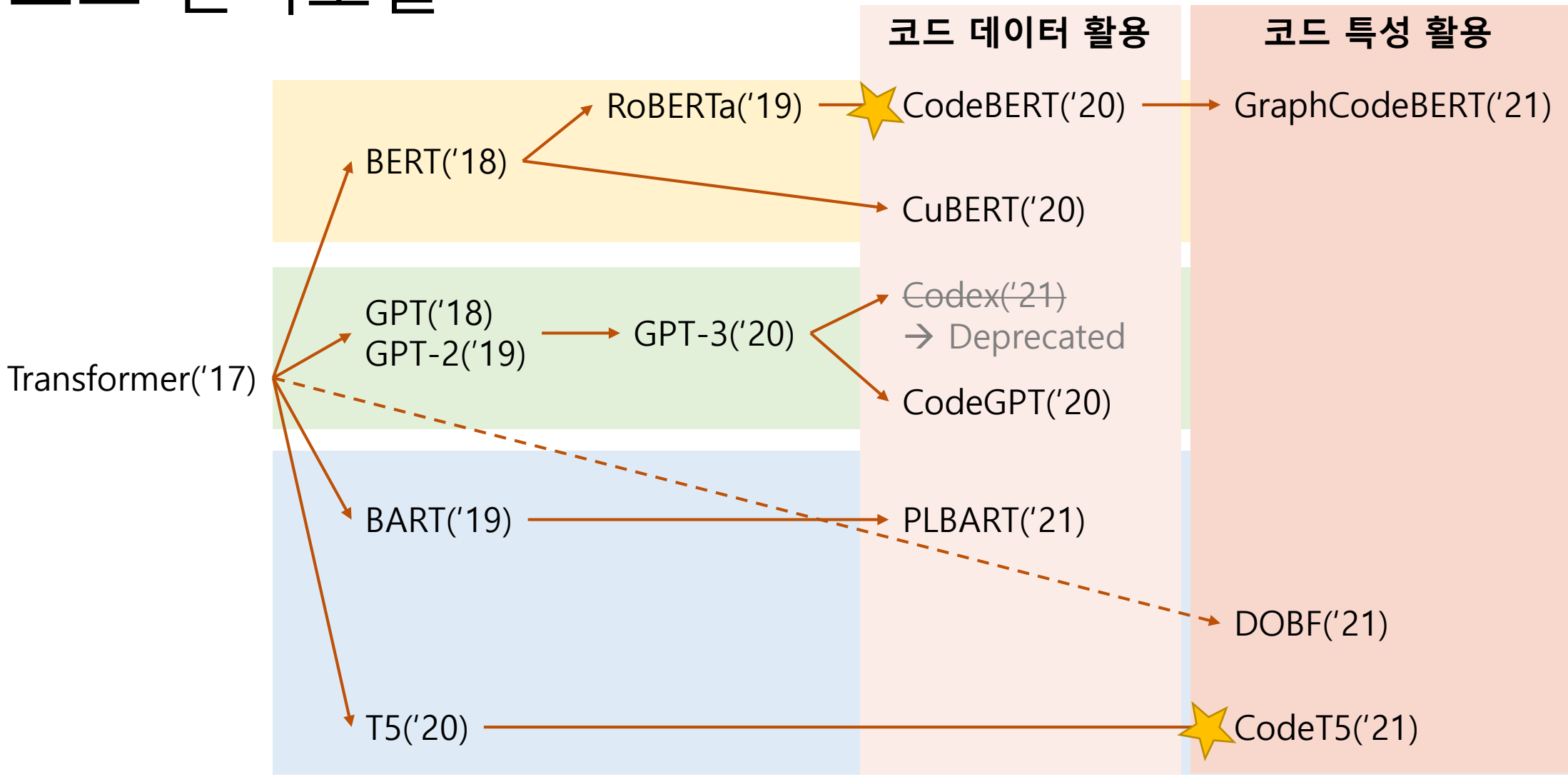
실습 1. Attention 계산해보기

모델 및 데이터 일람

자연어 언어모델


RoBERTa('19)
BERT('18)
Transformer Encoder
GPT('18)
GPT-2('19)
GPT-3('20)
Transformer('17)
Transformer Decoder
BART('19)
Transformer Encoder-Decoder
T5('20)

코드 언어모델


코드 데이터 활용
코드 특성 활용
Transformer('17)
BERT('18)
RoBERTa('19)
CodeBERT('20)
GraphCodeBERT('21)
CuBERT('20)
GPT('18)
GPT-2('19)
GPT-3('20)
~~Codex('21)~~
→ Deprecated
CodeGPT('20)
BART('19)
PLBART('21)
DOBF('21)
T5('20)
CodeT5('21)

거대 언어모델

- GPT-3(2020) 이후 거대 언어 모델 (Large Language Model) 널리 사용

그룹	모델 이름	규모	구조	공개 여부	발표 시기	분류
Facebook	InCoder	1B, 6B	Decoder-only	공개	2022.04.	코드 특화
Salesforce	CodeRL	770M+	CodeT5 + 강화학습	공개	2022.07.	코드 특화
BigScience	BLOOM	175B+	Decoder-only	공개	2022.11.	범용
Facebook	LLaMa	7B~65B	Transformer	제한적 공개	2023.02.	범용
OpenAI	GPT-4	unknown	GPT3	유료 API	2023.03.	범용
Salesforce	CodeT5+	110M~770M	T5	공개	2023.05.	코드 특화
Facebook	LLaMa2	7B~70B	Transformer	공개	2023.07	범용

대화형 언어모델

- 채팅 형식을 강화학습으로 학습
  - (장점): 강화학습으로 "사람이 더 좋아할만한 출력" 학습
  - (단점): Prefix 제한 불가능

그룹	모델 이름	규모	구조	공개 여부	발표 시기	분류
OpenAI	InstructGPT	Unknown	GPT3	유료 API	2022.01.	범용
Salesforce	CodeGen	350M, 2.7B, 16B	Decoder-only	공개	2022.03.	코드 특화
OpenAI	ChatGPT	Unknown	GPT3	유료 API	2022.12.	범용
Salesforce	InstructCodeT5+	16B	CodeT5	공개	2023.05.	코드 특화
Salesforce	CodeGen2	1B, 7B, 16B	Decoder-only	공개	2023.05.	코드 특화
Facebook	LLaMa2-Chat	7B~70B	Transformer	공개	2023.07.	범용

코드 데이터셋 – 코드 분류, 요약

그룹	데이터 이름	종류	입력	출력
Microsoft	CodeXGLUE	Clone detection	코드 쌍	이진분류
		Defect detection	코드	이진분류
		Type prediction	코드	타입 분류
		Code summarization	코드	자연어 요약
		Code search	자연어, 코드 쌍	이진 분류
		Text-to-code generation	자연어 요약	코드
Google	CuBERT	Defect detection	코드	이진분류

코드 데이터셋 – 코드 생성

그룹	데이터 이름	종류	입력	출력
OpenAI	HumanEval	(original)	자연어/코드 프롬프트	자동 완성 코드
		Infill	자연어/코드 프롬프트	자동 완성+수정 코드
Google	MBPP		자연어 프롬프트	자동 생성 코드
Microsoft	CodeXGLUE	Cloze test	코드	토큰
		Code completion	코드 프롬프트	자동 완성 코드
		Code repair	코드	자동 수정 코드
		Code translation	자바 코드	파이썬 코드
		Text-to-code generation	자연어 요약	코드

모델 사용할 때 필요한 GPU 메모리

- GPU 메모리 최소 사용량:
  - 파라미터 개수 × 파라미터 크기 = 모델 최소 크기
  - 예시: Facebook Incoder 1B 모델 사용하려면 최소 5 GB (= 1.3B * 4 byte) 메모리 필요
- Inference 할 때:
  - 입력 변수의 크기와 중간 상태 변수만큼 메모리 추가 사용
  - 경험적으로 모델 크기의 20% 내외
- 학습할 때:
  - 역전파(back propagation)를 위해 중간 상태 변수를 모두 유지하기때문에 메모리 많이 사용
  - 데이터 배치 크기가 작을수록 메모리 덜 사용
  - Fine-tuning 학습의 경우 고정된 레이어가 많을수록 메모리 덜 사용

실습 2. Fine-tuning 학습해보기

API 에서 독립하기

- 언어모델을 보조적인 확률 도구로 사용한다면 작은 모델 파인 튜닝으로도 충분
- 공개된 거대 언어모델 이용해도 GPT-3 수준의 생성 가능
- 비결정적인 동작 제어 가능
- 서버 비용 vs API 이용료

- 참고할 만한 자료
  - Natural Language Processing with Transformers book: homepage
  - Hugging Face 에서 제공하는 Transformers 강좌: homepage, wikidocs.net (우리말 번역)