CH01. LLM 지도

2024-12-14

태영

0. 개요

- LLM(Large Language Model)의 작동 원리:
  - 다음 단어를 예측해 문장을 생성하는 단순하지만 강력한 기초 원리를 통해 작동.
  - ChatGPT와 같은 모델이 이러한 기초를 기반으로 발전.
- LLM 연구의 필요성:
  - ChatGPT 등장 이후 빠르게 확산된 LLM 서비스와 연구 흐름 이해.
  - LLM의 발전 단계와 핵심 메커니즘에 대한 통찰 제공.
- 1장에서 제공하는 내용:
  - 기술적 언어 모델의 기본 구조 및 학습 방식 소개.
  - 세부 사항보다는 전체적인 흐름과 맥락을 파악할 수 있도록 구성.
- 향후 다룰 주제:
  - LLM이 우리 삶과 기술적 환경에 미치는 영향.
  - 다양한 애플리케이션에서 LLM의 활용 사례와 방향성 탐구.

1.1 딥러닝과 언어모델링

- LLM의 기술적 기반:
  - 딥러닝(Deep Learning)은 데이터의 패턴을 학습하는 신경망(neural network) 기술로, 기계 학습(machine learning)의 한 분야.
  - 정형 데이터뿐 아니라 비정형 데이터(unstructured data)에서도 뛰어난 패턴 인식 능력을 발휘.
- LLM의 연구 분야:
  - 자연어 처리(Natural Language Processing): 사람이 이해하고 생성할 수 있는 언어 연구.
  - 자연어 생성(Natural Language Generation): 사람처럼 텍스트를 생성하는 기술.
- 언어 모델의 정의:
  - 다음에 올 단어를 예측하며 문장을 생성.
  - LLM은 딥러닝 기반의 언어 모델.
- 역사적으로 중요한 사건:
  - 2013년: 구글의 Word2Vec 발표 – 단어를 숫자로 표현.
  - 2017년: 트랜스포머 아키텍처(Transformer Architecture) 발표 – 기계 번역 성능 향상.
  - 2018년: OpenAI의 GPT-1 모델 공개 – 트랜스포머 기반 언어 모델 활용.

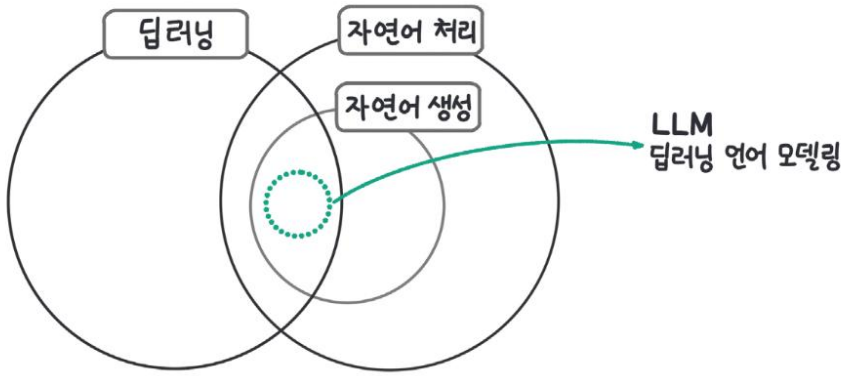

그림 1.1 LLM이 딥러닝 및 자연어 처리에서 차지하는 위치

그림 1.2 딥러닝과 언어 모델링 관점에서 주요 사건

1.1.1 데이터의 특징을 스스로 추출하는 딥러닝

- 딥러닝의 주목:
  - 2012년 이미지넷 대회에서 **알렉스넷(AlexNet)**의 성공으로 딥러닝의 가능성 주목.
  - 기존 모델 대비 오류율을 혁신적으로 낮춤(26% → 16%).
- 딥러닝의 특징:
  - 복잡한 문제를 간단하면서도 범용적인 방법으로 해결.
  - 기존 머신러닝과 비교해 "데이터의 특징(feature)을 자동으로 학습"하는 점이 가장 큰 차별점.
- 딥러닝 문제 해결 과정:
  - 문제 유형에 따라 모델 선택.
  - 문제에 대한 학습 데이터를 준비.
  - 데이터를 반복적으로 모델에 입력해 학습 진행.
- 기존 접근법과 차이:
  - 과거에는 사람이 직접 데이터를 설계하고 특징을 선택.
  - 딥러닝은 데이터를 스스로 분석하고 학습하며, 더욱 높은 정확도와 효율성을 보임.

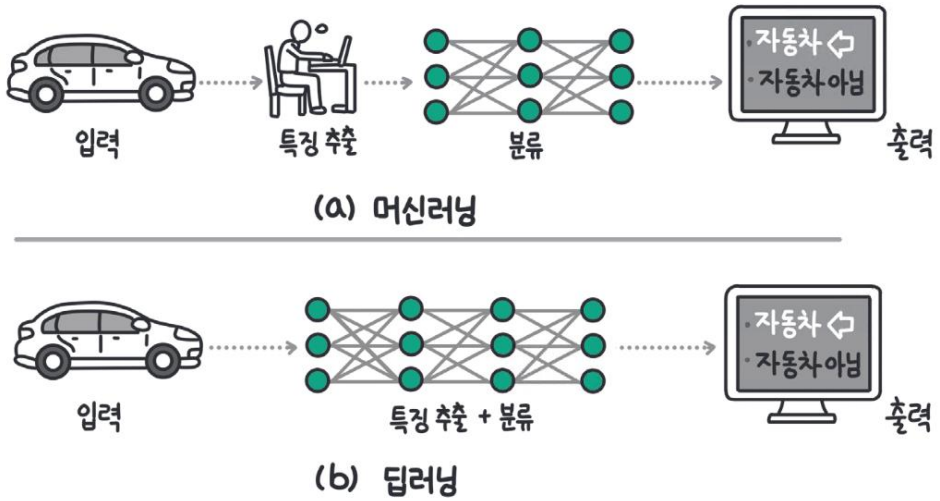

그림 1.3 머신러닝과 딥러닝의 차이
(출처: https://www.softwaretestinghelp.com/data-mining-vs-machine-learning-vs-ai/)

1.1.2 임베딩: 딥러닝 모델이 데이터를 표현하는 방식

- 임베딩이란?
  - 데이터를 숫자로 표현하여 의미와 특징을 숫자의 집합으로 담는 방법.
  - 딥러닝 모델이 데이터를 학습하기 위한 핵심 개념 중 하나.
- MBTI를 활용한 임베딩 이해:
  - 사람의 성향을 4개의 숫자로 단순화하여 표현(MBTI 검사 결과 예시).
  - 복잡한 데이터(예: 텍스트, 이미지, 비디오)를 벡터로 변환.
- 임베딩의 활용:
  - 검색 및 추천: 검색어와 관련된 상품 추천.
  - 클러스터링: 유사한 데이터끼리 묶기.
  - 이상치 탐지: 비정상적 데이터 탐지.
- 대표적인 임베딩 모델: Word2Vec
  - 2013년 구글에서 발표한 모델로 단어를 벡터로 변환.
  - 단어의 의미를 숫자의 집합으로 표현하며, 자연어 처리(NLP)에서 활용.
- 임베딩의 특징:
  - 숫자 하나로는 의미를 명확히 알기 어렵지만, 벡터 전체가 단어의 의미를 압축적으로 담음.
  - 데이터 간 거리를 계산해 관련성과 유사성을 평가.

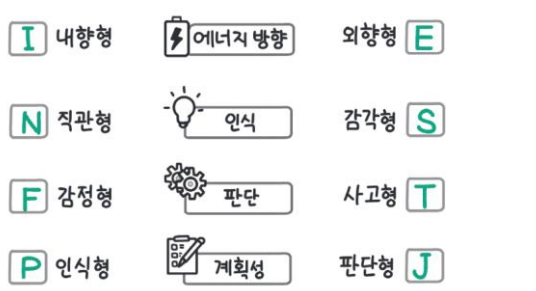

그림 1.4 MBTI의 네 가지 범주(출처: https://www.testmbti.net/mbti-검사-16문항)

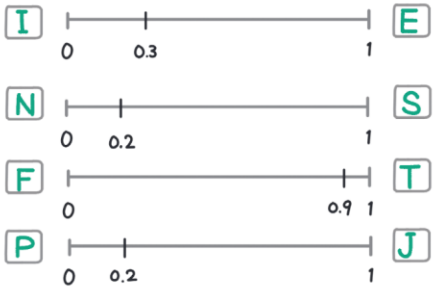

그림 1.5 MBTI 검사 결과 예시

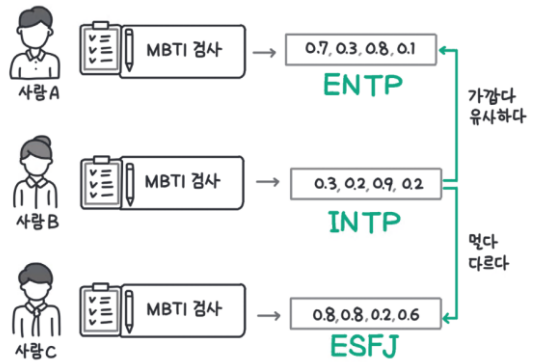

그림 1.6 데이터를 숫자로 표현할 경우의 장점

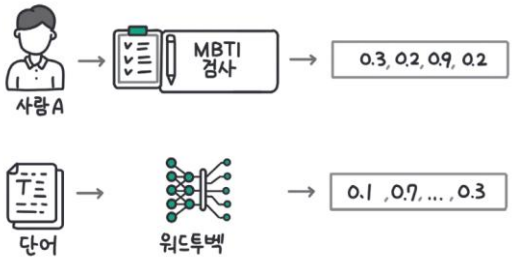

그림 1.7 워드투벡을 활용해 단어를 임베딩으로 표현하기

표 1.1 MBTI 검사의 숫자 표현과 임베딩 모델 숫자 표현의 공통점과 차이점

	MBTI	임베딩 모델(예: 워드투벡)
공통점	• 데이터나 개체를 숫자 집합으로 표현한다. • 숫자로 표현하기 때문에 거리나 유사도 같은 정보를 활용할 수 있다.	
차이점	• 사람을 4개의 숫자로 표현한다. • 사람이 MBTI 검사를 설계했기 때문에 각 숫자의 의미가 명확하다.	• 일반적으로 수십~수만 개의 숫자로 표현한다. • 사람이 직접 각 숫자의 의미를 정의하지 않으므로 숫자 하나하나의 의미를 파악하기 어렵다.

1.2.3 언어 모델링 : 딥러닝 모델의 언어 학습법 1/2

- 언어 모델링이란?
  - 텍스트의 다음 단어를 예측하며 문장을 생성하는 모델.
  - 대량 데이터로 언어의 특징을 학습하고 문제 해결에 응용.
- 전이 학습(Transfer Learning):
  - 사전 학습(Pre-training): 대규모 데이터로 모델의 기본 구조 학습.
  - 미세 조정(Fine-tuning): 특정 문제 해결을 위해 추가 학습.
- 특징과 이점:
  - 적은 데이터로도 높은 성능 발휘.
  - 다양한 과제에 유연하게 활용 가능.
- 응용 사례:
  - 이미지 분류(예: 새, 고양이, 개).
  - 유방암 데이터로 새로운 분류 작업 수행.

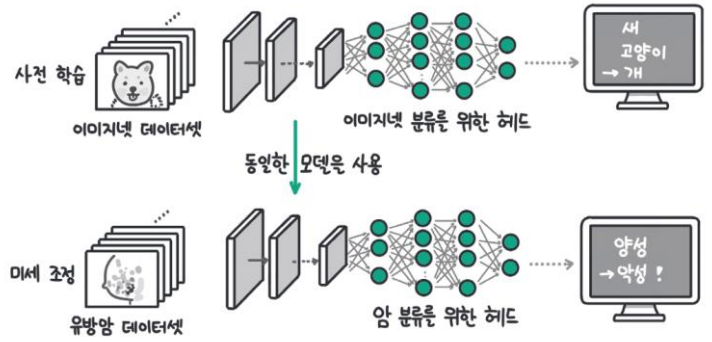

그림 1.8 이미지 인식 분야의 전이 학습
(출처: https://www.kaggle.com/datasets/nancyalaswad90/breast-cancer-dataset)

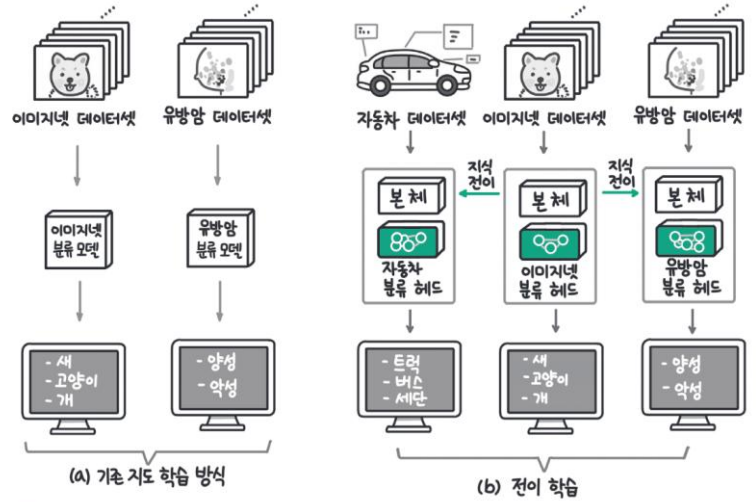

그림 1.9 기존의 지도 학습 방식과 전이 학습 방식의 차이
(출처: https://learning.oreilly.com/library/view/natural-language-processing/9781098136789/ch01.html)

1.2.3 언어 모델링 : 딥러닝 모델의 언어 학습법 2/2

- 자연어 처리 분야에서의 전이 학습의 발견:
  - 2018년 fast.ai의 연구로 사전 학습이 적은 데이터에서도 효과적임을 확인.
  - 텍스트 분류와 같은 다운스트림 작업에서 미세 조정(fine-tuning)을 통해 성능 향상.
    - 다운스트림 작업 이란, 사전 학습된 모델을 활용하여 특정 문제를 해결하는 응용 작업을 의미
  - 1.10(a) – 다음 단어 예측 하는 방식으로 사전 학습, (b) – 다운 스트림 과제의 데이터 셋으로 미세 조정, (c) – 텍스트 분류 미세 조정 (레이블 없음)
- 트랜스포머의 등장:
  - 2017년 구글의 논문 "Attention is All You Need"로 트랜스포머 구조 소개.
  - 순환신경망(RNN)의 한계를 극복하고 딥러닝 모델의 효율성을 크게 향상.
- 대표적인 트랜스포머 모델:
  - BERT: 양방향 Encoder 기반으로 자연어 이해 작업에 최적화.
  - GPT: 생성적 사전 학습 모델로 텍스트 생성에 강점.
- 트랜스포머의 의의:
  - 사전 학습을 통해 레이블 없는 데이터도 효과적으로 활용 가능.
  - 자연어 처리에서 가장 대표적인 사전 학습 방식으로 자리 잡음.

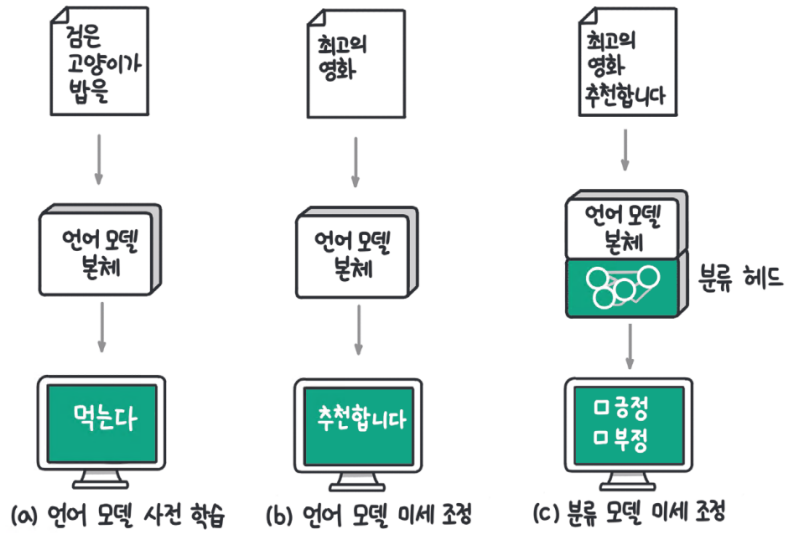

그림 1.10 언어 모델링을 통한 학습이 분류 성능에 미치는 영향
(출처: 「Universal Language Model Fine-Tuning」, https://arxiv.org/abs/1801.06146)

1.2 언어 모델이 챗GPT가 되기 까지

- 트랜스포머 아키텍처 공개(2017):
  - AI 분야에서 획기적인 기술로, 언어 모델의 발전 기반 마련.
- GPT 시리즈의 등장(2018~2020):
  - 트랜스포머를 활용한 모델로 언어 생성 능력 강화.
  - GPT 시리즈를 통해 대규모 언어 모델(LLM)로 발전.
- ChatGPT의 탄생(2022):
  - AI 기술의 정점으로 대화형 LLM의 새로운 기준 수립.
  - 트랜스포머 아키텍처와 정렬 기술(Alignment)의 결합으로 사용자 친화적 대화 기능 구현.

그림 1.11 언어 모델에서 챗GPT까지의 과정

1.2.1 RNN에서 트랜스포머 아키텍처로 1/2

- RNN의 특징:
  - 텍스트 데이터(시퀀스)를 순차적으로 처리하여 다음 단어를 예측.
  - 지금까지의 입력 데이터를 압축해 하나의 **잠재 상태 (hidden state)**로 유지.
  - 장점: 메모리를 적게 사용하며 빠르게 계산 가능.
  - 단점: 긴 입력 데이터에서는 의미 손실 및 성능 저하.
- 트랜스포머의 등장(2017):
  - RNN의 한계를 극복하기 위해 어텐션(attention) 메커니즘 도입.
  - 순차적 처리 없이 전체 맥락을 계산해 더 정확한 예측 가능.
- 차이점:
  - RNN: 이전 단어의 맥락만 활용해 예측.
  - 트랜스포머: 입력 데이터 전체를 고려해 예측.

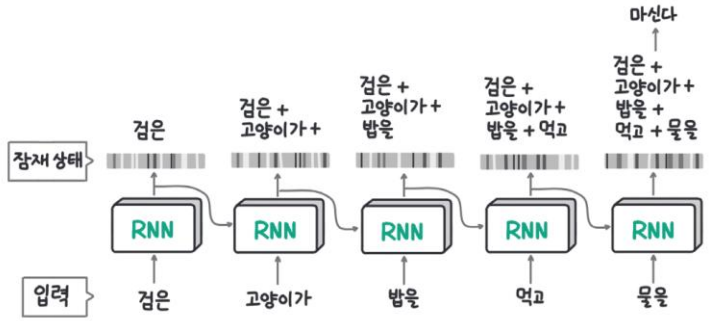

그림 1.12 입력을 순차적으로 처리하는 RNN 모델의 방식

그림 1.13 RNN에서의 맥락 압축

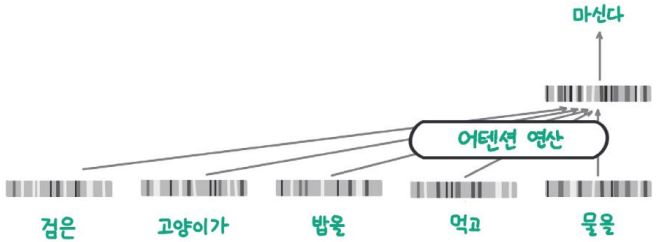

그림 1.14 트랜스포머 아키텍처의 어텐션 연산

1.2.1 RNN에서 트랜스포머 아키텍처로 2/2

- 트랜스포머의 어텐션 메커니즘:
  - 입력 텍스트 전체를 활용해 정확한 예측 가능.
  - RNN과 달리 순차적 처리 없이 병렬 연산을 통해 학습 속도 향상.
  - 긴 입력 데이터에서도 맥락 정보를 유지하여 성능 높임.
- 한계와 단점:
  - 어텐션 과정에서 많은 메모리와 연산량 필요.
  - 긴 입력 데이터는 예측 시간이 증가.
- 효율성과 성능 비교:
  - 트랜스포머: 높은 성능을 자랑하지만 메모리 효율성이 낮음.
  - RNN: 메모리 효율성이 뛰어나지만 성능이 제한적.
  - 새로운 아키텍처 연구가 진행 중으로, 효율성과 성능을 모두 만족할 가능성이 기대됨.

그림 1.15 연산 과정에서 많은 메모리를 사용하는 어텐션 연산

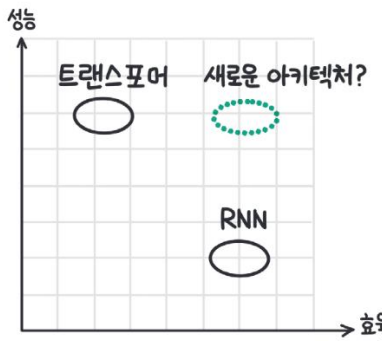

그림 1.16 메모리 효율성(RNN)과 성능(트랜스포머) 사이의 새로운 아키텍처

1.2.2 GPT 시리즈로 보는 모델 크기와 성능의 관계

- GPT 시리즈의 발전:
  - GPT-1 (2018): 1.7억 파라미터.
  - GPT-2 (2019): 15억 파라미터로 확장.
  - GPT-3 (2020): 1,750억 파라미터로 대규모 확장.
- 모델 크기와 성능:
  - 모델 크기와 학습 데이터셋의 확장은 성능 향상에 기여.
  - GPT-3는 인간의 언어 생성 능력과 유사하다는 평가를 받음.
- 언어 모델과 압축:
  - 학습 과정은 데이터의 언어 패턴을 압축하여 모델에 저장.
  - 메타의 라마(LLaMA) 모델 예: 10TB 데이터를 학습해 최종 140GB로 압축.
- 한계:
  - 학습 데이터가 커질수록 성능은 향상되지만, 모델 크기의 증가가 성능 향상을 보장하지는 않음.
  - 학습 데이터 크기가 최대 모델 크기의 상한.

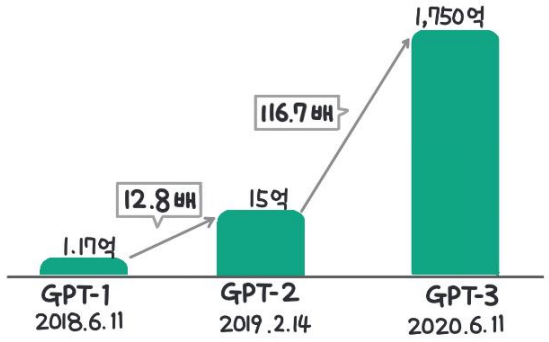

그림 1.17 GPT 버전에 따른 모델 크기 비교
(출처: NIA 「THE AI REPORT 2023-1 리포트」 'ChatGPT는 혁신의 도구가 될 수 있을까? ChatGPT 활용 사례 및 전망', 김태원 수석연구원(AI · 미래전략센터), https://kenss.or.kr/51/11908649)

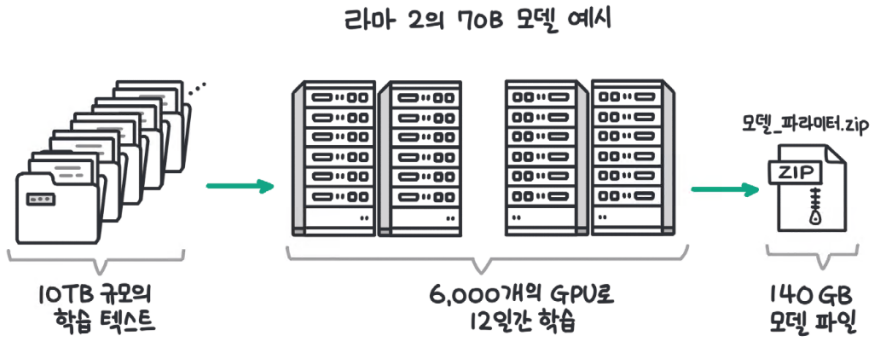

그림 1.18 텍스트 생성에서의 압축(출처: 안드레이 카르파티(Andrej Karpathy)의 'LLM 소개(Intro to Large Language Model)' 유튜브 동영상, https://www.youtube.com/watch?v=zjkBMFhNj_g&t=257s)

1.2.3 챗GPT의 등장

- GPT-3의 한계:
  - 1,750억 파라미터로 뛰어난 텍스트 생성 능력을 가졌으나, 사용자 요청을 제대로 이해하거나 응답하지 못함.
- 정렬(Alignment)의 중요성:
  - LLM이 사용자 요청에 맞는 응답을 생성하며, 사용자 가치를 반영하도록 설계.
  - 응답 품질 향상뿐만 아니라 사용자 이해와 편의를 위한 다양한 관점 고려.
- ChatGPT의 개선 사항:
  - 지도 미세 조정(Supervised Fine-Tuning):
    - 사람이 정리한 데이터셋(지시 데이터셋)을 기반으로 추가 학습.
    - 사용자 요청에 맞는 적절한 응답을 생성하도록 학습.
  - RLHF(Reinforcement Learning from Human Feedback):
    - 사용자 피드백을 활용해 모델을 강화 학습.
    - 선호 데이터셋을 기반으로 더 나은 응답 선택.
- 한계와 과제:
  - 사용자 요청에 맞춰 응답 하는 것이 항상 옳은 것은 아닐 수 있음.
  - 예: 폭탄이나 약물 제조 하는 정보를 제공하거나 사용자 오해를 초래할 가능성.

1.3 LLM 애플리케이션의 시대가 열리다

- LLM의 영향력:
  - ChatGPT의 충격 이후, 여러 조직이 LLM을 활용한 애플리케이션 개발에 주력.
  - LLM이 우리의 일상과 기술 환경에 미치는 영향이 점점 확대됨.
- 애플리케이션 개발의 핵심:
  - 효율적인 학습: LLM을 효과적으로 활용하기 위한 최적화 기법 필요.
  - 검색 증강 생성(RAG): 검색 결과를 활용해 LLM의 성능을 강화.
  - sLLM(small LLM): 경량화된 모델로 특정 환경에서 효율적인 성능 발휘.
- 향후 전망:
  - LLM 기반 애플리케이션이 다양한 산업과 분야에서 혁신을 주도.
  - 효율성과 활용도를 극대화하기 위한 기술 개발이 계속될 전망.

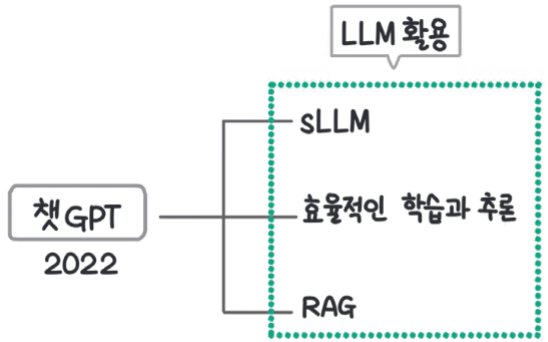

그림 1.19 LLM을 활용한 애플리케이션을 개발할 때 알아둬야 할 개념

1.3.1 지식 사용법을 획기적으로 바꾼 LLM

- LLM의 다재다능함:
  - 자연어 이해와 생성 두 측면에서 뛰어난 성능을 보임.
  - 다양한 작업에서 사용자 요청에 맞는 적응적 수행(multitasking) 가능.
- 다재다능함의 필요성:
  - 대부분의 작업이 언어 이해와 생성을 포함하여 복합적으로 얽혀 있음.
  - 예: 코드 작성, 문서 요약 및 보고서 작성, 의사결정 지원 등.
- LLM의 장점:
  - 기존 모델들이 여러 작업에 각각 필요한 복잡도를 단일 LLM으로 대체.
  - 복잡도를 줄이고 효율성을 높여 AI 활용성을 극대화.
- 영향과 기대:
  - 지식 습득, 요약, 활용 과정을 간소화하여 새로운 통찰을 빠르게 제공.
  - 인간이 해결하기 어려운 복잡한 작업에도 기대와 우려를 동시에 불러일으킴.

그림 1.21 개발자가 수행하는 업무 예시

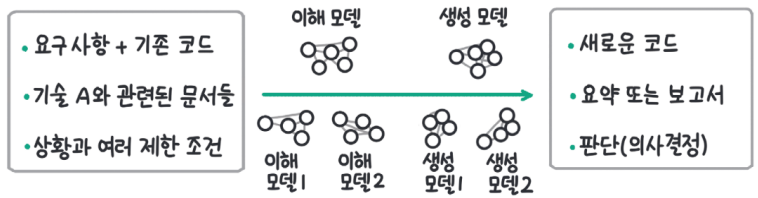

그림 1.22 기존 자연어 처리 모델을 활용한 작업

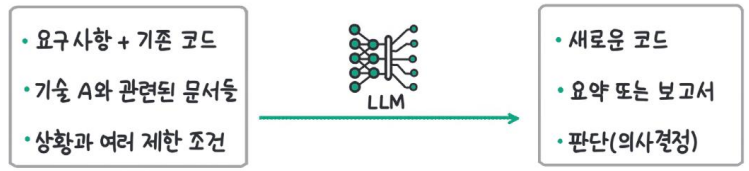

그림 1.23 LLM을 활용한 작업

1.3.2 sLLM: 더 작고 효율적인 모델 만들기

- LLM 활용 방식:
  - 상업용 API(예: OpenAI GPT-4) 활용.
  - 오픈소스 LLM을 직접 학습하여 특정 도메인에 맞춘 모델 생성.
- sLLM의 특징:
  - 도메인별 데이터로 추가 학습하여 모델 크기는 작지만 높은 성능 발휘.
  - 특정 작업 및 환경에 최적화된 경량 모델.
- 대표 사례:
  - 메타의 라마(LLaMA): 오픈소스 LLM 연구를 선도.
  - 구글의 젬마-2(Gemini-2): 특정 도메인에 최적화된 모델.
  - 이코노소프트 Phi-3: 텍스트를 SQL로 변환하는 언어 추론 능력 강화.
- 기대 효과:
  - 조직 특화 sLLM 개발로 다양한 산업 요구 충족.
  - 경량화된 모델로 효율성과 성능의 균형 달성.

1.3.3 더 효율적인 학습과 추론을 위한 기술

- 트랜스포머 기반 LLM의 연산 비용:
  - 성능 향상을 위해 모델 크기가 커지며 연산량과 GPU 비용 급증.
  - GPU 수요 증가로 인해 높은 비용과 자원 부족 문제가 대두.
- 효율성 개선 기술:
  - 양자화(Quantization): 모델 파라미터를 더 적은 비트로 표현해 연산 비용 절감.
  - LoRA(Low Rank Adaptation): 모델의 일부만 학습하여 효율적 학습과 추론 가능.
  - 무거운 어텐션 최적화: 병렬 연산 개선을 통해 GPU 사용 효율성 증대.
- 기대 효과:
  - 적은 수의 GPU로도 높은 성능 달성 가능.
  - 연구 비용 절감과 GPU 접근성 확대.
  - 에너지 소비 및 탄소 배출 감소로 환경적 이점 제공.

1.3.4 LLM의 환각 현상을 대처하는 검색 증강 생성(RAG) 기술

- LLM의 환각 현상:
  - LLM이 잘못된 정보를 사실처럼 생성하는 문제.
  - 원인: 학습 데이터의 한계와 사실 여부를 판단하지 못하는 구조적 한계.
- 문제 해결 방안:
  - RAG(Retrieval Augmented Generation):
    - 외부 데이터를 미리 검색하여 정보를 추가.
    - LLM의 답변 신뢰도를 높이고 오류를 줄임.
  - 지도 미세 조정:
    - 지시 데이터셋을 통해 LLM을 개선, 환각 현상을 완화.
- 전문가 의견:
  - 존 슐만(John Schulman): 환각 현상은 지시 데이터로 조정 가능.
  - 안드레이 카르파티(Andrej Karpathy): LLM은 "꿈꾸는 것과 비슷한" 생성 구조를 가짐.
- 기대 효과:
  - 잘못된 정보 생성을 줄이고 신뢰성 높은 응답 제공.
  - LLM의 실용성을 강화하고 오류에 대한 사용자 불안을 해소.

1.4 LLM의 미래: 인식과 행동의 확장

- LLM의 발전 방향:
  - 멀티 모달 LLM: 이미지, 비디오, 오디오 등의 다양한 데이터 처리 가능.
  - 에이전트 기능: 계획 수립, 의사결정, 행동 수행 능력 강화.
  - 새로운 아키텍처: 긴 입력 데이터를 효율적으로 처리하며 성능 향상.
- 주요 기술 발전:
  - GPT-4의 멀티 모달 처리 능력: 텍스트 외 다양한 형식 데이터 처리 가능.
  - RAG 기술과의 통합: 텍스트와 이미지를 결합한 검색 및 생성 기능 강화.
- 에이전트의 두뇌 역할:
  - LLM을 기반으로 한 자동화 시스템의 중심 엔진으로 활용.
  - 상황 인식 및 행동 계획을 통해 복합적 문제 해결 가능.
- 향후 기대:
  - LLM 기반 에이전트의 활용 확대로, 다양한 분야에서 혁신적 변화 예고.
  - 멀티 모달과 에이전트 기술의 융합으로 실생활 응용 가능성 극대화.

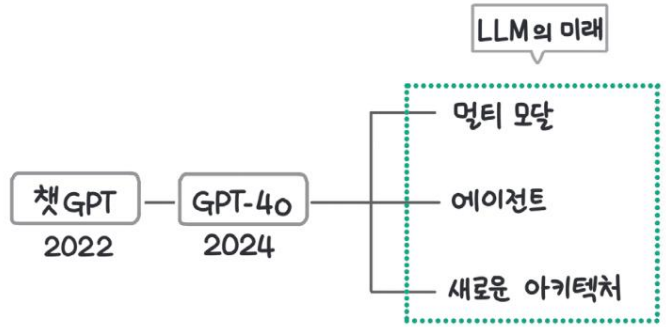

그림 1.25 LLM이 움직이고 있는 방향

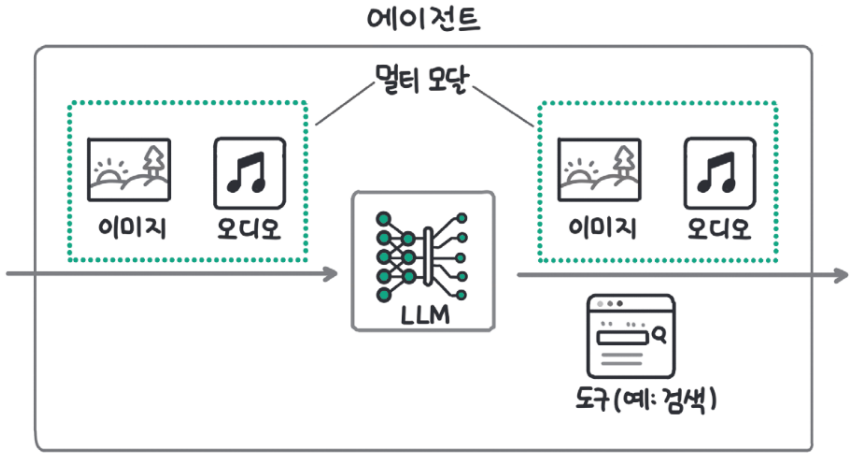

그림 1.26 다양한 데이터 형식을 이해하는 멀티 모달 LLM과 스스로 문제를 해결하는 에이전트

1.5 정리

- LLM의 현재와 가능성:
  - ChatGPT와 같은 대규모 모델은 정렬 기술을 통해 엄청난 가능성을 증명.
  - 최적화된 경량 모델(sLLM)로 효율적이고 높은 성능 실현.
- 멀티 모달 처리의 발전:
  - 텍스트 외에도 이미지, 비디오 등 다양한 데이터 처리 가능.
  - 기존 한계를 넘어서는 기술로 빠르게 발전 중.
- 미래 전망:
  - LLM은 인간의 뇌처럼 다양한 작업을 자동화하며 혁신을 이끌 것.
  - 에이전트 기술과 결합해 실생활 문제 해결 가능성 확대.
- 핵심 기술: 트랜스포머 아키텍처
  - LLM의 기본 구조로, OpenAI의 GPT 시리즈에 활용되어 대화 가능 모델 개발 성공.
  - 효율적 학습과 추론을 위해 다양한 기술과 함께 발전 중.

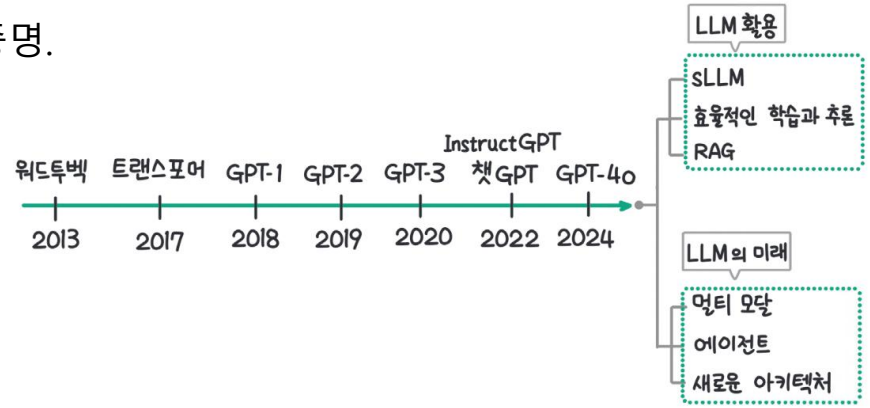

그림 1.27 LLM 지도