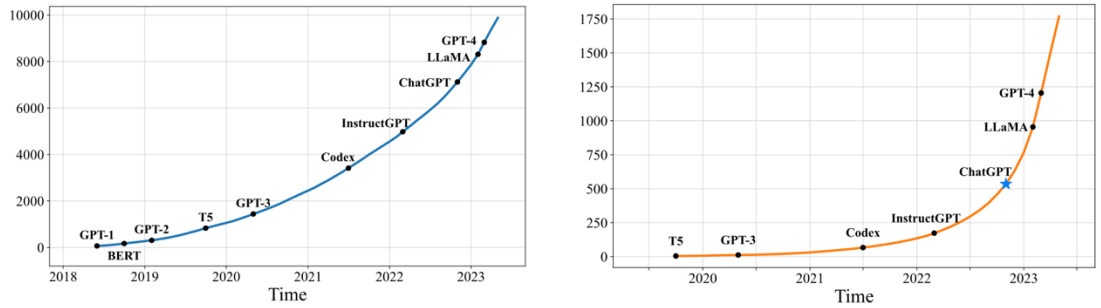
LLM survey

1. 대규모 언어 모델의 부상

- 대규모 언어 모델(Large Language Model, LLM):
  - 수십억 개 이상의 파라미터를 가진 거대한 신경망
  - 방대한 텍스트 데이터를 학습하여 인간의 언어를 이해하고 생성하는 능력을 갖춘 인공지능 모델
  - 특히 OpenAI의 ChatGPT 등장은 전 세계적으로 엄청난 사회적 파급력을 일으킴
  - 며, LLM 기술이 단순한 연구 단계를 넘어 우리 삶의 방식을 혁신할 잠재력을 지녔음을 증명.
  - LLM은 인공 일반 지능(AGI)의 가능성에 대한 논의를 불러일으킴



(a) Query="Language Model"

(b) Query="Large Language Model"

Fig. 1: The trends of the cumulative numbers of arXiv papers that contain the keyphrases *"language model"* (since June 2018) and *"large language model"* (since October 2019), respectively. The statistics are calculated using exact match by querying the keyphrases in title or abstract by months. We set different x-axis ranges for the two keyphrases, because "language models" have been explored at an earlier time. We label the points corresponding to important landmarks in the research progress of LLMs. A sharp increase occurs after the release of ChatGPT: the average number of published arXiv papers that contain *"large language model"* in title or abstract goes from 0.40 per day to 8.58 per day (Figure 1(b)).

Wayne Xin Zhao, et al., "A Survey of Large Language Models", 2025

2. 언어 모델(LM)의 진화: 통계 모델에서 대규모 언어 모델까지

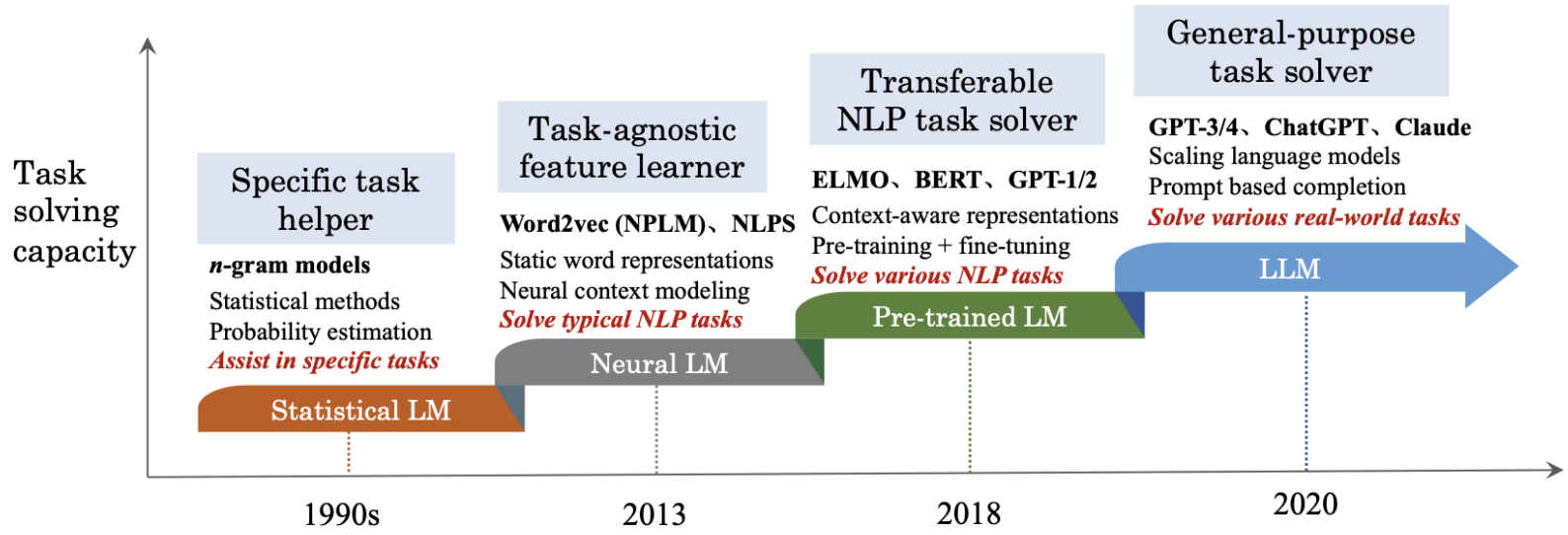


Fig. 2: An evolution process of the four generations of language models (LM) from the perspective of task solving capacity. Note that the time period for each stage may not be very accurate, and we set the time mainly according to the publish date of the most representative studies at each stage. For neural language models, we abbreviate the paper titles of two representative studies to name the two approaches: NPLM [1] ("*A neural probabilistic language model*") and NLPS [2] ("*Natural language processing (almost) from scratch*"). Due to the space limitation, we don't list all representative studies in this figure.

2. 언어 모델(LM)의 진화: 통계 모델에서 대규모 언어 모델까지

- 언어 모델의 4단계 발전 과정
  - 1단계: **통계적 언어 모델** (Statistical Language Models, SLM) 1990년대에 주류를 이룬 SLM은 단어 시퀀스의 등장 확률을 통계적으로 모델링했다. 대표적인 n-gram 모델은 특정 단어의 확률을 이전 n-1개의 단어를 기반으로 추정했다. 이러한 모델들은 범용적인 언어 이해 능력을 갖추기보다는 정보 검색, 음성 인식과 같은 **특정 작업을 보조하는 역할(specific task helper)**을 수행하는 데 중점을 두었다.
  - 2단계: **신경망 언어 모델** (Neural Language Models, NLM) 2000년대 초반, 신경망이 도입되면서 NLM 시대가 열렸다. NLM의 가장 큰 기여는 단어를 고차원 벡터로 표현하는 분산 표현(distributed representation) 개념을 도입한 것이다. 특히 **word2vec**과 같은 모델은 특정 작업에 국한되지 않고 단어의 의미적 관계를 벡터 공간에 인코딩하는 작업 불가지론적 표현(task-agnostic representations) 학습에 초점을 맞추었다.
  - 3단계: **사전 학습 언어 모델 (Pre-trained Language Models, PLM)** 2018년을 기점으로 Transformer 아키텍처와 대규모 코퍼스를 활용한 '사전 학습 및 미세 조정(pre-training and fine-tuning)' 패러다임이 정립되었다. ELMo, BERT, GPT-2와 같은 PLM들은 대규모 비지도 텍스트로 언어의 일반적인 패턴을 학습한 뒤, 특정 작업에 맞게 모델을 미세 조정했다. 이를 통해 모델은 문맥을 깊이 이해하는 표현을 학습하여 **다양한 NLP 작업을 해결(solve various NLP tasks)**할 수 있게 되었고, 전례 없는 성능 향상을 이끌었다.
  - 4단계: **대규모 언어 모델 (Large Language Models, LLM)** 2020년 이후, 모델과 데이터의 크기를 확장하면 성능이 예측 가능하게 향상된다는 **'스케일링 법칙(scaling law)'**이 발견되었다. GPT-3, PaLM과 같이 파라미터 수가 수천억 개에 달하는 모델들은 특정 임계점을 넘어서면서, 작은 모델에서는 나타나지 않던 **인컨텍스트 학습(in-context learning)**과 같은 **'창발적 능력(emergent abilities)'**을 보였다. 이로써 **LLM은 별도의 미세 조정 없이도 **다양한 실제 세계의 작업을 해결(solve various real-world tasks)**할 수 있는 **범용 작업 해결사(general-purpose task solvers)**로 진화**했다.

3. 기반 구축 (GPT)

- GPT-1 (2018): '생성적 사전 학습(Generative Pre-Training)' 개념을 정립했다. Transformer의 **디코더-온리(decoder-only)** 아키텍처를 사용하여 다음 단어를 예측하는 방식으로 대규모 텍스트를 학습하고, 이후 특정 작업에 맞게 미세 조정하는 하이브리드 접근법의 기틀을 마련했다.
- GPT-2 (2019): 모델 크기를 15억 개 파라미터로 확장하고, 명시적인 미세 조정 없이 언어 모델링 자체만으로 다양한 작업을 수행할 수 있다는 **'비지도 다중 작업 학습자(unsupervised multitask learner)'**로서의 가능성을 탐구했다. 이 접근법의 핵심 철학은 "비지도 학습 목표의 전역 최솟값은 (다양한 작업에 대한) 지도 학습 목표의 전역 최솟값이기도 하다"라는 가설에 기반한다. 즉, 모델이 세상의 모든 텍스트에 대해 다음 단어를 예측할 만큼 강력하다면, 이는 텍스트로 표현될 수 있는 모든 작업을 암묵적으로 학습했음을 의미한다.

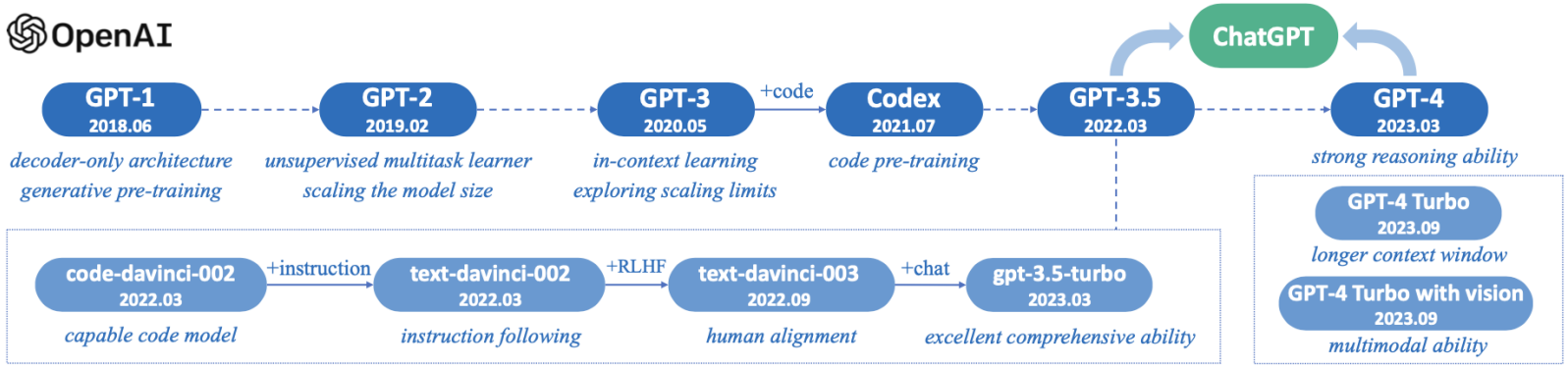


Fig. 4: A brief illustration for the technical evolution of GPT-series models. We plot this figure mainly based on the papers, blog articles and official APIs from OpenAI. Here, *solid lines* denote that there exists an explicit evidence (*e.g.*, the official statement that a new model is developed based on a base model) on the evolution path between two models, while *dashed lines* denote a relatively weaker evolution relation.

3. 기반 구축 (GPT)

- **GPT-3 (2020): 모델 크기를 1750억 개 파라미터로 극적으로 확장하면서 LLM의 시대가 본격적으로 열렸다**. GPT-3는 프롬프트에 몇 가지 예시를 제공하는 것만으로 새로운 작업을 수행하는 '인컨텍스트 학습(**In-Context Learning, ICL**)' 능력을 명확하게 보여주었다. 이로써 모델을 학습시키는 '사전 학습'과 모델을 활용하는 'ICL'이 모두 다음 텍스트를 예측하는 언어 모델링 패러다임으로 수렴되었다.
- GPT-3의 성공 이후, OpenAI는 모델의 능력을 더욱 강화하기 위해 두 가지 주요 접근법을 탐구했다.
  - 코드 데이터 학습: 방대한 GitHub 코드 코퍼스로 GPT-3를 미세 조정하여 Codex 모델을 개발했다. 코드는 논리적 구조가 명확하기 때문에, 코드 데이터 학습은 모델의 복잡한 추론 능력과 수학 문제 해결 능력을 크게 향상시키는 효과를 가져왔다. 이는 GPT-3.5 시리즈의 기반이 되었다.
  - **인간 피드백 기반 강화 학습(RLHF):** LLM이 인간의 지시를 더 잘 따르고, 유용하며 안전한 답변을 생성하도록 만드는 **정렬(alignment)** 기술이다. RLHF는 InstructGPT를 위해 갑자기 발명된 것이 아니라, 인간의 선호도로부터 학습하는 2017년 연구와 PPO(Proximal Policy Optimization) 알고리즘 개발 등 OpenAI에서 수년간 이어진 연구의 정점이었다. InstructGPT를 통해 정립된 3단계 RLHF 알고리즘은 모델이 유해하거나 편향된 콘텐츠 생성을 줄이고 인간의 가치와 더 잘 부합하도록 만들었다.
- **ChatGPT** (2022): InstructGPT와 유사한 방식으로 학습되었지만, 대화 형식에 특별히 최적화되었다. 방대한 지식, 뛰어난 추론 능력, 다중 턴 대화의 문맥 추적 능력, 그리고 인간의 가치에 부합하는 안전성까지 갖추며 AI 챗봇의 역사에 한 획을 그었다.
- **GPT-4** (2023): 기존의 텍스트 입력을 넘어 이미지까지 이해하는 멀티모달(multimodal) 능력을 도입했다. 또한, 더 작은 모델의 성능을 기반으로 대규모 모델의 최종 성능을 정확하게 예측하는 **예측 가능한 스케일링(predictable scaling)**과 같은 새로운 개발 방법론을 제시하며 기술적 성숙도를 한 단계 끌어올렸다.

4. LLM의 핵심 원리 및 기술

- LLM이 이전 세대의 언어 모델과 구별되는 가장 큰 특징은 압도적인 규모에서 비롯되는 독특한 특성들이다.
- 스케일링 법칙은 LLM의 성능(주로 언어 모델링 손실 값)이 모델 크기(파라미터 수 N), 데이터 크기(토큰 수 D), 그리고 투입된 컴퓨팅 자원(C)과 예측 가능한 관계를 맺고 있음을 수학적으로 설명하는 경험적 법칙이다. **Chinchilla (DeepMind, 2022)는** 최적의 성능을 위해서는 **모델 크기와 데이터 크기를 균형 있게** 확장해야 한다고 주장한바 있다.
- 스케일링 법칙의 의미와 한계 : 스케일링 법칙은 LLM 개발에 중요한 지침을 제공했지만, 모든 것을 설명하지는 못한다. Chinchilla 법칙 이후, LLM 개발 트렌드는 단순히 모델을 키우는 것에서 고품질 데이터를 대규모로 확보하여 균형 있게 학습시키는 방향으로 전환되었다. 또 때로는 모델이 커질수록 특정 작업의 성능이 오히려 저하되는 역스케일링(inverse scaling) 현상도 관찰된다. 또한, 스케일링 법칙으로는 다음에 설명할 '창발적 능력'의 등장을 예측할 수 없다.
- **창발적 능력**이란 "작은 모델에서는 나타나지 않지만, 모델의 규모가 특정 임계점을 넘어서면서 갑자기 발현되는 능력"을 의미한다. 대표적인 창발적 능력은 다음과 같다.
  - **인컨텍스트 학습 (In-Context Learning) 모델**의 가중치를 업데이트하지 않고, 프롬프트에 몇 개의 예시(demonstration)를 제공하는 것만으로 새로운 작업을 수행하는 능력이다. GPT-3(175B)에서 명확하게 관찰되기 시작했으며, 이는 LLM을 유연한 문제 해결사로 만드는 핵심 능력이다.
  - **지시 사항 준수 (Instruction Following) 자연어**로 주어진 지시 사항을 이해하고 그에 따라 적절한 결과물을 생성하는 능력이다. 다양한 작업을 자연어 지시문 형태로 학습시키는 **'지시 튜닝(Instruction Tuning)'**을 통해 강화될 수 있으며, LaMDA(68B)와 같은 모델에서부터 유의미한 성능 향상이 나타났다.
  - **단계별 추론 (Step-by-step Reasoning)** 복잡한 문제(예: 수학 응용 문제)를 해결하기 위해 중간 추론 과정을 생성하고 이를 바탕으로 최종 답을 도출하는 능력이다. '**생각의 사슬(Chain-of-Thought, CoT)**' 프롬프팅을 통해 발현되며, PaLM과 같은 100B 이상의 대규모 모델에서 그 효과가 뚜렷하게 나타난다.

4. LLM의 핵심 원리 및 기술

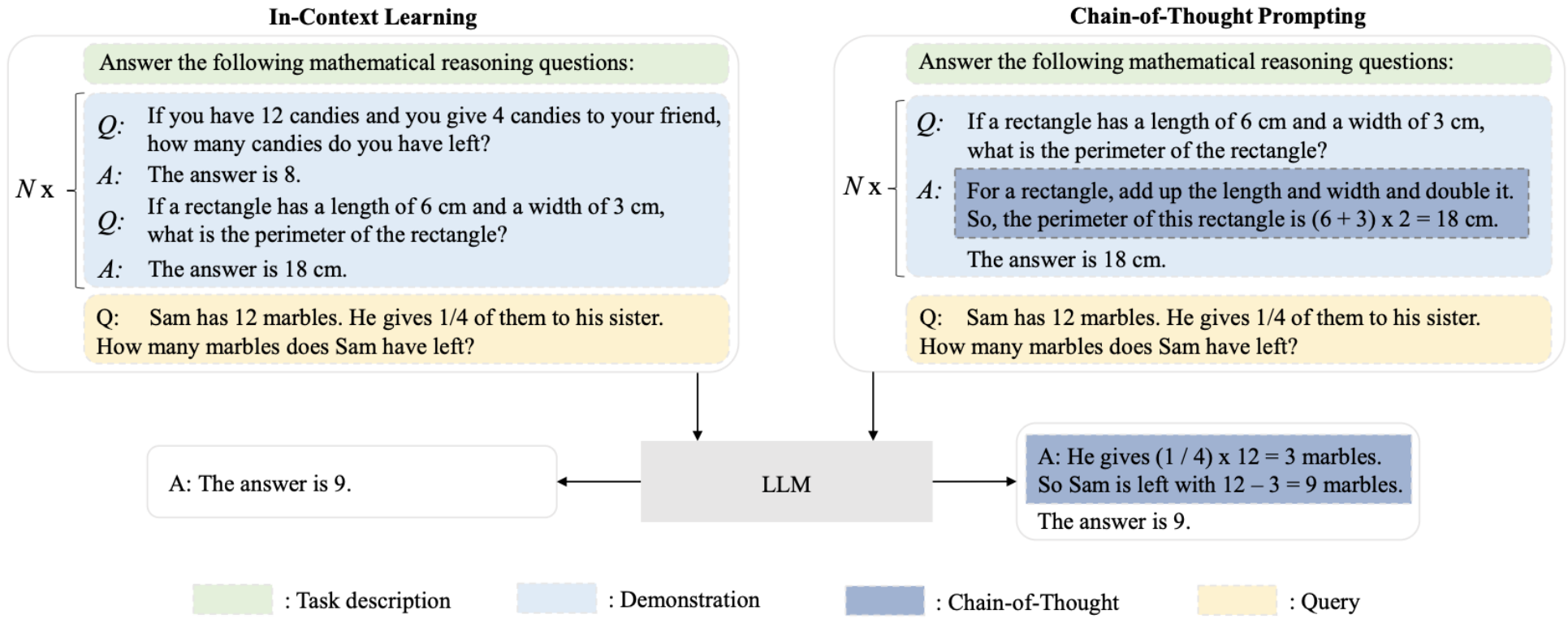


Fig. 14: A comparative illustration of in-context learning (ICL) and chain-of-thought (CoT) prompting. ICL prompts LLMs with a natural language description, several demonstrations, and a test query, while CoT prompting involves a series of intermediate reasoning steps in prompts.

4. LLM의 핵심 원리 및 기술

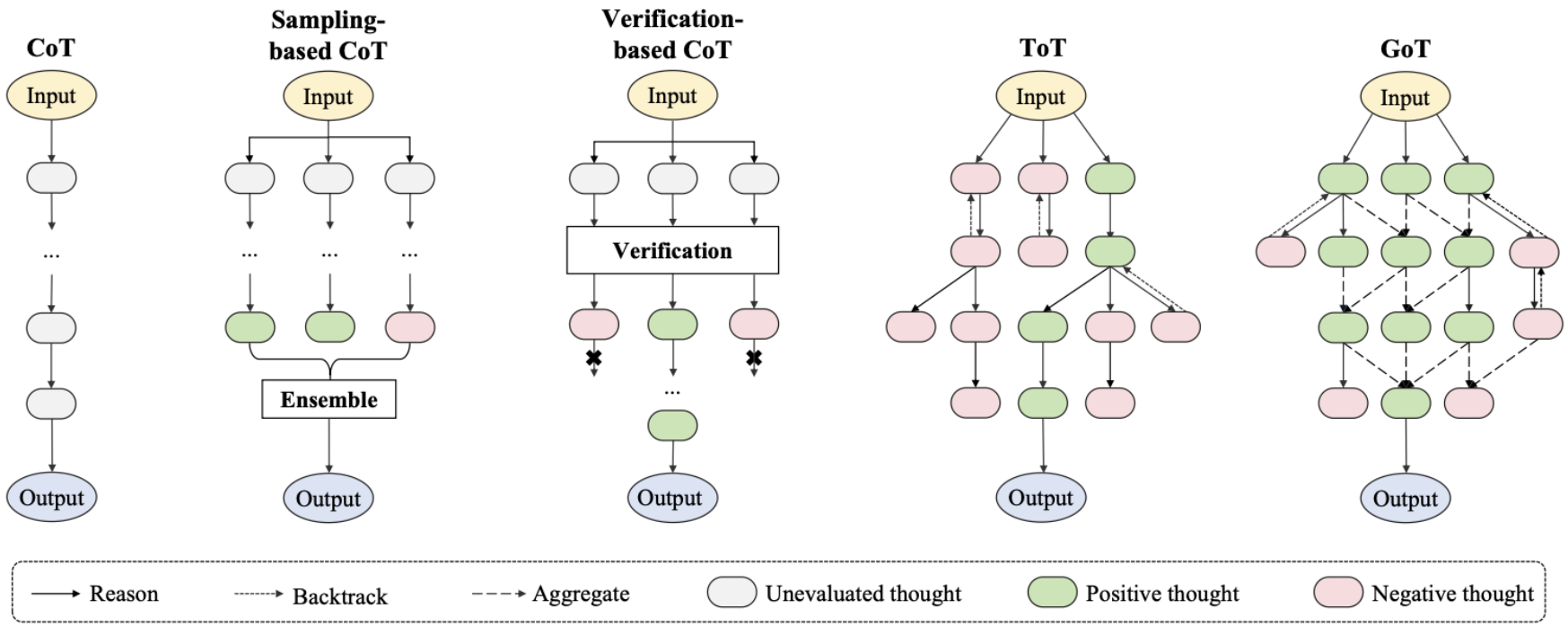


Fig. 15: An illustration of the evolution of CoT prompting strategies. It begins with the basic CoT approach and progresses to enhanced CoT generation techniques, including sampling-based and verification-based methods. Finally, it extends to variations of the chain structure, such as trees and graphs. Here, "thought" refers to an intermediate reasoning step as stated in [33, 444].

4. LLM의 핵심 원리 및 기술

- 그러나 창발적 능력이 실제 '상전이' 현상인지에 대한 학술적 논쟁이 존재한다. 일부 연구에서는 이러한 능력이 갑자기 나타나는 것처럼 보이는 이유가 정확도와 같은 **불연속적인 평가 지표(discontinuous evaluation metrics)**의 사용 때문일 수 있다고 주장한다. 즉, 모델의 내부적인 능력은 점진적으로 향상되지만, 특정 임계값을 넘어야만 평가 지표 상에서 성능이 급격히 도약하는 것처럼 보일 수 있다는 것이다. 이 논쟁은 LLM의 작동 원리에 대한 더 근본적인 이해가 필요함을 시사한다.
- LLM의 성공을 이끈 핵심 기술 요약
  - **스케일링 (Scaling)**: 모델 크기, 데이터 규모, 컴퓨팅 자원을 대규모로 확장하는 것이 성능 향상의 전제 조건이다
  - **학습 (Training)**: 거대한 모델을 안정적으로 학습시키기 위한 분산 학습 알고리즘이 필수적이다.
  - **능력 발현 (Ability Eliciting)**: 사전 학습된 모델이 가진 잠재적 능력을 끌어내는 기술이다. **생각의 사슬(CoT)** 프롬프팅이나 **지시 튜닝(Instruction Tuning)**과 같은 기법을 통해, 모델이 내재된 추론 능력이나 지시 이해 능력을 특정 작업에서 발휘하도록 유도한다.
  - **도구 활용 (Tools Manipulation)**: LLM은 텍스트 데이터로 학습되었기 때문에 부정확한 계산이나 최신 정보 부족과 같은 본질적인 한계를 가진다. 이를 보완하기 위해 계산기, 검색 엔진, 코드 인터프리터와 같은 외부 도구(API)를 호출하여 그 결과를 활용하는 능력이 중요해지고 있다.

5. LLM 아키텍처 심층 분석

TABLE 7: Detailed formulations for the network configurations. Here, Sublayer denotes a FFN or a self-attention module in a Transformer layer, d denotes the size of hidden states, $\mathbf{p}_i$ denotes position embedding at position i, A_{ij} denotes the attention score between a query and a key, r_{i-j} denotes a learnable scalar based on the offset between the query and the key, and $\mathbf{R}_{\Theta,t}$ denotes a rotary matrix with rotation degree $t \cdot \Theta$.

Configuration	Method	Equation
Normalization position	Post Norm [22]	$\text{Norm}(\mathbf{x}+\text{Sublayer}(\mathbf{x}))$
	Pre Norm [26]	$\mathbf{x}+\text{Sublayer}(\text{Norm}(\mathbf{x}))$
	Sandwich Norm [274]	$\mathbf{x}+\text{Norm}(\text{Sublayer}(\text{Norm}(\mathbf{x})))$
Normalization method	LayerNorm [275]	$\frac{\mathbf{x}-\mu}{\sigma}\cdot\gamma+\beta,\quad \mu=\frac{1}{d}\sum_{i=1}^{d}x_i,\quad \sigma=\sqrt{\frac{1}{d}\sum_{i=1}^{d}(x_i-\mu))^2}$
	RMSNorm [276]	$\frac{\mathbf{x}}{\text{RMS}(\mathbf{x})}\cdot\gamma,\quad \text{RMS}(\mathbf{x})=\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}$
	DeepNorm [277]	$\text{LayerNorm}(\alpha\cdot\mathbf{x}+\text{Sublayer}(\mathbf{x}))$
Activation function	ReLU [278]	$\text{ReLU}(\mathbf{x})=\max(\mathbf{x},\mathbf{0})$
	GeLU [279]	$\text{GeLU}(\mathbf{x})=0.5\mathbf{x}\otimes[1+\text{erf}(\mathbf{x}/\sqrt{2})],\quad \text{erf}(x)=\frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$
	Swish [280]	$\text{Swish}(\mathbf{x})=\mathbf{x}\otimes\text{sigmoid}(\mathbf{x})$
	SwiGLU [281]	$\text{SwiGLU}(\mathbf{x}_1,\mathbf{x}_2)=\text{Swish}(\mathbf{x_1})\otimes\mathbf{x_2}$
	GeGLU [281]	$\text{GeGLU}(\mathbf{x}_1,\mathbf{x}_2)=\text{GeLU}(\mathbf{x_1})\otimes\mathbf{x_2}$
Position embedding	Absolute [22]	$\mathbf{x}_i=\mathbf{x}_i+\mathbf{p}_i$
	Relative [82]	$A_{ij}=\mathbf{W}_q\mathbf{x}_i\mathbf{x}_j^T\mathbf{W}_k^T+r_{i-j}$
	RoPE [282]	$A_{ij}=\mathbf{W}_q\mathbf{x}_i\mathbf{R}_{\Theta,i-j}\mathbf{x}_j^T\mathbf{W}_k^T=(\mathbf{W}_q\mathbf{x}_i\mathbf{R}_{\Theta,i})(\mathbf{W}_k\mathbf{x}_j R_{\Theta,j})^T$
	ALiBi [283]	$A_{ij}=\mathbf{W}_q\mathbf{x}_i\mathbf{x}_j^T\mathbf{W}_k^T-m(i-j)$

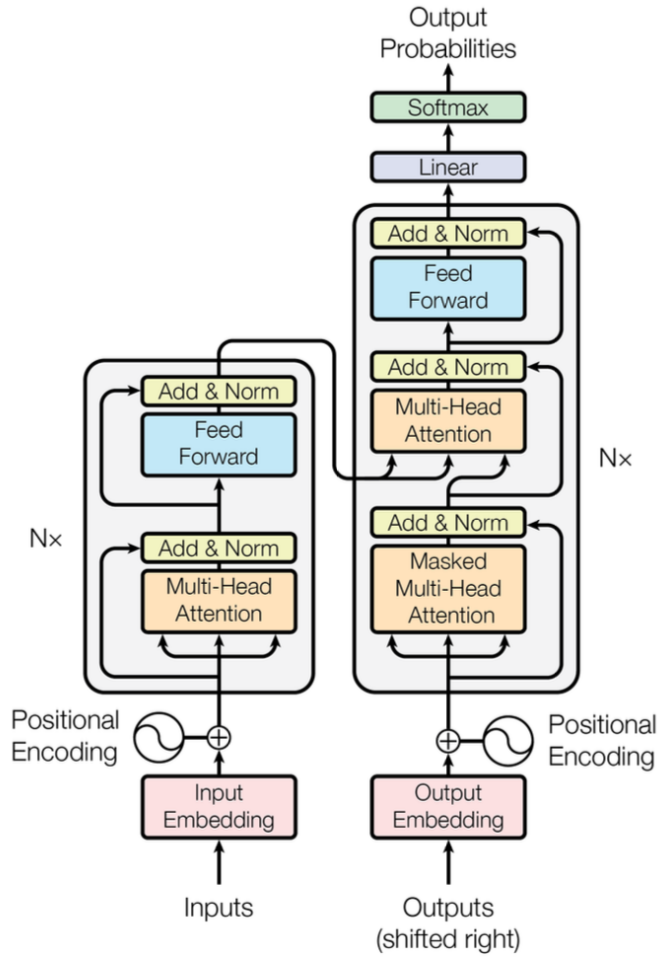


Figure 1: The Transformer - model architecture.

5. LLM 아키텍처 심층 분석

- 어텐션 메커니즘 최적화
- 어텐션 계산은 **시퀀스 길이에 따라 제곱에 비례하는 계산량(quadratic complexity)**을 가져 LLM의 병목이 된다. 이를 최적화하기 위한 기법들이 개발되었다.
- **FlashAttention**: GPU 메모리 계층 구조(SRAM, HBM)의 I/O 특성을 고려하여 어텐션 계산 커널을 최적화한 기법이다. 불필요한 메모리 읽기/쓰기 작업을 줄여 기존 어텐션 대비 속도와 메모리 사용량을 크게 개선했다.
- **PagedAttention**: LLM 서빙 시 생성되는 키-값(KV) 캐시를 운영체제의 페이징 기법처럼 관리하여 메모리 파편화를 줄이고 GPU 활용률을 극대화하는 기법이다. 이를 통해 서빙 처리량을 크게 향상시킬 수 있다.

- 사전 학습 목표(Pre-training Tasks) 분석
- LLM은 대규모 코퍼스로부터 다음 두 가지 주요 사전 학습 목표를 통해 언어적 지식과 패턴을 학습한다.
- 언어 모델링 (Language Modeling): 가장 보편적인 사전 학습 방식으로, 이전 토큰 시퀀스가 주어졌을 때 다음 토큰을 예측하도록 모델을 학습시킨다. 이는 GPT와 같은 디코더-온리 아키텍처의 핵심 학습 목표이며, 모델이 텍스트 생성 능력을 자연스럽게 습득하게 한다.
- 디노이징 오토인코딩 (Denoising Autoencoding): 입력 텍스트의 일부를 의도적으로 손상(예: 마스킹)시키고, 모델이 원본 텍스트를 복원하도록 학습시키는 방식이다. T5와 같은 인코더-디코더 모델에서 주로 사용되며, 문맥 이해 능력을 강화하는 데 효과적이다.

6. LLM 학습 및 적응 방법론

- LLM의 강력한 능력은 두 가지 핵심 단계를 통해 구축되고 정제된다. 첫 번째는 방대한 텍스트 데이터로부터 세상의 지식과 언어 패턴을 학습하는 **사전 학습(Pre-training) 단계이며**, 두 번째는 사전 학습된 모델을 인간의 의도에 맞게 유용하고 안전하게 만드는 **적응 튜닝(Adaptation Tuning) 단계**이다.
- 사전 학습 데이터 준비 과정: 사전 학습 데이터의 품질은 LLM의 성능을 좌우하는 가장 중요한 요소 중 하나이다. 고품질 코퍼스를 구축하는 과정은 크게 세 단계로 나눌 수 있다.

- 데이터 수집 및 구성
- LLM은 일반 텍스트 데이터와 전문 텍스트 데이터를 혼합하여 학습한다.
  - • 일반 텍스트 데이터: 웹페이지(예: CommonCrawl), 도서(예: BookCorpus), 대화 텍스트 등은 모델이 폭넓은 상식과 언어적 유창성을 습득하는 기반이 된다.
  - • 전문 텍스트 데이터: 과학 문헌(예: arXiv), 코드(예: GitHub), 백과사전(예: Wikipedia) 등은 모델의 특정 분야 지식과 논리적 추론 능력을 강화한다.
- 주요 모델들은 이러한 데이터 소스들을 특정 비율로 혼합하는 데이터 혼합(Data Mixture) 전략을 사용한다. 예를 들어, LLaMA(65B)의 학습 데이터 구성은 CommonCrawl (67%), C4 (15%), GitHub (4.5%), Wikipedia (4.5%), 도서 (4.5%), ArXiv (2.5%), StackExchange (2%)로, 웹페이지 데이터가 큰 비중을 차지하면서도 코드, 학술 자료 등을 포함하여 균형 잡힌 능력을 추구한다.

6. LLM 학습 및 적응 방법론

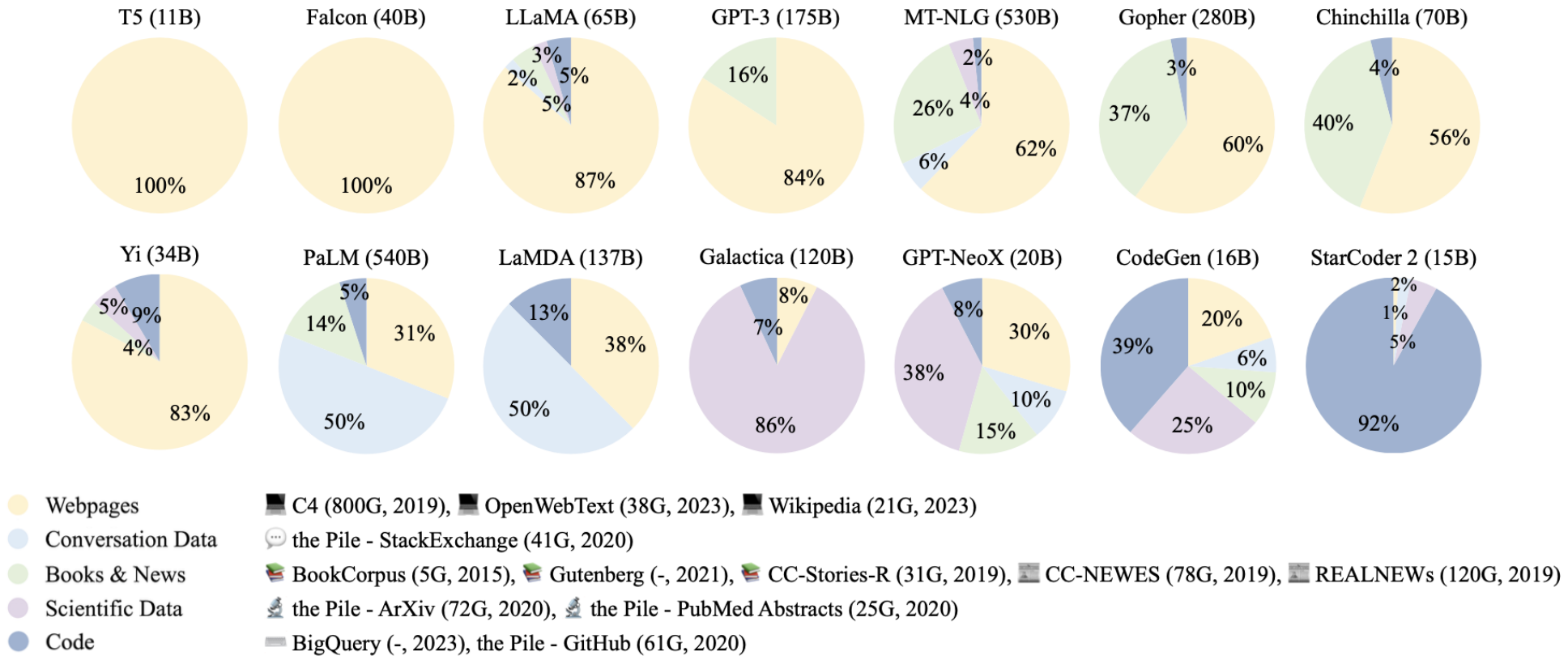


Fig. 6: Ratios of various data sources in the pre-training data for existing LLMs.

6. LLM 학습 및 적응 방법론

- 데이터 전처리 파이프라인
- 수집된 원시 데이터는 그대로 사용하기 어렵기 때문에, 다음과 같은 정교한 전처리 파이프라인을 거쳐 고품질 코퍼스로 정제된다.
  - **품질 필터링 (Quality Filtering):** 저품질 데이터를 걸러내는 과정이다. 고품질 데이터(예: 위키피디아)를 학습한 분류기를 사용하거나, 문장 길이, 특수문자 비율 등 휴리스틱 규칙을 기반으로 저품질 문서를 제거한다.
  - **중복 제거 (De-duplication):** 문장, 문서, 데이터셋 수준에서 중복된 내용을 제거한다. 이는 데이터의 다양성을 확보하고, 특정 내용이 과도하게 학습되어 발생하는 훈련 불안정성 및 편향 문제를 완화하는 데 매우 중요하다.
  - **개인정보 제거 (Privacy Reduction)**: 이름, 주소, 전화번호 등 개인 식별 정보(PII)를 탐지하고 제거하여 프라이버시 침해 위험을 줄인다.
  - **토큰화 (Tokenization)**: 원시 텍스트를 모델이 처리할 수 있는 작은 단위인 '토큰(token)' 시퀀스로 분할하는 과정이다. 어휘 크기와 시퀀스 길이의 균형을 맞추기 위해, 단어보다 작은 단위로 텍스트를 분할하는 하위 단어(subword) 토크나이저(예: BPE, WordPiece)가 주로 사용된다.

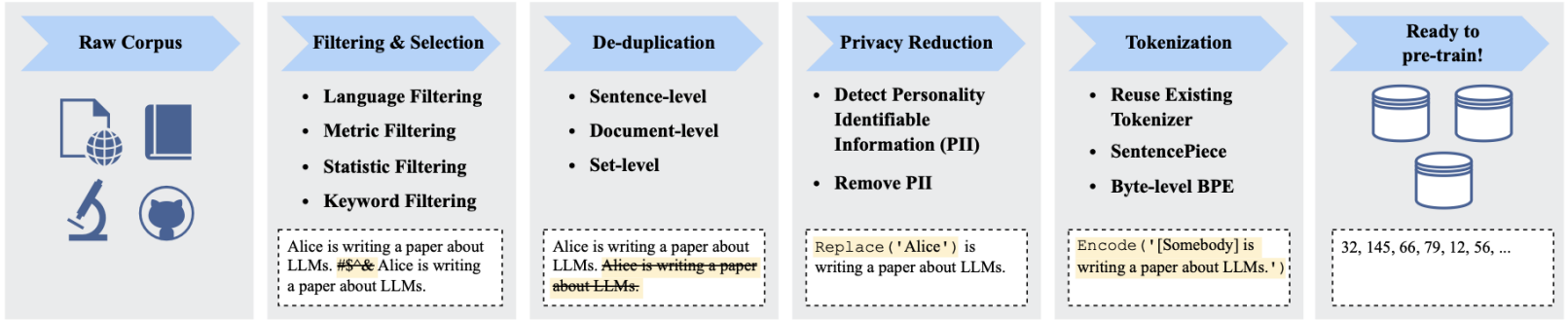


Fig. 7: An illustration of a typical data preprocessing pipeline for pre-training large language models.

6. LLM 학습 및 적응 방법론

- **데이터 스케줄링 (Data Scheduling)**: 훈련 과정에서 어떤 데이터를 어떤 순서로 사용할지 결정하는 전략 또한 중요하다.
  - **데이터 혼합 (Data Mixture)**: 전체 훈련 과정에서 각 데이터 소스가 차지하는 비율을 정하는 전략이다. 이는 모델이 특정 능력(예: 코딩, 추론)을 갖추도록 데이터 구성을 설계하는 데 사용된다.
  - **데이터 커리큘럼 (Data Curriculum)**: 훈련 단계에 따라 데이터의 구성이나 순서를 변경하는 전략이다. 예를 들어, 초기에는 고품질의 일반 텍스트로 학습하고, 후반부에는 특정 도메인 데이터의 비중을 높여 전문성을 강화할 수 있다.
  - **적응 튜닝(Adaptation Tuning) 방법론** : 사전 학습을 마친 LLM은 방대한 지식을 갖춘 '기반 모델(Foundation Model)'이지만, 아직 특정 작업을 수행하거나 인간의 복잡한 요구를 이해하는 데에는 미숙하다. 적응 튜닝은 이러한 기반 모델을 특정 목적에 맞게 미세 조정하여 그 잠재력을 최대한 끌어내는 과정이다.
- **지시 튜닝 (Instruction Tuning)**
  - 개념: 지시 튜닝은 LLM의 사전 학습 목표(다음 단어 예측)와 사용자의 실제 목표(지시를 유용하고 안전하게 따르는 것) 사이의 간극을 줄이기 위한 핵심적인 방법론이다. 사전 학습된 LLM을 (지시문, 출력) 형태의 데이터셋으로 미세 조정함으로써, 모델은 단순히 텍스트를 이어가는 것을 넘어 주어진 지시의 의도를 파악하고 이전에 보지 못한 새로운 작업에 대해서도 일반화하는 능력을 학습한다.
  - 지시 데이터셋 구축 방법: 지시 튜닝의 성공은 고품질 지시 데이터셋 확보에 달려 있으며, 구축 방법은 크게 세 가지로 나뉜다.
    - 기존 NLP 데이터셋 활용: SQuAD(질의응답), CNN/DailyMail(요약)과 같은 기존 NLP 데이터셋에 "다음 질문에 답하세요:", "이 글을 요약하세요:"와 같은 자연어 작업 설명을 추가하여 지시 형식으로 변환한다. (예: P3, FLAN 데이터셋)
    - 실제 사용자 데이터 활용: 실제 사용자들이 LLM에게 요청한 쿼리나 대화 데이터를 수집하여 데이터셋을 구축한다. 이는 모델이 현실 세계의 다양하고 예측 불가능한 요구에 더 잘 대응하도록 돕는다. (예: ShareGPT, Dolly 데이터셋)
    - 합성 데이터 생성: 강력한 LLM(예: GPT-4)을 사용하여 소수의 시드(seed) 데이터로부터 대규모의 새로운 지시 데이터를 자동으로 생성하는 방법이다. 인간의 수동 레이블링에 드는 높은 비용과 제한된 규모의 문제를 해결하여 Alpaca, WizardLM과 같은 데이터셋을 빠르게 구축할 수 있게 되면서 지배적인 방법론이 되었다.

6. LLM 학습 및 적응 방법론

- **정렬 튜닝 (Alignment Tuning)**
  - 필요성: LLM이 단순히 지시를 잘 따르는 것을 넘어, 인간 사회의 가치 및 선호도와 일치하는 방식으로 작동하도록 만드는 과정이다. 정렬의 목표는 모델을 유용하고(Helpful), 정직하며(Honest), 무해한(Harmless), 이른바 '3H' 원칙에 부합하도록 만드는 것이다.
  - 인간 피드백 기반 강화 학습 (RLHF): 정렬 튜닝을 위한 가장 대표적인 방법론으로, InstructGPT에서 정립된 3단계 프로세스를 따른다.
    - 지도 미세 조정 (Supervised Fine-Tuning, SFT): 인간 전문가가 직접 작성한 고품질의 (지시문, 응답) 시연 데이터로 모델을 1차 미세 조정한다. 이 단계는 모델이 바람직한 응답의 스타일과 형식을 학습하도록 가르친다.
    - 보상 모델 (Reward Model, RM) 학습: 하나의 지시문에 대해 SFT 모델이 여러 개의 응답을 생성하면, 인간 평가자가 어떤 응답이 더 나은지 선호도 순위를 매긴다. 이 **쌍대 비교 데이터(pairwise comparison data)**를 사용하여, 인간이 어떤 응답을 선호할지 예측하는 보상 모델을 학습시킨다.
    - 강화 학습 (Reinforcement Learning): SFT 모델의 정책(policy)을 PPO와 같은 강화 학습 알고리즘을 사용하여 다시 미세 조정한다. 이때 모델은 보상 모델로부터 더 높은 점수를 받는 응답을 생성하도록 업데이트되며, 이를 통해 모델의 출력을 인간의 선호도에 부합하는 방향으로 '조정'한다.

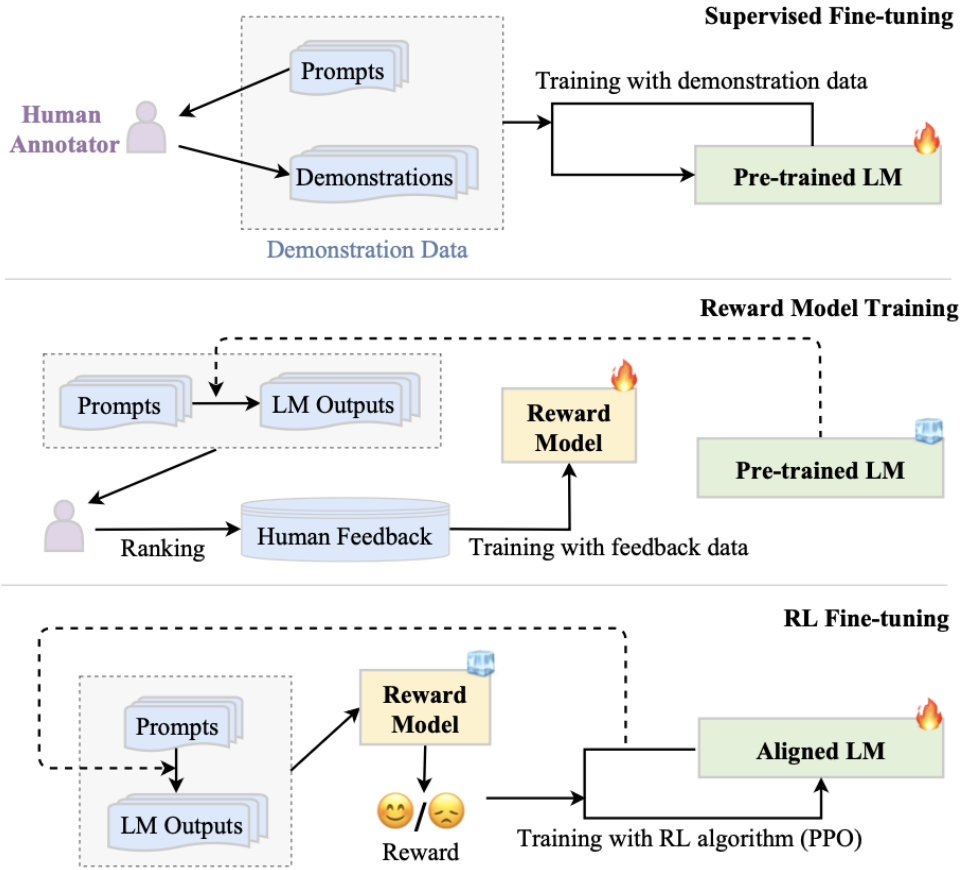


Fig. 12: The workflow of the RLHF algorithm.

6. LLM 학습 및 적응 방법론

- **파라미터 효율적 미세 조정 (PEFT)**
  - LLM의 모든 파라미터를 미세 조정하는 것은 막대한 계산 비용을 요구한다. **파라미터 효율적 미세 조정(PEFT)**은 사전 학습된 모델의 대부분 파라미터는 고정한 채, 소수의 추가 파라미터만 업데이트하여 계산 비용과 메모리 사용량을 크게 줄이는 방법론이다.
  - **LoRA (Low-Rank Adaptation)**: 가장 널리 사용되는 PEFT 기법 중 하나이다. LoRA는 미세 조정 중 가중치의 변화량(ΔW)이 낮은 '내재적 차원(intrinsic rank)'을 가질 것이라는 가설에 기반한다. **따라서 거대한 ΔW 행렬을 직접 학습하는 대신, LoRA는 ΔW를 재구성하는 두 개의 훨씬 작은 저차원 행렬(A와 B)의 곱으로 근사한다.** 이를 통해 수억에서 수십억에 달하는 학습 가능한 파라미터 수를 수천에서 수백만 수준으로 극적으로 줄일 수 있다.

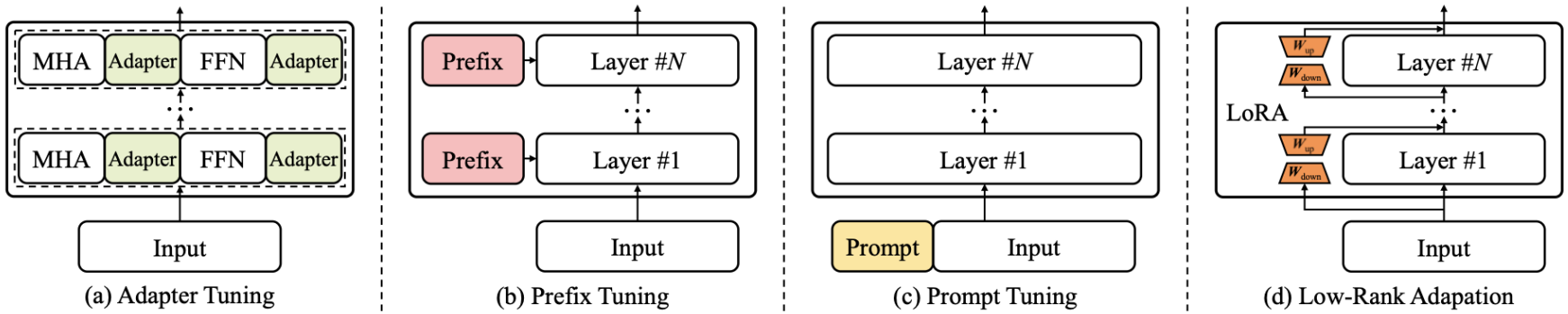


Fig. 13: An illustration of four different parameter-efficient fine-tuning methods. MHA and FFN denote the multi-head attention and feed-forward networks in the Transformer layer, respectively.

7. 기술적 한계점

- **환각 (Hallucination)**
  - 개념: 사실이 아닌 정보를 마치 사실인 것처럼 그럴듯하게 꾸며내어 생성하는 현상입니다. 예를 들어, 존재하지 않는 논문을 인용하거나 역사적 사실을 왜곡하는 답변을 매우 자신감 있게 제시할 수 있습니다.
  - 원인: LLM은 '진실'을 판단하는 능력을 가진 것이 아니라, 훈련 데이터에 기반하여 통계적으로 '가장 확률이 높은' 다음 단어를 예측하는 모델이기 때문에 발생합니다. 훈련 데이터에 포함된 오류, 모델이 지식을 부정확하게 암기한 경우, 또는 확률적으로 그럴듯한 문장을 만드는 훈련 목표와 사실성을 유지하는 것 사이의 불일치 등이 복합적으로 작용합니다.
- **지식 단절 (Knowledge Cutoff)**
  - 개념: LLM의 지식은 사전 학습에 사용된 데이터의 특정 시점에 고정되어 있습니다. 따라서 그 시점 이후에 발생한 최신 사건이나 새롭게 발견된 정보에 대해서는 전혀 알지 못합니다. 예를 들어, 어제 발표된 신제품에 대해 물으면 모른다고 하거나 잘못된 정보를 생성할 수 있습니다.
  - 원인: LLM을 한 번 훈련시키는 데에는 막대한 비용과 시간이 소요되기 때문에, 실시간으로 세상의 모든 정보를 반영하여 모델을 업데이트하는 것은 현실적으로 불가능합니다. (앞서 배운 '검색 증강 생성(RAG)' 기술이 이 문제의 효과적인 해결책 중 하나입니다.)
- **편향성 (Bias)**
  - 개념: LLM은 인터넷의 방대한 텍스트 데이터를 학습 자료로 사용합니다. 이 데이터에는 인종, 성별, 국적, 직업 등에 대한 인간 사회의 다양한 편견과 고정관념이 그대로 녹아있습니다. LLM은 이러한 편향을 그대로 학습하여, 답변을 생성하는 과정에서 특정 집단에 대한 차별적인 내용을 재현하거나 심지어 증폭시킬 수 있습니다.
  - 원인: 근본적인 원인은 훈련 데이터 자체가 인간 사회의 편향을 반영하고 있기 때문입니다. '쓰레기는 쓰레기를 낳는다(Garbage in, garbage out)'는 말처럼, 편향된 데이터로 학습한 모델은 편향된 결과를 생성할 수밖에 없습니다.

8. 주요 과제 및 미래 방향 분석

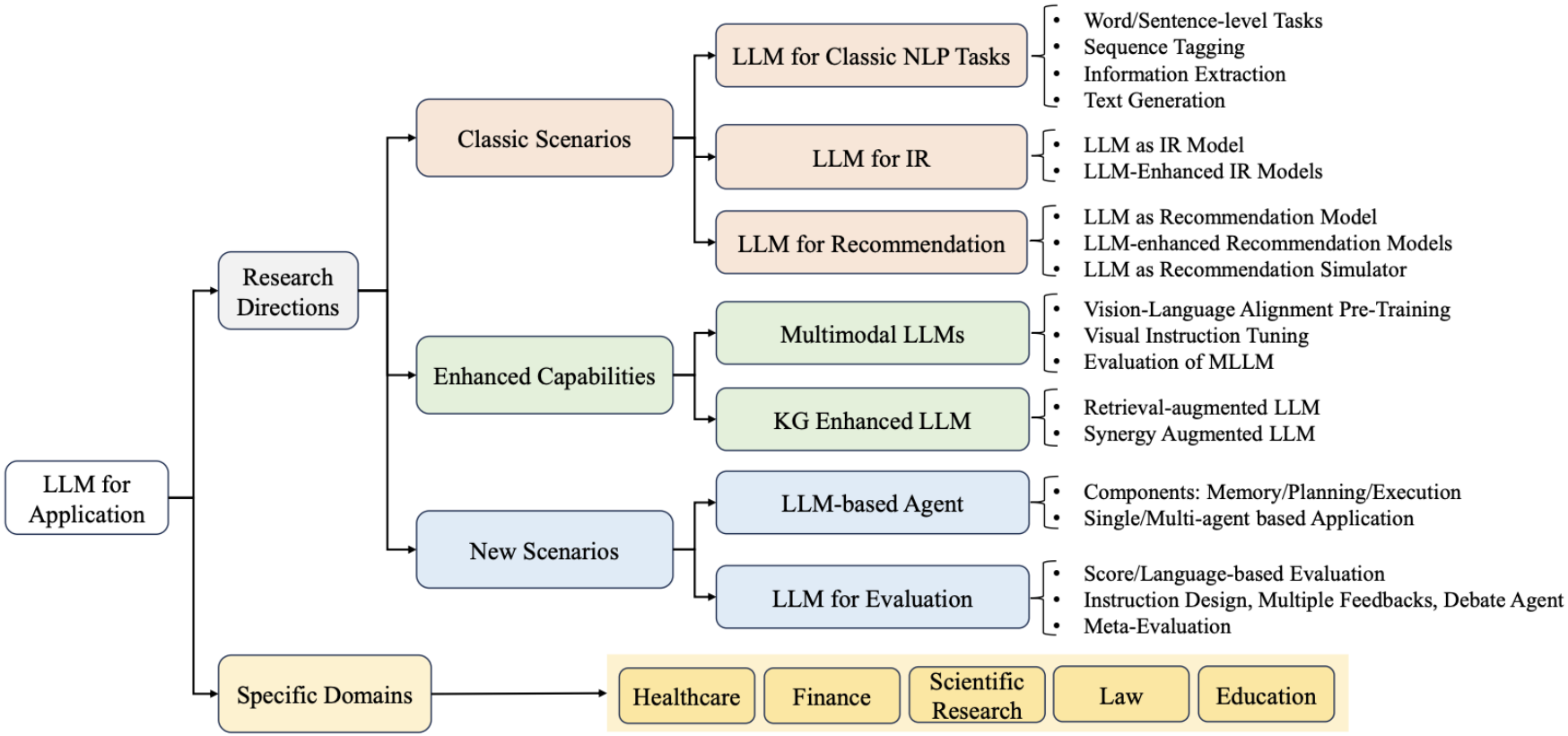


Fig. 18: The applications of LLMs in representative research directions and downstream domains.

8. 주요 과제 및 미래 방향 분석

- LLM 기술은 놀라운 발전을 이루었지만, 여전히 해결해야 할 근본적인 과제들을 안고 있다. 향후 연구는 다음과 같은 네 가지 방향으로 집중될 것으로 예상된다.
- **기초 및 원리 (Basics and Principles)**: '다음 단어 예측'이라는 단순한 목적 함수로부터 어떻게 복잡한 추론과 같은 고차원적 능력이 창발하는지에 대한 이론적 프레임워크의 부재는 여전히 핵심적인 도전 과제이다. 이 메커니즘을 규명하는 것은 더 신뢰성 있고 예측 가능한 모델을 구축하는 데 필수적이다.
- **모델 아키텍처 (Model Architecture)**: 현 주류인 Transformer 아키텍처는 시퀀스 길이에 따라 계산량이 급증하여 비용이 높고 추론 속도가 느리다는 한계를 가진다. 상태 공간 모델(SSM)과 같은 새로운 아키텍처 탐색과 FlashAttention과 같은 시스템 수준 최적화를 통해 기존 아키텍처의 효율성을 극대화하려는 연구가 계속될 것이다.
- **모델 활용 (Model Utilization)**: 프롬프팅(ICL, CoT 등)은 LLM의 잠재력을 끌어내는 강력한 방법이지만, 그 작동 원리는 아직 명확히 규명되지 않았다. 어떤 프롬프트가 왜 더 효과적인지에 대한 근본적인 연구를 통해, 특정 작업에 최적화된 프롬프트를 더 효율적으로 찾고 설계하는 방법론이 발전할 것이다.
- **애플리케이션 및 생태계 (Application and Ecosystem)**: LLM을 핵심 두뇌로 활용하여 외부 도구를 사용하고, 스스로 계획을 세워 복잡한 작업을 수행하는 **자율 에이전트(Autonomous Agent)**와 같은 지능형 시스템 개발이 가속화될 것이다. 이는 LLM이 단순한 챗봇을 넘어, 실제 환경과 상호작용하는 주체로 발전할 가능성을 시사하며, LLM 기반 애플리케이션 생태계는 더욱 성장하여 우리 삶의 다양한 영역에 깊숙이 통합될 것이다.