언어모델(Language Model)과 거대언어모델(Large Language Model)

Hyopil Shin

hpshin@snu.ac.kr

http://nlp.snu.ac.kr

목차

언어모델(Language Model)

Predict Next Word

- P(새빨간 거짓말) > P(새빨간 희망)
- P(이 강의는 참 재미있어요)
- P(재미있어요| 이 강의는)

Assign a Probability to a sentence

- Statistical Language Model: N-gram based -Unigram, Bigram, Trigram…
- Neural Network based: Word Embedding (Static Language Model)
- Transformer –based (Dynamic Contextual) : Autoencoding(BERT), Autoregressive(GPT)
- Large Language Model : GPT, PaLM, LLaMA, LLaMA2, HyperCloverX…

확률적 언어모델(Statistical Language Model): N-Gram

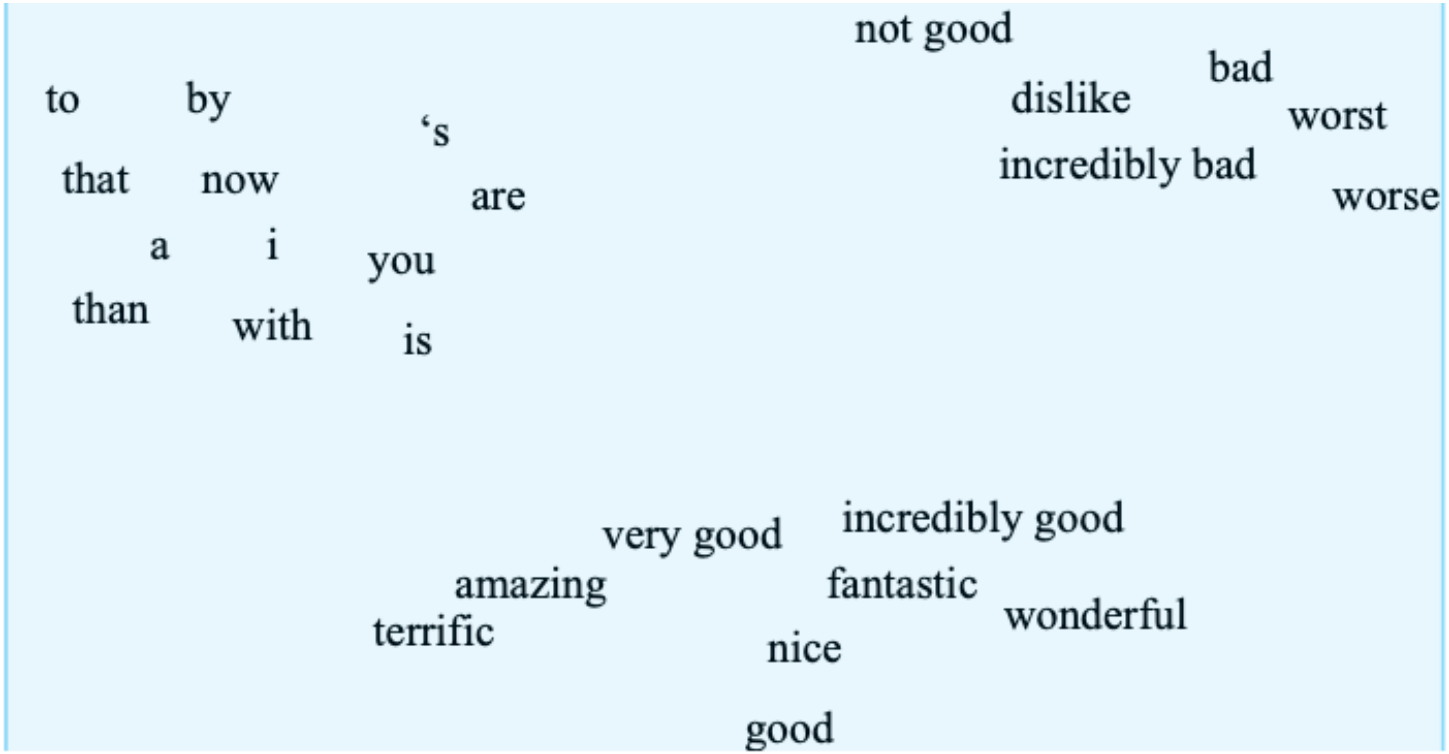
- 문장의 확률을 어떻게?
  - 조건부 확률: P(B|A) = P(A, B)/P(A) → P(A, B) = P(A)p(B|A)
    - P(A, B, C, D) = P(A)P(B|A)P(C|A, B)P(D|A, B, C)
  - Chain Rule:
    - $P(X_1, X_2, X_3, \ldots X_n) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2) \ldots P(X_n|X_1, \ldots X_{n-1})$
    - P("이 강의는 참 재미 있어요") = P(이) X P(강의는|이) X P(참|이 강의는) X P(재미 | 이 강의는 참) X P(있어요 | 이 강의는 참 재미)
    - P(있어요 | 이 강의는 참 재미) = Count(이 강의는 참 재미 있어요) / Count(이 강의는 참 재미)
- Markov Assumption
  - P(있어요| 이 강의는 참 재미) ≈ P(있어요 | 재미) 또는 P(있어요 | 참 재미)
  - P("이 강의는 참 재미 있어요") = P(이) X P(강의는|이) X P(참|강의는) X P(재미 | 참) X P(있어요 | 재미)

$$P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i \mid w_{i-k} \ldots w_{i-1})$$

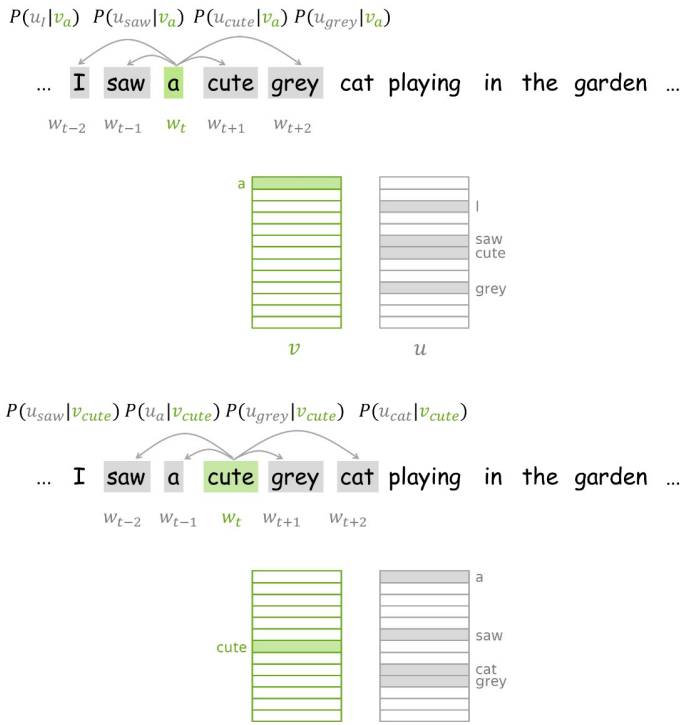
$$P(w_1 w_2 \ldots w_n) = \prod_i P(w_i \mid w_1 w_2 \ldots w_{i-1})$$

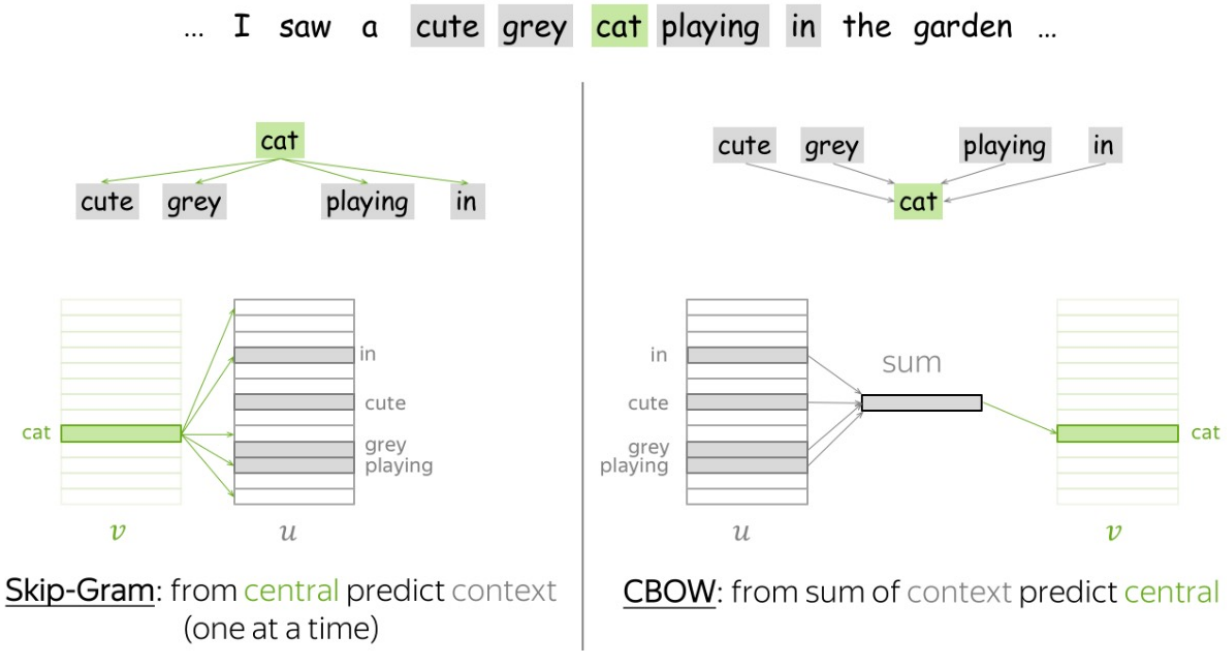
Neural Network based Language Model: 단어 임베딩(Word Embedding)

- Numericalization : 텍스트를 숫자로 바꾸는 방법
  - One-hot encoding, N-Gram 언어 모델, Bag of Words 언어 모델…
- Vector 의미론(Semantics)
  - 단어의 의미를 어떻게 표상할 수 있는가?
    - 동음이의어, 유사어, 다의어, 관련어…
- Word Embedding
  - Distributional Hypothesis
  - Word2Vec, Glove, FaxtText..
  - Static Word Embedding (다의어, 동의어 구별이 되지 않음)

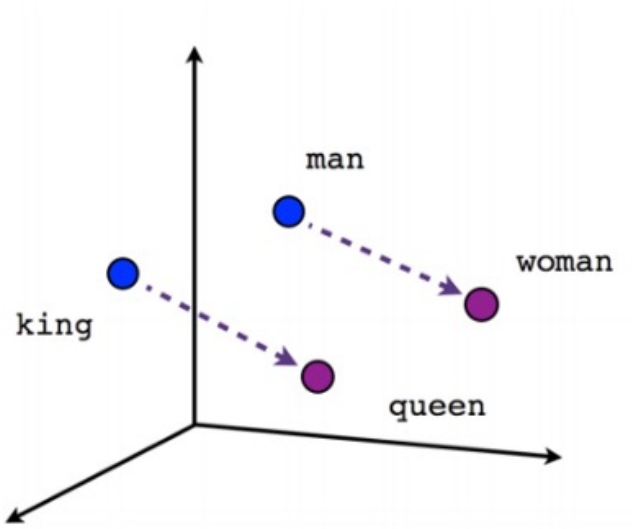


Word2Vec Language Modeling: Skip-gram/Continuous bag of words

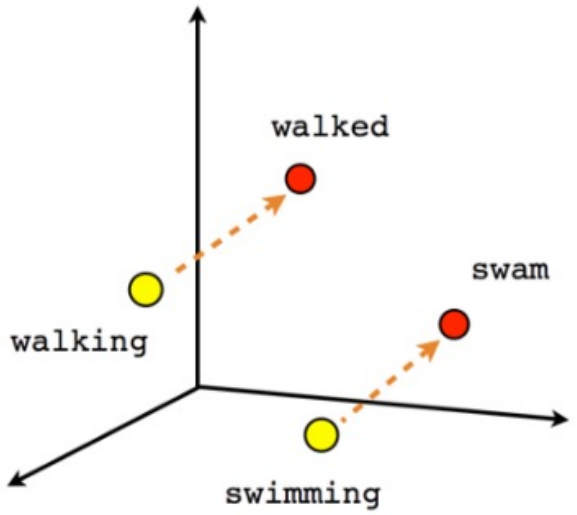




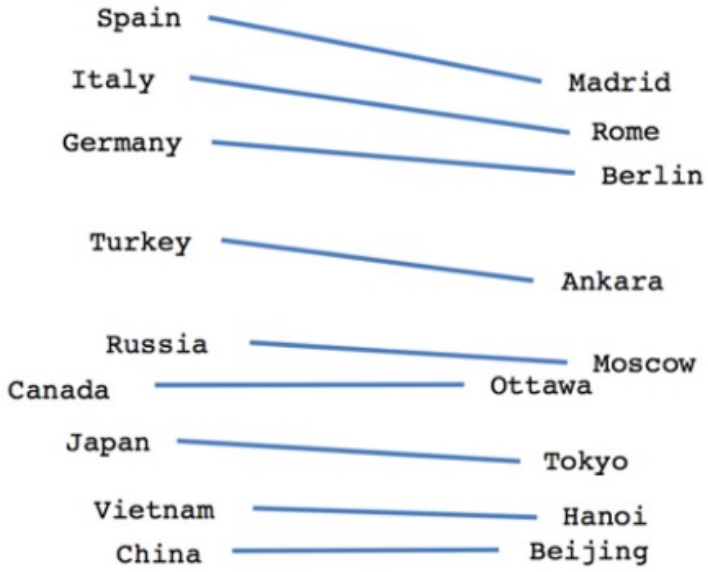
단어 임베딩



Male-Female



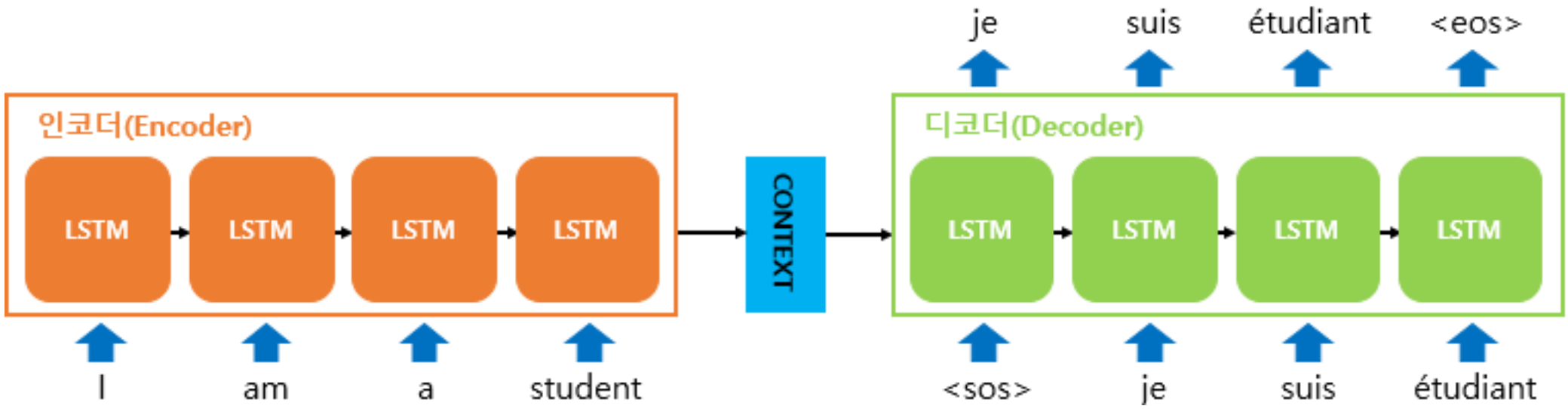
Verb tense

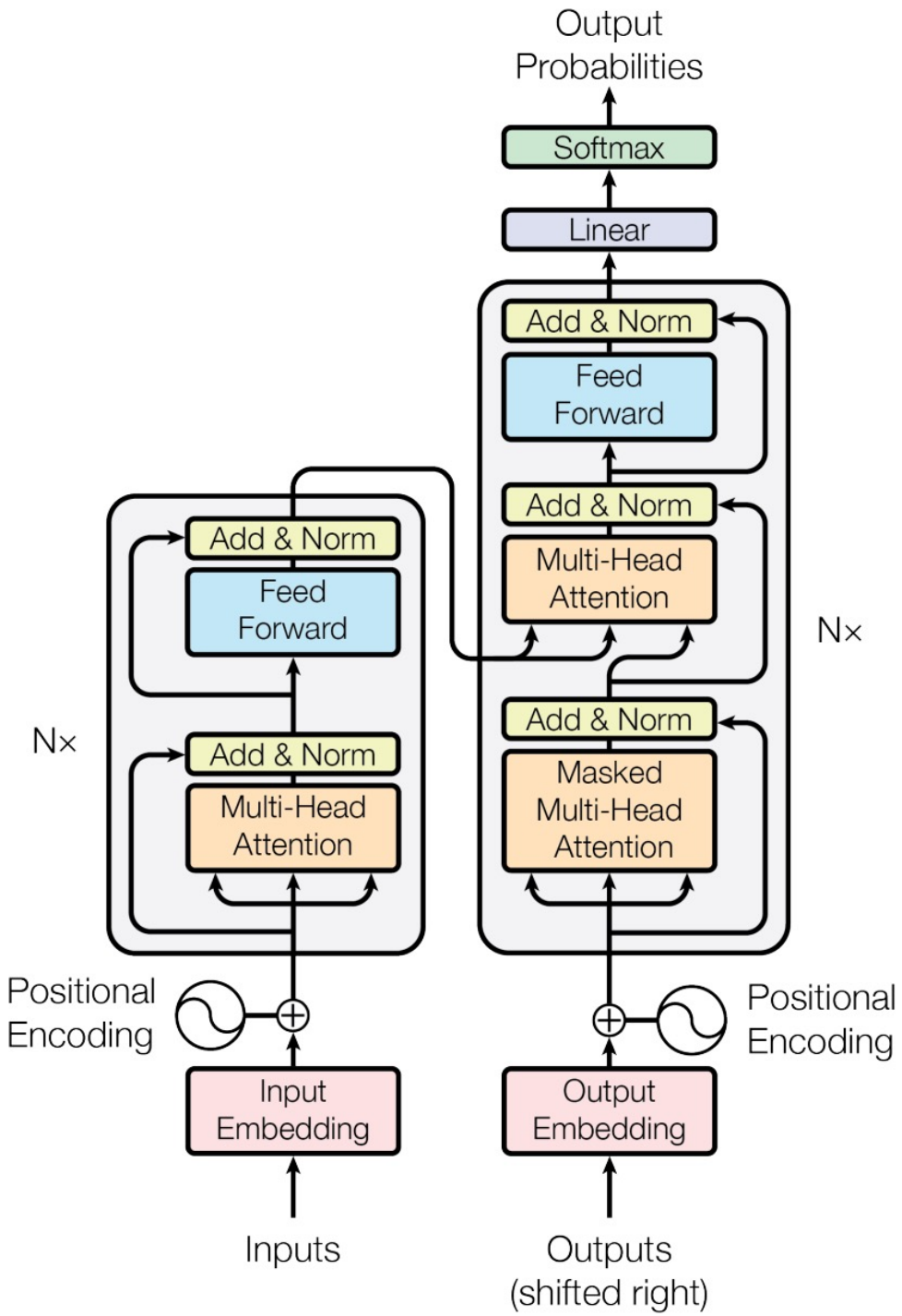


Country-Capital

Transformer-based Language Model: Sequence To Sequence/Attention

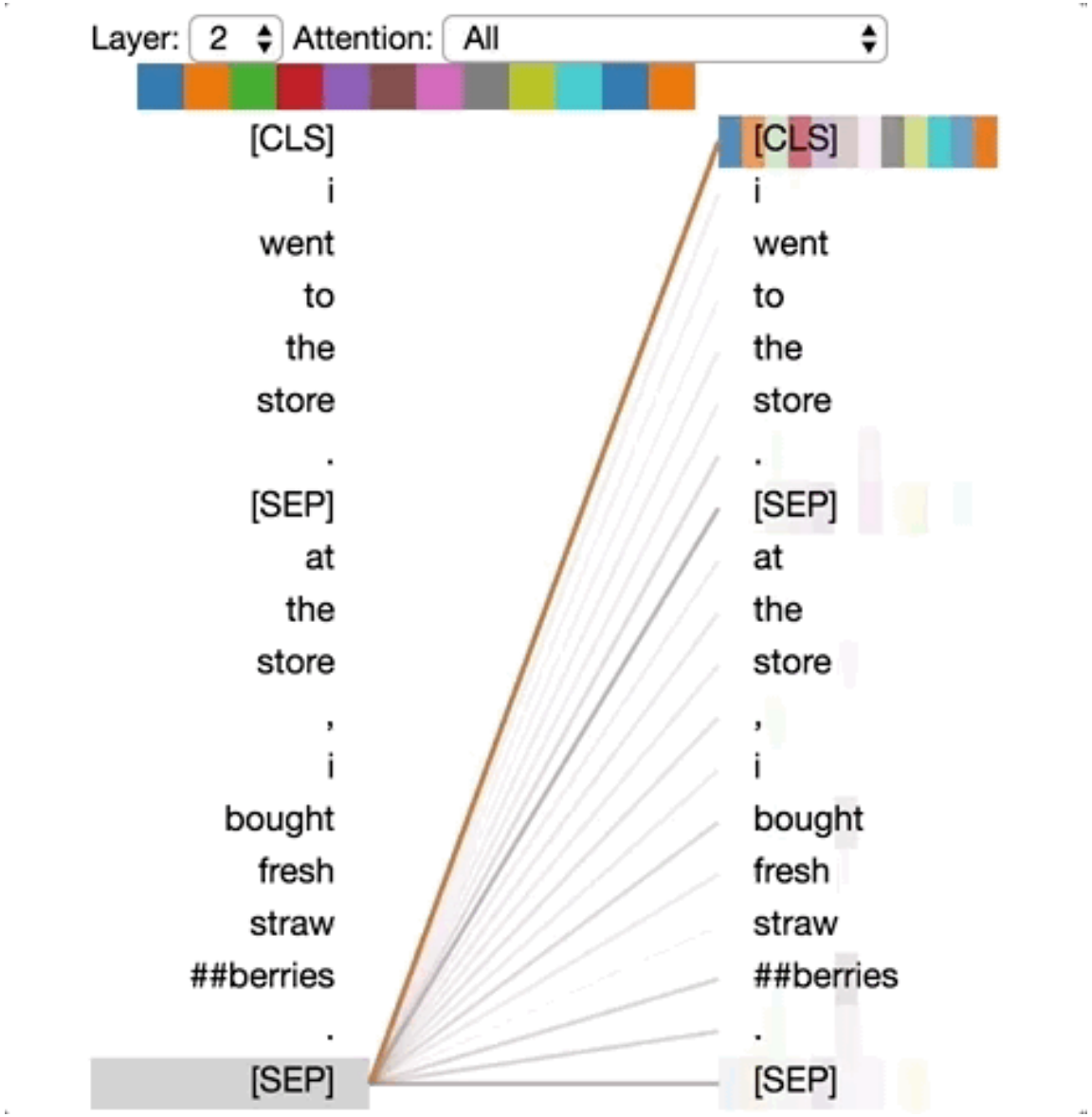
- Sequence to sequence Model
- Encoder-Decoder





Transformer

- Parallel Processing
- Tokenization
- Positional Embedding
- Three types of Attentions: Multi-head Self-Attention, Masked Multi-head Self-Attention, Cross Attention
- Layer Normalization
- Residual Connection



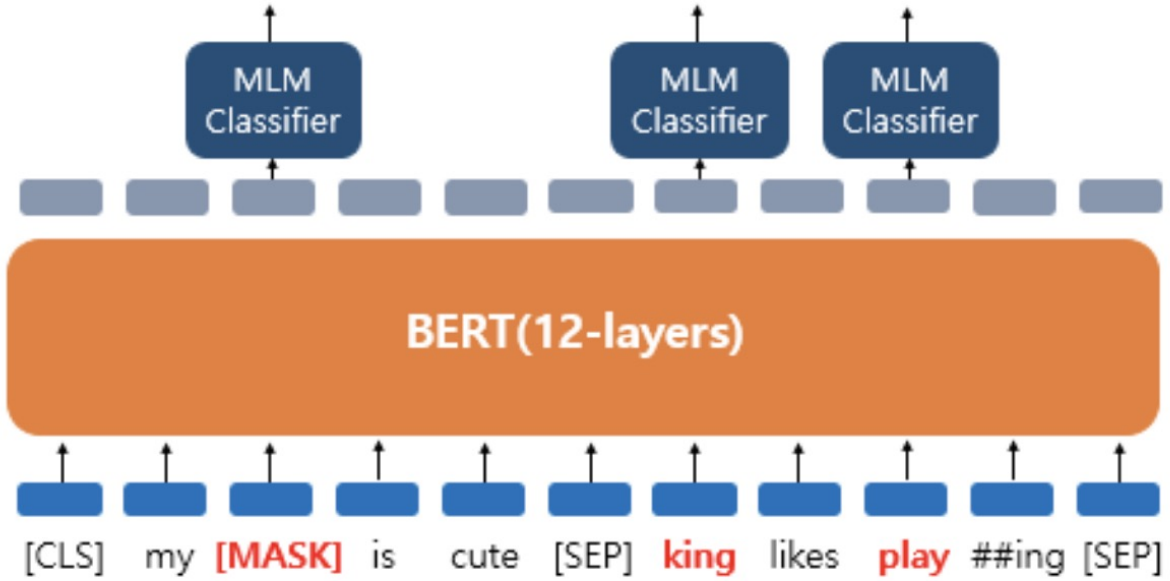
트랜스포머(Transformer)

Self Attention

https://towardsdatascience.com/deconstructing-bert-distilling-6-patterns-from-100-million-parameters-b49113672f77

Masked Language Modeling(MLM)in BERT

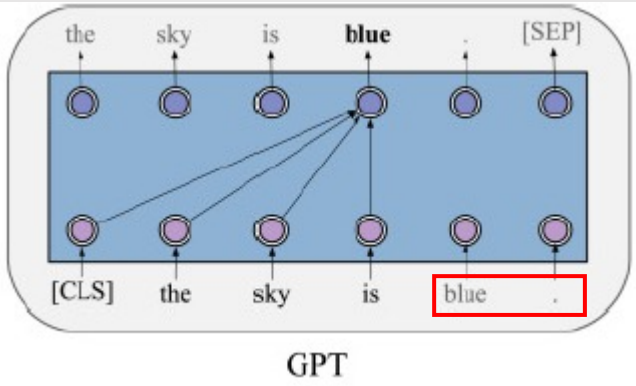
- 80% of the time the words were replaced with the masked token [MASK]
- 10% of the time the words were replaced with random words
- 10% of the time the words were left unchanged



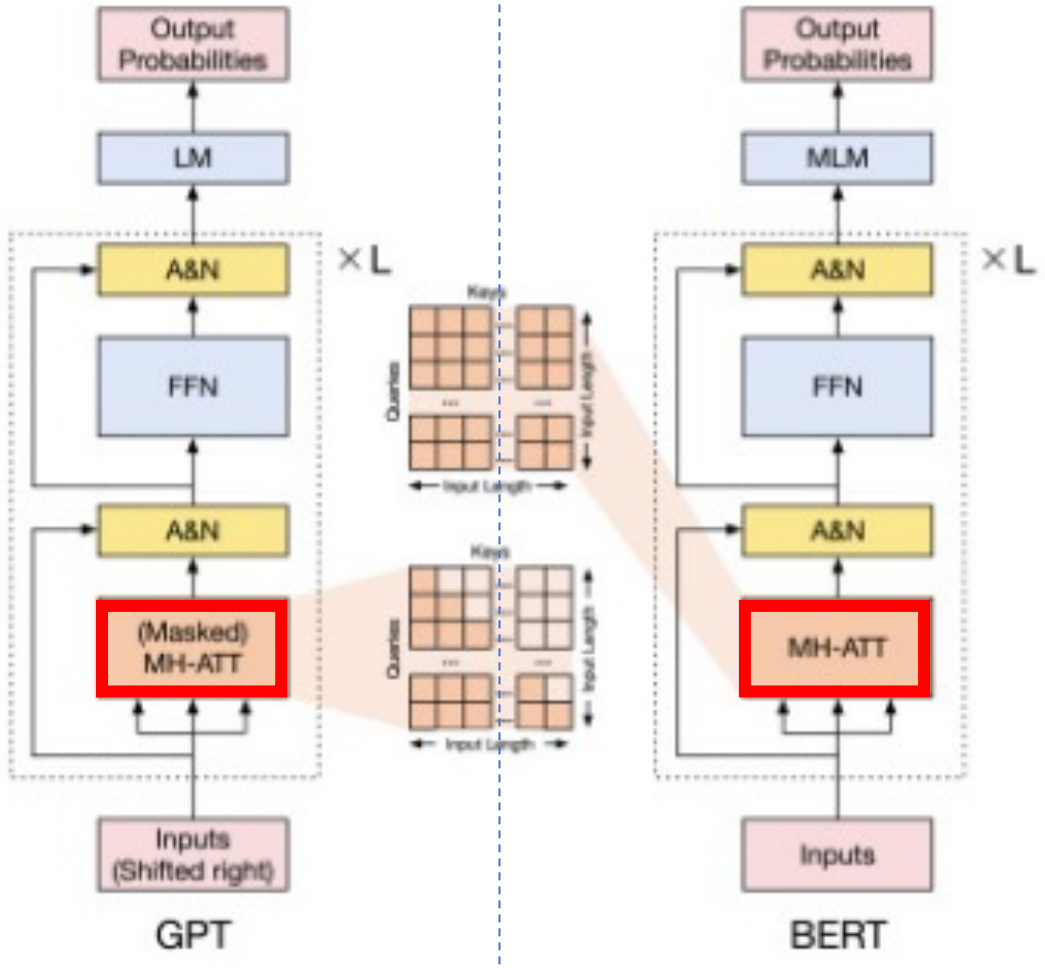
GPT

Generative Pre-trained Transformer

- Given large-scale corpora without labels, GPT optimizes a standard **autoregressive language modeling**, that is, maximizing the conditional probabilities of all the words by taking their previous words as contexts.
- The adaptation procedure of GPT to specific tasks is fine-tuning, by using the pre-trained parameters of GPT as a start point of downstream tasks.



GPT

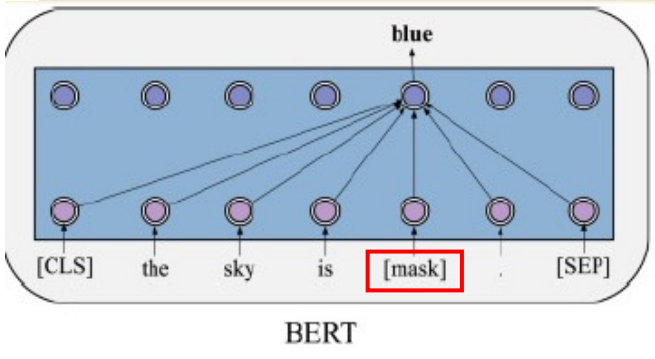


GPT BERT

BERT

Bidirectional Encoder Representations from **Transformers**

- In the pre-training phase, BERT applies **autoencoding language modeling** rather than autoregressive language modeling used in GPT. More specifically, inspired by cloze (Taylor, 1953), the objective masked language modeling (MLM) is designed.
- By modifying inputs and outputs with the data of downstream tasks, BERT could be fine-tuned for any NLP tasks.



BERT

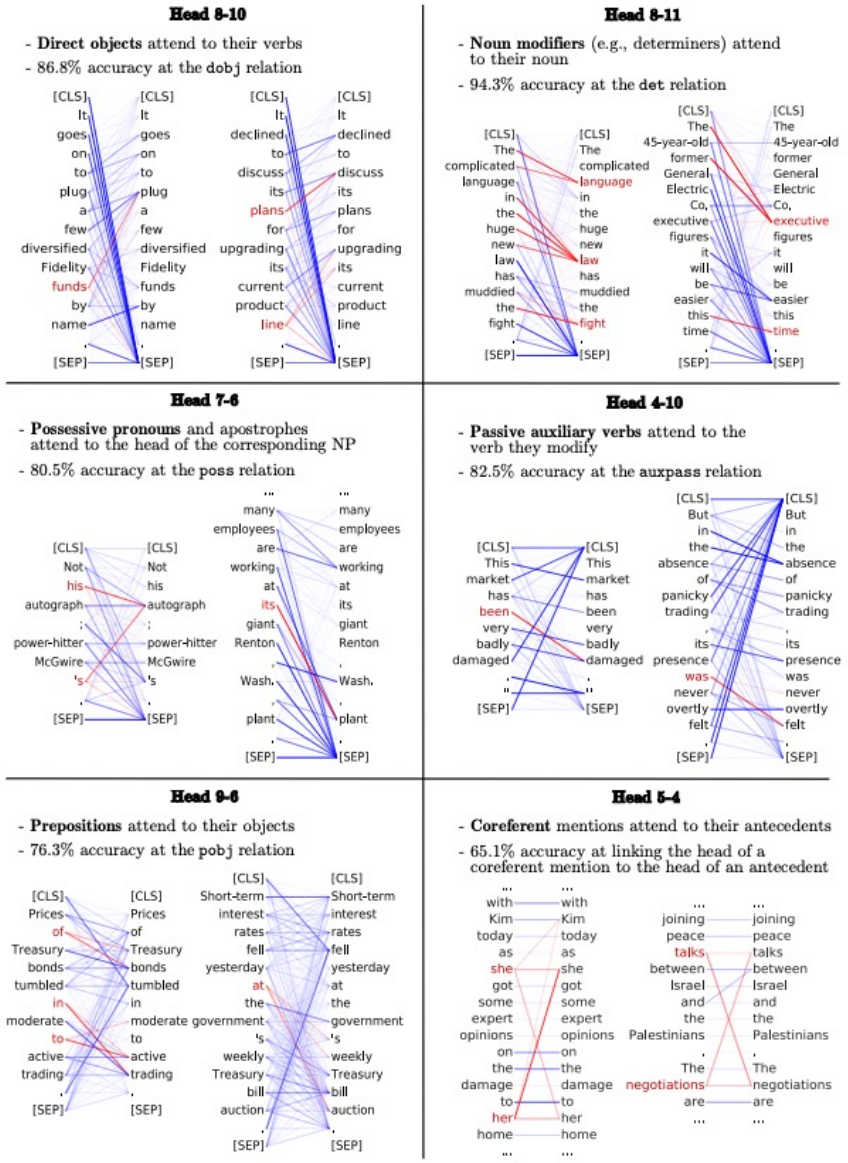
Investigating Linguistic Knowledge in Language Models

- Many recent studies have searched for evidence that neural networks(NNs) learn representations that implicitly encode grammatical concepts
- Still it is an open question how the linguistic knowledge of state-of-the-art language models varies across the linguistic phenomena
- Two Methods for evaluating NN's linguistic knowledge
  - Probing tasks : in which a classifier is trained to directly predict grammatical properties of a sentence or part of a sentence using only the NN's learned representation as input
  - Acceptability judgements: to address the same question without the need for training data labeled with grammatical concepts
    - Acceptability judgements are the main form of behavioral data used in generative linguistics to measure human linguistic competence

Analyzing the Attention Mechanism: Clark et al. 2019 - What Does BERT Look At? An Analysis of BERT's Attention

- They propose methods for analyzing the attention mechanisms of pre-trained models and apply them to BERT
- They show that certain attention heads correspond well to linguistic notions of syntax and coreference
  - They find heads that attend to the direct objects of verbs, determiners of nouns, objects of prepositions, and coreferent mentions with remarkably high accuracy
  - They propose an attention-based probing classifier and use it to further demonstrate that substantial syntactic information is captured in BERT's attention
  - They treat each head as a simple no-training-required classifier that, given a word as input, outputs the most-attended-to other word: Particular heads correspond remarkably well to particular relations
    - Heads that find direct objects of verbs, determiner of nouns, objects of prepositions, and objects of possessive pronouns with > 75% accuracy

BERT ATTENTION HEADS THAT CORRESPOND TO LINGUISTIC PHENOMENA



Acceptability Judgement: BLiMP(The Benchmark of Linguistic Minimal Pairs for English) (Warstadt et al. 2020)

- Uses minimal pairs to infer whether LMs detect specific grammatical contrasts
- Consists of 67 minimal pair paradigms, each with 1,000 sentence pairs in mainstream American English grouped into 12 categories
  - Refer Minimal pair types as *paradigms* and categories *phenomena*
  - Automatically generate the data from linguist-crafted grammar templates, and automatic labels are validated with crowd-sourced human judgments
  - Each minimal pair type corresponds to exactly one paradigm, a particular fact about English grammar may be illustrated by multiple paradigms
    - Determiner-nouns agreement vs. changing the number marking of the nouns

  - *Rachelle had bought those chair*
- Overall accuracy on BLiMP is simply the proportion of the 67,000 minimal pairs in which the model assigns a higher probability to the acceptable sentence

Phenomenon	N	Acceptable Example	Unacceptable Example
ANAPHOR AGR.	2	*Many girls insulted themselves.*	*Many girls insulted herself.*
ARG. STRUCTURE	9	*Rose wasn't disturbing Mark.*	*Rose wasn't boasting Mark.*
BINDING	7	*Carlos said that Lori helped him.*	*Carlos said that Lori helped himself.*
CONTROL/RAISING	5	*There was bound to be a fish escaping.*	*There was unable to be a fish escaping.*
DET.-NOUN AGR.	8	*Rachelle had bought that chair.*	*Rachelle had bought that chairs.*
ELLIPSIS	2	*Anne's doctor cleans one important book and Stacey cleans a few.*	*Anne's doctor cleans one book and Stacey cleans a few important.*
FILLER-GAP	7	*Brett knew what many waiters find.*	*Brett knew that many waiters find.*
IRREGULAR FORMS	2	*Aaron broke the unicycle.*	*Aaron broken the unicycle.*
ISLAND EFFECTS	8	*Whose hat should Tonya wear?*	*Whose should Tonya wear hat?*
NPI LICENSING	7	*The truck has clearly tipped over.*	*The truck has ever tipped over.*
QUANTIFIERS	4	*No boy knew fewer than six guys.*	*No boy knew at most six guys.*
SUBJECT-VERB AGR.	6	*These casseroles disgust Kayla.*	*These casseroles disgusts Kayla.*

Transformer 기반의 Language Model: 사전학습모델 (Pre-Trained Models)

Decoders or autoregressive models:

- GPT, GPT-2, GPT3. CTRL(Conditional Transformer Language Model for Controllable Generation, Transformer-XL, Reformer, XLNet

Encoders or autoencoding models

- BERT, ALBERT, RoBERTa, DistilBERT, ConvBERT, XLM, XLM-RoBERTa, FlauBERT, ELECTRA, Funnel Transformer, Longformer

Sequence-to-sequence models

- BART, Pegasus, MarianMT, T5, MT5, Mbart, ProphetNet, XLM-ProphetNet

Multimodal models

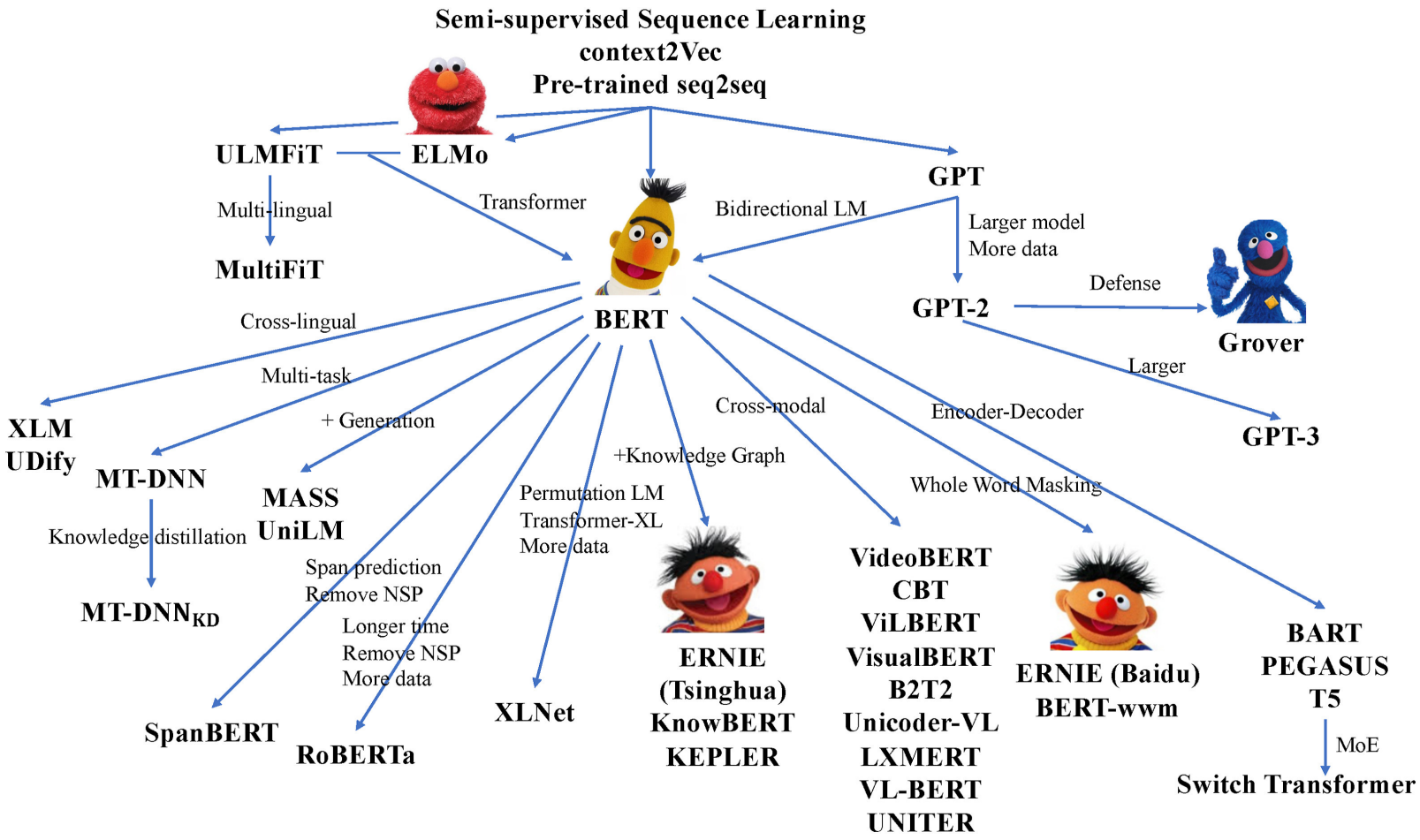
- MMBT

Retrieval-based models

- DPR, RAG

Pre-Trained Language Model (사전학습언어모델)

https://ars.els-cdn.com/content/image/1-s2.0-S2666651021000231-gr8.jpg



거대언어모델(Large Language Model)

사전학습모델(Pre-trained Language Model)의 모델크기나 데이터 크기를 확장하면 downstream task에서 모델 용량이 향상됨

- LLM은 기존 언어모델에서 볼 수 없던 새로운 능력을 보여줌
- LLM은 인간이 AI를 개발하고 사용하는 방식에 변화를 가져옴
  - LLM에 접근하는 방식은 prompt를 통해서 주로 이루어짐
  - 따라서 LLM이 이해할 수 있는 방식으로 지시문(instruction)을 만들어야 함
- chatGPT와 GPT-4의 출현으로 Artificial General Intelligence의 가능성 모색됨

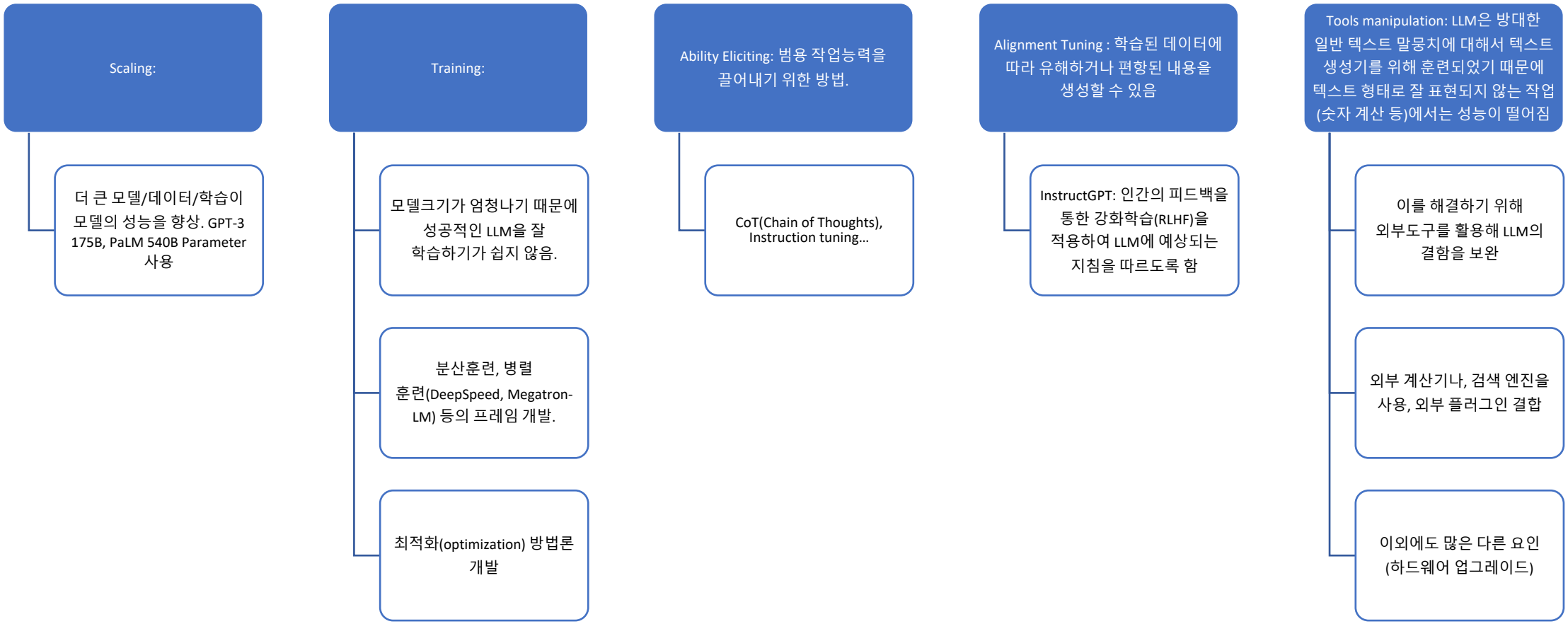
그럼에도 불구하고 LLM의 기본원리는 잘 밝혀지지 않음

- 작은 PLM이 아닌 LLM에서 새로운 기능이 발생하는지 의문
- LLM을 구축하기 위해서는 엄청난 양의 데이터와 고성능의 GPU가 대량으로 필요하기 때문에 훈련 비용이 많이 듬
  - 몇 거대 기업들만 훈련 가능한 상황
  - 구축된 모델 및 사용된 데이터셋의 비공개

LLM의 배경

- 일반적으로 거대언어모델은 트랜스포머(Transformer) 아키텍쳐를 기반으로 수천억 또는 그 이상의 parameter를 포함하는 언어모델을 지칭
  - Scaling: 거대언어모델의 능력향상은 모델의 크기에 따라 성능이 대략적으로 증가하는 scaling 법칙으로 부분적으로 설명가능.
  - Emergent Ability: “작은 모델에는 존재하지 않지만 큰 모델에서 발생하는 능력”. 규모가 일정수준에 도달하면 성능이 무작위보다 크게 상승함
    1. In-Context Learning(ICL): GPT-3에서 처음으로 도입. 언어모델에 자연어 지시 또는 여러 작업의 데모가 제공되었다면, 추가 학습이나 gradient 업데이트 없이 입력 테스트의 순서를 완성하여 예상 출력을 생성함
    2. Instruction-following: 자연어 설명을 통해 포맷된 다중 작업 데이터셋을 혼합하여 미세조정(instruction tuning)하면, LLM은 명령어 형태로 설명되는 보이지 않는 작업에서도 잘 수행됨
    3. Step-by-step Reasoning: 수학 연산, 논리적 추론 등을 위해 사고의 연쇄(Chain-of-Thoughts)를 사용하여, 최종 답을 도출하기 위한 중간 추론 단계가 포함된prompt를 활용하여 적용

LLM의 주요 기술(Key Technics)



List of Large Langauge Models (https://en.wikipedia.org/wiki/Large_language_model)

Name	Release date[a]	Developer	Number of parameters[b]	Corpus size	License[c]	Notes
BERT	2018	Google	340 million[55]	3.3 billion words[55]	Apache 2.0[56]	An early and influential language model,[2] but encoder-only and thus not built to be prompted or generative[57]
XLNet	2019	Google	~340 million[58]	33 billion words		An alternative to BERT; designed as encoder-only[59][60]
GPT-2	2019	OpenAI	1.5 billion[61]	40GB[62] (~10 billion tokens)[63]	MIT[64]	general-purpose model based on transformer architecture
GPT-3	2020	OpenAI	175 billion[24]	300 billion tokens[63]	public web API	A fine-tuned variant of GPT-3, termed GPT-3.5, was made available to the public through a web interface called ChatGPT in 2022.[65]
GPT-Neo	March 2021	EleutherAI	2.7 billion[66]	825 GiB[67]	MIT[68]	The first of a series of free GPT-3 alternatives released by EleutherAI. GPT-Neo outperformed an equivalent-size GPT-3 model on some benchmarks, but was significantly worse than the largest GPT-3.[68]
GPT-J	June 2021	EleutherAI	6 billion[69]	825 GiB[67]	Apache 2.0	GPT-3-style language model
Megatron-Turing NLG	October 2021[70]	Microsoft and Nvidia	530 billion[71]	338.6 billion tokens[71]	Restricted web access	Standard architecture but trained on a supercomputing cluster.
Ernie 3.0 Titan	December 2021	Baidu	260 billion[72]	4 Tb	Proprietary	Chinese-language LLM. Ernie Bot is based on this model.
Claude[73]	December 2021	Anthropic	52 billion[74]	400 billion tokens[74]	Closed beta	Fine-tuned for desirable behavior in conversations.[75]
GLaM (Generalist Language Model)	December 2021	Google	1.2 trillion[76]	1.6 trillion tokens[76]	Proprietary	Sparse mixture-of-experts model, making it more expensive to train but cheaper to run inference compared to GPT-3.
Gopher	December 2021	DeepMind	280 billion[77]	300 billion tokens[78]	Proprietary	
LaMDA (Language Models for Dialog Applications)	January 2022	Google	137 billion[79]	1.56T words,[79] 168 billion tokens[78]	Proprietary	Specialized for response generation in conversations.
GPT-NeoX	February 2022	EleutherAI	20 billion[80]	825 GiB[67]	Apache 2.0	based on the Megatron architecture
Chinchilla	March 2022	DeepMind	70 billion[81]	1.4 trillion tokens[81][78]	Proprietary	Reduced-parameter model trained on more data. Used in the Sparrow bot.
PaLM (Pathways Language Model)	April 2022	Google	540 billion[82]	768 billion tokens[81]	Proprietary	aimed to reach the practical limits of model scale
OPT (Open Pretrained Transformer)	May 2022	Meta	175 billion[83]	180 billion tokens[84]	Non-commercial research[d]	GPT-3 architecture with some adaptations from Megatron
YaLM 100B	June 2022	Yandex	100 billion[85]	1.7TB[85]	Apache 2.0	English-Russian model based on Microsoft's Megatron-LM.

List of Large Language Models

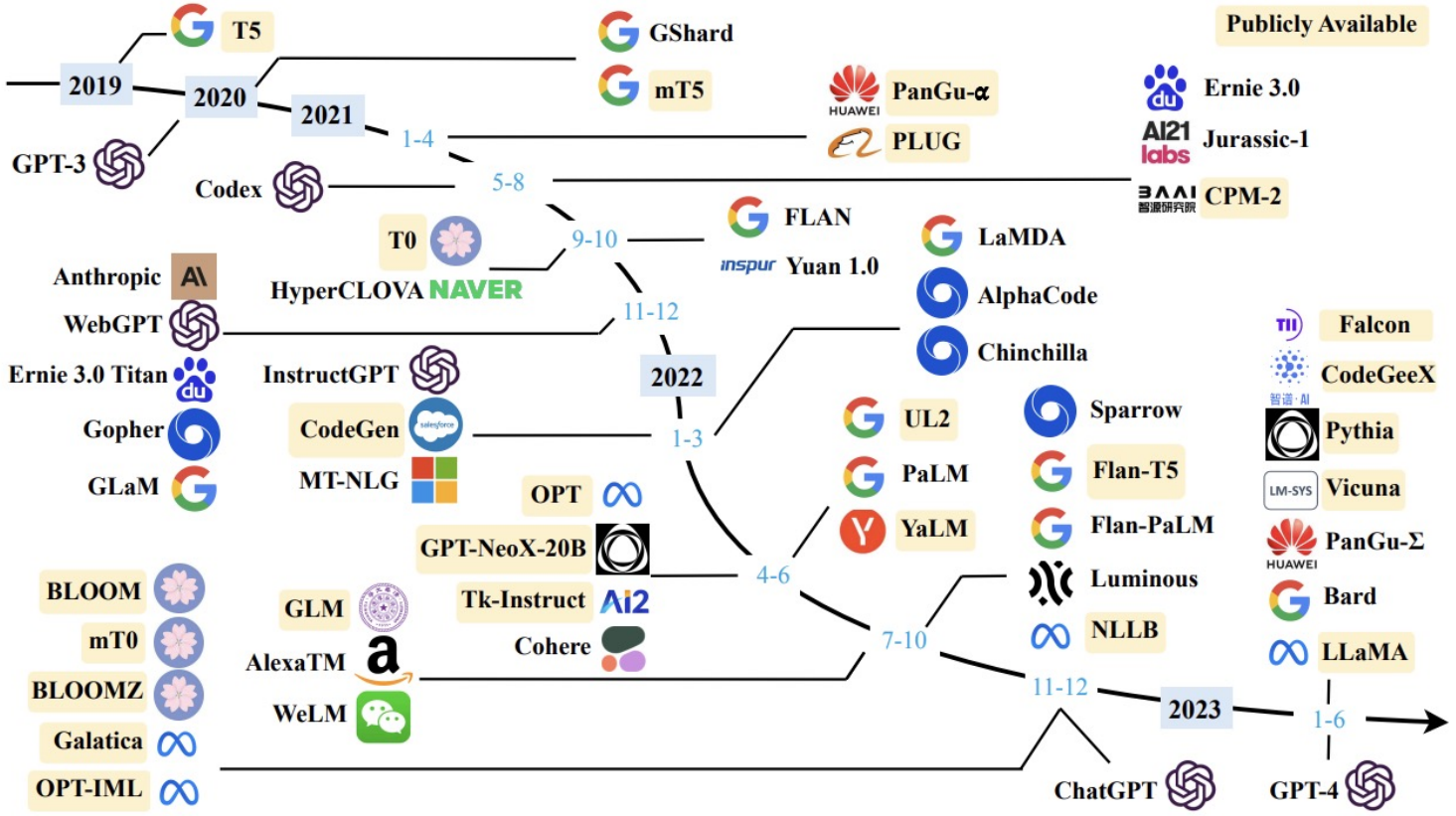
Minerva	June 2022	Google	540 billion[86]	38.5B tokens from webpages filtered for mathematical content and from papers submitted to the arXiv preprint server[86]	Proprietary	LLM trained for solving "mathematical and scientific questions using step-by-step reasoning".[87] Minerva is based on PaLM model, further trained on mathematical and scientific data.
BLOOM	July 2022	Large collaboration led by Hugging Face	175 billion[88]	350 billion tokens (1.6TB)[89]	Responsible AI	Essentially GPT-3 but trained on a multi-lingual corpus (30% English excluding programming languages)
Galactica	November 2022	Meta	120 billion	106 billion tokens[90]	CC-BY-NC-4.0	Trained on scientific text and modalities.
AlexaTM (Teacher Models)	November 2022	Amazon	20 billion[91]	1.3 trillion[92]	public web API[93]	bidirectional sequence-to-sequence architecture
LLaMA (Large Language Model Meta AI)	February 2023	Meta	65 billion[94]	1.4 trillion[94]	Non-commercial research[e]	Trained on a large 20-language corpus to aim for better performance with fewer parameters.[94] Researchers from Stanford University trained a fine-tuned model based on LLaMA weights, called Alpaca.[95]
GPT-4	March 2023	OpenAI	Exact number unknown, approximately 1 trillion [f]	Unknown	public web API	Available for ChatGPT Plus users and used in several products.
Cerebras-GPT	March 2023	Cerebras	13 billion[97]		Apache 2.0	Trained with Chinchilla formula.
Falcon	March 2023	Technology Innovation Institute	40 billion[98]	1 Trillion tokens (1TB)[98]	Apache 2.0[99]	The model is claimed to use only 75% of GPT-3's training compute, 40% of Chinchilla's, and 80% of PaLM-62B's.
BloombergGPT	March 2023	Bloomberg L.P.	50 billion	363 billion token dataset based on Bloomberg's data sources, plus 345 billion tokens from general purpose datasets[100]	Proprietary	LLM trained on financial data from proprietary sources, that "outperforms existing models on financial tasks by significant margins without sacrificing performance on general LLM benchmarks"
PanGu-Σ	March 2023	Huawei	1.085 trillion	329 billion tokens[101]	Proprietary	
OpenAssistant[102]	March 2023	LAION	17 billion	1.5 trillion tokens	Apache 2.0	Trained on crowdsourced open data
PaLM 2 (Pathways Language Model 2)	May 2023	Google	340 billion[103]	3.6 trillion tokens[103]	Proprietary	Used in Bard chatbot.[104]

상업적으로 사용 가능한 공개 언어 모델

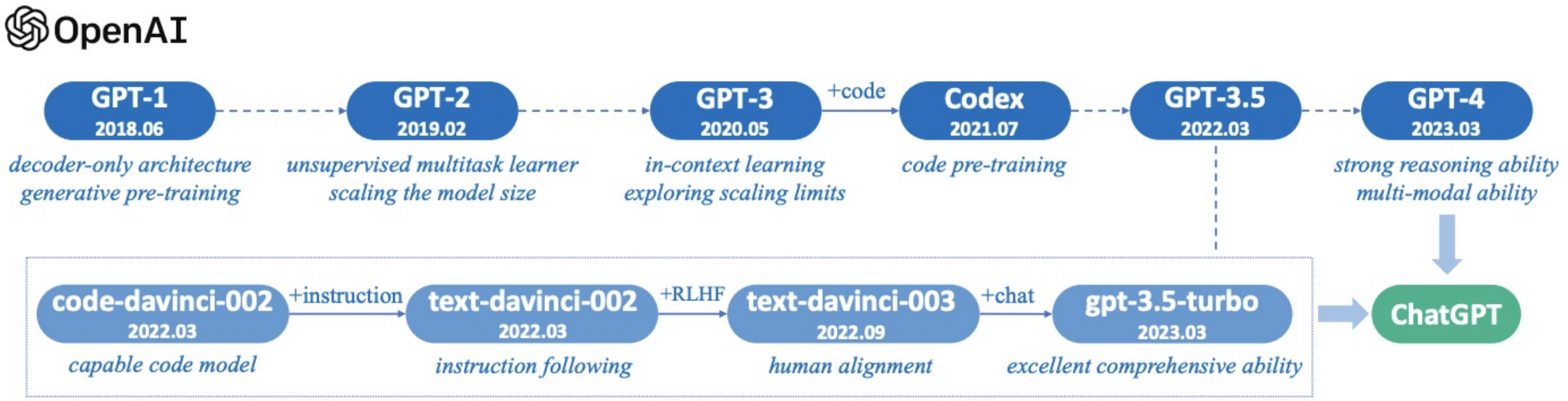
	License	Data	Architecture	Weights	Size	Checkpoints	Language
Meta Llama2	Llama license	Open	Open	Open	7, 13, 70	Yes	English / Multilingual
EleutherAI Pythia	Apache 2.0	Open	Open	Open	7, 12	Yes	English
EleutherAI Polyglot	GPL-2.0	Open	Open	Open		Yes	English / Multilingual
GPT-J	MIT	Open	Open	Open	6	Yes	English
Databricks Dolly 2	Apache 2.0	Open	Open	Open	7, 12	Yes	English
Cerebras-GPT	Apache 2.0	Open	Open	Open	7, 13	Yes	English / Multilingual
StableLM	CC BY-SA-4.0	Open	Open	Open	3, 7, (15, 30, 65, 175)	Yes	English
Mosaic MPT	Apache 2.0	Open	Open	Open	7, 30	Yes	English
Falcon GPT	Apache 2.0	Open	Open	Open	7, 40	Yes	English

Technical Evolution of GPT-series Models

A timeline of Existing Large Language Models(Zhao et al. (2023))

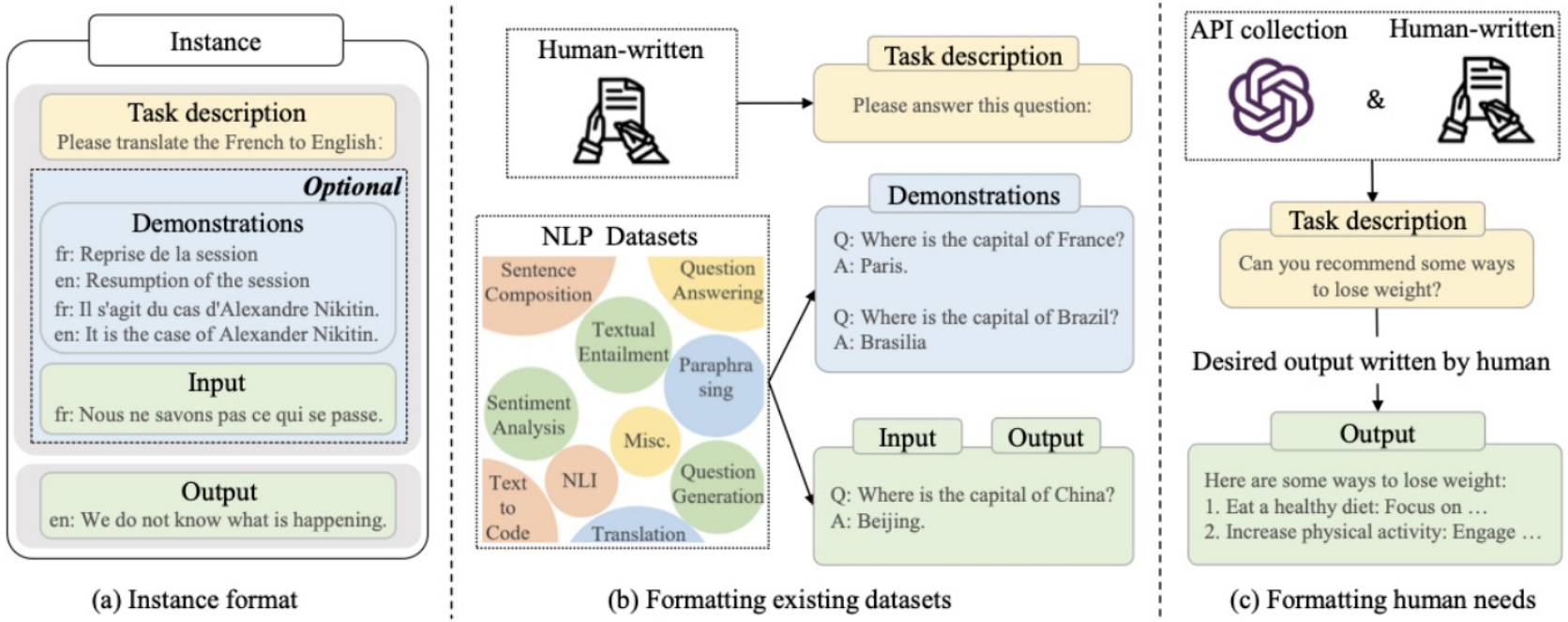


Technical Evolution of GPT-series Models

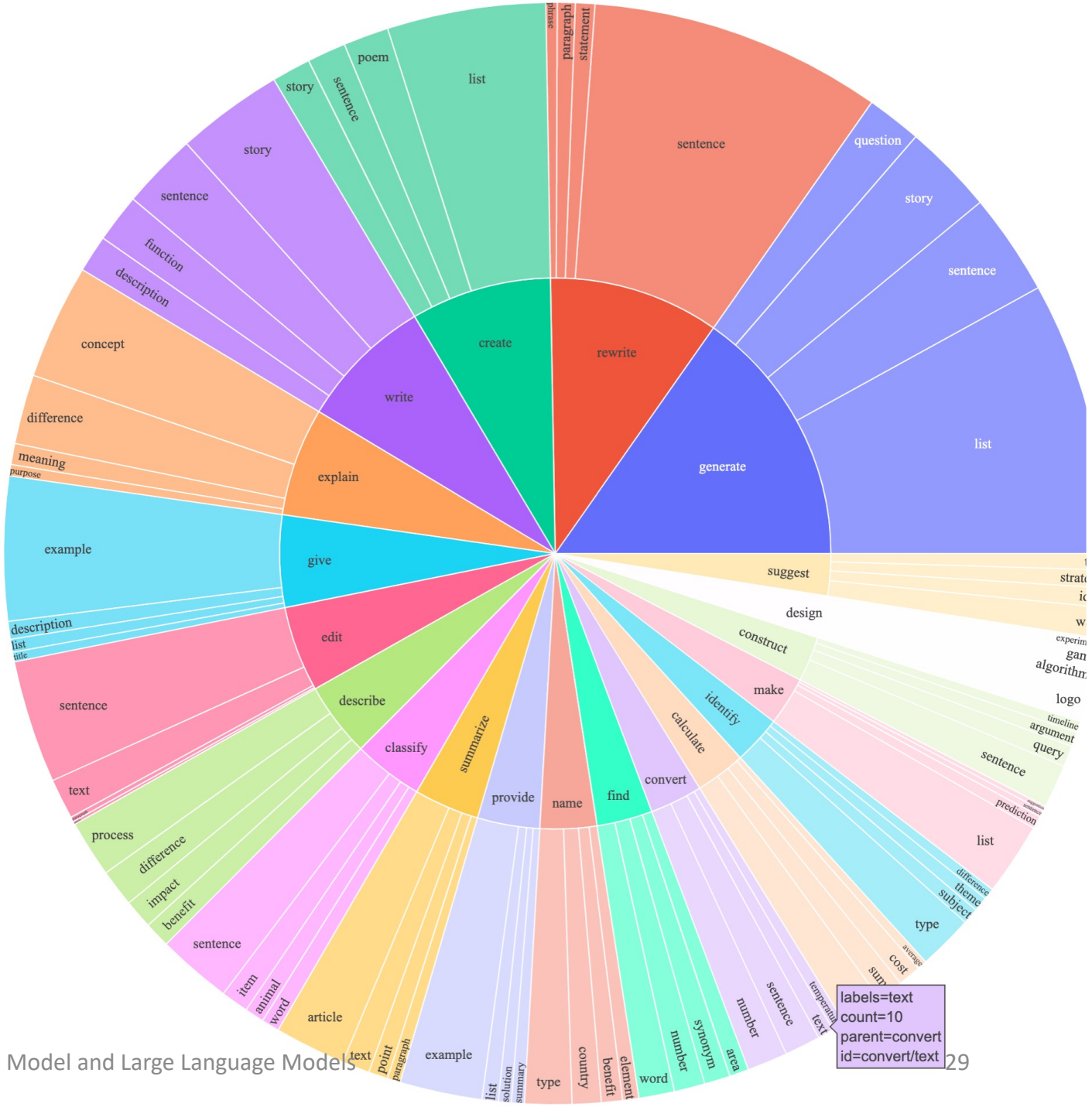


Adaptations of LLMs: Instruction Tuning

- 사전 훈련 후 LLM은 다양한 작업을 위해 일반적인 능력을 학습
  - Instruction Tuning: LLM의 능력을 향상시키는 것을 목표
  - Alignment Tuning: LLM의 행동을 인간의 가치 또는 선호도에 맞추는 것



Instruction-Following Model by Alpaca



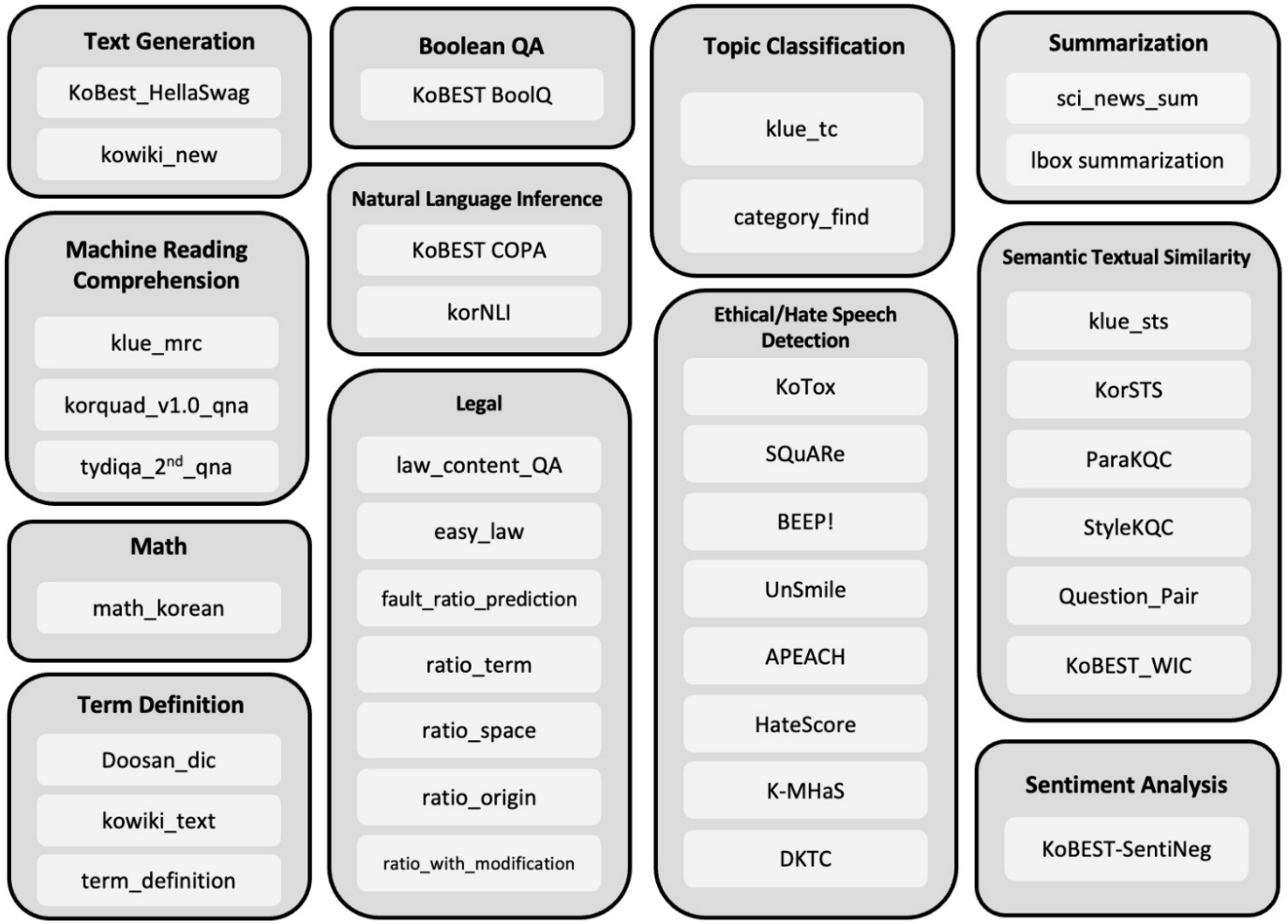
Instruction-Following (Response) Model Dataset (Alpaca)

```
...
{
    "instruction": "Describe a time when you had to make a difficult decision.",
    "input": "",
    "output": "I had to make a difficult decision when I was working as a project manager at a const
},
{
    "instruction": "Identify the odd one out.",
    "input": "Twitter, Instagram, Telegram",
    "output": "Telegram"
},
...
```

Instruction-Following (Response) Model Dataset (GPT4ALL)


```
{
    "id": "user_oriented_task_235",
    "motivation_app": "Yelp",
    "instruction": "전문 분야에 따라 레스토랑, 홈 서비스, 자동차 서비스, 기타 중 하나로 비즈니스를 분류합니다.",
    "instances": [
        {
            "input": "견적을 받으려면 650-636-4884로 전화하거나 웹사이트를 방문하세요. 이 매장은 신품 타이어 및 일반 자동
            "output": "Auto Services"
        }
    ]
},
```

Instruction DataSets in DaG



Key Factors for Instance Construction

Scaling the instructions

- 일반적으로 LLM의 task 수를 확장하면 일반화 능력을 크게 확장할 수 있다고 알려져 있으나 과제를 더 추가해도 추가적인 이득을 얻지 못할 수도 있음
- 길이, 구조, 창의성 등 여러 면에서 task 설명의 다양성을 높이는 것이 중요
- Task 당 instance의 수는 일반적으로 적은 수의 instance가 모델의 일반화 성능을 충족시키는 것으로 밝혀짐

Formatting design

- 입력문에 task description과 optional demonstration을 추가할 수 있는데, task description은 거대언어모델이 그 task를 이해하는데 가장 중요한 부분
- 적절한 수의 예제를 데모로 사용하면 성능의 개선이 이루어질 수 있음

Instance의 수보다는 Instruction의 다양성이 더 중요

특정 데이터에 한정된 task 보다는 인간이 필요로 하는 task를 레이블하는 것이 더 유용함

Improvement Strategies

Enhancing the instruction complexity

- 더 많은 taks 요구나 더 많은 추론 단계로 이루어진 복잡한 instruction을 구성하여 LLM의 모델의 성능을 향상.

Increasing the topic diversity

- 다양한 주제로 된 instance 데이터셋 구축이 모델의 성능을 향상
- 그러나 다양한 주제를 self-instruct 방법으로 구축하기는 쉽지 않음.

Scaling the instruction number

- Instruction의 수가 모델의 성능에 영향을 미치고 더 많은 instruction을 사용하면 task의 지식을 확장하게 되고 거대언어모델의 instruction의 능력을 증대시키게 됨

Balancing the instruction Difficulty

- Synthetic하게 instruction을 만들 경우 너무 쉽거나 어려운 instruction이 생성될 수 있고 이 경우 학습이 불안정하거나 LLM에 과적합이 생기기 쉬움
- LLM의 perplexity 점수를 사용하여 너무 쉽거나 어려운 instruction을 제거

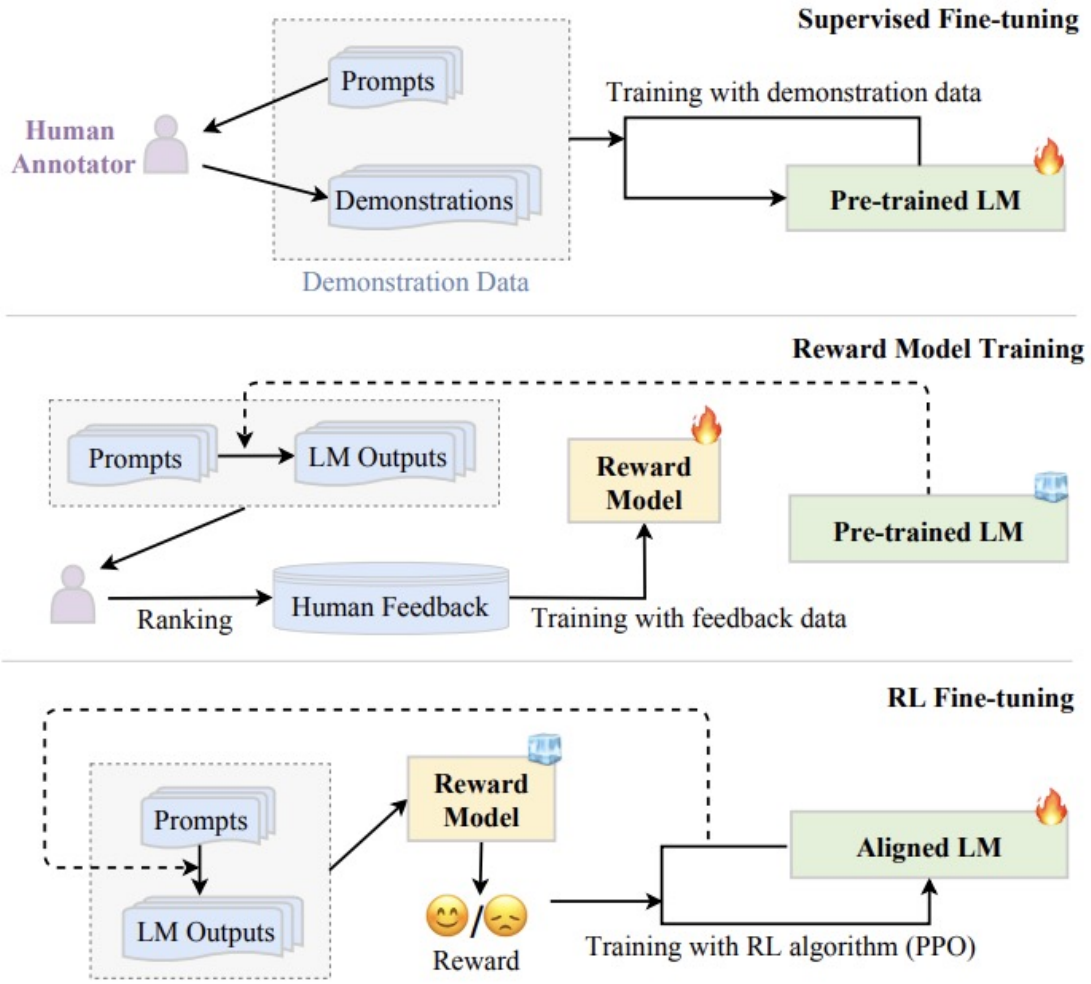
Results of Instruction-tuning experiment

TABLE 8: Results of instruction-tuning experiments (all in a single-turn conversation) based on the LLaMA (7B) and LLaMA (13B) model under the chat and QA setting. We employ four instruction improvement strategies on the Self-Instruct-52K dataset, *i.e.*, enhancing the complexity (*w/ complexity*), increasing the diversity (*w/ diversity*), balancing the difficulty (*w/ difficulty*), and scaling the instruction number (*w/ scaling*). *Since we select the LLaMA (7B)/(13B) model fine-tuned on Self-Instruct-52K as the baseline, we omit the win rate of the fine-tuned model with Self-Instruct-52K against itself.

Models	Dataset Mixtures	Instruction Numbers	Lexical Diversity	Chat	QA	
				AlpacaFarm	MMLU	BBH3k
LLaMA (7B)	① FLAN-T5	80,000	48.48	23.77	38.58	32.79
	② ShareGPT	63,184	77.31	81.30	38.11	27.71
	③ Self-Instruct-52K	82,439	25.92	/*	37.52	29.81
	② + ③	145,623	48.22	71.36	41.26	28.36
	① + ② + ③	225,623	48.28	70.00	43.69	29.69
	③ Self-Instruct-52K	82,439	25.92	/*	37.52	29.81
	w/ complexity	70,000	70.43	76.96	39.73	33.25
	w/ diversity	70,000	75.59	81.55	38.01	30.03
	w/ difficulty	70,000	73.48	79.15	32.55	31.25
	w/ scaling	220,000	57.78	51.13	33.81	26.63
LLaMA (13B)	① FLAN-T5	80,000	48.48	22.12	34.12	34.05
	② ShareGPT	63,184	77.31	77.13	47.49	33.82
	③ Self-Instruct-52K	82,439	25.92	/*	36.73	25.43
	② + ③	145,623	48.22	72.85	41.16	29.49
	① + ② + ③	225,623	48.28	69.49	43.50	31.16
	③ Self-Instruct-52K	82,439	25.92	/*	36.73	25.43
	w/ complexity	70,000	70.43	77.94	46.89	35.75
	w/ diversity	70,000	75.59	78.92	44.97	36.40
	w/ difficulty	70,000	73.48	80.45	43.15	34.59
	w/ scaling	220,000	57.78	58.12	38.07	27.28

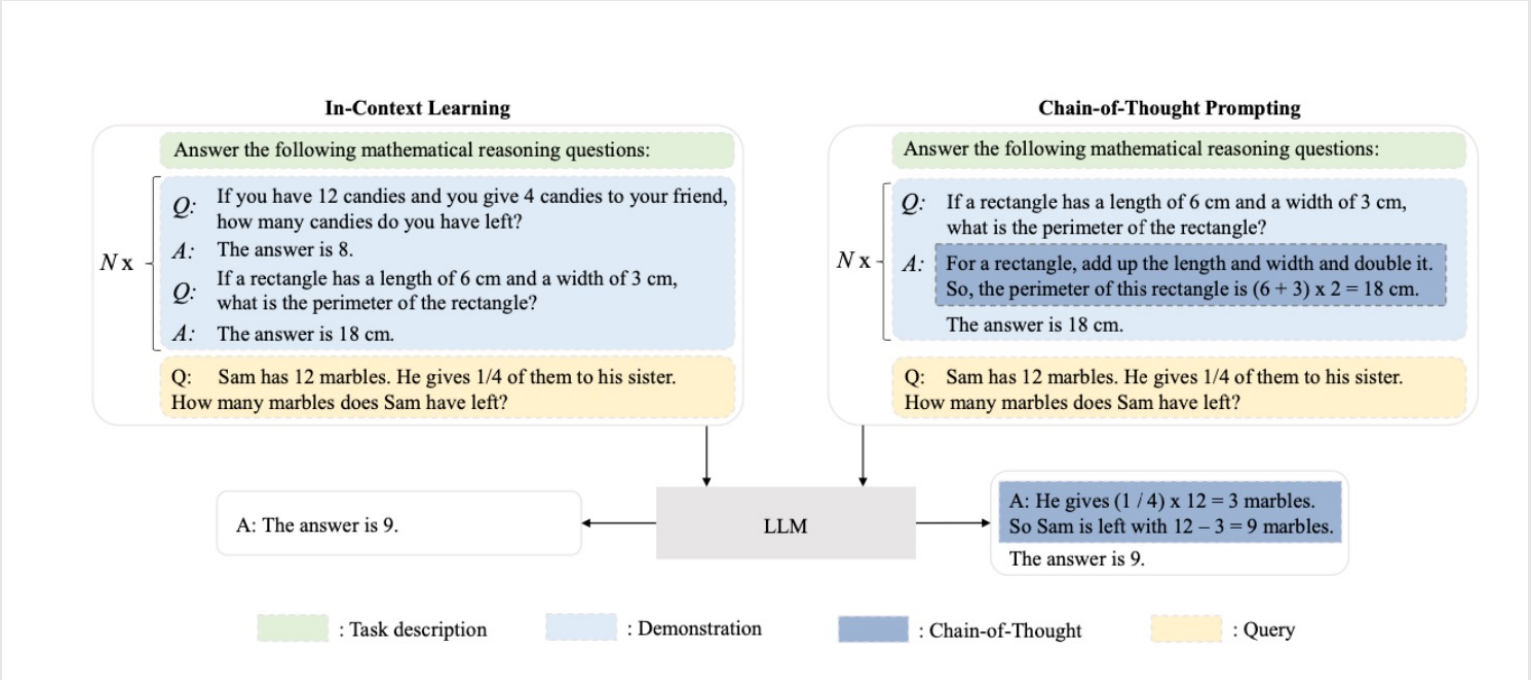
Adaptations of LLMs: Alignment Tuning

- LLM이 인간의 기대에 따라 행동하도록 tuning
  - 유용성: LLM은 가능한 간결하고 효율적인 방식으로 사용자의 과제 해결이나 질문에 대한 답변을 지원하려는 시도를 명확히 보여줘야 함
  - 정직성: LLM은 조작된 정보 대신 정확한 콘텐츠를 사용자에게 제공해야 함
  - 무해성: 모델이 생성하는 페이지가 불쾌감을 주거나 차별적이지 않아야 함
- 사람의 피드백을 통한 강화학습(Reinforcement Larningn from Human Feedback)



Utilization

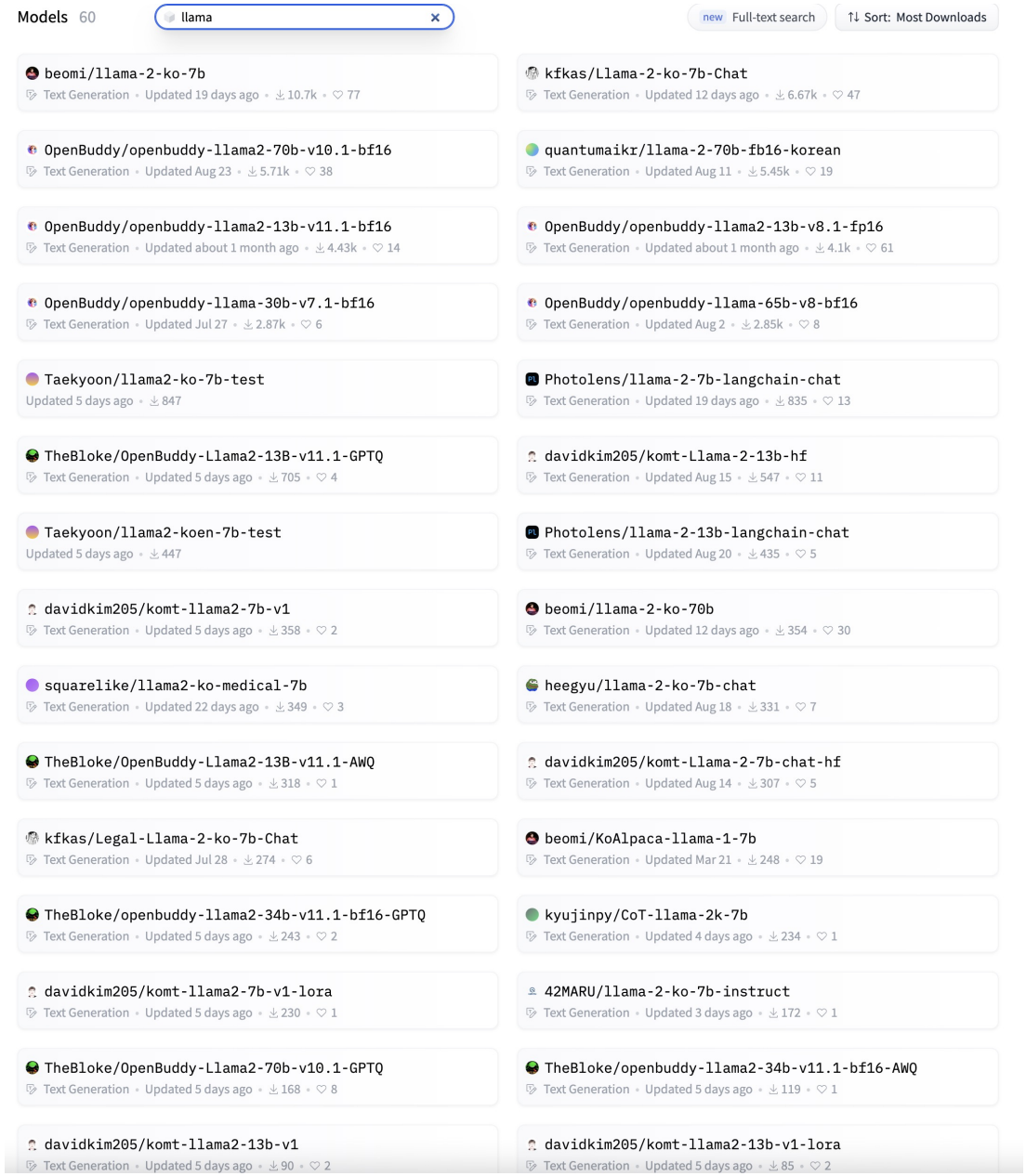
- Prompting :사전 훈련이나 적응 튜닝 후 LLM을 사용하기 위한 중요한 접근 방법
  - Task에 따른 prompt는 직접 만들거나 또는 자동으로 최적화할 수 있음
  - 상황내학습(in-context learning): 자연어 텍스트 형태로 과제 설명 및 또는 데모를 공식화하는 prompting
  - 연쇄적 사고 prompt(chain-of-thought Prompt): 일련의 중간 추론 단계를 prompt에 포함시켜 상황내 학습을 강화하기 위함



한국어 거대언어모델 연구 동향

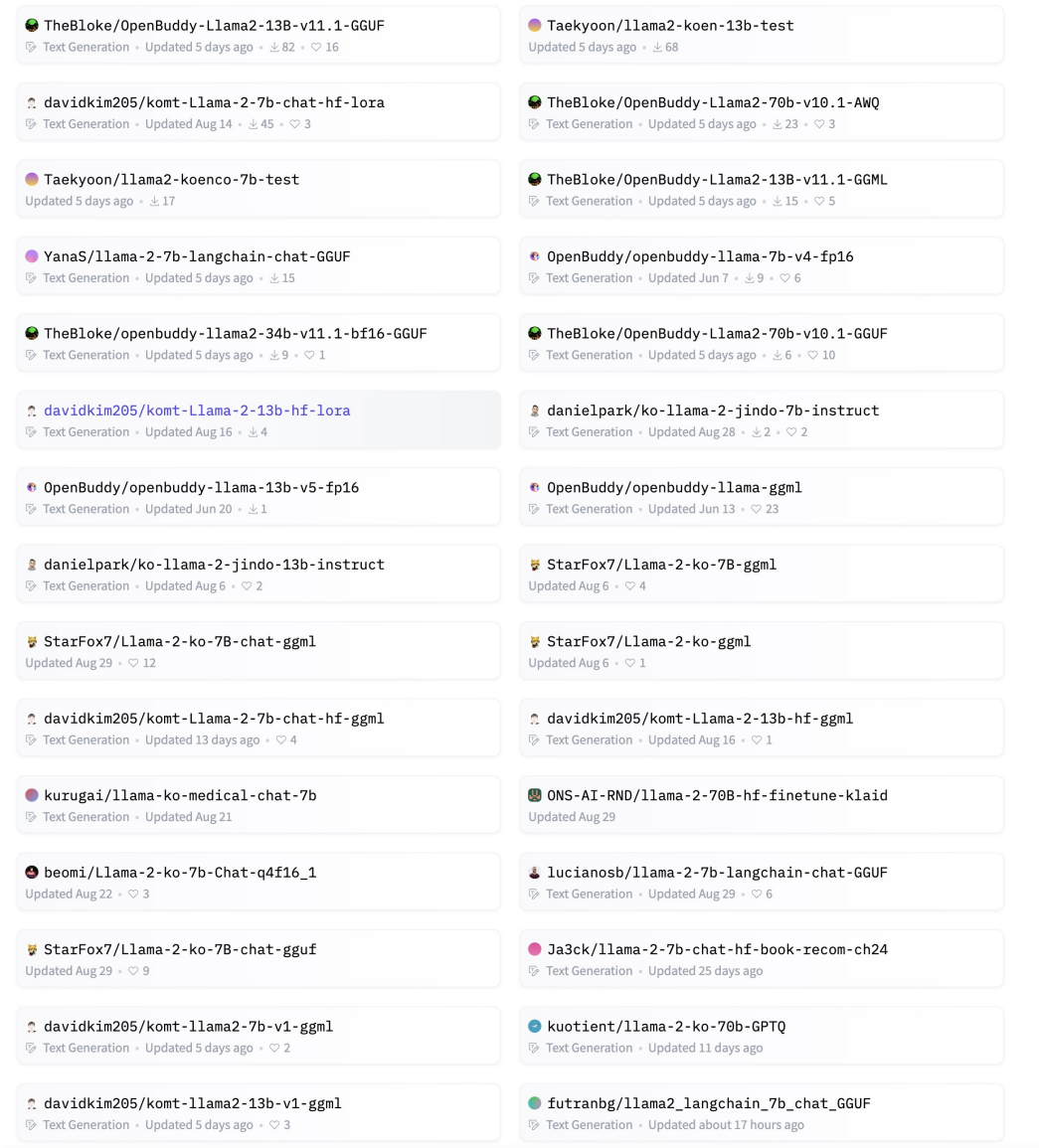
모델	특징
Naver : HyperClover/HyperCloverX	●한국어 560B 토큰으로 82B 패러미터의 **GPT-3 모델 학습** ●chatGPT 식의 모델 공개 (HyperCloverX)
KaKaoBrain: KoGPT **(https://huggingface.co/kakaobrain/kogpt)**	●한국어 **2000억 토큰** 규모의 한국어 데이터셋으로 학습 ●텍스트를 분류, 검색, 요약 생성하는데 적합한 모델 ●욕설, 음란, 정치적 내용에 대한 처리를 하지 않고 학습 ●학습데이터에 등장하지 않는 방언, 한국어가 아닌 경우에 성능이 좋지 않을 수 있음
EleutherAI: Polyglot-ko **(https://huggingface.co/EleutherAI/polyglot-ko-1.3b)**	●TuNiB Ai에서 수집한 1.2TB 규모의 한국어 데이터로 학습 ●오픈자료로 현재 **한국어 instruction tuning에 많이 활용**되고 있음
KoAlpaca **(https://github.com/Beomi/KoAlpaca)**	●**Stanford Alpaca 모델을 한국어로 구현** ●백본 모델로 한국어 모델은 Polyglot-ko(5.8B) 모델을, 영문+한국어 기반 모델은 LLaMA를 사용 ●Stanford Alpaca에서 사용된 instruction-following 셋을 번역기로 돌려 그대로 사용하여 잘못된 번역, 잘못된 표현 등이 존재 ●**RLHF 등의 강화학습 방법을 적용하지 않음**
고려대학교:KULLM **(https://github.com/nlpai-lab/KULLM)**	●백본 모델로 Polyglot-ko를 사용 ●Stanford Alpaca뿐만 아니라 **GPT4ALL, Vicuna,** 그리고 **Dolly**의 데이터를 합하여 대략 15만여 개의 instruction-following 데이터 셋 사용 ●KoAlpaca와 마찬가지로 **영어의 데이터 셋을 번역기를 통해 번역한 결과를 그대로 사용** ●RLHF 등의 강화학습 방법을 적용하지 않음
upstage llama-30b-instruct-2048 (https://huggingface.co/upstage/llama-30b-instruct-2048)	●upstage가 **LLaMA를 30B-parameter로 미세조정한 모델**로 7월 20일자 뉴스기사에 의하면 허깅페이스 오픈 LLM 리더보드에서 메타의 LLaMA2에 이어 2위를 차지했다고 함 ●LLaMA는 한국어 데이터 학습이 적어 한국어의 경우 백본으로 쓰기에는 성능이 좋지 않은 것으로 알려져 있음 ●**한국어 거대모델 미세조정에 잘 기능할지는 불분명함**

한국어 거대언어모델 동향 Huggingface에 탑재된 Model (LLaMA 기반 모델)



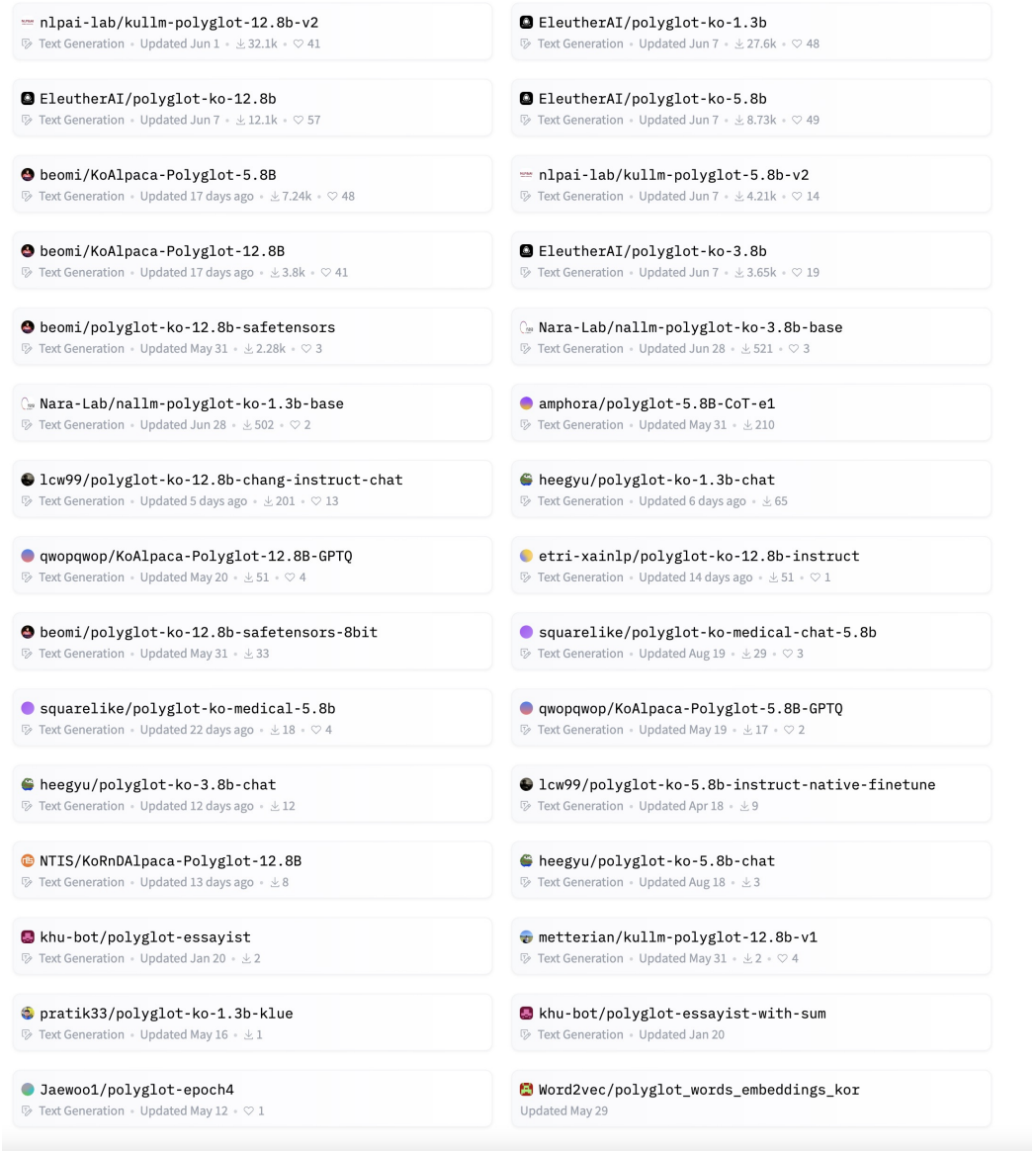
한국어 거대언어모델 동향

Huggingface에 탑재된 Model (LLaMA 기반 모델들)



한국어 거대언어모델 동향

Huggingface에 탑재된 Model (Polyglot-Ko 기반 모델들)



References

- Zhao et al. (2023), A Survey of Large Language Models. https://arxiv.org/pdf/2303.18223.pdf
- Timothy Tan (2020), Evolution of Language Models: N-Grams, Word Embeddings, Attention and Transformers, https://towardsdatascience.com/evolution-of-language-models-n-grams-word-embeddings-attention-transformers-a688151825d2
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.