

# Fine-tuning a clinical domain LLM

2025-08-02, 15:30 ~ 17:30

Jiho Kim, Sujeong Im

KAIST AI @ Edlab (Advised by Edward Choi)

# Speaker Bio

## Jiho Kim (김지호)

### Education

- KAIST Electrical Engineering, B.Sc (2017-2021)
- KAIST Kim Jaechul Graduate School of AI, MS & Ph.D (2021-)

### Research Interests

- Natural Language Processing
- Consistency Check
- Machine Learning for Healthcare

## Sujeong Im (임수정)

### Education

- POSTECH Creative IT Engineering, B.Sc (2018-2022)
- KAIST Kim Jaechul Graduate School of AI, M.Sc (2023-)

### Research Interests

- Foundation Model
- Natural Language Processing
- Machine Learning for Healthcare

# Table of Contents

- How to build a clinical domain Large Language Model (LLM)? (40 mins)
  - (Large) Language Model
  - How to build a (large) language model?
  - Building an instruction-following LLM in the clinical domain
  - Asclepius (Kweon and Kim et al., ACL 2024 Findings)
- Hands-on Session: Fine-tuning a clinical domain LLM (80 mins)
  - Environment Setup & Colab Practice
  - LLM memory layout
  - Parameter-Efficient Fine-Tuning (LoRA/QLoRA)

# Language Model

## Language model

🌐 31 languages ▾

Article [Talk](#)

[Read](#) [Edit](#) [View history](#) [Tools](#) ▾

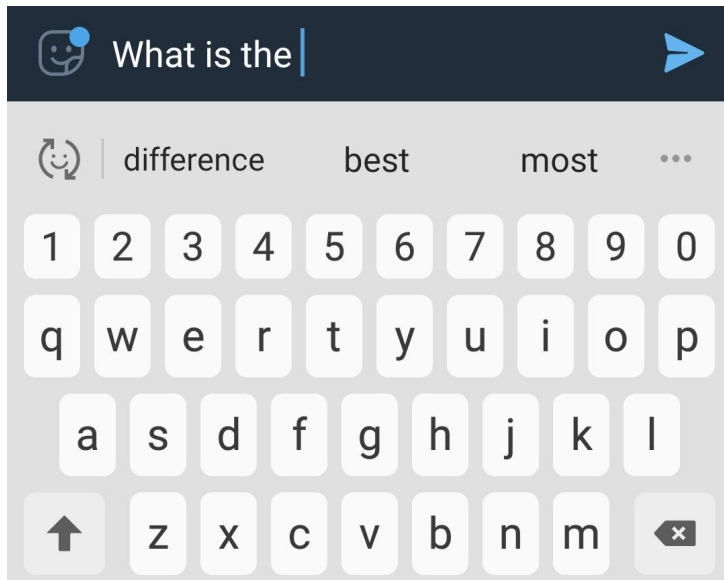
From Wikipedia, the free encyclopedia

A **language model** is a probabilistic [model](#) of a natural language.<sup>[1]</sup> In 1980, the first significant statistical language model was proposed, and during the decade IBM performed ‘[Shannon-style](#)’ experiments, in which potential sources for language modeling improvement were identified by observing and analyzing the performance of human subjects in predicting or correcting text.<sup>[2]</sup>

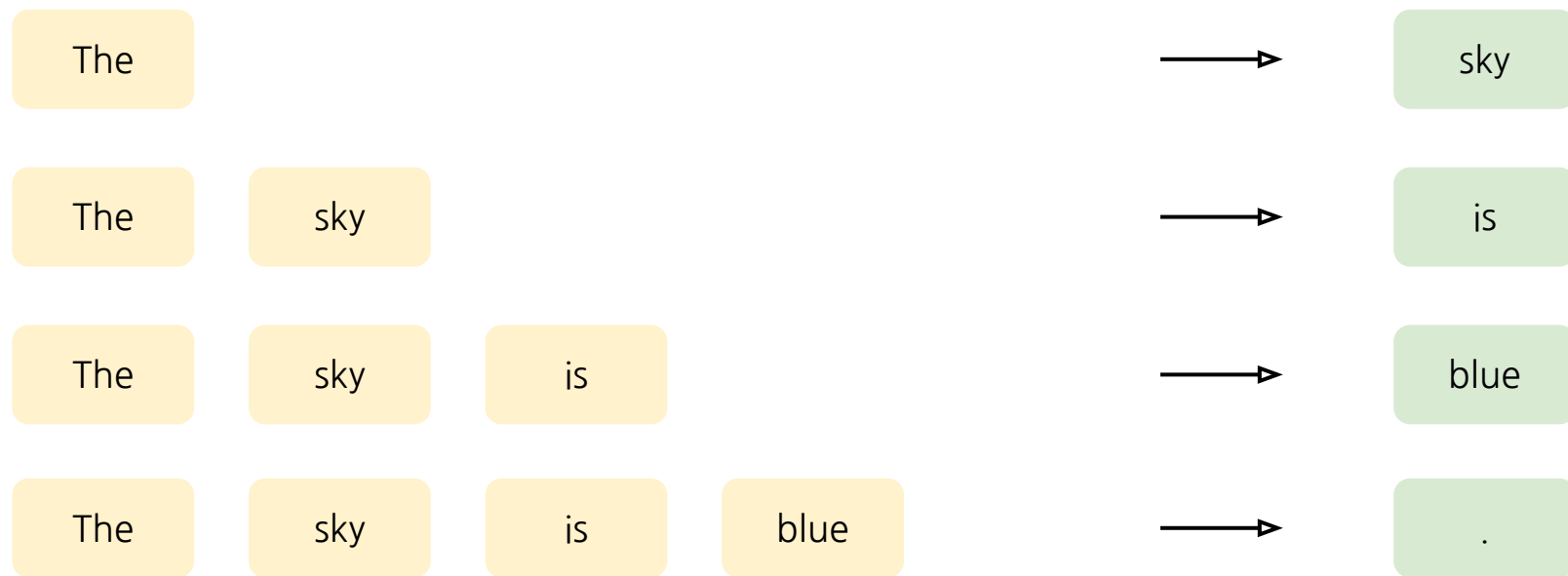
Language models are useful for a variety of tasks, including [speech recognition](#)<sup>[3]</sup> (helping prevent predictions of low-probability (e.g. nonsense) sequences), [machine translation](#),<sup>[4]</sup> [natural language generation](#) (generating more human-like text), [optical character recognition](#), [handwriting recognition](#),<sup>[5]</sup> [grammar induction](#),<sup>[6]</sup> and [information retrieval](#).<sup>[7][8]</sup>

[Large language models](#), currently their most advanced form, are a combination of larger datasets (frequently using words [scraped](#) from the public internet), [feedforward neural networks](#), and [transformers](#). They have superseded [recurrent neural network](#)-based models, which had previously superseded the pure statistical models, such as [word  \$n\$ -gram language model](#).

# We deal with LMs every day!

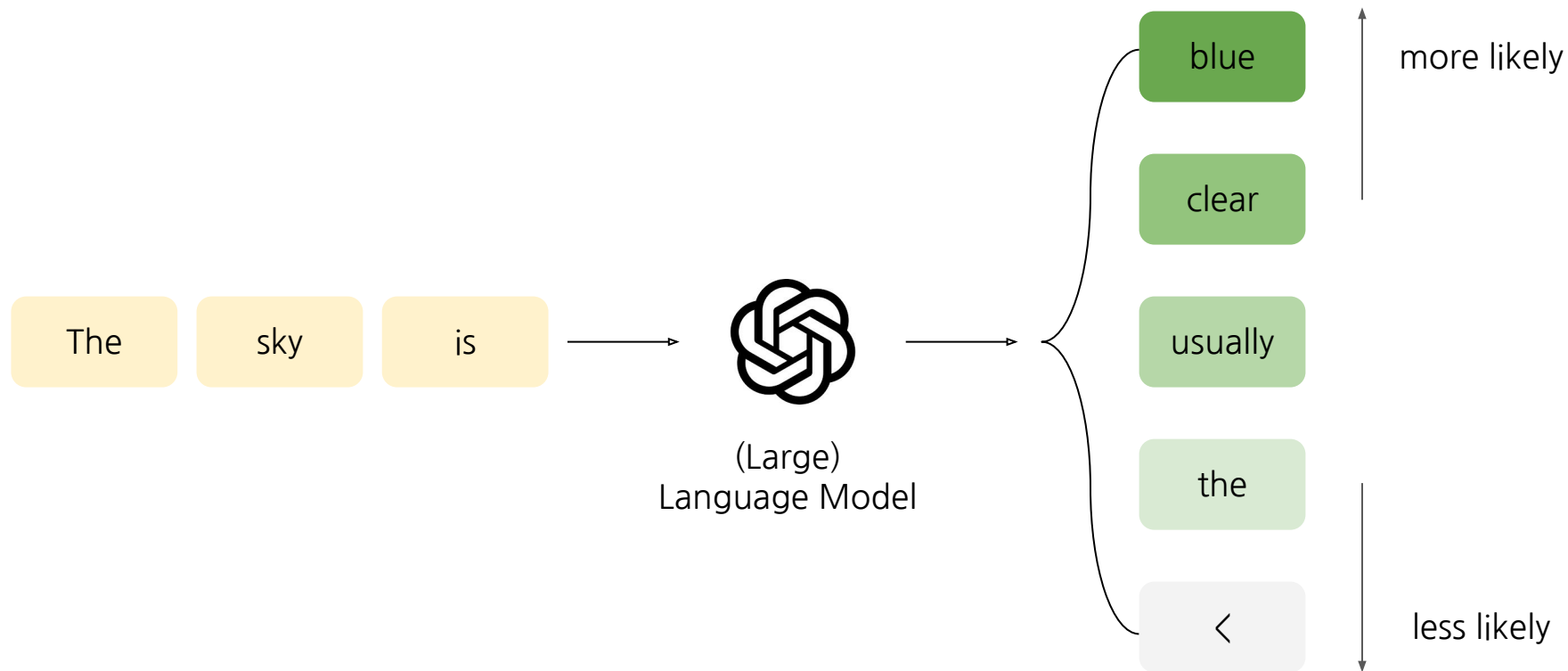


# How to train an LM?



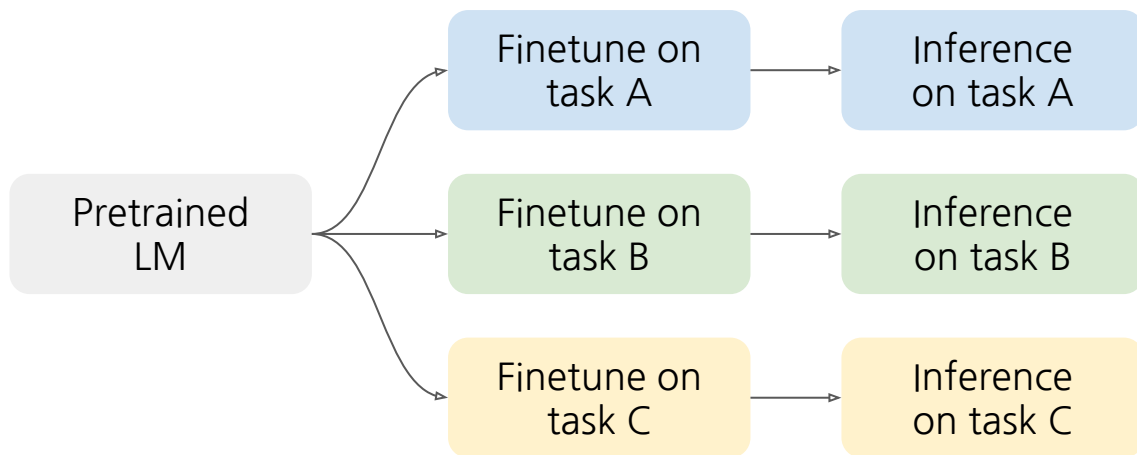
**Next Token Prediction** task for the sentence "The sky is blue."

# Text Generation via a Probabilistic Model



# How to build a (large) language model?

- Pre-training and Fine-tuning
  - e.g., BERT (2018), T5 (2019)

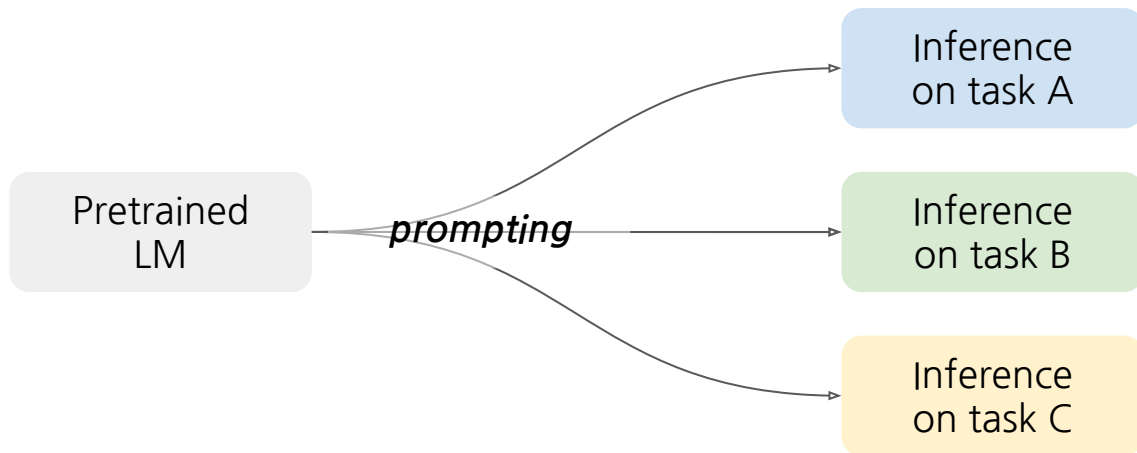


(-) Task-specific training → One specialized model for each task



# How to build a (large) language model?

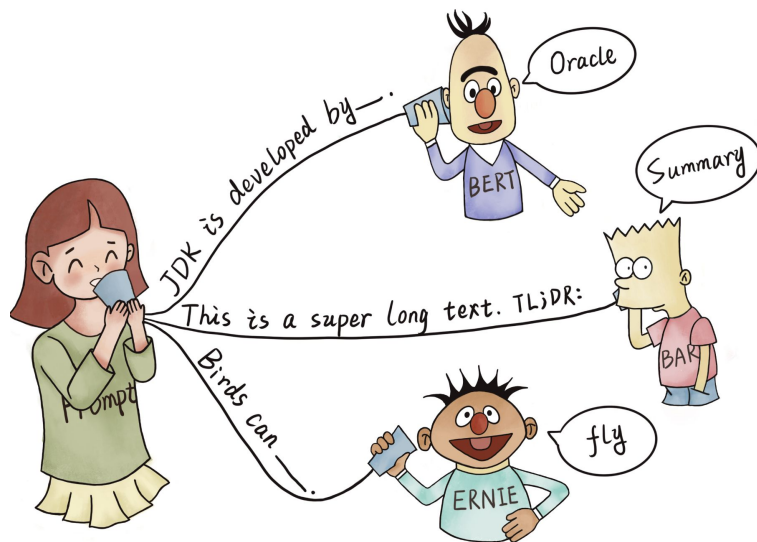
- Pre-training and Prompting
  - e.g., GPT-3 (2020)



(+) Improve performance via few-shot prompting or prompt engineering

# How to build a (large) language model?

- Pre-training and Prompting



## Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

1	Translate English to French:	← task description
2	sea otter => loutre de mer	← examples
3	peppermint => menthe poivrée	
4	plush girafe => girafe peluche	
5	cheese => .....	← prompt

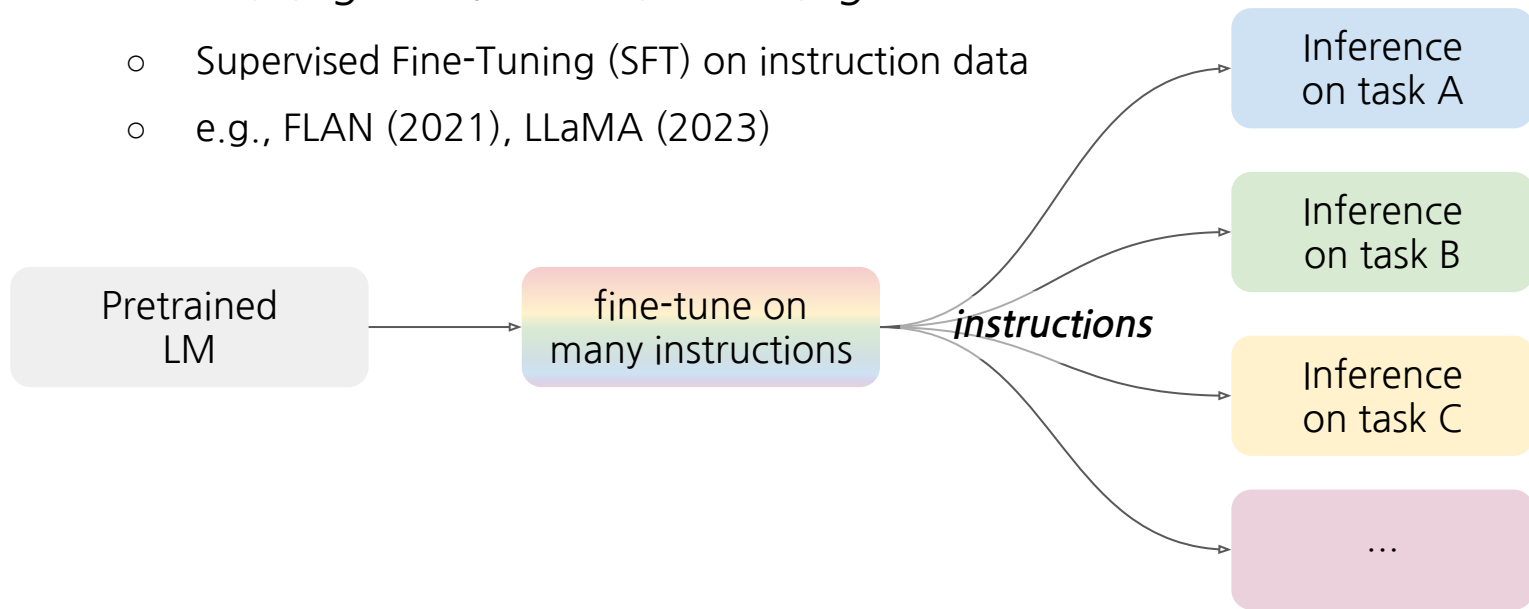
(-) Forced few-shot prompting

(-) Manual efforts for the prompting technique

(-) Not aligned with natural instructions

# How to build a (large) language model?

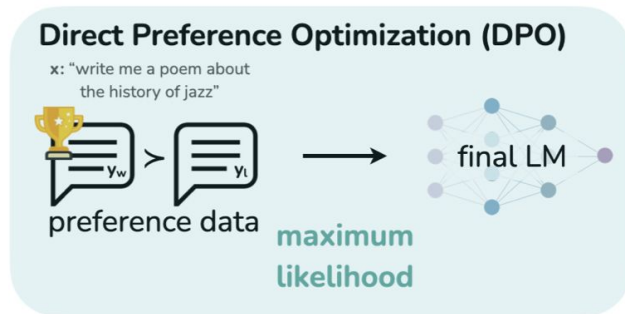
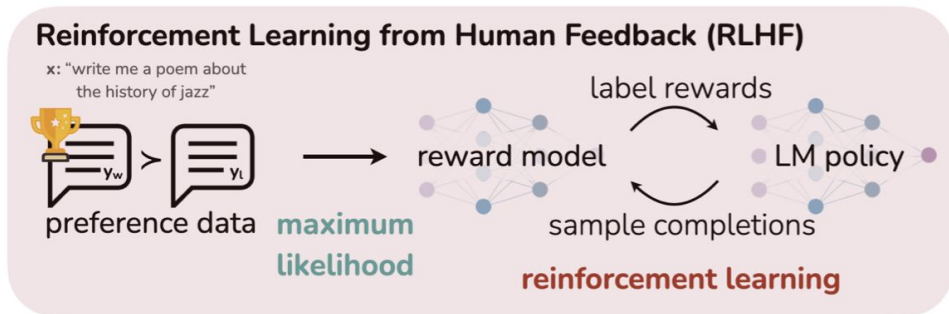
- Pre-training and Instruction tuning
  - Supervised Fine-Tuning (SFT) on instruction data
  - e.g., FLAN (2021), LLaMA (2023)



(+) model learns to perform many tasks via natural language instructions

# How to build a (large) language model?

- Pre-training and Alignment tuning
  - Supervised Fine-Tuning (SFT) on instruction data
    - + Alignment learning on preference data (e.g., RLHF, DPO)
  - e.g., InstructGPT (2022), ChatGPT (2022), Llama 2 (2023), Llama 3 (2024)



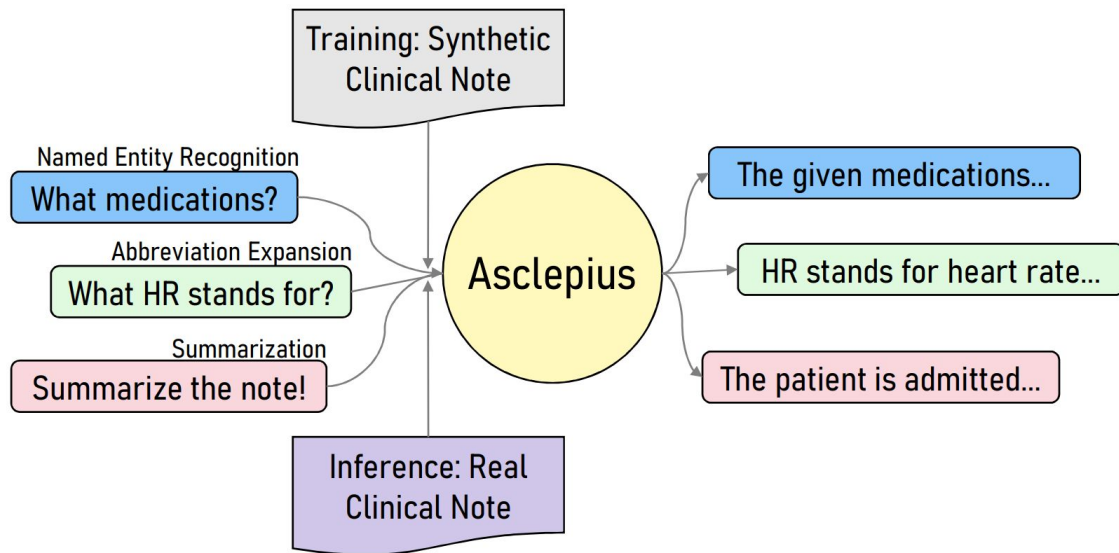
# Building an instruction-following LLM

- How can we build an instruction-following LLM?
  - Prepare a pre-trained large language model (e.g., LLaMA 7B)
  - Perform supervised fine-tuning on instruction data (e.g., Alpaca 52K dataset)
- How can we build an instruction-following LLM in the clinical domain?
  - Prepare a pre-trained large language model
  - Pre-training on clinical corpus for domain adaptation
  - Perform supervised fine-tuning using domain-specific clinical instruction data
    - Today, we will focus on instruction-following data tailored for clinical notes!

# Imagine a clinical LLM

- Given a clinical note, a clinical LLM can perform these tasks as follows:
  - “What medical procedures were performed on the patient during her hospital course, as mentioned in the discharge summary?” **Named Entity Recognition**
  - “What abbreviation was expanded using the acronym ‘ANH’ in the diagnosis section of the discharge summary?” **Abbreviation Expansion**
  - “When was the patient started on oral acyclovir and what was the duration of treatment?” **Temporal Information Extraction**
  - “Can you summarize the patient’s hospital course, treatment, and diagnoses according to the given discharge summary?” **Summarization**
  - “What was the reason for the patient’s transfer to ICU and what was the treatment plan for infection-induced respiratory failure?” **Question Answering**

# Asclepius: Publicly Shareable Clinical Large Language Model Built on Synthetic Clinical Notes (Kweon and Kim et al., ACL 2024 Findings)



# Real clinical note

Admission Date: [\*\*2118-8-10\*\*]  
Discharge Date: [\*\*2118-8-12\*\*]  
Date of Birth: [\*\*2073-12-25\*\*]  
Sex: F  
...  
Discharge Diagnosis:  
AVM  
Radionecrosis  
...  
Discharge Instructions:  
- DISCHARGE INSTRUCTIONS  
FOR CRANIOTOMY/HEAD INJ  
URY  
- Have a family member check y  
our incision daily for signs of inf  
ection  
- Take your pain medicine as pre  
scribed

Real Clinical Note

- Semi-Structured Text about Patient Activity
- Properties
  - Semi-structured: Associated with headers
  - Acronyms
  - Typos
- Problem: Protected Health Information (PHI)
  - Use GPT: PHI  $\Rightarrow$  Impractical
  - Human Annotation: Require Experts  $\Rightarrow$  cost
  - Machine Annotation: PHI  $\Rightarrow$  Impractical



# Case report

Admission Date: [\*\*2118-8-10\*\*]  
Discharge Date: [\*\*2118-8-12\*\*]  
Date of Birth: [\*\*2073-12-25\*\*]  
Sex: F  
...  
Discharge Diagnosis:  
AVM  
Radionecrosis  
...  
Discharge Instructions:  
- DISCHARGE INSTRUCTIONS  
FOR CRANIOTOMY/HEAD INJURY  
- Have a family member check y  
our incision daily for signs of inf  
ection  
- Take your pain medicine as pre  
scribed

Real Clinical Note

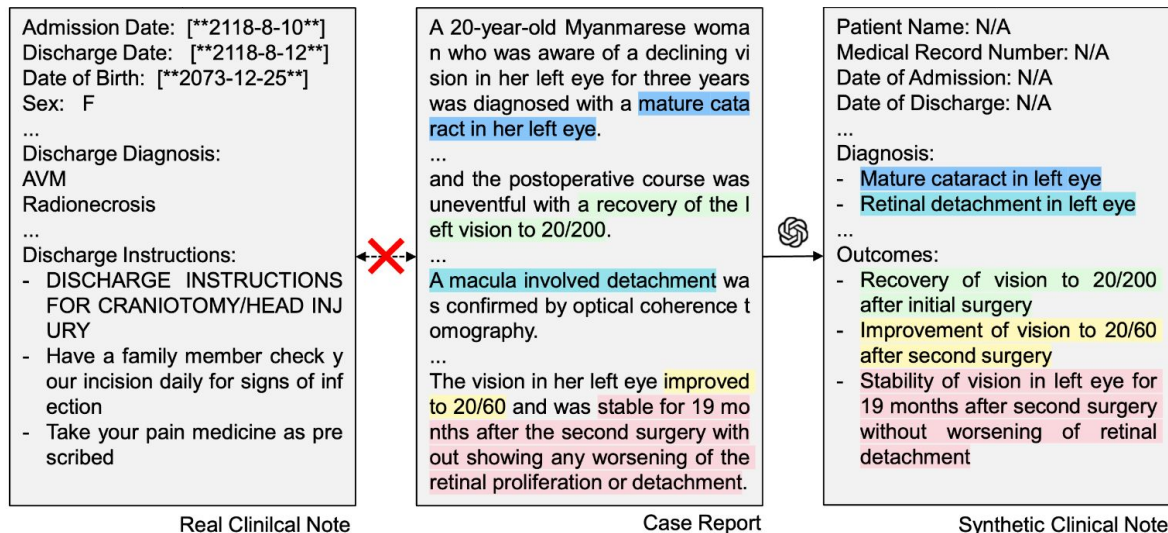


A 20-year-old Myanmar woman who was aware of a declining vision in her left eye for three years was diagnosed with a mature cataract in her left eye.  
...  
and the postoperative course was uneventful with a recovery of the left vision to 20/200.  
...  
A macula involved detachment was confirmed by optical coherence tomography.  
...  
The vision in her left eye improved to 20/60 and was stable for 19 months after the second surgery without showing any worsening of the retinal proliferation or detachment.

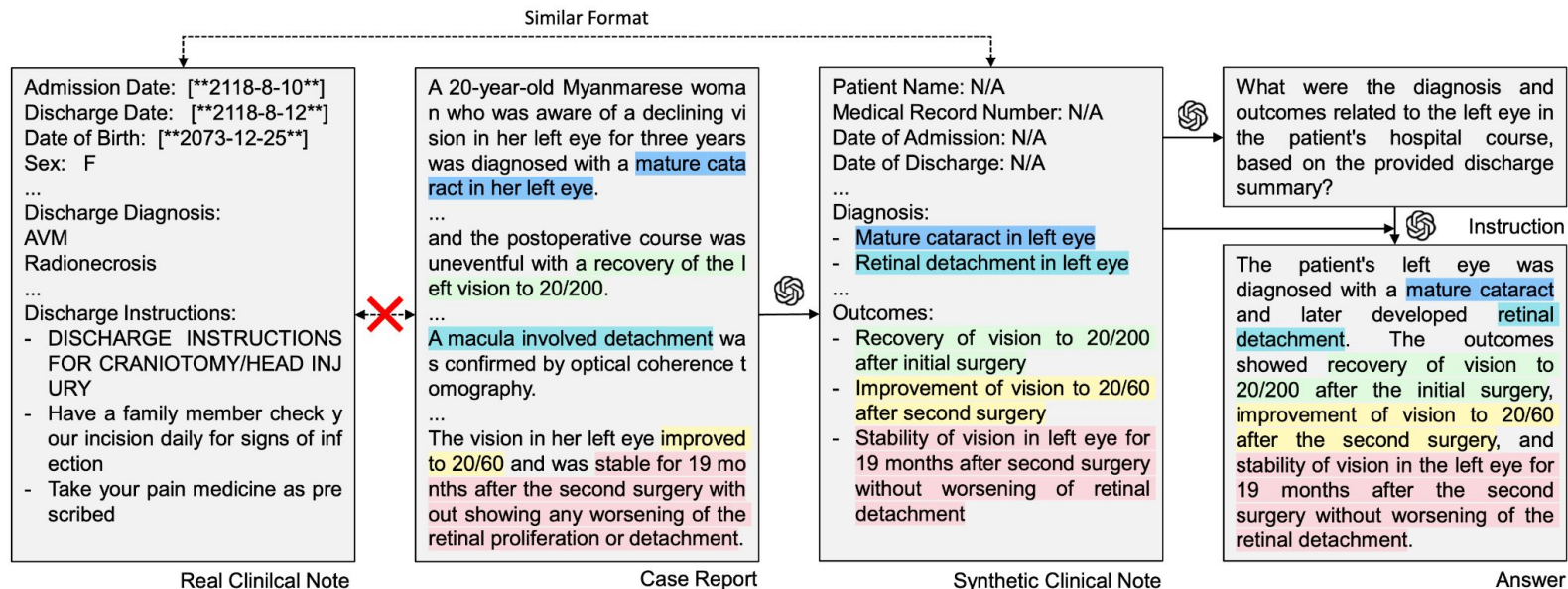
Case Report

- To share “case” with community
  - No PHI ⇒ Sharable
- Properties
  - Plain text
  - Less acronyms
  - Well-written
- Contents are similar to the notes
- e.g., PMC (PubMed Central) case report

# Synthetic clinical note generation



# Clinical instruction/response data generation



- (clinical note, instruction, response) triples  $\Rightarrow$  all synthetics!

Downloads last month 498

---

Use this dataset

Edit dataset card

---

Repository:

Paper:

Size of downloaded dataset files:

Github

arxiv.org

402 MB

Size of the auto-converted Parquet files:

Number of rows:

199 MB

158,114

---

Models trained or fine-tuned on starmppcc/Asclep...

starmppcc/Asclepius-Llama2-7B

Text2Text Generation • Updated Ja... • ↗ 888 • ♥ 12

starmppcc/Asclepius-Llama2-13B

Text2Text generation • Updated Ja... • ↗ 925 • ♥ 11

mzradermacher/Asclepius-Llama3-8B-ii-G...

Updated 4 days ago • ↗ 495

TheBloke/Asclepius-13B-GGUF

# Asclepius-Llama3-8B

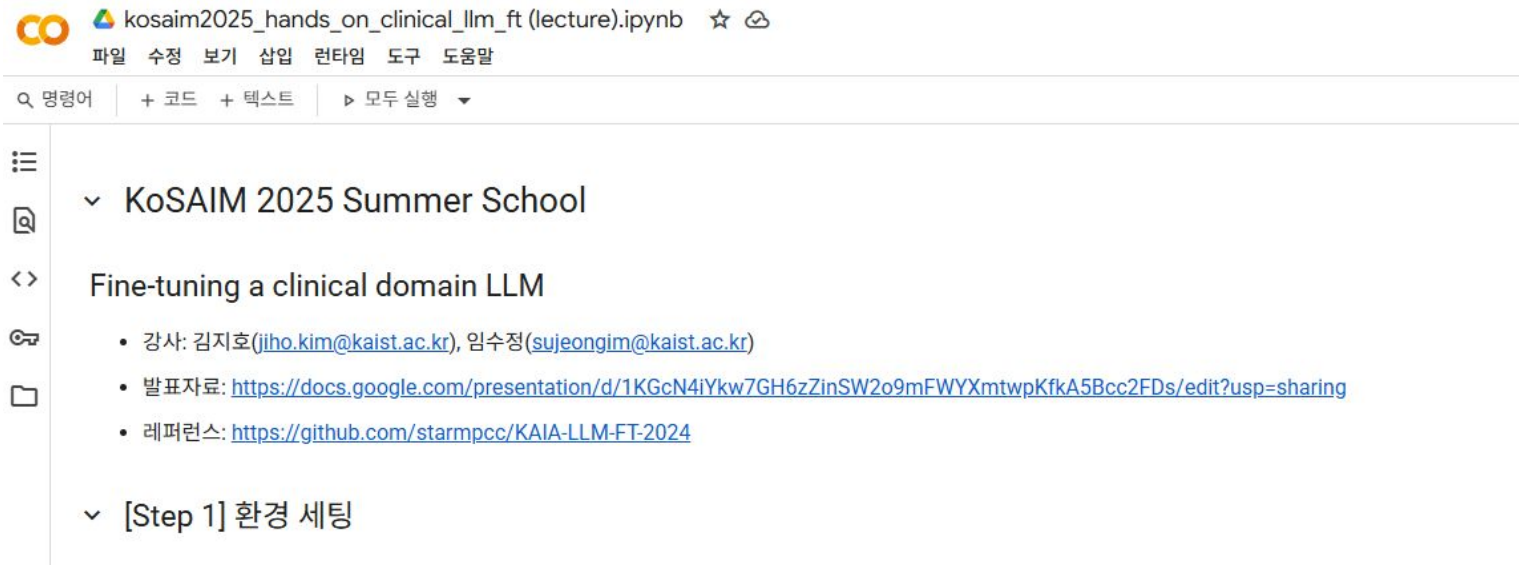
- How can we build an instruction-following LLM in the clinical domain?
  - Prepare a pre-trained large language model
    - use Llama3-8B model
  - Pre-training on clinical corpus for domain adaptation
    - Pre-training (1 epoch): 2h 59m with 4x A100 80G
    - dataset: synthetic clinical notes
  - Perform supervised fine-tuning using domain-specific clinical instruction data
    - Instruction fine-tuning (3 epoch): 30h 41m with 4x A100 80G
    - dataset: clinical instruction-response pairs with synthetic clinical notes

# Hands-on Session:

## Fine-tuning a clinical domain LLM

# Environment Setup

- <https://github.com/jiho283/>
- <https://github.com/jiho283/Clinical-LLM-HandsOn-Session>



The screenshot shows the Jupyter Notebook interface for the file 'kosaim2025\_hands\_on\_clinical\_llm\_ft (lecture).ipynb'. The left sidebar contains a table of contents with the following items:

- ☰
- 📁
- 🔗
- 🔑
- 📁

The main content area displays the following structure:

- ▼ KoSAIM 2025 Summer School
  - <> Fine-tuning a clinical domain LLM
    - 🔑 강사: 김지호([jiho.kim@kaist.ac.kr](mailto:jiho.kim@kaist.ac.kr)), 임수정([sujeongim@kaist.ac.kr](mailto:sujeongim@kaist.ac.kr))
    - 📄 발표자료: <https://docs.google.com/presentation/d/1KGcN4iYkw7GH6zZinSW2o9mFWYXmtwpKfkA5Bcc2FDs/edit?usp=sharing>
    - 📄 레퍼런스: <https://github.com/starmppcc/KAIA-LLM-FT-2024>
  - ▼ [Step 1] 환경 세팅

# Environment Setup

CO kosaim2025\_hands\_on\_clinical\_llm\_ft (lecture).ipynb ☆ ☁

파일 수정 보기 삽입 런타임 도구 도움말

명령어 코드 텍스트

모두 실행 Ctrl+F9

이전 셀 실행 Ctrl+F8

초점이 맞춰진 셀 실행 Ctrl+Enter

선택항목 실행 Ctrl+Shift+Enter

셀 및 하위 셀 실행 Ctrl+F10

실행 중단 Ctrl+M |

세션 다시 시작 Ctrl+M .

세션 다시 시작 및 모두 실행

런타임 연결 해제 및 삭제

런타임 유형 변경

세션 관리

리소스 보기

▼ KoSAIM 2025

Fine-tuning a clinical LLM

- 강사: 김지호(jiho.kim@kci.go.kr)
- 발표자료: <https://www.kci.go.kr/ko/publications/publicationDetail.do?seq=123456789>
- 레퍼런스: <https://arxiv.org/abs/2009.00793>

▼ [Step 1] 환경 설정

▼ 패키지 설치

### 런타임 유형 변경

런타임 유형

Python 3

하드웨어 가속기 ?

☐ CPU ☒ T4 GPU ☐ A100 GPU ☐ L4 GPU

☐ v2-8 TPU ☐ v6e-1 TPU ☐ v5e-1 TPU

프리미엄 GPU를 이용하시겠어요? [추가 컴퓨팅 단위 구매](#)

취소 저장



# Colab Objectives

- Goal: Fine-tuning a clinical domain LLM
- Environment: Google Colab
- Dataset: starmppcc/Asclepius-Synthetic-Clinical-Notes
- Model: microsoft/phi-2 (2.7B)
- **CAUTION (주의)**
  - LLM 학습하는 과정에서 Colab을 절대 끄지 마시기 바랍니다.
    - 새로고침 금지
    - 코랩 내에서 다른 버튼 클릭 금지
    - 실행 중지 금지

# Deep learning memory layout

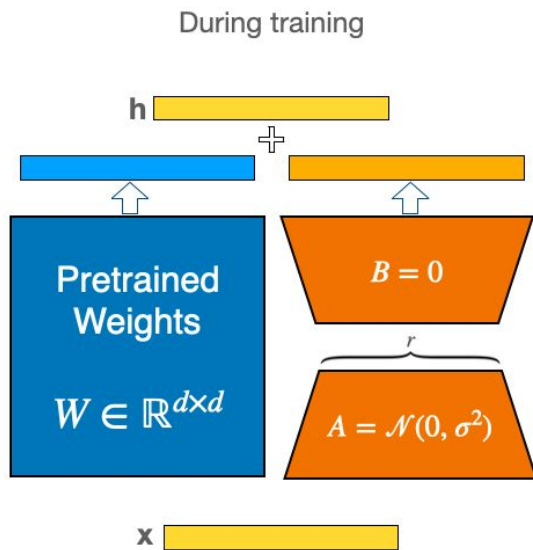
- Model size: B (billion) scale
  - $x\text{B parameters} = x\text{B floating point numbers} = 2x \text{ GB (bf16/fp16)}$
- Deep Learning Memory Requirements
  - model parameter:  $2x \text{ GB}$
  - gradient state:  $2x \text{ GB}$
  - optimizer state:  $2x \sim 12x \text{ GB}$
  - Total:  $6 \sim 16x \text{ GB} + \text{alpha}$
- Our requirements
  - model: phi-2 (2.7B)
  - GPU VRAM: Colab T4 (16GB)
  - $2.7 * 6 = 16.2$

# Can You Run it?

- <https://huggingface.co/spaces/Vokturz/can-it-run-llm>



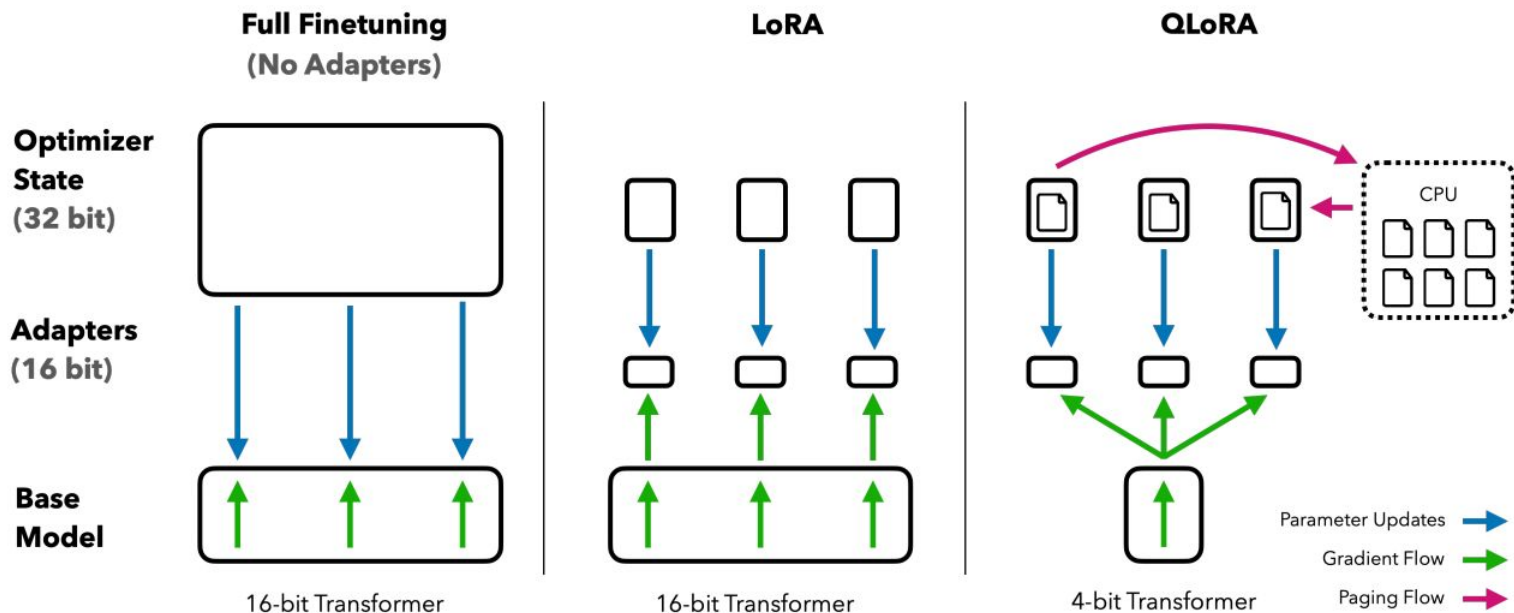
# LoRA (Hu and Shen et al., 2021)



$$h = Wx + BAx$$
$$h = \underbrace{(W + BA)}_{W_{merged}}x$$



# QLoRA (Dettmers and Pagnoni et al., 2023)



# Parameter-Efficient Fine-Tuning (PEFT)

- <https://github.com/huggingface/peft>

Prepare a model for training with a PEFT method such as LoRA by wrapping the base model and PEFT configuration with `get_peft_model`. For the bigscience/mt0-large model, you're only training 0.19% of the parameters!

```
from transformers import AutoModelForSeq2SeqLM
from peft import get_peft_config, get_peft_model, LoraConfig, TaskType
model_name_or_path = "bigscience/mt0-large"
tokenizer_name_or_path = "bigscience/mt0-large"

peft_config = LoraConfig(
    task_type=TaskType.SEQ_2_SEQ_LM, inference_mode=False, r=8, lora_alpha=32, lora_dropout=0.
)

model = AutoModelForSeq2SeqLM.from_pretrained(model_name_or_path)
model = get_peft_model(model, peft_config)
model.print_trainable_parameters()
"trainable params: 2359296 || all params: 1231940608 || trainable%: 0.19151053100118282"
```

# Thank you :D

If you require any further information, feel free to contact us:  
[jiho.kim@kaist.ac.kr](mailto:jiho.kim@kaist.ac.kr), [sujeongim@kaist.ac.kr](mailto:sujeongim@kaist.ac.kr)