Fine-tuning a clinical domain LLM

2024-08-10, 15:30 ~ 17:30

Seongsu Bae, Sujeong Im
KAIST AI @ Edlab (Advised by Edward Choi)

Speaker Bio

Seongsu Bae (배성수)

Education

- Hanyang University Mathematics, B.Sc (2013-2019)
- KAIST Kim Jaechul Graduate School of AI, M.Sc (2020-2022)
- KAIST Kim Jaechul Graduate School of AI, Ph.D (2022-)

Research Interests

- Semantic Machine
- Multimodal Learning
- 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

Language Model

Language model

文A 31 languages

Article Talk Read Edit View history Tools

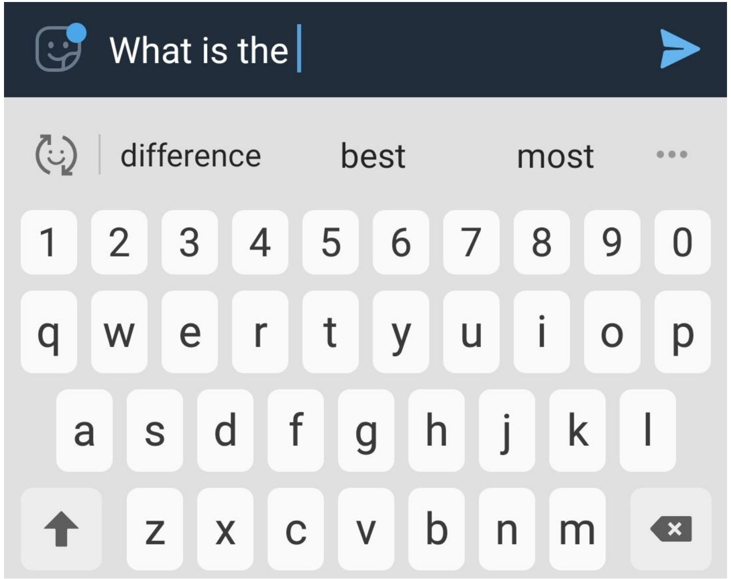
From Wikipedia, the free encyclopedia

A **language model** is a probabilistic model of a natural language.[1] 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.[2]

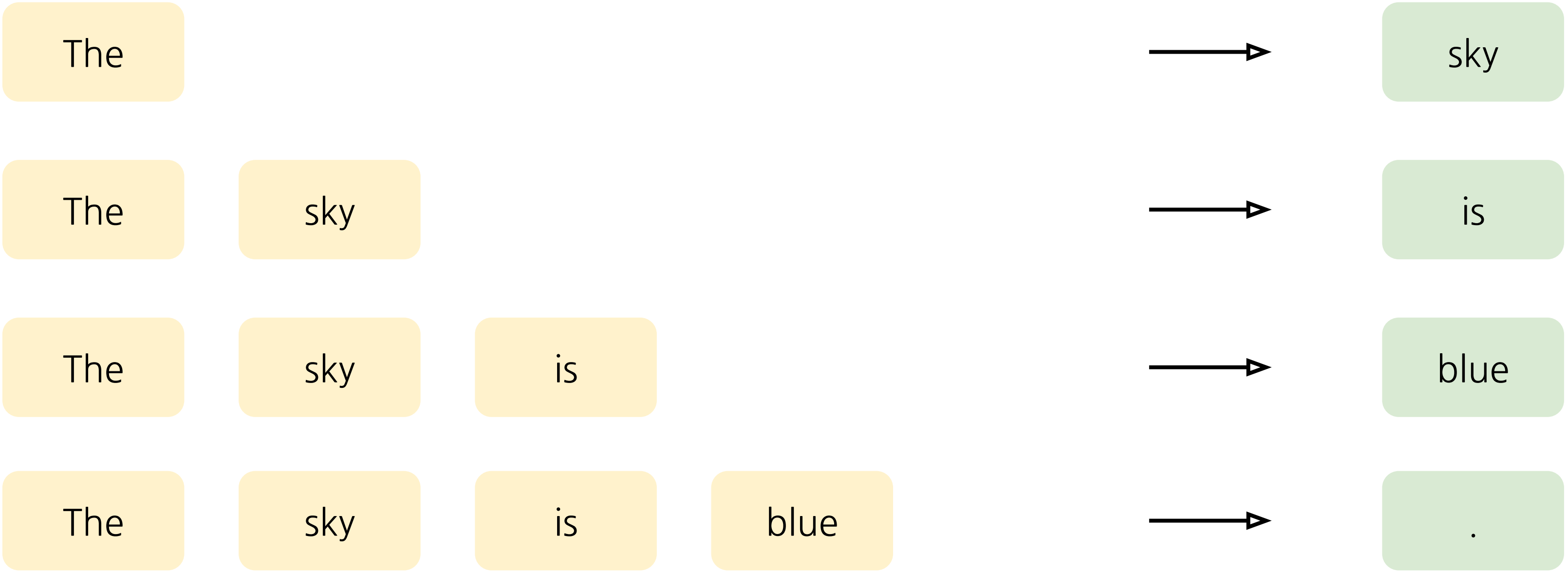
Language models are useful for a variety of tasks, including speech recognition[3] (helping prevent predictions of low-probability (e.g. nonsense) sequences), machine translation,[4] natural language generation (generating more human-like text), optical character recognition, handwriting recognition,[5]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

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 They have superseded recurrent neural network-based models, which had previously superseded the pure statistical models, such as word *n*-gram language model.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!

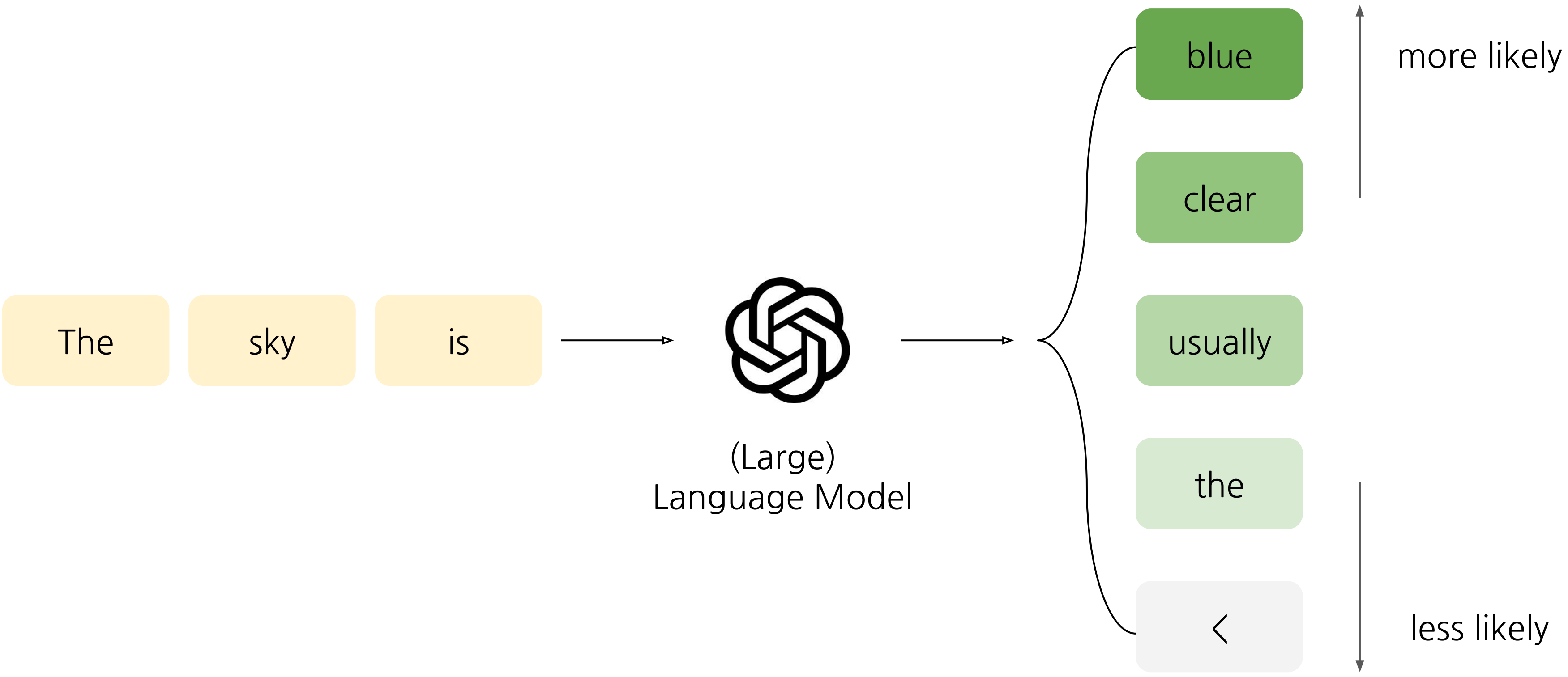


How to train a LM?



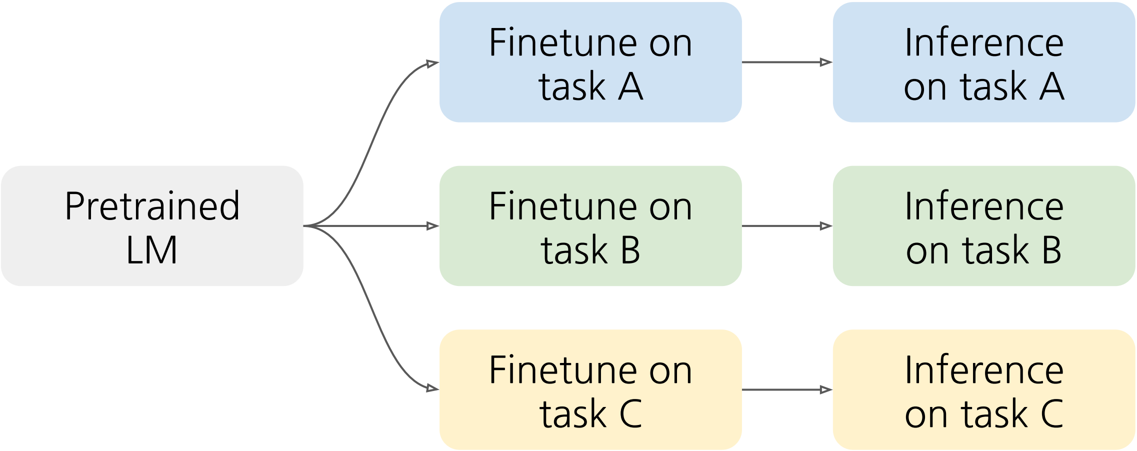
Next Token Prediction task for the sentence “The sky is blue.”

Text Generation via a Probabilistic Model



How to build a (large) language model?

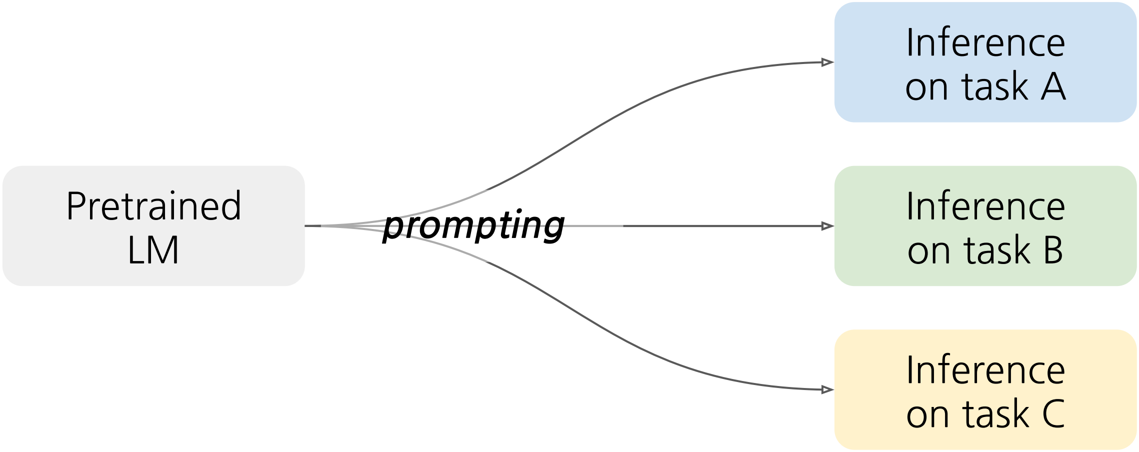
- Pre-training and Fine-tuning
  - e.g., BERT (2018), T5 (2019)



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

How to build a (large) language model?

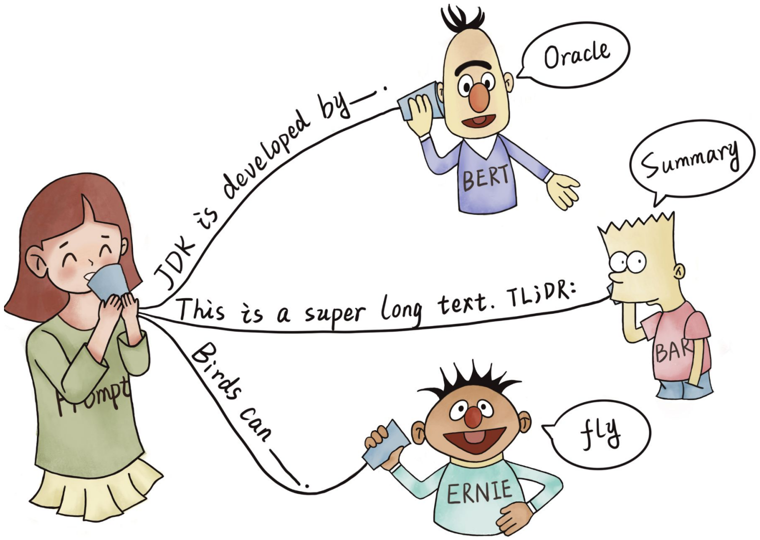
- Pre-training and Prompting
  - e.g., GPT-3 (2020)



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

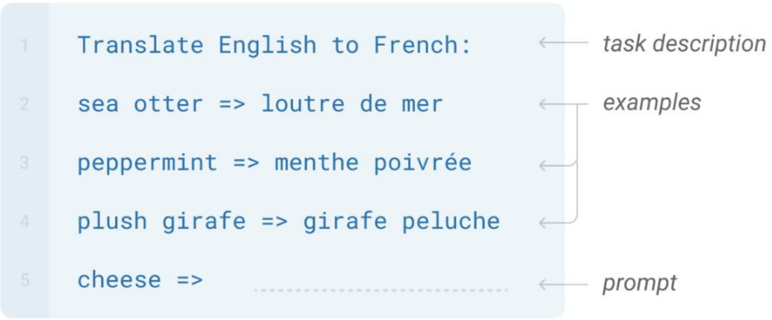
How to build a (large) language model?

- 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.



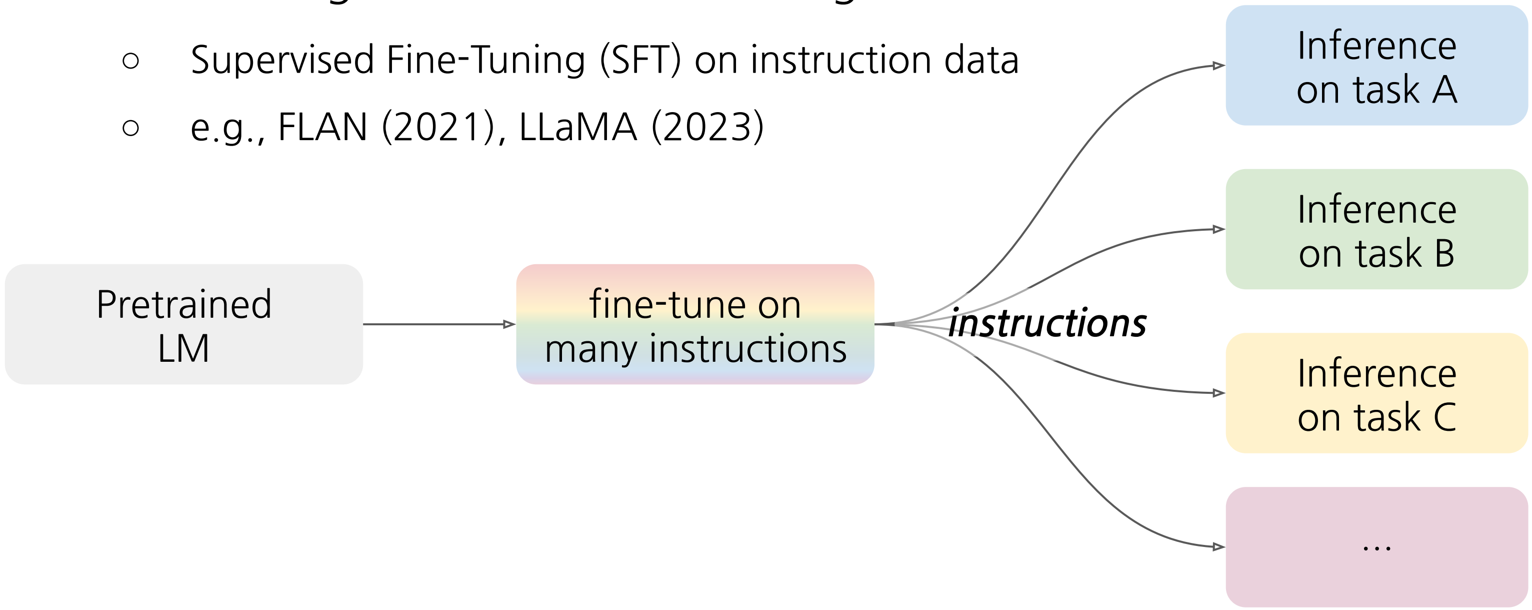
(-) Forced few-shot prompting

(-) Manual efforts for the prompting technique

(-) Not aligned with natural instructions

How to build a (large) language model?

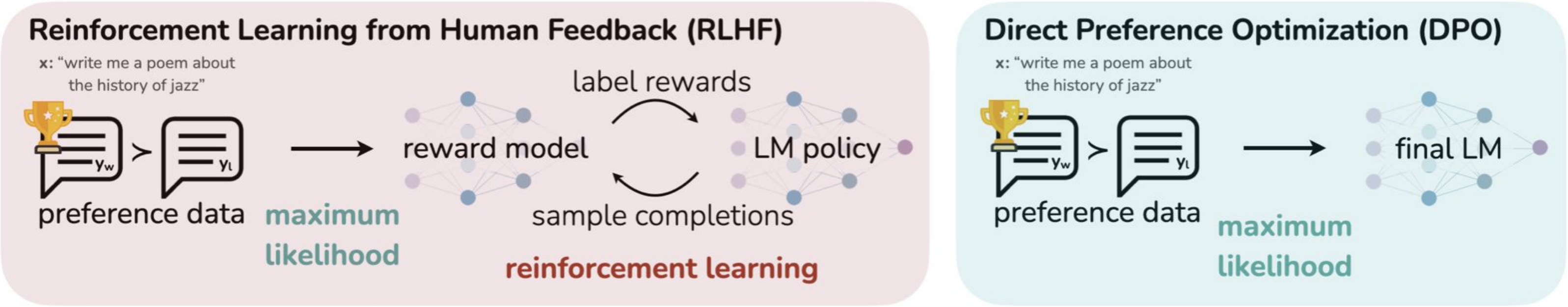
- Pre-training and Instruction tuning
  - Supervised Fine-Tuning (SFT) on instruction data
  - e.g., FLAN (2021), LLaMA (2023)



(+) 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

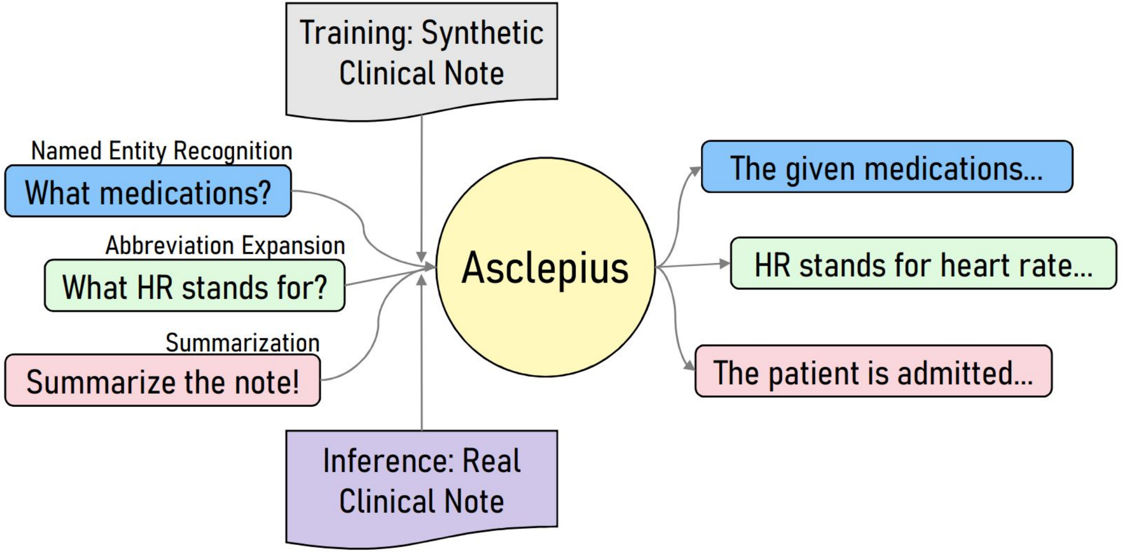
- How can we build an instruction-following LLM?
  - Prepare a pre-trained large language model (e.g., LLaMA 7B)
  - Perform supervised fine-tuning on instruction data (e.g., Alpaca 52K dataset)

- 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 (Gweon and Kim et al., ACL 2024 Findings)



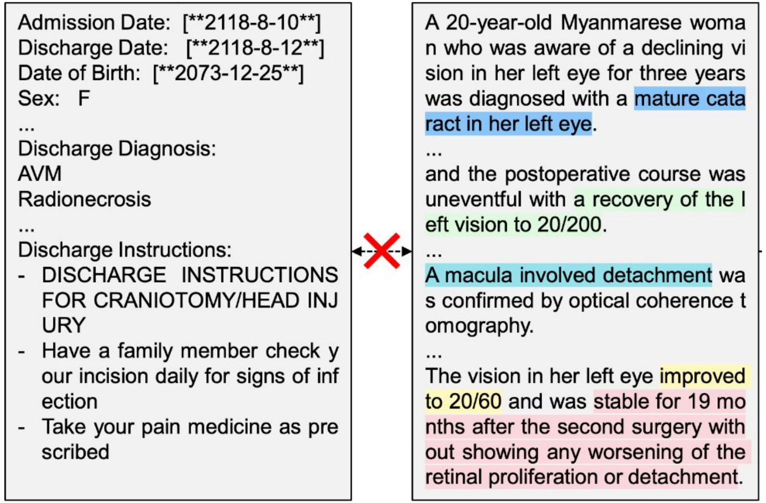
Real clinical note

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 Clinilcal Note

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

Case report

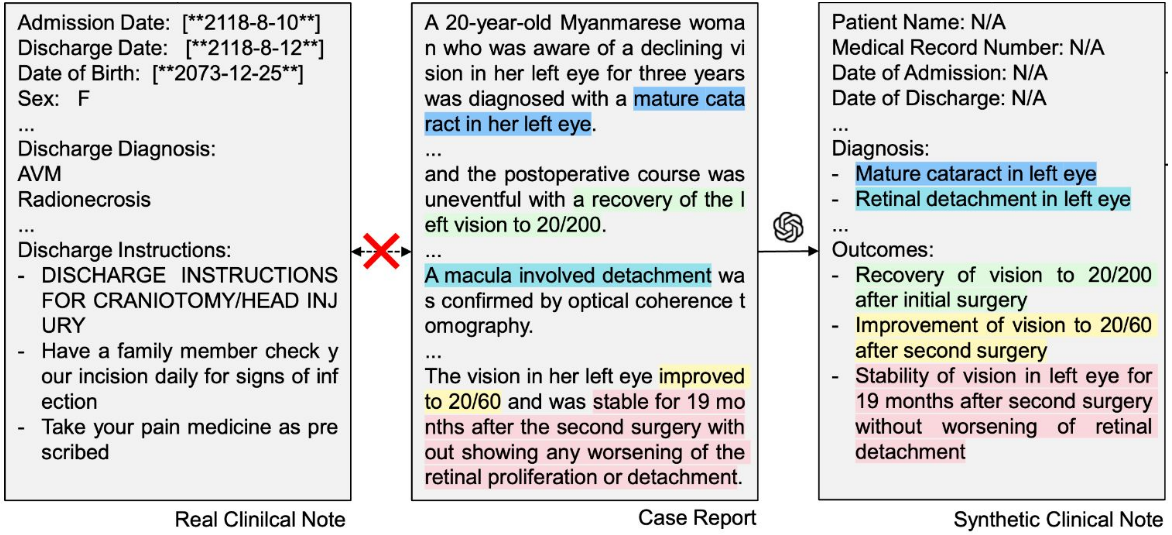


Real Clinilcal Note

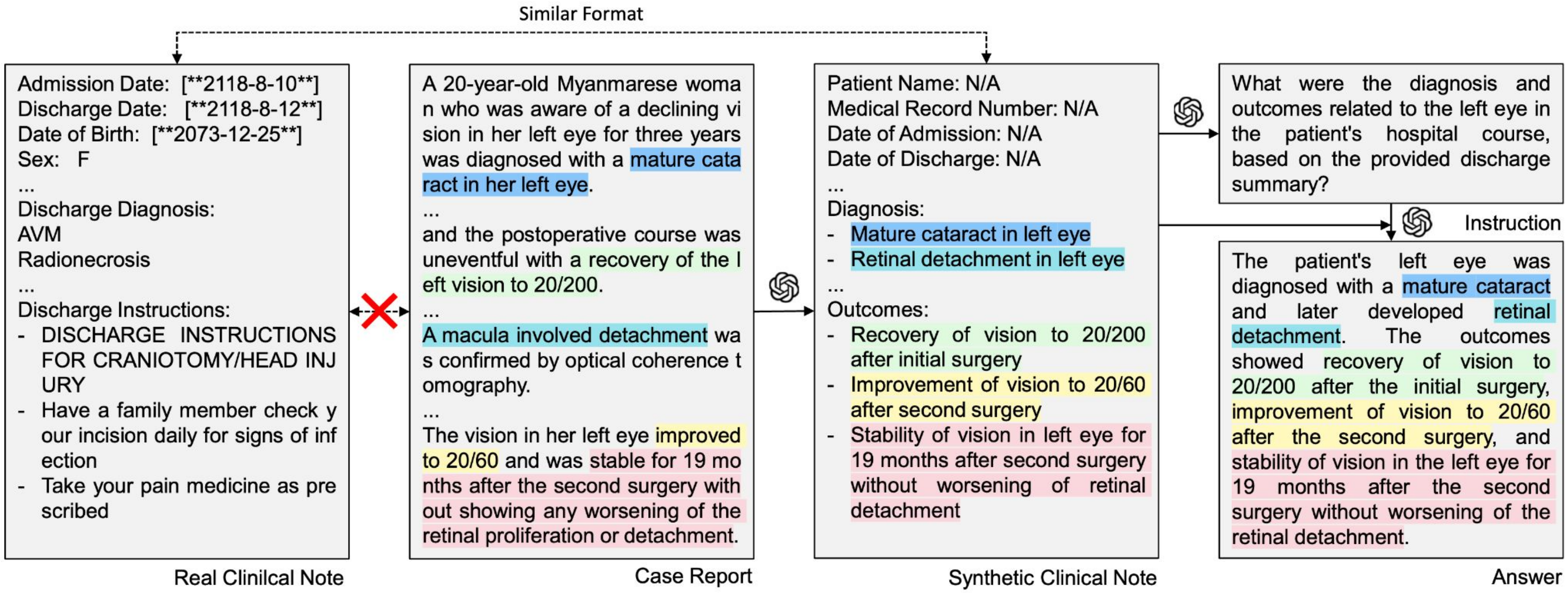
Case Report

- 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

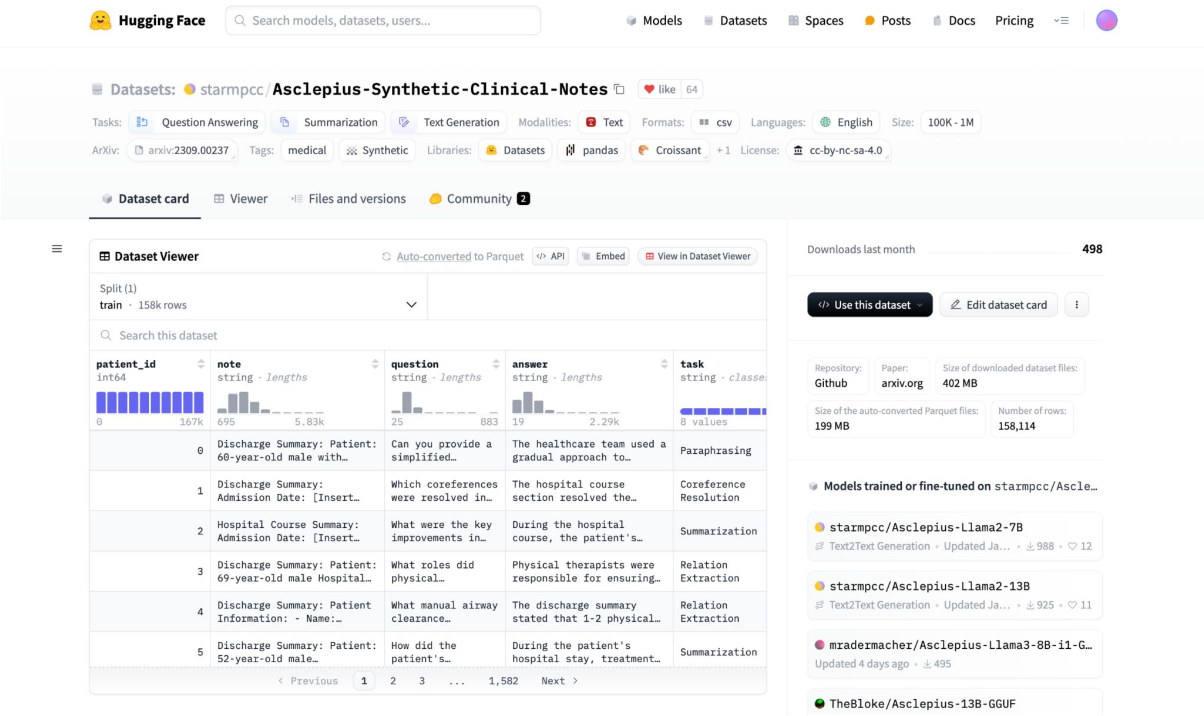


Clinical instruction/response data generation



Final dataset

- (clinical note, instruction, response) triples ⇒ all synthetics!



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

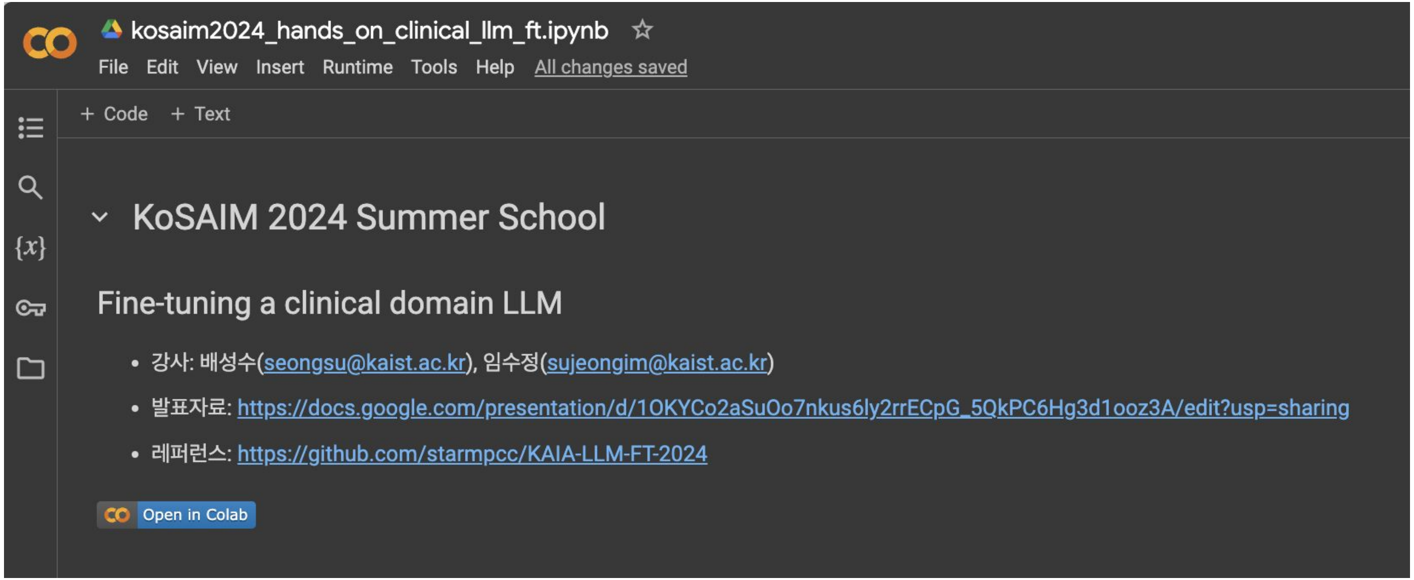
Environment Setup

- https://github.com/baeseongsu/

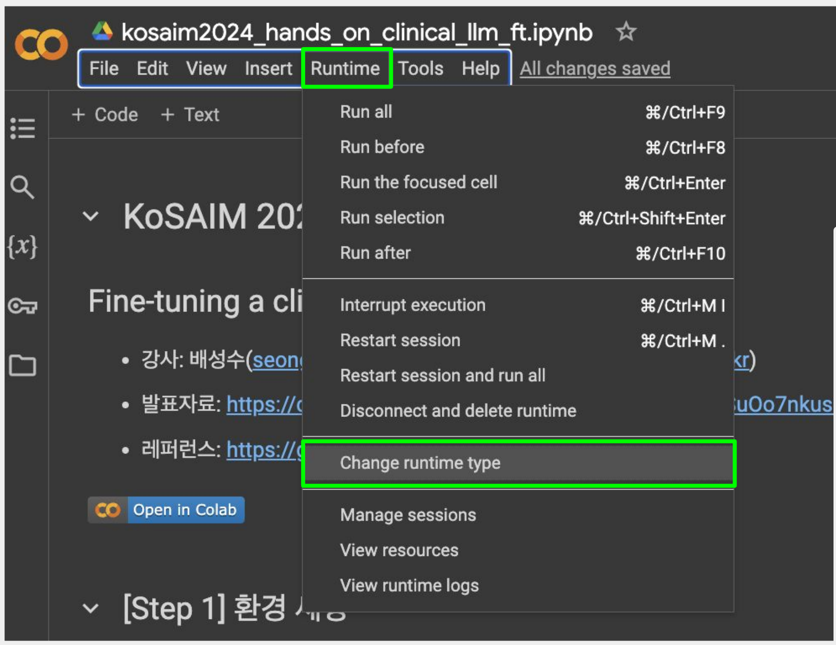
colab link

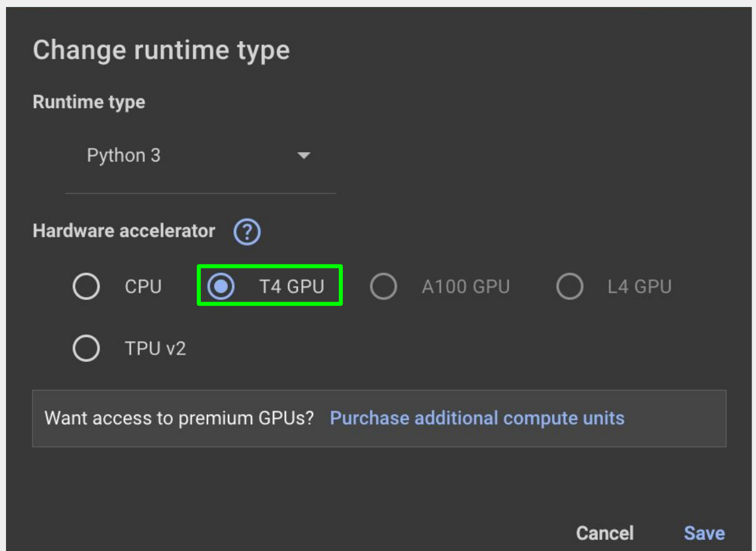
Environment Setup

- https://github.com/baeseongsu/KoSAIM2024-Clinical-LLM



Environment Setup





Colab Objectives

- Goal: Fine-tuning a clinical domain LLM
- Environment: Google Colab
- Dataset: starmpcc/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**

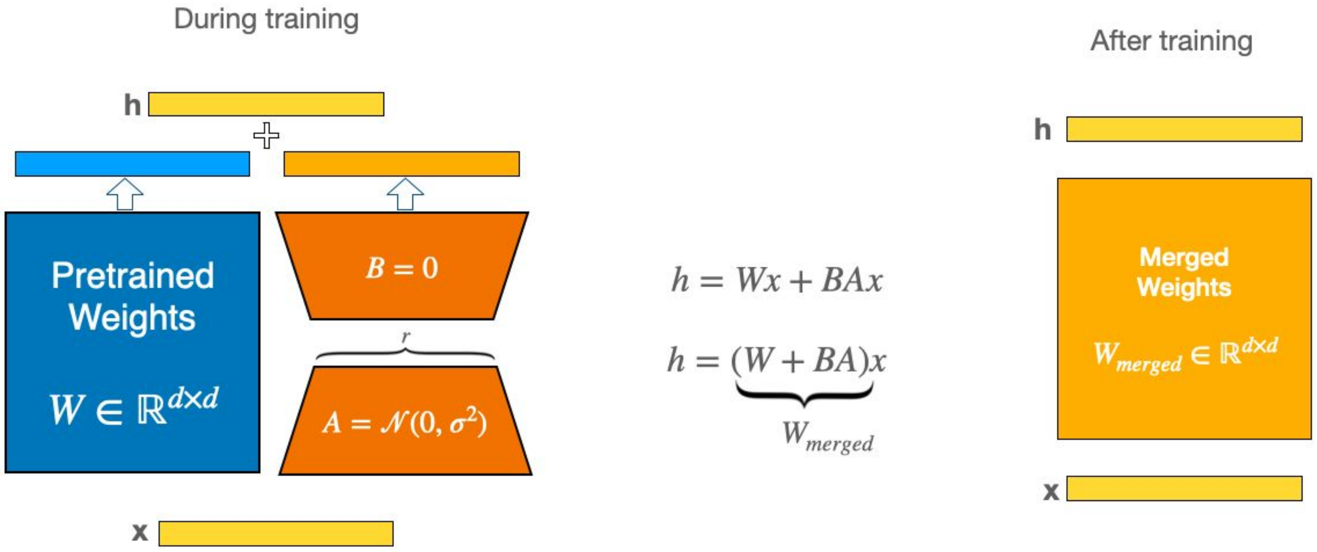
Deep learning memory layout

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

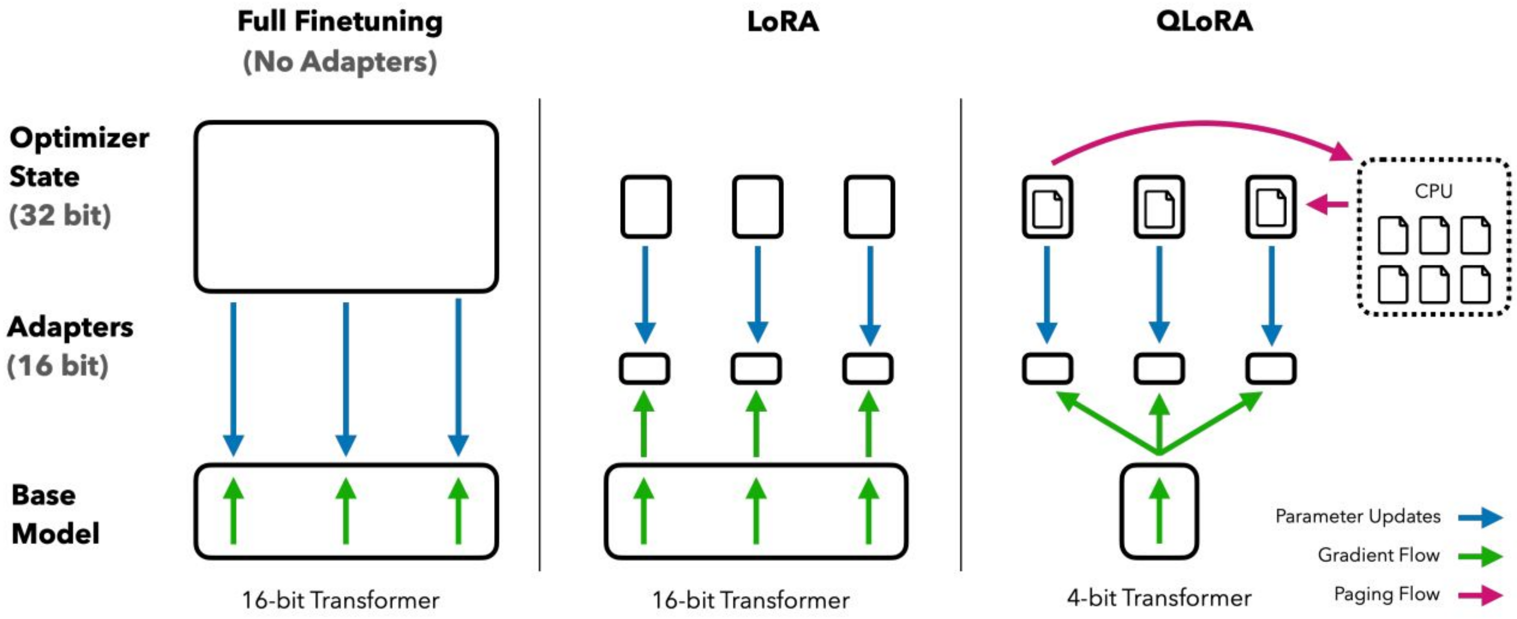
Can You Run it?

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

LoRA (Hu and Shen et al., 2021)



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:

seongsu@kaist.ac.kr, sujeongim@kaist.ac.kr