SKKU Lab Recommendation Service : FindMyLab

Team J (Last Lap) 김현진 송민석 장민석 조재희

2024-Fall Capstone Design Project

Backend Part – Tagging

Tag Collection

Abstract of Paper

The next generation of artificial intelligence (AI) is required to be capable of proper communication to enable eloquent interaction with human beings. Thus, AI models require powerful language understanding and generation capabilities. ….

GPT Prompt Engineering

- Artificial intelligence
- AI communication
- Language understanding
- task-oriented dialogue systems
- dialogue research
- language generation

Backend Part – tagging

Clustering

Synonym

Database manage

Database management

Database management system

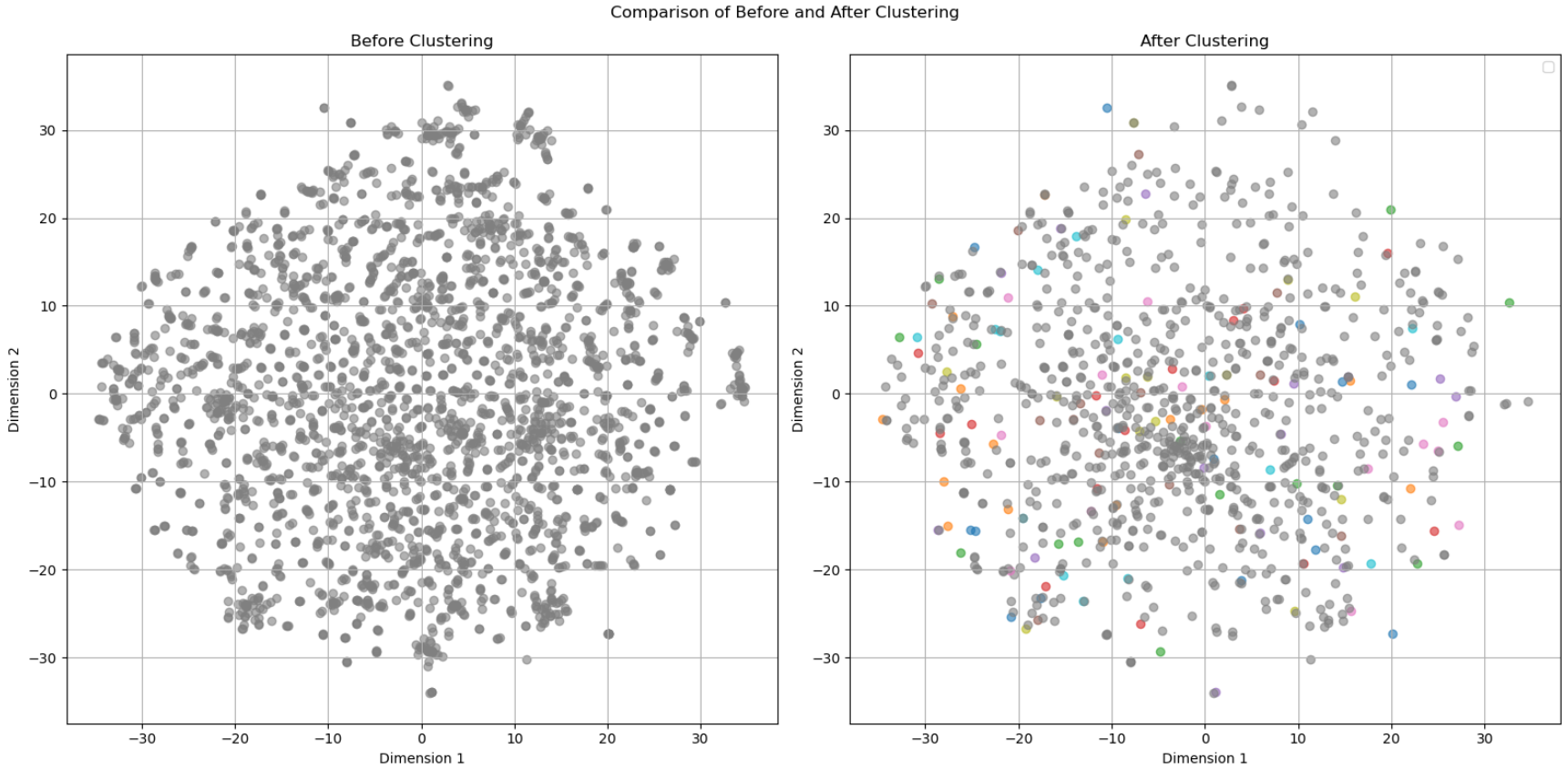
Similar Word

Medical technology

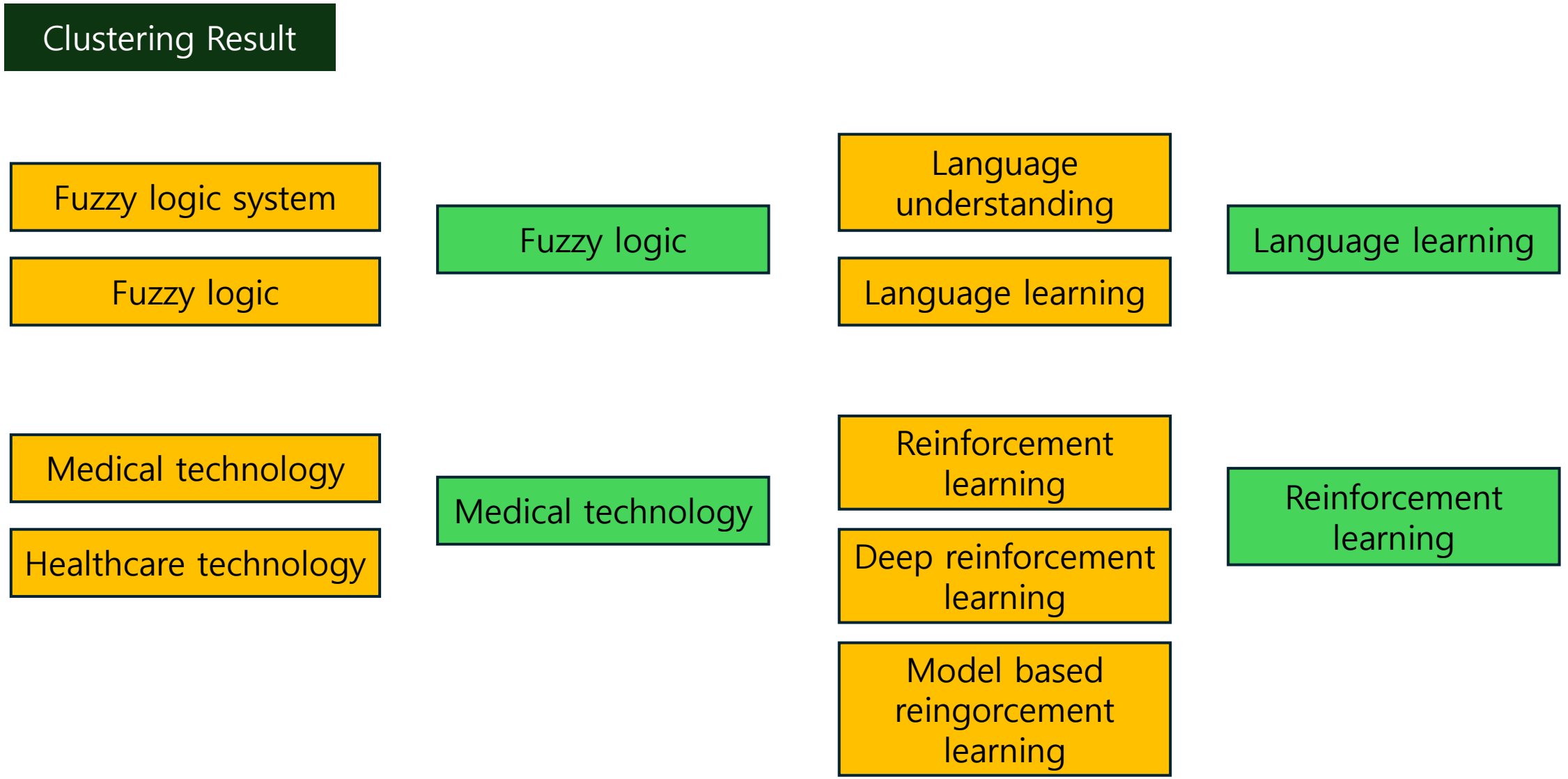
Healthcare technology

Backend Part – tagging

Clustering



Backend Part – tagging



Backend Part – Performance Tuning

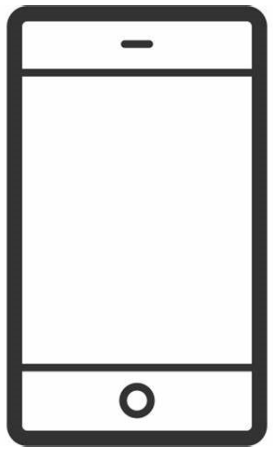
System Log

```
app listening on port  3000
DB is Connected
Get request
Embedding Time: 5.966s
Query Search Time: 61.59ms
Tag Search Time: 99.441ms
End
```

[Server system log]

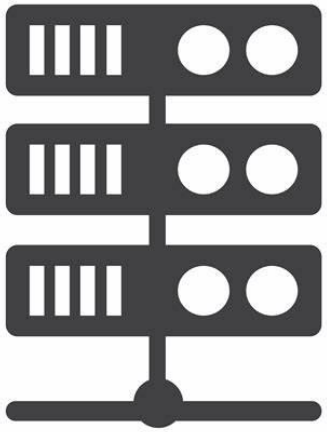
Backend Part – Performance Tuning

Embedding Process



[User]

Search
Word



AWS Server (Slow)

Create child process
and System call

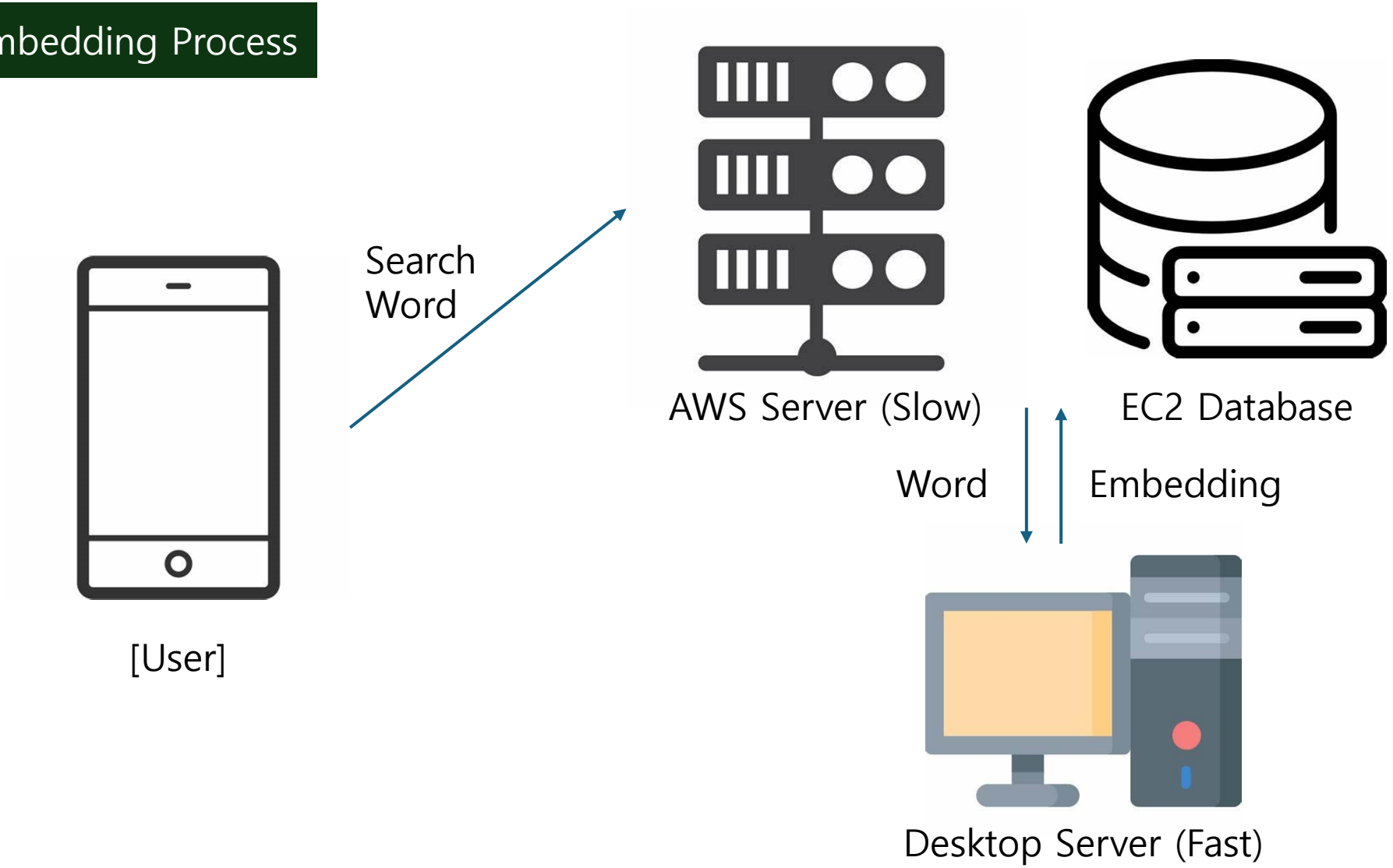
python™

[Embedding Program]

```
const pythonProcess = spawn('python', ['emb.py', input]);
let data = '';
let error = '';
return new Promise((resolve, reject) => {
    pythonProcess.stdout.on('data', (chunk) => {
        data += chunk.toString();
    });
    pythonProcess.stderr.on('data', (chunk) => {
        error += chunk.toString();
    });
    pythonProcess.on('close', () => {
        if (error) {
            reject(error);
        } else {
            try {
                resolve(JSON.parse(data));
            } catch (err) {
```

Backend Part – Performance Tuning

Embedding Process



Backend Part – Performance Tuning

Result

Issue of time

> 5secs

```
app listening on port  3000
DB is Connected
Get request
Embedding Time: 52.795ms
Query Search Time: 9.91ms
Tag Search Time: 92.14ms
End
```

[Server system log]

AI Part – finetune

data-preprocessing

CSV → json

NAME	Abstract	Keywords
고영중		
고영중		
고영중		
…		
…		

{

“query” : key word (expected search query: using extracted keyword from abstract),
“pos”: [source abstract]

}

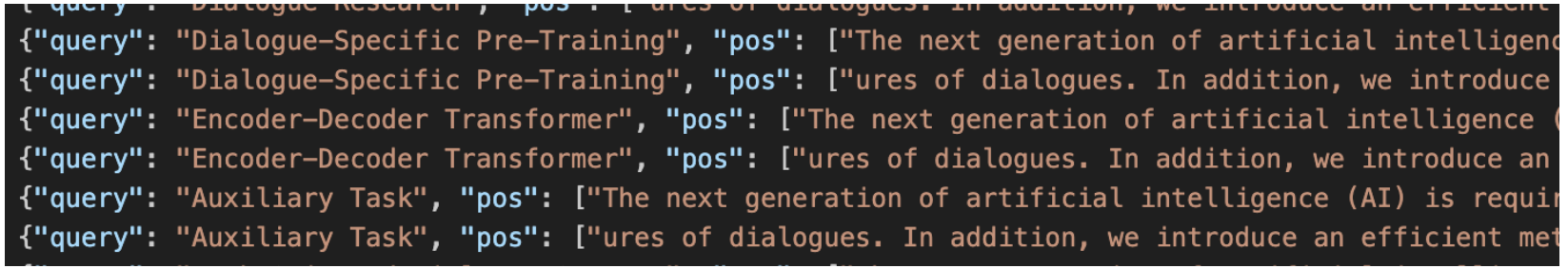
AI Part – finetune

data-preprocessing

512 tokenize

{ "**query**" : key word A, **"pos"**: [source abstract A-1]}
{ "**query**" : key word A, **"pos"**: [next source abstract A-2]}

{ "**query**" : key word B, **"pos"**: [source abstract B-1]}
{ "**query**" : key word B, **"pos"**: [next source abstract B-2] }



total: 18536

AI Part – finetune

data-preprocessing

only pos json →hard negative mining

```
!python -m FlagEmbedding.baai_general_embedding.finetune.hn_mine \
--model_name_or_path BAAI/bge-m3 \
--input_file /data/projects/jaehee/jupyter/processed_dataset_pos_only.jsonl \
--output_file /data/projects/jaehee/jupyter/processed_dataset_hdn.jsonl \
--range_for_sampling 15-200 \
--negative_number 10 \
--use_gpu_for_searching
```

AI Part – finetune

data-preprocessing

result of hard negative mining

{ "**query**" : key word A, **"pos"**: [source abstract A-1], **"neg"**:[mining result 1], ... ,[mining result N] }
{ "**query**" : key word A, **"pos"**: [next source abstract A-2] , **"neg"**:[mining result 1], ... ,[mining result N] }

{ "**query**" : key word B, **"pos"**: [source abstract B-1] , **"neg"**:[mining result 1], ... ,[mining result N] }
{ "**query**" : key word B, **"pos"**: [next source abstract B-2] , **"neg"**:[mining result 1], ... ,[mining result N] }

AI Part – finetune

Model Change

! issue of out of memory
→ changed to small model

SBERT-base-nli-v2 → BAAI/bge-small-en-v1.5

AI Part – search algorithm

Model Change

BAAI/bge-small-en-v1.5

Model Name	Dimension	Sequence Length	Average (56)	Retrieval (15)	Clustering (11)	Pair Classification (3)
BAAI/bge-large-en-v1.5	1024	512	**64.23**	**54.29**	46.08	87.12
BAAI/bge-base-en-v1.5	768	512	63.55	53.25	45.77	86.55
BAAI/bge-small-en-v1.5	384	512	62.17	51.68	43.82	84.92

AI Part – finetune

Model config

```
!torchrun --nproc_per_node 1 \
-m FlagEmbedding.baai_general_embedding.finetune.run \
--output_dir ./checkpoint_v1_bs32 \
--model_name_or_path BAAI/bge-small-en-v1.5 \
--train_data ./processed_dataset_hdn.jsonl \
--learning_rate 1e-5 \
--fp16 \
--num_train_epochs 14 \
--per_device_train_batch_size 32 \
--dataloader_drop_last True \
--normlized True \
--temperature 0.02 \
--query_max_len 64 \
--passage_max_len 512 \
--train_group_size 2 \
--negatives_cross_device \
--logging_steps 10 \
--save_steps 1000 \
--query_instruction_for_retrieval ""
```

batch size: 32
epoch: 14

AI Part – finetune

Ex)

search keywords

```
['Compiler toolchain provenance',
 'BERT-based system',
 'Toolchain classification',
 'Binary code similarity detection',
 'Machine learning',
 'Fine-tuning process',
 'Binary analysis',
 'Signature-based tool',
 'Emerging compilation toolchains (Rust Go Nim)',
 'C and C++ compilers (GCC clang)']
```

AI Part – finetune

Ex)

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 January 2024, accepted 11 January 2024, date of publication 17 January 2024,
date of current version 26 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3355098

RESEARCH ARTICLE

ToolPhet: Inference of Compiler Provenance From Stripped Binaries With Emerging Compilation Toolchains

HOHYEON JANG, NOZIMA MURODOVA, AND HYUNGJOON KOO
Department of Computer Science and Engineering, Sungkyunkwan University, Suwon-si, Gyeonggi-do 16419, Republic of Korea
Corresponding author: Hyungjoon Koo (kevin.koo@skku.edu)

This work was supported in part by the Basic Science Research Program through National Research Foundation of Korea (NRF) Grant funded by the Ministry of Education, Government of South Korea, under Grant 2022R1F1A1074373; and in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Korean Government [Minister of Science, Information and Communications Technology (MSIT)], Graduate School of Convergence Security, Sungkyunkwan University, under Grant 2022-0-01199.

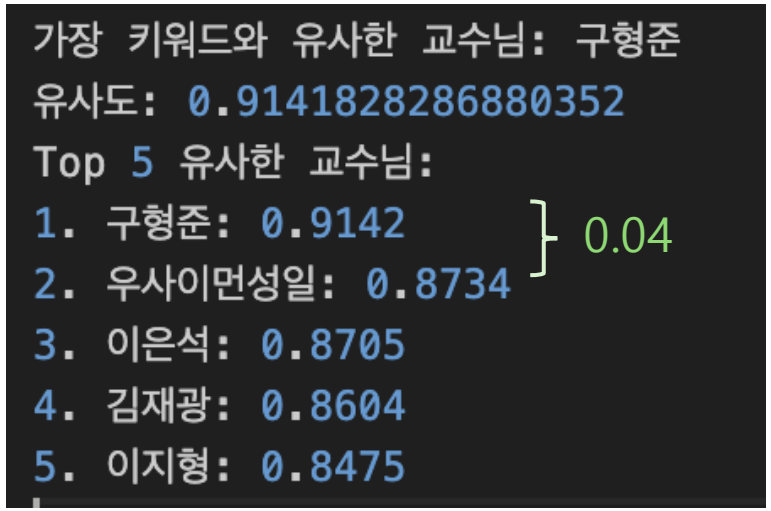
The keywords are highly similar to those used in Professor Koo Hyungjun's actual research papers

AI Part – finetune

RESULT

<before>

<after>



AI Part – finetune

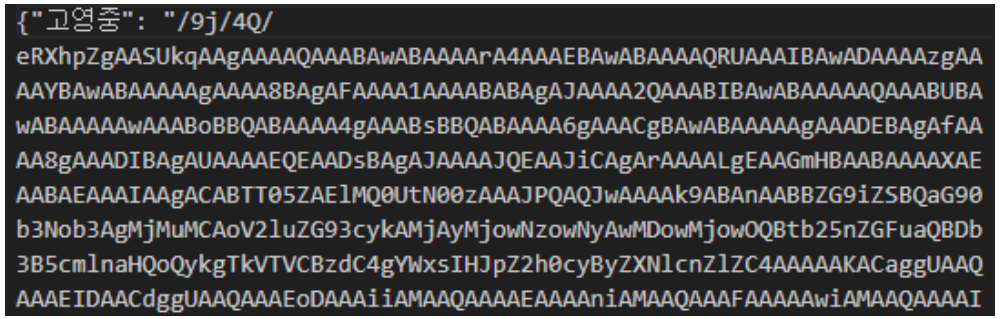
Compare

```python
from FlagEmbedding import FlagModel
sentences_1 = ["Dialogue-Specific Pre-Training"]
sentences_2 = ["The next generation of artificial intelligence (AI) is required to be capable of proper communication to enable e

model = FlagModel('BAAI/bge-small-en', use_fp16=True)
fine_tuned_model = FlagModel('/data/projects/jaehee/jupyter/checkpoint_v1_bs32/checkpoint-3000', use_fp16=True)

# 기존 모델로 각각 임베딩
embeddings_1 = model.encode(sentences_1)
embeddings_2 = model.encode(sentences_2)

# 기존 모델로부터 나온 임베딩으로 유사도 계산
similarity_from_base_model = embeddings_1 @ embeddings_2.T

# 파인 튜닝 모델로 각각 임베딩
embeddings_1 = fine_tuned_model.encode(sentences_1)
embeddings_2 = fine_tuned_model.encode(sentences_2)

# 파인 튜닝 모델로부터 나온 임베딩으로 유사도 계산
similarity_from_fine_tuned_model = embeddings_1 @ embeddings_2.T

print('기존 모델:', similarity_from_base_model)
print('파인 튜닝된 모델:', similarity_from_fine_tuned_model)
```

```
기존 모델: [[0.923 0.813]]
파인 튜닝된 모델: [[0.825 0.578]]
```

Frontend Part

Image data processing : base 64



Base 64 encoded image data

Decode and output the encoded image data

Demo Video


localhost:3001
www.BANDICAM.com
FindMyLab
FindMyLab
키워드 검색
Example : AI, Network

Thank you.