

# 영어분석을위한기계학습 – 6th Assignment Final Report

프로젝트명: BonCahier AI — 수업 자료 자동 요약과 번역 및 핵심 정리 도구

과목명: 영어분석을위한기계학습

학번: 201903774

이름: 한형준

전공: 언어인지과학과

제출일: 2025-12-06

## 1. 프로젝트 개요

현대 대학 수업 환경에서 학생들은 여러 과목의 강의 자료(PPT, PDF)를 동시에 관리해야 하고, 과제·시험 일정이 몰리는 시기에는 짧은 시간 안에 많은 슬라이드를 정리해야 하는 문제가 있다. 특히 다음과 같은 어려움이 존재한다.

- 텍스트 양이 많은 슬라이드(특히 이론 설명 위주 과목)의 경우,  
직접 요약 노트를 만드는 데 시간이 과도하게 소요됨
- 수업 자료가 영어 등 외국어로 작성된 경우,  
번역과 이해를 동시에 해야 하므로 인지적 부담이 큼
- 한 학기 동안 받은 모든 파일(PPT, PDF)을 시험 직전에 다시 열어보며  
핵심만 정리하기가 매우 비효율적

이를 해결하기 위해, 본 프로젝트에서는 강의 자료(PPT, PDF)를 업로드하면 슬라이드/페이지 단위로 텍스트를 추출하고, KoBART 기반 모델로 핵심 요약과 번역을 자동 생성하는 도구 “BonCahier AI”를 설계·구현하였다.

## 2. 진행 과정 요약 (Assignment 3–5)

### 2.1 Assignment 3 – 프로젝트 제안서 작성

Assignment 3에서는 먼저 문제 정의와 서비스 아이디어를 구체화했다.

- 목표 서비스:
  - PPT/PDF 강의 자료 업로드 → 핵심 요약 + 번역 + 간단한 설명 자동 생성
- 주요 기능:

- 파일 업로드(PPT/PDF) 및 텍스트 추출(PyMuPDF, python-pptx)
  - 문단/슬라이드 단위 요약
  - 외국어 자료에 대한 한국어 번역
  - 결과를 Markdown/PDF 형식으로 저장
  - 기대 효과:
    - 시험 대비, 과제 준비, 발표 준비 시  
"핵심 요약 노트"를 자동으로 얻어 **복습 효율 극대화**
- 이 단계에서는 실제 구현보다는 전체 그림(UX 시나리오, 모델 후보, 기술 스택)을 설계하는 데 집중했다.

## 2.2 Assignment 4 – 데이터 수집 및 분석

Assignment 4에서는 강의 슬라이드 요약 태스크와 유사한 구조를 갖는 **요약 데이터셋을 수집·분석(EDA)**하였다.

- 한국어 요약 데이터:
  - daekeun-ml/naver-news-summarization-ko
  - 약 2만 건 이상의 뉴스 기사(document)와 요약문(summary) 쌍
  - 결측치 거의 없고, 카테고리 분포·문장 길이 분포 등을 확인
- 영어 요약 데이터:
  - gopalkalpande/bbc-news-summary
  - BBC 뉴스 기사와 요약문 쌍

EDA 결과, 한국어 네이버 뉴스 데이터는 **요약 태스크의 대표적인 벤치마크로 활용 가능**했고, 문장 길이와 요약 길이 분포도 강의 슬라이드의 "설명형 텍스트"와 어느 정도 유사하다고 판단하였다. 따라서 Assignment 5에서는 이 데이터셋을 **슬라이드 요약 태스크의 proxy corpus**로 사용하기로 했다.

## 2.3 Assignment 5 – KoBART 요약 모델 학습 및 평가

Assignment 5에서는 HuggingFace의 gogamza/kobart-base-v2를 기반으로 네이버 뉴

스 요약 데이터에 대해 KoBART 한국어 요약 모델을 미세조정(fine-tuning) 했다.

- 데이터 전처리 및 분할:
  - Train/Validation/Test = 8: 1: 1 비율
  - 재현성을 위해 seed = 42 고정
- 학습 설정:
  - MAX\_SOURCE\_LENGTH = 512, MAX\_TARGET\_LENGTH = 128
  - BATCH\_SIZE = 4, NUM\_TRAIN\_EPOCHS = 4, LEARNING\_RATE = 3e-5
- 모델 저장:
  - 학습 완료 후 OUTPUT\_DIR에 모델과 토크나이저 저장 → 이후 evaluation.ipynb, inference.ipynb, 그리고 이번 Assignment 6의 boncahier\_service.py에서 재사용 가능하도록 구조화

평가는 ROUGE-1, ROUGE-2, ROUGE-L 지표를 중심으로 이루어졌으며, 기본 KoBART 모델 대비 한국어 뉴스 요약 품질이 향상되는 것을 목표로 하였다.

지표	값
eval_loss	0.505
ROUGE-1	35.62
ROUGE-2	14.46
ROUGE-L	34.99
ROUGE-Lsum	44.90

### 3. Assignment 6 – 모델 서비스화 구조

Assignment 6의 목표는 지금까지 학습한 모델을 실제로 사용할 수 있는 서비스 형태로 만들고, 이를 최소 5회 이상 사용해 본 뒤, 전체 과정을 최종 보고서에 정리하는 것이다.

이번 단계에서 구현한 핵심은 **CLI 기반 요약·번역 도구**인 boncahier\_service.py이다.

#### 3.1 시스템 아키텍처 개요

##### 1. 입력 단계

- 사용자로부터 .pdf 또는 .pptx 파일 경로를 인자로 받는다 (-i/--input).

## 2. 텍스트 추출 단계

- PDF: PyMuPDF(fitz)를 사용하여 각 페이지의 텍스트를 추출
- PPTX: python-pptx를 사용하여 각 슬라이드의 텍스트를 추출

## 3. 언어 감지(간단 heuristic)

- 텍스트 내에 한글(유니코드 범위 \uac00-\ud7a3)이 포함되었는지 확인하여 대략적으로 “한국어 vs 비한국어(주로 영어)”를 구분

## 4. 번역 단계 (옵션)

- 비한국어(주로 영어)로 추정되는 텍스트는 Helsinki-NLP/opus-mt-tc-big-en-ko 모델로 영→한 번역 후 요약
- 한국어 요약 결과는 필요시 Helsinki-NLP/opus-mt-ko-en으로 한→영 번역

## 5. 요약 단계

- KoBART 기반 요약 모델(gogamza/kobart-base-v2 또는 fine-tuned 모델)로 문단 단위 한국어 요약 수행

## 6. 결과 저장

- 각 페이지/슬라이드별로
  - 원문 텍스트
  - 한국어 요약
  - (옵션) 영어 번역 요약
  - 비고(번역 사용 여부, 에러 등)
- 위의 내용을 Markdown 파일(summary\_{파일이름}\_{타임스탬프}.md)로 저장

이 구조를 통해, 하나의 커맨드로 강의 슬라이드 전체에 대한 요약 노트를 얻을 수 있다.

### 3.2 주요 스크립트 설명 – boncahier\_service.py

- extract\_from\_pdf(path)
  - PyMuPDF를 이용해 PDF 각 페이지의 텍스트를 추출하고, {"id": "page\_1", "kind": "page", "text": "..."} 형태 리스트로 반환.

- extract\_from\_pptx(path)
  - python-pptx로 슬라이드별 텍스트를 모아 {"id": "slide\_1", "kind": "slide", "text": "..."} 형태로 반환.
- load\_summarization\_model(model\_dir, model\_name)
  - model\_dir가 존재하면 그 디렉토리(학습 완료된 KoBART)를 사용
  - 없으면 model\_name (기본: gogamza/kobart-base-v2)에서 모델 로드
- summarize\_text(...)
  - KoBART 요약 모델로 한국어 문단 요약 수행
- translate (text, direction, cache)
  - direction이 "en-ko" 또는 "ko-en"인 경우, 적절한 Helsinki-NLP 번역 모델을 로드하여 번역 수행
- process\_file(...)
  - 입력 파일 하나를 처리해서 최종 Markdown 요약 파일을 생성하는 상위 함수
- main ()
  - CLI 인자를 파싱하고, process\_file()을 호출하여 실제로 요약을 수행

#### 4. 실제 사용 결과 (5회 이상)

회차	사용 일 시	입력 파일 이름	파일 유 형	언어	주요 목 표	소감
1	2025-12-06 11:00	201903774한형준15주발표자료.pptx	ppt	한국어	요약	핵심 키워드 중심으로 요약은 나쁘지 않다.
2	2025-12-06 11:50	ml4e-lecture-week13.pdf	pdf	영어	요약&번역	영어->한국어 번역에 심각한

						오류가 있다.
3	2025-12-06 12:03	2025fall_7.DNN과 RNN.pdf	pdf	한국어	요약&번역	한국어->영어 번역에도 다소 오류가 있으나 영어->한국어 번역에 비해 심각하지 않음
4	2025-12-06 12:30	NLP_Lec3&4_NLP_TM_slides.pdf	pdf	영어	요약	번역에 비해 핵심 키워드 중심을 찾지만 전처리 과정이 필요함
5	2025-12-06 12:40	PythonReview.pdf	pdf	영어	요약	프로그래밍 언어가 포함된 슬라이드는 오류가 다수 검출됨

스크린샷:

1회차:

# 4번 퀸 - 강의 PPTX/PDF 업로드 후 실행 (업로드 방식)  
from google.colab import files  
uploaded = files.upload()  
# 여기서 .pdf 또는 .pptx 파일 선택

2019037747번은 15주별자료.pdf  
2019037747번은 15주별자료.pptx

# 딕셔너리 형태의 파일명 확인  
list(uploaded.keys())

[1] 1'2019037747번은 15주별자료.pptx'

# 파일 이름 꺼내고 아래와 같이 실행하기  
input\_filenames = list(uploaded.keys())[0] # 첫 번째 업로드 파일  
input\_path = f'{content}/content/{input\_filenames}'  
print(input\_path)  
# python /content/initial/benchmark\_service.py  
# -i "{input\_path}"  
# --start-page 1  
# --end-page 21  
# --no-translate  
# -o "(0000).pdf"

-- INPUT: /content/2019037747번은 15주별자료.pptx

[Serialization] Loading model from: /content/drive/MyDrive/benchmark/models/robot\_ko.nms  
[Serialization] Loading model from: /content/drive/MyDrive/benchmark/models/robot\_en.nms  
2025-12-06 02:04:01.253189 External([local\_xla/xla/stream\_executor/cuda/cuda\_ifc.cc:46]) Unable to register cufft factory: Attempting to register factory for plugin cufft when one has already been registered  
WARNING: The logs messages before absl::InitializeFlag() is called are written to STDERR  
0000 00:00:00.764995641 28312 2151 cuda\_bias.cc:[407] Unable to register cuDAS factory: Attempting to register factory for plugin cuDNN when one has already been registered  
0000 00:00:00.764995641 300752 2151 computation\_iplace.cc:[177] computation placer already registered. Please check linkage and avoid linking the same target more than once.  
0000 00:00:00.764995641 300752 2151 computation\_iplace.cc:[177] computation placer already registered. Please check linkage and avoid linking the same target more than once.  
0000 00:00:00.764995641 300752 2151 computation\_iplace.cc:[177] computation placer already registered. Please check linkage and avoid linking the same target more than once.  
0000 00:00:00.764995641 300802 2151 computation\_iplace.cc:[177] computation placer already registered. Please check linkage and avoid linking the same target more than once.  
2025-12-06 02:04:01.306928 [TensorFlow/core/platform/cpu\_feature\_guard.cc:20] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.  
[TensorFlow/core/platform/cpu\_feature\_guard.cc:20] To disable this behavior, set the TF\_CUDNN\_DISABLE\_MKL environment variable to 1.  
# 만약 파일이 한글로 되었습니다. 결과 파일 : /content/drive/MyDrive/benchmark/outputs/summary\_2019037747번은 15주별자료\_20251206-020417.ad

[1] # 맨 끝 결과 확인  
import glob, os  
os

마지막 단계 준비 도구

6thAssignment.ipynb - Colab

outputs - Google Drive

한국외국어 대학교 e-Class Sy

HUFS-LAI-MLAE-2025-2/assig

(101) Megachurches are h

```

# 파일 이름 끝과 아래와 같이 일치하기
1 input_fnames = list(uploaded.keys())[:10] # 첫 번째 몇개도 파일
2 input_paths = ["/content/{}".format(i) for i in input_fnames]
3 print("input_fnames:", input_fnames)
4 print("input_paths:", input_paths)
5

6 python /content/finetune/boncailier_service.py \
7 -i /content/outputs \
8 --model-dir "/content/drive/MyDrive/boncailier" \
9 --start-page 61 \
10 --end-page 71 \
11 --o "/output/DIR"

... INPUT: /content/all4e/lecture-week13.pdf
[SerializedModelConfig] Model config file is located at /content/drive/MyDrive/boncailier/models/kobart_ko_news
You passed max_label_size which is incompatible to the 'label' size of length 2.
2025-12-06 02:51:40.14963: E external/xla/xla/stream_executor/cuda/cuda_ltt.cc:407] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has already been registered
WARNING: All the log messages before absl::InitializeLog() is called are written to STDERR
0000 00:00:17649500.175749 [3999 computation_placer.cc:1427] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered
0000 00:00:17649500.194247 [3999 computation_placer.cc:1771] computation placer already registered. Please check linkage and avoid linking the same target more than once.
0000 00:00:17649500.194247 [3999 computation_placer.cc:1771] computation placer already registered. Please check linkage and avoid linking the same target more than once.
0000 00:00:17649500.194296 [3999 computation_placer.cc:1771] computation placer already registered. Please check linkage and avoid linking the same target more than once.
0000 00:00:17649500.194300 [3999 computation_placer.cc:1771] computation placer already registered. Please check linkage and avoid linking the same target more than once.
2025-12-06 02:51:40.159717: I tensorflow/compiler/mlir/tensorflow/cu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable MLIR-based tensor conversion and other optimizations, rebuild TensorFlow with the appropriate compiler flags.
[tokenizer] Loading model: Helsinki-NLP/opus-xtc-biafra-en (direction:en)
tokenizer_config.json: 100x 341/341 [0.00-0.00-0.00, 2.29Ms/s]
sentencepiece.bpe.model: 100x 100000000 [0.00-0.00-0.00, 740kB/s]
target_spk: 100x 819/819 [0.00-0.00-0.00, 740kB/s]
vocab.json: 9594 [0.00-0.00-0.00, 32.0Ms/s]
sentencepiece.bpe.model: 100x 100000000 [0.00-0.00-0.00, 740kB/s]
config.json: 1.72Ms [0.00-0.00-0.00, 5.39Ms/s]
0 1000 418/418 [0.00-0.00-0.00, 16Ms/s]
model_safetensors: 0 1000 292/293 [0.00-0.00-0.00, 2.19Ms/s]
[tokenizer] Loading model: Helsinki-NLP/opus-xtc-biafra-en (direction:ko-en)
tokenizer_config.json: 100x 44/44.0 [0.00-0.00-0.00, 348kB/s]
config.json: 1.39Ms [0.00-0.00, 7.59Ms/s]
sentencepiece.bpe.model: 100x 100000000 [0.00-0.00-0.00, 35.1Ms/s]
target_spk: 100x 819/819 [0.00-0.00-0.00, 121Ms/s]
vocab.json: 1.72Ms [0.00-0.00-0.00, 5590kB]
sentencepiece.bpe.model: 100x 100000000 [0.00-0.00-0.00, 112Ms/s]
model_safetensors: 0 0.00/312W [0.00-0.00-0.00, 78Ms]
generation_config.json: 100x 293/293 [0.00-0.00-0.00, 2.46Ms/s]
model_safetensors: 100x 312W/312W [0.00-0.00-0.00, 79.36Ms]
[오류] 편집되었었습니다. 결과 파일은 /content/drive/MyDrive/boncailier/outputs/summary_all4e-lecture-week13.20251206-025148.ko
```

AM 11:53 T4(Python 3)

4°C 초원 오전 11:53 2025-12-06

마지막 단계 준비 도구

6thAssignment.ipynb - Colab

outputs - Google Drive

한국외국어 대학교 e-Class Sy

HUFS-LAI-MLAE-2025-2/assig

(101) Megachurches are h

```

# 파일 열어보기
1 with open(files[0], "r", encoding="utf-8") as f:
2     print(f.read()) # 일부만 확인

... ## 한국어 요약 결과

작은 능력 그들은 익기 인기 요리 푸드 하나와 침대 앤 d ph . 혼다 일식 bombard.crainebard craina p 혼다 일 양 양 maxi .혼다 일 abard는 베기 인기 인기 요리 푸드명 하나와 앤으로 d, P D T N 로 악관, 악관, 악사 능력 그들은
## (음선) 있어 번역 요청
Indecencies, they have a tab doh.doh with an instrument-like transparency and a bonds-like bombard. crainada-like maxi. abard is a tab with a white-fledded dish, PTN with a tab, an apothecary, and an apothecary.

## 비고
- X 원문이 비한국어로 추정되어 'en→ko' 번역 후 요약을 수행했습니다.

...
## Unit 7 - page_57 (page)
## 원본 텍스트
...
Adding nonlinearities in self-attention
* Note that there are no element-wise
nonlinearities in self-attention;
* self-attention layers just
re-verages value vectors.
* Easy fix: add a feed-forward network to
post-process each output vector
* min_MLPoutput:
* = W2 * RelU(W1 * output + b2
https://arxiv.org/pdf/1706.07920.pdf
Attention is All You Need (https://arxiv.org/pdf/1706.07920.pdf)

... ## 한국어 요약 결과

웹 사이트는 귀하가 웹 사이트를 탐색하는 동안 귀하의 경험을 향상시키기 위해 주기를 사용하는데 이 주기를 통해서 할로윈 마인 블루 원 주기는 웹 사이트의 기본적인 기능을 수행하는 데 필수적이므로 보관우편에 저장되어 또한 웹 사이트가 사용 방식을 분석하고
## (음선) 있어 번역 요청
The web site uses cookies to improve your experience while you are browsing for web sites, which are required to perform the basic features of web sites, so that cookies are stored in the browser, as well as using my third cookie, which helps the s
## 비고
- X 원문이 비한국어로 추정되어 'en→ko' 번역 후 요약을 수행했습니다.

...

```

AM 11:53 T4(Python 3)

4°C 초원 오전 11:55 2025-12-06

3회차:

The screenshot shows a Google Colab notebook titled "6thAssignment.ipynb". The code cell contains Python code for generating a PDF from uploaded files and saving it as "2025fall\_7.DNN.pdf". The output cell shows the generated PDF file. A summary cell at the bottom displays the generated PDF.

```
# # 4번 폴 - 경로 PDF 파일을 추 실행 (업로드 받식)
# !from google.colab import files
# !uploaded = files.upload() # 여기서 .pdf 또는 .ptpx 파일 선택

# [파일] 2025fall_7.DNN.pdf
# 2025fall_7.DNN.pdf[application/pdf] - 1570373 bytes, last modified: 2025. 12. 6. - 100% done
Saving 2025fall_7.DNN.pdf to 2025fall_7.DNN.pdf

# # 딕셔너리 형태의 파일명 확인
# !list(uploaded.keys())
# ... ['2025fall_7.DNN.pdf']

# # 파일 이름 등록 아래와 같이 실행하기
# !input_filename = !list(uploaded.keys())[0] # 첫 번째 업로드 파일
# !output_filename = input_filename.replace('.pdf', '.txt')
# !print('INPUT:', input_path)
# !print('OUTPUT:', output_path)

# !python content/2025fall_7.DNN/boncaker_service.py \
#     --input-dir '!{input_dir}' \
#     --model-dir '!{MODEL_DIR}' \
#     --start-page 3 \
#     --end-page 20 \
#     --o '!{OUTPUTDIR}'
```

마지막 단계 준비 도구

6thAssignment.ipynb - Colab

Google Drive

한국외국어 대학교 e-Class Sy

HUFS-LAI-MAE-2025-2/assig

(102) Vndeopop is a Bad Person

(102) バーチャルアート

파일 수정 보기 삽입 터미널 도구 도움말

Q. 쟁쟁어 + 코드 + 텍스트 > 모두 실행

```

# ④ 번째 파일 - 각의 PPTX/PDF 업로드 후 실행 (업로드 방식)
1 from google.colab import files
2 uploaded = files.upload() # 여기서 .pdf 또는 .pptx 파일 선택
3
4 uploaded.keys()
5
[{"NLP_Lec384_NLP_TM_slides.pdf": "NLP_Lec384_NLP_TM_slides.pdf"}]
6
7 # 딕셔너리 형태의 파일명 확인
8 len(uploaded.keys())
9
10 ['NLP_Lec384_NLP_TM_slides.pdf']
11
12
13 # 파일 이름 풀고 아래와 같이 실행하기
14 input_filename = list(uploaded.keys())[0] # 첫 번째 업로드 파일
15
16 uploaded_path = f"/content/{input_filename}"
17 print(f"Uploading file {input_filename} to {uploaded_path}")
18
19 #!/usr/bin/python
20 #!/usr/bin/python3
21 #!/usr/bin/python2
22
23 #!/bin/bash
24
25 #!/bin/python
26
27 #!/bin/python3
28
29 #!/bin/python2
30
31 #!/usr/bin/python
32
33 #!/usr/bin/python3
34
35 #!/usr/bin/python2
36
37 #!/bin/bash
38
39 #!/bin/python
40
41 #!/bin/python3
42
43 #!/bin/python2
44
45 #!/usr/bin/python
46
47 #!/usr/bin/python3
48
49 #!/usr/bin/python2
50
51 #!/bin/bash
52
53 #!/bin/python
54
55 #!/bin/python3
56
57 #!/bin/python2
58
59 #!/usr/bin/python
60
61 #!/usr/bin/python3
62
63 #!/usr/bin/python2
64
65 #!/bin/bash
66
67 #!/bin/python
68
69 #!/bin/python3
70
71 #!/bin/python2
72
73 #!/usr/bin/python
74
75 #!/usr/bin/python3
76
77 #!/usr/bin/python2
78
79 #!/bin/bash
80
81 #!/bin/python
82
83 #!/bin/python3
84
85 #!/bin/python2
86
87 #!/usr/bin/python
88
89 #!/usr/bin/python3
90
91 #!/usr/bin/python2
92
93 #!/bin/bash
94
95 #!/bin/python
96
97 #!/bin/python3
98
99 #!/bin/python2
100
101 #!/usr/bin/python
102
103 #!/usr/bin/python3
104
105 #!/usr/bin/python2
106
107 #!/bin/bash
108
109 #!/bin/python
110
111 #!/bin/python3
112
113 #!/bin/python2
114
115 #!/usr/bin/python
116
117 #!/usr/bin/python3
118
119 #!/usr/bin/python2
120
121 #!/bin/bash
122
123 #!/bin/python
124
125 #!/bin/python3
126
127 #!/bin/python2
128
129 #!/usr/bin/python
130
131 #!/usr/bin/python3
132
133 #!/usr/bin/python2
134
135 #!/bin/bash
136
137 #!/bin/python
138
139 #!/bin/python3
140
141 #!/bin/python2
142
143 #!/usr/bin/python
144
145 #!/usr/bin/python3
146
147 #!/usr/bin/python2
148
149 #!/bin/bash
150
151 #!/bin/python
152
153 #!/bin/python3
154
155 #!/bin/python2
156
157 #!/usr/bin/python
158
159 #!/usr/bin/python3
160
161 #!/usr/bin/python2
162
163 #!/bin/bash
164
165 #!/bin/python
166
167 #!/bin/python3
168
169 #!/bin/python2
170
171 #!/usr/bin/python
172
173 #!/usr/bin/python3
174
175 #!/usr/bin/python2
176
177 #!/bin/bash
178
179 #!/bin/python
180
181 #!/bin/python3
182
183 #!/bin/python2
184
185 #!/usr/bin/python
186
187 #!/usr/bin/python3
188
189 #!/usr/bin/python2
190
191 #!/bin/bash
192
193 #!/bin/python
194
195 #!/bin/python3
196
197 #!/bin/python2
198
199 #!/usr/bin/python
200
201 #!/usr/bin/python3
202
203 #!/usr/bin/python2
204
205 #!/bin/bash
206
207 #!/bin/python
208
209 #!/bin/python3
210
211 #!/bin/python2
212
213 #!/usr/bin/python
214
215 #!/usr/bin/python3
216
217 #!/usr/bin/python2
218
219 #!/bin/bash
220
221 #!/bin/python
222
223 #!/bin/python3
224
225 #!/bin/python2
226
227 #!/usr/bin/python
228
229 #!/usr/bin/python3
230
231 #!/usr/bin/python2
232
233 #!/bin/bash
234
235 #!/bin/python
236
237 #!/bin/python3
238
239 #!/bin/python2
240
241 #!/usr/bin/python
242
243 #!/usr/bin/python3
244
245 #!/usr/bin/python2
246
247 #!/bin/bash
248
249 #!/bin/python
250
251 #!/bin/python3
252
253 #!/bin/python2
254
255 #!/usr/bin/python
256
257 #!/usr/bin/python3
258
259 #!/usr/bin/python2
260
261 #!/bin/bash
262
263 #!/bin/python
264
265 #!/bin/python3
266
267 #!/bin/python2
268
269 #!/usr/bin/python
270
271 #!/usr/bin/python3
272
273 #!/usr/bin/python2
274
275 #!/bin/bash
276
277 #!/bin/python
278
279 #!/bin/python3
280
281 #!/bin/python2
282
283 #!/usr/bin/python
284
285 #!/usr/bin/python3
286
287 #!/usr/bin/python2
288
289 #!/bin/bash
290
291 #!/bin/python
292
293 #!/bin/python3
294
295 #!/bin/python2
296
297 #!/usr/bin/python
298
299 #!/usr/bin/python3
300
301 #!/usr/bin/python2
302
303 #!/bin/bash
304
305 #!/bin/python
306
307 #!/bin/python3
308
309 #!/bin/python2
310
311 #!/usr/bin/python
312
313 #!/usr/bin/python3
314
315 #!/usr/bin/python2
316
317 #!/bin/bash
318
319 #!/bin/python
320
321 #!/bin/python3
322
323 #!/bin/python2
324
325 #!/usr/bin/python
326
327 #!/usr/bin/python3
328
329 #!/usr/bin/python2
330
331 #!/bin/bash
332
333 #!/bin/python
334
335 #!/bin/python3
336
337 #!/bin/python2
338
339 #!/usr/bin/python
340
341 #!/usr/bin/python3
342
343 #!/usr/bin/python2
344
345 #!/bin/bash
346
347 #!/bin/python
348
349 #!/bin/python3
350
351 #!/bin/python2
352
353 #!/usr/bin/python
354
355 #!/usr/bin/python3
356
357 #!/usr/bin/python2
358
359 #!/bin/bash
360
361 #!/bin/python
362
363 #!/bin/python3
364
365 #!/bin/python2
366
367 #!/usr/bin/python
368
369 #!/usr/bin/python3
370
371 #!/usr/bin/python2
372
373 #!/bin/bash
374
375 #!/bin/python
376
377 #!/bin/python3
378
379 #!/bin/python2
380
381 #!/usr/bin/python
382
383 #!/usr/bin/python3
384
385 #!/usr/bin/python2
386
387 #!/bin/bash
388
389 #!/bin/python
390
391 #!/bin/python3
392
393 #!/bin/python2
394
395 #!/usr/bin/python
396
397 #!/usr/bin/python3
398
399 #!/usr/bin/python2
400
401 #!/bin/bash
402
403 #!/bin/python
404
405 #!/bin/python3
406
407 #!/bin/python2
408
409 #!/usr/bin/python
410
411 #!/usr/bin/python3
412
413 #!/usr/bin/python2
414
415 #!/bin/bash
416
417 #!/bin/python
418
419 #!/bin/python3
420
421 #!/bin/python2
422
423 #!/usr/bin/python
424
425 #!/usr/bin/python3
426
427 #!/usr/bin/python2
428
429 #!/bin/bash
430
431 #!/bin/python
432
433 #!/bin/python3
434
435 #!/bin/python2
436
437 #!/usr/bin/python
438
439 #!/usr/bin/python3
440
441 #!/usr/bin/python2
442
443 #!/bin/bash
444
445 #!/bin/python
446
447 #!/bin/python3
448
449 #!/bin/python2
450
451 #!/usr/bin/python
452
453 #!/usr/bin/python3
454
455 #!/usr/bin/python2
456
457 #!/bin/bash
458
459 #!/bin/python
460
461 #!/bin/python3
462
463 #!/bin/python2
464
465 #!/usr/bin/python
466
467 #!/usr/bin/python3
468
469 #!/usr/bin/python2
470
471 #!/bin/bash
472
473 #!/bin/python
474
475 #!/bin/python3
476
477 #!/bin/python2
478
479 #!/usr/bin/python
480
481 #!/usr/bin/python3
482
483 #!/usr/bin/python2
484
485 #!/bin/bash
486
487 #!/bin/python
488
489 #!/bin/python3
490
491 #!/bin/python2
492
493 #!/usr/bin/python
494
495 #!/usr/bin/python3
496
497 #!/usr/bin/python2
498
499 #!/bin/bash
500
501 #!/bin/python
502
503 #!/bin/python3
504
505 #!/bin/python2
506
507 #!/usr/bin/python
508
509 #!/usr/bin/python3
510
511 #!/usr/bin/python2
512
513 #!/bin/bash
514
515 #!/bin/python
516
517 #!/bin/python3
518
519 #!/bin/python2
520
521 #!/usr/bin/python
522
523 #!/usr/bin/python3
524
525 #!/usr/bin/python2
526
527 #!/bin/bash
528
529 #!/bin/python
530
531 #!/bin/python3
532
533 #!/bin/python2
534
535 #!/usr/bin/python
536
537 #!/usr/bin/python3
538
539 #!/usr/bin/python2
540
541 #!/bin/bash
542
543 #!/bin/python
544
545 #!/bin/python3
546
547 #!/bin/python2
548
549 #!/usr/bin/python
550
551 #!/usr/bin/python3
552
553 #!/usr/bin/python2
554
555 #!/bin/bash
556
557 #!/bin/python
558
559 #!/bin/python3
560
561 #!/bin/python2
562
563 #!/usr/bin/python
564
565 #!/usr/bin/python3
566
567 #!/usr/bin/python2
568
569 #!/bin/bash
570
571 #!/bin/python
572
573 #!/bin/python3
574
575 #!/bin/python2
576
577 #!/usr/bin/python
578
579 #!/usr/bin/python3
580
581 #!/usr/bin/python2
582
583 #!/bin/bash
584
585 #!/bin/python
586
587 #!/bin/python3
588
589 #!/bin/python2
590
591 #!/usr/bin/python
592
593 #!/usr/bin/python3
594
595 #!/usr/bin/python2
596
597 #!/bin/bash
598
599 #!/bin/python
600
601 #!/bin/python3
602
603 #!/bin/python2
604
605 #!/usr/bin/python
606
607 #!/usr/bin/python3
608
609 #!/usr/bin/python2
610
611 #!/bin/bash
612
613 #!/bin/python
614
615 #!/bin/python3
616
617 #!/bin/python2
618
619 #!/usr/bin/python
620
621 #!/usr/bin/python3
622
623 #!/usr/bin/python2
624
625 #!/bin/bash
626
627 #!/bin/python
628
629 #!/bin/python3
630
631 #!/bin/python2
632
633 #!/usr/bin/python
634
635 #!/usr/bin/python3
636
637 #!/usr/bin/python2
638
639 #!/bin/bash
640
641 #!/bin/python
642
643 #!/bin/python3
644
645 #!/bin/python2
646
647 #!/usr/bin/python
648
649 #!/usr/bin/python3
650
651 #!/usr/bin/python2
652
653 #!/bin/bash
654
655 #!/bin/python
656
657 #!/bin/python3
658
659 #!/bin/python2
660
661 #!/usr/bin/python
662
663 #!/usr/bin/python3
664
665 #!/usr/bin/python2
666
667 #!/bin/bash
668
669 #!/bin/python
670
671 #!/bin/python3
672
673 #!/bin/python2
674
675 #!/usr/bin/python
676
677 #!/usr/bin/python3
678
679 #!/usr/bin/python2
680
681 #!/bin/bash
682
683 #!/bin/python
684
685 #!/bin/python3
686
687 #!/bin/python2
688
689 #!/usr/bin/python
690
691 #!/usr/bin/python3
692
693 #!/usr/bin/python2
694
695 #!/bin/bash
696
697 #!/bin/python
698
699 #!/bin/python3
700
701 #!/bin/python2
702
703 #!/usr/bin/python
704
705 #!/usr/bin/python3
706
707 #!/usr/bin/python2
708
709 #!/bin/bash
710
711 #!/bin/python
712
713 #!/bin/python3
714
715 #!/bin/python2
716
717 #!/usr/bin/python
718
719 #!/usr/bin/python3
720
721 #!/usr/bin/python2
722
723 #!/bin/bash
724
725 #!/bin/python
726
727 #!/bin/python3
728
729 #!/bin/python2
730
731 #!/usr/bin/python
732
733 #!/usr/bin/python3
734
735 #!/usr/bin/python2
736
737 #!/bin/bash
738
739 #!/bin/python
740
741 #!/bin/python3
742
743 #!/bin/python2
744
745 #!/usr/bin/python
746
747 #!/usr/bin/python3
748
749 #!/usr/bin/python2
750
751 #!/bin/bash
752
753 #!/bin/python
754
755 #!/bin/python3
756
757 #!/bin/python2
758
759 #!/usr/bin/python
760
761 #!/usr/bin/python3
762
763 #!/usr/bin/python2
764
765 #!/bin/bash
766
767 #!/bin/python
768
769 #!/bin/python3
770
771 #!/bin/python2
772
773 #!/usr/bin/python
774
775 #!/usr/bin/python3
776
777 #!/usr/bin/python2
778
779 #!/bin/bash
780
781 #!/bin/python
782
783 #!/bin/python3
784
785 #!/bin/python2
786
787 #!/usr/bin/python
788
789 #!/usr/bin/python3
790
791 #!/usr/bin/python2
792
793 #!/bin/bash
794
795 #!/bin/python
796
797 #!/bin/python3
798
799 #!/bin/python2
800
801 #!/usr/bin/python
802
803 #!/usr/bin/python3
804
805 #!/usr/bin/python2
806
807 #!/bin/bash
808
809 #!/bin/python
810
811 #!/bin/python3
812
813 #!/bin/python2
814
815 #!/usr/bin/python
816
817 #!/usr/bin/python3
818
819 #!/usr/bin/python2
820
821 #!/bin/bash
822
823 #!/bin/python
824
825 #!/bin/python3
826
827 #!/bin/python2
828
829 #!/usr/bin/python
830
831 #!/usr/bin/python3
832
833 #!/usr/bin/python2
834
835 #!/bin/bash
836
837 #!/bin/python
838
839 #!/bin/python3
840
841 #!/bin/python2
842
843 #!/usr/bin/python
844
845 #!/usr/bin/python3
846
847 #!/usr/bin/python2
848
849 #!/bin/bash
850
851 #!/bin/python
852
853 #!/bin/python3
854
855 #!/bin/python2
856
857 #!/usr/bin/python
858
859 #!/usr/bin/python3
860
861 #!/usr/bin/python2
862
863 #!/bin/bash
864
865 #!/bin/python
866
867 #!/bin/python3
868
869 #!/bin/python2
870
871 #!/usr/bin/python
872
873 #!/usr/bin/python3
874
875 #!/usr/bin/python2
876
877 #!/bin/bash
878
879 #!/bin/python
880
881 #!/bin/python3
882
883 #!/bin/python2
884
885 #!/usr/bin/python
886
887 #!/usr/bin/python3
888
889 #!/usr/bin/python2
890
891 #!/bin/bash
892
893 #!/bin/python
894
895 #!/bin/python3
896
897 #!/bin/python2
898
899 #!/usr/bin/python
900
901 #!/usr/bin/python3
902
903 #!/usr/bin/python2
904
905 #!/bin/bash
906
907 #!/bin/python
908
909 #!/bin/python3
910
911 #!/bin/python2
912
913 #!/usr/bin/python
914
915 #!/usr/bin/python3
916
917 #!/usr/bin/python2
918
919 #!/bin/bash
920
921 #!/bin/python
922
923 #!/bin/python3
924
925 #!/bin/python2
926
927 #!/usr/bin/python
928
929 #!/usr/bin/python3
930
931 #!/usr/bin/python2
932
933 #!/bin/bash
934
935 #!/bin/python
936
937 #!/bin/python3
938
939 #!/bin/python2
940
941 #!/usr/bin/python
942
943 #!/usr/bin/python3
944
945 #!/usr/bin/python2
946
947 #!/bin/bash
948
949 #!/bin/python
950
951 #!/bin/python3
952
953 #!/bin/python2
954
955 #!/usr/bin/python
956
957 #!/usr/bin/python3
958
959 #!/usr/bin/python2
960
961 #!/bin/bash
962
963 #!/bin/python
964
965 #!/bin/python3
966
967 #!/bin/python2
968
969 #!/usr/bin/python
970
971 #!/usr/bin/python3
972
973 #!/usr/bin/python2
974
975 #!/bin/bash
976
977 #!/bin/python
978
979 #!/bin/python3
980
981 #!/bin/python2
982
983 #!/usr/bin/python
984
985 #!/usr/bin/python3
986
987 #!/usr/bin/python2
988
989 #!/bin/bash
990
991 #!/bin/python
992
993 #!/bin/python3
994
995 #!/bin/python2
996
997 #!/usr/bin/python
998
999 #!/usr/bin/python3
1000
1001 #!/usr/bin/python2
1002
1003 #!/bin/bash
1004
1005 #!/bin/python
1006
1007 #!/bin/python3
1008
1009 #!/bin/python2
1010
1011 #!/usr/bin/python
1012
1013 #!/usr/bin/python3
1014
1015 #!/usr/bin/python2
1016
1017 #!/bin/bash
1018
1019 #!/bin/python
1020
1021 #!/bin/python3
1022
1023 #!/bin/python2
1024
1025 #!/usr/bin/python
1026
1027 #!/usr/bin/python3
1028
1029 #!/usr/bin/python2
1030
1031 #!/bin/bash
1032
1033 #!/bin/python
1034
1035 #!/bin/python3
1036
1037 #!/bin/python2
1038
1039 #!/usr/bin/python
1040
1041 #!/usr/bin/python3
1042
1043 #!/usr/bin/python2
1044
1045 #!/bin/bash
1046
1047 #!/bin/python
1048
1049 #!/bin/python3
1050
1051 #!/bin/python2
1052
1053 #!/usr/bin/python
1054
1055 #!/usr/bin/python3
1056
1057 #!/usr/bin/python2
1058
1059 #!/bin/bash
1060
1061 #!/bin/python
1062
1063 #!/bin/python3
1064
1065 #!/bin/python2
1066
1067 #!/usr/bin/python
1068
1069 #!/usr/bin/python3
1070
1071 #!/usr/bin/python2
1072
1073 #!/bin/bash
1074
1075 #!/bin/python
1076
1077 #!/bin/python3
1078
1079 #!/bin/python2
1080
1081 #!/usr/bin/python
1082
1083 #!/usr/bin/python3
1084
1085 #!/usr/bin/python2
1086
1087 #!/bin/bash
1088
1089 #!/bin/python
1090
1091 #!/bin/python3
1092
1093 #!/bin/python2
1094
1095 #!/usr/bin/python
1096
1097 #!/usr/bin/python3
1098
1099 #!/usr/bin/python2
1100
1101 #!/bin/bash
1102
1103 #!/bin/python
1104
1105 #!/bin/python3
1106
1107 #!/bin/python2
1108
1109 #!/usr/bin/python
1110
1111 #!/usr/bin/python3
1112
1113 #!/usr/bin/python2
1114
1115 #!/bin/bash
1116
1117 #!/bin/python
1118
1119 #!/bin/python3
1120
1121 #!/bin/python2
1122
1123 #!/usr/bin/python
1124
1125 #!/usr/bin/python3
1126
1127 #!/usr/bin/python2
1128
1129 #!/bin/bash
1130
1131 #!/bin/python
1132
1133 #!/bin/python3
1134
1135 #!/bin/python2
1136
1137 #!/usr/bin/python
1138
1139 #!/usr/bin/python3
1140
1141 #!/usr/bin/python2
1142
1143 #!/bin/bash
1144
1145 #!/bin/python
1146
1147 #!/bin/python3
1148
1149 #!/bin/python2
1150
1151 #!/usr/bin/python
1152
1153 #!/usr/bin/python3
1154
1155 #!/usr/bin/python2
1156
1157 #!/bin/bash
1158
1159 #!/bin/python
1160
1161 #!/bin/python3
1162
1163 #!/bin/python2
1164
1165 #!/usr/bin/python
1166
1167 #!/usr/bin/python3
1168
1169 #!/usr/bin/python2
1170
1171 #!/bin/bash
1172
1173 #!/bin/python
1174
1175 #!/bin/python3
1176
1177 #!/bin/python2
1178
1179 #!/usr/bin/python
1180
1181 #!/usr/bin/python3
1182
1183 #!/usr/bin/python2
1184
1185 #!/bin/bash
1186
1187 #!/bin/python
1188
1189 #!/bin/python3
1190
1191 #!/bin/python2
1192
1193 #!/usr/bin/python
1194
1195 #!/usr/bin/python3
1196
1197 #!/usr/bin/python2
1198
1199 #!/bin/bash
1200
1201 #!/bin/python
1202
1203 #!/bin/python3
1204
1205 #!/bin/python2
1206
1207 #!/usr/bin/python
1208
1209 #!/usr/bin/python3
1210
1211 #!/usr/bin/python2
1212
1213 #!/bin/bash
1214
1215 #!/bin/python
1216
1217 #!/bin/python3
1218
1219 #!/bin/python2
1220
1221 #!/usr/bin/python
1222
1223 #!/usr/bin/python3
1224
1225 #!/usr/bin/python2
1226
1227 #!/bin/bash
1228
1229 #!/bin/python
1230
1231 #!/bin/python3
1232
1233 #!/bin/python2
1234
1235 #!/usr/bin/python
1236
1237 #!/usr/bin/python3
1238
1239 #!/usr/bin/python2
1240
1241 #!/bin/bash
1242
1243 #!/bin/python
1244
1245 #!/bin/python3
1246
1247 #!/bin/python2
1248
1249 #!/usr/bin/python
1250
1251 #!/usr/bin/python3
1252
1253 #!/usr/bin/python2
1254
1255 #!/bin/bash
1256
1257 #!/bin/python
1258
1259 #!/bin/python3
1260
1261 #!/bin/python2
1262
1263 #!/usr/bin/python
1264
1265 #!/usr/bin/python3
1266
1267 #!/usr/bin/python2
1268
1269 #!/bin/bash
1270
1271 #!/bin/python
1272
1273 #!/bin/python3
1274
1275 #!/bin/python2
1276
1277 #!/usr/bin/python
1278
1279 #!/usr/bin/python3
1280
1281 #!/usr/bin/python2
1282
1283 #!/bin/bash
1284
1285 #!/bin/python
1286
1287 #!/bin/python3
1288
1289 #!/bin/python2
1290
1291 #!/usr/bin/python
1292
1293 #!/usr/bin/python3
1294
1295 #!/usr/bin/python2
1296
1297 #!/bin/bash
1298
1299 #!/bin/python
1300
1301 #!/bin/python3
1302
1303 #!/bin/python2
1304
1305 #!/usr/bin/python
1306
1307 #!/usr/bin/python3
1308
1309 #!/usr/bin/python2
1310
1311 #!/bin/bash
1312
1313 #!/bin/python
1314
1315 #!/bin/python3
1316
1317 #!/bin/python2
1318
1319 #!/usr/bin/python
1320
1321 #!/usr/bin/python3
1322
1323 #!/usr/bin/python2
1324
1325 #!/bin/bash
1326
1327 #!/bin/python
1328
1329 #!/bin/python3
1330
1331 #!/bin/python2
1332
1333 #!/usr/bin/python
1334
1335 #!/usr/bin/python3
1336
1337 #!/usr/bin/python2
1338
1339 #!/bin/bash
1340
1341 #!/bin/python
1342
1343 #!/bin/python3
1344
1345 #!/bin/python2
1346
1347 #!/usr/bin/python
1348
1349 #!/usr/bin/python3
1350
1351 #!/usr/bin/python2
1352
1353 #!/bin/bash
1354
1355 #!/bin/python
1356
1357 #!/bin/python3
1358
1359 #!/bin/python2
1360
1361 #!/usr/bin/python
1362
1363 #!/usr/bin/python3
1364
1365 #!/usr/bin/python2
1366
1367 #!/bin/bash
1368
1369 #!/bin/python
1370
1371 #!/bin/python3
1372
1373 #!/bin/python2
1374
1375 #!/usr/bin/python
1376
1377 #!/usr/bin/python3
1378
1379 #!/usr/bin/python2
1380
1381 #!/bin/bash
1382
1383 #!/bin/python
1384
1385 #!/bin/python3
1386
1387 #!/bin/python2
1388
1389 #!/usr/bin/python
1390
1391 #!/usr/bin/python3
1392
1393 #!/usr/bin/python2
1394
1395 #!/bin/bash
1396
1397 #!/bin/python
1398
1399 #!/bin/python3
1400
1401 #!/bin/python2
1402
1403 #!/usr/bin/python
1404
1405 #!/usr/bin/python3
1406
1407 #!/usr/bin/python2
1408
1409 #!/bin/bash
1410
1411 #!/bin/python
1412
1413 #!/bin/python3
1414
1415 #!/bin/python2
1416
1417 #!/usr/bin/python
1418
1419 #!/usr/bin/python3
1420
1421 #!/usr/bin/python2
1422
1423 #!/bin/bash
1424
1425 #!/bin/python
1426
1427 #!/bin/python3
1428
1429 #!/bin/python2
1430
1431 #!/usr/bin/python
1432
1433 #!/usr/bin/python3
1434
1435 #!/usr/bin/python2
1436
1437 #!/bin/bash
1438
1439 #!/bin/python
1440
1441 #!/bin/python3
1442
1443 #!/bin/python2
1444
1445 #!/usr/bin/python
1446
1447 #!/usr/bin/python3
1448
1449 #!/usr/bin/python2
1450
1451 #!/bin/bash
1452
1453 #!/bin/python
1454
1455 #!/bin/python3
1456
1457 #!/bin/python2
1458
1459 #!/usr/bin/python
1460
1461 #!/usr/bin/python3
1462
1463 #!/usr/bin/python2
1464
1465 #!/bin/bash
1466
1467 #!/bin/python
1468
1469 #!/bin/python3
1470
1471 #!/bin/python2
1472
1473 #!/usr/bin/python
1474
1475 #!/usr/bin/python3
1476
1477 #!/usr/bin/python2
1478
1479 #!/bin/bash
1480
1481 #!/bin/python
1482
1483 #!/bin/python3
1484
1485 #!/bin/python2
1486
1487 #!/usr/bin/python
1488
1489 #!/usr/bin/python3
1490
1491 #!/usr/bin/python2
1492
1493 #!/bin/bash
1494
1495 #!/bin/python
1496
1497 #!/bin/python3
1498
1499 #!/bin/python2
1500
1501 #!/usr/bin/python
1502
1503 #!/usr/bin/python3
1504
1505 #!/usr/bin/python2
1506
1507 #!/bin/bash
1508
1509 #!/bin/python
1510
1511 #!/bin/python3
1512
1513 #!/bin/python2
1514
1515 #!/usr/bin/python
1516
1517 #!/usr/bin/python3
1518
1519 #!/usr/bin/python2
1520
1521 #!/bin/bash
1522
1523 #!/bin/python
1524
1525 #!/bin/python3
1526
1527 #!/bin/python2
1528
1529 #!/usr/bin/python
1530
1531 #!/usr/bin/python3
1532
1533 #!/usr/bin/python2
1534
1535 #!/bin/bash
1536
1537 #!/bin/python
1538
1539 #!/bin/python3
1540
1541 #!/bin/python2
1542
1543 #!/usr/bin/python
1544
1545 #!/usr/bin/python3
1546
1547 #!/usr/bin/python2
1548
1549 #!/bin/bash
1550
1551 #!/bin/python
1552
1553 #!/bin/python3
1554
1555 #!/bin/python2
1556
1557 #!/usr/bin/python
1558
1559 #!/usr/bin/python3
1560
1561 #!/usr/bin/python2
1562
1563 #!/bin/bash
1564
1565 #!/bin/python
1566
1567 #!/bin/python3
1568
1569 #!/bin/python2
1570
1571 #!/usr/bin/python
1572
1573 #!/usr/bin/python3
1574
1575 #!/usr/bin/python2
1576
1577 #!/bin/bash
1578
1579 #!/bin/python
1580
1581 #!/bin/python3
1582
1583 #!/bin/python2
1584
1585 #!/usr/bin/python
1586
1587 #!/usr/bin/python3
1588
1589 #!/usr/bin/python2
1590
1591 #!/bin/bash
1592
1593 #!/bin/python
1594
1595 #!/bin/python3
1596
1597 #!/bin/python2
159
```

마지막 단계 준비 도구

6thAssignment.ipynb - Colab

outputs - Google Drive

한국외국인 대학교 e-Class Sy...

HUFS-LAI-ML4E-2025-3/assig...

102) Vnotepop is a Bad Per...

(102) ボルシカ - アリジーノ

6thAssignment.ipynb

파일 수정 보기 설정 허락된 도구 드롭박스

문서에 + 코드 + 텍스트 > 모든 설정

RAM 디스크

임시 파일 열기

```
with open('file4', 'r', encoding='utf-8') as f:  
    print(f.read(10000)) # 일본문만 확인
```

## 한국어 예상 결과

Note because <x>0 returns True or False according, the(same)  
the same>: defneis<x>0 if the, simillid, shows, ethatall(<x>  
total()): total()more, alone->including sus.

##

# Unit 16 - page\_17 (page)

### 일본 예상

§ tuplesaredefinedwith(parentheses): (1),(2,(3),(for example,  
• In essence, you can think of a tuple as a list that is not mutable.  
thus, (indexing)worksaswith list objects.  
§ tuplesdetails:  
• tuples(1,(2,(3  
• ((1,100  
• tuple()<x>0 because it would change t.  
§ If you know that you aren't going to change your object(at all),  
then tuple is probably a good choice over list.  
§ The built-in tuple() function will convert its argument to a tuple.  
and tuple will convert its argument to a list.  
§ Examples:  
• <x>(1,(2,(3  
• <x>((1,100))#Throwsanexception.  
• <x>((1,100))#Works fine, unchanged.  
• copy<x>=copy<x>#True  
The copy()&deepcopy()functions  
tuple

##

# Unit 17 - page\_18 (page)

### 일본 예상

## 로그 발췌 예시:

1회차:

## Unit 8 — slide\_10 (slide)

### ### 원본 텍스트

...

인간 대상 연구에서 발견된 공격적인 행동과 전전두엽 전부 기능 장애의 연관성: 반사회적인 개개인에게는 회백질과 혈역학 반응의 감소, 배외측 전전두엽 피질의 왼쪽, 안와 전두 피질 오른쪽, 전방 대상 피질 오른쪽에 지역적인 뇌 혈류가 관찰되었다.<sup>63)</sup>

쥐 대상 연구: 내측 전전두엽 피질의 기능을 억제하면 수컷 쥐 간의 공격성이 증가하나 활성화하면 공격성이 감소하고<sup>69)</sup>, 세로토닌이 감소할 수록 공격성이 증가한다.<sup>70)</sup>

전두엽 손상이 목표에 대한 공격적인 반응과 관련이 있고, 측두엽 손상이 격렬하지만 집중되지 않은 분노와 공격성과 관련이 있다는 가정도 있다.<sup>72)</sup>

배외측 전전두엽 피질의 손상이 자기 제어를 실패하게 하여 분노 조절에 결함을 일으키고 도발 당하는 경우 공격적인 충동으로 이어진다.<sup>73)</sup>

공격적인 행동과 전전두엽 피질 장애의 상호 연관성

...

### **### 한국어 요약 결과**

반사회적인 개개인에게는 회백질과 혈역학 반응의 감소, 배외측 전전두엽 피질의 왼쪽, 안와 전두 피질 오른쪽, 전방 대상 피질 오른쪽에 지역적인 뇌 혈류가 관찰되었다.<sup>73)</sup>

공격적인 행동과 전전 두엽 피질 장애의 상호 연관성: 내측 전 전두엽피질의 기능을 억제하면 수컷 쥐 간의 공격성이 증가하나 활성화하면 공격성이 감소하고<sup>69)</sup>, 세로토닌이 감소할 수록 공격성이 증가한다.<sup>70)</sup>

측두엽 손상이 목표에 대한 공격적인 반응과 관련이

2회차:

**## Unit 7 — page\_67 (page)**

### **### 원본 텍스트**

...

### Adding nonlinearities in self-attention

- Note that there are no element-wise nonlinearities in self-attention;
- stacking more self-attention layers just re-averages value vectors
- Easy fix: add a feed-forward network to post-process each output vector
- $m_i = \text{MLP}_{\text{output}}(i)$
- 

$$= W_2 * \text{ReLU}(W_1 \text{output}_i + b_1) + b_2$$

<https://web.stanford.edu/class/cs224n/>

Attention Is All You Need (<https://arxiv.org/pdf/1706.03762.pdf>)

...

### ### 한국어 요약 결과

웹 사이트는 귀하가 웹 사이트를 탐색하는 동안 귀하의 경험을 향상시키기 위해 쿠키를 사용하는데 이 쿠키들 중에서 필요에 따라 분류 된 쿠키는 웹 사이트의 기본적인 기능을 수행하는 데 필수적이므로 브라우저에 저장되며 또한 웹 사이트가 사용 방식을 분석하고 이해하는 데 도움이되는 제 3 자 쿠키를 사용한다. 이러한 쿠키를 거부 할 수도 있다. 그러나 이러한 쿠키 중 일부를 선택 해제하면 검색 환경에 영향을 미칠 수 있다. 이러한 쿠키들을 선택 해제 하면 검색 환경에 영향 미칠 수도 있다. 이런 쿠키들 중 일부 선택 해제된 쿠키와 제 3자 쿠키를 사용할 수 있다. 또한 이러한 쿠키들은 웹 사이

### ### (옵션) 영어 번역 요약

The web site uses cookies to improve your experience while you are browsing for web sites, which are required to perform the basic features of web sites, so that cookies are stored in the browser, as well as using my third cookie, which helps the site to analyze and understand how they are used.

### ### 비고

- ※ 원문이 비한국어로 추정되어 `en→ko` 번역 후 요약을 수행했습니다.

3회차:

### ## Unit 10 — page\_12 (page)

### ### 원본 텍스트

...

#### PREVIEW

v순환신경망(Recurrent Neural Network)과 LSTM (Long Short-Term

Memory)

♣순환신경망은 시간성정보를 활용하여 순차데이터를 처리하는 효과적인 학습모델

♣매우 긴 순차데이터(예, 30단어 이상의 긴 문장)를 처리하는데에는 장기의존성을 잘다

루는 LSTM을 주로 사용(LSTM은 선별 기억 능력을 가짐)

v최근에는 순환신경망을 생성모델로 사용

♣예, CNN과 LSTM이 협력하여 자연영상에 주석을다는 문제를 풀

...

### ### 한국어 요약 결과

'Rurrent Neural Network)과 LSTM (Long Short-Term Memory)

어에우긴순차데이터(예, 30단어이상의긴문장)를처리하는데에는장기의존성을잘드러내고 최근에는순환신경망을생성모델로사용

한데예, CNN과LSTM이협력하여자연영상에주석을다는문제를풀을문제를풀다는문제를 풀을 는가?

CNN은 CNN의협력하여 CNN에게주석다는문제를함바, CNTM이협력

### **### (옵션) 영어 번역 요약**

When it comes to dealing with long-term dependencies and long-term sequences, which are more than 30 words long-term sentences, CNN andLSTM have recently been used to create a model that creates problems that address the problem of commenting on natural images in cooperation with CNN?

4회차:

### **## Unit 9 — page\_24 (page)**

#### **### 원본 텍스트**

...

What is TM? (cont'd)

☞Text Mining vs. Data Mining

→Text Mining

) Dealing with unstructured textual data

- Free-form text
  - » No pre-defined or organized in any way
- } Requiring an extra step
  - The unstructured data has to be organized and structured in a way that allows the data modeling and analytics to occur
- } Concerned with the detection of patterns in NL texts
  - Data Mining
- } Handling structured (highly formatted) data
- } Combining disciplines including statistics, AI and machine learning to apply directly to structured data
- } Concerned with the detection of patterns in database

Nov. 25 & Dec. 01, 2025

Introduction to Language Technology 2025

24

...

### ### 한국어 요약 결과

'Tith unstructured data modeling and analytics to occur

Concerned with the detection of patterns in NL texts

Data Mining disciplines including statistics, AI and machine and apply directly patted  
daa

'Introd

5호차:

## ## Unit 16 — page\_17 (page)

### ### 원본 텍스트

``

§ Tuples(are(defined(with(parentheses: (1,(2,(3),(for(example.

- In(essence,(you(can(think(of(a tuple as(a list that(is(not(mutable.
- Thus,(indexing(works(as(with list objects.

§ But(assignment(fails:

- t=((1,(2,(3)

t[1]=(100

- This(won't(work(because(it(would(change t.

§ If(you(know(that(you(aren't(going(to(change(your(object(at(all,(

then tuple is(probably(a(good(choice(over list.

§ The(built-in tuple() function(will(convert(its(argument(to(a tuple,(  
and list() will(covert(its(argument(to list.

§ Examples:

§ x=((1,(2,(3)

y=(list(x)

x[1]=(100##(Throws(an(exception.

y[1]=(100##(Works(fine;(changes(y.

tuple(list(x))(==x##(True!

The\$basics\$of\$dicts,\$and\$tuples

**tuple**

---

### ### 한국어 요약 결과

Thus,(indexing(works)as(with part list objects)) But(assignment(fails,  
(1,(2,(3),(for(example,)))) function(will(convert(its(argument(to list)) will(convert, its,\_  
If(won't(wou(that(you(know(thou(

## 5. 배운 점 및 한계, 향후 개선 방향

### 5.1 배운 점

- 데이터 수집-EDA-모델 학습-서비스화까지 하나의 흐름을 직접 경험하면서, "단순한 Colab 실습"이 아니라 실제로 사용 가능한 도구를 만드는 과정을 익혔다.
- 한국어 요약에 특화된 KoBART 모델을 사용해 보면서, **인퍼런스 속도·토큰 길이·Beam Search** 파라미터가 결과 품질과 시간에 어떤 영향을 미치는지 체감할 수 있었다.
- PyMuPDF, python-pptx, transformers, HuggingFace Datasets 등을 함께 사용하면서 실제 ML 프로젝트에서 흔히 쓰이는 라이브러리들의 역할을 이해하게 되었다.

### 5.2 한계

- 학습 데이터가 **뉴스 기사**이기 때문에, 교수님 슬라이드의 레이아웃/요약 스타일과는 약간의 차이가 있다.
- 영어 슬라이드의 경우, en→ko → 한국어 요약 → ko→en 순서로 파이프라인이 구성되어 en→ko 번역이 ko→en 번역보다 오류 가능성이 심각하게 높다.
- 텍스트 추출 품질은 슬라이드의 특징에 따라 달라지므로, 표/이미지로 이루어지거나 수식, 프로그래밍 언어, 웹사이트 링크가 다수 포함된 슬라이드는 요약이 어렵다.

### 5.3 향후 개선 방향

- 강의 슬라이드 전용 데이터(예: 실제 PPT 텍스트 + 요약 노트)를 구축하여

KoBART를 **슬라이드 도메인에 특화된 모델로 재학습해 볼 수 있다.**

- 웹 UI(예: Streamlit, Flask)를 없어서, 명령어에 익숙하지 않은 사용자도 **드래그 앤 드롭으로 사용할 수 있는 서비스**로 확장 가능하다.
- 번역 부분은 HuggingFace의 다른 번역 모델이나, 필요하다면 외부 API(예: DeepL, Google Translate)를 조합해 품질을 끌어올릴 수 있다.
- 더욱 효율적이고 성능 높은 영어 데이터 및 모델을 다시 선정하여 en→ko 번역의 품질을 향상시킬 수 있다.

## 6. 결론

본 프로젝트는 **강의 자료 요약·번역 도구 BonCahier AI**를 기획(Assignment 3), 데이터 수집 및 분석(Assignment 4), KoBART 모델 학습 및 평가(Assignment 5), 그리고 실제 사용 가능한 CLI 서비스로 구현(Assignment 6)하는 전 과정을 다루었다.

그 결과, PDF/PPTX 형식의 수업 자료를 입력으로 받아 **슬라이드/페이지 단위로 핵심 내용을 요약하고 번역해 주는 실질적인 도구**를 완성하였고, 이를 실제 강의 자료에 적용하여 **시험 대비용 요약 노트 자동 생성**이라는 초기 목표를 달성할 수 있었다.