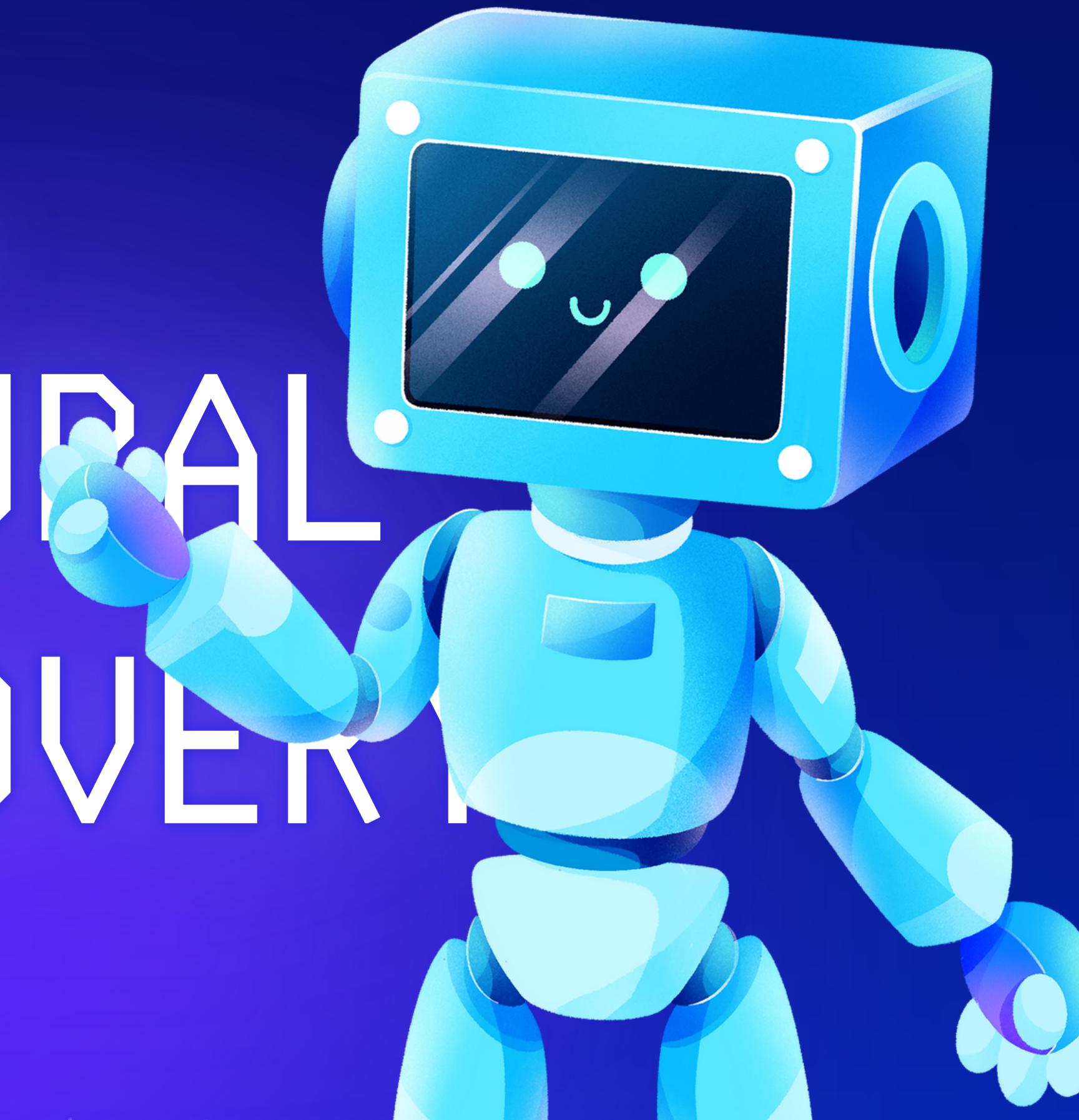




ARTIFICIAL INTELLIGENCE



NATURAL DISCOVERY

By Alexander Aronowitz



TABLE OF CONTENTS

• 주제 선정 배경	01
• 역할 분담	02
• 이시명(말뭉치 기반 다국어 번역 프로그램)	03
• 변주영(아마존 독서 리뷰 요약)	04
• 양현우(호불호 분석 프로그램)	05
• 명노아(잠뜰TV 댓글 생성 프로그램)	06
• 결론 & 시연	07



주제 선정 배경

자연어 처리 분석 비중의 증가

+

트랜스포머 학습

+

다양한 기능 제공 소프트웨어 개발

-> NLP Secretary(자연어처리 비서)

이때까지 배운 걸 써먹자!



You

일상생활에 스며드는 자연어 처리 복합 모델



ChatGPT

일상생활에서 스며드는 자연어 처리(NLP) 복합 모델은 우리의 언어를 이해하고 상호작용하는 데 사용됩니다. 이러한 모델은 다양한 기술과 기능을 통합하여 다음과 같은 다양한 사용 사례에 적용될 수 있습니다:

1. **가상 비서 및 개인화된 서비스:** 스마트폰이나 스마트 홈 기기와 같은 장치에서 음성 인식 기능을 통해 사용자의 음성 명령을 인식하고 이에 대한 응답을 생성합니다. 이러한 모델은 사용자의 의도를 이해하고, 적절한 작업을 수행하기 위해 다양한 NLP 기술을 활용합니다.

2. **인공지능 번역 및 다국어 커뮤니케이션:** 다국어 번역 모델은 다양한 언어 간의 텍스트를 자동으로 번역하여 사용자들이 서로 다른 언어로 소통할 수 있도록 돕습니다. 이러한 모델은 기계 번역, 언어 감지, 문맥 이해 등의 기술을 결합하여 사용됩니다.

3. **소셜 미디어 감성 분석:** 트위터, 페이스북, 인스타그램 등의 소셜 미디어 플랫폼에서 사용자의 감정을 분석하여 특정 이벤트나 제품에 대한 피드백을 이해하고 추론합니다. 이를 통해 기업은 제품 및 서비스 개선에 활용할 수 있습니다.

4. **문서 요약 및 정보 추출:** 복합 NLP 모델은 긴 문서를 읽고 요약하거나, 특정 정보를 추출하여 요약된 내용을 제공합니다. 이는 뉴스 기사, 보고서, 과학 논문 등과 같은 대량의 텍스트를 처리하는데 유용합니다.

5. **자연어 이해 기반 검색 엔진:** 검색 엔진은 사용자의 검색 쿼리를 이해하고 관련 정보를 검색 결과로 제공하기 위해 자연어 처리 기술을 사용합니다. 이러한 검색 엔진은 사용자의 의도를 파악하고, 관련 검색어를 식별하여 정확한 검색 결과를 제공합니다.

이러한 다양한 사용 사례를 위해 자연어 처리 복합 모델은 텍스트 이해, 음성 인식, 감정 분석, 정보 추출 등의 다양한 기술을 통합하고 발전시켜 나가고 있습니다. 이는 우리의 일상생활에서 더욱 효과적인 소통과 정보 이해를 도모합니다.

역할 분담

	이시영	명노아	변주영	양현우
데이터 수집 & 전처리	✓	✓	✓	✓
모델 생성	✓	✓	✓	✓
전체 웹페이지 구축		✓		✓
개인 페이지 구축	✓	✓	✓	✓
Readme	✓		✓	

말뭉기 기반 다국어 번역기

이지명

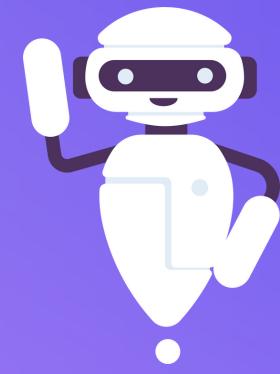


말뭉기 기반 외계어 쌤쌤기

이지명



PROJECT OBJECTIVES



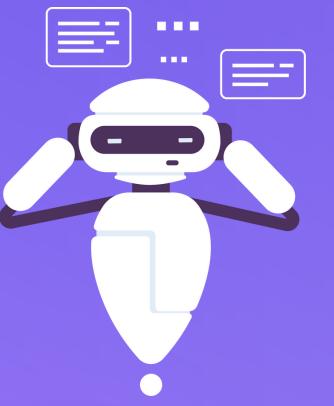
DESTROYED
MODEL

TRANSFORMER
Facebook/bart
Google/bart
전이학습



CREATE MODEL

seq2seq



EXECUTE IN WEB

Flask 와 연결
DB연결

기존 라이브러리 이용하지 않는 모델

AI - HUB 한국어-영어 번역(병렬) 말뭉치 1K ~ 20K

Seq2seq – encoder – decoder

AdamW

22~1000 epochs



TRANSFORMER 있으라

3대의 컴퓨터

3개의 가상환경

12개의 모델



태초에 트랜스포머 모델이 있었다

— 교재

```
Epoch 1: 100%|██████████| 1250/1250 [2:51:29<00:00, 8.23s/batch, accuracy=0.000804, loss=0.0124]
Epoch 2: 100%|██████████| 1250/1250 [2:28:19<00:00, 7.12s/batch, accuracy=0.000402, loss=0.00622]
Epoch 3: 100%|██████████| 1250/1250 [3:37:36<00:00, 10.44s/batch, accuracy=0.000268, loss=0.00251]
Epoch 4: 100%|██████████| 1250/1250 [2:15:42<00:00, 6.51s/batch, accuracy=0.000201, loss=0.000498]
Epoch 5: 100%|██████████| 1250/1250 [2:14:41<00:00, 6.46s/batch, accuracy=0.000161, loss=0.000352]
```

트랜스포머는 구글 BERT를 낳고

— 최초 BERT 모델

```
class BERTTrainer :  
  
    def __init__(self, bert : BERT, vocab_size : int, train_dataloader : DataLoader, test_dataloader : DataLoader = None,  
                 lr : float = 1e-4, betas = (0.9, 0.999), weight_decay : float = 0.01, warmup_steps = 10000,  
                 with_cuda : bool = True, cuda_devices = None, log_freq : int = 10) :~
```

Original: 밥이나 먹죠. - Translated: 밥이나 먹죠.

구글 BERT는 페이스북 BART를 낳고 – 영-독 BART모델 한-영 모델 전환

```
# 토크나이저 정의
tokenizer = BartTokenizer.from_pretrained('facebook/bart-large')
```

Original: 나는 학생입니다 - Translated:

페이스북 BART는 깃허브 모델을 낳고

— KO-BART 모델

[study](#) / [algorithm](#) / [02.transformer_intermediate](#) / [02.transformer_intermediate.ipynb](#)

Original: 밥이나 먹죠. - Translated: .

깃허브 모델은 허깅페이스 모델을 낳고

– 한-영 허깅페이스 모델

```
model = TranslateModel.from_pretrained("Helsinki-NLP/opus-mt-ko-en")
```

Original: 걔는 홀씨가 됐다구 NaNaNaNaNa. - Translated: I don't know how long it will take.

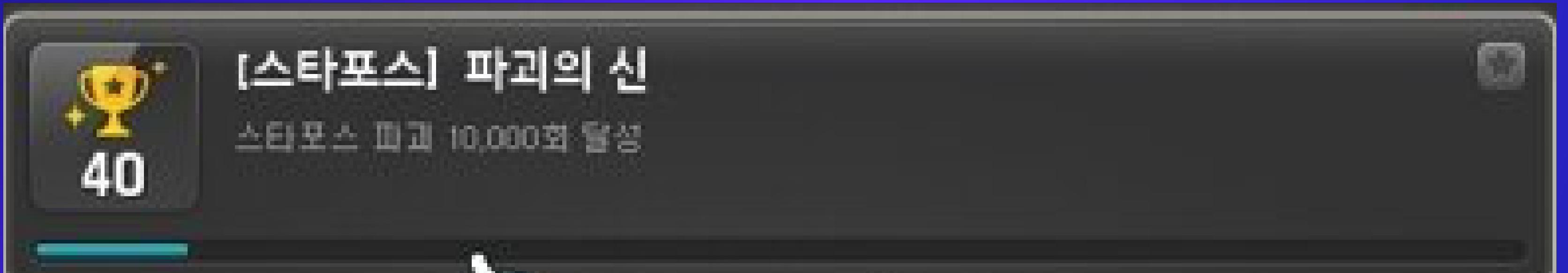
Original: 나는 친구에게 그 철학가의 책을 선물해주겠다고 말했다. - Translated: I don't know how long it will take you to finish the project.

허깅페이스 모델은 SEQ2SEQ 모델을 낳고

– 교재 + GITHUB

```
class Seq2Seq(nn.Module):  
    def __init__(self, encoder, decoder):  
        super().__init__()  
        self.encoder = encoder  
        self.decoder = decoder
```

Original : 안녕하세요. 저는 학생입니다. Translation: the price is a lot of the same .



▼ 3. 어록

- 흰 하늘 찢고 고름 섞인 개구리 양심![2][3]
- 네 발바닥 즐거운! 푸르다! 손! 밤! 아홉의 오른쪽 물거품!
- 심심한 장미를 콧구멍에—!!
- 행복은 불타는 이단 옆차기!
- 딱딱하게 끓는 망치 바르면!
- 무거운 해 늙어 태어나면 개나리 웃지요!
- 크낙새 뿌리 무침? 파란 냄새 삼각형!
- 팔짝 뛰는 토끼색 칠한 재채기 세 쌍만 던져!
- 사랑은 착한 빠다귀!
- 여보![4]
- 잃어버린 줄무늬 의표... 다움이 너무 많은, 모레.
- 기름칠 한 평화! 애국자 잡탕 떨꾹질!
- 흥미로운 거미줄의 어금니!
- 동쪽이 세 개! 우, 송아지! 늦가을은 빨래집게의 맹장을 따라!



4개의 최종 모델

번역은 많은 데이터와 많은 학습이 필요하다

현재 인프라에만 맨땅에 헤딩은 거의 불가능

전이학습이 나은 방법이었을까?

짧은 기간 내에 다양한 모델의 구조 생김새를 익힘

아마존 독서 리뷰 요약

변주영



PROJECT OBJECTIVES



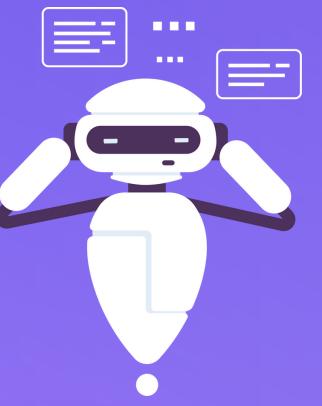
DATASET

[kaggle]
Amazon Books
Reviews Dataset



CREATE MODEL

BART
전이학습
파연어 편僻어 수행



EXECUTE IN WEB

Flask에 모델 연결
DB 연결 (결과 추가)

데이터 소개

텍스트 요약 모델을 만들고 싶다. 그러면... 무슨 데이터를 쓰지?

주제	영화 자막 -> 줄거리 요약	리뷰 요약
입력	영화 한 편당 수백줄의 자막	서너줄 길이의 본문
출력	두세줄 요약 줄거리	리뷰의 제목 능 전체 리뷰의 요약
느낌	결론 : 쉽지 않을 것 같다.	결론 : 해볼만 할 것 같다!

데이터 소개

MOHAMED BEKHEET · UPDATED 2 YEARS AGO

145

New Notebook

Download (1 GB)

Amazon Books Reviews

Goodreads-books reviews and descriptions of each book

출처 : <https://www.kaggle.com/datasets/mohamedbakhet/amazon-books-reviews>

About this file					
This file has information about 3M book reviews for 212404 unique book and users who gives these reviews for each book.					
review/helpfulness	# review/score	# review/time	# review/summary	# review/text	
helpfulness rating of the review, e.g. 2/3	rating from 0 to 5 for the book	time of given the review	the summary of text review	the full text of a review	
19% vi... 0% 81%	0/0 1/1 Other (1801268)	30% 10% 60%		1592316 unique values	2062649 unique values
-of-	7/7	4.0	940636800	Nice collection of Julie Strain images	This is only for Julie Strain fans. It's a collection of her photos -- about 80 pages worth with a n...
	10/10	5.0	1095724800	Really Enjoyed It	I don't care much for Dr. Seuss but after reading Philip Nel's book I changed my mind--that's a good...

데이터 전처리

```
### 300만 행 존재 (2.66GB)
```

```
reviewDF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000000 entries, 0 to 2999999
Data columns (total 10 columns):
 #   Column            Dtype  
 --- 
 0   Id                object  
 1   Title             object  
 2   Price              float64 
 3   User_id            object  
 4   profileName        object  
 5   review/helpfulness object  
 6   review/score       float64 
 7   review/time         int64   
 8   review/summary      object  
 9   review/text         object  
dtypes: float64(2), int64(1), object(7)
memory usage: 228.9+ MB
```

```
### 두 컬럼 중 결측치 값이 존재하는지 확인 => 존재
```

```
reviewDF[['review/summary', 'review/text']].isna().sum()
```

```
review/summary    407
review/text        8
dtype: int64
```

```
### 두 컬럼 중 값이 NaN인 행 제거
```

```
reviewDF = reviewDF.dropna(subset=['review/summary', 'review/text'])
```

결측치 체크, 제거

데이터 전처리

1만 개의 행 사용

자연어 전처리 (선택적)

데이터셋 생성

데이터로더 생성

```
def str_preprocessing(x: str):
    """ str이 아니면 그냥 반환
    if not isinstance(obj/x, class_or_tuple/str):
        return x

    """ 소문자로 변환
    x = x.lower()

    """ 소괄호로 둘러싸인 문자열 삭제
    pattern1 = r'\([^\)]*\)'
    x = re.sub(pattern=pattern1, repl='', string=x)

    """ 대괄호로 둘러싸인 문자열 삭제
    pattern2 = r'\[[^\]]*\]'
    x = re.sub(pattern=pattern2, repl='', string=x)

    """ 축약어, 슬랭 처리
    x = contractions.fix(s=x)

    return x
```

```
""" 자연어 전처리 수행
train = train.applymap(str_preprocessing)
valid = valid.applymap(str_preprocessing)
test = test.applymap(str_preprocessing)
```

데이터 전처리

1만 개의 행 사용

자연어 전처리 (선택적)

데이터셋 생성

데이터로더 생성

```
def make_dataset(data, tokenizer, device):
    tokenized = tokenizer( # tokenized : transformers.tokenization_utils_base.BatchEncoding
        text=data['review/text'].tolist(), # data 안의 text를 list로 변환
        padding="longest", # 가장 긴 시퀀스의 길이에 맞춰 패딩을 추가
        truncation=True, # 입력 데이터가 모델의 최대입력길이를 초과하는 경우 자르는 작업
        return_tensors="pt", # 토큰화된 결과를 pytorch tensor로 반환
        max_length=256
    ) # tokenized : 'input_ids', 'attention_mask'로 구성 (둘 다 tensor이고, shape : [3000, 3913])
    labels = []
    input_ids = tokenized["input_ids"].to(device)
    attention_mask = tokenized["attention_mask"].to(device)
    for target in data['review/summary']:
        ### 요약 문자열을 토큰화, 정수화 하여 텐서로 저장
        labels.append(tokenizer.encode(target, return_tensors="pt").squeeze())
    ### 입력으로 주어진 시퀀스 중 가장 긴 길이에 맞춰서 패딩을 수행한다. (shape : [30])
    labels = pad_sequence(sequences=labels, batch_first=True, padding_value=-100).to(dtype=device)
    return TensorDataset(input_ids, attention_mask, labels)

def get_dataloader(dataset, sampler, batch_size):
    data_sampler = sampler(dataset)
    dataloader = DataLoader(dataset=dataset, sampler=data_sampler, batch_size=batch_size)
    return dataloader
```

데이터 전처리

1만 개의 행 사용

자연어 전처리 (선택적)

데이터셋 생성

데이터로더 생성

```
def make_dataset(data, tokenizer, device):
    tokenized = tokenizer( # tokenized : transformers.tokenization_utils_base.BatchEncoding
        text=data['review/text'].tolist(), # data 안의 text를 list로 변환
        padding="longest", # 가장 긴 시퀀스의 길이에 맞춰 패딩을 추가
        truncation=True, # 입력 데이터가 모델의 최대입력길이를 초과하는 경우 자르는 작업
        return_tensors="pt", # 토큰화된 결과를 pytorch tensor로 반환
        max_length=256) # tokenized : 'input_ids', 'attention_mask'로 구성 (둘 다 tensor이고, shape : [3000, 3913])
    labels = []
    input_ids = tokenized["input_ids"].to(device)
    attention_mask = tokenized["attention_mask"].to(device)
    for target in data['review/summary']:
        ### 요약 문자열을 토큰화, 정수화 하여 텐서로 저장
        labels.append(tokenizer.encode(target, return_tensors="pt").squeeze())
    ### 입력으로 주어진 시퀀스 중 가장 긴 길이에 맞춰서 패딩을 수행한다. (shape : [30])
    labels = pad_sequence(sequences=labels, batch_first=True, padding_value=-100).to(dtype=device)
    return TensorDataset(input_ids, attention_mask, labels)

def get_dataloader(dataset, sampler, batch_size):
    data_sampler = sampler(dataset)
    dataloader = DataLoader(dataset=dataset, sampler=data_sampler, batch_size=batch_size)
    return dataloader
```

바로 이 부분 !!!

데이터 전처리

The screenshot shows a Jupyter Notebook interface with several tabs open. The active tab is '예제 7.18~7.23 BART 모델 실습.ipynb'. The code cell contains the following Python code:

```
print("Saved the model weights")
```

The terminal output shows a progress bar for 'Downloading builder script' at 100% completion, followed by a stack trace for an `IndexError`:

```
IndexError
Cell In[6], line 61
    59 best_loss = 10000
    60 for epoch in range(epochs):
---> 61     train_loss = train(model, optimizer, train_dataloader)
    62     val_loss, val_accuracy = evaluation(model, valid_dataloader)
    63     print(f"Epoch {epoch + 1}: Train Loss: {train_loss:.4f} Val Loss: {val_loss:.4f}
Val Rouge {val_accuracy:.4f}")

Cell In[6], line 23
    20 train_loss = 0.0
    22 for input_ids, attention_mask, labels in dataloader:
---> 23     outputs = model(input_ids=input_ids, attention_mask=attention_mask, labels=labels)
    25     loss = outputs.loss
    26     train_loss += loss.item()

File c:\ProgramData\anaconda3\envs\TEXT_017_220_38\lib\site-
packages\torch\nn\modules\module.py:1511, in Module._wrapped_call_impl(self, *args, **kwargs)
    1509     return self._compiled_call_impl(*args, **kwargs) # type: ignore[misc]
    1510 else:
-> 1511     return self._call_impl(*args, **kwargs)

File c:\ProgramData\anaconda3\envs\TEXT_017_220_38\lib\site-
packages\torch\nn\modules\module.py:1520, in Module._call_impl(self, *args, **kwargs)
    1515 # If we don't have any hooks, we want to skip the rest of the logic in
    1516 # this function, and just call forward.
    1517 if not (self._backward_hooks or self._backward_pre_hooks or
self._forward_hooks
...
    2235     # remove once script supports set_grad_enabled
    2236     _no_grad_embedding renorm_(weight, input, max_norm, norm_type)
-> 2237 return torch.embedding(weight, input, padding_idx, scale_grad_by_freq, sparse)

IndexError: index out of range in self
```

The message 'Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...' is displayed at the bottom.

예제와 똑같은 코드인데
왜 에러가 뜨지 ???

데이터 전처리

그렇게 장장 3시간이 흘러가던 중.....

패딩의 길이가 왜 다를까?

```
x, y, z = train_dataset[0]
print(x.shape)
print(y.shape)
print(z.shape)
✓ 0.0s
torch.Size([3372])
torch.Size([3372])
torch.Size([38])
```

```
x, y, z = valid_dataset[0]
print(x.shape)
print(y.shape)
print(z.shape)
✓ 0.0s
torch.Size([2990])
torch.Size([2990])
torch.Size([40])
```

```
x, y, z = test_dataset[0]
print(x.shape)
print(y.shape)
print(z.shape)
✓ 0.0s
torch.Size([3001])
torch.Size([3001])
torch.Size([30])
```

내 컴퓨터

```
x, y, z = train_dataset[0]
print(x.shape)
print(y.shape)
print(z.shape)
✓ 0.0s
torch.Size([1024])
torch.Size([1024])
torch.Size([38])
```

```
x, y, z = valid_dataset[0]
print(x.shape)
print(y.shape)
print(z.shape)
✓ 0.0s
torch.Size([1024])
torch.Size([1024])
torch.Size([40])
```

```
x, y, z = test_dataset[0]
print(x.shape)
print(y.shape)
print(z.shape)
✓ 0.0s
torch.Size([1024])
torch.Size([1024])
torch.Size([30])
```

정상 작동 컴퓨터

데이터 전처리

그리고 알아낸 놀라운 비밀

관리자: Anaconda Powershell Prompt

tbb	2021.8.0	h59b6b97_0	
terminado	0.17.1	py38haa95532_0	
textsearch	0.0.24	pypi_0	pypi
threadpoolctl	3.4.0	pypi_0	pypi
tifffile	2023.7.10	pypi_0	pypi
tinycss2	1.2.1	py38haa95532_0	
tokenizers	0.19.1	pypi_0	pypi
tomli	2.0.1	py38haa95532_0	
torchaudio	2.2.2	py38_cpu	pytorch
torchinfo	1.8.0	pyhd8ed1ab_0	conda-forge
torchmetrics	1.3.2	pyhd8ed1ab_0	conda-forge
torchtext	0.17.2	pypi_0	pypi
torchvision	0.17.2	py38_cpu	pytorch
tornado	6.3.3	py38h2bbff1b_0	
tqdm	4.66.2	pyhd8ed1ab_0	conda-forge
traitlets	5.7.1	py38haa95532_0	
transformers	4.40.0	pypi_0	pypi
typing_extensions	4.9.0	py38haa95532_1	
typing_extensions	4.9.0	py38haa95532_1	
urllib3	2.1.0	py38haa95532_1	
vc	14.2	h21ff451_1	
vs2015_runtime	14.27.29016	h5e58377_2	
wcwidth	0.2.5	pyhd3eb1b0_0	
webencodings	0.5.1	py38_1	
websocket-client	0.58.0	py38haa95532_4	
werkzeug	3.0.2	pyhd8ed1ab_0	conda-forge
wheel	0.41.2	py38haa95532_0	
widgetsnbextension	4.0.10	py38haa95532_0	
win_inet_pton	1.1.0	py38haa95532_0	
winpty	0.4.3	4	

내 컴퓨터

관리자: Anaconda Powershell Prompt

smart-open	7.0.4		pypi_0	pypi
sniffio	1.3.0		py38haa95532_0	
soupsieve	2.5		py38haa95532_0	
sqlalchemy	2.0.29		pypi_0	pypi
sqlite	3.41.2		h2bbff1b_0	
stack_data	0.2.0		pyhd3eb1b0_0	
sympy	1.12		py38haa95532_0	
tbb	2021.8.0		h59b6b97_0	
terminado	0.17.1		py38haa95532_0	
threadpoolctl	2.2.0		pyh0d69192_0	
tinycss2	1.2.1		py38haa95532_0	
tokenizers	0.15.1		py38h49fca51_0	
tomli	2.0.1		py38haa95532_0	
torchaudio	2.2.2		py38_cpu	pytorch
torchtext	0.17.2		py38	pytorch
torchvision	0.17.2		py38_cpu	pytorch
tornado	6.3.3		py38h2bbff1b_0	
tqdm	4.65.0		py38hd4e2768_0	
traitlets	5.7.1		py38haa95532_0	
transformers	4.37.2		py38haa95532_0	
typing_extensions	4.9.0		py38haa95532_1	
typing_extensions	4.9.0		py38haa95532_1	
ucrt	10.0.22621.0		h57928b3_0	conda-forge
urllib3	2.1.0		py38haa95532_1	
vc	14.2		h21ff451_1	
vc14_runtime	14.38.33130		h82b7239_18	conda-forge
vs2015_runtime	14.38.33130		hcb4865c_18	conda-forge
wcwidth	0.2.5		pyhd3eb1b0_0	
webencodings	0.5.1		py38_1	
websocket-client	0.58.0		py38haa95532_4	

정상 작동 컴퓨터

데이터 전처리

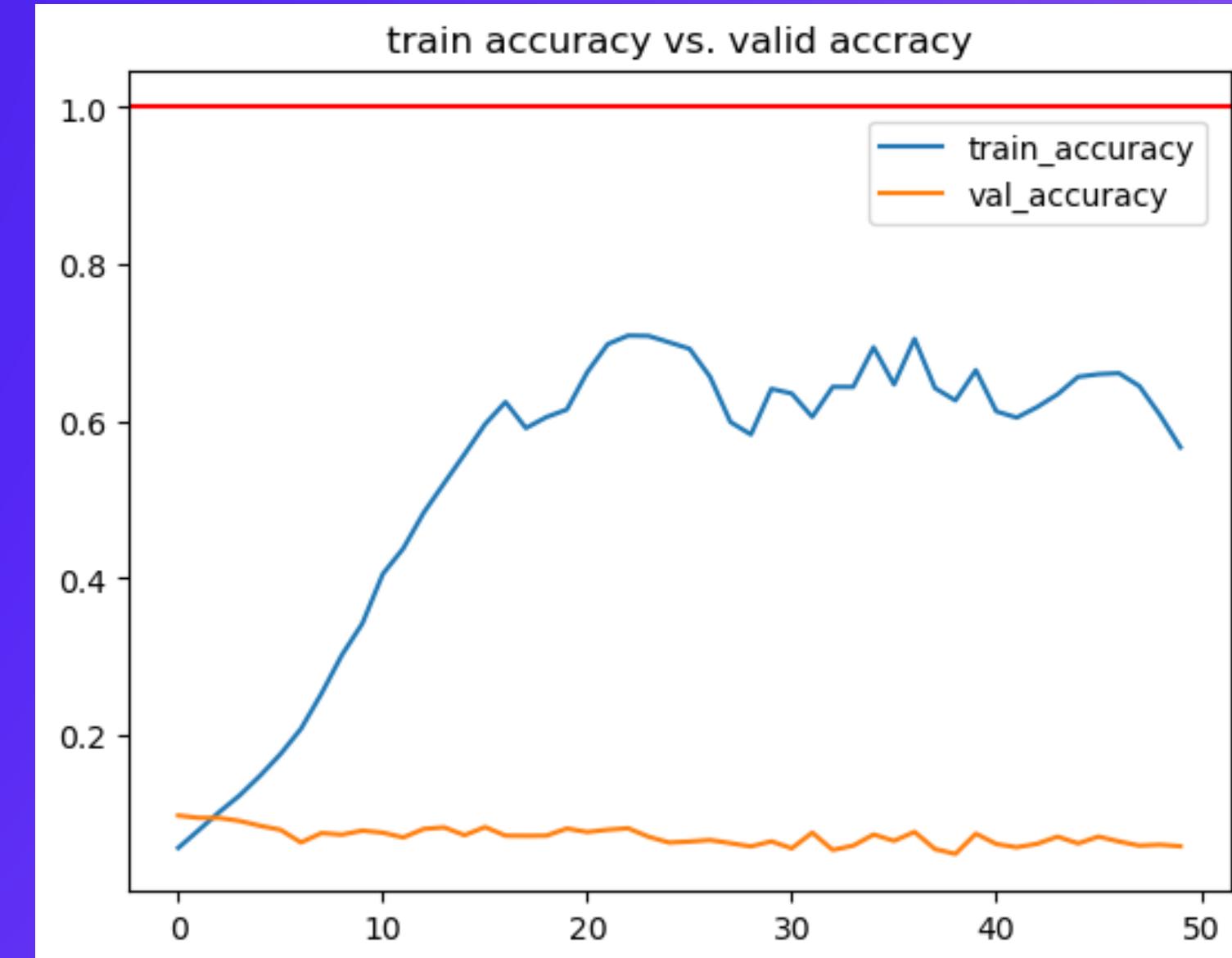
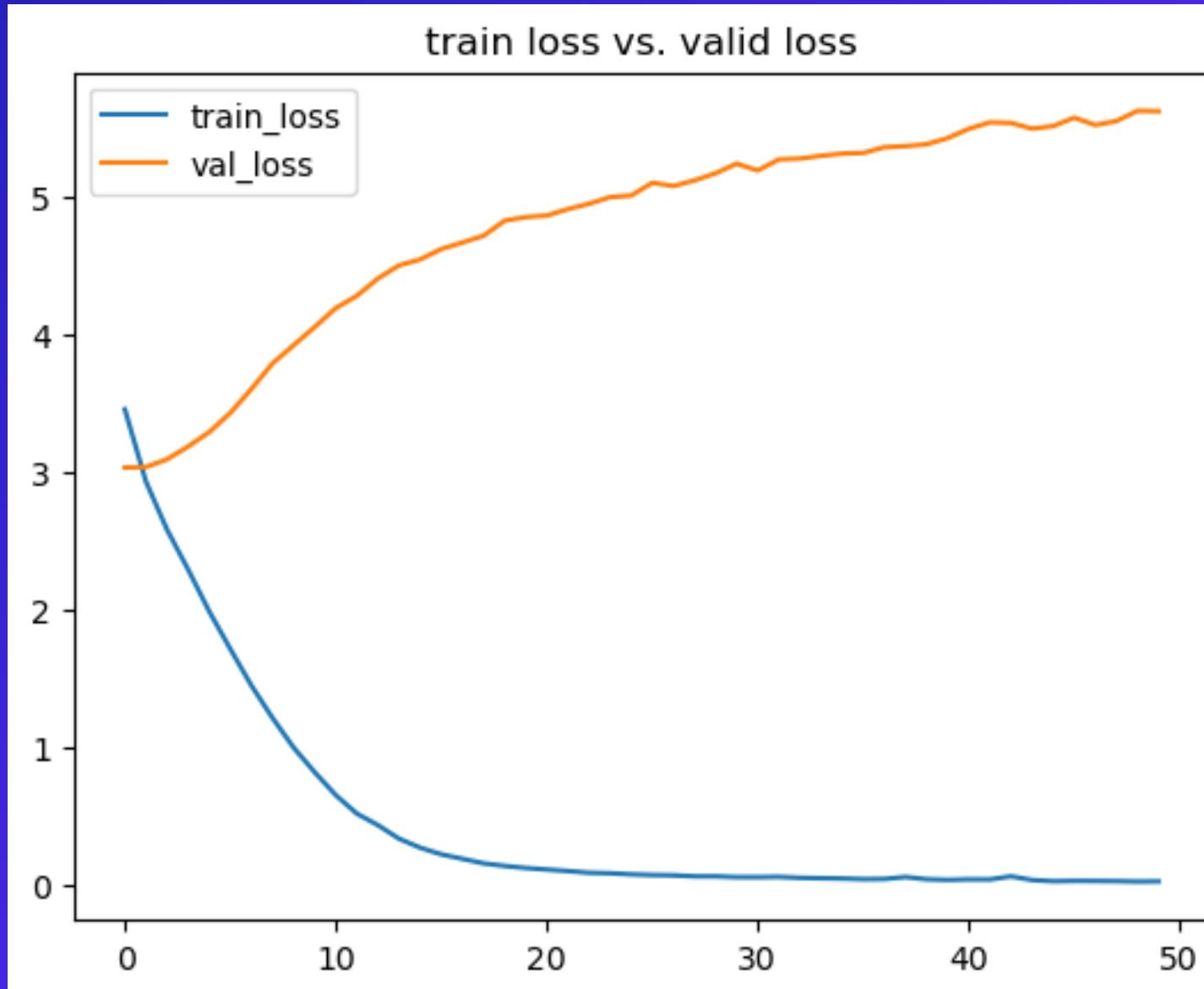
트랜스포머 버전에 따라 BartTokenizer 의 작동이 달랐던 것이 원인이었던 것...

```
def make_dataset(data, tokenizer, device):
    tokenized = tokenizer( # tokenized : transformers.tokenization_utils_base.BatchEncoding
        text=data['review/text'].tolist(), # data 안의 text를 list로 변환
        padding="longest", # 가장 긴 시퀀스의 길이에 맞춰 패딩을 추가
        truncation=True, # 입력 데이터가 모델의 최대입력길이를 초과하는 경우 자르는 작업
        return_tensors="pt", # 토큰화된 결과를 pytorch tensor로 반환
        max_length=1024 # transformers 4.37.2 에서는 없어도 자동으로 길이고정됨. transformers 4.40.0 에서는 max_length를 줘야 고정됨
    ) # tokenized : 'input_ids', 'attention mask' 로 구성 (둘 다 tensor이고, shape : [3000, 3913])
```

트랜스포머 4.37.2 에서는 **max_length** 없이도 길이를 통일해 주고,

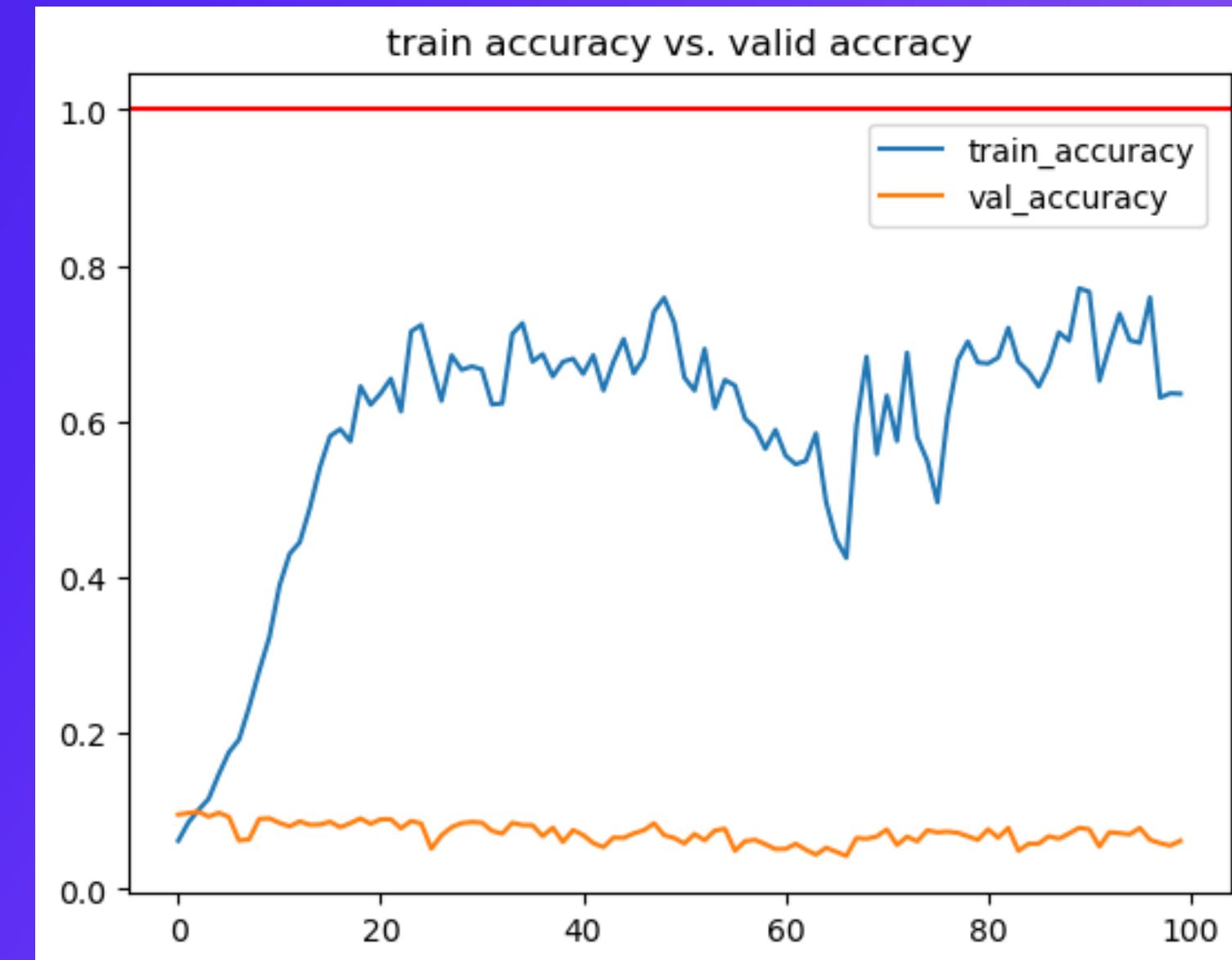
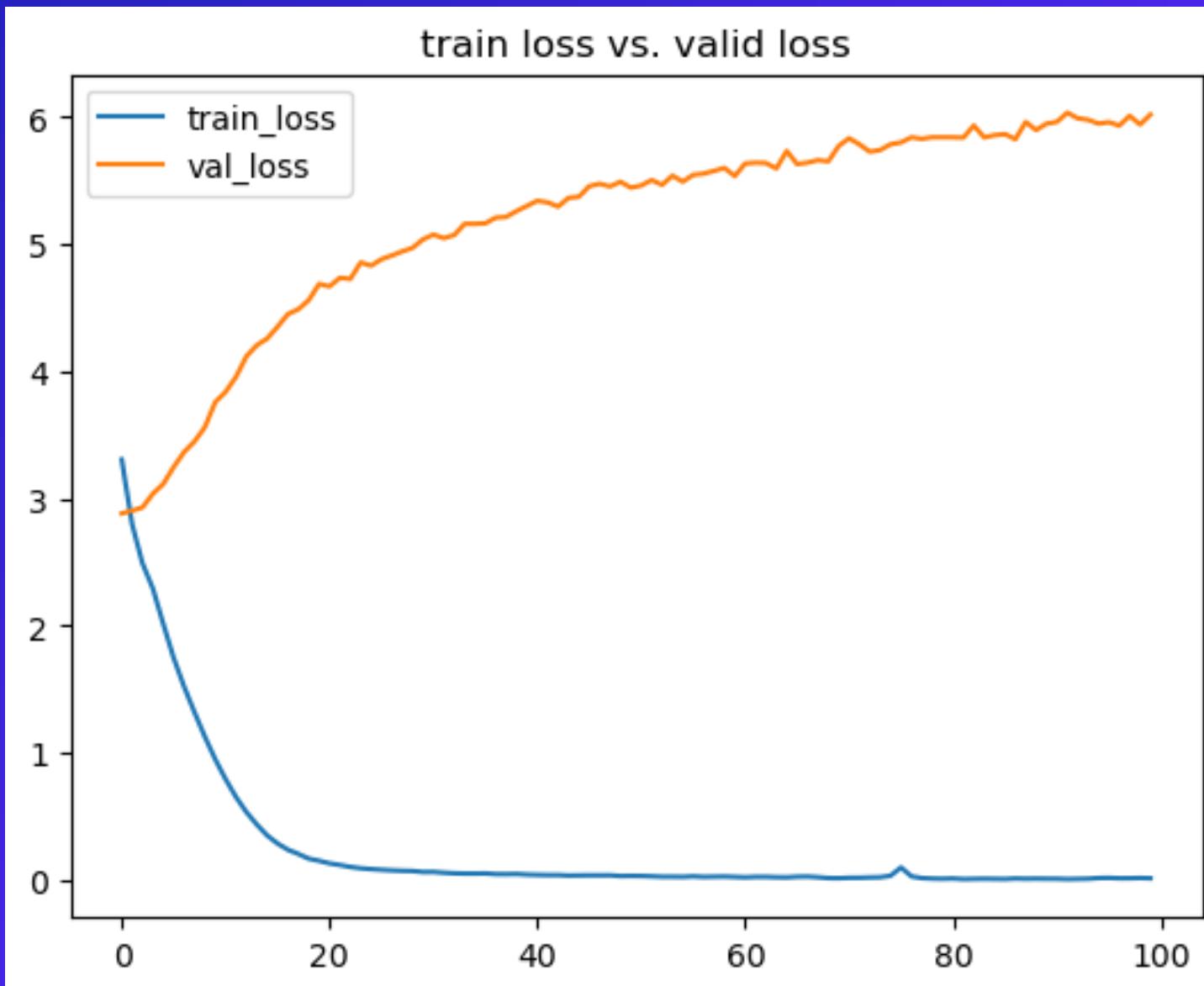
트랜스포머 4.40.0 에서는 **max_length** 없으면 제일 긴 문자열 길이로 패딩됩니다.

모델 학습 (1) - 자연어 전처리 X



하긴... 전처리를 안 해 줬으니까...

모델 학습 (2) - 자연어 전처리 ○



왜 별로 차이가 없지;;

모델 테스트 (predict)



Amazon Customer

★★★★★ but I enjoyed it so much

Reviewed in the United States on October 16, 2017

Verified Purchase

I was hoping to spread this out to a couple of weeks, but I enjoyed it so much, I listened to it in a few days. Jim Dale does such a wonderful job narrating. His voice is soothing and easy to listen to and I love the voices he does for each character. I've read the books myself a dozen times, but it's nice to be able to listen while doing something else.

4 people found this helpful

'Great book, but not enough for a few days'

전처리 X Epoch 1

'Great book! More to read in a week! '

전처리 X Epoch 50

'great book for a few days. I loved it! '

전처리 O Epoch 1

'great book to read for a lot of people! '

전처리 O Epoch 100

모델 테스트 (predict)



Online Purchase

★★★★★ Perfect gift from someone how has never met a child

Reviewed in the United States on January 27, 2023

Verified Purchase

This book is like it was generated by some AI code. It's supposed to convey a nice message, but the illustrations are really awkward. It's like some thing projected what would make a surgery sweet baby/kids book.

The content is sort of geared towards infants and above, but it's an unmanageably large size to read with a child and has gigantic pull out pages, which means the book ends up on the floor while the parent juggles with the curious child.

It's a perfect gift from someone that has never met a child - it looks nice on the outside, but useless and surgery on the inside, and should best be put on a shelf somewhere to languish. It's empty calories in book form.

"Not what you'd expect from a baby book."

전처리 X Epoch 1

"Surprisingly good - it's not that great of a book"

전처리 X Epoch 50

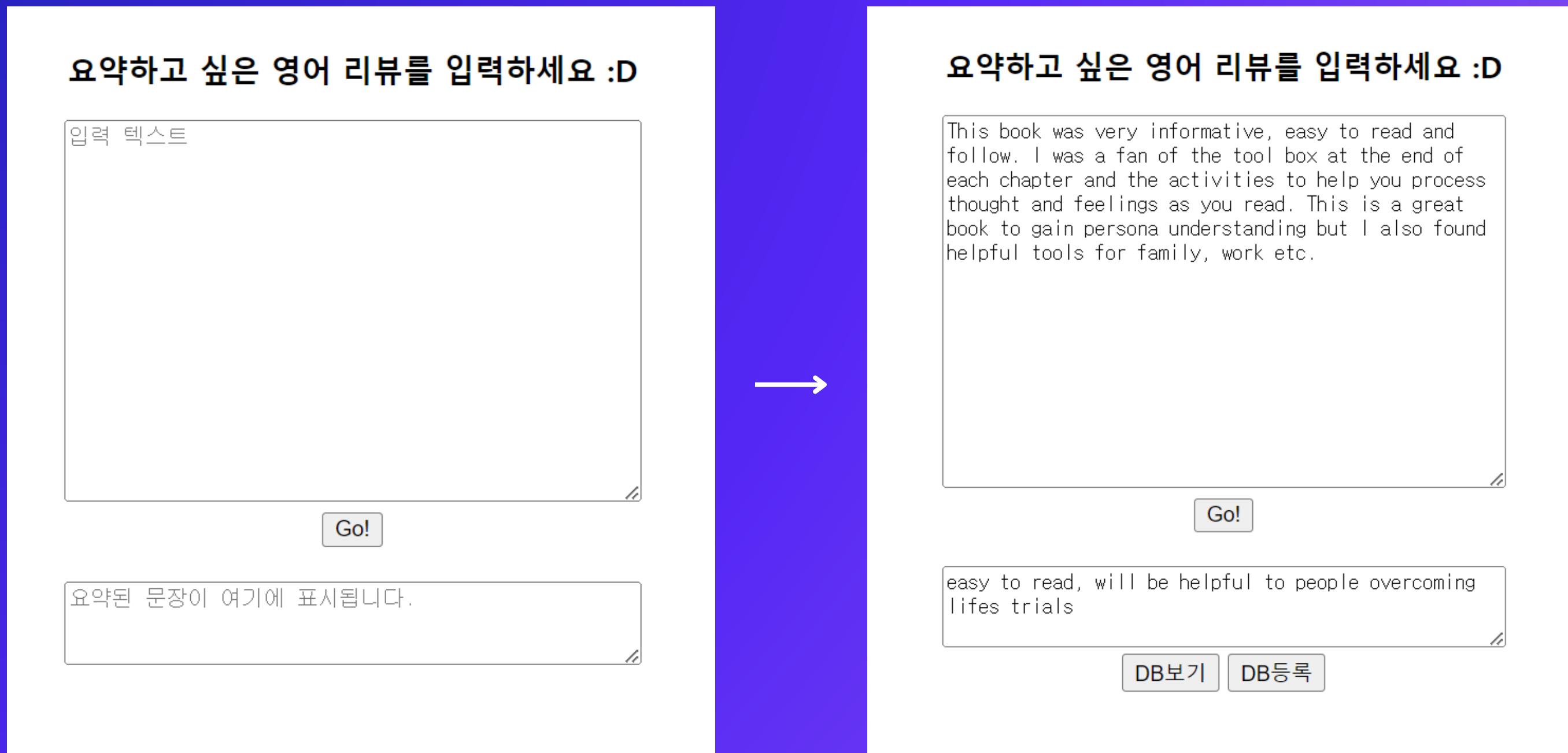
"it looks nice on the outside, but it's empty calories in book form"

전처리 O Epoch 1

'it is not that bad - it is just pretty boring'

전처리 O Epoch 100

모델 - Flask 연동



기본 화면

요약 모델 실행 화면

결론

본문의 제목은 충분히 요약에 해당한다고 생각하여 사용하였으나,

요약이 잘 된 제목의 비중이 적은 것 같다.

그러므로, 의무적으로 적은 경우(좋은 요약이라 보기 힘든 경우)들을 삭제해 주었으면 더 좋았을 것 같다.

이를 해결하기 위해, 본문과 제목의 연관성이 크도록 작성된 논문이나 신문을 학습시키면

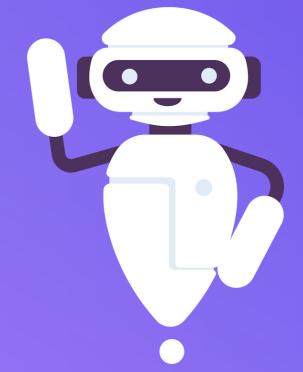
훨씬 더 좋은 결과를 보일 수 있지 않을까 하는 생각을 하였다.

호불호 분석 프로그램

양현우



PROJECT OBJECTIVES



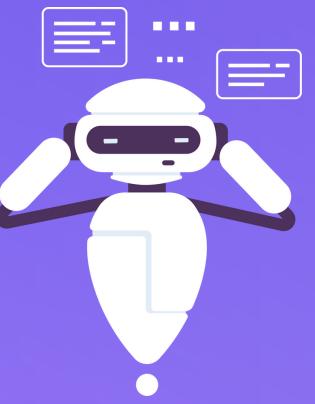
DATASET

[Korpora]
NAVER.Sentiment
Movie.Corporus



CREATE MODEL

KoBERT(불가능)
BERT



EXECUTE IN WEB

Flask 에 모델 연결
DB 연결 (결과 추가)

-release)

🏠 / README.md

Korpora: Korean Corpora

Korpora는 오픈소스 말뭉치들의 다운로드와 전용화를 용할 수 있도록 돋기 위해 만들었습니다. 말뭉치들은

말뭉치 목록

- Korpora 패키지가 제공하는 말뭉치 목록

말뭉치 이름	
korean_chatbot_data	챗봇 트레이닝 데이터
kcbert	KcBERT 모델
korean_hate_speech	한국어 혐오 말뭉치
korean_petitions	청와대 국민청원
kornli	Korean NLP
korsts	Korea STS
namuikitext	나무위키 텍스트

데이터 설정 단계

KORPORA

- korpora는 대규모 한국어 말뭉치를 제공하는 오픈소스 라이브러리

NSMC

- 네이버 영화 리뷰 말뭉치



초기 계획

- 트랜스포머의 주 기능 중 하나 감정분석을 목표로 시작
- 라벨링된 데이터셋 찾기(ai-hub, korpora, 등등 한글로 된 데이터 중심)
- koBERT란?
- koBERT로 뭔가를 만들 수 있지 않을까?

KoBERT?



- KoBERT (Korean Bidirectional Encoder Representations from Transformers)
- 기존 BERT의 한국어 성능 한계를 극복하기 위해 SKT Brain에서 개발한 모델
- 기존 대비 27%의 토큰만으로 2.6% 이상의 성능 향상을 이끌어냄

설치 단계

- 사전(Vocabulary)
 - 크기 : 8,002
 - 한글 위키 기반으로 학습한 토크나이저(SentencePiece)
 - Less number of parameters(92M < 110M)

Requirements

- see [requirements.txt](#)

How to install

- Install KoBERT as a python package

```
pip install git+https://git@github.com/SKTBrain/KoBERT.git@master
```

- If you want to modify source codes, please clone this repository

```
git clone https://github.com/SKTBrain/KoBERT.git
cd KoBERT
pip install -r requirements.txt
```

```
%pip install git+https://git@github.com/SKTBrain/KoBERT.git@master
✓ 27.5s

Collecting git+https://****@github.com/SKTBrain/KoBERT.git@master
  Cloning https://****@github.com/SKTBrain/KoBERT.git (to revision master) to c:\users\h
  Resolved https://****@github.com/SKTBrain/KoBERT.git to commit 47a69af87928fc24e20f571
    Preparing metadata (setup.py): started
    Preparing metadata (setup.py): finished with status 'done'
  Collecting boto3<=1.15.18 (from kobert==0.2.3)
    Using cached boto3-1.15.18-py2.py3-none-any.whl.metadata (5.3 kB)
  Collecting gluonnlp<=0.10.0,>=0.6.0 (from kobert==0.2.3)
    Using cached gluonnlp-0.10.0.tar.gz (344 kB)
    Preparing metadata (setup.py): started
    Preparing metadata (setup.py): finished with status 'done'
  Collecting mxnet<=1.7.0.post2,>=1.4.0 (from kobert==0.2.3)
    Using cached mxnet-1.7.0.post2-py2.py3-none-win_amd64.whl.metadata (402 bytes)
  Collecting onnxruntime<=1.8.0,==1.8.0 (from kobert==0.2.3)
    Using cached onnxruntime-1.8.0-cp38-cp38-win_amd64.whl.metadata (3.0 kB)
  Collecting sentencepiece<=0.1.96,>=0.1.6 (from kobert==0.2.3)
    Using cached sentencepiece-0.1.96-cp38-cp38-win_amd64.whl.metadata (10 kB)
  Collecting torch<=1.10.1,>=1.7.0 (from kobert==0.2.3)
    Using cached torch-1.10.1-cp38-cp38-win_amd64.whl.metadata (24 kB)
  Collecting transformers<=4.8.1,>=4.8.1 (from kobert==0.2.3)
    Using cached transformers-4.8.1-py3-none-any.whl.metadata (48 kB)
Requirement already satisfied: numpy>=1.16.6 in e:\user\administrator\anaconda3\envs\my_
Collecting protobuf (from onnxruntime<=1.8.0,==1.8.0->kobert==0.2.3)
  Using cached protobuf-5.26.1-cp38-cp38-win_amd64.whl.metadata (592 bytes)
Collecting flatbuffers (from onnxruntime<=1.8.0,==1.8.0->kobert==0.2.3)
...
1. loosen the range of package versions you've specified
2. remove package versions to allow pip attempt to solve the dependency conflict

Note: you may need to restart the kernel to use updated packages.
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
Running command git clone --filter=blob:none --quiet 'https://****@github.com/SKTBrain/KoBERT.git@master'
ERROR: Cannot install kobert because these package versions have conflicting dependencies
ERROR: ResolutionImpossible: for help visit https://pip.pypa.io/en/latest/topics/dependency-resolution/
```

설치 단계?

```
ERROR: Cannot install kobert because these package versions have conflicting dependencies
ERROR: ResolutionImpossible: for help visit https://pip.pypa.io/en/latest/topics/dependencies/
```

```
boto3 <=1.15.18
gluonnlp >= 0.6.0, <=0.10.0
mxnet >= 1.4.0, <=1.7.0.post2
onnxruntime == 1.8.0, <=1.8.0
sentencepiece >= 0.1.6, <=0.1.96
torch >= 1.7.0, <=1.10.1
transformers >= 4.8.1, <=4.8.1
```

- torch, transformers 말고는 도저히 모르겠다...
- BERT로 타협

데이터 설정, 정제

```
corpus = Korpora.load("nsmc")
df = pd.DataFrame(corpus.test).sample(10000, random_state=17)
train, valid, test = np.split(
    df.sample(frac=1, random_state=17), [int(0.6 * len(df)), int(0.8 * len(df))])
)

def make_dataset(data, tokenizer, device):
    tokenized = tokenizer(
        text = data.text.tolist(),
        padding="longest",
        truncation=True,
        return_tensors="pt"
    )
    input_ids = tokenized["input_ids"].to(device)
    attention_mask = tokenized["attention_mask"].to(device)
    labels = torch.tensor(data.label.values, dtype=torch.long).to(device)
    return TensorDataset(input_ids, attention_mask, labels)

def get_dataloader(dataset, sampler, batch_size):
    data_sampler = sampler(dataset)
    dataloader = DataLoader(dataset, sampler=data_sampler, batch_size=batch_size)
    return dataloader
```

데이터 로더

```
epochs = 100
batch_size = 32
device = "cuda" if torch.cuda.is_available() else "cpu"
tokenizer = BertTokenizer.from_pretrained(
    pretrained_model_name_or_path="bert-base-multilingual-cased", do_lower_case=False
)
```

```
train_dataset = make_dataset(train, tokenizer, device)
train_dataloader = get_dataloader(train_dataset, RandomSampler, batch_size)

valid_dataset = make_dataset(valid, tokenizer, device)
valid_dataloader = get_dataloader(valid_dataset, SequentialSampler, batch_size)

test_dataset = make_dataset(test, tokenizer, device)
test_dataloader = get_dataloader(test_dataset, RandomSampler, batch_size)

print(train_dataset[0])
```

데이터 학습, 검증

```
def train(model, optimizer, dataloader):
    model.train()
    train_loss = 0.0

    for input_ids, attention_mask, labels in dataloader:
        outputs = model(
            input_ids=input_ids,
            attention_mask=attention_mask,
            labels=labels
        )

        loss = outputs.loss
        train_loss += loss.item()

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

    train_loss = train_loss / len(dataloader)
    return train_loss
```

```
def evaluation(model, dataloader):
    with torch.no_grad():
        model.eval()
        criterion = nn.CrossEntropyLoss()
        val_loss, val_accuracy = 0.0, 0.0

        for input_ids, attention_mask, labels in dataloader:
            outputs = model(
                input_ids=input_ids,
                attention_mask=attention_mask,
                labels=labels
            )

            logits = outputs.logits

            loss = criterion(logits, labels)
            logits = logits.detach().cpu().numpy()
            label_ids = labels.to("cpu").numpy()
            accuracy = calc_accuracy(logits, label_ids)

            val_loss += loss
            val_accuracy += accuracy

    val_loss = val_loss/len(dataloader)
    val_accuracy = val_accuracy/len(dataloader)
    return val_loss, val_accuracy
```

커널 사망

```
best_loss = 10000
for epoch in range(epochs):
    train_loss = train(model, optimizer, train_dataloader)
    val_loss, val_accuracy = evaluation(model, valid_dataloader)
    print(f'Epoch {epoch +1}: Train Loss: {train_loss:.4f} Val Loss: {val_loss:.4f} Val Accuracy: {val_accuracy:.4f}')

    if val_loss < best_loss:
        best_loss = val_loss
        torch.save(model.state_dict(), "./BERTForSequenceClassification.pt")
        print("save the model weights")
```

The Kernel crashed while executing code in the current cell or a previous cell.

Please review the code in the cell(s) to identify a possible cause of the failure.

Click [here](#) for more info.

View Jupyter [log](#) for further details.

학습 결과(FROM HOME)

```
Epoch 1: Train Loss: 0.6028 Train Accuracy: 0.6579 Val Loss: 0.5018 Val Accuracy: 0.7515
save the model weights
Epoch 2: Train Loss: 0.4437 Train Accuracy: 0.7942 Val Loss: 0.4842 Val Accuracy: 0.7629
save the model weights
Epoch 3: Train Loss: 0.3382 Train Accuracy: 0.8534 Val Loss: 0.4766 Val Accuracy: 0.7877
save the model weights
Epoch 4: Train Loss: 0.2505 Train Accuracy: 0.8944 Val Loss: 0.5663 Val Accuracy: 0.7793
Epoch 5: Train Loss: 0.1892 Train Accuracy: 0.9259 Val Loss: 0.5899 Val Accuracy: 0.7852
Epoch 6: Train Loss: 0.1353 Train Accuracy: 0.9468 Val Loss: 0.7263 Val Accuracy: 0.7877
```

철야의 노래

검증

13 우리는 꿈뜰이들에게 행복을 주셔서 감사합니다.

긍정적 확률
:0.917729377746582

2024-04-
22 13:27:01

아아 군대가기 싫다

부정적 확률
:0.8007347583770752

16 사람이 죽으면 먼저 가있던 반려동물이 마중나온다는 이야기가 있다. 나는 이 이야기를 무척 좋아한다. 그때가 되면 우리는 서로를 이해하게 될 것이다

긍정적 확률
:0.865452229976654

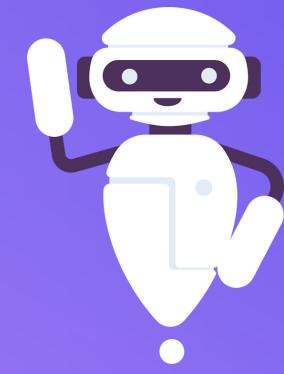
결론



잠뜰TV 댓글
생성 프로그램
명노아



PROJECT OBJECTIVES



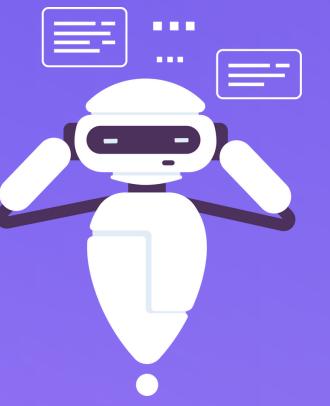
DATA CRAWLING

구글 API 키 사용
댓글 정보
유튜버 정보 크롤링
정규표현식 사용



CREATE MODEL

전이 학습
Word2Vec 사용
KoGPT2 사용
모델 학습(5시간)
Epoch3 VS Epoch100



EXECUTE IN WEB

Flask 와 연결
DB연결
CRUD기능 구현
BootStrap
CSS & JS 구현

유튜브 댓글 크롤링

일일이 언제 다하지?

The screenshot shows a Google Cloud search interface with the search term 'youtube' entered. The results are filtered under the '문서 및 튜토리얼' (Documents and Tutorials) tab. The results list several documents related to integrating YouTube with BigQuery:

- YouTube 채널 전송 | BigQuery**
YouTube용 BigQuery Data Transfer Service를 사용하면 YouTube 채널 보고서와 관련한 반복적인 로드 작업을 자동으로 예약하고 관리할 수 있습니다.
유형: 문서
- YouTube 채널 보고서 변환 | BigQuery**
BigQuery로 전송된 YouTube 채널 보고서는 다음 BigQuery 테이블 및 뷰로 변환됩니다. BigQuery에서 테이블과 뷰를 볼 때 suffix 값은 전송을 생성할 때 구성한 테이블 서 ...
유형: 문서
- YouTube 콘텐츠 소유자 전송 | BigQuery**
YouTube용 BigQuery Data Transfer Service를 사용하면 YouTube 콘텐츠 소유자 보고서와 관련한 반복적인 로드 작업을 자동으로 예약하고 관리할 수 있습니다.
유형: 문서
- YouTube 채널 보고서에서 데이터 로드 | BigQuery**
YouTube 채널 보고서에서 BigQuery로 반복되는 로드 작업을 예약합니다.
유형: 문서
- YouTube 분석 | Integration Connectors**
YouTube 분석 커넥터를 사용하면 YouTube 분석에서 지원되는 작업을 수행할 수 있습니다. 시작하기 전에 YouTube 분석 커넥터를 사용하기 전에 다음 태스크를 수행합니다.
유형: 문서
- YouTube 콘텐츠 소유자 보고서에서 데이터 로드**
YouTube 콘텐츠 소유자 보고서에서 BigQuery로 반복되는 로드 작업을 예약합니다.

Google Cloud Platform에서
유튜브 관련 API 키 제공

유튜브 댓글 크롤링

```
comments = list()
ID = "8fWqUaVkJSo"
api_obj = build('youtube', 'v3', developerKey='[REDACTED]')
response = api_obj.commentThreads().list(part='snippet,replies', videoId=ID, maxResults=100).execute()

# 유튜브 API 문서에서 확인할 수 있는 변수명
# textDisplay: 댓글의 내용
# authorDisplayName: 댓글 작성자
# publishedAt: 댓글 작성 시간
# likeCount: 좋아요 수
while response:
    for item in response['items']:
        comment = item['snippet']['topLevelComment']['snippet']
        comments.append([comment['textDisplay'], comment['authorDisplayName'], comment['publishedAt'], comment['likeCount']])

        if item['snippet']['totalReplyCount'] > 0:
            for reply_item in item['replies']['comments']:
                reply = reply_item['snippet']
                comments.append([reply['textDisplay'], reply['authorDisplayName'], reply['publishedAt'], reply['likeCount']])

    if 'nextPageToken' in response:
        response = api_obj.commentThreads().list(part='snippet,replies', videoId=ID, pageToken=response['nextPageToken'], maxResults=100).execute()
    else:
        break
```

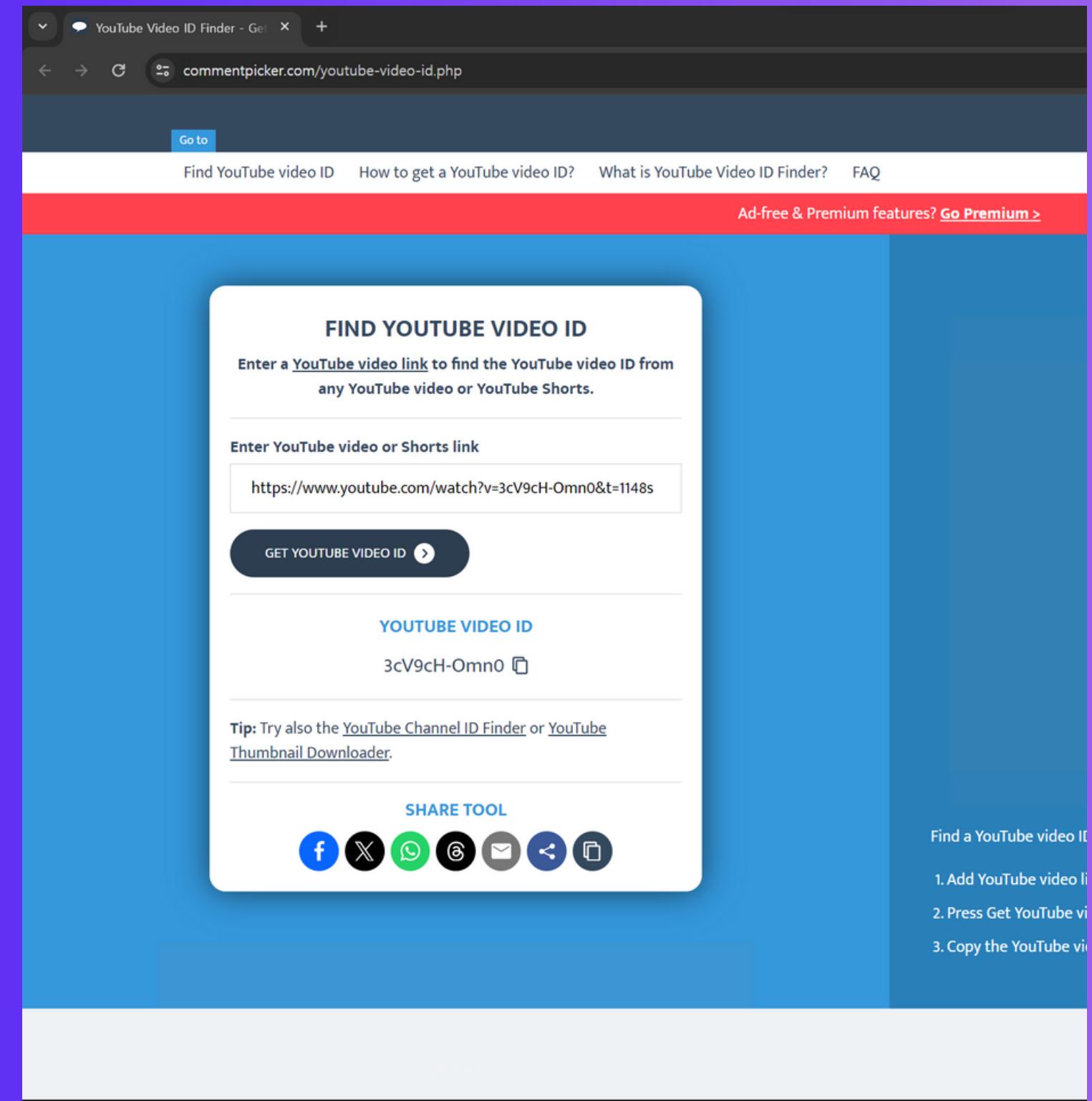
유튜브 댓글 크롤링

일일이 언제 다하지?

유튜브 댓글 크롤링

최근 날짜 기준(4.19) : 동영상 72개 VideoID 수집

```
ID_list=[ 'ta6mirThQBC', "nYFMht12_Vg", "l0F2f1VKfqc", "EkJ3Ib0znGQ",  
'C-CLlmpFD288', 'B_Sy69HPGPQ', 'aY474FVarFg', 'mla6nfz0hbw',  
'Jx2FnWAtYvI', '8y7dzODwzXs', "QYnDm6-UYUI", "5aW7vYdMUK0",  
'4XX8vPnUVEg', 'Iyt8qg8yeX4', 'NYP6pkSXowc', '4xz_9SGAmBE',  
'rNGKgI0-RTA', 'yEA1hhI7w50', 'kuiQWoB1wfA', 'qSkZsnDGgas',  
'XBcpwKz3wmc', 'jdZGlp3WICM', 'uTIWvJJZp1_E', 'LVM6sAXGdTl',  
'GAZBK8KHDHU', 'ecHSsSYQX5I', 'PhIh26JK3Lg', 'efPL9-txcAk',  
'hS0wdooquJU', 'c4nTzJFNGLc', 'JBGfEil0Qg8', 'gUtWzQJ6ChA',  
'-LzEnjMLKKY', 'kEwFq-jjA8Q', '9iMpUzF5QSo', '3IrXfolzdI8',  
'RrdVZAmWTyc', 'd3oS6-4PRsU', 'G8tXwlBqUok', 'r-44KzvCoP0',  
'6zGuJOS7XFk', 'TFH0cweeme8', 'ecZmFQ8gLGU', '4nTWE1KIpJg',  
'hIV7TlFbn0c', '_SFJ4YLcTCQ', '5IbnKeHeSMk', 'w_GbrgRu9tk',  
'w-KL6aSLyjA', 'Bc5muJ1ZL2w', 'TXAluhqHNCw', 'P9DHF7u_n08',  
'FGBtxD3YJy0', 'Uqn7wF4gkh0', '4gm2p0d8E74', '6kArUiagM2E',  
'nQhx81EVV1U', '2Gwnh0jbnE4', 'Z7QdRi5fDLs', '74If-D3CV1Y',  
'ojn2Cxp-N3k', 'ikmgIskVYuc', 'Cr-Jgtnp5Bs', 'XpzKq1xCffs',  
'Od1Y80qHdIo', 'vZyKE0BMAfg', '6vjjYoges6I', 'NRIm0cuQ3bE',  
,]  
  
# 동영상 범위(72개) : https://www.youtube.com/watch?v=NRIm0cuQ3bE&t=1155s  
for i in ID_list:  
    Scraping_Comments(i)
```



유튜브 댓글 크롤링

74000개 데이터 확보

comment	3781
와	634
오	355
ㅋㅋ	182
ㅋㅋㅋ	178
안녕하세요	168
뜰하	154
빠	144
우왕	144
사랑해요	129
우와	124
시간전	115
헐	114
ㅋㅋㅋㅋ	114
ㅋㅋㅋㅋㅋ	103
감사합니다	98
왕	93
와우	93
스포방지선	87
ㅠㅠ	83
ㅋㅋㅋㅋㅋㅋ	79
꿀잼	79
헉	78
잠뜰님오늘도너무너무재미있었어요그리고잠뜰님항상영상을만들어주셔서감사합니다잠뜰님힘내세요그리고잠뜰님최고입니다	75
재밌어요	73
재밌다	72
오늘도꿀잼영상재미있게잘보고갑니다ㅎㅎㅎ	69
분전	57
좋아요안받기일차	56

1 comment, author, date, num_likes

- 73911 마법, @user-pf3ei4wi6x, 2024-01-28T07:14:59Z, 0
- 73912 1빠다악 15초 전, @Doexn_, 2024-01-28T07:14:59Z, 1
- 73913 아니다악, @user-lk7pr8se7z, 2024-01-28T07:15:30Z, 1
- 73914 @@user-lk7pr8se7z 제가 이 영상에 왔을때 이영상이 15초 전에 올라왔다고 떠있었어음, @Doexn_, 2024-01-28T07:25:21Z, 0
- 73915 와 대박!! 이게바로 이세카이?..., @woohyukjang3473, 2024-01-28T07:14:58Z, 0
- 73916 오늘도 영상 잘 볼게요!!!, @ohoeui, 2024-01-28T07:14:57Z, 0
- 73917 오, @user-vg9tx3wn4d, 2024-01-28T07:14:57Z, 0
- 73918 마법소녀 잠뜰님, @cqfhwl, 2024-01-28T07:14:55Z, 1
- 73919

Gensim(유사도 모델)

```
tokenizer = Okt()
tokens = [tokenizer.morphs(review) for review in corpus.comment]

from gensim.models import Word2Vec

word2vec = Word2Vec(
    sentences=tokens,
    vector_size=128,
    window=5,
    min_count=1,
    sg=1,
    epochs=3,
    max_final_vocab=10000
)

word2vec.save("./word2vec.model")
word2vec = Word2Vec.load("./word2vec.model")

word = "마인크래프트"
print(word2vec.wv[word])
print(word2vec.wv.most_similar(word, topn=5))
print(word2vec.wv.similarity(w1=word, w2="연기력"))
```

word = "마인크래프트"

```
print(word2vec.wv.most_similar(word, topn=5))
```

```
[('♥', 0.865932047367096), ('멋지신', 0.7457040548324585), ('🐶❤', 0.7267171740531921), ('송강', 0.7040014863014221), ('진화', 0.6655195951461792)]
```

=> ♥, '멋지신', 🐶❤, '송강', '진화'

transformer 생성

기존에 있는 GPT모델에 흥미

GPT2 모델 사용 -> 실패

koGPT2 모델 발견!

KoGPT2 Public

Watch 14 Fork 98 Star 516

master 3 Branches 0 Tags Go to file Add file Code

bage79 Merge pull request #52 from BM-K/master e2bba02 · 2 years ago 62 Commits

imgs Update README.md 3 years ago
kogpt2-base-v2 @ 6f383b7 add chit-chat example 3 years ago
.gitmodules add to huggingface.co 3 years ago
LICENSE Update LICENSE 3 years ago
README.md fix example code cleaning 2 years ago

README License

- KoGPT2 (한국어 GPT-2) Ver 2.0
 - Tokenizer
 - Model
 - Performances
 - Classification or Regression
 - Data
 - Demo
 - User Contributed Examples
 - Related press releases
 - Contacts
 - License

KoGPT2 (한국어 GPT-2) Ver 2.0

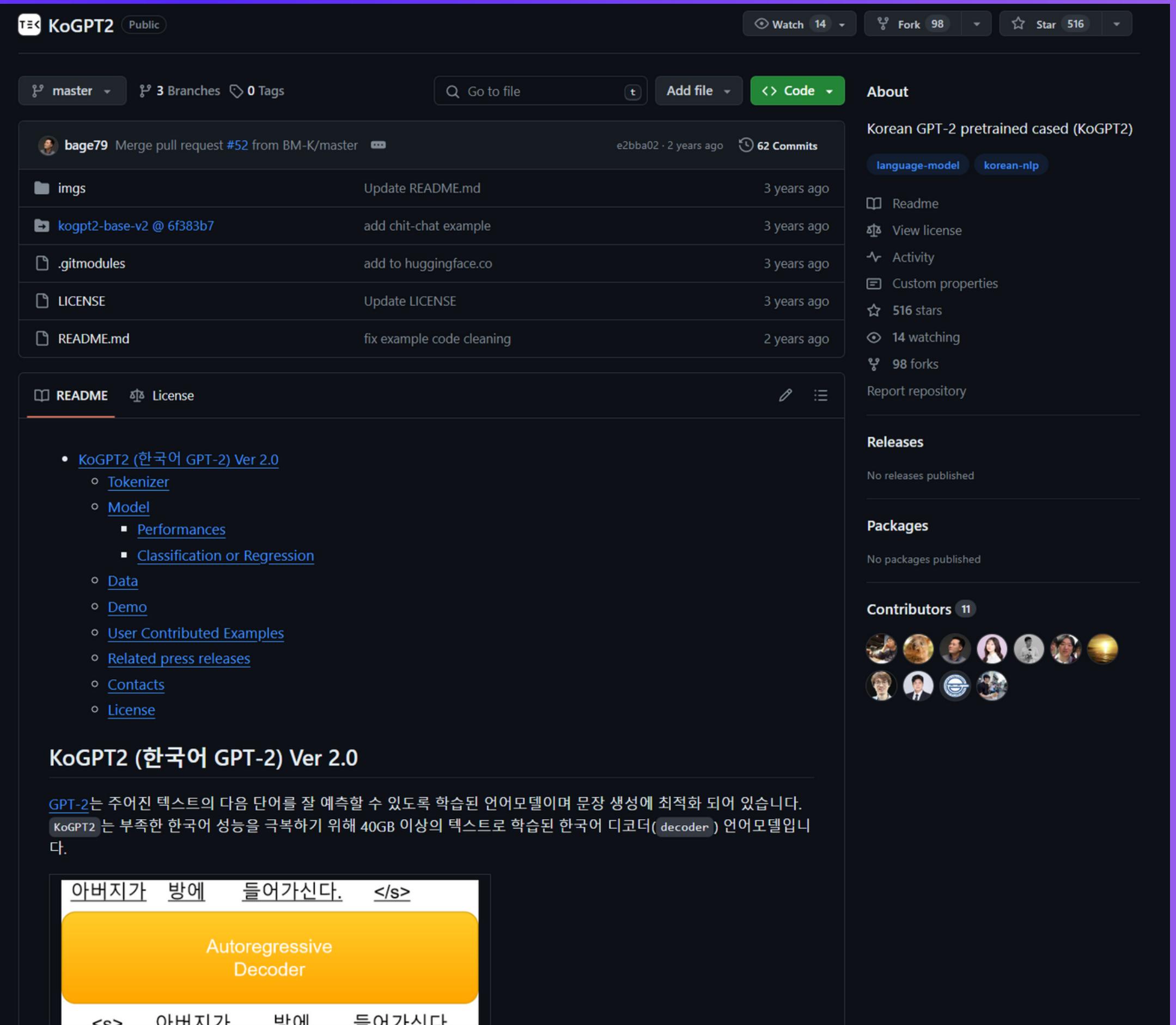
GPT-2는 주어진 텍스트의 다음 단어를 잘 예측할 수 있도록 학습된 언어모델이며 문장 생성에 최적화 되어 있습니다. KoGPT2는 부족한 한국어 성능을 극복하기 위해 40GB 이상의 텍스트로 학습된 한국어 디코더(decoder) 언어모델입니다.

아버지가 방에 들어가신다. </s>

Autoregressive Decoder

<s> 아버지가 방에 들어가신다

About Korean GPT-2 pretrained cased (KoGPT2)
language-model korean-nlp
Readme View license Activity Custom properties 516 stars 14 watching 98 forks Report repository
Releases No releases published
Packages No packages published
Contributors 11



transformer 생성

```
from transformers import PreTrainedTokenizerFast
tokenizer = PreTrainedTokenizerFast.from_pretrained("skt/kogpt2-base-v2", bos_token='</s>', eos_token='</s>', unk_token='<unk>', pad_token='<pad>', mask_token='<mask>')
import torch
from transformers import GPT2LMHeadModel

model = GPT2LMHeadModel.from_pretrained('skt/kogpt2-base-v2')
text = '잠뜰 TV'
input_ids = tokenizer.encode(text, return_tensors='pt')
gen_ids = model.generate(input_ids,
                         max_length=128,
                         repetition_penalty=2.0,
                         pad_token_id=tokenizer.pad_token_id,
                         eos_token_id=tokenizer.eos_token_id,
                         bos_token_id=tokenizer.bos_token_id,
                         use_cache=True)
generated = tokenizer.decode(gen_ids[0])
print(generated)
```

잠뜰 TV를 통해서도 볼 수 있다.

이번 전시는 지난해 11월부터 올 1월까지 진행한 '2018 서울국제광고제'에서 수상한 작품들로 구성됐다.

전시에는 총 4000여 점의 작품이 출품됐으며 이중 약 1000점이 넘는 작품들이 소개된다.

특히 이번 전시회에서는 국내 최초로 선보이는 미디어아트 작품을 비롯해 다양한 장르의 광고작품들이 대거 공개돼 눈길을 끈다.

미디어 아트는 현대인의 일상을 표현하는 예술 장르로 최근 주목받고 있는 디지털 기술을 활용해 새로운 형태의 광고를 제작하고 이를 활용한 마케팅을 시도하는 것을 말한다.

또한, 미디어는 인간의 삶을 풍요롭게 하는 매개체로 인식되고 있으며, 이러한 메시지를 전달하는 매체

토큰ナイ저,

모델 불러오기

-> 텍스트 입력하고 사용하면 끝

transformer 학습

```
tokenizer = PreTrainedTokenizerFast.from_pretrained("skt/kogpt2-base-v2", bos_token='</s>', eos_token='</s>', unk_token='<unk>', pad_token='<pad>', mask_token='<mask>')
model = GPT2LMHeadModel.from_pretrained("skt/kogpt2-base-v2")

from transformers import TextDataset, DataCollatorForLanguageModeling

# 텍스트 데이터셋 생성

dataset = TextDataset(tokenizer=tokenizer, file_path='./test.txt', block_size=128, overwrite_cache=False)

# 언어 모델링을 위한 데이터 콜레이터 생성
data_collator = DataCollatorForLanguageModeling(tokenizer=tokenizer, mlm=False)

# 학습 설정
training_args = TrainingArguments(
    output_dir='./results',
    overwrite_output_dir=True,
    num_train_epochs=3,
    per_device_train_batch_size=64,
    save_steps=10_000,
    save_total_limit=2,
)

# Trainer 객체 생성 및 학습
trainer = Trainer(
    model=model,
    args=training_args,
    data_collator=data_collator,
    train_dataset=dataset,
)

# 학습 실행
trainer.train()
```

Could not render content for 'application/vnd.jupyter.widget-view+json'
{"model_id": "f73571e964a042faae179eebe485ebde", "version_major": 2, "version_minor": 0}

실증 랩에서 돌린 뒤(5시간),
결과를 보러 오니 데이터를 못 불러옴



[EPOCH 3]

100%

1836/1836 [8:09:68<00:00, 3.74it/s]



[EPOCH 100]

100%

61192/61200 [4:34:35<00:02, 3.74it/s]

loss, epoch, norm-value 같은
실증랩에 있었지만 날라감...

transformer 결과

키워드: “잠뜰”

3EPOCH

Generated sequence 1: 잠뜰님!!!!!!!!!!!!!!

100EPOCH

Generated sequence 1: 잠뜰티비는 야생 베이스 컨텐츠가 젤루 재밌는것 같아용

안녕하세요 저는 학교 도덕선플달기 수행평가를 하고있는 중학생입니다.
전부터 꾸준히 매일매일 재밌고 신박한 영상을 올려주셔서 감사하고 앞으로도
영상많이올려주세요!!!

transformer 결과

키워드: “1등”

3EPOCH

100 EPOCH

Generated sequence 1: 1등 ㅋㅋㅋ
<a href=""<https://www.youtube.com/watch?v=5IbnKeHeSMk&t=294>">4:54 라더님 웃는 게 웬지 모르게 웃겨서 저도 모르게 웃었네요 ㅋㅋㅋㅋㅋ

transformer + Gensim 결과

키워드：“마크”

3EPOCH

판시 잠뜰님 ..?
계시네요
공포가
모드 퀄리티 대박
에서 덕개님 연기 너무 잘하시네요

100EPOCH

판돈으로 게임을 하기도 했으며 왕국의 여왕이자 마녀의 눈을 피해 도망치기도 했으며 어인들의 나라에서 살아남기도 했으며 지독히도 끔찍한 악몽에서 깨어나기도 했으며 그리고 동료였던 그러나 적이 된 그들과 싸우기도 했습니다. 그리고 동료였던 그러나 적이 된 그들과 싸우기도 했습니다. 그리고 동료였던 그러나 적이 된 그들과 싸우기도 했습니다.

계십니다

잠뜰님은 일만 하시면 극한 직업이시네요..

잠뜰님 꿈뜰이들의 마음도 찾아주세요

이번이 몇 번째 직업이죠?

공포물이였다가 다시 평범하진 아는 엔딩

수상한 사건내가 구독한가?

..?

공포가 웃집 좀비

이웃집 좀비 안돼

벌써 설레네요. 어릴 때부터 잠뜰님을 봤는데 항상 멋지세요. 존경합니다. 영원한 팬이에요!!!

왜 마지막이야 안돼!!!!!!!!!!

.

모드인가??? 대박이다 대박.. ..? 이거지 예

저번 콘텐츠에서 봤을 때 너무 재밌었는데 또 나와서 너무 좋네요

너무 마지막에 진심 아니냐고요 근데 시리즈 나오면 완전 꿀잼일듯

..?

Flask와 연결하기 전...

The screenshot shows a Stack Overflow question page with the following details:

- Question Title:** To obtain the channel id you can view the source code of the channel page and find either `data-channel-external-id="UCjXfkj5iapKJrhYfAF9ZGg"` or `"externalId": "UCjXfkj5iapKJrhYfAF9ZGg"`.
- Upvotes:** 258
- Content:** `UCjXfkj5iapKJrhYfAF9ZGg` will be the channel ID you are looking for.
- Author:** Lemmings19 (1,433 reputation)
- Answered:** May 1, 2013 at 20:52 by Henry Moshkovich (2,632 reputation)
- Comments:** 6 (Naveed Ahmad), 2 (klauskpm), 21 (mjlescano), 2 (derekantican), 15 (TD540)
- Links:** [Show 8 more comments](#)
- Second Answer:** An easy answer is, your YouTube Channel ID is `UC + {YOUR_ACCOUNT_ID}`. To be sure of your YouTube Channel ID or your YouTube account ID, access the [advanced settings](#) at your [settings page](#). And if you want to know the YouTube Channel ID for any channel, you could use the [solution](#) @mjlescano gave.
Link: https://www.googleapis.com/youtube/v3/channels?key={YOUR_API_KEY}&forUsername={USER_ID}
- Author:** TD540 (Nov 16, 2020 at 7:14)
- Upvotes:** 79
- Comments:** 1 (right here)
- Answered:** Apr 1, 2019 at 14:33 by TD540 (Nov 16, 2020 at 7:14)

유튜브

1. 채널 제목
2. 구독자 수
3. 영상 개수
4. 채널 설명



소프트코딩으로 정보 갖고오는 함수 구현

Flask와 연결하기 전...

```
36 # 유튜브 채널 ID
37 CHANNEL_ID = 'UCg7rkxrTnIhiHEpXY1ec9NA'
38
39 # 채널 정보 가져오기
40 channel_info = get_channel_info(CHANNEL_ID)
41
42 # 채널 정보 출력
43 if __name__ == '__main__':
44     print('채널 제목:', channel_info['channel_title'])
45     print('구독자 수:', channel_info['subscriber_count'])
46     print('영상 개수:', channel_info['video_count'])
47     print('채널 설명:', channel_info['channel_description'])
48
```

PROBLEMS 26 OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER COMMENTS

영상 개수: 2522

채널 설명: 마인크래프트를 비롯한 다양한 게임들을
재밌는 콘텐츠로 만들고 있습니다 😊

[비즈니스 문의] : contact@asmb1.kr
[기타 문의] : skhh1221@gmail.com

구독하기 버튼과 알림 버튼을 누르시면 매일 재밌는 영상을
제일 먼저 만나보실 수 있습니다!
우리 꿈뜰이분들, 시청자분들 항상 감사드려요

#마인크래프트 #게임

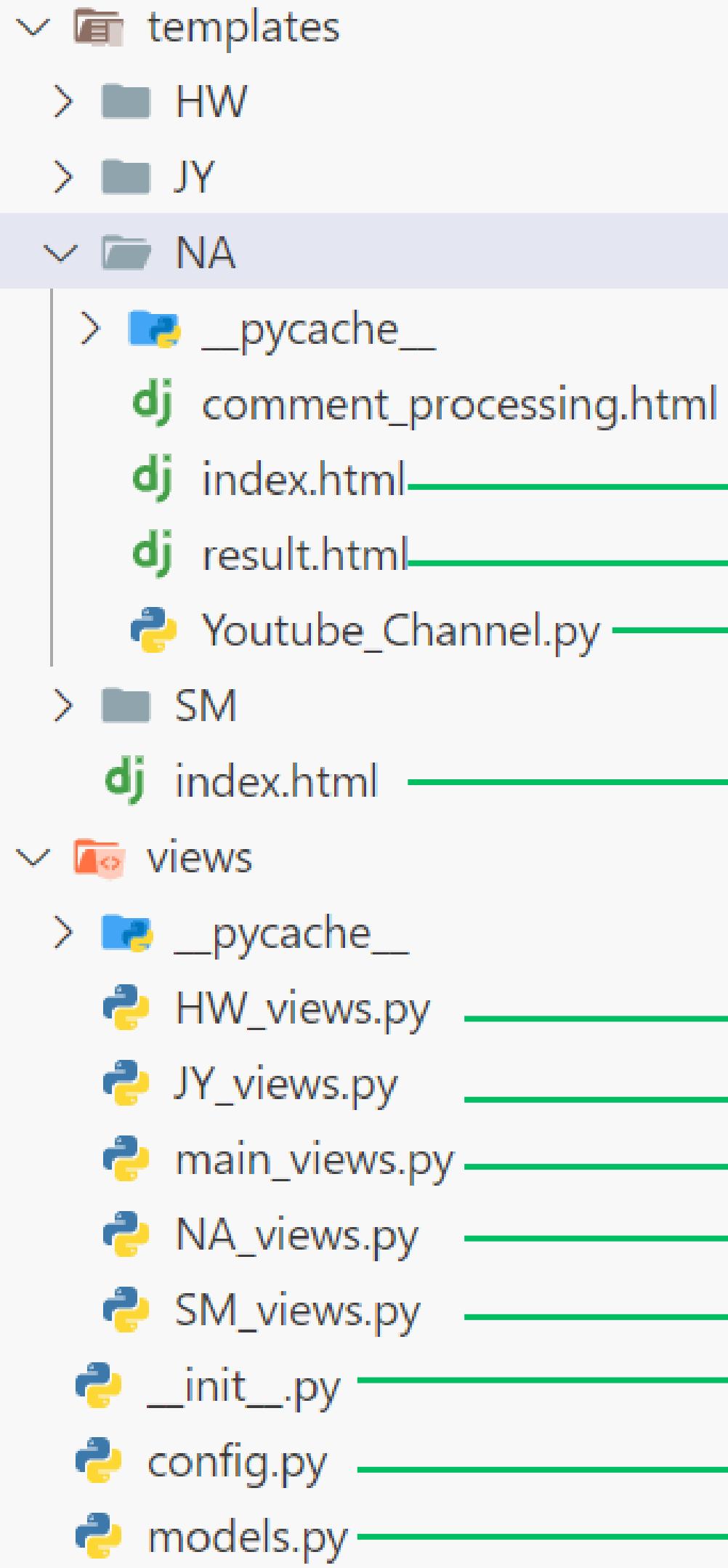
PS C:\Users\mathn\Desktop\KDT05_FLASK_Project>

웹에 데이터 전송하기로 결정

Flask(Web) 구성 1/2



Flask(Web) 구성 2/2



templates

HW

JY

NA

pycache

dj comment_processing.html

dj index.html

dj result.html

py Youtube_Channel.py

SM

dj index.html

views

pycache

py HW_views.py

py JY_views.py

py main_views.py

py NA_views.py

py SM_views.py

py __init__.py

py config.py

py models.py

댓글 수정(update), 삭제(delete) 관리 화면

댓글 생성 화면

댓글 생성 결과 화면

유튜브 채널 정보 수집 함수

첫 화면

현우님의 Blueprint

주영님의 Blueprint

첫 화면 Blueprint

노아님의 화면 Blueprint

시명님의 화면 Blueprint

처음 실행되는 파일, 데이터베이스 연결 및 Blueprint 연결

MySQL 및 SQLAlchemy 관련 변수 선언

데이터베이스 클래스 생성

NA/index.html

기본값 input 2개와 submit 버튼 구성

기본적인 설명과
제출 버튼 상호작용 요소 추가

키워드 관련 댓글 생성 프로그램

기본값 = "수현"
Ko_GPT2를 활용하여 만든 트랜스포머 기반 댓글
생성기
값 입력시 : 4초 소요

기본값 = "마인크래프트"
word2vec를 활용하여 만든 유사 단어 추출기
값 입력시 : 0.2초 소요

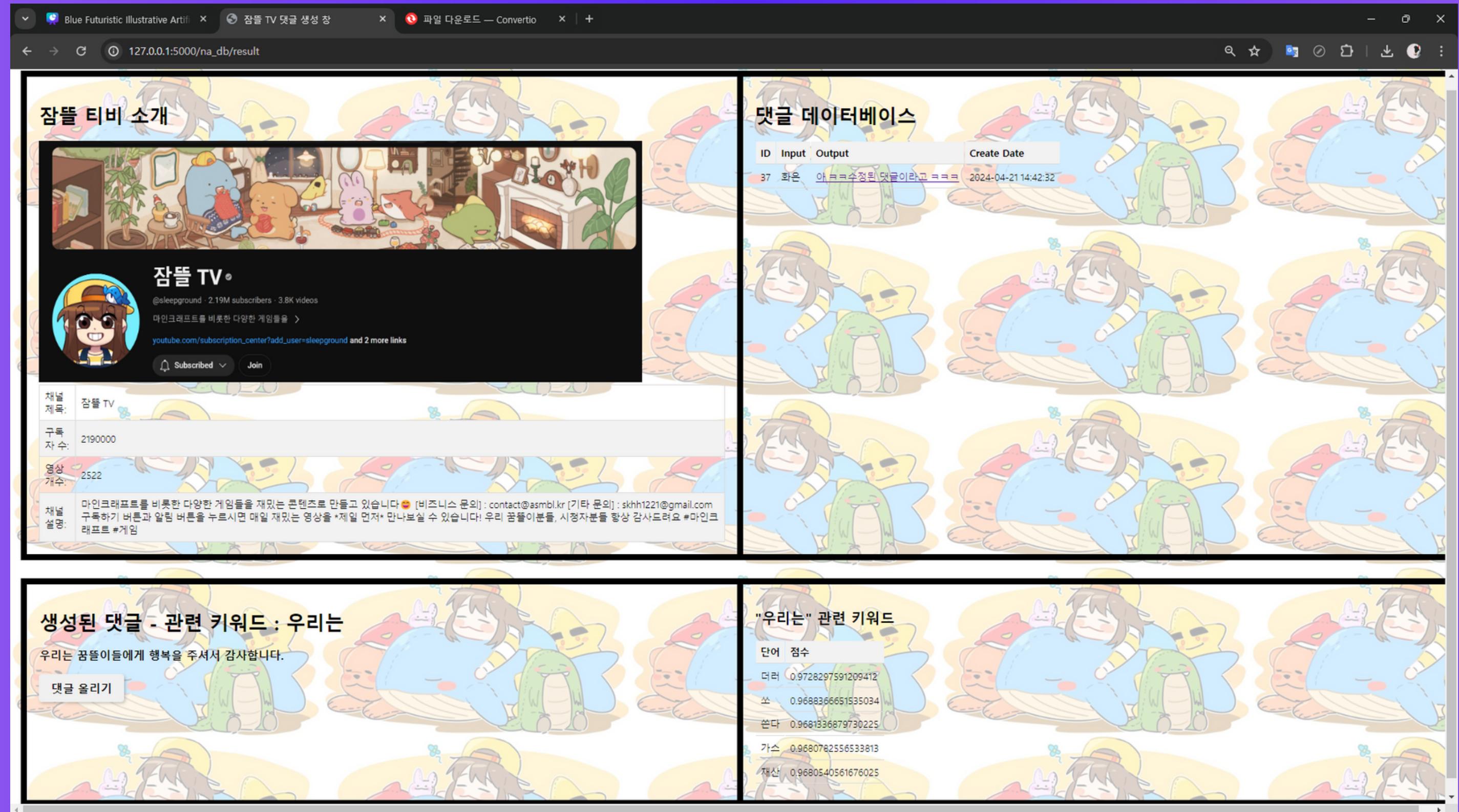
수현

마인크래프트

제출

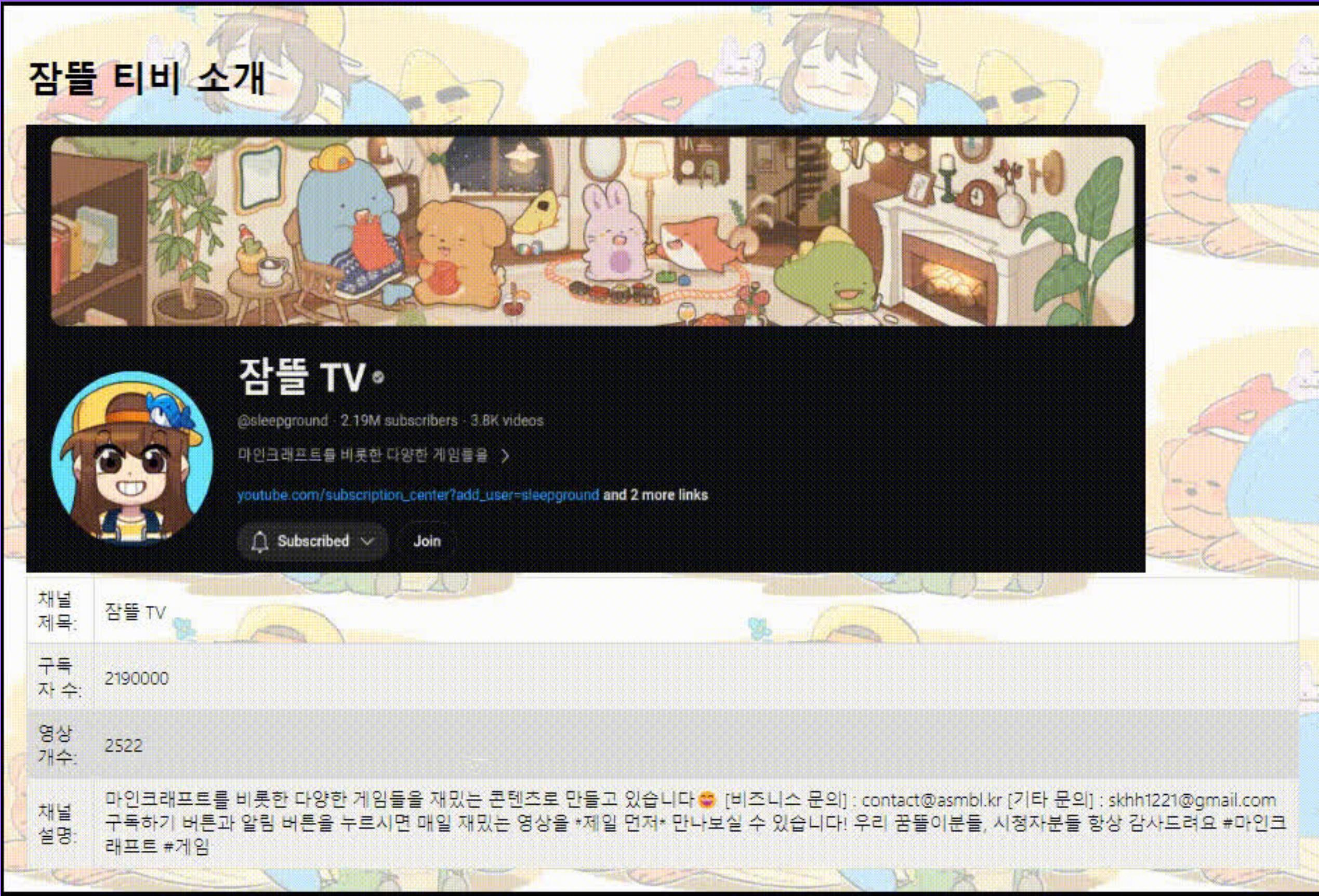


NA/result.html



크게 4가지로 구성

NA/result.html: 잠뜰 티비 소개



Youtube_Channel.py

get_channel_info를 사용하여
정보 획득 후 표시

```
<table id="channel-info-table">
  <tr>
    <td>채널 제목:</td>
    <td>{{ channel_info.channel_title }}</td>
  </tr>
  <tr>
    <td>구독자 수:</td>
    <td>{{ channel_info.subscriber_count }}</td>
  </tr>
  <tr>
    <td>영상 개수:</td>
    <td>{{ channel_info.video_count }}</td>
  </tr>
  <tr>
    <td>채널 설명:</td>
    <td>{{ channel_info.channel_description }}</td>
  </tr>
</table>
```

NA/result.html: 데이터베이스 소개

댓글 데이터베이스

ID	Input	Output	Create Date
43	잠은 다	잠은다더니 그 컨텐츠였구나... 이게 뭔뜻인가요?	2024-04-21 16:45:37
42	덕개	덕개님 연기톤만 들으면 을 거 같애.. 뭔가 나중에 절이 필요하지만 절을 못 찾아서 삐리뽀를 버리거나 삐리뽀가 자의적 판단을 해서 배신 할 것 같아요... 진짜 삐리뽀만 남는거아냐??? ...?	2024-04-21 16:45:23
41	덕개	덕개님 연기톤만 들으면 을 거 같애.. 뭔가 나중에 절이 필요하지만 절을 못 찾아서 삐리뽀를 버리거나 삐리뽀가 자의적 판단을 해서 배신 할 것 같아요... 진짜 삐리뽀만 남는거아냐??? ...?	2024-04-21 16:45:20
40	소중	소중하고 같이 있고싶죠.. 덕개님의 연기는 정말 대박이에요 보는 내내 슬퍼서 눈물이 나왔답니다 그리고 수현님의 배신? 합정은 설마했지만.. 과연 어떻게 진행될까요.. 연기가 배우해도 될거같다는 생각합니다!! 너무 멋있고 다들 연기력이 좋으니까! 활수현님 과 덕개님이 빠진게 너무 슬프고 수현님도 살아	2024-04-21 16:45:07
39	장난	장난쟁이처럼 뛰어왔습니다 어디서 많이 본 도박장?.. 제목도 경고문도 다 재그네용 오늘도 재밌게 볼게요	2024-04-21 16:44:49
38	우리 는	우리는 꿈풀이들에게 행복을 주셔서 감사합니다.	2024-04-21 16:44:32

최신 등록 순으로 DB 출력

```
{% if table_list %}  
<table>  
  <thead>  
    <tr>  
      <th>ID</th>  
      <th>Input</th>  
      <th>Output</th>  
      <th>Create Date</th>  
    </tr>  
  </thead>  
  <tbody>  
    {% for data in table_list %}  
      <tr>  
        <td>{{ data.id }}</td>  
        <td>{{ data.input }}</td>  
        <td>  
          <a href="/na_db/detail/{{ data.id }}">{{ data.output }}</a>  
        </td>  
        <td>{{ data.create_date }}</td>  
      </tr>  
    {% endfor %}  
  </tbody>  
</table>  
{% else %}  
<p>댓글이 없습니다</p>  
{% endif %}
```

NA/result.html : 댓글 등록 소개(필살기)

```
{% if result %}  
<h1>생성된 댓글 - 관련 키워드 : {{cache_input}}</h1>  
<h3>{{result}}</h3>  
<form action="/na_db/upload" method="POST">  
  <input type="hidden" name="cache_input" value="{{cache_input}}"/>  
  <input type="hidden" name="cache_output" value="{{cache_output}}"/>  
  <input type="submit" value="댓글 올리기"/>  
</form>  
{% endif %} {% if not result and not result2%}  
<button onclick="window.location.href='/na_db/'">  
  댓글 다시 만들기</button>  
>{% endif %}
```

“댓글 올리기” 버튼을 클릭하면 바로 해당 화면에 뜰 수 있도록 **hidden** 속성을 활용하여 데이터 값을 저장 후 전송

생성된 댓글 - 관련 키워드 : 힘들어

힘들어요잠뜰님 영상을 보면서 눈물 흘리는 거 진짜 뜸한데 정말 인상깊은 영상입니다.. 정말 놀랐어요 덕님이 연기를 잘 하신다는 걸 알고는 있었지만..진짜 이번 영상 레전드중에 레전드네요 다른 장면들도 다 너무 멋있네요 다른 장면들도 다 너무 웃기고 슬프고 감동적이고.. 제가 다 반대로 잠뜰님이 연기를 너무 잘 하셔서 모든 감정들이 다들 힘들어요

댓글 올리기

NA/result.html: 연관 단어 출력

부가적 기능
해당 기능을 조합하여,
연관 단어 - 댓글 생성 기능을 구현했으나,
댓글 생성 당 5초가 걸리기에 반영하지 않음



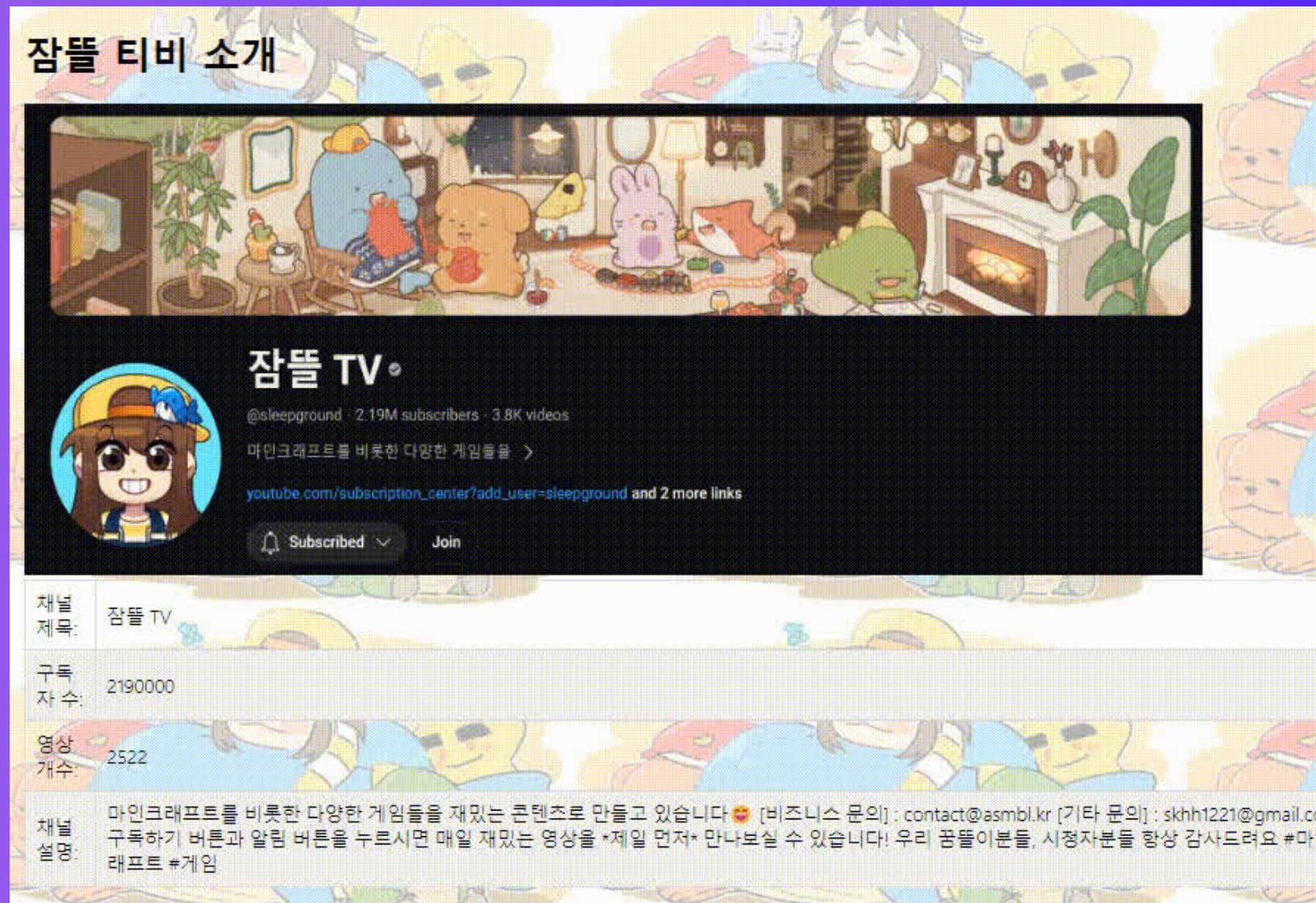
NA/comment_processing.html

ID	Input	Output	Create Date
43	잠은 다	잠은다더니 그 컨텐츠였구나... 이게 뭔뜻인가요?	2024-04-21 16:45:37
42	덕개	덕개님 연기톤만 들으면 을 거 같애.. 뭔가 나중에 절이 필요하지만 절을 못 찾아서 삐리뽀를 버리거나 삐리뽀가 자의적 판단을 해서 배신 할 것 같아요... 진짜 삐리뽀만 남는거아냐??? ...?	2024-04-21 16:45:23
41	덕개	덕개님 연기톤만 들으면 을 거 같애.. 뭔가 나중에 절이 필요하지만 절을 못 찾아서 삐리뽀를 버리거나 삐리뽀가 자의적 판단을 해서 배신 할 것 같아요... 진짜 삐리뽀만 남는거아냐??? ...?	2024-04-21 16:45:20
40	소중	소중하고 같이 있고싶죠.. 덕개님의 연기는 정말 대박이에요 보는 내내 슬퍼서 눈물이 나왔답니다 그리고 수현님의 배신? 합정은 설마했지만.. 과연 어떻게 진행될까요.. 연기가 배우해도 될거같다는 생각합니다!! 너무 멋있고 다들 연기력이 좋으니까! 활수현님과 덕개님이 빠진게 너무 슬프고 수현님도 살아	2024-04-21 16:45:07
39	장난	장난쟁이처럼 뛰어왔습니다 어디서 많이 본 도박장?.. 제목도 경고문도 다 재그네용 오늘도 재밌게 볼게요	2024-04-21 16:44:49
38	우리 는	우리는 꿈풀이들에게 행복을 주셔서 감사합니다.	2024-04-21 16:44:32

하이퍼링크를 누르면,
댓글 관리창으로 넘어감

NA/comment_processing.html

“삭제”버튼을 누를 시, DB에서 해당 id를 가진 행 삭제
“수정”버튼을 누를 시, DB에서 해당 id를 가진 행 내용을 바꿈



ID	Input	Output	Create Date
44	힘들어	힘들어요 잠뜰님 영상을 보면서 눈물 흘리는 거 진짜 뜻한데 정말 인상깊은 영상입니다. 정말 놀랐어요 덕님이 연기를 잘 하신다는 걸 알고는 있었지만 진짜 이번 영상 레전드중에 레전드네요 다른 장면들보다 너무 멋있네요 다른 장면들도 다 너무 웃기고 슬프고 감동적이고 제가 다 반대로 잠뜰님이 연기를 너무 잘 하셔서 모든 감정들이 다를 힘들어요	2024-04-21 16:47:45
43	잠온다	잠온다더니 그 컨텐츠였구나... 이게 뭔뜻이가요?	2024-04-21 16:45:37
42	덕개	덕개님 연기톤만들으면 을 거 같애 원가 나중에 졸이 필요하지만 철을 못 찾아서 빼리쁘를 버리거나 빼리쁘가 자의적 판단을 해서 배신 할 것 같아요 진짜 빼리쁘만 남는거아냐??????	2024-04-21 16:45:23
41	덕개	덕개님 연기톤만들으면 을 거 같애 원가 나중에 졸이 필요하지만 철을 못 찾아서 빼리쁘를 버리거나 빼리쁘가 자의적 판단을 해서 배신 할 것 같아요 진짜 빼리쁘만 남는거아냐??????	2024-04-21 16:45:20
40	소중	소중하고 같이 있고싶죠. 덕개님의 연기는 정말 대박이에요 보는 내내 슬퍼서 눈물이 나왔답니다 그리고 수현님의 배신! 학점은 살마했지만 과연 어울리게 진화될까요 연기가 배우해도 들판같다는 생각합니다 너무 멋있고 다들 연기력이 좋으니까 활수현님과 덕개님이 빠지게 너무 슬프고 수현님도 살아	2024-04-21 16:45:07
39	장난	잔난쟁이처럼 뛰어왔습니다 어디서 많이 본 도박장.. 제목도 경고문도 다개그네용 오늘도 재밌게 볼게요	2024-04-21 16:44:49
38	우리	우리는 꿈풀이들에게 행복을 주셔서 감사합니다.	2024-04-21 16:44:32

views/NA_views.py

```
#=====
# '/' 경로로 들어오는 GET 요청에 대해 index 페이지를 렌더링합니다.
# '/result' 경로로 들어오는 GET 및 POST 요청에 대해서는 결과를 보여줍니다.
# 사용자가 텍스트를 입력하면 해당 텍스트를 기반으로 댓글을 생성하고,
# 사용자가 유사도를 확인하고자 할 때는 Word2Vec 모델을 사용하여 가장 유사한 단어를 찾습니다.
# 결과를 템플릿으로 렌더링하여 사용자에게 보여줍니다.
#=====

@bp.route('/')
def index():
    return render_template("NA/index.html")
```

views/NA_views.py

```
#=====
# '/result' 경로로 들어오는 GET 및 POST 요청에 대해서는 결과를 보여줍니다.
# 사용자가 텍스트를 입력하면 해당 텍스트를 기반으로 댓글을 생성하고,
# 사용자가 유사도를 확인하고자 할 때는 Word2Vec 모델을 사용하여 가장 유사한 단어를 찾습니다.
# 결과를 템플릿으로 렌더링하여 사용자에게 보여줍니다.
=====

@bp.route('/result', methods=['GET', 'POST'])
def show_result():

    req_dict = request.form.to_dict() # 값들 갖고 오기
    text = req_dict.get('name')
    simir_text = req_dict.get('simir')

    if text : # 댓글 생성 함수=====1-1
        result = generate_text(text)
        # 알파벳과 특수기호를 제거하는 정규표현식
        pattern = r'[^가-힣\.,!?]+'
        # 각 내용에 대해 정규표현식을 적용하여 필터링
        result = re.sub(pattern, '', result)

    if simir_text: # 유사도 함수=====1-2
        word2vec = Word2Vec.load("명노아/model/Gensim.model")
        result2 = word2vec.wv.most_similar(simir_text, topn=5)

    # 테이블 값 갖고오기
    table_list = NA_DB.query.order_by(NA_DB.create_date.desc())
    CHANNEL_ID = 'UCg7rkxrTnIhiHEpXY1ec9NA'

    # 채널 정보 가져오기
    channel_info = get_channel_info(CHANNEL_ID)
    # 값이 있음에 따라 HTML요소로 넘겨주기
    if text and simir_text:
        return render_template("NA/result.html", result=result, table_list=table_list, result2=result2, cache_input=text,
        cache_output=result, channel_info=channel_info)
    elif text:
        return render_template("NA/result.html", result=result, table_list=table_list, cache_input=text, cache_output=result,
        channel_info=channel_info)
    elif simir_text:
        return render_template("NA/result.html", table_list=table_list, result2=result2, cache_input=text, cache_output=result,
        channel_info=channel_info)
    else :
        return render_template("NA/result.html", table_list=table_list, channel_info=channel_info)
```

views/NA_views.py

```
#=====
# 이 함수는 특정 댓글의 세부 정보를 가져와서 보여줍니다.
#
# Parameters:
#     comment_id (int): 댓글의 고유 식별자입니다.
#
# Returns:
#     template: NA/comment_processing.html 템플릿을 렌더링하고, 해당 댓글의 정보를 전달합니다.
#=====

@bp.route('/detail/<int:comment_id>', methods=['GET', 'POST'])
def detail(comment_id):
    comment = NA_DB.query.get(comment_id)
    return render_template("NA/comment_processing.html", comment=comment)
```

views/NA_views.py

```
#=====
# 이 함수는 특정 댓글을 삭제합니다.
#
# Parameters:
#     comment_id (int): 삭제할 댓글의 고유 식별자입니다.
#
# Returns:
#     redirect: 'NA.show_result' 라우트로 리디렉션합니다.
#=====

@bp.route('/delete/<int:comment_id>', methods=['GET', 'POST'])
def delete_comment(comment_id):
    comment = NA_DB.query.get(comment_id)
    db.session.delete(comment)
    db.session.commit()
    table_list = NA_DB.query.order_by(NA_DB.create_date.desc())
    return redirect(url_for('NA.show_result'))
```

views/NA_views.py

```
#=====
# 이 함수는 사용자가 입력한 댓글을 업로드합니다.
#
# Returns:
#     redirect: 'NA.show_result' 라우트로 리디렉션합니다.
#=====

@bp.route('/upload', methods=['GET', 'POST'])
def upload_comment():
    req_dict = request.form.to_dict() # 값을 갖고 오기
    text = req_dict.get('cache_input')
    result= req_dict.get('cache_output')
    q = NA_DB(input=text, output=result, create_date=datetime.now())
    db.session.add(q)
    db.session.commit()
    return redirect(url_for('NA.show_result'))
```

설연

