

Transformer 사용하기

✓ 1.환경준비

✓ (1) 라이브러리 설치

```
# !pip install transformers
# colab에서는 이미 설치되어 있음
```

✓ (2) 라이브러리 Import

```
from transformers import pipeline
```

✓ 2.다양한 활용

- 영어로 테스트 할 수 있는 예제를 사용해 봅니다.

✓ (1) 감성 분석

- transformer로 생성된 감성분석 모델을 다운받아 사용해 봅니다.

```
classifier = pipeline(task = "sentiment-analysis", model = 'bert-base-multilingual-cased')

/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingf
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or dat
warnings.warn(
config.json: 100% 625/625 [00:00<00:00, 7.94kB/s]
model.safetensors: 100% 714M/714M [00:11<00:00, 88.6MB/s]
Some weights of BertForSequenceClassification were not initialized from the model checkpoint at
You should probably TRAIN this model on a down-stream task to be able to use it for predictions .
tokenizer_config.json: 100% 49.0/49.0 [00:00<00:00, 349B/s]
vocab.txt: 100% 996k/996k [00:00<00:00, 3.99MB/s]
tokenizer.json: 100% 1.96M/1.96M [00:00<00:00, 6.56MB/s]
```

```
# sentiment-analysis 모델 파이프라인 생성
# 기본값 : distilbert-base-uncased-finetuned-sst-2-english
```

```
classifier = pipeline("sentiment-analysis")

No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned-sst-2-english a
Using a pipeline without specifying a model name and revision in production is not recommended.
config.json: 100% 629/629 [00:00<00:00, 30.1kB/s]
model.safetensors: 100% 268M/268M [00:05<00:00, 33.9MB/s]
tokenizer_config.json: 100% 48.0/48.0 [00:00<00:00, 2.44kB/s]
vocab.txt: 100% 232k/232k [00:00<00:00, 11.5MB/s]
```

```
# 모델 사용
text = ["I've been waiting for a HuggingFace course my whole life.",
        "I hate this so much!",
        "I have a dream.",
        "She was so happy."]

classifier(text)

[{'label': 'POSITIVE', 'score': 0.9598048329353333},
 {'label': 'NEGATIVE', 'score': 0.9994558691978455},
 {'label': 'POSITIVE', 'score': 0.9997022747993469},
 {'label': 'POSITIVE', 'score': 0.9998832941055298}]
```

- 영어 문장을 2~3개 추가해서 긍/부정 확률을 확인해 봅시다.

```
text = ['I feel good because I have a girlfriend',
        'The game boss is too difficult',
        'I succeeded in finding a job and went out to eat with my family.'
        ]

classifier(text)

[{'label': 'POSITIVE', 'score': 0.9998742341995239},
 {'label': 'NEGATIVE', 'score': 0.9997159838676453},
 {'label': 'POSITIVE', 'score': 0.9974942207336426}]
```

코딩을 시작하거나 AI로 코드를 선택하세요.

✓ (2) Zero-shot classification

- Target 범주를 정해주고, 분류하도록 해 봅시다.

```
# Zero-shot 분류 파이프라인 생성
classifier = pipeline(task = "zero-shot-classification", model="facebook/bart-large-mnli")

config.json: 100% 1.15k/1.15k [00:00<00:00, 67.0kB/s]
model.safetensors: 100% 1.63G/1.63G [00:17<00:00, 134MB/s]
tokenizer_config.json: 100% 26.0/26.0 [00:00<00:00, 1.33kB/s]
vocab.json: 100% 899k/899k [00:00<00:00, 12.0MB/s]
merges.txt: 100% 456k/456k [00:00<00:00, 16.3MB/s]
tokenizer.json: 100% 1.36M/1.36M [00:00<00:00, 20.1MB/s]

# 후보 레이블 지정
candidate_labels = ["tech", "politics", "business", "finance"]

# 분류하고자 하는 텍스트
text = "This is a tutorial about using transformers in natural language processing."

# 분류 수행
result = classifier(text, candidate_labels)

# 결과 출력
print(f"Labels: {result['labels']}")
print(f"Scores: {result['scores']}")

Labels: ['tech', 'business', 'politics', 'finance']
Scores: [0.9759427905082703, 0.014607219956815243, 0.005425842013210058, 0.0040240720845758915]
```

- 새로운 후보 레이블을 지정하고, 문장을 입력하여 분류를 시도해 봅시다.

```
# 후보 레이블 지정
candidate_labels = ["tech", "politics", "business", "finance", "sports"]
```

```
# 분류하고자 하는 텍스트
text = "Korea will win this Asian Under-23 Cup"

# 분류 수행
result = classifier(text, candidate_labels)

# 결과 출력
print(f"Labels: {result['labels']}")
print(f"Scores: {result['scores']}")

Labels: ['sports', 'tech', 'business', 'politics', 'finance']
Scores: [0.8442844152450562, 0.07076362520456314, 0.04445340856909752, 0.022742336615920067, 0.017756246030330658]
```

✓ (3) 번역

- 한국어를 영어로 번역해 봅시다.

```
# 한국어에서 영어로 번역하는 파이프라인 생성
translator_ko_to_en = pipeline(task = "translation", model="haee9/translation_en_ko")

config.json: 100% 1.41k/1.41k [00:00<00:00, 76.2kB/s]
model.safetensors: 100% 310M/310M [00:03<00:00, 87.2MB/s]
generation_config.json: 100% 288/288 [00:00<00:00, 14.5kB/s]
tokenizer_config.json: 100% 818/818 [00:00<00:00, 52.6kB/s]
source.spm: 100% 842k/842k [00:00<00:00, 9.99MB/s]
target.spm: 100% 813k/813k [00:00<00:00, 11.3MB/s]
vocab.json: 100% 1.85M/1.85M [00:00<00:00, 23.3MB/s]
special_tokens_map.json: 100% 74.0/74.0 [00:00<00:00, 4.29kB/s]
/usr/local/lib/python3.10/dist-packages/transformers/models/arian/tokenization_arian.py:197: U
warnings.warn("Recommended: pip install sacrosanct")
```

```
# 번역하고자 하는 한국어 텍스트
text_ko = "안녕하세요, 오늘 미세먼지가 무척 심하네요."

# 번역 수행
translated_text_en = translator_ko_to_en(text_ko, max_length=60)

# 번역된 영어 텍스트 출력
print(f"Translated Text (KO to EN): {translated_text_en[0]['translation_text']}")

Translated Text (KO to EN): Hello, there's a lot of fine dust today.
```

- 다양한 한글 문장을 입력해서 영어로 번역해 봅시다.

```
# 번역하고자 하는 한국어 텍스트
text_ko = "저는 취업을 하기위해 kt에이블 스쿨에서 공부 중에 있습니다"

# 번역 수행
translated_text_en = translator_ko_to_en(text_ko, max_length=60)

# 번역된 영어 텍스트 출력
print(f"Translated Text (KO to EN): {translated_text_en[0]['translation_text']}")

Translated Text (KO to EN): I'm studying at KT Aable School to get a job.
```

코딩을 시작하거나 AI로 코드를 생성하세요.

✓ (4) 요약

✓ 1) 영문 텍스트 요약

```
# 텍스트 요약 파이프라인 생성, 여기서는 BART 모델을 사용
```

```
summarizer = pipeline(task = "summarization", model="facebook/bart-large-cnn")
```

config.json: 100%	1.58k/1.58k [00:00<00:00, 59.0kB/s]
model.safetensors: 100%	1.63G/1.63G [00:17<00:00, 163MB/s]
generation_config.json: 100%	363/363 [00:00<00:00, 17.1kB/s]
vocab.json: 100%	899k/899k [00:00<00:00, 17.3MB/s]
merges.txt: 100%	456k/456k [00:00<00:00, 8.98MB/s]
tokenizer.json: 100%	1.36M/1.36M [00:00<00:00, 41.1MB/s]

```
# 요약하고자 하는 여러 문장이나 긴 텍스트
```

```
text = """
```

```
The global economy is facing unprecedented challenges due to the impact of the COVID-19 pandemic. Many countries are experiencing significant downturns, with industries such as travel, hospitality, and retail particularly affected. Governments around the world are implementing various fiscal and monetary policies in an attempt to mitigate the economic fallout. This includes measures such as lowering interest rates, providing financial assistance to businesses and individuals, and introducing stimulus packages aimed at boosting economic activity. Despite these efforts, the path to recovery is expected to be long and uncertain, with many experts predicting a slow return to pre-pandemic levels of economic growth.
```

```
"""
```

```
# 텍스트 요약 수행
```

```
summary = summarizer(text, max_length=80, min_length=30, do_sample=False)
```

```
# 요약된 텍스트 출력
```

```
print(f"Summary: {summary[0]['summary_text']}")
```

```
Summary: The global economy is facing unprecedented challenges due to the impact of the COVID-19 pandemic. Many countries are experiencing significant
```

- 영문으로 된 기사나 소설 등에서 한 단락을 붙여 놓고 요약을 시도해 보시다.
- 시도해 볼 만한 사이트
 - CNN(<https://edition.cnn.com/>)
 - Nature(<https://www.nature.com/>)
 - Gartner(<https://www.gartner.com/en>)
 - 영문소설(<https://www.gutenberg.org/>)

```
# 요약하고자 하는 여러 문장이나 긴 텍스트
```

```
text = """Once Trump's attorneys and the district attorney's office used up their 10 peremptory strikes to remove jurors, things moved quickly.
```

```
The judge rejected Trump's challenges to remove jurors for cause because they had expressed negative opinions about Trump, telling the former president's lawyer that the seated jury includes seven men and five women. The new jurors empaneled Thursday include an investment banker, a security engineer, a retired wealth manager, a speech therapist, and a speech therapist. The jury pool was quickly whittled down from a second panel of 96 Thursday morning, after nearly 50 prospective jurors said they did not feel they could be impartial.
```

```
# 텍스트 요약 수행
```

```
summary = summarizer(text, max_length=80, min_length=30, do_sample=False)
```

```
# 요약된 텍스트 출력
```

```
print(f"Summary: {summary[0]['summary_text']}")
```

```
Summary: The seated jurors empaneled Thursday include an investment banker, a security engineer, a retired wealth manager, a speech therapist, and a speech therapist mo
```

코딩을 시작하거나 AI로 코드를 선택하세요.

코딩을 시작하거나 AI로 코드를 선택하세요.

코딩을 시작하거나 AI로 코드를 선택하세요.

✓ 2) 한글 텍스트 요약

```
# 텍스트 요약 파이프라인 생성, 여기서는 BART 모델을 사용
```

```
summarizer = pipeline("summarization", model="ainize/kobart-news")
```

```
config.json: 100% 1.45k/1.45k [00:00<00:00, 36.8kB/s]
You passed along `num_labels=3` with an incompatible id to label map: {'0': 'NEGATIVE', '1': 'PO'
You passed along `num_labels=3` with an incompatible id to label map: {'0': 'NEGATIVE', '1': 'PO'
pytorch_model.bin: 100% 496M/496M [00:09<00:00, 20.6MB/s]
tokenizer_config.json: 100% 302/302 [00:00<00:00, 15.5kB/s]
tokenizer.json: 100% 682k/682k [00:00<00:00, 27.5MB/s]
special_tokens_map.json: 100% 239/239 [00:00<00:00, 15.4kB/s]
```

소설 상록수 중에서...

```
input_text = ''
```

가을 학기가 되자, ∞일보사에서 주최하는 학생계몽운동에 참가하였던 대원들이 돌아왔다. 오늘 저녁은 각처에서 모여든 대원들을 위로하는 다과회가 그 신문사 누싱 오록백 명이나 수용할 수 있는 대강당에는 전 조선의 방방곡곡으로 흩어져서 한여름 동안 땀을 흘려 가며 활동한 남녀 대원들로 빈틈없이 들어찼다.

폭양에 그들은 그들의 시커먼 얼굴! 큰 박덩이만큼씩 한 전등이 드문드문하게 달린 천장에서 내리비치는 불빛이 휘황할수록, 흰 벽을 등지고 앉은 그네들의 얼굴은 「만호 장안의 별처럼 깔린 등불이 한눈에 내려다보이도록 사방의 유리창을 활짝 열어제쳤건만, 건장한 청년들의 코와 몸에서 풍기는 훈김이 우거진 공발 속을 들어; 정각이 되자 P학당의 취주악대는 코넷, 트럼본 같은 번쩍거리는 악기를 들고 연단 앞줄에 가 벌려 선다. 지휘자가 손을 내젓는 대로 힘차게 연주하는 것은 유명한 독

텍스트 요약 수행

```
summary = summarizer(input_text)
```

요약된 텍스트 출력

```
print(f"Summary: {summary[0]['summary_text']}")
```

Summary: 2개월이 되자, ∞일보사에서 주최하는 학생계몽운동에 참가하였던 대원들이 돌아와 오늘 저녁은 각처에서 모여든 대원들을 위로하는 다과회가 그 신문 오록백 명이나 수용할 수 있는 대강당에는 전 조선의 방방곡곡으로 흩어져서 한여름 동안 땀을 흘려 가며 활동한 남녀 대원들로 빈틈없이 들어찼다.

• 한글 문장 요약

- 다양한 글을 입력하고 요약을 시도해 봅시다.

소설 상록수 중에서...

```
input_text = '''레버쿠젠이 무패 행진을 이어갔다. 44경기 무패로 21세기 유럽 10대 리그 최강 무패를 달성했다.
```

레버쿠젠은 19일 오전 4시(한국시간) 영국 런던에 위치한 런던 스타디움에서 열린 2023-24시즌 유럽축구연맹(UEFA) 유로파리그(UEL) 8강 2차전에서 웨스트햄 유나이티드에게 1-0으로 승리했다. 전반 13분 블라디미르 초우팔의 패스를 받은 제로르 보웬이 우측면에서 왼발 멀리 크로스를 올렸다. 미카엘 안토니오가 머리로 미 후반에 들어와 레버쿠젠이 동점을 만들었다. 후반 44분 요시프 스타니시치의 패스를 받은 제레미 프림퐁이 페널티 박스 안에서 수비와 대치했다. 프림퐁은 페널티 빅 경기 후 사비 알론소 감독은 영국 매체 'TNT스포츠'와 인터뷰를 통해 "웨스트햄은 정말 좋은 선수들, 에너지, 믿음을 가지고 있었다. 유럽대항전에서 4강을 넘어 더 이어 "우리에게도 위기가 있었지만 무너지지 않고 잘 헤쳐나갔다. 두 번째, 세 번째 실점을 하지 않은 것이 주요했다. 계속 올라갈 수 있어 행복하다. UEL에서는 언

텍스트 요약 수행

```
summary = summarizer(input_text)
```

요약된 텍스트 출력

```
print(f"Summary: {summary[0]['summary_text']}")
```

Summary: 레버쿠젠은 19일 오전 4시(한국시간) 영국 런던에 위치한 런던 스타디움에서 열린 2023-24시즌 유럽축구연맹(UEFA) 유로파리그(UEL) 8강 2차전에서 웨

코딩을 시작하거나 AI로 코드를 생성하세요.

코딩을 시작하거나 AI로 코드를 생성하세요.

✓ (5) 문장 생성

영문 생성 모델 다운로드

```
generator = pipeline("text-generation", model="distilgpt2") #distill : 경량화
```

```
config.json: 100% 762/762 [00:00<00:00, 43.4kB/s]
model.safetensors: 100% 353M/353M [00:06<00:00, 83.3MB/s]
generation_config.json: 100% 124/124 [00:00<00:00, 2.06kB/s]
tokenizer_config.json: 100% 26.0/26.0 [00:00<00:00, 417B/s]
vocab.json: 100% 1.04M/1.04M [00:00<00:00, 15.7MB/s]
merges.txt: 100% 456k/456k [00:00<00:00, 6.57MB/s]
tokenizer.json: 100% 1.36M/1.36M [00:00<00:00, 21.7MB/s]
```

문장 생성 실행

```
generator("In this course, we will teach you how to",
          max_length=30,          # 생성할 최대 토큰 수
          num_return_sequences=3) # 생성할 문장
```

```
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncation=True` to explicitly truncate examples to
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
[{'generated_text': "In this course, we will teach you how to do the right thing, which are important, so we won't stop here and say, \""},
 {'generated_text': 'In this course, we will teach you how to solve and change the world with our own simple principles and
procedures.>\n\n\nLet'},
 {'generated_text': "In this course, we will teach you how to use an open API to create data that is open source:\n\n\nWe'll learn that using"}]
```

- 영어 문장을 조금 적은 후, 추가 문장을 생성해 봅시다.

코딩을 시작하거나 AI로 코드를 생성하세요.

코딩을 시작하거나 AI로 코드를 생성하세요.

코딩을 시작하거나 AI로 코드를 생성하세요.

3.Hugging Face 에서 직접 사용하기



<https://huggingface.co/>

- 허깅페이스에서
- 한국어 요약 모델 **ainize/kobart-news** 를 찾아서
- 사용해 봅시다.

```
from transformers import PreTrainedTokenizerFast, BartForConditionalGeneration
```

```
# Load Model and Tokenize
```

```
tokenizer = PreTrainedTokenizerFast.from_pretrained("ainize/kobart-news")
model = BartForConditionalGeneration.from_pretrained("ainize/kobart-news")
```

```
# Encode Input Text
```

```
input_text = ''
```

국내 전반적인 경기침체로 상가 건물주의 수익도 전국적인 감소세를 보이고 있는 것으로 나타났다.

수익형 부동산 연구개발기업 상가정보연구소는 한국감정원 통계를 분석한 결과 전국 중대형 상가
순영업소득(부동산에서 발생하는 임대수입, 기타수입에서 제반 경비를 공제한 순소득)이 1분기 ㎡당
3만4200원에서 3분기 2만5800원으로 감소했다고 17일 밝혔다.

수도권, 세종시, 지방광역시에서 순영업소득이 가장 많이 감소한 지역은 3분기 1만3100원을 기록한 울산으로, 1분기 1만9100원 대비 31.4% 감소했다.
이어 대구(-27.7%), 서울(-26.9%), 광주(-24.9%), 부산(-23.5%), 세종(-23.4%), 대전(-21%), 경기(-19.2%), 인천(-18.5%) 순으로 감소했다.
지방 도시의 경우도 비슷했다.

경남의 3분기 순영업소득은 1만2800원으로 1분기 1만7400원 대비 26.4% 감소했으며

제주(-25.1%), 경북(-24.1%), 충남(-20.9%), 강원(-20.9%), 전남(-20.1%), 전북(-17%), 충북(-15.3%) 등도 감소세를 보였다.

조현택 상가정보연구소 연구원은 "올해 내수 경기의 침체된 분위기가 유지되며

상가, 오피스 등을 비롯한 수익형 부동산 시장의 분위기도 경직된 모습을 보였고

오피스텔, 지식산업센터 등의 수익형 부동산 공급도 증가해 공실의 위험도 늘었다"며

"실제 올 3분기 전국 중대형 상가 공실률은 11.5%를 기록하며 1분기 11.3% 대비 0.2% 포인트 증가했다"고 말했다.

그는 "최근 소셜커머스(SNS를 통한 전자상거래), 음식 배달 중개 애플리케이션, 중고 물품 거래 애플리케이션 등의

사용 증가로 오프라인 매장에 영향을 미쳤다"며 "향후 지역, 콘텐츠에 따른 상권 양극화 현상은 심화될 것으로 보인다"고 덧붙였다.

```
...
```


```
input_ids = tokenizer.encode(input_text, return_tensors="pt")
```

```
# Generate Summary Text Ids
```

```
summary_text_ids = model.generate(
    input_ids=input_ids,
    bos_token_id=model.config.bos_token_id,
    eos_token_id=model.config.eos_token_id,
    length_penalty=2.0,
    max_length=142,
    min_length=56,
    num_beams=4,
)
```

```
# Decoding Text
```

```
print(tokenizer.decode(summary_text_ids[0], skip_special_tokens=True))
```

 You passed along `num_labels=3` with an incompatible id to label map: {'0': 'NEGATIVE', '1': 'POSITIVE'}. The number of labels will be overwritten to 2. 국내 국내 전반적인 경기침체로 상가 건물주의 수익도 전국적인 감소세를 보이고 있는 것으로 나타났고 전국적인 감소세를 보이고 있는 것으로 나타났으며 국내

코딩을 시작하거나 AI로 코드를 생성하세요.