

NEWS

# 최신티렌드 알아보기

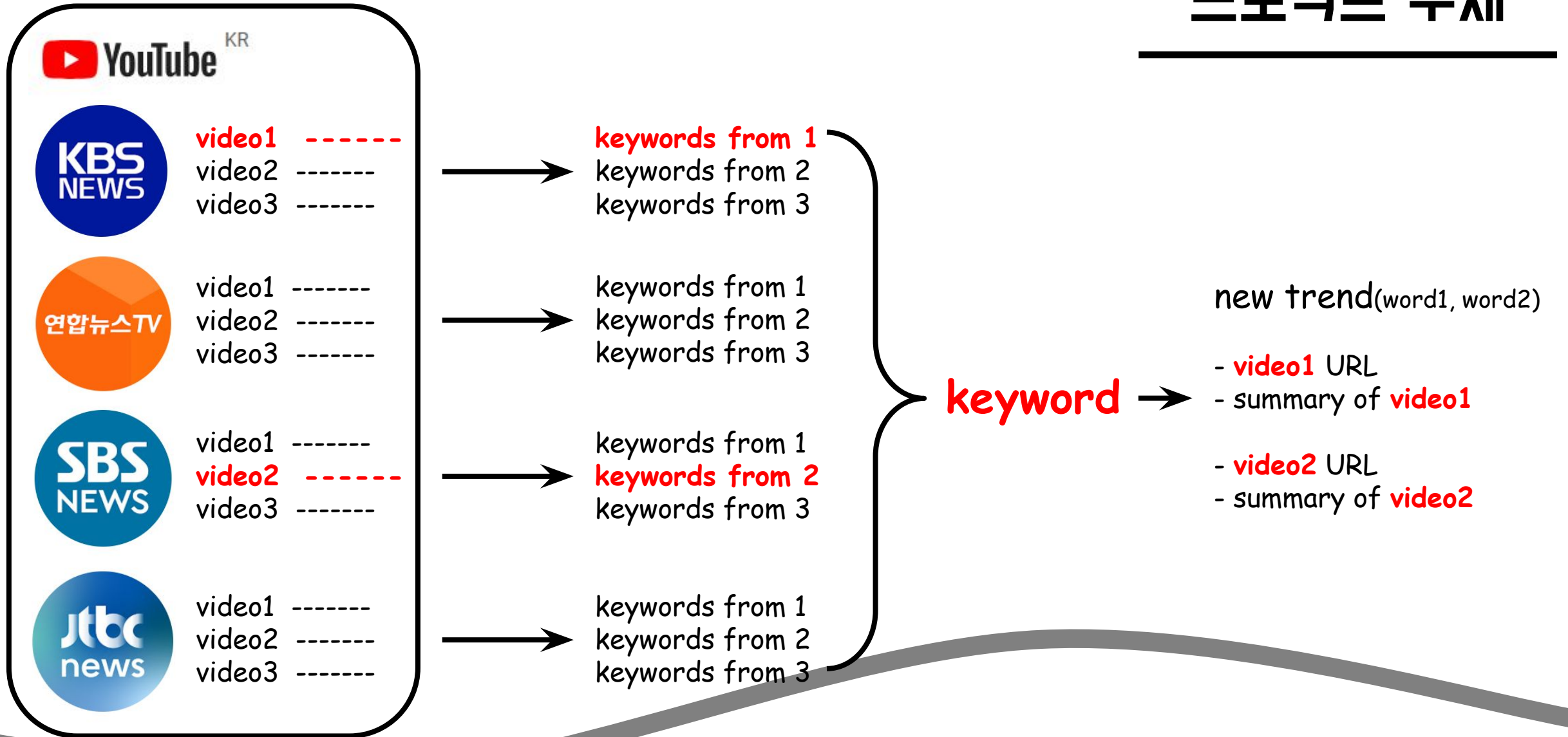
youtube news channel 

---

이승복

- > 프로젝트 주제
- > 프로젝트 선정 이유
- > 사용기술
- > 프로젝트 설명(code)
- > 마무리

# 프로젝트 주제



# 프로젝트 선정이유

---

openvino

자연어처리 (NLP)

강화학습

Python 기초

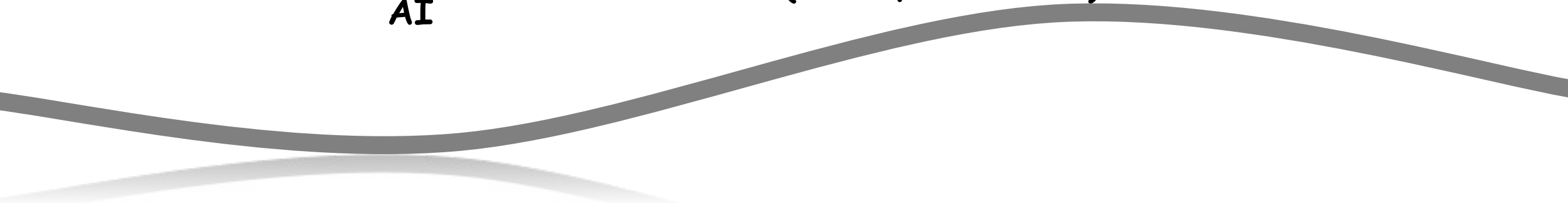
각종 API 사용

Fine Tuning

vision

AI

각종 모델들 (Llama, YOLO ...)



프로젝트 선정이유

---

openvino

강화학습

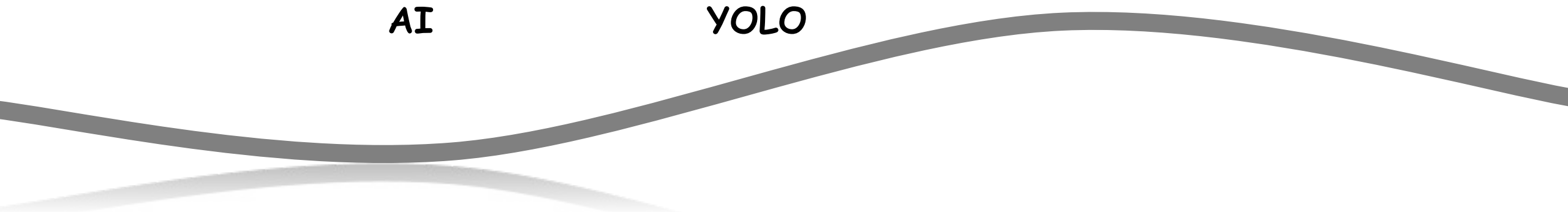
# 블랙박스 AI 도우미

Fine Tuning

vision

AI

YOLO



프로젝트 선정이유

---

openvino

자연어처리 (NLP)

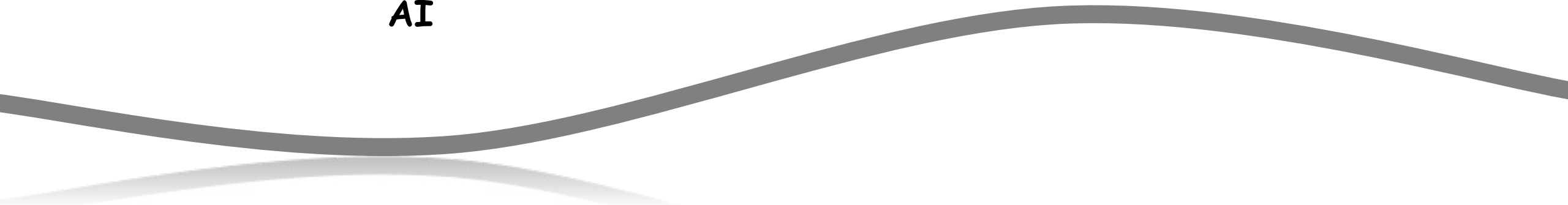
## 최신 트렌드 알아보기

각종 API 사용

Fine Tuning

Llama3.2, GPT4

AI



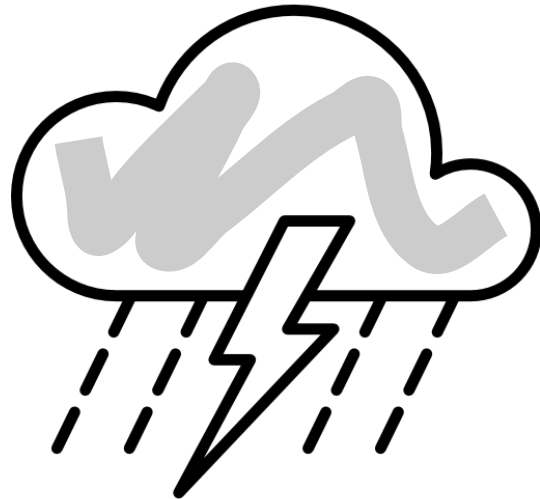
1. **youtube api**를 이용해 최신 영상의 mp3파일 불러오기(채널ID, 영상ID 불러오기)
2. **whisper** 이용하여 음성을 텍스트로 변환
3. "**keybert 모델**"을 사용하여 텍스트에서 키워드 추출
  - 3-1. 첫번째는 직접 모델을 불러와서 파이썬에서 바로 적용 -> 키워드 바로 제공
  - 3-2. 두번째는 onnx파일로 모델을 가져와서 "**openvino**"로 최적화 -> 임베딩 벡터 형태로 제공  
-> 텍스트를 토큰화하여 추출된 임베딩 벡터로 유사도 계산하여 키워드 추출
- 4-1. openai 에서 **gpt 4**를 이용해서 키워드 기반으로 문장 생성
- 4-2. Hugging face 에서 **llama3.2**를 이용해서 키워드 기반으로 문장 생성
  - > onnx파일로 모델을 가져와서 "**openvino**"로 최적화
  - > 한국어 데이터로 **Fine Tuning**해서 성능 높여주기

속도 ↑ 정확도 ↓

속도 ↓ 정확도 ↑







keyword

비, 흐림

#youtube api key를 통해 6개의 뉴스사에서 정보 가져오기 채널 ID가 필요하기 때문에 채널 ID부터 찾기

YOUTUBE\_API\_SERVICE\_NAME = "youtube"

YOUTUBE\_API\_VERSION = "v3"

channel\_list=['연합뉴스TV', 'YTN', 'MBCNEWS', 'KBS News', 'SBS 뉴스', 'JTBC News']

youtube = build(YOUTUBE\_API\_SERVICE\_NAME, YOUTUBE\_API\_VERSION, developerKey=API\_KEY)

channels={}

# 채널의 최신 동영상 가져오는 함수

def get\_latest\_videos(channel\_id):

url = f"https://www.googleapis.com/youtube/v3/search?key={API\_KEY}&channelId={channel\_id}&part=snippet,id&order=date&maxResults=10"

response = requests.get(url)

data = response.json()

video\_ids = []

for item in data['items']:

if item['id']['kind'] == 'youtube#video':

video\_ids.append(item['id']['videoid'])

return video\_ids

# 동영상 정보 가져오기

def get\_video\_details(video\_ids):

video\_details = []

video\_id\_str = ','.join(video\_ids) # API 요청을 위해 비디오 ID를 쉼표로 구분

url = f"https://www.googleapis.com/youtube/v3/videos?key={API\_KEY}&id={video\_id\_str}&part=contentDetails,snippet,liveStreamingDetails"

response = requests.get(url)

data = response.json()

for item in data['items']:

# 실시간 동영상인지 확인

if item['snippet']['liveBroadcastContent'] == "none":

duration = isodate.parse\_duration(item['contentDetails']['duration'])

if duration <= timedelta(minutes=2): # 2분 이하만 선택

video\_data = {

'title': item['snippet']['title'],

'videoid': item['id'],

'publishedAt': item['snippet']['publishedAt'],

'duration': duration

}

video\_details.append(video\_data)

return video\_details

ID

연합뉴스TV 의 채널 ID : UCHCOPwqNfZ0uik0vFyhGwg

YTN 의 채널 ID : UChlg13UHC0nwUGzWzbJ3H5w

MBCNEWS 의 채널 ID : UCF4Wxdo3inmxP-Y59wXDsfw

KBS News 의 채널 ID : UCcQTRi69dsVYHN3exePtZ1A

SBS 뉴스 의 채널 ID : UCKinYTS9IHq0EwR1Sze2JTW

JTBC News 의 채널 ID : UCsU-l-vHLiaMfV-ceaYz5rQ

프로젝트 설명

Info

--- 연합뉴스TV ---

Title: [날씨] 전국 호리고 남부 호우 집중...내일까지 이어져 / 연합뉴스TV (YonhapnewsTV)

Video ID: GyqMr\_dbois

Published At: 2024-10-13T23:16:28Z

Duration: 0:01:40

Title: [알면 좋은 뉴스] 해외 독자들도 '한강알이'...한강 소설 불티나게 판매·품질 外 / 연합뉴스TV (YonhapnewsTV)

Video ID: HnZSj7vGNyO

Published At: 2024-10-13T23:14:22Z

Duration: 0:01:59

--- YTN ---

Title: 발사대에 다시 착륙...스페이스X 시험비행 성공 / YTN

Video ID: CXhaf6vtSdg

Published At: 2024-10-13T19:39:18Z

Duration: 0:01:46

Title: 이스라엘 중부서 40명 부상...헤즈볼라 "드론 공격" / YTN

Video ID: oWKv-HM1JJ8

Published At: 2024-10-13T19:37:10Z

Duration: 0:00:31

--- MBCNEWS ---

Title: 2024년 10월 14일 (월) 오늘의 주요 뉴스 #Shorts

Video ID: bLbVduiNWCw

Published At: 2024-10-13T21:24:35Z

Duration: 0:01:00

Title: [날씨] 충청·남부 비...주 후반 비 온 뒤 쌀쌀 (2024.10.13/뉴스데스크/MBC)

Video ID: ulzF5sjRW3E

Published At: 2024-10-13T11:39:19Z

Duration: 0:01:20

--- KBS News ---

Title: [날씨] 충청·남부 중심 가을비...기온 평년보다 높아 / KBS 2024.10.14.

Video ID: hFJa0\_5YKV8

Published At: 2024-10-13T23:05:10Z

youtube

# 프로젝트 설명

```
import yt_dlp
import whisper

model = whisper.load_model("large") # 1시간 40분 - 2시간 정도..
# model = whisper.load_model("base") # 이거는 한 10분 정도인데 음성이 좀 더 정확하지는 않고
# model = whisper.load_model("small") #
# model = whisper.load_model("medium")
import subprocess

#동영상 주소로 다운로드
urls=[]
for id in video_id_list:
    urls.append('https://www.youtube.com/watch?v='+id)

# 각 동영상을 MP3로 다운로드
for url in urls:
    # 출력 파일명
    command = [
        "yt-dlp",
        "-x",
        "--audio-format", "mp3",
        "-o", "%(id)s", # 파일명을 동영상 id로 설정
        url
    ]
    result=[]
    for i in range(len(video_id_list)):
        result.append(model.transcribe('/content/'+video_id_list[i]+'.mp3'))
```

## title , main

전국 흐리고 남부 호우 집중...내일까지 이어져 한주의 시작인 오늘 전국 하늘이 흐리겠고 곳곳에 비가 내리겠습니다. 특히 해외 독자들도 '한강알이'...한강 소설 불티나게 판매·품절 外 노벨 문학상으로 역사를 썼고 앞으로 또 새로운 역사를 써 발사대에 다시 착륙...스페이스X 시험비행 성공 일론 머스크의 우주기업 스페이스X가 세계 최초로 로켓을 발사대에 다시 이스라엘 중부서 40명 부상...헤즈볼라 "드론 공격" 이스라엘 테라비브 북쪽 50km 정도에 위치한 소도시 빈야미나에서 무 지금까지 생생 지구촌이었습니다. 공동성명을 냈지만 네타냐후 이스라엘 총리는 전투 지역에서 유엔군 철수를 요구하고 충청·남부 비...주 후반 비 온 뒤 쌀쌀 ( 지금까지 생생 지구촌이었습니다. 흐리면 흐린 대로 계절의 정취를 느낄 수 있 충청·남부 중심 가을비...기온 평년보다 높아 오늘은 주로 충청과 남부 지방에 가을비가 내리겠습니다. 레이더 화면을 보 뉴욕총영사, 광복절 기념사 논란에 "소신 발언...사과할 뜻 없다" 김 의원 미국 뉴욕 총영사가 국정감사에서 국회 외교통일 충청·남부·제주 중심 비...내일까지 전국 5~80mm 충청과 남부, 제주를 중심으로 비가 내리겠습니다. 내일까지 예상되는 '일본도 살인' 도검 판매업체 대표 2명 입건 SBS 김지경입니다. 지난 7월 발생한 일본도 살인 사건의 가해자에게 도검을 '완전체 결실' 약속한 옥주현! 보고 있을 핑클 멤버들에게 한마디 핑클 완전체 재결합 소망도 있다고 하셨는데 너무나 있 전국 대체로 '흐림'...오후부터 곳곳 비 내일 대체로 흐린 가운데 오후부터는 일부 지역에 비가 오겠습니다. 올해까지 부산 "유산으로 120년간 상금 지급 중"...노벨 재단의 재테크 비결은? 한강 작가가 받을 노벨상 상금이 13억 원이 넘는 걸로 알

# whisper

# 프로젝트 설명

```
from keybert import KeyBERT

# KeyBERT 모델
kw_model = KeyBERT(model=None)

# 한국어 불용어 리스트
stop_words = [
    '은', '는', '이', '가', '을', '를', '에', '의', '도', '와', '으로', '그리고', '그', '이것', '저것',
    '너', '나', '우리', '당신', '모두', '또한', '에서', '보다', '까지', '부터', '한테', '만', '왜', '어디',
    '누구', '무엇'
]

# 키워드 추출
for i in range(len(news_title_main_list)):
    text = news_title_main_list[i]
    keywords = kw_model.extract_keywords(
        text,
        keyphrase_ngram_range=(1, 3), # (1~3 단어 조합)
        stop_words=stop_words, # 불용어 리스트
        use_mmr=True,
        diversity=0.9,
        top_n=10 # 상위 10개의 키워드 추출
    )

    print(keywords)
```

keyword with **keybert**

[('남부지방에 비가', 0.6832), ('이번 후반부터는 북쪽에서', 0.5049), ('하늘 표정을', 0.4619), ('김수진', 0.4236), ('내일', 0.3588), ('기온은  
[('해외 독자들도 한강알이', 0.6948), ('플랫폼인 아마존에서는', 0.4058), ('판매 품질 노벨', 0.3333), ('한창입니다 미국의 최대', 0.3087), ('한·  
[('화성 탐사 로켓', 0.7106), ('스페이스x 시험비행 성공', 0.5738), ('69km에서 1단 추진체', 0.4373), ('지금까지는 발사대가 아니라', 0.2873), ('  
[('이스라엘 중부서 40명', 0.7182), ('드론 공격', 0.4312), ('부상', 0.3703), ('향해 대규모 무인기', 0.2463), ('50km', 0.2064), ('소도시', 0.1:  
[('가운데 민주당은 정권의', 0.6158), ('대통령실과 여당에선 불쾌하다는', 0.5635), ('전투 지역에서 유엔군', 0.4683), ('국민의힘 한동훈', 0.382  
[('넘는 비가', 0.6419), ('남부 후반 쌀쌀', 0.4475), ('내일', 0.41), ('24도', 0.291), ('지형적인 영향을', 0.2172), ('강하고 많은', 0.1908), ('  
[('남부에도 가끔씩 비가', 0.7816), ('23도', 0.4014), ('충남과 호남 지방', 0.3198), ('내일', 0.3114), ('레이더 화면을 보면', 0.2996), ('먼바다  
[('뉴욕 유엔 한국대표부에서', 0.6776), ('사과할', 0.5272), ('국회 외교통일위원회', 0.2895), ('정치적 중립을', 0.2733), ('없다', 0.2135), ('지  
[('22도로 비가 내리는', 0.6025), ('남해안 지역은 시간당', 0.494), ('지역은 10에서 60mm', 0.4667), ('서울과 강릉', 0.4545), ('17 6도', 0.3941)  
[('일본도 살인 사건의', 0.7005), ('이들은 일본도를 제조', 0.4839), ('경찰은 국내', 0.4235), ('총포화약법을', 0.3933), ('김지경입니다 지난 7월  
[('아직 자리에서', 0.7105), ('저희도', 0.5701), ('완전체 재결합', 0.4307), ('질문', 0.4251), ('within 중간', 0.3624), ('diligently and', 0.3:  
[('5에서 60mm의 비가', 0.7544), ('등에는 5에서 60mm의', 0.4381), ('대체로 흐림 오후부터', 0.3888), ('80mm', 0.3358), ('내일', 0.3094), ('부스  
[('노벨상 상금은 100만', 0.8081), ('현재는 기금의 절반가량이며', 0.4168), ('글로벌 금융위기 800만', 0.3233), ('스웨덴', 0.3136), ('120년', 0.2

keyword ver1



# bert model to ONNX

```
from transformers import AutoTokenizer, AutoModel
```

```
# KLUE/BERT 모델 및 토크나이저 로드
```

```
tokenizer = AutoTokenizer.from_pretrained("klue/bert-base")
```

```
model = AutoModel.from_pretrained("klue/bert-base")
```

```
# 예시 입력 문장
```

```
text = "한국 경제는 매우 빠르게 성장하고 있습니다."
```

```
# 토크나이저로 입력 데이터 준비
```

```
inputs = tokenizer(text, return_tensors='pt')
```

```
# 모델 추론 (PyTorch에서 실행)
```

```
outputs = model(**inputs)
```

Model  
Optimizer

openvino

## 프로젝트 설명

```
import torch
```

```
# 모델을 ONNX 형식으로 변환
```

```
torch.onnx.export(
```

```
    model,
```

```
    (inputs['input_ids'], inputs['attention_mask']), # 필요한 입력 텐서들
```

```
    "klue_bert_model.onnx", # 저장할 ONNX 파일 이름
```

```
    input_names=['input_ids', 'attention_mask'], # 입력 이름
```

```
    output_names=['output'], # 출력 이름
```

```
    dynamic_axes={'input_ids': {0: 'batch_size'}, 'attention_mask': {0: 'batch_size'}, 'output': {0: 'batch_size'}},
```

```
    opset_version=14 # ONNX 버전
```

```
)
```

```
# OpenVINO Model Optimizer 사용
```

```
!mo --input_model klue_bert_model.onnx --output_dir ./klue_openvino --compress_to_fp16
```

openvino

```
# 중복 키워드 제거 및 상위 n개의 고유한 키워드를 추출하는 함수
def select_top_keywords(all_keywords, top_n=5, filter_words=None):
    # 필터링: 조사나 자주등장하는 대명사들 제거
    if filter_words is None:
        filter_words = [ '.', '네', '등', '있', '싶', '것', '않',
                        '"', "'", '?', '!', '%', ',', '...', ':', ';', '“’, ’”, ‘’, ’’, ', '/', '|',
                        '누가', '언제', '어디', '무엇', '어떻게', '왜',
                        '이렇게', '저렇게', '이게', '그게', '저게', '이', '그', '저' ]

    # 필터링: '#'이 포함된 키워드 및 사용자 정의 필터 단어 제거
    filtered_keywords = [
        kw for kw in all_keywords
        if not re.match(r'#', kw) and kw not in filter_words
    ]

    # 키워드 빈도 계산 (중복 제거)
    keyword_counter = Counter(filtered_keywords)

    # 빈도순으로 상위 n개의 키워드를 추출
    most_common_keywords = keyword_counter.most_common(top_n)

    # 상위 n개의 키워드 반환
    return [keyword for keyword, _ in most_common_keywords]
```

[ '비', '남부', '전국', '오늘', '하늘', '충청', '내리', '호우', '기상', '온통' ]  
 [ '한강', '노래', '소설', '작품', '해외', '들', '앞', '노벨', '역사', '새로운' ]  
 [ '스페이스', '로켓', '발사', '발사대', '비용', '재', '예정', '착륙', '추진', '성공' ]  
 [ '이스라엘', '해', '정도', '북쪽', '빈', '다쳤', '부상자', '밝혔', '아직', '레바논' ]  
 [ '국민', '언급', '공개', '입장', '반응', '청주', '경부고속도로', '근처', '지금', '공동' ]  
 [ '비', '충청', '서울', '남부', '지방', '아침', '차츰', '영동', '제주', '생생' ]  
 [ '비', '높', '기온', '부산', '서울', '남부', '오후', '내리', '광주', '먼' ]  
 [ '발언', '뉴욕', '국회', '기념', '광복절', '충영', '정치', '사과', '김', '생각' ]  
 [ '비', '전국', '지역', '서울', '낮', '남부', '제주', '충청', '내일', '부산' ]  
 [ '판매', '경찰', '입건', 'SBS', '살인', '업체', '허가', '불법', '또', '외' ]  
 [ '핑', '약속', '멤버', '꼭', '모', '재결', '말', '늘', '아직', '정확' ]  
 [ '비', '대체로', '오후', '충청', '남부', '전국', '곳곳', '부산', '울산', '많' ]  
 [ '상금', '노벨', '원', '넘', '돈', '금융', '투자', '노벨상', '기금', '경제' ]

keyword ver2

# 프로젝트 설명

```
for keywords in key_words_list:
    # 키워드를 기반으로 문장을 생성하기 위한 프롬프트, 메시지 생성
    prompt = (
        f"The following keywords are ordered by importance. The first word is the most important, "
        f"and the last word is the least important. Create a sentence using these keywords in this order: "
        f"{', '.join(keywords)})."
    )

    messages = [
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": f"Generate a sentence using the following keywords: {', '.join(keywords)}."}
    ]

    response = openai.ChatCompletion.create(
        model="gpt-4",
        messages=messages,
        max_tokens=100,
        temperature=0.7
    )

    # 생성된 텍스트 출력
    generated_text = response['choices'][0]['message']['content'].strip()
    keywords_to_sentence.append(generated_text)
    print("Generated Sentence:", generated_text)
```

## Keywords to sentence Feat GPT4

Generated Sentence: "오늘 기상 예보에 따르면 전국적으로 비가 내리며, 특히 남부와 충청 지역에는 호우가 예상되어 하늘은 온통 구름으로 덮여 있습니다."

Generated Sentence: "한강의 새로운 소설 작품은 역사를 노래하며, 그의 노벨 앞으로 해외에서도 많이 듣게 될 것입니다."

Generated Sentence: "스페이스 회사는 비용 절감을 위해 재사용 가능한 로켓을 개발하였고, 이 로켓은 발사대에서 성공적으로 발사 예정이며, 추진 시스템을 통해 착륙도 재차"

Generated Sentence: "이스라엘 북쪽에 위치한 레바논에서 해매던 정도는, 아직 빈 부상자를 찾지 못했으며, 다친 사람들이 있다는 사실을 밝혔습니다."

Generated Sentence: "지금 경부고속도로 근처 청주에서 공개된 국민들의 입장을 언급하는데, 이에 대한 공동 반응이 활발하게 이루어지고 있습니다."

Generated Sentence: "아침에 서울과 제주, 그리고 영동 지방에는 생생한 비가 내리기 시작하며, 차츰 남부와 충청 지방으로도 비가 확산될 예정입니다."

Generated Sentence: "오후에 서울과 광주에 비가 내리며, 부산과 남부 지역은 기온이 높아질 것으로 예상됩니다."

Generated Sentence: "김 총영이 뉴욕에서 발언하며, 광복절 기념 행사에서의 국회의 정치적 사과에 대한 생각을 공유했습니다."

Generated Sentence: "내일 낮에는 전국적으로 비가 내릴 예정이며, 특히 서울, 부산, 충청 지역과 남부 지역, 그리고 제주도에는 강한 비가 예상됩니다."

Generated Sentence: "SBS에 따르면, 판매 업체가 허가 없이 불법으로 살인 도구를 판매하여 경찰에 입건되었고, 이에 대해 외부에서는 또 한 번의 경각심을 불러일으켰습니다."

Generated Sentence: "핑 멤버들이 아직 모두 모이지 않아서, 정확한 재결을 내리지는 못했지만, 우리는 약속했어, 꼭 모두가 모일 때 그 말을 늘 하기로."

Generated Sentence: "전국적으로 비가 내리고 있지만, 오후에는 대체로 비가 그치며 충청과 남부 지역, 그리고 부산과 울산 등 곳곳에서는 많은 비가 예상됩니다."

Generated Sentence: "노벨상의 상금은 수억 원이 넘는 돈으로, 이를 금융 투자에 활용하면 큰 경제적 이익을 가져올 수 있는 기금이 될 수 있습니다."

# GPT4

# 프로젝트 설명

```
from transformers import AutoTokenizer, AutoModelForCausalLM
import torch

# Hugging Face API 키
api_token = '...'

# 모델 및 토크나이저 로드
model_id = "meta-llama/llama-3.2-3B-Instruct"
tokenizer = AutoTokenizer.from_pretrained(model_id, use_auth_token=api_token)
model = AutoModelForCausalLM.from_pretrained(model_id, use_auth_token=api_token)
```

# ONNX로 변환

```
dummy_input = tokenizer("안녕하세요", return_tensors="pt")
torch.onnx.export(
```

```
    model,
    (dummy_input["input_ids"],),
    "llama_3.2.onnx",
    input_names=["input_ids"],
    output_names=["output"],
    dynamic_axes={"input_ids": {0: "batch_
```

```
    # 프롬프트 리스트를 기반으로 문장 생성
    prompts = ['비', '날씨', '높다', '주말', '서울', '기온', '많이', '한낮', '전국']
    generated_text = generate_text_from_prompts(prompts)
    print(generated_text)
```

```
!mo --input_model llama_3.2.onnx --output-
```

```
## openvino로 최적화해서 사용해봤는데 너무 느림... 한국어 데이터를 파인튜닝하려고 했으나 실패함...
## 생성된 문장 : 기온 상승 서울에서부터 강원도까지의 기온 상승은 1일 2도, 2일 3도, 3일 4도, 4일 5도, 5일 6도, 6일 7도, 7일 8도, 8일 9도, 9일 10도, 10일 11도, 11일 12도, 12일 13
```

# 프롬프트 리스트를 기반으로 문장 생성

```
prompts = ['rain', 'weather', 'high', 'weekend', 'seoul', 'temperature', 'many']
generated_text = generate_text_from_prompts(prompts)
print(generated_text)
```

```
## 생성된 영어 문장 - Seoul, South Korea - A high-pressure system is expected to bring clear skies and warm temperatures to Seoul over the weekend, with temperatures expected to reach as high as 28° C
##
##           However, a low-pressure system is expected to bring rain to the city on Saturday, with a high of 22° C (72° F) and a low of 18° C (64° F).
##           The rain is expected to clear by Sunday morning, leaving behind a beautiful day with plenty.
## 해석 - 고기압 시스템이 주말 동안 서울에 맑은 하늘과 따뜻한 기온을 가져올 것으로 예상되며, 일요일에는 기온이 최고 28° C (82° F)까지 오를 것으로 보입니다.
##           하지만 저기압 시스템이 토요일에 서울에 비를 가져올 것으로 예상되며, 최고 기온은 22° C (72° F), 최저 기온은 18° C (64° F)로 예상됩니다.
##           비는 일요일 아침까지 그치고, 맑고 화창한 날씨가 이어질 것으로 보입니다.
```

## Llama3.2, openvino



predict  
keyword  
비, 흐림



keyword  
비, 남부

## 프로젝트 설명

가장 많이 등장한 단어는 : '비' '남부' 입니다.

가장 트렌디한 뉴스의 정보는? >>

1번째의 뉴스 Url: [https://www.youtube.com/watch?v=GygMr\\_dbois](https://www.youtube.com/watch?v=GygMr_dbois), 제목: 전국 흐리고 남부 호우 집중...내일까지 이어져

키워드는 ['비', '남부', '전국', '오늘', '하늘', '충청', '내리', '호우', '기상', '온통'] 입니다.

키워드 기반 뉴스 요약 "오늘 기상 예보에 따르면 전국적으로 비가 내리며, 특히 남부와 충청 지역에는 호우가 예상되어 하늘은 온통 구름으로 덮여 있습니다."

6번째의 뉴스 Url: <https://www.youtube.com/watch?v=u1zF5sjRW3E>, 제목: 충청·남부 비...주 후반 비 온 뒤 쌀쌀 (

키워드는 ['비', '충청', '서울', '남부', '지방', '아침', '차츰', '영동', '제주', '생생'] 입니다.

키워드 기반 뉴스 요약 "아침에 서울과 제주, 그리고 영동 지방에는 생생한 비가 내리기 시작하며, 차츰 남부와 충청 지방으로도 비가 확산될 예정입니다."

7번째의 뉴스 Url: [https://www.youtube.com/watch?v=hFJa0\\_5YKV8](https://www.youtube.com/watch?v=hFJa0_5YKV8), 제목: 충청·남부 중심 가을비...기온 평년보다 높아

키워드는 ['비', '높', '기온', '부산', '서울', '남부', '오후', '내리', '광주', '먼'] 입니다.

키워드 기반 뉴스 요약 "오후에 서울과 광주는 비가 내리며, 부산과 남부 지역은 기온이 높아질 것으로 예상됩니다."

9번째의 뉴스 Url: <https://www.youtube.com/watch?v=0lWcbFCeKUK>, 제목: 충청·남부·제주 중심 비...내일까지 전국 5~80mm

키워드는 ['비', '전국', '지역', '서울', '낮', '남부', '제주', '충청', '내일', '부산'] 입니다.

키워드 기반 뉴스 요약 "내일 낮에는 전국적으로 비가 내릴 예정이며, 특히 서울, 부산, 충청 지역과 남부 지역, 그리고 제주도에 강한 비가 예상됩니다."

12번째의 뉴스 Url: <https://www.youtube.com/watch?v=e30DYPJbh2w>, 제목: 전국 대체로 '흐림'...오후부터 곳곳 비

키워드는 ['비', '대체로', '오후', '충청', '남부', '전국', '곳곳', '부산', '울산', '많'] 입니다.

키워드 기반 뉴스 요약 "전국적으로 비가 내리고 있지만, 오후에는 대체로 비가 그치며 충청과 남부 지역, 그리고 부산과 울산 등 곳곳에서는 많은 비가 예상됩니다."

# Final

## 아쉬운 점

1. whisper를 시연할 수 있을만큼 속도 문제를 개선
  2. streamlit을 이용해 최소한의 UI라도 제공
  3. Fine Tuning이 제대로 이루어져서 Llama 3.2로도 문장을 생성
- 