강원대학교
AI 소프트웨어학과

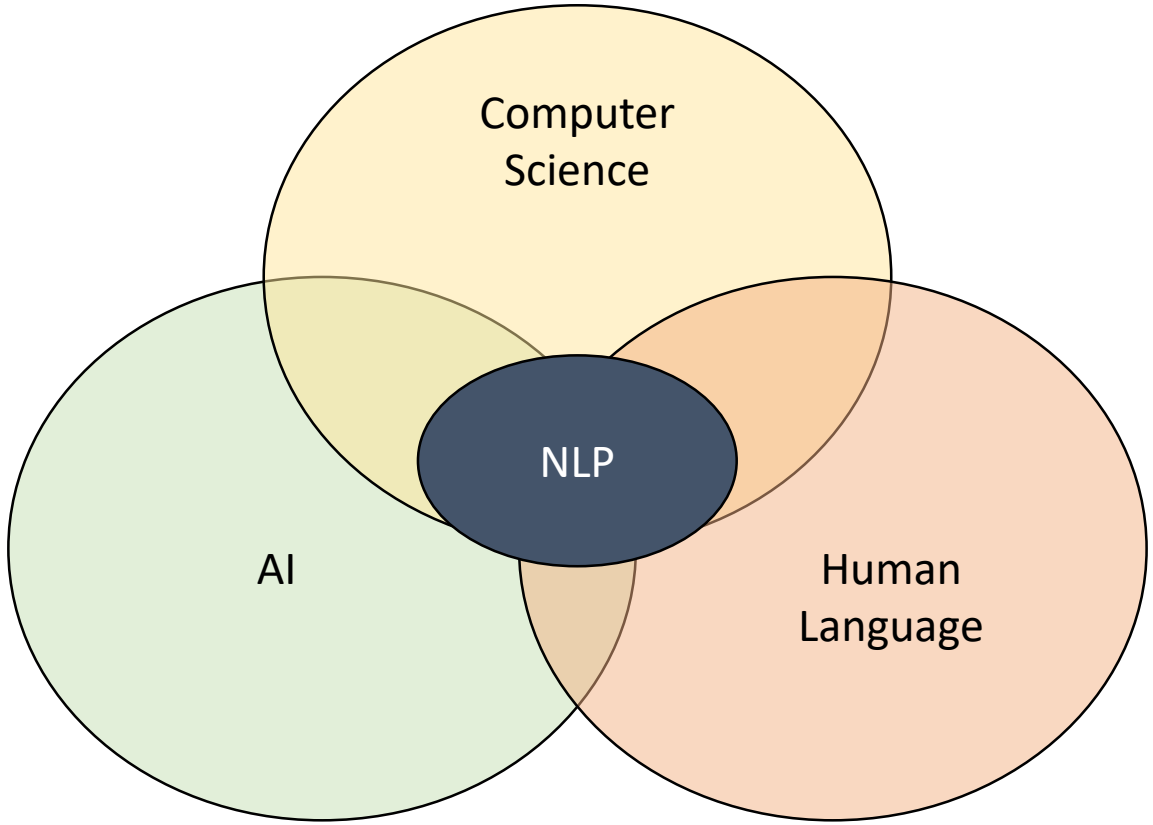
인공지능
- 텍스트 마이닝 -

KNU
강원대학교

인공지능 텍스트 분석

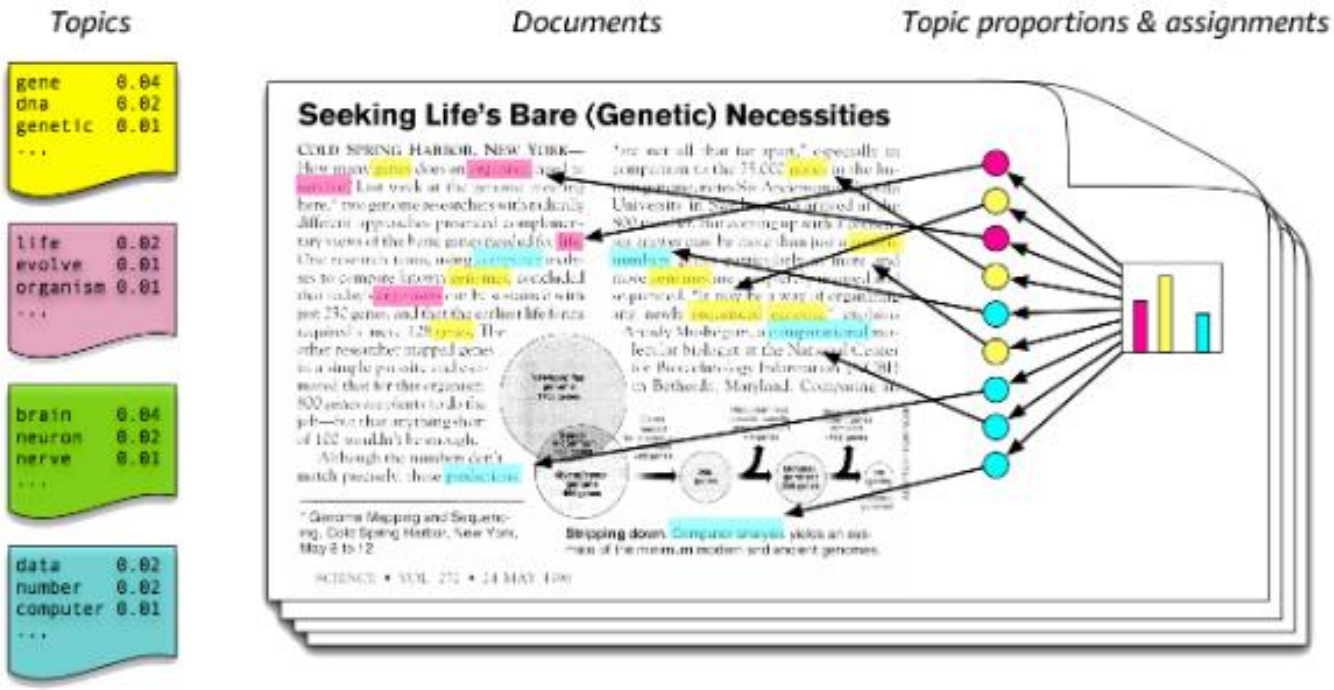
- NLP(Natural Language Processing)는 기계가 사람의 언어에 대해 처리하는 계산적 기술의 집합

→ 의미분석, 감성분석, 음성인식, 번역 등이 존재



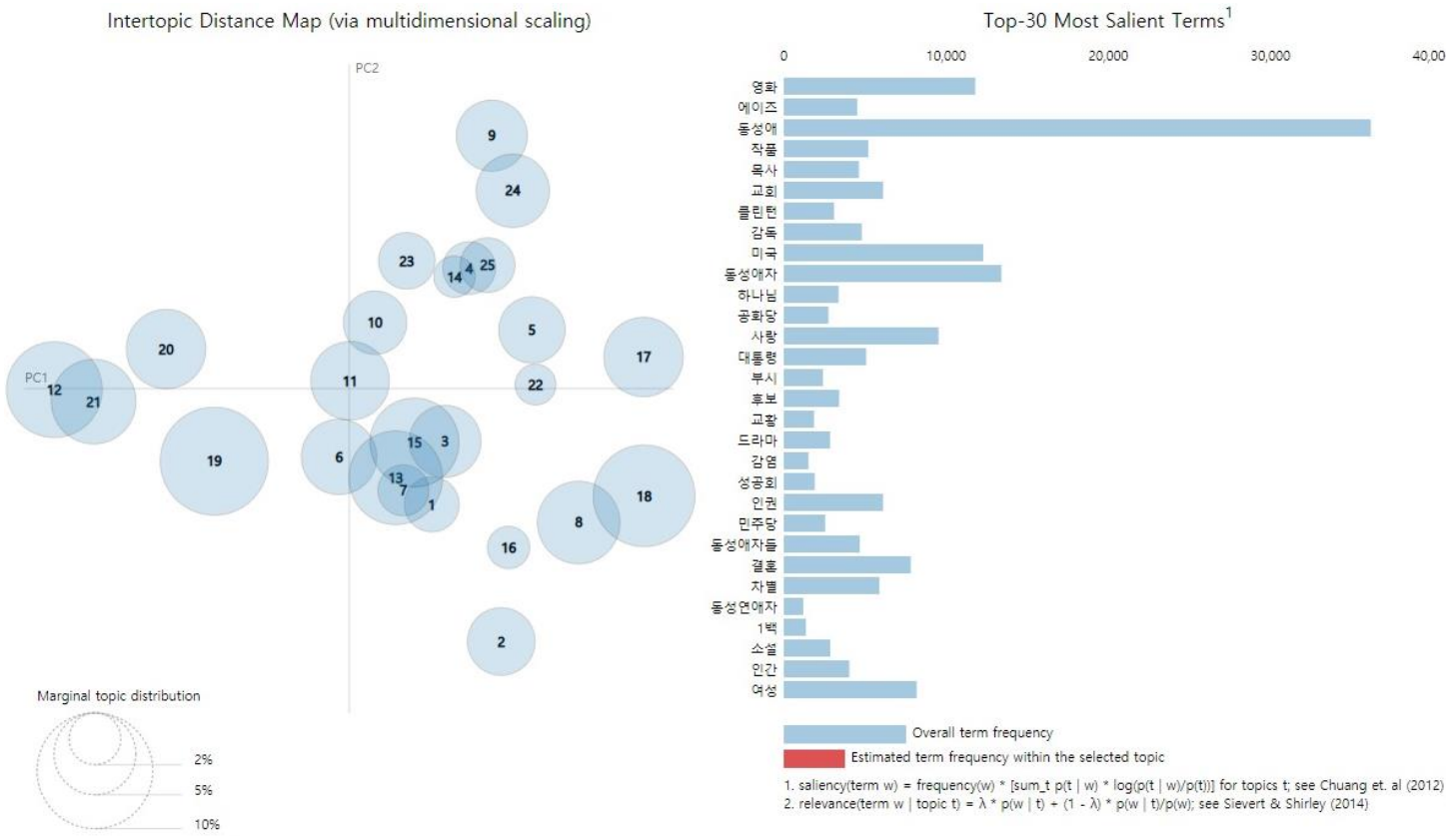
인공지능 텍스트 분석

- **토픽모델링(Topic Modeling) : 단어, 말뭉치(corpus)로 부터 숨겨진 의미를 찾고 키워드별로 주제를 묶어 주는 모델로 문서에 대한 확률 분포를 가정해 분류해주는 방법**



인공지능 텍스트 분석

- **토픽모델링(Topic Modeling) : 단어, 말뭉치(corpus)로 부터 숨겨진 의미를 찾고 키워드별로 주제를 묶어 주는 모델로 문서에 대한 확률 분포를 가정해 분류해주는 방법**



01 텍스트 마이닝

인공지능 텍스트 분석

- 워드 클라우드(Word Cloud)는 텍스트를 분석해 사람들의 관심사, 키워드, 개념 등을 파악할 수 있도록 빈도수를 단순히 카운트하여 시각화 시킨 방법

인공지능 텍스트 분석

- **활용 사례 : 사람들의 댓글 및 의견들을 통해 새로운 가치 및 의미를 찾는 것에 활용**

  → 음식의 새로운 조합, 사람들의 흥미요소, 전혀 연관이 없는 새로운 가치

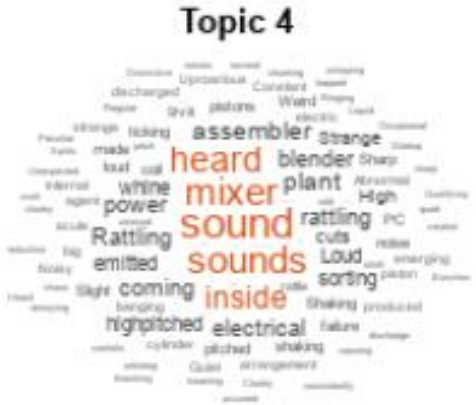
인공지능 텍스트 분석

- **활용 사례 : 사람들의 댓글 및 의견들을 통해 새로운 가치 및 의미를 찾는 것에 활용**

  → 음식의 새로운 조합, 사람들의 흥미요소, 전혀 연관이 없는 새로운 가치

인공지능 텍스트 분석

- **활용 사례 : 음악의 장르 파악, 논문의 키워드를 파악, 글쓴이의 성향을 파악**

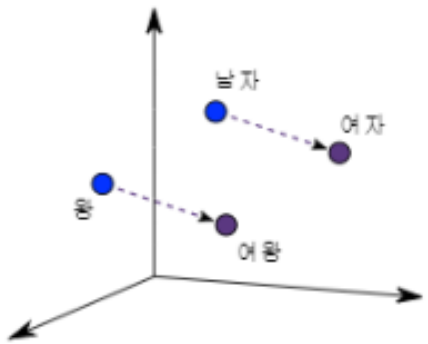


인공지능 텍스트 분석

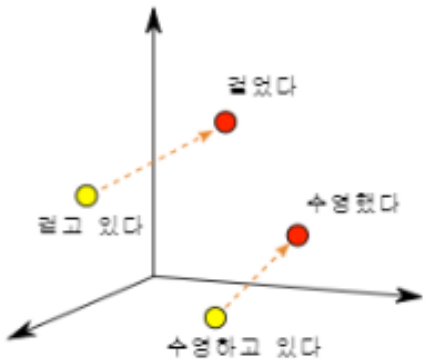
- 감성 분석(Sentiment Analysis)이란 텍스트에 들어있는 의견이나 감성, 평가, 태도 등의 주관적인 정보를 컴퓨터를 통해 분석하는 과정

인공지능 텍스트 분석

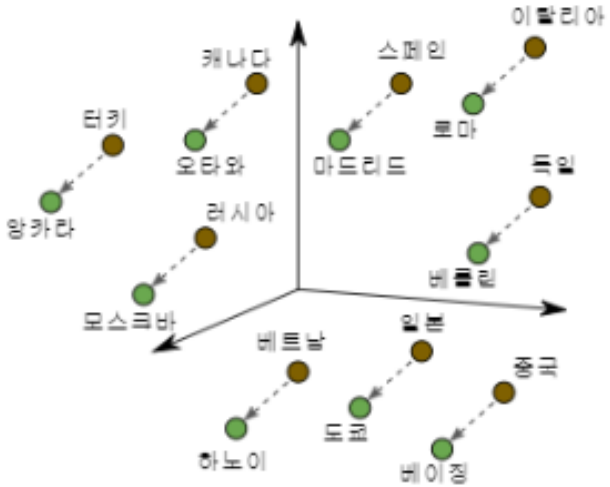
- NLP를 하기위해 텍스트를 컴퓨터가 이해할 수 있도록 숫자로 바꾸는 작업이 필요함
- 사람의 경우는 문맥을 통해 문장 및 의미를 구별하는 것이 가능함 → 임베딩(Embedding)
- 즉 자연어를 수치화 한 것으로 벡터로 표현하는 것을 말하고 임베딩은 그 과정까지 모두를 포함하는 의미



남자-여자



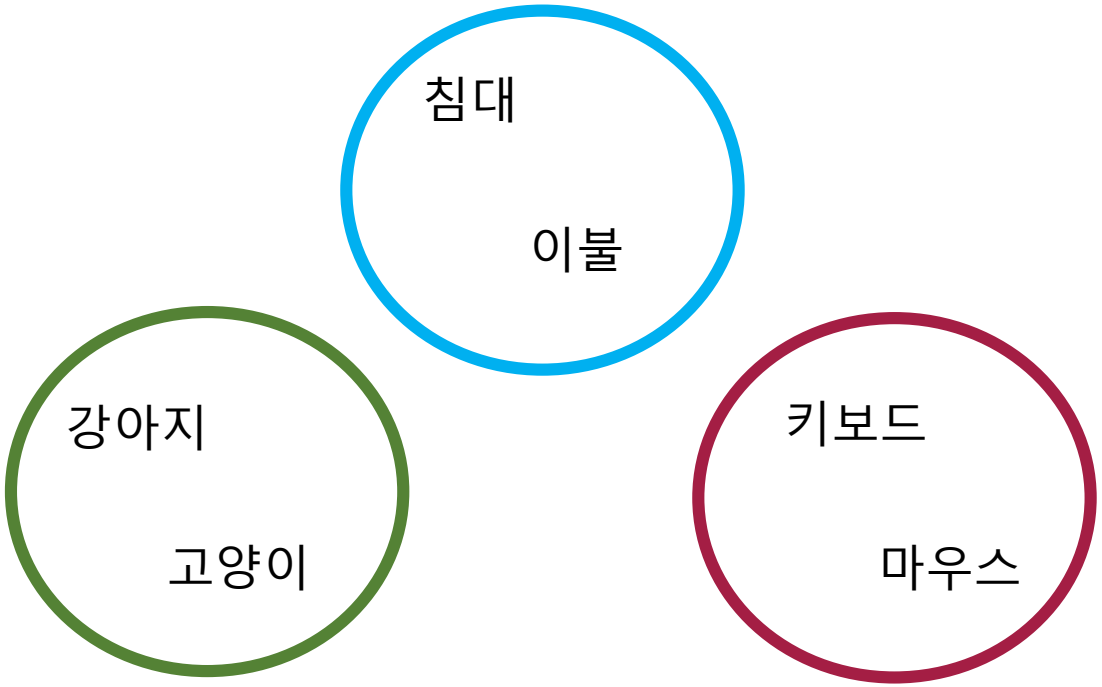
동사 시제

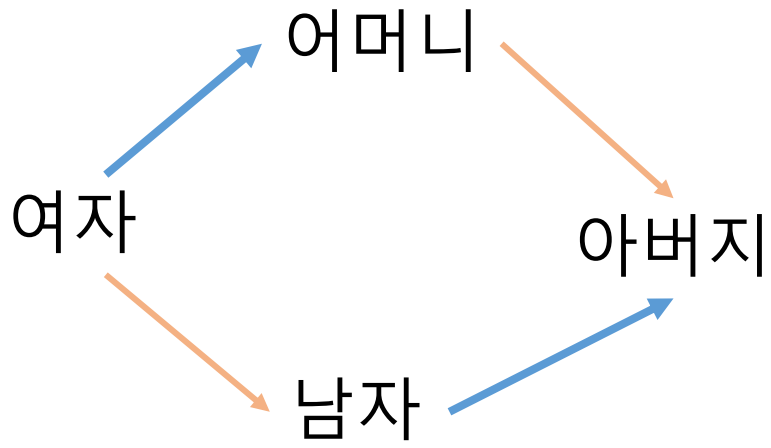


국가-수도

인공지능 텍스트 분석

- **임베딩(Embedding) : 전체 단어들 간의 관계에 맞춰 해당 언어의 특성을 갖는 벡터로 바꿔주므로 단어 간의 의미를 파악해 문법적 관계를 알 수 있음**

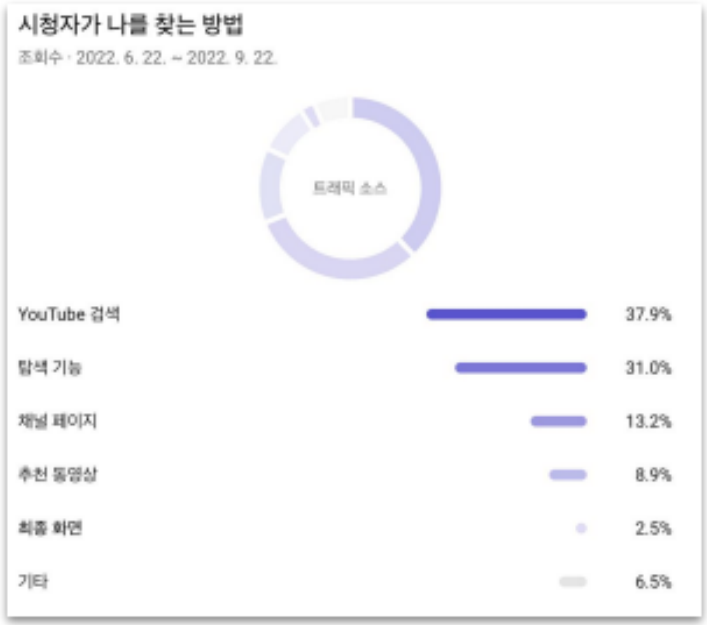


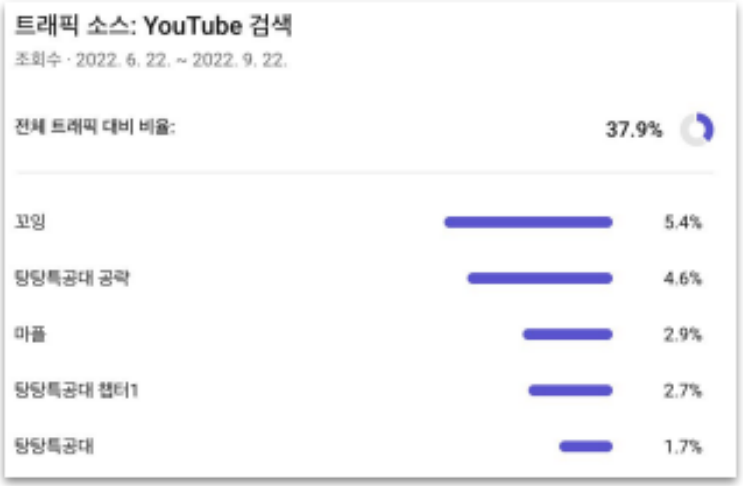


01 텍스트 마이닝

인공지능 텍스트 분석

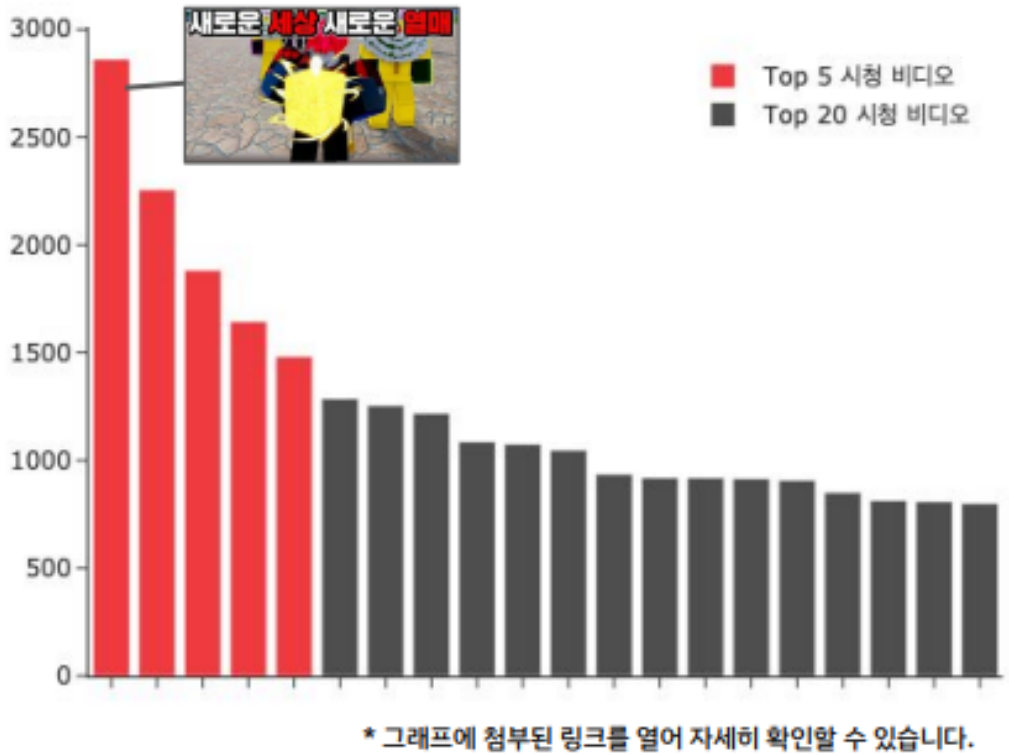
텍스트 분석을 위한 댓글 크롤러 만들기





인공지능 텍스트 분석

텍스트 분석을 위한 댓글 크롤러 만들기



* 그래프에 첨부된 링크를 열어 자세히 확인할 수 있습니다.

채널의 시청시간이 가장 높은 영상은 새로운 열매 콘텐츠입니다.

가장 높은 시청시간을 갖고있는 영상의 경우 새로운 아이템 혹은 how to를 다룬 정보성 영상으로 로블록스 게임에 적용하고자 하는 니즈를 갖고 있습니다.

해당 채널의 총 시청 시간(hr) : 59,735.22
상위 5개 비디오의 시청 시간(hr) : 10,126.0
상위 5개 비디오의 시청 시간 점유율 : 16.95%

인공지능 텍스트 분석

텍스트 분석을 위한 댓글 크롤러 만들기

01 블록스피스
검색 횟수 : 38,048
전체 검색 대비: 24.0%

02 블피
검색 횟수 : 22,774
전체 검색 대비: 14.36%

03 탕탕특공대 공략
검색 횟수 : 8,116
전체 검색 대비: 5.12%

04 로블록스
검색 횟수 : 7,022
전체 검색 대비: 4.43%

06 탕탕특공대 챕터1
검색 횟수 : 6,070
전체 검색 대비: 3.83%

07 탕탕특공대 챕터5
검색 횟수 : 5,388
전체 검색 대비: 3.4%

08 꼬잉
검색 횟수 : 5,336
전체 검색 대비: 3.37%

09 블피 코드
검색 횟수 : 3,425
전체 검색 대비: 2.16%

인공지능 텍스트 분석

텍스트 분석을 위한 댓글 크롤러 만들기

다시 도약하는 대한민국
함께 잘사는 국민의 나라

신직업	1인 미디어 특화 데이터 분석가			
정의	소셜 미디어 데이터를 수집 및 분석해 마케팅 전략을 도출하고 콘텐츠 가치에 따른 비즈니스 정책 수립			
필요 역량	▪ 데이터 수집, 분석, 시각화 ▪ 비즈니스 매니지먼트, 전략			
교육	기간	단기☐	중기☑	장기☐
	난이도	하☐	중☑	상☐
현황	▪ 인플루언서 마케팅, 기업의 소셜 미디어 마케팅이 활성화되며 소셜 미디어 내 데이터 분석을 통한 판매, 마케팅 전략의 중요성 증대			
향후 전망	▪ 소셜 미디어의 데이터 수집, 분석과 이를 활용한 비즈니스 전략을 마련할 수 있는 분석가에 대한 수요가 높아질 것으로 예상(전문가 양**)			

01 텍스트 마이닝

인공지능 텍스트 분석

- 유튜버의 동영상 댓글을 크롤링 하고, 워드 클라우드 형태를 만들어보자

- 각자의 단어 사전을 만들어 댓글의 긍정과 부정을 판단하는 모델을 만들어보자

01 텍스트 마이닝

인공지능 텍스트 분석(유튜브 동영상 크롤링)

텍스트 분석을 위한 댓글 크롤러 만들기

- python=3.10.9
- pip install selenium
- pip install openpyxl
- pip install pandas
- pip install wordcloud

Anaconda에서 가상환경 생성

- conda create -n 생성하고 싶은 이름 python=설치 버전
- pip install jupyter notebook
- pip install ipykernel

- python -m ipykernel install --user --name=커널이름 --display-name 원하는이름

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
import time
from openpyxl import Workbook
import pandas as pd
from bs4 import BeautifulSoup
import warnings
warnings.filterwarnings('ignore')
```

01 텍스트 마이닝

인공지능 텍스트 분석(유튜브 동영상 크롤링)

텍스트 분석을 위한 댓글 크롤러 만들기

Driver를 읽어오기
- driver = webdriver.Chrome("chromedriver.exe") → 크롬 드라이브를 실행함

크롬화면을 최대로 키운다
- driver.maximize_window()

- a=input("유튜브명 : ")

search_query안에 원하는 유튜버 채널을 입력함
- url = "https://www.youtube.com/results?search_query={}".format(a)

해당 홈페이지로 접속
- driver.get(url)

페이지 로드 대기시간 설정
- driver.implicitly_wait(3) →설정을 안해주면 3초 대기 없이 바로 다음 문장을 실행해 오류 딜레이 오류가 생김

01 텍스트 마이닝

인공지능 텍스트 분석(유튜브 동영상 크롤링)

텍스트 분석을 위한 댓글 크롤러 만들기

대기시간 설정

- time.sleep(1.5) → 대기 시간 중간중간에 넣어주기
- driver.refresh()
- element = driver.find_element(By.ID, "text") → 드라이버에 특정 요소를 찾아라
- element.click() → 해당 요소를 클릭하라
- element = driver.find_element(By.LINK_TEXT, "동영상") → 드라이버에 특정 요소를 찾아라
- element.click() → 해당 요소를 클릭하라
- time.sleep(3) → 대기 시간 중간중간에 넣어주기

인공지능 텍스트 분석(유튜브 동영상 크롤링)

텍스트 분석을 위한 댓글 크롤러 만들기

스크롤을 내려가지 않을 때 까지 내리는 명령어

스크롤을 내렸을 때 더 이상 내용이 없을 때

- last_height = driver.execute_script("return document.documentElement.scrollHeight")
- num_scrolls = 2 → 스크롤을 내리는 횟수 지정하기
- url_list = [] → 크롤링 되는 url을 담을 list 만들기

인공지능 텍스트 분석(유튜브 동영상 크롤링)

```
for i in range(num_scrolls): → 앞에서 지정한 num_scrolls 값에 따라 for문 시작
    #처음부터 끝까지 스크롤을 내려 해당내용 크롤링
    driver.execute_script("window.scrollTo(0, document.documentElement.scrollHeight)")
    time.sleep(1.5)
    # 스크롤을 내렸을 때 더 이상 내용이 없을 때
    new_height = driver.execute_script("return document.documentElement.scrollHeight")

    if new_height == last_height: → 더 이상 내용이 없으므로 종료해라
        break

    html_source = driver.page_source  → 페이지 소스를 받아와라
    soup = BeautifulSoup(html_source, "html.parser")→ 페이지 소스에서 html.parser에 들어가라

# 동영상의 url이 포함된 요소를 찾기
parent_elements = soup.find_all(class_ = 'yt-simple-endpoint style-scope ytd-playlist-thumbnail')
# 동영상의 요소의 href값을 추출해 저장하기
for parent_element in parent_elements:
    url = parent_element['href']
    url_list.append('https://www.youtube.com' + url)
```

인공지능 텍스트 분석(유튜브 동영상 크롤링)

```
# 저장된 url 리스트를 프린트 하시오
url_list=list(set(url_list))
print(url_list)

#유료 가입 광고 무시하기
try:
    driver.find_element_by_css_selector("#dismiss-button > a").click()
except:
    pass
```

인공지능 텍스트 분석(댓글 크롤링)

```
driver = webdriver.Chrome("chromedriver.exe")

urls=url_list[0:가지고 오고 싶은 범위까지]
id =[]
comment=[]

for url in urls:
    driver.get(url)
    driver.implicitly_wait(3)
    time.sleep(3)

    last_height = driver.execute_script("return document.documentElement.scrollHeight")
    while True:
```

인공지능 텍스트 분석(댓글 크롤링)

전장 for문 안에 넣기

```
while True:
        driver.execute_script("window.scrollTo(0, document.documentElement.scrollHeight);")
        time.sleep(1.5)

        new_height = driver.execute_script("return document.documentElement.scrollHeight")
        if new_height == last_height:
            break
        last_height = new_height

    time.sleep(1.5)

    try:
        driver.find_element_by_css_selector("#dismiss-button > a").click()
    except:
            pass
```

인공지능 텍스트 분석(댓글 크롤링)

```
except:
        pass

#여기 이어서
html_source = driver.page_source
    soup = BeautifulSoup(html_source, 'html.parser')

    id_list = soup.select("div#header-author > h3 > #author-text > span")
    comment_list = soup.select("yt-formatted-string#content-text")
    for i, j in zip(id_list,comment_list):
        id.append(i)
        comment.append(j)
```

인공지능 텍스트 분석(Word Cloud)

```
id_final = []
comment_final = []

for i in range(len(id)):
    temp_id = id[i].text
    temp_id = temp_id.replace('₩n', '')
    temp_id = temp_id.replace('₩t', '')
    temp_id = temp_id.replace('    ', '')
    id_final.append(temp_id) # 댓글 작성자

    temp_comment = comment[i].text
    temp_comment = temp_comment.replace('₩n', '')
    temp_comment = temp_comment.replace('₩t', '')
    temp_comment = temp_comment.replace('    ', '')
    comment_final.append(temp_comment) # 댓글 내용
```

Id, comment에 불필요한 문자들 제거

인공지능 텍스트 분석(Word Cloud)

```
# 저장된 내용을 아이디, 댓글 내용를 가지도록 dict으로 저장
pd_data = {"아이디" : id_final , "댓글 내용" : comment_final}

# 데이터 프레임으로 변환
youtube_pd = pd.DataFrame(pd_data)

# 데이터를 xlsx로 저장 index=False 데이터의 순서 번호 제거
youtube_pd.to_excel("지정하고 싶은 이름.xlsx", index=False)
df = pd.read_excel(" 지정하고 싶은 이름.xlsx")→ 엑셀 파일을 불러들여
df.to_csv(" 지정하고 싶은 이름 ", index=False, encoding="utf-8-sig") → 다시 csv파일로 변환함(인코딩 오류의 문제)
```

01 텍스트 마이닝

인공지능 텍스트 분석(Word Cloud)

아이디	댓글 내용
Suyoung	각본/연출 빠니보틀, 주연 곽준빈... 부부가 함께 열심히 사는 모습이 보기 좋네요 :) 파이팅♡
김영찬	진짜 곽준빈 박재한은 하늘이 맺어준 인연 아닐까..
정봉이	형님 진짜 13분동안 숨 참으면서 봤습니다... 특히 랩 하실 땐, 제 2의 셔플댄스 사건이 터진 줄 알았습니다
Wi Kim	정말 놀랐습니다. 다른건 아니고 전에 웰컴스토어 봤을때 곽튜브님의 연기를 잡아내는 거하고 이번에 찐따록에서 빠니보틀님
PowerBa	랩하실 때 등에 식은땀 나버리고 말았습니다.. 준빈쿤 연기력 무엇?
박미진	발연기인데 작품을 밀어 부친 빠니의 우정이 대단합니다^^
Jimmy	군대파트에서 삐걱 거렸지만 "진짜"들의 대화에서는 빛을 발하는 곽준빈 폼 미쳤다.
holy mol	동물의왕국에 준빈쿤이라니....빠니보틀 본인 이야기도 잔뜩 버무렸네요ㅋㅋㅋㅋㅋ
안녕안녕	신입생으로서 다큐 보는 느낌이었습니다개강총회는 도저히 스킵 없이 볼 수 없네요
줄의 JUR	아 진자 대리수치심 오지는 거 보면 정말 잘 만든 웹드라마가 아닐 수 없습니다...
주주부부	앜ㅋㅋㅋ다음편 기다려지는 동시에 어떤 수치스러움이 기다릴지 너무 두렵네요 ㅋㅋㅋㅋㅋㅋ연기가 생각보다 너무 좋으시니
박종현	자신의 상처를 예술로 표현하는 준빈쿤 폼 미쳐ㅆ다
야생차	착하고 유머 감각있는 준빈쿤 찐으로 웃는 얼굴 보기 좋음
김Daniel	연출 진짜 개 숨막힌다 .. 와 .. 보틀은 천재다 ㄹㅇ
모리의 빈	와 후배가 젓가락 놔줄 때 준빈 웃는거 보고 나도 찐으로 같이 웃었다 ㅋㅋㅋㅋ 준빈쿤 행복해야해!!!
냥냥냥냥	곽으로 투영하는 빠니의 자전적 이야기..둘은 역시 자웅동체 암수한쌍..첫 편부터 뭔가 어색하고ㅋㅋ 불편한게그냥 이 드라마
호호	나레이션으로 진행되는 형식이라 예전에 너무 재미있게봤던 짝 프로그램이 생각나기도해서 너무 기대되네요ㅋㅋㅋㅋ!!! 잘
JY MOOI	이게 바로 찐휴먼다큐지! 배우 데뷔 축하드립니다
누군지 알	형님 연기 소름 돋아서 1분만에 스탑했습니다. 하지만 형의 팬인 만큼 13번 나눠서 꼭 다보겠습니다
정리어터	라방이나 방송에서 들었던 내용을 실제 드라마라니..벌써 기대기대..어색하면서 어색하지않은척..입꼬리가 들썩들썩댑니다.
din O	준빈이형 마지막에 웃는데 왤케 킹받지ㅋㅋㅋㅋ 재밌게 보고 갑니다
예슈화나	9:08 이부분이 디테일이 미친게힙합좋아한다고 하고 래퍼 누구있는지 잘 몰라서 잠시 생각하다가 예능 많이나오는 넉살떠올
wodfd w	와 진짜 하이퍼릴리즘 현장감 미쳤다 보면서 개소름돋음 너무 현실적이어서 ㅋㅋㅋㅋㅋㅋㅋ
재똥이	연기까지 완벽한 준빈이형 폼 미쳤다

인공지능 텍스트 분석(Word Cloud)

텍스트 wordcloud 생성

```
# 그래프를 그릴 수 있는 패키지 설치
pip install matplotlib


# 폰트 다운로드
import matplotlib.font_manager as fm

for f in fm.fontManager.ttflist:
    if 'Nanum' in f.name:
        print(f.name, f.fname)

# 폰트 경로 설정
font_path = 'C:/Windows/Fonts/NanumPen.ttf'
```

인공지능 텍스트 분석(Word Cloud)

텍스트 wordcloud 생성

#방법 1 : 불필요한 단어를 설정하고 제거하기

```
from wordcloud import WordCloud, STOPWORDS

stopwords = set(STOPWORDS)
stopwords.add("ㅋㅋ")
stopwords.add("ㅎㅎ")
stopwords.add("너무")
stopwords.add("진짜")
stopwords.add("와")
```

인공지능 텍스트 분석(Word Cloud)

텍스트 wordcloud 생성

#방법 2 : 불필요한 단어를 설정하고 제거하기

```
with open("stopwords.txt", encoding="utf-8") as f:
    stopwords = set(f.read().splitlines())
```

인공지능 텍스트 분석(Word Cloud)

텍스트 wordcloud 생성

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# CSV 파일을 pandas 데이터프레임으로 읽어옴
df = pd.read_csv('파일이름.csv',encoding='utf-8')

# 댓글 내용이 저장된 컬럼 선택
comment_col = '댓글 내용'
comment_list = df[comment_col].tolist()
new_list = []
for x in comment_list:
    new_list.append(str(x))

# 모든 댓글 내용을 하나의 문자열로 결합
text = ' '.join(new_list)
```

인공지능 텍스트 분석(Word Cloud)

텍스트 wordcloud 생성

```
# 워드 클라우드 생성
wordcloud = WordCloud(font_path=font_path,width=800, height=800,
                      background_color='white',stopwords=stopwords, min_word_length=4).generate(text)

# 단어의 빈도 계산
wordcloud.generate_from_frequencies(wordcloud.process_text(text.lower()))
word_freq = wordcloud.process_text(text.lower())

# 워드 클라우드 출력
plt.figure(figsize=(8, 8), facecolor=None)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.tight_layout(pad=0)

# 파일로 저장
plt.savefig('wordcloud.png')
```

인공지능 텍스트 분석(Word Cloud)

텍스트 wordcloud 생성

 # 해당 숫자보다 낮은 빈도의 단어를 도출함

딕셔너리에 단어 : 빈도수를 입력하고 단어가 특정길이 보다 작을 때 이것을 새로운 딕셔너리에 추가함

 # 낮은 빈도의 단어를 프린트함

특정 빈도수 보다 작은 단어들과 그 단어의 빈도를 아래와 같이 프린트 하시오

```
맛있겠네: 2
먹구싶다: 1
맛깔나게: 1
먹으시네ㅋㅋㅋ방금: 1
배부르게먹었는데: 1
이거보니: 1
먹고싶네ㅋㅋ내일: 1
돈까스다ㅋㅋㅋㅋ: 1
먹방하나는: 1
```

01 텍스트 마이닝

인공지능 텍스트 분석(긍부정 단어 도출)

긍부정 단어사전

```
def train_sentiment_analysis_model(positive_file, negative_file):
    # 긍정 단어와 부정 단어를 읽어서 리스트로 변환합니다.
    with open(positive_file, 'r', encoding='utf-8') as f:
        positive_words = f.read().splitlines()

    with open(negative_file, 'r', encoding='utf-8') as f:
        negative_words = f.read().splitlines()

    # 긍정적인 문장과 부정적인 문장을 학습 데이터로 생성합니다.
    positive_sentences = [" ".join(positive_words)] * len(positive_words)
    negative_sentences = [" ".join(negative_words)] * len(negative_words)

    # 학습 데이터와 레이블을 생성합니다.
    X = positive_sentences + negative_sentences
    y = [1] * len(positive_sentences) + [0] * len(negative_sentences)

    # CountVectorizer를 사용하여 단어의 빈도수를 측정합니다.
    vectorizer = CountVectorizer(token_pattern=r"₩b₩w+₩b") → 단어의 빈도를 측정하는데 사용
    X = vectorizer.fit_transform(X) → 행렬도 문자를 반환

    # LogisticRegression을 사용하여 모델을 학습합니다.
    clf = MultinomialNB() → 변환된 문자 행렬을 학습함(다른 Classification 알고리즘으로도 사용 가능)
    clf.fit(X, y)

    return vectorizer, clf
```

```
pip install -U scikit-learn scipy

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
import numpy as np
Import itertools
```

positive_sentences

```
['good nice gg 최고였다 멋지다 영화 감독 줄거리는 love',
 'good nice gg 최고였다 멋지다 영화 감독 줄거리는 love',
 'good nice gg 최고였다 멋지다 영화 감독 줄거리는 love',
 'good nice gg 최고였다 멋지다 영화 감독 줄거리는 love',
 'good nice gg 최고였다 멋지다 영화 감독 줄거리는 love',
 'good nice gg 최고였다 멋지다 영화 감독 줄거리는 love',
 'good nice gg 최고였다 멋지다 영화 감독 줄거리는 love',
 'good nice gg 최고였다 멋지다 영화 감독 줄거리는 love',
 'good nice gg 최고였다 멋지다 영화 감독 줄거리는 love']
```

negative_sentences

```
['bad nedd nervous 별로 싫다 지루하고',
 'bad nedd nervous 별로 싫다 지루하고',
 'bad nedd nervous 별로 싫다 지루하고',
 'bad nedd nervous 별로 싫다 지루하고',
 'bad nedd nervous 별로 싫다 지루하고',
 'bad nedd nervous 별로 싫다 지루하고']
```

인공지능 텍스트 분석(긍부정 단어 도출)

긍부정 단어사전

```
def train_sentiment_analysis_model(positive_file, negative_file):
    # 긍정 단어와 부정 단어를 읽어서 리스트로 변환합니다.
    with open(positive_file, 'r', encoding='utf-8') as f:
        positive_words = f.read().splitlines()

    with open(negative_file, 'r', encoding='utf-8') as f:
        negative_words = f.read().splitlines()

    # 긍정적인 문장과 부정적인 문장을 학습 데이터로 생성합니다.
    positive_sentences = [" ".join(positive_words)] * len(positive_words)
    negative_sentences = [" ".join(negative_words)] * len(negative_words)

    # 학습 데이터와 레이블을 생성합니다.
    X = positive_sentences + negative_sentences
    y = [1] * len(positive_sentences) + [0] * len(negative_sentences)

    # CountVectorizer를 사용하여 단어의 빈도수를 측정합니다.
    vectorizer = CountVectorizer(token_pattern=r"₩b₩w+₩b") → 단어의 빈도를 측정하는데 사용
    X = vectorizer.fit_transform(X) → 행렬도 문자를 반환

    # LogisticRegression을 사용하여 모델을 학습합니다.
    clf = MultinomialNB() → 빈도수로 분류하는 모델(다른 Classification 알고리즘으로도 사용 가능)
    clf.fit(X, y)

    return vectorizer, clf
```

```
pip install -U scikit-learn scipy

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
import numpy as np
```

대체 가능

```
max_permutations = 20

pos_per = list(itertools.islice(itertools.permutations(positive_words), max_permutations))
neg_per = list(itertools.islice(itertools.permutations(negative_words), max_permutations))

for p in positive_sentences:
    pos_list.append(list(p))

for i in range(0,len(pos_list)):
    a=[" ".join(pos_list[i])]
    positive_list.append(a)

for num in positive_list:
    pos_convert.append(num[0])

for n in negative_sentences:
    neg_list.append(list(n))

for i in range(0,len(neg_list)):
    a=[" ".join(neg_list[i])]
    negative_list.append(a)

for num in negative_list:
    neg_convert.append(num[0])
```

인공지능 텍스트 분석(긍부정 단어 도출)

긍부정 단어사전

```
def predict_sentiment_analysis(text, vectorizer, clf):
    # 입력된 텍스트를 벡터화합니다.
    X = vectorizer.transform([text])

    # 모델을 사용하여 감성을 예측합니다.
    y_pred = clf.predict(X)

    # 예측 결과에 따라 출력 메시지와 카운트 정보를 선택합니다.
    if y_pred[0] == 1:
        result = {"sentiment": "긍정적인 단어", "positive_count": 1, "negative_count": 0}
    else:
        result = {"sentiment": "부정적인 단어", "positive_count": 0, "negative_count": 1}

    # 입력된 텍스트에 포함된 긍정 단어와 부정 단어를 카운트합니다.
    for word in text.split():
        if word in positive_words:
            result["positive_count"] += 1
        elif word in negative_words:
            result["negative_count"] += 1

    return result
```

인공지능 텍스트 분석(긍부정 단어 도출)

긍부정 단어사전

```
with open('text/positive_words.txt', 'r',encoding='utf-8') as f:
    positive_words = f.read().splitlines()

with open('text/negative_words.txt', 'r',encoding='utf-8') as f:
    negative_words = f.read().splitlines()


vectorizer, clf=train_sentiment_analysis_model("text/positive_words.txt","text/negative_words.txt")

text = input("분석할 문자열을 입력하세요: ")

predict_sentiment_analysis(text, vectorizer, clf)
```

01 텍스트 마이닝

인공지능 텍스트 분석(긍부정 단어 도출)

df = pd.read_csv("파일이 위치한 경로.csv", encoding='utf-8-sig') → csv파일을 데이터 프레임 형태로 불러오기
comment_col = '댓글 내용'
comment_list = df[comment_col].tolist() → 데이터 프레임에서 comment_col에 지정한 변수를 모두 리스트로 바꾸는 함수
str(x) → x에 대한 형식을 문자로 바꾸는 함수

	아이디	댓글 내용	감정
0	빠니보틀 Pani Bottle	안녕하세요 국내 최고령 찐따(아마도)입니다	부정적인 단어
1	Laurine Mescle	안녕하세요! 저는 첫 번째 줄에 있는 소녀들 중 한 명이었는데, 정말 좋았어요! 칸...	부정적인 단어
2	멍부	와....구독자 1000명 되어서 울었었다는 그 빠니보틀 맞냐... 가슴이 웅장해진...	부정적인 단어
3	고유빈	칸에 초청받는게 꿈인 영화과 입시생 고등학생입니다. 빠니보틀님 영상 보며 세계여행...	부정적인 단어
4	이예쓰 세계여행 Lee Yes travel	잠시 지치시기도 했지만, 다시 더 활짝 웃는 모습 볼 수 있어서 너무 좋습니다!여행...	부정적인 단어
...	...	...	...
1914	티티	형은 그냥 혼자가 가장 느낌있어 ㅇㅇ 이제 혼자 브라질 밀림 가자.	부정적인 단어
1915	갈길감	혼자 배낭매고 여기저기 돌아다니고, 가끔 지인들 만나는 영상을 올리던 빠니가 그립다...	부정적인 단어
1916	일본미국에서사업중.매국노X kats.no.	개싫다	부정적인 단어
1917	박상일	물개옆에 저렇게 가까히 눕는 행동 참...상식밖에 행동이네요 외국인이 봤을때 동양인...	부정적인 단어
1918	sy kim	오킹만 다른 맴버였으면 좋았을걸.....남자가 봐도 중간에 보기 싫더라구요.....	부정적인 단어

1919 rows × 3 columns

인공지능 텍스트 분석(긍부정 단어 도출)

긍부정 단어사전

```
df = pd.read_csv('파일이 위치한 경로.csv', encoding='utf-8-sig')
comment_col = '댓글 내용'
comment_list = df[comment_col].tolist()
new_list = []
for x in comment_list:
    new_list.append(str(x))
```

긍정과 부정을 판단하는 함수를 활용해
긍정과 부정 문장의 결과를
emotion이라는 리스트를 만들어 저장하시오