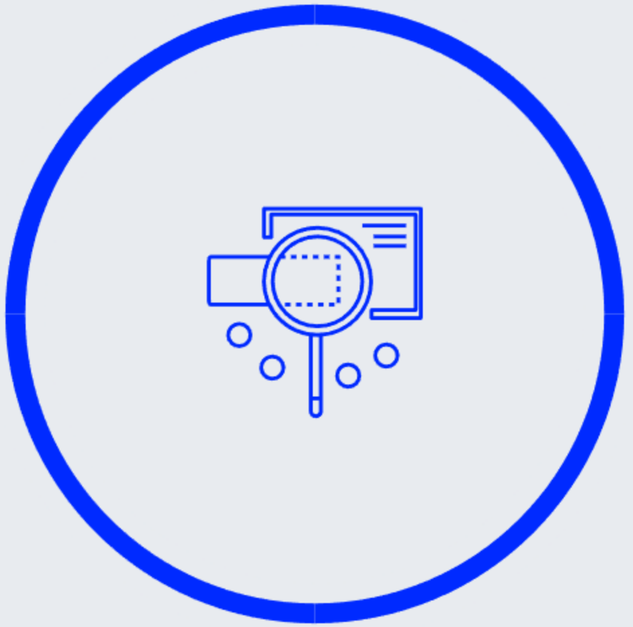
Project Presentation

[프로젝트 발표] - 축구 선수 친밀도 계산

20191658 장정안

목차

프로젝트 소개



프로젝트 코드 소개

프로젝트 결과

프로젝트 소개 – 개요

축구 선수 친밀도 계산

원래 축구를 좋아해서 축구 관련 기사를 이용해서 프로젝트를 진행하려고 했는데, 참고하라고 올려주신 프로젝트에 스포츠 선수 친밀도 계산을 하는 프로젝트가 있어서 스포츠 중 축구를 택해서 축구 선수 친밀도 계산을 하기로 결정하였다.
실습을 하면서 KCC150 한국어 뉴스 파일을 사용하였는데, 축구 관련한 문장만이 있는 것이 아니었다. 본인이 하려는 프로젝트에 있어서 조금은 오래된 자료라는 점과 축구 선수에 대한 데이터가 적을 것이라는 점이 문제라고 생각이 들었다. 그래서 직접 크롤링을 통한 축구 기사 자료 수집을 한 뒤에 형태소 분석 후 워드 임베딩으로 모델링 하여 결과물을 도출하였다

크롤링 키워드(예시)

1. 축구
2. EPL
3. 라리가
4. K리그
5. 분데스리가

내가 알고자 하는 축구 선수(예시)

1. 손흥민
2. 황희찬
3. 이강인
4. 김민재
5. 백승호

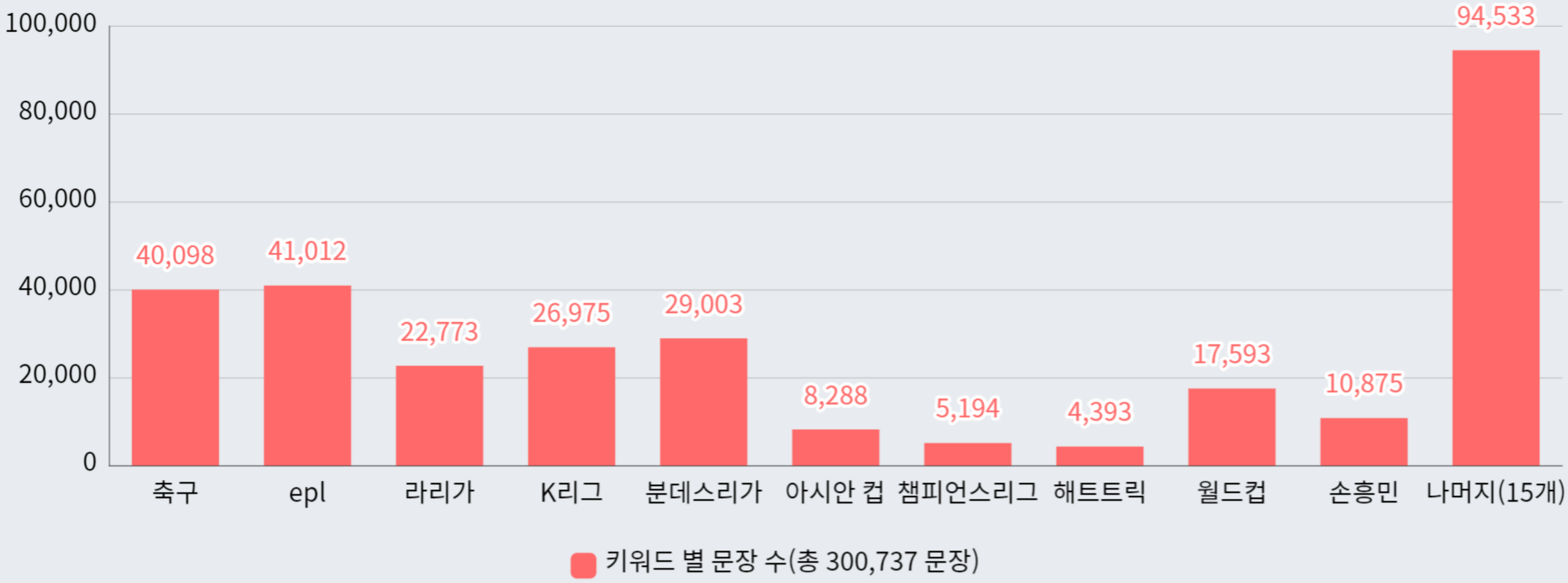
손흥민 선수와 친한 선수(결과 예시)

케인

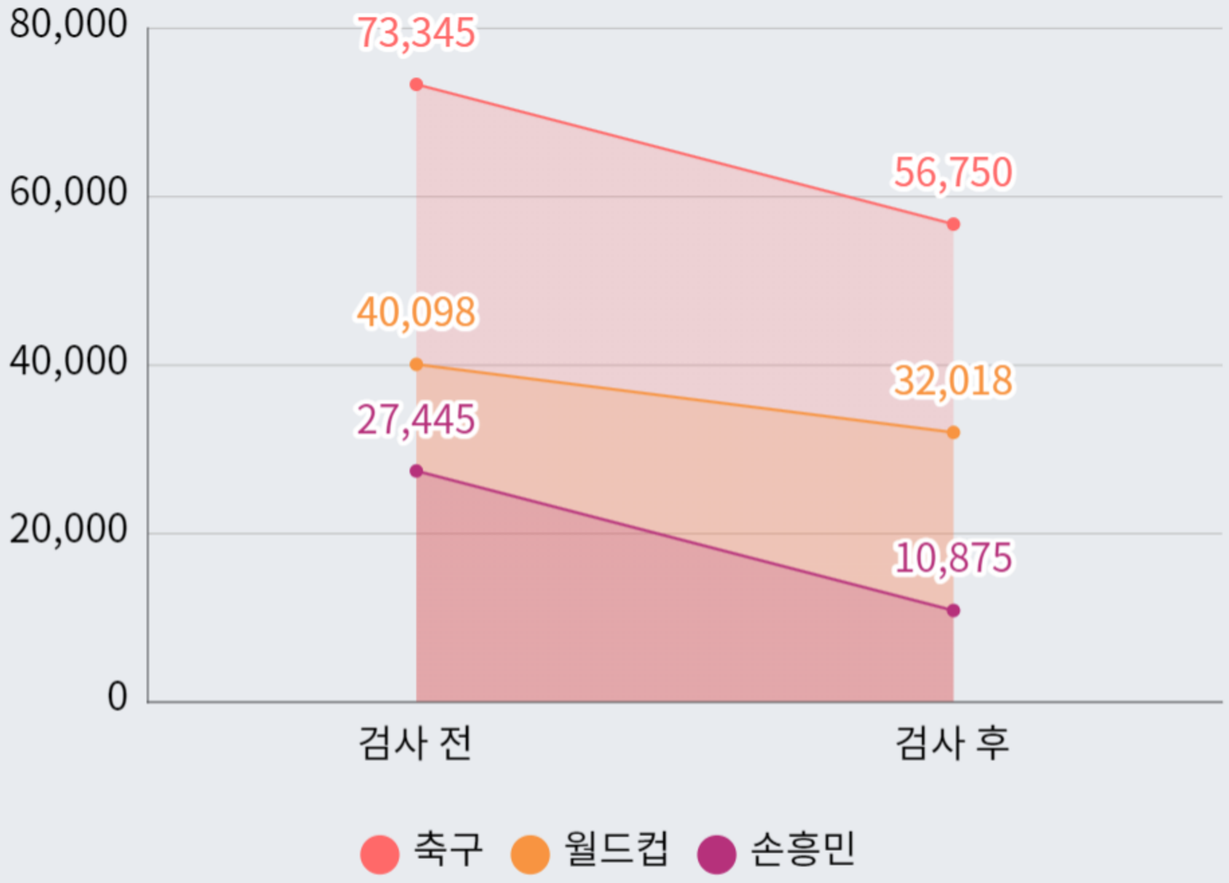
황의조

히샬리송

프로젝트 소개 – 자료 수집 키워드



프로젝트 소개 – 제목 유사도 검사 후 문장 수 비교



프로젝트 코드 소개

```
#
import requests
from bs4 import BeautifulSoup
import re
import pandas as pd

USER_AGENT = "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.3 Safari/605.1.15"

# 페이지 url 형식에 맞게 바꾸어 주는 함수 만들기
# 입력값 수를 1, 11, 21, 31 ...만들어 주는 함수
def make_page_num(num):
    if num == 1:
        return num
    elif num == 0:
        return num + 1
    else:
        return num + 9 * (num - 1)

# 크롤링할 url 생성하는 함수 만들기(검색어, 크롤링 시작 페이지, 크롤링 종료 페이지)
def make_url(search, start_pg, end_pg):
    if start_pg == end_pg:
        start_page = make_page_num(start_pg)
        url = "https://search.naver.com/search.naver?where=news&sm=tab_pge&query=" + search + "&start=" + str(start_page)
        print("Generated URL: ", url)
        return [url]
    else:
        urls = []
        for i in range(start_pg, end_pg + 1):
            page = make_page_num(i)
            url = "https://search.naver.com/search.naver?where=news&sm=tab_pge&query=" + search + "&start=" + str(page)
            urls.append(url)
```

1. 웹 크롤링 코드(네이버뉴스)

```
import csv


with open('NaverNews_축구.csv', 'r', encoding='utf-8') as csv_file:
    reader = csv.reader(csv_file)
    with open('축구_txt', 'w', encoding='utf-8') as txt_file:
        for row in reader:
            txt_file.write(row[0] + '\n')
```

2. 수집 자료 전처리 코드

```
# most_similar 결과를 수치를 기반으로 한 막대그래프를 시각화하여 보여주는 메소드
def visualize_most_similar(model, keywords):
    font_name = fm.FontProperties(fname="c:/Windows/Fonts/malgun.ttf").get_name()
    rc('font', family=font_name)
    mpl.rcParams['axes.unicode_minus'] = False #한글 폰트 사용시 마이너스 폰트 깨짐 해결

    similar_words = model.wv.most_similar(positive=[keyword,u'등료','곰비'], negative=[u'불화'], topn=5)
    words = [word[0] for word in similar_words]
    scores = [word[1] for word in similar_words]
    plt.bar(words, scores)
    plt.xticks(rotation=90)

    plt.show()

# network node 그래프
def setEdges_simWords(model,word,n1=10,n2=5):
    simWords = model.wv.most_similar(word,topn=n1)

    G=nx.Graph()
    for (w2,wgt) in simWords:   # 1차 확장
        G.add_edge(word,w2,weight=round(wgt,2))

    for (w2,wgt) in simWords:
            simWords2 = model.wv.most_similar(w2,topn=n2)
            for (w3,wgt) in simWords2:  # 2차 확장
                G.add_edge(w2,w3,weight=round(wgt,2))

    return G
```

3. 최종 결과물 코드

프로젝트 코드 결과

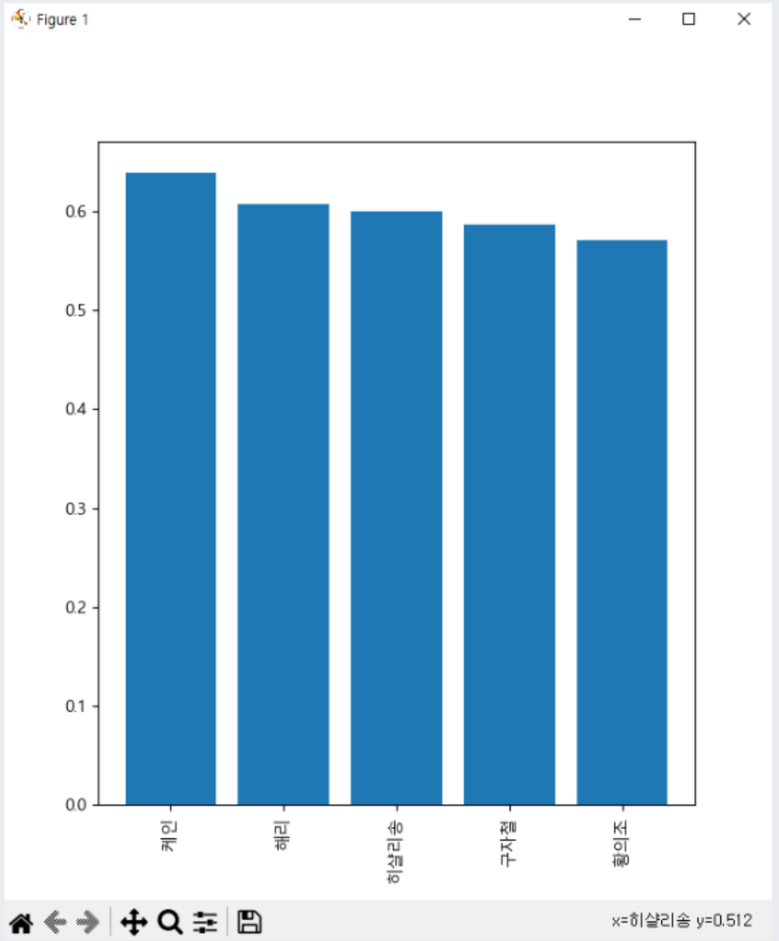
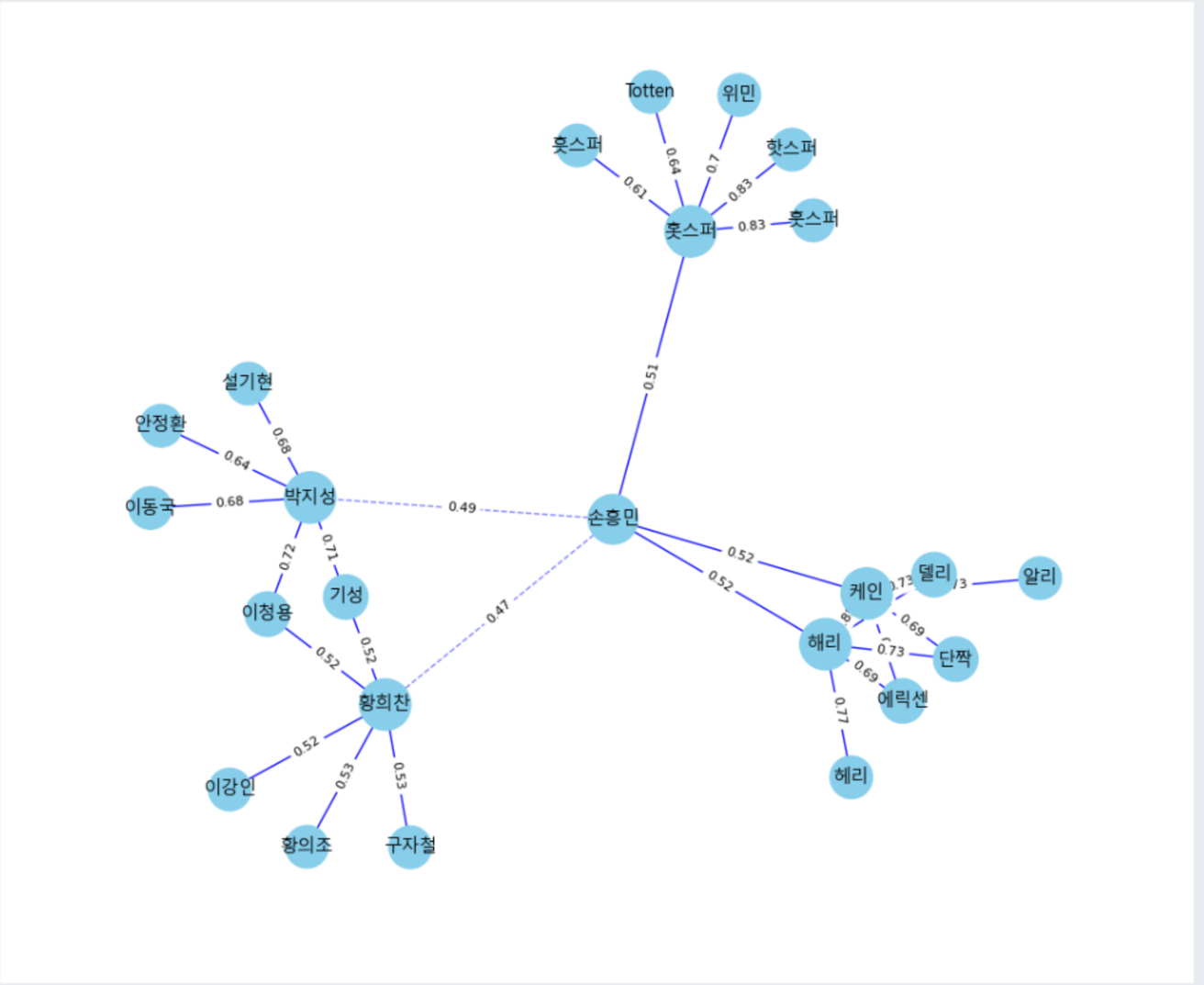


Figure 1
0.6
0.5
0.4
0.3
0.2
0.1
0.0
케인
해리
허설리송
구자철
황의조
x=허설리송 y=0.512


Totten
위민
훗스퍼
핫스퍼
훗스퍼
훗스퍼
0.61
0.64
0.7
0.83
0.83
0.51
설기현
안정환
이동국
박지성
0.68
0.64
0.68
0.49
손흥민
0.72
0.71
0.52
0.52
이청용
기성
케인
델리
알리
해리
단짝
에릭센
헤리
0.47
0.52
0.52
황희찬
이강인
황의조
구자철
0.52
0.53
0.53
0.77
0.69
0.73
0.69

프로젝트 결과 – 개선할 점

목표로 했던 선수에 대한 친밀도 계산은 잘 이루어졌다. 이것으로 친밀도 계산의 목표 선수와 밀접한 키워드 자료 수집이 굉장히 중요하다라는 것을 알 수 있다.

손흥민

손흥민 선수를 특정 키워드로 데이터를 수집을 진행하였고, 특히 손흥민 선수와 더 밀접하다고 생각되는 키워드로 문장을 더 많이 수집하였다. 그 결과로 손흥민 선수에 대한 친밀도 계산 결과는 예측한 것과 거의 유사하지만, 다른 선수들을 키워드로 검색했을 때 예측했던 것과 차이가 있었음을 알 수 있다.
따라서 원하는 선수의 친밀도 계산을 위해서 그 선수 이름에 해당하는 키워드와 그 선수와 밀접하게 관련된 키워드의 자료 수집이 중요하다는 결론을 내릴 수 있었다. 만약 실제로 선수간 친밀도 계산을 축구 팀 내에서 선수를 영입하거나, 팀원들의 관계 조사 등을 할 때 그 선수와 관련된 더 많은 데이터를 수집하여 진행하게 되면 더 정확한 결과 값들을 얻을 수 있을것이다.

Thank you

감사합니다.