Capstone Project

Cha Jae Hyuck

Aprill 2, 2019

Data Scientist Nanodegree

Detect correlation between morpheme in articles and real epidemic data

Definition

Project Overview

전염병은 많은 사람들의 목숨을 빼앗았고, 지금 순간에도 많은 병들이 잠재적으로 퍼지고 있다. 사람들은 공기, 곤충, 동물, 물 같은 다양한 원인에 의해 전염병에 걸린다.

하지만, 언제, 어디서, 어떻게 전염되고, 발생했는지 또는 종결이 되었는지 일반 사람들은 알지 못하기 때문에, 전염병에 대해 매우 무지하다. 일반 대중들은 오직, 뉴스 기사나 정부 발표에 의해서 크고 강력한 전염병들만 파악하게 된다. 그러므로, 많이 알려지지 않거나, 작은 전염병들은 지나치게 된다. 그래서, 이 프로젝트에서는 쓰여진 기사 내용 안의 질병 관련 단어들의 빈도수와 실제 질병의 발생된 유무 관계를 분석하여 실제로 발생하고 사라진 전염병들을 확인한다. 앞으로 더 나아가 있을 전염병을 예방하기 위하여 전염병 발생하기 전 기사 내용 안에 전염병을 암시하는 어떤 전조적인 단어가 있는지에 대해 찾아보려 한다.

Problem Statement

목적은 특정 날짜 범위를 기준으로 그 안의 기사들 내용에서의 질병 관련 단어들의 빈도수와 실제 질병의 유무 관계를 찾는 것이다. 작업들은 다음과 같다.

1. 네이버 뉴스 기사들을 수집한다.(뉴스 기사들은 2015 년 5 월 20 일부터 수집한 뉴스 기사들이다.)
2. 모든 뉴스 기사들을 1 년 단위로 형태소를 분석해서, '환자'이라는 형태소의 빈도수를 뽑아 평균을 계산한다. (결과값 : 평균 값) –'환자'를 선택한 이유: 모집단에서 추출한 표본집단 샘플에서 형태소 분석을 돌려봤을 때, 병명을 제외한 형태소 중 가장 빈도수가 높았다.
3. 평균을 기준으로 모든 뉴스 기사들을 1 달 단위로 형태소를 분석해서 평균보다 '환자' 형태소의 빈도수가 높은 달만 선택한다. (결과 값 : 높은 달(형태소 목록/빈도수)
4. 선택된 달의 형태소들을 바탕으로, 질병관리본부의 데이터를 이용하여 실제 그 날짜 구간에 어떤 전염병이 발생 했는지의 유무를 확인한다. (확인 방법은 빈도수가 가장 많이 기록된 달의 형태소에서 병명을 수동으로 뽑아내서(결과 값: 형태소 목록 안에 있는 병명), 실제 데이터에다가 그 병명이 가장 많이 발생된 달을 뽑아서(결과 값 : 해당 달) 확인하는 방법)

Analysis

Data Exploration

수집 기사 데이터는 2015 년부터 지금까지 수집한 여러 포털에서의 모든 뉴스기사를 수집한 것이다. Scrapy 라는 웹 크롤링 파이썬 라이브러리를 이용하여 수집할 데이터의 필드를 정의하고(item.py), 정보를 얻기 위한 코드를

작성했다.(spider.py) 크롤링한 웹 기사 데이터는 개인 DB 에 저장했다. 데이터는 총 개이며, 우리는 이 데이터에서 기사 내용과 수집 시간을 이용한다. 수집 기사 데이터는 다음과 같은 필드가 있다.

*id : 수집된 기사 인덱스 (Integer)

*aid: 기사 번호 (Integer)

*title : 기사 제목 (String)

*content : 기사 내용 (String)

*aDate : 수집 시간 (TimeStamp)

*nUrl : 기사 url – 네이버 (String)

*pUrl : 기사 원본 url (String)

*nClass : 기사 분야 (String)

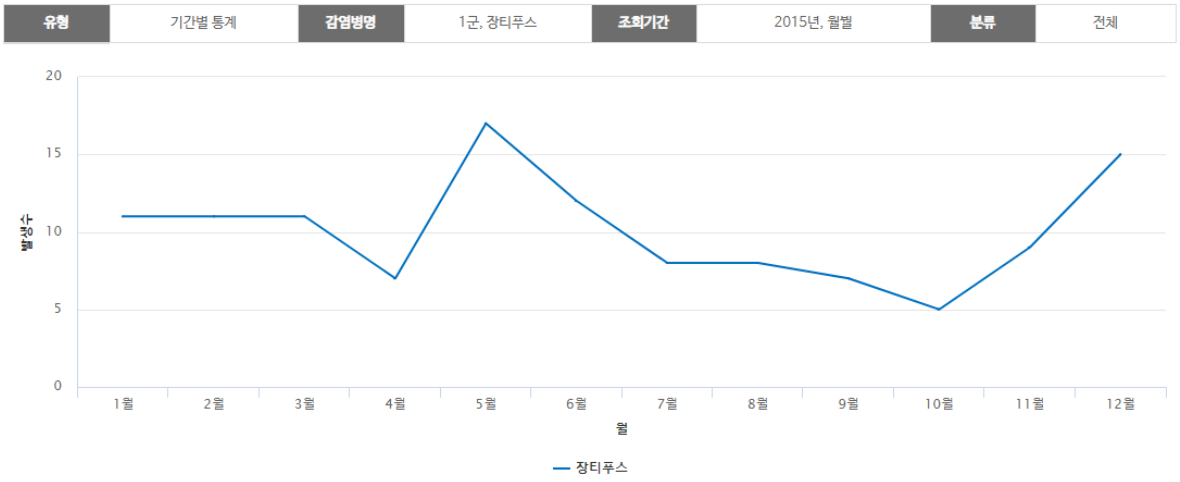
*press : 기사 출처 (String)

*subclass, pdf, numComment etc.

	id [PK] integer	aid bigint	title character varying (60)	content text	aDate timestamp with time zone	nUrl text	pUrl text	subClass character varying (10)	nClass character varying (10)	press character varying (10)	pdf character varying (20)	numComment integer
1	2	8860957	문재인 대통령, '마타헬라 ...		2018-10-17 19:34:00+00	https...	http:/...		정치	뉴시스	0008860957.pdf	0
2	3	8860876	김성태 "육군 탈영자 최근 ...		2018-10-17 18:27:00+00	https...	http:/...		정치	뉴시스	0008860876.pdf	0
3	4	8860940	문 대통령, '마타헬라 대통...		2018-10-17 19:23:00+00	https...	http:/...		정치	뉴시스	0008860940.pdf	0
4	5	8860698	서구 인구정책 민간주진단		2018-10-17 17:09:00+00	https...	http:/...		정치	뉴시스	0008860698.pdf	0
5	6	10407670	문 대통령, 마테헬라 이탈...		2018-10-17 19:15:00+00	https...	http:/...		정치	연합뉴스	0010407670.pdf	0
6	7	1293645	작심 반박 나선 청와대 "한...		2018-10-17 19:35:00+00	https...	http:/...		정치	MBN	0001293645.pdf	0
7	8	2487521	"폐지 말아 근근이 하루살...		2018-10-17 18:09:00+00	https...	http:/...		경제	디지털타임스	0002487521.pdf	0
8	9	4028126	"증선위 저분 취소해달라"...		2018-10-17 18:16:00+00	https...	http:/...		경제	한국경제	0004028126.pdf	0
9	10	8860910	서울 아파트 분양가 3.3㎡...		2018-10-17 18:42:00+00	https...	http:/...		경제	뉴시스	0008860910.pdf	0
10	11	4226847	제21회 농과학기술대상 시...		2018-10-17 17:27:00+00	https...	http:/...		경제	이데일리	0004226847.pdf	0
11	12	2487547	건설경기 재건 나선 쌍용건...		2018-10-17 18:10:00+00	https...	http:/...		경제	디지털타임스	0002487547.pdf	0
12	13	4110363	"세계경제 짧은 잔치 끝나...		2018-10-17 17:23:00+00	https...	http:/...		경제	파이낸셜뉴스	0004110363.pdf	0
13	14	334699	소비 물었는데 매출 쑥... 막...		2018-10-17 17:35:00+00	https...	http:/...		경제	한국일보	0000334699.pdf	0
14	15	845496	공유사무실 세계적 기업 '...		2018-10-17 19:37:00+00	https...	http:/...		경제	부산일보	0000845496.pdf	0
15	16	3643321	김장호 대장 빈소찾은 산악...		2018-10-17 18:32:00+00	https...	http:/...		사회	뉴스1	0003643321.pdf	0
16	17	2899712	유지원 비리 근절, 지금이 '...		2018-10-17 17:42:00+00	https...	http:/...		사회	경향신문	0002899712.pdf	0
17	18	4236530	[매경CEO특강] 고한승 삼...		2018-10-17 17:47:00+00	https...	http:/...		사회	매일경제	0004236530.pdf	0
18	19	4236531	은행권 돌기 트렌드 반영한...		2018-10-17 17:47:00+00	https...	http:/...		사회	매일경제	0004236531.pdf	0

Data Visualization

이 그래프는 특정 연도의 월별 장티푸스 발생수 이다. 이와 비슷하게, 1~4 군 감염병들의 각 연도별/월별 발생수를 확인할 수 있다. 이 데이터들은 우리가 분석한 데이터와 실제 데이터의 일치함을 비교하는 기준으로 이용된다.



구분	계	1월	2월	3월	4월	5월	6월	7월	8월	9월	10월	11월	12월
장티푸스	121	11	11	11	7	17	12	8	8	7	5	9	15

<출처> 감영병 포털 : http://www.cdc.go.kr

Methodology

Data Preprocessing

기사 수집 데이터는 공급이 되고, 100% 정확도를 가지고 있습니다. 그래서, 전처리가 필요 없습니다.

Implementation

-모든 뉴스 기사들을 1 년 단위로 형태소를 분석해서(2015 년은 5 월~12 월), '환자'이라는 형태소의 빈도수를 뽑아 평균을 계산한다.(Konlpy, Counter, 평균 계산)

```
def get_tag(text, ntags):
    spliter = Okt()
    nouns = spliter.nouns(text)

    count = Counter(nouns)
    return_list = []

    for n, c in count.most_common(ntags):
        if (n == "환자"):
            temp = {'tag': n, 'count': c}
            return_list.append(temp)

    return return_list
```

```
import time
start_time = time.time()

noun_list = []
count_list = []

for i in range(5, 13):
    article_df_month = article_df.loc[(article_df['aDate'].dt.month == i)]

    noun_count = 200
    tags = get_tag(article_df_month.to_string(), noun_count)
    #output_file_name = "article_count.txt"
    #open_output_file = open(output_file_name, 'w', -1, "utf-8")

    for tag in tags:
        noun = tag['tag']
        count = tag['count']
        print(str(i) + "월", noun, ":", count)
        noun_list.append(str(i))
        count_list.append(count)
        #open_output_file.write('{} {}₩n'.format(noun, count))
    #open_output_file.close()
    #print(time.time() - start_time)
```

```
patient_df = pd.DataFrame({"Month" : noun_list, "Patient" : count_list})
patient_df
```

```
total = patient_df.sum(axis = 0)[1]
day = 12 + 30 + 31 + 31 + 30 + 31 + 30 + 31
month_avr = (total / day) * 30
print("Month Average :", month_avr)

select_month = patient_df[patient_df['Patient'] > month_avr]
select_month
```

```
Month Average : 2331.6371681415926
```

	Month	Patient
0	5	2899
1	6	10629

-평균을 기준으로 모든 뉴스 기사들을 1 달 단위로 형태소를 분석해서 평균보다 '환자' 형태소의 빈도수가 높은 달만 선택하고, 그 달의 형태소 목록을 보여준다. (konlpy, Counter, 최대빈도 달 계산) 최(결과 값 : 높은 달(형태소 목록/빈도수)

```
def get_tags(text, ntags):
    spliter = Okt()
    nouns = spliter.nouns(text)

    count = Counter(nouns)
    return_list = []

    for n, c in count.most_common(ntags):
        temp = {'tag': n, 'count': c}
        return_list.append(temp)

    return return_list
```

```
for i in range(select_month.shape[0]):
    month = select_month['Month'][i]

    article_df_month = article_df.loc[(article_df['aDate'].dt.month == int(month))]

    noun_count = 10
    tags = get_tags(article_df_month.to_string(), noun_count)

    print(str(month) + "월")
    for tag in tags:
        noun = tag['tag']
        count = tag['count']
        print(noun, ":", count)
```

month_avr : 월별 "환자" 형태소의 빈도수의 1 년 평균 (2015 년의 경우 5 월 20 일~ 12 월 31 일까지의 데이터이기 때문에 "환자" 형태소의 빈도수/일(day) * 30 으로 평균을 계산)

select_month : "환자" 형태소의 빈도수가 평균보다 높은 달

5월
메르스 : 5477
환자 : 2899
중동 : 1752
명 : 1722
호흡기 : 1473
증후군 : 1345
감염 : 1281
기자 : 1020
의심 : 1014
발생 : 766
6월
메르스 : 67305
기자 : 17393
명 : 11821
환자 : 10629
중동 : 10512
서울 : 10133
호흡기 : 9231
병원 : 8575
증후군 : 8269
뉴스 : 6312

-선택된 달의 형태소들을 바탕으로, 질병관리본부의 데이터를 이용하여 실제 그 날짜 구간에 어떤 전염병이 발생 했는지의 유무를 확인한다.(수동으로 찾은 병명 형태소를 input 으로 실제 데이터에서 그 병명이 가장 많이 발생한달을 계산) (확인 방법은 빈도수가 가장 많이 기록된 달의 형태소에서 병명을 수동으로 뽑아내서(결과 값: 형태소 목록 안에 있는 병명), 실제 데이터에다가 그 병명이 가장 많이 발생된 달을 뽑아서(결과 값 : 해당 달) 확인하는 방법)

```
mers_df = pd.read_csv("MERS.csv")
patient_df['Month'] = patient_df['Month'].astype(int)
mers_df['Month'] = mers_df['Month'].astype(int)
merge_2015 = pd.merge(patient_df, mers_df)
merge_2015.corr()
```

Refinement

get_tag() : "환자" 형태소의 빈도수를 list 로 반환하는 함수. konlpy 의 Okt 를 사용, nouns()함수를 통해서 명사를 추출하고 Counter 객체를 이용하여 빈도수 파악.

@param1 : 형태소를 분석할 텍스트

@param2 : 빈도수가 높은 명사를 추출하고자 하는 개수

get_tags() : 빈도수가 높은 n 개의 형태소를 list 로 반환하고 출력하는 함수. Konlpy 의 Okt 를 사용, nouns()함수를 통해서 명사를 추출하고 Counter 객체를 이용하여 빈도수 파악.

@param1 : 형태소를 분석할 텍스트

@param2 : 빈도수가 높은 명사를 추출하고자 하는 개수

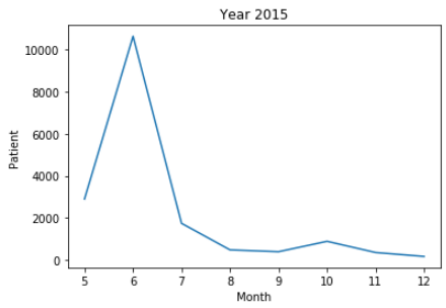
Results

Model Evaluation and Validation

2015 년

-각 달의 기사 내용들 중 '환자' 형태소 빈도수

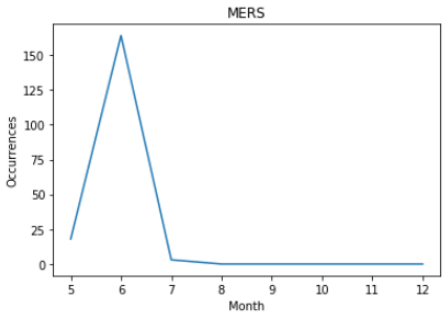
```
plt.title("Year 2015")
plt.plot(noun_list, count_list)
plt.ylabel('Patient')
plt.xlabel("Month")
plt.show()
```



-실제 '메르스' 발생 '환자'에 관한 질병관리본부의 데이터

```
plt.title("MERS")
mers_df = mers_df[mers_df["Month"] >= 5]
plt.ylabel('Occurrences')
plt.xlabel("Month")
plt.plot(mers_df["Month"], mers_df["Occurrences"])
```

```
[<matplotlib.lines.Line2D at 0x2d4b011b5c0>]
```

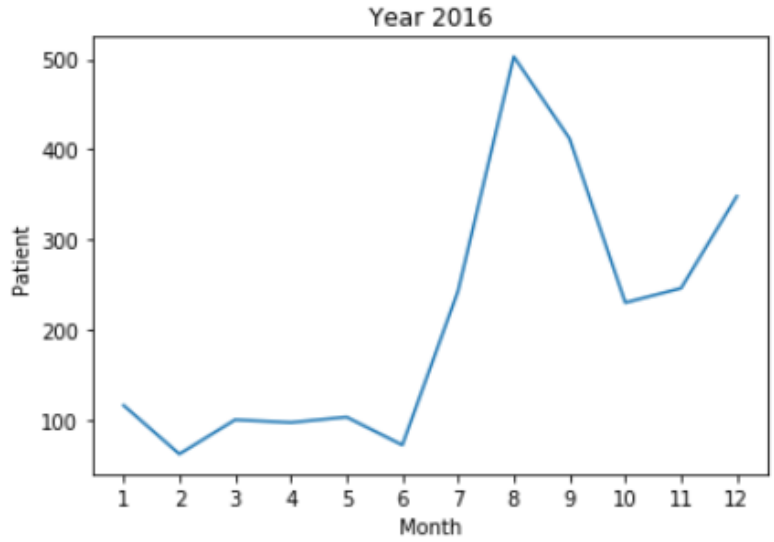


-'환자' 형태소 빈도수와 '메르스'환자수의 상관관계

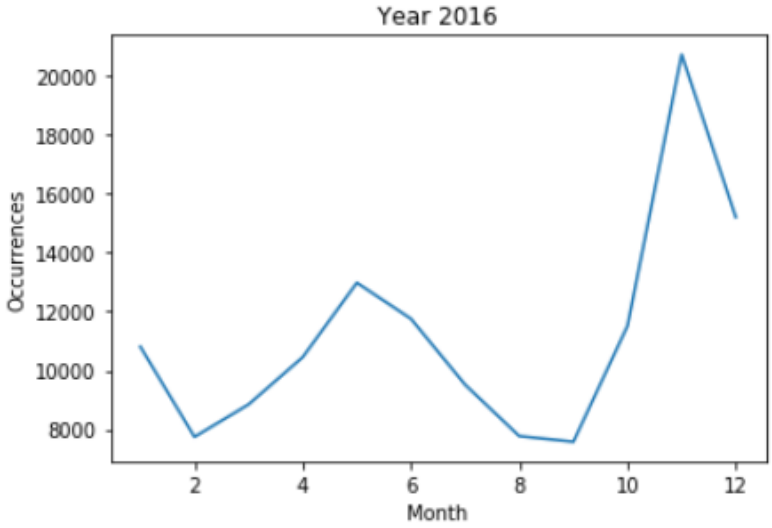
	Month	Patient	Occurrences
Month	1.000000	-0.603977	-0.486350
Patient	-0.603977	1.000000	0.985725
Occurrences	-0.486350	0.985725	1.000000

2016 년

-각 달의 기사 내용들 중 '환자' 형태소 빈도수



-실제 모든 전염병 발생 '환자'에 관한 질병관리본부의 데이터

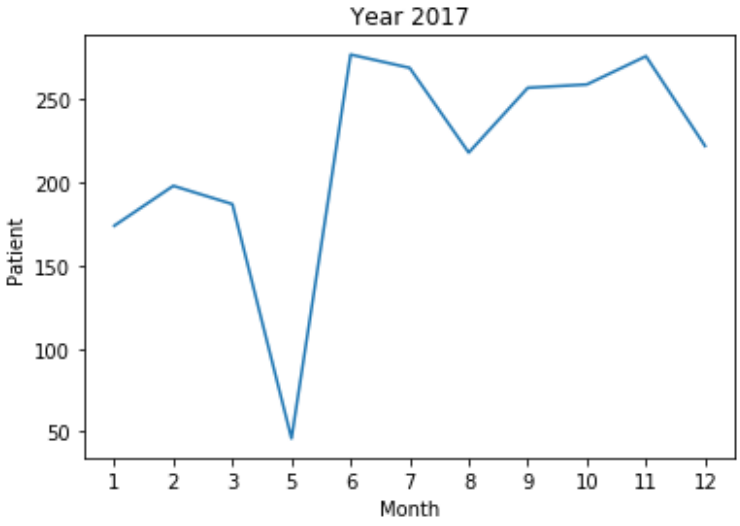


-'환자' 형태소 빈도수와 모든 전염병 환자수의 상관관계

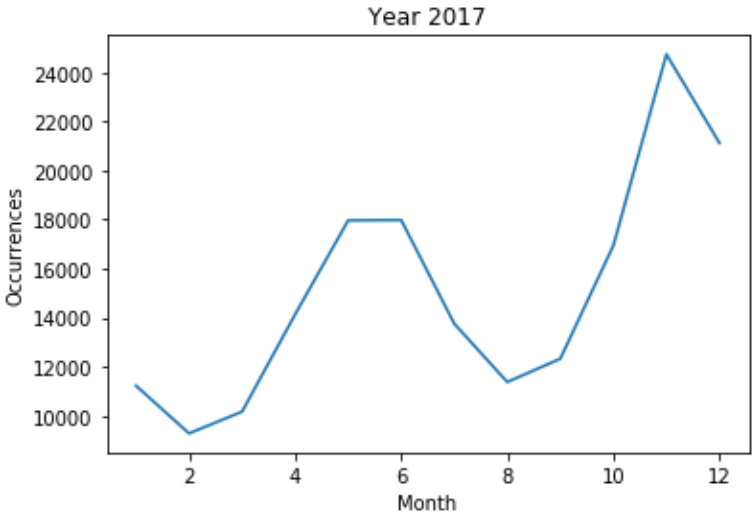
	Month	Patient	Occurrences
Month	1.000000	0.696169	0.507285
Patient	0.696169	1.000000	-0.043851
Occurrences	0.507285	-0.043851	1.000000

2017 년

-각 달의 기사 내용들 중 '환자' 형태소 빈도수



-실제 모든 전염병 발생 '환자'에 관한 질병관리본부의 데이터



-'환자' 형태소 빈도수와 모든 전염병 환자수의 상관관계

	Month	Patient	Occurrences
Month	1.000000	0.490746	0.703117
Patient	0.490746	1.000000	0.179343
Occurrences	0.703117	0.179343	1.000000

Justification

- 2015 년 : '환자'라는 형태소가 가장 많이 출현한 6 월달 기사에 실제로도 가장 많은 '메르스'환자가 발생고, 2015 년 내, 기록된 '환자'형태소 빈도수와 발생한 환자의 상관관게는 0.985725 로 매우 강한 양의 상관관계를 보인다.
- 2016 년 : "환자" 형태소가 전체적으로 빈도수가 높지 않았고, 빈도수와 감염자 수 총합에 대한 상관관계분석에서 -0.043851 라는 수치를 보였습니다. 실제 2016 년에는 법정전염병에 의한 국가재난사태가 없었기 때문에 이러한 결과가 나온 것으로 파악합니다.

- 2017 년 : "환자" 형태소의 빈도수가 전제적으로 균등하게 낮은 빈도수를 보였습니다. 빈도수와 감염자 수의 총합에 대한 상관관계분석에서 0.179343 라는 수치를 보여 관계가 약하다고 보여집니다. 2017 년에도 법정전염병에 의한 국가재난사태가 없었기 때문에 이러한 결과가 나온 것으로 파악된다. "환자" 형태소의 빈도수로는 일반적인 전염병을 파악하기에는 부족하지만 법정전염병에 의한 국가재난사태에 대해서는 파악 가능한것으로 보인다.

Conclusion

Reflection

- 2015 년과 2017 년에는 매우 기사 안의 '환자'라는 형태소 빈도수와 실제 특정 전염병에 걸린 환자의 수가 강한 양의 상관관계를 가지는 것으 볼수 있지만, 반면 2016 년에는 오히려 음의 상관관계를 가지는 것을 볼 수 있다.

Improvement

- 실제 데이터와 비교를 할 대상이 단지 '환자'라는 형태소만 있어서, 매우 약한 기준이었다. 좀 더, 명확하고, 힘을 실어줄 수 있는 edge 를 추가해야한다.
- 그리고, 분야 상관없이 모든, 기사의 내용에 관해서 형태소를 분석을 하게 되므로, 그때 당시, 사회적,정치적 이슈가 많이 나오는 시대에는, 편협적으로 '전염병'에 관련된 기사와 형태소가 매우 적은 문제도 있었다.
- 앞으로 더 나아가, 좀 더 보완을 해서 기사 데이터와 실제 데이터를 가지고, 기사 데이터의 어떤 특정 형태소가 실제 데이터와 연관이 있음을 증명할 수 있는 논리를 생각해봐야 할 것 같다.