Capstone Project

Cha Jae Hyuck

Aprill 2, 2019

Data Scientist Nanodegree

Detect precursor phenomenon words for preventing infectious diseases

Definition

Project Overview

The plague has taken the lives of many people, and at the moment, many diseases are potentially spreading. People are infected by a variety of causes, such as air, insects, animals, and water. However, since ordinary people don't know when, where and how epidemics start or disappear, so, they are very ignorant about the epidemic. the most people recognizes large and powerful epidemics only through news, articles and government announcements. Therefore, little known or small infectious diseases remains unnoticed to the public.

In this project, we analyze the frequency of words related illness in the articles and the relationship between the occurrence of actual diseases and find the actual diseases that have occurred and disappear.

Problem Statement

The purpose is to find the relationship between the frequency of disease-related words in the contents of articles and the presence or absence of actual diseases within that range of dates (2015, 2016, 2017). The tasks are as follows:

1. Collect news articles in NAVER. (https://news.naver.com/) (=news articles have been collected since May 20, 2015.)
2. Analyze morphemes of every news article by a year term and calculate the average frequency of how many times the term 'patient' has been used in article . (Why we chose 'patients'? : when we looked at morphological analysis in the sample group extracted from the population, the most frequent morpheme except the pathologic name was high.)
3. After analyzing all of the news articles morphemes by a monthly basis, select the months that have higher frequency of using the term 'patients' in the news compared to the average number.
4. Based on the analyzed morphemes of the selected month, check the occurrence of an epidemic in that day using the actual data from KOSIS(Korean Statistical Information Service).

Analysis

Data Exploration

Collected articles data are all the news articles from various portals that have been collected since 2015. Using Python library web crawler called 'Scrapy', we define the fields of the data to be collected(item.py)and write code to get the

information (spider.py). The crawled web article data was stored in a personal DB. Data is about 500,000 total, and we use article content and collection time in this data. The collection article data has the following fields:

*id : Collected article Index (Integer)

*aid: article identified number(Integer)

*title : article title(String)

*content : article content(String)

*aDate : collected time (TimeStamp)

*nUrl : article url from NAVER (String)

*pUrl : article url from original NEWS site (String)

*nClass : article section (String)

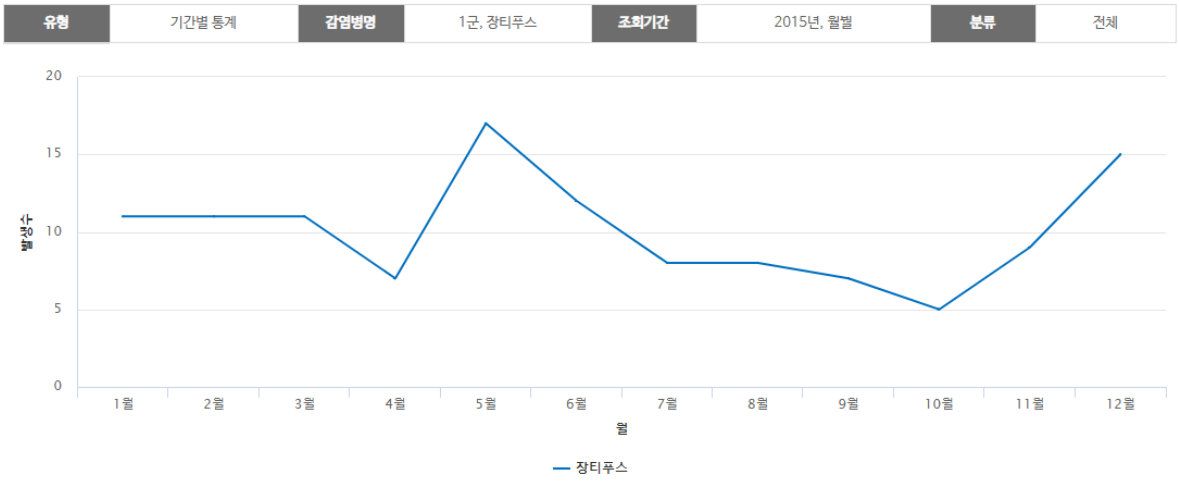
*press : article company (String)

*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 팔아 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

This graph is the monthly patients number of typhoid fever in a certain year. Similary, the number of incidents for each year /month of Group 1to 4 infectious diseases can be confirmed. These data are uses as a basis for comparing the correspondence between the data analyzed by us and the actual data.



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

<source> Infection portals : http://www.cdc.go.kr

Methodology

Data Preprocessing

Article collection data is supplied and has 100% accuracy. So, no preprocessing is required.

Implementation

-Analyze all the news articles on a yearly basis and calculate the average by calculating the frequency of the term 'patient'. (konlpy 의 대한 설명, knolpy/counter 를 사용해 형태소 구하고, 평균을 구하는 것에 대한 중요코드에 대한 설명 필요)

```
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

- select the months that have higher frequency of using the term 'patients' and show a list of morphemes for that month.

```
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 : The average of the frequencyofthe 'patient' morphemes per month → calculated as " the number of 'patient' morphemes / day *30 " because of the data from May 20 to December 31 for 2015.

select_month : The frequency of 'patient' morphemes is higher than average

```
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
```

- Based on the analyzed morphemes of the selected month, check the occurrence of an epidemic in that day using the actual data from KOSIS.

```
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() : A function that returns the frequency of the 'patient' morpheme as a list using konlpy's Okt and extracts nouns with nouns() and figures out frequency with Counter object.

@param1 : Text to analyze morpheme

@param2 : The number of nouns to be extracted with high frequency

get_tags() : A function that returns a list of n morphemes with a high frequency and prints them.

@param1 : Text to analyze morpheme

@param2 : The number of nouns to be extracted with high frequency

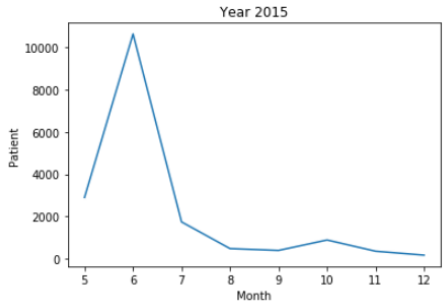
Results

Model Evaluation and Validation

<2015>

- The frequency of 'patient' morpheme among articles of each month

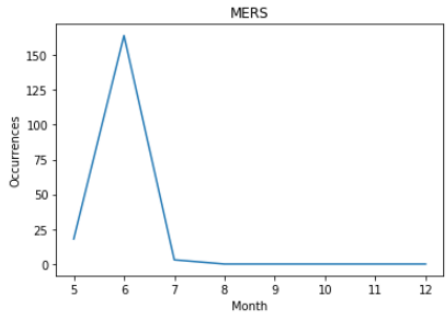
```
plt.title("Year 2015")
plt.plot(noun_list, count_list)
plt.ylabel('Patient')
plt.xlabel("Month")
plt.show()
```



- Real data on patients that actually occurred 'MERS'

```
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>]
```

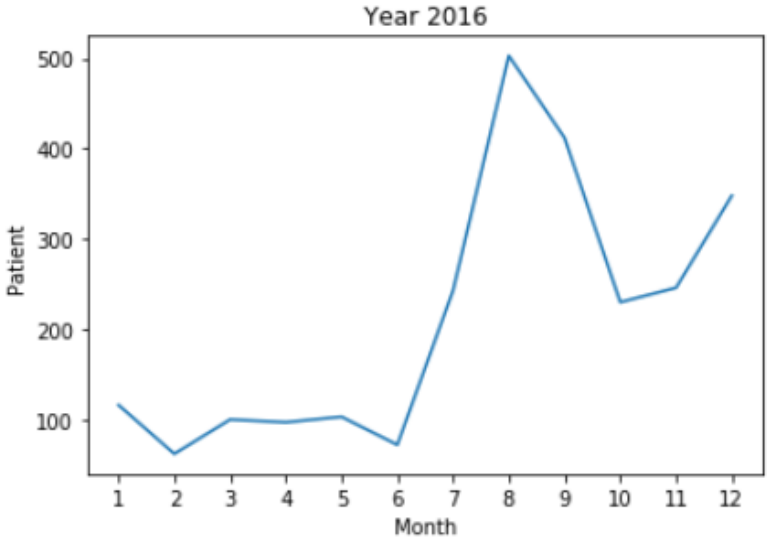


-Correlation between 'patient' morphological frequency and 'MERS' patient number

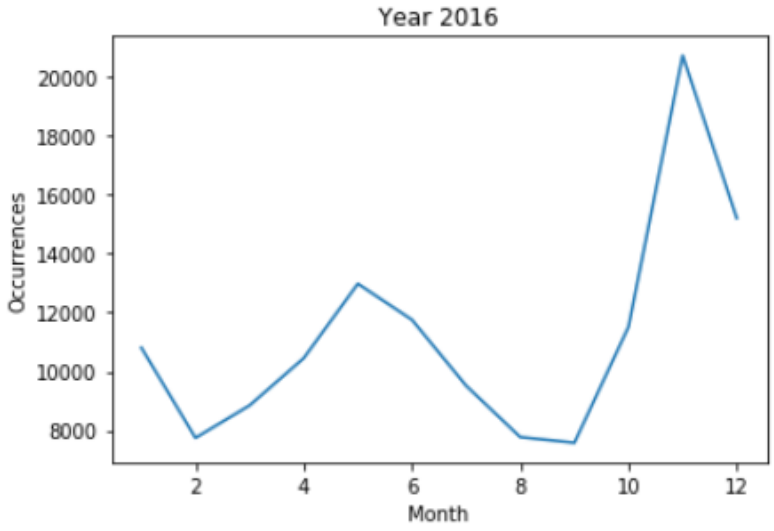
	Month	Patient	Occurrences
Month	1.000000	-0.603977	-0.486350
Patient	-0.603977	1.000000	0.985725
Occurrences	-0.486350	0.985725	1.000000

<2016>

- The frequency of 'patient' morpheme among articles of each month



- Real data on patients that actually occurred all infectious disease



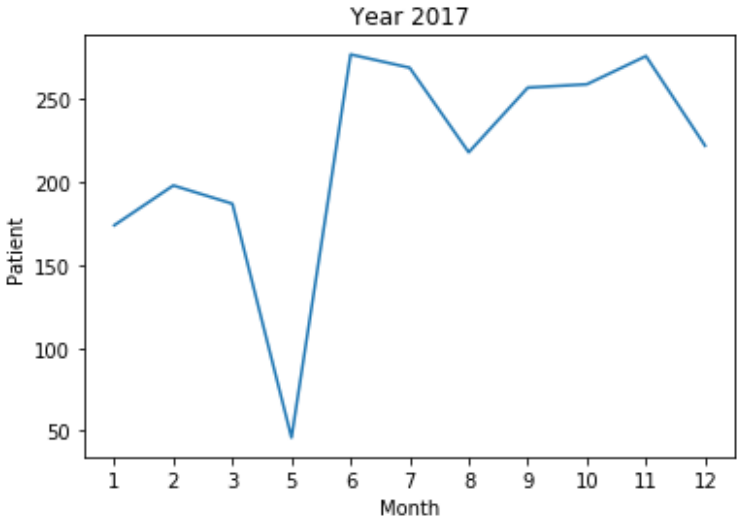
-

-Correlation between 'patient' morphological frequency and all epidemic patient number

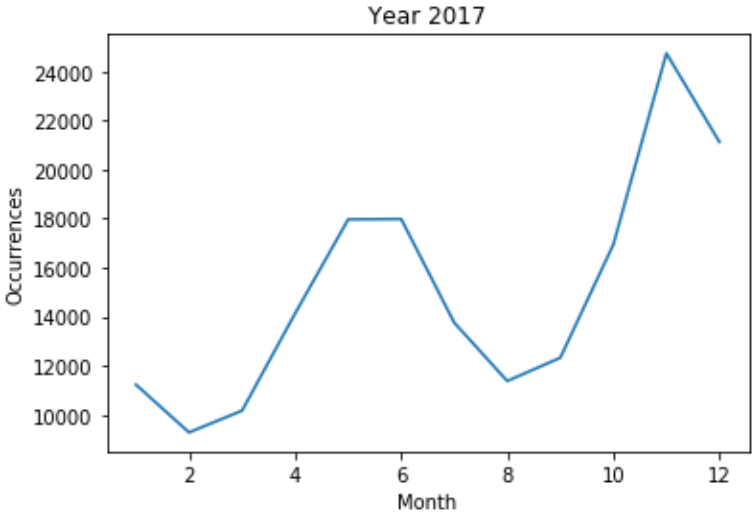
	Month	Patient	Occurrences
Month	1.000000	0.696169	0.507285
Patient	0.696169	1.000000	-0.043851
Occurrences	0.507285	-0.043851	1.000000

<2017>

- The frequency of 'patient' morpheme among articles of each month



- Real data on patients that actually occurred all infectious disease



-Correlation between 'patient' morphological frequency and all epidemic patient number

	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 year: The news article written in June showed the most abundant usage of the morpheme ‘patient’. And it was actually the same month as when there were highest number of MERS patients among the globe. In addition,

  In 2015, the usage frequency of the morpheme ‘patient’ did have a very strong positive correlation to the number of patients that actually suffered from disease, which was up to 0.985725.

- 2016 year : The 'patient’ morpheme did not have a high frequency as a whole, and the correlation between the frequency and the total number of infected persons was -0.043851.

  In fact, in 2016, there was no national disaster caused by a pandemic..

- 2017 year : The frequencies of the "patient" morphemes were equally low evenly. The correlation between frequency and number of infections is 0.179343, which is considered to be weak.

In 2017, there is no national disaster caused by a contagious disease. Although the frequency of "patient" morphemes is not sufficient to identify common infectious diseases, it seems possible to understand the state of calamity caused by a pandemic.

Conclusion

Reflection

- In 2015 and 2017, there is a strong positive correlation between the frequency of morpheme "patient" in the article and the actual number of infectious diseases, whereas in 2016 it is seen as having a negative correlation.

Improvement

- It was a very weak criterion because there was only a 'patient' morpheme to compare with actual data. We need to add more and clear and empowering edges.
- And, since we analyze the morpheme with only contents of all articles regardless of field. Therefore, at that time, when there were many social and political issues, there were very few articles and morphemes related to 'infectious diseases'.
- In the future, I would like to reconsider the logic of proving that particular morphemes of article and another edges are associated with the actual data.