It’s PRL, Dude!

20171366 강범규

PRL ?

- PRL : *Physical Review Letters*
- APS (American Physical Society)에서 1958년부터 출간중인 저널
- IF : 9.185 (2021년 기준)
- 물리 분야에서 높은 인지도를 보유
- Physical Review Letters (https://journals.aps.org/prl/)

About “It’s PRL, dude” …

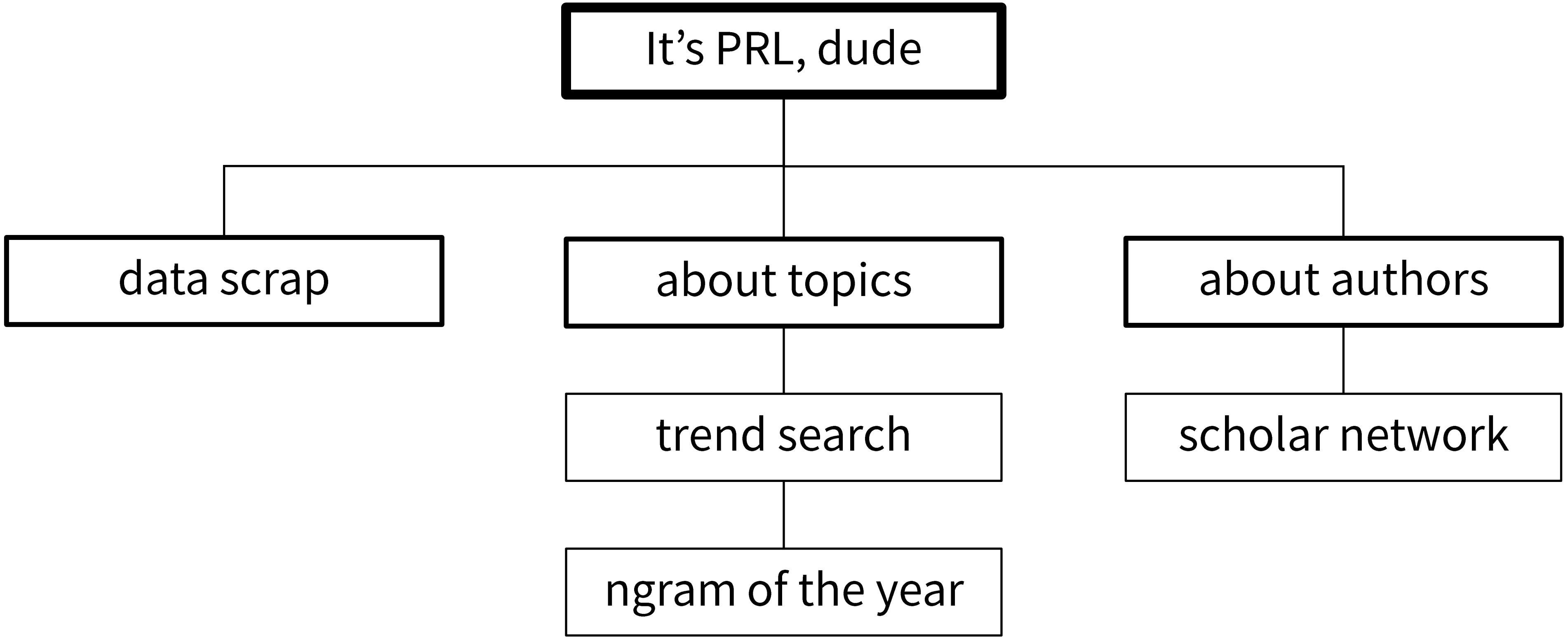
“대학원 생활에 필요한 Tool이 있다면?”

“연구 동향을 쉽게 확인할 수 있을까?”

“우리 교수님은 얼마나 **인싸* 이실까?”

>> “It’s PRL, dude !” <<

About “It’s PRL, dude” …


It’s PRL, dude

data scrap

about topics

about authors

trend search

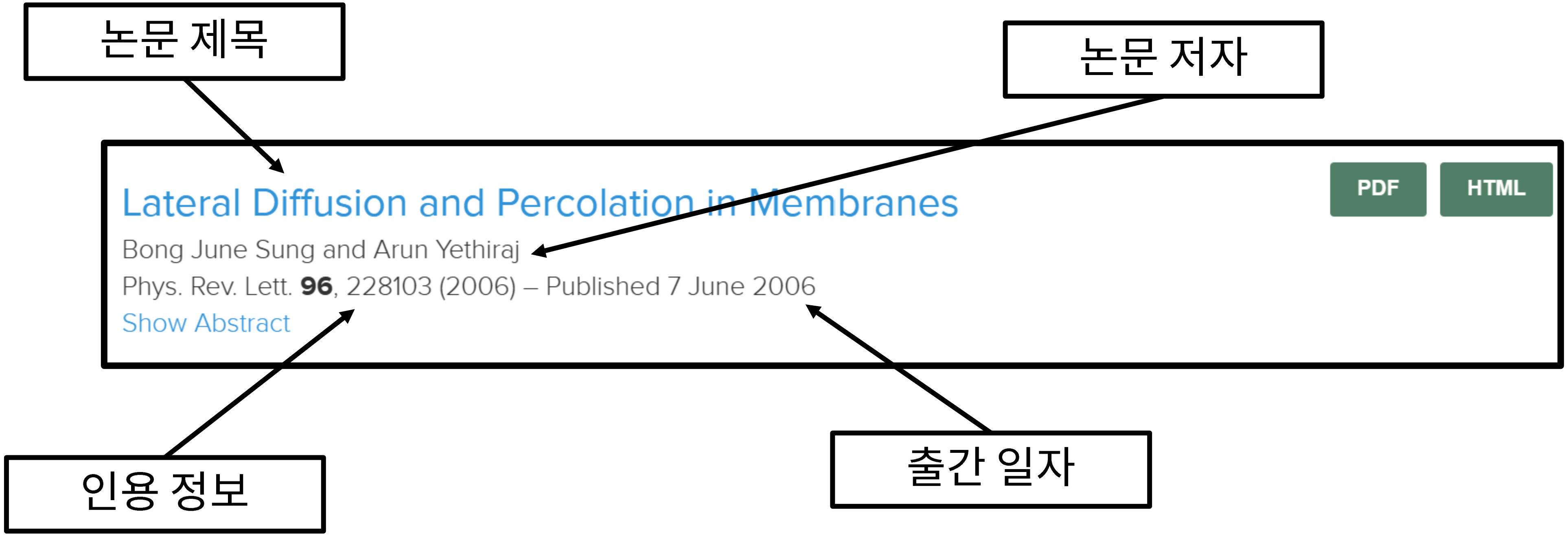
scholar network

ngram of the year

Data Scrap

- Physical Review Letters (https://journals.aps.org/prl/issues)

  : 현재까지 출간된 모든 논문의 데이터 보유



논문 제목

논문 저자

Lateral Diffusion and Percolation in Membranes

Bong June Sung and Arun Yethiraj

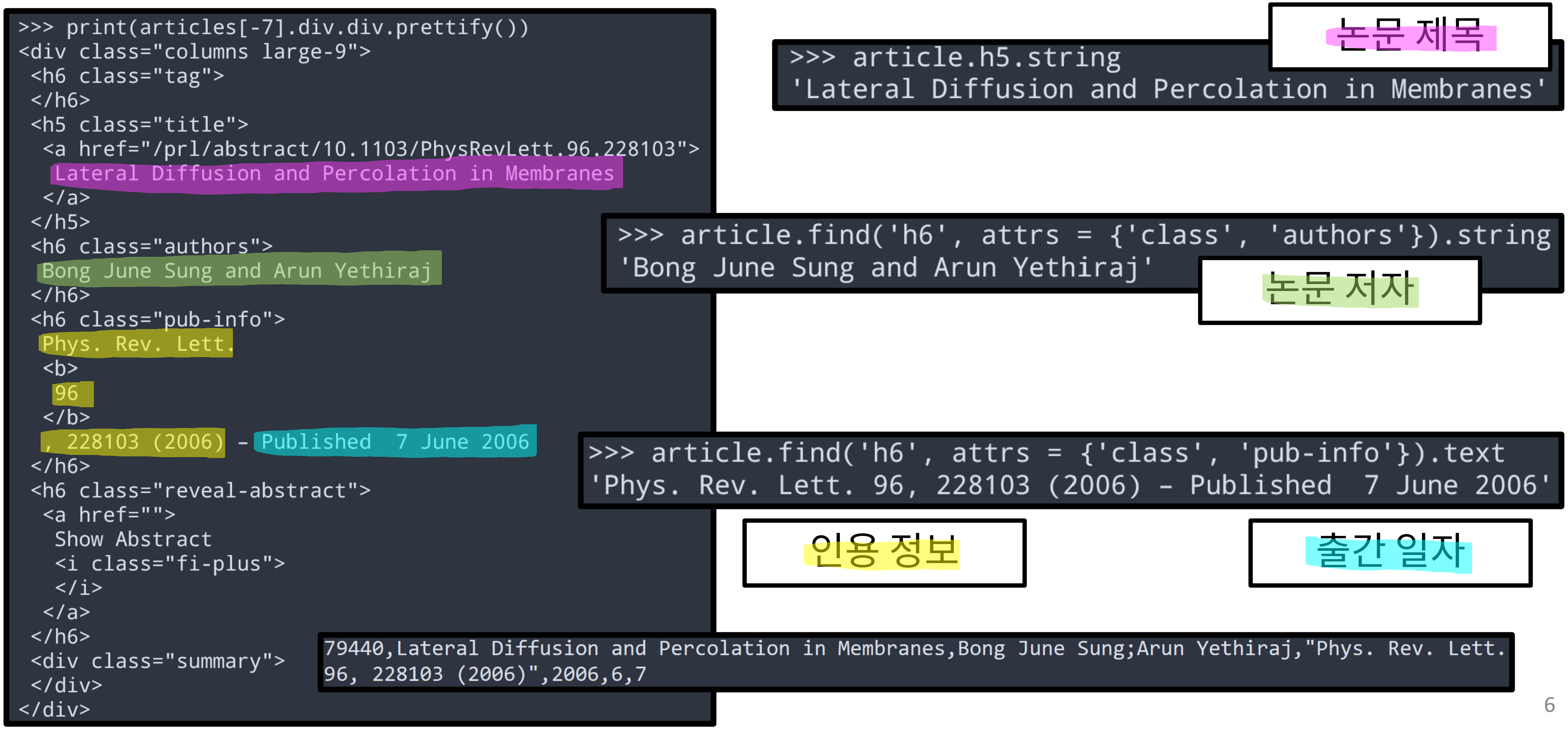
Phys. Rev. Lett. **96**, 228103 (2006) – Published 7 June 2006

Show Abstract

인용 정보

출간 일자

Data Scrap



```
>>> print(articles[-7].div.div.prettify())
<div class="columns large-9">
 <h6 class="tag">
 </h6>
 <h5 class="title">
  <a href="/prl/abstract/10.1103/PhysRevLett.96.228103">
   Lateral Diffusion and Percolation in Membranes
  </a>
 </h5>
 <h6 class="authors">
  Bong June Sung and Arun Yethiraj
 </h6>
 <h6 class="pub-info">
  Phys. Rev. Lett.
  <b>
   96
  </b>
  , 228103 (2006) - Published  7 June 2006
 </h6>
 <h6 class="reveal-abstract">
  <a href="">
   Show Abstract
   <i class="fi-plus">
   </i>
  </a>
 </h6>
 <div class="summary">
 </div>
</div>
```

논문 제목

```
>>> article.h5.string
'Lateral Diffusion and Percolation in Membranes'
```

```
>>> article.find('h6', attrs = {'class', 'authors'}).string
'Bong June Sung and Arun Yethiraj'
```

논문 저자

```
>>> article.find('h6', attrs = {'class', 'pub-info'}).text
'Phys. Rev. Lett. 96, 228103 (2006) - Published  7 June 2006'
```

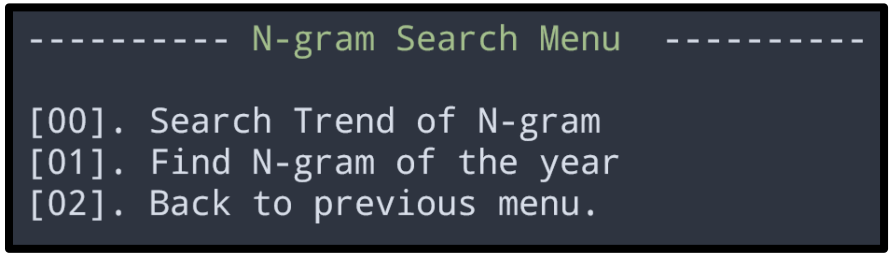
인용 정보

출간 일자

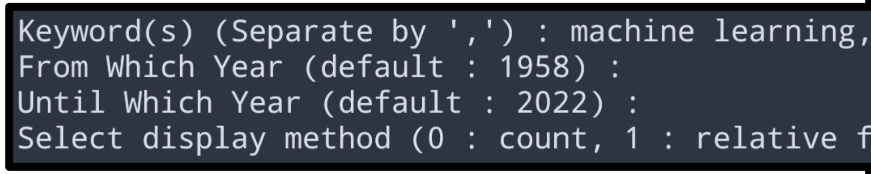
```
79440,Lateral Diffusion and Percolation in Membranes,Bong June Sung;Arun Yethiraj,"Phys. Rev. Lett.
96, 228103 (2006)",2006,6,7
```

Trend Search

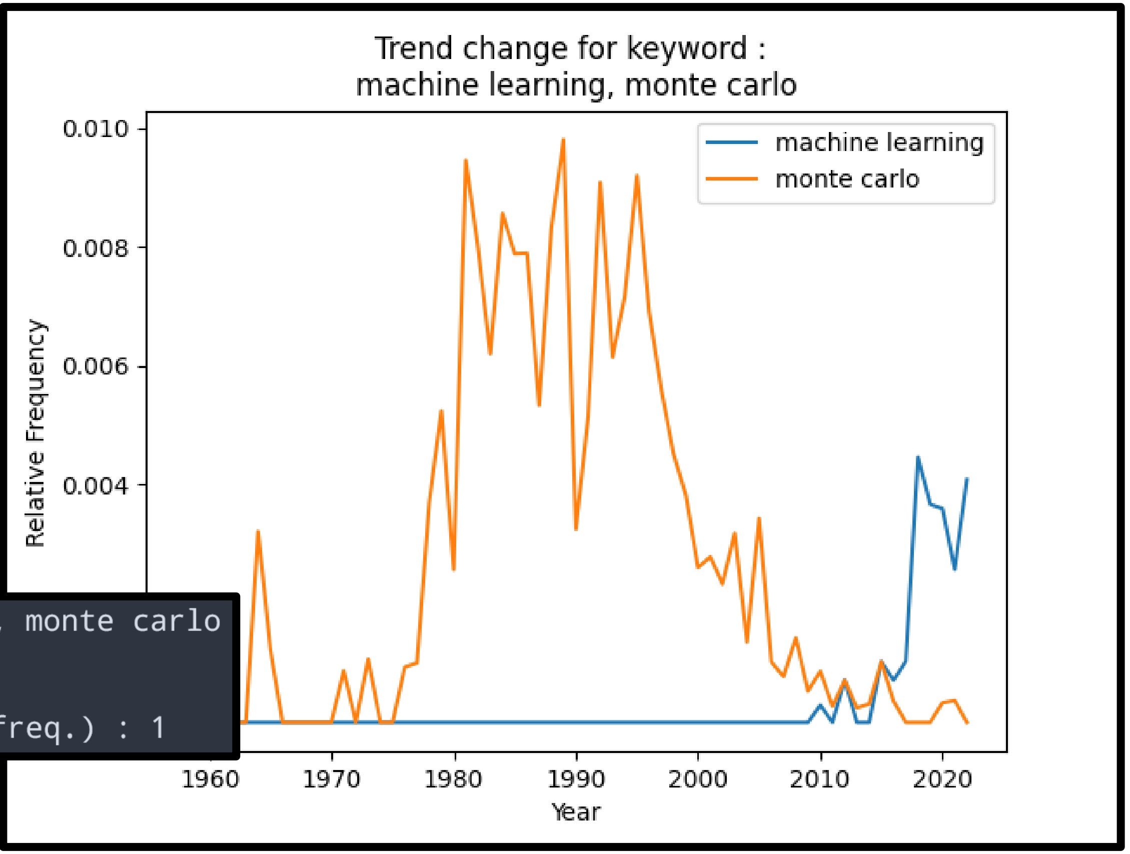
- 검색어의 빈도 trend를 시각화



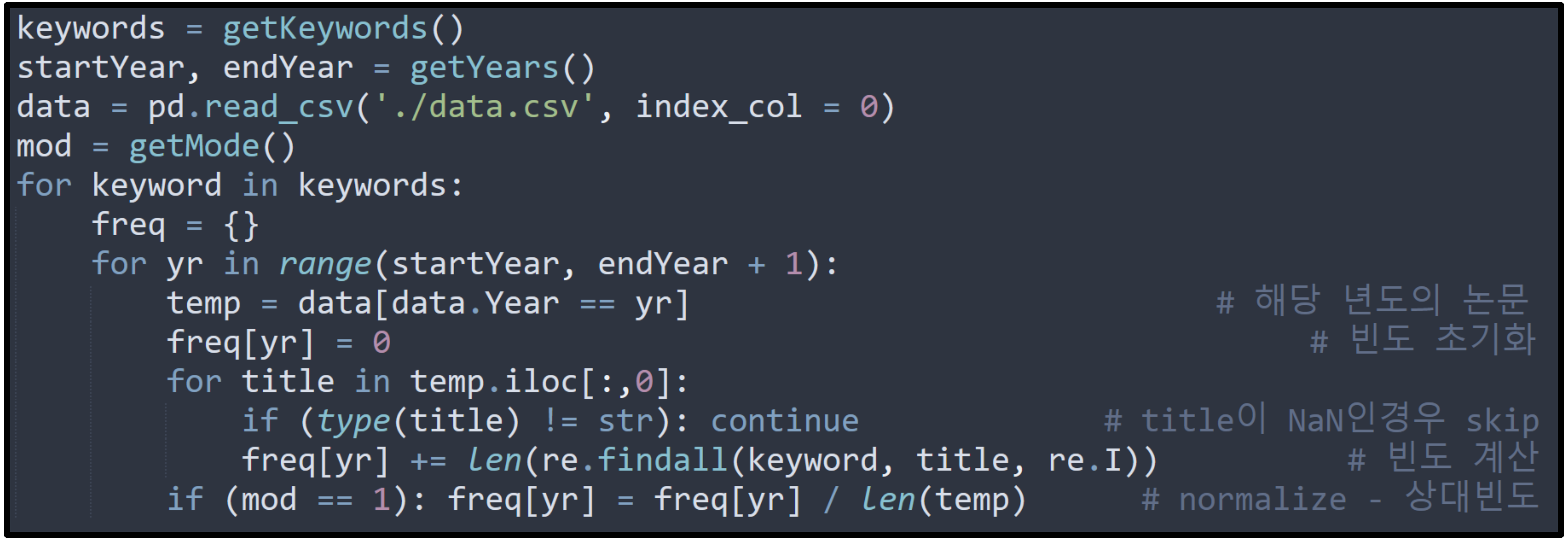
```
---------- N-gram Search Menu  ----------

[00]. Search Trend of N-gram
[01]. Find N-gram of the year
[02]. Back to previous menu.
```



```
Keyword(s) (Separate by ',') : machine learning, monte carlo
From Which Year (default : 1958) :
Until Which Year (default : 2022) :
Select display method (0 : count, 1 : relative freq.) : 1
```



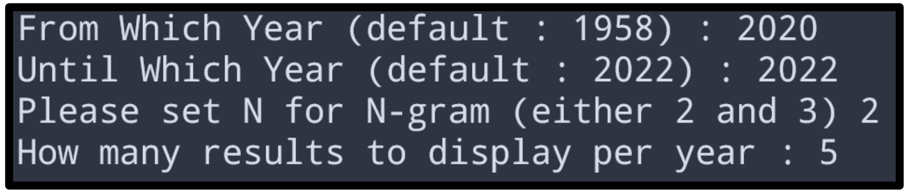
Trend Search



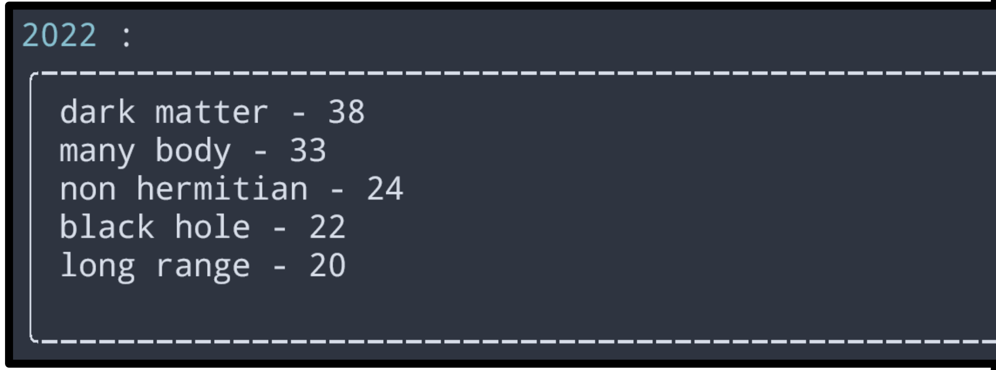
```
keywords = getKeywords()
startYear, endYear = getYears()
data = pd.read_csv('./data.csv', index_col = 0)
mod = getMode()
for keyword in keywords:
    freq = {}
    for yr in range(startYear, endYear + 1):
        temp = data[data.Year == yr]                            # 해당 년도의 논문
        freq[yr] = 0                                                 # 빈도 초기화
        for title in temp.iloc[:,0]:
            if (type(title) != str): continue                 # title이 NaN인경우 skip
            freq[yr] += len(re.findall(keyword, title, re.I))          # 빈도 계산
        if (mod == 1): freq[yr] = freq[yr] / len(temp)      # normalize - 상대빈도
```

Ngram of the Year

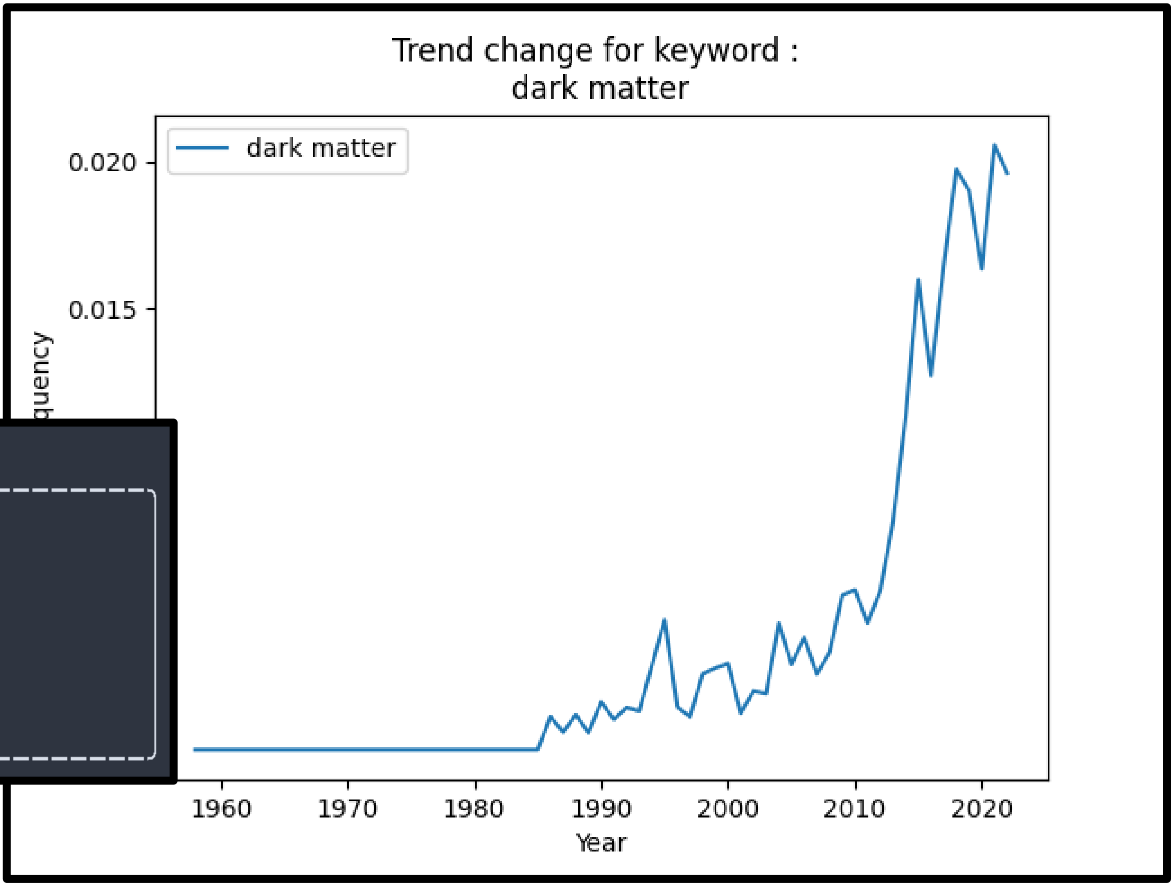
- 연도별 최고빈도 ngram 표시



```
From Which Year (default : 1958) : 2020
Until Which Year (default : 2022) : 2022
Please set N for N-gram (either 2 and 3) 2
How many results to display per year : 5
```

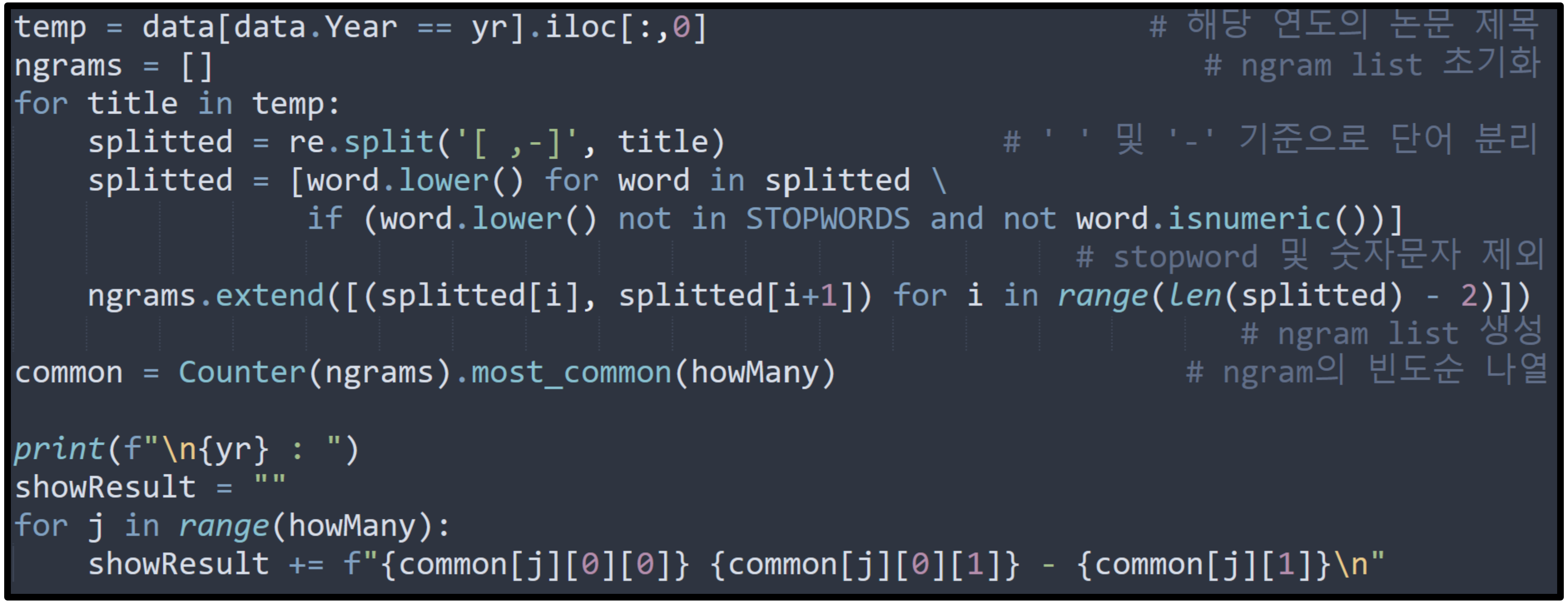


```
2022 :
 ------------------------------------------------------------
  dark matter - 38
  many body - 33
  non hermitian - 24
  black hole - 22
  long range - 20

 ------------------------------------------------------------
```



Trend change for keyword :
dark matter

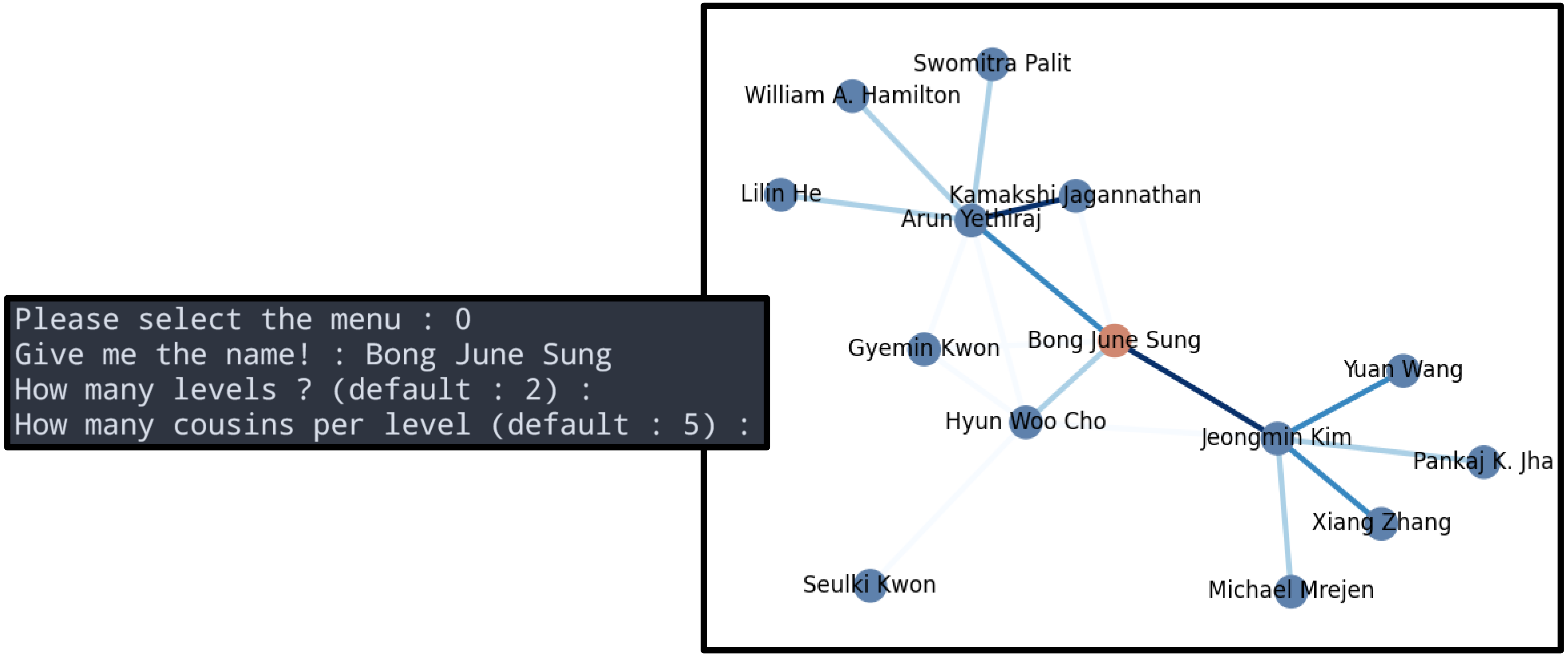
Ngram of the Year



```python
temp = data[data.Year == yr].iloc[:,0]                        # 해당 연도의 논문 제목
ngrams = []                                                      # ngram list 초기화
for title in temp:
    splitted = re.split('[ ,-]', title)               # ' ' 및 '-' 기준으로 단어 분리
    splitted = [word.lower() for word in splitted \
                if (word.lower() not in STOPWORDS and not word.isnumeric())]
                                                          # stopword 및 숫자문자 제외
    ngrams.extend([(splitted[i], splitted[i+1]) for i in range(len(splitted) - 2)])
                                                                   # ngram list 생성
common = Counter(ngrams).most_common(howMany)                   # ngram의 빈도순 나열

print(f"\n{yr} : ")
showResult = ""
for j in range(howMany):
    showResult += f"{common[j][0][0]} {common[j][0][1]} - {common[j][1]}\n"
```

Scholar Network

- Find your academic cousin !



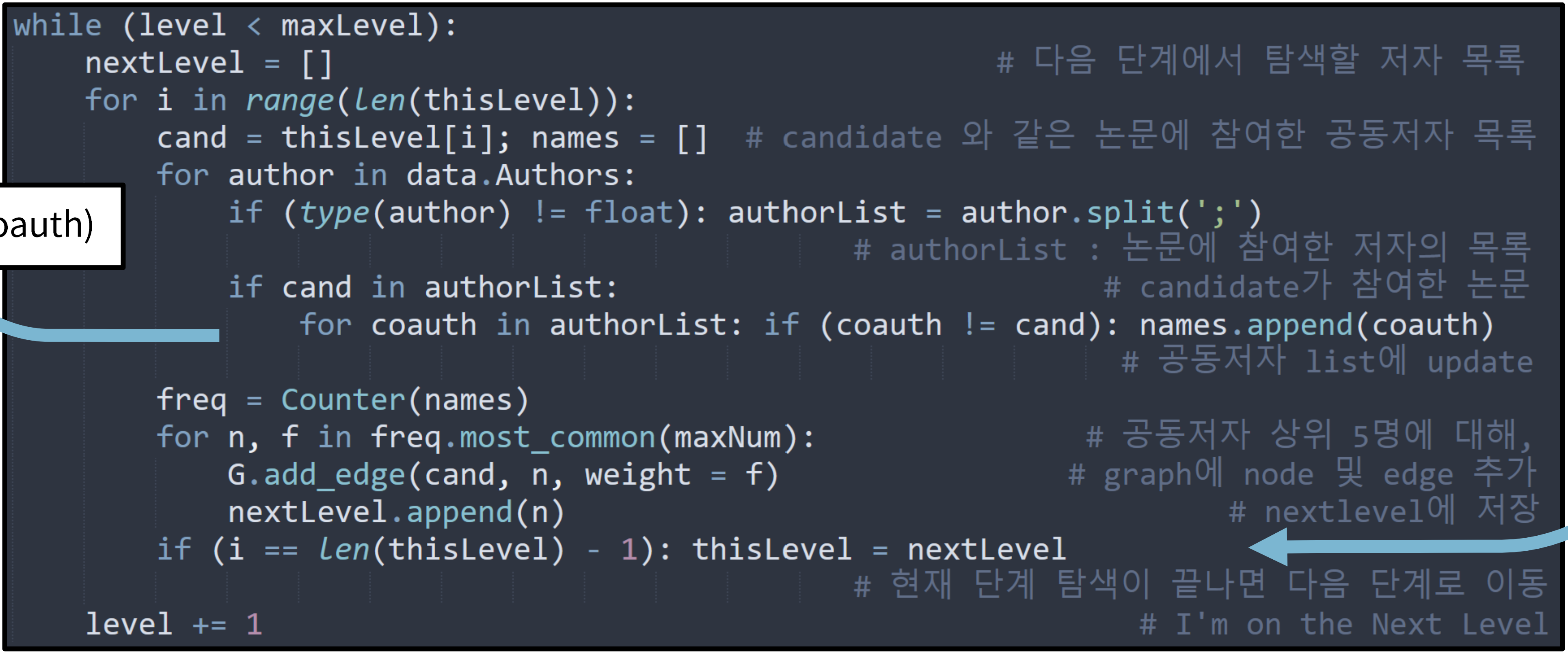
```
Please select the menu : 0
Give me the name! : Bong June Sung
How many levels ? (default : 2) :
How many cousins per level (default : 5) :
```

Scholar Network

nextlevel

[Karina, Winter, Giselle, Ningning]

.append(coauth)



```
while (level < maxLevel):
    nextLevel = []                                           # 다음 단계에서 탐색할 저자 목록
    for i in range(len(thisLevel)):
        cand = thisLevel[i]; names = []  # candidate 와 같은 논문에 참여한 공동저자 목록
        for author in data.Authors:
            if (type(author) != float): authorList = author.split(';')
                                                  # authorList : 논문에 참여한 저자의 목록
            if cand in authorList:                             # candidate가 참여한 논문
                for coauth in authorList: if (coauth != cand): names.append(coauth)
                                                                   # 공동저자 list에 update
        freq = Counter(names)
        for n, f in freq.most_common(maxNum):                  # 공동저자 상위 5명에 대해,
            G.add_edge(cand, n, weight = f)                   # graph에 node 및 edge 추가
            nextLevel.append(n)                                        # nextlevel에 저장
        if (i == len(thisLevel) - 1): thisLevel = nextLevel
                                              # 현재 단계 탐색이 끝나면 다음 단계로 이동
    level += 1                                                    # I'm on the Next Level
```

Thanks for Listening !

https://github.com/SinsuSquid/It-s-PRL-dude