User input keywords popularity analysis

Have a look at the demonstration website. In this app, we allow users to input their keywords. Users can get the line chart, and the frequency of keywords mentioned in the news dataset.

Users can input multiple keywords separated by space; and specify “and” or “or” condition. We can count the news containing all of the keywords with “and” condition; and count the news containing at least one of the keywords with “or” condition.

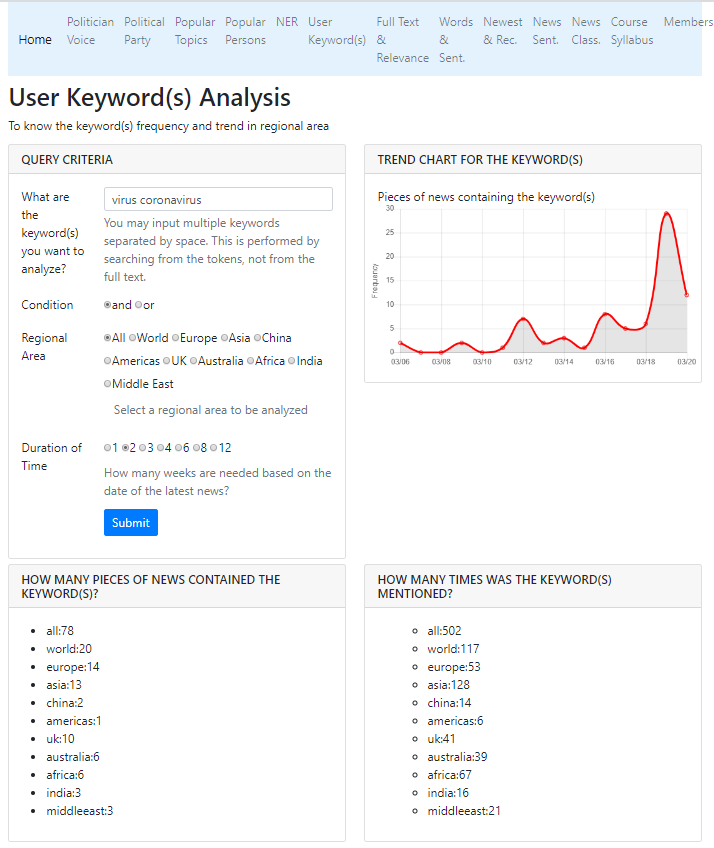
Users should specify the duration of time. Our system searches keywords from the limited pieces of news within the specified recent weeks.

Our system can search the keywords from either “tokens\_v2” or “content” column. In this app, we search from the “tokens\_v2” column, not from the full text “content” column. However, you can do it by either way.

How can we find Chinese or English keywords from a news text? We can use “in” or panda’s “contains()” function.

In this app, we need to show the trend chart for the keyword frequency. In this app counting time-based frequency of keywords is a difficult task.

Let’s get started!



<http://163.18.22.32:8000/userkeyword/> 

Step 0: Create a project by duplicating previous project

We will continue from our previous project.

I suggest you rename your project folder with a meaningful name.

Step1: Open the project with your favorite Editor

Open our project using VS Code.

Step 2: Create “app\_user\_keyword” app

Step 2: Create a new APP named “app\_user\_keyword” in your website root.

python manage.py startapp app\_user\_keyword

Step3: Copy data file to the dataset folder

We will use the preprocessed data file, so copy it into the dataset folder.

Step 4: settings.py

|  |  |
| --- | --- |
| |  | | --- | | ALLOWED\_HOSTS = ['localhost', '127.0.0.1']  INSTALLED\_APPS = [  …..  ' app\_top\_keyword',  ' app\_top\_person',  'app\_top\_ner\_analysis',  'app\_user\_keyword',  ]  TEMPLATES = [  {  'BACKEND': 'django.template.backends.django.DjangoTemplates',  'DIRS': [os.path.join(BASE\_DIR, 'templates')],  'APP\_DIRS': True,  'OPTIONS': …..  ……  ……  },  ] | |

Step 5: website\_configs/urls.py

website\_configs/urls.py

|  |  |
| --- | --- |
| |  | | --- | | from django.urls import path  from django.urls import include  urlpatterns = [  # top keywords  path('topword/', include('app\_top\_keyword.urls')),  # top persons  path('topperson/', include('app\_top\_person.urls')),  # top name entity keyword  path('topner/', include('app\_top\_ner.urls')),  # user keyword analysis  path('userkeyword/', include('app\_user\_keyword.urls')),  ] | |

Step 6: app\_user\_keyword/urls.py

In folder app\_user\_keyword, create a Python file named “urls.py”.

app\_user\_keyword/urls.py

|  |  |
| --- | --- |
| |  | | --- | | from django.urls import path  from . import views  app\_name="app\_user\_keyword"  urlpatterns = [       path('', views.home, name='home'),     path('api\_get\_top\_userkey/', views.api\_get\_top\_userkey),  ]  '''  # the first way:  The url path on the browser will be  http://localhost:8000/userkeyword/  # the second way:  The url path on the browser will be  http://localhost:8000/userkeyword/top\_userkey/  The ajax url is as the following:  $.ajax({      type: "POST",      url: "api\_get\_top\_userkey/",  ''' | |

Step 7: views.py

What are the differences between CNN and CNA news?

(1) The data csv file name is different.

(2) The name of news category is different.

news\_categories = ['all', 'world', 'europe', 'asia', 'china', 'americas', 'uk', 'australia', 'africa','india','middleeast']

news\_categories = ['政治', '科技', '運動', '證卷', '產經', '娛樂', '生活', '國際', '社會', '文化', '兩岸', '全部']

(3) the category “all” in the filter\_dataFrame() should be changed to “全部”.

(4) the category “all” in the count\_keyword() should be changed to “全部”.

views.py

|  |
| --- |
| from django.shortcuts import render  import pandas as pd  from django.http import JsonResponse  from django.views.decorators.csrf import csrf\_exempt  from datetime import datetime, timedelta  import re  # (1) we can load data using read\_csv()  # df is a global variable  # df = pd.read\_csv('dataset/cna\_news\_preprocessed.csv', sep='|')  # (2) we can load data using reload\_df\_data() function  def reload\_df\_data():  # make df be a global variable  global df  df = pd.read\_csv('app\_user\_keyword/dataset/cna\_news\_preprocessed.csv', sep='|')  # We should reload df when necessary  reload\_df\_data()  def home(request):  return render(request, 'app\_user\_keyword/home.html')  # When POST is used, make this function be exempted from the csrf  @csrf\_exempt  def api\_get\_top\_userkey(request):  # (1) get keywords, category, condition, and weeks passed from frontend  userkey = request.POST['userkey']  cate = request.POST['cate']  cond = request.POST['cond']  weeks = int(request.POST['weeks'])  key = userkey.split()    # (2) make df\_query global, so it can be used by other functions  global df\_query  # (3) filter dataframe  df\_query = filter\_dataFrame(key, cond, cate,weeks)  #print(len(df\_query))  # (4) get frequency data  key\_freq\_cat, key\_occurrence\_cat = count\_keyword(df\_query, key)  print(key\_occurrence\_cat)    # (5) get line chart data  # key\_time\_freq = [  # '{"x": "2019-03-07", "y": 2}',  # '{"x": "2019-03-08", "y": 2}',  # '{"x": "2019-03-09", "y": 13}']  key\_time\_freq = get\_keyword\_time\_based\_freq(df\_query)  # (6) response all data to frontend home page  response = {  'key\_occurrence\_cat': key\_occurrence\_cat,  'key\_freq\_cat': key\_freq\_cat,  'key\_time\_freq': key\_time\_freq, }  return JsonResponse(response)  # Searching keywords from "content" column  # Here this function uses df.content column, while filter\_dataFrame() uses df.tokens\_v2  def filter\_dataFrame(user\_keywords, cond, cate, weeks):  # end date: the date of the latest record of news  end\_date = df.date.max()    # start date  start\_date = (datetime.strptime(end\_date, '%Y-%m-%d').date() - timedelta(weeks=weeks)).strftime('%Y-%m-%d')  # (1) proceed filtering: a duration of a period of time  # 期間條件  period\_condition = (df.date >= start\_date) & (df.date <= end\_date)    # (2) proceed filtering: news category  # 新聞類別條件  if (cate == "全部"):  condition = period\_condition # "全部"類別不必過濾新聞種類  else:  # category新聞類別條件  condition = period\_condition & (df.category == cate)  # (3) proceed filtering: keywords  # and or 條件  if (cond == 'and'):  # query keywords condition使用者輸入關鍵字條件and  condition = condition & df.content.apply(lambda text: all((qk in text) for qk in user\_keywords)) #寫法:all()  elif (cond == 'or'):  # query keywords condition使用者輸入關鍵字條件  condition = condition & df.content.apply(lambda text: any((qk in text) for qk in user\_keywords)) #寫法:any()  # condiction is a list of True or False boolean value  df\_query = df[condition]  return df\_query  # \*\* How many pieces of news were the keyword(s) mentioned in?  # \*\* How many times were the keyword(s) mentioned?  # For the query\_df, count the occurence and frequency for each category.  # (1) cate\_occurence={} 被多少篇新聞報導 How many pieces of news contain the keywords.  # (2) cate\_freq={} 被提到多少次? How many times are the keywords mentioned  news\_categories = ['政治', '科技', '運動', '證卷', '產經', '娛樂', '生活', '國際', '社會', '文化', '兩岸', '全部']  def count\_keyword(df\_query, query\_keywords):  cate\_occurrence = {}  cate\_freq = {}    # 字典初始化  for cate in news\_categories:  cate\_occurrence[cate] = 0 # {'政治':0, '科技':0}  cate\_freq[cate] = 0    for idx, row in df\_query.iterrows():  # count the number of articles各類別篇數統計  cate\_occurrence[row.category] += 1 # {'政治':+1, '科技':0}  cate\_occurrence['全部'] += 1    # count the keyword frequency各類別次數統計  # 計算這一篇文章的content中重複含有多少個這些關鍵字(頻率)  freq = sum([ len(re.findall(keyword, row.content, re.I)) for keyword in query\_keywords] )  cate\_freq[row.category] += freq # 在該新聞類別中累計頻率  cate\_freq['全部'] += freq # 在"全部"類別中累計頻率  return cate\_freq, cate\_occurrence  def get\_keyword\_time\_based\_freq(df\_query):  date\_samples = df\_query.date  query\_freq = pd.DataFrame({'date\_index': pd.to\_datetime(date\_samples), 'freq': [1 for \_ in range(len(df\_query))]})  data = query\_freq.groupby(pd.Grouper(key='date\_index', freq='D')).sum()  time\_data = []  for i, idx in enumerate(data.index):  row = {'x': idx.strftime('%Y-%m-%d'), 'y': int(data.iloc[i].freq)}  time\_data.append(row)  return time\_data  print("app\_user\_keyword was loaded!") |

Step 8: home.html (Template)

In the app folder, create a HTML file named “home.html”

app\_user\_keyword/templates/app\_user\_keyword/home.html

home.html

|  |
| --- |
| {% extends 'base.html' %} {% block title %}  使用者關鍵詞查詢  {% endblock %} {% block content %}  <div class="col-lg-12">  <h1>分析你關心的關鍵詞</h1>  <p>可以針對你輸入的個別關鍵詞進行熱門程度分析</p>  </div>  <div class="col-lg-6 mb-2">  <!-- 輸入條件區塊開始 -->  <div class="card">  <div class="card-header">  <h3 class="h6 text-uppercase mb-0">輸入條件</h3>  </div>  <div class="card-body">  <div class="mb-3 row">  <label class="col-md-3 col-form-label">關心哪個關鍵詞?</label>  <div class="col-md-9">  <input id="input\_keyword" name="userkey" value="烏克蘭 俄羅斯" class="form-control" />  <div class="form-text text-muted">全文搜尋關鍵字，可輸入多個，空白隔開。任何人名、產品、片語、特定詞皆可。</div>  </div>  </div>  <div class="mb-3 row">  <label class="col-sm-3 col-form-label">條件</label>  <div class="col-md-9 mb-3">  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="cond\_and" value="and" name="condradio" />  <label class="form-check-label" for="cond\_and">and</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="cond\_or" value="or" name="condradio" checked />  <label class="form-check-label" for="cond\_or">or</label>  </div>  </div>  </div>  <div class="mb-3 row">  <label class="col-sm-3 col-form-label">新聞類別</label>  <div class="col-md-9">  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="cate\_all" value="全部" name="cateradio" checked />  <label class="form-check-label" for="cate\_all">全部</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="cate\_politics" value="政治" name="cateradio" />  <label class="form-check-label" for="cate\_politics">政治</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="cate\_tech" value="科技" name="cateradio" />  <label class="form-check-label" for="cate\_tech">科技</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="cate\_sports" value="運動" name="cateradio" />  <label class="form-check-label" for="cate\_sports">運動</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="cate\_stocks" value="證卷" name="cateradio" />  <label class="form-check-label" for="cate\_stocks">證卷</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="cate\_economy" value="產經" name="cateradio" />  <label class="form-check-label" for="cate\_economy">產經</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="cate\_entertainment" value="娛樂" name="cateradio" />  <label class="form-check-label" for="cate\_entertainment">娛樂</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="cate\_life" value="生活" name="cateradio" />  <label class="form-check-label" for="cate\_life">生活</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="cate\_international" value="國際" name="cateradio" />  <label class="form-check-label" for="cate\_international">國際</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="cate\_society" value="社會" name="cateradio" />  <label class="form-check-label" for="cate\_society">社會</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="cate\_culture" value="文化" name="cateradio" />  <label class="form-check-label" for="cate\_culture">文化</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="cate\_china" value="兩岸" name="cateradio" />  <label class="form-check-label" for="cate\_china">兩岸</label>  </div>  </div>  </div>  <div class="mb-3 row">  <label class="col-md-3 col-form-label">最近多少周?</label>  <div class="col-md-9">  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="wk1" value="1" name="wkradio" />  <label class="form-check-label" for="wk1">1</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="wk2" value="2" name="wkradio" checked />  <label class="form-check-label" for="wk2">2</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="wk3" value="3" name="wkradio" />  <label class="form-check-label" for="wk3">3</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="wk4" value="4" name="wkradio" />  <label class="form-check-label" for="wk4">4</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="wk6" value="6" name="wkradio" />  <label class="form-check-label" for="wk6">6</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="wk8" value="8" name="wkradio" />  <label class="form-check-label" for="wk8">8</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" id="wk12" value="12" name="wkradio" />  <label class="form-check-label" for="wk12">12</label>  </div>  <div class="form-text text-muted">以最新資料時間為準，往前推多少周?</div>  </div>  </div>  <div class="mb-3 row">  <div class="col-md-9 ms-auto">  <button type="button" id="btn\_ok" class="btn btn-primary">查詢</button>  </div>  </div>  </div>  </div>  </div>  <!-- 輸入區塊結束 -->  <!-- 顯示區塊 -->  <div class="col-lg-6 mb-2">  <div class="card">  <div class="card-header">  <h3 class="h6 text-uppercase mb-0">出現頻率以時間呈現</h3>  </div>  <div class="card-body">  <small>觀察每個時間點的有多少篇報導(聲量大小)</small>  <div class="row">  <canvas id="keyword\_time\_line\_chart"></canvas>  </div>  </div>  </div>  </div>  <!-- 區塊結束 -->  <!-- 同時出現的關鍵字區塊 -->  <div class="col-lg-6 mb-2">  <div class="card">  <div class="card-header">  <h3 class="h6 text-uppercase mb-0">熱門程度:有幾篇新聞報導提到它?</h3>  </div>  <div class="card-body">  <ul id="keyword\_article\_count"></ul>  </div>  </div>  </div>  <!-- 區塊結束 -->  <!-- 熱門程度區塊 -->  <div class="col-lg-6 mb-2">  <div class="card">  <div class="card-header">  <h3 class="h6 text-uppercase mb-0">熱門程度:提到它的次數?</h3>  </div>  <div class="card-body">  <ul id="keyword\_frequency"></ul>  </div>  </div>  </div>  <!-- 區塊結束 -->  {% endblock %} {% block extra\_js %}  <!-- 這裡的java scrip等頁面初始化之後才載入與執行 -->  <!-- chartjs圖js -->  <script src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.13.0/moment.min.js"></script>  <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.7.3/Chart.min.js"></script>  <!-- 程式碼區 -->  <script>  call\_ajax()    //\*\*按鈕事件  $('#btn\_ok').on('click', function () {  call\_ajax()  }) //event function    $("input[name='cateradio']").on('change', function () {  call\_ajax()  }) //event function    $("input[name='wkradio']").on('change', function () {  call\_ajax()  }) //event function    $("input[name='condradio']").on('change', function () {  call\_ajax()  }) //event function    function call\_ajax() {  const userkey = $('#input\_keyword').val()  const weeks = $("input[name='wkradio']:checked").val()  const cate = $("input[name='cateradio']:checked").val()  const cond = $("input[name='condradio']:checked").val()    if (userkey.length < 2) {  alert('輸入關鍵字不可空白或小於兩個中文字!')  return 0  }    $.ajax({  type: 'POST',  url: 'api\_get\_top\_userkey/',  //url: 'http://163.18.23.20:8000/userkeyword/api\_get\_top\_userkey/',  data: {  userkey: userkey,  cate: cate,  weeks: weeks,  cond: cond  }, // pass to server  success: function (received) {  const article\_count = received['key\_occurrence\_cat']  console.log(article\_count)  $('#keyword\_article\_count').empty()    //將內容加上li標籤附加起來顯示  for (let key in article\_count) {  let paste = '<li>' + key + ':' + article\_count[key] + '</li>'  $('#keyword\_article\_count').append(paste)  }  const kwfreq = received['key\_freq\_cat']  console.log(kwfreq)  $('#keyword\_frequency').empty()  for (let key in kwfreq) {  let paste = '<li>' + key + ':' + kwfreq[key] + '</li>'  $('#keyword\_frequency').append(paste)  }    const data\_key\_time\_freq = received['key\_time\_freq']  console.log(data\_key\_time\_freq)  showtimechart(data\_key\_time\_freq)  } //function  }) //ajax  } //call\_ajax()    // 宣告全域變數用於存放圖表實例  let line\_chart = null    function showtimechart(data\_key\_time\_freq) {  //取得繪圖元件  const ctx\_key\_time = document.getElementById('keyword\_time\_line\_chart').getContext('2d')    const myoptions = {  type: 'line',  data: {  datasets: [  {  label: 's2',  borderColor: 'red',  data: data\_key\_time\_freq // your data here!  }  ]  },  options: {  legend: {  display: false  },  scales: {  xAxes: [  {  type: 'time',  time: {  unit: 'day',  displayFormats: {  //day: 'DD-MM-YYYY'  day: 'MM/DD'  }  }  }  ],  yAxes: [  {  ticks: {  beginAtZero: true  },  display: true,  scaleLabel: {  display: true,  labelString: '出現次數'  }  }  ]  }  }  }    // 檢查並清除舊圖  if (line\_chart) {  line\_chart.destroy()  }    // 畫新圖  line\_chart = new Chart(ctx\_key\_time, myoptions)  }  </script>  {% endblock %} |

Step 9: base.html

Only one line you need to modify:

For CNA news (中央社)

<a class="nav-link" href="{% url 'app\_user\_keyword:home' %}">你的關鍵詞熱門度分析</a>

{% url 'app\_user\_keyword:home' %}

Step 10: Run server

python manage.py runserver 8000

What does your web page look like?