**『巴哈姆特資訊看版探勘』**

F109118121\_許唐維

**i.Data crawl**

We crawl three topics in bahamut website, and we use manual headers to access because bahamut has anti-crawl feature.

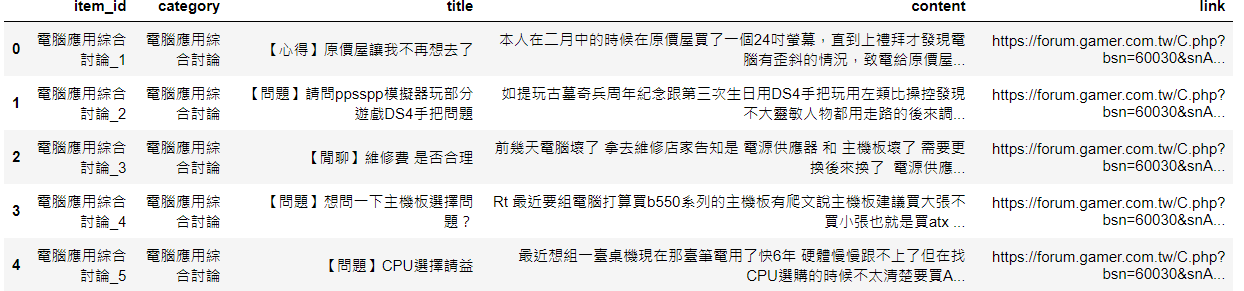
|  |
| --- |
| bsn\_links =['60030', '60001', '60559']  bsn\_categories=['電腦應用綜合討論', '電視遊樂器綜合討論', '智慧型手機']  base\_url = 'https://forum.gamer.com.tw/'  HEADERS = {      'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.92 Safari/537.36',  } |

First, we get all article pages from every topic at the first page.

Then we get the article total reply page to get all the replies.

|  |
| --- |
| def get\_article\_url\_list(forum\_url):      r = requests.get(forum\_url, headers=HEADERS)      if r.status\_code != requests.codes.ok:          print("載入失敗")          return []        article\_url\_list = []      soup = BeautifulSoup(r.text,'lxml')      item\_blocks = soup.select('table.b-list > tr[class="b-list\_\_row b-list-item b-imglist-item"]')      for item\_block in item\_blocks:          title\_block = item\_block.select\_one('.b-list\_\_main\_\_title')          article\_url = f"https://forum.gamer.com.tw/{title\_block.get('href')}"          article\_url\_list.append(article\_url)      return article\_url\_list    def get\_article\_total\_page(soup):      article\_total\_page = soup.select\_one('.BH-pagebtnA > a:last-of-type').text      return int(article\_total\_page)    #主題頁面資訊  def get\_article\_info(article\_url):      soup = BeautifulSoup(requests.get(article\_url, headers=HEADERS).text,'lxml')      article\_title = soup.select\_one('h1.c-post\_\_header\_\_title').text        article\_total\_page = get\_article\_total\_page(soup)  #獲得總樓層的數量        reply\_info\_list = []      for page in range(article\_total\_page):          crawler\_url = f"{article\_url}&page={page + 1}"          reply\_list = get\_reply\_info\_list(crawler\_url)          reply\_info\_list.extend(reply\_list)          random.uniform(1, 3)        article\_info = {          'title': article\_title,          'url': article\_url,          'reply': reply\_info\_list[0],          'category': category      }      return article\_info  def get\_reply\_info\_list(url):        reply\_info\_list = []      soup = BeautifulSoup(requests.get(url, headers=HEADERS).text,'lxml')      reply\_blocks = soup.select('section[id^="post\_"]')        for reply\_block in reply\_blocks:            reply\_info = reply\_block.select\_one('.c-article\_\_content').text          reply\_info = re.sub(r'\n+',"", reply\_info)          reply\_info\_list.append(reply\_info)      return reply\_info\_list |

We use dataframe to save our data, which display below.



Then we use CkipLab to tokenize our content and count the word frequency, which shows below.

|  |
| --- |
| category\_freq = []  word = []  for i in range(0, len(df2)):        tokens = ws(df2.all\_contents[i])      tokens\_pos = pos(tokens)      word\_pos\_pair = [list(zip(w,p)) for w, p in zip(tokens, tokens\_pos)]      with open('stops\_chinese\_traditional.txt', 'r', encoding='utf8') as f:          stops = f.read().split('\n')      allowPOS=['Na','Nb','Nc','VA','VAC','VB','VC']      tokens\_v2 =[]      for wp in word\_pos\_pair:          tokens\_v2.append([w for w,p in wp if w not in stops and (len(w) >= 2) and p in allowPOS])      tokens\_pos = pos(tokens\_v2)      word\_pos\_pair = [list(zip(w,p)) for w, p in zip(tokens\_v2, tokens\_pos)]        word.append(tokens\_v2[0])      keyfreqs =[]      filtered\_words =[]      for wp in word\_pos\_pair:          word\_frequency(wp)      counter = Counter(filtered\_words)      keyfreqs.append(counter.most\_common(200))        category\_freq.append(keyfreqs[0]) |



We use freq to present our midterm project with Django.

**ii.** **Django**

We use Django to create analysis website.

1. Create a new Django project

Step 1: Create a folder named “midtest”

mkdir midtest

Step 2: Go into the folder “midtest”

cd midtest

Step 3: Create a project configures folder named website\_configs

django-admin startproject website\_configs .

2. Create an APP

Step 1: Create an APP named “app\_top\_keyword”

django-admin startapp app\_top\_keyword

Step 2: setting website\_configs’s settings.py and urls.py

|  |
| --- |
| **Setting setting.py**  import os  ALLOWED\_HOSTS = ['127.0.0.1']  INSTALLED\_APPS = [  …………      'app\_top\_keyword',  ]  TEMPLATES = [      {  ……………       'DIRS': [os.path.join(BASE\_DIR, 'templates')],  ………………  **Setting urls.py**  from django.contrib import admin  from django.urls import path  from django.urls import include  urlpatterns = [      #path('admin/', admin.site.urls),      path('topword/', include('app\_top\_keyword.urls')),  ] |

3. Setting app\_top\_keyword’s views.py and create urls.py

|  |
| --- |
| **Setting views.py**  from django.views.decorators.csrf import csrf\_exempt  from django.shortcuts import render  from django.http import JsonResponse  import pandas as pd  from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator  import matplotlib.pyplot as plt  # render渲染網頁  def home(request):      return render(request, 'app\_top\_keyword/home.html')  # read df  df\_topkey = pd.read\_csv(      'app\_top\_keyword/dataset/baha\_dataset\_freq.csv', sep=',')  # prepare data  data = {}  for idx, row in df\_topkey.iterrows():      data[row['category']] = eval(row['top\_keys'])  # We don't use it anymore, so delete it to save memory.  del df\_topkey  # POST: csrf\_exempt should be used  # 指定這一支程式忽略csrf驗證  @csrf\_exempt  def api\_get\_cate\_topword(request):      cate = request.POST.get('news\_category')      # cate = request.POST['news\_category']  # this command also works.      topk = request.POST.get('topk')      topk = int(topk)      print(cate, topk)      chart\_data, wf\_pairs = get\_category\_topword(cate, topk)      response = {          'chart\_data': chart\_data,          'wf\_pairs': wf\_pairs,      }      print(response)      return JsonResponse(response)  def get\_category\_topword(cate, topk=10):      wf\_pairs = data[cate][0:topk]      print(data)      words = [w for w, f in wf\_pairs]      freqs = [f for w, f in wf\_pairs]      chart\_data = {          "category": cate,          "labels": words,          "values": freqs}      return chart\_data, wf\_pairs  print("app\_top\_keywords--類別熱門關鍵字載入成功!")  **Create app\_top\_keyword urls.py**  **In folder app\_top\_keyword, create a python file named “urls.py”**  create app\_top\_keywrod/urls.py  **Setting app\_top\_keywrod/urls.py**  from django.urls import path  from app\_top\_keyword import views  # Declare a namespace for this APP  app\_name = 'app\_top\_keyword'  urlpatterns = [      # For home      path('', views.home, name='home'),  # app\_top\_keyword:home      # For Ajax      path('api\_get\_cate\_topword/', views.api\_get\_cate\_topword),  ] |

4. loading dataset

In folder app\_top\_keyword, create a file named “dataset”and loading freq.csv.



5. create tempates in file midtest, then create navbar.html

|  |
| --- |
| **Create midtest/tempates/navbar.html**    **Setting nabar.html**  <div class="col-lg-12 mb-2">      <nav class="navbar navbar-expand-lg navbar-light" style="background-color: #e3f2fd;">          <a class="navbar-brand" href="#">巴哈網站大數據</a>          <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"                  aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">              <span class="navbar-toggler-icon"></span>          </button>          <div class="collapse navbar-collapse" id="navbarSupportedContent">              <ul class="navbar-nav mr-auto">                  <li class="nav-item">                      <a class="nav-link" href="{% url 'app\_top\_keyword:home' %}">熱門關鍵詞分析</a>                  </li>              </ul>          </div>      </nav>  </div> |

6. create app\_top\_key\_word/templates/app\_top\_key\_word /home.html

|  |
| --- |
| **Create app\_top\_key\_word/templates/app\_top\_key\_word /home.html**  **Setting home.html**  <!DOCTYPE html>  <html lang="en">  <head>      <title>輿情分析平台</title>      <meta charset="utf-8">      <meta name="viewport" content="width=device-width, initial-scale=1">      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>      <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>  </head>  <body>      <div class="container">          <div class="row">              <!-- Here insert the navigation bar -->              {%include 'navbar.html'%}              <div class="col-lg-12">                  <h1>巴哈三個專版的關鍵詞</h1>                  <p>熱門度分析:可以了解專版關注那些重要的議題</p>              </div>              <!-- 新聞類別選單------------------------------->              <div class="col-lg-6 mb-2">                  <div class="card">                      <div class="card-header">                          <h3 class="h6 text-uppercase mb-0">熱門關鍵字瀏覽與繪圖(資料週期:資料截止時間為上週五)</h3>                      </div>                      <div class="card-body">                          <!--新聞類別選單 form group-->                          <div class="form-group row">                              <label class="col-sm-3 form-control-label">專版類別</label>                              <div class="col-md-9">                                  <select id="cate-selected" name="news\_category" class="form-control">                                      <!--<option>請選擇</option>-->                                      <option>電腦應用綜合討論</option>                                      <option>電視遊樂器綜合討論</option>                                      <option>智慧型手機</option>                                  </select>                                  <small class="form-text text-muted">請選擇專版類別                                  </small>                              </div>                          </div>                          <!--form group-->                          <!--熱門詞多少個?form group-->                          <div class="form-group row">                              <label class="col-md-3 form-control-label">多少個熱門詞?</label>                              <div class="col-md-9">                                  <input id="topk-selected" name="topk" value="10"                                      class="form-control form-control-success">                                  <small class="form-text text-muted">內定值為10                                  </small>                              </div>                          </div>                          <!--form group-->                          <!--submit按鈕form group-->                          <div class="form-group row">                              <div class="col-md-9 ml-auto">                                  <button type="button" id="btn-ok" class="btn btn-primary">查詢</button>                              </div>                          </div>                          <!--form group-->                      </div>                      <!--card body-->                  </div>                  <!--column-->              </div><!-- 區塊結束 -->              <!-- 繪圖區塊--------------------------------------------------------->              <div class="col-lg-6 mb-5">                  <div class="card">                      <div class="card-header">                          <h3 class="h6 text-uppercase mb-0">熱門關鍵字繪圖</h3>                      </div>                      <div class="card-body">                          <canvas id="mychart"></canvas>                      </div>                  </div>              </div><!-- 區塊結束 -->              <!-- 熱門關鍵字區塊----------------------------------------------------->              <div class="col-lg-6 mb-5">                  <div class="card">                      <div class="card-header">                          <h3 class="h6 text-uppercase mb-0">熱門關鍵字</h3>                      </div>                      <div class="card-body">                          <ul id="topkeys"></ul>                      </div>                  </div>              </div><!-- 區塊結束 -->          </div> <!-- row結束-->      </div> <!-- container結束-->  </body>  </html>  <!-- chartjs圖js-->  <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.7.3/Chart.min.js"></script>  <!-- 程式碼區 -->  <script>  // Write your JS code here!  call\_ajax();  //let cate = $('#cate-selected').val();  //console.log(cate);  //let topk = $('#topk-selected').val();  //console.log(topk);  //alert(topk);     $('#btn-ok').on('click', function () {      console.log("按下按鈕");      call\_ajax();    showTopKeys(wf\_pairs);     });   //\*新聞類別選單select被選中值有改變時，執行以下事件     $('#cate-selected').on('change', function () {       let cate = $('#cate-selected').val();             console.log(cate);     }); //event function  // Exercise#2: Define a function  // Please paste showTopKeys function here!      //\* 顯示關鍵詞資料函數      function showTopKeys(items) {          //先清除前一次的資料          $('#topkeys').empty();          //將內容加上li標籤附加起來，顯示在顯示區"topkeys"          for (let i = 0; i < items.length; i++) {              let item\_li = "<li>" + items[i] + "</li>";              $('#topkeys').append(item\_li);          }      } //function      // Call function when btn\_ok is clicked  // Exercise#4: Define “call\_ajax” function to perform Ajax  // Call ajax function when page is loaded and button is clicked.     // See what the data received from backend API looks like.     // Display word frequency pairs.     function call\_ajax() {          let cate = $('#cate-selected').val();          let topk = $('#topk-selected').val();          $.ajax({              type: "POST",              //url: "/topword/api\_get\_cate\_topword/",              url: "http://127.0.0.1:8000/topword/api\_get\_cate\_topword/",              //url: "http://163.18.22.32:8000/topword/api\_get\_cate\_topword/",              //url: "api\_get\_cate\_topword/", //Not recommended!              data: { "news\_category": cate, "topk": topk },              success: function (received) {                  console.log(received);                  let chart\_data = received.chart\_data;                  let wf\_pairs = received.wf\_pairs;                  console.log(wf\_pairs);                  showTopKeys(wf\_pairs);                  showChart(chart\_data);                  //showChart(chart\_data);              } //success function          }); //ajax      } //call\_ajax   //\*\*繪圖函數showChart()   function showChart(chart\_data) {  // 畫圖需要的數據資料  let values = chart\_data.values;  let labels = chart\_data.labels;  let category = chart\_data.category;  //第1個變數: 餵給chart的資料data  let data = {      labels: labels,      datasets: [{          label: category,          data: values,          backgroundColor: randomColors(values.length),          borderColor: randomColors(values.length),          borderWidth: 1,      }],  };  //第2個變數: chart的選項  指定y坐標軸從零開始顯示  let options = {      scales: {          yAxes: [{              ticks: {                  beginAtZero: true              }          }]      },  };  //取得在前面html區域欲顯示的圖代號  let canvas\_mychrat = document.getElementById("mychart");  //\*\*先清除前一個圖 再繪新圖  // 可以印出barchart物件是否存在  // console.log(window.barchart);  //先清除前一個圖 再繪新圖 if 有以下兩種寫法皆可  // if (window.barchart)  //若存在則為true  // if (typeof (barchart) != "undefined"){  if (window.barchart) {      barchart.destroy();  }  //\*\*繪圖(產生一個圖物件變數名稱為barchart)  // 必須全域變數--注意:前面不要有let, var, const等修飾詞  // 理由: 我們要讓它存在於網頁全域變數，  // 這樣我們才方便判斷是否有前一次的圖，如果存在有，要刪除之，否則，很多張圖會疊在一起  barchart = new Chart(canvas\_mychrat, {      type: 'bar',      data: data,      options: options,  });  //\*\* 產生隨機顏色  function randomColors(num\_colors) {      let colors = [];      for (i = 0; i < num\_colors; i++) {          let r = Math.floor(Math.random() \* 255);          let g = Math.floor(Math.random() \* 255);          let b = Math.floor(Math.random() \* 255);          let rgb = `rgba(${r},${g},${b},0.5)` // (red, green, blue, alfa) alfa透明度          colors.push(rgb);      }      return colors;  }  } //show chart function  </script> |

7.Start Django Sever, show the midtest websire

Step 1: Start Django Sever

python manage.py runserver

step 2: 127.0.0.1:8000/topword/ show the webside

