

HW1 LISP Report

執行環境：Ubuntu 16.04

Question1:

```
walltsai@walltsai-System-Product-Name: ~/Programming_Language/HW2
1 import urllib.request
2 import re
3 import matplotlib.pyplot as plt; plt.rcParams()
4 import numpy as np
5 import matplotlib.pyplot as plt
6
7 author = input("Input Author : ");
8 #author = "Ian Goodfellow"
9 author = author.replace(" ", "+")
10 start = 0
11 size = 50
12 url = "https://arxiv.org/search/?query=" + author + "&searchtype=author&size=" + str(size) + "&order=announced_date_first&start=" + str(start)
13 content = urllib.request.urlopen(url)
14 html_str = content.read().decode('utf-8')
15
16 pattern = 'of[\\s\\S]*?results for author: <span'
17 result = re.findall(pattern,html_str)
18 num_of_result = int(result[0].split("of ")[1].split(" results for author: <span")[0].strip())
19 #print(num_of_result)
20
21 pattern = 'originally announced[\\s\\S]*?</p>'
22 result = re.findall(pattern,html_str)
23 pattern2 = 'class="author[\\s\\S]*?</p>'
24 result2 = re.findall(pattern2,html_str)
25
26 year = []
27 freq = []
28 f = 0
29 print("[ Author: " + author + " ]")
```

26,1 頂端

```
walltsai@walltsai-System-Product-Name: ~/Programming_Language/HW2
30 while start < num_of_result:
31     for i in range(0,len(result)):
32         if result2[i].find(author.replace("+", " ")) >= 0:
33             r = result[i]
34             y_list = r.split();
35             yr = y_list[3].strip(".")
36             if yr in year:
37                 f += 1
38             else:
39                 freq.append(f)
40                 f = 1
41                 year.append(yr)
42         start += size
43         url = "https://arxiv.org/search/?query=" + author + "&searchtype=author&size=" + str(size) + "&order=announced_date_first&start=" + str(start)
44         content = urllib.request.urlopen(url)
45         html_str = content.read().decode('utf-8')
46         result = re.findall(pattern,html_str)
47         result2 = re.findall(pattern2,html_str)
48         freq.append(f)
49         freq.pop(0)
50
51 y_pos = np.arange(len(year))
52 plt.bar(y_pos, freq, align='center', alpha=0.5)
53 plt.xticks(y_pos,year)
54 plt.show()
```

54,4 底端

執行方式：

```
walltsai@walltsai-System-Product-Name:~/Programming_Language/HW2$ python3 question1.py
Input Author : Ian Goodfellow
```

說明：

先 import urllib,re,matplotlib 等 module

Input “author name”(名字中間是空格)

url 設計：“https://arxiv.org/search/?query=” + author
+ “&searchtype=author&size=” + str(size) +
“&order=announced_date_first&start=” + str(start)
(size= 這頁最多文章量 start = 這頁開始的文章編號-1
announce_date_first = 文章由舊至新)

照範例 code 打開 html

第一個 pattern 用於找出總 results 數

並存在 num_of_result

第二個 pattern 用於找出所有包含年份的 string, 存在 result

pattern2 用於找出一篇文章的所有作者, 存在 result2

year 和 freq 分別用於存 所有年份 和 次數 的 list

print("[Author: " + author + "]")

進入迴圈

while start < num_of_result (這頁有文章)

for i in range(0, len(result))

if result2[i] 包含作者

r = result[i]

將 r 之中的年份切出存於 yr

if yr in year (非初見之年份)

f++

else

freq.append(f) (將上個年份的次數存入)

f = 1

year.append(yr) (將新的年份存入)

start += size (換頁)

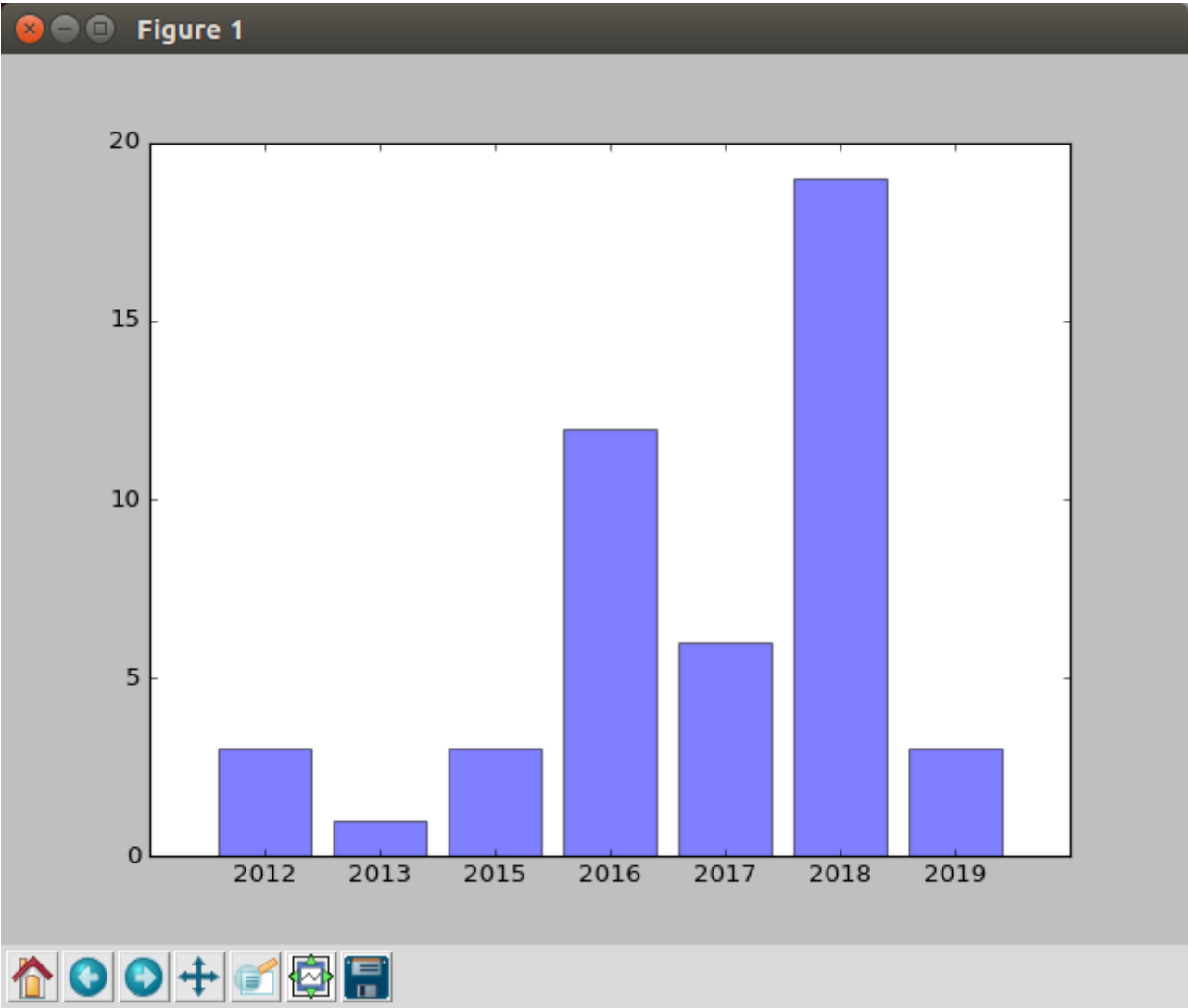
更新 url 打開 html 再用 pattern 於找出所有包含年份的 string

freq.append(f) (將最一個年份的次數存入)

freq.pop(0) (freq[0]是多餘的)

year 為橫軸 freq 為縱軸 畫出柱狀圖

執行結果：



Question2:

```
walltsai@walltsai-System-Product-Name: ~/Programming_Language/HW2
1 import urllib.request
2 import re
3
4 author = input("Input Author : ");
5 #author = "Ian Goodfellow"
6 author = author.replace(" ", "+")
7 start = 0
8 size = 50
9 url = "https://arxiv.org/search/?query=" + author + "&searchtype=author&size=" + str(
    size) + "&order=announced_date_first&start=" + str(start)
10 content = urllib.request.urlopen(url)
11 html_str = content.read().decode('utf-8')
12
13 pattern = 'of[\\s\\S]*?results for author: <span'
14 result = re.findall(pattern,html_str)
15 num_of_result = int(result[0].split("of ")[1].split(" results for author: <span")[0].
    strip())
16 print(num_of_result)
17
18 pattern = 'searchtype=author&query[\\s\\S]*?</a>'
19 result = re.findall(pattern,html_str)
20
21 co_author = []
22 co_low = []
23 time = []
24 print("[ Author: " + author + " ]")
"question2.py" 46L, 1504C 已寫入 16,1 頂端

w
~
~
~
25 while start < num_of_result:
26     for r in result:
27         name = re.search(">[\\s\\S]*?<",r).group().strip("><")
28         while name.startswith(" "):
29             name = name[1:len(name)]
30         if name.lower() in co_low:
31             i = co_low.index(name.lower())
32             time[i] += 1
33         else:
34             co_author.append(name)
35             co_low.append(name.lower())
36             time.append(1)
37     start += size
38     url = "https://arxiv.org/search/?query=" + author + "&searchtype=author&size=" +
    str(size) + "&order=announced_date_first&start=" + str(start)
39     content = urllib.request.urlopen(url)
40     html_str = content.read().decode('utf-8')
41     result = re.findall(pattern,html_str)
42 co_author.sort()
43 for r in co_author:
44     if r.lower() != author.replace("+", " ").lower():
45         i = co_low.index(r.lower())
46         print(r + ": " + str(time[i]) + " times")
~
~
~
28,1-4 底端
```

執行方式：

```
walltsai@walltsai-System-Product-Name:~/Programming_Language/HW2$ python3 question2.py
Input Author : Ian Goodfellow
```

說明：

先 import urllib,re 等 module

Input “author name”(名字中間是空格)

url 設計：同 question1

照範例 code 打開 html

第一個 pattern 用於找出總 results 數

並存在 num_of_result

第二個 pattern 用於找出所有包含人名的超連結

co_author 和 time 分別用於存 其他作者 和 次數 的 list

co_low 用於存 其他作者 的 lowercase 的 list

print("[Author: " + author + "]")

進入迴圈

while start<num_of_result (這頁有文章)

for r in result

將 r 之中的 名字 切出存於 name

while name.startswith(" ") (防止開頭是空格)

name = name[1:len(name)]

if name.lower() in co_low (非初見之作者)

i = co_low.index(name.lower()) (找到他的 index)

time[i] += 1

else (新作者)

co_author.append(name) (新作者存入)

co_low.append(name.lower()) (小寫新作者存入)

time.append(1) (新作者次數)

start += size (換頁)

更新 url 打開 html 再用 pattern 於找出所有包含年份的 string

co_author.sort() (將 list 由 a-z 排序)

for r in co_author

if r.lower() != author.replace("+", " ").lower(): (排除作者本人)

i = co_low.index(r.lower()) (找到他的 index)

print(r + ": " + str(time[i]) + " times")

執行結果：

```
walltsai@walltsai-System-Product-Name:~/Programming_Language/HW2$ python3 question2.py
Input Author : Ian Goodfellow
[ Author: Ian+Goodfellow ]
Aaron Courville: 10 times
Abhibhav Garg: 1 times
Alan Yuille: 1 times
Alec Radford: 1 times
Aleksander Madry: 1 times
Alex Kurakin: 1 times
Alexander Belopolsky: 1 times
Alexander Matyasko: 1 times
Alexandre de Brébisson: 1 times
Alexey Kurakin: 7 times
Alireza Makhzani: 1 times
Amjad Almahairi: 1 times
Ananthram Swami: 1 times
Anatoly Belikov: 1 times
Andrew Harp: 1 times
Andrew M. Dai: 3 times
Andrew M. Saxe: 1 times
Andy Chu: 1 times
Andy Davis: 1 times
Anish Athalye: 1 times
Arnaud Bergeron: 2 times
Ashish Agarwal: 1 times
Augustus Odena: 6 times
Aurko Roy: 2 times
Avital Oliver: 1 times
Balaji Lakshminarayanan: 1 times
Been Kim: 2 times
Ben Hamner: 1 times
Bing Xu: 1 times
Brendan Frey: 1 times
Brian Cheung: 1 times
Catherine Olsson: 4 times
Chelsea Finn: 1 times
Chetan Ramaiah: 1 times
Chiyuan Zhang: 1 times
Chris Olah: 1 times
Christian Szegedy: 2 times
Christof Angermueller: 1 times
Christopher Olah: 1 times
Cihang Xie: 2 times
Colin Raffel: 4 times
Craig Citro: 1 times
Cristian Grozea: 1 times
Da Xiao: 1 times
Dan Boneh: 2 times
Dan Mane: 1 times
David Andersen: 1 times
David Berthelot: 2 times
David Thaler: 1 times
David Warde-Farley: 5 times
Derek Murray: 1 times
Dimitris Athanasakis: 1 times
Dimitris Metaxas: 1 times
Dimitris Tsipras: 1 times
Dong-Hyun Lee: 1 times
Dumitru Erhan: 2 times
Dzmitry Bahdanau: 1 times
Ekin D. Cubuk: 1 times
Eugene Brevdo: 1 times
```


Fangxiang Feng: 1 times
Fangzhou Liao: 1 times
Fartash Faghri: 2 times
Florian Tramèr: 2 times
Frédéric Bastien: 3 times
Gamaleldin F. Elsayed: 2 times
Garrison Cottrell: 1 times
Geoffrey Irving: 1 times
George E. Dahl: 1 times
Greg S. Corrado: 1 times
Guillaume Alain: 1 times
H. Brendan McMahan: 2 times
Han Zhang: 1 times
Harini Kannan: 1 times
Ian J. Goodfellow: 15 times
Ilya Mironov: 2 times
Ilya Sutskever: 1 times
Jacob Buckman: 1 times
James Bergstra: 4 times
Jan Chorowski: 1 times
Jascha Sohl-Dickstein: 2 times
Jean Pouget-Abadie: 1 times
Jeffrey Dean: 1 times
Jianyu Wang: 1 times
Jingjing Xie: 1 times
Joan Bruna: 1 times
John Park: 1 times
John Shawe-Taylor: 1 times
Jonas Rauber: 2 times
Jonathan Uesato: 1 times
Jonathon Shlens: 3 times
Josh Bleecher Snyder: 1 times
Josh Levenberg: 1 times
Julian Ibarz: 1 times
Julius Adebayo: 2 times
Jun Zhu: 1 times
Junjiajia Long: 1 times
Justin Bayer: 1 times
Justin Gilmer: 4 times
Karen Hambardzumyan: 1 times
Kunal Talwar: 3 times
Kyunghyun Cho: 1 times
Li Zhang: 2 times
Lukasz Kaiser: 1 times
Lukasz Romaszko: 1 times
Luke Metz: 1 times
Maithra Raghu: 1 times
Manjunath Kudlur: 1 times
Marius Popescu: 1 times
Martin Wattenberg: 1 times
Martín Abadi: 4 times
Matthieu Devin: 1 times
Maxim Milakov: 1 times
Mehdi Mirza: 5 times
Michael Isard: 1 times
Michael Muelly: 1 times
Mihaela Rosca: 1 times
Ming Liang: 1 times
Moritz Hardt: 1 times
Motoki Abe: 1 times
Navdeep Jaitly: 1 times
Nicholas Carlini: 4 times

Nicolas Ballas: 1 times
Nicolas Bouchard: 2 times
Nicolas Boulanger-Lewandowski: 1 times
Nicolas Papernot: 10 times
Olivier Breuleux: 1 times
Oriol Vinyals: 1 times
Pascal Lamblin: 2 times
Patrick McDaniel: 4 times
Paul Barham: 1 times
Paul Christiano: 2 times
Paul Hendricks: 1 times
Pierre Luc Carrier: 1 times
Pierre-Luc Carrier: 1 times
Pieter Abbeel: 1 times
Radu Ionescu: 1 times
Rafal Jozefowicz: 1 times
Rajat Monga: 1 times
Rami Al-Rfou: 1 times
Razvan Pascanu: 2 times
Reuben Feinman: 1 times
Rob Fergus: 1 times
Ruifan Li: 1 times
Rujun Long: 1 times
Ryan P. Adams: 1 times
Ryan Sheatsley: 1 times
Sacha Arnoud: 1 times
Samaneh Azadi: 1 times
Samuel S. Schoenholz: 1 times
Samy Bengio: 3 times
Sandy Huang: 1 times
Sangxia Huang: 1 times
Sanjay Ghemawat: 1 times
Seiya Tokui: 1 times
Sergey Levine: 1 times
Shakir Mohamed: 1 times
Sherjil Ozair: 1 times
Sherry Moore: 1 times
Shreya Shankar: 1 times
Somesh Jha: 1 times
Stephan Zheng: 1 times
Surya Bhupatiraju: 1 times
Takeru Miyato: 1 times
Takuya Akiba: 1 times
The Theano Development Team: 1 times
Thomas Leung: 1 times
Tianqi Chen: 1 times
Tianyu Pang: 1 times
Tim Salimans: 1 times
Tom B. Brown: 2 times
Tom Brown: 2 times
Trevor Darrell: 1 times
Vahid Behzadan: 1 times
Valentin Bisson: 1 times
Vicki Cheung: 1 times
Vinay Shet: 1 times
Vincent Dumoulin: 2 times
Wieland Brendel: 1 times
Will Cukierski: 1 times
Willi Gierke: 1 times
William Fedus: 2 times
Wojciech Zaremba: 2 times
Xavier Bouthillier: 1 times

```
Xi Chen: 1 times
Xiaojie Wang: 1 times
Xiaolin Hu: 1 times
Yan Duan: 1 times
Yang Song: 1 times
Yangqing Jia: 1 times
Yao Qin: 1 times
Yao Zhao: 1 times
Yaroslav Bulatov: 1 times
Yash Sharma: 1 times
Yerkebulan Berdibekov: 1 times
Yi-Lin Juang: 1 times
Yichuan Tang: 1 times
Yingbo Zhou: 1 times
Yinpeng Dong: 2 times
Yoshua Bengio: 12 times
Yuzhe Zhao: 1 times
Z. Berkay Celik: 1 times
Zhi Li: 1 times
Zhifeng Chen: 1 times
Zhishuai Zhang: 2 times
Zhonglin Han: 1 times
Zhou Ren: 1 times
Úlfar Erlingsson: 2 times
walltsai@walltsai-System-Product-Name:~/Programming_Language/HW2$
```