Group 4 Cafoogle

組員：資管二 楊易勳、黃婷筠、盧柏瑜、潘嘉威

* 第四組github連結：https://github.com/DanielYang1209/da_final

1. Introduction / Our topic and motivation

主題：

台灣咖啡廳的後記心得（類似部落格文章）

動機和目標：

市面上的消費者在購買或使用店家所提供的商品與服務前，通常會先觀察其他消費者累積的經驗知識和評論，判斷能否解決消費者的問題或是滿足消費者的需求。

因此我們期望能透過優化Google搜尋引擎，在使用者搜尋「台灣 咖啡廳」後幫助使用者統整出關於網路上一些部落格、美食評論家或者消費者之心得，讓使用者能快速了解哪些咖啡廳符合他們的消費習慣，而非被一些非相干的網站干擾，像是線上訂餐、線上訂位、咖啡廳聯絡、交通方式和咖啡廳官網等等。

透過優化搜尋引擎，不只能幫助使用者以更簡潔快速的方式找出適合度咖啡廳，更能建立起對品牌的信任度與好感度。

2. Search tricks / Our score formulation

```
keywords.add(new Keyword("comment", 3));
keywords.add(new Keyword("blog", 2));
keywords.add(new Keyword("recommendation", 3));
keywords.add(new Keyword("experience", 1));
keywords.add(new Keyword("animal", 1));
keywords.add(new Keyword("cafe", 1));
keywords.add(new Keyword("coffee", 2));

keywords.add(new Keyword("online", -1));
keywords.add(new Keyword("online order", -1));
keywords.add(new Keyword("contact", -3));
keywords.add(new Keyword("direction", -1));
keywords.add(new Keyword("official page", -2));
keywords.add(new Keyword("Gin Gin", -4));
```

關鍵字分數分配：

(1) 分數為正的關鍵字：

- comment 3分
- recommendation 3分
- blog 2分
- coffee 2分
- experience 1分

- animal 1分
- cafe 1分

→ 這些是我們設定上要讓使用者找的內容，也就是咖啡廳評論的部分，所以關鍵字會有評論以及跟其相關的評論、部落格、推薦指數、美食札記這些字詞。不同關鍵字的權重如上所述，其中animal 和cafe此兩個關鍵字是以爬蟲的方式找到的更多關鍵字。

(2) 分數為負的淘汰字：

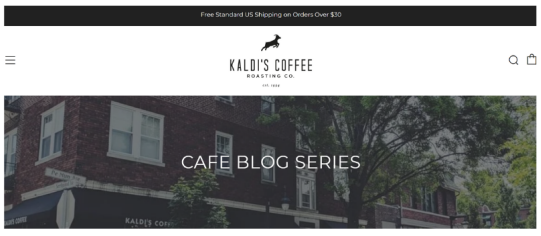
- Gin Gin -4分
- contact -3分
- official page -2分
- online -1分
- online order -1分
- direction -1分

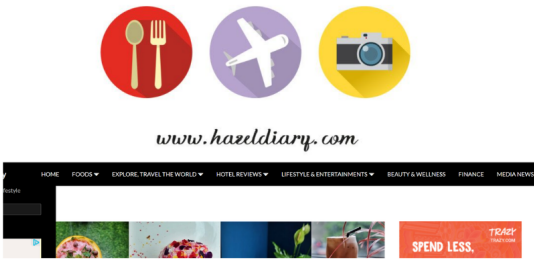
→ 因為線上訂位訂餐等內容是我們不想搜尋到的，所以就反其道而行訂為負分，降低網頁分數來減少搜尋到的可能性。不同淘汰字的權重如上所述，其中Gin Gin此關鍵字是以爬蟲的方式找到的更多關鍵字。

<u>我們最一開始**rank**的網頁：</u>

主要有以下四個，從此四個做延伸找更多的子網頁

https://kaldiscoffee.com/pages/cafe-blog-series
https://hazeldiary.com/2017/08/cafe-hopping-in-taipei-taiwan/
https://sya.tw/ting-tao-cafe/
https://kellyrosie12.com/blog/post/469412900

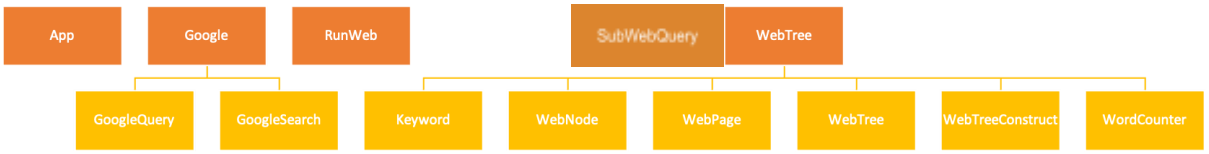




<u>生成更多關鍵字方式：</u>

我們利用python的爬蟲抓取更多，在下面 stage4會做更多闡述。

3. System design / Class diagrams proposal sample



App : main method執行處
RunWeb : 執行網頁
SubWebQuery : 尋找子網頁
Google :
–GoogleQuery : 將關鍵字放入google進行搜尋
–GoogleSearch : 呼叫GoogleQuery執行
WebTree :
–Keyword : 關鍵字名稱、關鍵字的weight
–WebNode : 計算每個Node的分數
–WebPage : url、名稱、關鍵字count、計算關鍵字score
–WebTree Class : set Order, 把tree畫出來後與score一同印出
–WebTreeConstruct : 透過ArrayList建構一個Tree
–WordCounter : 計算關鍵字的出現次數

4. Schedule / How and when to accomplish stages

1. **Stage 0 (HW3) Keyword Counting**

→利用了WordCounter這個class來計算關鍵字出現次數

```java
package webTree;

import java.io.BufferedReader;

public class WordCounter {
    private String urlStr;
    private String content;

    public WordCounter(String urlStr){
        this.urlStr = urlStr;
    }

    private String fetchContent() throws IOException,Exception{
        try {
        URL url = new URL(this.urlStr);
        URLConnection conn = url.openConnection();

            InputStream in = conn.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(in));

            String retVal = "";

            String line = null;

            while ((line = br.readLine()) != null){
                retVal = retVal + line + "\n";
            }

            return retVal;
        } catch (Exception e) {
            // TODO: handle exception
            return "";
        }
    }

    public int countKeyword(String keyword) throws IOException,Exception{
        if (content == null){
            content = fetchContent();
        }

        //To do a case-insensitive search, we turn the whole content and keyword into upper-case:
        content = content.toUpperCase();
        keyword = keyword.toUpperCase();

        int retVal = 0;
        int fromIdx = 0;
        int found = -1;

        while ((found = content.indexOf(keyword, fromIdx)) != -1){
            retVal++;
            fromIdx = found + keyword.length();
        }

        return retVal;
    }
}
```

2. Stage 1 (HW6) Page Ranking

```java
package webTree;

import java.io.IOException;

public class WebPage {
    public String url;
    public String name;
    public WordCounter counter;
    public double score;

    public WebPage(String url, String name){
        this.url = url;
        this.name = name;
        this.counter = new WordCounter(url);
    }

    public void setScore(ArrayList<Keyword> keywords) throws Exception{
        score = 0;
//      1. calculate score
        for(int i=0; i<keywords.size(); i++){
            score+=keywords.get(i).weight*counter.countKeyword(keywords.get(i).name);
        }
    }
    public double getScore(){
        return score;
    }
}
```

→WebPage這個class負責set和get 各個Page的分數

```java
package webTree;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.SSLSession;


public class webTreeConstruct {
    public WebPage rooPage;
    private WebTree tree;
    private String name;
    public String url;
    private ArrayList<Keyword> keywords = new ArrayList<Keyword>();
    public webTreeConstruct(String url,String name){
        this.url=url;
        this.name=name;
        this.rooPage = new WebPage(url,name);
        this.tree=new WebTree(rooPage);

        keywords.add(new Keyword("comment", 3));
        keywords.add(new Keyword("blog", 2));
        keywords.add(new Keyword("recommendation", 3));
        keywords.add(new Keyword("experience", 1));
        keywords.add(new Keyword("animal", 1));
        keywords.add(new Keyword("cafe", 1));
        keywords.add(new Keyword("coffee", 2));

        keywords.add(new Keyword("online", -1));
        keywords.add(new Keyword("online order", -1));
        keywords.add(new Keyword("contact", -3));
        keywords.add(new Keyword("direction", -1));
        keywords.add(new Keyword("official page", -2));
        keywords.add(new Keyword("Gin Gin", -4));

    }
    public void contructTree() throws Exception {
        tree.setPostOrderScore(keywords);
        tree.eularPrintTree();
    }

    public void addChild(String url, String name){
        this.tree.root.addChild(new WebNode(new WebPage(url, name)));
    }
}
```

→WebTreeConstruct這個class負責以Post Order的方式排序score，再add進不同tree裡，接著以eularPrintTree的方式排出

3. Stage 2 (HW6) Site Ranking

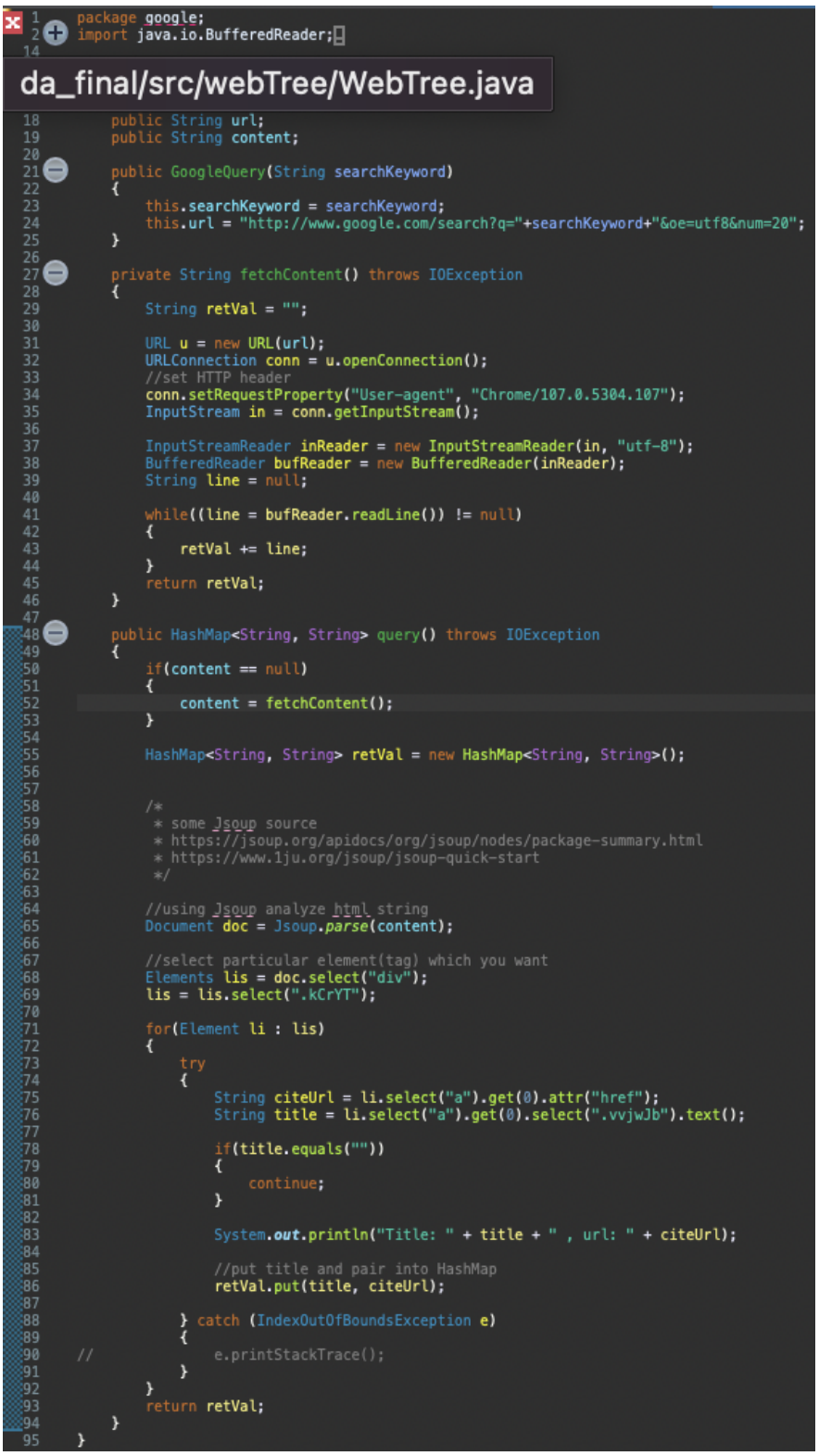
```java
package webTree;

import java.io.IOException;

public class WebNode {
    public WebNode parent;
    public ArrayList<WebNode> children;
    public WebPage webPage;
    public double nodeScore;//This node's score += all its children's nodeScore

    public WebNode(WebPage webPage){
        this.webPage = webPage;
        this.children = new ArrayList<WebNode>();
    }

    public void setNodeScore(ArrayList<Keyword> keywords) throws Exception{
        //this method should be called in post-order mode

        //compute webPage score
        webPage.setScore(keywords);
        //set webPage score to nodeScore
        nodeScore = webPage.score;

        //nodeScore += all children's nodeScore
        for(WebNode child : children){
            nodeScore += child.nodeScore;
        }
    }

    public void addChild(WebNode child){
        //add the WebNode to its children list
        this.children.add(child);
        child.parent = this;
    }

    public boolean isTheLastChild(){
        if(this.parent == null) return true;
        ArrayList<WebNode> siblings = this.parent.children;

        return this.equals(siblings.get(siblings.size() - 1));
    }

    public int getDepth(){
        int retVal = 1;
        WebNode currNode = this;
        while(currNode.parent != null){
            retVal++;
            currNode = currNode.parent;
        }
        return retVal;
    }
}
```

→WebNode的class，這部分和助教lab給的code雷同

```java
public class WebTree {

    public WebNode root;

    public WebTree(WebPage rootPage){
        this.root = new WebNode(rootPage);
    }

    public void setPostOrderScore(ArrayList<Keyword> keywords) throws Exception{
        setPostOrderScore(root, keywords);
    }

    private void setPostOrderScore(WebNode startNode, ArrayList<Keyword> keywords) throws Exception{
        //2. compute the score of children nodes via post-order, then setNodeScore for startNode
        //for(int j=0; j<startNode.; j++){
        if(startNode.isTheLastChild() || !startNode.children.isEmpty()){
            for(int i=0; i<startNode.children.size(); i++){
                this.root=startNode.children.get(i);
                setPostOrderScore(keywords);
            }
        }
        startNode.setNodeScore(keywords);
        this.root=startNode;
    }

    public void eularPrintTree(){
        eularPrintTree(root);
    }

    private void eularPrintTree(WebNode startNode){
        int nodeDepth = startNode.getDepth();

        if(nodeDepth > 1) System.out.print("\n" + repeat("\t", nodeDepth-1));

        System.out.print("(");
        System.out.print(startNode.webPage.name + "," + startNode.nodeScore);

        //3. print child via pre-order
        if(!startNode.children.isEmpty()){
            for(int i=0; i<startNode.children.size(); i++){
                if(!startNode.children.get(i).isTheLastChild() || !startNode.children.isEmpty()){
                    this.root=startNode.children.get(i);
                    eularPrintTree();
                }else{
                    this.root=startNode.parent;
                }
            }
        }
        System.out.print(")");

        if(startNode.isTheLastChild()) System.out.print("\n" + repeat("\t", nodeDepth-2));
    }

    private String repeat(String str, int repeat){
        String retVal = "";
        for(int i = 0; i < repeat; i++){
            retVal += str;
        }
        return retVal;
    }
}
```

→WebTree仔細寫出了WebTreeConstruct裡的setPostOrderScore()和eularPrintTree()的methods

4. Stage 3 (HW 8&9) Refine the rank of Google



```
package google;
import java.io.BufferedReader;
```

da_final/src/webTree/WebTree.java

```
    public String url;
    public String content;

    public GoogleQuery(String searchKeyword)
    {
        this.searchKeyword = searchKeyword;
        this.url = "http://www.google.com/search?q="+searchKeyword+"&oe=utf8&num=20";
    }

    private String fetchContent() throws IOException
    {
        String retVal = "";

        URL u = new URL(url);
        URLConnection conn = u.openConnection();
        //set HTTP header
        conn.setRequestProperty("User-agent", "Chrome/107.0.5304.107");
        InputStream in = conn.getInputStream();

        InputStreamReader inReader = new InputStreamReader(in, "utf-8");
        BufferedReader bufReader = new BufferedReader(inReader);
        String line = null;

        while((line = bufReader.readLine()) != null)
        {
            retVal += line;
        }
        return retVal;
    }

    public HashMap<String, String> query() throws IOException
    {
        if(content == null)
        {
            content = fetchContent();
        }

        HashMap<String, String> retVal = new HashMap<String, String>();


        /*
         * some Jsoup source
         * https://jsoup.org/apidocs/org/jsoup/nodes/package-summary.html
         * https://www.1ju.org/jsoup/jsoup-quick-start
         */

        //using Jsoup analyze html string
        Document doc = Jsoup.parse(content);

        //select particular element(tag) which you want
        Elements lis = doc.select("div");
        lis = lis.select(".kCrYT");

        for(Element li : lis)
        {
            try
            {
                String citeUrl = li.select("a").get(0).attr("href");
                String title = li.select("a").get(0).select(".vvjwJb").text();

                if(title.equals(""))
                {
                    continue;
                }

                System.out.println("Title: " + title + " , url: " + citeUrl);

                //put title and pair into HashMap
                retVal.put(title, citeUrl);

            } catch (IndexOutOfBoundsException e)
            {
//              e.printStackTrace();
            }
        }
        return retVal;
    }
}
```

→Google Query的class，這部分和助教lab給的code相同

```java
The declared package "google" does not match the expected package ""

import java.io.IOException;

public class googleSearch
{
    public static HashMap google(String searchKeyword)
    {
        try
        {
            /*
             * Using different keyword depends on the last number of your student ID
             * 0,1:Tomato
             * 2,3:Liver
             * 4,5:Pokemon
             * 6,7:Tissue
             * 8,9:Process
             */
            HashMap map=new GoogleQuery(searchKeyword).query();
            System.out.println(map);
//          GoogleQuery g = new GoogleQuery("NCCU");
//          g.query();
            //System.out.println("F");
            return map;
        }
        catch (IOException e)
        {
            e.printStackTrace();
            return null;
        }
    }
}
```

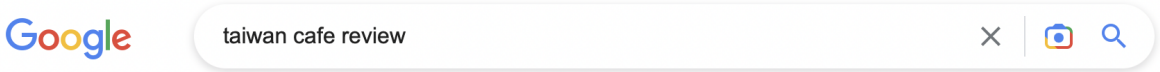
→Google Search，這部分和助教lab給的code相同

5. Stage 4 (HW10) Semantics Analysis

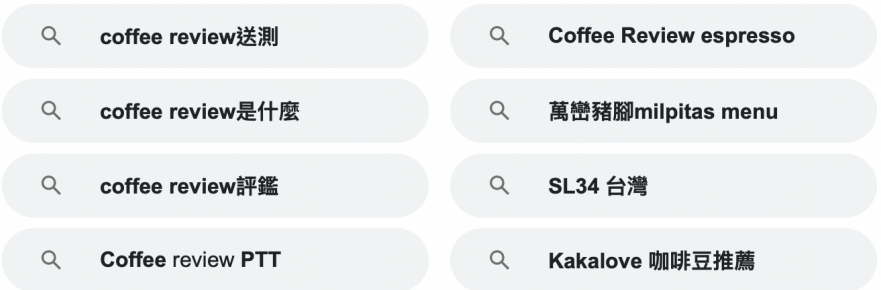
找出更多關鍵字的方法：

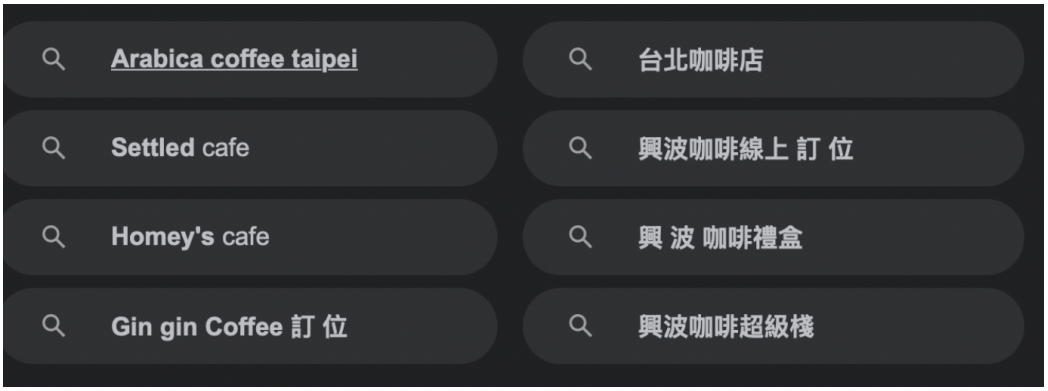
利用beautifulsoup網路爬蟲套件，用python去寫。利用去google查詢以下四項片語(taiwan cafe review, taiwan cafe recommendation, taiwan cafe blog, taiwan cafe review)後，爬取出現在搜尋結果最下面框框中的字，以下如圖：

假使我們查詢taiwan cafe review



滑到搜尋結果最下面會出現更多推薦的關鍵字搜尋，我們爬的目標就是這些：





如以上圖:我們發現查詢後Gin gin coffee出現的次數很多，但是Gin gin coffee並非我們所想要的關鍵字。因此將Gin gin 加入到score負分(-4)的地方。

```
import cfscrape, bs4
from urllib.request import urlopen
##taiwan cafe recommendations
site = "https://www.google.com/search?q=taiwan+cafe+recommenda
scraper1 = cfscrape.create_scraper()
result1 = scraper1.get(site)
soup = bs4.BeautifulSoup(result1.text, "html.parser")
keywords = soup.find_all("div", class_="s75CSd OhScic AB4Wff")
for keyword in keywords:
    print(keyword.text.strip())
print(" ")
##taiwan café blog
site = "https://www.google.com/search?q=taiwan+caf%C3%A9+blog+
result2 = scraper1.get(site)
soup = bs4.BeautifulSoup(result2.text, "html.parser")
keywords = soup.find_all("div", class_="s75CSd OhScic AB4Wff")
for keyword in keywords:
    print(keyword.text.strip())
print(" ")
##taiwan café popular
site = "https://www.google.com/search?q=taiwan+caf%C3%A9+popul
result3 = scraper1.get(site)
soup = bs4.BeautifulSoup(result3.text, "html.parser")
keywords = soup.find_all("div", class_="s75CSd OhScic AB4Wff")
for keyword in keywords:
    print(keyword.text.strip())
print(" ")
##taiwan café review
site = "https://www.google.com/search?q=taiwan+caf%C3%A9+revie
result4 = scraper1.get(site)
soup = bs4.BeautifulSoup(result4.text, "html.parser")
keywords = soup.find_all("div", class_="s75CSd OhScic AB4Wff")
for keyword in keywords:
    print(keyword.text.strip())
```

以上圖是beautifulsoup爬蟲的程式碼

6. Stage 5 (HW11) Publish our work online
→移至底下第6點Demo的部分

7. Stage 6 App
→移至底下第6點Demo的部分

5. Challenges / Techniques that we need but have a hard time to learn on our own

(1) 如何使用java程式碼在Google搜尋引擎中:
→ 最初還沒學到，不過我們最後在lab11後學會了這項能力。

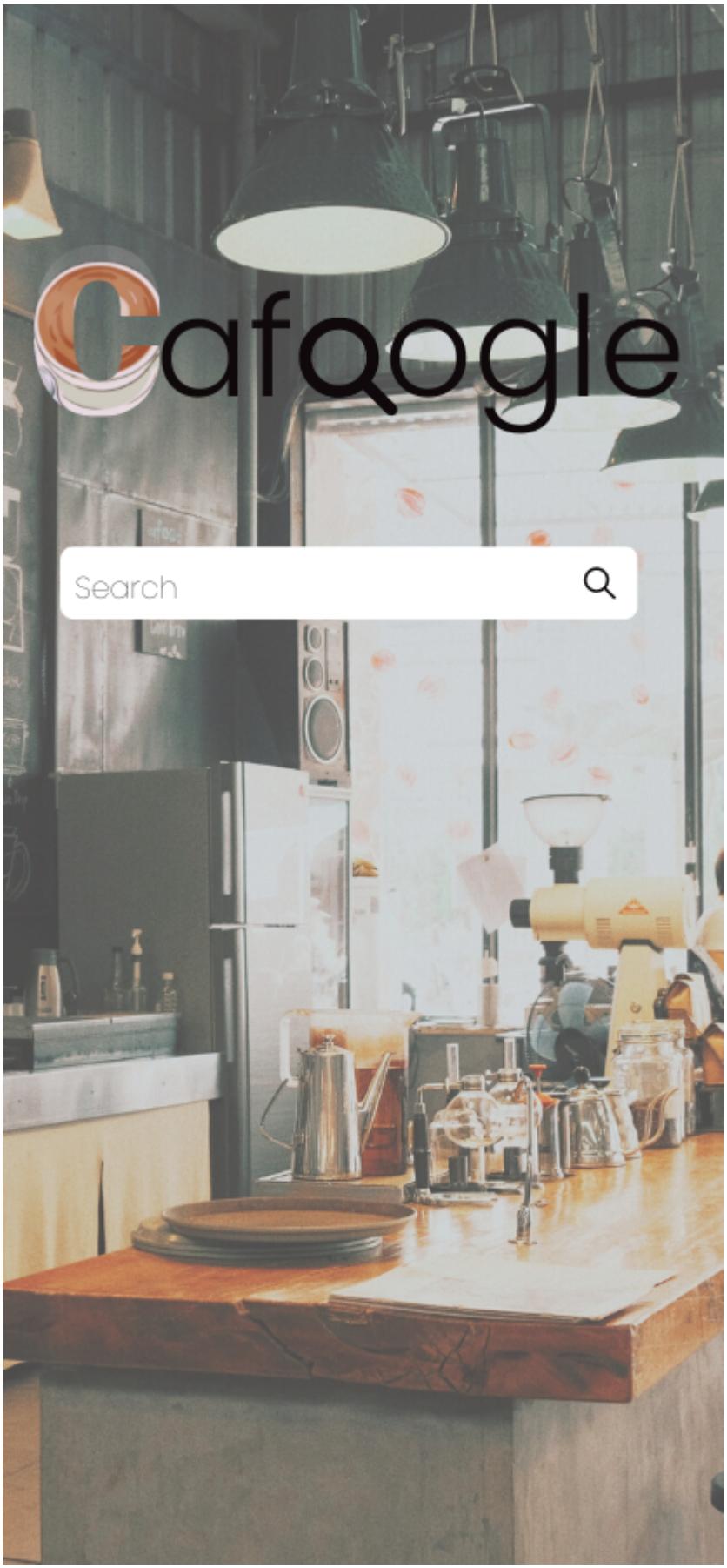
(2) 如何使用java程式碼在網站中近進一步取得相關搜尋關鍵字:
→ 我們最後是使用爬蟲，不過這樣的方法得出的關鍵字還是必須經過人工篩選，將不必要的手動輸入到負分關鍵字的地方，將蠻常出現且重要的手動輸入到正分的關鍵字中。

(3) 關鍵字的score都是用我們自己手動人工分配的，有沒有什麼方法能利用程式碼去感應該關鍵字和主題連結的重要性和相關性並且賦予分數。

6. Demo

1. Website

2. Application Interface（預期成果）



Cafoogle

Search

You've searched taiwan cafe

Sort by Most Relevant ▾

https://www.tripadvisor.com/Restaurants-g293913-c8-Taipei.html

THE 10 BEST Cafés in Taipei (Updated 2023) - Tripadvisor

Cafés in Taipei · 1. Rilakkuma CafE · 2,688 reviewsOpen Now · 2. Moomin Cafe · 541 reviews · 3. Fika Fika Cafe · 140 reviewsOpen Now · 4. Toasteria Cafe Yong Kang.

https://www.yelp.com/biz/taiwan-cafe-milpitas-2

TAIWAN CAFE - 1180 Photos & 373 Reviews - Yelp

Recommended Reviews - Taiwan Cafe ; 568 N Abel St. Milpitas, CA 95035 ; (408) 586-8885 ; Visit Website. https://www.taiwancafemilpitas.com ; Full menu ; More Info.

評分：3.5 · 373 則評論 · 消費：$11-30

https://www.yelp.com/biz/taiwan-café-boston-2

Taiwan Café - 34 Oxford St, Boston, MA - Yelp

COVID update: Taiwan Café has updated their hours, takeout & delivery options. Rated 3.6/5 with 1280 reviews of Taiwan Café "Ever had steamed pork/crab ...

評分：3.5 · 1,280 則評論 · 消費：$11-30