二、研究計畫內容

（一）摘要

本研究的目的是要利用網頁爬蟲程式，來進行網頁內容的檢測。利用 Python 語言撰寫爬蟲程式，並將結果儲存在 SQLite 資料庫中。本研究將進行的檢測項目包括以下幾項。一、檢查網頁超鏈結的有效性，檢測出已經失效或連結到不存在網址的超鏈結。二、檢查網頁上所放置的檔案格式，檢測出提供使用者下載的檔案格式及數量。三、檢查網頁上是否包含有個人資料，檢測網頁上是否放置如身份證字號或電話號碼等的個人資料。這些檢測出來的結果，可以提供給網頁管理人員做為改善的參考。

（二）研究動機與研究問題

現在網頁的內容愈來愈多也愈來愈龐大，對於網頁內容的管理者來說，要了解網頁內容的狀況是一項極大的挑戰，如果沒有一個自動化的工具來協助網頁內容管理者的話，勢必造成網頁內容管理者的困擾。

由於 Python 程式語言的興起，再加上 Python 程式語言擁有眾多的套件，其中的一項套件就是網頁資料擷取與分析，也就是所謂的網頁爬蟲。利用這一個套件，可以很輕易地自動擷取網頁的內容。因此，本研究就是希望利用 Python 的網頁爬蟲程式，來自動進行網頁內容的擷取，將擷取的結果提供給網頁管理人員參考。

一般而言，網頁最常遇到的一個狀況，就是點了某一個超鏈結之後，出現無法顯示網頁或是檔案不存在，這是因為當初設計超鏈結時，是鏈結到另一個網頁內容，然而隨著時間的經過，網頁管理者可能已經移除該網頁內容但是超鏈結依舊存在，因此會造成超鏈結失效。如果網頁管理人員要把失效的超鏈結找出來的話，就必須要一個一個的點進去，這是一件相當費時費工的工作。如果可以透過網頁爬蟲程式，自動找到某個頁面上所有的超鏈結，並且自動嘗試鏈結看看超鏈結是否有效，則可以幫助網頁管理人員省下不少的工作。

其次，由於目前國發會正在努力推動開放文件格式(Open Document Format，ODF)，其優點為格式開放、跨平台、跨應用程式的特性、可與國際間交換、適用於長久保存並可避免版本升級等問題。在推動過程中，其中有一項要求是各機關在網頁上提供可編輯的檔案，必須是符合 ODF 格式。也就是說，原本放在網頁上提供給使用者下載編輯的檔案，不能是特定商業軟體的格式(例如微軟的 Word、Excel 等等)。對於網頁管理人員來說，如果要一頁一頁地檢查，又是一件費時費工的工作。如果可以利用網頁爬蟲程式，自動找出放置在網頁上供使用者下載的檔案格式，則可以減輕網頁管理人員的負擔。

此外，個人資料保護法的通過，讓大家對於個資保護的工作更為警慎，如果

不小心洩漏個資，則可能會造成不必要的困擾。網頁管理人員經過多年的維運，已經放置相當多的內容或檔案在公開的網頁上，其中很有可能包含了個人資料。如果要網頁管理人員一一檢查，又必須耗一段時間。如果能有自動化的程式，可以自動檢查網頁上是否有個資，可以減輕網頁管理人員的工作。個人資料可能包括身份證字號或是電話號碼等，在 Python 中可以利用正規表示方式，輕易地將個人資料找出。個人資料除了以網頁方式呈現之外，也可能是在檔案內(包括 Word、Excel、PowerPoint、或是 PDF)，Python 程式語言也可以自動開啟這些檔案，檢查看看裡面是否含有個人資料。

本研究是基於以上幾個動機，希望開發出一套網頁爬蟲程式，來進行網頁內容檢測，最後提供給網頁內容管理人員參考。

(三) 文獻回顧與探討

網路已經盛行很長的時間了，網路上已有很多爬蟲程式，每個程式都有各自提供的功能可以給我們研究的參考。

Sébastien Ailleret 獨立開發『Larbin』，該軟體主要在 Linux 進行開發，且以 C/C++開發語言，此軟體目的是能夠跟蹤頁面的 url 進行擴展的抓取，最後為搜尋引擎提供廣泛的數據來源。而 Larbin 只是單純的爬蟲程式，其他功能都沒有提供，像是儲存到資料庫等等皆不支援。

Yong-Siang Shih 開發『Scrapy + Python 3: PTT 資料抓取與分析』，利用 Scrapy 程式庫結合 Python 程式，固定間隔一段時間自動抓取文章與下面留言數，並做出用與分析來收集文章下面討論內容，將內容利用圖表顯示。

陳威宇等人開發『Crawlzilla』，利用 Java 與 JavaScript SHELL 在 Linux 上運作的爬蟲程式，除了基本的 HTML 外，還能分析網頁上的文件，如 doc、pdf、ppt 等多種文件格式，讓搜尋引擎不只是網頁搜尋引擎，而是網站的完整資料索引庫。

『Sinawler』為開源軟體原名為『新浪微博爬蟲』以 C#.NET 為開發語言，利用 SQL Server 做為後台資料庫，主要功能提供給新浪微博使用者，以該用戶的關注人、粉絲為線索，沿人脈關係搜集用戶基本信息、微博數據、評論數據。

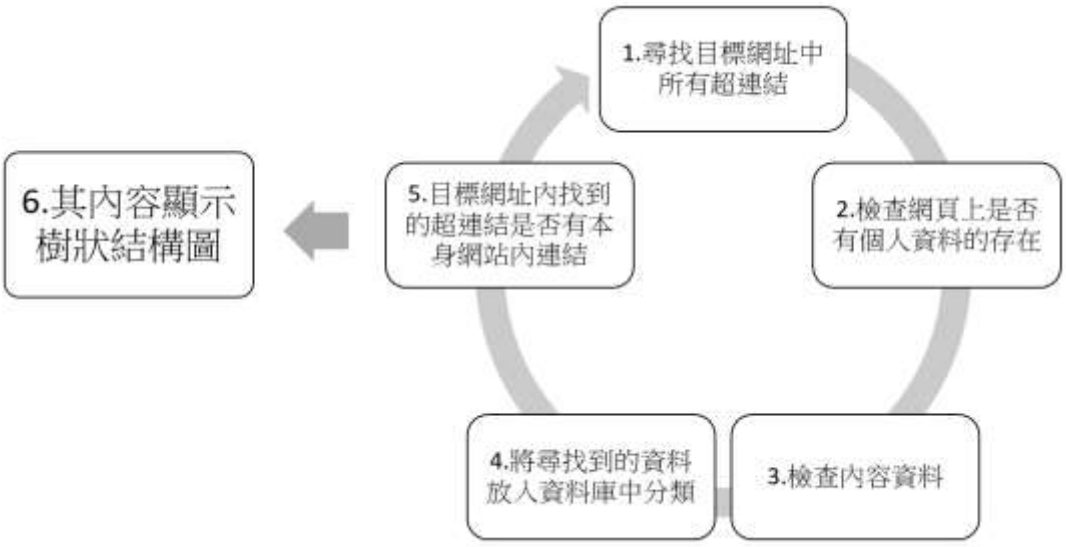
Vnikic 開發『Web-Harvest』，Web-Harvest 是一個 Java 開源 Web 數據抽取工具。它能夠收集指定的 Web 頁面並從這些頁面中提取有用的數據。Web-Harvest 主要是運用了像 XSLT、XQuery 正則表式等這些技術來實現對 text/xml 的操作。

以上都是爬蟲程式的應用，而本研究注重於超鏈結上的檢查，抓取超鏈結內容，將超鏈結網頁內容進行檢查，找出是否有個人資料的存在，收集相關資訊，提供將資料存放資料庫之功能，將全部內容使用圖表方式呈現。

（四）研究方法及步驟

4.1 研究方法

本程式以 Python 為開發語言，利用 Python 的 BeautifulSoup 套件作為爬蟲功能之開發，將找到之內容利用正規表示法做各種檢查、比對，如果內容為 doc、ppt、pdf 等文件格式，可以自動開啟檔案檢查文件中內容，使用 SQLite 資料庫儲存相關資料，最後將內容顯示出來。運作步驟如圖一所示。



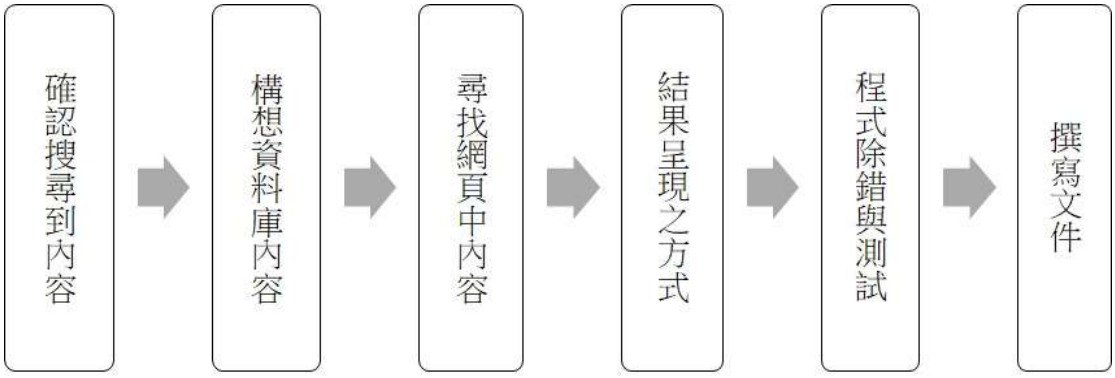
圖一、系統運作步驟流程圖

系統運作步驟說明如下：

1. 檢查網頁上是否有個人資料的存在。
2. 使用爬蟲程式尋找目標網址中所有超鏈結。
3. 若超鏈結內容檔案為 doc、ppt、pdf 等文件格式，自動開啟檔案檢查檔案中內容。
4. 將尋找到的超鏈結資料檢查相關資訊(例如:檔案名稱、格式與超連結是否失效)放入 SQLite 資料庫中分類。
5. 若從目標網址內尋找到的超鏈結為本身網站內連結，將其鏈結當成新目標網址，重複第一到第四步驟。
6. 最後其內容利用繪圖程式顯示樹狀結構圖。

4.2 研究步驟

研究步驟如圖二所示，說明如下。



圖二、研究步驟流程圖

1. 研究網頁 HTML 語言之架構內容，確定爬蟲程式搜尋到的內容。

(一)利用 BeautifulSoup 套件作為爬蟲程式開發，以下程式碼為利用本套件中使用 requests 獲取網頁原始碼，利用 BeautifulSoup 解析 HTML 文件。

```
import requests
from bs4 import BeautifulSoup
import re
response = requests.get("目標網址")
soup = BeautifulSoup(response.text,'html.parser')
```

(二)解析 HTML 文件後，可以單獨列出各項資料。以下程式碼以列出所有網頁超連結 href 屬性為目的。

```
for x in soup.findAll('a',href=re.compile('doc')):
    print(x['href'])
```

2. 設計資料庫中資料表內欄位與資料型態。

(一)使用 Python 內建資料庫功能，只要 import sqlite3 即可使用。以下程式碼建立資料庫及表單中欄位，ID 代表著該筆資料專屬辨別、Link 為該筆網址、upperID 為該筆上層網址的專屬 ID、Survival 為該筆連結是否存亡、personal 為檢查網頁中是否有個人資料的存在。

```
import sqlite3
conn=sqlite3.Connection('test2.sqlite')
sqlstr='CREATE TABLE IF NOT EXISTS href \
("ID" TEXT PRIMARY KEY NOT NULL ,"Link" TEXT,"upperID" TEXT,"Survival"
TEXT,"personal" TEXT)'
conn.execute(sqlstr)
```

(二)使用資料庫，當中操作不能少了新增資料、修改、刪除與列出資料等功能。

1. insert 新增程式碼

```
sqlstr="insert into href
values({},{},{},{},{})".format(ID,Link,upperID,Survival,personal)
conn.execute(sqlstr)
conn.commit()
```

2. update 更改程式碼

```
sqlstr ="update href set Link = {} where Link = {}".format(new,old)
conn.execute(sqlstr)
conn.commit()
```

3. delete 刪除程式碼

```
sqlstr = "delete from href where ID={}".format(id)
conn.execute(sqlstr)
conn.commit()
```

4. fetchone 尋找單筆資料程式碼

```
find='select * from href where ID={} '.format(id)
cursor = conn.execute(find)
row = cursor.fetchone()
```

5. fetchall 列出所有內容程式碼

```
cursor = conn.execute(find)
rows = cursor.fetchall()
```

3. 研究利用程式碼開啟文件檔案(Word、Excel、PowerPoint、PDF)

(一)開啟 word 檔案要額外安裝 python-docx 套件。使用該套件讀取檔案時需要轉換成 docx 檔案才可成功讀取。

```
import docx
```

使用參數 16 表示將 doc 轉換成 docx

```
doc.SaveAs(r"D:\\test2.docx",16)
```

(二)讀寫 Excel 分別要安裝兩種套件，xlwt 其功能是使程式可以寫入 Excel 資料，xlrd 其功能是使程式可以讀取。

```
import xlrd
import xlrt
```

(三)操作 PowerPoint 功能需要安裝 Window com，安裝後即可以將 PowerPoint 內容輸出到指定文件

```
import win32com
from win32com.client import Dispatch, constants
```

(四)對 pdf 的操作前需要額外安裝 PDFMiner3K 的套件，使用後 Python 會把 pdf 轉換成字串，然後用 StringIO 轉換成文件對象。

```
from pdfminer.pdfinterp import PDFResourceManager, process_pdf
from pdfminer.converter import TextConverter
from pdfminer.layout import LAParams
from io import StringIO
```

4. 研究用正規表示法尋找網頁中是否有個人資料。

```
id = re.findall(r'[A-Z][1-2][0-9]{8}',res2)
cel = re.findall(r'[09]{2}[0-9]{8}',res2)
tele = re.findall(r'[0][2-8][0-9]{8}',res2)
```

5. 構想最後結果呈現之方式。

6. 進行程式除錯與測試。

7. 完成文件之撰寫。

（五）預期結果

預期執行程式的主畫面如圖三所示，爬蟲程式進行搜尋畫面如圖四所示，查詢結果畫面如圖五所示。

圖三、預期主畫面

圖四、預期搜尋畫面

圖五、預期查詢結果畫面

（六）參考文獻

[1] 每日頭條-33 款可用來抓數據的開源爬蟲軟體工具--
https://kknews.cc/tech/bx2ml6.html

[2] larbin -- https://github.com/ictxiangxin/larbin

[3] Scrapy + Python 3: PTT 資料抓取與分析--
http://city.shaform.com/blog/2016/02/28/scrapy.html

[4] crawlzilla -- https://code.google.com/archive/p/crawlzilla/

[5] sinawler -- https://code.google.com/archive/p/sinawler/

[6] web-harvest -- http://web-harvest.sourceforge.net/

（七）需要指導教授指導內容

1. 專題研究相關方向、內容、問題。

2. 程式寫作相關想法、問題、演算法。

3. 介面架設相關觀念、問題。

4. 正式文件撰寫。