Acemap report

daxingcheng 515030910485

May 27, 2018

1 Introduction

The content of this project is that in acemap project, in the case of known scholars home page url, analyze the scholar's home page content, analyze the meaning of each field, extract scholars information, so as to perfect the homepage of scholars in the acemap.

Introduction of Acemap: Acemap is different from the traditional academic search engines as a new era of academic search tools, it will show the relationship of each paper, author in the form of a map, let people can intuitive understanding the relationship between each paper from the map.

2 Demand analysis

This paper is divided into two parts, one is how to use the NER (named entity recognition) to do information extraction from scholars home page, the second is to use web crawler to get hyperlinks on their homepages, and associated url is obtained by clustering. So that to increase the range of extracting information, and to improve the accuracy.

The extraction target is the author's name, address, email, home page and career.

3 Specific implementation of part I

3.1 NER introduction:

NER (Named Entity Recognition), also known as special name Recognition, is a common task in natural language processing, and it is widely used. A named entity usually refers to an entity that has special meaning or is highly referential in a text, usually including a person's name, place name, institution name, time, proper name, etc. NER system extracts the above entities from unstructured text and identifies more categories of entities according to business requirements, such as product name, model, price, etc. So the concept of an entity can be

very broad, and any particular piece of text that is required by the business can be called an entity.

3.2 introduction of spaCy:

SpaCy is a Python Natural Language Processing tool kit, which was born in the middle of 2014. It is called "industrial-strength Natural Language Processing in Python", and it is a Python NLP tool kit with Industrial Strength. Cython is widely used in spaCy to improve the performance of related modules, which is different from the more academic Python NLTK, so it has practical value in the industry application.

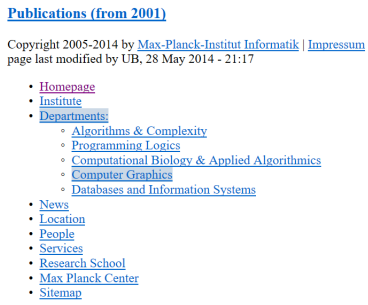
The sample code for NER using spaCy is as follows:

3.3 problem A

The career part of scholars will cause great trouble to the extraction of information, and the information of the career part is very messy, including the school name, address and other information, causing interference.

The solution is using the method of classification. The text is divided into three parts: academic career, scholar introduction and others. We only use NER on the part of scholar introduction.

The classification method is to extract the characteristics of each paragraph. The features of office and Email are the introduction of scholars, and the information of year and awards are the career of scholars



3.4 problem B

Some information of scholars does not exist directly on the page of the scholar's homepage, but provides links. The second part of this paper solves this problem.

2005 -	Chair, European Association for Computer Graphics Eurographics)
2004 -	Election to DFG Fachkollegium Informatik
2004	International Review Panel, Computer Science Departments
	TU Delft, Univ. Leiden, The Netherlands
2004 -	Executive Committee, Solid Modeling Association (Steering Committee
	ACM Solid Modeling Symposium)
2003 -	Co-Director, Max Planck Center for Visual Computing and Communication
	Stanford/Saarbrücken
2003 - 2004	Vice Chair, European Association for Computer Graphics (Eurographics)
2001 - 2003	Managing Director, Max-Planck-Institute for Computer Science
2000 - 2003	Perspectivenkommission CPTS, Max-Planck-Society
1998 -	Scientific Advisory Board, Leibniz Minerva Center, Hebrew University, Jerusalem, Israel
1998 - 1999	Stv. Sprecher, DFG Sonderforschungsbereich 603, Model Based Analysis and

4 Specific implementation of part ii

use beautifulsoup to extract hyperlinks from web pages

cluster analysis was carried out for all the urls obtained through the crawler, together with the original urls. All the urls in the category of the original urls are the related urls of the original urls.

original url: http://hanj.cs.illinois.edu/

related url:

```
http://www.cs.uiuc.edu/
http://www.uiuc.edu/
http://www.uiuc.edu/cgi-bin/where_is?bldg=dcl&room=2123
http://dml.cs.uiuc.edu/
http://dais.cs.illinois.edu/
http://senate.illinois.edu/ep0944.html
https://webmail.illinois.edu/owa/
https://my.cs.illinois.edu/
http://hanj.cs.illinois.edu
http://hanj.cs.illinois.edu/projs/social_media.htm
http://hanj.cs.illinois.edu/projs/structnet.htm
http://hanj.cs.illinois.edu/bkl
http://hanj.cs.illinois.edu/projs/csbibcube.htm
http://hanj.cs.illinois.edu/projs/eventcube.htm
http://hanj.cs.illinois.edu/projs/patternmine.htm
http://hanj.cs.illinois.edu/projs/infonet.htm
http://hanj.cs.illinois.edu/projs/biobdi.htm
http://hanj.cs.illinois.edu/projs/streamine.htm
http://hanj.cs.illinois.edu/projs/cps.htm
http://hanj.cs.illinois.edu/projs/movemine.htm
http://hanj.cs.illinois.edu/projs/conflicts.htm
http://illimine.cs.uiuc.edu/
http://hanj.cs.illinois.edu/others.html
http://citeseer.ist.psu.edu
http://dais.cs.uiuc.edu/
http://www.cs.uiuc.edu/
http://www.uiuc.edu/
>>>
```

5 conclusion

Through this project, I learned relevant information of NLP, understood the specific implementation method of NER, and learned how to use spaCy. In addition to the academic improvement, I cooperated with other students in this project and with the help of the seniors and teachers, I learned how to work with others to complete projects.

6 code of part ii

```
import requests
from bs4 import BeautifulSoup
import json
from sklearn.cluster import KMeans

#输入url，返回同类url
def geturl(url):
    try:
        r=requests.get(url)
    except:
        return False
    soup = BeautifulSoup(r.text)
    x=url

    #特征集合
    l=[]
    for link in soup.find_all('a'):
        url=link.get('href')
        if url!=None:
            if url[0:4]=='http':
                url=url.replace('/',' ')
                url=url.split()[1].split('.')
                if 'www' in url:
                    url.remove('www')
                for j in url:
                    if j not in l:
                        l.append(j)
    if l==[]:
        return False
    url=x
    url=url.replace('/',' ')
    url=url.split()[1].split('.')
    if 'www' in url:
        url.remove('www')
    for j in url:
        if j not in l:
            l.append(j)
```

```
#特征向量
feature=[]
for link in soup.find_all('a'):
    c=link.get('href')
    if c!=None:
        if c[0:4]=='http':
            c=c.replace('/',' ')
            c=c.split()[1].split('.')
            if 'www' in c:
                c.remove('www')
            u=[0]
            u=u*len(l)
            for j in c:
                u[l.index(j)]=1
            feature.append(u)

#kmeans聚类
if len(feature)<=3:
    n=len(feature)
else:
    n=3
clf = KMeans(n_clusters=n)
s = clf.fit(feature)
```

```
#输出同类url
c=x
c=c.replace('/',' ')
c=c.split()[1].split('.')
if 'www' in c:
    c.remove('www')
u=[0]
u=u*len(l)
for j in c:
    u[l.index(j)]=1
n=0
m=[]
for link in soup.find_all('a'):
    c=link.get('href')
    if c!=None:
        if c[0:4]=='http':
            if  clf.labels_[n]==clf.predict([u]):
                m.append(c)
            n=n+1
return m
```