# CMSC5733 Social Computing

## Tutorial 1: Python and Web Crawler

Shenglin Zhao

The Chinese University of Hong Kong

slzhao@cse.cuhk.edu.hk

# Tutorial Overview

- Python basics and useful packages
- Web Crawler

# Why Python?

- Simple, easy to read syntax
- Object oriented
- Huge community with great support
- Portable and cross-platform
- Powerful standard libs and extensive packages
- Stable and mature
- FREE!

# Python Programming Language

- Download Python 2.7.10 or 3.4.3 at

  http://www.python.org/download/

- Set up tutorials

  http://www.youtube.com/watch?v=4Mf0h3HphEA

  Or

  https://developers.google.com/edu/python/set-up

# Installing Packages

- Tools for easily download, build, install and upgrade Python packages
  - easy_install
    - Installation instruction: https://pypi.python.org/pypi/setuptools/1.1.4#installation-instructions
  - pip
    - In terminal input: easy_install pip
- Distribution package : Anoconda, http://www.continuum.io/downloads

# Python Packages

- mysql-python package for MySQL
  - Quick install
    - ✓ Download: http://sourceforge.net/projects/mysql-python/
    - ✓ easy_install mysql-python or pip install mysql-python
  - MySQL Python tutorial:
    - ✓ http://zetcode.com/db/mysqlpython/
  - Example

```
# remember to install MySQLdb package before import it
import MySQLdb as mdb
# connect with mysql
con = mdb.connect(host='localhost',user='root',passwd='',db='limitssystem')
# get connection
cur = con.cursor()
sql = "select f_id,f_name,f_action from function"
# execute sql
cur.execute(sql)
# get the result
result = cur.fetchall()
for r in result:
        f_id = r[0]
        f_name = r[1]
        f_action = r[2]
        print f_id,unicode(f_name,"utf-8"),f_action
```

# Python Packages

- urllib2 package
  - Reading a web page
  - Example:

```
import urllib2
# Get a file-like object for the Python Web site's home page.
url = "http://www.python.org"
try:
        f = urllib2.urlopen(url)
        # Read from the object, storing the page's contents in 's'.
        s = f.read()
        print unicode(s,"utf-8")
        f.close()
except IOError:
        print ‘problem reading url:’,url
```

# Python Packages

- BeautifulSoup package
  - Extracting HTML data
  - Quick Install
    - easy_install BeautifulSoup or pip install BeautifulSoup
  - Example

```
from BeautifulSoup import BeautifulSoup
html = '<html><body><p class="title">Title</p></body></html>'
soup = BeautifulSoup(html)
print soup.p                # <p class="title">Title</p>
print soup.p["class"]       # title
print soup.p.string         # Title
```

# Python Packages

- Scrapely package
  - What is Scrapely:

    Scrapely is a library for extracting structured data from HTML pages. Given some example web pages and the data to be extracted, scrapely constructs a parser for all similar pages.
  - https://github.com/scrapy/scrapely
  - Example

```
from scrapely import Scraper
s = Scraper()  #instantiating the Scraper class
#proceed to train the scraper by adding some page and the data you expect to
#scrape from there
url1 = 'http://pypi.python.org/pypi/w3lib/1.1'
data = {'name': 'w3lib 1.1', 'author': 'Scrapy project', 'description': 'Library of web-
related functions'}
#train the data
s.train(url1, data)
url2 = 'http://pypi.python.org/pypi/Django/1.3'
print s.scrape(url2)
```

# Python Packages

- Scrapy Package
  - What is Scrapy:
    - Scrapy is a fast high-level screen scraping and web crawling framework, used to crawl websites and extract structured data from their pages.
  - http://scrapy.org/
  - Quick Install:
    - easy_install -U Scrapy or pip install Scrapy

# Python Programming Language

- Video tutorials for python
  - http://www.youtube.com/watch?v=4Mf0h3HphEA
  - http://www.youtube.com/watch?v=tKTZoB2Vjuk
- Document tutorials for python
  - http://www.learnpython.org/
  - https://developers.google.com/edu/python/ (suggested!)

# Web Crawler

- Definition:
  - A Web crawler is a computer program that browses the World Wide Web in a methodical, automated manner. (Wikipedia)
- Utilities:
  - Gather pages from the Web.
  - Support a search engine, perform data mining and so on.
- Object:
  - Text, video, image and so on.
  - Link structure.

# Features of a crawler

- Must provide:
  - Robustness: spider traps
    - Infinitely deep directory structures: http://foo.com/bar/foo/bar/foo/...
    - Pages filled a large number of characters.
  - **Politeness:** which pages can be crawled, and which cannot
    - Explicit: robots.txt (later example)
    - Implicit: do not visit the same page too frequently

# Features of a crawler (Cont'd)

- Should provide:
  - Distributed
  - Scalable
  - Performance and efficiency
  - Quality
  - Freshness
  - Extensible

# Robots.txt

- Protocol for giving spiders ("robots") limited access to a website, originally from 1994
  - www.robotstxt.org/wc/norobots.html
- Website announces its request on what can(not) be crawled
  - For a server, create a file `/robots.txt`
  - This file specifies access restrictions

# An example of robots.txt

- No robot should visit any URL starting with "/yoursite/temp/", except the robot called “searchengine":

```
User-agent: *
Disallow: /yoursite/temp/

User-agent: searchengine
Disallow:
```

# Web Crawler

- Demo for crawling book.douban.com
  - Crawling all the book information
  - Book link: http://book.douban.com/subject/24753751/
  - Douban API: https://api.douban.com/v2/book/24753751
  - Steps: Breadth first crawling all the webpages, get all the book link and book id. According to every book id, get douban API information.
  - Demo

# Web Crawler

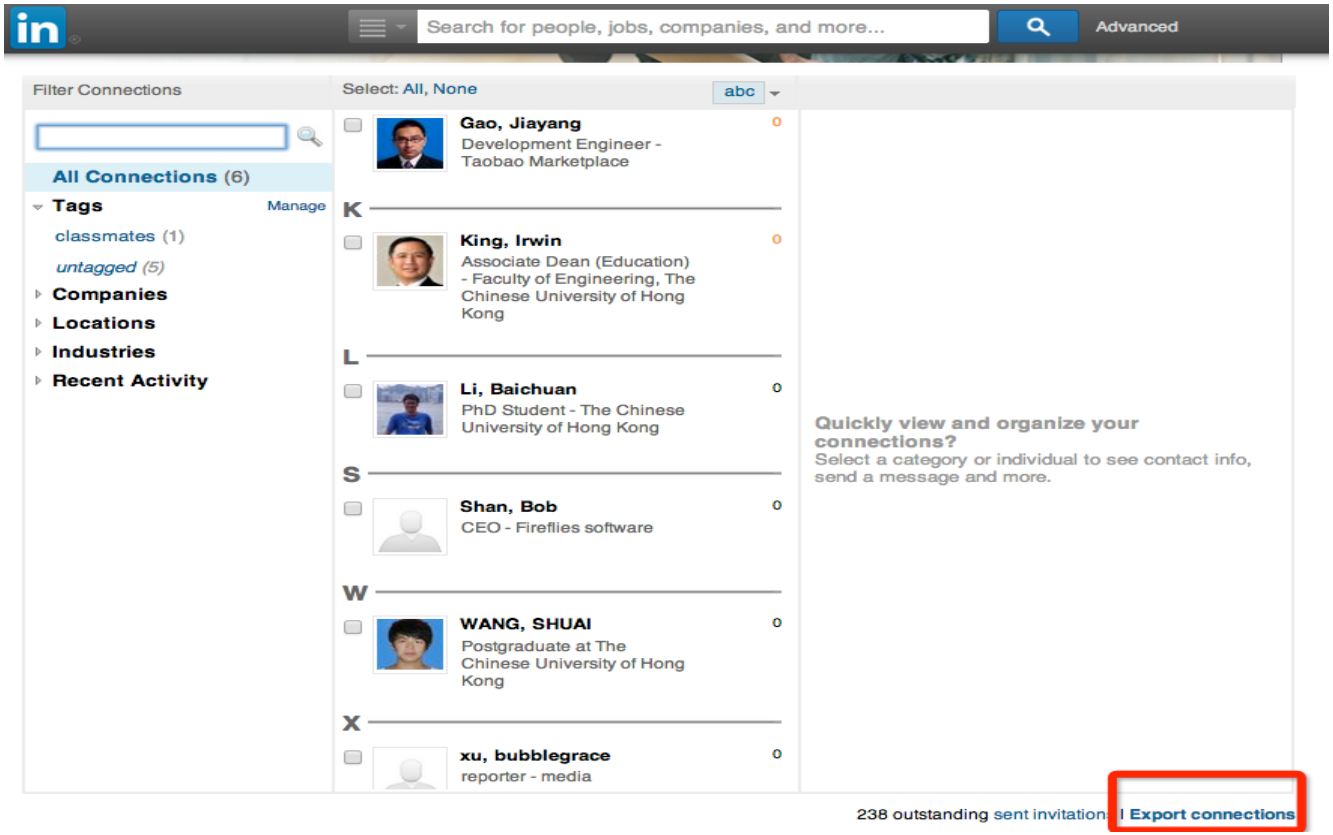
- Python package for twitter API
  - Twitter libraries for Python: https://dev.twitter.com/docs/twitter-libraries
  - Python-twitter for demo: https://dev.twitter.com/docs/twitter-libraries
  - Applying for oAuth authentication: https://dev.twitter.com/apps/new
  - Demo

# Web Crawler

- Python package for linkedIn API
  - Website: https://github.com/ozgur/python-linkedin
  - linstallation:
    - $ pip install python-linkedin
    - $ pip install requests
    - $ pip install requests_oauthlib
  - Applying for Authentication
    - https://developers.linkedin.com/documents/authentication

# Web Crawler

- An easy way to extract my connection profile from linkedin



# Web Crawler

- Tools for crawling Sina Weibo
  - Using Sina Weibo API: http://open.weibo.com/wiki/%E5%BE%AE%E5%8D%9AAPI
  - Using tools from cnpameng.com: http://www.cnpameng.com/
  - Download data source from datatang.com: http://www.datatang.com

# Web Crawler

- Scrapinghub
  - Providing web crawling and data processing solutions
  - http://scrapinghub.com/

# References

- http://www.python.org
- https://developers.google.com/edu/python/
- https://github.com/scrapy/scrapely
- http://scrapy.org/
- https://dev.twitter.com/docs/twitter-libraries