LING 408/508: Programming for Linguists

Lecture 16
October 26th
Adminstrivia

• Homework 6 graded

• Homework 7 out today
  – if possible do it by next time
  – if not, next Saturday by midnight
Today's Topics

• regexper.com
• Homework 6 Review
regexper.com

• (recommended by Gus Hahn-Powell)
• Visualize regular expressions as an automaton:

\b.+ed\b

Display

\b(.+)ed\b

Display

\b word boundary \rightarrow any character \rightarrow \text{group } #1 \rightarrow ed \rightarrow word boundary
Homework 6 review

• From UIUC POS Tagger demo: sample.txt

Helicopters will patrol the temporary no-fly zone around New Jersey's MetLife Stadium Sunday, with F-16s based in Atlantic City ready to be scrambled if an unauthorized aircraft does enter the restricted airspace.

Down below, bomb-sniffing dogs will patrol the trains and buses that are expected to take approximately 30,000 of the 80,000-plus spectators to Sunday's Super Bowl between the Denver Broncos and Seattle Seahawks.

The Transportation Security Administration said it has added about two dozen dogs to monitor passengers coming in and out of the airport around the Super Bowl.

On Saturday, TSA agents demonstrated how the dogs can sniff out many different types of explosives. Once they do, they're trained to sit rather than attack, so as not to raise suspicion or create a panic.

TSA spokeswoman Lisa Farbstein said the dogs undergo 12 weeks of training, which costs about $200,000, factoring in food, vehicles and salaries for trainers.

Dogs have been used in cargo areas for some time, but have just been introduced recently in passenger areas at Newark and JFK airports. JFK has one dog and Newark has a handful, Farbstein said.
Homework 6 review

• Question 1: write a regex that finds all the **acronyms** in the article.

![Regex Example]

- String: Helicopters will patrol the temporary no-fly zone arou
- Regex: `[A-Z][A-Z]+`
- Global match (g): checked
- Matches: TSA, TSA, JFK, JFK
Homework 6 review

• Question 2: write a regex that finds all and only the **numeric items** in the article.
Homework 6 review

• Question 3: write a regex that finds all Noun-Noun compounds
Homework 6 review

• Question 4: write a regex that finds all the main verbs (exclude auxiliaries) in the article.

\[(\text{will}|\text{to}|\text{does})\ (\text{[a-z]+})|([\text{a-z}]+\text{ed}|\text{said}|\text{has})\b\]
Homework 6 review

• Question 5: write a regex that finds all the passive verbs

Regex: `(be|been|are) \(\w+ed\)\b`

be scrambled, be, scrambled
are expected, are, expected
been used, been, used
been introduced, been, introduced
The server side

• So far, all the web programming has been **client-side** only
  – i.e. the Javascript code is running on the browser

• Let's build a webserver
  – the client-side will send form information to the **server-side** to be processed
Building a Webserver

• We'll use cgi-bin and bash scripts initially ...

• Apache2 is the most common webserver software
  — unfortunately, configuration are different on OSX and Ubuntu (we will cover both here)
Common Gateway Interface (CGI)

- The glue between a webserver and programs that run on the computer (server) hosting the webserver

1. Normally, a webserver sends out **static webpages** in response to (URL) requests from a client (your web browser).
2. Sometimes, we want the **request to run a program** (a script or binary) on the server that does some computation and generates some result to be displayed on the client (as a webpage).
Apache Webserver on OSX

Commands to be entered at a Terminal

- Apache version (OSX 10.11 El Capitan):
  - httpd -v (/usr/sbin/httpd)
  - Server version: Apache/2.4.16 (Unix)
  - Server built: Aug 22 2015 16:51:57

- Apache webserver control:
  - sudo apachectl start (/usr/sbin/apachectl)
  - sudo apachectl stop
  - sudo apachectl -k restart (after configuration change)
  - apachectl configtest (check configuration)
  - Syntax OK
  - ps -ax | grep httpd
  - 15709 ?? 0:00.09 /usr/sbin/httpd -D FOREGROUND
  - 15713 ?? 0:00.00 /usr/sbin/httpd -D FOREGROUND
  - 15715 ttys000 0:00.00 grep httpd
  - sudo apachectl stop
  - ps -ax | grep httpd
  - 15723 ttys000 0:00.00 grep httpd
Apache Webserver on OSX

- `sudo apachectl start`
- On a browser, enter: [http://localhost/](http://localhost/)

Safari Can’t Connect to the Server

Safari can’t open the page “localhost” because Safari can’t connect to the server “localhost”.

It works!
Apache Webserver on OSX

Default static webpage storage location:

- http://localhost/
- /Library/WebServer/Documents/index.html
- Let's create index.html ourselves!
- sudo nano /Library/Webserver/Documents/index.html

(nano is a simple text editor
^ means use the Control key)
Apache Webserver on OSX

Default static webpage storage location:
  - http://localhost/
  - /Library/WebServer/Documents/index.html
  - Let's create index.html ourselves!
  - sudo nano /Library/WebServer/Documents/index.html

My MacBook Pro

File: /Library/WebServer/Documents/index.html
Apache Webserver on OSX

• Configuration file:
  – /etc/apache2/httpd.conf
DocumentRoot: The directory out of which you will serve your documents. By default, all requests are taken from this directory, but symbolic links and aliases may be used to point to other locations.

DocumentRoot "/Library/WebServer/Documents"

<Directory "/Library/WebServer/Documents">

# Possible values for the Options directive are "None", "All", or any combination of:
# Indexes Includes FollowSymLinks SymLinks ifOwnerMatch ExecCGI MultiViews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.4/mod/core.html#options
# for more information.
#
Options FollowSymLinks Multiviews
MultiviewsMatch Any
#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
# AllowOverride FileInfo AuthConfig Limit
#
AllowOverride None
#
# Controls who can get stuff from this server.
#
Require all granted
</Directory>
Apache Webserver on OSX

Static webpages

- storage locations:
  - http://localhost/~sandiway/ (no need to be superuser)
  - mkdir ~sandiway/Sites (/Users/username/Sites)
  - index.html
  - sudo nano /etc/apache2/users/sandiway.conf

create this file ...

```html
<Directory "/Users/sandiway/Sites/">
AllowOverride All
Options Indexes Multiviews FollowSymLinks
Require all granted
</Directory>
```
Apache Webserver on OSX

Static webpages

• storage locations:
  – http://localhost/~sandiway/
  – sudo nano /etc/apache2/httpd.conf

```bash
# uncomment mod_userdir.so line (remove the #)
```

```conf
#LoadModule cgi_module libexec/apache2/mod_cgi.so
#LoadModule dav_fs_module libexec/apache2/mod_dav_fs.so
#LoadModule dav_lock_module libexec/apache2/mod_dav_lock.so
#LoadModule vhost_alias_module libexec/apache2/mod_vhost_alias.so
LoadModule negotiation_module libexec/apache2/mod_negotiation.so
LoadModule dir_module libexec/apache2/mod_dir.so
#LoadModule imagemap_module libexec/apache2/mod_imagemap.so
#LoadModule actions_module libexec/apache2/mod_actions.so
#LoadModule speling_module libexec/apache2/mod_speling.so
#LoadModule userdir_module libexec/apache2/mod_userdir.so
LoadModule alias_module libexec/apache2/mod_alias.so
#LoadModule rewrite_module libexec/apache2/mod_rewrite.so
#LoadModule php5_module libexec/apache2/libphp5.so
LoadModule hfs_apple_module libexec/apache2/mod_hfs_apple.so
```
Apache Webserver on OSX

Static webpages

- storage locations:
  - [http://localhost/~sandiway/](http://localhost/~sandiway/)
  - `sudo nano /etc/apache2/httpd.conf`

```plaintext
# Include /private/etc/apache2/extra/httpd-multilang-errordoc.conf

# Fancy directory listings
Include /private/etc/apache2/extra/httpd-autoindex.conf

# Language settings
#Include /private/etc/apache2/extra/httpd-languages.conf

# User home directories
Include /private/etc/apache2/extra/httpd-userdir.conf
```

uncomment httpd_userdir.conf line
*(remove the #)*
Apache Webserver on OSX

Static webpages

- storage locations:
  - http://localhost/~sandiway/
  - sudo nano /etc/apache2/extra/httpd-userdir.conf

```bash
GNU nano 2.0.6 File: /etc/apache2/extra/httpd-userdir.conf

UserDir Sites
#
# Control access to UserDir directories. The following is an example
# for a site where these directories are restricted to read-only.
#
Include /private/etc/apache2/users/*,conf
<IfModule bonjour_module>
    RegisterUserSite customized-users
</IfModule>
```
Apache Webserver on OSX

Static webpages
- storage locations:
  - http://localhost/~sandiway/
  - sudo apachectl -k restart
  - create a file ~sandiway/Sites/index.html

```html
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html><head>
<title>Sandiway Fong</title>
<style>
span {font-family:Menlo; font-size:14}
</style>
</head>
<body>
<h1>Sandiway Fong ~/Sites</h1>

This is Users/sandiway/Sites/index.html

</body></html>
```
Apache Webserver on OSX

Static webpages

• storage locations:
  – http://localhost/~sandiway/
Apache Webserver on OSX

- `/var/log/apache2/access_log`

```plaintext
<table>
<thead>
<tr>
<th>Time</th>
<th>IP Address</th>
<th>Request</th>
<th>Status Code</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>25/0ct/2015:21:01:29</td>
<td>-</td>
<td>GET</td>
<td>200</td>
<td>215</td>
</tr>
<tr>
<td>25/0ct/2015:21:02:40</td>
<td>-</td>
<td>GET</td>
<td>404</td>
<td>207</td>
</tr>
<tr>
<td>25/0ct/2015:21:02:41</td>
<td>-</td>
<td>GET</td>
<td>304</td>
<td>-</td>
</tr>
<tr>
<td>25/0ct/2015:21:02:42</td>
<td>-</td>
<td>GET</td>
<td>404</td>
<td>207</td>
</tr>
</tbody>
</table>
```

Homework 7

• For Mac owners
  – set up Apache2 on your mac
  – http://localhost/
  – http://localhost/~yourusername/

  – Create two different index.html webpages at these locations
  – Show your system works!
  – Consult the web if things don't work...
Homework 7

• For Ubuntu owners
  – set up Apache2 in VirtualBox (*see following slides*)
  – [http://localhost/](http://localhost/)

  – Create two different index.html webpages at these locations
  – Show your system works!
Apache Webserver on Ubuntu

- Ubuntu:
  - `sudo apt-get update`
  - `sudo apt-get install apache2`

```
sandiway@sandiway-VirtualBox:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2-bin apache2-data libapr1 libaprutil1 libaprutil1-db-dbd-sqlite3
  libaprutil1-ldap
Suggested packages:
  apache2-doc apache2-suexec-pristine apache2-suexec-custom apache2-utils
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data libapr1 libaprutil1 libaprutil1-db-dbd-sqlite3
  libaprutil1-ldap
0 upgraded, 7 newly installed, 0 to remove and 125 not upgraded.
Need to get 1,268 kB of archives.
After this operation, 5,244 kB of additional disk space will be used.
Do you want to continue? [Y/n] n
```
Apache Webserver on Ubuntu

- **Ubuntu:**
  - `sudo apt-get install apache2`
Apache2 on Ubuntu

- Apache webserver:
  - `sudo apache2ctl start` (/usr/sbin/apache2ctl)
  - `sudo apache2ctl stop`
  - `sudo apache2ctl restart`

```
http://localhost/
```

Not running...
Apache2 on Ubuntu

- Apache webserver:
  - `sudo apache2ctl start` (/usr/sbin/apache2ctl)

by default it should be running...

DocumentRoot
/var/www/html
/var/www/html/index.html
Apache2 on Ubuntu

```
sendi@sendi-VirtualBox:~$ ps aux | grep apache2
sendi  4133  0.0  0.2  16184 2192 pts/4  S+   22:21  0:00 grep --color=au
to apache2
sendi@sendi-VirtualBox:~$ apache2ctl start
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message.
(13)Permission denied: AH00072: make_sock: could not bind to address [::]:80
(13)Permission denied: AH00072: make_sock: could not bind to address 0.0.0.0:80
no listening sockets available, shutting down
AH00015: Unable to open logs
Action 'start' failed.
The Apache error log may have more information.
sendi@sendi-VirtualBox:~$ ps aux | grep apache2
sendi  4139  0.0  0.2  16184 2192 pts/4  S+   22:22  0:00 grep --color=au
to apache2
```

- `ps aux | grep apache2` (looks for apache2 processes)
- `apache2ctl start` (doesn't work, not superuser)
- `sudo apache2ctl start` (correct!)
Apache2 on Ubuntu

```
sudo apache2ctl start
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message

sandiway@sandiway-VirtualBox:~$ ps aux | grep apache2
root  4145  0.0  0.4  71312  4296 ?   Ss   22:22 0:00 /usr/sbin/apache2
www-data 4146  0.0  0.6  360476  6228 ?   Sl   22:22 0:00 /usr/sbin/apache2
www-data 4147  0.0  0.6  360476  6228 ?   Sl   22:22 0:00 /usr/sbin/apache2
www-data 4148  0.0  0.6  360476  6228 ?   Sl   22:22 0:00 /usr/sbin/apache2
www-data 4149  0.0  0.6  360476  6228 ?   Sl   22:22 0:00 /usr/sbin/apache2
to apache2
```

```
sudo apache2ctl start (correct!)
ps aux | grep apache2 (looks for apache2 processes)
```
Apache2 on Ubuntu

- Master configuration file:
  - `/etc/apache2/httpd.conf`
Apache2 on Ubuntu

- cd /etc/apache2/
- grep -r "DocumentRoot" .

/etc/apache2/sites-enabled/000-default.conf
Apache2 on Ubuntu

- /etc/apache2/sites-enabled/000-default.conf
Apache2 on Ubuntu

• Logs are in directory: /var/log/apache2/
  – access.log
  – error.log
Apache2 on Ubuntu

- User web files in ~/public_html
Apache2 on Ubuntu

• To enable user web files in ~/public_html
  – sudo a2enmod userdir
  – apache2ctl –k restart
  – http://localhost/~sandiway/