ENGL 516X:
Methods of Formal Linguistic Analysis

Semester: Spring '18

Instructor: Sowmya Vajjala

Iowa State University, USA

13 Feb 2018

Class outline

- Regular Expressions
- Using regex in Python code
- Regex in Python: Practice
- Links to Regex Practice exercises

Regular Expressions

Regular Expressions

- Regular expressions are used to do do pattern based information extraction from data.
- They have their own syntax for doing pattern matching in different ways.
- They are very useful to process text and manipulate it.
- Regular expressions in python are in a module called "re" and you can use them in your code once you add a "import re" statement in your program/console.
- They can simplify a lot of your tasks, but they themselves can be very complicated.
- pythex.org - is what I will use today to explain the syntax. We will use import re in our code next week.

RegEx syntax

1. ^{} matches the beginning of a line. For example,
   - a pattern ^Th matches all lines in a text file that start with Th
2. $ matches the end of a line. For example,
   - a pattern Th$ matches all lines in a text file that end with Th
3. \s matches a white space character
4. \S matches a non-white space character.

RegEx syntax

1. . matches any character
2. * -applies to the immediately preceding character and indicates to match zero or more of the preceding character(s).
   - for example, te* matches all locations where there is a t, te, tt, tete etc.
3. + - applies to the immediately preceding character and indicates to match one or more of the preceding character(s).
   - for example, te+ matches all locations where there is a te, tete, tetete etc.
4. {} is used next to a regular expression to indicate a range of occurrences of that expression. e.g., t{1,3} matches: t,tt,ttt.

RegEx syntax - continued

The power of square brackets

1. [aeiou]- matches a single character as long as the character is in this set.
2. You can also specify ranges in square brackets. For example, [a-z0-9] matches all characters in lower case or a single digit.
3. When the first character after the square brackets is a caret (^), it works like a "not" keyword. So, [^a-z0-9] matches all characters that are not lower cased letters, and not numbers.

Escape Character

What do you do if you want to match a ? or a . which also carry a meaning in regex?

Escape Character

What do you do if you want to match a ? or a . which also carry a meaning in regex?
We "escape" them to tell regex module that these are real characters and not regex syntax. This is done using a \ character.

So, st\. is a pattern that searches for all occurences of "st." in a string.

Regex practice on http://pythex.org

Go to APLING program homepage (apling.engl.iastate.edu) and copy the welcome message there into pythex test string area. Now, try to write regex patterns to get the following:

1. All occurences of the word "is" (Not th**is**, ling**is**tics, etc. Only "is") - \bis\b
2. All occurences of the letter e, irrespective of the case. - $e|E$
3. All occurences of "es" where it occurs in the middle of the word (i.e., es should not be followed by a space, comma, fullstop etc) - [a-z]es[a-z]

RegEx: recap exercise

- What is the regular expression to select all phone-numbers from the following text:

```
University Assistance
515-294-4428 Department of Public Safety
515-294-3322 Department of Residence
515-294-5100 Facilities Planning and Management
515-294-4444 Help Van
Poison Control
800-222-1222 Iowa Poison Control Center
```

(source: `http://info.iastate.edu/emergency/`)

RegEx: recap exercise

- What is the regular expression to select all phone-numbers from the following text:

  ```
  University Assistance
  515-294-4428 Department of Public Safety
  515-294-3322 Department of Residence
  515-294-5100 Facilities Planning and Management
  515-294-4444 Help Van
  Poison Control
  800-222-1222 Iowa Poison Control Center
  ```

  (source: `http://info.iastate.edu/emergency/`)

- My expression: ([0-9]+-*)+

RegEx: recap exercise

- What is the regular expression to select all phone-numbers from the following text:

  ```
  University Assistance
  515-294-4428 Department of Public Safety
  515-294-3322 Department of Residence
  515-294-5100 Facilities Planning and Management
  515-294-4444 Help Van
  Poison Control
  800-222-1222 Iowa Poison Control Center
  ```

  (source: `http://info.iastate.edu/emergency/`)
- My expression: ([0-9]+-*)+
- Slightly complex, but more precise one: ([0-9]{3}-){2}[0-9]{4}

Using RegEx in Python

Python's "re" module

- You should import "re" module of python using the statement: import re
- Two useful functions in re module are: search() and findall().
- search() in re module is similar to the find() method for Strings, but just more sophisticated.
  - re.search("XX[0-9]",str) searches for the first occurence of "XX" followed by a digit in a string and returns the corresponding match.
- findall() returns all matches of a pattern in a string, as a list of matches.

Searching a text with RegEx

re.search() function

An example code: RegexSearchExamples.py

A short detour: What is a List?

- A list is a sequence of values. A string can be seen as a list of characters. There can be a list of strings as well.
- These values in a list are called "elements" or "items".
- Lists in Python are identified by the presence of square brackets.
- Examples:
  1. [516,515,520,540]- is a list of integers
  2. ["Python","Java","Perl","R","Ruby"]- is a list of strings.
  3. ['spam', 2.0, 5, [10, 20]]- is list with elements of various data types. There is a list inside a list too!

An Example list and its use

Look at this code:

```
str = "Look at this code"
demoList = str.split(" ") #space is here the "delimiter".
#Splits the string wherever there is a space.
print(demoList)
['Look', 'at', 'this', 'code']
print(len(demoList)
# prints the number of items in a demoList object. 4 here.
print(demoList[1])
# prints "at".
```

More on lists on thursday.

Extracting information from a text using RegEx

re.findall() function

Example of findall() and a comparison with search() - Example codes on Canvas.

What is this pattern doing? -1

```
if re.search('^X\S*: [0-9.]+', line):
    print(line)
```

What is this pattern doing? -1

```
if re.search('^X\S*: [0-9.]+', line):
    print(line)
```

What is this pattern doing? -2

```
x = re.findall('^Details:.*rev=([0-9]+)', line)
```

Work with this program

Download grepProgram.py and mbox.txt files from Canvas. Using this program, find out the number of times did:

- source@collab.sakaiproject.org receive emails.
- source@collab.sakaiproject.org appear in the text anywhere.
- email addresses from the domain iupui.edu
- Example interaction with the program:

```
Enter a regular expression: ^Author
mbox.txt had 1798 lines that matched ^Author
```

Resources for PythonRegex

1. For learning:
   - Python Docs link `https://goo.gl/TTunhz`
   - `http://regexone.com/references/python`
2. For practice:
   - `http://pythex.org/`
   - `https://regex101.com/#python`
   - `http://www.pyregex.com/`
   - `https://txt2re.com/`

Next Class

- Topics: Lists in python
- Readings: Chapter on Lists in the textbook
- Do: The other exercise at the end of the chapter on regular expressions.