INTRODUCTION TO NATURAL LANGUAGE PROCESSING
Regular Expression – Spamlord

January 17, 2018

1 Problem

Your goal in this assignment is to write a program using regular expression that extract phone numbers and email addresses.

Email is usually expressed as follows:

```
jurafsky@stanford.edu
```

Your result should return:

```
jurafsky@stanford.edu
```

But you also need to deal with more complex examples created by people trying to shield their addresses, such as the following types of examples:

```
jurafsky(at)cs.stanford.edu
jurafsky at csli dot stanford dot edu
<script type="text/javascript">obfuscate(’stanford.edu’,’jurafsky’)</script>
```

For all of the above you should return the corresponding email address:

```
jurafsky@standford.edu
jurafsky@cs.stanford.edu
jurafsky@csli.stanford.edu
```

Similarly, for phone numbers, you need to handle examples like the following:

```
TEL +1-650-723-0293
Phone: (650) 723-0293
Tel (+1): 650-723-0293
<a href="contact.html">TEL</a> +1&thinsp;650&thinsp;723&thinsp;0293
```

All of which should return the following canonical form:

```
650-723-0293
```

The standard format of the email and the phone number in the following form:

user@example.com

650-555-1234

In order to make it easier for you to do this and other homeworks, we will be giving you some data to test your code on, what is technically called a development test set. It was stored in the directory data/. This is a document with some emails and phone numbers, together with the correct answers, so that you can make sure code extracts all and only the right information.

A good regular expression if and only if the amount of email and telephone number are extracted more accurate. You will be highly appreciated.

Note that you will not know the test data set that teachers can use to test your extracted email addresses and phone numbers. Therefore in order to increase the number of correct results, students should research and deduce many representations of email addresses and phone numbers.

2 Installation

Installation environment: Python. Students can access the link `https://www.youtube.com/watch?v=4Mf0h3HphEA` to view how to install Python.

You are provided with the program to extract *python/SpamLord.py*.

You write a regular expression describing the email addresses and phone numbers according to the form $my\_first\_pat = “regexpatterns”$.

The function *process_file* takes in a file object (or any iterable of strings) and returns a list of tuples representing emails or phone numbers $(filename, type, value)$ where type is either 'e', for email, or 'p' for phone number, and value is just the actual email or phone number.

You can update the method process_file or write more another method to support this method.

Commands to execute the program:

```
cd python
python SpamLord.py ../data/dev/ ../data/devGOLD
```

It will run your code on the files contained in data/dev/ and compare the results of a simple regular expression against the correct results. The results will look something like this:

```
True Positives (4)      ############
balaji e                balaji@stanford.edu
nass e                  nass@stanford.edu
shoham e                shoham@stanford.edu
thm e                   pkrokel@stanford.edu
False Positives (1)     ############
psyoung e               young@stanford.edu
False Negatives (110)   ############
ashishg e               ashishg@stanford.edu
ashishg e               rozm@stanford.edu
ashishg p               650-723-1614
...
```

Where:

- **The true positive**: emails and phone numbers which your code correctly matches.
- **The false positive**: emails and phone numbers which your code matches but which are not correct.
- **The false negative**: emails and phone numbers which your code did not match, but which do exist in the html file.

Your goals, the, is to reduce the number of false positives and negatives to zero.

3 Submission

Submission requirements (students must comply with the following requirements, **wrong format would be no point**)

- Each student submitted a compressed folder, the folder name is the student ID number (eg. 1500000.zip)
- The folder containing files SpamLord.py