CSCE-638, Programming Assignment #1 SpamLord

Li-Hen Chen 928003907

1. How to compile and run your code.

We program the code in Python3 and execute the program under the same directory of the SpamLord.py file.

Eg.

In the terminal just type:

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

2. Results and Analysis

For extracting email, although there are some simple cases like "huangrh@cse.tamu.edu" or "huangrh at cse dot tamu dot edu", there are some complex cases such as "<script type="text/javascript">obfuscate('cse.tamu.edu','huangrh')</script>". For extracting phone number, there are some cases we have to deal with such as "(979) 862-2908 ", "979-862-2908" and "&thinsp;979&thinsp;862&thinsp;2908".

- Email Extraction

We should find out the possible email expression first in order to solve the problem and below is the basic email format I found.

```
<td class="value">ouster (followed by &ldquo;@cs.stanford.edu&rdquo;)</td>
<td class="value">teresa.lynn (followed by "@stanford.edu")</td>
<dd>		<em>melissa&#x40;graphics.stanford.edu</em>
<address>engler WHERE stanford DOM edu</address>
email: pal at cs stanford edu,
d-l-w-h-@-s-t-a-n-f-o-r-d-.-e-d-u
<dd>		<em>ada&#x40;graphics.stanford.edu</em>
Email: uma at cs dot stanford dot edu
hager at cs dot jhu dot edu
(Fedora) Server at cs.stanford.edu Port 80
<script> obfuscate('stanford.edu','jurafsky'); </script>
```

Except for the last example, we can use regular expression to divide each part. First part is characters previous “@”. There may by dash between characters so we just use a regular expression to match it. In the second part, there are some cases with “followed by” so we also use ? to filter whether the sentence contains “followed by”. In third part, there are some cases “@, @ , at , where ,&#x40;,” so we decide to set different regular expression to judge which one is contained in the sentence. For the remaining parts we use same idea to filter the cases like “., ;, dot , dt , DOM” and “.edu, .com, -e-d-u”. For the last example “obfuscate('stanford.edu','jurafky');”, we just have to set a special regular expression to deal with it.

- Phone Number Extraction

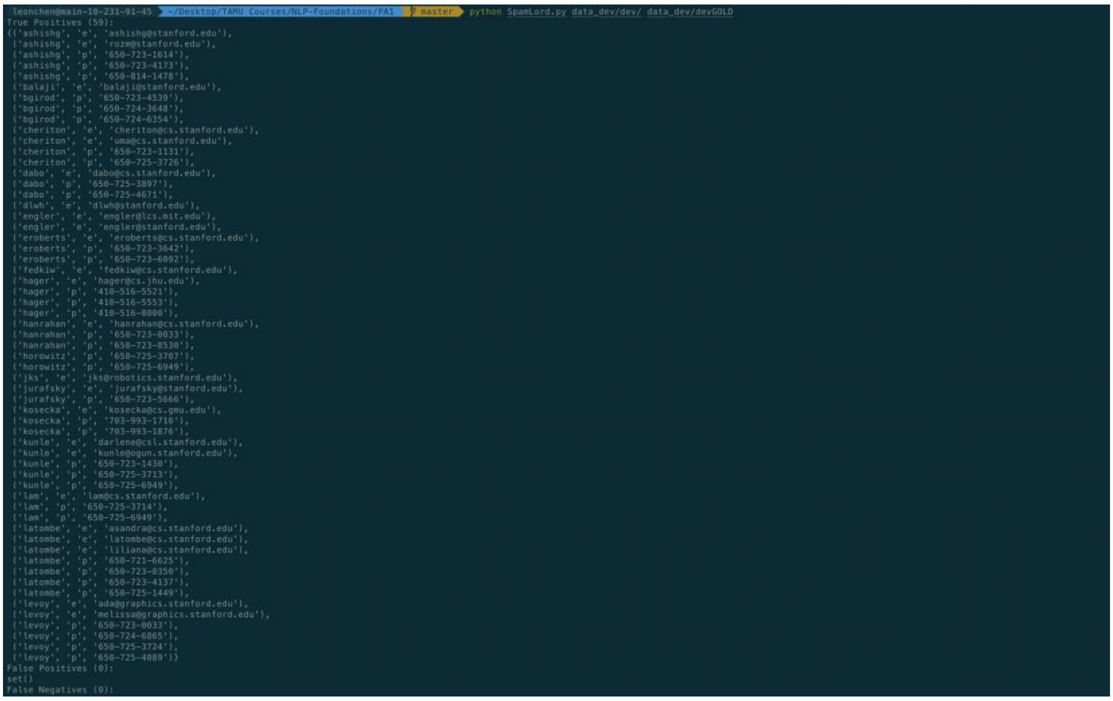
Compared to the email address, the phone number patter is relatively easy. At first part, we set a regular expression to handle to “(123)” and “123”. Then we just have to deal with whether the remaining part contains dash and replace it to the normal phone number.

- Test results on dev dataset

True Positive: 59

False Negative: 0

False Positive: 0



```
 leonchen@main-10-231-91-45  ~/Desktop/TAMU Courses/NLP-Foundations/PA1  master  python SpamLord.py data_dev/dev/ data_dev/devGOLD
True Positives (59):
{('ashishg', 'e', 'ashishg@stanford.edu'),
 ('ashishg', 'e', 'rozm@stanford.edu'),
 ('ashishg', 'p', '650-723-1614'),
 ('ashishg', 'p', '650-723-4173'),
 ('ashishg', 'p', '650-814-1478'),
 ('balaji', 'e', 'balaji@stanford.edu'),
 ('bgirod', 'p', '650-723-4539'),
 ('bgirod', 'p', '650-724-3648'),
 ('bgirod', 'p', '650-724-6354'),
 ('cheriton', 'e', 'cheriton@cs.stanford.edu'),
 ('cheriton', 'e', 'uma@cs.stanford.edu'),
 ('cheriton', 'p', '650-723-1131'),
 ('cheriton', 'p', '650-725-3726'),
 ('dabo', 'e', 'dabo@cs.stanford.edu'),
 ('dabo', 'p', '650-725-3897'),
 ('dabo', 'p', '650-725-4671'),
 ('dlwh', 'e', 'dlwh@stanford.edu'),
 ('engler', 'e', 'engler@lcs.mit.edu'),
 ('engler', 'e', 'engler@stanford.edu'),
 ('eroberts', 'e', 'eroberts@cs.stanford.edu'),
 ('eroberts', 'p', '650-723-3642'),
 ('eroberts', 'p', '650-723-6092'),
 ('fedkiw', 'e', 'fedkiw@cs.stanford.edu'),
 ('hager', 'e', 'hager@cs.jhu.edu'),
 ('hager', 'p', '410-516-5521'),
 ('hager', 'p', '410-516-5553'),
 ('hager', 'p', '410-516-8000'),
 ('hanrahan', 'e', 'hanrahan@cs.stanford.edu'),
 ('hanrahan', 'p', '650-723-0033'),
 ('hanrahan', 'p', '650-723-8530'),
 ('horowitz', 'p', '650-725-3707'),
 ('horowitz', 'p', '650-725-6949'),
 ('jks', 'e', 'jks@robotics.stanford.edu'),
 ('jurafsky', 'e', 'jurafsky@stanford.edu'),
 ('jurafsky', 'p', '650-723-5666'),
 ('kosecka', 'e', 'kosecka@cs.gmu.edu'),
 ('kosecka', 'p', '703-993-1710'),
 ('kosecka', 'p', '703-993-1876'),
 ('kunle', 'e', 'darlene@csl.stanford.edu'),
 ('kunle', 'e', 'kunle@ogun.stanford.edu'),
 ('kunle', 'p', '650-723-1430'),
 ('kunle', 'p', '650-725-3713'),
 ('kunle', 'p', '650-725-6949'),
 ('lam', 'e', 'lam@cs.stanford.edu'),
 ('lam', 'p', '650-725-3714'),
 ('lam', 'p', '650-725-6949'),
 ('latombe', 'e', 'asandra@cs.stanford.edu'),
 ('latombe', 'e', 'latombe@cs.stanford.edu'),
 ('latombe', 'e', 'liliana@cs.stanford.edu'),
 ('latombe', 'p', '650-721-6625'),
 ('latombe', 'p', '650-723-0350'),
 ('latombe', 'p', '650-723-4137'),
 ('latombe', 'p', '650-725-1449'),
 ('levoy', 'e', 'ada@graphics.stanford.edu'),
 ('levoy', 'e', 'melissa@graphics.stanford.edu'),
 ('levoy', 'p', '650-723-0033'),
 ('levoy', 'p', '650-724-6865'),
 ('levoy', 'p', '650-725-3724'),
 ('levoy', 'p', '650-725-4089')}
False Positives (0):
set()
False Negatives (0):
```

3. Any known bugs, problems, or limitations of your program

Although we did perfect on dev dataset, I did not handle the cases such as "&thinsp;979&thinsp;862&thinsp;2908". As results, there may be some errors in the test dataset. I believe the examples I listed have been dealt with perfectly.