Homework 2

Joyce LJ Woznica

# Introduction

The objective of this homework is to match email addresses and phone numbers in the provide HTML files despite attempts at obfuscation of this information. This document outlines each regular expression used with a description of the intended matches and any improvement on the matches provided.

# Description

The approach taken for this exercise is to create regular expressions and try to improve the number of “true positives” (an actual email address or phone number that ***is*** matched by the expressions provided) and reduce the number of “false negatives” (an actual email address or phone number that ***is not*** matched by the expressions provided).

# Part 1

In this section, various regular expressions were used to try to match obfuscated email addresses and phone numbers. My approach was to first do the best I could with the email addresses and then finish up with phone numbers. This section shows the iterations and the outcome of each.

## Baseline

First the code is run without any regular expressions to detect email addresses or phone numbers. This provides a baseline to improve. This provides us with the following information:

﻿Summary: tp=0, fp=0, fn=117

## Email Addresses

Each iteration is listed with the new true positives. Additional matches are shown in green font for each iteration.

### Iteration 1

For this iteration, the following regular expression was used:

* Pattern added: '([A-Za-z]+)@([A-Za-z]+)\.edu'
  + This expression is used to detect any number of alphabetic characters (upper and/or lower case) followed by an “@” sign and then any number of alphabetic characters (upper and/or lower case) followed by “.edu”   
    *Note: This pattern was intended to catch the most obvious email address formats.*
  + Examples of matches:
    - “balaji@stanford.edu”
    - "nass@stanford.edu”
* Run Output: Summary: tp=4, fp=1, fn=113
  + Matches:  
    True Positives (4):

{('balaji', 'e', 'balaji@stanford.edu'),

('nass', 'e', 'nass@stanford.edu'),

('shoham', 'e', 'shoham@stanford.edu'),

('thm', 'e', 'pkrokel@stanford.edu')}

### Iteration 2

For this iteration, the following regular expression was used:

* Pattern appended: '([A-Za-z.]+)@([A-Za-z.]+)\.edu'
  + This expression is used to detect any number of alphabetic characters (upper and/or lower case) and any special characters followed by an “@” sign and then any number of alphabetic characters (upper and/or lower case) and any character except a line terminator followed by “.edu”.  
    *Note: This pattern was intended to catch the addresses with “.” in the name or URL.*
  + Examples of matches:
    - “fedkiw@cs.stanford.edu”
    - "kosecka@cs.gmu.edu"
* Run Output: Summary: tp=19, fp=0, fn=98
  + Matches:

### True Positives (19):

### {('balaji', 'e', 'balaji@stanford.edu'),

### ('cheriton', 'e', 'cheriton@cs.stanford.edu'),

### ('engler', 'e', 'engler@lcs.mit.edu'),

### ('eroberts', 'e', 'eroberts@cs.stanford.edu'),

### ('fedkiw', 'e', 'fedkiw@cs.stanford.edu'),

### ('hanrahan', 'e', 'hanrahan@cs.stanford.edu'),

### ('kosecka', 'e', 'kosecka@cs.gmu.edu'),

### ('kunle', 'e', 'darlene@csl.stanford.edu'),

### ('kunle', 'e', 'kunle@ogun.stanford.edu'),

### ('nass', 'e', 'nass@stanford.edu'),

### ('nick', 'e', 'nick.parlante@cs.stanford.edu'),

### ('psyoung', 'e', 'patrick.young@stanford.edu'),

### ('rinard', 'e', 'rinard@lcs.mit.edu'),

### ('shoham', 'e', 'shoham@stanford.edu'),

### ('thm', 'e', 'pkrokel@stanford.edu'),

### ('widom', 'e', 'siroker@cs.stanford.edu'),

### ('widom', 'e', 'widom@cs.stanford.edu'),

### ('zelenski', 'e', 'zelenski@cs.stanford.edu'),

### ('zm', 'e', 'manna@cs.stanford.edu')}

### Iteration 3

For this iteration, the following regular expression was used:

* Pattern appended: '([A-Za-z.]+)\s@\s([A-Za-z.]+)\.edu'
  + This expression is used to detect any number of alphabetic characters (upper and/or lower case) and special characters followed by a space, then an “@” sign followed by a space and then any number of alphabetic characters (upper and/or lower case) and any character except a line terminator followed by “.edu”   
    *note: This pattern was intended to catch the addresses with an extra part of the URL (like cs.standford.edu or lcs.mit.edu).*
  + Examples of matches:
    - “rozm @ stanford.edu”
    - “ullman @ cs.stanford.edu”
* Run Output: Summary: tp=22, fp=1, fn=95
  + Matches:  
    True Positives (22):

{('ashishg', 'e', 'ashishg@stanford.edu'),

('ashishg', 'e', 'rozm@stanford.edu'),

('balaji', 'e', 'balaji@stanford.edu'),

('cheriton', 'e', 'cheriton@cs.stanford.edu'),

('engler', 'e', 'engler@lcs.mit.edu'),

('eroberts', 'e', 'eroberts@cs.stanford.edu'),

('fedkiw', 'e', 'fedkiw@cs.stanford.edu'),

('hanrahan', 'e', 'hanrahan@cs.stanford.edu'),

('kosecka', 'e', 'kosecka@cs.gmu.edu'),

('kunle', 'e', 'darlene@csl.stanford.edu'),

('kunle', 'e', 'kunle@ogun.stanford.edu'),

('nass', 'e', 'nass@stanford.edu'),

('nick', 'e', 'nick.parlante@cs.stanford.edu'),

('psyoung', 'e', 'patrick.young@stanford.edu'),

('rinard', 'e', 'rinard@lcs.mit.edu'),

('shoham', 'e', 'shoham@stanford.edu'),

('thm', 'e', 'pkrokel@stanford.edu'),

('ullman', 'e', 'ullman@cs.stanford.edu'),

('widom', 'e', 'siroker@cs.stanford.edu'),

('widom', 'e', 'widom@cs.stanford.edu'),

('zelenski', 'e', 'zelenski@cs.stanford.edu'),

('zm', 'e', 'manna@cs.stanford.edu')}

### Iteration 4

For this iteration, the following regular expression was used:

* Pattern appended: r'([A-Za-z.]+)\s+@\s+([A-Za-z.]+)\.[A-Za-z]+'
  + This raw expression is used to detect any number of alphabetic characters (upper and/or lower case) and special characters followed by any number of spaces, then an “@” sign followed by any number of spaces and then any number of alphabetic characters (upper and/or lower case) and any character except a line terminator followed by “.” and then any number of alphabetic characters.  
    *Note: Unfortunately, this pattern only added “dabo” to the true positives.*
  + Examples of matches:
    - “dabo @ cs.stanford.edu”
* Run Output: Summary: tp=23, fp=1, fn=94
  + Matches:  
    True Positives (23):

{('ashishg', 'e', 'ashishg@stanford.edu'),

('ashishg', 'e', 'rozm@stanford.edu'),

('balaji', 'e', 'balaji@stanford.edu'),

('cheriton', 'e', 'cheriton@cs.stanford.edu'),

('dabo', 'e', 'dabo@cs.stanford.edu'),

('engler', 'e', 'engler@lcs.mit.edu'),

('eroberts', 'e', 'eroberts@cs.stanford.edu'),

('fedkiw', 'e', 'fedkiw@cs.stanford.edu'),

('hanrahan', 'e', 'hanrahan@cs.stanford.edu'),

('kosecka', 'e', 'kosecka@cs.gmu.edu'),

('kunle', 'e', 'darlene@csl.stanford.edu'),

('kunle', 'e', 'kunle@ogun.stanford.edu'),

('nass', 'e', 'nass@stanford.edu'),

('nick', 'e', 'nick.parlante@cs.stanford.edu'),

('psyoung', 'e', 'patrick.young@stanford.edu'),

('rinard', 'e', 'rinard@lcs.mit.edu'),

('shoham', 'e', 'shoham@stanford.edu'),

('thm', 'e', 'pkrokel@stanford.edu'),

('ullman', 'e', 'ullman@cs.stanford.edu'),

('widom', 'e', 'siroker@cs.stanford.edu'),

('widom', 'e', 'widom@cs.stanford.edu'),

('zelenski', 'e', 'zelenski@cs.stanford.edu'),

('zm', 'e', 'manna@cs.stanford.edu')}

### Iteration 5

For this iteration, the following regular expression was used:

* Pattern appended: r'([A-Za-z.]+)@([A-Za-z.]+)\.[A-Za-z]+'
  + This raw expression is used to detect any number of alphabetic characters (upper and/or lower case) and special characters followed an “@” sign followed any number of alphabetic characters (upper and/or lower case) and any character except a line terminator followed by “.” and then any number of alphabetic characters.  
    *Note: Unfortunately, this only added a single match to the true positives.*
  + Examples of matches:
    - “uma@cs.stanford.EDU”
* Run Output: Summary: tp=24, fp=1, fn=93
  + Matches:  
    True Positives (24):

{('ashishg', 'e', 'ashishg@stanford.edu'),

('ashishg', 'e', 'rozm@stanford.edu'),

('balaji', 'e', 'balaji@stanford.edu'),

('cheriton', 'e', 'cheriton@cs.stanford.edu'),

('cheriton', 'e', 'uma@cs.stanford.edu'),

('dabo', 'e', 'dabo@cs.stanford.edu'),

('engler', 'e', 'engler@lcs.mit.edu'),

('eroberts', 'e', 'eroberts@cs.stanford.edu'),

('fedkiw', 'e', 'fedkiw@cs.stanford.edu'),

('hanrahan', 'e', 'hanrahan@cs.stanford.edu'),

('kosecka', 'e', 'kosecka@cs.gmu.edu'),

('kunle', 'e', 'darlene@csl.stanford.edu'),

('kunle', 'e', 'kunle@ogun.stanford.edu'),

('nass', 'e', 'nass@stanford.edu'),

('nick', 'e', 'nick.parlante@cs.stanford.edu'),

('psyoung', 'e', 'patrick.young@stanford.edu'),

('rinard', 'e', 'rinard@lcs.mit.edu'),

('shoham', 'e', 'shoham@stanford.edu'),

('thm', 'e', 'pkrokel@stanford.edu'),

('ullman', 'e', 'ullman@cs.stanford.edu'),

('widom', 'e', 'siroker@cs.stanford.edu'),

('widom', 'e', 'widom@cs.stanford.edu'),

('zelenski', 'e', 'zelenski@cs.stanford.edu'),

('zm', 'e', 'manna@cs.stanford.edu')}

### Iteration 6

For this iteration, the following regular expression was used:

* Pattern appended: r'(\w+)\b.[A-Z].\*\b(stanford).[A-Za-z]+.edu'
  + This raw expression is used to detect any number of alphanumeric characters and underscore followed by a boundary (or empty string) at the start or end of a word followed by and alphabetic character (upper case) followed by a “.” and then any boundary and then the word “stanford” followed by a “.” And then any number of alphabetic characters (upper and/or lower case) followed by “.edu”.  
    *Note: This pattern was more “hardcoded” to match the word ‘stanford’ which gave me two more matches.*
  + Examples of matches:
    - “engler WHERE stanford DOM edu”
    - “subh AT stanford DOT edu”
* Run Output: Summary: tp=26, fp=1, fn=91
  + Matches:  
    True Positives (26):

{('ashishg', 'e', 'ashishg@stanford.edu'),

('ashishg', 'e', 'rozm@stanford.edu'),

('balaji', 'e', 'balaji@stanford.edu'),

('cheriton', 'e', 'cheriton@cs.stanford.edu'),

('cheriton', 'e', 'uma@cs.stanford.edu'),

('dabo', 'e', 'dabo@cs.stanford.edu'),

('engler', 'e', 'engler@lcs.mit.edu'),

('engler', 'e', 'engler@stanford.edu'),

('eroberts', 'e', 'eroberts@cs.stanford.edu'),

('fedkiw', 'e', 'fedkiw@cs.stanford.edu'),

('hanrahan', 'e', 'hanrahan@cs.stanford.edu'),

('kosecka', 'e', 'kosecka@cs.gmu.edu'),

('kunle', 'e', 'darlene@csl.stanford.edu'),

('kunle', 'e', 'kunle@ogun.stanford.edu'),

('nass', 'e', 'nass@stanford.edu'),

('nick', 'e', 'nick.parlante@cs.stanford.edu'),

('psyoung', 'e', 'patrick.young@stanford.edu'),

('rinard', 'e', 'rinard@lcs.mit.edu'),

('shoham', 'e', 'shoham@stanford.edu'),

('subh', 'e', 'subh@stanford.edu'),

('thm', 'e', 'pkrokel@stanford.edu'),

('ullman', 'e', 'ullman@cs.stanford.edu'),

('widom', 'e', 'siroker@cs.stanford.edu'),

('widom', 'e', 'widom@cs.stanford.edu'),

('zelenski', 'e', 'zelenski@cs.stanford.edu'),

('zm', 'e', 'manna@cs.stanford.edu')}Iteration 4

### Iteration 7

For this iteration, the following regular expression was used:

* Pattern appended: r'([a-z]+).at <!--.+>.(stanford).+edu'
  + This raw expression is used to detect any number of alphabetic characters (lower case) followed by the expression “<!---.+>” which should return <!-> and any additional matching special characters (like “spam pigs”) followed by the word “stanford” followed by “.edu”.   
    *Note: Sadly, this one took a lot of effort to figure out and only added one match. I guess no one else was as adamant about not being caught with spam.*
  + Examples of matches:
    - “vladlen at <!-- die!--> stanford <!-- spam pigs!--> dot <!-- die!--> edu”
* Run Output: Summary: tp=27, fp=1, fn=90
  + Matches:  
    True Positives (27):

{('ashishg', 'e', 'ashishg@stanford.edu'),

('ashishg', 'e', 'rozm@stanford.edu'),

('balaji', 'e', 'balaji@stanford.edu'),

('cheriton', 'e', 'cheriton@cs.stanford.edu'),

('cheriton', 'e', 'uma@cs.stanford.edu'),

('dabo', 'e', 'dabo@cs.stanford.edu'),

('engler', 'e', 'engler@lcs.mit.edu'),

('engler', 'e', 'engler@stanford.edu'),

('eroberts', 'e', 'eroberts@cs.stanford.edu'),

('fedkiw', 'e', 'fedkiw@cs.stanford.edu'),

('hanrahan', 'e', 'hanrahan@cs.stanford.edu'),

('kosecka', 'e', 'kosecka@cs.gmu.edu'),

('kunle', 'e', 'darlene@csl.stanford.edu'),

('kunle', 'e', 'kunle@ogun.stanford.edu'),

('nass', 'e', 'nass@stanford.edu'),

('nick', 'e', 'nick.parlante@cs.stanford.edu'),

('psyoung', 'e', 'patrick.young@stanford.edu'),

('rinard', 'e', 'rinard@lcs.mit.edu'),

('shoham', 'e', 'shoham@stanford.edu'),

('subh', 'e', 'subh@stanford.edu'),

('thm', 'e', 'pkrokel@stanford.edu'),

('ullman', 'e', 'ullman@cs.stanford.edu'),

('vladlen', 'e', 'vladlen@stanford.edu'),

('widom', 'e', 'siroker@cs.stanford.edu'),

('widom', 'e', 'widom@cs.stanford.edu'),

('zelenski', 'e', 'zelenski@cs.stanford.edu'),

('zm', 'e', 'manna@cs.stanford.edu')}

### Iteration 8

For this iteration, the following regular expression was used:

* Pattern appended: r'([a-z]+)&#x40;(graphics.stanford).edu'
  + This raw expression is used to detect any number of alphabetic characters (lower case) followed by the expression ‘&#x40;’ followed by “graphics.stanford” and then ending with “.edu”.  
    *Note: Another hardcoded attempt to get “graphics.standford” emails.*
  + Examples of matches:
    - “ada&#x40;graphics.stanford.edu”
    - “melissa&#x40;graphics.stanford.edu”
* Run Output: Summary: tp=29, fp=1, fn=88
  + Matches:  
    True Positives (29):

{('ashishg', 'e', 'ashishg@stanford.edu'),

('ashishg', 'e', 'rozm@stanford.edu'),

('balaji', 'e', 'balaji@stanford.edu'),

('cheriton', 'e', 'cheriton@cs.stanford.edu'),

('cheriton', 'e', 'uma@cs.stanford.edu'),

('dabo', 'e', 'dabo@cs.stanford.edu'),

('engler', 'e', 'engler@lcs.mit.edu'),

('engler', 'e', 'engler@stanford.edu'),

('eroberts', 'e', 'eroberts@cs.stanford.edu'),

('fedkiw', 'e', 'fedkiw@cs.stanford.edu'),

('hanrahan', 'e', 'hanrahan@cs.stanford.edu'),

('kosecka', 'e', 'kosecka@cs.gmu.edu'),

('kunle', 'e', 'darlene@csl.stanford.edu'),

('kunle', 'e', 'kunle@ogun.stanford.edu'),

('levoy', 'e', 'ada@graphics.stanford.edu'),

('levoy', 'e', 'melissa@graphics.stanford.edu'),

('nass', 'e', 'nass@stanford.edu'),

('nick', 'e', 'nick.parlante@cs.stanford.edu'),

('psyoung', 'e', 'patrick.young@stanford.edu'),

('rinard', 'e', 'rinard@lcs.mit.edu'),

('shoham', 'e', 'shoham@stanford.edu'),

('subh', 'e', 'subh@stanford.edu'),

('thm', 'e', 'pkrokel@stanford.edu'),

('ullman', 'e', 'ullman@cs.stanford.edu'),

('vladlen', 'e', 'vladlen@stanford.edu'),

('widom', 'e', 'siroker@cs.stanford.edu'),

('widom', 'e', 'widom@cs.stanford.edu'),

('zelenski', 'e', 'zelenski@cs.stanford.edu'),

('zm', 'e', 'manna@cs.stanford.edu')}

### Iteration 9

For this iteration, the following regular expression was used:

* Pattern appended: '([A-Za-z.]+)\s<at symbol>\s([A-Za-z.]+)\.edu'
  + This expression is used to detect any number of alphabetic characters (upper and/or lower case) and special characters followed by a space and then the words “<at symbol>” followed by another space and then any number of alphabetic characters (upper and/or lower case) and any character except a line terminator followed by “.edu”.
  + Examples of matches:
    - “manning <at symbol> cs.stanford.edu”
    - “dbarros <at symbol> cs.stanford.edu”
* Run Output: Summary: tp=31, fp=1, fn=86
  + Matches:  
    True Positives (31):

{('ashishg', 'e', 'ashishg@stanford.edu'),

('ashishg', 'e', 'rozm@stanford.edu'),

('balaji', 'e', 'balaji@stanford.edu'),

('cheriton', 'e', 'cheriton@cs.stanford.edu'),

('cheriton', 'e', 'uma@cs.stanford.edu'),

('dabo', 'e', 'dabo@cs.stanford.edu'),

('engler', 'e', 'engler@lcs.mit.edu'),

('engler', 'e', 'engler@stanford.edu'),

('eroberts', 'e', 'eroberts@cs.stanford.edu'),

('fedkiw', 'e', 'fedkiw@cs.stanford.edu'),

('hanrahan', 'e', 'hanrahan@cs.stanford.edu'),

('kosecka', 'e', 'kosecka@cs.gmu.edu'),

('kunle', 'e', 'darlene@csl.stanford.edu'),

('kunle', 'e', 'kunle@ogun.stanford.edu'),

('levoy', 'e', 'ada@graphics.stanford.edu'),

('levoy', 'e', 'melissa@graphics.stanford.edu'),

('manning', 'e', 'dbarros@cs.stanford.edu'),

('manning', 'e', 'manning@cs.stanford.edu'),

('nass', 'e', 'nass@stanford.edu'),

('nick', 'e', 'nick.parlante@cs.stanford.edu'),

('psyoung', 'e', 'patrick.young@stanford.edu'),

('rinard', 'e', 'rinard@lcs.mit.edu'),

('shoham', 'e', 'shoham@stanford.edu'),

('subh', 'e', 'subh@stanford.edu'),

('thm', 'e', 'pkrokel@stanford.edu'),

('ullman', 'e', 'ullman@cs.stanford.edu'),

('vladlen', 'e', 'vladlen@stanford.edu'),

('widom', 'e', 'siroker@cs.stanford.edu'),

('widom', 'e', 'widom@cs.stanford.edu'),

('zelenski', 'e', 'zelenski@cs.stanford.edu'),

('zm', 'e', 'manna@cs.stanford.edu')}

### Iteration 10

For this iteration, the following regular expression was used:

* Pattern appended: '([A-Za-z.]+)<del>@([A-Za-z.]+)\.edu'
  + This expression is used to detect any number of alphabetic characters (upper and/or lower case) and special characters followed by a space and then the words “<del>” followed by another space and then any number of alphabetic characters (upper and/or lower case) and any character except a line terminator followed by “.edu”.
  + Examples of matches:
    - “latombe<del>@cs.stanford.edu”
    - “liliana<del>@cs.stanford.edu”
* Run Output: Summary: tp=34, fp=1, fn=83
  + Matches:  
    True Positives (34):

{('ashishg', 'e', 'ashishg@stanford.edu'),

('ashishg', 'e', 'rozm@stanford.edu'),

('balaji', 'e', 'balaji@stanford.edu'),

('cheriton', 'e', 'cheriton@cs.stanford.edu'),

('cheriton', 'e', 'uma@cs.stanford.edu'),

('dabo', 'e', 'dabo@cs.stanford.edu'),

('engler', 'e', 'engler@lcs.mit.edu'),

('engler', 'e', 'engler@stanford.edu'),

('eroberts', 'e', 'eroberts@cs.stanford.edu'),

('fedkiw', 'e', 'fedkiw@cs.stanford.edu'),

('hanrahan', 'e', 'hanrahan@cs.stanford.edu'),

('kosecka', 'e', 'kosecka@cs.gmu.edu'),

('kunle', 'e', 'darlene@csl.stanford.edu'),

('kunle', 'e', 'kunle@ogun.stanford.edu'),

('latombe', 'e', 'asandra@cs.stanford.edu'),

('latombe', 'e', 'latombe@cs.stanford.edu'),

('latombe', 'e', 'liliana@cs.stanford.edu'),

('levoy', 'e', 'ada@graphics.stanford.edu'),

('levoy', 'e', 'melissa@graphics.stanford.edu'),

('manning', 'e', 'dbarros@cs.stanford.edu'),

('manning', 'e', 'manning@cs.stanford.edu'),

('nass', 'e', 'nass@stanford.edu'),

('nick', 'e', 'nick.parlante@cs.stanford.edu'),

('psyoung', 'e', 'patrick.young@stanford.edu'),

('rinard', 'e', 'rinard@lcs.mit.edu'),

('shoham', 'e', 'shoham@stanford.edu'),

('subh', 'e', 'subh@stanford.edu'),

('thm', 'e', 'pkrokel@stanford.edu'),

('ullman', 'e', 'ullman@cs.stanford.edu'),

('vladlen', 'e', 'vladlen@stanford.edu'),

('widom', 'e', 'siroker@cs.stanford.edu'),

('widom', 'e', 'widom@cs.stanford.edu'),

('zelenski', 'e', 'zelenski@cs.stanford.edu'),

('zm', 'e', 'manna@cs.stanford.edu')}

### Iteration 11

For this iteration, the following regular expression was used:

* Pattern appended: '([A-Za-z.]+) [at]+ ([A-Za-z.]+)\.edu'
  + This expression is used to match any single character (upper or lower case a-z) including a period one to unlimited times followed by a space and then the word “at” one to unlimited times then a space and then a group matching any single character (upper or lower case a-z) including a period one to unlimited times followed by “.edu”.
  + Examples of matches:
    - “lam at cs.stanford.edu”
* Run Output: Summary: tp=35, fp=1, fn=82
  + Matches:  
    True Positives (35):

{('ashishg', 'e', 'ashishg@stanford.edu'),

('ashishg', 'e', 'rozm@stanford.edu'),

('balaji', 'e', 'balaji@stanford.edu'),

('cheriton', 'e', 'cheriton@cs.stanford.edu'),

('cheriton', 'e', 'uma@cs.stanford.edu'),

('dabo', 'e', 'dabo@cs.stanford.edu'),

('engler', 'e', 'engler@lcs.mit.edu'),

('engler', 'e', 'engler@stanford.edu'),

('eroberts', 'e', 'eroberts@cs.stanford.edu'),

('fedkiw', 'e', 'fedkiw@cs.stanford.edu'),

('hanrahan', 'e', 'hanrahan@cs.stanford.edu'),

('kosecka', 'e', 'kosecka@cs.gmu.edu'),

('kunle', 'e', 'darlene@csl.stanford.edu'),

('kunle', 'e', 'kunle@ogun.stanford.edu'),

('lam', 'e', 'lam@cs.stanford.edu'),

('latombe', 'e', 'asandra@cs.stanford.edu'),

('latombe', 'e', 'latombe@cs.stanford.edu'),

('latombe', 'e', 'liliana@cs.stanford.edu'),

('levoy', 'e', 'ada@graphics.stanford.edu'),

('levoy', 'e', 'melissa@graphics.stanford.edu'),

('manning', 'e', 'dbarros@cs.stanford.edu'),

('manning', 'e', 'manning@cs.stanford.edu'),

('nass', 'e', 'nass@stanford.edu'),

('nick', 'e', 'nick.parlante@cs.stanford.edu'),

('psyoung', 'e', 'patrick.young@stanford.edu'),

('rinard', 'e', 'rinard@lcs.mit.edu'),

('shoham', 'e', 'shoham@stanford.edu'),

('subh', 'e', 'subh@stanford.edu'),

('thm', 'e', 'pkrokel@stanford.edu'),

('ullman', 'e', 'ullman@cs.stanford.edu'),

('vladlen', 'e', 'vladlen@stanford.edu'),

('widom', 'e', 'siroker@cs.stanford.edu'),

('widom', 'e', 'widom@cs.stanford.edu'),

('zelenski', 'e', 'zelenski@cs.stanford.edu'),

('zm', 'e', 'manna@cs.stanford.edu')}

## Phone Numbers

After trying many different iterations for emails that did not improve the number of true positives of 31 by matching telephone numbers. The status before starting to match phone numbers is:

﻿Summary: tp=35, fp=1, fn=82.

### Iteration 1

For the first iteration for phone numbers, the following regular expression was used:

* Pattern appended: '(\d{3})-(\d{3})-(\d{4})'
  + This expression is used to detect three (3) numeric digits followed by a dash (“-“) then three more numeric digits followed by another dash and then four (4) numeric digits.   
    *Note: This pattern is intended to match standard phone numbers.*
  + Examples of matches:
    - “650-724-1915”
    - “650-725-3726”
* Run Output: Summary: tp=54, fp=1, fn=63
  + Matches:  
    True Positives (54):

{('ashishg', 'e', 'ashishg@stanford.edu'),

('ashishg', 'e', 'rozm@stanford.edu'),

('balaji', 'e', 'balaji@stanford.edu'),

('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'),

('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', 'p', '410-516-8000'),

('hanrahan', 'e', 'hanrahan@cs.stanford.edu'),

('kosecka', 'e', 'kosecka@cs.gmu.edu'),

('kunle', 'e', 'darlene@csl.stanford.edu'),

('kunle', 'e', 'kunle@ogun.stanford.edu'),

('lam', 'e', 'lam@cs.stanford.edu'),

('latombe', 'e', 'asandra@cs.stanford.edu'),

('latombe', 'e', 'latombe@cs.stanford.edu'),

('latombe', 'e', 'liliana@cs.stanford.edu'),

('levoy', 'e', 'ada@graphics.stanford.edu'),

('levoy', 'e', 'melissa@graphics.stanford.edu'),

('manning', 'e', 'dbarros@cs.stanford.edu'),

('manning', 'e', 'manning@cs.stanford.edu'),

('nass', 'e', 'nass@stanford.edu'),

('nick', 'e', 'nick.parlante@cs.stanford.edu'),

('psyoung', 'e', 'patrick.young@stanford.edu'),

('rajeev', 'p', '650-723-4377'),

('rajeev', 'p', '650-723-6045'),

('rajeev', 'p', '650-725-4671'),

('rinard', 'e', 'rinard@lcs.mit.edu'),

('shoham', 'e', 'shoham@stanford.edu'),

('subh', 'e', 'subh@stanford.edu'),

('subh', 'p', '650-724-1915'),

('subh', 'p', '650-725-3726'),

('subh', 'p', '650-725-6949'),

('thm', 'e', 'pkrokel@stanford.edu'),

('ullman', 'e', 'ullman@cs.stanford.edu'),

('ullman', 'p', '650-494-8016'),

('ullman', 'p', '650-725-2588'),

('ullman', 'p', '650-725-4802'),

('vladlen', 'e', 'vladlen@stanford.edu'),

('widom', 'e', 'siroker@cs.stanford.edu'),

('widom', 'e', 'widom@cs.stanford.edu'),

('widom', 'p', '650-723-0872'),

('widom', 'p', '650-723-7690'),

('widom', 'p', '650-725-2588'),

('zelenski', 'e', 'zelenski@cs.stanford.edu'),

('zelenski', 'p', '650-723-6092'),

('zelenski', 'p', '650-725-8596'),

('zm', 'e', 'manna@cs.stanford.edu')}

### Iteration 2

After a solid improvement with the first iteration of phone number matches, another regular expression was used:

* Pattern appended: r'.+(\d{3}).[^0-9](\d{3})[^0-9](\d{4})'
  + This raw expression is used to detect any character (one or unlimited) except line terminators followed by three (3) numeric digits (0-9) followed any character (one or unlimited) except line terminators but not a numeric digit between 0 and 9, followed by 3 more numeric digits, a single character that is not 0 through 9 and then 4 numeric characters.  
    *Note:. Lots of new matches (40 additional).*
  + Examples of matches:
    - “(650) 724-6354”
    - “(650) 724-3648”
* Run Output: Summary: tp=94, fp=1, fn=23
  + Matches:  
    True Positives (94):

{('ashishg', 'e', 'ashishg@stanford.edu'),

('ashishg', 'e', 'rozm@stanford.edu'),

('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'),

('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', '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'),

('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'),

('manning', 'e', 'dbarros@cs.stanford.edu'),

('manning', 'e', 'manning@cs.stanford.edu'),

('manning', 'p', '650-723-7683'),

('manning', 'p', '650-725-1449'),

('manning', 'p', '650-725-3358'),

('nass', 'e', 'nass@stanford.edu'),

('nass', 'p', '650-723-5499'),

('nass', 'p', '650-725-2472'),

('nick', 'e', 'nick.parlante@cs.stanford.edu'),

('nick', 'p', '650-725-4727'),

('ok', 'p', '650-723-9753'),

('ok', 'p', '650-725-1449'),

('psyoung', 'e', 'patrick.young@stanford.edu'),

('rajeev', 'p', '650-723-4377'),

('rajeev', 'p', '650-723-6045'),

('rajeev', 'p', '650-725-4671'),

('rinard', 'e', 'rinard@lcs.mit.edu'),

('rinard', 'p', '617-253-1221'),

('rinard', 'p', '617-258-6922'),

('serafim', 'p', '650-725-1449'),

('shoham', 'e', 'shoham@stanford.edu'),

('subh', 'e', 'subh@stanford.edu'),

('subh', 'p', '650-724-1915'),

('subh', 'p', '650-725-3726'),

('subh', 'p', '650-725-6949'),

('thm', 'e', 'pkrokel@stanford.edu'),

('thm', 'p', '650-725-3383'),

('thm', 'p', '650-725-3636'),

('thm', 'p', '650-725-3938'),

('ullman', 'e', 'ullman@cs.stanford.edu'),

('ullman', 'p', '650-494-8016'),

('ullman', 'p', '650-725-2588'),

('ullman', 'p', '650-725-4802'),

('vladlen', 'e', 'vladlen@stanford.edu'),

('widom', 'e', 'siroker@cs.stanford.edu'),

('widom', 'e', 'widom@cs.stanford.edu'),

('widom', 'p', '650-723-0872'),

('widom', 'p', '650-723-7690'),

('widom', 'p', '650-725-2588'),

('zelenski', 'e', 'zelenski@cs.stanford.edu'),

('zelenski', 'p', '650-723-6092'),

('zelenski', 'p', '650-725-8596'),

('zm', 'e', 'manna@cs.stanford.edu'),

('zm', 'p', '650-723-4364'),

('zm', 'p', '650-725-4671')}

### Iteration 3

For this iteration, the following regular expression was used:

* Pattern appended: r'.?(\d{3})[^0-9](\d{3})[^0-9](\d{4})'
  + This raw expression is used to detect any character (except line terminator) zero or one times, followed by 3 numeric digits, followed by a single character not in 0 to 9, followed by another 3 numeric digit group, followed by a single character not in 0 to 9 and then a 4 digit group of numeric digits.
  + Examples of matches:
    - “(650)723-1614”
    - “(650)725-6949”
* Run Output: Summary: tp=106, fp=1, fn=11
  + Matches:  
    True Positives (106):

{('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'),

('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', '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'),

('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'),

('manning', 'e', 'dbarros@cs.stanford.edu'),

('manning', 'e', 'manning@cs.stanford.edu'),

('manning', 'p', '650-723-7683'),

('manning', 'p', '650-725-1449'),

('manning', 'p', '650-725-3358'),

('nass', 'e', 'nass@stanford.edu'),

('nass', 'p', '650-723-5499'),

('nass', 'p', '650-725-2472'),

('nick', 'e', 'nick.parlante@cs.stanford.edu'),

('nick', 'p', '650-725-4727'),

('ok', 'p', '650-723-9753'),

('ok', 'p', '650-725-1449'),

('pal', 'p', '650-725-9046'),

('psyoung', 'e', 'patrick.young@stanford.edu'),

('rajeev', 'p', '650-723-4377'),

('rajeev', 'p', '650-723-6045'),

('rajeev', 'p', '650-725-4671'),

('rinard', 'e', 'rinard@lcs.mit.edu'),

('rinard', 'p', '617-253-1221'),

('rinard', 'p', '617-258-6922'),

('serafim', 'p', '650-725-1449'),

('shoham', 'e', 'shoham@stanford.edu'),

('shoham', 'p', '650-723-3432'),

('shoham', 'p', '650-725-1449'),

('subh', 'e', 'subh@stanford.edu'),

('subh', 'p', '650-724-1915'),

('subh', 'p', '650-725-3726'),

('subh', 'p', '650-725-6949'),

('thm', 'e', 'pkrokel@stanford.edu'),

('thm', 'p', '650-725-3383'),

('thm', 'p', '650-725-3636'),

('thm', 'p', '650-725-3938'),

('tim', 'p', '650-724-9147'),

('tim', 'p', '650-725-2340'),

('tim', 'p', '650-725-4671'),

('ullman', 'e', 'ullman@cs.stanford.edu'),

('ullman', 'p', '650-494-8016'),

('ullman', 'p', '650-725-2588'),

('ullman', 'p', '650-725-4802'),

('vladlen', 'e', 'vladlen@stanford.edu'),

('widom', 'e', 'siroker@cs.stanford.edu'),

('widom', 'e', 'widom@cs.stanford.edu'),

('widom', 'p', '650-723-0872'),

('widom', 'p', '650-723-7690'),

('widom', 'p', '650-725-2588'),

('zelenski', 'e', 'zelenski@cs.stanford.edu'),

('zelenski', 'p', '650-723-6092'),

('zelenski', 'p', '650-725-8596'),

('zm', 'e', 'manna@cs.stanford.edu'),

('zm', 'p', '650-723-4364'),

('zm', 'p', '650-725-4671')}

### Iteration 4

For this iteration, the following regular expression was used:

* Pattern appended: '(?:[(])([0-9]{3})(?:[)])[ ]\*([0-9]{3})-([0-9]{4})'
  + This expression is used to match a left parenthesis “(“ followed by a 3 digit numeric group of numbers from 0 to 9, followed by a right parenthesis “)” and then a single character (zero to unlimited amount), followed by 3 numeric digits and a hyphen “-“ followed by 4 numeric digits..  
    *Note: Only one additional match was found.*
  + Example of matches:
    - “(650) 723-3334”
* Run Output: Summary: tp=107, fp=1, fn=10
  + Matches:  
    True Positives (107):

{('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'),

('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', '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'),

('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'),

('manning', 'e', 'dbarros@cs.stanford.edu'),

('manning', 'e', 'manning@cs.stanford.edu'),

('manning', 'p', '650-723-7683'),

('manning', 'p', '650-725-1449'),

('manning', 'p', '650-725-3358'),

('nass', 'e', 'nass@stanford.edu'),

('nass', 'p', '650-723-5499'),

('nass', 'p', '650-725-2472'),

('nick', 'e', 'nick.parlante@cs.stanford.edu'),

('nick', 'p', '650-725-4727'),

('ok', 'p', '650-723-9753'),

('ok', 'p', '650-725-1449'),

('pal', 'p', '650-725-9046'),

('psyoung', 'e', 'patrick.young@stanford.edu'),

('rajeev', 'p', '650-723-4377'),

('rajeev', 'p', '650-723-6045'),

('rajeev', 'p', '650-725-4671'),

('rinard', 'e', 'rinard@lcs.mit.edu'),

('rinard', 'p', '617-253-1221'),

('rinard', 'p', '617-258-6922'),

('serafim', 'p', '650-723-3334'),

('serafim', 'p', '650-725-1449'),

('shoham', 'e', 'shoham@stanford.edu'),

('shoham', 'p', '650-723-3432'),

('shoham', 'p', '650-725-1449'),

('subh', 'e', 'subh@stanford.edu'),

('subh', 'p', '650-724-1915'),

('subh', 'p', '650-725-3726'),

('subh', 'p', '650-725-6949'),

('thm', 'e', 'pkrokel@stanford.edu'),

('thm', 'p', '650-725-3383'),

('thm', 'p', '650-725-3636'),

('thm', 'p', '650-725-3938'),

('tim', 'p', '650-724-9147'),

('tim', 'p', '650-725-2340'),

('tim', 'p', '650-725-4671'),

('ullman', 'e', 'ullman@cs.stanford.edu'),

('ullman', 'p', '650-494-8016'),

('ullman', 'p', '650-725-2588'),

('ullman', 'p', '650-725-4802'),

('vladlen', 'e', 'vladlen@stanford.edu'),

('widom', 'e', 'siroker@cs.stanford.edu'),

('widom', 'e', 'widom@cs.stanford.edu'),

('widom', 'p', '650-723-0872'),

('widom', 'p', '650-723-7690'),

('widom', 'p', '650-725-2588'),

('zelenski', 'e', 'zelenski@cs.stanford.edu'),

('zelenski', 'p', '650-723-6092'),

('zelenski', 'p', '650-725-8596'),

('zm', 'e', 'manna@cs.stanford.edu'),

('zm', 'p', '650-723-4364'),

('zm', 'p', '650-725-4671')}

Other attempts were made with no additional matches, so these are not documented. Our final status after these iterations for emails and phone numbers was:

* Summary: tp=107, fp=1, fn=10

# Part 3 (Option)

The next step in the homework assignment was to select further matching patterns and possible alter the existing python function provided *process\_file* to attempt to catch the remaining 10 items that were not matched in previous methods.

*Note: I got very confused in this part, but tried to do my best to add patterns to assist in fidnig new things. I tried to add patterns to match ‘com’ and ‘edu’ and semi-colons and e-d-u. I also updated the line and the sections of the process\_file function, but I ran out of time to document and detail what I did.*

## Additional Regular Expressions

Multiple patterns were added to try to match additional emails.

The first expression added was

* Pattern appended:   
  '([A-Za-z.\_0-9]+)\s\*(?:@|\sat\s)\s\*([A-Za-z.\_0-9]+)(?:\.|\s)([A-Za-z]+)'
  + This pattern matches characters from the sets of upper and lower case characters, a single character of either “.” or “\_” followed by a single character from 0 through 9. Then matching a whitespace character (\s\*) from zero to unlimited times and then matching the character “@” or space “at” space then again characters from upper and lower case letters and digits from 0 to 9 then matching a “.” followed by whitespace and then characters from upper and lower case letters.
  + Run Output:  
    Summary: tp=108, fp=44, fn=9
  + Additional Match:  
    ('jurafsky', 'e', 'jurafsky@stanford.edu')
  + Issues: Led to 44 false positives.

The next attempt was to add this pattern:

* Pattern appended:  
  '([A-Za-z.\_0-9]+)\s\*(?:@|\sat\s)\s\*([A-Za-z.\_0-9]+)(?:\.|\s)([A-Za-z]{3})'
  + This matches the same as above, but constricts the URL for the email server to only 3 characters (.org, .edu, .com, etc.)
  + Run Output:  
    Summary: tp=108, fp=44, fn=9
  + Additional Match:  
    ('jurafsky', 'e', 'jurafsky@stanford.edu')
  + Issues: Led to 44 false positives.

Then I started grasping at things that I found online:

'([A-Za-z.\_0-9]+)\s\*(?:@|\sat\s)\s\*([A-Za-z.\_0-9]+)(?:\.|\s)(com|edu)\W')  
‘('([A-Za-z.\_0-9]+)\s\*(?:@|\sat\s)\s\*([A-Za-z.;\_0-9]+)\s\*(?:\.|\s|;)((?:com|edu))\W')  
'([A-Za-z.\_0-9-]+)@([A-Za-z.;\_0-9-]+)\.(-e-d-u)')

I found some examples and try to add some lines to the python function *process\_file* to accommodate for these patterns:

Substitutions in the line:

# before applying the patterns

# matching DOT, dot, DOM and dt with the line

line = re.sub('\s?\<\!.+?\>\s?', ' ', line)

line = re.sub('\s+(?:DOT|dot|DOM|dt)\s+', '.', line)

if '-@-' in line:

line = re.sub( '-' , '' , line)

Then I had to change the way the code accommodated if the line had more than 2 entities:

if len(m) <= 2:

email = '{}@{}.edu'.format(m[0],m[1])

elif len(m) == 3:

eServe = re.sub( '(?:\s|\;)' , '.' , m[1])

eName = re.sub( '-' , '' , m[0])

eServe = re.sub( '-' , '' , eServe)

tld = re.sub( '-' , '' , m[2])

email = '{}@{}.{}' .format(eName,eServe, tld)

res.append((name,'e',email))

All this work led to 111 true positives and 6 false negatives. At this point, I stopped trying.

# Conclusion

I hope I never have to work with regular expressions if I change careers. I tried my best with this. Worked on it for hours and still couldn’t figure it out. I decided to turn something in with what I did accomplish in hopes for partial credit. This is, indeed, the first time I have given up on homework in my 58 years..

IST664 Natural Language Processing: Homework 2

2019-05-10

Using these ten expressions, we get a result of 105 true positive results, no false positive results, and 12 false negative results.

1. Exercise 2
   1. With the regular expressions listed in exercise 1, the following list of regular expressions could not be matched:

* [d-l-w-h-@-s-t-a-n-f-o-r-d-.-e-d-u](mailto:d-l-w-h-@-s-t-a-n-f-o-r-d-.-e-d-u)
* hager at cs dot jhu dot edu
* jks at robotics;stanford;edu
* obfuscate(‘stanford.edu’, ‘jurafsky’)
* lam at cs.stanford.edu
* dbarros <at symbol> cs.stanford.edu
* manning <at symbol> cs.stanford.edu
* pal at cs stanford edu
* support at gradiance dt com

These regular expressions, though matched in the testing phase, could not be extracted from the documents because there were additional whitespace characters intertwined with the expressions that were not contemplated during the test phase.

In cases like ‘jks at robotics;stanford;edu’, the reason this expression couldn’t be matched was because of the semicolons that obfuscated the regular expression within the test phase. Similarly, the [d-l-w-h-@-s-t-a-n-f-o-r-d-.-e-d-u](mailto:d-l-w-h-@-s-t-a-n-f-o-r-d-.-e-d-u) email could not be retrieved because of the dashes between the letters. Though these were able to be identified individually, when the groups were constructed, I could not find a way of ignoring the dashes.

Finally, the email address ‘obfuscate(‘stanford.edu’, ‘jurafsky’) was built in a way not expected, as there were no ways of two groups with a middle at symbol to identify the expression. Thus, a different expression needed to be created to extract this email.

* 1. From what research I could find, the consensus is that the best way to obfuscate email addresses is by using HTML tags around and within the email address. Similarly, there are many websites that provide email obfuscation by using methods, such as encryption and appending different, non-relevant characters around and within the email address.

Finally, one last advice I’ve seen about obfuscation is to hide the addresses within Python and JavaScript code, which makes it hard for web crawlers to separate the code from the email address, as they intertwine seamlessly.

For example, emails could be created as:

* <href>[mailto:martin @ <domain>example</domain>.edu</href>endmail](mailto:martin%20@%20%3cdomain%3eexample%3c/domain%3e.edu%3c/href%3eendmail).
* <a href="javascript:window.location.href=atob('<?= base64\_encode("mailto:email@example.com") ?>')" style="unicode-bidi: bidi-override; direction: rtl;"><?= strrev("email@example.com") ?></a>
* <h3 id="email">[hello@gmail.com</h3](mailto:hello@gmail.com%3c/h3)>