**NLP Homework 2: Contact Finder:**

**Using Regular Expressions to find email addresses and phone numbers**

Part 1: Creating Regex Patterns to Match emails and phone numbers

Solution:

* 1. [eramosmo@g.syr.EDU](mailto:eramosmo@g.syr.EDU)
  2. [balaji@stanford.edu](mailto:balaji@stanford.edu)
  3. 20 True Positives
  4. Match regular email addresses with either edu or EDU in the end.
  5. eramosmo @ g.syr.edu
  6. ashishg @ stanford.edu
  7. 5 True Positives
  8. Find regular addresses that have one or more spaces in before or after the @ symbol.
  9. eramosmo AT g.syr. DOT edu
  10. 1 True Positives
  11. Match regular emails with AT instead of @ and DOT instead of a period.
  12. eramosmo WHERE g.syr DOM edu
  13. 1 True Positives
  14. Match regular emails with WHERE and DOM instead of @ and a period correspondingly.
  15. eramosmo<del>@g.syr.edu
  16. 3 True Positives
  17. Find regular emails with <del> instead of the @ symbol.
  18. eramosmo &#x40;g.syr.edu
  19. 2 True Positives
  20. Find regular emails with &#x40; instead of the @ symbol.
  21. eramosmo <at symbol> g.syr.edu
  22. 2 True Positives
  23. Find regular emails with <at symbol> instead of the @ symbol.

**Was able to find 33 Email Addresses Using the Values Above**

* 1. 479-925-8904
  2. ('cheriton', 'p', '650-723-1131')
  3. 19 True Positives
  4. Regular phone number with no parenthesis and dashes in between area code.
  5. ('ashishg', 'p', '650-723-1614')
  6. 8 True Positives
  7. Parenthesis around the area code.
  8. ('bgirod', 'p', '650-723-4539')
  9. Parenthesis and one space after each group of numbers.
  10. 39 True Positives
  11. ('jurafsky', 'p', '650-723-5666')
  12. 2 True Positives
  13. Parenthesis and one or more spaces after each group of numbers.
  14. [479] 925-8904
  15. (nass, 'p', '650-725-2472)
  16. 2 True Positives
  17. Brackets instead of parenthesis around the area code.

1. (\d{3})[\s]+(\d{3})-(\d{4})
   1. 479 925-8904
   2. ('shoham', 'p', '650-723-3432')
   3. 2 True Positives
   4. One or more spaces after the area code.

**Was able to find 72 Phone Numbers Using the Regex Above**

**Using this list of regular expressions above, we were able to get 105 True Positives and 12 False Negatives.**

**Program Output:**

Assuming ContactFinder.py called in directory with data folder

True Positives (105):

{('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', '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'),

('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')}

False Positives (0):

False Negatives (12):

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

('hager', 'e', 'hager@cs.jhu.edu'),

('jks', 'e', 'jks@robotics.stanford.edu'),

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

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

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

('ouster', 'e', 'teresa.lynn@stanford.edu'),

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

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

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

('ullman', 'e', 'support@gradiance.com'),

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

**Part 2:**

1. **List examples that you found you could not match:**
2. [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)
   1. Reason it won’t work is that even if I can find the email, the regular expression cant take out the dashes. The output of the program will include the dashes in the email.
   2. '([A-Za-z.-]+)@([A-Za-z.-]+)\.[edu|EDU]'
3. hager at cs dot jhu dot edu
   1. Similar to the one above, I can’t replace a dot with a period in the program after finding it.
4. jks at robotics;stanford;edu
   1. Similar to the two above except I can’t replace a semicolon with a period.
   2. '([A-Za-z.-]+)@([A-Za-z.-;]+)\.[edu|EDU]'
5. obfuscate('stanford.edu','jurafsky')
   1. The code made a function to create the email address that I can’t access. I can see it manually but creating a code to find this will have to be very specific.
6. ouster (followed by &ldquo;@cs.stanford.edu&rdquo;)
   1. Too much work to find a single example of this. Also since the edu is followed by more words, it is hard to identify by itself with so much nonsense around it.
7. pal at cs stanford edu
   1. No peroids to use in between cs and Stanford. Cant create one after finding it.
8. vladlen at <!-- die!--> stanford <!-- spam pigs!--> dot <!-- die!--> edu
   1. Hard to create a specific regular expression to find this since there is a lot of trash in between.
9. **Search the web and find a couple of additional examples of obscured email addresses or phone number and report on them.**
10. Obfuscate Email Address:

<SCRIPT LANGUAGE="JavaScript">  
user = '*name*';  
site = '*domain.com*';  
document.write('<a href=\"mailto:' + user + '@' + site + '\">');  
document.write(user + '@' + site + '</a>');  
</SCRIPT>

* 1. The great thing about Obfuscate, is that the function can easily change the way a data miner can look at it. By Changing the order, or the parameters, the user computer will have no way of finding patterns in the data.

1. <!--email\_off-->*you* *email addresses, goes here*<!--/email\_off-->
   1. The adding of dumb comments around the email helps us obscure the actual email. The comments add a layer which a computer cannot easily identify unless it is trained to do so. Now, the layer might be unique and will require manual intervention to access it.

**Optimal way of Obscuring Email Addresses:**

The reason why regular expressions are useful is its ability to find patterns in text data. Consequently, what if we get rid of patterns in the way we encode the emails. A random number generator will help us choose different Obfuscate Email functions and write filler comments aroura the emails that only our own decoder can make sense of it. This will keep the scammer mining our details from getting all of them at the same time. One will have to scrub manually to be able to find any kind of pattern which will not exist.

Therefore, mixing the two types of email obscuring with a random number generator will help us keep scammers away.