

Quinn E Knudsen

NLP HOMEWORK 2: contact finder NLP

1. **Regex Pattern Matching**

For the first portion of this assignment, nine unique regex functions were written for email patterns, and three were written for phone numbers. Before adding in addition line modifications within the process\_file loop, these regex patterns yielded 105 true positives, 0 false positives and 12 false negatives. These were highly positive results, but still did not capture 12 of the email addresses hidden by the evil genius known as the ‘SpamLord!’. As an email sleuth, this first section will provide a glimpse into the file collected to expose the SpamLord! once and for all.

**Part 1** **Email Pattern Match**

* ([A-Za-z.]+)@([A-Za-z.]+)\.[A-Za-z]+')
  + This expression seeks to match one or more uppercase or lowercase letters with a middle @ sign. The sign should be a standalone, without any special characters found in many examples such as ashishg@stanford.edu.
* ([A-Za-z.]+)\s+@\s+([A-Za-z.]+)\.[A-Za-z]+')
  + This expression also seeks to match one or more uppercase or lowercase letters as well as special characters. This is designed to handle whitespaces that exist around a special character such as those found in ullman @ cs.stanford.edu.
* ([a-z]+)&#x40;(graphics.stanford).edu')
  + This regex expression matches the tricky ‘&#x40;’ string, followed by ‘graphics.stanford’ and ends with a .edu address. An example email address captured by this expression is: ada&#x40;graphics.stanford.edu.
* (\w+)\b.[A-Z].\*\b(stanford).[A-Za-z]+.edu')
  + This special regex captures a single word followed by uppercase lettersand special characters. The word also then was ‘stanford’ in the middle followed by additional upper or lowercase words and ‘.edu’ at the end. This expression handles tricky SpamLord! intricacies such as engler WHERE stanford DOM edu.
* ([a-z.]+)\b[<\>|(followed by &ldquo;][+.?@([a-z.]+).edu](mailto:+.?@([a-z.]+).edu)')
  + This expression begins similarly by searching for one or more lowercase letters and special characters. The second sections seeks and open <\> or ‘followed by &ldquo;’ special characters and the associated sign. Finally, it searchers for .edu. An example email found here would be asandra<del>@cs.stanford.edu.
* ^([a-z]+).?\bat\b\s(\W.+).edu+')
  + This regex takes in a single word followed by an uppercase and/or special character. Key to its success, this regex searches then on the word ‘stanford’ and additional upper or lowercase text and a domain name of ‘.edu’. This captures, by far, my favorite email vladlen at <!-- die!--> stanford <!-- spam pigs!--> dot <!-- die!--> edu
* ([a-z]+).at <!--.+>.(stanford).+edu')
  + This expression searches for lowercase letters followed by the expression <!--.+> ‘stanford’ and ‘.edu’.
* ([A-Za-z.\_0-9]+)\s\*(?:@|\sat\s)\s\*([A-Za-z.;\_0-9]+)\s\*(?:\.|\s|;)\s\*((?:com|edu))\W')
  + This searchers for one or more characters, digits, or periods with either an @ sign or a written out ‘at’ that is surrounded by spaces. It also includes in the search zero or more than one space, character, digits, or period followed by a period or space that ends in either ‘com’ or ‘edu’ and ends with an alphanumeric character to not capture unwanted information. An example email here is lam at cs.stanford.edu.
* ([A-Za-z.\_0-9-]+)@([A-Za-z.;\_0-9-]+)\.(-e-d-u)')
  + This last email regex captures one or more characters, digits, or periods separated by ‘-‘ with an @ before capturing the same information in the latter half. Finally, it cpautes ‘-e-d-u’ specifically as seen in this tricky example d-l-w-h-@-s-t-a-n-f-o-r-d-.-e-d-u.

**Part 2** **Phone Pattern Match**

* (\d{3})-(\d{3})-(\d{4})')
  + This first phone expression matches phone numbers that have three digits followed by three digits and end in four digits. These digit groupings must be spectated by a dash sign.
* .+(\d{3}).[^0-9](\d{3})[^0-9](\d{4})')
  + The next two expression are nearly identical. This expression in particular seeks to match the first three digits preceded by characters that do not end a line. It then matches on a noncharacter digit before seeking three new digits. Finally, part 2 is repeated before seeking four digits.
* .?(\d{3})[^0-9](\d{3})[^0-9](\d{4})')
  + This expression mirrors the prior, with the modification of the initial special character match at the beginning of the expression being optional.

1. **Line & Email Pattern Matching**

Additional coding was completed within the process\_file loop to further thwart the SpamLord!’s goal of world [wide web] domination. Nine re.sub line edits were made, one if statement was added with a re.sub, and one modification to the email parameter was made in an attempt to collect more email addresses (including those with ‘.com’).

* ('\<at\ssymbol\>', '@', line)
  + This line replaces <at symbol> with an @ sign.
* ('\<del\>', '', line)
  + This line removes <del> and replaces it with an empty space.
* ('\s?\<\!.+?\>\s?', ' ', line)
  + This removes everything inside of <\!.+?\> or inside of the brackets that would be useful in other occurrences of vladen’s spam pigs messaging.
* ('\s\(followed.+?\@', '@', line)
  + This takes everything between and including followed by and replaces it with an @.
* ('[sS]erver' , ' ' , line)
  + This replaces server with a blank.
* ('\s+(?:DOT|dot|DOM|dt)\s+', '.', line)
  + This is a sleeker way to replace DOT, dot, DOM, dt with an actual ‘.’.
* ('&#x40;', '@', line)
  + This edit is redundant with the one regex written earlier, but may be the more useful way to approach these one offs in the future rather than writing search for it. This replaces &#x40 with an @ sign.
* ('\s+(?:AT|WHERE)\s+', '@', line)
  + This replaces AT and WHERE with an @ sign.
* if '-@-' in line:
* line = re.sub( '-' , '' , line)
  + Replaces ‘-‘ with a blank only if the line has -@- to best handle the cumbersome [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.
* email = '{}@{}.edu' .format(m[0],m[1]) or '{}@{}.com' .format(m[0],m[1])
  + Intended to capture ‘.edu’ and ‘.com’ although the efficacy is uncertain based upon the one false positive and one false negative containing .com.

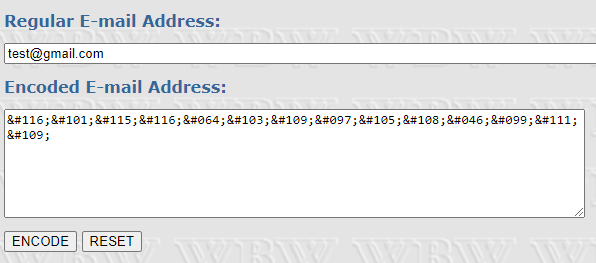
Cumulatively, this manifested in 112 true positives, two false positives, and five false negatives detailed in the appendix.



1. **Missed Opportunities**
2. jks at robotics;stanford;edu
   1. The combination of the ‘at’ and utilization of colons and semicolons helped this one escape detection. I had hoped that the regex expression '([A-Za-z.\_0-9]+)\s\*(?:@|\sat\s)\s\*([A-Za-z.;\_0-9]+)\s\*(?:\.|\s|;)((?:com|edu))\W' would capture this and/or the ‘pal at cs stanford edu’ address.
3. obfuscate(‘stanford.edu’, ‘jurafsky’)
   1. This one avoided detection as it was hard to group the middle with a symbol or expression.
4. pal at cs stanford edu
   1. This follows the same line of thought as the first missed email. A better custom expression might help find this one as well. An if statement may also help this one within the loop.
5. support at gradiance dt com
   1. Much like the previous examples, a customized if re.sub statement to convert at to @, remove the spaces, and add a period before .edu|.com. It is key to make sure .com is properly captured.
6. (650) 723-3334&nbsp
   1. I believe this avoided detection due to the & sign at the end.
7. For the two false positives, jks at robotics;stanford;edu and support at gradiance dt com, something must have occurred with the re.sub where the format became a little funky and further or more properly refined re.subs might be needed to course correct (subbing the ; or the at|dt and stripping the white space/replacing with period).
8. **Concluding Remarks**

Although it was fun making the SpamLord! a villain, I was ultimately bested in this endeavor and will need some additional recon to thwart his villainous activities. Although retrospectively, am I the true villain here in creating a WebCrawler to scrape contact information? The plot has thickened.

All kidding aside, some research on the web provides some recommendations for avoiding these crawlers such as reformatting the address (JavaScript is recommended), removing email addresses and creating forms, and using Google reCaptcha. The articles generally concluded by saying spam filters are likely the best defense. In 2015 Google claimed their filters are able to stop 99.9% of spam and misclassify spam on 0.05% of the time. This has likely increased over time. I also found this [site](http://www.wbwip.com/wbw/emailencoder.html?regEmail=test%40gmail.com&codeEmail=) that encodes your email into something like the following:



1. **Appendix**

**True Positives (112):**

{('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'),

('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'),

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

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

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

('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', 'e', 'uma@cs.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')}

**False Positives (2):**

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

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

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

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

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

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

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

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

**False Negatives (5):**

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

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

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

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

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

**Summary: tp=112, fp=2, fn=5**