Zane Alderfer

Professor Larche

IST 664

November 14, 2023

Homework 2

1.) Using the initial regular expressions provided to use in the sample code: epatterns.append(r'([A-Za-z.]+)@([A-Za-z.]+)\.edu'), epatterns.append(r'([A-Za-z.]+)\s@\s([A-Za-z.]+)\.edu'), and ppatterns.append(r'(\d{3})-(\d{3})-(\d{4})'). This returned 41 true positives and 76 false negatives.

• The first pattern I appended to the list was epatterns.append(r'(\w+)\b.[A-Z].\*\b(stanford).[A-Za-z]+.edu') which returned these two extra email addresses: **True Positives (2): {('subh', 'e', 'subh@stanford.edu'), ('engler', 'e', 'engler@stanford.edu')}** Looking at the regex, the “(\w)” covers one word, the \b is a word boundary of zero width, the [A-Z] covers the alphabet, another word boundary of zero with the word stanford and then [A-Za-z] to cover DOM in this case and then “edu” is covered at the end. This was ideal for an address like “engler WHERE stanford DOM edu.”

• The second pattern is epatterns.append(r'([a-z.]+)\b[(followed by &ldquo;]+.?@([a-z.]+).edu'). This returns: **True Positives (2): {('ouster', 'e', 'ouster@cs.stanford.edu'), ('ouster', 'e', 'teresa.lynn@stanford.edu')}.** The “([a-z.]+)” covers all of the lower case letters plus a period followed by the expression “followed by &ldquo” which both of these raw emails have and then an @ symbol with the “edu” once again at the end.

• The third pattern was epatterns.append(r'([a-z.]+)\b[<del>]+.?@([a-z.]+).edu'). This returns: **True Positives (3): {('latombe', 'e', 'asandra@cs.stanford.edu'), ('latombe', 'e', 'latombe@cs.stanford.edu'), ('latombe', 'e', 'liliana@cs.stanford.edu')}.** Aside from the other normal expression pieces that I’ve used for the previous 2 regex patterns, this regex adds a portion of “<del>” which these three raw emails have such as “asandra<del>@cs.stanford.edu.”

• The fourth regex pattern I added was epatterns.append(r'(\w+)\b.[A-Z].\*\b(stanford).[A-Za-z]+.edu'). This returned: **True Positives (2): {('subh', 'e', 'subh@stanford.edu'), ('engler', 'e', 'engler@stanford.edu')}.** So this is the second regex that was able to return Engler’s email but I was more so aiming for “uma at cs dot stanford dot edu.” The first portion once again calls a single word covering all the letters of the alphabet and then the start \* looks for zero or more of the previous item and then the word “stanford” and again every letter of the alphabet to cover “dot” and finally “edu”.

• The final regex pattern I was able to make for the emails was epatterns.append(r'([a-z]+).at <!--.+>.(stanford).+edu'). This returned only one more email which was **True Positives (1): {('vladlen', 'e', 'vladlen@stanford.edu')}.** The first portion of the expression covers the alphabet once again and then the “at <!--.+>” is very specific to this email. That email is “ladlen at <!-- die!--> stanford <!-- spam pigs!--> dot <!-- die!--> edu”. That specific piece of the regex covers those three portions of “</>.”

• Although I added 5 expressions, this only added 8 emails in total to the true positives

• The first regex pattern I made for phone numbers was ppatterns.append(r'.?(\d{3})[^0-9](\d{3})[^0-9](\d{4})'). The .? repeats any periods throughout the phone in case of the format of XXX.XXX.XXXX. The (\d{3}) is any 3 digits in any order. The (\d{4}) is exactly the same except 4 digits instead of 3. Finally the “[^0-9]” covers any start of a string 0-9 for the 3 and 4 digit regex expressions. This returned 31 phone numbers into the true positive category which can be found in the python output.

• The second regex pattern for phone numbers is ppatterns.append(r'.+(\d{3}).[^0-9](\d{3})[^0-9](\d{4})'). This returned 40 true positives. There’s obviously some overlap between the two regex patterns as well the original regex pattern given to us. The difference between this one and last one would be that the plus sign in the beginning would indicate that the expression follows can be duplicated one or more times followed by a period and then strings starting with 0-9 of length 3 and length 4.

The final list after all the regex expressions would be 101 true positives to 14 false negatives.

Assuming ContactFinder.py called in directory with data folder

True Positives (101):

{('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', 'p', '650-723-1131'),

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

('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', 'p', '650-723-0033'),

('levoy', 'p', '650-724-6865'),

('levoy', 'p', '650-725-3724'),

('levoy', 'p', '650-725-4089'),

('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', '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')}

False Positives (0):

False Negatives (16):

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

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

('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'),

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

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

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

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

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

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

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

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

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

Summary: tp=101, fp=0, fn=16

2a.) In the python file, I left the regex expressions that didn’t pan out or add anything to the list as just a comment in the code. The expressions that didn’t work are: False Negatives (16):

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

• I wasn’t able to actually view the email in the document for this person so I couldn’t tell what was going wrong.

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

• This email is completely spelled out with “at” and “dot” for the @ symbol and the periods. I was able to get a false positive with one of the commented regex (r'([A-Za-z. ]+)\s@\s([A-Za-z. ]+)\.edu') but couldn’t manage to get a true positive

**• ('dlwh', 'e', 'dlwh@stanford.edu'),**

• This email looks like d-l-w-h-@-s-t-a-n-f-o-r-d-.-e-d-u. I was able to get a false positive with r'([A-Za-z.-]+)@([A-Za-z.-]+)\.-e-d-u' but wasn’t able to get it into a true positive.

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

• I was able to get this as a false positive with the same regex from Dabo’s email as they both have the same format but once again unable to get a true positive

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

• I was once again able to get a false positive with regex (r'([A-Za-z.]+)\sat\s([A-Za-z.;]+);edu'). The format of this email looks like (jks at robotics;stanford;edu)

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

• The email format looks like (obfuscate('stanford.edu','jurafsky'). I was not sure how to approach this one.

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

• I was able to get a false positive with

(r'([A-Za-z. ]+)\sat\s([A-Za-z.]+).edu') but couldn’t get it to be a true positive. The format of the email looks like (lam at cs.stanford.edu). Adding the spaces and the “at” instead of @ didn’t account for everything.

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

• The format of this email looks like (&#x40;graphics.stanford.edu). I wasn’t sure how to approach this one exactly.

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

• Same as above

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

• The format looks like (manning <at symbol> cs.stanford.edu) I was once again able to get a false positive with

(r'([a-z.]+)\b[<at symbol>]+.?@([a-z.]+).edu') but wasn’t able to get a true positive.

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

• Same as above

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

• I was unable to find the format of the email in the original document.

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

• I used this regex (r'([A-Za-z. ]+)\sat\s([A-Za-z.]+).edu') and was able to once again get a false positive.

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

• The format for this number was odd. It was ((650) 723-3334&nbsp). I tried this regex (r'.+(\d{3}).[^0-9](\d{3})([^0-9a-z&]{8})') but wasn’t successful in returning it.

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

• The format here looks like (subh AT stanford DOT edu). I attempted this regex (r'([A-Za-z.]+)\sAT\s([A-Za-z.;]+)\sDOT\sedu') but wasn’t successful.

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

• The format for this email is (ullman @ cs.stanford.edu). Based on the regex I created, I can’t seem to figure out why this email wouldn’t fall under the scope of any of them.

2b.) An odd email I found online that I’d imagine regex could have some trouble distinguishing would be (ÿÞ\* < anyname @anydomain.com>). Another example that could be tough for regex is (\*@domain name.xxx). Another example I could think of are languages that use a different alphabet in an email address. The regex would have to be adjusted to be able to recognize emails with foreign alphabets.