**HW2 - SpamLord! Using Regular Expressions to find email addresses and phone numbers**

1.

**Email:**

([A-Za-z.\-]+)\s\*@\s\*([A-Za-z.\-]+)\.[\-Ee]+[\-Dd]+[Uu]+

Following are some of the examples:

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

Obfuscated: ashishg @ stanford.edu

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

Obfuscated: rozm @ stanford.edu

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

Obfuscated: balaji@stanford.edu

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

Obfuscated: cheriton at cs.stanford.edu

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

Obfuscated: uma at cs.Stanford.EDU

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

Obfuscated: dabo @ cs.stanford.edu

 ('dlwh', 'e', 'dlwh@stanford.edu')

Obfuscated: d-l-w-h-@-s-t-a-n-f-o-r-d-.-e-d-u

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

Obfuscated: darlene@csl.stanford.edu

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

Obfuscated: kunle@ogun.stanford.edu

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

Obfuscated: siroker@cs.stanford.edu

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

Obfuscated: widom@cs.stanford.edu

([A-Za-z.]+)\s+[Aa][Tt]\s+([A-Za-z.]+\s\*[Dd]?[Oo]?[Tt.;]?\s?[A-Za-z]+\s\*)[Dd]?[Oo]?[Tt.;]?\s?edu

Following are some of the examples:

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

Obfuscated: cheriton at cs.stanford.edu

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

Obfuscated: hager at cs dot jhu dot edu

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

Obfuscated: jks at robotics;stanford;edu

 ('lam', 'e', 'lam@cs.stanford.edu')

Obfuscated: lam at cs.stanford.edu

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

 Obfuscated: pal at cs stanford edu

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

Obfuscated: serafim at cs dot stanford dot edu

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

Obfuscated: subh AT stanford DOT edu

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

Obfuscated: uma at cs dot stanford dot edu

**Mobile:**

(\[\*\(\*[2-9]\d{2})\s\*\]\*\)\*\s\*-\*\s\*(\d{3})\s\*-\*\s\*(\d{4})

The phone numbers in North America have the format NXX-YYY-ZZZZ. Where N cannot be 0 or 1. Hence, the above REGEX works to extract the numbers.

Some of the examples are below along with their obfuscated forms:

('ashishg', 'p', '650-723-1614')

Obfuscated: (650)723-1614

 ('ashishg', 'p', '650-723-4173')

Obfuscated: (650)723-4173

 ('ashishg', 'p', '650-814-1478')

Obfuscated: (650)814-1478

 ('bgirod', 'p', '650-723-4539')

Obfuscated: (650) 723-4539

 ('bgirod', 'p', '650-724-3648')

Obfuscated: (650) 724-3648

 ('bgirod', 'p', '650-724-6354')

Obfuscated: (650) 724-6354

 ('cheriton', 'p', '650-723-1131')

Obfuscated: 650-723-1131

 ('cheriton', 'p', '650-725-3726')

Obfuscated: 650-725-372

 ('dabo', 'p', '650-725-3897')

Obfuscated: (650) 725-3897

 ('dabo', 'p', '650-725-4671')

Obfuscated: (650) 725-4671

There were also some false positive which were picked up from different parts of the html files. Eg. CSS, particularly mso-font-signature.

('bgirod', 'p', '593-177-5729'),

 ('latombe', 'p', '611-985-1107')

 ('latombe', 'p', '689-107-3717')

 ('latombe', 'p', '689-676-9869')

 ('latombe', 'p', '698-693-6769')

 ('latombe', 'p', '794-676-9869')

 ('latombe', 'p', '830-552-1902')

 ('latombe', 'p', '870-145-1791')

 ('nass', 'p', '375-742-1353')

 ('nass', 'p', '910-769-9828')

The following REGEX help in fitting another 9 false negatives:

Though these are specific regular expression, it helps in fitting 9 email IDs which is close to 7% of the total number of email IDs, so it has been adding to the program.

([A-Za-z.]+)\<del\>@([A-Za-z.]+)\.edu

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

Obfuscated: asandra<del>@cs.stanford.edu

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

Obfuscated: latombe<del>@cs.stanford.edu</a>

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

Obfuscated: liliana<del>@cs.stanford.edu

([A-Za-z.]+)\&\#x40\;([A-Za-z.]+)\.edu

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

Obfuscated: ada&#x40;graphics.stanford.edu

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

Obfuscated: melissa&#x40;graphics.stanford.edu

([A-Za-z.]+)\s\*\<at symbol\>\s\*([A-Za-z.]+)\.edu

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

Obfuscated: manning <at symbol> cs.stanford.edu

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

Obfuscated: dbarros <at symbol> cs.stanford.edu

([A-Za-z.]+)\s\(followed by [&ldquo;"]+@([A-Za-z.]+)\.edu

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

Obfuscated: ouster (followed by &ldquo;@cs.stanford.edu

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

Obfuscated: teresa.lynn (followed by "@stanford.edu

Complete output:

True Positives (113):

{('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'),

 ('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', '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', 'e', 'pal@cs.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', '650-723-3334'),

 ('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', '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'),

 ('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 (10):

{('bgirod', 'p', '593-177-5729'),

 ('latombe', 'p', '611-985-1107'),

 ('latombe', 'p', '689-107-3717'),

 ('latombe', 'p', '689-676-9869'),

 ('latombe', 'p', '698-693-6769'),

 ('latombe', 'p', '794-676-9869'),

 ('latombe', 'p', '830-552-1902'),

 ('latombe', 'p', '870-145-1791'),

 ('nass', 'p', '375-742-1353'),

 ('nass', 'p', '910-769-9828')}

False Negatives (4):

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

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

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

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

Summary: tp=113, fp=10, fn=4

**2a. Well obfuscated edu email ids:**

Jurafsky:

In this file, a script has been written to make the email id obfuscate so that no spammer can find it using regular expression.

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

<script type="text/javascript"><!--

function obfuscate( domain, name ) { document.write('<a href="mai' +

'lto:' + name + '@' + domain + '">' + name + '@' + domain + '</' + 'a>'); }

// --></script>

<script> obfuscate('stanford.edu','jurafsky'); </script>

vladlen:

In this file, comments have been added in between to prevent regular expressions from matching the email address.

vladlen at <!-- die!--> stanford <!-- spam pigs!--> dot <!-- die!--> edu

Engler:

In this file, where and DOM have been placed instead of @ and . respectively. REGEX can be written for this, however, since this is a rare scenario, I choose to ignore it.

engler WHERE stanford DOM edu

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

**Com email id:**

Ullman:

This is an email id ending with com which is outside the scope of my analysis.

support at gradiance dt com

2b.

The following is a way to hide email ids from a source file.

One of the ways to obscure the email addresses has been seen in the file Jurafsky. This can also be done by using CSS.

<style>

my-email::after {

content: attr(data-domain);

}

my-email::before {

content: attr(data-username) "\0040"; <!--@ sign-->

}

</style>

<my-email data-username="jurafsky" data-domain="stanford.edu"></my-email>

There are some other methods using JavaScript with which we can perform some encoding and decoding which will allow us to add a fake email address to our HTML file but for which display shows as the true email id.

Eg. Using script to reverse the email id.

Using script convert ASCII values for each character separated by a delimiter to the actual email id in the front end. And so on.