SI 630: Homework 0 – Regular Expressions

Due: Wednesday, January 17, 5:30pm

Email addresses are everywhere online. Especially in academic web pages, people provide their email as way of easily getting in touch with them. However, unscrupulous spammers also look for these addresses to send unwanted email to people. As a result, some web page authors have resorted to obfuscating their address so that a human could still figure out what the address is without a machine being able to easily detect it. For example, someone might write `myname@domain.edu` as `myname at domain dot edu`.

You've been asked to perform a security audit for a large university. They want to know what kinds of email addresses might be recoverable from each web page. Conveniently, they've already put together all of the web pages for you into a single file, where each page is one line. Further, every page is guaranteed to have one email on it at most (some could have none). However, they have no idea how the addresses are formatted!

Being a supportive university, they have also given you some of their manually-checked pages and associated emails to test your solution on, `trial-pages.txt` and `trial-pages.emails.txt` and a scoring program to verify your output, `scorer.py` which takes in two files, the gold standard and your solution, and prints the score. The university staff are kind enough to also point out that since this is a manual labeling, they probably left off a bunch of edge cases for how people obfuscate their emails.

Problem 1 (90 points). Write a program that uses regular expressions to extract and canonicalize email addresses from web pages. Hint: groups may come in handy here. You will be provided with a large file of web pages on Canvas, `webpages.txt`, where each page is on a separate line. Your program will write to a user-specified file the canonicalized email address found on each page or `None` if no email address was found. By canonicalized, we mean that if the author wrote `myname at domain dot edu`, you would output `myname@domain.edu` in your file. Your output should have the same number of output lines as the input file.

Your code should work on the command line and take two arguments: (1) the name of the input file to process and (2) the name of the output file to write to. For example, if you're writing in python, you would execute your program like

```
python extract_emails.py webpages.txt email-outputs.txt
```

You should submit three items as a part of your solution:

- Your code
- A txt file named `email-outputs.txt` with the emails you extracted
- A txt, pdf, or docx file with your answer to the following question

Problem 2 (10 points). Your effort should hopefully show that with regular expressions, you can find most kinds of obfuscated email addresses. However, this discoverability doesn't bode well for people who still want to obfuscate their addresses. In a few sentences, describe your recommendations for strategies people could try to hide their email addresses from spammers.