Instructions for Text Anonymization Using Python

We have built a prototype system to remove officer identification from narrative data. Using a list of police officer names the system will scan each text for matching names and replace them with an hash ID assigned to the officer. Badge numbers are also replaced.

Our prototype system works for test data we have created but needs to be tested on actual data by CMPD. Once this has been done we can make any improvements necessary to ensure the system functions correctly and anonymizes all texts.

Note that the hash IDs assigned to the officers by the current system are not the actual hashes (we do not have the officer names and hashes so we are unable to do this our end). The final code will need some slight modifications to enable us to use the correct hashes.

The code requires you have Python 3.5 installed along with the `pandas` package (if you don't have it then type `pip install pandas` into the command-line/terminal window and hit enter to install it). The other packages used are string, re, random, and csv, which should all be included with any Python distribution.

Instructions for use (using test data):

First run the system on the test data we created to ensure that you get the same results. All officer names in the test reports should be replaced by an ID code and all badge numbers should be replaced by the term BADGENUM.

The compressed folder you have been sent contains three files: an excel spreadsheet containing some test officer names (names_test.xls), an excel spreadsheet containing some test police reports (reports_test.xls), and a Python script that will anonymizes the text (anon.py). Please extract these files and ensure they are all placed in the same directory. Move this folder to a useful place (e.g. Home/anon).

Open the command line or terminal and change directory to enter the folder (e.g. `cd anon`). When you are inside this folder you can run the Python script my typing the following command, `python anon.py`

Follow the instructions in the terminal and enter the names of the two input files when asked.

The script will then run and print out the original reports and the anonymized versions in the terminal screen.

The anonymized reports will be saved in a new file called anon_reports.csv that will appear in the same directory. You can open this file and view the anonymized reports.

Instructions for use (using real data):

To test the system on real data all you need to do is create two new input files with the same structure as the test input files.

1. Names.xls should contain three columns titled last_name, first_name, and middle_name. This information should be easy to extract from your databases and convert to Excel format.
2. Reports.xls should contain one column titled narrative. Each row will contain an individual narrative or report.

Next you can simply run the code and input the new file names when requested. The system will print out a lot of data if there are a large number of reports. Before running it we suggest you open the Python file and delete the ## in front of line 289 (afterwards it should look like `reports = random.sample(reports, 20)`) and save the file. This will enable you to select 20 random reports and evaluate the results. Running the program on a large number of reports may take some time and will print out a large amount of information in the terminal window.

It is important to note that this system will currently only remove the names of officers listed in the input file. If there are names of officers that appear in the anonymized text then the names were likely not in the input. To include these names for removal you can simply add them to the names file. Our final system will include the ability to easily do this as the reports are anonymized.