

Classification on emails that contain personal information

Lewis Munro

2469780

Motivation

Government funded companies, get their funding through tax i.e. schools, hospitals etc, are legally required to release the data they store to the public unless the data being released contains personal information. Personal information can be many things, health conditions, religion, confidential information etc. It takes a long time to go through each individual file and classify if it is personal or not and a way to do this automatically is becoming more necessary.

Aims

This project will develop a webapp that users can upload email collections to, these collections will be automatically classified, and the results are then displayed to the user. The webapp will also provide reasons for an email being considered personal and allow for the user to classify the emails further manually. Graphs and data will be given to show more reasoning for the classifications made and provide the user with further details about their collection

Progress

- Research has been done on classifiers, specifically the best one to use in this scenario
- Tutorials on different classifiers have been done to give a better understanding on how to use them
- Framework for the webapp has been chosen, Django is used to build/ host the webapp while python is used for the classifier
- The classifier is complete using Random Forest from scikit learn
- The webapp foundations are complete allowing for the classifier to be run and the data to be uploaded to an SQL database
- The results re displayed in a table with all the emails, the emails can be clicked to re direct to a page with more specific details of that emails classification
- Graphs are implemented to show more data about the collection, number of personal emails, who has sent the most personal emails and the most important features from the classifier
- The design of the webapp is mostly done, the table still needs congifering

Problems and risks

Problems

- I did not take into account how I would store the data from the classifier therefore it had to be reconfigured multiple times until a valid solution was found
- The pre processing script I was given did not work with the email collection I was given therefore some time was spent fixing it which delayed some other tasks
- The classifier script was originally a jupyter notebook file and Django had troubles running it so I had to convert the notebook to a standard python file.
- I had to do extra research into Django plugins to get the data to display correctly which delayed some other tasks

Risks

- It is hard to tell if the classifier is working correctly, some of the important features do not seem like they would correctly identify if an email is personal. Fix: add my own stop words to get rid of some of the features i.e. "www", "com", "@".
- The webapp has a lot of dependencies and may not run correctly on other systems. Fix: have a setup file which install all the dependencies to a machine before running the webapp.

Plan

- Week 1-2: finish implementing all features for the webapp
- Week 3: plan user testing/ possible topic modelling
- Week 4-5: carry out user testing
- Week 6-7: implement/ fix any issues brought up in the user testing
- Week 7-10: write up dissertation