TSOP proposal.

NOTICE: this is a draft of a Tor Summer of Privacy 2015 proposal.

Motivation.

When people wish to publish online anonymously there are many potential ways to be de-anonymized. There exist easy to use tools like the Tor Browser and MAT for anonymizing Internet traffic and document meta-data. However, even when these tools are properly used publishers still risk being de-anonymized by stylometric analyses of their texts. Anonymouth [0] is a Java application that helps authors to alter their writing style and better understand their anonymity. However, Anonymouth is not an easy-to-use tool like the Tor Browser is. It requires the user to select the feature set and machine learning method, which is confusing and therefore unsafe for most users. Also Anonymouth has not been maintained for over two years and has not incorporated recent results from the field of computational stylometry, most notably the detection writing obfuscation.

Proposed contribution.

I would like to build an application for authors who wish to publish anonymously. The main functions of the application are:

1. Provide user with information about what the application can and cannot do for them.
2. The user can present one sensitive document and multiple documents representing their regular writing style. The application gives the user an understanding of their anonymity using a dataset of writing samples from other authors. This dataset ships with the application.
3. The user can also provide text samples from other authors (e.g. from persons in their anonymity set) and determine how anonymous they are within that set.
4. The user can imitate another author (like fan fiction) and see how well they perform at this.
5. Gender attribution. User is notified what gender is detected.
6. Detection of obfuscation. User is warned when text obfuscation becomes detectable.
7. For steps 2-6 the user can get suggestions/recommendations as how the text should be altered in order to hide themselves better among other authors.

Deliverables.

The first version of this application uses a command line interface (CLI). This is because my expertise does not lie in building GUI. The content that the user provides to the application should be plain text files. The CLI could work with the following commands:

user@user:~/ APP [options] [input files]
[APP output]

user@user:~/ APP --gender-attribution --input-document [document to be analyzed]
Gender detected: [gender]
How to 'change gender' of this document:
[Recommendations]

user@user:~/ APP --anonymity-score --input-document [document to be analyzed] --author-

documents [folder with the authors public documents]
Anonymity-score: [xx]%
[Explanation of anonymity score]
How to improve anonymity-score:
[Recommendations]

The application would require one of these options:

(1) --gender-attribution
(2) --anonymity-score
(3) --imitation-score
(4) --obfuscation-detection
(5) --help

Where options (1)-(3) can be combined with option (4).

While the application should communicate all relevant information to the user, I believe it should also hide which exact methods are used and why. For a more technical explanation of the application a reference to the development documentation, code and relevant scientific papers will be made.

Creating a good working CLI application will be my main focus.

What the application will not do:

The application will not alter a text in an automated way. Rather it will provide suggestions/recommendations on what characteristics should be changed.

The application will not have an integrated text editor.

The application will not provide live feedback. After a text has been adjusted the user should re-run the application.

Project realization.

While performing research for my thesis (See section personal background) I have separately implemented contributions 2-6 described in section 'Proposed contribution'. The 7th contribution I plan to implement in a similar way as Anonymouth has done, described in this paper [1].

The work that I would be doing during the TSOP is:
- Combine existing peaces of research code into a usable program. (contributions 2-6)
- Extend the program with a 'recommendations' component as described in [2]. (contribution 7)
- Write documentation about what the program can and cannot do for users. (contribution 1)

The code that I have already written is in python and uses the nltk, pattern, numpy and scipy libraries.

Planning:

May 25: Start working on combining existing code into an application, implement CLI and the general flow of the program.
June 08: Start building a separate, proof of concept of the recommendations component. (contribution 7).

June 15: Think of various ways to communicate the recommendations to the user.
June 15: Implement recommendations into the application, deliver basic application with CLI.
June 29: Week for writing documentation, code commenting and preparing for mid-term evaluation.
Juli 06: Start work on 'smart' recommendations. I am thinking of:
- Synonym suggestions. These help change the letter frequency, vocabulary richness and word length distributions.
- Pointing out which specific part of a text is responsible for a revealing feature value.
- Provide canonical examples of changes. This can be useful for features that are difficult to understand.
Juli 27: Away for a week.
August 03: Continue work on smart recommendations or start working on a GUI.
August 17: Wrapping up. Writing documentation, making sure all code is commented, polish the work.

Personal background.

I am a research master student in Artificial Intelligence at the UvA (Netherlands) with a focus on natural language processing and machine learning. During my bachelor and masters I have worked within groups of between 2 and 9 members with often different technical backgrounds. During the last year of my bachelor I worked at a startup with 2 other programmers for 5 months. There I was responsible for data analysis using NLP and adding features to back-end services. Also during my bachelor I designed and implemented the front-end of a virtual book club [2], a site for following discussions and books and engaging with other readers on the platform.
The past 6 months I have been working on my thesis which is about computational stylometry in adversarial settings. I am currently writing my thesis of which I will send a draft along with my proposal. The major results and accomplishments are already documented in this early draft. However, the writing is still very early stage, missing introduction and conclusion chapters and not as coherent as my final report will be.
If I would be selected for TSOP I would pause finishing my thesis to work full time for the Tor Project from May 25 until August 25, except for a one week vacation. During this period I can write a bi-monthly report to the tor-dev mailing list on my progress. My thesis supervisor (Ilya Markov) has given me a lot of freedom during my research, while sometimes providing reflection or suggestions. This worked very well for me and I hope to have a similar relationship with my Tor mentor.
After my TSOP I would continue to follow the subject of computational stylometry and contribute to this project where I see an opportunity to improve anonymity of authors. I would also consider writing a basic GUI.
I am very happy that the topic of my thesis has allowed me to combine my two passions: Privacy enhancing technologies (PET) and Artificial Intelligence applied to Natural Language Processing (AI/NLP). By applying for the Tor Summer of Privacy I hope to continue contributing to PET and working with AI/NLP. I would like to use my scientific research while making a usable and free application for authors.

Please contact me at remi.py@yandex.com for any questions you may have.

[0] https://github.com/psal/anonymouth
[1] https://www.cs.drexel.edu/~sa499/papers/anonymouth.pdf
[2] http://www.thebookclub.nl (Dutch, 2013)