



Anonymouth

Software Requirements Specification

V1.0

ArmonEntezari
Karan Hansaria
Chris McGuire
Jon McGrath
Paul Vu

Table of Contents

1.	Introduction.....	3
1.1	Abstract.....	3
1.2	Purpose.....	3
1.3	Intended Audience.....	4
1.4	Background.....	4
2.	Functional Requirements	4
2.1	Architecture	4
2.2	GUI	5
2.2.1	Main Screen.....	7
2.2.1.1	Text Area	7
2.2.1.2	Feature and Suggestion Area.....	8
2.2.1.3	History Area	11
2.2.1.4	Legend Toolbox.....	12
2.2.1.5	Submit Button	12
2.2.2	Project Wizard.....	13
2.2.3	File Menu.....	15
2.2.4	Edit Menu.....	16
2.2.5	Tools Menu.....	17
2.2.6	Windows Menu	18
2.2.7	Help Menu.....	18
3.	Non-Functional Requirements.....	19
3.1	Software Constraints.....	19
3.2	Other Constraints.....	19
4.	Use Cases.....	19
4.1	Initialize Writing Sample.....	19

4.2	Analyze Writing Sample.....	20
4.3	Make Change for Anonymity.....	20
	4.3.1Change Punctuation.....	20
	4.3.2Offer Synonyms.....	20
	4.3.3Restructure Sentences.....	20
5.	Future Improvements.....	21
6.	References.....	21
7.	List of Figures.....	21
8.	Glossary.....	22

1. Introduction

1.1 Abstract

Anonymouth is designed for anyone who wishes to hide their writing style. The target audience is writers who want to keep their writing from being identified. Anonymouth interfaces with a stylometry tool called Stylo that analyzes a piece of writing and sends its analysis to our tool. Anonymouth then uses that analysis to suggest improvements to hide the author's identifiable writing characteristics.

Data from the stylometry process is interpreted after Stylo finishes its analysis. The sentence structure, common word usage, the complexity of the vocabulary, punctuation usage, and the number of distinct words in a writing sample are examples of the types of data that can be interpreted. Anonymouth then uses this data to recommend changes in the writing sample by offering synonyms for key words, changing the punctuation in sentences, or rewording sentences to mimic the writing style of another author. The goal is to give the writer the ability to easily hide their writing style.

User interaction is mainly with a GUI that resembles a word processor and allows the user to anonymize a new piece of writing or import previous work.

1.2 Purpose

This document defines the specifications for the Anonymouth tool.

1.3 Intended Audience

The document is intended for the designers, stakeholders, testers, coders, and future users of Anonymouth.

1.4 Background

Many people require anonymity in order to protect themselves from harm. Anonymouth is also intended for a broader audience of writers who are curious about their personal writing style and wish to see more about it.

Stylometry is the study of linguistic style. It examines characteristic features of the way people communicate. It was used to analyze the writing styles of writers and identify writers who wished to remain anonymous. An example of the identification of anonymous authors using stylometry techniques is the federalist papers. James Madison, John Jay, and Alexander Hamilton [2] wrote the federalist papers anonymously. However, using stylometry tools, historians were able to identify the authors of this document.

2. Functional Requirements

Priority	Description
1	This is the highest priority level; all requirements of this level must be fully satisfied and verified in order for the software system to be released.
2	Requirements of this priority are not expected to be verified by the current release of Anonymouth. The unfulfillment of these requirements must not create hazards for the system. However, these requirements are expected to be fully satisfied and verified by the next minor release of the system. The features satisfying these non-verified requirements must be clearly documented to the end user as such.
3	Requirements of this priority are not expected to be fully satisfied in the current release of Anonymouth. But, these are expected to be fully satisfied and verified at the next major release of the system.
4	These requirements are of the lowest priority and are not within the current scope of the system design. These requirements are included only to indicate where the software is expected to change in future development.

[1]

2.1 Architecture

Anonymouth will be designed to function directly alongside its sister project, Stylo. The two applications will run in parallel and communicate using pipelining methods. The purpose of the entire system is to provide an authorship-hiding writing interface. Users can submit

writings and receive feedback measuring the anonymity of their writing, as well as suggestions for specific changes to improve anonymity. Anonymouth will receive feature data from Stylo and will use this data to determine changes to suggest to the user which will cause the greatest increase in anonymity.

The system will consist of a backend which will use feature data to determine suggested changes for the writing, and a front-end GUI which will convey these suggestions to the user.

0010 - The System will be able to call Stylo and send it a corpus and a document to be analyzed (Priority 1).

0020 - The System will receive a list of features that identify the author of the document from Stylo (Priority 1).

0021 - The features received from Stylo will have a rating associated with them noting how important that feature was in identifying the author (Priority 1).

0030 - Anonymouth will be able to take the features identified from Stylo and recognize them in the submitted documents (Priority 1).

0040 - Anonymouth will be able to come up with multiple suggestions for each feature for how the user could change their document to make it less identifiable (Priority 1).

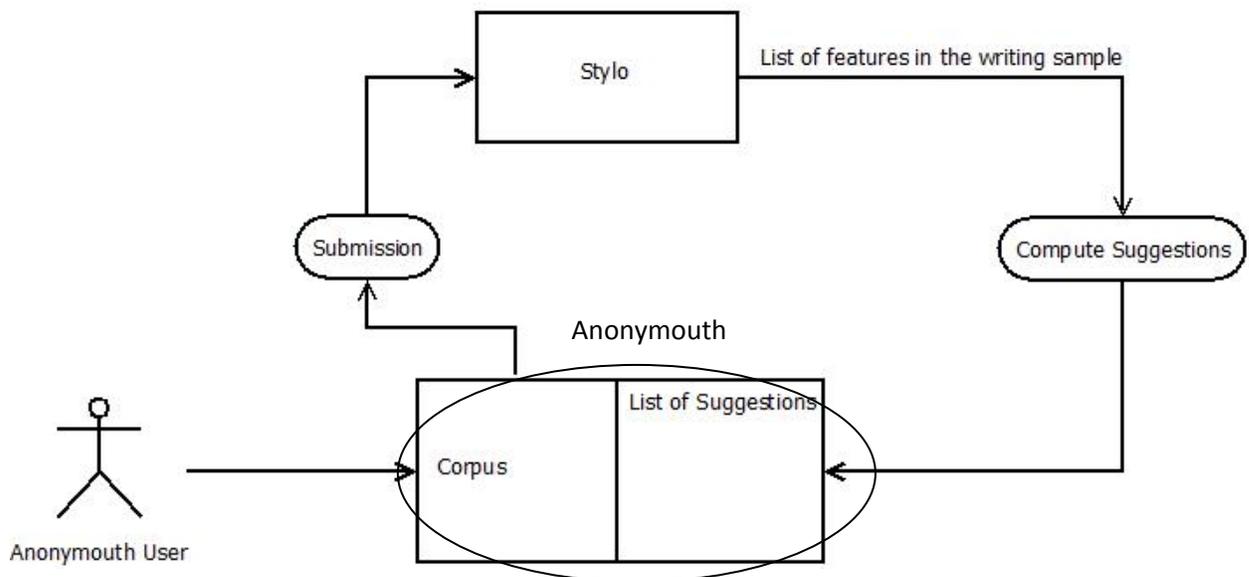


Figure 1 – Higher level architecture between Anonymouth and Stylo

2.2GUI

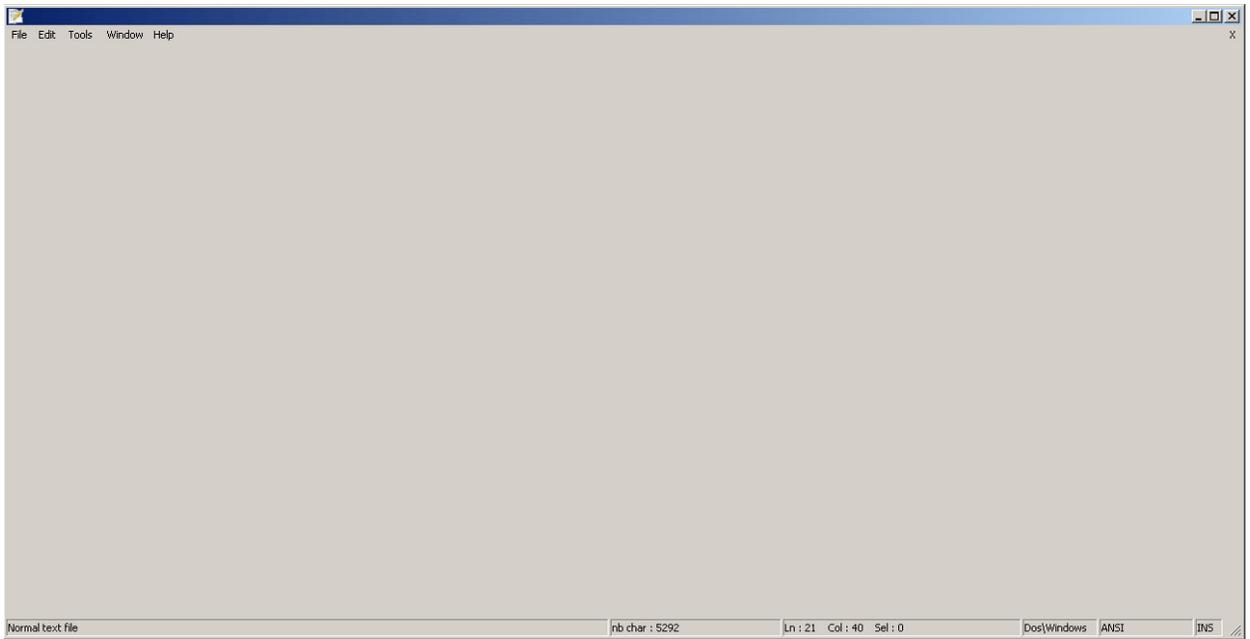


Figure 2 - Anonymouth upon start-up

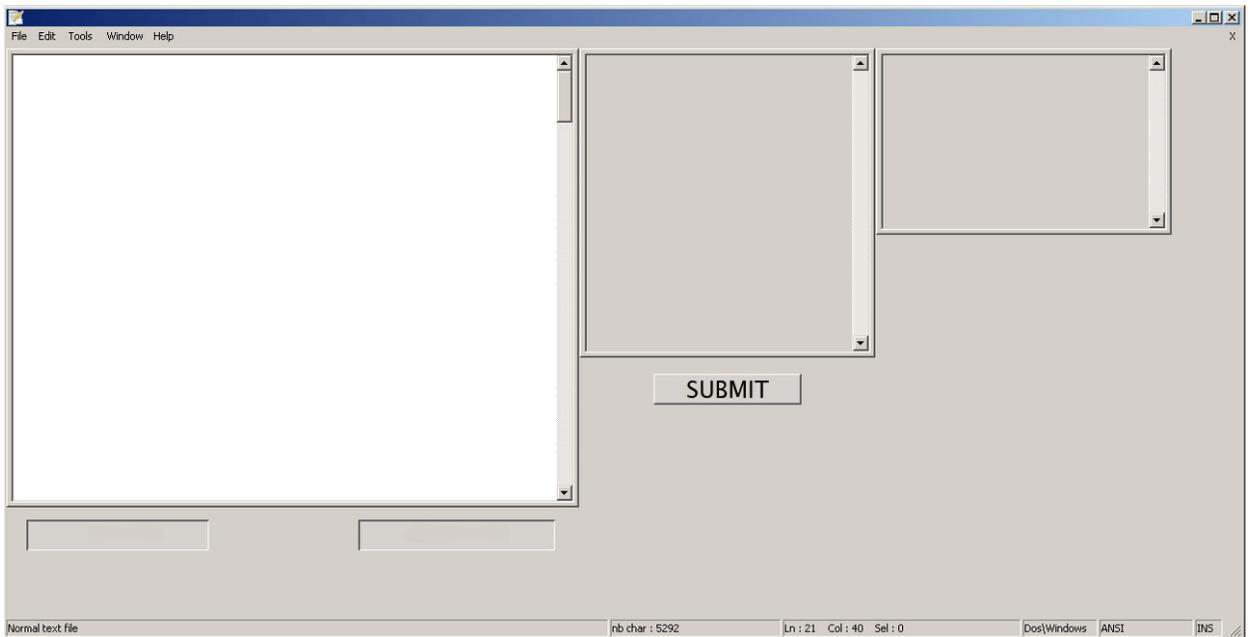


Figure 3 - Anonymouth with a blank new document

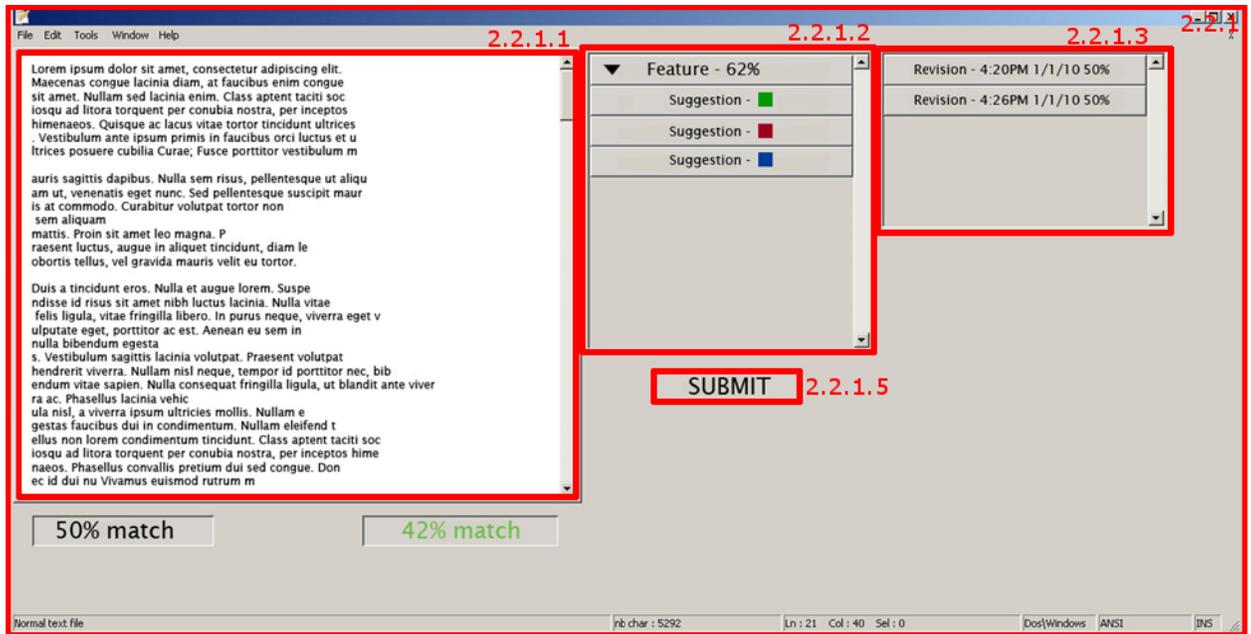


Figure 4 – Main Screen after submission

2.2.1 Main Screen

0050 - The main screen of the GUI that the user is presented with consists of a menu and an application window with three separate sections: Text area, Feature/Suggestion area and History area(Priority 1).

0060 - Initially, the application window will be blank except for a button that says “Project Wizard” (Priority 1).

0070 - Upon opening a new “project,”creating one, or going through the wizard, the application will load up the main sections(Priority 1).

2.2.1.1 Text Area

The Text Area represents the word processing aspects of Anonymouth. It is where the user and make changes to the text of the document they are working on.

0080 - The user will be able to enter the text that they wish Anonymouth to analyze into the text area(Priority 1).

0090 - Copy and paste will give the user a way to import his or her work(Priority 1).

0100 - Users are also able to edit the document directly with standard text processing functions to submit for the analysis – copy, paste, undo, etc.(Priority 1).

0101 - The text area will allow for standard character input from the keyboard (Priority 1).

0102 - The text area will resemble a typical graphical word processor such as notepad and will accept mouse interaction by having the mouse click change the cursor(Priority 1).

0103 - The text area will accept click and dragging to highlight areas(Priority 1).

0110 - Directly underneath and linked to the text area are two sections which hold the percentage of likeliness that the current document identifies its author based on previous works(Priority 1).

0111 - The section on the left holds the piece’s current likeliness of author identification which stays static until the piece is submitted as a new version with the Submit button when it will update (Priority 1).

0112 - The section on the right reports the piece’s “new” percentage which will track the piece’s changes to the likeliness (Priority 1).

0113 - Both these sections will be grayed out until the user has clicked Submit (Priority 1).

2.2.1.2 Feature and Suggestion Area

The Feature and Suggestion Area will be where the user can see what aspect of the documents identify the writer and how to change them to make them more anonymous.

0120 - The Feature and Suggestion area will be directly to the right of the text area. (Priority 1).

0130 - This section will initially be blank until the user has submitted a document for analysis (Priority 1).

0140 - The section will populate after the program has analyzed the piece that the user submits (Priority 1).

0150 - The section will give feedback to the user using a cascading tree structure of features that represent the document's identifiable properties unique to the user's writing style(Priority 1).

0160 -Clicking on any individual feature will expand down further into a list of suggestions(Priority 1).

0170 -These suggestions, when clicked, scroll the text area (2.2.1.1) to the location of that feature and subsequently highlight it(Priority 1).

0180 - The appropriate area will then be highlighted with a corresponding color for that type of feature (Priority 1).

0181 - These colors can be found and categorized in the legend (2.2.1.4) (Priority 1).

0190 - Hovering over a suggestion in the feature and suggestion area with the mouse will give the user a tooltip to right click for more options(Priority 1).

0191 - This submenu will let the user perform more actions based on the suggestions (Priority 1).

0192 - These options are: replace, compare, and ignore (Priority 1).

0200 - Selecting "replace" replaces the user's feature with the suggestion (Priority 1).

0201 - Selection "replace" will update the new likeliness of the document accordingly.

0210 - Selecting "compare" updates the percentage temporarily without replacing the text (Priority 1).

0211 - Clicking anywhere in the application will reset this value to its pre-compare state.

0220 - Clicking ignore will allow the user to ignore that feature and its suggestions(Priority 1).

0221 – An ignored suggestion will be removed from the list and no longer be visible (Priority 1)

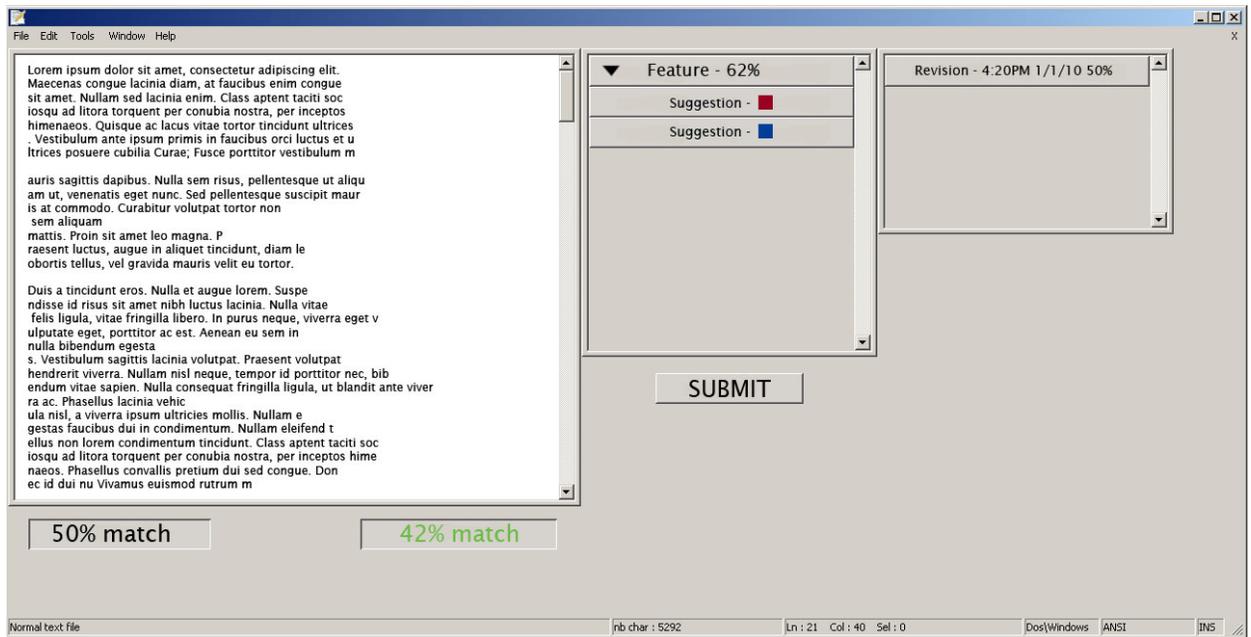


Figure 5 – Main Screen after ignoring the “green-colored” suggestion

0230 - The user will be able to undo an ignored feature by right clicking in the feature and suggestion area, which will cause a tooltip to appear with an “undo ignore” option. (Priority 3).

0240 - If there are no suggestions, this submenu appears, but the first two options are disabled (Priority 1).

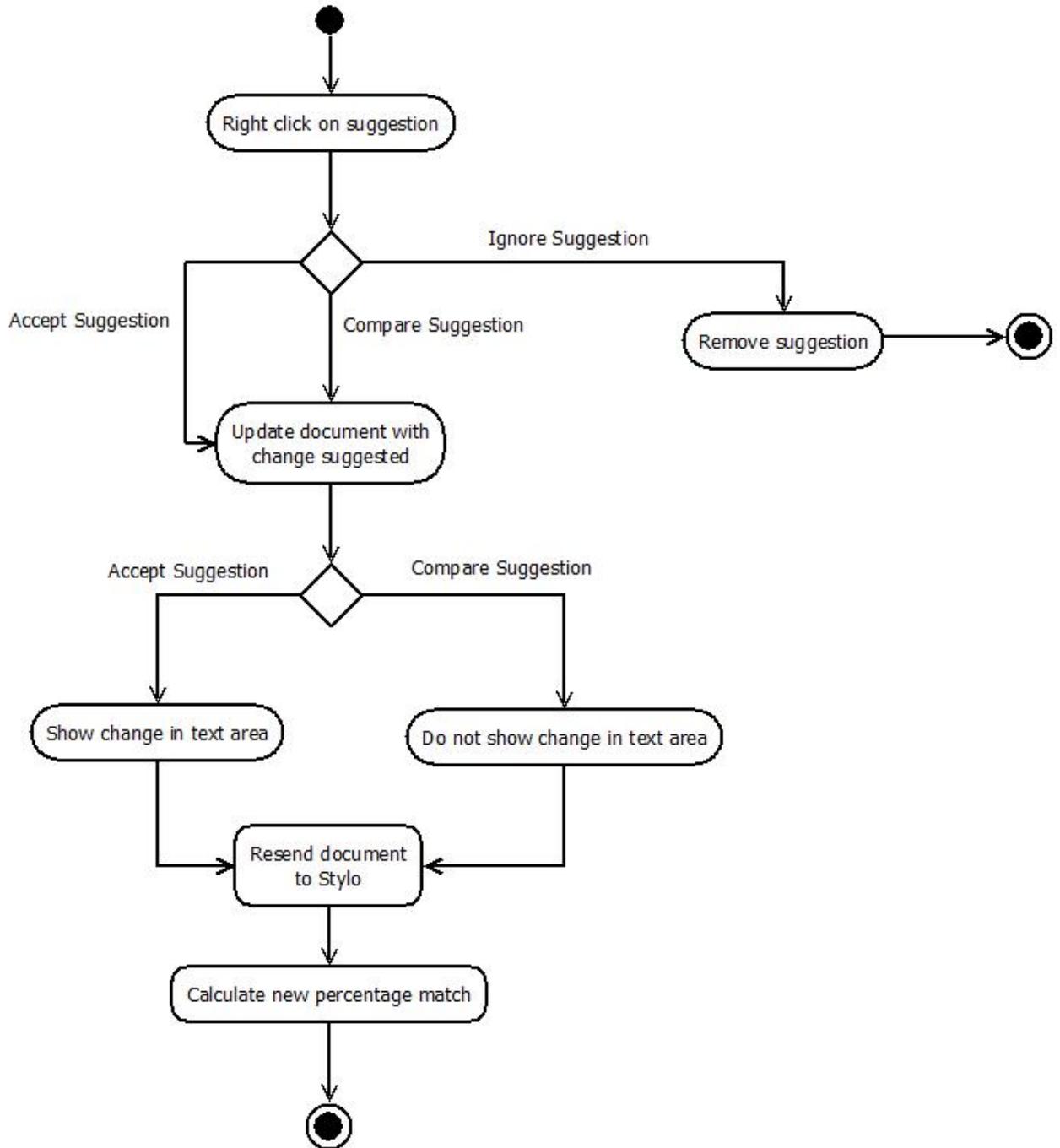


Figure 6 – Flow of actions after right clicking a suggestion in the Features and Suggestions Area (2.2.1.2)

2.2.1.3 History Area

The History Area will allow the user to see all the previous revisions to the documents and how they have affected the identifiability of the document.

0250 - The History Area will be directly to the right of the Feature and Suggestion area (Priority 1).

0260 - The History area will display previous revisions to the file with a history of previous feature vectors (Priority 1).

0270 - The feature vectors will be the percentage of how identifiable that revision of the document was (Priority 1).

0280 - The user will be able to double click the revisions to revert the text area (2.2.1.1) to that previous version (Priority 2).

2.2.1.4 Legend Toolbox

To work with the suggestions area and the feature highlights will be a dynamic legend which explains which colors represent which feature category.

0290 - Beneath the feature and suggestion area will be a dynamic legend which color codes the features by category (Priority 1).

0300 - Clicking on a color in this toolbox will filter the suggestions accordingly in the feature and suggestion area (2.2.1.2) (Priority 1).

0310 - The default filter is none and all suggestions will show initially (Priority 1).

0320 - This window will be floating, detached from the main area, moveable, and closeable (Priority 2).

0330 - The window can be toggled off and on in the windows menu (2.2.6) (Priority 2).

2.2.1.5 Submit Button

Below the three windows is a single button. The button's function is to submit the piece for evaluation.

0340 - The window will contain a submit button (Priority 1).

0350 - When clicked, the writing in the text area (2.2.1.1) will be evaluated (Priority 1).

0360 - After the evaluation is complete, the features tree (req 0120) as well as the new percentage match (req 0110) will be updated with new values for the submitted writing (Priority 1).

0370 - Clicking Submit will also save the current version and its statistics in the history (Priority 1).

2.2.2 Project Wizard

The project wizard will be a series of three screens to facilitate the process of anonymization.

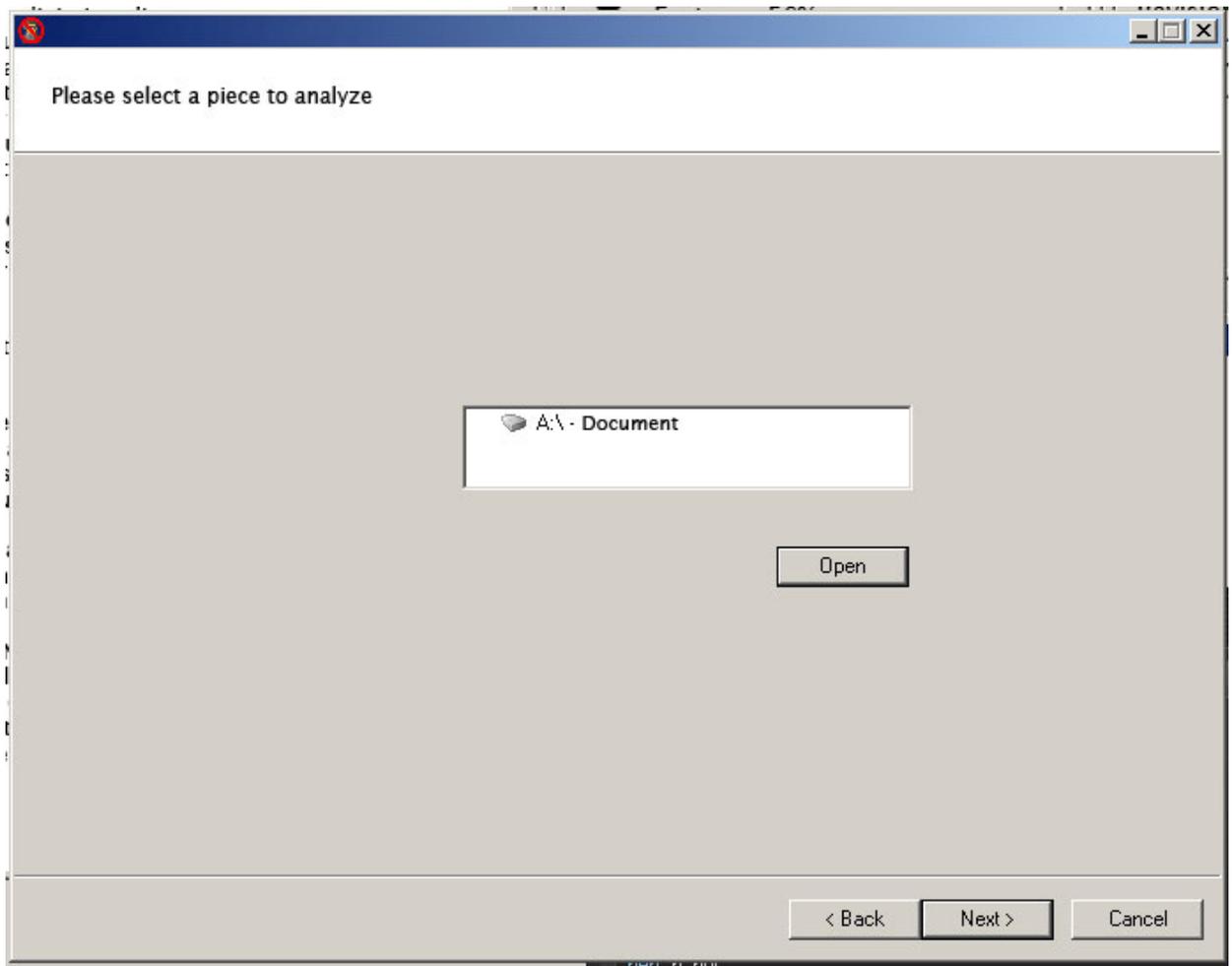


Figure 7 - Step 1 of Project Wizard (2.2.2)

0380 - After selecting the project wizard, a window will open asking the user if they would like to import a document (Priority 1).

0381 - The import document window will have two buttons, “yes” and “no” (Priority 1).

0390 - The “yes” button will open a file browser window to allow the user to import a document (Priority 1).

0391 - The file browser window will only display file types that are able to be imported (rtf, doc, txt, odt) (Priority 2).

0392 - If there is an error when importing a file, an error message will be given saying “An error has occurred, please try again” (Priority 1).

0393 - The file browser window will have two buttons, “open” and “cancel” (Priority 1).

0394 - Clicking the cancel button will take the user back to the import document window (req 0380) (Priority 1).

0395 - Clicking the open button with no file selected will take the user back to the import document window (req 0380) (Priority 1).

0396 - Clicking the open button with a file selected will load that file into the text area once the wizard is done and open the corpus selection window of the wizard (Priority 1).

0397 - Clicking no on the import document window will have the text area be blank once the wizard is done and open the corpus selection window of the wizard (Priority 1).

0400 - The corpus selection window will contain a file browser when the user can select a folder or files containing their work (Priority 1).

0410 - The corpus selection window will list all of the files that will be used in the corpus (Priority 1).

0420 - Documents, once added, can be removed through the action of clicking on an item and clicking on a button marked with a red ‘x’ (Priority 1).

0430 - Once the corpus has been chosen, users are able to move to the final screen which will be a review of their choices and let the user, optionally, save the project to a file (Priority 1).

0440 - Clicking on Okay button will take the user to the main screen and will populate it (Priority 1).

2.2.3 File Menu

The file menu will provide the user with access to the application's most basic functions.

0450 - The window will have a file menu (Priority 1).

0460 - The file menu will have sub-options: New Project, Open Project, Save, Save As, Close Project, Recent Projects, and Export (Priority 1).

0470 - The New Project option will open the sections of the main screen and allow a new project to be started (Priority 1).

0480 - Clicking New Project option the will ask for sample writings to analyze against (Priority 1).

0490 - The new project option will prompt users if he or she would like to import a piece from file or if they would like to run the wizard (Priority 1).

0500 - The Open Project option opens a file browser which only displays Anonymouth project file types (Priority 1).

0510 - Selecting a project from the Open Project file browser will load that project's text, recent suggestions and history (Priority 1).

0520 - If there is project currently open when opening a project or creating a new project, a window will appear asking if the user would like to save their current project (Priority 1).

0521 - After the user has chosen to save or not save their current project, the current project is closed and the new one is opened (Priority 1).

0530 - Clicking the save button will open file browser window if the project has not been previously saved (Priority 1).

0531 - If the project has been previously saved, the save button will save the file (Priority 1).

0540 - The Save As button will open a file browser window and allow the user to save their project (Priority 1).

0541 - When in the Save/Save as file browser, if the user tries to name the project the same name as a project that already exists in that folder, a window will ask the user if they would like to overwrite the old file (Priority 1).

0550 - The Recent Projects list will open a tab when highlighted listing the previous 5 projects saved (Priority 1).

0560 - The Export option will save the text of the project to a supported text document (Priority 1).

0570 - The Close Project option will ask the user if they would like to save their current project(Priority 1).

0571 - After the user decides to save or not save their current project, the current project is cleared from the application window(Priority 1).

0580 - If no project is open, the Save, Save As, Close Project, and Export options will not be selectable (Priority 1).

2.2.4 Edit Menu

The edit menu will provide the user with text editing tools.

0590 - The edit menu will have sub-options: Undo, Redo, Cut, Copy, Paste, and Find (Priority 1).

0600 - The Undo button will revert the last change the user made in the text area(Priority 1).

0601 - The Undo button will be grayed out if the user has not made any changes to the text area(Priority 1).

0602 - The Undo Button will keep a stack of changes and each press of Undo will step back in the changes the user has made (Priority 3).

0610 - The Redo button will undo the last Undo the user made in the text area(Priority 1).

0611 - The Redo button will be grayed out if the user has not just clicked the Undo button(Priority 1).

0612 - The Redo button can be called multiple times if undo has been called multiple times (Priority 3).

0620 - Copy will copy the highlighted text into the clipboard(Priority 1).

0630 - Cut will remove the highlighted text and move it into the clipboard(Priority 1).

0640 - Paste will place the text in the clipboard at the location of the cursor(Priority 1).

0650 - Find will open a window asking for the user to input the text they are searching for(Priority 2).

0651 - If the text they are searching for is found in the document, the text area will scroll down and highlight the first occurrence of the text(Priority 2).

0660 - The Edit menu will be inaccessible until a project exists(Priority 1).

2.2.5 Tools Menu

The tools menu will contain a link to manage the reference materials or the works that the writer is being comparing against.

0670 - The tools menu will contain sub-options: Manage Corpus and Preset Corpus

0680 - The Manage Corpus option will open a new window where the user can add or remove documents from the corpus(Priority 1).

0681 - The Manage Corpus window will list the current documents in the corpus with an 'x' next to them (Priority 1).

0682 - Clicking the 'x' will remove that file from the corpus(Priority 1).

0683 - Clicking the add document button will open a file browser, letting the user select the file/files that they want to add(Priority 1).

0684 - If a user tried to leave this window with no files in the corpus a window will appear saying that they need to have at least one file in the corpus(Priority 1).

0690 - The Preset Corpus menu will contain presets to offer suggestions to emulate other authors such as Shakespeare (Priority 2).

2.2.6 Windows Menu

This menu controls the open and hidden windows and sections of the application.

0700 -The windows menu will contain an option to display or hide the legend toolbox(Priority 1).

0710 - It also contains an option to hide and show the history area(Priority 1).

2.2.7 Help Menu

This menu item will give the user version information as well as provide an overview and help for the overall program.

0720 - The Help Menu will contain sub-options: Tutorial, Contents, and About(Priority 1).

0730 - Clicking the Tutorial will open a window with a step by step guide on how to do the import and analysis functions(Priority 1).

0740 - Selecting contents will take the user to the help document with frequently asked questions(Priority 2).

0750 - Selecting About will display a small splash screen with information about the application (Priority 2).

3. Non-Functional Requirements

3.1 Software Constraints

Python version 2.6 or greater is installed on the system running the software. It should be portable to all popular operating systems:

- Linux
- Mac OS X
- Windows

3.2 Other Constraints

The pickled object from the Stylo tool must be a list of features and it will be running when we pickle that program back our results.

4. Use Cases

4.1 Initialize Writing Sample

The user has two options to perform analysis on their writing sample. Either upload it from an existing file or paste it into the text area of the GUI. Upon completion of one of these options, the analysis process can begin.

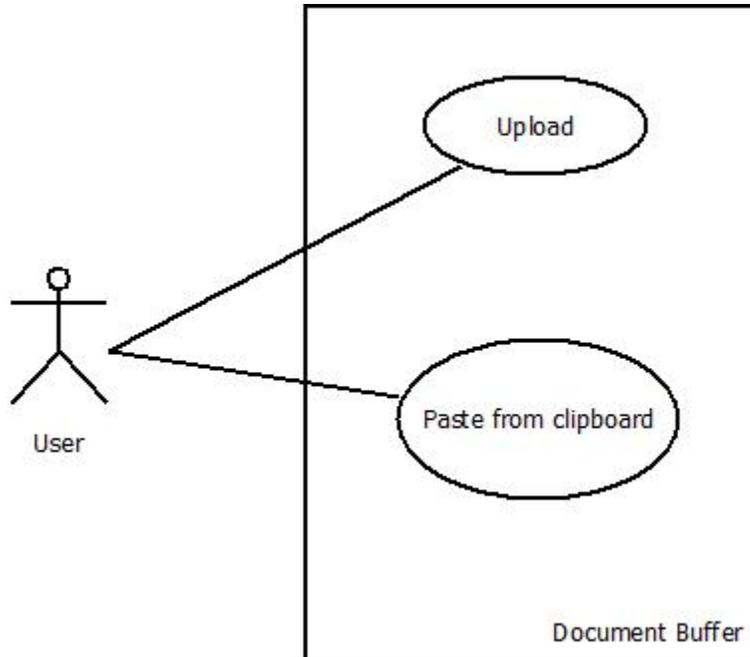


Figure 8 – Use Case

4.2 Analyze Writing Sample

After the user presses the “Submit” button, the writing sample and all corpus documents are submitted to Stylo’s tool via command line as plain text. Then, Stylo uses algorithms to compute values for all the linguistic features of the writing sample. This computation is returned to Anonymouth, again via command line, as a pickled list. This list is used to make suggestions.

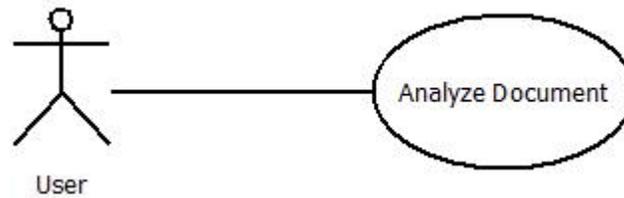


Figure 9 - Use Case

4.3 Make Change for Anonymity

Our tool offers several ways an author can change their writing sample for anonymity.

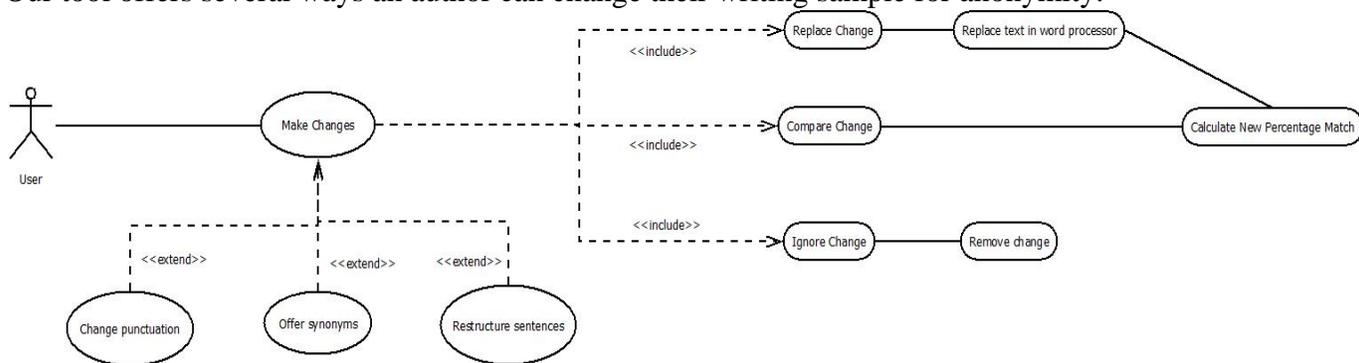


Figure 10 - Use Case

4.3.1 Change Punctuation

If a writing sample uses punctuation frequently or not, our tool will suggest locations to add or remove them.

4.3.2 Offer Synonyms

Synonyms for keywords that are used more than often throughout the sample will be recommended.

4.3.3 Restructure Sentences

Some authors tend to structure their sentences the same in the sentences they write. A good writing sample will have a mixture of sentences structured differently.

5. Future Improvements

One future improvement will be to make our GUI customizable. Users can therefore drag sections of the GUI, such as the Features and Suggestions Area (2.2.1.2), anywhere on the screen.

Another improvement to our existing GUI will allow the user to preview files in a pane before they add them to the text area (2.2.1.1).

6. References

[1] - <https://www.cs.drexel.edu/~jsalvage/Summer2010/CS451/SRSVodka.pdf> . Vodka Requirements Document 2007

[2] - "The Federalist - Contents." *Constitution Society Home Page*. 4 Oct. 2008. Web. 24 Nov. 2010. <<http://www.constitution.org/fed/federa00.htm>>.

[3] - Button, A. Mouse. "What Is Graphical User Interface? - A Word Definition From the Webopedia Computer Dictionary." *Webopedia: Online Computer Dictionary for Computer and Internet Terms and Definitions*. Web. 24 Nov. 2010. <http://www.webopedia.com/TERM/G/Graphical_User_Interface_GUI.html>.

[4] - "About." *Python Programming Language – Official Website*. Web. 24 Nov. 2010. <<http://www.python.org/about/>>.

[5] - "Synonym | Define Synonym at Dictionary.com." *Dictionary.com | Find the Meanings and Definitions of Words at Dictionary.com*. Web. 24 Nov. 2010. <<http://dictionary.reference.com/browse/synonym>>.

7. List of Figures

Figure 1 – Higher level architecture of the architecture between Anonymouth and Stylo.....	5
Figure 2 - Anonymouth upon start-up	6
Figure 3 - Anonymouth with a blank new document	6
Figure 4 – Main Screen after submission	7
Figure 5 – Main Screen after ignoring the “green-colored” suggestion	10

Figure 6 – Flow of actions after right clicking a suggestion in the Features and Suggestions Area (2.2.1.2)	11
Figure 7 - Step 1 of Project Wizard (2.2.2).....	13
Figure 8 – Use Case	19
Figure 9 - Use Case.....	20
Figure 10 - Use Case.....	20

8. Glossary

Back-end: Processes the commands received from the front-end GUI.

Feature:A writing characteristic that can be identified. For example, a common sentence structure or common synonyms.

Front-end: Also known as the GUI.

GUI:Acronym for Graphical User Interface. It is a program interface that takes advantage of the computer’s graphics capabilities to make the program easier to use [3].

Pickled Object: From docs.python.org: “algorithm for serializing and de-serializing a Python object structure. “Pickling” is the process whereby a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream is converted back into an object hierarchy.”

Python: A remarkably powerful dynamic programming language that is used in a wide variety of application domains [4].

Synonym: A word having the same or nearly the same meaning as another in the language [5].

Stylometry:The study of writing style. Used to examine the writing style and identify the author of a piece of writing.