# Table of Contents

# List of Figures

Fig 1.1 Block Diagram .....................................................................................................................8

Fig 2.1 Use Case Diagram ...............................................................................................................9

# Chapter 1

# Introduction

This chapter provides the overview of the project. The first paragraph of every chapter should provide the chapter summary.

## Brief

Style in an document when we say, we aren't speaking of lower-level copyediting concerns such as punctuation or formatting. Rather than we are talking about more significant level concerns like the writer's selection of words and syntactic developments gives a bit of composing its specific 'feel'. Writer work collaboratively, attempting to consolidate their styles can bring about the tedious revision, disappointment, and relational clash, and even then it is not always successful. It is a tough task of making a common single style from numerous styles. Usually to written language, the application for study of linguistic style is Stylometry.

Style change detection is closely related to the fields of stylometry, plagiarism detection, and text segmentation. All of them have in common that they rely on intrinsic stylometric analyses of documents, without referring to external documents. Style Change Detection is to identify whether a document is a single author or multi-author and finding exact positions within a multi-author document at which the author switches. Detecting these positions is a crucial part of processing multi-author documents for purposes of authorship identification.

Multi-authorship analysis provides a report about the author changing position in a collaborating documents. It gives the number of author are there in a document or the position of the authors.

### Relevance to Course Modules

Style change within a document is related to the field of stylometry which is the linguistic study of the writing style. There are many approaches through which change in a writing style can be detected. All of the approaches that are formed is closely to the area of machine learning as the model predict the value whether there is a style change within a document or not. There is use of different classification technique and classifiers to detect the change. This project is also work on machine learning model and classifiers to detect the style change in a document. Dataset is provided to the model to make predictions.

### Project Background

Authorship Attribution is one of the oldest and one of the newest problems in information retrieval. It is the science of construing characteristics of the author from the characteristics of documents written by that author may be a problem with an extended history and a good range of application. The appearance of current statistics made it conceivable to research questions of authorship in a more refined style, and the advancement of present-day computers and huge corpora have made it pragmatic to research these questions algorithmically by means of data retrieval strategies. Authorship attribution (or to utilize another close equivalent term, "stylometry") with the advent of corpus linguistics has become a famous and gainful zone of examination. Most of the researchers assume that people have a characteristic pattern of language, a sort of fingerprint that can be detected in their writings.

Van Halteren has gone so far as to term this a “human stylome,” that a author can be distinguished by properties of their writings, their stylome as it were. There are great hypothetical purposes behind accepting that such a trait may exist. Since each individual needs to learn "language" without help from anyone else, and their experience as language learners contrast, so will the "language" they learn contrast in miniature perspectives. A new advancement in ongoing research has been the development of multivariate strategies tuned to distributional features rather than the

simple presence or absence of features. One might focus on properties like average word length

or vocabulary richness rather than looking at specific word. In contrast to specific words, these properties are consistently accessible.

There are various undertakings identified with authorship attribution, however the majority of the research has been focused on enormous documents. An interesting problem to tackle for smaller texts is the one of style change detection: given a text document, identify whether style change occurs anywhere in it. This formulation usually entails a consistent distribution of text segments from multiple authors.

## Literature Review

There is an exponential growth of digital information produced every day in the form of texts written in natural languages, such as magazines, books, websites, newspapers, reports. Stylometry outgrew prior strategies of examining writings for proof of legitimacy, writer personality, and different inquiries. The advanced act of the order got exposure from the investigation of authorship problems in English Renaissance dramatization. In authorship attribution corpora's study on stylometry is one of the main approaches. Fernanda and Carlos perform a task associated with authorship attribution in which there is a stylometric analysis of stylistic variables in a brief and long text. M. Grabchak Z. Zhang uses an approach based on comparing the entropy of two samples. In which they suggest splitting a large document into equal length of smaller parts and then comparing it. One of all the foremost punctual works on style change detection is performed by Glover and Hirst by analyzing stylometry features to measure the author boundaries. It gives an idea about stylistically inconsistent when two or more authors collaborate on a document by each contributing pieces of text. Meyer Zu Eißen and Stein were the primary to research intrinsic plagiarism detection based on style change detection using word frequency classes. Some of researches propose an unsupervised method to decompose multi-author documents into authorial threads by applying clustering methods on lexical features. Tschuggnall et al. proposed an unsupervised decomposition approach based on grammar tree representations. George A. lozana uses the title an elephant in the room describing multi-authorship and the assessment of individual author. It describes the analysis of single author and multi-author by the mean of h-index. Rexha et al. use stylistic features to predict the number of authors who wrote a text. Gianella employs Bayesian modeling to separates a document by authorship. More methodologies include that of Graham et al. , who utilize neural networks with several stylometric features. Stylometry work include authorship attribution and style change detection is also a part of PAN CLEF every year with a new task. There are some approaches introduce for style change detection. It gives some approaches for single author and multi-author document by deep learning convolutional neural networks and recurrent neural networks operating on features. BERT approach have been proposed for style change detection.

### Analysis from Literature Review (in the context of your project)

Style change detection in a document is text related study as stylometry or text segmentation. Feature extraction is performed to check the sample of a text to see the author's properties. There are many approaches done in the field of stylometry and authorship attribution that help in the study of forensic linguistic studies. Almost much of the research is based on the classification and feature extraction of text to check the change within a document.

### Methodology and Software Lifecycle for This Project

We used an supervised methodology for style change detection, which aims to check whether a document is single author or multi-author and if the document is multi-author where the exact position of an author changes. It combine a TF.IDF representation of the document with features and make predictions via an ensemble of diverse classifiers including SVM, Random Forest, MLP, and LightGBM. Whenever the model detects that style change is present, apply it recursively, looking to find the specific positions of the change in the document.

## Rationale behind Selected Methodology

### Rationale behind Selected Methodology

There are many approaches or methodology are for purpose of authorship attribution and in field of stylometry for style change detection. It can be perform using deep neural networks, Bert and B-maximal Clustering, n-grams and some more approaches but it works on finding whether author change or not at paragraph level and this approaches or methodology not provides the position where the author changes in a document. The selected methodology gives the exact position of change if the authors change within a document.

## Problem Definition

This chapter discusses the precise problem to be solved. It should extend to include the outcome.

### Problem Statement

Today, the use of the work of an author without its authorization, known as textural plagiarism. The field of automatic plagiarism detection raises new questions: how to find if a text has been written by one or more authors? Also, the position where the author changes in a document.

The system predicted that a document is a single author or multi-author. Then the exact position where the style change in a document. The system performs feature extraction and computation on the given text. It split a text into segments and calculates the feature vector of segments and compare each segment and find the position of the author if it is multi-author.

## Deliverables and Development Requirements

Deliverables and development requirements.

### Current System (if applicable to your project)

A brief description of an existing system.

The following figure is a sample figure, Figure 2.1. You are required to follow the same style of numbering and caption for the whole report.

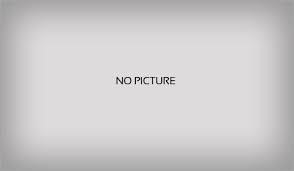


Figure 2.1: Sample picture

The following table (Table 2.1) is sample table; you are required to follow the same style of numbering and caption for the whole report.

**Table 32.1: Sample Table**

|  |  |  |
| --- | --- | --- |
| Header 1 | Header 2 | Header 3 |
| Text | Text | Text |
|  |  |  |

The following list style is the sample to consistently follow in the whole report.

* List items 1
* List items 2

# Chapter 2

# Software Requirement Specification (SRS)

## Requirement Analysis

The required system is to provide the information whether the given document is single author or multi-author. If the documents multi-author than the position where it changes in multi-authorship analysis.

*General Aspects:*

* Dataset of style change detection is from PAN which is based on StackExchange. The dataset is consists of training set 3000 documents and validation and test set of 1500 documents.
* Text file document is provided to the system to find the change in the written document text.
* Input document is examined for multi-authorship analysis by the system.

*Analysis:*

* On the document the feature extraction is performed to highlight the different vector of the text.
* The computed vectors of text is compared with each other to extract the difference between the words.
* This is performed with the supervised learning with classification of the document text.

## Use Cases Diagram(s)

### Detailed Use Case

Diagram

Description automatically generated

## Use case description

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1 |
| **Use Case Name:** | Enter document to check change |
| **Actors:** | Primary Actor : User  Secondary Actor : Developer |
| **Description:** | User enter a document to check whether there is a style change in the provide document. And the position where change occurs. |
| **Trigger:** | A user want to check change in file. |
| **Preconditions:** | PRE 1 : A document must be in language that a the system recognize.  PRE 2 : A document must be a text file. |
| **Postconditions:** | POST 1 : Document show that change occurs in document.  POST 2 : It provide clearly where the style change in a document. |
| **Normal Flow:** | 1. Provide a Document 2. The Document than given to the model for making prediction. 3. After prediction is done by system to check multi-authorship if multi-author or not the given document 4. The predicted document is provided to the user according to requirements. |
| **Alternative Flow:** | None |
| **Exceptions:** | 1. It inform a user when enter wrong format file.   1a. Ask user to enter file with the language that is sported by the system.   1. If user delete the upload document than terminate the use case and restart it. |
| **Assumptions:** | None |

**Table 1 Show the detail of use case**

## Functional Requirements

|  |  |
| --- | --- |
| **Identifier** | RQ -1 |
| **Title** | Style Change |
| **Requirement** | The requirement is that a document is provide by the user to check the if there is a change in the document. The system is all about multi-authorship analysis so this requirement is for style change detection in the document. |
| **Source** | Multi-authorship Analysis |
| **Rationale** | In Multi-authorship analysis to check whether document is single author or multi-author. |
| **Dependencies** | RQ -2 |
| **Priority** | High |

**Functional Requirement 1**

## Functional Requirement 2

|  |  |
| --- | --- |
| **Identifier** | RQ -2 |
| **Title** | Position of Change |
| **Requirement** | The requirement is occurred when there is a change in a document. If there is a style change detection within a document than find the position where that change occurs. If there is no style change than no need to check position. |
| **Source** | Multi-authorship Analysis |
| **Rationale** | In Multi-authorship analysis to check where the style change occurs in the document. |
| **Dependencies** | RQ -1 |
| **Priority** | High |

## Functional Requirement 3

|  |  |
| --- | --- |
| **Identifier** | RQ -3 |
| **Title** | Sentence Modifier |
| **Requirement** | The requirement is occurred when there is a change in a document and we have to show the position where the style change within text of the document. The system ought to provide the highlight of text where the author changes in a file. |
| **Source** | Style Breach Analysis |
| **Rationale** | In Multi-authorship analysis to show an attractive output to the user where the breach occurs. |
| **Dependencies** | RQ -2 |
| **Priority** | Medium |

## Functional Requirement 4

|  |  |
| --- | --- |
| **Identifier** | RQ -4 |
| **Title** | Upload Requirement |
| **Requirement** | The requirement is needed as the upload option to be provided as the system takes the file through an interface. |
| **Source** | Multi-authorship Analysis |
| **Rationale** | In Multi-authorship analysis to upload a document to the system. |
| **Priority** | Medium |

## Non-Functional Requirements

## Usability

USE-1: The system should provide a friendly environment and ease of use to the users.

USE-2: User can get result document with one press button if possible. Because one of the software feature is time saving.

## Performance

PER-1: Calculation time and response time should be as meager as could be expected under the circumstances, since one of the software's highlights are timesaving.

PER-2: The entire pattern of summing up a page ought not to be more than 30 seconds to 3 pages in length record.

## Accuracy

ACC-1: The predicted result must more accurate as possible within document the system should provide high accuracy in term of style change occurs or not.

ACC-2: The positions return by the system where the breach occurs must be accurate.

## Look and Style

LAS-1: The system should give an simple look and style as the user easily understand the where what to do using the system.

## Availability

AVA-1: The system must be available for the corporate Intranet for all time when the site is access by user. It should be available for user whenever the site access.

## Reliability

REL-1: This software will be developed with machine learning, feature engineering. There may be a less chance of site failure.

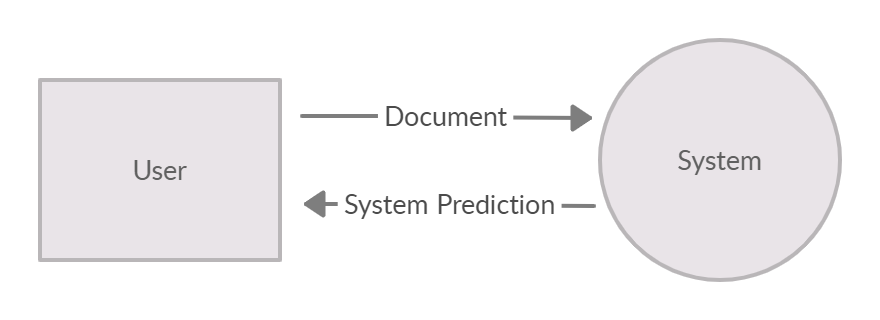
# Chapter 3

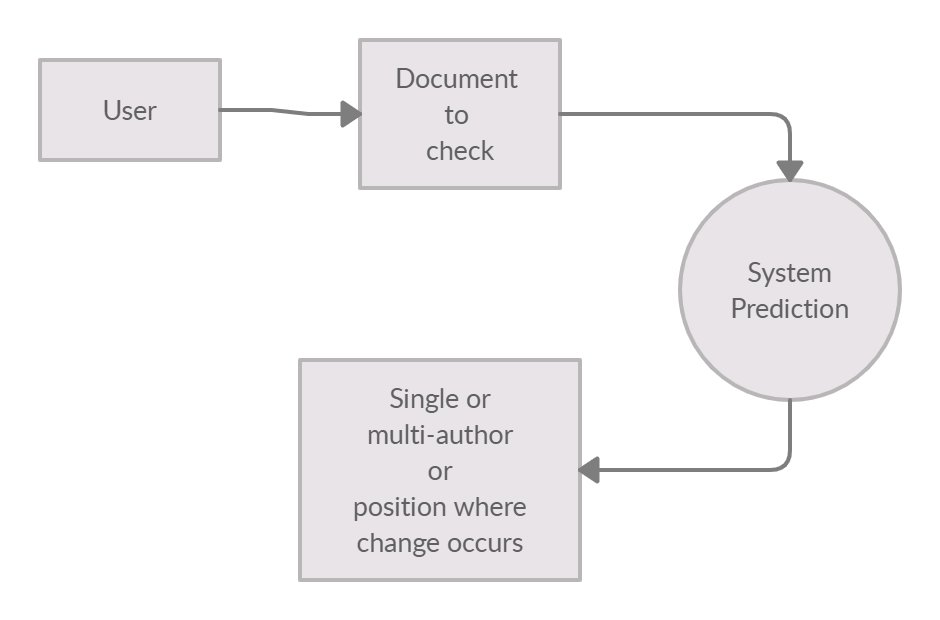
# Design and Architecture

The following parts of Software Design Description (SDD) report should be included in this chapter.

## System Architecture

### Data Representation [Diagram + Description]





### Process Flow/Representation

Diagram

Description automatically generated

First of all, the phase include is feature extraction takes place and replace long numbers with the specific token. In the second phase, there is feature computation as it filters out the long character sequence and long word, streams of words replace file path with the special token. In a text document to find out such difference, we split the document into equal-length segments. By calculating the feature vector for each segment then finding the maximum difference between the values of each feature for any pair of segments. To get more potential information focuses, applied a sliding window across each document with an overlap of one-third of the segment size. For the style changes text representation is used as grammar contraction, tautology, lexical, frequent words, readability and many more. For finding the positions of author changes in a document repeat this procedure of splitting and searching for changes recursively until the length of the text fragment becomes less than 20 sentences and more check for respective fragment

## Design Models [along with descriptions]

# Chapter 4

# Implementation

This chapter will discuss implementation details supported by UML diagrams (if applicable). You will not put your source code here. Any of the following sections may be included based on your project.

## Algorithm

Mention the algorithm(s) used in your project to get the work done with regards to major modules. Provide a pseudocode **OR** a natural language explanation regarding the functioning of main features. Be sure to use the correct syntax and semantics for algorithm representations.

### External APIs

Describe the APIs used in the following table.

Table 8 Details of APIs used in the project

|  |  |  |  |
| --- | --- | --- | --- |
| Name of API | Description of API | Purpose of usage | List down the function/class name in which it is used |
| NLTK | Perform Tasks like classification, stemming, tagging, parsing, semantic reasoning, and tokenization in Python | To reduce the impact of long words split the words and for common words NLTK words corpus is used |  |
| NumPy | Adding support for large, multidimensional  arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays |  |  |
| TensorFlow | For dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications |  |  |
| Pandas | For data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series |  |  |
| Scikit-learn | It features various algorithms like support vector machine, random forests, and k-neighbors. |  |  |
| SciPy | For various commonly used tasks in scientific programming, including linear algebra, integration |  |  |
| Textstat | Python package to calculate statistics from text, which helps to decide readability, complexity and grade level of a particular corpus. |  |  |

## User Interface

Details about user interface with descriptions.

# Chapter 5

# Testing and Evaluation

This chapter may include the following sections. (Students are required to perform the testing both manually and automatedly).

## Manual Testing

### System testing

Once the system has been successfully developed, testing has to be performed to ensure that the system working as intended. This is also to check that the system meets the requirements stated earlier. Besides that, system testing will help in finding the errors that may be hidden from the user. There are few types of testing which includes the unit testing, functional testing and integration testing. The testing must be completed before it is being deploy for user to use.

### Unit Testing

#### Unit **Testing** 1: Login as FYP Committee

#### Testing Objective: To ensure the login form is working correctly

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Test case/Test script | Attribute  value | and | Expected result | Result |
| 1. | Verify user login after click on the „Login‟ button on login form with correct input data | Username: L001 Password:  1234 |  | Successfully log into the main page of the system as FYP Committee member. | Pass |
| 2. |  |  |  |  |  |

#### Unit Testing 2: Edit Profile

### Testing Objective: To ensure the edit profile form is working properly.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Test case/Test script | Attribute  value | and | Expected result | Result |
| 1. |  |  |  |  |  |
| 2. |  |  |  |  |  |

## Functional Testing

The functional testing will take place after the unit testing. In this functional testing, the functionality of each of the module is tested. This is to ensure that the system produced meets the specifications and requirements.

### Functional Testing 1: Login with different roles

#### Objective: To ensure that the correct page with the correct navigation bar is loaded.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Test case/Test script | Attribute and value | Expected result | Result |
| 1. | Login as a „FYP  Committee‟ member. | Username: L001 Password: 1234 | Main page for the FYP Committee member is loaded with the FYP  Committee  navigation bar | Pass |
| 2. |  |  |  |  |

## Integration Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Test case/Test script | Attribute and  value | Expected result | Result |
| 1. | Login as “FYP  Committee” member | Username: L001 Password: 1234 | Login successful and the FYP  Committee page with its navigation bar is loaded and in the view profile page | Pass |
| 2. | Upload student record for Project 1 | - | File successfully uploaded and return to the upload page. Student records are updated. | Pass |
| 3. | View supervising  student | - | The list of supervisees shown on the screen. | Pass |

## Automated Testing

Tools used:

|  |  |  |  |
| --- | --- | --- | --- |
| Tool Name | Tool Description | Applied on [list of related test cases / FR / NFR] | Results |
|  |  |  |  |
|  |  |  |  |

# Chapter 6 Conclusion and Future Work

This chapter concludes the project and highlights future work.

## Conclusion

## Future Work

## References

References to any book, journal paper or website should properly be acknowledged. Please consistently follow the style. The following are few examples of different resources i.e. journal article, book, and website.

1. Lyda M.S. Lau, Jayne Curson, Richard Drew, Peter Dew and Christine Leigh, (1999), Use Of VSP Resource Rooms to Support Group Work in a Learning Environment, ACM 99, pp-2. **(Journal paper example)**
2. Hideyuki Nakanishi, Chikara Yoshida, Toshikazu Nishmora and TuruIshada, (1996), FreeWalk: Supporting Casual Meetings in a Network, pp 308-314 **(paper on web)** http://www.acm.org/pubs/articles/proceedings/cscw/240080/p308-nakanishi.pdf
3. Ali Behforooz& Frederick J.Hudson, (1996), Software Engineering Fundamentals, Oxford University Press. Chapter 8, pp255-235. **(book reference example)**
4. Page Author, Page Title, http://www.bt.com/bttj/archive.htm, Last date accessed**. (web**

site)

## APPENDIX A

(if any name it)