VIVEKANAND EDUCATION SOCIETY’S
INSTITUTE OF TECHNOLOGY

Department of Computer Engineering

Project Report on

STYLOMETRY BASED AUTHORSHIP IDENTIFICATION

In partial fulfilment of the Fourth Year, Bachelor of Engineering

(B.E.) Degree in Computer Engineering at the University of Mumbai

Academic Year 2017-2018

Project Mentor

Mrs. Sujata Khedkar

Submitted by

Shashank Agnihotri, D17C/02
Anshul Agarwal, D17B/02
Mahak Pancholi, D17B/51
Pooja Hande, D17C/23

(2017-18)

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

Department of Computer Engineering

Certificate

This is to certify that ***Shashank Agnihotri, Anshul Agarwal, Mahak Pancholi, Pooja Hande*** of Fourth Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the project on "***Stylometry Based Authorship Identification***" as a part of their coursework of PROJECT-II for Semester-VIII under the guidance of their mentor ***Mrs. Sujata Khedkar*** in the year 2017-2018.

Programme Outcomes	Grade
PO1,PO2,PO3,PO4,PO5,PO6,PO7, PO8, PO9, PO10, PO11, PO12 PSO1, PSO2	

Date:

Project Guide:

Mrs. Sujata Khedkar

Project Report Approval
For
B. E (Computer Engineering)

This thesis/dissertation/project report entitled ***Stylometry Based Authorship Identification*** by ***Shashank Agnihotri, Anshul Agarwal, Mahak Pancholi, Pooja Hande*** is approved for the degree of ***B.E Computer Engineering.***

Internal Examiner

____

External Examiner

____

Head of the Department

____

Principal

____

Date:
Place: Mumbai , Chembur.

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Shashank Agnihotri, D17C/02

Anshul Agarwal, D17B/02

Mahak Pancholi, D17B/51

Pooja Hande, D17C/23

Date:

ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to Associate Professor **Mrs. Sujata Khedkar** (Project Guide) for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Computer Department **Dr.(Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J.M. Nair ,** for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement at several times.

Computer Engineering Department

COURSE OUTCOMES FOR B.E PROJECT

Learners will be to:-

Course Outcome	Description of the Course Outcome
CO 1	Able to apply the relevant engineering concepts, knowledge and skills towards the project.
CO2	Able to identify, formulate and interpret the various relevant research papers and to determine the problem.
CO 3	Able to apply the engineering concepts towards designing solution for the problem.
CO 4	Able to interpret the data and datasets to be utilized.
CO 5	Able to create, select and apply appropriate technologies, techniques, resources and tools for the project.
CO 6	Able to apply ethical, professional policies and principles towards societal, environmental, safety and cultural benefit.
CO 7	Able to function effectively as an individual, and as a member of a team, allocating roles with clear lines of responsibility and accountability.
CO 8	Able to write effective reports, design documents and make effective presentations.
CO 9	Able to apply engineering and management principles to the project as a team member.
CO 10	Able to apply the project domain knowledge to sharpen one's competency.
CO 11	Able to develop professional, presentational, balanced and structured approach towards project development.
CO 12	Able to adopt skills, languages, environment and platforms for creating innovative solutions for the project.

ABSTRACT of the project

“Every person is unique”, we have been hearing this since ages. Every person has a unique identity, a unique fingerprint, a unique retina and a lot more. These features play a vital role in identification of individuals for security purposes. Unfortunately, when it comes to security of written pieces or words from an individual, these primary unique identities are futile. One cannot identify a writer from a written piece of text on the basis of retina or fingerprint scans, sometimes even the signature can be forged, in such situations for security purposes and intellectual property rights it becomes very important to identify the true author. Stylometry plays an important role in this.

Every author has a unique style of writing, measure of this style of writing is called Stylometry. This paper proposes to identify authors from text based on their style of writing. First a data set consisting of novels will be used to train the system for multiple authors, then a random text would be given to the system to identify the author correctly, if the author predicted by the system is similar to the author claimed then the information is authentic otherwise the author claiming to be the writer is a fraud. For stylometry, over the ages, many features have been focused on, but this paper proposes new features to be used for this purpose. While writing, there are many unconscious styles that are incorporated by the author, these features have been unnoticed till date, but can play a vital role in accurate and fast identification of authors. These features include: intellectual property right’, ‘chapter length’ and frequency of particular words per thousand words.The algorithms that can be used to train the system are Decision tree, SVM or Multilayer Perceptron.

INDEX

Chapter 1 : Introduction

In this chapter, we give a brief introduction to the project. The motivation behind choosing to work on this project is also given. We mention the shortcomings of already existing systems and define the problem, while recapitulating the relevance of our system. Here, the methodology that we will be using is highlighted.

1.1 Introduction

Stylometry is a behavioral feature that a person exhibits during writing and can be extracted and used potentially to check the identity of the author of documents. Stylometry or style of writing can be a powerful feature providing proof for right ownership of a document.

Stylometry based techniques are usually very efficient in capturing the writing style of an author. With the help of stylometry myriad of problems can be resolved. In this project we will use stylometric features to correctly identify the author of a given document.

1.2 Motivation

Plagiarism is stealing, appropriation of another person's thoughts, ideas, or linguistic expression, and the representation of them as one's original work. Plagiarism is considered academic dishonesty and a breach of journalistic ethics. It is subject to sanctions like penalties, suspension, and even expulsion. Plagiarism in itself is not a crime: it is not defined or punished by law, but rather by institutions. With author attribution techniques we can identify the most appropriate author from group of potential suspects of a piece of work and find evidences to support the same. Plagiarism of linguistic style or expression can be picked up, which is not possible with current systems in place. Stylometry based author attribution can also prove to be a strong tool in fraud detection, forgery, email classification et cetera. One of the interesting applications of stylogenetics of a document is author profiling, in which one can derive gender, nationality, age group, mental health of the author from the given document.

Identifying the style in which a document is written can be of great consequence. It can be used to verify author of a legal document (verify the author of last will), users with multiple account on same online forum, review spamming, public security (find authors of anonymous illegal documents and threats), School essays authorship verification (co-authorship) and much more.

In this project we have chosen to address author attribution problem, as a viable and scalable solution for multi-author dataset is yet to be formulated. Existing systems do not take into consideration the linguistic style of the author. Thus, these systems can be fooled by cleverly rearranged or rephrased sentences. Our proposed system will be capable of detecting aforementioned cases of plagiarism, which alludes current systems.

1.3 Drawback of Existing Systems

Text-alignment is currently the preferred technique for estimating the degree of similarity with existing written works. A plagiarism detection service looks for matching strings of words between the document it's looking at and the ones it has in its index. This is true for a local plagiarism checker, such as WCopyFind, search engine-based systems such as Copyscape and Plagium and high-end system such as Turnitin. Due to its dependency on other documents it becomes increasingly tedious and time-consuming to scale up to the growing number of online and offline documents.

Since plagiarism detection tools can only detect copying, or more specifically similar phrases, there are two areas where they are particularly weak.

1. **Non-Verbatim Plagiarism:** Plagiarism that involves the rewriting, translating or otherwise redrafting the text can't be detected. This can be difficult to get away with as most plagiarism detectors are extremely sensitive, but since plagiarism detectors don't analyze the content of the work, just the words, it can't see if you lifted the idea or information if you didn't also lift the words. This is a common problem in academia, which treats this kind of plagiarism equally as seriously as verbatim plagiarism.
2. **Common Phrasing/Attributed Use:** Second, though many plagiarism checkers will make an attempt to separate out attributes use, given the variety of attribution styles it isn't always possible. Also, given how common some phrases are in the English language, many plagiarism checkers will report matches that are actually just coincidence.

1.4 Problem Definition

In this project, we attempt to solve author attribution problem using stylometric features. A viable solution is yet to be implemented for large scale author identification, and therefore by extension, for plagiarism detection. Current systems use text-alignment algorithms which are unable to differentiate and capture linguistic style and context of the document. Due to this, these systems can be easily fooled by a cleverly arranged and rephrased documents.

We attempt to incorporate and distinguish between styles large number of authors. A multi-author corpus of documents is used as dataset. Most of previous researches on authorship attribution were completely or partially based on frequency analysis of most frequent words, especially those also called function words. Function words were found to be among the best features for authorship attribution. In this project we will use a mix of frequency and context words.

1.5 Relevance of the Project

Plagiarism is a widespread problem around the world. It can take various forms : copying and pasting text without acknowledging its source, recycling, self plagiarism, purchasing papers from an agency or a ghostwriter and submitting them as one's own.

In the United Kingdom, for example, almost 50,000 university students were caught cheating from 2012 to 2015. This is only the reported cases , many more cases remain undetected. Some famous politicians have been implicated in plagiarism scandals. Following the public scandal revolving around plagiarism identified in their dissertations, German Defense Minister Karl-Theodor zu Guttenberg resigned in 2011 and German Education Minister Annette Schavan in 2013.

With such increasing popularity and availability of digital text data, authorships of digital texts can not be taken for granted due to the ease of copying and parsing.

1.6 Methodology used

Dirty data often gives sparse feature set. It is not fit for training using machine learning algorithms such as MLP, SVM etcetera. Data is cleaned by first tokenizing and removing all unwanted symbols and numbers. Then the document is stemmed using porter's algorithm. After data cleaning document is processed to generate word embedding.

Instead of relying on pre-computed co-occurrence counts, Word2Vec takes raw text as input and learns a word by predicting its surrounding context given its surrounding context using gradient descent with randomly initialized vectors. The algorithm reads each word in the corpus and generates a vector by predicting its surrounding context. This preserves the semantic context in the vectors.

The feature vectors generated by word2vec model of gensim is trained using multi-layer perceptron.

Chapter 2 : Literature Survey

Fraud, cheating, and, plagiarism detection are major challenges for detection at large scale. Verifying that a student solved an assignment alone is extremely hard to verify in an online setting. Much research has been conducted on author attribution. For our research on the topic, papers on popular journals such as IEEE, IJCSCN, IJCA, springer et cetera,were referred.

---

1. Krause, Markus, "Stylometry-based Fraud and Plagiarism Detection for Learning at Scale", 2015 5th KSS Workshop, Karlsruhe, Germany [1]

Abstract

Fraud detection in free and natural text submissions is a major challenge for educators in general. It is even more challenging to detect plagiarism at scale and in online class such as Massive Open Online Courses. In this paper, we introduce a novel method then analyses the writing style of an author (stylometry) to identify plagiarism. We will show that our system scales to thousands of submissions and students. For a given test set of ~4000 users our algorithm shows F-scores of over 90%.

Inference

(Markus Krause et al. ,2015) [1] extracted a set of features from each blog the corpus. A total of 164 individual features were used(character frequency, word length frequency, sentence length frequency, part of speech tag frequency, word specificity frequency). To generate **POS Penn-Treebank tagger from the NLTK library** was used. From these annotated sentences, feature vector for each sentence was calculated.To train the support vector machine for an author b **bootstrap samples of an author as positive examples**, were taken. Proposed method was effective in detecting the authorship with a **mean F-score of 0.91.**

2. Ángel Hernández-Castañeda, Hiram Calvo , "Author Verification Using a Semantic Space Model.", Computación y Sistemas, Vol. 21, No. 2, 2017, pp. 167–179 [2]

Abstract

In this work we propose to solve the author verification problem using a semantic space model through **Latent Dirichlet Allocation (LDA).** We experiment with the corpus used in the author identification tasks at PAN 2014 and PAN 2015. These datasets consist of subsets in the following languages: English, Spanish, Dutch and Greek. Each problem contained in these corpora is formed by one to five known documents which were written by one author and one unknown document. The task is to predict whether the unknown document was written by the author who wrote the known documents. They processed the documents in the dataset and captured the fingerprint of authors by generating a **probabilistic distribution of words in the documents**. In PAN 2015 classification, we achieved 81.6%, 75.4%, 74.1%, 67.1% accuracy for each English, Spanish, Dutch and Greek subset respectively. In particular for the English subset, we outreached the best result reported in both competitions.

Inference

Ángel Hernández-Castañeda1 et al [2] used semantic space model for style determination. In this work,semantic information to find features that help to discriminate texts was obtained through Latent Dirichlet Allocation. Latent Dirichlet Allocation is a probabilistic generative model for discrete data collections such as a collection of texts; it represents documents as a mix of different topics. Each topic consists of a set of words that keep some link between them. Due to the fact that LDA is a stochastic method, the obtained result for each experiment can be different. To represent each problem, all documents in the dataset were processed with LDA. Then, we obtain vectors which represent known and unknown documents. Based on a specific problem, each known-document's vector and the unknown-document's vector were subtracted.

3. J. Hurtado, N. Taweewitchakreeya and X. Zhu, "Who wrote this paper? Learning for authorship de-identification using stylometric featuress," Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014), Redwood City, CA, 2014, pp. 859-862.[3]

Abstract

In this paper, (authors) we propose to combine stylometric features and neural networks for authorship de-identification. (their)Our research mainly focuses on scientific publications, because scholarly journals are publicly available with plenty of labeled data to learn an author's style or traits. The main challenge of authorship de-identification is to identify features which can properly capture an author's writing style. In

the proposed design, we choose a combination of stylometric features, including lexical, syntactic, structural and content-specific features, to represent each author's style and use them to build classification models.

We manually collect publications from computer science and biomedicine domains and validate our designs by using a number of classification methods. Our experiments show that among four well-known classifiers, Multilayer Perceptron (MLP) classifiers achieve the best performance for authorship de-identification.

Inference

Jose Hurtado, Napat Taweewitchakreeya and Xingquan Zhu [3] evaluated a framework using **four different types of classifiers, including Multilayer Perceptron (MLP), Random forest (RF), Support vector machines (SVM), and k-nearest neighbors (k-NN)**. In addition to that they also carried out experiments on two domain specific fields, including computer science and biomedical fields, in order to validate that their method is robust for data from different domains. Their experiments show that combining all identified features and MLP learner produces best classification results for authorship de-identification, and their method is also effective for data from different domains.

In order to learn the writing style of each author, they employed a supervised learning framework. Given a set of abstracts collected from a number of authors, they processed these abstracts to extract a set of stylometric features. After feature extraction, they used those features along with each author's name to generate the training data and learn classification models from the training data.

4. P. Das, R. Tasmim and S. Ismail, "An experimental study of stylometry in Bangla literature," 2015 2nd International Conference on Electrical Information and Communication Technologies (EICT), Khulna, 2015, pp. 575-580.[4]

Abstract

Every writer has a different style of writing of their own. By analyzing various kinds of features we can identify and specify some characteristics in a writer's writing which is known as stylogenetics. In this paper we gathered Bangla blogs written by four different Bangladeshi writers. Using machine learning methods we tried to identify special Stylometry features in their writing style. We analyzed various features in their writings, for example, percentage of unique words, word length, sentence length, and frequency of some parts of speech, number of suffix, frequency of first word, second word, second last word and last word of a sentence, counting average number of question marks per document, frequency of word by its position in a sentence etc. We gathered statistical data from analyzing those features and tried to find the variance among these writers using the statistical data.

Inference

Prapti Das, Rishmita Tasmim, Sabir Ismail [4] used 50 blogs each written by four different Bangladeshi writers were used for this study. **Stylogenetics** is clustering-based stylistic analysis of literary corpora. It is a way of analyzing written texts to learn about the writer. Analysis of twelve features according to the style of Bangla literature was possible. **Standard deviation and Jaccard Similarity** are comparatively easy to calculate. It was possible to extend the texts written in languages other than English. **But these two methods do not identify all the authors from the data set. The system can be trained only for a particular language and doesn't support different language data set.** Machine learning methods like Cosine similarity or Clustering can also be used to find some Stylometric features of a writer.

5. He, Congzhou & Rasheed, Khaled. (2004). "Using Machine Learning Techniques for Stylometry", Proceedings of the International Conference on Artificial Intelligence, IC-AI '04, Volume 2 & Proceedings of the International Conference on Machine Learning; Models, Technologies & Applications,2004[5]

Abstract

In this paper we describe our work which attempts to recognize different authors based on their style of writing (without help from genre or period). Fraud detection, email classification, deciding the authorship of famous documents like the Federalist Papers 1 , attributing authors to pieces of texts in collaborative writing, and software forensics are some of the many uses of author attribution. In this project, we train decision trees and neural networks to "learn" the writing style of five Victorian authors and distinguish between them based on certain features of their writing which define their style. The texts chosen were of the same genre and of the same period to ensure that the success of the learners would entail that texts can be classified on the style or the "textual fingerprint" of authors alone. We achieved 82.4% accuracy on the test set using decision trees and 88.2% accuracy on the test set using neural networks.

Inference

Ramyaa, Congzhou He, Khaled Rasheed [5] proposed a base paper in which they were able to capture an intangible concept like style. **Both decision trees and Artificial Neural Networks** yield a significantly higher accuracy rate than random guess. This brings us to the conclusion which shows that the assumption is well justified that there is a quantifiable unconscious aspect in an author's style. The neural networks yield a better performance as compared to decision tree, capturing the intangible concept like style. But decision trees are human readable which makes the possibility of defining style. The methods used required fewer input variables than traditional statistics.

6. Marcelo Luiz Brocardo, Issa Traore, Sherif Saad, Isaac Woungang , "Authorship Verification for Short Messages using Stylometry", 2013 International Conference on Computer, Information and Telecommunication Systems, pp. 1-6 [6]

Abstract

Authorship verification can be checked using stylometric techniques through the analysis of linguistic styles and writing characteristics of the authors. Stylometry is a behavioral feature that a person exhibits during writing and can be extracted and used potentially to check the identity of the author of online documents. Although stylometric techniques can achieve high accuracy rates for long documents, it is still challenging to identify an author for short documents, in particular when dealing with large author's populations. These hurdles must be addressed for stylometry to be usable in checking authorship of online messages such as emails, text messages, or twitter feeds. In this paper, we pose some steps toward achieving that goal by proposing a supervised learning technique combined with n-gram analysis for authorship verification in short texts. Experimental evaluation based on the Enron email dataset involving 87 authors yields very promising results consisting of an Equal Error Rate (EER) of 14.35% for message blocks of 500 characters.

Inference

This paper used n-gram algorithm for classification of data-set. Marcelo Luiz Brocardo, Issa Traore, Sherif Saad and Isaac Woungang [6] stated that the best configuration was achieved with block size of 500 characters, achieving EER below 15%. Investigation in block sizes of 250 and 500 characters represent significantly shorter messages compared to the messages used so far in the literature for identity verification. A long text is splitted in chunks of 500 characters. Investigation on even shorter messages (e.g. 10 to 50 characters) is still to be done to be able to cover a broader range of online messages such as twitter feeds and text messages. **The methodologies included calculating EER,FER and TER.** A smaller verification block lead to increased verification error rates and vice-versa. It is less suitable for larger written works. It's accuracy drops significantly when we use a larger data-set. **The con for this system is that this system can work optimally only for 3-gram or 4-gram algorithms and when the entities of the data-set are extremely small such as short emails, personal messages etc. Moreover the limit to the number of the authors that can be identified accurately is very small.**

7. Ramnial H., Panchoo S., Pudaruth S, "Authorship Attribution Using Stylometry and Machine Learning Techniques", 2016 Intelligent Systems Technologies and Applications. Advances in Intelligent Systems and Computing, vol 384. Springer, Cham[7]

Abstract

Plagiarism is considered to be a highly unethical activity in the academic world. Text-alignment is currently the preferred technique for estimating the degree of similarity with existing written works. Due to its dependency on other documents it becomes increasingly tedious and time-consuming to scale up to the growing number of online and offline documents. Thus, this paper aims at studying the use of stylometric features present in a document in order to verify its authorship. Two machine learning algorithms, namely k-NN and SMO, were used to predict the authenticity of the writings. A computer program consisting of 446 features was implemented. Ten PhD theses, split into different segments of 1000, 5000 and 10000 words, were used, totaling 520 documents as our corpus. Our results show that authorship attribution using stylometry method has generated an accuracy of above 90%, except for 7-NN with 1000 words. We also showed how authorship attribution can be used to identify potential cases of plagiarism in formal writings.

Inference

The detailed prediction results for author attribution for the segments of different sizes and different number of features were tested with **the nearest neighbour (kNN) and support vector machines (SMO) algorithm** from Weka. A cross-validation technique with 10 folds was used for all the thirty experiments. SMO outperformed the kNN algorithm. kNN using 446 features produced higher classification accuracies than using 153 features. The number of features used did not have any significant effect on SMO as the performance measures are almost the same in both cases. For smaller segments of 1,000 words, 7-NN performed with an accuracy value of 86%. The accuracy continued to increase until 13-NN, after which it started to decrease. SMO with 446 features performed slightly better than SMO with 153 features.**Nearest neighbour(NN),Support Vector Machine(SMO)together can be used for authorship attribution with high accuracy.**

8. Maciej Eder, Jan Rybicki and Mike Kestemont , "Stylometry with R: A Package for Computational Text Analysis",The R Journal Vol. 8/1, Aug. 2016 [8]

Abstract

This software paper describes ‘Stylometry with R’ (stylo), a flexible R package for the high- level analysis of writing style in stylometry. Stylometry (computational stylistics) is concerned with the quantitative study of writing style, e.g. authorship verification, an application which has considerable potential in forensic contexts, as well as historical research. This paper introduces the possibilities of stylo for computational text analysis, via a number of dummy case studies from English and French literature. It demonstrate how the package is particularly useful in the exploratory statistical analysis of texts, e.g. with respect to authorial writing style. Because stylo provides an attractive graphical user interface for high-level exploratory analyses, it is especially suited for an audience of novices, without programming skills (e.g. from the Digital Humanities). More experienced users can benefit from such implementation of a series of standard pipelines for text processing, as well as a number of similarity metrics.

Inference

Stylo is a package available at CRAN and at GitHub repository used by the Maciej Eder, Jan Rybicki and Mike Kestemont[8] that allows loading textual data either from **R objects,** or directly from corpus files stored in a dedicated folder. Metadata of the input texts are expected to be included in the file names. The file name convention assumes that any string of characters followed by an underscore becomes a class identifier (case sensitive). In final scatter plots and dendrograms, colors of the samples are assigned according to this convention; common file extensions are dropped and stylo offers a rich set of options to load texts in various formats from a file system (preferably encoded in UTF-8 Unicode, but it also supports other encodings, e.g. under Windows). Apart from raw text, stylo allows to load texts encoded according to the guidelines of the Text Encoding Initiative, which is relatively prominent in the community of text analysis researchers.To preprocess the data, stylo offers a number of tokenizers. Tokenization refers to the process of dividing a string of input texts into countable units, such as word tokens. **It has graphics and charting capabilities. Downloading, installing and loading stylo is straightforward.**

9. Lakshmi, Pushpendra Kumar Pateriya, "A Study on Author Identification through Stylometry ",Lakshmi et al , International Journal of Computer Science & Communication Networks,Vol 2(6), pp. 653-657 [9]

Abstract

Author identification is a critical point to be ensured, because many people are used to copy the content of others. Stylometry can be used for the author identification for text documents. As the non–repudiation and integrity of the message are the major concerns. Stylometry is not only identifying a writing pattern but we can also identify the gender of the human. So this document discussed about identification of author, authentication through stylometry technique. In this paper different stylometric techniques are discussed.

Inference

In this paper, the authors have discussed author profiling from the style of the paper. They argue that, the many details about the author, such as her nationality, gender, age group et cetera can be derived from a given document. They have used various techniques in pattern classification, such as **Bayesian Theory, Decision Trees, Neural Networks or k nearest neighbor (KNN)**. Their program uses the KNN algorithm which used to classify objects based on the basis of their similarities or distance metric. Each subset was run against the other yielding 76.72% and 66.72% accuracy. The author faces some difficulties and their future work is to extend the authentication task to identify patterns in frequently used misspelled and misused words.

10. Ganapathi N V Raju, Ch. Sadhvi, P Tejaswini and Y Mounica, "Style based Authorship Attribution on English Editorial Documents", International Journal of Computer Applications 159(4), pp. 5-8, February 2017. [10]

Abstract

The aim of the authorship attribution is identification of the author/s of unknown document(s). Every author has a unique style of writing pattern. The present paper identifies the unique style of an author(s) using lexical stylometric features. The lexical feature vectors of various authors are used in the supervised machine learning algorithms for predicting the unknown document. The highest average accuracy achieved is 97.22 using SVM algorithm.

Inference

Three kinds of style based text features were considered for the experiment. They are character-based, word-based and function words. I all 150 features were selected for identifying the author's task. The research was conducted on English editorial documents with style based character, word, function word based features and feature value extraction was implemented in our Java program. The style based features are implemented on a collection of 250 editorial documents from the seven leading columnists of India i.e...(1) M.J.Akbar, (2) Chetan Bhagat, (3) A.S.Panneerselvan, (4) C.Raja Mohan and (5) Tavleen Singh. 50 documents of each author has been considered for both training and testing purpose. **The highest average accuracy achieved is 97.22 using SVM algorithm.**

Chapter 3 : Requirements

In this chapter, we enlist the functional and nonfunctional requirements of our project. The constraints of the system are also listed. Hardware, software requirements as well as techniques and tools we might need for implementing the project, are mentioned. The algorithm which is used in the existing system is mentioned along with our project proposal as shown below in this chapter.

3.1 Functional Requirements

1.Data Collection Module: For obtaining the documents, corpus to perform the analysis.

2.Data processing Module: Pre-processing activities, feature extraction and selections.

3.Machine Learning Module: for building a predictive model on training set.

4.Visualisation: User interface for displaying output statistics of feature based sentiment analysis.

3.2 Non-Functional Requirements

1.Reliability: Reliability is an attribute of any computer-related component that consistently performs according to its specifications.

2.Portability: Portability is a characteristic attributed to a computer program if it can be used in operating systems other than the one in which it was created without requiring major rework.

3.Ease to use: The user interface must be easy to use, so that users can upload documents and check identify correct author.

3.3 Constraints

1. Requirement of known documents of undisputed ownership, for the system to analyse and understand the style of the author.
2. Requirement of relatively large corpus, to achieve high efficacy.
3. Classifiers built may not achieve 100% accuracy.
4. Prediction accuracy depends on the data set and the ML algorithm used for training. The data set should be adequate enough.

3.4 Hardware, Software, Techniques and Tools - Requirements

3.4.1 Hardware Requirements

1. 8 GB RAM or more
2. 4GB Graphic card

3. 5th generation intel i7 processor or above

3.4.2 Software Requirements

1. Data Processing: Scikit Sklearn Python, Java.
2. User Interface: Xampp, HTML, CSS, JavaScript, Bootstrap, PHP, Ajax, JQuery.
3. Data Visualization: Sklearn libraries.

3.4.3 Techniques

1. Multilayer Perceptron(Neural Network)
2. SVM
3. Decision Tree
4. Tf-idf
5. Doc2Vec
6. Word2Vec

3.4.4 Tools

1. Visual Studio Code
2. Jetbrains Pycharm
3. Sklearn Predictor
4. Xampp Server

3.5 Selection of Hardware, Software, Technology and Tools

3.5.1 Hardware

High requirements of RAM and Graphic card for processing large amount of data quickly so that faster results are obtained for the best user experience.

3.5.2 Software

Java is used for Pre-Processing and Scikit Sklearn is used to implement the algorithms. The UI is implemented with the help of basic web technologies for better visualization.

3.5.3 Techniques

The algorithms such as DT, MLP and SVM are used for training the model to obtain the best possible results for our dataset. These can be easily implemented using the Sklearn Predictor.

3.5.4 Tools

Various softwares and IDEs are used to develop the application with ease. Sklearn Predictor is used to implement the ML algorithms. Xampp Server is required for supporting the user interface.

Chapter 4 : Proposed Design

In this chapter, each module of the system is defined at length, discussing how the data will flow in the system. Each module involved in the system is described in detail. The algorithms used and the scheduling diagram is also given. Also the gantt charts represent the timeline of the work done.

4.1 Block Diagram

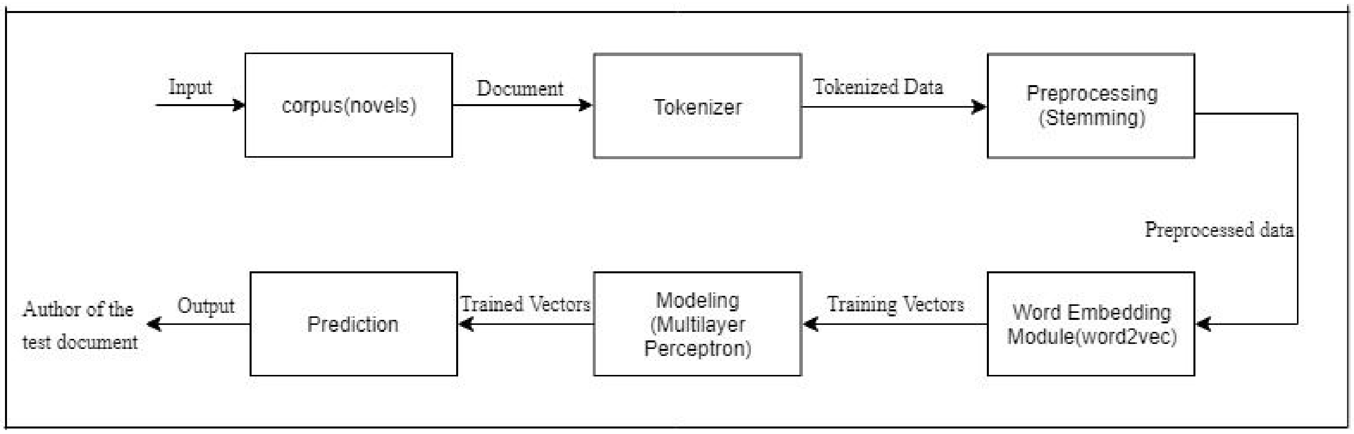


Figure 1: Block Diagram

In above block diagram, there are following modules:

1. **Input:** Input will be documents, essays or novels of each author, with the author being undisputed owner of that document.
2. **Generation of vectors:** Word embedding is used to obtain vectors for each unique word in the document. Word embedding is a feature learning techniques in natural language processing where words or phrases from the vocabulary are mapped to vectors of real numbers. Gensim library was used for this purpose.
3. **Data Training:** The algorithm moves over each word in the corpus and repeats the training step in an online fashion. The interesting property that word vectors obtained this way exhibit is that they encode not only syntactic but also semantic relationships between words. Word2vec embeddings can be obtained using skip gram model or CBOW model. We have used CBOW model for our project. Once the vectors are generated for a document in the corpus, they are saved in a gensim model file using pickle. For testing, this model is loaded in the memory and the vectors for test file are compared with this model.

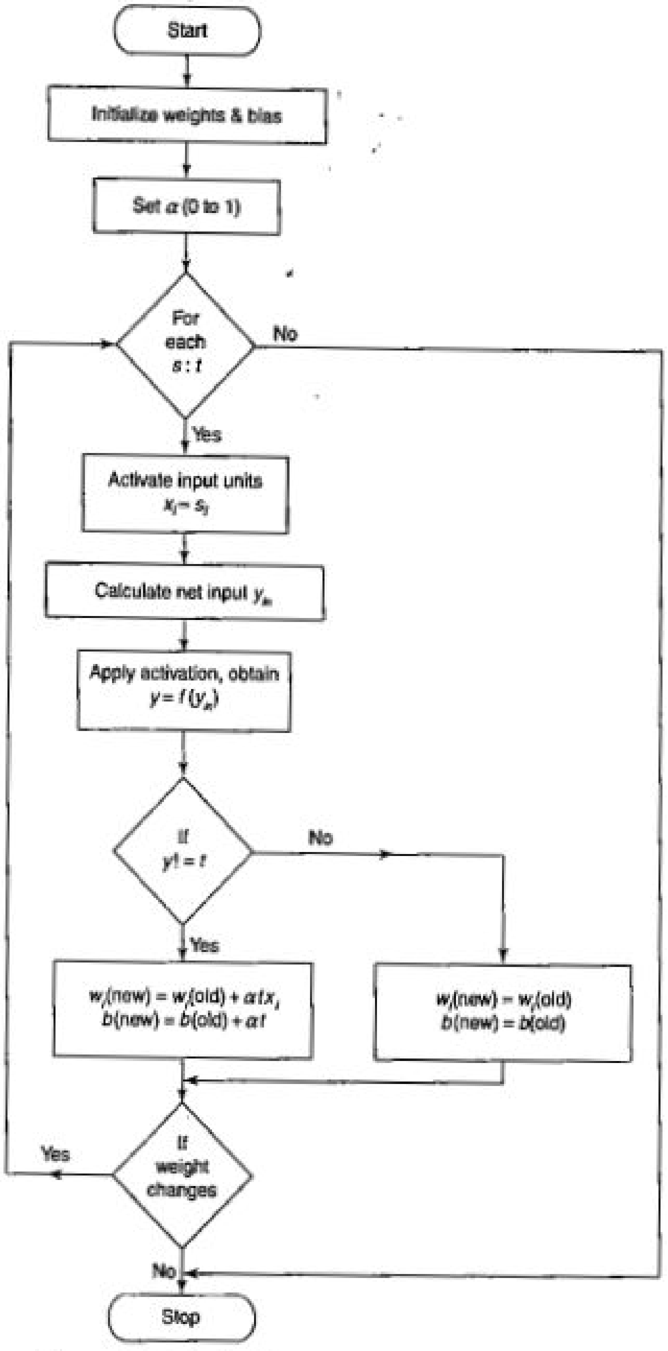

Start
Initialize weights & bias
Set α (0 to 1)
For each $s : t$
No
Yes
Activate input units $x_i = s_i$
Calculate net input y_{in}
Apply activation, obtain $y = f(y_{in})$
If $y != t$
No
Yes
$w_i(\text{new}) = w_i(\text{old}) + \alpha t x_i$
$b(\text{new}) = b(\text{old}) + \alpha t$
$w_i(\text{new}) = w_i(\text{old})$
$b(\text{new}) = b(\text{old})$
Yes
If weight changes
No
Stop

Figure 2: Flowchart for MLP

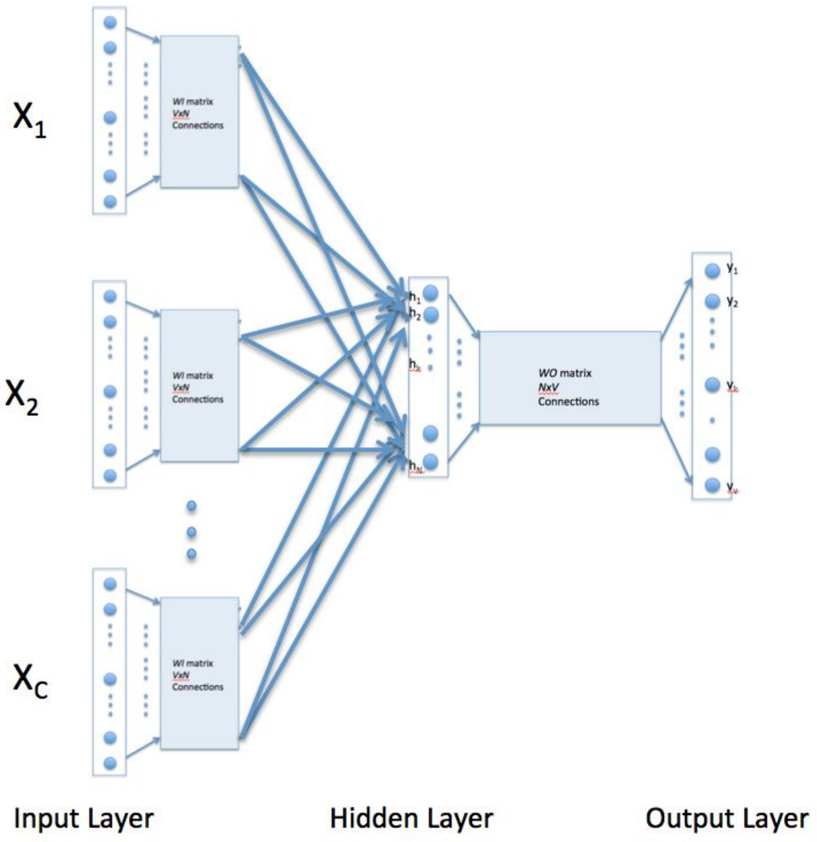


Figure 3: Working of MLP Classifier

4. **Prediction:** Given a document of an unknown author, the system will identify the owner of the document. If a person falsely claims to be the owner, then the system will be successful in identifying it.

4.2 Design of the proposed system

4.2.a Data Flow Diagram

Level 0:

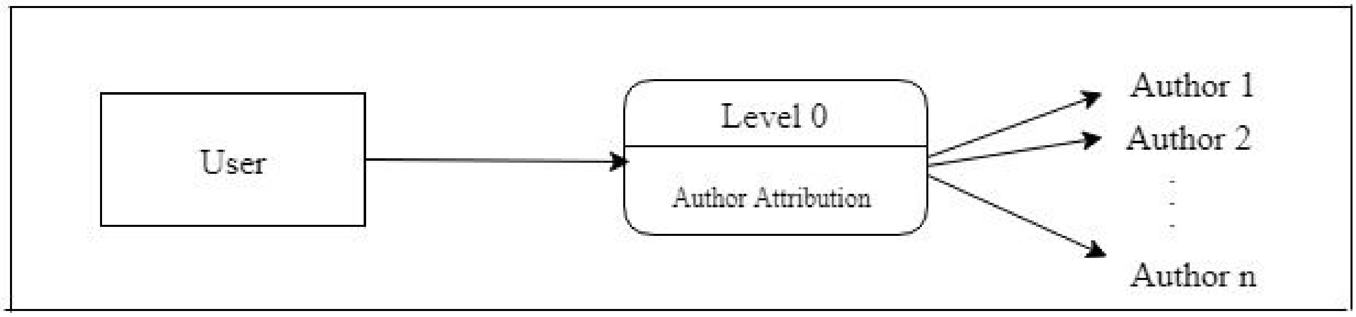


Figure 4: DFD Level 0

The figure above shows a context Data Flow Diagram that is drawn for Author Identification using Stylometry System. It contains a process that represents the system to model, in this case, the "Author Attribution". It also shows the participant who will interact with the system, called the external entities. In this example, User and Authors are the two entities who will interact with the system. In between the process and the external entities, there are data flow (connectors) that indicate the existence of information exchange between the entities and the system .Here the information is Document and Review Articles.

Level 1:

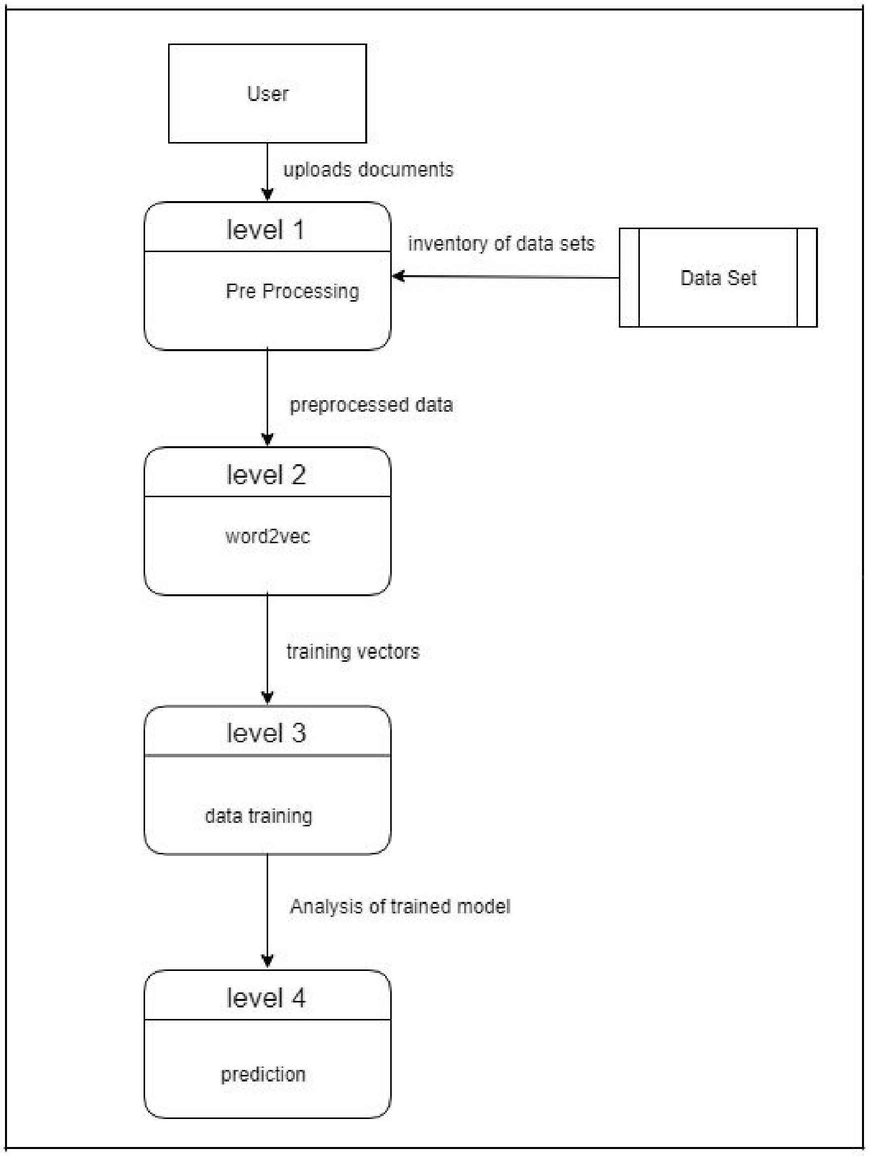


Figure 5: DFD Level 1

The figure above shows the level 1 DFD, which is the decomposition of the Author Attribution process shown in the context DFD. The Author Attribution System Data Flow Diagram contains four processes, one external entity and one data stores. A User can give the Document to the Pre Processing process, and the Inventory of dataset is provided by the data store. The Document with reduced dimensions is produced by pre processing process will be given to word2vec. Now the training vectors generated from the document is given to the data training process, using the experience from the learning the system will be trained. Analysis of trained model is done by the Prediction process thus giving the output.

Level 2:

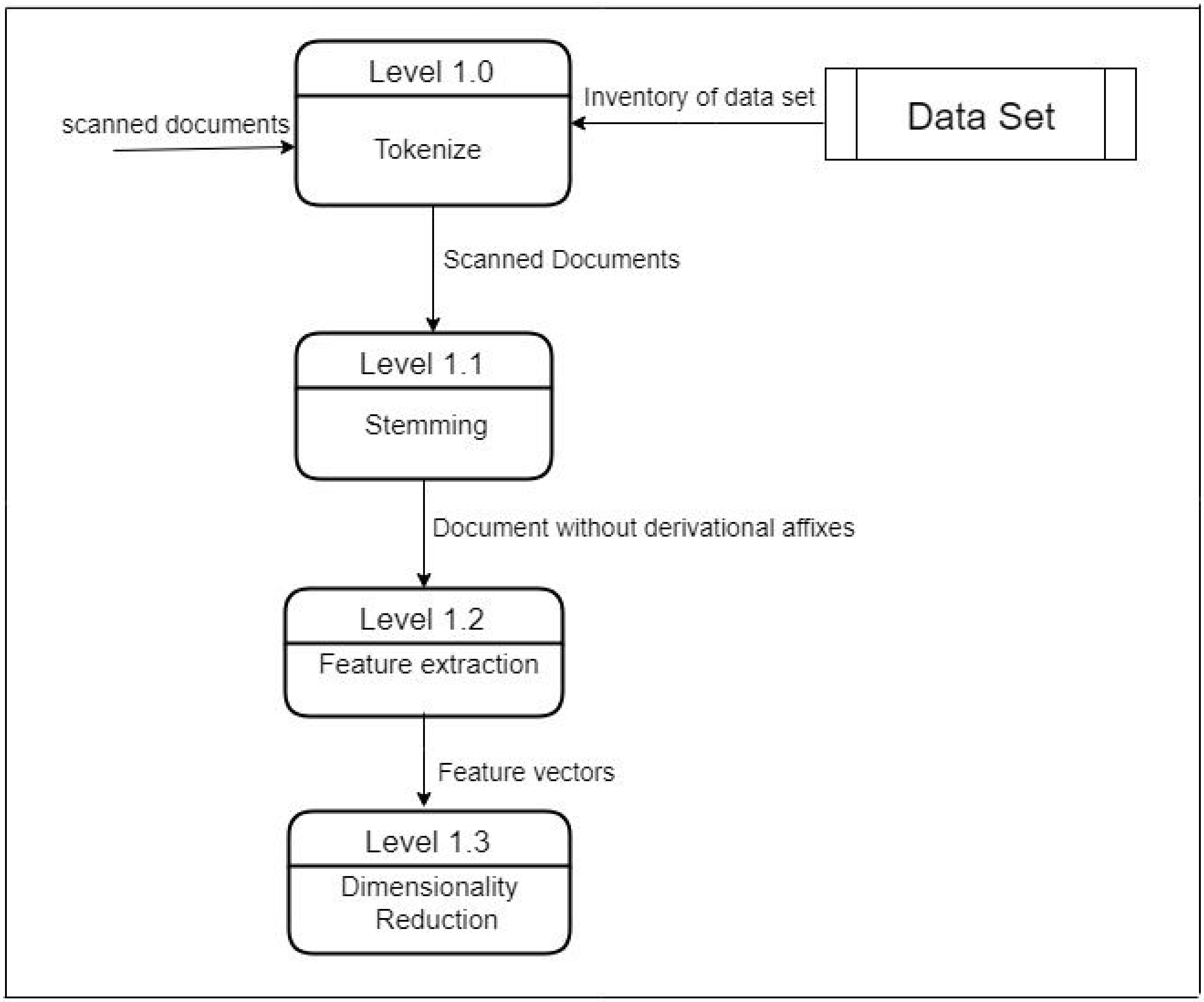


Figure 6: DFD Level 2 (1)

The figure above shows the level 2 DFD, which is the decomposition of the Preprocessing process shown in the Figure 3. The Pre Processing System Data Flow Diagram contains three processes and one data stores. The scanned document is provided to the Preprocessing process then the pre-processing process scans the documents and sends it to the Stemming process using porter's algorithm, and the dataset is provided by the data store. The Stemming process will generate a document that does not have derivational affixes and pass it for Feature Extraction. After getting vectors using word2vec model, the feature set is subjected to principal component analysis.

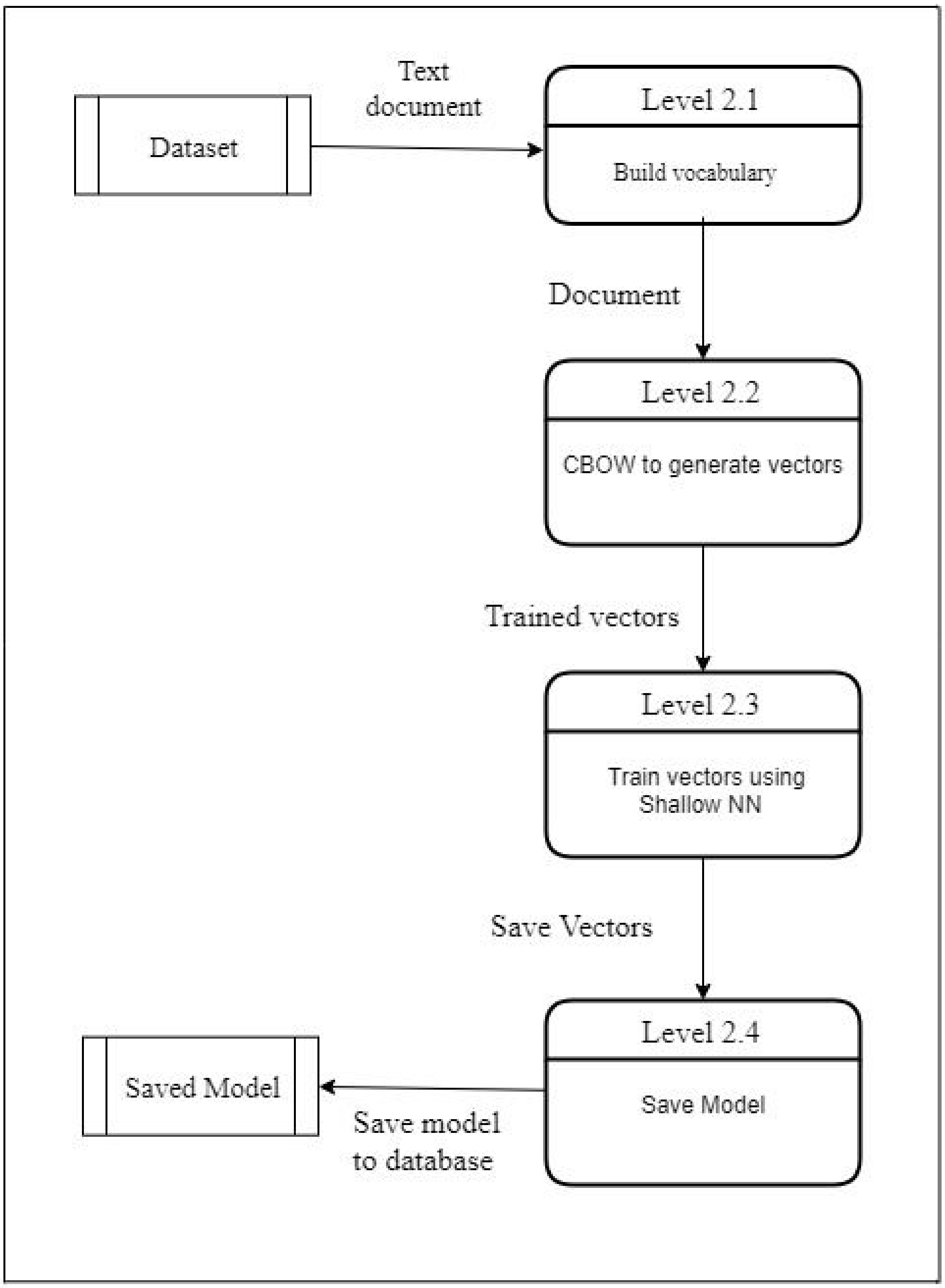


Figure 7: DFD Level 2 (2)

The figure above shows DFD for feature extraction module. Word embedding is used for feature extraction. The algorithm then moves over each word in the corpus and repeats the training step in an online fashion. The interesting property that word vectors obtained this way exhibit is that they encode not only syntactic but also semantic relationships between words. After scanning the document, a vocabulary of unique words is built. Embedding for these words is obtained by CBOW model in Gensim library. The word embedding for each document is saved before scanning next document.

4.2.b ER Diagram

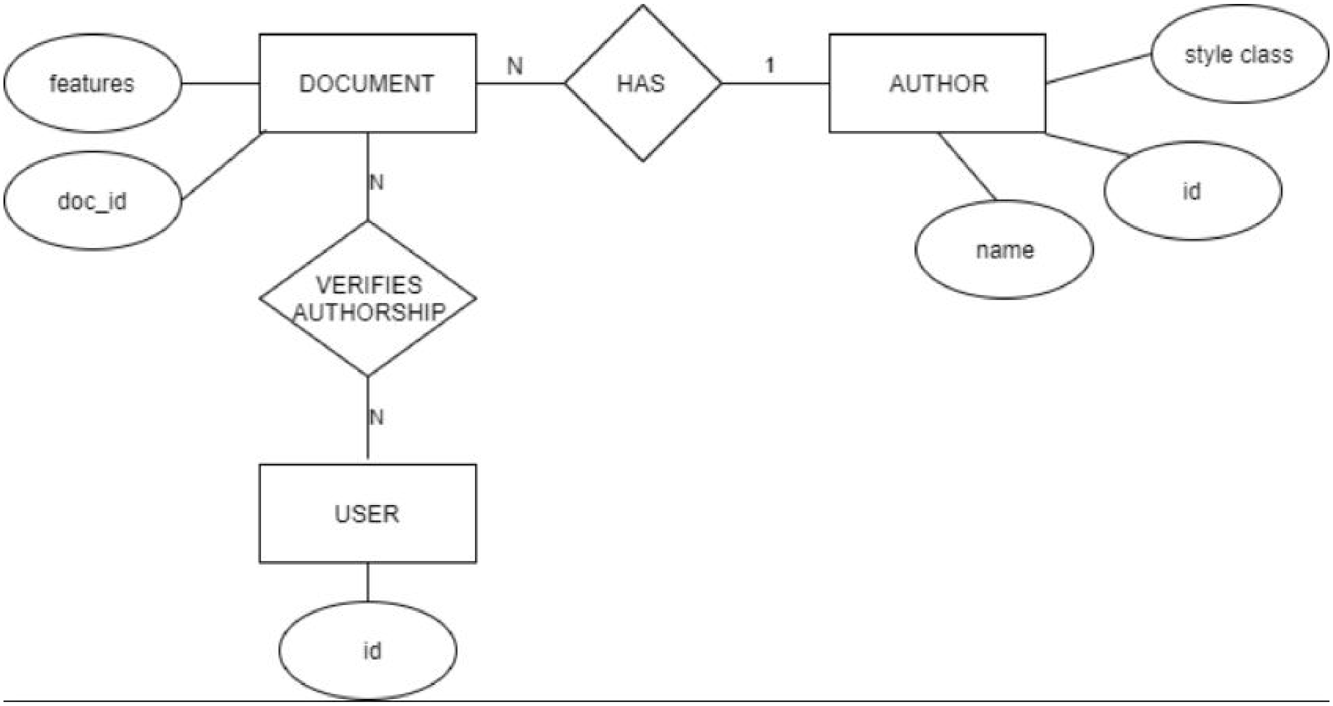
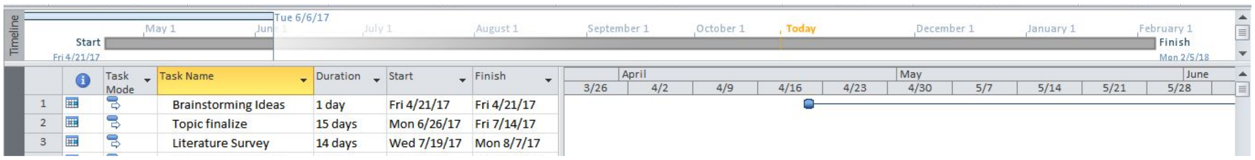


Figure 8: ER Diagram

Every Author will have multiple documents but every document will have a unique author. The attributes for every author would be the author's 'style of writing' or 'stylography', an 'author id' and the author's 'name'.

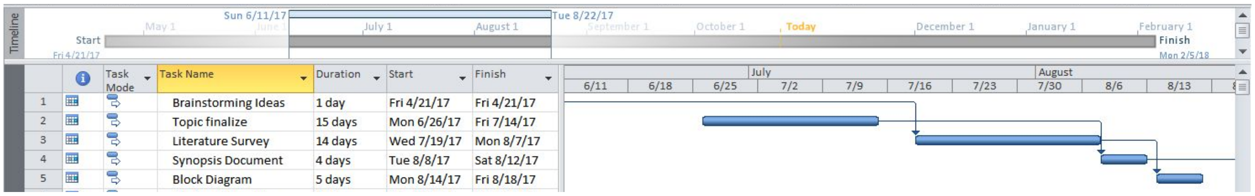
Every document would first be verified by the user or the admin. The user will have a unique id while the attributes of the document are: 'document's id' and 'features extracted'.

4.3 Project Scheduling and Tracking using Timeline / Gantt Chart



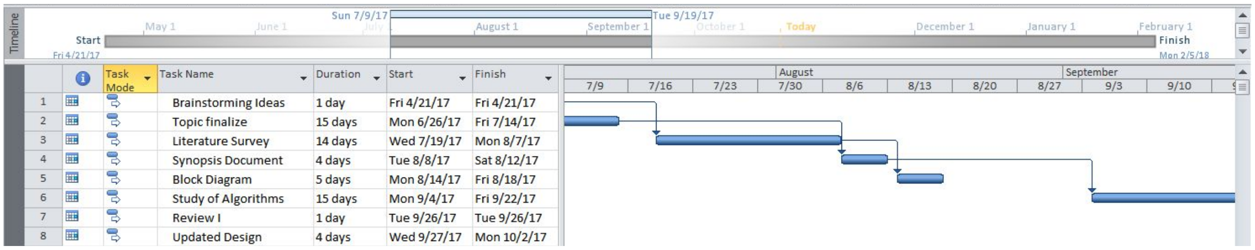
	Task Mode	Task Name	Duration	Start	Finish
1		Brainstorming Ideas	1 day	Fri 4/21/17	Fri 4/21/17
2		Topic finalize	15 days	Mon 6/26/17	Fri 7/14/17
3		Literature Survey	14 days	Wed 7/19/17	Mon 8/7/17

Figure 9: Gantt Chart(1)



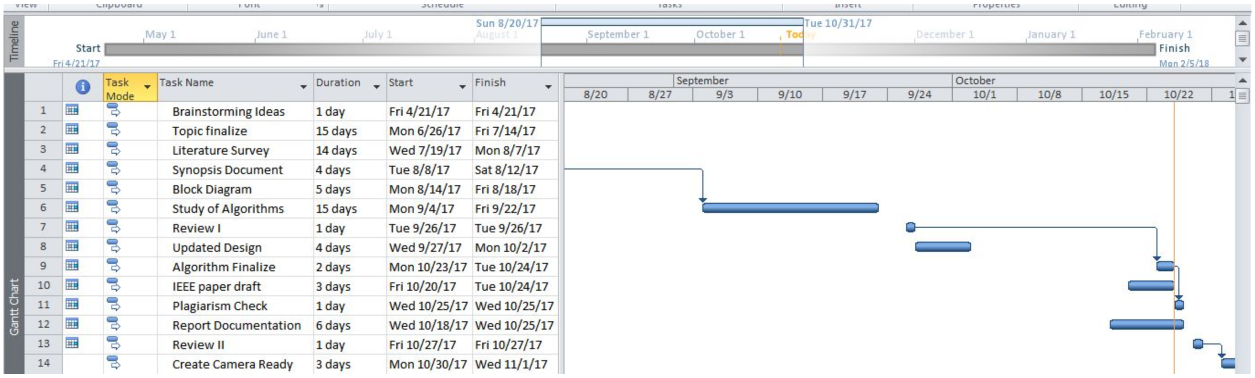
	Task Mode	Task Name	Duration	Start	Finish
1		Brainstorming Ideas	1 day	Fri 4/21/17	Fri 4/21/17
2		Topic finalize	15 days	Mon 6/26/17	Fri 7/14/17
3		Literature Survey	14 days	Wed 7/19/17	Mon 8/7/17
4		Synopsis Document	4 days	Tue 8/8/17	Sat 8/12/17
5		Block Diagram	5 days	Mon 8/14/17	Fri 8/18/17

Figure 10: Gantt Chart(2)



	Task Mode	Task Name	Duration	Start	Finish
1		Brainstorming Ideas	1 day	Fri 4/21/17	Fri 4/21/17
2		Topic finalize	15 days	Mon 6/26/17	Fri 7/14/17
3		Literature Survey	14 days	Wed 7/19/17	Mon 8/7/17
4		Synopsis Document	4 days	Tue 8/8/17	Sat 8/12/17
5		Block Diagram	5 days	Mon 8/14/17	Fri 8/18/17
6		Study of Algorithms	15 days	Mon 9/4/17	Fri 9/22/17
7		Review I	1 day	Tue 9/26/17	Tue 9/26/17
8		Updated Design	4 days	Wed 9/27/17	Mon 10/2/17

Figure 11: Gantt Chart(3)



	Task Mode	Task Name	Duration	Start	Finish
1		Brainstorming Ideas	1 day	Fri 4/21/17	Fri 4/21/17
2		Topic finalize	15 days	Mon 6/26/17	Fri 7/14/17
3		Literature Survey	14 days	Wed 7/19/17	Mon 8/7/17
4		Synopsis Document	4 days	Tue 8/8/17	Sat 8/12/17
5		Block Diagram	5 days	Mon 8/14/17	Fri 8/18/17
6		Study of Algorithms	15 days	Mon 9/4/17	Fri 9/22/17
7		Review I	1 day	Tue 9/26/17	Tue 9/26/17
8		Updated Design	4 days	Wed 9/27/17	Mon 10/2/17
9		Algorithm Finalize	2 days	Mon 10/23/17	Tue 10/24/17
10		IEEE paper draft	3 days	Fri 10/20/17	Tue 10/24/17
11		Plagiarism Check	1 day	Wed 10/25/17	Wed 10/25/17
12		Report Documentation	6 days	Wed 10/18/17	Wed 10/25/17
13		Review II	1 day	Fri 10/27/17	Fri 10/27/17
14		Create Camera Ready	3 days	Mon 10/30/17	Wed 11/1/17

Figure 12: Gantt Chart(4)

Chapter 5 : Implementation Details

The GUI and result of the system is described briefly. Here, the various models which are been implemented in our project are mentioned. Project implementation is the phase where visions and plans become reality.Implementation means carrying out the activities that has been planned. Implementation is the action that must follow any preliminary thinking in order for something to actually happen. In an information technology context, implementation encompasses all the processes involved in getting new software or hardware operating properly in its environment, including installation, configuration, running, testing, and making necessary changes.

5.1 Algorithms for the respective modules developed

1. Machine Learning Based Algorithms using Term Frequency.
2. Machine Learning Based Algorithms with Principal Component Analysis(PCA) using Term Frequency.
3. Machine Learning Based Algorithms using TF-IDF.
4. Machine Learning Based Algorithms with Principal Component Analysis and K-fold Validation.
5. Machine Learning Based Algorithms with K-fold Validation.
6. Word embedding based algorithms.
    6.1. sentence 2 vector
    6.2. word 2 vector

Corpus :

A corpus comprising of 438 authors was obtained using Project Gutenberg: Project Gutenberg electronic text archive, which contains some 50,000 free electronic books, hosted at http://www.gutenberg.org/. NLTK supports an API that can be used to download text from the Gutenberg archive. The training corpus consists of two documents by each author.

Stemming:

It is the process of reducing inflected,derived words to their word stem, base or root form. Porter’s algorithm was for stemming the corpus. The corpus was cleaned in order to remove numbers, combination of numbers and letters etcetera.

Principal Component Analysis:

Principal Component Analysis refers to analysis of data which will be responsible to identify the patterns and then finding the patterns to reduce the dimensions of the dataset drastically, taking into consideration the minimal loss of the information. One of the way for performing Principal Component Analysis is by choosing a subset of Principal Components and Variables.

Term Frequency- Inverse Document Frequency

In information retrieval, tf–idf or TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf-idf value increases proportionally to the number of times a word appears in the document and is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. Nowadays, tf-idf is one of the most popular term-weighting schemes; 83% of text-based recommender systems in the domain of digital libraries use tf-idf.

5.1.1 Machine Learning Based Algorithms using Term Frequency

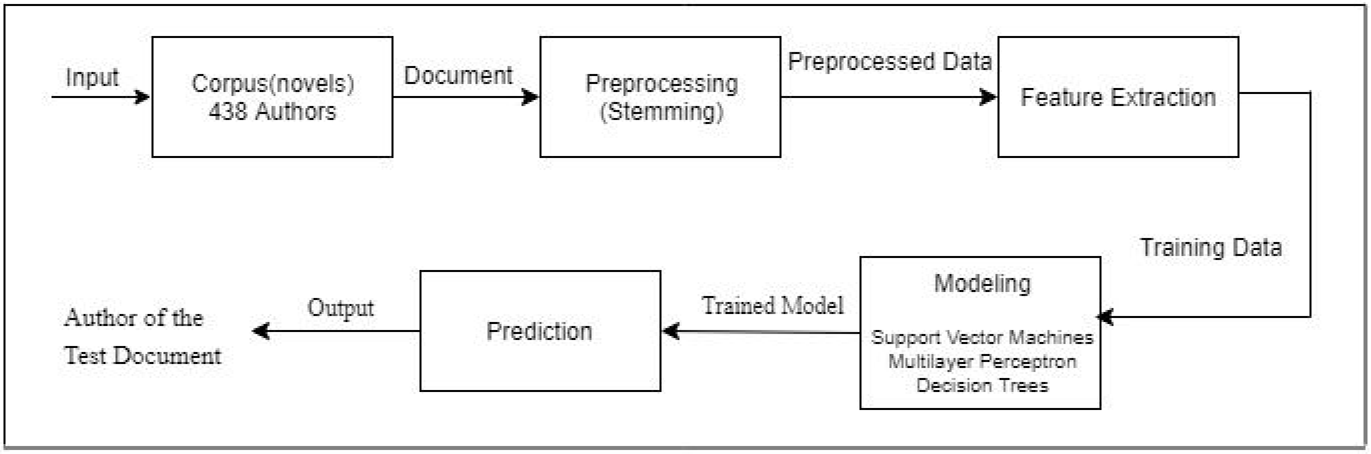


Figure 13: Model 1

Algorithm:

1. Perform Stemming on the dataset.
2. Calculate the frequency of only lexical features* of the documents.
3. Divide the frequencies into 3 categories, low frequency, mid frequency (count 500 to 1000) and high frequencies.
4. Split the low, mid and high frequency features table into Training and Testing Data set.
5. Train the system on using Decision Tree Classifier, SVM and Neural Network.
6. Predict the authors for unseen features.

This approach gives us an accuracy of 37% to 90% but the system is not dynamic, and only lexical features are considered here which is not ideal.

5.1.2 Machine Learning Based Algorithms with Principal Component Analysis(PCA) using Term Frequency

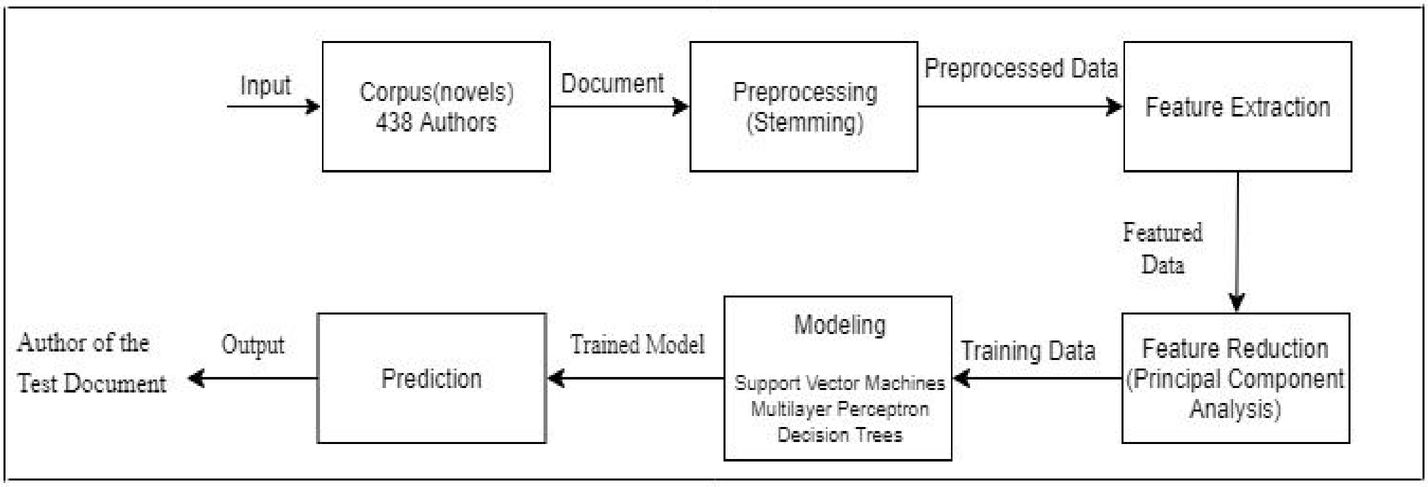


Figure 14: Model 2

Algorithm:

1. Perform Stemming on the dataset.
2. Calculate the frequencies of lexical features of the documents and bag-of-words**.
3. Like Module 1 divide the frequencies into 3 categories, low frequency, mid frequency and high frequencies.
4. Perform Principal Component Analysis on mid frequency document.
5. Split the low, mid and high frequency features table into Training and Testing Data set.
6. Train the system on using Decision Tree Classifier, SVM and Neural Network.
7. Predict the authors for unseen features.

This module gave us a low accuracy as compared to model 1 as PCA wiped out essential features being used in prediction. This approach gives us an accuracy of 30% to 82%. Moreover, here too only lexical features were being considered.

5.1.3 Machine Learning Based Algorithm using TF-IDF

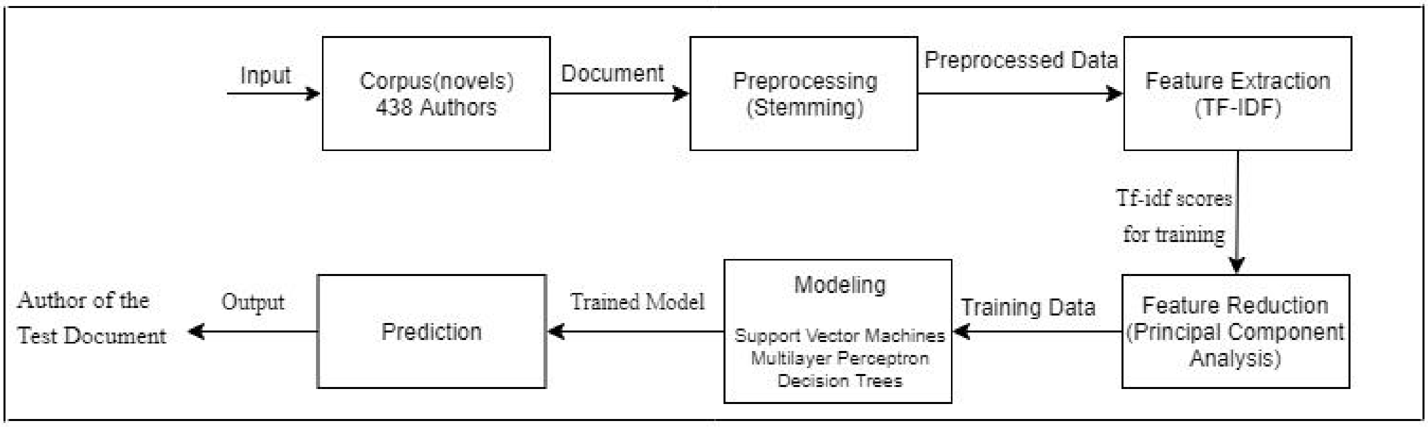


Figure 15: Model 3

Algorithm:

1. Perform stemming on the entire dataset
2. Calculate the “Term Frequency–Inverse Document Frequency” i.e. tf-idf score for the stemmed dataset.
3. Perform Principal Component Analysis (PCA) on the result of step 2.
4. Split the result table of step 3 into Training and Testing dataset.
5. Train the system using Decision Tree Classifier, SVM and MLP.
6. Predict the authors for unseen feature vectors.

This module gave us a very low accuracy of 10% to 32% because tf-idf score is not a very suitable approach for our dataset, which are large documents from many different authors which the number of documents per author varying a lot. Moreover, the system was static, that is to test or train the system on a new file or author, the entire system had to be run again.

5.1.4 Machine Learning Based Algorithms with Principal Component Analysis and K-fold Validation

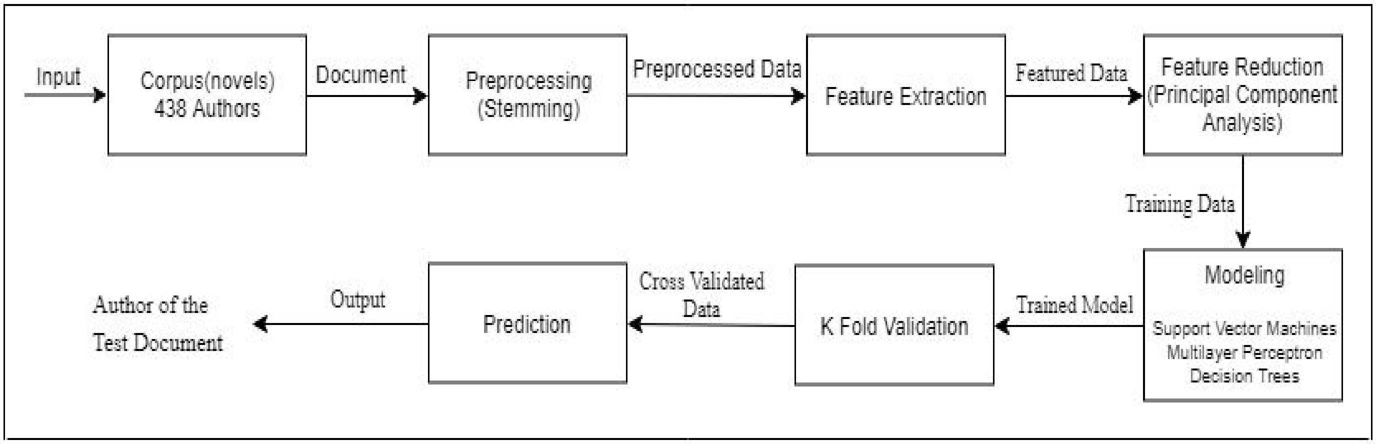


Figure 16: Model 4

Algorithm:

1. Perform Stemming on the dataset.
2. Calculate the frequencies of lexical features of the documents and bag-of-words.
3. Like Module 1 divide the frequencies into 3 categories, low frequency, mid frequency and high frequencies.
4. Perform Principal Component Analysis on mid frequency document.
5. Split the low, mid and high frequency features table into Training and Testing Data set.
6. Train the system using Decision Tree Classifier, SVM and Neural Network.
7. Perform k-fold cross validation on the dataset(k=10).
8. Predict the authors for unseen features.
9. Calculate the accuracy, the mean of the accuracy and standard deviation of the accuracy.

This module gave us a accuracy in the range of 31% to 85%. There was a bit drop in accuracy due to PCA.

5.1.5 Machine Learning Based Algorithms with K-fold Validation

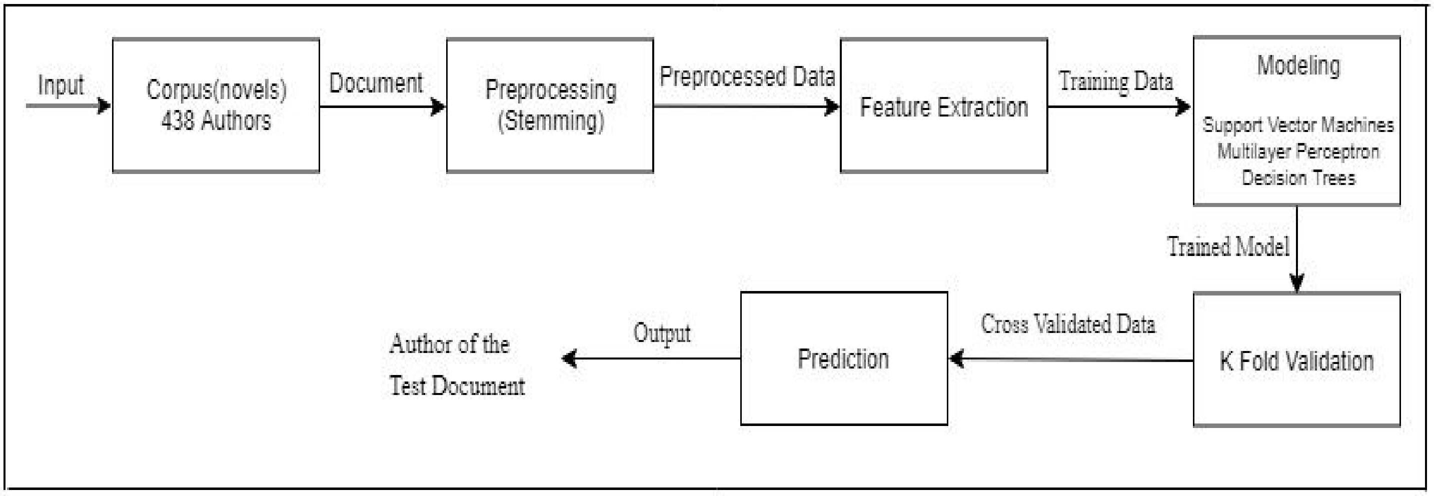


Figure 17: Model 5

Algorithm:

1. Perform Stemming on the dataset.
2. Calculate the frequencies of lexical features of the documents and bag-of-words.
3. Like Module 1 divide the frequencies into 3 categories, low frequency, mid frequency and high frequencies.
4. Split the low, mid and high frequency features table into Training and Testing Data set.
5. Train the system using Decision Tree Classifier, SVM and Neural Network.
6. Perform k-fold cross validation on the dataset.
7. Predict the authors for unseen features.

8. Calculate the accuracy, the mean of the accuracy and standard deviation of the accuracy.

The only difference between Module 5 and Module 4 is performing PCA, however this small change had a huge impact of almost 7% - 8% on the accuracy of the system, Module 5 gave us an accuracy of around 41% to 92%.

5.1.6 Word Embedding Based Algorithms

The Authorship Attribution (AA) task consists in identifying the author of a given text among a list of candidates authors. In this approach, the problem is treated as a supervised classification task, when a classifier is built using a training set and the task consists in classifying correctly the samples from a testing set. Word embeddings after cleaning the training data, we use the Word2vec method to obtain the vectors for each document. The Word2vec module offers two possible approaches to build the model, the Distributed Model (DM), which tries to predict the context of a given element and the Distributed Bag of Words (DBOW), which tries to predict the word given the context.

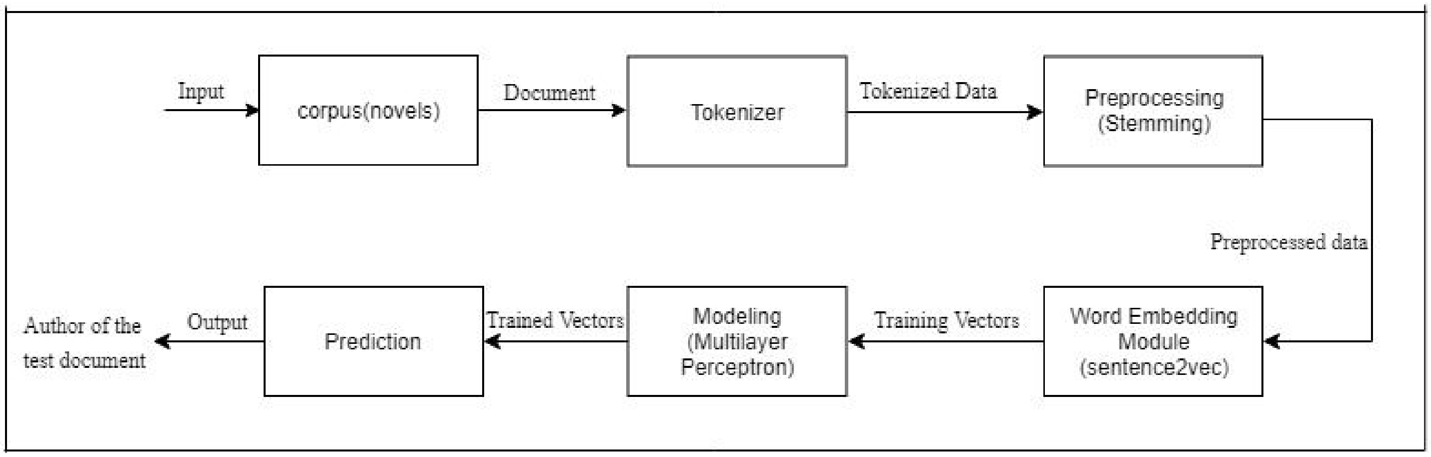


Figure 18: Sentence2Vec

Algorithm :

1. Tokenize dataset and perform stemming.
2. Obtain vectors for each sentence in a document.
3. Train model using these vectors and machine learning algorithms such as neural networks and svm.
4. Test the trained model.

Accuracy by this model on a 5 author dataset trained using MLP is 68% and on a 15 author dataset is 49%.

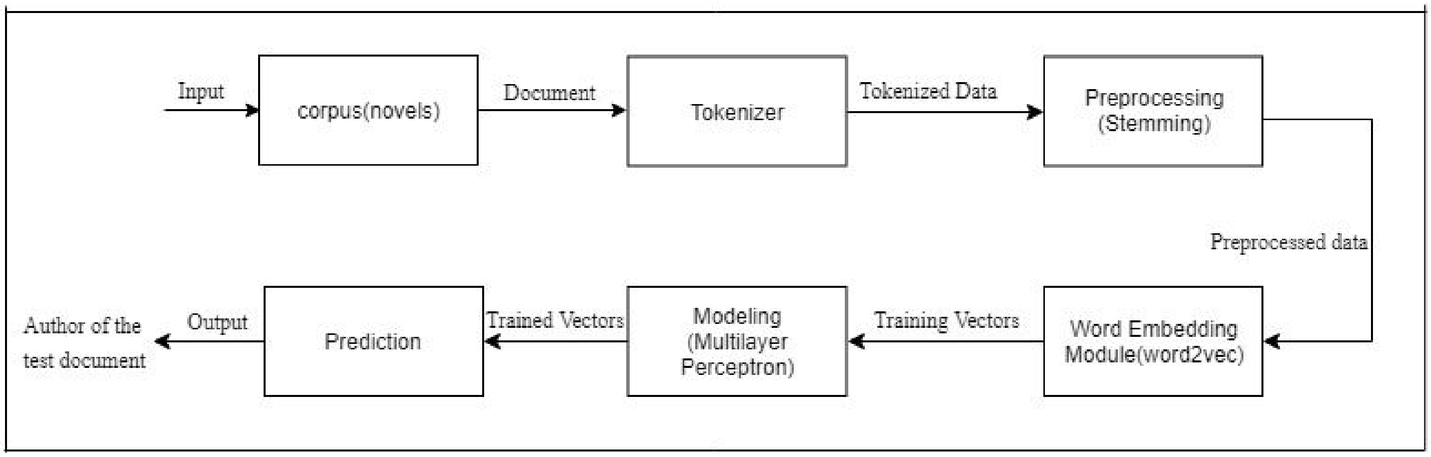


Figure 19: Word2Vec

Algorithm :

1. Tokenize dataset and perform stemming.
2. Obtain word2vec embeddings for each document (vector size = 300).
3. Train model using these vectors and machine learning algorithms such as neural networks and svm.
4. Test the trained model.

Accuracy by this model on a 15 author dataset trained using MLP is 83% . On 5 authors accuracy is 98.11%

Example:

Suppose the input text is : “ this elaborate and excellent collection ”. After tokenize module the output will be : [‘this’, ‘elaborate’,’and’ , ‘excellent’ , ‘collection’]. The word2vec embedding module will represent each unique word as a vector. These vectors are generated by CBOW algorithm which considers continuous bag of words to derive semantic as well as syntactic features for the word in consideration. Elaborate : [1.23, 0.22, ...], excellent : [0.34, 1.46, ...]. The next module takes in these vectors as input and tries to learn an author's writing style based on it. It is supervised learning using MLP.

Code:

test.py

```
import pandas as pd
import gensim
from gensim.models.doc2vec import TaggedDocument
from nltk import RegexpTokenizer
from nltk.corpus import stopwords
from os import listdir
from os.path import isfile, join
import os
```

```
from nltk.tokenize import sent_tokenize, word_tokenize
import numpy as np
import pylab
#Data cleaning
def nlp_clean(data):
  new_data = []
  for d in data:
    new_str = d.lower()
    dlist = tokenizer.tokenize(new_str)
    dlist = list(set(dlist).difference(stopword_set))
    new_data.append(dlist)
  return new_data
#Generating iterator
class LabeledLineSentence(object):
   def __init__(self, doc_list, labels_list):
      self.labels_list = labels_list
      self.doc_list = doc_list
   def __iter__(self):
      for idx, doc in enumerate(self.doc_list):
          yield gensim.models.doc2vec.LabeledSentence(doc,[self.labels_list[idx]])
path = "C:/xampp/htdocs/BE/upload_folder"
#now create a list that contains the name of all the text file in your data #folder
#create a list data that stores the content of all text files in order of their names in docLabels
data =[]
d = []
tagged_docs=[]
docLabels = []
authors = []
words=[]
vec = []
row = 0
tokenizer = RegexpTokenizer(r'\w+')
stopword_set = set(stopwords.words('english'))
```

```
turn = 0
dir_path = "C:/xampp/htdocs/BE/upload_folder"
file_path = [x[0] for x in os.walk(dir_path)]
for each_dir in file_path[0:]:
    each_file = os.listdir(each_dir)
    if len(each_file) > 0:
        for file in each_file:
            full_file_path = each_dir+'/'+file
            file_path=full_file_path.split(os.sep)
            file_content = open(full_file_path,'rb').read()
            #Tokenize into sentences
            data=sent_tokenize(str(file_content))
            #Data cleaning
            data= nlp_clean(data)
            i=0
            del authors[:]
            del words[:]
            del   vec[:]
            d.extend(data)
            i =0
            #Generate vectors
            model = gensim.models.Word2Vec(d, min_count=1,size=300)
            for word in model.wv.vocab:
                vec.append( model.wv[word])
                words.append(word)
            df1 = pd.DataFrame(np.array(vec).reshape(len(model.wv.vocab),300))
            df = pd.DataFrame(np.array(words).reshape(len(words),1))
            df = pd.concat([df,df1],axis=1)
            #Saving vectors to a csv file
            df.to_csv('C:/xampp/htdocs/BE/word2vec_test.csv', mode='a', header = None)
            del data[:]
            del d[:]
array= df._values
```

```
from sklearn import preprocessing
from sklearn import utils
import csv
from sklearn.decomposition import PCA
import pandas as pd

X =array[:,2:300]
print()
import random
import numpy as np
from sklearn.neural_network import MLPClassifier
#Classifier
import pickle
clf = pickle.load(open("C:/xampp/htdocs/BE/model.pkl", 'rb'))
#Prediction
Z = clf.predict(X)
print('The prediction of the author is:')
print(Z)
print()
Z= np.ndarray.tolist(Z)
Edmund_Goldsmid = 0
EdwardBulwerLytton = 0
JohnBurroughs = 0
JohnGreenleafWhittier = 0
MartinAnderson = 0
#Counting authors predicted
for name in Z:
    if name == 'edmund goldsmid':
        Edmund_Goldsmid=Edmund_Goldsmid+1
    if name == 'Edward Bulwer-Lytton':
        EdwardBulwerLytton+=1
    if name == 'John Burroughs':
        JohnBurroughs+=1
    if name == 'john greenleaf whittier':
        JohnGreenleafWhittier+=1
    if name == 'martin anderson nexo':
        MartinAnderson+=1
print()
print('Edmund Goldsmid Count=')
print(Edmund_Goldsmid)
print()
print('Edward Bulwer-Lytton Henry Lewis Count=')
print(EdwardBulwerLytton)
print()
print('John Burroughs Count=')
print(JohnBurroughs)
print()
print('John Greenleaf Whittier Sue Count=')
print(JohnGreenleafWhittier)
print()
print('Martin Anderson Count=')
print(MartinAnderson)
#Visualization
import matplotlib.pyplot as plt
#Data to plot
labels = 'Edmund Goldsmid', 'Edward Bulwer-Lytton', 'John Burroughs', 'John Greenleaf Whittier', 'Martin
Anderson'
sizes = [Edmund_Goldsmid, EdwardBulwerLytton, JohnBurroughs, JohnGreenleafWhittier,
MartinAnderson]
colors = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue', 'red']
explode = (0, 0, 0, 0, 0)  # explode 1st slice
#Plot
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
        autopct='%1.1f%%', shadow=True, startangle=140)
plt.axis('equal')
plt.savefig('C:/xampp/htdocs/BE/chart.png')
```

```
print("successful")
```

test.php

```
<html>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<style>
body {
    font-family: inherit;
    background-image: url("background.jpg");
}
h1{text-shadow: 2px 2px 5px grey;}
p{text-shadow: 2px 2px 4px grey;}
.sidenav {
    height: 100%;
    width: 250px;
    position: fixed;
    z-index: 1;
    top: 0;
    left: 0;
    background-color: #111;
    overflow-x: hidden;
    padding-top: 20px;
}
.sidenav a {
    padding: 6px 8px 6px 16px;
    text-decoration: none;
    font-size: 18px;
    color: #818181;
    display: block;
}
.sidenav a:hover {
    color: #f1f1f1;
}
```

```
.main {
   margin-left: 250px; /* Same as the width of the sidenav */
   font-size: 18px; /* Increased text to enable scrolling */
   padding: 0px 10px;
}
@media screen and (max-height: 450px) {
   .sidenav {padding-top: 15px;}
   .sidenav a {font-size: 18px;}
}
ul
{
list-style-type: none;
}
</style>
</head>
<body style="color:white;">
<script>
function myFunction() {
   var result="<?php  py_call(); ?>";
   var s= document.getElementById('res');
   s.value = result
  return false;
}
</script>
<div class="sidenav">
  <!-- Sidebar -->
      <div id="sidebar-wrapper">
         <ul class="sidebar-nav">
            <li class="sidebar-brand">
               <a href="index.html">
                  SAI
               </a>
            </li>
            <li>
                <a href="upload.html">Upload a file</a>
            </li>
            <li>
                <a href="test.php">Analyse</a>
            </li>
            <li>
                <a href="charts.html">Comparitive Charts</a>
            </li>
            <li>
                <a href="about.html">About</a>
            </li>
            <li>
                <a href="https://ves.ac.in/vesit/contact-us/">Contact</a>
            </li>
        </ul>
    </div>
    <!-- /#sidebar-wrapper -->
</div>
<div class="main">
    <div class="container">
        <h1 style="color:red; text-align: center">Author of the document</h1>
        <form method="post">
            <button class="btn btn-lg btn-info" onclick="myFunction()">Predict</button>
        </form>
        <!-- <textarea style="color:red" id="res" rows="4" cols="50"></textarea> -->
        <div class="row">
            <div class="col-sm-8-pull-4">
                <div class="thumbnail">
                    <a href="chart.png" target="_blank">
                    <img class="img-responsive" src="chart.png" alt="Comparitive">
                    <div class="caption">
                        <p style="text-align: center">Prediction</p>
                    </div>
                    </a>
                </div>
            </div>
        </div>
    </div>
//To run the backend code
<?php
ini_set('max_execution_time', 3000); //300 seconds = 5 minutes
function py_call()
{
    $python = "C:/Users/Admin/AppData/Local/Programs/Python/Python36/python.exe";
    $file = "C:/xampp/htdocs/BE/test.py";
    $cmd = "$python $file";
    $op = exec($cmd);
    echo $op;
}
?>
</div>
</body>
</html>
```

5.2 Comparative analysis with the existing Algorithms

Existing System	SAI
In traditional studies on authorship attribution, the focus is on small sets of authors. Generally the set contains less than 50 authors, in some cases it is as low as 10 authors.	In our project we have considered 438 authors to solve author attribution problem. This is a very large corpus compared to the ones used by the existing systems.
For adding new entries to the dataset the whole process of preprocessing, feature extraction and training needs to be done again and again which takes up more memory space and increases the time and computational complexities.	For adding new entries to the dataset whole process of preprocessing and training need not be done again and again which reduces the memory requirements and has relatively less time and computational complexities.
The accuracy of the existing systems drop sharply when the corpus size is increased to a large dataset. The highest accuracy achieved in any existing system never exceeded 85% despite considering only a few authors.	The accuracy of our system does not drop much when the number of authors increases. The highest accuracy achieved by our model is 98.11% when 5 authors were considered.
The existing systems invests a lot of time and processing power in preprocessing steps like stemming and principal component analysis which might decrease the space to be occupied by the corpus but results in the loss of vital information regarding the data.	In our system instead of losing vital information due to Principal Component Analysis and Stemming, we perform data cleaning and use bag of words in preprocessing. This enables us to have a preprocessed corpus while no vital information is lost.
Owing to the low accuracies and high error rates of the existing systems, they lose their credibility to be used by cyber police and security systems in forensics.	Our system has a high accuracy due to which it empowers cyber security forces to rely on its results if used in cyber security and forensics.

Table 1: Comparison between existing and proposed system

5.3 Evaluation of the developed System

Model 1: Machine Learning Based Algorithms using Term Frequency

Algorithm	**Low Frequency**	**Mid Frequency**	**High Frequency**
SVM (C=1.5, Degree=3, Kernel=poly)	57-62%	65-70%	**85-90**%
Neural Network - MLP (hidden_layer_sizes=(32, 32) , alpha = 0.001)	60-63%	60-65%	60-70%
Decision Tree (CART)	37-40%	37-42%	43-47%

Table 2: Comparison of Model 1

Model 2: Machine Learning Based Algorithms with PCA using Term Frequency (n = Number of Components)

Algorithm	**Low Frequency**(n=15000-20000)	**Mid Frequency**(n=700)	**High Frequency**(n=1000 - 2500)
SVM (C=1.5, Degree=3, Kernel=poly)	60-65%	64-67%	**77-82**%
Neural Network - MLP (hidden_layer_sizes=(32,32) , alpha = 0.001)	53-56%	57-59%	54-58%
Decision Tree (CART)	30-32%	32-34%	34-37%

Table 3: Comparison of Model 2

Model 3: Machine Learning Based Algorithms Using TF-IDF.

Algorithm	**Accuracy**
SVM (C=1.5, Degree=3, Kernel=poly)	**22-32%**
Neural Network - MLP (hidden_layer_sizes=(32,32) ,alpha = 0.001)	8-10%
Decision Tree (CART)	3-10%

Table 4: Comparison of Model 3

Model 4: Machine Learning Based Algorithms with PCA and K-fold validation (n=Number of Components and K = 10)

Algorithm	**Low Frequency**(n=15000-20000)	**Mid Frequency**(n=700)	**High Frequency**(n=1000 - 2500)
SVM (C=1.5, Degree=3, Kernel=poly)	60-66%	65-68%	**84-85%**
Neural Network - MLP(hidden_layer_sizes=(32,32) ,alpha = 0.001)	56-58%	70-73%	68-70%
Decision Tree (CART)	31-34%	35-37%	40-43%

Table 5: Comparison of Model 4

Model 5: Machine Learning Based Algorithms with K-fold validation (K=10)

Algorithm	**Low Frequency**	**Mid Frequency**	**High Frequency**
SVM (C=1.5, Degree=3, Kernel=poly)	60-62%	75-77%	**90-92%**
Neural Network - MLP(hidden_layer_sizes =(32,32) ,alpha = 0.001)	66-68%	75-77%	67-75%
Decision Tree (CART)	41-45%	44-55%	47-52%

Table 6: Comparison of Model 5

Model 6: sentence2Vec

Algorithm	**Accuracy**
SVM (C=1.5, Degree=3, Kernel=poly)	33%(5 authors) , 15%(15 authors)
Neural Network - MLP(hidden_layer_sizes=(32,32) ,alpha=0.001)	**68%**(5 authors) , 49%(15 authors)
Decision Tree (CART)	41%(5 authors) , 22%(15 authors)

Table 7: Comparison of Model 6

Model 7: sentence2Vec with PCA

Algorithm	**Accuracy**
SVM (C=1.5, Degree=3, Kernel=poly)	41%(5 authors) , 19%(15 authors)
Neural Network - MLP (hidden_layer_sizes=(32,32) , alpha = 0.001)	**70%**(5 authors) , 67%(15 authors)
Decision Tree (CART)	54%(5 authors) , 43%(15 authors)

Table 8: Comparison of Model 7

Model 8: word2Vec (vector size = 300)

Algorithm	**Accuray**
SVM (C=1.5, Degree=3, Kernel=poly)	65%(5 authors),
Neural Network - MLP(hidden_layer_sizes=(32,32) , alpha = 0.001)	**98.11%** (5 authors), 84%(15 authors)
Decision Tree (CART)	97.45%(5 authors), 45%(15 authors)

Table 9: Comparison of Model 8

Chapter 6 : Testing

In this chapter, various modules of the system are tested, discussing how easily the system can be used by the user of the system. Various types of testing such as unit testing, integration testing and user acceptance testing are performed in this chapter.

6.1 Unit Testing

In this stage of testing, the individual units of the application are tested. The errors reported during the test are fixed and the testing is performed recursively until all the errors are fixed and the application gives desired output as mentioned in the requirements.

The following table shows the various test cases under unit testing:

Test Case No.	Test Case	Description	Input	Expected Output	Actual Output	Pass/ Fail
1.	Testing file edmund.txt	A document written by author Edmund Goldsmid on which the system has not been trained before is uploaded to the system to predict the author	A text file written by Edmund Goldsmid	The pie chart given as output should contain high percentage, or a large sector showing Edmund Goldsmid as the predicted or probabilistic author	The System predicts that there is a 57% probability that the author of the document uploaded is Edmund Goldsmid	Pass
2.	Testing file ‘edward b.txt’	A document written by author Edward Bulwer-Lytton on which the system has not been trained before is uploaded to the system to predict the author	A text file written by Edward Bulwer-Lytton	The pie chart given as output should contain high percentage, or a large sector showing Edward Bulwer-Lytton as the predicted or probabilistic author	The System predicts that there is a 42% probability that the author of the document uploaded is Edward Bulwer-Lytto n	Pass

3.	Testing file john b.txt	A document written by author John Burroughs on which the system has not been trained before is uploaded to the system to predict the author	A text file written by John Burroughs	The pie chart given as output should contain high percentage, or a large sector showing John Burroughs as the predicted or probabilistic author	The System predicts that there is a 51% probability that the author of the document uploaded is John Burroughs	Pass
4.	Testing file 'john green.txt'	A document written by author John Greenleaf Whittier on which the system has not been trained before is uploaded to the system to predict the author	A text file written by John Greenleaf Whittier	The pie chart given as output should contain high percentage, or a large sector showing John Greenleaf Whittier as the predicted or probabilistic author	The System predicts that there is a 74% probability that the author of the document uploaded is John Greenleaf Whittier	Pass
5.	Testing file martin.txt	A document written by author Martin Anderson Nexo on which the system has not been trained before is uploaded to the system to predict the author	A text file written by Martin Anderson Nexo	The pie chart given as output should contain high percentage, or a large sector showing Martin Anderson Nexo as the predicted or probabilistic author	The System predicts that there is a 44% probability that the author of the document uploaded is Martin Anderson Nexo	Pass
6.	Testing the buttons and navigation of the front-end	On click of the buttons and list, appropriate pages must open and the desired output must be obtained. This testing must be done for every button and click function. The navigation of all pages from every single page	Click on the buttons: 'USE SAI' 'Comparative Charts' 'About' 'Contact' 'Upload a file'	The desired page must open and in case of uploading a file the file selected by the author should upload to the system with	All the functionalities of the front-end were upto the mark and running and functioning smoothly without any	Pass

		also has to be checked and verified to avoid error in navigation if the user is at a certain page and desires to navigate to another page shown in the available tabs or pages.		ease and on clicking contact us the VESIT Contact Us page must open.	errors or malicies.	

Table 10: Unit Test Cases

6.2 Integration Testing

The output of the trained model is used to predict the author of the document. The output of the model is displayed on the Graphical User Interface on the Xampp Server. The python code at the back end is integrated with the web technologies in the front end. After resolving a few minor bugs, the modules worked perfectly.

6.3 User Acceptance Testing

The system is tested on multiple users of different ages to understand the user acceptance of the application. Within a few minutes, the users were able to use to application easily. With a simple UI design, the users were easily able to understand the purpose and the usage of the application. The users were able to explore the features of the app by themselves.

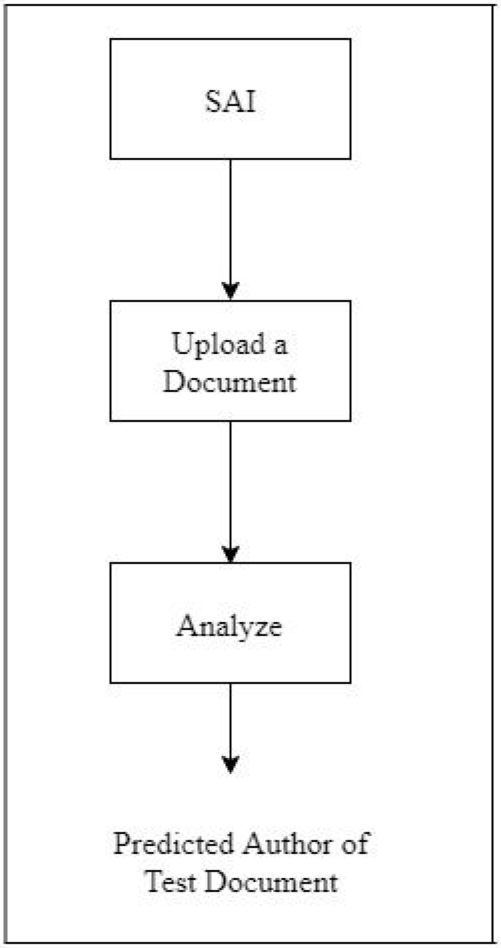
Some of the comments given by the users were:

- The UI is consistent, familiar and responsive.
- The analytics provides a bird’s eye view of the entire scenario.
- The side navbar is attractive and quite trendy.
- The background image used gives a realistic feel of the system.

Chapter 7: Result Analysis

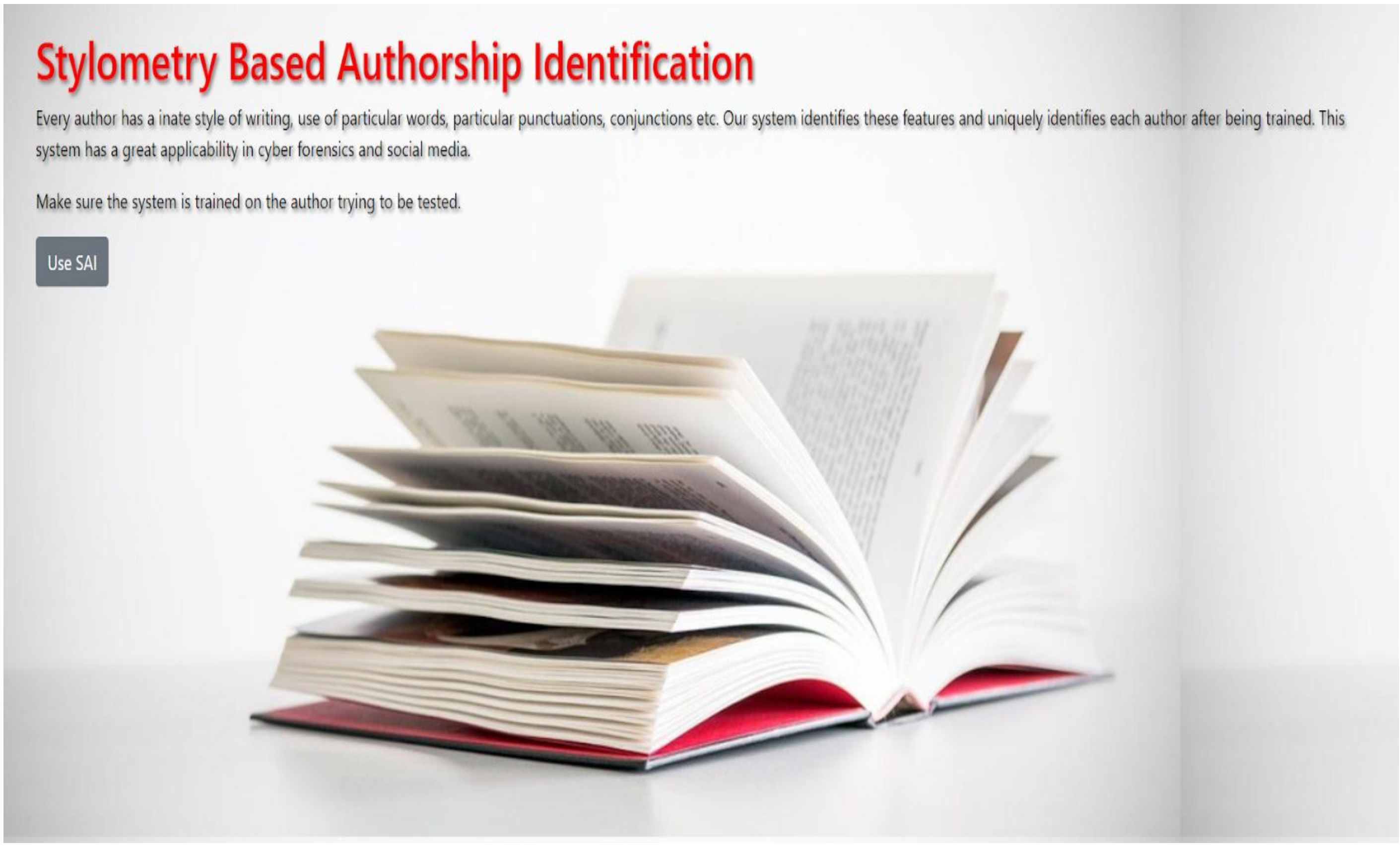
The Result Analysis focuses mainly on the working aspect i.e. what all functions are provided in the system, the screenshots attached in this chapter helps in better understanding to the user as how the system will look like(User Interface) and how the simulation model is realized practically.
The reports generated section in this chapter shows the comparative analysis between different models developed.

7.1 Simulation Model



An application has been created that the user can use to check for the authorship of a document. The user must upload the document whose authorship is in question, by clicking on the upload file button. The document will be analysed and vectors for the same will be generated. These vectors will be tested against the trained model. To view output of the algorithm, the user must click on analyse. This analysis is represented in the form of pie chart showing the probability that the uploaded document was authored by a particular author.

7.2 Screenshots of the User Interface

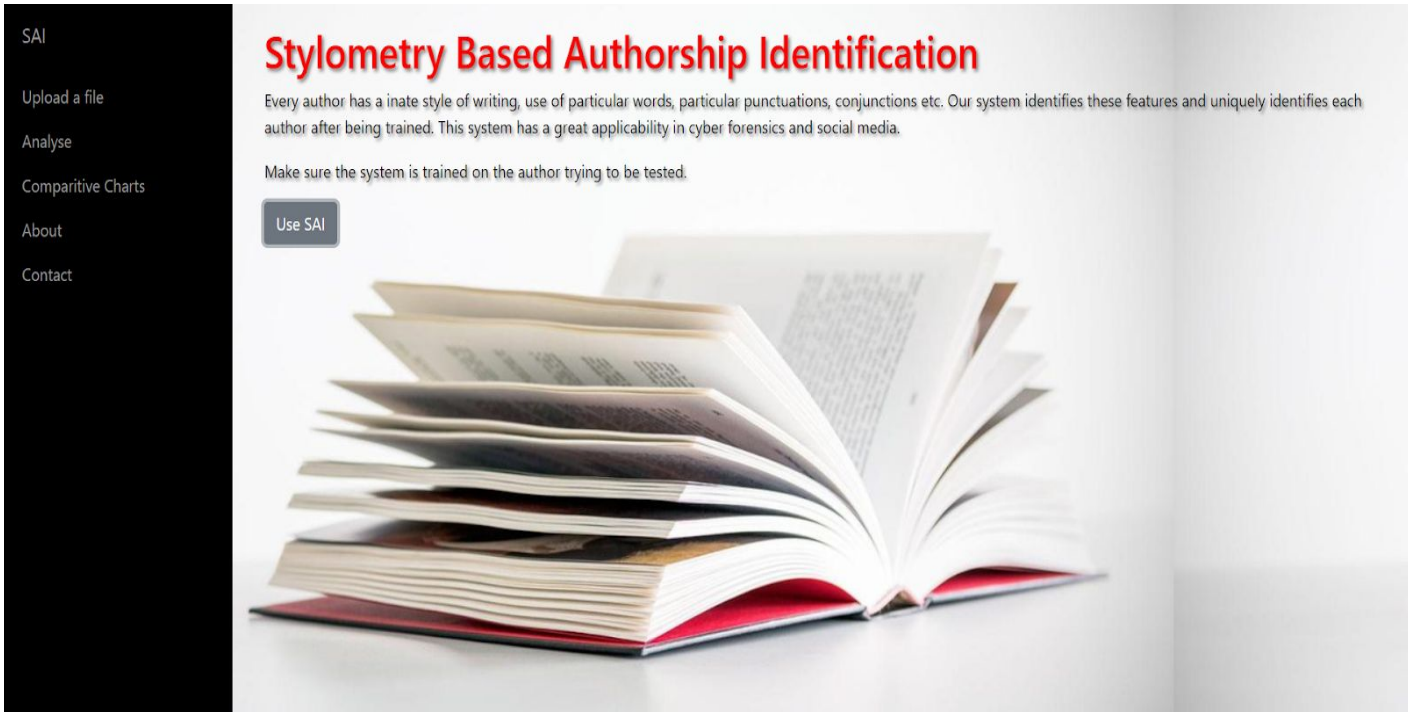


Stylometry Based Authorship Identification

Every author has a inate style of writing, use of particular words, particular punctuations, conjunctions etc. Our system identifies these features and uniquely identifies each author after being trained. This system has a great applicability in cyber forensics and social media.

Make sure the system is trained on the author trying to be tested.

Figure 20: Home Page

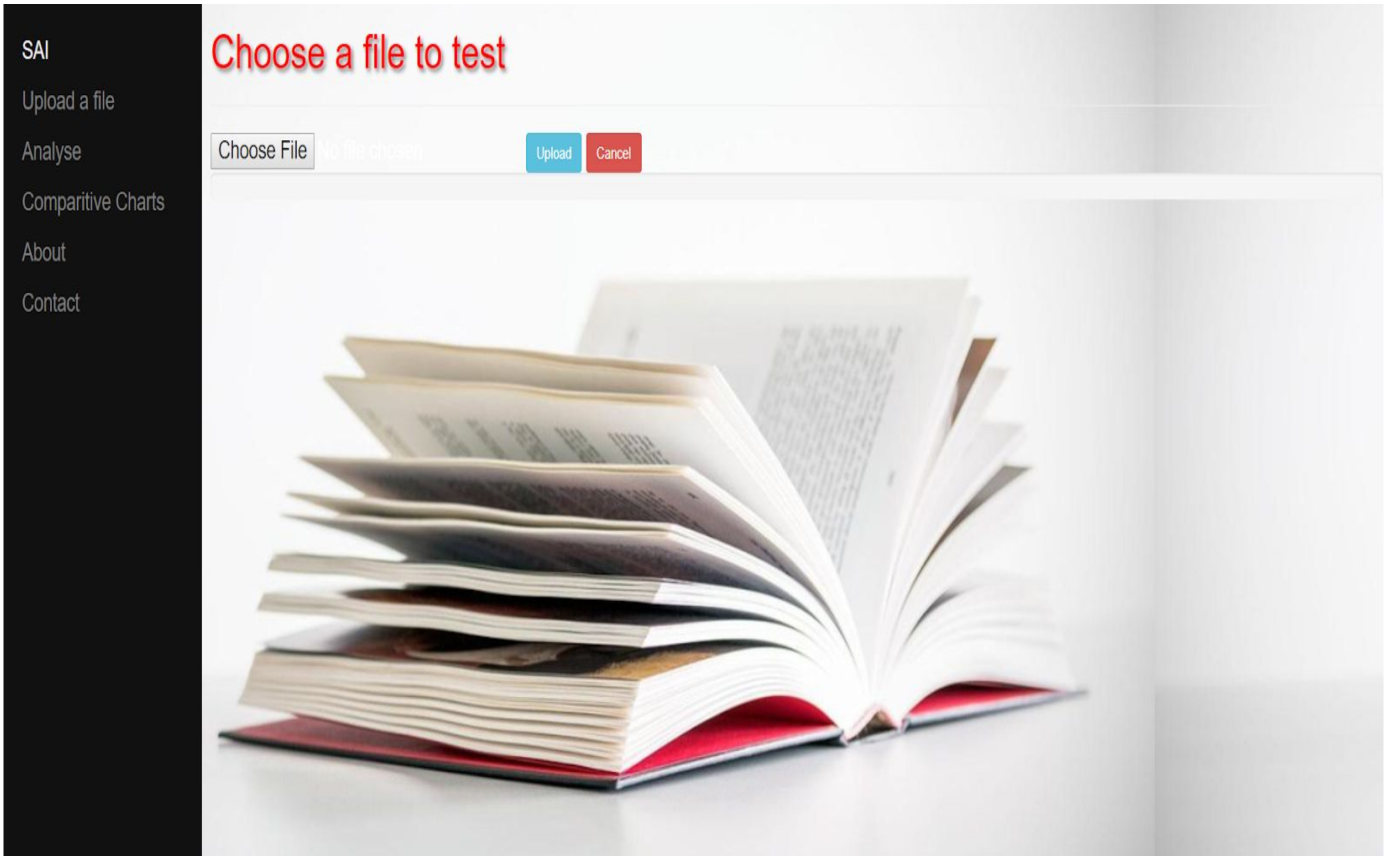


Stylometry Based Authorship Identification

Every author has a inate style of writing, use of particular words, particular punctuations, conjunctions etc. Our system identifies these features and uniquely identifies each author after being trained. This system has a great applicability in cyber forensics and social media.

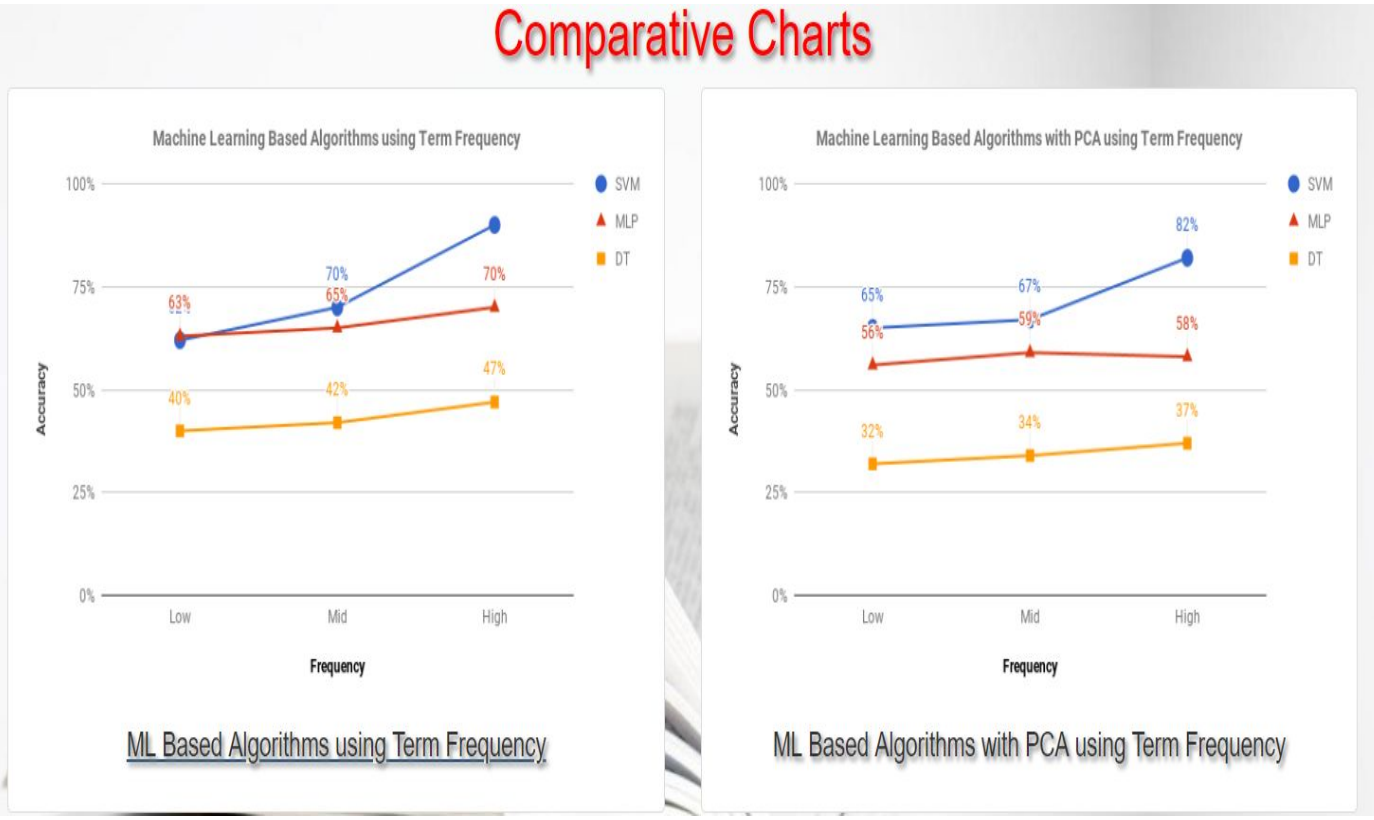
Make sure the system is trained on the author trying to be tested.

Figure 21: Menu



Choose a file to test

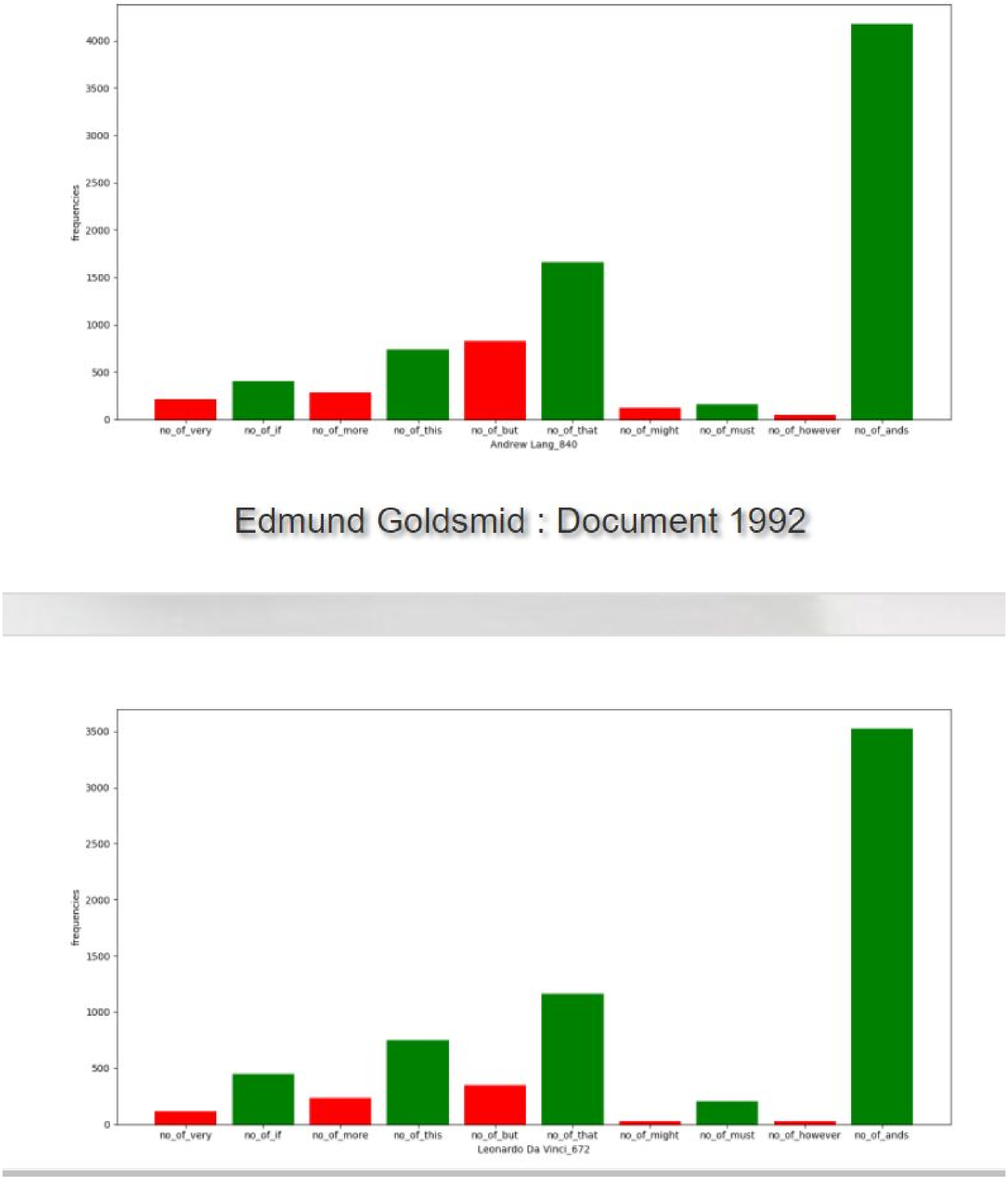
Figure 22: Upload


Comparative Charts

ML Based Algorithms using Term Frequency

ML Based Algorithms with PCA using Term Frequency

Figure 23: Charts (1)



Edmund Goldsmid : Document 1992

Figure 24: Charts (2)

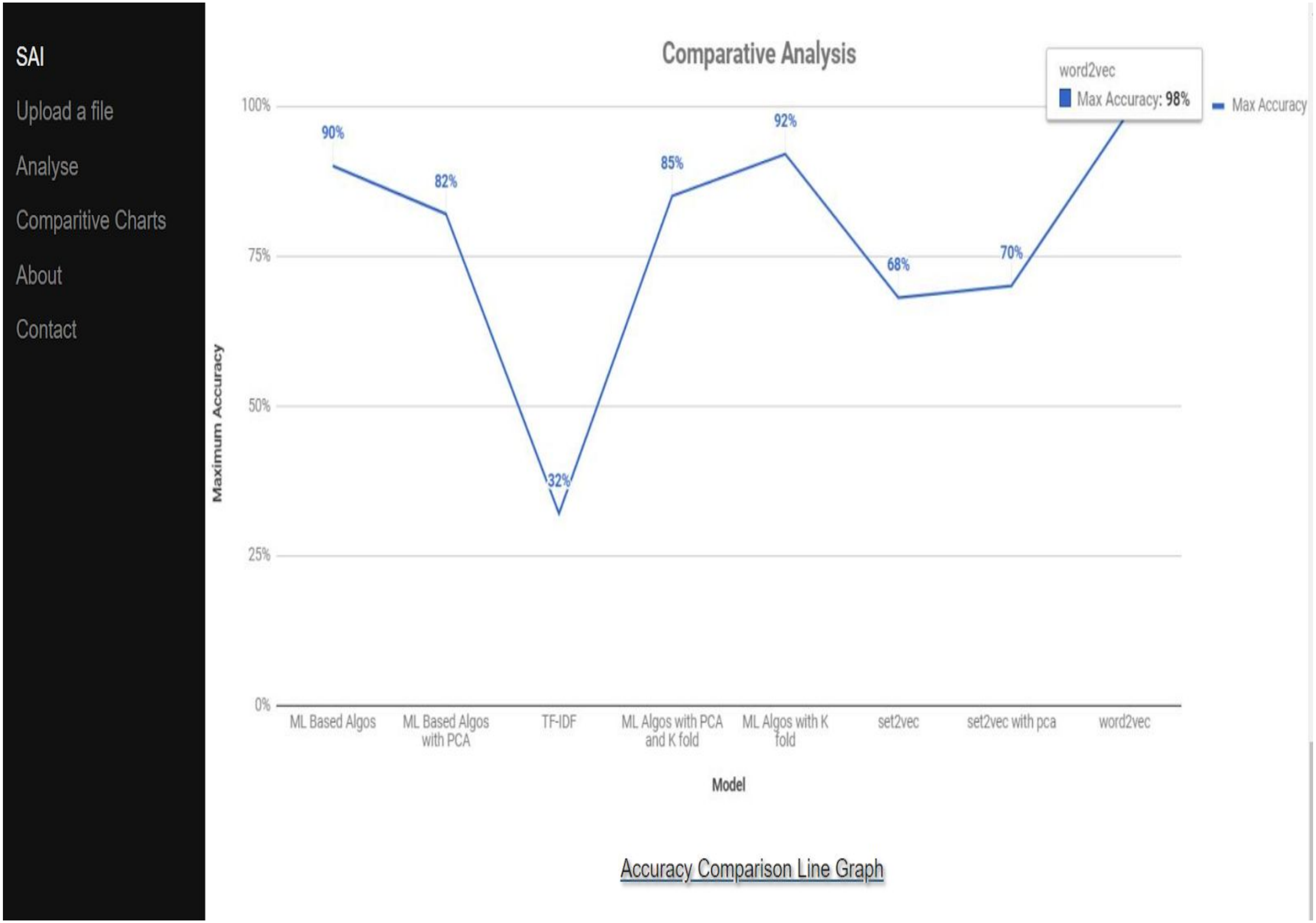
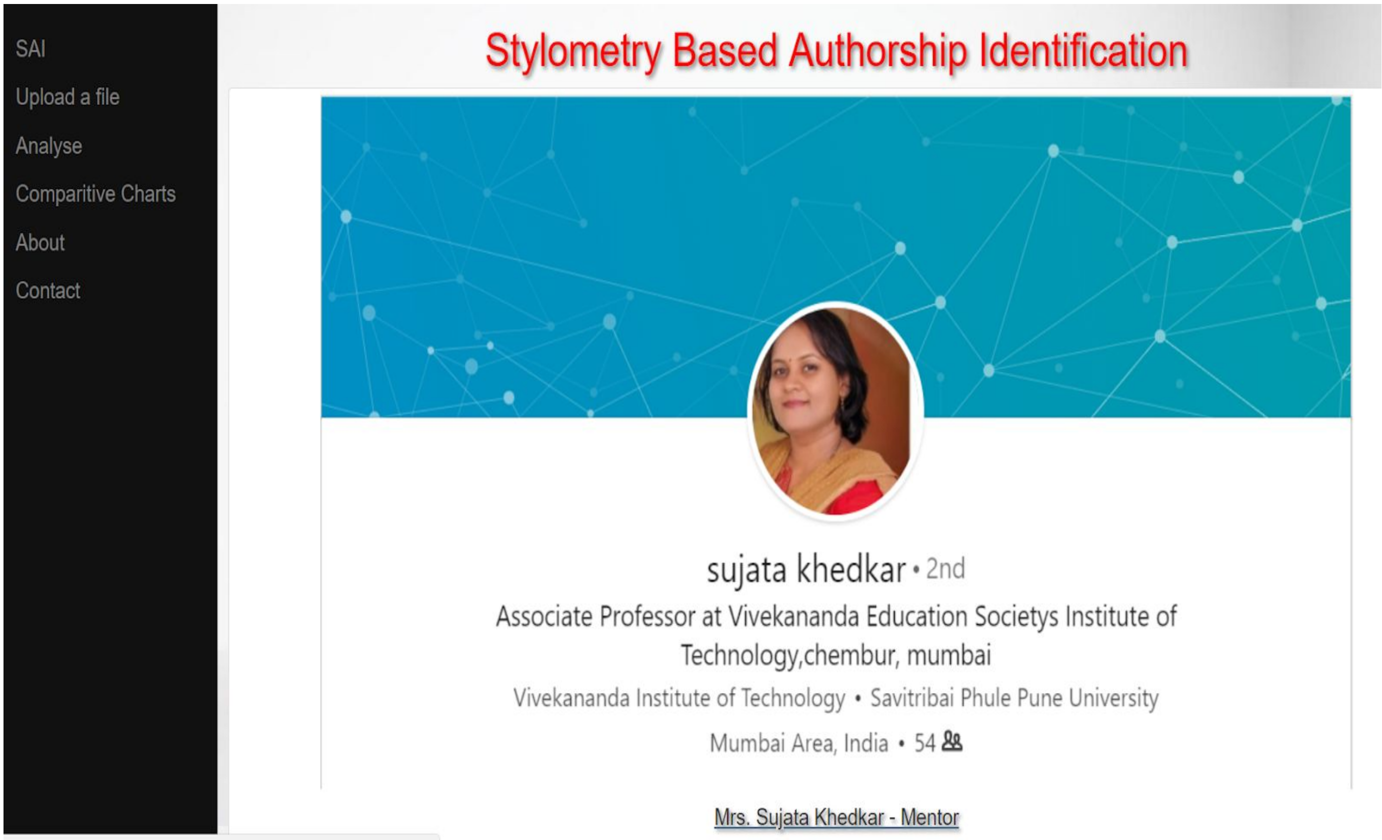


Figure 25: Charts (3)



Stylometry Based Authorship Identification

sujata khedkar • 2nd

Associate Professor at Vivekananda Education Societys Institute of Technology,chembur, mumbai

Vivekananda Institute of Technology • Savitribai Phule Pune University

Mumbai Area, India • 54

Mrs. Sujata Khedkar - Mentor

Figure 26: About Us (1)

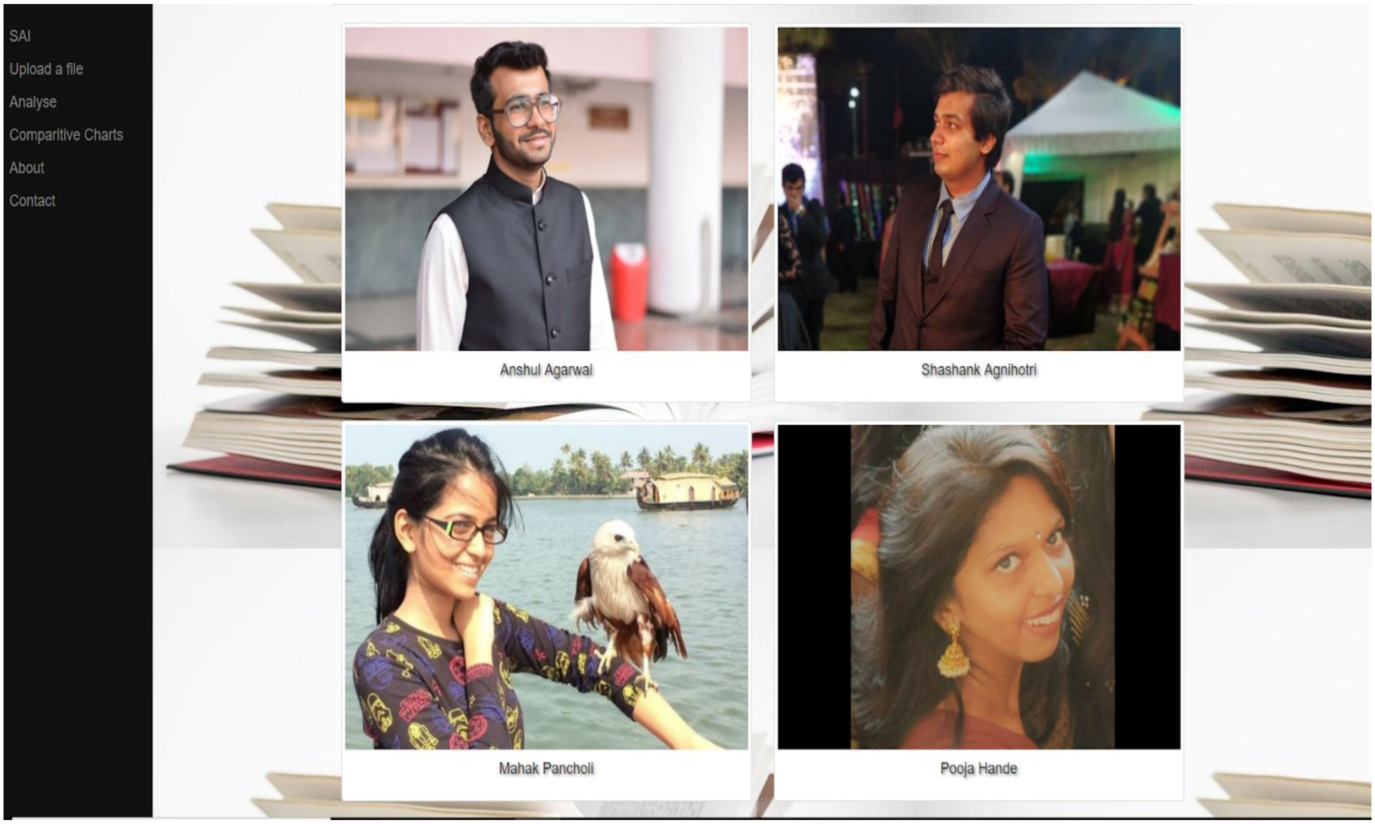


Figure 27: About Us (2)

7.3 Graphical Outputs

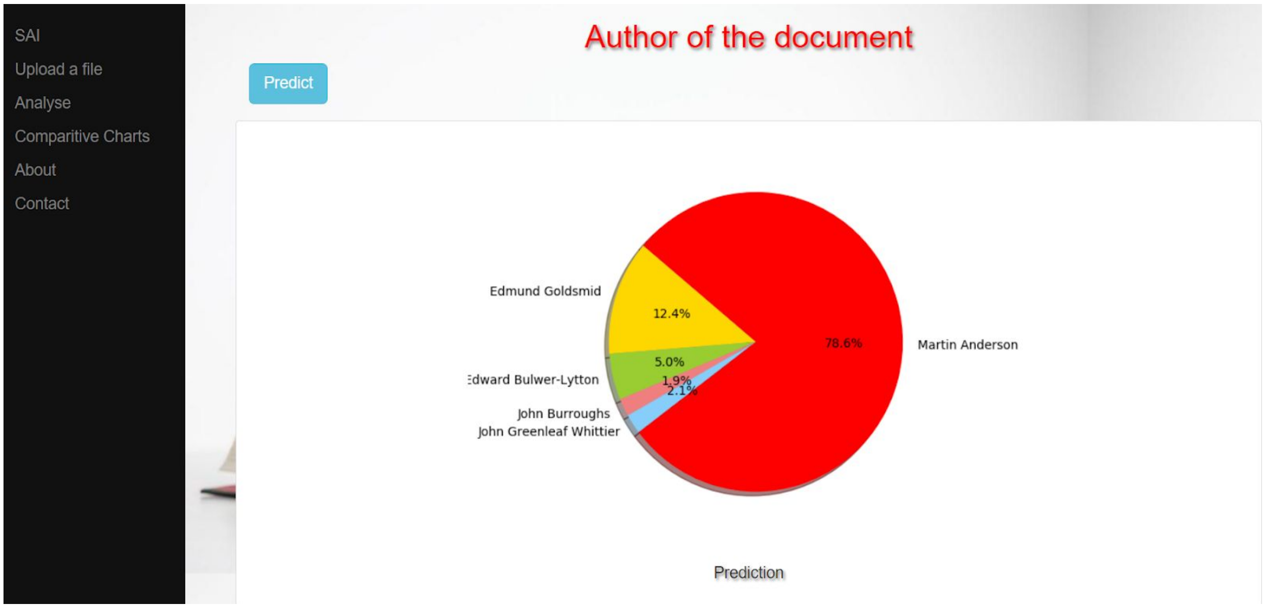


Figure 28: Graphical Output Generated

The output generated by the system will be in the form of a pie chart which will show the probability that the uploaded test document was authored by a particular author. The maximum percentage section in the pie chart is predicted to be the author of the test document uploaded by the user of the system. In the above figure , the system predicts that the test document has a 58.4% chances of being written by Martin Anderson which is greater than any other author. So the system predicted that the author of the test document is 'Martin Anderson'.

7.4 Reports Generated

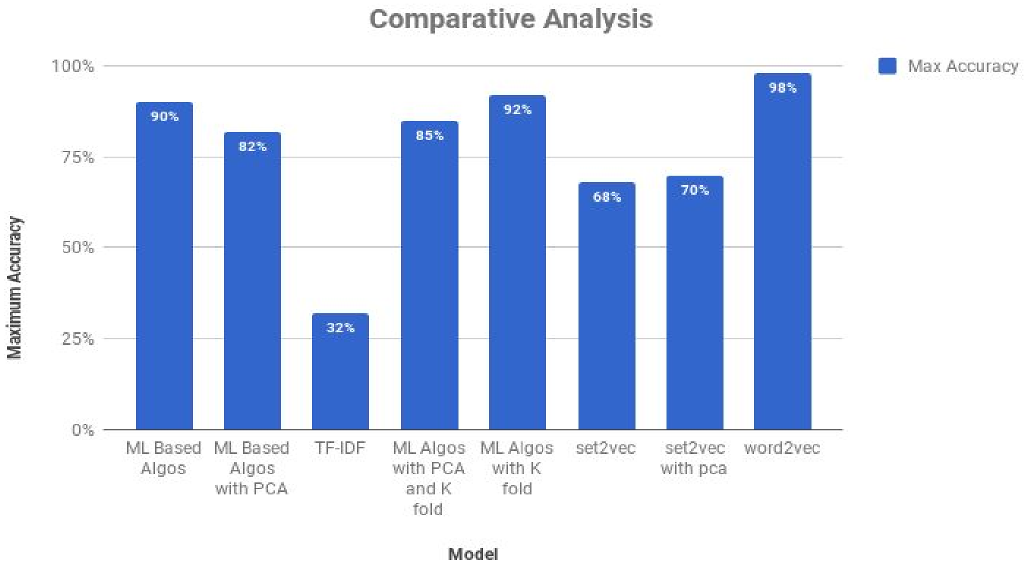


Figure 29: Comparing accuracies of alternative algorithms (Bar)

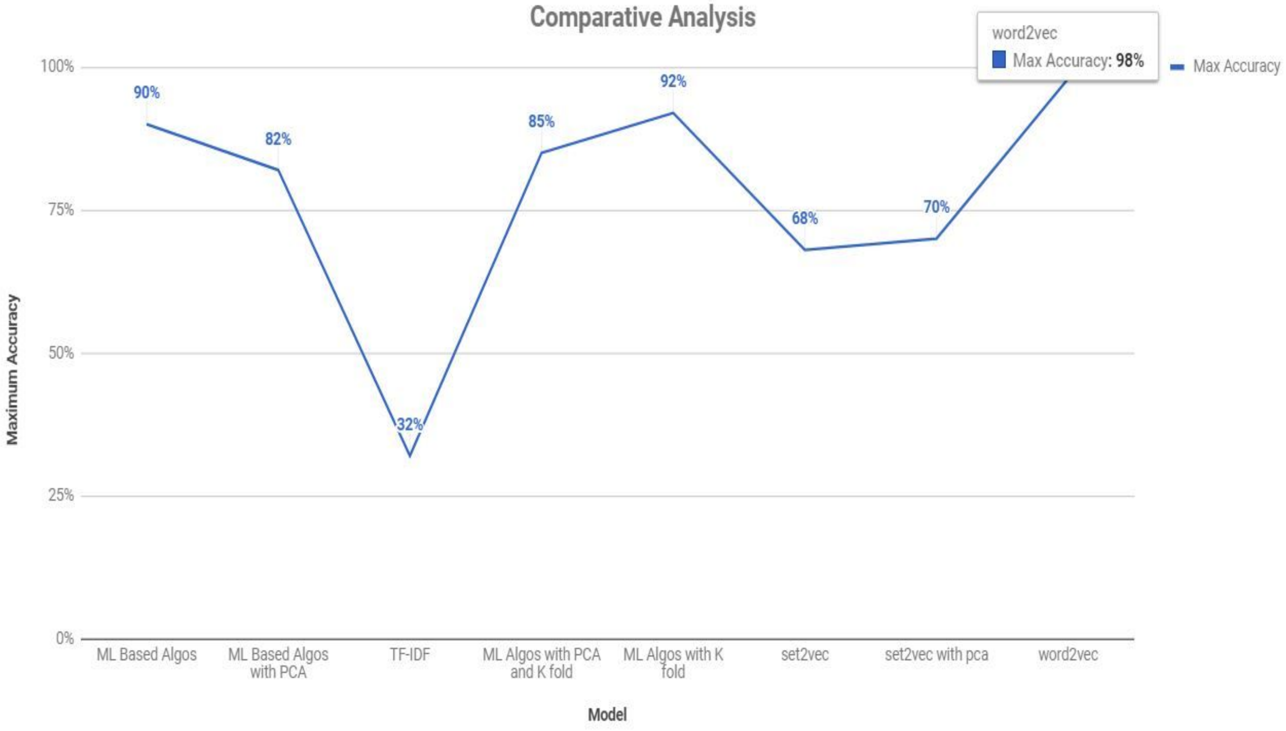


Figure 30: Comparing accuracies of alternative algorithms (Line)

In the figure, accuracy obtained from all the training models is shown in the form of a bar graph and line graph. The accuracies were calculated as the ratio of correct predictions with total predictions. Highest accuracy was obtained using word2vec model.

Author Identification

The author of the test document is given as input to the system. Processing using word2vec takes place at the back end and we get the percentage of probability of the authors that can be the author of the test document. The graph displayed gives us a clear and bird's eye view of the author of the test document.

Chapter 8: Conclusion

This chapter discusses the Limitations and Future Scope of our project. The summarization of the project is also done over here.

8.1 Limitations

Prediction accuracy depends on the data set considered in training the system and the time, space and computational complexities heavily depends on the relationship and mutual importance of every word with respect to the other words in a single document and documents written by other authors.

For fast computation and processing the system should have high computational power which includes a fast RAM (minimum of 8GB), a powerful processor and a high end graphic card.

8.2 Conclusion

Thus this project addresses an old but unsolved problem of accurate and reliable author identification using stylometry. If successful, stylometry would a vital role in cybercrime forensics and would help the world solve ages of mysteries regarding ownership of various writing pieces by authors. It would be used to identify anonymous works and saying by comparing it the the style of authors and famous personalities of those days.

For this very purpose, the project proposed the features to be extracted from the document which hopefully would assist in increasing accuracy and reducing a few redundant dimensions. This project certainly proposes methods that would overcome the known drawbacks of previous works in this field.

8.3 Future Scope

Our system requires many words from each author to train itself to their writing style. The future scope for our project is improving it to predict authors successfully even on a small dataset from each other, like twitter tweets or short online messages and posts. This will enable cyber police in forensics when the case pertains to authenticity or accurately determining the true author of a message or post present online by just studying the writing styles prevalent on the online and social networking portals and those stored in the present models saved after forensics performed on previous cases. Another application can be identification of multiple users as the author of a single document which does not take place in the current model.

References

[1] Krause, Markus, "Stylometry-based Fraud and Plagiarism Detection for Learning at Scale", 2015 5th KSS Workshop, Karlsruhe, Germany

[2] Ángel Hernández-Castañeda, Hiram Calvo , "Author Verification Using a Semantic Space Model.", Computación y Sistemas, Vol. 21, No. 2, 2017, pp. 167–179

[3] J. Hurtado, N. Taweewitchakreeya and X. Zhu, "Who wrote this paper? Learning for authorship de-identification using stylometric features," Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014), Redwood City, CA, 2014, pp. 859-862.

[4] P. Das, R. Tasmim and S. Ismail, "An experimental study of stylometry in Bangla literature," 2015 2nd International Conference on Electrical Information and Communication Technologies (EICT), Khulna, 2015, pp. 575-580.

[5] He, Congzhou & Rasheed, Khaled. (2004). "Using Machine Learning Techniques for Stylometry", Proceedings of the International Conference on Artificial Intelligence, IC-AI '04, Volume 2 & Proceedings of the International Conference on Machine Learning; Models, Technologies & Applications,2004

[6] Marcelo Luiz Brocardo, Issa Traore, Sherif Saad, Isaac Woungang , "Authorship Verification for Short Messages using Stylometry", 2013 International Conference on Computer, Information and Telecommunication Systems, pp. 1-6

[7] Ramnial H., Panchoo S., Pudaruth S, "Authorship Attribution Using Stylometry and Machine Learning Techniques", 2016 Intelligent Systems Technologies and Applications. Advances in Intelligent Systems and Computing, vol 384. Springer, Cham

[8] Maciej Eder, Jan Rybicki and Mike Kestemont , "Stylometry with R: A Package for Computational Text Analysis",The R Journal Vol. 8/1, Aug. 2016

[9] Lakshmi, Pushpendra Kumar Pateriya, "A Study on Author Identification through Stylometry ",Lakshmi et al , International Journal of Computer Science & Communication Networks,Vol 2(6), pp. 653-657

[10] Ganapathi N V Raju, Ch. Sadhvi, P Tejaswini and Y Mounica, "Style based Authorship Attribution on English Editorial Documents", International Journal of Computer Applications 159(4), pp. 5-8, February 2017.

[11] A. Rocha et al., "Authorship Attribution for Social Media Forensics," in IEEE Transactions on Information Forensics and Security, vol. 12, no. 1, pp. 5-33, Jan. 2017.

[12] Helena Ǵ omez-Adorno, Posadas-Duŕ an, Juan-Pablo, Grigori Sidorov, David Pinto," Document Embeddings Learned on Various Types of n-grams for Cross-Topic Authorship Attribution" , in Computing 2018, pp. 1-16.

[13] Đlker Nadi Bozkurt, Özgür Bağlıoğlu, Erkan Uyar, "Authorship Attribution Performance of various features and classification methods", in Proceedings Bozkurt Authorship AP, 2017

[14] Crawford, Michael, et al. "Survey of review spam detection using machine learning techniques." Journal of Big Data 2.1 (2015): 23.

[15] Willett, Peter. "The Porter stemming algorithm: then and now." Program 40.3 (2006): 219-223.

[16] Jolliffe, Ian T. "Principal Component Analysis and Factor Analysis." Principal component analysis. Springer New York, 1986. 115.

Appendix

List of Figures

List of Tables

Paper Publication

Paper 1

Author Identification using Stylometry

Mentor, Mrs. Sujata Khedkar
Associate Professor
Computer Engineering, VESIT,
Chembur
sujata.khedkar@ves.ac.in

Shashank Agnihotri
B.E. Computer Engineering,
VESIT, Chembur
shashank.agnihotri@ves.ac.in

Anshul Agarwal
B.E. Computer Engineering,
VESIT, Chembur
anshul.agarwal@ves.ac.in

Mahak Pancholi
B.E. Computer Engineering,
VESIT, Chembur
mahak.pancholi@ves.ac.in

Pooja Hande
B.E. Computer Engineering,
VESIT, Chembur
pooja.hande@ves.ac.in

***Abstract*—“Every person is unique”, we have been hearing this since ages. Every person has a unique identity, a unique fingerprint, a unique retina and a lot more. These features play a vital role in identification of individuals for security purposes. Unfortunately, when it comes to security of written pieces or words from an individual, these primary unique identities are futile. One cannot identify a writer from a written piece of text on the basis of retina or fingerprint scans, sometimes even the signature can be forged, in such situations for security purposes and intellectual property rights it becomes very important to identify the true author. Stylometry plays an important role in this. Every author has a unique style of writing, measure of this style of writing is called Stylometry. This paper proposes to identify authors from text based on their style of writing. First a data set consisting of articles, short stories and emails will be used to train the system for multiple authors, then a random text would be given to the system to identify the author correctly, if the author predicted by the system is similar to the author claimed then the information is authentic otherwise the author claiming to be the writer is a fraud. For stylometry, over the ages, many features have been focused on, but this paper proposes new features to be used for this purpose. While writing, there are many unconscious styles that are incorporated by the author, these features have been unnoticed till date, but can play a vital role in accurate and fast identification of authors. These features include: ‘intellectual property right’, ‘chapter length’ and frequency of particular words per thousand words. The algorithms used to train the system can be Decision tree, Naive Bayesian or Multilayer Perceptron.**

Keywords—feature extraction, data set, Decision tree, artificial intelligence, machine learning, supervised learning.

I. Introduction

Various attempts have been made to identify author using stylometry. Most of the attempts made use of similar feature extractions but different data sets and algorithms. Every system had a drawback that couldn’t be overlooked. Jose Hurtado, Napat Taweewitchakreeya, and Xingquan Zhu in their paper[1] used multilayer perceptron, random forest, SVM and k-nearest neighbour for training the data. Here the MLP learner, combined with the six categories of stylometric features, provides better performance over other classifiers and baseline approaches however Random forest and k-nearest neighbours give low accuracy and only few authors can be identified accurately. While in [2] Kohonen Self Organising Maps and backpropagation is used which is suitable to capture an intangible concept like style and in this fewer input variables are required as compared to the traditional statistics but this can be implemented only for small number of authors. [3] seems to cover all the drawbacks of [1] and [2] and other related works. [3] uses LDA and Naive Bayes for classification which enables it to do semantic analysis of corpus however it brings in a new drawback with it: to classify a new unknown document it would be necessary to reprocess all documents including new ones, this is an onerous and time consuming task. Thus this paper proposes a new methodology that encompasses almost all the benefits of [1], [2], [3] and [4] as it overcomes their drawbacks. Stemming and Principal Component Analysis will provide a sharp edge in cutting down the processing time while increasing the efficiency of the new proposed system. Moreover focus on a new set of features will provide better accuracy and including a 'Pre-Processing stage' in the system will tremendously decrease the payload on the system when adding new data sets to the already trained system.

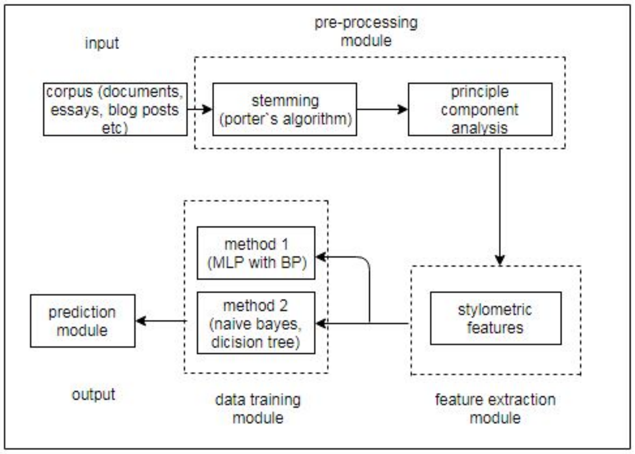

input
pre-processing module
corpus (documents, essays, blog posts etc)
stemming (porter`s algorithm)
principle component analysis
method 1 (MLP with BP)
method 2 (naive bayes, dicision tree)
prediction module
stylometric features
output
data training module
feature extraction module

Fig. 1. Block diagram of the proposed system

II. Preprocessing

A. Stemming

Stemming, a crude heuristic process is used to chop the end of the words in order to achieve the goal correctly and more easily. It focuses on removing the derivational affixes as well.

Porter's Algorithm as mentioned in [6] can be used. 5 phases of word reductions are applied sequentially in Porter's algorithm. Each phase consists of various conventions to select the rules which are suitable.The example of the same can be that a rule can be selected from a particular rule group and hence applying it to the suffix with the largest length.

B. Principal Component Analysis

Principal Component Analysis refers to analysis of data which will be responsible to identify the patterns and then finding the patterns to reduce the dimensions of the dataset drastically, taking into consideration the minimal loss of the information. One of the way for performing Principal Component Analysis is by choosing a subset of Principal Components and Variables as mentioned in section 6 of [7].

III. Feature Extraction

A. Mainstream Methods

1. **type-token ratio:** The type- token ratio denotes the tendency of the author to repeat certain words in a sentence. It is used to define the richness of the vocabulary of the author. Higher type-token ratio denotes varied vocabulary.
2. **mean word length:** Longer words tend to have more formal styles and pedantic as compared to the shorter words which are typically used for informal spoken language.
3. **mean sentence length:** Sentence length is a vital factor wherein the longer sentences are usually associated with careful planned writing, whereas shorter sentences symbolize more of a spoken language characteristic.
4. **standard deviation of sentence length:** It is one of the important marker of style. Standard deviation denotes the variation of a sentence length.
5. **mean paragraph length:** The paragraph length is used to determine the occurrences of dialogues.
6. **number of commas per thousand tokens:** Commas play a crucial role, which denote the ongoing flow of ideas within a sentence.
7. **number of ands per thousand tokens:** Ands are the markers used to represent coordination. It is frequently used in spoken production.
8. **number of buts per thousand tokens:** Buts are the markers of coordination, used to represent the contrastive linking.
9. **vocabulary:** Every authors selected vocabulary was chosen.

B. New Methods

1. **chapter length:** It will denote the length of the sample chapter.
2. **number of colons per thousand tokens:** colons indicate the reluctance of an author to stop a sentence where(s) he could.
3. **number of quotation marks per thousand**

tokens: The frequent use of quotations marks denote the involvement feature.

4. **number of exclamation marks per thousand tokens:** Exclamation mark is a marker of strong emotions.
5. **number of hyphens per thousand tokens**: Hyphens play an important signal as some of the authors use hyphenated words more than other authors.
6. **number of howevers per thousand tokens:** "However" forms a contrastive pair with "but" in a sentence structure which can act as an important marker as well.
7. **number of 'if's per thousand tokens:** Ifs will denote the samples of subordination.
8. **number of 'that's per thousand tokens:** 'That's in a sentence usually denote the subordination and also can be used as demonstratives.
9. **number of 'more's per thousand tokens:** More is used as an indicator of author's liking for a comparative structure.
10. **number of 'must's per thousand tokens:** Modal verbs are potential candidates for expressing tentativeness. Musts are more often used non-epistemically.
11. **number of 'might's per thousand tokens:** 'Might's are a marker which are usually used epistemically.
12. **number of 'this's per thousand tokens:** This is a marker which is used in the case of anaphoric reference.
13. **number of 'very's per thousand tokens:** Verys are significant because of it's emphasis on it's modifies.
14. **part-of-speech tagging (PoS tagging):** Penn Treebank PoS tagging denotes annotations. (ex. CC for coordinating conjunction, SYM for symbol)

IV. Intended Methodology

A. Method 1

Author identification would be done using a mixture of two algorithms. The primary algorithm would be Decision tree but instead of using values of features in the Decision tree, their conditional probabilities would be used. Multinomial Naive Bayes algorithm would be used for this purpose. While the training the system, on the data set, the system will calculate the priors for every tag word (focused features), then on the basis of the documents calculate the conditional probability of these features to the documents. Then on the basis of conditional probabilities the Decision Tree would be formed to classify each document to its respective author, in the process giving the system a measure of style of writing for every author.

Then when an author needs to be identified for a given random document, the priors and conditional probabilities would be calculated using the Multinomial Naive Bayes algorithm for that document and the values of the conditional probabilities would be used as an input in the trained Decision Tree. This would ensure high accuracy and fast computation when compared to that of both algorithms individually.

B. Method 2

Multilayer Perceptron would be used for training the system on the dataset.The Multilayer Perceptron for better accuracy will have backpropagation with it. The features to be focused on by the system, as discussed in section III of this paper, would be the factors of the hidden layers in the Multilevel Perceptron.The target state would be grouping all documents from the same author in one category or class. This would be partially supervised learning. As the system would initially not know the true authors of the documents but just have the documents. On the basis of the features, the documents will pass several layers of the perceptron to be distinguished on the basis of stylometry and then the output would be compared with the desired output. The errors would be back propagated and the weights would be adjusted to finally categorize all the documents correctly under their respective authors. This will give us a unique identity and stylometry for every author. This would be tagged to every author to help us in author identification and verification in case of new documents. Then when a new document would be tested against the MLP the system would already be trained and the weights adjusted, thus the system would accurately be able to identify the true author and plagiarism, if any.

V. Specifications and Tools

A. Hardware Specifications

1. DDR4 RAM

2. 4 GB Graphic card

3. 6th generation intel processor or above

B. Software Specifications

1. Data Processing : Anaconda IPython notebook, NLTK, GraphLab, Matlab, Octaves
2. User Interface : Xampp, Html, CSS, JavaScript, Bootstrap, Php
3. Analytical Tools : Knime, Weka
4. Data Visualization : Tableau,D3.js

C. Tools

1. Jetbrains Webstorm
2. *R Studio*
3. *Jetbrains Pycharm*
4. *Scikit*
5. *nltk library*
6. *MatLab*
7. *Octave*

VI. Results

The system will calculate the chances of each author having the chances of writing the document, and the author which has the highest percentage would be identified by the system to be the true author. If the author claimed and the author identified by the system are same then the claim is validated, if not then the author has falsely claimed to be the author of that document.

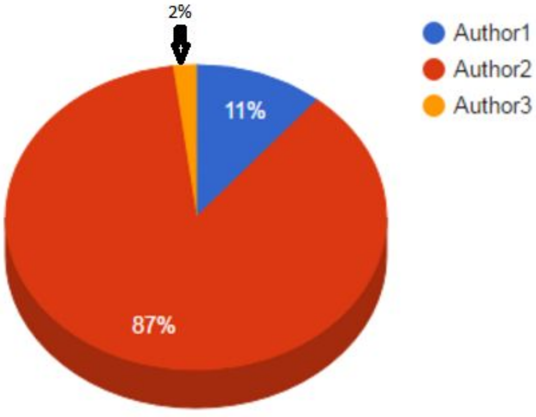


Fig. 2. Sample output

This is how the result will look. Author 1 has 11% chances of being the author of the given document, Author 2 has 87% chances and Author 3 has 2% chances of being the author of the given document. Thus clearly Author 2 is identified by the system as the true author of the document.

VII. Constraints

The techniques used to determine the authors will be requiring relatively large dataset of novels, to train and test our system. Large dataset is required to accurately determine author in a large pool of authors.

The proposed algorithm might not work, with high precision, for a smaller dataset.

VIII. Future Scope

Determining the most favorable algorithm or developing a new algorithm for training the data-set is something in sight as future scope of this project. Moreover the proposed system only takes into account syntax for feature extraction in stylometry, extending the feature extraction to semantic features would develop the system and increase its accuracy many folds. However, mechanisms for this purpose haven't been developed yet, so incorporating this feature in the system remains as a future scope waiting for developments of suitable and feasible mechanisms to make this a reality.

Stylometry will play an important role in identification of potential social media hazards and in cracking cyber crime cases. Being able to incorporate short messages like tweets, Facebook posts or WhatsApp messages to train data and identify the author would be helpful and play an instrumental role in this field. This use of stylometry is also something that has to be left for future development as currently for accurately identifying authors there is substantial amount to written text that is needed to train the system.

IX. Conclusion

Thus this paper addresses an old but unsolved problem of accurate and reliable author identification using stylometry. If successful, stylometry would a vital role in cybercrime forensics and would help the world solve ages of mysteries regarding ownership of various writing pieces by authors. It would be used to identify anonymous works and saying by comparing it the the style of authors and famous personalities of those days.

For this very purpose, the paper proposed new features to be extracted from the document which hopefully would assist in increasing accuracy and reducing a few redundant dimensions. The paper even proposes a new algorithm that could be identified for faster computation and better accuracy. This new proposed algorithm is basically a combination of two well known and used algorithm. This paper certainly proposes methods that would overcome the known drawbacks of previous works in this field.

References

[1] Hurtado, Jose, Napat Taweewitchakreeya, and Xingquan Zhu. "Who wrote this paper? learning for authorship de-identification using stylometric featuress." Information reuse and integration (IRI), 2014 IEEE 15th international conference on. IEEE, 2014.

[2] Ramyaa, Congzhou He, and Khaled Rasheed. "Using machine learning techniques for stylometry." Proceedings of International Conference on Machine Learning. 2004.

[3] Hernández-Castañeda, Ángel, and Hiram Calvo. "Author Verification Using a Semantic Space Model." Computación y Sistemas 21.2 (2017).

[4] Brocardo, Marcelo Luiz, et al. "Authorship verification for short messages using stylometry." Computer, Information and Telecommunication Systems (CITS), 2013 International Conference on. IEEE, 2013.

[5] Crawford, Michael, et al. "Survey of review spam detection using machine learning techniques." Journal of Big Data 2.1 (2015): 23.

[6] Willett, Peter. "The Porter stemming algorithm: then and now." Program 40.3 (2006): 219-223.

[7] Jolliffe, Ian T. "Principal Component Analysis and Factor Analysis." Principal component analysis. Springer New York, 1986. 115-128.

Draft of Paper 2

Stylometry Based Authorship Identification

Mentor, Mrs. Sujata Khedkar
Associate Professor
Computer Engineering, VESIT,
Chembur
sujata.khedkar@ves.ac.in

Shashank Agnihotri
B.E. Computer Engineering, VESIT,
Chembur
shashank.agnihotri@ves.ac.in

Anshul Agarwal
B.E. Computer Engineering, VESIT,
Chembur
anshul.agarwal@ves.ac.in

Mahak Pancholi
B.E. Computer Engineering, VESIT,
Chembur
mahak.pancholi@ves.ac.in

Pooja Hande
B.E. Computer Engineering, VESIT,
Chembur
pooja.hande@ves.ac.in

***Abstract*—"Every person is unique", we have been hearing this since ages. Every person has a unique identity, a unique fingerprint, a unique retina and a lot more. These features play a vital role in identification of individuals for security purposes. Unfortunately, when it comes to security of written pieces or words from an individual, these primary unique identities are futile. One cannot identify a writer from a written piece of text on the basis of retina or fingerprint scans, sometimes even the signature can be forged, in such situations for security purposes and intellectual property rights it becomes very important to identify the true author. Stylometry plays an important role in this. Every author has a unique style of writing, measure of this style of writing is called Stylometry. This paper proposes to identify authors from text based on their style of writing. First a data set consisting of articles, short stories and emails will be used to train the system for multiple authors, then a random text would be given to the system to identify the author correctly, if the author predicted by the system is similar to the author claimed then the information is authentic otherwise the author claiming to be the writer is a fraud. For stylometry, over the ages, many features have been focused on, but this paper proposes new features to be used for this purpose. While writing, there are many unconscious styles that are incorporated by the author, these features have been unnoticed till date, but can play a vital role in accurate and fast identification of authors. These features include: 'intellectual property right', 'chapter length', 'the importance of a word with respect to the other words in a document' and frequency of particular words per thousand words. The algorithms used to train the system can be Decision tree, Naive Bayesian or Multilayer Perceptron.**

Keywords—feature extraction, data set, Decision tree, artificial intelligence, machine learning, supervised learning, word2vec, sentence2vec, doc2vec.

I. Introduction

Various attempts have been made to identify author using stylometry. Most of the attempts made use of similar feature extractions but different data sets and algorithms. Every system had a drawback that couldn't be overlooked. Jose Hurtado, Napat Taweewitchakreeya, and Xingquan Zhu in their paper[1] used multilayer perceptron, random forest, SVM and k-nearest neighbour for training the data. Here the MLP learner, combined with the six categories of stylometric features, provides better performance over other classifiers and baseline approaches however Random forest and k-nearest neighbours give low accuracy and only few authors can be identified accurately. While in [2] Kohonen Self Organising Maps and backpropagation is used which is suitable to capture an intangible concept like style and in this fewer input variables are required as compared to the traditional statistics but this can be implemented only for small number of authors. [3] seems to cover all the drawbacks of [1] and [2] and other related works. [3] uses LDA and Naive Bayes for classification which enables it to do semantic analysis of corpus however it brings in a new drawback with it: to classify a new unknown document it would be necessary to reprocess all documents including new ones, this is an onerous and time consuming task. Thus this paper proposes a new methodology that encompasses almost all the benefits of [1], [2], [3] and [4] as it overcomes their drawbacks. Stemming and Principal Component Analysis will provide a sharp edge in cutting down the processing time while increasing the efficiency of the new proposed system. Moreover focus on a new set of features will provide better accuracy and including a ‘Pre-Processing stage’ in the system will tremendously decrease the payload on the system when adding new data sets to the already trained system.

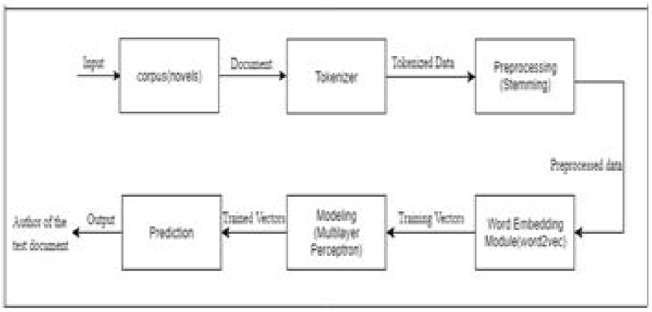


Fig 1. Block Diagram of final developed system

II. Preprocessing

A. Stemming

Stemming refers to a crude heuristic process which is commonly used to chop off the end of the words so as to achieve the desired goal easily and more correctly. It focuses on removing the derivational affixes as well.

Porter’s Algorithm as mentioned in [6] can be used. 5 phases of word reductions are applied sequentially in Porter’s algorithm. Each phase consists of various conventions to select the rules which are suitable.The example of the same can be that a rule can be selected from a particular rule group and hence applying it to the suffix with the largest length.

B. Data Cleansing

Data cleansing which is also known as data cleaning is the process in which we detect and correct the corrupt and records which are inaccurate from a record set, database or some table and then identifies inaccurate, incomplete, irrelevant or incorrect parts of the data and then modifying, deleting or replacing the dirty data. Data cleansing may be performed interactively with data wrangling tools, or as batch processing through scripting.

C. Principal Component Analysis

Principal Component Analysis refers to analysis of data which will be responsible to identify the patterns and then finding the patterns to reduce the dimensions of the dataset drastically, taking into consideration the minimal loss of the information. One of the way for performing Principal Component Analysis is by choosing a subset of Principal Components and Variables as mentioned in section 6 of [7].

III. Feature Extraction

A. Adopted Methods

1. *number of commas per thousand tokens:* Commas play a crucial role, which denote the ongoing flow of ideas within a sentence.

2. *number of ands per thousand tokens:* Ands are the markers used to represent coordination. It is frequently used in spoken production.

3. *number of buts per thousand tokens:* Buts are the markers of coordination, used to represent the contrastive linking.

4. *vocabulary:* Every authors selected vocabulary was chosen.

5. *number of colons per thousand tokens:* Colons indicate the reluctance of an author to stop a sentence where(s) he could.

6. *Frequency of words from bag-of-words:* The frequency of every word used is measured and words and their occurrence counts are mapped into categories of ‘high frequency’, ‘mid frequency’ and ‘low frequency’ for further processing.

7. *part-of-speech tagging (PoS tagging):* Penn Treebank PoS tagging denotes annotations. (ex. CC for coordinating conjunction, SVM for symbol)

8. *Word2vec:* The Authorship Attribution (AA) task consists in identifying the author of a given text among a list of candidates authors. In this approach, the problem is treated as a supervised

classification task, when a classifier is built using a training set and the task consists in classifying correctly the samples from a testing set. Word embeddings after cleaning the training data, we use the Word2vec method to obtain the vectors for each document. The Word2vec module offers two possible approaches to build the model, the Distributed Model (DM), which tries to predict the context of a given element and the Distributed Bag of Words (DBOW), which tries to predict the word given the context.

9. *Tf-idf:* In information retrieval, tf–idf or TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf-idf value increases proportionally to the number of times a word appears in the document and is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. Nowadays, tf-idf is one of the most popular term-weighting schemes; 83% of text-based recommender systems in the domain of digital libraries use tf-idf.

IV. Methodologies adopted

A. Method 1

Algorithm:

1. Perform Stemming on the dataset.
2. Calculate the frequency of only lexical features* of the documents.
3. Divide the frequencies into 3 categories, low frequency, mid frequency (count 500 to 1000) and high frequencies.
4. Split the low, mid and high frequency features table into Training and Testing Data set.
5. Train the system on using Decision Tree Classifier, SVM and Neural Network.
6. Predict the authors for unseen features.

This approach gives us an accuracy of 37% to 90% but the system is not dynamic, and only lexical features are considered here which is not ideal.

B. Method 2

Algorithm:

1. Perform Stemming on the dataset.
2. Calculate the frequencies of lexical features of documents and bag-of-words.
3. Like Module 1 divide the frequencies into 3 categories, low frequency, mid frequency and high frequencies.
4. Perform Principal Component Analysis on mid frequency document.
5. Split the low, mid and high frequency features table into Training and Testing Data set.
6. Train the system on using Decision Tree Classifier, SVM and Neural Network.Predict the authors for unseen features.

This module gave us a low accuracy as compared to model 1 as PCA wiped out essential features being used in prediction. This approach gives us an accuracy of 30% to 82%. Moreover, here too only lexical features were being considered.

C. Method 3

Algorithm:

1. Perform stemming on the entire dataset
2. Calculate the “Term Frequency–Inverse Document Frequency” i.e. tf-idf score for the stemmed dataset.
3. Perform Principal Component Analysis (PCA) on the result of step 2.
4. Split the result table of step 3 into Training and Testing dataset.
5. Train the system using Decision Tree Classifier, SVM and MLP.
6. Predict the authors for unseen feature vectors.

This module gave us a very low accuracy of 10% to 32% because tf-idf score is not a very suitable approach for our dataset, which are large documents from many different authors which the number of documents per author varying a lot. Moreover, the system was static, that is to test or train the system on a new file or author, the entire system had to be run again.

D. Method 4

Algorithm:

1. Perform Stemming on the dataset.
2. Calculate the frequencies of lexical features of the documents and bag-of-words.
3. Like Module 1 divide the frequencies into 3 categories, low frequency, mid frequency and high frequencies.
4. Perform Principal Component Analysis on mid frequency document.
5. Split the low, mid and high frequency features table into Training and Testing Data set.
6. Train the system using Decision Tree Classifier, SVM and Neural Network.
7. Perform k-fold cross validation on the dataset(k=10).
8. Predict the authors for unseen features.
9. Calculate the accuracy, the mean of the accuracy and standard deviation of the accuracy.

This module gave us a accuracy in the range of 31% to 85%. There was a bit drop in accuracy due to PCA.

E. Method 5

Algorithm:

1. Perform Stemming on the dataset.
2. Calculate the frequencies of lexical features of the documents and bag-of-words.
3. Like Module 1 divide the frequencies into 3 categories, low frequency, mid frequency and high frequencies.
4. Split the low, mid and high frequency features table into Training and Testing Data set.
5. Train the system using Decision Tree Classifier, SVM and Neural Network.
6. Perform k-fold cross validation on the dataset.
7. Predict the authors for unseen features.
8. Calculate the accuracy, the mean of the accuracy and standard deviation of the accuracy.

The only difference between Module 5 and Module 4 is performing PCA, however this small change had a huge impact of almost 7% - 8% on the accuracy of the system, Module 5 gave us an accuracy of around 41% to 92%.

F. Method 6

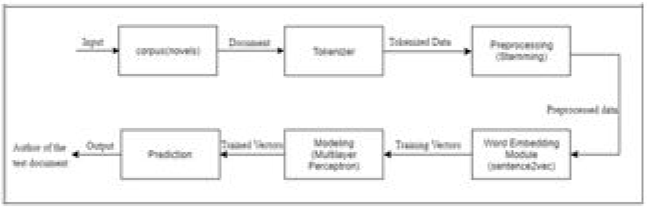


Fig 2: Model 6

Algorithm:

1. Tokenize dataset and perform stemming.
2. Perform Data cleansing and cleaning.
3. Perform sentence2vector operations on the cleaned dataset.
4. Obtain vectors for each sentence in a document.
5. Train model using these vectors and machine learning algorithms such as neural networks and SVM.
6. Test the trained model.

Accuracy by this model on a 5 author dataset trained using MLP is 68% and on a 15 author dataset is 49%.

G. Method 7

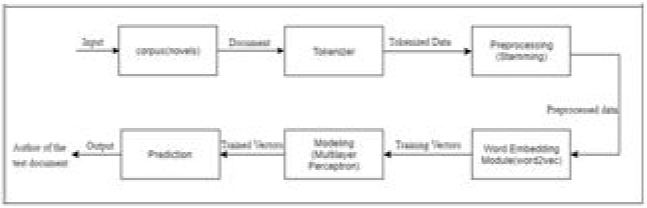


Fig 3: Model 7

Algorithm:

1. Tokenize dataset and perform stemming.
2. Obtain word2vec embeddings for each document.
3. Train model using these vectors and machine learning algorithms such as neural networks and SVM.
4. Test the trained model.

Accuracy by this model on a 15 author dataset trained using MLP is 83% . On 5 authors accuracy is 98.11%.

V. RESULTS

The system will calculate the chances of each author having the chances of writing the document, and the author which has the highest percentage would be identified by the system to be the true author. If the author claimed and the author identified by the system are same then the claim is validated, if not then the author has falsely claimed to be the author of that document.

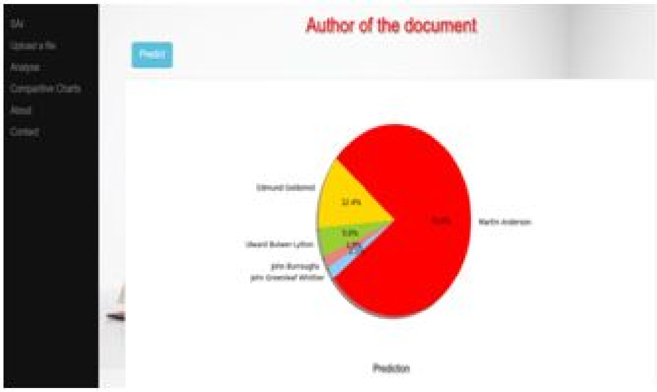


Fig 4. Sample Output

This is how the result will look. There is a 78% probability that Martin was the author of the document given to the system for prediction, however there is a 32% probability that Edmund was the author too, while 5% probability of Edward being the author and some more smaller probabilities of some other known authors on which the system is trained to be the author of the document being tested.

As the probability of Martin being the true author of the system is the highest, based on the stylometry based tests, Martin is identified as the one true author of the document.

VI. COMPARISONS

Comparing the various built and tested models and their accuracies for the different machine learning algorithms the system was trained on. The primary machine learning algorithms used were SVM, Neural Network – Multilayer Perceptron (MLP) with 32 hidden layers and Decision Tree.

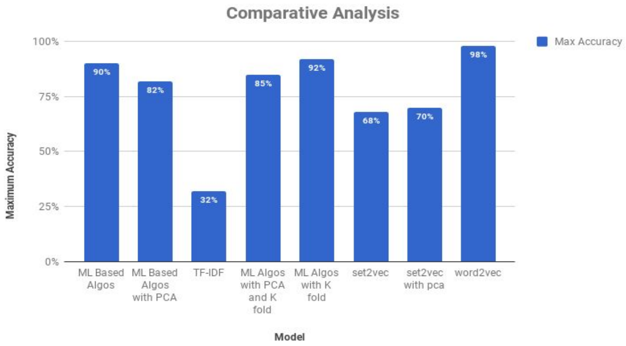


Fig 5. Sample Output

Parameters of SVM: (C=1.5, Degree=3, Kernel=poly)
Parameters of MLP: (hidden_layer_sizes (32,32) , alpha=0.001)
Type of Decision Tree: (CART). CART is an abbreviation for Classification & Regression Trees.

A. Model 1: Word Frequency based model without PCA

Algorithm	**Low Frequency**	**Mid Frequency**	**High Frequency**
SVM	57-62%	65-70%	**85-90%**
MLP	60-63%	60-65%	60-70%
Decision Tree	37-40%	37-42%	43-47%

Table 1: Comparisons of Model 1

B. Model 2: Word Frequency based model with PCA

Algorithm	**Low Frequency**	**Mid Frequency**	**High Frequency**
SVM	60-65%	64-67%	**77-82%**
MLP	53-56%	57-59%	54-58%
Decision Tree	30-32%	32-34%	34-37%

Table 2: Comparisons of Model 2

C. Model 3: ML Based Model using TF-IDF

Algorithm	Accuracy
SVM	22-32%
MLP	8-10%
Decision Tree	3-10%

Table 3: Comparisons of Model 3

D. Model 4: Word Frequency based model with PCA and k-fold cross validation

Algorithm	Low Frequency	Mid Frequency	High Frequency
SVM	60-66%	65-68%	**84-85%**
MLP	56-58%	70-73%	68-70%
Decision Tree	31-34%	35-37%	40-43%

Table 4: Comparisons of Model 4

E. Model 5: Word Frequency based model without PCA and k-fold cross validation.

Algorithm	Low Frequency	Mid Frequency	High Frequency
SVM	60-62%	75-77%	**90-92%**
MLP	66-68%	75-77%	67-75%
Decision Tree	41-45%	44-55%	47-52%

Table 5: Comparisons of Model 5

F. Model 6: sentence2vector

Algorithm	Accuracy
SVM	33%(5 authors) , 15%(15 authors)
MLP	**68%**(5 authors) , 49%(15 authors)
Decision Tree	41%(5 authors) , 22%(15 authors)

Table 6: Comparisons of Model 6

G. Model 7: word2vector

Algorithm	Accuracy
SVM	65%(5 authors),
MLP	**98.11%** (5 authors), 84%(15 authors)
Decision Tree	97.45%(5 authors), 45%(15 authors)

Table 7: Comparisons of Model 7

On comparing the results obtained by testing the above models it becomes very important to identify a single algorithm or model to adapt for the final Stylometry based Authorship Identification System being built.

Thus the maximum accuracies obtained from each developed model were collected and represented in graphical forms for better and clear understanding of the ideal model to be finally adopted. A close study of the accuracies revealed that the 'word2vec' model provided the highest accuracy to the system under development and thus after passing further tests developed using specific test cases, 'word2vec' was finally chosen.

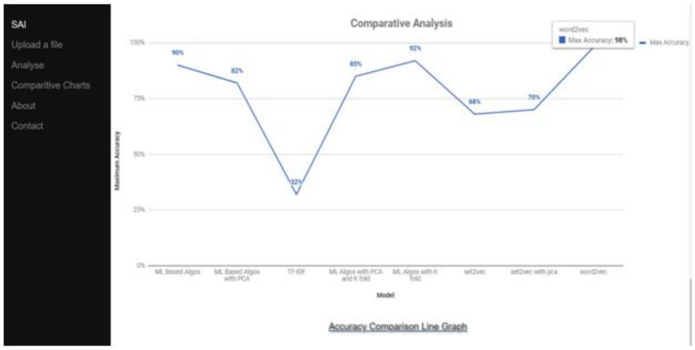


Fig.6 Accuracy comparison line graph

VII. CONSTRAINTS

With increasing communication and interaction between people, there are times that a single piece of document has not been written by a single author but has multiple authors, in such a case, the developed system fails to identify the multiple authors of the document and identifies the single largest probabilistic author as the true author of the document.

VIII. FUTURE SCOPE

Stylometry will play an important role in identification of potential social media hazards and in cracking cyber crime cases. Being able to incorporate short messages like tweets, Facebook posts or WhatsApp messages to train data and identify the author would be helpful and play an instrumental role in this field. This use of stylometry is also something that has to be left for future development as currently for accurately identifying authors there is substantial amount to written text that is needed to train the system.

Moreover, surpassing the constraint in the current system, of identifying documents written by not one, but multiple authors is another important aspect that can be added to the future scope of this developed system.

IX. CONCLUSION

Thus this paper addresses an old but unsolved problem of accurate and reliable author identification using stylometry. If successful, stylometry would a vital role in cybercrime forensics and would help the world solve ages of mysteries regarding ownership of various writing pieces by authors. It would be used to identify anonymous works and saying by comparing it the the style of authors and famous personalities of those days.

For this very purpose, the paper proposed new features to be extracted from the document which hopefully would assist in increasing accuracy and reducing a few redundant dimensions. The paper even proposes a new algorithm that could be identified for faster computation and better accuracy. This new proposed algorithm is basically a combination of two well known and used algorithm. This paper certainly proposes methods that would overcome the known drawbacks of previous works in this field.

REFERENCES

[1] Hurtado, Jose, Napat Taweewitchakreeya, and Xingquan Zhu. "Who wrote this paper? learning for authorship de-identification using stylometric features." Information reuse and integration (IRI), 2014 IEEE 15th international conference on. IEEE, 2014.

[2] Ramyaa, Congzhou He, and Khaled Rasheed. "Using machine learning techniques for stylometry." Proceedings of International Conference on Machine Learning. 2004.

[3] Hernández-Castañeda, Ángel, and Hiram Calvo. "Author Verification Using a Semantic Space Model." Computación y Sistemas 21.2 (2017).

[4] Brocardo, Marcelo Luiz, et al. "Authorship verification for short messages using stylometry." Computer, Information and Telecommunication Systems (CITS), 2013 International Conference on. IEEE, 2013.

[5] Crawford, Michael, et al. "Survey of review spam detection using machine learning techniques." Journal of Big Data 2.1 (2015): 23.

[6] Willett, Peter. "The Porter stemming algorithm: then and now." Program 40.3 (2006): 219-223.

[7] Jolliffe, Ian T. "Principal Component Analysis and Factor Analysis." Principal component analysis. Springer New York, 1986. 115.

[8] Krause, Markus, "Stylometry-based Fraud and Plagiarism Detection for Learning at Scale", 2015 5th KSS Workshop, Karlsruhe, Germany

[9] P. Das, R. Tasmim and S. Ismail, "An experimental study of stylometry in Bangla literature," 2015 2nd International Conference on Electrical Information and Communication Technologies (EICT), Khulna, 2015, pp. 575-580.

[10] Ramnial H., Panchoo S., Pudaruth S, "Authorship Attribution Using Stylometry and Machine Learning Techniques", 2016 Intelligent Systems Technologies and Applications. Advances in Intelligent Systems and Computing, vol 384. Springer, Cham

[11] Maciej Eder, Jan Rybicki and Mike Kestemont , "Stylometry with R: A Package for Computational Text Analysis",The R Journal Vol. 8/1, Aug. 2016

[12] Lakshmi, Pushpendra Kumar Pateriya, "A Study on Author Identification through Stylometry ",Lakshmi et al , International Journal of Computer Science & Communication Networks,Vol 2(6), pp. 653-657

[13] Ganapathi N V Raju, Ch. Sadhvi, P Tejaswini and Y Mounica, "Style based Authorship Attribution on English Editorial Documents", International Journal of Computer Applications 159(4), pp. 5-8, February 2017.

[14] A. Rocha et al., "Authorship Attribution for Social Media Forensics," in IEEE Transactions on Information Forensics and Security, vol. 12, no. 1, pp. 5-33, Jan. 2017.

[15] Helena G ́omez-Adorno, Posadas-Dur ́an, Juan-Pablo, Grigori Sidorov, David Pinto," Document Embeddings Learned on Various Types of n-grams for Cross-Topic Authorship Attribution" , in Computing 2018, pp. 1-16.

[16] Đlker Nadi Bozkurt, Özgür Bağlıoğlu, Erkan Uyar, "Authorship Attribution Performance of various features and classification methods", in Proceedings Bozkurt Authorship AP, 2017.