A MINI PROJECT REPORT

On

AUTHOR IDENTIFICATION OF HORROR NOVELS

Submitted in partial fulfillment of the requirement of
University of Mumbai for the Course

Natural Language Processing
In
Computer Engineering (VIII SEM)

Submitted By
Nikita Bhalekar(15202015)
Siddhant Bhadsavle(16102038)
Shilpa Chandra(16102004)

Subject Incharge
Prof. Mayuri Jain

Department Of Computer Engineering
A. P. SHAH INSTITUTE OF TECHNOLOGY
THANE – 400 615
UNIVERSITY OF MUMBAI

Academic Year 2019 – 20

Department of Computer Engineering
A. P. Shah Institute of Technology
Thane – 400 615

CERTIFICATE

This is to certify that the requirements for the project report entitled '**Author Identification of Horror Novels**' have been successfully completed by the following students:

Name	**Roll No.**
Nikita Bhalekar	4
Siddhant Bhadsavle	5
Shilpa Chandra	8

in partial fulfillment of the course Natural Language Processing in Computer Engineering (VIII SEM) of Mumbai University in the Department of Computer Engineering, A. P. Shah Institute of Technology during the Academic Year 2019 – 20.

External Examiner

_____ _____

(Prof. Mayuri Jain)
Subject Incharge

Date:

Place: Thane

Department of Computer Engineering
A. P. Shah Institute of Technology
Thane – 400 615

PROJECT APPROVAL

This project entitled "Author Identification of Horror Novels" by Nikita Bhalekar, Siddhant Bhadsavle and Shilpa Chandra are approved for the course Natural Language Processing in Computer Engineering (VIII sem) of Mumbai University in the Department of Computer Engineering.

Subject Incharge:

Prof. Mayuri Jain

_____

Date:

Place: Thane

Department of Computer Engineering
A. P. Shah Institute of Technology
Thane - 400 615

DECLARATION

We declare that this written submission for Natural Language Processing mini project entitled "Author Identification of Horror Novels" 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 ideas / data / fact / source in our submission. We understand that any violation of the above will cause disciplinary action by the institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission has not
been taken when needed.

Project Group Members:

Nikita Bhalekar & Sign:
_____

Siddhant Bhadsavle & Sign:
_____

Shilpa Chandra & Sign:
_____

_____

Date:

Place:

Table of Contents

Abstract

Author Profiling is a task in Natural Language Processing which aims to predict authors based on their specific profile characteristics by analyzing their written documents. Nowadays, its relevance has been highlighted thanks to several applications in computer forensics, security and marketing. Each author can have their own style of writing, such as some author will use a specific set of words most of the time in their work, making their style of writing unique. The process of author profiling basically includes three steps like identifying specific features to be extracted from the text, building an standard representation (for example bag-of-words model) for the target profile, building a classification model using a standard classifier for the target profile respectively. In this project the task will be to identify the authors based on horror novel snippets. The three authors taken into consideration are Edgar Allan Poe, HP Lovecraft and Mary Shelley in the corpus.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Fundamentals

Python programming language is required to be used while implementing this project because the need of using NLTK (Natural Language Took Kit) and Scikit-Learn appears.

In this proposed system the dataset used is contains only one attribute that is text snippet of the three authors considered and the prediction column which contains the author's name.

For implementing this project the basics of Natural Language Processing are to be known such as Lemmatisation, Removal of Stop Words, Bag of Words and Classifiers mostly that deal with frequency distribution like Naive Bayes.

- Lemmatisation means reducing the inflectional forms of each word into a common base or root. It takes into consideration the morphological analysis of the words. To do so, it is necessary to have detailed dictionaries which the algorithm can look through to link the form back to its lemma.

- Stop Words are usually articles (a, an, the), prepositions (in, on, under, …) and other frequently occurring words that do not provide any key or necessary information. We would not want these words to take up space in our database, or taking up valuable processing time. For this, we can remove them easily, by storing a list of words that you consider to stop words. NLTK(Natural Language Toolkit) in python has a list of stopwords stored in 16 different languages.

- Bag of Words is a method to extract features from text documents. These features can be used for training machine learning algorithms. It creates a vocabulary of all the unique words occurring in all the documents in the training set. The basic steps included are shown in Fig 1.1. Generated vectors are then input to the classifier algorithm.

Fig 1.1: Process of bag-of-words

- For Classifiers first the raw text data will be transformed into feature vectors and new features will be created using the existing dataset. For this purpose CountVectorizer will be used which is a matrix notation of the dataset in which every row represents a document from the corpus, every column represents a term from the corpus, and every cell represents the frequency count of a particular term in a particular document. Then finally the Multinomial Naive Bayes Algorithm (Classifier) will be used to predict the Author.

1.2 Objectives

The objective of this project is as follows:

1. To understand the implementation of text classification in python.
2. To understand how does text is extracted from corpus and used as features for classification algorithm.
3. To understand how does CountVectorizer Work and is different from TF-IDF Vectorizer.
4. To conclude that which author has the most unique style of writing out of the three authors by creating a Classification Report.
5. To correctly map the text-snippets with author based on user inputs.

1.3 Scope

The scope of the project is to correctly identify the author based on his/her famous text-snippets from their books(horror novels only) and to determine which author has has the most unique style of writing.

1.4 Organization of the Report

The report is organized as follows: The introduction is given in Chapter 1. It describes the fundamental terms used in this project. It motivates to study and understand the different techniques used in this work. This chapter also presents the outline of the objective of the report. The Chapter 2 describes the review of the relevant various techniques in the literature systems. It describes the pros and cons of each technique. The Chapter 3 presents the Theory and proposed work. It describes the major approaches used in this work. The societal and technical applications are mentioned in Chapter 4. The summary of the report is presented in Chapter 5.

Chapter 2

Literature Survey

2.1 Introduction

The papers we have used have helped in this project to find a way out about the work that has been done. The ideas have been referenced from the papers and have been modified according to the purpose of this project.

2.2 Literature Review

1. H. Ayral and S. Yavuz, "An automated domain specific stop word generation method for natural language text classification," 2011 International Symposium on Innovations in Intelligent Systems and Applications, Istanbul, 2011, pp. 500-503.

From this paper the technique of automatically generating stop words has been studied. Furthermore this paper has implemented bayesian natural language classifier whereas in our paper Multinomial Naive Bayes Classifier has been implemented. Bag of words model is used to test the generated words in this paper but in our proposed system the bag of words model is used to generate word vectors which are then given as input to the Classifier.

2. Barathi Ganesh H B, Reshma U and Anand Kumar M, "Author identification based on word distribution in word space," 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, 2015, pp. 1519-1523.

From this paper unigram and bigram features of word space are studied and are taken into consideration but it is very difficult to implement using NLTK as the functions have been changed from time to time. The authors have approached the author profiling using models like support vector machine whereas in our project we have implemented Multinomial Naive Bayes Classifier.

2.3 Literature Summary

SN	Techniques	Author & Year of Publication	Advantages and Disadvantages
1.	Stop words, bag of words, classifiers	H. Ayral and S. Yavuz, 2011	Advantage: The advantage is being able to test the stop words for its appropriateness. Disadvantage: There is no use of wasting computational time to test stop words rather stop words should be completely removed from the queries for faster process of classification.

2.	uni/bi grams, classifiers	Barathi Ganesh H B, Reshma U and Anand Kumar M, 2015	Advantages: Similar features are tested using uni/bi grams which helps when the nltk corpus for the same data is available. Disadvantage: The uni/bi grams hardly works well with the nltk libraries now after the recent updates. Hence we have used bag of words and countverctorizer to find the similar features of a particular author

Table 2.3 : Literature survey summary

Chapter 3

Implementation Details

3.1 Overview

The text-snippets from the dataset are first prepared for the classification algorithm. For that the dataset is first loaded using pandas and then the text processing steps are as follows:

- Removal of any punctuation
- Lemmatisation
- Removal of stop words
- Label encoding of classes is done for example each author gets a numeric number assciated to it to fasten the computation
- Feature engineering is done using Bag of Words
- And then the model is finally trained using MultinomialNB
- Performance of the model is also checked using Classification Report

Once the model is ready the deployment of the model is done using Flask Framework. Flask is a micro web framework in python it is highly popular to deploy machine learning models on web. Once the deployment is done the user can eneter the text-snippets of any author out of the three and can get the prdiction which author's text-snippet was it.

3.1.1 Existing Methodology and Systems

The same set of features such as style of writing of a particular author by suing certain set of words was usually been done by N-grams. The prediction would then been done suing a support vector machine.

3.1.2 Proposed Methodology and System

In this proposed system the certain set of words of a particular author are identified using Bag of Words. This extract features from the text-snippets and creates a vocabulary of all the unique words occurring in them. These feat are then converted to vector using CountVectorizer. These vectors are then in turn given as input to the Classifier algorithms.

3.2 Implementation Details

The model is first trained on the Jupyter Notebook and the perfomance analysis is done using

- Training Accuracy
- Validation Accuracy
- Precision, Recall and F1–Score

The model is then deployed to the web (done using flask) runs on http://127.0.0.1:5000/ on the local machine. Once deployed the user can enter the text by any of the three authors mentioned in the corpus. And when the user is done with the text he/she clicks on the identify the author part. Once this button gets clicked. The flask home page redirects it to the http://127.0.0.1:5000/predict page. When the server enters this page it collects the text that user had entered and then gives it as input to the @app.route('/predict', methods=['POST']). This route contains the function where all the text processing, classification done is explained in 3.2.1.

3.2.1 Methodology

1. The project is first implemented using Jupyter Notebook on a local machine. The dataset was loaded using Pandas.

2. Text Processing is done using three techniques:

    - Removal of Punctuation : All the punctuation marks are removed from all the text-snippets (attribute) from the dataset i.e., the corpus.
    - Lemmatisation : Inflected forms of a word are known as lemma. So, the lemma of a word are grouped under the single root word. This is done to make the vocabulary of words in the corpus contain distinct words only.
    - Removal of Stopwords : Stop-words are usually articles (a, an, the), prepositions (in, on, under, …) and other frequently occurring words that do not provide any necessary information to the meaning of the sentence. They are removed from all the text-snippets present in the corpus.

3. Assigning Numeric Value to Classes:

    For the classification, the classes are the three authors as mentioned. But in the dataset, it can be seen that labels are non-numeric (MWS, EAP and HPL short for Mary Shelley, Edgar Allan Poe and HP Lovecraft respectively). These are label encoded to make them numeric, starting from 0 depicting each label in the alphabetic order i.e., (0 for EAP, 1 for HPL and 2 for MWS).

4. Word Cloud Visualization:

    For the classification algorithm to work each author has his own style of writing such as using a certain set of words repeatedly. A visualization of the mostly-used words to the least-used words of the three authors is done.

5. Bag-of-words:

Bag of Words is a method to extract features from text documents. These features can be used for training machine learning algorithms. It creates a vocabulary of all the unique words occurring in corpus. For this CountVectorizer is been used i.e., counting word occurrence. The reason behind of using this approach is that a keyword or important word will occur again and again. So if the number of occurrence represent the importance of word. More frequency means more importance.

6. Training the Model:

Multinomial Naive Bayes Algorithm (Classifier) has been used. First, it calculates the fraction of documents in each class: i.e., Probability Distribution over Vocabulary. Then Probability of each word per class is calculated. For calculating this probability, we will find the average of each word for a given class. For class j and word i, the average is given by:

$$P(i|j)= \frac{\text{wordij}}{\text{wordj}}$$

Combining probability distribution of P with fraction of documents belonging to each class. The vocabulary dataframe is created which basically gets the counts of each word in the vocabulary. Combining probability distribution of P with fraction of documents belonging to each class is the last step of Multinomial Naive Bayes.

7. Model Performance Analysis:

- Training Accuracy
- Validation Accuracy
- Precision, Recall and F1–Score

8. Flask:

For deploying the model on web so that user can access it.

3.2.2 Details of packages, data set

Packages used are numpy, pandas, nltk, flask, sklearn.
From nltk two main packages have been imported such as stopwords(to remove stop words), WordNetLemmatizer(to. Find the inflected forms of the words).
From sklearn some important pacakges that have been imported are CountVectorizer(to convert the keywords into vectors), train_test_split(to spilt the corpus into testing and training), MultinomialNB(for classification the input here are those vectors).

Chapter 4

Project Inputs and Outputs

4.1 Inputs Details

The dataset is taken from kaggle which is an online community of data scientists and machine learning practitioners which was freely available.

The dataset contains the attributes text and id. Only text is used as the attribute in the project which are the text-snippets of the three authors mentioned.

The authors is the class column which are MWS, EAP and HPL short for Mary Shelley, Edgar Allan Poe and HP Lovecraft.

The text is the text is then processed for classification.

4.2 Evaluation Parameters Details

In this project we have used model.score to determine the accuracy of the model.

In multilabel classification like ours, this function computes subset accuracy: the set of labels predicted for a sample must (of training data) match the corresponding set of labels in training data.

We have also used classification report to display the precision, recall, F1, and support scores for the model.

Precision is the ability of a classifier not to label an instance positive that is actually negative. For each class it is defined as as the ratio of true positives to the sum of true and false positives. Said another way, “for all instances classified positive, what percent was correct?”

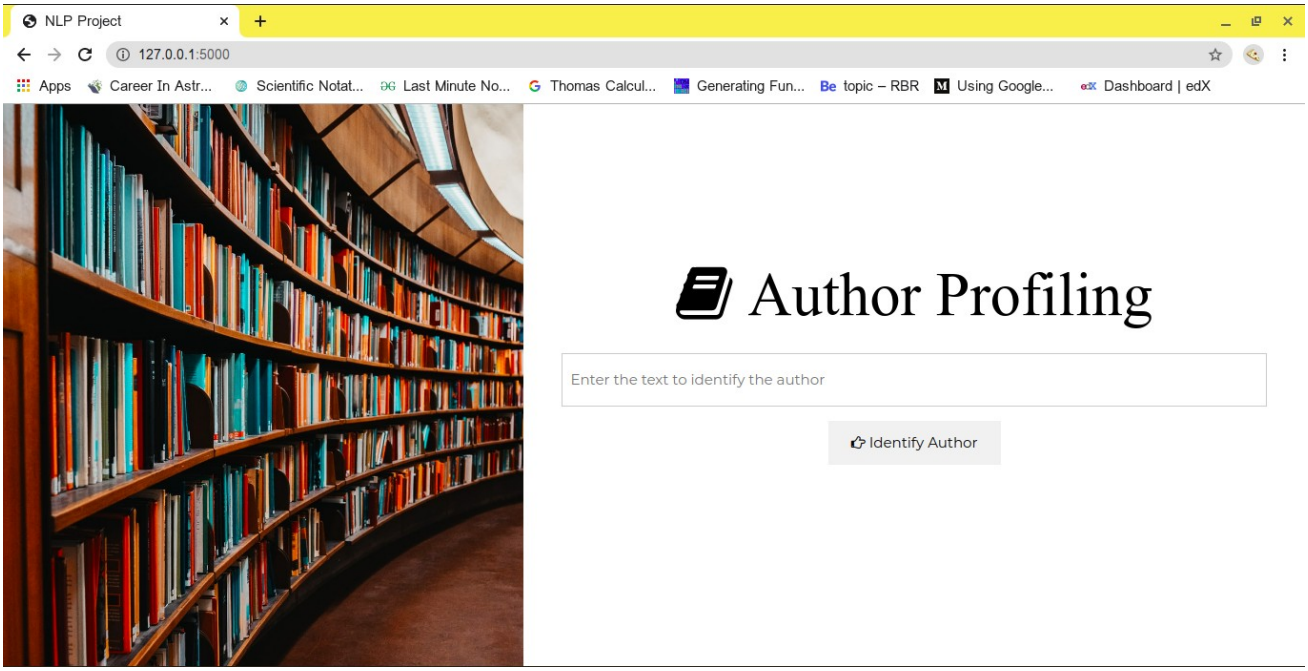
Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives. Said another way, “for all instances that were actually positive, what percent was classified correctly?”

The F1-score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. Generally speaking, F1-scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy.

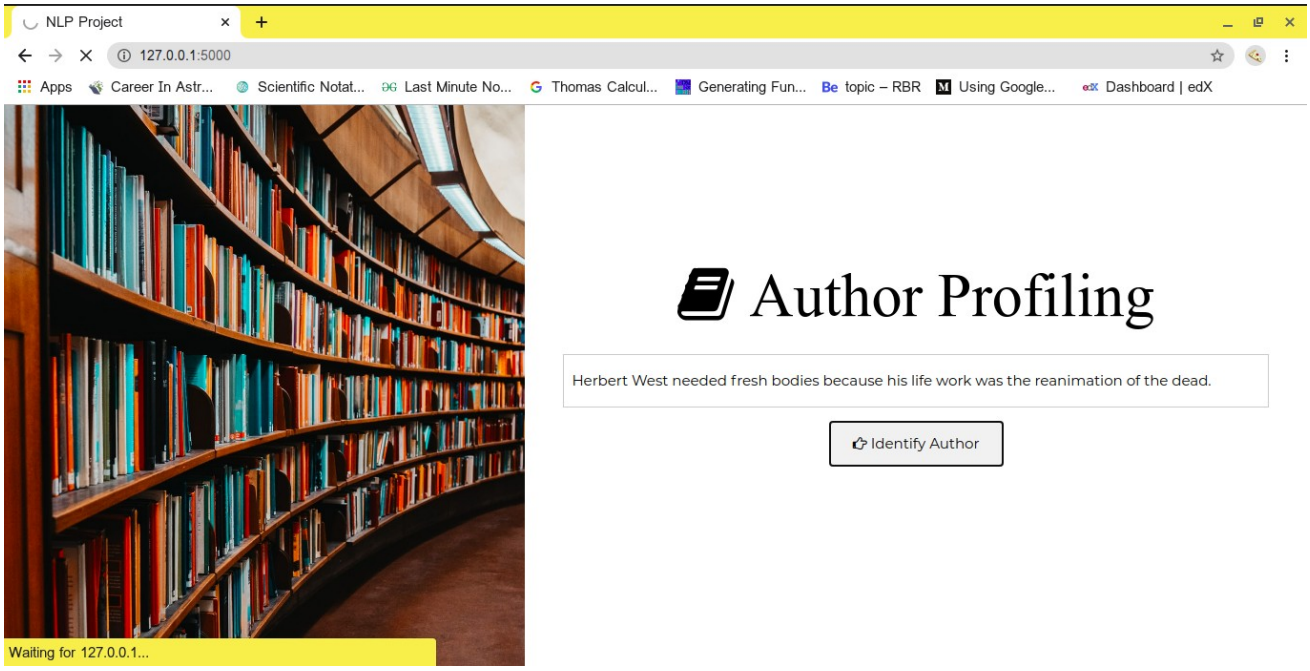
Support is the number of actual occurrences of the class in the specified dataset. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the classifier and could indicate the need for stratified sampling or rebalancing. Support doesn’t change between models but instead diagnoses the evaluation process.

Further explanation of each of these paramaters related to the project is explained in 4.3.

4.3 Output Details and Screenshots

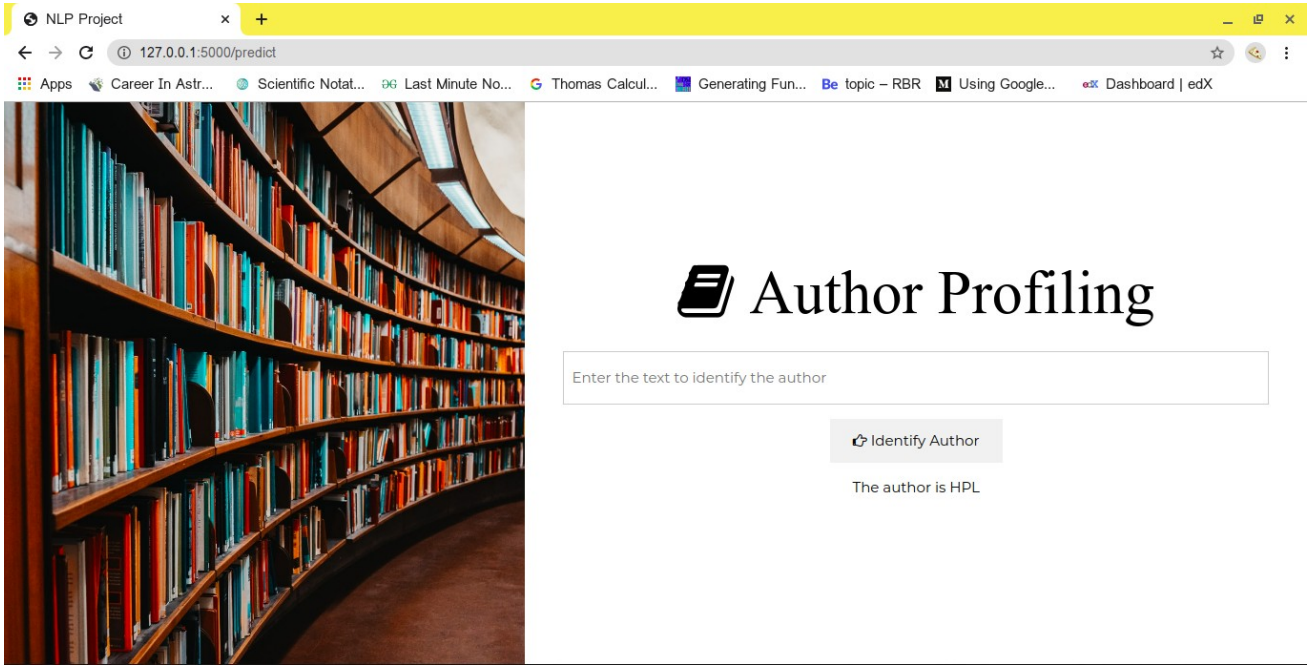


This is the home page of the project. The model has been deployed on the web by flask.



The user then enters the text-snippet of his/her desired author to be identified (out of the three authors in our corpus)

On Click Identify Author the result appears as seen below:



The author here identified is HPL i.e., HP Lovecraft which is correct.

Now let's look at the evaluation process.

```
In [8]:
1 from sklearn.naive_bayes import MultinomialNB
2 model = MultinomialNB()
3 model = model.fit(text_bow_train, y_train)
4

In [9]:
1 model.score(text_bow_train, y_train)

Out[9]: 0.9074889867841409
```

Here the accuracy of the model can be seen which is 90.74% i.e., the percentage of set of labels predicted for a sample must (of training data) match the corresponding set of labels in training data.

Next the classification report has been generated which display the precision, recall, F1, and support scores for the model.
Precision is the ability of a classifier not to label an instance positive that is actually negative.
Recall is the ability of a classifier to find all positive instances.
The F1-score is a weighted harmonic mean of precision.
Support is the number of actual occurrences of the class in the specified dataset.
These parameters has the following values in our project as seen below:

In [13]:

```
from sklearn.metrics import classification_report
predictions = model.predict(text_bow_test)
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       0.85      0.82      0.84      1562
           1       0.87      0.83      0.85      1149
           2       0.81      0.87      0.84      1205

    accuracy                           0.84      3916
   macro avg       0.84      0.84      0.84      3916
weighted avg       0.84      0.84      0.84      3916
```

From the image we can conclude that Author 2 i.e., in our dataset Mary Shelley has the most unique style of writing as :
The recall value of Author 2 is the most i.e., 0.87. Recall means all instances that were actually positive, what percent was classified correctly and that value is greatest for Author 2. Which means Author 2 has been correctly classified more than other two Authors.
Hence we can conclude that Author 2 which is Mary Shelley has the most unique style of writing as this Author's writing can more accurately be classified by our model.

Chapter 5

Summary and Future Scope

5.1 Summary

The proposed system has been successfully been implemented by us. We have taken input from the user and classified it correctly by analyzing and processing the style of writing of the three authors that we have considered.

Our system also tells us which Author out of the three has the most unique style of writing which in this case is Mary Shelley as this Author has been most correctly classified by our model than the others.

5.2 Future Scope

In this proposed system we have only taken into consideration the three Authors from only Horror genre. In future we can take more Author from various genres. Make analysis as to which genres have the most unique style of writing.

References

[1] Barathi Ganesh H B, Reshma U and Anand Kumar M, "Author identification based on word distribution in word space," 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, 2015, pp. 1519-1523.

[2] H. Ayral and S. Yavuz, "An automated domain specific stop word generation method for natural language text classification," 2011 International Symposium on Innovations in Intelligent Systems and Applications, Istanbul, 2011, pp. 500-503.

Acknowledgment

We have great pleasure in presenting the report on **AUTHOR IDENTIFICATION OF HORROR NOVELS.**

We take this opportunity to express our sincere thanks towards teacher Prof. Mayuri Jain, Department of Computer Engineering, APSIT Thane for providing the technical guidelines and suggestions regarding line of work.

We would like to express our gratitude towards her constant encouragement, support and guidance through the development of project.

We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Nikita Bhalekar:

15202015

Siddhant Bhadsavle:

16102038

Shilpa Chandra:

16102004