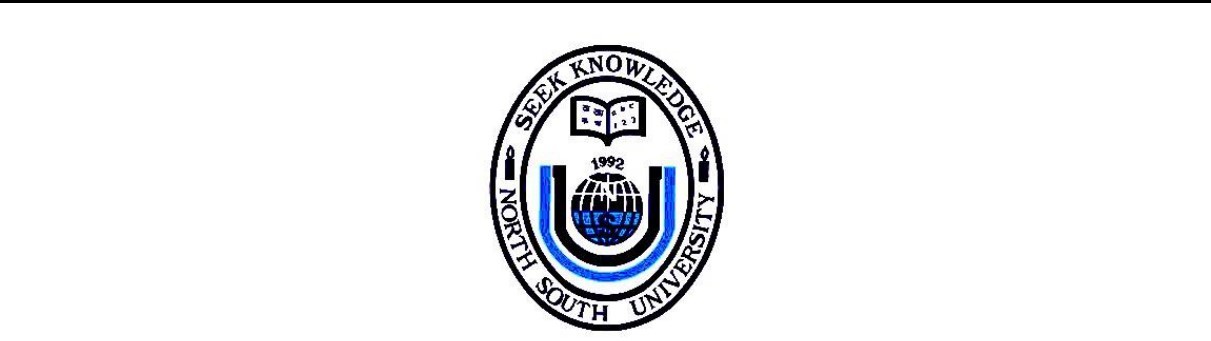
**Department of Electrical and Computer Engineering North South University**



**Author Detection using Neural Networks**

**Name: MD. Ashiqul Amin ID: 1520349642**

**Name: MD. Abdullah Al Waqib ID: 1621493042**

**Faculty Advisor:**

**Dr. Mohammad Ashrafuzzaman Khan**

**Assistant professor**

**Department of ECE**

**Spring 2021**

Table of Contents

Chapter 1: Introduction

Chapter 2: Literature Review

Chapter 3: Methodology

Chapter 4: Results

Chapter 5: Conclusion

Chapter 1: Introduction

Author detection has been a very important and practical problem in Natural Language Processing. The problem is to identify the author of a document from a given list of possible authors. A large amount of work exists on this problem in literature.

We develop ideas based on this work in order to build our own model for author detection. It has become even more important with the widespread use of various forums where anonymity is allowed. A good system for author detection can handle cases of misuse of anonymity. Such a system can also be used for plagiarism detection. Analysis of author of historical texts has already been done many times using techniques for author attribution. Various spoofing e-mails can also be detected by using these techniques.

We have used the spooky author detection dataset which was easily available at Kaggle. Author detection from text is a significant domain of research in natural language processing. It can be used in books retrieval from a website or from a public domain. For example, a user can search an author by using a part of text or some group of words that an author might frequently use. We have built a model that we automatically detect an author by looking at the text. The model doesn’t take much time to train as it uses less parameters and converges very quickly. The model also predicts correctly 7 times out of 10.

Chapter 2: Literature Review

A text can be used for classification by gleaning information from its content and style. In our approach, we used similar techniques using NLP tools to correctly classify a text written by an author. There are many papers from which, we could draw information for our research but we would like to present some in the related works. Firstly, Diri and Amasyali presented a paper in 2003 where they built a model to correctly classify from a dataset comprising text from 18 different authors. The authors used Turkish texts and data processing is used to avoid non-unique content in the texts for example: punctuation, stop words etc.

The authors have also used Style markers to help in the classification that is; they have marked the part of the speech according to the category of grammar the word is.

Secondly, Stamatatos, Fakotakis and Kokkinakis proposed a paper where the text is categorized in terms of author and genre. They have taken full advantage of the NLP tools that are available. In the end, they propose a set of style markers including analysis-level measures that represent the way in which the input text has been analyzed and capture useful stylistic information without additional cost.

Chapter 3: Methodology

This chapter comprises the workflow of our research. It shades light on the dataset used, techniques of data processing, visualization and finally preparing the data for input to a machine learning model.

**Dataset**

The dataset contains text from the fictional work written by authors: Edgar Allan Poe, Hp Lovecraft and Mary Shelley. The dataset contains three columns which are as follows: id, text and author. The "id" field is provided to uniquely identify each row in the dataset as shown in Figure 1. On the other hand, the "text" column comprises the long chunk of text taken from the fictional work of the authors and lastly, "author" column provides the label of each row. By looking at the dataset, we gleaned that features are represented as text and the corresponding label is the author name. Prior to model selection and prediction, we have done some preprocessing on the text so that we only feed in the significant features into the model. The data processing is completed in mainly three step which are punctuation removal, lemmatization and stop words removal.

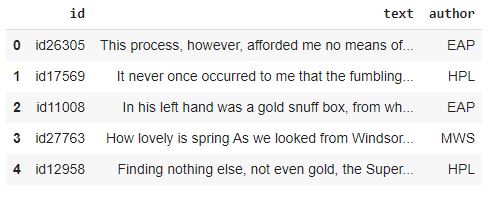


Figure 1: A sample of the Dataset used

**Data Preprocessing:**

In this subsection, we discuss about the steps taken for data preprocessing which is the preliminary step to every machine learning research.

**Removing Punctuation:**

The text features are processed to remove all the punctuations since a model cannot interpret anything by looking at the punctuation and it will not add to the performance of our model. Therefore, texts are preprocessed for lemmatization.

**Lemmatization:**

Transformation of word into its lemma is a important step in natural language processing. The reason is an English word might have different types of form, to illustrate that, study can be written as studying and studied. Therefore, with lemmatization, the word is transformed into its most basic form so that the word with a same meaning across different tense is considered as a single feature for the model.

**Stop words removal:**

Before training a model, stop words are removed because they can occur in ample and they provide no uniqueness to the text therefore, using these words will increase the size of the vocabulary leading to more computational power and providing no improvement to the model whatsoever.

**Bag of Words:**

After removal of punctuation, stopwords and transforming the words into its lemma. We covert the long string of text into a list of binary numbers. This is done by calculating the size of vocabulary which represent the total number of unique words present in the dataset. In our research, there are 22275 unique words in the dataset. Each corresponding text from an author is converted into a list of length 22275 which is filled with 0s and 1s. If a particular word from the vocabulary is present in the text, then we place 1 in the exact same index number respect to the index the word in placed in the dictionary. Likewise, 0 is placed if the word is not present in the text. Finally, the text is made ready for usage as an input into our model but the author column is in string data type. Therefore, we used LabelEncoder class to transform the author label into a corresponding number as our model will only understand numerical value.

**CHAPTER 4: Results**

After 100 epochs, the model reached a training accuracy of 99.9% and a testing accuracy of 77.1% which is an optimal texting accuracy given that we are using bag of words for the prediction. The intuition behind the models learning is that an individual author has different styles of writing and using words. Therefore, some authors find use the word “love” more often than others whereas other authors might use “hate” more frequently. This is how the model converges based on the cross-entropy loss function we used in our model. The word cloud plotted using the text database of each other is shown in Figure 2, 3, 4 below. By observing the figure, we can clearly state that each author has used certain words different times. The size of the words increases with respect to the time it is being used by the author.

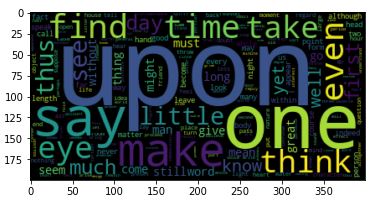


Figure 2: Word cloud of Edgar Allan Poe

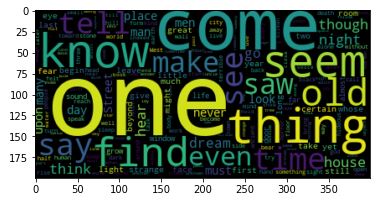


Figure 3: Word cloud of Mary Shelley

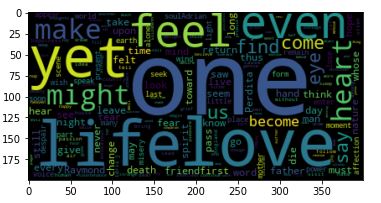
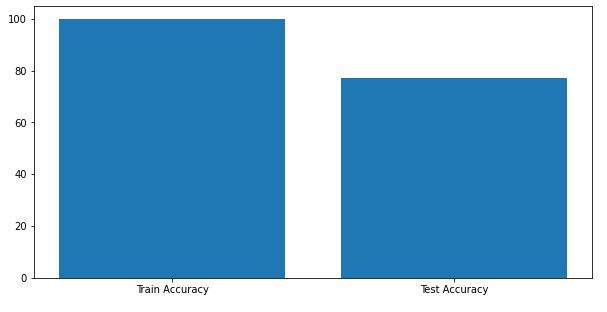
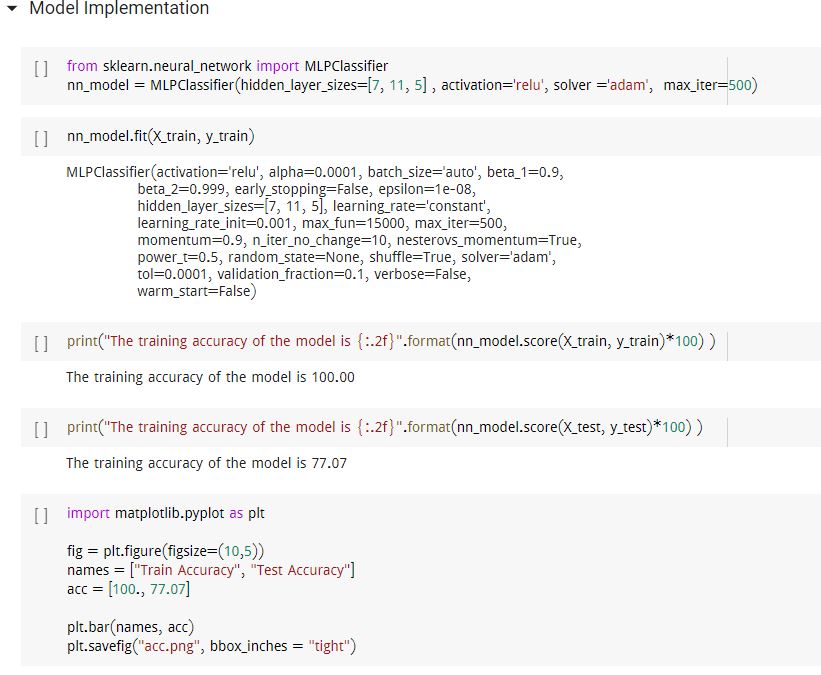
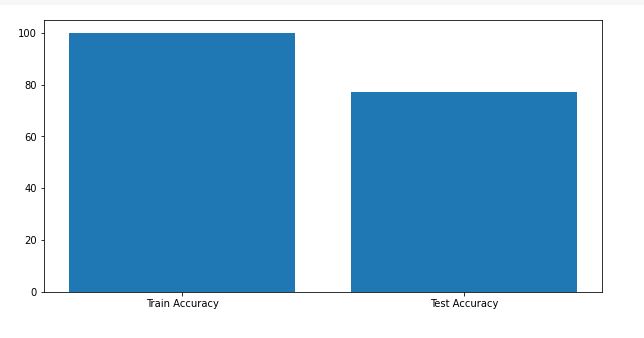


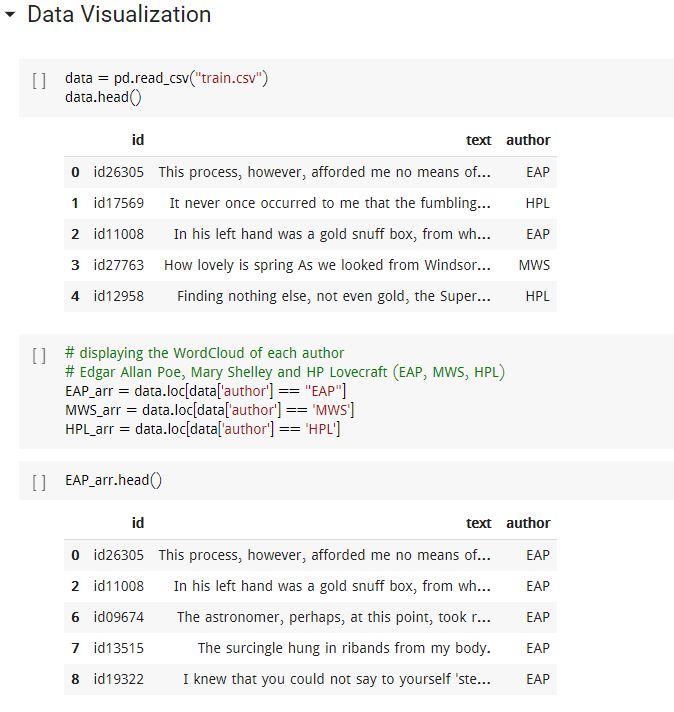
Figure 4: Word cloud of HP Lovecraft

Finally, we reached a very good training accuracy by just looking at the bag of words used by the author. Our model can predict 7 times correctly out of 10 just by looking at the text written by the author. Figure 5 shows the bar plot of accuracies of our model. There is some over-fitting of the model which we determined by looking at the gap between training and testing accuracies. We look to reduce this over-fitting in future by using more data and some regularization effect.

Figure 5: Training and Testing Accuracy

**Here some code snippet of our project:**

**** ****





Chapter 5: Conclusion

The accuracy obtained by our model is close to the accuracy of the baseline model. This project shows that features and fingerprinting features are crucial to the task of author detection. Author detection is a technique of detecting author of a given document. Author detection method includes extraction of features from a given text data such as word length, vocabulary richness, use of digits, Prepositions, Word-length, Pronouns, Conjunctions and so on. Statistical computations are applied on these feature set and compared on both training and test data. Process of detecting author is to provide training data as text written by different authors.