REPORT DOCUMENT

Makabongwe Lwethu Sibisi

ST10145439

Table of Contents

INTRODUCTION

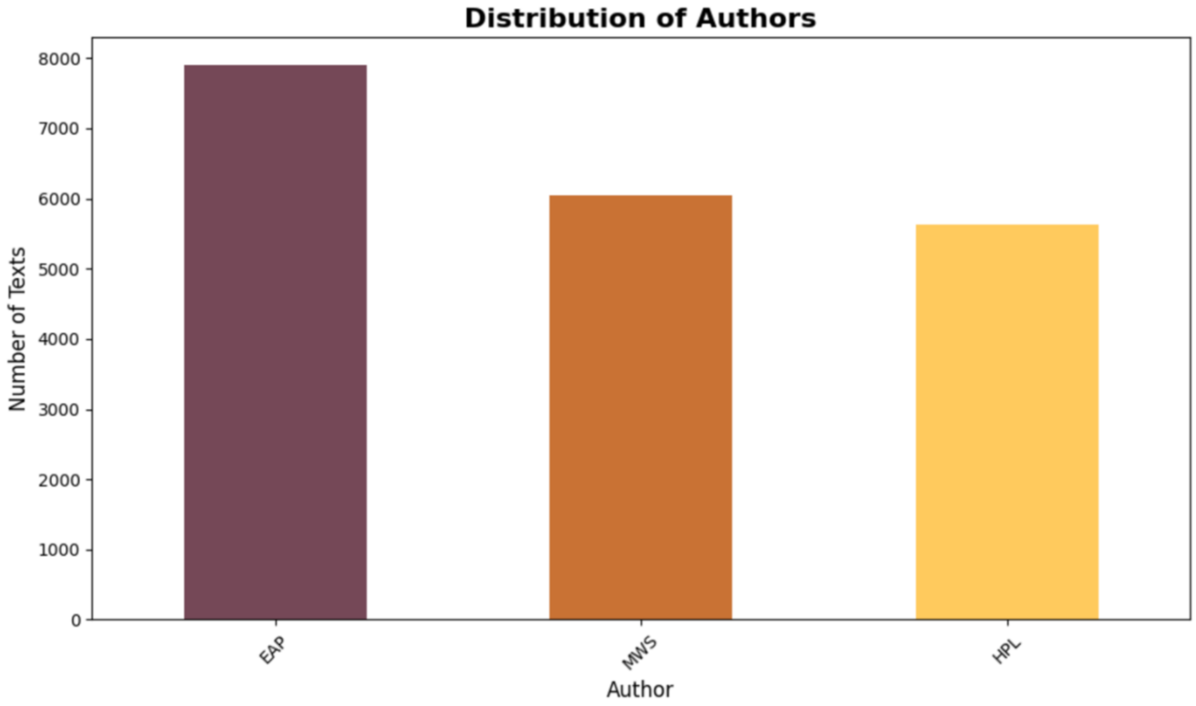
The purpose of this project was to build a Long-Term Short-Term Recurrent Neural Network (LSTM) capable of predicting the author of a literary passage based on writing style. This capability allows fans of the publication to interact with a bot that “guesses” the author, enhancing engagement and showcasing modern textual analysis. The following report details each stage of the process: data cleaning, exploratory data analysis, feature engineering, neural model construction, and the interpretation of results

Data cleaning

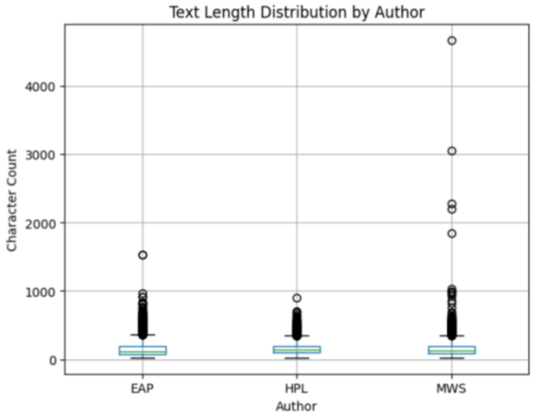
A comprehensive cleaning process was essential to ensure the reliability of all downstream analyses and models. The original Spooky Books dataset consisted of nearly 20,000 entries across three fields: unique identifier, text, and author label. The first verification step confirmed there were no missing values or duplicates, establishing a solid foundation for processing. Substantial cleaning was performed on the text, converting all text data to lowercase and removing special characters and punctuation. Stopwords were eliminated from the analysis using natural language processing tools. Finally, author names were converted into numerical labels, optimizing them for machine learning. These processes standardised the inputs, thus minimizing noise and maximising the ability to uncover stylistic patterns within the dataset.

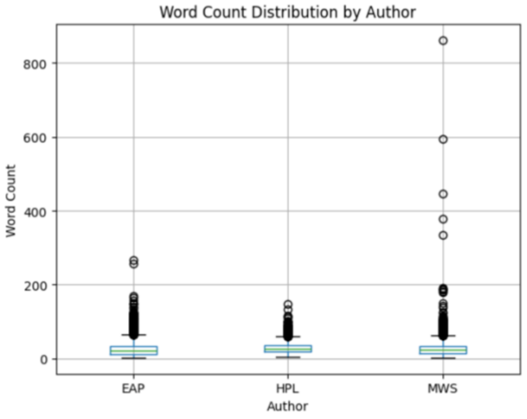
Exploratory Data Analysis

Thorough exploratory data analysis provided insight into the class distributions, stylistic variance, and challenges for textual classification. The first visualization illustrated the distribution of samples per author, revealing that Edgar Allan Poe was the most represented, followed by Mary Shelley, then H.P. Lovecraft. This minor class imbalance was taken into account during modelling.

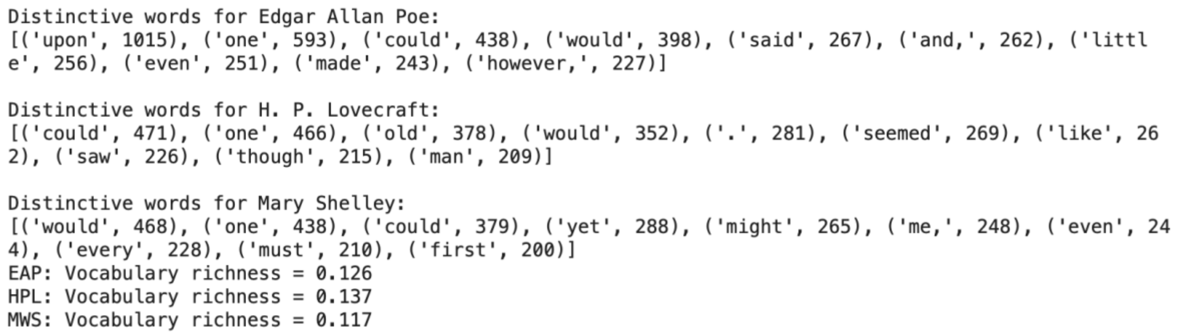


Further analysis examined text length and word count by author. Boxplot visualizations showed that texts attributed to Mary Shelley typically featured the greatest word counts and longest passages, while H.P. Lovecraft's texts tended to be shorter and more concise. Edgar Allan Poe’s samples fell between these two extremes, highlighting distinctive writing habits across authors.





Vocabulary analysis explored the most characteristic words for each author, confirming clear stylistic distinctions. For example, Poe’s texts frequently featured words like “upon” and “however,” Lovecraft’s often used “old” and “seemed,” while Shelley’s texts included “would” and “might.” A calculation of vocabulary richness (unique word ratio per author) revealed Lovecraft as the most diverse, with Shelley the least. These findings provided initial evidence for machine learning models to learn and exploit stylistic cues.



```
Distinctive words for Edgar Allan Poe:
[('upon', 1015), ('one', 593), ('could', 438), ('would', 398), ('said', 267), ('and,', 262), ('littl
e', 256), ('even', 251), ('made', 243), ('however,', 227)]

Distinctive words for H. P. Lovecraft:
[('could', 471), ('one', 466), ('old', 378), ('would', 352), ('.', 281), ('seemed', 269), ('like', 26
2), ('saw', 226), ('though', 215), ('man', 209)]

Distinctive words for Mary Shelley:
[('would', 468), ('one', 438), ('could', 379), ('yet', 288), ('might', 265), ('me,', 248), ('even', 24
4), ('every', 228), ('must', 210), ('first', 200)]
EAP: Vocabulary richness = 0.126
HPL: Vocabulary richness = 0.137
MWS: Vocabulary richness = 0.117
```

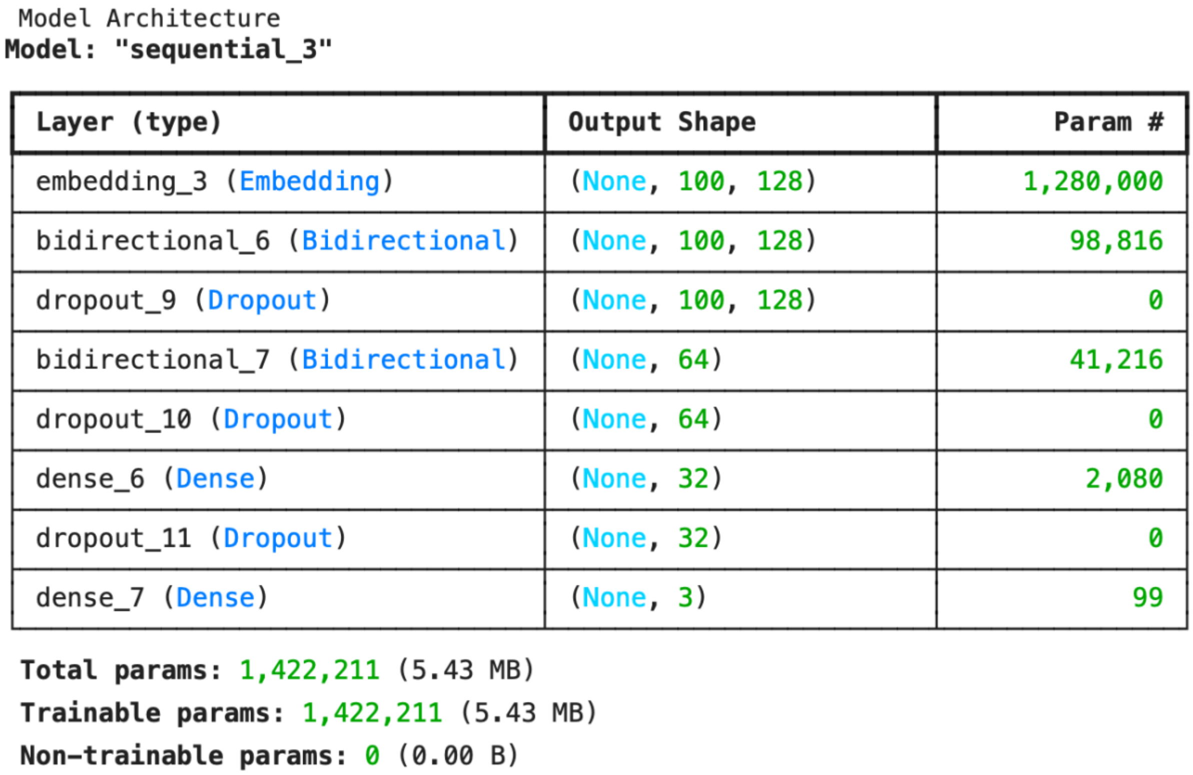
Feature Engineering and Preprocessing

Every original dataset column proved relevant for author prediction. Preprocessing was conducted to convert the cleaned text into a format suitable for deep learning. Tokenization mapped each word to a unique integer, generating a vocabulary of the most common 10,000 words. Cleaning steps ensured that no irrelevant or redundant features were included; all sequences were padded or truncated to a standardized length of 100 tokens, meeting the neural network’s input requirements. The final numerical representation of the data preserved both word order and stylistic structure, prerequisites for successful sequential learning with LSTM models.

Model Development and Training

A custom bidirectional LSTM neural network was constructed to learn the structure and nuance of each author’s writing style. The architecture consisted of embedding layers translating words into dense numeric vectors, followed by two stacked bidirectional LSTM layers (each with 64 units) that processed the text sequence in both directions.

Dropout regularization was applied to mitigate overfitting. The model was completed with a softmax output layer, enabling it to differentiate the three distinct authors. Additional training best practices(early stopping and adaptive learning rate reduction) were employed to prevent overtraining and loss of generalization. In total, the model utilized about 1.4 million trainable parameters, balancing deep learning capability with efficiency. Training lasted up to 30 epochs, with regular progress monitoring.



Model Architecture

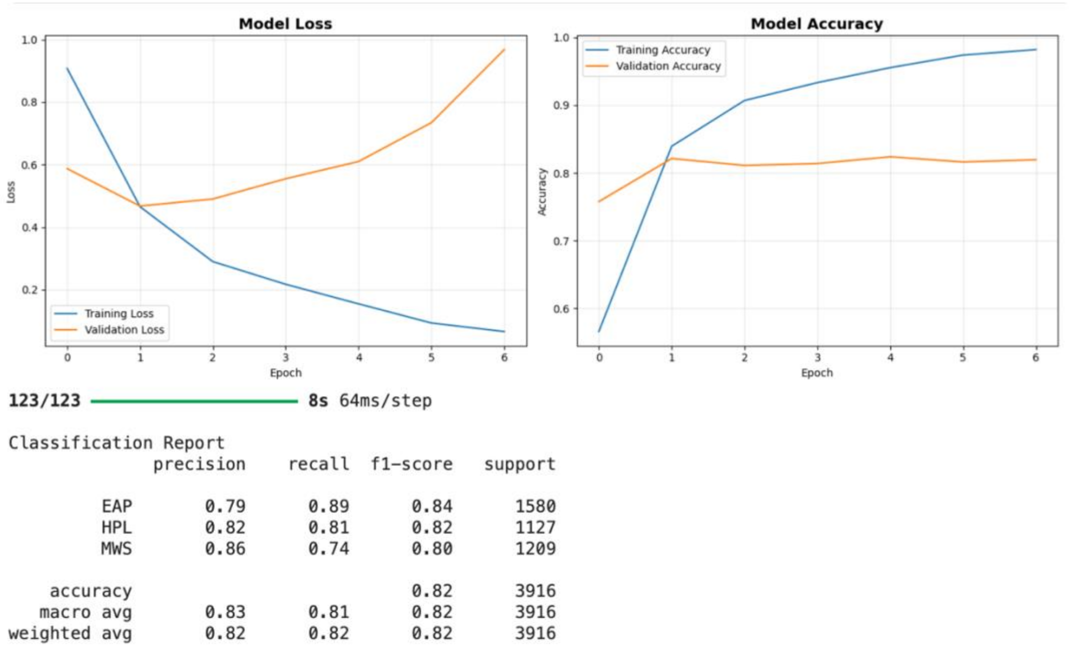
Model: "sequential_3"

Layer (type)	Output Shape	Param #
embedding_3 (Embedding)	(None, 100, 128)	1,280,000
bidirectional_6 (Bidirectional)	(None, 100, 128)	98,816
dropout_9 (Dropout)	(None, 100, 128)	0
bidirectional_7 (Bidirectional)	(None, 64)	41,216
dropout_10 (Dropout)	(None, 64)	0
dense_6 (Dense)	(None, 32)	2,080
dropout_11 (Dropout)	(None, 32)	0
dense_7 (Dense)	(None, 3)	99

Total params: 1,422,211 (5.43 MB)
Trainable params: 1,422,211 (5.43 MB)
Non-trainable params: 0 (0.00 B)

Model Results and Interpretation

Model evaluation was performed on a reserved test split. Visualization of model training and validation accuracy over epochs demonstrated rapid improvement and convergence, followed by early stopping to avoid excess overfitting. The model consistently delivered around 82% validation accuracy on unseen data, a strong result for the task.



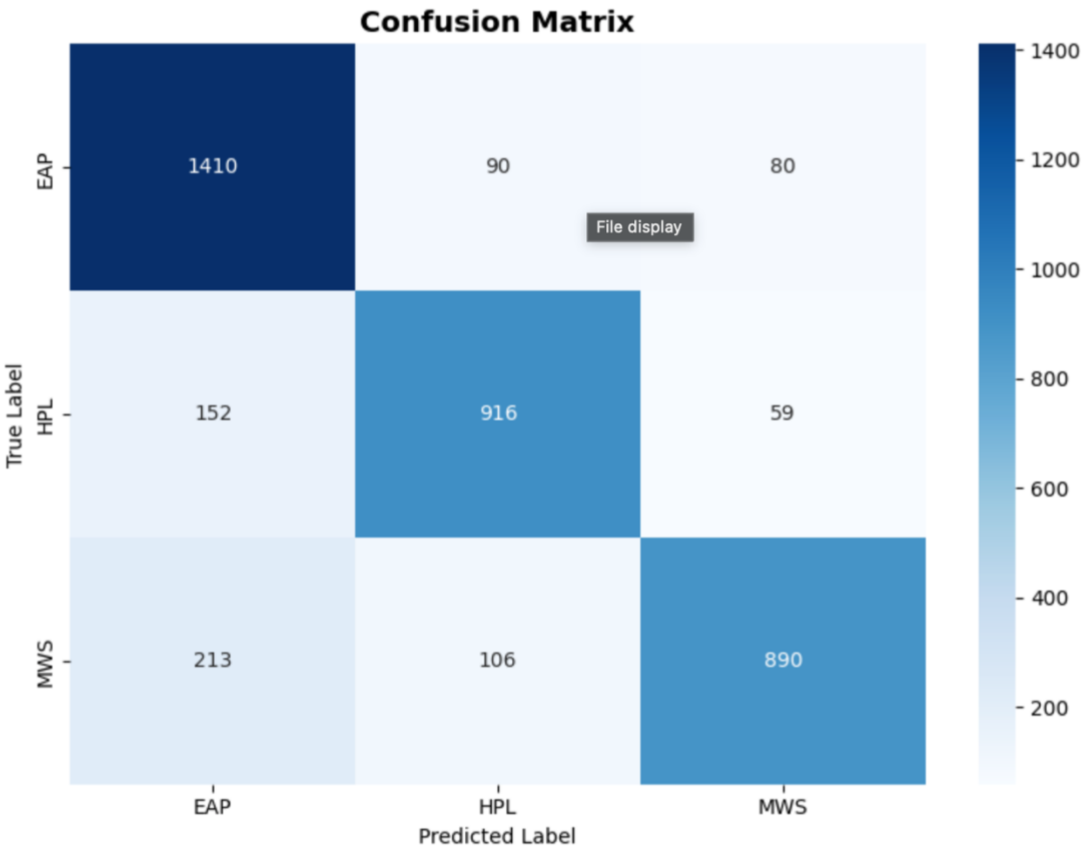
```
123/123 ━━━━━━━━━━━━━━━━━━━━ 8s 64ms/step

Classification Report
              precision    recall  f1-score   support

         EAP       0.79      0.89      0.84      1580
         HPL       0.82      0.81      0.82      1127
         MWS       0.86      0.74      0.80      1209

    accuracy                           0.82      3916
   macro avg       0.83      0.81      0.82      3916
weighted avg       0.82      0.82      0.82      3916
```

Further analysis included confusion matrices and classification reports. These revealed balanced performance across all three author classes, with occasional confusion mostly between the styles of Poe and Shelley. Precision, recall, and f1-scores for each author were summarized to stakeholders, demonstrating robust and reliable identification across the corpus. These results confirm that the LSTM model successfully learned to classify texts by style, meeting both technical and business objectives.



```
=Final Validation Accuracy: 0.8212
```

Business Impact and Application

The developed author prediction bot now reliably classifies textual passages, enabling fans to receive interactive predictions and insights on writing style. The approach demonstrated here is extensible to more authors, genres, and languages. By leveraging deep learning, the system captures complex stylistic signatures, strengthening both product engagement and analytical credibility. This project provides a proven roadmap for future author identification automation and related textual analysis solutions.