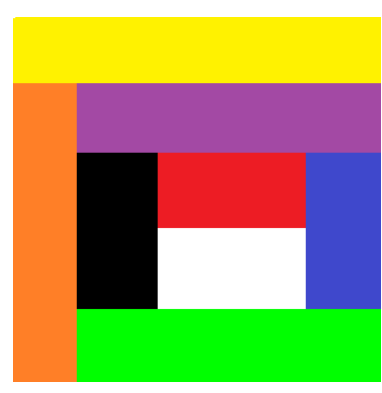




Background

Forensic authorship analysis

The verification and attribution of authorship is a popular task in various domains from software programs to musical scores. A particularly challenging domain is forensic authorship of texts. The central problem is that not only should the results be accurate, but that the method used must be explainable and understood by non-specialists, namely the jurors. The aim of this research is to design, develop and evaluate of an authorship analysis application. A requirements analysis showed that the extant tools were unsatisfactory. Software design documents and prototype were developed for consultation purposes with forensic linguists at the Aston Institute for Forensic Linguists, United Kingdom. This research aims to create a practical

tool that can be used in forensic contexts. **Table 1** shows the different functions of the four domains in this research

Table 1: Overview of Functions for Authorship Analysis Research Domains

Method	Function
Human-guided	To attribute authorship using evidence that can be explained in court
Automated	To verify the evidence collected at scale
Deep learning	To verify the authorship attribution
Explainable AI	To explain the attribution and inform human-guided analysis

Current work

Human-guided

Currently, forensic linguists use multiple tools to manually identify shared linguistic features. However, this is not only time-consuming but error-prone. The primary aim is to identify which distinguishing features occur in both the questioned or disputed document and the known datasets.

We aim to create a one-stop web application that can be used by both lay users (e.g. students) and expert users (i.e. forensic linguists assigned cases by the police force). **Table 2** shows the web app functions valued by experts.

Table 2: Core functions of web app

Function	Description
Search	Users search for features using word tokens, part-of-speech (POS) tags, and/or regular expressions.
Consistency	Users identify shared features among a dataset of one author to establish that prototypicality of the feature.
Comparison	Users identify which prototypical features occur in both the questioned document and the known documents.
Rarity	Users identify the frequency of bigrams by author, e.g. 4 in 10,000.

The web app is developed in the Django framework. A screenshot of the search function is shown in **Fig. 1**. Here you can see that it is possible to search using word token, POS tag and regex.



Fig. 1: Screenshot of the Search function

Fig. 2. shows how the consistency algorithm that detects shared bigrams is displayed, showing the bigrams occurring in all datasets first, then those occurring in all but one, etc.

Occurrence	Bigram	Document A	Document B	Document C	Document D
4	it is	4 (0.002%)	3 (0.0015%)	5 (0.0025%)	2 (0.001%)
4	from the	2 (0.001%)	1 (0.0005%)	3 (0.0015%)	1 (0.0005%)
4	with the	1 (0.0005%)	1 (0.0005%)	2 (0.001%)	1 (0.0005%)
3	as a	12 (0.006%)	11 (0.0055%)	10 (0.005%)	0 (0%)
3	by the	9 (0.0045%)	0 (0%)	8 (0.004%)	7 (0.0035%)
3	that the	0 (0%)	7 (0.0035%)	6 (0.003%)	5 (0.0025%)

Fig. 2: Screenshot of Consistency function

Automated

In the automated analysis, user identifies potentially discriminating shared features, e.g. bigrams. The web app then evaluates the likelihood of authorship based on the relative differences in frequency starting with the highest frequency items and continuing until an authorship decision can be made.

Deep learning and Explainable AI

We are investigating the feasibility of using bespoke fine-grained POS tags to provide additional linguistic information to both identify and explain authorship. Tamura and Blake (2024) subdivided the adjective tag JJ in the Penn TreeBank tagset as show in **Table 3**.

Table 3: Sub-categorization of Adjectives from Previous Research

Tag	Sub-type	Examples
JJ_lim	Limit adjectives	fantastic, awful
JJ_ext	Extraposition	It is interesting that
JJ_presp	Present participle	moving, boring
JJ_pp	Past participle	moved, bored

To test whether this approach could be applied to authorship analysis, we categorized adjectives (JJ) in Shakespeare and Bryant texts from the Gutenberg corpus. The occurrence frequency of each POS tag per paragraph was calculated, and LightGBM was used for classification, employing Leave-One-Out Cross-Validation (LOOCV) for validation. SHAP values were then used to determine the contribution of each POS tag to the classification results. The preliminary results are given in **Tables 4 and 5**. SHAP was used to visualize the contribution of each feature to the prediction (**Fig. 3**).

Table 4: Confusion matrix of model

		Predicted value	
		False	True
Actual value	False	923	261
	True	224	960

Table 5: Evaluation metrics of model

Metric	Value
AUC (ROC)	0.8734
F1 score	0.7983
Accuracy	0.7951

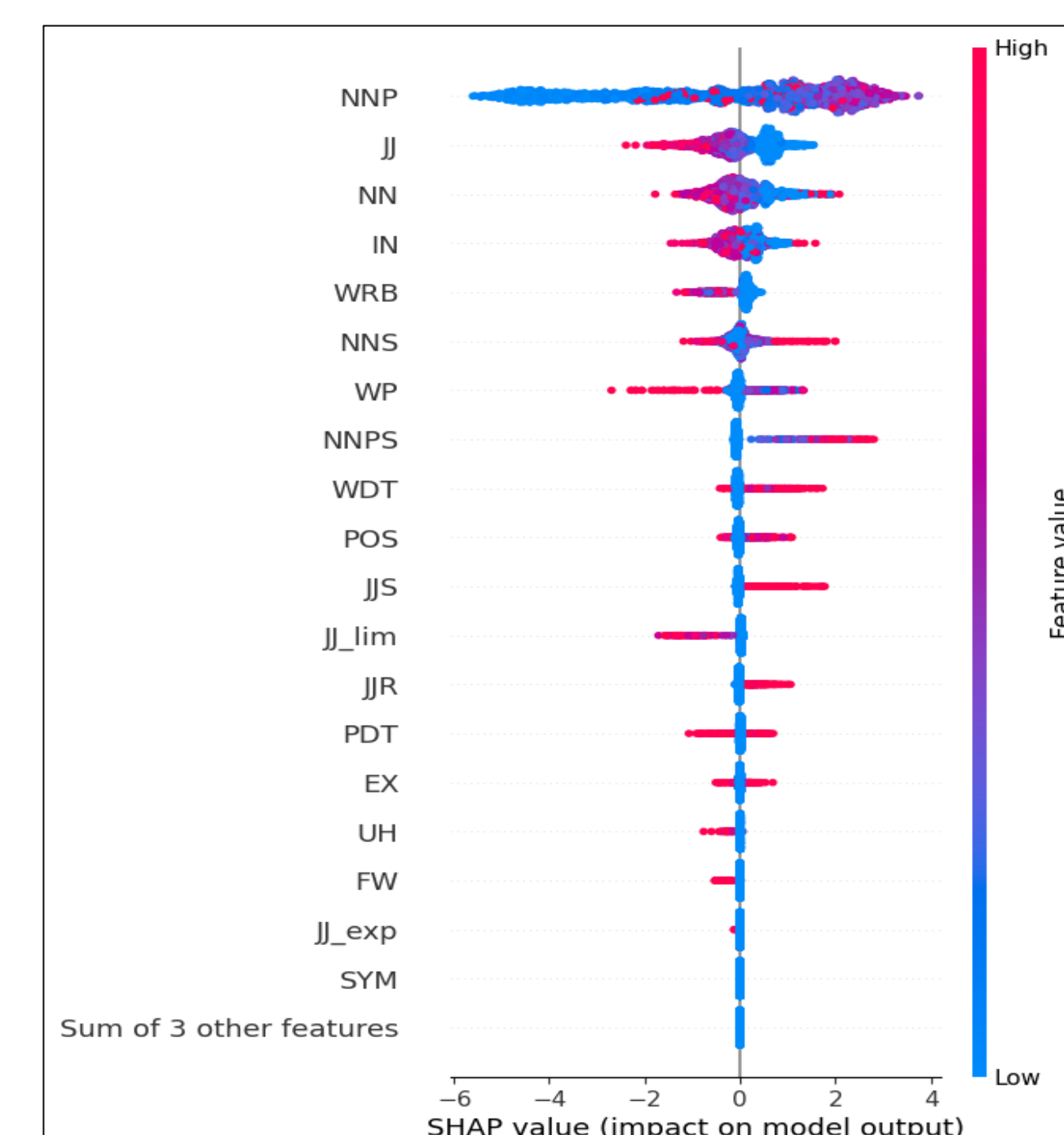


Fig. 3: Screenshot of Consistency function