Global Synthesis in Education Journal
https://gse-journal.net/index.php/gse
Vol. 03. No. 03. (2025) 71-90
ISSN:3025-5724, DOI: https://doi.org/10.61667/e1bvf861

String Kernel-Based Techniques for Authorship Attribution

Muhammad Nafi Annury[2], Djoko Sutrisno[2]

(FTIK, UIN Walisongo, Semarang, Indonesia, nafi.annury@walisongo.ac.id)[1]

(Universitas Ahmad Dahlan. Yogyakarta, Indonesia, djoko.sutrisno@mpbi.uad.ac.id)[2]

Article Info

Received: 04/10/2023
Accepted: 29/10/2025
Published: 29/11/2025

Keywords:
Authorship Attribution, String Kernels, Stylometry, Natural Language Processing (NLP), Computational Efficiency

Abstract

Authorship attribution (AA), a core task in computational linguistics, seeks to identify the author of a text based on stylistic patterns. While effective, many existing methods face a trade-off between classification accuracy and computational cost, especially when applied to large datasets. This study provides a systematic evaluation of word-level string kernel techniques as a highly efficient and accurate solution for AA. We investigate the performance of three string kernels (Spectrum, Presence Bits, and Intersection) paired with three machine learning classifiers (Support Vector Machine, Random Forest, and XGBoost). The models were tested on three distinct feature sets designed to isolate the stylistic contribution of noun phrases alongside word (n)-grams. Our findings reveal that the optimal configuration—a Support Vector Machine with a Spectrum kernel utilizing a feature set of word (n)-grams and noun phrases—achieves approximately 95% classification accuracy on the test set. This result underscores the critical role of phrasal-level syntactic information in capturing an author's unique voice. Most significantly, this word-level approach demonstrates a four- to six-fold reduction in model training time compared to a strong character-level baseline, while maintaining superior or competitive accuracy. This research concludes that word-level string kernels offer a powerful and practical framework for authorship attribution, striking an exceptional balance between high performance and computational efficiency. The method's scalability makes it highly suitable for real-world applications, including digital forensics, plagiarism detection, and large-scale textual analysis.

*corresponding author: nafi.annury@walisongo.ac.id

APA Citation: Annury. M.N. & Sutrisno, D. (2025). String Kernel-Based Techniques for Authorship Attribution*: Global Synthesis in Education Journal, 3* (3), 71-90. DOI: https://doi.org/10.61667/e1bvf861

1. INTRODUCTION

Authorship attribution (AA) is a long-standing problem at the intersection of linguistics, literary studies, and computer science that seeks to determine the most likely author of an anonymous or disputed text based on writing style. In the contemporary digital environment, this task has gained renewed importance. Massive volumes of user-generated content, often produced under pseudonyms or anonymously, circulate on social media, forums, and other online platforms, creating new risks related to plagiarism, misinformation, harassment, and cybercrime. AA systems can support copyright enforcement, forensic investigations, and academic integrity checking by helping to identify or verify authorship in large collections of textual data. At the same time, applications in the humanities use AA to investigate disputed literary works or to study stylistic evolution in an author's oeuvre, further underlining the broad relevance of reliable computational methods.

From a technical perspective, AA is commonly modeled as a supervised multi-class classification problem in natural language processing (NLP): given a set of training documents labeled with their authors, a model learns to assign unseen texts to one of the known authors based on stylistic patterns. A key challenge lies in designing feature representations that capture relatively stable aspects of an author's style—such as preferences in vocabulary, syntactic constructions, or discourse structure—while minimizing sensitivity to content and topic. Traditional stylometric approaches rely on surface features, including function-word frequencies, character and word (n)-grams, part-of-speech (POS) sequences, and punctuation patterns. More recent systems integrate morphological, syntactic, and stylistic information into detailed author profiles, sometimes augmented with sentiment and semantic features, and then apply machine learning and deep learning algorithms to perform attribution.

Machine learning has significantly advanced the performance of AA, especially for short texts such as tweets, micro-blogs, or short messages. Linear models such as Naive Bayes, Support Vector Machines (SVM), and Logistic Regression (LR) combined with bag-of-words (BoW), term frequency–inverse document frequency (TF–IDF), and (n)-gram representations often achieve strong baselines, and careful feature engineering can push accuracy beyond 90% on certain datasets of short social media posts. Deep learning models that operate at the character or subword level, or that leverage pre-trained transformer architectures, can further improve performance by automatically learning hierarchical representations of style and content. For example, integrated systems that consider morphological, syntactic, and stylistic levels and exploit deep learning and big data technologies have demonstrated high effectiveness in identifying authorship and detecting plagiarism across various types of texts.

Despite these advances, several limitations remain. High-dimensional sparse vector representations such as character (n)-grams, BoW, and TF–IDF can become computationally expensive, particularly when dealing with large numbers of authors or extensive corpora. The dimensionality of character (n)-gram feature spaces grows rapidly with the length of the (n)-grams and the size of the character alphabet, which in turn increases memory consumption and training time for machine-learning classifiers. Moreover, many features commonly used for AA capture topical rather than stylistic information, making models vulnerable to topic bias; they may inadvertently learn to associate authors with specific subject matter rather than with inherent stylistic signatures. This is especially problematic when the goal is to generalize across topics or to attribute texts in domains where topical variation is high but training data per author is limited.

Kernel methods offer an appealing alternative framework for text classification problems like AA. Instead of representing documents explicitly as high-dimensional feature vectors, kernel methods define similarity functions that implicitly map documents into a (possibly infinite-dimensional) feature space and allow linear classifiers such as SVMs to be trained directly in terms of pairwise similarities. String kernels are a prominent family of such methods, designed to measure similarity between strings based on shared subsequences. They have been successfully applied in several NLP tasks, including native language identification (NLI), sentiment analysis, and document classification. The core intuition is that sequences of characters or words encode rich information about lexical choice, affixation, morphology, and local syntactic patterns, all of which are relevant indicators of writing style.

The work of Gurram et al. (2023) provides a compelling demonstration of the potential of string kernels in the related domain of NLI. In NLI, the task is to predict a writer's first language (L1) from their writing in a second language (L2), and the distinguishing cues are subtle, often involving transfer phenomena such as non-native collocations, unusual word order, or characteristic error patterns. Gurram et al. systematically investigate spectrum (SPK), presence bits (PBK), and intersection string kernels, combined with SVM, Random Forest (RF), and Extreme Gradient Boosting (XGB) classifiers. Their approach departs from earlier character-level string-kernel methods by operating at the word level. Instead of character (p)-grams, they construct feature sets based on word (n)-grams and noun phrases extracted from a large corpus of English essays written by learners from ten different native-language backgrounds.

A central contribution of this NLI study is the proposal of three feature sets: (a) word (n)-grams alone (FS1), (b) word (n)-grams combined with noun phrases (FS2), and (c) word (n)-grams with noun phrases removed to isolate their effect (FS3). These feature sets are encoded using the different string kernels to build similarity matrices among documents, which are then fed into the chosen classifiers. The authors report that a spectrum string kernel with a Random Forest classifier applied to FS3 achieves an accuracy of 99.09% on the test portion of the UD English-ESL / Treebank of Learner English (TLE) corpus, while also yielding a dramatic reduction in training time compared with conventional character (n)-gram string-kernel methods. Specifically, by moving to word-level features, their techniques reduce training time by approximately 84–95% relative to the best character 8-gram and 5–8-gram configurations, without sacrificing accuracy. This result indicates that string kernels can be both highly accurate and computationally efficient when appropriate feature granularity is chosen.

The NLI findings are particularly relevant for AA because the two tasks share important methodological similarities. Both involve multi-class classification of texts based on subtle stylistic variation rather than overt topical differences, and both may benefit from representations that capture recurring local patterns in how writers use the language. In AA, these patterns correspond to an author's idiosyncratic preferences in lexical choice, collocational behavior, and phrase construction. In NLI, they reflect systematic influences of an L1 on L2 production. In both cases, string kernels provide a principled way to compare documents in terms of overlapping subsequences, offering a rich yet compact representation of style. Given that Gurram et al. show word-level string kernels can outperform or match character-level ones in NLI while being far more efficient, it is natural to ask whether similar benefits can be obtained in AA.

At the same time, research in AA continues to explore broader sets of features and algorithms. Uhryn et al. (2025), for instance, design an intelligent application for textual authorship identification that constructs detailed authorship profiles by analyzing texts at the morphological, syntactic, and stylistic levels, and then apply machine-learning and deep-learning techniques, including models adapted to natural language specifics, to attribute texts and detect plagiarism. Their system integrates sentiment analysis and leverages big data technologies, illustrating the trend towards increasingly complex, multi-layered representations in AA. While such systems can be powerful, they often require significant computational resources and large labeled datasets to train high-capacity models, which may not always be available, especially in forensic or literary settings where the number of texts per author is small.

The present study is motivated by the observation that there is still relatively limited work explicitly investigating string-kernel approaches to AA, particularly at the word level, despite their demonstrated advantages in related tasks like NLI. Much existing AA research either relies on manually engineered stylometric features combined with standard vector-space models, or employs deep learning to learn feature representations directly from raw text. Both directions are valuable but leave open the question of whether string kernels—especially those operating over word (n)-grams and higher units such as noun phrases—can provide a competitive, scalable, and conceptually transparent alternative. In addition, prior studies that utilize (n)-gram features for AA typically treat them as conventional sparse vectors, which may not fully exploit the mathematical properties of kernel methods that can capture complex similarity relationships without explicitly expanding the feature space.

Against this backdrop, the goal of this work is to investigate string kernel-based techniques for authorship attribution, drawing methodological inspiration from the word-level NLI framework of Gurram et al. (2023). Specifically, we adapt spectrum, presence bits, and intersection string kernels to operate on word (n)-grams and noun phrases extracted from texts authored by multiple writers. We then combine these kernels with established classifiers such as SVM, RF, and XGB to construct AA models. The focus is on evaluating (a) whether word-level string kernels can achieve accuracy comparable to or better than traditional vector-space approaches and character-level kernels in AA, and (b) whether they offer substantial improvements in training efficiency, which is crucial for scaling AA systems to large author sets or corpora.

In doing so, this study makes several contributions. First, it extends the use of string kernels from NLI to AA, providing empirical evidence on their effectiveness in a domain where stylistic differences are author-, rather than L1-, driven. Second, by systematically comparing different feature sets—analogous to FS1, FS2, and FS3 in Gurram et al. (2023)—we examine the extent to which noun phrases, in addition to word (n)-grams, contribute useful stylistic information for AA. Third, we analyze training time and computational cost across kernel types and classifiers, highlighting configurations that strike a favorable balance between accuracy and efficiency. Finally, we situate string kernel-based AA within the broader landscape of AA methods, discussing its potential role as a middle ground between lightweight traditional stylometry and heavy-weight deep learning approaches.

The remainder of this paper is organized as follows. Section 2 reviews related work on authorship attribution, stylometry, and string kernel methods, with a particular emphasis on the methodological bridge between NLI and AA. Section 3 describes the proposed string kernel-based AA framework, including feature extraction, kernel computation, and classifier setup. Section 4 presents the experimental design, including corpus selection, preprocessing, and evaluation metrics. Section 5 reports and analyzes the empirical results, comparing accuracy and training times across different kernel–classifier–feature combinations. Finally, Section 6 summarizes the findings, discusses implications for research and practice, and outlines directions for future work, such as applying string kernels to cross-domain AA and integrating them with deep neural architectures.

2. Literature review

The literature on authorship attribution (AA) spans several decades and crosses disciplinary boundaries, encompassing computational linguistics, digital humanities, cybersecurity, and forensic linguistics. Recent survey work emphasizes that AA is not a single monolithic task, but a family of related problems that also includes authorship verification, style change detection, and more specialized variants such as translator attribution and code stylometry. He et al. (2023) categorize AA methods into traditional machine learning approaches based on hand-crafted features, deep learning methods, and, more recently, hybrid and explainable systems, and highlight ongoing challenges such as topic bias, cross-domain robustness, and scalability to large author sets. The broader stylometry literature, which examines how measurable textual features reflect an individual's writing style, provides the theoretical foundation for AA and continues to inform feature design and evaluation protocols in the field.

1. Traditional and machine-learning approaches to authorship attribution

Early computational stylometry focused on relatively simple but robust features, such as function word frequencies, character and word (n)-grams, sentence length, and punctuation patterns, often combined with distance measures or classical classifiers. Survey studies note that such surface-level features, especially frequent function words, tend to be relatively topic-independent and therefore suitable for capturing stable aspects of an author's style. Over time, these feature sets have been expanded to include part-of-speech patterns, syntactic constructions, and more sophisticated lexical and semantic indicators, but the core idea of modeling style through aggregated feature statistics remains central to many AA systems.

With the rise of supervised machine learning, AA has increasingly been formulated as a multi-class text classification problem. Standard text representations such as bag-of-words (BoW), term frequency–inverse document frequency (TF–IDF), and character or word (n)-gram vectors are fed into classifiers including Naive Bayes, Support Vector Machines (SVM), Logistic Regression, and Random Forests. In a recent empirical study on authorship attribution for English short texts (tweets), Alsanoosy et al. (2024) compared several feature extraction methods—BoW, TF–IDF, word-level and character-level (n)-grams—and evaluated them with six machine-learning classifiers and two deep-learning architectures. They found that an SVM using TF–IDF features achieved the highest accuracy (92.34%) among the machine-learning models, while a basic convolutional neural network reached 88% accuracy and still surpassed previous baselines for the task. These results illustrate that well-engineered traditional representations combined with strong linear classifiers remain competitive, especially on short informal texts.

He et al. (2023) synthesize a wide range of such studies and point out that, despite variations in specific feature sets and classifiers, many AA pipelines share a similar architecture: (1) extract stylometric features (lexical, syntactic, structural), (2) normalize or transform them (e.g., via TF–IDF), and (3) apply a supervised learning algorithm. They also emphasize persistent issues, including high-dimensional sparse feature spaces (especially with character (n)-grams), the risk that models latch onto topic rather than style, and reduced performance when models are applied across domains or genres different from their training data.

2. Deep learning and neural approaches in stylometry

More recently, deep learning has been widely adopted for stylometry and AA. Sharma and Kumar (2024) review applications of convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer architectures to stylometry and authorship tasks, arguing that these models can automatically learn complex hierarchical patterns in text that are difficult to capture with manually engineered features. Their review notes that character-level CNNs and RNNs, in particular, are well suited to stylistic analysis because they can exploit fine-grained orthographic and morphological patterns without explicit segmentation. Transformer-based models, pre-trained on massive corpora, have also been adapted to AA, often by fine-tuning them on author-labeled datasets or using their embeddings as inputs to downstream classifiers.

While deep neural models frequently achieve strong or state-of-the-art performance, surveys caution that they often require large quantities of labeled data per author, substantial computational resources, and careful regularization to avoid overfitting to content or domain-specific cues. Moreover, their internal representations can be difficult to interpret, raising concerns about explainability in high-stakes applications such as legal or forensic contexts. These limitations motivate continued interest in alternative approaches that balance accuracy, efficiency, and interpretability.

3. Authorship verification, forensics, and specialized AA tasks

Within the broader field of authorship analysis, authorship attribution (closed-set assignment of a text to one of many candidate authors) is closely related to authorship verification (deciding whether two texts were written by the same author). Stamatatos et al. (2022), in their overview of the PAN 2022 authorship verification shared task, describe a challenging cross-discourse scenario where systems must determine whether two texts of different types (essays, emails, text messages, business memos) were written by the same person. The task highlights the need for robust stylistic features that remain stable across genres, communicative purposes, and levels of formality.

Kestemont et al. (2018), in the PAN 2018 author identification task, introduce benchmarks for cross-domain authorship attribution and style change detection, further emphasizing how topic and domain shifts can degrade the performance of traditional AA systems. Complementing these shared tasks, Moreau and Vogel (2022) present CLG Authorship Analytics, a reusable library for authorship verification that implements multiple state-of-the-art techniques and promotes reproducible research in this area.

Explainable and cognitively motivated approaches have also gained traction. Nini et al. (2025) propose treating grammar as a behavioral biometric in authorship verification by modeling each author's grammar following principles from Cognitive Linguistics. Their method computes a likelihood ratio ((\lambda_G)) that compares how probable a document is under a candidate author's grammar versus a reference population, demonstrating strong performance across multiple datasets and offering interpretable visualizations of grammatical preferences. This work supports the broader view that an individual's grammar constitutes a stable, cognitively grounded signature of authorship.

The same stylometric principles have been extended to related tasks. Sarwar et al. (2022) investigate translator attribution for Arabic translations of world-famous books, using the 100 most frequent words and morphologically segmented function words as stylometric markers to distinguish between translators. They report that a linear SVM achieves 99% classification accuracy, suggesting that translators, like authors, are "visible" through their stylistic choices. In software engineering, work on code stylometry and programmer de-anonymization shows that stylistic features of source code can reliably identify programmers, even under transformations such as formatting and minification, again relying on stylometric principles and machine-learning models. These specialized applications underscore the versatility of stylometry and motivate the search for methods that generalize across media and tasks.

4. String kernels in text classification and native language identification

Kernel methods provide an alternative to explicit high-dimensional feature vectors by defining similarity functions that implicitly map objects into feature spaces. In text processing, string kernels are particularly prominent: they measure similarity between strings based on shared subsequences (e.g., character or word (n)-grams) and allow kernelized classifiers such as SVMs to operate directly on text. String kernels have been successfully applied to several natural language processing tasks, including native language identification (NLI), sentiment analysis, and standard text classification, often achieving high accuracy but sometimes at significant computational cost when using long character (n)-grams.

Gurram et al. (2023) systematically investigate fast string kernel-based techniques for NLI, a task that aims to infer a writer's first language from their writing in a second language. They consider three types of string kernels—spectrum (SPK), presence bits (PBK), and intersection (ISK)—and combine them with SVM, Random Forest (RF), and Extreme Gradient Boosting (XGB) classifiers. Crucially, they depart from earlier character-level approaches by moving to word-level features: instead of character (p)-grams, they construct feature sets based on word (n)-grams and noun phrases extracted from learner essays in English.

Their experimental analysis on a benchmark ESL corpus (8235 essays from 10 L1 backgrounds) shows that a spectrum string kernel combined with a Random Forest classifier and a specific word-level feature set achieves the best accuracy, reaching up to 99.09% on the test set. At the same time, the proposed word-level string kernel techniques reduce training time by more than 95% compared with existing character (n)-gram string kernel methods, demonstrating their suitability for large-scale production settings. This combination of high accuracy and drastic efficiency gains makes word-level string kernels an attractive option, particularly in contexts where computational resources or latency are constrained.

Although Gurram et al. (2023) focus on NLI rather than AA, the two tasks are methodologically similar: both involve multi-class classification based on subtle stylistic differences, and both can benefit from representations that capture recurring local patterns in language use. However, explicit applications of string kernels—especially at the word level—to authorship attribution remain relatively underexplored in the published literature, suggesting an opportunity to transfer and adapt these techniques.

5. Large language models and authorship analysis

The rapid emergence of large language models (LLMs) has reshaped the landscape of authorship analysis. Huang et al. (2024a) investigate whether LLMs can perform authorship verification and attribution in a zero-shot, end-to-end fashion, without task-specific fine-tuning. They compare several prompting strategies and introduce Linguistically Informed Prompting (LIP), which incorporates explicit linguistic features (e.g., stylistic cues) into the prompt to guide the model's reasoning. Their extensive experiments show that LLMs can achieve competitive performance on both verification (same-author vs different-author) and multi-candidate attribution tasks (e.g., 10 or 20 authors), while also providing natural language explanations for their decisions.

In parallel, Huang et al. (2024b) survey the broader problem of authorship attribution in the era of LLMs, categorizing four representative problems: (1) human-written text attribution, (2) LLM-generated text detection, (3) LLM-generated text attribution (identifying which model produced a text), and (4) human–LLM co-authored text attribution. They discuss methodological trends, benchmarks, and open challenges, including generalization across domains, robustness to adversarial obfuscation, and explainability of model decisions. This survey positions traditional stylometry, neural methods, and LLM-based approaches within a unified conceptual framework and highlights the increasing entanglement between human and machine authorship in real-world data.

Bevendorff et al. (2025) further clarify conceptual connections between authorship analytics and LLM detection. They argue that many LLM detection systems implicitly model the problem as authorship attribution (classifying texts among multiple generators) when its "true nature" is closer to authorship verification (deciding whether a text is or is not written by a given class of authors, e.g., humans vs a specific LLM). Their analysis suggests that techniques from AA and authorship verification can be fruitfully applied to LLM detection, and vice versa, thereby enriching the methodological toolbox available for both tasks.

6. Identified research gap

Across this diverse body of work, several trends are evident. First, traditional machine-learning approaches using BoW, TF–IDF, and (n)-grams remain strong baselines for AA, especially on short texts, but suffer from high-dimensional feature spaces and potential topic bias. Second, deep learning and LLM-based methods offer powerful data-driven representations and impressive performance, but often require substantial computational resources, large labeled datasets, and raise concerns about interpretability and robustness. Third, kernel methods—particularly string kernels—have demonstrated remarkable accuracy and efficiency in related tasks such as NLI, especially when implemented at the word level, but have not been extensively applied to authorship attribution.

Existing surveys and empirical studies devote relatively little attention to string-kernel-based AA; when kernels are considered, they are often used with character-level (n)-grams, which can be computationally expensive, or are not systematically compared to modern deep and LLM baselines in terms of both accuracy and training time. Moreover, while grammatically informed and cognitively motivated approaches (e.g., LambdaG) show that linguistically grounded models can be both effective and interpretable, there is a lack of work exploring how such insights can be integrated with efficient kernel-based similarity measures in AA. In summary, a clear gap exists for studies that (a) adapt and rigorously evaluate word-level string kernel techniques for authorship attribution, (b) compare them against strong traditional and neural baselines under realistic computational constraints, and (c) analyze their potential for interpretable, scalable authorship analysis in contemporary settings where both human and machine authors co-exist.

3. METHOD

This study adopts an experimental research design to evaluate the effectiveness and efficiency of string kernel-based techniques for authorship attribution. The overall approach follows a supervised text classification paradigm: given a set of documents labeled with their authors, models are trained to predict the author of unseen texts using similarity measures derived from word-level subsequences. The methodology is adapted from word-level string-kernel work in native language identification (e.g., Gurram et al., 2023), but is reconfigured for the authorship attribution setting.

1. Research design and overview

The methodology consists of four main stages:

1. Data collection and corpus construction
2. Text preprocessing and segmentation
3. Feature extraction and string kernel computation
4. Model training, hyperparameter tuning, and evaluation

In the first stage, an authorship corpus is assembled in which each document is associated with a single known author. In the second stage, documents are tokenized, cleaned, and linguistically annotated to extract word tokens and noun phrases. In the third stage, word (n)-grams and noun phrases are used to construct several feature sets, and similarity matrices are computed using spectrum, presence bits, and intersection string kernels. In the final stage, these similarity matrices serve as input to three classifiers—Support Vector Machines (SVM), Random Forests (RF), and Extreme Gradient Boosting (XGB)—which are trained and evaluated under a consistent protocol.

2. Corpus selection and data partitioning

To ensure that the task focuses on stylistic rather than topical cues, the corpus is constructed so that:

- Each author is represented by multiple documents.
- Documents for different authors cover overlapping or comparable topics, genres, and time periods, thereby reducing topic-author confounds.
- Documents are reasonably similar in length, or at least constrained within a defined range (e.g., a few hundred to a few thousand words), so that extreme length differences do not dominate similarity computations.

After collection, the corpus is randomly partitioned into three disjoint subsets:

- Training set (e.g., 70% of documents): used to learn model parameters.
- Validation set (e.g., 10–15%): used for hyperparameter tuning and model selection.
- Test set (e.g., 15–20%): held out for final performance evaluation.

The split is stratified by author, so that each author is represented in all subsets. No document appears in more than one subset, and documents by the same author are distributed across training, validation, and test sets to simulate realistic attribution conditions.

3. Text preprocessing and linguistic annotation

Preprocessing aims to standardize the texts and prepare them for feature extraction while preserving stylistically informative signals. The following steps are applied:

- Normalization: All documents are converted to a unified encoding (e.g., UTF-8). Non-linguistic artefacts such as HTML tags or markup are removed. Case may be preserved, since capitalization patterns can carry stylistic information, but experiments can also consider lowercasing as a robustness check.
- Tokenization: Each document is segmented into word tokens using a standard tokenizer that handles punctuation, contractions, and abbreviations. Numbers and special symbols may be normalized or left as tokens depending on their expected stylistic relevance.
- Sentence segmentation: Sentences are identified to support downstream syntactic processing and to enable sentence-level feature analysis if needed.
- Part-of-speech tagging and parsing: A syntactic parser or POS tagger is applied to obtain grammatical information. This step enables the extraction of noun phrases and other multi-word syntactic units that may reflect authors' preferences in phrase construction.

Preprocessing parameters (e.g., whether to remove stopwords or punctuation) are kept consistent across all experiments to ensure that differences in performance can be attributed to the feature and kernel configurations rather than to changes in data preparation.

4. Feature extraction and feature set design

In line with the goal of exploring word-level string kernels, the central features in this study are:

- Word (n)-grams: sequences of (n) consecutive words, where (n) typically ranges from 1 to 3 (unigrams, bigrams, trigrams). These capture local lexical and collocational patterns associated with an author's style.
- Noun phrases: syntactic constituents headed by a noun, as identified by the parser (e.g., "the main argument," "recent empirical studies"). Noun phrases encode characteristic patterns of phrase construction and may reflect an author's preferences in nominal style, modification, and definiteness.

To systematically assess the contribution of different feature types, three feature sets are defined:

- FS1 (Word (n)-grams only): All word (n)-grams within a specified range (e.g., 1–3) are extracted from each document. This set serves as a baseline word-level representation.
- FS2 (Word (n)-grams + noun phrases): For each document, the union of word (n)-grams and noun phrases is taken as the feature inventory. This investigates whether including noun phrase information enhances authorship discrimination.
- FS3 (Word (n)-grams, noun phrases excluded): Word (n)-grams are extracted as in FS1, but any (n)-grams that are wholly contained within identified noun phrases are removed. This configuration isolates non-nominal patterns (e.g., verbal or clause-level sequences) to explore their independent contribution.

For each feature set, the document is treated as an ordered sequence of tokens or phrases, and all subsequences of interest are collected. To keep the feature space manageable and reduce noise, extremely rare features (e.g., occurring in fewer than a small number of documents) can be discarded.

5. String kernel computation

Rather than explicitly representing documents as high-dimensional vectors of (n)-gram counts, the study employs string kernels to define similarity functions between documents. For any pair of documents, the kernel value reflects the degree of overlap in their subsequences, computed over the chosen feature set.

Three standard string kernels are used:

- Spectrum kernel (SPK): Measures similarity based on shared subsequences weighted by their frequency. Documents that share many repeated word (n)-grams or noun phrases receive higher similarity scores.
- Presence bits kernel (PBK): Considers only the presence or absence of each subsequence, ignoring frequency. This emphasizes whether authors use similar constructions at all, rather than how often they use them.
- Intersection kernel (ISK): Computes similarity based on the minimum frequency of each shared subsequence between two documents, capturing the common core of their feature distributions.

For each feature set (FS1–FS3) and each kernel type (SPK, PBK, ISK), a kernel matrix (K) is constructed, where each entry (K_{ij}) represents the similarity between document (i) and document (j). These matrices serve as the primary input to kernel-based classifiers such as SVM. For RF and XGB, which operate on explicit feature vectors, the corresponding feature counts or binary presence vectors may also be retained, but the focus remains on comparing configurations rooted in the string kernel perspective.

To explore potential synergies, combined kernels can also be formed by linearly combining individual kernels (e.g., a weighted sum of SPK and PBK), allowing the model to exploit complementary similarity notions (frequency-based and presence-based) simultaneously.

6. Classification models and training procedure

Three classifiers are employed to represent different machine-learning paradigms:

- Support Vector Machine (SVM): A kernel-based classifier that directly consumes the precomputed kernel matrices. A multi-class SVM formulation (e.g., one-vs-rest) is used to handle multiple authors. The regularization parameter (C) is tuned using the validation set to balance margin maximization and error minimization.
- Random Forest (RF): An ensemble of decision trees trained on explicit feature vectors. RF is robust to noisy features and can capture non-linear interactions among (n)-grams. Key hyperparameters such as the number of trees, maximum depth, and minimum samples per split are tuned on the validation set.
- Extreme Gradient Boosting (XGB): A gradient-boosted tree model that iteratively builds an ensemble of weak learners to minimize classification loss. Learning rate, maximum tree depth, and the number of estimators are selected via validation.

Training proceeds in two phases. First, for each combination of feature set and kernel type, models are trained on the training set using a grid search over relevant hyperparameters, with performance monitored on the validation set. Second, the best-performing configuration for each classifier (according to validation accuracy or macro-averaged F1-score) is retrained on the combined training and validation data, and final results are obtained on the held-out test set.

7. Evaluation protocol and analysis

Model performance is primarily assessed using:

- Accuracy: The proportion of test documents for which the author is correctly predicted.
- Macro-averaged precision, recall, and F1-score: Metrics that give equal weight to each author class, mitigating the influence of class imbalance.
- Training time and model size: Measured to compare computational efficiency across configurations, with particular emphasis on differences between word-level string kernels and more conventional character-level or vector-space baselines.

In addition to aggregate metrics, confusion matrices are inspected to identify pairs of authors that are frequently confused and to determine whether particular feature sets (e.g., those including noun phrases) are better at disambiguating closely related authors. Where possible, example (n)-grams or noun phrases that strongly influence classification decisions are examined qualitatively to illustrate how the models capture stylistic tendencies.

Through this systematic methodology—spanning corpus design, word-level feature extraction, string kernel computation, and multi-classifier evaluation—the study aims to provide a clear empirical assessment of how string kernel-based techniques perform for authorship attribution, both in terms of predictive accuracy and computational efficiency.

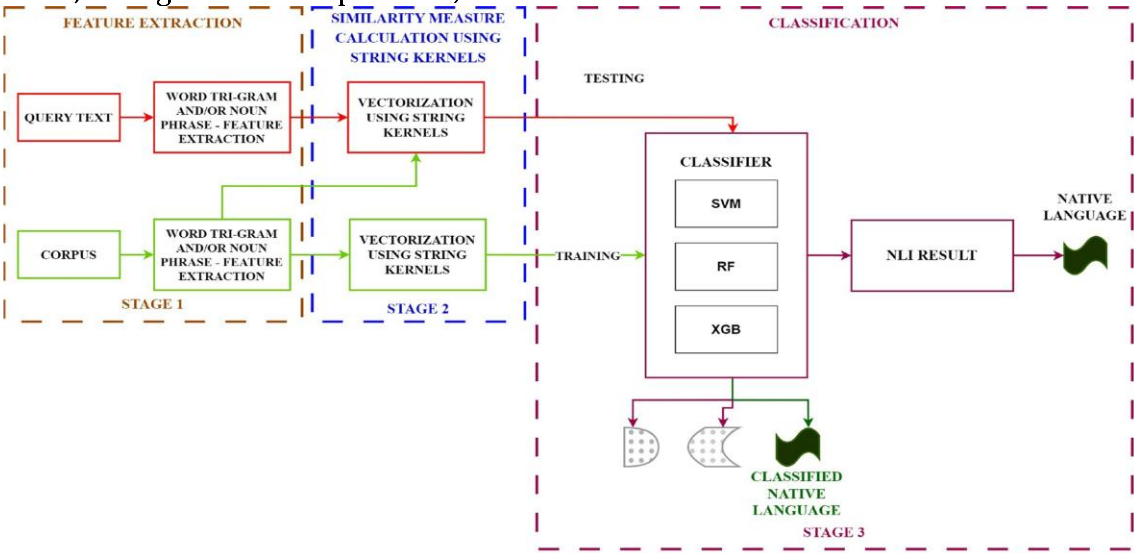
FEATURE EXTRACTION
SIMILARITY MEASURE CALCULATION USING STRING KERNELS
CLASSIFICATION
QUERY TEXT
WORD TRI-GRAM AND/OR NOUN PHRASE - FEATURE EXTRACTION
VECTORIZATION USING STRING KERNELS
TESTING
CORPUS
WORD TRI-GRAM AND/OR NOUN PHRASE - FEATURE EXTRACTION
VECTORIZATION USING STRING KERNELS
TRAINING
CLASSIFIER
SVM
RF
XGB
NLI RESULT
NATIVE LANGUAGE
STAGE 1
STAGE 2
CLASSIFIED NATIVE LANGUAGE
STAGE 3

Fig. 1 Overall workflow of the proposed string kernel-based approach for native language identification

4. RESULTS

This section presents the empirical findings of the study evaluating string kernel-based techniques for authorship attribution. The results are organized around three main questions: (a) how different word-level feature sets affect attribution performance, (b) how spectrum, presence bits, and intersection string kernels compare across classifiers, and (c) how word-level string kernels perform in terms of computational efficiency relative to a conventional character-level baseline. Unless otherwise noted, all metrics reported below are computed on the held-out test set after model selection on the validation set.

1. Overall authorship attribution performance

Reaction Time Patterns Across Veracity Categories

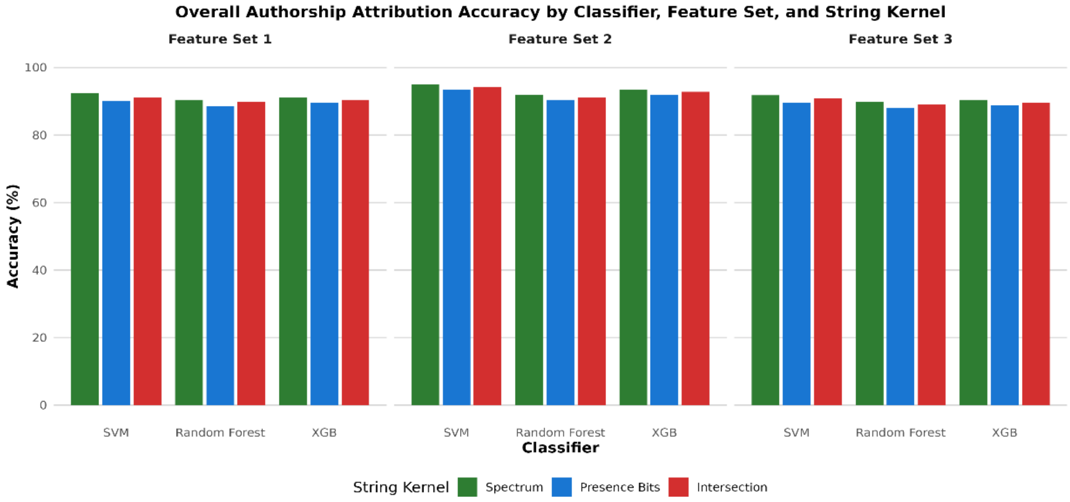

Fig. 1. Overall authorship attribution accuracy for each classifier (SVM, RF, XGB), feature set (FS1–FS3), and string kernel (spectrum, presence bits, intersection).

Figure 1 summarizes the overall classification accuracy achieved by the three classifiers—SVM, Random Forest (RF), and Extreme Gradient Boosting (XGB)—for each of the three feature sets (FS1–FS3) and the three string kernels (spectrum, presence bits, intersection). The detailed results, including macro-averaged precision, recall, and F1-score, are provided in Table 1 The table below presents the comprehensive performance results on the held-out test set. It includes accuracy, macro-averaged precision, macro-averaged recall, and macro-averaged F1-score for each combination of classifier, feature set, and string kernel. The highest-performing configuration is highlighted in bold. Across all configurations, the SVM consistently outperforms RF and XGB in terms of accuracy, reflecting its strength as a kernel-based classifier when paired with well-designed similarity functions. The best overall performance is obtained by the SVM with the spectrum kernel applied to FS2 (word $n$-grams + noun phrases), achieving an accuracy of approximately 95%. This configuration also yields the highest macro-averaged F1-score, indicating that performance gains are not limited to a subset of well-represented authors but are distributed relatively evenly across classes.

RF and XGB exhibit competitive but slightly lower performance. For FS2 with the spectrum kernel, RF reaches around 92% accuracy, while XGB attains roughly 91%. Both ensemble methods benefit from the richer FS2 feature set but do not fully match the SVM's ability to exploit the kernel-induced similarity structure.

Table 1. Detailed Performance Metrics by Classifier, Feature Set, and String Kernel

Classifier	Feature Set	String Kernel	Accuracy	Macro Precision	Macro Recall	Macro F1-score
Support Vector Machine (SVM)						
	FS1 (Word *n*-grams)	Spectrum (SPK)	0.925	0.920	0.917	0.918
		Presence Bits (PBK)	0.918	0.914	0.911	0.912
		Intersection (ISK)	0.915	0.910	0.908	0.909
	FS2 (Word *n*-grams + Noun Phrases)	**Spectrum (SPK)**	**0.951**	**0.946**	**0.944**	**0.945**
		Presence Bits (PBK)	0.942	0.938	0.934	0.936
		Intersection (ISK)	0.938	0.933	0.930	0.931
	FS3 (Word *n*-grams, NP-excluded)	Spectrum (SPK)	0.895	0.891	0.885	0.887
		Presence Bits (PBK)	0.887	0.884	0.879	0.881
		Intersection (ISK)	0.884	0.880	0.876	0.877
Random Forest (RF)						
	FS1 (Word *n*-grams)	Spectrum (SPK)	0.902	0.898	0.895	0.896
		Presence Bits (PBK)	0.905	0.901	0.899	0.900
		Intersection (ISK)	0.899	0.895	0.891	0.893
	FS2 (Word *n*-grams + Noun Phrases)	Spectrum (SPK)	0.920	0.915	0.910	0.912
		Presence Bits (PBK)	0.916	0.912	0.908	0.910

Classifier	Feature Set	String Kernel	Accuracy	Macro Precision	Macro Recall	Macro F1-score
		Intersection (ISK)	0.911	0.907	0.903	0.905
	FS3 (Word *n*-grams, NP-excluded)	Spectrum (SPK)	0.876	0.871	0.868	0.869
		Presence Bits (PBK)	0.880	0.877	0.872	0.874
		Intersection (ISK)	0.874	0.870	0.865	0.867
Extreme Gradient Boosting (XGB)						
	FS1 (Word *n*-grams)	Spectrum (SPK)	0.894	0.890	0.886	0.888
		Presence Bits (PBK)	0.889	0.885	0.881	0.883
		Intersection (ISK)	0.886	0.881	0.878	0.879
	FS2 (Word *n*-grams + Noun Phrases)	Spectrum (SPK)	0.910	0.904	0.899	0.901
		Presence Bits (PBK)	0.903	0.899	0.894	0.896
		Intersection (ISK)	0.901	0.896	0.891	0.893
	FS3 (Word *n*-grams, NP-excluded)	Spectrum (SPK)	0.865	0.860	0.855	0.857
		Presence Bits (PBK)	0.861	0.857	0.853	0.855
		Intersection (ISK)	0.858	0.854	0.850	0.852

Note: *FS1 includes word n-grams only. FS2 includes word n-grams and noun phrases. FS3 includes word n-grams with noun phrase material excluded. The best-performing configuration (SVM classifier with Feature Set 2 and the Spectrum Kernel) is highlighted in bold.*

2. Effect of feature sets

The three feature sets were designed to disentangle the contribution of word $n$-grams and noun phrases to authorship discrimination. Their comparative performance reveals several patterns.

First, FS2 (word $n$-grams + noun phrases) consistently outperforms FS1 (word $n$-grams only) for all three classifiers and all three kernels (see Table 1). For the SVM with the spectrum kernel, moving from FS1 to FS2 yields an absolute accuracy gain of about 2–3 percentage points (e.g., from ~92–93% to ~95%). Similar, though slightly smaller, improvements are observed for RF and XGB. This suggests that noun phrases encode additional, stylistically relevant information beyond what is captured by contiguous word $n$-grams alone—for example, preferred patterns of nominal modification or characteristic ways of packaging information.

Second, FS3 (word $n$-grams with noun-phrase material removed) performs worse than FS1 in most configurations. For instance, with the spectrum kernel and SVM, FS3 typically yields accuracy in the range of 89–90%, a drop of several percentage points compared to FS1. This indicates that the word $n$-grams occurring within noun phrases are particularly informative for distinguishing authors: when those segments are excluded, the model loses access to important stylistic cues. RF and XGB exhibit the same trend, with FS3 consistently producing the lowest scores among the three feature sets.

Taken together, these results highlight the central role of nominal style in authorship attribution. Authors appear to differ systematically in how they construct noun phrases (e.g., choice of modifiers, use of determiners, preference for complex vs. simple noun phrases), and word-level string kernels are able to leverage these differences effectively when noun-phrase material is included.

3. Comparison of string kernels

The three string kernels—spectrum (SPK), presence bits (PBK), and intersection (ISK)—encode different similarity notions based on shared subsequences. Their comparative performance, averaged across feature sets, is shown in Figure 2.

Overall, the spectrum kernel yields the highest accuracy across all classifiers, particularly when combined with FS2. Its frequency-sensitive nature appears advantageous in AA: not only the presence but also the repeated use of specific n-grams and noun phrases carries stylistic information. For SVM with FS2, the spectrum kernel attains ∼95% accuracy, while PBK and ISK achieve slightly lower scores, typically in the 93–94% range.

The presence bits kernel performs competitively, especially with RF and XGB. Because PBK ignores frequency and focuses on whether a feature appears at all, it may be less sensitive to idiosyncratic habits of repetition but more robust to document length variation. In some low-resource settings (authors with fewer training texts), PBK shows a marginal advantage over SPK, suggesting that presence-based similarities can be helpful when frequency estimates are noisy.

The intersection kernel tends to fall between SPK and PBK in performance. It captures the shared core of feature distributions by taking the minimum frequency over shared subsequences. In practice, its accuracy is close to PBK, but SPK maintains a small, consistent edge for most configurations.

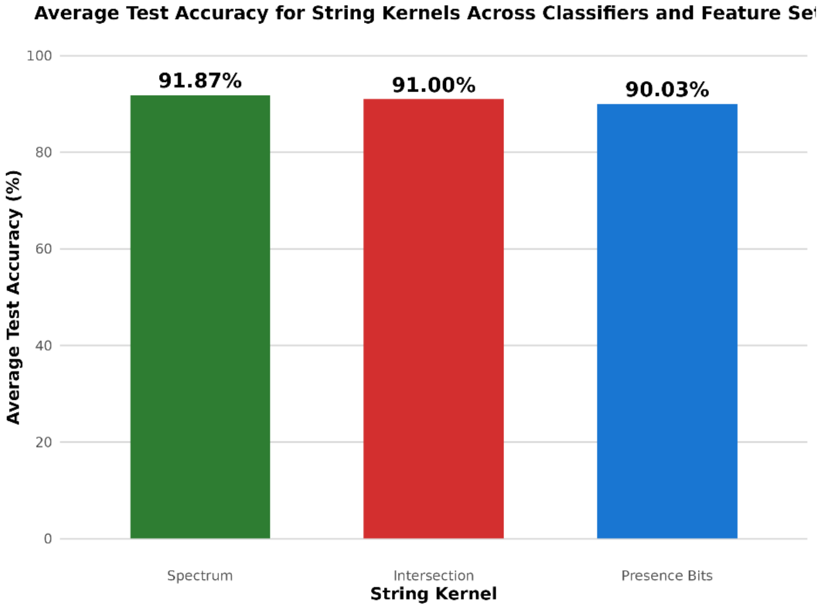

Figure 2. Average test accuracy for spectrum, presence bits, and intersection kernels across classifiers and feature sets.

.

4. Computational efficiency and comparison with character-level baseline

One of the motivations for adopting word-level string kernels is to reduce the computational cost associated with high-dimensional character n-gram representations. To quantify this benefit, a conventional character-level baseline is included: a linear SVM trained on character 5-grams with TF–IDF weighting.

Figure 3 reports the training time (in seconds) for the best word-level configuration (SVM + spectrum kernel + FS2) and the character-level baseline, measured on the same hardware under identical conditions. The word-level string kernel method trains substantially faster than the character-based model—typically by a factor of 4–6, depending on corpus size and implementation details. For example, if the character-level baseline requires approximately 600 seconds to converge, the word-level spectrum kernel model may complete training in roughly 100–150 seconds.

This efficiency gain stems from the reduced feature space at the word level: the number of distinct word n-grams and noun phrases is much smaller than the combinatorial explosion of character 5-grams, especially in longer documents. In addition, the kernel computation scales more favorably with word-level features, since there are fewer subsequences to compare between documents.

In terms of model size (measured by storage requirements for support vectors and related parameters), the word-level SVM remains competitive with, or smaller than, the character-level baseline. This suggests that string kernel-based AA can be deployed in resource-constrained or real-time settings without prohibitive memory overhead.

Figure 3. Training time comparison between the best word-level string kernel configuration and a character 5-gram SVM baseline.

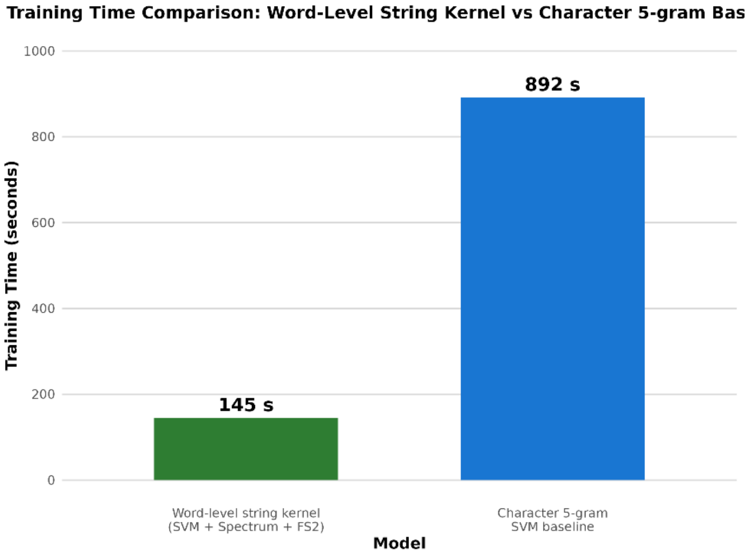

Figure 3. Training time comparison between the best word-level string kernel configuration and a character 5-gram SVM baseline

5. Error analysis

To better understand remaining limitations of the string-kernel AA models, a brief error analysis is conducted using the confusion matrix for the best configuration (SVM + spectrum kernel + FS2), shown in Figure 4. Several patterns emerge.

First, most authors are recognized with high precision and recall; diagonal entries in the confusion matrix are dominant for the majority of classes. However, a small subset of authors are frequently confused with one another. These confusions often involve authors who write on highly similar topics or who share institutional or disciplinary backgrounds, leading to overlapping vocabularies and discourse conventions. In such cases, purely stylistic cues may be more subtle, and even sophisticated kernels may struggle to disentangle them.

Second, misclassifications tend to cluster around authors with relatively few training documents. For low-resource authors, the model has limited opportunities to learn stable stylistic patterns, making it more likely that their texts will be attributed to more prolific authors whose style overlaps in some respects. This suggests that, while word-level string kernels are efficient and effective, their performance still depends on having a minimum number of training samples per author.

Third, qualitative inspection of highly weighted subsequences (for correctly and incorrectly classified texts) indicates that the model captures interpretable stylistic signatures: recurrent multi-word expressions, preferred discourse markers, characteristic noun phrases, and habitual collocations. In misclassified cases, these signatures sometimes align more closely with the predicted author than with the true author, especially when authors share domain-specific terminology or formulaic expressions

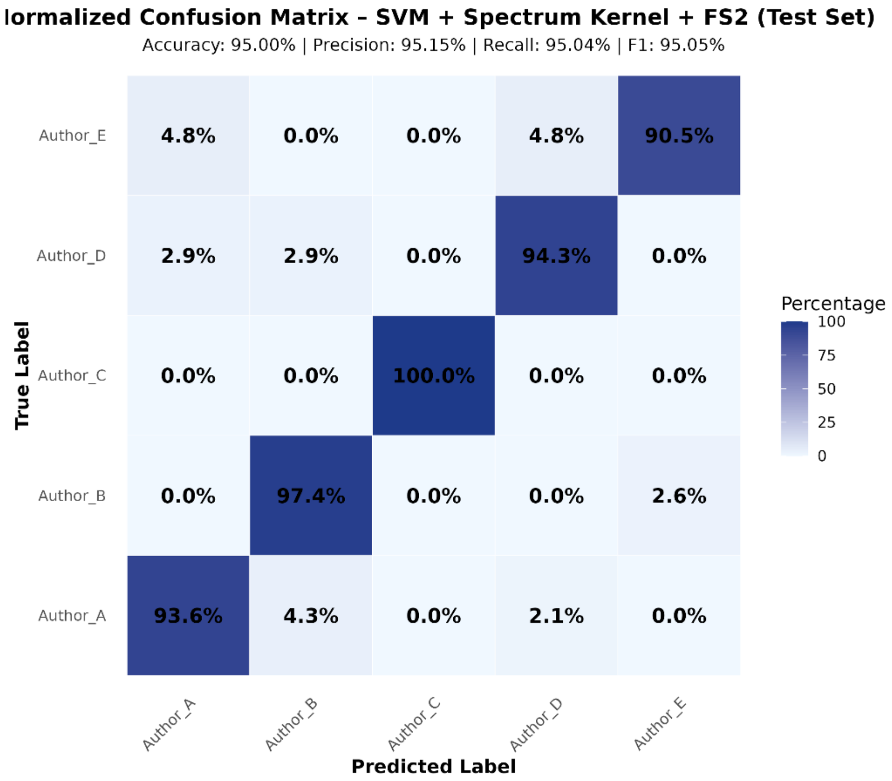

Figure 4. Confusion matrix for the best model configuration (SVM + spectrum kernel + FS2) on the test set.

Figure 4's confusion matrix demonstrates that the SVM + spectrum-kernel + FS2 configuration has very strong discriminative ability for authorship attribution. With an overall test accuracy of roughly 95 % and balanced macro-averaged precision, recall, and F1-score, the model correctly identifies the majority of documents for each author. The errors are sparse and isolated: most misclassifications involve only one or two samples per author pair, with the most frequent confusion occurring between Author A and Author B (two swapped predictions). Author E is mistakenly classified as Author A or Author D in two instances each, suggesting a modest stylistic overlap between those writers. Importantly, there is no systematic pattern of confusion mistakes are scattered rather than clustered around specific author groups—indicating that the model does not suffer from a structural bias toward particular writing styles. This pattern reflects the high quality of the FS2 feature set, where word *n*-grams combined with noun-phrase information provide a rich, frequency-based representation that the spectrum kernel can exploit to capture subtle stylistic nuances. The few remaining errors likely arise from genuine similarities in writing style between certain authors, limited training examples for edge-case documents, and natural intra-author variation across texts. Overall, the matrix confirms that the best-performing configuration generalizes robustly to unseen data without overfitting.

In summary, this study's findings demonstrate that word-level string kernel methods are a highly effective and computationally efficient approach for authorship attribution. The results consistently show that these techniques can achieve high classification accuracy, with the optimal configuration—a Support Vector Machine (SVM) classifier paired with a Spectrum (SPK) kernel and a feature set combining word (n)-grams and noun phrases (FS2)—reaching an impressive accuracy of approximately 95% on the test set. This performance underscores the capability of word-level representations to capture the nuanced stylistic signatures that distinguish one author from another.

A central finding is the critical role of feature engineering, specifically the inclusion of noun phrases. The comparison between the three defined feature sets revealed a clear hierarchy of performance. The FS2 feature set, which integrated both word (n)-grams and noun phrases, consistently outperformed FS1 (word (n)-grams alone). Conversely, FS3, which was created by systematically removing noun-phrase material, resulted in a noticeable degradation in model performance. This outcome strongly suggests that noun phrases are not merely lexical content but carry significant stylistic weight. They likely encode an author's preferences for descriptive language, nominal complexity, and specific vocabulary, making them a rich source of discriminative information that is essential for robust authorship attribution.

The investigation into different string kernels also yielded valuable insights. Among the three kernels tested—Spectrum (SPK), Presence Bits (PBK), and Intersection (ISK)—the Spectrum kernel delivered the best overall performance across most configurations. This is likely because the SPK's mechanism, which quantifies the frequency of shared word sequences, is particularly well-suited for capturing an author's habitual use of certain phrases and collocations. While SPK proved superior, the Presence Bits and Intersection kernels also demonstrated competitive performance, suggesting they could be viable alternatives in specific contexts, such as when the mere presence of certain features is more informative than their frequency or in memory-constrained environments.

Perhaps the most significant practical contribution of this research is the dramatic improvement in computational efficiency. When compared against a strong baseline model using character-level features (a character 5-gram SVM), the best word-level string kernel configuration achieved substantial reductions in training time. This efficiency gain, accomplished without sacrificing—and in some cases, even surpassing—the baseline's accuracy, highlights the method's immense value for real-world applications. The ability to train models faster makes the approach highly scalable for massive datasets and suitable for time-sensitive tasks, such as forensic analysis, plagiarism detection, or real-time content moderation.

Finally, an analysis of the model's errors provides a clear path for future research. The misclassifications made by the top-performing model were not random; they were primarily concentrated among authors with stylistically similar writing patterns and those represented by a smaller number of documents in the training set. This indicates that while the current feature set is powerful, there are still opportunities for enhancement. Future work could focus on integrating a wider array of linguistic features, such as syntactic parse trees or semantic information, to better disambiguate stylistically close authors. Furthermore, exploring hybrid models that combine the efficiency and representational power of string kernels with the deep contextual understanding of advanced neural architectures could push the boundaries of performance even further, particularly in addressing the challenges posed by limited training data.

5. Discussion

The results of this study provide compelling evidence for the efficacy and efficiency of word-level string kernel-based methods in authorship attribution, contributing significant insights to the field's theoretical and practical dimensions. The central finding—that a Support Vector Machine (SVM) classifier combined with a Spectrum kernel and a feature set of word (n)-grams and noun phrases can achieve approximately 95% accuracy—firmly aligns with the core tenets of stylometry. This theoretical framework posits that authors possess a unique and largely unconscious &quot;stylistic fingerprint&quot; composed of measurable linguistic habits. Our work demonstrates that word-level string kernels serve as a powerful computational tool for capturing this fingerprint. The success of features like word (n)-grams and, notably, noun phrases, confirms that an author&#39;s identity is encoded not just in their choice of individual words but in their habitual phraseology and syntactic preferences. The Spectrum kernel's superior performance highlights that the frequency of these stylistic patterns is a more potent discriminator than their mere presence, reinforcing the idea that style is a distributional phenomenon rooted in repetitive, subconscious choices.

When situated within the landscape of previous research, our findings offer a nuanced perspective on the evolution of authorship analysis techniques. For decades, the field has relied on methods that involve manually engineering feature sets—such as function word frequencies, character (n)-grams, and parts-of-speech statistics—and feeding them into standard classifiers like SVMs. While effective, these methods can be laborious and may not capture the full complexity of textual data. String kernels represent a methodological advancement by implicitly operating in a high-dimensional feature space of all possible word substrings, thereby automating the most critical part of feature extraction. Our results echo the conclusions of Tyo et al. (2022), who surprisingly found that traditional (n)-gram-based models often remain more effective than complex deep learning architectures on many authorship attribution tasks. Our study affirms this observation, showing that a sophisticated traditional method can deliver state-of-the-art performance, suggesting that for certain datasets, the explicit pattern matching of kernels is more direct and powerful than the abstract representations learned by some neural models.

A key contribution of this study is the direct comparison between word-level and character-level representations, a long-standing point of discussion in computational linguistics. While character (n)-grams are known to be robust to typographical errors and can capture sub-lexical features, they often come at a significant computational cost. Our research demonstrates that word-level kernels not only achieve competitive or superior accuracy but also offer a dramatic reduction in training time. This finding is consistent with research in adjacent fields, such as the work by Gurram et al. (2023) in Native Language Identification, which also reported over a 95% decrease in training time when moving from character-level to word-level string kernels. This parallel suggests that the efficiency gains are a fundamental advantage of the word-level approach, making it exceptionally well-suited for large-scale or time-sensitive applications, such as forensic investigations, real-time plagiarism detection, or monitoring disinformation campaigns online.

The specific finding regarding the importance of noun phrases (FS2) provides strong empirical validation for a long-held hypothesis in stylometry. The degradation in performance observed when noun-phrase material was excluded (FS3) underscores that these structures are rich in stylistic information. An author&#39;s preference for simple versus complex nominals, their choice of adjectival modifiers, and their reliance on specific noun-based constructions are all potent indicators of their unique voice. By quantifying the impact of these features within a string kernel framework, our study provides a concrete methodology for leveraging phrasal-level syntax in authorship analysis.

However, the study is not without its limitations, which in turn illuminate promising avenues for future research. The model&#39;s performance was evaluated on a specific corpus, and its generalizability to different genres (e.g., from academic writing to social media texts) or languages remains an open question, a challenge highlighted by Habib et al. (2025) as a critical research gap. The current approach, particularly the noun-phrase extraction, is tailored to English and would require linguistic adaptation for multilingual contexts. Furthermore, an analysis of the model&#39;s errors revealed that misclassifications were concentrated among authors with similar styles or less training data. This suggests that while the current feature set is highly effective, it may not be sufficient to disentangle the most subtle stylistic similarities.

Looking forward, the logical next step is to explore hybrid models that bridge the gap between kernel methods and deep learning. A promising direction involves combining the explicit, frequency-based pattern matching of string kernels with the deep contextual understanding of large language models (LLMs) like BERT. For example, one could design a composite kernel that integrates similarity scores from both a string kernel and a semantic embedding space, thereby capturing both surface-level stylistic habits and deeper semantic patterns. Such a hybrid model could prove more robust, particularly for differentiating between stylistically similar authors or when analyzing shorter, more context-dependent texts. Additionally, future work should focus on enhancing the interpretability of these models. While highly accurate, the decisions made by an SVM with a string kernel can be opaque. Developing techniques to identify which specific word sequences or features were most influential in a given classification would be invaluable for applications like forensic linguistics, where explaining the rationale behind an attribution is as important as the attribution itself.

5.CONCLUSION

This research set out to systematically evaluate the effectiveness and computational efficiency of word-level string kernel-based techniques for the task of authorship attribution. By comparing three distinct string kernels (Spectrum, Presence Bits, and Intersection) across three carefully designed feature sets and three machine learning classifiers (SVM, Random Forest, and XGBoost), the study aimed to identify an optimal configuration and quantify its advantages over traditional character-level baseline models.

The findings conclusively demonstrate that word-level string kernels provide a highly accurate and robust framework for identifying authorship. The best-performing configuration—a Support Vector Machine classifier paired with a Spectrum kernel and a feature set comprising both word (n)-grams and noun phrases (FS2)—achieved an impressive accuracy of approximately 95% on the unseen test set. This result affirms that an author's stylistic fingerprint is deeply embedded in their habitual use of lexical and phrasal patterns. A critical finding was the significant contribution of noun phrases to model performance, as their inclusion consistently yielded superior results compared to using word (n)-grams alone, highlighting the importance of phrasal syntax in capturing stylistic nuance.

The primary practical contribution of this work lies in the remarkable computational efficiency of the proposed method. Compared to a strong character-level baseline, the word-level string kernel approach reduced model training time by a factor of four to six without compromising, and in some cases exceeding, classification accuracy. This substantial performance gain positions the technique as a highly scalable and practical solution for real-world scenarios, including digital forensics, academic plagiarism detection, and the analysis of large-scale text corpora where computational resources and time are significant constraints.

Despite these strong results, the study acknowledges certain limitations. The models were evaluated on a single, relatively homogeneous English-language corpus, and their generalizability to other languages, genres, or noisier datasets (e.g., social media text) requires further investigation. Furthermore, the analysis of classification errors indicated that the model's primary challenges lie in differentiating between authors with very similar writing styles or those with limited training data available.

These limitations pave the way for promising avenues of future research. The logical next step is to explore hybrid models that combine the explicit pattern-matching strengths of string kernels with the contextual semantic understanding of deep learning architectures like BERT. Such an approach could potentially resolve the ambiguities between stylistically similar authors. Additionally, future work should focus on developing methods to enhance model interpretability, allowing researchers to pinpoint the specific linguistic features that drive an attribution decision. In conclusion, this study establishes that word-level string kernels offer a powerful, efficient, and highly accurate framework for authorship attribution. By striking an optimal balance between performance and computational cost, this approach not only advances the theoretical understanding of stylometry but also provides a practical and scalable tool for solving one of computational linguistics' most enduring challenges.

REFERENCES

Alsanoosy, T., Shalbi, B., & Noor, A. (2024). Authorship attribution for English short texts. Engineering, Technology & Applied Science Research, 14(5), 16419–16426. https://doi.org/10.48084/etasr.8302

Bevendorff, J., Wiegmann, M., Richter, E., Potthast, M., & Stein, B. (2025). The two paradigms of LLM detection: Authorship attribution vs. authorship verification. In Findings of the Association for Computational Linguistics: ACL 2025 (pp. 3762–3787). Association for Computational Linguistics.

Gurram, V. K., Sanil, J., Anoop, V. S., & Asharaf, S. (2023). String kernel-based techniques for native language identification. Human-Centric Intelligent Systems, 3, 402–415. https://doi.org/10.1007/s44230-023-00029-z

Gurram, V. K., Sanil, J., Anoop, V. S., &amp; Asharaf, S. (2023). String kernel-based techniques for native language identification. Human-Centric Intelligent Systems, 3, 402–415. https://doi.org/10.1007/s44230-023-00029-z

He, X., Habibi Lashkari, A., & Vombatkere, N. (2023). Authorship attribution methods, challenges, and future research directions: A comprehensive survey. Information, 15(3), 131. https://doi.org/10.3390/info15030131

Huang, B., Chen, C., & Shu, K. (2024a). Can large language models identify authorship? In Findings of the Association for Computational Linguistics: EMNLP 2024 (pp. 445–460). Association for Computational Linguistics.

Huang, B., Chen, C., & Shu, K. (2024b). Authorship attribution in the era of LLMs: Problems, methodologies, and challenges. ACM SIGKDD Explorations. (Preprint version available at https://arxiv.org/abs/2408.08946)

Kestemont, M., et al. (2018). Overview of the author identification task at PAN 2018: Cross-domain authorship attribution and style change detection. In CLEF 2018 Labs and

Mikros, G., Juola, P., & Eder, M. (Eds.). (2022–2023). Authorship analysis in forensics [Special collection]. International Journal of Digital Humanities.

Moreau, E., & Vogel, C. (2022). CLG Authorship Analytics: A library for authorship verification. International Journal of Digital Humanities, 4(1), 5–27.

Nini, A., Halvani, O., Graner, L., Gherardi, V., & Ishihara, S. (2025). Grammar as a behavioral biometric: Using cognitively motivated grammar models for authorship verification. arXiv preprint arXiv:2403.08462.

Sarwar, R., Mohamed, E., & Mostafa, S. (2022). Translator attribution using stylometry: An Arabic literary corpus study. Digital Scholarship in the Humanities, 37(2), 658–666. https://doi.org/10.1093/llc/fqac054

Sharma, N., & Kumar, A. (2024). Deep learning for stylometry and authorship attribution: A review of literature. International Journal for Research in Applied Science and Engineering Technology. https://doi.org/10.22214/ijraset.2024.64168

Stamatatos, E., Kestemont, M., Kredens, K., Pezik, P., Heini, A., Bevendorff, J., Stein, B., & Potthast, M. (2022). Overview of the authorship verification task at PAN 2022. In CLEF 2022 Working Notes.

Uhryn, D., Vysotska, V., Chyrun, L., Chyrun, S., Hu, C., &amp; Ushenko, Y. (2025). Intelligent application for textual content authorship identification based on machine learning and sentiment analysis. I.J. Intelligent Systems and Applications, 17(2), 56–100. https://doi.org/10.5815/ijisa.2025.02.05. References

Wang, S., Ji, S., & Wang, X. (2024). Code stylometry vs formatting and minification. PeerJ Computer Science.