STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LXX**, Number 1-2, 2025
DOI: 10.24193/subbi.2025.02

OLDIES BUT GOLDIES: THE POTENTIAL OF CHARACTER N-GRAMS FOR ROMANIAN TEXTS

DANA LUPŞA, SANDA-MARIA AVRAM, AND RADU LUPŞA

Abstract. This study addresses the problem of authorship attribution for Romanian texts using the ROST corpus, a standard benchmark in the field. We systematically evaluate six machine learning techniques — Support Vector Machine (SVM), Logistic Regression (LR), k-Nearest Neighbors (k-NN), Decision Trees (DT), Random Forests (RF), and Artificial Neural Networks (ANN), employing character n-gram features for classification. Among these, the ANN model achieved the highest performance, including perfect classification in four out of fifteen runs when using 5-gram features. These results demonstrate that lightweight, interpretable character n-gram approaches can deliver state-of-the-art accuracy for Romanian authorship attribution, rivaling more complex methods. Our findings highlight the potential of simple stylometric features in resource-constrained or under-studied language settings.

1. Introduction

The authorship determination of a text requires distinguishing the author particularities that are reflected in the written text. Instruments used to do automated authorship attribution (AA) can be characterized along several dimensions, as detailed in the following. *Dataset characteristics* include language, dataset size, class imbalance, and text type (e.g., emails, novels, social media posts, or cross-domain/genre scenarios) [22]. *Preprocessing steps* include text normalization and punctuation handling [17]. *Feature extraction* encompasses character-based features (e.g., character types - letters, digits; character n-grams), lexical features (e.g., word frequencies, word n-grams, stop words, function words), syntactic features (e.g., parts of speech, sentence and phrase

Received by the editors: 25 June 2025.

2010 *Mathematics Subject Classification.* 68T50.

1998 *CR Categories and Descriptors.* I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *Text Analysis*; I.2.6 [**Artificial Intelligence**]: Learning – *Induction.*

Key words and phrases. Authorship Attribution, Machine Learning, Character n-gram.

structure), and semantic features (e.g., topic modeling, word embeddings, contextual representations from large language models) [10, 14, 22, 24]. *Computational methods* rang from traditional statistical techniques (e.g., stylometry, PCA) to machine learning algorithms (e.g., Support Vector Machines, Random Forests), deep learning models (e.g., Recurrent Neural Networks, transformers), and modern large language model (LLM) approaches [12]. *Evaluation metrics* include accuracy, macro-averaged accuracy (especially for imbalanced datasets), precision, recall, F1-score, area under the curve, and confusion matrices [22].

English remains the most studied language in AA research due to the availability of data and the focus on computational linguistics [22, 10]. However, research on Romanian texts has seen growing interest in recent years, both by efforts to support under-resourced languages and by practical applications such as historical and literary analysis, plagiarism detection, and cybercrime investigations [5, 3, 2, 15].

The size of the dataset significantly influences the choice of methodology. Deep learning and LLM-based approaches typically require large datasets for effective training, validation, and testing. In contrast, classical machine learning methods are often more suitable for small to mid-scale datasets (e.g., 10,000–20,000 texts, tens of authors) [11].

Key challenges associated with AA datasets include [23, 14]:

- limited text availability (e.g., few samples per author),
- class imbalance (e.g, in representation of authors or genres),
- generalizability issues (e.g., on cross-domain or cross-genre texts).

The aforementioned challenges are especially pronounced for under-resourced languages like Romanian [3, 2, 15].

Dataset partitioning into training, testing, and validation subsets also affects performance outcomes, as different splits can yield varying results due to random sampling effects. Thus, researchers commonly employ multiple splits or cross-validation techniques to obtain more reliable performance estimates.

While preprocessing transforms raw text into a cleaner format, the feature representation stage, where texts are converted to numerical vectors, is critical to classification success. The choice of features can have a profound impact on the results, often independent of the classification algorithm used, making feature engineering a central focus of AA research.

Different classification algorithms may yield varying results on the same data and features. Consequently, studies often compare multiple algorithms or maintain algorithm consistency when evaluating features. Furthermore, optimizing algorithm hyperparameters can significantly enhance performance.

In this paper, we investigate the effectiveness of character n-gram features to improve authorship attribution performance on the Romanian ROST dataset.

The remainder of this paper is organized as follows. Section 2 reviews related work on authorship attribution, with a focus on approaches relevant to Romanian texts and the ROST dataset. Section 3 presents the experimental setup and results, including dataset description, model configurations, and detailed analyses of classification performance across different feature and parameter settings. Finally, Section 4 concludes the paper with a summary of key findings and outlines directions for future research.

2. Related Work

2.1. **Challenges in Romanian Authorship Attribution.** Authorship attribution (AA) in under-resourced languages, such as Romanian, presents a distinct set of challenges that set it apart from work in high-resource languages. One of the primary obstacles is the limited availability of annotated corpora, which restricts both the scale and diversity of training data available for model development and evaluation. This scarcity is compounded by a significant class imbalance, as datasets such as the ROST [3] corpus exhibit uneven representation of authors, ranging from as few as 27 to as many as 60 texts per author, and substantial variation in text length, from short stories of 90 words to extensive works exceeding 39,000 words. The ROST dataset also encompasses a wide range of genres, including stories, fairy tales, novels, articles, and sketches, and spans a broad historical period from 1850 to 2023, further increasing heterogeneity and complicating model generalization.

These factors introduce biases and variability that are difficult to control, making it challenging to develop robust and generalizable AA models for Romanian [3, 2]. Addressing such issues is essential for advancing the field, as models trained on unbalanced or limited data may fail to perform reliably across different authors, genres, or time periods.

Recent research has begun to address these challenges by developing hybrid models that combine handcrafted linguistic features with contextualized embeddings, tailored specifically for Romanian and other under-resourced languages. However, the linguistic complexity of Romanian, characterized by rich morphology and flexible word order, means that effective solutions for English or other major languages often do not transfer directly [15]. Similar difficulties have been reported in other under-resourced languages, such as Albanian, where the lack of large annotated corpora continues to impede progress in authorship attribution research [13].

Moreover, the scarcity of large, publicly available Romanian corpora limits the applicability of data-intensive deep learning methods, making careful feature engineering and rigorous evaluation protocols especially important for achieving reliable results in this context.

2.2. **N-gram Features in Authorship Attribution.** N-grams are contiguous sequences of N items, such as characters, words, or other tokens, extracted from text to capture patterns indicating an author's unique style. Typically, n-grams are constructed at the character, word, or syntactic level [24].

Character n-grams have been extensively used in authorship attribution research due to their ability to encode stylistic nuances, lexical patterns, word order tendencies, and punctuation or capitalization habits [22]. Despite their widespread use in languages such as English, Romanian texts remain relatively understudied using this approach [22, 3].

2.2.1. *Advantages of Character N-grams.*

- **Robustness:** Character n-grams are resilient to infrequent errors such as grammatical mistakes or punctuation slips because their discriminative power derives from frequent, recurring patterns [23].
- **Preservation of Stylistic Traits:** Subtle author-specific variations, such as repeated punctuation marks or distinctive capitalization, are naturally encoded within character n-grams [12]. For instance, some authors rarely use exclamation marks, while others employ them frequently. Similarly, preferences for sentence length can be reflected through punctuation usage patterns, such as the relative frequency of periods versus commas, which correspond to shorter or longer sentence structures, respectively [9].
- **Language Independence:** Character n-grams do not require deep linguistic knowledge of grammar or semantics, making them applicable across diverse languages without modification.
- **Computational Efficiency:** Extracting character n-grams is both straightforward and computationally inexpensive, as it involves direct processing of raw text without the need for complex preprocessing.

2.2.2. *Limitations of Character N-grams.*

- **Redundancy:** Due to the overlapping nature of n-grams, each n-gram shares $N-1$ characters with adjacent ones. Therefore, many n-grams represent slight variations of the same lexical unit (e.g., `"in_"`, `"in."`, `"in!"`). While this redundancy can reinforce stylistic signals such as affixation (prefix/suffix) or punctuation preferences, it may also lead to overfitting if not properly managed.
- **Sparsity and High Dimensionality:** Compared to word-level n-grams, character n-grams, especially for larger N, tend to generate

very high-dimensional and sparse feature spaces, which can pose challenges for model training and generalization [23, 20].

2.3. **Computational Methods.** A wide range of classification algorithms has been applied to authorship attribution (AA) leveraging character n-gram features [4, 6, 7, 1, 18, 26]. Next, we provide a concise overview of the principal methods utilized in this study:

Support Vector Machine (SVM) is a supervised learning model that identifies an optimal hyperplane to separate author classes by maximizing the margin between data points of different classes. It excels in handling high-dimensional feature spaces such as those generated by n-grams and is robust against overfitting and noise, making it a popular choice in stylometric analysis.

Logistic Regression (LR) is a probabilistic linear model used for binary and multiclass classification. It maps input features to class probabilities via the logistic (sigmoid) function and applies regularization (L1 or L2 penalties) to prevent overfitting. LR provides interpretable coefficients, which can be valuable for understanding the contribution of specific n-grams, particularly in smaller datasets.

k-Nearest Neighbors (k-NN) is a non-parametric, instance-based classifier that assigns a class label based on the majority vote of the k nearest neighbors in the feature space. It relies on distance metrics such as Euclidean, cosine, or Minkowski distance. While simple to implement and intuitive, k-NN can be computationally expensive for large datasets and high-dimensional n-gram features.

Decision Trees (DT) recursively partition the feature space by selecting feature thresholds that maximize class purity, typically using criteria like entropy or Gini impurity. The resulting tree structure yields interpretable decision rules that can highlight stylometric patterns, such as frequent character sequences. However, DTs are prone to overfitting, especially with high-dimensional n-gram data.

Random Forest (RF) is an ensemble method that constructs multiple decision trees using random subsets of the data and features, aggregating their predictions via majority voting. This approach reduces overfitting through bagging and feature randomness. RF handles noisy, high-dimensional n-gram data effectively and provides measures of feature importance, enabling identification of discriminative n-grams.

Artificial Neural Networks (ANN) consist of interconnected layers of neurons that learn hierarchical feature representations through backpropagation. Activation functions such as ReLU enable the modeling of complex, non-linear relationships among features. While deep ANNs typically require large datasets, shallow architectures (1–2 hidden layers) are well-suited for moderate-sized corpora like ROST [3].

Among the aforementioned methods, SVM and RF have traditionally dominated AA research [23] due to their strong performance with stylometric features such as n-grams. ANNs, particularly deep learning models, are gaining traction but generally demand larger datasets. Logistic Regression and k-NN often serve as baseline benchmarks, while standalone Decision Trees are less commonly employed due to their susceptibility to overfitting.

2.4. **Evaluation Metrics.** Authorship attribution studies commonly employ the following evaluation metrics to assess model performance:

2.4.1. *Accuracy.* Accuracy measures the proportion of correctly classified texts across all classes. While widely used, accuracy can be misleading for imbalanced datasets, as it may overemphasize performance on majority classes.

2.4.2. *Macro-Accuracy (Balanced Accuracy).* Macro-accuracy addresses class imbalance by calculating accuracy for each class independently and then averaging the results:

$$\text{Macro-Accuracy} = \frac{1}{C}\sum_{i=1}^{C} \text{Accuracy}_{Class_i}$$

where C is the number of classes (authors). This metric ensures all authors contribute equally to the final score, making it more representative for datasets with uneven class distributions.

2.4.3. *Precision, Recall, and F1-Score.* These metrics provide complementary insights into model performance, particularly for imbalanced data:

- **Precision** The proportion of correctly attributed texts for an author out of all texts predicted as that author. High precision minimizes false attributions.
- **Recall** The proportion of correctly attributed texts for an author out of all texts actually written by that author. High recall indicates strong retrieval of true positives.
- **F1-Score** The harmonic mean of precision and recall. It reflects how well a system is at finding relevant items (precision) and all of them (recall), without being skewed by abundant irrelevant items. Particularly important for imbalanced datasets, where, if one class significantly outnumbers the others, a model can achieve high accuracy by simply predicting the majority class.

2.4.4. *Confusion Matrix.* A table showing the number of correct and incorrect predictions for each class. Reveals which authors are frequently confused. It summarizes the true positives, true negatives, false positives, and false negatives, from which accuracy is derived.

3. Experiments and results

3.1. **The dataset.** Motivated by the relative scarcity of research on Romanian authorship attribution (AA) and the challenges posed by the ROST corpus, our work aims to advance the field by exploring lightweight yet effective feature representations. The ROST dataset, comprising approximately 400 texts authored by 10 writers, is characterized by significant class imbalance and diverse text lengths and genres, making it a challenging benchmark for AA [3, 2].

Previous studies have established important baselines on ROST: Avram et al. (2022) introduced the dataset and evaluated five classification methods, including Artificial Neural Networks (ANN), Multi Expression Programming (MEP), k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), and C5.0 Decision Trees, using inflexible part-of-speech tags as features. Their best macro-accuracy was 80.94%, with an overall error rate of 20.40% [3]. Avram (2023) leveraged a Romanian-specific BERT model, achieving approximately 87% macro-accuracy [2]. Nitu et al. (2024) proposed a hybrid Transformer architecture combining handcrafted linguistic features (lexical, syntactic, semantic, discourse markers) with BERT embeddings, reaching a state-of-the-art F1-score of 0.95 and reducing the error rate to 4% [15].

While transformer-based approaches demonstrate impressive performance, their reliance on large pretrained models and substantial computational resources limits their accessibility, especially in resource-constrained environments. In contrast, our study investigates character n-grams, a lightweight, interpretable feature set, to assess whether they can achieve comparable performance on the ROST dataset. By systematically evaluating n-gram sizes ranging from 2 to 5, we aim to demonstrate that simpler, computationally efficient methods can provide competitive accuracy while offering greater interpretability and ease of deployment.

3.2. **Data Preprocessing.** In the raw text preprocessing phase, we apply *normalization* to standardize the textual data. Therefore, texts were initially preprocessed to address specific character encoding variations like:

- **Diacritic standardization:** converting Romanian characters such as Ş/ş and Ţ/ţ to their canonical forms Ș/ș and Ț/ț;
- **Punctuation unification:**
  - convert smart quotes („ “ ” ’) to straight quotes (")
  - convert en/em dashes (–—) to hyphens (-)

  - convert ellipses (…) to triple periods (. . .)
- **Whitespace regularization:** collapse multiple contiguous whitespace into a single space

These steps ensure consistency and reduce noise in the dataset, facilitating more reliable downstream analysis.

In addition to normalization, we performed the following preprocessing steps:

- **Case Handling:** We conducted experiments using both lowercase text and the original case, to assess the impact of letter casing on authorship attribution.
- **Digit Replacement:** Since the specific values of the numbers were not relevant to our analysis, all digits were replaced with a special character (@), which does not occur in the original texts.
- **Punctuation Preservation:** All other special characters (punctuation marks) were left unchanged, as patterns in punctuation usage may reflect individual authorial style.
- **Whitespace Encoding:** Recognizing the potential importance of whitespace as a stylistic marker, we replaced spaces and tabs with the underscore character (_), and newlines with the dollar sign ($). This allows us to explicitly encode paragraph boundaries and whitespace usage into the N-grams as distinct features.

3.3. **Feature Extraction Process.** We employ character N-grams to construct stylometric representations of texts, following these steps:

3.3.1. *N-gram Definition and Range.* Drawing on prior stylometric studies (Houvardas et al., 2006; Ramezani et al., 2013; Ntoulas et al., 2006; Smith & Jones, 2022) [8, 19, 21, 24], we analyze character sequences of length $N = 2$ to $N = 5$. This range captures both local orthographic patterns (e.g., bigrams) and longer morphological features (e.g., pentagrams).

3.3.2. *Vectorization via TF-IDF.* Each document is transformed into a numerical vector using Term Frequency-Inverse Document Frequency (TF-IDF) weighting. This approach emphasizes N-grams that are statistically distinctive to individual documents while down-weighting common sequences.

3.3.3. *Comprehensive Feature Inclusion.* To avoid potential information loss from premature feature selection, we retain all extracted character N-grams during initial modeling. This ensures maximal preservation of potential stylistic markers.

3.3.4. *Feature Matrix Construction.* The final representation is a $D \times F$ matrix, where:

- $D =$ Number of documents
- $F =$ Total unique N-grams across all N values
- Cell values = TF-IDF weights for N-gram/document pairs

3.4. **Experimental Setup and Classification Methods.**

3.4.1. *Implementation Details.* All experiments were conducted in Python, utilizing the following libraries and tools:

- **Data Processing and Manipulation:** `numpy` and `pandas` for numerical operations and data handling.
- **Feature Extraction:** using `TfidfVectorizer` from `scikit-learn` to generate character N-gram feature representations.
- **Model Training and Evaluation:** `scikit-learn` for implementing and evaluating machine learning classifiers.

3.4.2. *Classification Algorithms.* We explored several supervised learning algorithms: Support Vector Machine (SVM), Logistic Regression (LR), k-Nearest Neighbor (k-NN), Decision Trees (DT), Random Forest (RF), and Artificial Neural Networks (ANN).

3.4.3. *Model Evaluation.* We use standard metrics including *accuracy*, *macro-accuracy* (aka. balanced accuracy or macro-averaged accuracy), and the *classification report* (precision, recall, F1-score), as provided by `scikit-learn`.

3.5. **Experimental Procedure.** For each of the classification methods described above, we conducted a set of experiments to identify the configuration yielding the best performance. Our investigation included the following key aspects:

- **Parameter Exploration:** For k-NN, we evaluated five different values of k to determine the optimal neighborhood size.
- **Robustness to Randomness:** To mitigate the effects of stochastic variability inherent in certain models, we repeated the training and evaluation of DT, RF, and ANN five times, each with a different random seed. This procedure allowed us to assess the stability and reliability of the results.
- **Evaluation Metrics:** Performance was assessed using multiple metrics, including accuracy and balanced accuracy, to provide a comprehensive understanding of classifier effectiveness across potentially imbalanced classes.

- **Data Splitting:** We employed randomly selected train-test splits to preserve class distribution during evaluation, ensuring fair and representative performance estimates.

This rigorous experimental protocol ensures that observed performance differences reflect genuine model capabilities rather than artifacts of data sampling or initialization.

ML Model	**Hyperparameters**	**Variation Parameters**
SVM	kernel: linear	—
LR	solver: lbfgs, penalty: l2	—
k-NN	metric: minkowski	n_neighbors = $\{3, 5, 7, 9, 11\}$
DT	criterion: gini	randomstate = $\{7, 17, 42, 67, 101\}$
RF	n_estimators: 100 criterion: gini	randomstate = $\{7, 17, 42, 67, 101\}$
ANN	activation: relu hidden layer sizes: (100,50)	randomstate = $\{7, 17, 42, 67, 101\}$

Table 1. Summary of machine learning models and their hyperparameters. To ensure that performance differences reflect model characteristics rather than random variation, DT, RF, ANN were each trained and evaluated five times using different random seeds. For k-Nearest Neighbors (k-NN), five different values of the number of neighbors were tested.

3.6. **Parameter Tuning and Model Selection.** The dataset was partitioned into training (80%) and testing (20%) subsets. This splitting procedure was repeated five times to generate distinct train-test partitions, ensuring robustness and reliability of the evaluation. For each partition, all six machine learning models (SVM, LR, k-NN, DT, RF, ANN) were trained and evaluated using the hyperparameter configurations detailed in Table 1.

Furthermore, to assess the impact of text casing on model performance, experiments were conducted on both the original case and fully lowercased versions of the texts.

3.7. **Results Overview and Key Findings.**

3.7.1. *Impact of Letter Casing on Classification Performance.* We conducted experiments comparing classification performance on original-case versus fully lowercased texts. The observed differences in Macro-Accuracy (MAcc) were generally minor. The largest difference, 0.031, occurred with the k-Nearest Neighbors (k-NN) classifier at n-gram size $N = 3$, where the lowercase representation slightly outperformed the original case.

Figure 1 depicts the variation in MAcc differences across all classifiers and n-gram sizes. Overall, lowercase texts tend to yield marginally better results,

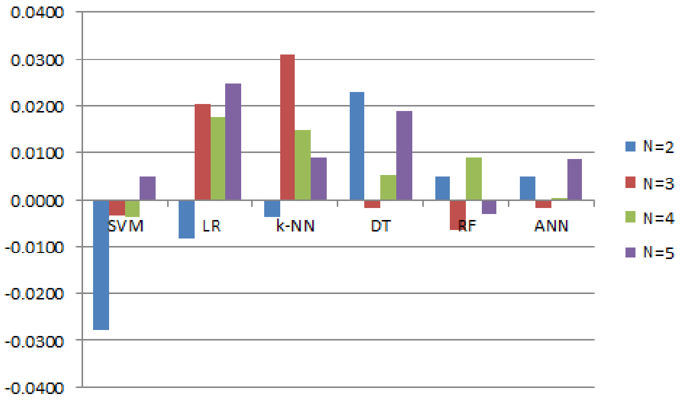


FIGURE 1. Differences in Macro-Accuracy (MAcc) averages between fully lower-cased and original-case texts for each N-gram size (N = 2 to 5) and for each of the six classification methods: SVM, LR, k-NN, DT, RF, and ANN. The values used to calculate the differences presented in the chart, were computed as follows: for each N-gram size, each classification method and each (letter) casing, several runs were executed (as described in Table 1 and in section 3.6) and the results were averaged.

particularly for $k > 3$ and for the k-NN and LR classifiers. However, in most cases, the difference remains below 0.02, which is negligible and likely falls within the range of normal variation due to hyperparameter tuning.

For ANN, the average difference between casing strategies is even smaller, approximately 0.009. Interestingly, at n-gram size $N = 5$, the trend reverses slightly, with original-case texts performing better, as further illustrated in Figure 4.

These findings suggest that letter casing has no consistent or substantial impact on classification performance across the evaluated methods. The minor differences observed are unlikely to affect practical outcomes or model selection decisions.

3.7.2. *The Impact of N-gram Size on Classification Performance.* Figure 2 presents the variation in of Macro-Accuracy (MAcc) with respect to the N-gram size parameter k (ranging from 2 to 5) for each of the six classification methods: SVM, LR, k-NN, DT, RF, and ANN.

It is important to note that in Figures 2e and 2f, the y-axis scale is adjusted to focus on the observed variations in MAcc, rather than spanning the full range from 0 to 1. This scaling facilitates a clearer visualization of performance trends as k changes.

A noticeable decline in MAcc is observed for RF (Figure 2e) as the N-gram size increases. Conversely, ANN shows a general upward trend in MAcc with larger N-gram dimensions. For SVM, the highest MAcc is achieved at $k = 4$, while k-NN attains its peak performance at $k = 2$. The remaining methods do not exhibit a clear monotonic trend in MAcc relative to the N-gram size.

3.7.3. *Best Results.* ANN achieved the best overall performance, with average accuracy and macro-accuracy (MAcc) values exceeding 0.9, as summarized in Table 2. Tables 3 and 4 present the average accuracy and MAcc for each model across N-gram sizes ($N = 2$ to 5), reported separately for original-case and lowercase texts.

Model	**Macro-Accuracy (avg)**	**Accuracy (avg)**
SVM	0.761	0.762
LR	0.701	0.725
k-NN	0.608	0.590
DT	0.632	0.631
RF	0.874	0.884
ANN	0.934	0.935

Table 2. The average of Macro-Accuracy and Accuracy scores obtained for all six classification methods (SVM, LR, k-NN, DT, RF, ANN).

ANN consistently outperformed other models, except at $k = 2$, where RF achieved superior results. This trend is illustrated graphically in Figure 3, which visualizes the MAcc values from the aforementioned tables.

For ANN, the highest performance was observed at $k = 4$ using original-case letters, and at $k = 5$ using lowercase letters—with lowercase slightly outperforming original case by 0.003 in MAcc. Conversely, RF achieved the best results at $k = 2$, with average macro-accuracy values of 0.914 (original case) and 0.919 (lowercase). Examining RF results further (Figures 2e) we observe a slight decrease in average accuracy as k increases, although MAcc values across different k are not clearly separated.

Notably, ANN achieved perfect classification (accuracy = 1.0) in one of the test runs. This raises the question of whether such perfect accuracy reflects a consistent pattern for that particular train-test split, or if it is an isolated occurrence. To address this, we conducted an extended series of 15 experiments using different random seeds:

$\{7, 17, 29, 31, 37, 41, 42, 43, 47, 53, 59, 67, 83, 101, 137\}$

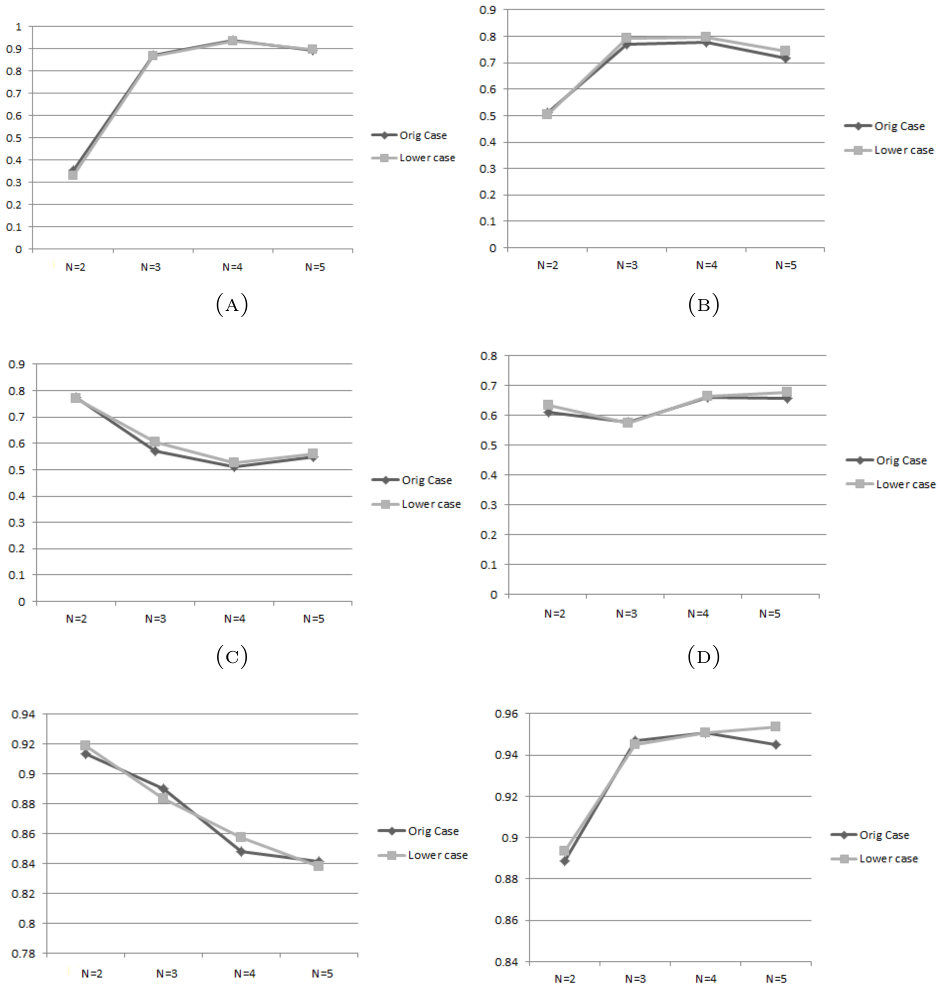
for both original-case and lowercase texts, for that specific split . As shown in Figure 4, perfect accuracy was reached in four cases: twice for $random\_state = 42$ (in both casing conditions) and two additional times for original-case texts.

Model		2-gram	3-gram	4-gram	5-gram
SVM	MAcc avg	0.328	0.869	0.934	0.899
	Acc avg	0.309	0.874	0.933	0.911
LR	MAcc avg	0.502	0.791	0.795	0.744
	Acc avg	0.531	0.807	0.812	0.770
k-NN	MAcc avg	0.772	0.603	0.526	0.559
	Acc avg	0.765	0.589	0.502	0.535
DT	MAcc avg	0.635	0.575	0.666	0.676
	Acc avg	0.633	0.581	0.667	0.675
RF	MAcc avg	0.919	0.883	0.857	0.838
	Acc avg	0.916	0.893	0.869	0.857
ANN	MAcc avg	0.894	0.945	0.951	0.954
	Acc avg	0.899	0.944	0.951	0.952

TABLE 3. The Macro-Accuracy (MAcc) and Accuracy (Acc) scores obtained for different N-gram sizes ($N = 2$ to 5), using fully lowercased text, for all six classification methods (SVM, LR, k-NN, DT, RF, ANN). Values shown represent the average of the macro-accuracy scores computed over multiple experimental runs.

Model		2-gram	3-gram	4-gram	5-gram
SVM	MAcc avg	0.356	0.872	0.938	0.894
	Acc avg	0.348	0.877	0.938	0.909
LR	MAcc avg	0.510	0.771	0.777	0.719
	Acc avg	0.541	0.793	0.798	0.748
k-NN	MAcc avg	0.755	0.572	0.511	0.550
	Acc avg	0.768	0.550	0.483	0.526
DT	MAcc avg	0.611	0.577	0.660	0.657
	Acc avg	0.603	0.575	0.662	0.655
RF	MAcc avg	0.914	0.890	0.848	0.841
	Acc avg	0.912	0.898	0.863	0.863
ANN	MAcc avg	0.889	0.947	0.951	0.945
	Acc avg	0.895	0.943	0.950	0.948

TABLE 4. The Macro-Accuracy (MAcc) and Accuracy (Acc) scores obtained for different N-gram sizes ($N = 2$ to 5), using original case text, for all six classification methods (SVM, LR, k-NN, DT, RF, ANN). Values shown represent the average of the macro-accuracy scores computed over multiple experimental runs.



(A)

(B)

(C)

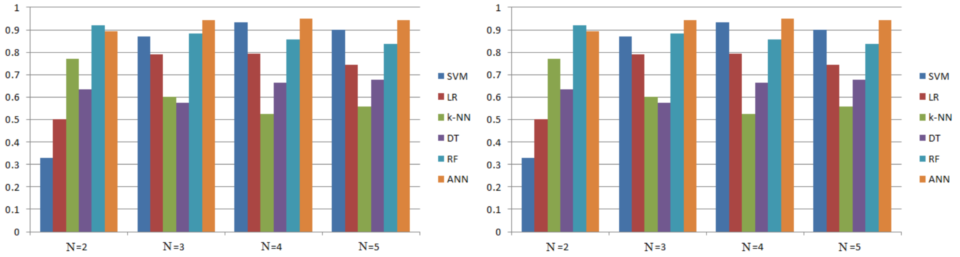
(D)

(E)

(F)

Figure 2. Variation of MAcc across different N-gram sizes ($N = 2$ to 5) for all six classification methods: (A) SVM, (B) LR, (C) k-NN, (D) DT, (E) RF, and (F) ANN. This analysis examines how varying the N-gram size affects the models' performance. Values shown represent the average of the macro-accuracy scores computed over multiple experimental runs.

The standard deviation of MAcc across these runs was approximately 0.02 (0.02039 for lowercase, 0.01938 for original case, and 0.02040 overall). No clear pattern emerged relating accuracy variation to the random seed. However,



(A)

(B)

Figure 3. Comparison of Macro-Accuracy (MAcc) scores for N-gram with $N = 2$ to 5, for original case text (A), and fully lowercased text (B) and for all six classification methods (SVM, LR, k-NN, DT, RF, ANN). Values shown represent the average of the macro-accuracy scores computed over multiple experimental runs.

original-case texts tended to perform slightly better on average, with a 0.01 higher MAcc (0.974 vs. 0.962 for lowercase), reversing the trend observed in Tables 3 and 4. Given the small magnitude of these differences, and the reversed trend in the second set of experiments for ANN, we conclude that casing (original vs. lowercase) does not have a significant impact on classification performance.

3.8. **Comparison with Existing Results.** In the literature, the best reported accuracies for authorship attribution often exceed 95%. For example, Posadas et al. [16] combined word N-grams with the Doc2Vec method, achieving over 98% accuracy on some test sets. Similarly, Zhang et al. [25] explored character-level convolutional neural networks (CNNs) and reported a minimal error rate of 1.31% when using N-grams.

Reported accuracies vary widely depending on the dataset, feature selection, and classification methods, typically ranging from approximately 70% upwards.

Authorship attribution for Romanian is still relatively underexplored, although recent interest has been increasing. Notably, Nițu et al. [15] reported an F1-score of 0.87 on a 19-author dataset, improving to 0.95 on ROST, the same dataset used in our study. Also on ROST, Avram et al. [3] achieved a lowest overall error rate of 20.40% across several machine learning models, while Avram et al. [2] reported macro-accuracy up to 87% using a Romanian pretrained BERT model.

Our approach, employing an Artificial Neural Network (ANN), achieved an initial average macro-accuracy of 0.935 across five data splits, with one split

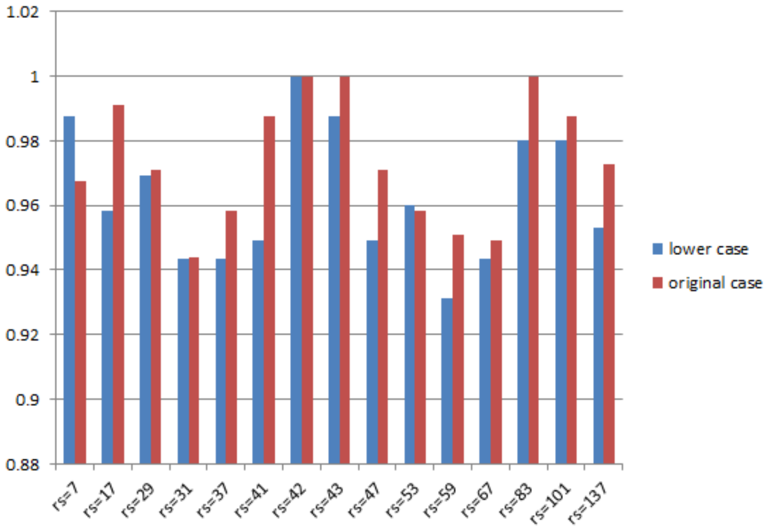


Figure 4. The Macro-Accuracy (MAcc) scores corresponding to each of the 15 experimental runs of the Artificial Neural Network (ANN) and for the one split that produced the highest classification accuracy in the previous experiments. **rs** denotes the *random_state* value used to initialize the ANN and shows the 15 distinct parameter settings.

yielding perfect classification (macro-accuracy = 1.0). To evaluate the consistency of this result, we extended the analysis to 15 additional runs with randomized seeds (Section 3.7.3). For original-case texts, the expanded trials demonstrated a mean macro-accuracy of 0.974 (with a standard deviation $\sigma \approx 0.02$), aligning with the 0.979 baseline from initial testing on the same split. Perfect classification occurred in four runs: once under lowercase transformation and three times with original casing. Notably, random seed 42 produced perfect classification under both text conditions.

These results demonstrate that our lightweight, character N-gram based approach with ANN matches or surpasses existing benchmarks for Romanian authorship attribution, offering a competitive alternative to more complex models such as BERT or transformer-based approaches.

4. Conclusions and Future Work

In this paper, we addressed the authorship attribution problem using a Romanian dataset previously employed in [2] and [3]. To our knowledge, this is the first study to apply character N-gram based methods for Romanian authorship attribution.

We evaluated six machine learning techniques: Support Vector Machine (SVM), Logistic Regression (LR), k-Nearest Neighbor (k-NN), Decision Trees (DT), Random Forest (RF), and Artificial Neural Networks (ANN). Among these, the ANN model achieved the best performance, including perfect classification in four out of thirty runs for 5-gram features.

For future work, we plan to extend our research in the direction of identifying significant character-level stylistic features. We also plan to extend our investigation by incorporating additional feature types such as word N-grams, part-of-speech tags, and other linguistic markers. These enhancements aim to further improve attribution accuracy. Given the good evaluation scores reported in some papers over the collection of texts we used, we also plan to develop a more challenging benchmark for Romanian and with a higher real-life relevance.

References

[1] Altman, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician 46*, 3 (1992), 175–185.

[2] Avram, S.-M. Bert-based authorship attribution on the romanian dataset called rost. *arXiv preprint arXiv:2301.12500* (2023).

[3] Avram, S.-M., and Oltean, M. A comparison of several ai techniques for authorship attribution on romanian texts. *Mathematics 10*, 23 (2022), 4589.

[4] Boser, B. E., Guyon, I. M., and Vapnik, V. N. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory* (1992), pp. 144–152.

[5] Dinu, L. P., Popescu, M., and Dinu, A. Authorship identification of romanian texts with controversial paternity. In *LREC* (2008).

[6] Fix, E., and Hodges, J. J. Discriminatory analysis: Non-parametric discrimination: Consistency properties. Tech. rep., USAF School of Aviation Medicine, 1951.

[7] Fix, E., and Hodges, J. J. Discriminatory analysis: Non-parametric discrimination: Small sample performance. Tech. rep., USAF School of Aviation Medicine, 1952.

[8] Houvardas, J., and Stamatatos, E. N-gram feature selection for authorship identification. In *Artificial Intelligence: Methodology, Systems, Applications* (2006).

[9] Howedi, F., and Mohd, M. Text classification for authorship attribution using naive bayes classifier with limited training data. *computer engineering and intelligent systems 5*, 4 (2014), 48–56.

[10] Kestemont, M. Function words in authorship attribution. from black magic to theory? In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL)* (2014), pp. 59–66.

[11] Kestemont, M., Tschuggnall, M., Stamatatos, E., Daelemans, W., Specht, G., Stein, B., and Potthast, M. Overview of the author identification task at pan-2018: cross-domain authorship attribution and style change detection. In *Working Notes Papers of the CLEF 2018 Evaluation Labs. Avignon, France, September 10-14, 2018/Cappellato, Linda [edit.]; et al.* (2018), pp. 1–25.

[12] Koppel, M., Schler, J., and Argamon, S. Computational methods in authorship attribution. *Journal of the American Society for information Science and Technology 60*, 1 (2009), 9–26.

[13] Misini, A., Canhasi, E., Kadriu, A., and Fetahi, E. Automatic authorship attribution in albanian texts. *Plos one 19*, 10 (2024), e0310057.

[14] Neal, T., Sundararajan, K., Fatima, A., Yan, Y., Xiang, Y., and Woodard, D. Surveying stylometry techniques and applications. *ACM Computing Surveys (CSuR) 50*, 6 (2017), 1–36.

[15] Nitu, M., and Dascalu, M. Authorship attribution in less-resourced languages: A hybrid transformer approach for romanian. *Applied Sciences 14*, 7 (2024), 2700.

[16] Posadas Durán, J., Gomez Adorno, H., Sidorov, G., Batyrshin, I., Pinto, D., and Chanona-Hernández, L. Application of the distributed document representation in the authorship attribution task for small corpora. *Soft Computing 21* (02 2017).

[17] Potthast, M., Barrón-Cedeno, A., Stein, B., and Rosso, P. Cross-language plagiarism detection. *Language Resources and Evaluation 45* (2011), 45–62.

[18] Quinlan, J. R. Induction of decision trees. *Machine learning 1*, 1 (1986), 81–106.

[19] Ramezani, R., Sheydaei, N., and Kahani, M. Evaluating the effects of textual features on authorship attribution accuracy. In *ICCKE 2013* (2013), pp. 108–113.

[20] Sapkota, U., Bethard, S., Montes, M., and Solorio, T. Not all character n-grams are created equal: A study in authorship attribution. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Human language technologies* (2015), pp. 93–102.

[21] Sari, Y., Vlachos, A., and Stevenson, M. Continuous n-gram representations for authorship attribution. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers* (Valencia, Spain, Apr. 2017), M. Lapata, P. Blunsom, and A. Koller, Eds., Association for Computational Linguistics, pp. 267–273.

[22] Stamatatos, E. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology 60*, 3 (2009), 538–556.

[23] Stamatatos, E. On the robustness of authorship attribution based on character n-gram features. *Journal of Law and Policy 21*, 2 (01 2013), 421–439.

[24] Wanwan, Z., and Jin, M. A review on authorship attribution in text mining. *Wiley Interdisciplinary Reviews: Computational Statistics 15* (04 2022).

[25] Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 649–657.

[26] Zurada, J. M. Introduction to artificial neural systems, 1992.

Department of Computer Science, Babeş-Bolyai University, Cluj-Napoca, Romania

Email address: `dana.lupsa@ubbcluj.ro`
Email address: `sanda.avram@ubbcluj.ro`
Email address: `radu.lupsa@ubbcluj.ro`