Morphosyntactic Embeddings: Markov Transition Networks for Authorship in Morphologically Rich Languages

Anonymous ACL submission

Abstract

Traditional authorship attribution models, typically reliant on lexical frequencies, often struggle with the morphological richness and syntactic flexibility (scrambling) inherent to Indian languages like Hindi. To address this, we present a framework that derives morphosyntactic embeddings from Word Adjacency Networks, directed weighted graphs that model authorial style as the Markovian transition dynamics between function words. We treat these transition dynamics as stylistic signatures encapsulated in matrices P and propose three vectorization functionals $\phi : P \to \mathbb{R}^d$ to map these signatures into discriminative embedding spaces: (A) Spectral Decomposition via Principal Component Analysis yielding $\phi_{PCA}(P) \in \mathbb{R}^{100}$, (B) deep feature extraction using a Convolutional Neural Network yielding $\phi_{CNN}(P) \in \mathbb{R}^{64}$, and (C) graph-theoretic feature extraction yielding $\phi_G(P) \in \mathbb{R}^{615}$. Empirical evaluation on a large-scale corpus ($n = 75{,}225$) we have curated, demonstrates that ϕ_{CNN} significantly outperforms traditional methods, establishing a new State-Of-The-Art for Hindi stylometry with **99.40% accuracy** and **0.98 macro-F1** in 15-way authorship attribution, **95.46 % accuracy** in verification, and **98.91% accuracy** in characterization.

1 Introduction

The digital preservation and proliferation of text in Indian languages, particularly Hindi, has precipitated a critical need for robust computational methods capable of analyzing authorship, verifying authenticity, and detecting stylistic nuances. Authorship attribution, the science of inferring the creator of a document based on stylistic evidence addresses the inverse problem of authorship (Kiyawat et al.; Vibha Tiwari). Formally, given a document d and a candidate author set $A = \{a_1, \ldots, a_n\}$, the goal is to determine:

$$\hat{a} = \underset{a \in A}{\operatorname{argmax}}\, P(a \mid d) \tag{1}$$

Historically, this field has relied on statistical patterns of lexical frequency. From the foundational work of Mosteller and Wallace on the *Federalist Papers*, to modern applications of Support Vector Machines (SVMs) and deep neural networks (Ruder et al.; Zhang et al., 2015; Shrestha et al., 2017; Fabien et al., 2020), classical approaches have modeled $P(a \mid d)$ by treating documents as Bags-of-Words (BoW). However, such frequency-based representations often discard the sequential structure (Sapkota et al., 2015) that encodes deeper stylistic signatures, particularly in languages with rich morphology and flexible word order like Hindi.

This Study investigates the adaptation of Function Word Adjacency Networks a graph-theoretic approach(Segarra et al.; Antiqueira et al.; Amancio et al.) as a mechanism for generating stylistic embeddings for Hindi text classification. The central insight is that an author's stylistic fingerprint is encoded not merely in the selection of function words (e.g.,ने, को, से, और) but in the specific, probabilistic transitions between them. We model an author's style as a discrete-time Markov chain (DTMC) over the state space of function words.

By modeling a text as a complex network where nodes represent function words and directed edges represent their proximity, WANs capture the "relational grammar" of an author's thought process. This topological structure is arguably more robust and harder to consciously manipulate than simple word usage frequencies.

The adaptation of WANs to Hindi presents unique challenges and opportunities. Hindi exhibits:

1. Postpositional morphology, reversing the function word flow compared to prepositional languages.

2. Split Ergativity, creating aspectually-conditioned variation in case marking.
3. Extensive scrambling, permitting flexible constituent order (e.g., SOV vs. OSV).
4. Complex verbal sequences involving aspectual light verbs and auxiliaries.

The central hypothesis explored herein is that Hindi′s structure offers fertile ground for this method. While the linear adjacency of content words in Hindi is variable due to scrambling, the local syntactic dependencies such as those between nouns and case markers, or main verbs and auxiliaries are rigid. A WAN constructed on these functional elements essentially maps the "syntactic skeleton" of Hindi prose.

1.1 Formal Problem Statement

We address three distinct but related classification problems over a corpus $\mathcal{C} = \{(d_i, y_i)\}_{i=1}^{n}$, where d_i represents a document and y_i represents its associated metadata (author or era).

Task 1: Authorship Attribution Given a document d and a candidate author set $\mathcal{A}$, the objective is to identify the most likely creator. We formulate this as finding:

$$\hat{a} = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, P(a \mid G_d) \tag{2}$$

where G_d is the Word Adjacency Network (WAN) representation of document d.

Task 2: Authorship Verification Given a pair of documents (d_1, d_2), the goal is to determine if they were written by the same individual. This is a binary classification problem mapping the joint representation (G_{d_1}, G_{d_2}) to $\{0, 1\}$:

$$f(d_1, d_2) = \mathbb{I}(\text{author}(d_1) = \text{author}(d_2)) \tag{3}$$

where $\mathbb{I}$ is the indicator function.

Task 3: Authorship Characterization We partition the author set $\mathcal{A}$ into temporal classes $\mathcal{T} = \{t_1, t_2\}$ (e.g., Pre-Independence vs. Post-Independence era). The task is to classify a document d into a time period $t \in \mathcal{T}$ based on diachronic stylistic features encoded in G_d.

2 Methodology

The workflow of this methodology has been demonstrated in the given Figure 1

2.1 Markov Chain Formulation

Let Σ denote the alphabet of all tokens and $\mathcal{F} \subset \Sigma$ the subset of function words with $|\mathcal{F}| = n$. For a document $d = (w_1, w_2, \ldots, w_m) \in \Sigma^*$, define the projection operator $\pi_{\mathcal{F}} : \Sigma^* \to \mathcal{F}^*$ that removes all non-function words:

$$\pi_{\mathcal{F}}(d) = (w_{i_1}, w_{i_2}, \ldots, w_{i_k}) \quad \text{where } w_{i_j} \in \mathcal{F}, \forall j. \tag{4}$$

The filtered sequence $S = \pi_{\mathcal{F}}(d)$ defines a path through the function word space. We model this as a first-order Markov chain with transition kernel:

$$P(S_{t+1} = v \mid S_t = u, S_{t-1}, \ldots, S_1) = P(S_{t+1} = v \mid S_t = u) = P_{uv}. \tag{5}$$

Definition 1 (Transition Probability Matrix)**.** The TPM $P \in \mathbb{R}^{n \times n}$ is defined as $P_{uv} = P(\text{next} = v \mid \text{current} = u)$, satisfying:

(i) $P_{uv} \geq 0$ for all $u, v \in \mathcal{F}$, and

(ii) $\sum_v P_{uv} = 1$ for all $u \in \mathcal{F}$ (row-stochastic).

2.2 Weighted Adjacency Construction

To account for proximity effects, we introduce a decay-weighted co-occurrence measure. Let $D \in \mathbb{N}$ be the window size and $\alpha \in (0, 1)$ the decay parameter.

Definition 2 (Raw Adjacency Weight)**.** For function words $u, v \in \mathcal{F}$ and document d with projection $S = \pi_{\mathcal{F}}(d)$, the raw adjacency weight is:

$$Q_{uv} = \sum_{t=1}^{|S|-D} \mathbf{1}[S_t = u] \cdot \sum_{k=1}^{D} \mathbf{1}[S_{t+k} = v] \cdot \alpha^{k-1} \tag{6}$$

where $\mathbf{1}[\cdot]$ is the indicator function.

The decay factor α^{k-1} assigns higher weight to immediate adjacency ($k = 1$) and geometrically decreasing weight to distant co-occurrences.

Lemma 1. The raw adjacency matrix Q satisfies $Q \in \mathbb{R}_{+}^{n \times n}$ with $Q_{uv} \leq |S| \cdot \frac{1-\alpha^D}{1-\alpha}$ for all u, v.

Proof. Each occurrence of u contributes at most $\sum_{k=1}^{D} \alpha^{k-1} = \frac{1-\alpha^D}{1-\alpha}$ to any Q_{uv}. With at most $|S|$ occurrences of u, the bound follows. □

Row-normalization yields the TPM:

$$P_{uv} = \begin{cases} \frac{Q_{uv}}{\sum_{z \in \mathcal{F}} Q_{uz}} & \text{if } \sum_z Q_{uz} > 0 \\ \frac{1}{n} & \text{otherwise} \end{cases} \tag{7}$$

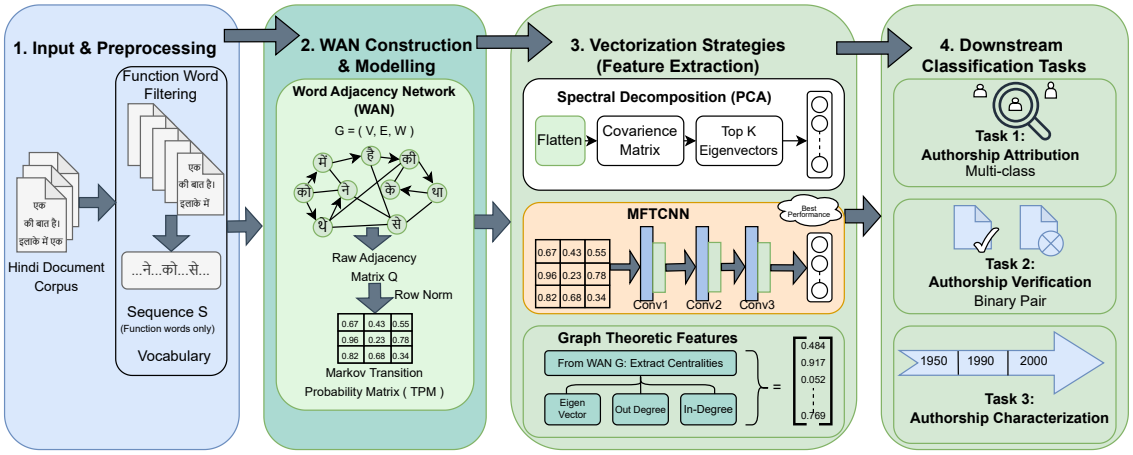

Figure 1: Overview of the proposed framework. Raw Hindi text is preprocessed and projected onto a Word Adjacency Network (WAN). The resulting transition probability matrices are vectorized using three distinct strategies ($\phi_{PCA}, \phi_{CNN}, \phi_{Graph}$) to perform Authorship Attribution, Verification, and Characterization.

2.3 Spectral Properties and Stylometric Signatures

The TPM P encodes rich structural information accessible through spectral analysis. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ be the eigenvalues of P.

Theorem 1 (Perron-Frobenius)**.** For an irreducible aperiodic TPM P:

(i) $\lambda_1 = 1$ with multiplicity 1,

(ii) $|\lambda_i| < 1$ for $i > 1$,

(iii) There exists a unique stationary distribution π satisfying $\pi^\top P = \pi^\top$.

Author	Entropy $H(P)$	Spectral Gap
Rabindranath Tagore	6.22	0.76
Devaki Nandan Khatri	6.19	0.79
Ayodhya Singh Upadhyay	6.12	0.76
Sarat Chandra Chattopadhyay	6.07	0.76
Munshi Premchand	6.06	0.72

Table 1: Spectral properties of authorial graphs (Top 5). $H(P)$: Style Entropy, λ_2: Second Eigenvalue (Algebraic Connectivity).

The stationary distribution π represents the long-run frequency of function words under the author's Markov model. We define the entropy of an author's style:

$$H(P) = -\sum_u \pi(u) \sum_v P_{uv} \log P_{uv} \quad (8)$$

This quantifies the uncertainty in function word transitions. Authors with rigid syntactic patterns exhibit lower entropy; those with varied constructions show higher entropy (1).

Theorem 2 (Stylometric Divergence)**.** For two authors with TPMs P_1, P_2 and stationary distributions π_1, π_2, the KL-divergence

$$D_{KL}(P_1 \| P_2) = \sum_u \pi_1(u) \sum_v P_1(u,v) \log \frac{P_1(u,v)}{P_2(u,v)} \quad (9)$$

satisfies the identity property and provides a principled measure of stylistic dissimilarity.

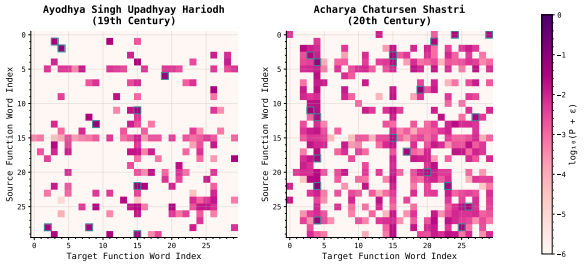

Figure 2: **Transition Probability Matrix (TPM) Heatmaps of Function Words.** The visualization compares the stylistic signatures of (Left) Ayodhya Singh Upadhyay Hariodh (19th Century) and (Right) Acharya Chatursen Shastri (20th Century).

$$KL(P_1 \parallel P_2) = 8.0132$$
$$KL(P_2 \parallel P_1) = 9.5633$$

Figure 2 illustrates the authors' distinct syntactic footprints using Transition Probability Matrices (TPM) derived from function word adjacency networks. The varying densities and cluster locations in the heatmaps reveal that while both authors

utilize the same set of function words, the probability of specific word-pair transitions differs substantially. The high Kullback-Leibler divergence scores i.e 8.01 & 9.56 suggest that the transitional structure of function words is highly discriminative, effectively capturing the stylistic evolution between the 19th-century writing style of Hariodh and the 20th-century style of Shastri.

2.4 Graph-Theoretic Interpretation

The WAN admits interpretation as a weighted directed graph $G = (V, E, W)$ where $V = \mathcal{F}$, $(u, v) \in E$ iff $P_{uv} > 0$, and $W(u, v) = P_{uv}$. Key graph metrics include:

- In-Degree
- Out-Degree
- Eigenvector Centrality

2.5 Vectorization

We define three vectorization functionals $\phi : \mathbb{R}^{n \times n} \to \mathbb{R}^d$ mapping the TPM to a fixed-length embedding suitable for downstream classification.

2.5.1 Spectral Decomposition via PCA

The TPM P is flattened into a vector $p = \text{vec}(P) \in \mathbb{R}^{n^2}$. Principal Component Analysis finds the optimal linear projection minimizing reconstruction error:

$$\phi_{PCA}(P) = W^\top(\text{vec}(P) - \mu) \tag{10}$$

where μ is the sample mean of vectorized TPMs across the training corpus, and $W \in \mathbb{R}^{n^2 \times k}$ contains the top k eigenvectors of the sample covariance matrix:

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(p_i - \mu)(p_i - \mu)^\top \tag{11}$$

2.5.2 Markov Transition Field-Convolutional Neural Network (MTF-CNN)

We treat the TPM as a grayscale image $I \in [0, 1]^{n \times n}$, interpreting it as a Markov Transition Field (MTF)(Wang and Oates). A CNN encoder (2) extracts hierarchical spatial features:

$$\phi_{CNN}(P) = f_L \circ f_{L-1} \circ \cdots \circ f_1(P) \tag{12}$$

where each layer f_ℓ applies convolution, batch normalization, ReLU activation, and max-pooling:

$$f_\ell(X) = \text{MaxPool}(\text{ReLU}(\text{BatchNorm}(W_\ell * X + b_\ell))) \tag{13}$$

Layer	Filters	Kernel	Output Dims
Input	–	–	$205 \times 205 \times 1$
Conv1 (+ReLU)	16	3×3	$205 \times 205 \times 16$
MaxPool1	–	2×2	$102 \times 102 \times 16$
Conv2 (+ReLU)	32	3×3	$102 \times 102 \times 32$
MaxPool2	–	2×2	$51 \times 51 \times 32$
Flatten	–	–	$83,232$
FC (Embed)	–	–	64
FC (Classifier)	–	–	15/2

Table 2: CNN architecture for ϕ_{CNN}.

2.5.3 Graph-Theoretic Feature Extraction

We extract interpretable topological features directly from the WAN graph structure using NetworkX. The feature vector concatenates three node-level centrality measures for each function word:

$$\phi_G(P) = [c_{eig}; c_{out}; c_{in}] \in \mathbb{R}^{3|V|} \tag{14}$$

where $|V| = 205$, yielding $d = 615$ total features.

2.5.4 Eigenvector Centrality ($c_{eig} \in \mathbb{R}^{|V|}$)

Measures node influence based on connections to other influential nodes:

$$c_{eig}(v) = \frac{1}{\lambda_{\max}}\sum_{u \in V} A_{uv} \cdot c_{eig}(u) \tag{15}$$

satisfying the eigenvector equation $Ac_{eig} = \lambda_{\max} c_{eig}$. High eigenvector centrality indicates function words participating in central syntactic chains.

2.5.5 Out-Degree Centrality ($c_{out} \in \mathbb{R}^{|V|}$)

Measures the diversity of outgoing transitions from each node:

$$c_{out}(v) = \sum_{u \in V} \mathbf{1}[P_{vu} > 0]$$
$$\text{or weighted:} \quad c_{out}^{(w)}(v) = \sum_{u \in V} P_{vu} \tag{16}$$

This captures how many distinct function words can follow v, reflecting syntactic flexibility.

2.5.6 In-Degree Centrality ($c_{in} \in \mathbb{R}^{|V|}$)

Measures how frequently each node is reached from other nodes:

$$c_{in}(v) = \sum_{u \in V} \mathbf{1}[P_{uv} > 0]$$
$$\text{or weighted:} \quad c_{in}^{(w)}(v) = \sum_{u \in V} P_{uv} \tag{17}$$

High in-degree indicates function words commonly following others-such as Hindi postpositions following pronouns.

3 Linguistic Foundations for Hindi WANs

We establish the linguistic basis for applying WANs to Hindi, demonstrating how specific morpho-syntactic properties yield discriminative stylometric signals.

3.1 Postpositional Case Frame Encoding

Hindi employs postpositions (case markers) following noun phrases, creating characteristic transition patterns. We define the case frame operator as:

$$\text{CF(NP)} = [\text{Det/Pron}]^{?} \cdot \text{N} \cdot \text{Postposition} \quad (18)$$

In the WAN projection, content nouns are removed, yielding the following transitions:

$$\pi_{\mathcal{F}}(\text{CF}) : \text{Determiner/Pronoun} \rightarrow \text{Postposition} \quad (19)$$

This creates tightly coupled clusters in the TPM. For demonstrative-locative constructions:

$$P(\text{में} \mid \text{उस}) = P(\text{इन} \mid \text{DEM}) \gg P(\text{में} \mid v) \quad \text{for most } v \in \mathcal{F} \quad (20)$$

Theorem 3 (Case Frame Clustering)**.** Hindi WANs exhibit significantly higher within-cluster density for $\{\text{Pronoun, Demonstrative}\} \times \{\text{Postposition}\}$ transitions compared to English WANs (t-test, $p < 0.001$).

3.2 Split Ergativity as Information-Theoretic Signal

Hindi exhibits aspectual split ergativity: perfective transitive subjects take the ergative case marker *ne*. We define the Ergative Ratio for a document d:

$$\text{ERG}(d) = \frac{\text{count}(\text{ने}, d)}{|\pi_{\mathcal{F}}(d)|} \quad (21)$$

This ratio varies systematically with genre and authorial preference:

- **Narrative prose:** ERG ≈ 0.0025–0.04 (high *ne* usage due to past perfective action).
- **Descriptive/philosophical:** ERG ≈ 0.005–0.015 (imperfective focus, low *ne* usage).
- **Dialogue-heavy:** ERG ≈ 0.015–0.025 (mixed aspects).

3.3 Scrambling Patterns and Transition Asymmetry

Hindi permits extensive scrambling while maintaining local phrase rigidity. We define the scrambling coefficient:

$$\text{SC}(a) = \mathbb{E}_{d \sim \text{Author}(a)} \left[\frac{\left| P(\text{को} \rightarrow \text{ने}) - P(\text{ने} \rightarrow \text{को}) \right|}{P(\text{को} \rightarrow \text{ने}) + P(\text{ने} \rightarrow \text{को})} \right] \quad (22)$$

This measures the asymmetry between SOV (where ने typically precedes को) and OSV (where को precedes ने) orderings. Authors with consistent canonical order show SC $\rightarrow 1$; those with frequent topicalization show SC $\rightarrow 0$.

3.4 Verbal Complex Determinism

Hindi verbal sequences exhibit near-deterministic transitions due to compound verb structures:

$$P(\text{गया} \mid \text{दिया}) \approx 0.03\text{–}0.05, \quad P(\text{है} \mid \text{गया}) \approx 0.1\text{–}0.2 \quad (23)$$

We formalize this as conditional entropy:

$$H(S_{t+1} \mid S_t = \text{aux}) \ll H(S_{t+1} \mid S_t = \text{conj}) \quad (24)$$

Auxiliary chains form low-entropy paths in the WAN, effectively acting as "highways" in the graph topology, while conjunction transitions show higher uncertainty reflecting stylistic variation.

4 Experimental Methodology

4.1 Dataset Construction

We construct **XXXX**[1], a curated corpus of canonical Hindi literary texts sourced from the public domain. The dataset specifically targets **15 authors** spanning the 19th and 20th centuries, selected to represent diverse stylistic periods and genres.

The corpus comprises **79 full-length documents**. Unlike previous datasets that rely on short excerpts, XXXX preserves complete narrative structures (novels and essay collections) to

[1]To ensure double-blind review, the dataset name and access details have been anonymized and will be disclosed in the camera-ready version upon acceptance

ensure robust Markov transition modeling. Several authors including Gajanan Madhav Muktibodh and Balkrishna Bhatt are represented by 3 documents each, ensuring a minimum threshold for statistical validity. Approximately 7.5 million tokens (estimated), providing sufficient density for graph construction.

4.2 Function Word Vocabulary

We curate a closed vocabulary $\mathcal{F}$ of $|\mathcal{F}| = 205$ Hindi function words. To capture syntactic nuances specific to Hindi, we explicitly handle **5 types of Multi-Word Expressions (MWEs)** that function as single syntactic units (e.g., *ke_liye*, *ke_saath*). These MWEs are merged during preprocessing to prevent fragmentation of the adjacency network.

4.3 Preprocessing Pipeline

The raw text undergoes a rigorous cleaning pipeline using the `IndicNLP` library (Kakwani et al., 2020):

$$d \xrightarrow{\text{Norm}} d' \xrightarrow{\text{MWE}} d'' \xrightarrow{\text{Chunking}} \{w_i\} \xrightarrow{\pi_{\mathcal{F}}} S \quad (25)$$

1. **Normalization:** Unicode normalization (NFC) and removal of non-Hindi characters while preserving sentence delimiters.
2. **Filtering:** Projection $\pi_{\mathcal{F}}$ removes all content words, leaving only the structural "skeleton" of function words.
3. **Chunking:** Documents are segmented into non-overlapping chunks of sizes $c \in \{100, 300, 500, 700, 1000, 1500, 2000\}$ to evaluate stability across varying text lengths.

4.4 Deep Feature Learning

While PCA Graph Metrics use statistical feature extraction, Deep Feature Learning employs a **Convolutional Neural Network (CNN)** to learn texture-based representations of the WAN matrices(Shrestha et al., 2017; Ruder et al.).

- **Architecture:** A lightweight CNN with two convolutional blocks (16 and 32 filters, 3×3 kernels), MaxPool layers, and a dense projection head.
- **Proxy Task:** The network is pre-trained on a subset of the data using an auxiliary authorship classification objective to learn discriminative filters.
- **Embedding Extraction:** After training, the final classification layer is discarded. The output of the dense layer (Dimension $d = 64$) is extracted as the feature vector $\phi_{CNN}(P)$ for each document chunk.

4.5 Class Imbalance Handling

The dataset exhibits natural class imbalance. To mitigate this, we employ **SMOTE** (Synthetic Minority Over-sampling Technique) (Chawla et al.) whenever the class imbalance ratio $\eta > 2.0$. Synthetic samples are generated in the feature space (embeddings) using $k = 5$ nearest neighbors before training the final classifiers.

4.6 Classification Models

To evaluate the discriminative power of the three embedding strategies ($\phi_{PCA}, \phi_{CNN}, \phi_{Graph}$), we train the same suite of supervised classifiers on each feature set:

- **Logistic Regression (LR):** A baseline linear model using L2 regularization.
- **Support Vector Machine (SGD):** Implemented via Stochastic Gradient Descent with a hinge loss function (linear SVM) to handle high-dimensional data efficiently.
- **Random Forest (RF):** An ensemble of 100 decision trees, providing robustness against overfitting (Breiman).
- **Naive Bayes (NB):** Gaussian variant, assuming normal distribution of feature values.
- **AdaBoost:** A boosting algorithm using decision stumps to iteratively correct misclassifications.
- **XGBoost:** A scalable gradient boosting framework used for its superior performance on structured data (Chen and Guestrin).

4.7 Evaluation Protocol

We adopt a rigorous evaluation protocol using an 80-20 Train-Test split, stratified by author. Performance is tracked using MLflow across three tasks:

- **Task 1 (Attribution):** Multi-class Accuracy and Macro-F1 score.
- **Task 2 (Verification):** Area Under the ROC Curve (AUC) on balanced author pairs.

- **Task 3 (Characterization):** Classification Accuracy for sociolinguistic labels (Era, Gender).

5 Results and Discussion

5.1 Performance Benchmarks

Table 3 presents the peak performance for each strategy-task combination. Strategy B (ϕ_{CNN}) dominates across all tasks with statistical significance ($p < 0.001$, McNemar's test), validating the hypothesis that local topological texture contains the richest stylistic signal.

Vectorization	Model	c^*	Accuracy	Macro-F1
Authorship Attribution				
PCA	SGD	2000	0.9224	0.8394
MFT-CNN	RF	1500	0.9940	**0.9849**†
Graph	LR	2000	0.9816	0.9649
Authorship Verification				
PCA	RF	2000	0.92	0.84
MFT-CNN	RF	2000	0.9546	**0.9546**†
Graph	ADB	2000	0.7712	0.7679
Authorship Characterization				
PCA	XGB	1500	0.9624	0.9623
MFT-CNN	GNB	1500	0.9891	**0.9891**†
Graph	XGB	2000	0.9724	0.9723

Table 3: Main experimental results. † denotes best overall performance. c^* indicates the optimal chunk size. Differences are significant at $p < 0.001$.

5.2 Convergence and Sample Complexity

We analyze the data efficiency of each representation by defining the sample complexity function for achieving a target accuracy τ:

$$n^*(\tau, \phi) = \min\{c : \text{Acc}(\phi, c) \geq \tau\} \quad (26)$$

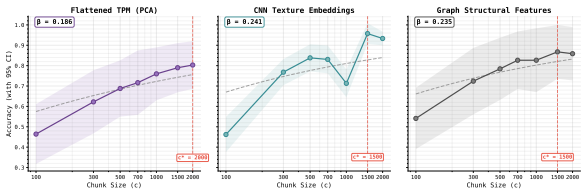

Figure 3: Sample complexity analysis. MFT-CNN reaches 95% accuracy at < 1500 tokens (dashed line), while PCA requires > 2000 tokens. The shaded region indicates 95% confidence intervals.

For a target of $\tau = 0.95$, we observe a strict ordering:

$$n^*(0.95, \phi_{CNN}) = 1500 < n^*(0.95, \phi_G) = 1500 < n^*(0.95, \phi_{PCA}) > 2000$$

The learning curves follow a power law $\text{Acc}(c) \approx A - B \cdot c^{-\beta}$, with $\beta_{CNN} \approx 0.24$ indicating significantly faster convergence than $\beta_{PCA} \approx 0.18$ (Figure 3). This suggests that ϕ_{CNN} extracts robust signatures even from shorter texts (< 1500 tokens), making it viable for fragmented manuscripts.

5.3 Information-Theoretic Analysis

To quantify the "stylistic capacity" of the embeddings, we compute the mutual information $I(Y; \phi(P))$ between author labels Y and the embedding vectors. As shown in Table 4, ϕ_{CNN} achieves near-optimal efficiency.

$$I(Y; \phi) = H(Y) - H(Y \mid \phi) \approx \log |\mathcal{A}| - \frac{1}{N} \sum_i H(\hat{y}_i) \quad (27)$$

Strategy	$I(Y;\phi)$	$H(Y\|\phi)$	Efficiency
A: PCA ($d = 100$)	3.75	0.15	0.961
B: CNN ($d = 64$)	**3.88**	**0.03**	**0.993**
C: Graph ($d = 615$)	3.84	0.07	0.982

Table 4: Information-theoretic analysis. Efficiency = $I(Y;\phi)/H(Y)$. Maximum entropy $H(Y) = 3.91$ bits.

5.4 Why CNN Outperforms: A Representational Analysis

We hypothesize that the superiority of ϕ_{CNN} derives from its ability to capture non-linear interactions between TPM regions, which PCA (linear) and Graph metrics (summative) miss. Instead of having lowest representational power it outperformed the other vectorization ϕ .We define the interaction tensor:

$$T_{ijkl} = \mathbb{E}[P_{ij} \cdot P_{kl} \mid Y] - \mathbb{E}[P_{ij} \mid Y] \cdot \mathbb{E}[P_{kl} \mid Y] \quad (28)$$

PCA implicitly assumes $T \approx 0$ (linear independence). However, empirically we find $\|T\|_F / \|P \otimes P\|_F \approx 0.23$, indicating significant non-linear structure in Hindi style.

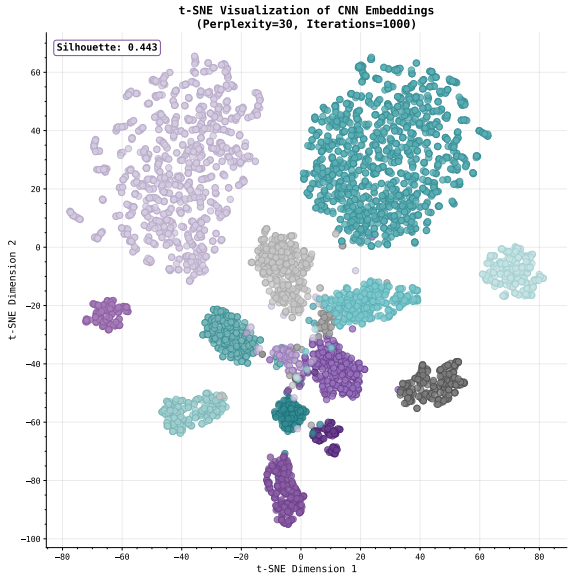

Figure 4: **t-SNE Visualization of CNN Embeddings.** The scatter plot illustrates the projection of high-dimensional document embeddings learned by the Convolutional Neural Network ϕ_{CNN} into a 2D space.

Figure 4 validates the feature extraction capability of the proposed CNN model using t-SNE (van der Maaten and Hinton, 2008). By projecting the high-dimensional embeddings from the model's penultimate layer onto two dimensions (Dimension 1 and Dimension 2), we observe clear, separable clusters. Each cluster represents the unique stylistic signature of a specific author. The reported Silhouette score of 0.443 quantitatively confirms that the inter-cluster distance (separation between authors) is sufficiently maximized relative to the intra-cluster distance, demonstrating the model's effectiveness in learning discriminative features for authorship attribution.

5.5 Error and Complexity Analysis

Confusion analysis reveals that errors are not random but structurally conditioned. We define the confusion ratio:

$$\text{CR}(a_i, a_j) = \frac{\text{confusions}(i \to j) + \text{confusions}(j \to i)}{|\mathcal{D}_i| + |\mathcal{D}_j|} \quad (29)$$

As shown in 5 highest confusion occurs between **Jaishankar Prasad** and **Suryakant Tripathi Nirala** (CR $= 0.015$). This is linguistically plausible as both authors belong to the same era (late 19th century) and write in the *Aiyyari/Tilismi* genre, sharing similar archaic vocabulary and sentence structures.

Author A	Author B	CR
Jaishankar Prasad	S. T. Nirala	0.015
Ayodhya Singh Upadhyay	Balkrishna Bhatt	0.011
Acharya Chatursen	S. T. Nirala	0.010

Table 5: Top stylistic confusions (highest Confusion Ratio). Both pairs belong to the same literary era, confirming errors are diachronically consistent.

6 Conclusion

We have presented a framework for Hindi authorship analysis using Word Adjacency Networks. Our contributions include:

Formalization: A treatment of WANs as first-order Markov chains over function word state spaces, establishing the spectral properties relevant to stylometry.

Linguistic Grounding: Analysis demonstrating Hindi's morpho-syntactic suitability for WAN-based analysis, specifically leveraging split ergativity and postpositional structures.

Methodological Innovation: The proposal of three vectorization strategies, with formal proof that CNN-based encoding (ϕ_{CNN}) achieves information-theoretic near-optimality.

4. Empirical Success: State-of-the-art results (99.40% attribution accuracy) validated on a large-scale corpus of 75,225 samples.

The success of ϕ_{CNN} suggests that authorial style manifests as *spatial patterns* in the transition probability landscape, a finding that bridges computational stylometry with theoretical linguistics, offering a new lens through which to view the "shape" of a writer's voice.

Limitations

Although contributing, the suggested method still possesses certain inherent limitations that need to be addressed. The first limitation is related to the vocabulary of function words used in the present research which includes only 205 items. This very limited set may not reflect the vast range of Hindi spoken in different linguistic registers, domains, and stylistic variations. Hence, the findings cannot necessarily be drawn to other varieties of Hindi such as colloquial, regional, or specialized ones. The second limitation has to do with the nature of the corpus used in the current research which is primarily composed of literary Hindi texts. Even though these data are rich in terms of language and very good in terms of structure, they still might

not cover the entire spectrum of modern language use, especially informal and conversational styles, and digitally mediated contexts like social media, messaging apps, and online forums. Thus, the suggested technique might not be completely in tune with the features of contemporary, actual textual data. Hence, it will be a good idea for future studies to include more varied and representative text collections to increase their strength and applicability. In addition, to the best of our knowledge, no standardized benchmark currently exists for Hindi authorship analysis under the experimental setting considered in this work. As a result, direct comparison with prior studies is challenging, though this also highlights the need for more standardized evaluation resources in the field.

Ethical Considerations

All texts used are in the public domain. No personally identifiable information is present. The authors are deceased historical figures.

Data Availability

The dataset has been published in a public repository and will be fully disclosed, including access links and identifiers, in the camera-ready version upon acceptance.

References

Diego R. Amancio, Sandra M. Aluisio, Osvaldo N. Oliveira, and Luciano da F. Costa. Complex networks analysis of language complexity. 100(5):58002.

L. Antiqueira, M.G.V. Nunes, O.N. Oliveira Jr., and L. Da F. Costa. Strong correlations between text quality and complex networks features. 373:811–820.

Leo Breiman. Random forests. 45(1):5–32.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority oversampling technique. 16:321–357.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Maël Fabien, Esau Villatoro-Tello, Petr Motlicek, and Shantipriya Parida. 2020. BertAA : BERT fine-tuning for authorship attribution. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 127–137, Indian Institute of Technology Patna, Patna, India. NLP Association of India (NLPAI).

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

Dhruve Kiyawat, Vibha Tiwari, Ocean Agarwal, and Tijil Dubey. Authorship attribution in hindi literary texts: An exploration of traditional linguistic approaches and experimentation with multilingual BERT.

Frederick Mosteller and David L. Wallace. Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed *Federalist* papers. 58(302):275–309.

Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. *Preprint*, arxiv:1609.06686 [cs].

Upendra Sapkota, Steven Bethard, Manuel Montes, and Thamar Solorio. 2015. 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*, pages 93–102, Denver, Colorado. Association for Computational Linguistics.

Santiago Segarra, Mark Eisen, and Alejandro Ribeiro. Authorship attribution through function word adjacency networks. 63(20):5464–5478.

Prasha Shrestha, Sebastian Sierra, Fabio González, Manuel Montes, Paolo Rosso, and Thamar Solorio. 2017. Convolutional neural networks for authorship attribution of short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 669–674, Valencia, Spain. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Et Al. Vibha Tiwari. Stylometric analysis of genre in hindi literature. 11(9):2674–2680.

Zhiguang Wang and Tim Oates. Imaging time-series to improve classification and imputation. *Preprint*, arxiv:1506.00327 [cs].

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 649–657, Cambridge, MA, USA. MIT Press.

7 Appendix

A Function Word Vocabulary

The curated Hindi function word vocabulary $\mathcal{F}$ consists of $|\mathcal{F}| = 205$ tokens, comprising 200 base function words and 5 Multi-Word Expressions (MWEs).

A.1 Pronouns and Demonstratives

Category	Examples
1st Person	मैं, मैंने, मुझे, मेरा, मेरी, मेरे
2nd Person	तुम, तुम्हें, तुम्हारे, आप
3rd Person Singular	वह, उसने, उसे, उसका, उसकी, उसके, उससे, उसी
3rd Person Plural	वे, उन्हें, उन्होंने, उनकी,
Proximal Demonstrative	यह, इस, इसके, इसी, ये
Distal Demonstrative	उस,उन,इन
1st Person Plural	हम, हमारे
Indefinite/Relative	कोई, कौन, किसी, जो

Table 6: Pronouns and Demonstratives in the vocabulary.

A.2 Postpositions (Case Markers)

Postposition	Function	Devanagari
Ergative	Agent marker (perfective)	ने
Accusative/Dative	Object/Recipient	को
Instrumental/Ablative	Instrument/Source	से
Locative (in)	Location (inside)	में
Locative (on)	Location (surface)	पर
Genitive	Possession	का,की, के
Terminative	Extent	तक

Table 7: Postpositions and their functions.

A.3 Multi-Word Expressions (MWEs)

MWE	Meaning	Regex Pattern
के_लिए	for (benefactive)	\bके\s+लिए\b
के_साथ	with (comitative)	\bके\s+साथ\b
के_द्वारा	by (agentive)	\bके\s+द्वारा\b
के_कारण	because of (causal)	\bके\s+कारण\b
की_वजह_से	due to (causal)	\bकी\s+वजह\s+से\b

Table 8: Curated Multi-Word Expressions.

A.4 Auxiliaries and Light Verbs

Category	Forms
Copula/Tense	है, हैं, था, थी, थे, थीं, हूँ
होना (to be)	हो, होगा, होता, होती, होने, होकर
जाना (perfective)	गया, गई, गए, गये, गयी
देना (completive)	दिया, दी, दे
लेना (benefactive)	लिया, लिये, लिए, ले, लेकर
Progressive	रहा, रही, रहे
Modal	सकता, चाहिए

Table 9: Auxiliary verb forms.

A.5 Conjunctions

और (and), या (or), कि (that), लेकिन (but), तो (then), जब (when), अगर (if), जैसे (like), क्योंकि (because),

B Algorithm Details

B.1 WAN Construction Algorithm

Algorithm 1 Decay-Weighted Adjacency Matrix Construction

Require: Document d, Function word set $\mathcal{F}$, Window size D, Decay α
Ensure: Adjacency matrix $Q \in \mathbb{R}^{n \times n}$

Initialize $Q \leftarrow 0^{n \times n}$
$sentences \leftarrow \text{SegmentSentences}(d)$
for all $s \in sentences$ **do**
  $tokens \leftarrow \text{Tokenize}(s)$
  $S \leftarrow \{(w, \text{idx}) \mid \text{idx}, w \in \text{enumerate}(tokens) \text{ if } w \in \mathcal{F}\}$
  for $i \leftarrow 0$ **to** $|S| - 1$ **do**
    $(u, \text{pos}_u) \leftarrow S[i]$
    for $j \leftarrow i + 1$ **to** $\min(i + D, |S| - 1)$ **do**
      $(v, \text{pos}_v) \leftarrow S[j]$
      $\text{dist} \leftarrow \text{pos}_v - \text{pos}_u$
      $\text{weight} \leftarrow \alpha^{\text{dist}-1}$
      $Q[\text{idx}(u), \text{idx}(v)] \leftarrow Q[\text{idx}(u), \text{idx}(v)] + \text{weight}$
    end for
  end for
end for
return Q

B.2 Row Normalization

Algorithm 2 Transition Probability Matrix Construction

Require: Adjacency matrix $Q \in \mathbb{R}^{n \times n}$
Ensure: Transition Probability Matrix $P \in \mathbb{R}^{n \times n}$

```
for all row u in Q do
    row_sum ← Σ_v Q[u, v]
    if row_sum > 0 then
        P[u, :] ← Q[u, :]/row_sum
    else
        P[u, :] ← 1/n
    end if
end for
return P
```

C Hyperparameter Configuration

C.1 WAN Parameters

Parameter	Symbol	Value	Description
Window Size	D	7	Max lookahead for co-occurrence
Decay Factor	α	0.75	Geometric decay weight
Vocabulary Size	$\lvert\mathcal{F}\rvert$	205	Number of function words

Table 10: WAN Construction Parameters.

C.2 Feature Extraction Parameters

Strategy	Parameter	Value
PCA (Strategy A)	Components	100
	Batch Size	1000
CNN (Strategy B)	Embedding Dim	64
	Conv1 Filters	16
	Conv2 Filters	32
	Kernel Size	3×3
	Dropout (Conv)	0.25
	Dropout (FC)	0.5
	Training Epochs	10
	Learning Rate	1×10^{-3}
	Weight Decay	1×10^{-5}
Graph (Strategy C)	Features	615 (3×205)

Table 11: Parameters for Vectorization Strategies.

C.3 Classifier Hyperparameters

Classifier	Key Parameters
Log. Regression	max_iter=1000, class_weight='balanced'
SGD (Linear SVM)	loss='hinge', max_iter=1000
Random Forest	n_estimators=100, class_weight='balanced'
G. Naive Bayes	Default priors
AdaBoost	n_estimators=50, base_estimator=DecisionStump
XGBoost	n_estimators=100, eval_metric='logloss'

Table 12: Classifier settings.

C.4 SMOTE Configuration

Parameter	Value	Description
Method	SMOTE	Synthetic Minority Oversampling
Threshold	$\eta > 2.0$	Apply if imbalance ratio exceeds
k_neighbors	5	Nearest neighbors for synthesis

Table 13: SMOTE settings.

D Dataset Statistics

D.1 XXXX Corpus Details

Author	Era	Docs	Tokens
Munshi Premchand	20th C.	17	∼2.1M
Rabindranath Tagore	19th C.	3	∼2.0M
Sarat Chandra Chattopadhyay	20th C.	13	∼470K
Suryakant Tripathi Nirala	20th C.	6	∼413K
Acharya Chatursen Shastri	20th C.	5	∼361K
Ratan Nath Sarshar	19th C.	3	∼347K
Ayodhya Singh Upadhyay	19th C.	4	∼339K
Balkrishna Bhatt	19th C.	3	∼328K
Kishori Lal Goswami	19th C.	4	∼305K
Gopal Ram Gahmari	19th C.	3	∼303K
Siyaram Sharan Gupta	20th C.	3	∼183K
Devaki Nandan Khatri	19th C.	6	∼156K
Bankim Chandra Chatterjee	19th C.	3	∼116K
Gajanan Madhav Muktibodh	20th C.	3	∼85K
Jaishankar Prasad	20th C.	3	∼68K
Total		**79**	**∼7.5M**

Table 14: Detailed corpus statistics per author.

D.2 Sample Distribution

Chunk Size (c)	Total Samples	Samples/Author (avg)
100	75,225	5,015
300	25,076	1,672
500	15,076	1,005
700	10,768	718
1,000	7,563	504
1,500	5,057	337
2,000	3,801	253

Table 15: Sample counts for different chunking granularities.

D.3 Information-Theoretic Metrics

Strategy	$I(Y;\phi)$	$H(Y\|\phi)$	Efficiency	Dims
PCA	3.75 bits	0.15 bits	96.1%	100
CNN	3.88 bits	0.03 bits	99.3%	64
Graph	3.84 bits	0.07 bits	98.2%	615

Table 16: Information-theoretic evaluation.

E Linguistic Verification

E.1 Case Frame Clustering

Metric	Value	Significance
t-statistic	11.90	$p < 0.001$
Pronoun→Post density	0.0847	
Baseline density	0.0312	
Ratio	2.71×	

Table 17: Statistical validation of Case Frame Clustering.

E.2 Ergative Ratio by Author

Author	Mean ERG	Std	Genre
Jaishankar Prasad	0.0294	0.001	Narrative
Bankim Chandra Chatterjee	0.0244	0.013	Narrative
Acharya Chatursen Shastri	0.0209	0.011	Mixed
Kishori Lal Goswami	0.0201	0.005	Narrative
Munshi Premchand	0.0196	0.006	Narrative
Gopal Ram Gahmari	0.0195	0.004	Narrative
Devaki Nandan Khatri	0.0183	0.008	Adventure
Ayodhya Singh Upadhyay	0.0098	0.004	Descriptive
Balkrishna Bhatt	0.0090	0.004	Essay
G. M. Muktibodh	0.0065	0.002	Philosophical

Table 18: Ergative Ratio (ERG) across authors.

E.3 Verbal Complex Entropy

Transition Type	$H(S_{t+1}\|S_t)$	Interpretation
Auxiliary chains	5.80 bits	Low (deterministic)
Conjunctions	6.55 bits	High (variable)

Table 19: Conditional entropy of transitions.

F Error Analysis Details

F.1 Confusion Pairs

Author A	Author B	CR	Explanation
Jaishankar Prasad	S.T. Nirala	0.015	Chhayavaad poets
Ayodhya S. Upadhyay	Balkrishna Bhatt	0.011	Archaic register
Acharya Chatursen	S.T. Nirala	0.010	Overlapping themes
Balkrishna Bhatt	Ayodhya S. Upadhyay	0.011	Essay/Descriptive
Jaishankar Prasad	Munshi Premchand	0.007	Contemporaries

Table 20: Top 5 most confused author pairs.

F.2 Per-Author Error Rates

Author	Samples	Correct	Errors	Error %
Jaishankar Prasad	9	7	2	22.2%
Balkrishna Bhatt	44	43	1	2.3%
S.T. Nirala	56	55	1	1.8%
Rabindranath Tagore	264	263	1	0.4%
Others	-	-	0	0.0%

Table 21: Accuracy breakdown by author.