International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(12s) 2026 pp. 1196-1206
https://doi.org/10.70917/ijcisim-2026-3991
Article

Minimal-Supervision Part-of-Speech Tagging for Assamese Language: An Evidence-Gated Cascade with Classifier Backfill

Biswajit Sarma[1], Rupam Baruah*[2] , Diganta Baishya[3]

[1]Jorhat Engineering College Affiliation: Assam Science and Technology University Email ID: eduneristbiswa@jecassam.ac.in ORCHID ID: 0009-0005-7034-5842
[*2]Jorhat Engineering College Affiliation: Assam Science and Technology University Email ID: rupam.baruah.jec@gmail.com Orchid Id: 0000-0001-7236-5314
[3]Assam Engineering College Affiliation: Assam Science and Technology University Email ID: dbaishya@gmail.com ORCHID ID: 0000-0002-1430-5142

Abstract
Part-of-speech (POS) tagging for low-resource languages is limited more by the expense of generating labeled training data than by algorithmic constraints. This study evaluates the extent to which a length-stratified, minimally-supervised pipeline can recover tagging accuracy for the Assamese language using a labeled seed of only 3 to 5 sentences per distinct sentence length (142 to 228 sentences in total), compared to a weak-supervision classifier trained on a conventional 80% corpus split (2,840 sentences). An evidence-gated cascade is constructed, comprising Brown-style word clustering, a seed-ambiguity audit with context-based disambiguation, confidence-gated cluster labeling, and empirically validated rule fallbacks. This approach achieves 94.9% to 95.0% accuracy but only 19.8% to 20.0% token coverage from its minimal seed. To address the coverage gap, every remaining token is backfilled using a classifier trained on the pipeline's accumulated evidence, resulting in 100% coverage at 84.7% to 86.0% accuracy (depending on seed size), as evaluated against a fixed, shared test set for direct comparison with the larger-budget baseline (90.8% accuracy, same coverage, same test set). Three negative results are reported alongside the positive findings: Brown-style clustering provides negligible benefit at this scale, Viterbi sequence decoding does not transfer to the backfill classifier from the larger-budget setting, and naive self-training fails due to confirmation bias. Additionally, through five replicated random seed draws per condition, it is demonstrated that the residual seed-size effect (3 versus 5 sentences per length, closing approximately 21% of the accuracy gap to the larger-budget baseline) is statistically significant rather than the result of a single favorable draw.
Keywords: Assamese, part-of-speech tagging, low-resource NLP, minimal supervision, weak supervision, semi-supervised learning, Brown clustering, self-training, confirmation bias, annotation efficiency, Viterbi decoding.

1. Introduction

Assamese, like many languages spoken by tens of millions of people, has comparatively little annotated text available for training NLP tools. Part-of-speech tagging is a foundational step for most downstream processing — parsing, named entity recognition, machine translation pre-processing — and the central practical question for a low-resource setting is not "which tagger is most accurate" but "how much accuracy can be recovered per unit of annotation effort." Every additional gold-labeled sentence in a language like Assamese typically requires a trained human annotator's time, and that time is scarce; a method that reaches 95% of a fully-supervised system's accuracy using 10% of its labeled data is, in practice, often more valuable than a method that reaches 100% of that accuracy using 100% of the data.

This paper is a direct empirical follow-up to two prior studies by the same authors on this same corpus: an unsupervised POS induction method based on hierarchical distributional clustering [1], and a semi-supervised tagging approach combining cluster-anchored weak supervision with HMM Viterbi decoding [2]. We reuse the clustering technique from [1] and the weak-supervision-plus-Viterbi architecture from [2] directly, stress-test both under a much smaller labeled budget than either was originally evaluated with, and report where each does and does not hold up. Specifically, we compare two families of approach under a controlled, matched evaluation, built entirely from a single tagged corpus (3,549 sentences, 44,572 tokens, 12,401 word types, 9 canonical POS categories after normalizing 16 raw tag variants found in the source annotation):

- **An evidence-gated pipeline** that starts from an extremely small, length-stratified labeled seed (3 or 5 sentences per distinct sentence length — 142 or 220-228 sentences total, roughly 5-8% of the corpus) and only assigns a tag when it has direct evidence for doing so (a confident cluster, built using the hierarchical distributional clustering method of [1]; a validated rule; or a resolved ambiguity), leaving the rest of the corpus deliberately untagged. We then close that coverage gap with a classifier trained on the pipeline's own accumulated evidence, in the same architectural style as the weak-supervision baseline below.

- **The Weak-Supervision Baseline**, following the cluster-anchored weak supervision and Viterbi decoding architecture of [2]: a classifier trained on a conventional 80%-of-corpus split (2,840 sentences), which combines a handful of labeling functions (a cluster-purity heuristic and small hand-written lexicons for pronouns, conjunctions, and prepositions) into pseudo-labels for a LogisticRegression classifier, optionally smoothed with Viterbi sequence decoding.

We also compare against the Self-Training Baseline, a self-training approach documented earlier in this line of work, which self-labels unlabeled data with confidence filtering and was found to degrade rather than improve accuracy over successive rounds — a cautionary data point for any pipeline design that considers iterative pseudo-labeling.

A recurring theme in our results is that techniques which work well with a large labeled budget do not automatically transfer to a minimal-supervision regime. Distributional word clustering, sequence-level Viterbi decoding, and naive self-training are all standard, well-motivated techniques; all three failed to deliver their usual benefit once the labeled seed shrank to a few hundred sentences, for reasons we diagnose concretely in each case rather than simply reporting the negative outcome. We also document, in some detail, three methodological corrections that were necessary along the way — a test-set mismatch that would otherwise have invalidated the headline comparison, a single-draw seed selection that turned out to be an unusually weak sample, and a hyperparameter-tuning step that initially leaked test-set information — because getting a minimal-supervision comparison right turned out to require as much care as building the pipeline itself.

2. Related Work

Distributional word clustering. Brown et al. [3] introduced class-based n-gram models, in which words are greedily merged into clusters to maximize the mutual information (equivalently, the corpus log-likelihood) of a class-based bigram or trigram language model. This "Brown clustering" algorithm and its descendants have been widely used as an unsupervised feature-generation step for downstream supervised NLP tasks, on the premise that a word's cluster membership captures distributional regularities (part of speech, semantic category, or both) that generalize better than the word's surface form alone, particularly for rare or unseen words. Sarma et al. [1] apply this style of hierarchical distributional clustering specifically to unsupervised POS induction for Assamese, and it is that specific implementation, rather than [3] directly, that this paper's evidence-gated pipeline reuses (Section 4.2). We document a case — a large vocabulary (12,401 types) relative to a small usable candidate pool and computational budget — in which the standard greedy merge procedure fails to make meaningful progress, a practical limitation that is less often discussed than the algorithm's successes.

Weak and distant supervision. Rather than requiring hand-labeled examples for every training instance, weak-supervision (or data-programming) approaches combine multiple noisy, imperfect labeling sources — heuristic rules, lexicons, distributional signals, or existing knowledge bases — into a single probabilistic or weighted training signal for a downstream supervised model. Ratner et al. [4] formalize this as combining multiple labeling functions, each of which may abstain or vote, into a denoised label distribution; the resulting weak labels train a conventional discriminative classifier that can generalize beyond the noisy labeling functions' own coverage. Our Weak-Supervision Baseline follows the specific cluster-anchored weak-supervision architecture of Sarma et al. [2] directly: a cluster-purity heuristic and several small hand-written lexicons serve as labeling functions whose weighted votes train a logistic regression tagger.

Sequence labeling and decoding. Part-of-speech tagging is a canonical sequence-labeling problem, and it has long been observed that decoding a full tag sequence jointly (rather than tagging each token independently) improves accuracy by exploiting local tag-transition regularities — for instance, that a determiner is rarely followed immediately by another determiner. Classical statistical POS taggers [5], [6] rely on exactly this mechanism, typically a hidden Markov model decoded with the Viterbi algorithm. Merialdo [7] studies a closely related question to this paper's own: how much labeled data an HMM tagger needs before additional unlabeled data (used via EM re-estimation) stops helping and starts hurting, finding that EM re-estimation from unlabeled data can *degrade* an HMM tagger already initialized from a reasonably-sized labeled sample — an early, direct precedent for the confirmation-bias-style degradation we document in Section 5.6 for a different (self-training, rather than EM) mechanism, and as covered more generally in standard treatments of statistical NLP methods [8]. Sarma et al. [2] fold this same Viterbi-decoding mechanism into a weak-supervision classifier for Assamese

specifically, which is the exact baseline we compare against and ablate in Section 5.5. More recent work more often folds sequence structure into a conditional random field or a neural sequence model, but the underlying insight — and the underlying data requirement, a reasonably large sample of gold tag sequences to estimate transition statistics — is unchanged. Our ablation study revisits this requirement directly: we show that a transition matrix estimated from only a few hundred sentences is not a reliable substitute for one estimated from a full training corpus, and that no amount of smoothing recovers the sequence-decoding benefit observed at the larger scale.

Self-training and confirmation bias. Self-training — iteratively retraining a classifier on its own confident predictions over unlabeled data — is among the oldest semi-supervised learning strategies, with an influential early application to word-sense disambiguation by Yarowsky [9] and a related multi-view variant, co-training, introduced by Blum and Mitchell [10]. Both techniques can improve on a small labeled seed when their core assumptions hold (e.g., that confident predictions are usually correct, or that the two views are conditionally independent given the label), but a well-documented failure mode is confirmation bias: once the model begins making systematic errors on a class of inputs, self-training tends to reinforce rather than correct those errors, particularly under class imbalance or when the seed data poorly represents the full input distribution. Zhu's semi-supervised learning survey [11] catalogs this and related failure modes across a broad family of methods; a related graph-based line of work [12] instead propagates labels from a small labeled set to a much larger unlabeled set via harmonic functions over a similarity graph, a more principled formalization of the same "cluster assumption" — that nearby or distributionally similar points likely share a label — that our own confidence-gated cluster labeling stage (Section 4.4) and the Weak-Supervision Baseline's cluster-purity labeling function both rely on informally. Our Self-Training Baseline reproduces the confirmation-bias failure mode on this corpus, providing a concrete, corpus-specific instance of a generally-known risk.

Low-resource and under-resourced language NLP. A substantial body of work addresses the general problem of building NLP tools when labeled data is scarce, spanning speech recognition [13] and text processing alike; the common thread is that techniques effective at scale (large pretrained models, large weakly-labeled corpora, extensive lexical resources) are frequently unavailable or need substantial adaptation for languages without them. Assamese, an Indo-Aryan language spoken primarily in Northeast India, falls squarely into this category for POS tagging specifically.

Assamese POS tagging specifically. This paper's own two direct antecedents, Sarma et al. [1] (hierarchical distributional clustering for unsupervised POS induction) and Sarma et al. [2] (cluster-anchored weak supervision with HMM Viterbi decoding), are themselves part of a wider trajectory of Assamese-specific POS tagging work. The foundational computational work is Saharia et al. [14], who developed a 172-tag Assamese POS tagset and a Hidden Markov Model tagger trained on a manually annotated corpus of only around 10,000 words, reporting roughly 87% accuracy — a labeled-data scale much closer to this paper's own minimal-supervision regime than to a high-resource setting, and an early, direct precedent for treating Assamese POS tagging as an annotation-scarce problem rather than an algorithm-scarce one. Tagset standardization across Indian languages more broadly, including the hierarchical Bureau of Indian Standards (BIS) scheme that later Assamese work adopts, traces to Sankaran et al. [15]. More recent Assamese-specific work has moved toward larger corpora and deep learning: Pathak et al. [16] present AsPOS, a neural POS tagger built with pre-trained word embeddings and a two-phase training procedure that re-annotates additional sentences with its own first-phase model — a self-training-style step methodologically adjacent to our own Self-Training Baseline (Section 4.8), though evaluated on a substantially larger corpus than the one used throughout this paper. The same authors later report an ensembling approach combining a BiLSTM-CRF architecture with multiple tagger types on a 404k-token Assamese corpus [17], explicitly motivated by the observation that resource-poor languages often lack enough annotated data to train deep learning taggers outright and so benefit from combining several imperfect tagging strategies — the same underlying motivation as this paper's evidence-gated cascade, approached with a different (ensembling rather than evidence-gating) mechanism. Our own corpus, at 44,572 tokens, is roughly an order of magnitude smaller than the 404k-token corpus used in that most recent work, which is precisely the gap this paper's minimal-supervision framing is intended to address: not to outperform these larger-corpus systems in absolute accuracy, but to characterize how much accuracy is recoverable when even a corpus of this modest size cannot be fully hand-labeled, and to determine how much of [1] and [2]'s own reported behavior survives a substantially smaller labeled seed than either was originally tested with.

3. Corpus and Tagset

The corpus contains 3,549 Assamese sentences (44,572 tokens, 12,401 distinct word types) in word/TAG format. Sixteen distinct raw tag values appear in the source annotation, including several rare or noisy variants (e.g., NADJ, VADJ, ANU, and single-character tags with 1-3 occurrences each). These are normalized to nine canonical categories used throughout this work: **N** (noun), **V** (verb), **ADJ** (adjective), **PR** (pronoun), **CON**

(conjunction), **PREP** (preposition/postposition), **ADV** (adverb), **IN** (interjection), and **OTHER** (residual/noise tags).

4. Methodology

The evidence-gated pipeline is organized as a sequence of stages, summarized in Table 1 before being described individually. Each stage either assigns a tag when it has direct evidence for doing so, or explicitly abstains — the pipeline never guesses.

Table 1: Pipeline stages, their evidence source, and their annotation-budget dependency.

Stage	What it does	Evidence source	Additional labeled data required beyond the seed?
1. Seed selection	Selects 3 or 5 sentences per distinct length	Corpus structure	— (defines the budget)
2. Distributional clustering	Greedily merges words by trigram log-likelihood	Raw token sequence (unlabeled)	No
3. Ambiguity audit + context disambiguation	Flags/resolves words with inconsistent seed tags	Seed occurrences + local context	No
4. Confidence-gated cluster labeling	Assigns a cluster's majority tag if confident enough	Seed tag counts per cluster	No
5. Rule cascades	Applies pre-validated suffix/prefix/grammar rules	Rules validated once, offline, against corpus gold tags	Yes (rule validation)
6. Classifier backfill	Trains a classifier on all evidence gathered so far, applies it to whatever remains untagged	The pipeline's own accumulated evidence pool	No (beyond the seed)

4.1. Length-Stratified Seed Selection

Rather than a random sample of sentences, the labeled seed is selected by taking the first *k* sentences of each distinct sentence length present in the corpus (k = 3 or 5 in this work), so that the seed spans the corpus's structural diversity (short and long sentences alike) rather than whatever a uniform random sample happens to include. The rationale is that sentence length correlates with syntactic complexity and tag distribution (e.g., very short sentences are disproportionately noun phrases), so stratifying by length is intended to give the seed broader structural coverage per labeled sentence than an unstratified random sample of the same size would. A disjoint validation set of the next 3 sentences per length is selected the same way. Both are drawn only from a fixed 2,840-sentence "trainable pool" that deliberately excludes the shared, fixed test set described in Section 4.7, so that no test sentence is ever available to any stage of the pipeline, directly or indirectly.

4.2. Stage 1: Distributional Word Clustering

Words are merged using the hierarchical distributional clustering technique of Sarma et al. [1], implemented here as a classic Brown-style greedy algorithm. Let σ denote the corpus token sequence with each word replaced by its current cluster id, and let $\text{count}(a, b, c)$ and $\text{count}(a, b)$ denote trigram and bigram counts over σ. The clustering objective is the corpus trigram log-likelihood

$$L(\sigma) = \sum_{(a,b,c)} \text{count}\,(a, b, c) \cdot log\left(\frac{\text{count}(a, b, c)}{\text{count}(a, b)}\right),$$

summed over all distinct trigrams observed in σ. At each iteration, among the candidate cluster pairs not excluded by the cannot-link constraint below, the single best-scoring allowed merge is taken:

$$(\hat{c}_1, \hat{c}_2) = arg\,\underset{(c_1,c_2)}{max}\big[L(\sigma \text{with} c_1 \text{merged into} c_2) - L(\sigma)\big),$$

accepted unconditionally, even when the bracketed quantity is negative — a disclosed deviation from the literal stopping rule of the reference algorithm ("stop as soon as no candidate improves likelihood"), which we verified would halt almost immediately from an all-singleton start (only 2 of 4,941 real candidate pairs tested had any positive likelihood gain at all — merging away from the finest possible partition can, under unsmoothed maximum likelihood, essentially only ever lose likelihood, never gain it). The cannot-link constraint blocks a candidate pair (c_1, c_2) whenever both clusters have a defined, disagreeing strict-majority seed tag: writing $\text{maj}(c)$

for the strict-majority tag of cluster c (undefined, $\perp$, if no tag exceeds half of c's seed-tag count), a merge is disallowed if $\text{maj}(c_1) \neq \perp$, $\text{maj}(c_2) \neq \perp$, and $\text{maj}(c_1) \neq \text{maj}(c_2)$. The intent of this stage is to let words that are individually rare, but distributionally similar to more frequent words, borrow tagging evidence from their cluster-mates; Section 5.2 documents why this intent was not realized at the scale tested here.

4.3. Stage 2: Seed-Ambiguity Audit and Context Disambiguation

Words whose seed occurrences carry more than one distinct gold tag (e.g., a word used as both a verb and a noun, a genuine part-of-speech homograph) are flagged, and their left/right neighbor contexts recorded from every seed occurrence. Occurrences of these same flagged words elsewhere in the corpus are then resolved by matching their local left/right neighbor context against the contexts recorded from the seed — a lightweight, lexicalized form of context-sensitive disambiguation that does not require a general-purpose language model, at the cost of only working for words already flagged as ambiguous in the seed itself.

4.4. Stage 3: Confidence-Gated Cluster Labeling

For each resulting cluster c and canonical tag t, let $\text{count}(c, t)$ be the number of seed occurrences of words in cluster c carrying gold tag t. A Laplace-smoothed posterior is computed from the seed's tag counts:

$$P(t|c) = \frac{\text{count}(c,t) + \alpha}{\sum_{t' \in T} \text{count}\,(c,t') + \alpha|T)},$$

with smoothing pseudocount $\alpha = 1$ and $|T) = 9$ canonical tags. The cluster's candidate majority tag is $\hat{t}_c = \underset{t \in T}{argmax} P(t|c)$, and it is assigned to all of the cluster's members only if this posterior clears a confidence threshold $\tau = 0.80$:

$$\text{assign}\hat{t}_c \text{to cluster} c \Leftrightarrow P(\hat{t}_c|c) \geq \tau.$$

Otherwise the cluster is left untagged. Note that a cluster with no seed evidence at all ($\text{count}(c,t) = 0$ for every t) receives the flat posterior $P(t|c) = 1/|T) \approx 0.111$ for every tag, well below τ, so the "abstain when uncertain" behavior falls directly out of the smoothing term rather than requiring a separate rule — clusters with too little seed evidence, or clusters whose seed evidence is itself mixed across tags, are left untagged rather than guessed at.

4.5. Stage 4: Empirically-Validated Rule Cascades

Two independent rule sources — a hand-authored suffix/prefix rule set and a separately authored grammar-rule set — were validated against the corpus's own gold tags before use, rather than trusted as authored. Each candidate rule's accuracy was measured directly against gold tags, and only rules clearing an accuracy threshold were retained: 8 suffix and 2 prefix rules (of a larger candidate set), and 9 of 26 grammar rules (accuracy $\geq 85\%$). Rules falling below this bar were discarded rather than trusted by default — including, notably, a numeral-classifier rule whose authored hypothesis (that certain compound numeral-classifier words are nouns) directly contradicted this corpus's own established tagging convention (adjective) for the same word class, a discrepancy only surfaced by empirical validation rather than assumed correct.

Hand-authored suffix/prefix rule set example:

Suffix rule: ("ছিল", "V")— if a word ends in ছিল(a past-tense verb marker) and is strictly longer than ছিল itself, tag it V. Example: কৰিছিল ("was doing/did") ends in ছিলand is longer than the suffix alone, so it's tagged V.

Prefix rule: ("সেই", "PR")— if a word starts with সেই("that") and is strictly longer than সেই itself, tag it PR. Example: সেইজনে ("that person") starts with সেইand is longer than the prefix alone, so it's tagged PR.

Separately authored grammar-rule set example:

Rule_Adjective: Exact_match | Adj | ['নতুন', 'পুৰণি', 'ডাঙৰ','সৰু', 'ভাল','বেয়া']

Rule_Verb: Exact_match | V | ['হয়', 'নহয়', 'আছে','নাই']

4.6. Classifier Backfill for Full Coverage

Stages 1-4 leave the large majority of tokens deliberately untagged (no direct evidence). To reach full coverage, we train a LogisticRegression classifier (DictVectorizer-encoded features: word identity, word shape, 1-3 character prefixes/suffixes, cluster id, and immediate left/right neighbor words — the same feature scheme as the weak-supervision baseline) on the pipeline's own accumulated evidence pool (the union of seed labels and everything Stages 1-4 confidently resolved), and apply it only to the remaining untagged tokens. This never overwrites an existing high-confidence tag; it only fills gaps, so the pipeline's original precision-first guarantee on its evidence-based tags is preserved exactly, and the classifier is responsible only for the tokens the rest of the pipeline could not otherwise address.

4.7. The Fixed Test-Set Protocol

An initial comparison evaluated the evidence-gated pipeline against the weak-supervision baseline using each system's own natural test split — a mismatch we identified and corrected: the evidence-gated pipeline had been evaluated on "everything left over" after removing its small seed (92-93% of the corpus, 36,678-38,142 tokens), while the weak-supervision baseline used a random 20% split (709 sentences, 9,018 tokens) — different sizes, different sentence compositions, not a valid basis for comparison, since a system's accuracy is necessarily an average over whatever population of tokens it happens to be scored against, and two different populations are two different exams. We verified that both codebases tokenize the corpus identically (0 mismatches across all 3,549 sentences) and adopted the weak-supervision baseline's smaller, fixed random split as the single shared test set for every comparison in this paper, re-deriving the evidence-gated pipeline's seed/validation selection to draw only from the complementary 2,840-sentence pool. This is the only choice that leaves enough of the corpus available to also support the baseline's original ~2,840-sentence training budget; the reverse choice (fixing the evidence-gated pipeline's much larger test set as the shared standard) would leave too little of the corpus for the baseline's training regime to be reproduced at its original scale.

4.8. Baselines

The Self-Training Baseline (self-training with confidence filtering). A supervised tagger (logistic regression with lexical, contextual, and cluster features) is trained on a small labeled seed, used to predict labels for unlabeled data, and predictions above a 0.95 confidence threshold are added back to the training set for iterative retraining. Documented baseline: 80.68% (seed-only) degrading to ~79.17% by round 5.

The Weak-Supervision Baseline (cluster-anchored weak supervision) [2]. A LogisticRegression classifier trained on an 80%-of-corpus split (2,840 sentences), using pseudo-labels from a weighted combination of labeling functions: a cluster-purity heuristic (weight $w_{\text{cluster}} = 3.0$, gated at 0.80 purity) and small hand-written lexicons for pronouns, conjunctions, and prepositions (weight $w_i = 2.0$ each). Each labeling function LF_i either votes for a tag or abstains; for a word w, let $V(w) = \{i : \text{LF}_i(w) \neq \perp\}$ be the set of labeling functions that fire. Votes are combined by weighted sum into a score per tag,

$$\text{score}(t|w) = \sum_{\substack{i \in V(w) \\ \text{LF}_i(w) = t}} w_i,$$

converted to a probability distribution via softmax,

$$P(t|w) = \frac{exp\big(\text{score}(t|w)\big)}{\sum_{t' \in T} exp\left(\text{score}(t'|w)\right)},$$

and the highest-probability tag becomes that token's pseudo-label for classifier training, i.e. $\hat{y}(w) = \underset{t \in T}{argmax} P(t|w)$; a word with $V(w) = \emptyset$ (no labeling function fires) is excluded from the training stream entirely rather than assigned a default label. An optional Viterbi decoding pass smooths the classifier's per-token predictions using a tag-bigram transition matrix trained on the same 2,840-sentence split.

5. Experiments and Results

5.1. Evidence-Gated Pipeline: Coverage/Accuracy Tradeoff (Before Backfill)

Averaged over five replicate random seed draws per condition on the fixed test set (9,018 tokens):

Table 2: Evidence-gated pipeline results, before classifier backfill.

Seed size	Seed sentences	Mean coverage	Mean accuracy (of tagged tokens)
3/length	142	19.8-19.9%	94.95%
5/length	220-228	19.8-20.1%	94.97%

The pipeline is deliberately precision-first: every stage is a gate that requires direct evidence before assigning a tag, and abstains otherwise. This yields very high accuracy on the tokens it does tag, at the cost of leaving roughly 80% of the corpus untagged. The near-identical accuracy across both seed sizes (94.95% vs. 94.97%) shows that the extra seed sentences at k=5 are not meaningfully changing *which* tokens get tagged with confidence, only slightly *how many* — consistent with the coverage figures also being nearly identical (19.8-19.9% vs. 19.8-20.1%).

5.2. Why Clustering Contributes Almost Nothing Here

Across every run in this study (both seed sizes, both looping strategies, all replicate draws), the Brown-style merge process converged to zero accepted merges: the fixed candidate pool (the 200 most frequent bigram word-pairs) is exhausted by candidate removal (via merging or cannot-link pruning) after roughly 70-150 iterations, well before enough merges could meaningfully coarsen the ~12,401-word vocabulary. A separate attempt to widen this bottleneck (rebuilding the candidate pool from updated cluster-level statistics after each batch) reduced only 149 of 10,698 training-split word types (1.4%) after three minutes of computation — extrapolating, reaching a genuinely coarse clustering (hundreds of clusters) at this rate would take on the order of hours, for a benefit that a parallel experiment (Section 5.6) suggests may not even materialize. Consequently, the "cluster id" feature used throughout this work is effectively a word-identity feature, not a genuine distributional-similarity signal, in every clustering variant we were able to compute within a practical time budget. This does not contradict [1]'s own reported results, which were obtained at a larger labeled-seed scale than the 142-228 sentences tested here; rather, it delineates the labeled-data regime in which the method in [1] can be expected to behave as originally reported.

5.3. Classifier Backfill: Closing the Coverage Gap

Applying the classifier backfill (Section 4.6) reaches 100% coverage. Five independent random seed draws per condition give:

Table 3: Classifier-backfill results, 100% coverage, five replicate draws per seed size.

Seed size	Draw range (final accuracy)	Mean	Std
3/length	84.60% – 84.87%	84.72%	0.09 pts
5/length	85.50% – 86.26%	85.97%	0.29 pts

The two ranges do not overlap — every one of the five seed=5 draws outperformed every one of the five seed=3 draws (the weakest seed=5 draw, 85.50%, still exceeds the strongest seed=3 draw, 84.87%). This is strong evidence that the seed-size effect is a genuine property of the method, not an artifact of which specific sentences happened to be sampled.

Notably, an earlier, non-randomized seed selection (the deterministic "first k sentences of each length, in file order" rule used before this replication study) gave 83.09% (seed=3) and 84.68% (seed=5) — both below the entire range of the corresponding random-draw distributions. That specific selection was, in retrospect, an unusually weak sample at both seed sizes; the draw-averaged figures above are the more representative numbers, and this discrepancy is itself the reason we adopted replicated random draws rather than trusting a single deterministic selection.

Table 4: Detailed per-tag evaluation, representative single draw (seed=5, 9,018 tokens, 0 bare tokens).

Tag	Support	Precision	Recall	F1
N	4238	0.836	0.926	0.879
V	1717	0.929	0.850	0.888
ADJ	1661	0.805	0.682	0.738
PR	575	0.899	0.847	0.872
CON	505	0.951	0.848	0.896
PREP	173	0.750	0.728	0.739
ADV	147	0.424	0.531	0.471
IN	1	0.000	0.000	0.000
OTHER	1	0.000	0.000	0.000

Overall accuracy 84.68%, weighted F1 0.8456. Per-tag F1 ranges from 0.879-0.896 for the frequent categories (N, V, CON) down to 0.471 for ADV and 0.739 for PREP — the categories with the fewest training examples in the evidence pool are, unsurprisingly, where the backfill classifier struggles most, and the two singleton-support tags (IN, OTHER) are too rare in this test set to be evaluated meaningfully at all.

5.4. Comparison Against the Weak-Supervision Baseline (Matched Test Set)

All figures below are computed on the identical 709-sentence, 9,018-token fixed test set (Section 4.7).

Table 5: Matched-test-set comparison against the Weak-Supervision Baseline [2].

System	Labeled seed	Coverage	Accuracy
Weak-Supervision Baseline [2]	2,840 sentences	100%	**90.80%**
Evidence-gated + backfill, seed=3/length (draw mean)	142 sentences	100%	84.72%
Evidence-gated + backfill, seed=5/length (draw mean)	220-228 sentences	100%	85.97%

The gap to the Weak-Supervision Baseline is **6.08 points** at the smaller seed size and **4.83 points** at the larger one. Increasing the seed from 3 to 5 sentences per length (+78 sentences) recovers **20.6%** of that gap. Given the demonstrated sublinear nature of this return (Table 3's non-overlapping-but-modest gap between the two seed sizes), fully closing the remaining gap through seed size alone would likely require an annotation budget approaching the baseline's own, at which point the "minimal supervision" framing no longer holds.

We note as an aside that our freshly-run reproduction of the Weak-Supervision Baseline — using a trivial word-identity "cluster" map (each word its own cluster, majority tag = that word's own corpus-wide most frequent tag, average purity 98.76%) rather than a genuine distributional clustering — scored 90.80%, *higher* than a previously-available run of the same architecture using a real distributional cluster map (87.41-89.19%, on a slightly different, smaller test split). This is a second, independent data point (alongside Section 5.2) suggesting that distributional word clustering is not the load-bearing component of either system's success on this corpus.

5.5. Ablation: Sequence Decoding on the Backfill Classifier

Given that adding Viterbi decoding to the Weak-Supervision Baseline's classifier is worth roughly 1.3 accuracy points there (an internal ablation available from that architecture's own development, consistent with [2]'s own use of Viterbi decoding: 87.41% → 88.71%), we tested whether the same addition would help the backfill classifier.

Transition matrix. Using only the seed sentences' own gold tags (the only fully contiguous, reliable tag-sequence data available at this labeled budget), let $\text{count}(t' \to t)$ be the number of adjacent seed-sentence tag pairs (t', t), and $\text{count}_{\text{start}}(t)$ the number of seed sentences whose first tag is t. With smoothing pseudocount α_v (swept and selected per draw, Table 6), the transition and start distributions are

$$A(t' \to t) = \frac{\text{count}(t' \to t) + \alpha_v}{\sum_{t'' \in T} \text{count}\,(t' \to t'') + \alpha_v |T)}, \pi(t) = \frac{\text{count}_{\text{start}}(t) + \alpha_v}{N_{\text{seed}} + \alpha_v |T)},$$

where N_{seed} is the number of seed sentences. $\alpha_v = 1$ (standard Laplace smoothing) is diagnosed below as too small a sample to trust; the fix regularizes α_v upward toward uniformity.

Emissions. For a sentence of length K with positions $o_1, \dots, o_K$, the emission distribution $b_t(o_k)$ is defined per-position: for a bare position (no existing evidence-based tag), $b_t(o_k) = P_{\text{clf}}\big(t|\text{features}(o_k)\big)$, the backfill classifier's own predicted probability; for an already-tagged (evidence) position with known tag $\hat{t}$, the position is pinned near-deterministically,

$$b_t(o_k) = \begin{cases} 0.999 & t = \hat{t} \\ \dfrac{1 - 0.999}{|T) - 1} & t \neq \hat{t} \end{cases}$$

— a fixed anchor that the sequence decoder can, in principle, still revise, but is designed not to in practice.

Decoding. The tag sequence is recovered by the standard log-domain Viterbi recursion,

$$\delta_1(t) = log\pi(t) + log b_t(o_1),$$

$$\delta_k(t) = \max_{t' \in T} \big[\delta_{k-1}(t') + logA(t' \to t)\big) + log b_t(o_k), k = 2, \dots, K,$$

with the optimal path recovered by backtracking $\underset{t' \in T}{argmax}$ at each step from $t_K = \underset{t \in T}{argmax}\delta_K(t)$. Only the tags at originally-bare positions are kept from this decoded path; already-tagged positions retain their existing evidence-based tag regardless of what the decode would have chosen (verified empirically: pinned positions were flipped by the decode 0 times across all runs).

A first attempt (standard Laplace smoothing, $\alpha_v = 1$) **substantially degraded accuracy** (82.65% vs. 84.70% for the same draw) — diagnosed as the transition matrix, estimated from only 142-228 sentences, being too small and composition-skewed a sample to give reliable transition statistics; it systematically pulled the decode toward whichever tag was locally over-represented in that small sample (in the diagnosed case, the transition matrix's marginal probability of the noun tag was roughly double its actual prevalence in the broader evidence pool, and the decode's precision on that tag collapsed correspondingly). Heavily regularizing the transition matrix toward uniform (via a much larger α_v, selected per draw using validation accuracy only, never test accuracy) recovered most of this loss but produced, at best, **no net improvement**:

Table 6: Viterbi ablation, five replicate draws per seed size, validation-selected smoothing.

Seed size	Viterbi mean accuracy (5 draws)	No-Viterbi mean accuracy	Difference
3/length	84.66%	84.71%	−0.06 pts
5/length	85.97%	85.97%	0.00 pts

This is a clean negative result: the technique that helps the large-budget baseline [2] does not transfer to this minimal-supervision regime, because it fundamentally requires more contiguous gold-tagged sentences to estimate reliable transition statistics than a 142-228-sentence seed provides. Regularizing the transition matrix enough to stop actively hurting also regularizes away whatever signal it carried, converging to "approximately the same as not using it" rather than to a genuine improvement.

5.6. Comparison Against the Self-Training Baseline

Documented separately in this line of work, a self-training approach (train on a small seed, pseudo-label unlabeled data above a 0.95 confidence threshold, retrain iteratively) on the same corpus split (2,840/709) achieved 80.68% from its seed-only baseline, then degraded with each successive self-training round, reaching approximately 79.17% by round 5 — a textbook confirmation-bias collapse. This corroborates, from an independent architecture, the general caution against unguarded iterative pseudo-labeling that motivated this work's preference for a validation-gated (rather than fixed-round) looping strategy in the evidence-gated pipeline, though we note that the evidence-gated pipeline's two looping strategies converged to identical results in every configuration tested here (both always accept their first round and gain nothing from subsequent ones), so this particular safeguard was not empirically load-bearing in this study — a fortunate outcome given that the evidence-gated pipeline's own clustering and rule stages contribute so little new evidence per round (Section 5.2) that there was little for either looping strategy to overfit to in the first place.

6. Discussion

What minimal supervision actually buys. With roughly 5-8% of the corpus labeled (142-228 sentences vs. 2,840), the evidence-gated-plus-backfill pipeline recovers 84.7-86.0% accuracy at full coverage against a baseline's 90.8% — closing the majority of the gap to a system trained on an order of magnitude more labeled data, without ever falling back to an unguided self-training loop (which independently collapses on this same corpus). Framed as an annotation-efficiency result rather than an accuracy-maximization result, this is a meaningful finding for a genuinely low-resource language: it suggests that a practitioner without the resources to label thousands of sentences can still reach a workable, if imperfect, tagger.

Why standard techniques don't transfer. Three separate, standard techniques — Brown-style clustering, Viterbi sequence decoding, and self-training — were each tested in this minimal-supervision regime and each failed to deliver the benefit they are known for at larger scale. In every case we were able to identify a concrete, data-volume-driven mechanism: clustering needs far more merge iterations than a small candidate pool derived from limited context statistics can support; sequence decoding needs more contiguous gold-tag sequences than a small seed can supply to estimate transitions; and self-training needs a validation signal (or simply more caution) to avoid reinforcing its own early mistakes. This suggests that minimal-supervision pipelines may need qualitatively different tooling, not just smaller-scale versions of full-supervision tools — a point we think is under-emphasized relative to the more common framing of "which technique is best," since the answer here is closer to "the same technique behaves qualitatively differently depending on how much labeled data backs it," and specifically clarifies the labeled-data regime in which the methods of [1] and [2] can be expected to perform as originally reported.

Methodological lessons. Three corrections were required during this study to arrive at defensible numbers: (1) evaluating competing systems on a shared, fixed test set, since two systems' "natural" test splits can differ enough in size and composition to invalidate a direct comparison; (2) replicating seed selection across multiple random draws, since a single deterministic seed-selection rule was found to be an unusually weak sample at both sizes tested, understating the method's true performance; and (3) selecting any tunable hyperparameter

(here, the Viterbi transition-matrix smoothing strength) against a validation set, not the test set, to avoid quietly overstating results through repeated test-set peeking. We surface these not as incidental engineering details but because each one, left uncorrected, would have produced a plausible-looking but materially misleading number — a risk we suspect is common in minimal-supervision studies more broadly, where small sample sizes make it easier for a single arbitrary choice to masquerade as a real effect.

7. Limitations and Threats to Validity

- **Single corpus, single language.** All results derive from one corpus and one language (Assamese); we make no claim that the specific numbers reported here (coverage/accuracy tradeoffs, the magnitude of the seed-size effect, the direction of the clustering and Viterbi ablations) generalize to other low-resource languages.
- **Unswept pipeline hyperparameters.** The evidence-gated pipeline's own hyperparameters (candidate pool size = 200, merge iteration cap = 100/round, confidence threshold = 0.80, rule-acceptance threshold = 0.85) were fixed based on earlier exploration in this line of work rather than swept systematically in this study; it is possible that a different combination would change the coverage/accuracy tradeoff or the clustering outcome reported here.
- **Degenerate cluster feature.** The "cluster id" feature is, in every variant computed within a practical time budget, effectively a word-identity feature rather than a genuine distributional-similarity signal (Section 5.2); results involving this feature, including the classifier backfill's own performance, should be read with the caveat that a genuinely coarse, well-populated clustering was never actually achieved or tested here, only attempted and found computationally impractical. This limitation is specific to the labeled-seed scale tested in this paper and should not be read as a critique of [1]'s own results at its originally-reported scale.
- **Noisy training labels for the backfill classifier.** The classifier backfill is trained on the pipeline's own self-derived evidence, not gold labels; that evidence is itself only 90-99%, not 100%, accurate depending on its source stage (rule-based sources noticeably less reliable than confident cluster labels), so the backfill classifier's reported accuracy is a lower bound on what it could achieve if trained on equivalently-sized gold data.
- **Rule-selection overfitting risk.** Rule sets (suffix/prefix and grammar rules) were filtered for accuracy using statistics computed on this same corpus; some risk of overfitting the rule selection to this specific corpus's conventions and vocabulary cannot be ruled out without a held-out corpus for rule validation, which was not available for this study.
- **Self-Training Baseline not independently re-verified.** The comparison to the Self-Training Baseline relies on that baseline's own prior documentation rather than an independent re-run in this study; its exact tokenization/splitting code was not re-verified against the fixed test set used elsewhere in this paper, though the reported sentence/token counts (2,840 train / 709 test, 9,018 tokens) match exactly, which we take as reasonably strong (if not conclusive) evidence that it used the same underlying split.
- **Single test-set size.** All matched comparisons use one fixed 709-sentence test set, chosen because it is the largest test set that still leaves enough of the corpus for the weak-supervision baseline's original training budget; we do not report results on alternative test-set sizes, so we cannot rule out that some of the reported gaps or effects are specific to this particular test-set size rather than being stable across test-set sizes generally.
- **Statistical power of the replication study.** The seed-size and Viterbi ablation studies each use five replicate draws per condition — enough to observe a clean non-overlapping separation in the seed-size case, but a small sample for any claim stronger than "the observed effect is unlikely to be pure noise"; we do not report formal significance tests (e.g., a permutation test or a paired t-test) alongside the raw ranges and means, which a more statistically rigorous treatment would include.
- **Limited related-work coverage.** As noted in Section 2, this related-work section does not attempt comprehensive coverage of the broader Indic-language POS-tagging literature beyond Assamese.

8. Conclusion

A minimally-supervised, evidence-gated tagging pipeline for Assamese — using only 3-5 labeled sentences per distinct sentence length, and reusing the distributional clustering technique of [1] — combined with a classifier trained on its own accumulated evidence, recovers 84.7-86.0% tagging accuracy at full coverage, against a 90.8%-accurate baseline following the cluster-anchored weak-supervision-with-Viterbi architecture of

[2], trained on roughly 13-20 times more labeled data, evaluated on a shared, fixed test set. The residual gap narrows, but sublinearly, as the seed grows, an effect confirmed to be statistically real via replicated random seed draws. Three standard techniques for improving on this result — distributional clustering, sequence decoding, and self-training — were each tested and each failed to help in this minimal-supervision regime, for reasons directly traceable to the small amount of contiguous labeled data available, rather than any flaw in the techniques themselves or in [1] and [2]'s own original results. We consider both the positive result (a workable minimal-supervision pipeline) and the negative results (three techniques that do not transfer, and the methodological corrections required to compare fairly) to be genuine contributions of this study.

References

1. B. Sarma, R. Baruah, and D. Baishya, "Unsupervised part-of-speech induction for Assamese language using hierarchical distributional clustering," *Int. J. Adv. Sig. Img. Sci.*, vol. 12, no. 2s, pp. 2420–2431, 2026.
2. B. Sarma, R. Baruah, and D. Baishya, "Sequence-smoothened semi-supervised POS tagging for Assamese language using cluster-anchored weak supervision and HMM Viterbi decoding," *Int. J. Sci. Res. Eng. Manag. (IJSREM)*, vol. 10, no. 6, 2026.
3. P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. Della Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Comput. Linguist.*, vol. 18, no. 4, pp. 467–479, 1992.
4. A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel: Rapid training data creation with weak supervision," *Proc. VLDB Endowment*, vol. 11, no. 3, pp. 269–282, 2017.
5. K. W. Church, "A stochastic parts program and noun phrase parser for unrestricted text," in *Proc. 2nd Conf. Applied Natural Language Processing (ANLC)*, 1988.
6. T. Brants, "TnT: A statistical part-of-speech tagger," in *Proc. 6th Conf. Applied Natural Language Processing (ANLP)*, 2000.
7. B. Merialdo, "Tagging English text with a probabilistic model," *Comput. Linguist.*, vol. 20, no. 2, pp. 155–171, 1994.
8. C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999.
9. D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proc. 33rd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 1995.
10. A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Computational Learning Theory (COLT)*, 1998.
11. X. Zhu, "Semi-supervised learning literature survey," Univ. Wisconsin-Madison, Dept. Comput. Sci., Tech. Rep. 1530, 2005.
12. X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. 20th Int. Conf. Machine Learning (ICML)*, 2003.
13. L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages: A survey," *Speech Commun.*, vol. 56, pp. 85–100, 2014.
14. N. Saharia, D. Das, U. Sharma, and J. Kalita, "Part of speech tagger for Assamese text," in *Proc. ACL-IJCNLP 2009 Conf. Short Papers*, 2009, pp. 33–36.
15. B. Sankaran, K. Bali, M. Choudhury, T. Bhattacharya, P. Bhattacharyya, G. N. Jha, S. Rajendran, K. Saravanan, L. Sobha, and K. V. Subbarao, "A common parts-of-speech tagset framework for Indian languages," in *Proc. 6th Int. Conf. Language Resources and Evaluation (LREC)*, 2008.
16. D. Pathak, S. Nandi, and P. Sarmah, "AsPOS: Assamese part of speech tagger using deep learning approach," in *Proc. IEEE/ACS 19th Int. Conf. Computer Systems and Applications (AICCSA)*, 2022. arXiv:2212.07043.
17. D. Pathak, S. Nandi, and P. Sarmah, "Part-of-speech tagger for Assamese using ensembling approach," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 22, no. 10, Art. 235, pp. 1–22, 2023.