StanLyric Semantic Atlas and Hierarchy Report

A Project Report on Cohere Embed v4, Graph Community Detection,

Hierarchical Lyric Analysis, and LLM-Assisted Labeling

Dheeraj Dhillon

June 2026

Abstract

StanLyric began as an interpretable lyric-fragment retrieval system and evolved into a semantic hierarchy over a cleaned 36,545-song corpus. This report documents the embedding-centered analysis, hierarchy construction, and LLM-assisted labeling that now power the portfolio interface. Each song was represented by a structured text containing its title, artist, and lyrics, then encoded with Cohere `embed-v4.0` using the clustering input mode and a 1,024-dimensional floating-point output. A resumable extraction pipeline completed 381 successful batched API calls, persisted 40 checkpoints, and assembled a row-aligned 142.8 MiB embedding matrix.

The embeddings were used to fit a three-dimensional UMAP atlas and to construct a connected, weighted fuzzy-neighborhood graph with 36,545 nodes and 834,320 edges. An initial weighted Leiden partition found 50 communities with modularity 0.5691. That flat result was useful but imposed a single semantic scale and contained a 5,651-song largest community. The final method therefore replaced the flat partition as the primary organization with a strict three-level hierarchy: 20 broad *Regions*, 139 middle *Communities*, and 175 fine-grained *Neighborhoods*. The middle layer was selected by a multiseed Constant Potts Model (CPM) resolution sweep using adjusted Rand agreement, normalized variation of information, adjacent-resolution persistence, balance, and fragmentation. Regions were constructed on a weighted quotient graph. Neighborhoods were accepted only when recursive splits passed minimum-size, seed-stability, graph-retention, embedding-cohesion, and NMF-topic-separation gates.

Every hierarchy node was first explained with binary-incidence class-based TF–IDF, prevalence lift, smoothed log odds, original-space centroid representatives, graph conductance, boundary songs, artist diversity, sampled cosine silhouette, and a bridge to an independently fitted 40-topic NMF model. Those evidence packets were then sent to GPT-5.5 to produce concise titles, one-line summaries, descriptions, confidence flags, and evidence keywords for all 334 Regions, Communities, and Neighborhoods. The resulting static browser artifacts power an interactive Three.js portfolio without shipping lyrics or dense vectors.

1 Executive Summary

The central engineering and modeling results are:

- A noisy 47,474-row prepared corpus was reduced to 36,545 canonical song documents through auditable metadata filtering and two forms of deduplication.
- Cohere `embed-v4.0` produced one 1,024-dimensional float embedding per song. The API input mode was `clustering`; the final matrix shape was $36{,}545 \times 1{,}024$.
- UMAP supplied both a three-dimensional browser layout and a symmetric fuzzy neighborhood graph grounded in the original cosine geometry.

- The graph was connected and sparse: 834,320 undirected weighted edges, unweighted density 0.001249, and weighted density 0.000238.
- The original 50-community Leiden baseline achieved modularity 0.5691, but it was a single-scale result and its largest community held 15.5% of the corpus.
- The final hierarchy contains 20 Regions, 139 Communities, and 175 Neighborhoods. Twenty-eight Communities passed adaptive split rules; 111 remained terminal.
- GPT-5.5 labeled all 334 hierarchy nodes using bounded evidence packets: 20 Regions, 139 Communities, and 175 Neighborhoods. The final label set contains 77 high-confidence and 257 medium-confidence descriptions.
- Region-level random-seed agreement was high (median ARI 0.9899). Middle-layer agreement was moderate (median ARI 0.5378), so uncertainty is retained as a per-song assignment-stability value rather than hidden.
- Accepted Neighborhood splits had median seed ARI 1.000, median internal edge-weight retention 0.9805, median embedding-cohesion gain 0.0149, and median Jensen–Shannon topic separation 0.0128.
- The final public hierarchy JSON is approximately 2.97 MiB and contains song identifiers, hierarchy assignments, LLM-authored titles and descriptions, compact quantitative evidence, and no lyric text or dense embedding vectors.

2 Problem Definition and Design Goals

The original StanLyric application answers a lexical information-retrieval question: given a remembered lyric fragment, which songs contain the strongest BM25 evidence? The embedding work asks a different but complementary question:

> How can an unlabeled lyrics corpus be organized into semantically meaningful, inspectable, and navigable groups without training a supervised classifier?

The intended output was not merely a clustering score. It needed to support:

1. **Semantic exploration.** Nearby songs should have related lyrical content even when they do not share exact query words.
2. **Multiple scales.** A user should be able to move between broad areas and narrower neighborhoods.
3. **Interpretability.** Every grouping should be accompanied by lexical, geometric, graph, artist, and topic-model evidence.
4. **Reproducibility.** API calls, checkpoints, corpus identity, random seeds, model settings, and resolution experiments should be recorded.
5. **Static deployment.** The final portfolio must run on GitHub Pages without a backend, a live Cohere call, or public lyric redistribution.

These requirements favored a graph-based unsupervised workflow. Dense embeddings supplied semantic coordinates, UMAP supplied a local-neighborhood graph and a visual projection, Leiden supplied scalable graph partitions, and lexical/NMF analyses supplied explanations.

3 System Overview

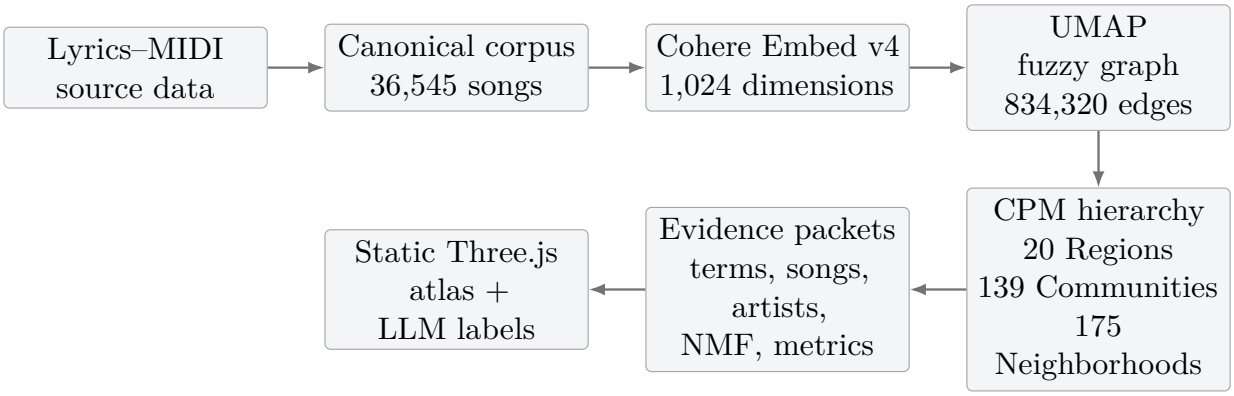

Figure 1: End-to-end semantic analysis and deployment pipeline.

The expensive representation step is performed once. All later analyses consume the saved embedding matrix, song index, graph, and model artifacts. This separation makes the community experiments repeatable without additional Cohere API calls.

4 Corpus Construction

4.1 Source and canonical schema

The source was the English Lyrics–MIDI Dataset published on Hugging Face under CC BY-NC-SA 4.0 [4]. The embedding analysis used the lyric subset, not the MIDI data. Arbitrary source columns were normalized to:

`doc_id`, `title`, `artist`, `lyrics`, `source`, and `lyrics_char_len`.

Lyrics shorter than 80 characters were excluded before the explicit cleaning stage. The final corpus contains 6,880 distinct artist strings and 30,705 distinct title strings. Median lyric length is 1,187 characters; mean length is 1,324.5 characters.

4.2 Auditable cleaning and deduplication

Normalization and cleaning were intentionally separate. The first stage exposed the source irregularities; the second stage recorded every removal. Table 1 gives the realized audit.

Table 1: Corpus cleaning audit.

Stage or removal reason	Rows
Prepared input	47,474
Unknown artist removed	4,904
Missing title removed	18
Exact lyric-copy rows removed	2,779
Normalized artist–title duplicate rows removed	3,228
Final canonical corpus	**36,545**

Exact lyrics were compared after case-folding and whitespace normalization. There were 1,569 exact-content duplicate groups. When several metadata labels described identical content, the ranking preferred the most frequent credible artist label, penalized obvious source-generated title artifacts, and used stable source order as a final tie-breaker.

Song identity was defined by case-, punctuation-, accent-, and whitespace-normalized artist and title components. There were 3,033 duplicate identity groups after exact-content cleaning. The retained row maximized a lyric quality score and then lyric length. Explicitly named versions such as *live*, *remix*, *acoustic*, and *demo* remained separate. This policy removed repeated records such as duplicate copies of "Lose Yourself" while preserving legitimate covers and versions such as different recordings of "Smooth Criminal."

5 Cohere Embedding Acquisition

5.1 Why Cohere Embed v4

Cohere Embed converts text into vectors suitable for semantic comparison, classification, and clustering. Embed v4 supports multiple output dimensions and a dedicated `clustering` input type [1, 2]. It also supports long and multilingual inputs, although this project deliberately capped text well below the model context limit.

The choice was pragmatic:

- the task required a semantic representation richer than lexical overlap;
- the corpus was unlabeled, so a clustering-oriented pretrained representation avoided annotation;
- 1,024 dimensions provided substantial capacity while reducing storage by one third relative to the 1,536-dimensional default;
- float output preserved the full representation for the first research pass; lower-precision compression remains a deployment experiment.

5.2 Text sent to the API

One API input was constructed per canonical song:

```
Title: <song title>
Artist: <artist name>
```

```
Lyrics:
<normalized lyric text>
```

Horizontal whitespace and repeated blank lines were normalized. The resulting string was truncated to 6,000 characters. Only 20 of 36,545 inputs reached the cap. The median API input was 1,239 characters and the mean was 1,375.8.

Including title and artist gives the model disambiguating metadata while leaving lyrics as the dominant portion of nearly every input. This choice also creates a known confound: songs by the same artist can become closer partly because the artist name is shared. A future ablation should compare lyrics-only, title-plus-lyrics, and full metadata inputs.

5.3 Realized API configuration

Table 2: Cohere embedding configuration recorded in the final manifest.

Parameter	Value
Model	`embed-v4.0`
Input type	`clustering`
Embedding type	`float`
Output dimension	1,024
Maximum texts per request	96
Configured input rate	1,900 texts/minute
Checkpoint interval	10 API batches
Text cap	6,000 characters
Title/artist prefix	enabled
Whitespace normalization	enabled

The complete run required 381 successful batched requests:

$$\left\lceil \frac{36{,}545}{96} \right\rceil = 381.$$

The request log contains three terminal failures from interrupted attempts and 807 retry events, primarily associated with trial-key rate and monthly-call limits encountered during development. These events did not duplicate final rows because completed document identifiers were read from checkpoint metadata before every resumed run.

5.4 Resumability and provenance

The extraction was checkpointed because API calls were expensive relative to local analysis. The final run produced 40 embedding parts and a memory-mappable matrix of shape $36{,}545 \times 1{,}024$. Operational details, saved artifact paths, and primary reproduction commands are collected in Appendix A.

6 From Embeddings to a Song Graph

6.1 Cosine geometry

For normalized song vectors $x_i, x_j \in \mathbb{R}^{1024}$, semantic similarity is measured by cosine similarity:

$$s(i,j) = \frac{x_i^\top x_j}{\|x_i\|_2 \|x_j\|_2}.$$

Cosine geometry is appropriate because direction encodes semantic content while vector magnitude is not treated as relevance.

6.2 UMAP projection and fuzzy graph

UMAP constructs a weighted fuzzy neighborhood graph before optimizing a low-dimensional layout [5]. The project fitted UMAP directly to the 1,024-dimensional embeddings with cosine distance, 30 neighbors, `min_dist`=0.12, three output dimensions, and random seed 42. The symmetric upper triangle of UMAP's learned graph was retained as the community graph. Community detection therefore used high-dimensional neighborhood memberships, not Euclidean distances between the rendered 3D points.

The final graph has:

- 36,545 song nodes;
- 834,320 undirected weighted edges;
- one connected component containing all songs;
- mean edge membership 0.1906 and median 0.1213;
- unweighted density 0.001249 and weighted density 0.000238.

This sparse graph retains local semantic relationships while avoiding the $O(n^2)$ storage of all pairwise similarities.

6.3 Projection diagnostics

UMAP is nonlinear and has no PCA-style explained-variance ratio. It was evaluated with trustworthiness and top-15 neighbor overlap on a fixed sample of 2,500 songs. A three-component randomized PCA model supplied a linear baseline.

Table 3: Three-dimensional projection diagnostics.

Projection	Trustworthiness	Top-15 neighbor overlap	Explained variance
UMAP	0.7534	0.1291	not defined
PCA	0.7379	0.0484	0.0609

UMAP preserved substantially more exact local neighbors than PCA, although an overlap of 0.1291 also warns that the atlas is an exploratory view rather than a lossless map. The browser exposes the projection diagnostics instead of describing visual distance as exact semantic distance.

7 Initial Flat Leiden Communities

7.1 Baseline construction

The first graph partition used weighted Leiden with the RBConfiguration objective, resolution 1.0, seed 42, and optimization until convergence. Leiden was preferred to Louvain because its refinement phase avoids arbitrarily badly connected communities and provides stronger connectivity guarantees [6].

Table 4: Original flat Leiden baseline.

Metric	Value
Communities	50
Weighted modularity	0.5691
Largest community	5,651 songs
Median community	451 songs
Smallest community	39 songs
Median embedding cohesion	0.6324
Median conductance	0.3825
Median normalized artist entropy	0.7705
Sampled cosine silhouette mean	-0.0114
Negative silhouette share	69.7%

The positive modularity and interpretable lexical labels demonstrated meaningful graph structure. Large groups emphasized themes such as romantic language, death and fear, dance and groove, country imagery, and rock-and-roll vocabulary. However, the partition also exposed three limitations:

1. **One semantic scale.** A 5,651-song group and a 39-song group were presented at the same conceptual level.
2. **Arbitrary resolution.** Resolution 1.0 yielded a useful result, but the exact count of 50 was not validated across random seeds or nearby resolutions.
3. **Overlapping geometry.** Near-zero negative silhouette values showed that many groups are not separated as convex clouds in the original embedding space.

The flat result was retained as an archived baseline, not discarded. Its comparison with the final 139-Community middle layer gives ARI 0.2216 and normalized VI 0.3274, indicating substantial reorganization rather than a simple subdivision of all 50 groups.

8 Making Communities Interpretable

Dense embedding dimensions are not literal words, so the model was explained using several independent views.

8.1 Binary-incidence class-based TF–IDF

The lyric vocabulary contains unigrams and bigrams with document frequency at least 5, maximum corpus frequency 75%, and a maximum of 40,000 features. Each term counts at most once per song. This binary incidence prevents repeated choruses from multiplying a label's importance.

Let $n_{c,t}$ be the number of songs in group c containing term t, $L_c = \sum_t n_{c,t}$, g_t the corpus song count for t, and $\overline{L}$ the average L_c. The implemented class-based weight is:

$$\mathrm{cTFIDF}_{c,t} = \frac{n_{c,t}}{L_c} \log\left(1 + \frac{\overline{L}}{g_t}\right).$$

The prevalence lift is:

$$\mathrm{lift}_{c,t} = \frac{n_{c,t}/|c|}{g_t/N}.$$

Terms must have lift above one and positive smoothed log-odds z-score. Repeated-token phrases such as "love love" are removed. Display ranking uses:

$$\mathrm{distinctiveness}_{c,t} = \mathrm{cTFIDF}_{c,t} \log(\mathrm{lift}_{c,t}).$$

This resembles the class-based TF–IDF approach used to explain embedding-derived clusters in BERTopic [12], with project-specific binary chorus control, log-odds filtering, and term-diversity rules.

8.2 Embedding and graph evidence

For each group c, normalized embeddings are averaged and the centroid is renormalized. Mean cohesion is:

$$\mathrm{cohesion}(c) = \frac{1}{|c|} \sum_{i \in c} x_i^\top \widehat{\mu}_c.$$

Songs with the highest $x_i^\top \widehat{\mu}_c$ are centroid representatives.

Weighted graph conductance is:

$$\phi(c) = \frac{\mathrm{cut}(c, V \setminus c)}{\min(\mathrm{vol}(c), \mathrm{vol}(V \setminus c))}.$$

Internal strength fraction is $2w_{\mathrm{in}}(c)/\,\mathrm{vol}(c)$. Boundary songs are ranked by the fraction of their incident edge weight that leaves the group and by the diversity of neighboring groups.

8.3 Artist diversity and silhouette

Artist concentration is summarized by distinct artist count, leading artist share, Shannon entropy, normalized entropy, and effective artist count. These metrics help detect groups that merely reproduce artist identity—an important check because artist metadata was included in the embedding input.

Cosine silhouette was estimated on a stratified 3,000-song sample, guaranteeing representation for every group. Silhouette compares within-group distance with distance to the nearest alternative group [10]. It is retained as a diagnostic, but not used as the sole selection target because Leiden optimizes graph communities, not spherical clusters.

8.4 Independent NMF topic bridge

Before the hierarchy work, a separate non-negative matrix factorization model was selected over TF–IDF lyric features. NMF provides additive, interpretable parts-based representations [11]. The search compared 20, 30, 40, 50, and 60 topics over seeds 42 and 123. Forty topics were selected with mean NPMI coherence 0.1711, diversity 0.8600, and seed stability 0.8975.

For each graph group, normalized NMF topic shares are averaged and compared with the corpus mean. The bridge score is:

$$B_{c,k} = \overline{\theta}_{c,k} \log_2 \left(1 + \frac{\overline{\theta}_{c,k}}{\overline{\theta}_{\text{corpus},k}} \right).$$

This bridge is valuable precisely because it is independent of Cohere's opaque embedding coordinates: dense semantic groups are corroborated with a word-derived model.

9 Hierarchical Community Method

9.1 Why CPM and three named levels

The final hierarchy uses the Constant Potts Model, a local resolution-parameterized objective designed to avoid the global resolution limit associated with modularity [7]. For a weighted graph partition $\mathcal{C}$,

$$Q_{\text{CPM}}(\mathcal{C};\gamma) = \sum_{c\in\mathcal{C}} \left[w_c - \gamma \binom{n_c}{2} \right],$$

where w_c is internal edge weight, n_c is community size, and γ controls density. Larger γ produces finer groups.

Three levels were chosen because they correspond to distinct user questions:

Region:	Which broad semantic area contains the song?
Community:	Which stable middle-scale lyrical grouping contains it?
Neighborhood:	Is there a smaller, well-supported local subdivision?

This is a strict hierarchy, not three independent clusterings: each Neighborhood has exactly one Community parent, and each Community has exactly one Region parent.

9.2 Middle layer: multiseed resolution selection

The middle layer was discovered first because it needed enough resolution to be interpretable while remaining a stable unit for both aggregation and recursive splitting. CPM was evaluated at eight resolutions:

$$\{0.00022, 0.00026, 0.00030, 0.00035, 0.00041, 0.00048, 0.00056, 0.00065\}.$$

Every resolution was run for seeds $11, 29, 42, 71, 101$, four Leiden iterations per run.

Seed agreement was measured with adjusted Rand index (ARI), corrected for chance [8], and normalized variation of information (NVI), an information distance between partitions [9]. The seed whose partition had the best mean ARI and NVI agreement with the others became the medoid. Medoid partitions at adjacent resolutions were also compared.

A candidate was eligible when it contained 45–190 Communities, no group held more than 16% of songs, and fewer than 3% of songs belonged to groups smaller than 30. Eligible candidates received:

$$S_{\text{middle}} = 0.55A_{\text{seed}} + 0.25A_{\text{adj}} + 0.15(1 - p_{\max}) + 0.05(1 - p_{\text{small}}).$$

Rather than always choosing the numerically largest score, the procedure chose the lowest resolution within 0.02 of the best eligible score. This parsimony rule selected $\gamma = 0.00048$, 139 Communities, and medoid seed 29.

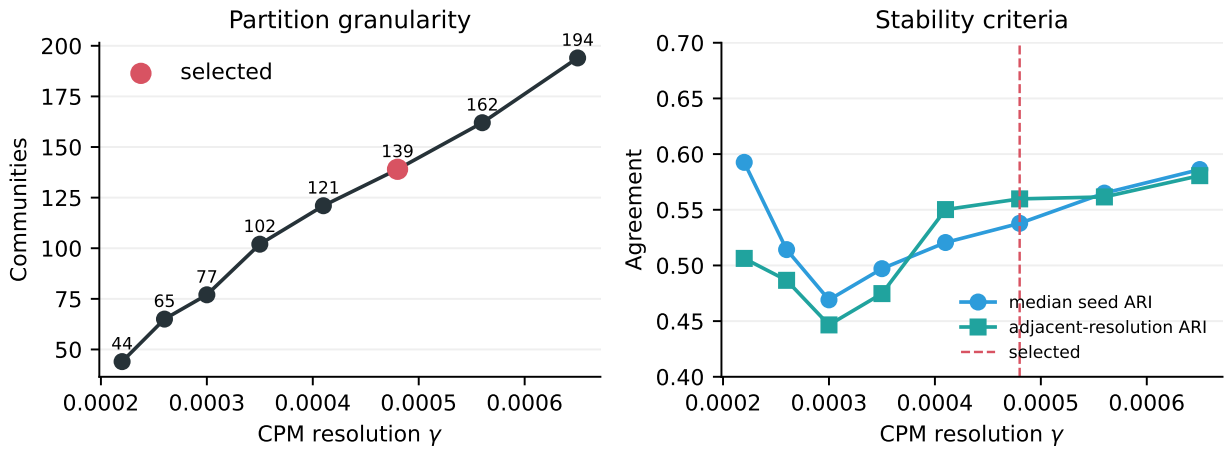

Figure 2: Middle-layer granularity and stability across the CPM resolution grid. The selected 139-Community solution is marked in red.

Table 5: Middle CPM resolution sweep. ARI is higher-is-better; NVI is lower-is-better.

γ	Groups	Seed ARI	Seed NVI	Adjacent ARI	Largest share	Status
0.00022	44	0.593	0.249	0.506	0.211	ineligible
0.00026	65	0.514	0.272	0.487	0.151	eligible
0.00030	77	0.469	0.275	0.447	0.105	eligible
0.00035	102	0.497	0.261	0.475	0.059	eligible
0.00041	121	0.521	0.243	0.550	0.055	eligible
0.00048	**139**	**0.538**	**0.236**	**0.560**	**0.051**	**selected**
0.00056	162	0.565	0.212	0.562	0.043	near-optimal
0.00065	194	0.586	0.201	0.581	0.042	ineligible

The selected ARI of 0.5378 is not presented as perfect stability. It says that the graph contains meaningful but ambiguous middle-scale boundaries. To preserve that information, every song receives an overlap-aligned assignment-stability score across seeds.

9.3 Coarse layer: Regions on the quotient graph

The 139 Communities were contracted into a weighted quotient graph. Edges between two quotient nodes equal the sum of all original cross-Community edge weights; each quotient node carries its

number of songs. CPM was then run on this graph at multiples of the selected middle resolution, again using five seeds and four iterations.

Region candidates were restricted to 6–24 groups and a largest Region share at most 45%. Their score was:

$$S_{\text{region}} = 0.60A_{\text{seed}} + 0.25A_{\text{adj}} + 0.15(1 - p_{\max}).$$

The selected resolution was $\gamma = 0.0001344$, yielding 20 Regions with medoid seed 11, seed ARI 0.9899, and seed NVI 0.0102. The largest Region holds 44.08% of songs, narrowly satisfying the balance gate. It is a broad love/heart/loneliness area, which is substantively plausible but remains an important coarse-layer imbalance.

9.4 Fine layer: constrained adaptive Neighborhoods

Every Community smaller than 200 songs became a terminal Neighborhood without further optimization. Larger Communities were independently tested at

$$\gamma_{\text{middle}} \times \{1.35, 1.65, 2.00, 2.40\} = \{0.000648, 0.000792, 0.000960, 0.001152\},$$

using seeds 11, 42, and 71.

A proposed split was accepted only if:

$$\begin{aligned} \text{children} &\geq 2, \\ \min_j |c_j| &\geq 30, \\ \text{medianARI} &\geq 0.80, \\ \text{edgeRetention} &\geq 0.55, \\ \Delta\,\text{cohesion} &\geq 0.003, \\ \text{JS}_{\text{topic}} &\geq 0.001. \end{aligned}$$

Edge retention is the fraction of the parent subgraph's total edge weight that remains internal after splitting. Cohesion gain compares size-weighted child centroid cohesion against parent cohesion. Topic separation is the size-weighted Jensen–Shannon divergence between child NMF topic profiles and the parent profile [13]. These gates require a split to be repeatable, structurally conservative, geometrically useful, and lexically distinguishable.

Twenty-eight Communities passed. Together they produced 64 child Neighborhoods; the other 111 Communities remained one-to-one terminal Neighborhoods, giving $64 + 111 = 175$ total.

10 Results

10.1 Hierarchy size and stability

Table 6: Final hierarchy structure.

Level	Nodes	Minimum	Median	Mean	Maximum
Regions	20	54	306.5	1,827.3	16,108
Communities	139	49	192.0	262.9	1,865
Neighborhoods	175	35	159.0	208.8	1,865

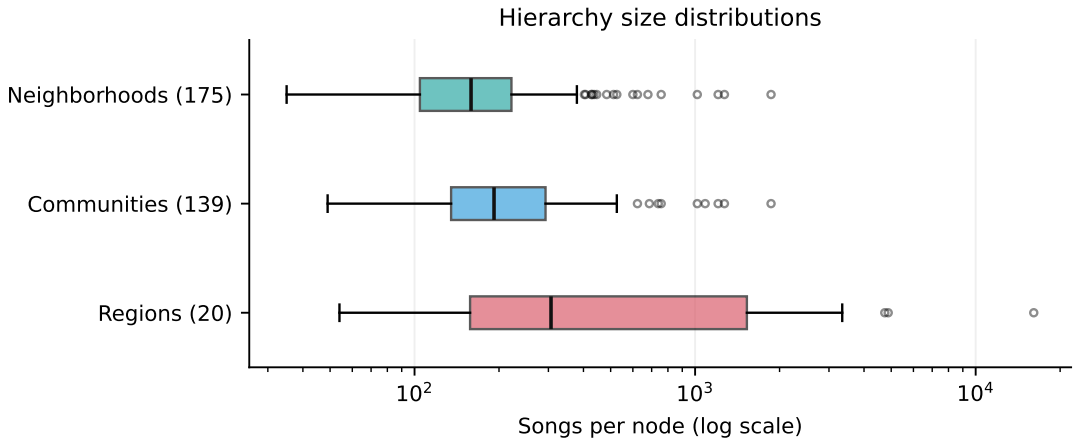

Figure 3: Node-size distributions. The logarithmic axis makes the long-tailed Region layer visible without hiding the smaller nodes.

Table 7: Partition stability. Assignment stability is measured per song after overlap-aligning labels across seeds.

Level	Seed ARI	Seed NVI	Mean assignment stability	Median stability
Regions	0.9899	0.0102	0.9977	1.0000
Communities	0.5378	0.2355	0.7239	0.8000
Neighborhoods	gate ≥ 0.80	–	0.9981	1.0000

The Neighborhood mean includes terminal one-to-one Communities, whose assignment is stable by construction. It should therefore be interpreted together with the accepted-split statistics in Table 8, not as an independent global clustering score.

10.2 Accepted split quality

Table 8: Realized metrics for the 28 selected Neighborhood splits.

Metric	Minimum	Median	Maximum
Parent songs	204	325	1,086
Children	2	2	3
Median seed ARI	0.8043	1.0000	1.0000
Internal weight retention	0.9266	0.9805	0.9948
Embedding cohesion gain	0.0041	0.0149	0.0384
NMF topic JS separation	0.0042	0.0128	0.0261

All selected splits cleared the configured thresholds by meaningful margins, especially graph retention and topic separation. The method was conservative: it preferred a terminal parent over a small or unstable refinement.

10.3 Structural and semantic diagnostics

Table 9: Median node diagnostics by level.

Level	Cohesion	Conductance	Internal fraction	Silhouette	Artists
Regions	0.6201	0.2817	0.7183	-0.0062	80.5
Communities	0.6411	0.4562	0.5438	-0.0145	69.0
Neighborhoods	0.6481	0.4707	0.5293	-0.0180	57.0

Cohesion rises as the hierarchy becomes finer, as expected. Conductance also rises and internal fraction falls because smaller groups expose more boundary edges. These metrics answer different questions and should not be collapsed into one "quality" number.

The global sampled silhouette means are -0.0145 for Regions, -0.0081 for Communities, and -0.0130 for Neighborhoods. Their small negative values indicate overlapping semantic neighborhoods rather than clean hyperspherical clusters. This does not contradict positive graph structure: Leiden optimizes weighted connectivity on a manifold graph, while silhouette compares every sampled point to centroid-free distance aggregates in the original space. The result should be described as a graph partition of overlapping language, not a set of well-separated Gaussian clusters.

10.4 Qualitative structure

The largest Region contains 16,108 songs and is characterized by love, heart, holding, and loneliness. Other broad Regions emphasize death/blood/fear, country/rock-and-roll imagery, Christmas/religious language, New York, and language containing profanity. These labels are descriptive outputs of the corpus and are not manually curated genre labels.

Middle Communities are more focused. Examples include:

- a 1,272-song death/flesh/blood group;
- a 1,208-song country/cowboy/train group;
- a 687-song rock-and-roll group;
- a 623-song dance/boogie/groove group;
- a 446-song sea/sailing/shore group.

Accepted fine splits show both the value and limits of automatic labels. A 687-song rock-and-roll Community separates into a 403-song core rock/rock-and-roll Neighborhood, a 247-song rolling-stone Neighborhood, and a 37-song residual group. A 494-song woman/lady Community separates into woman-centered, lady-centered, and smaller mixed-language groups. Some residual labels are noisy; this is why the system exposes representative songs, topic bridges, and diagnostics rather than treating four terms as a complete semantic definition.

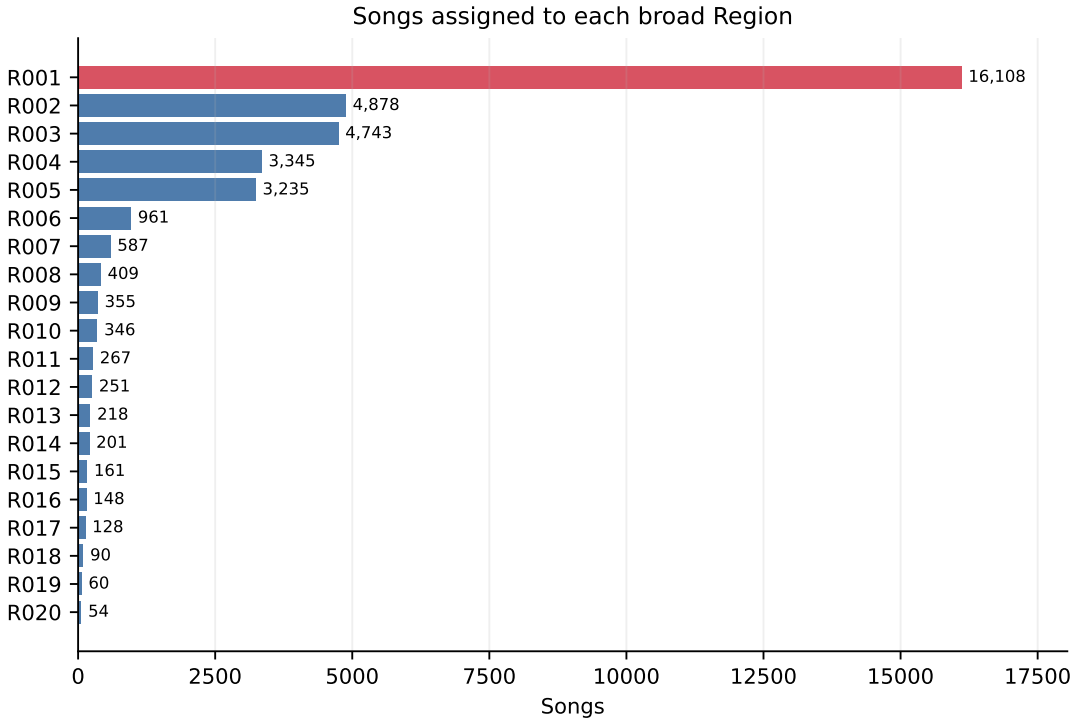

Figure 4: Region sizes. The selected coarse solution is stable, but the first Region remains intentionally broad.

10.5 Case study: "Lose Yourself"

The canonical Eminem recording maps to:

$$\texttt{R004} \rightarrow \texttt{C117} \rightarrow \texttt{N0153}.$$

`R004` is a 3,345-song broad Region dominated by explicit hip-hop vocabulary. `C117` is a 113-song Community labeled by terms associated with Dr. Dre, Shady, Slim, and Marshall; `N0153` is terminal and carries the same 113 songs. Region and Neighborhood assignment stability are 1.0, while the song's middle Community stability is 0.6. This is an informative result: the model places the song

in a coherent artist/hip-hop neighborhood, but nearby middle-scale boundaries move across seeds. The interface retains that uncertainty instead of converting it into false confidence.

11 LLM-Assisted Hierarchy Labeling

The graph and lexical evidence above produced technically useful node labels, but four slash-separated terms are not enough for public explanation. They can also over-emphasize sensitive words when a large group is characterized by explicit language. The final portfolio therefore adds a controlled LLM-labeling layer: GPT-5.5 receives a bounded evidence packet for each node and returns a title, summary, description, confidence flag, and evidence list.

11.1 Evidence packet

The labeling step uses only derived evidence. It does not send full lyrics, dense embeddings, or arbitrary neighboring text. Each node packet contains:

- node identity, level, rank, size, terminal status, and the original term-based label;
- embedding and graph diagnostics such as cohesion, conductance, internal strength fraction, sampled silhouette, and assignment stability;
- up to 12 distinctive c-TF–IDF terms with song prevalence, lift, smoothed log-odds, and distinctiveness;
- up to five NMF topic bridges, each represented by topic share, lift, and top topic terms;
- up to eight centroid-representative songs and six boundary songs;
- the 10 most represented artists, including dominant-artist share;
- compact parent, child, and sibling snapshots so the model can make titles distinct at the appropriate hierarchy level;
- a corpus summary confirming 36,545 songs, 20 Regions, 139 Communities, 175 Neighborhoods, and strict nesting.

The run manifest records a 14,000-character evidence budget per node. This keeps prompts large enough to include the main evidence but small enough to avoid turning each request into a raw data dump.

11.2 Prompt contract and outputs

The prompt asks the model to use only the provided evidence, avoid quoting lyrics, avoid unsupported genre or era claims, and avoid gratuitous profanity in titles. It also tells the model that Regions should sound broad, Communities more specific, and Neighborhoods more fine-grained. The requested JSON shape is:

```
{
  "labels": [{
    "node_id": "R001",
    "title": "2 to 8 words",
    "one_line_summary": "one sentence under 35 words",
    "description": "2 to 4 concise sentences",
    "confidence": "high|medium|low",
    "evidence_used": ["term or song"]
```

```
  }]
}
```

For example, the largest Region originally had the term label `love / heart / hold / lonely`. The LLM title became *Romantic Longing and Heartache*, with a summary that mentions love, need, hearts, closeness, loneliness, and goodbye. This is more readable than the raw term label while still anchored in the same evidence.

11.3 Validation and deployment

The generation run used prompt version `stanlyric_hierarchy_labeling_v1` and produced one valid record for each of the 334 hierarchy nodes: 20 Regions, 139 Communities, and 175 Neighborhoods. Seventy-seven labels were marked high confidence and 257 medium confidence; none were marked low confidence. The export step joined those records back onto the hierarchy nodes while preserving the original term labels as diagnostic evidence.

The LLM output is treated as presentation metadata, not as the source of the clustering. It cannot change membership, graph metrics, representatives, boundary songs, or stability scores. This separation is important: Cohere embeddings and Leiden define the structure; c-TF–IDF, NMF, artists, and graph diagnostics define the evidence; GPT-5.5 turns that evidence into readable language for the portfolio.

12 Static Portfolio Deployment

The research artifacts are larger and richer than a public browser needs. The deployment therefore separates analysis from presentation:

- `song_embeddings_float32.npy`: 142.8 MiB research matrix, not deployed;
- `stanlyric_embedding_atlas.json`: approximately 4.25 MiB, containing quantized 3D coordinates, metadata, hierarchy-independent graph neighbors, and no embeddings;
- `stanlyric_hierarchy.json`: approximately 2.97 MiB, containing hierarchy assignments, LLM labels, node metrics, terms, NMF bridges, representatives, and boundary songs;
- no public payload contains lyric text.

The Three.js page renders one point per song and supports title/artist search, rotation, zoom, semantic-neighbor links, and filtering at the Region, Community, or Neighborhood level. Selecting a song shows its complete strict hierarchy path. A synchronized explorer provides parent/child navigation and the complete interpretability contract, including the LLM-authored title and description for the selected node. This transforms the hierarchy from an offline table into an inspectable information-retrieval interface.

13 Limitations and Threats to Validity

1. **No labeled semantic ground truth.** Stability, graph structure, cohesion, topic separation, and manual inspection are complementary internal evidence. They do not prove that every group matches a human genre or theme.

2. **Moderate middle-layer stability.** Seed ARI 0.5378 and a 24.6% share of songs below 0.5 assignment stability show that many Community boundaries remain ambiguous.
3. **Broad coarse group.** The largest Region contains 44.1% of the corpus. It passes the predefined gate and is highly stable, but it limits coarse-level balance.
4. **Non-spherical structure.** Negative silhouette estimates warn against interpreting the groups as separated Euclidean balls.
5. **Metadata leakage.** Title and artist were intentionally embedded. Artist-level proximity may therefore reflect explicit metadata in addition to lyric style.
6. **Truncation.** Twenty unusually long songs were truncated at 6,000 characters. Later verses in those songs do not influence their embeddings.
7. **Corpus bias and quality.** The dataset is neither a complete catalog nor a statistically representative sample of music. Source transcriptions, artist naming, language coverage, and version labeling vary.
8. **Language-sensitive interpretation.** Cohere Embed v4 is multilingual, but the lexical interpretation uses English stop words and an English-oriented token pattern.
9. **LLM label interpretation.** GPT-5.5 titles and descriptions are evidence-grounded summaries, not human annotations or model-internal truths. They can over-generalize narrow evidence, so the interface keeps the quantitative evidence visible beside them.
10. **Sensitive language.** Unsupervised labels faithfully surface profanity and slurs present in lyrics. Public presentation requires context and should not imply endorsement.
11. **Proprietary model snapshot.** API-hosted model behavior can evolve. The manifest preserves model name and configuration, but not the model weights.
12. **Static hierarchy.** New songs require embedding, graph reconstruction or insertion, and renewed stability analysis.
13. **Dataset license.** The source is non-commercial share-alike. Public artifacts intentionally omit lyrics, but downstream use must still review dataset and song-rights obligations.

14 Conclusions and Next Experiments

The project demonstrates that a single set of Cohere song embeddings can support more than nearest-neighbor lookup. The same representation powers a local semantic graph, a reproducible 3D atlas, graph community detection, a strict multiscale hierarchy, representative-song selection, cross-model interpretation, and evidence-grounded LLM node labels.

The strongest result is not a particular community count. It is the construction of an auditable decision process: candidate resolutions are retained; random seed disagreement is measured; coarse aggregation preserves parentage; fine splits must improve multiple independent criteria; lexical and NMF explanations are computed after grouping; LLM descriptions are generated only after evidence is assembled; and uncertainty is exposed per song.

The next highest-value experiments are:

1. ablate title and artist metadata from embedding input;
2. compare 512-, 1,024-, and 1,536-dimensional Embed v4 outputs, including int8 storage;
3. add a small human evaluation of pair similarity and hierarchy labels;
4. test alternative graph neighborhood sizes and consensus partitions;
5. fit an out-of-sample assignment method for newly added songs;
6. compare content-based recommendation quality against lexical LSA and hybrid BM25–embedding

retrieval;
7. study overlapping or soft community membership for low-stability songs.

References

[1] Cohere. *Cohere's Embed Models: Details and Application*. https://docs.cohere.com/docs/cohere-embed. Accessed June 2026.

[2] Cohere. *Embed API (v2) Reference*. https://docs.cohere.com/v2/reference/embed. Accessed June 2026.

[3] Cohere Team. Introducing Embed 4: Multimodal search for business. Cohere Blog, 2025. https://cohere.com/blog/embed-4.

[4] Alexandr Sigalov. *Lyrics–MIDI Dataset*. Hugging Face Datasets, DOI 10.57967/hf/8809. https://huggingface.co/datasets/asigalov61/Lyrics-MIDI-Dataset.

[5] Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv:1802.03426*, 2018; revised 2020. https://arxiv.org/abs/1802.03426.

[6] Vincent A. Traag, Ludo Waltman, and Nees Jan van Eck. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9:5233, 2019. doi:10.1038/s41598-019-41695-z.

[7] Vincent A. Traag, Paul Van Dooren, and Yurii Nesterov. Narrow scope for resolution-limit-free community detection. *Physical Review E*, 84:016114, 2011. doi:10.1103/PhysRevE.84.016114.

[8] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985. doi:10.1007/BF01908075.

[9] Marina Meilă. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007. doi:10.1016/j.jmva.2006.11.013.

[10] Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. doi:10.1016/0377-0427(87)90125-7.

[11] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999. doi:10.1038/44565.

[12] Maarten Grootendorst. BERTopic: Neural topic modeling with a class-based TF–IDF procedure. *arXiv:2203.05794*, 2022. https://arxiv.org/abs/2203.05794.

[13] Jianhua Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. doi:10.1109/18.61115.

A Reproducibility and Artifact Inventory

A.1 Primary commands

From the repository root:

```
python -u -m scripts.embed_songs_cohere \
  --corpus data/processed/corpus.parquet \
  --output-dir data/processed/cohere_embeddings \
  --model embed-v4.0 \
  --input-type clustering \
  --output-dimension 1024 \
  --batch-size 96 \
  --inputs-per-minute 1900 \
  --checkpoint-batches 10

python -u -m scripts.build_embedding_atlas \
  --embedding-dir data/processed/cohere_embeddings \
  --output-dir outputs/embedding_communities \
  --neighbors 30 \
  --min-dist 0.12 \
  --resolution 1.0 \
  --seed 42

python -u -m scripts.analyze_embedding_communities

python -u -m scripts.build_hierarchical_communities

export OPENAI_MODEL="gpt-5.5"
python -u -m scripts.label_hierarchy_nodes --generate --batch-size 1

python -u -m scripts.export_hierarchy_labels
```

The companion notebook performs the embedding-community and hierarchy analysis through imported functions and displays candidate tables, hierarchy sizes, split diagnostics, and node-level evidence:

`notebooks/07_embedding_leiden_communities.ipynb`

A.2 Embedding resumability and provenance

The embedding extraction infrastructure was designed around scarce API calls:

1. A SHA-256 fingerprint over stable corpus metadata prevented incompatible corpora from sharing an output directory.
2. Each successful batch appended request timing, estimated token load, limiter sleep, and row boundaries to a JSONL log.
3. Every ten batches, embeddings were compressed into an NPZ part and aligned public metadata were written to Parquet.
4. The manifest recorded the model, text preparation, dimensions, rate settings, completed parts, and final artifact paths.
5. On interruption, in-memory completed batches were flushed before the exception was re-raised.
6. Assembly verified row counts and dimensions, then wrote a memory-mappable float32 matrix and a row-aligned song index.

The final embedding artifact consists of 40 parts and a matrix of shape $36{,}545 \times 1{,}024$. Its theoretical

float32 size is:

$$36{,}545 \times 1{,}024 \times 4 = 149{,}688{,}320 \text{ bytes} \approx 142.8 \text{ MiB}.$$

A.3 Primary saved artifacts

Table 10: Primary saved artifacts.

Path	Purpose
`data/processed/corpus.parquet`	Canonical lyrics corpus and stable song identifiers.
`data/processed/cohere_embeddings/manifest.json`	Corpus fingerprint, API/text configuration, checkpoint inventory, and final matrix provenance.
`data/processed/cohere_embeddings/song_embeddings_float32.npy`	Memory-mappable 36,545 × 1,024 embedding matrix.
`data/processed/cohere_embeddings/song_embedding_index.parquet`	Row-aligned public song metadata.
`outputs/embedding_communities/embedding_knn_edges.parquet`	UMAP fuzzy graph edge list used by Leiden.
`outputs/embedding_communities/embedding_projection_diagnostics.json`	UMAP and PCA fidelity metrics.
`outputs/embedding_communities/leiden_song_communities.parquet`	Archived 50-community baseline membership.
`outputs/hierarchical_communities/middle_resolution_stability.parquet`	Multiseed and adjacent-resolution diagnostics for the middle layer.
`outputs/hierarchical_communities/region_resolution_stability.parquet`	Coarse quotient-graph candidate diagnostics.
`outputs/hierarchical_communities/neighborhood_split_diagnostics.parquet`	Every tested recursive split, gate result, score, and selected candidate.
`outputs/hierarchical_communities/hierarchical_song_membership.parquet`	Strict Region–Community–Neighborhood path and per-song stability.
`outputs/hierarchical_communities/hierarchy_terms.parquet`	Complete c-TF–IDF, prevalence, lift, and log-odds evidence.
`outputs/hierarchical_communities/hierarchy_nmf_topics.parquet`	NMF topic bridges for all 334 hierarchy nodes.
`outputs/hierarchical_communities/hierarchical_communities.graphml`	Portable hierarchy graph for external graph tools.
`outputs/hierarchical_communities/labeling/node_evidence.jsonl`	Bounded evidence packets sent to the LLM.

Continued on next page

Path	Purpose
`outputs/hierarchical_communities/labeling/hierarchy_node_descriptions.jsonl`	Validated GPT-5.5 titles, summaries, descriptions, confidence flags, and evidence lists.
`outputs/hierarchical_communities/labeling/llm_raw_responses.jsonl`	Raw API responses retained for audit and parser debugging.
`outputs/hierarchical_communities/labeling/stanlyric_hierarchy_labeled.json`	Final no-lyrics hierarchy payload with labels joined onto nodes.
`outputs/hierarchical_communities/stanlyric_hierarchy.json`	Compact no-lyrics browser payload before LLM labels were joined.

B Exact Configuration Snapshot

Table 11: Final hierarchy and labeling configuration.

Component	Values
Middle resolutions	0.00022, 0.00026, 0.00030, 0.00035, 0.00041, 0.00048, 0.00056, 0.00065
Middle seeds / iterations	11, 29, 42, 71, 101 / 4
Middle balance gates	45–190 groups; largest share ≤ 0.16; small-song share ≤ 0.03
Region multipliers	0.19, 0.20, 0.21, 0.22, 0.23, 0.24, 0.25, 0.26, 0.28, 0.32
Region seeds / iterations	11, 29, 42, 71, 101 / 4
Region gates	6–24 groups; largest song share ≤ 0.45
Neighborhood multipliers	1.35, 1.65, 2.00, 2.40
Neighborhood seeds / iterations	11, 42, 71 / 2
Neighborhood gates	parent ≥ 200; child ≥ 30; ARI ≥ 0.80; retention ≥ 0.55; cohesion gain ≥ 0.003; topic JS ≥ 0.001
Interpretability vocabulary	binary unigrams/bigrams; `min_df`=5; `max_df`=0.75; 40,000 features
Silhouette sample	3,000 songs, cosine metric, stratified by node
LLM prompt version	`stanlyric_hierarchy_labeling_v1`
LLM model	`gpt-5.5`
LLM evidence budget	14,000 characters per node
LLM evidence caps	12 terms; 5 NMF topics; 8 representatives; 6 boundary songs; 10 artists; 12 children; 10 siblings

C Reference Metrics Snapshot

Table 12: Compact result card for future comparisons.

Metric	Final value
Songs	36,545
Embedding dimensions	1,024
Successful API calls	381
Graph edges	834,320
UMAP trustworthiness	0.7534
UMAP top-15 overlap	0.1291
Flat baseline communities	50
Flat baseline modularity	0.5691
Regions / Communities / Neighborhoods	20 / 139 / 175
Accepted adaptive splits	28
Region seed ARI	0.9899
Middle seed ARI	0.5378
Mean Community assignment stability	0.7239
Mean Neighborhood assignment stability	0.9981
LLM-labeled hierarchy nodes	334
High / medium / low confidence labels	77 / 257 / 0