GRANTHA HERITAGE TRACKER · SIH26096 · TECHNICAL REPRODUCIBILITY DOSSIER

Reproducibility, Dataset Lineage & Numbers Explained

Cross-script unsupervised topic grouping, held-out evaluation, and 3-D Archive Map (PCA + UMAP) on Digital Library of India scans.

Live App: http://34.123.104.155:8000/ **3-D Archive Map:** galaxy.html **Study Numbers:** study.html

Commit: `57a2c27` (`fix/vertex-ai-accuracy-pipeline`)

13,358	8 / 7	37.2%	+14.5 pp
Usable books (of 15,288 harvested; 87.4%)	Languages / Scripts (up to 1960 or undated)	Cross-script k-NN@10 after centring (vs 2.7% raw)	Test T2 subject gain over best simple guess (p=0.0005)

1. Dataset & Direct Source Links

Every book in Grantha comes from the public **Digital Library of India (DLI)** mirror hosted on the Internet Archive. No proprietary or private corpus is used.

- **Primary Collection:** https://archive.org/details/digitallibraryindia
- **Internet Archive Scrape API:** https://archive.org/services/search/v1/scrape (builds the per-language sampling frame)
- **Item Metadata Endpoint:** `https://archive.org/metadata/{id}` (catalogue tags, year, page count, file list)
- **Page Image Endpoint:** `https://archive.org/download/{id}/page/n{leaf}_w1200.jpg` (5 body pages at 15%, 30%, 50%, 70%, 85% through each book)

Sampling Rules (`pipeline/config.py`)

Items are filtered to publication year `<= 1960` (or undated) with at least `20` pages (`MIN_PAGES = 20`). Up to `2,000` books per language are drawn using a fixed seed (`SEED = 2026`), stratified across 5 era bands (`pre-1900` , `1900-1919` , `1920-1939` , `1940-1960` , `undated`). Whole-book text comes from the archive's own OCR for **12,263** books (`10,581` hOCR + `1,682` DjVu) and our Tesseract 5 OCR across `77,634` pages for **1,095** books. Deterministic 60/20/20 splits are assigned by identifier hash: **8,129 train** / **2,638 validation** / **2,591 test**.

Code	Language	Script	Harvested	Usable	Retention	Median OCR Conf.
`eng`	English	Latin	2,000	1,976	98.8%	91.9%
`ben`	Bengali	Bengali	2,000	1,930	96.5%	73.4%
`guj`	Gujarati	Gujarati	2,000	1,886	94.3%	71.5%
`hin`	Hindi	Devanagari	2,000	1,806	90.3%	77.1%
`san`	Sanskrit	Devanagari	2,000	1,562	78.1%	61.0%
`kan`	Kannada	Kannada	1,608	1,460	90.8%	77.1%
`tam`	Tamil	Tamil	1,766	1,416	80.2%	63.6%
`tel`	Telugu	Telugu	1,914	1,322	69.1%	55.4%
Total (8 languages, 7 scripts)			**15,288**	**13,358**	**87.4%**	—

2. How to Reproduce Everything End-to-End

The entire pipeline runs inside the project's Docker container (Python 3.12, Tesseract 5, scikit-learn, sentence-transformers). Because embeddings use Google Vertex AI (`text-multilingual-embedding-002` , feature set `V`), place `GOOGLE_CLOUD_PROJECT` and `HERITAGE_ADC_FILE` (or `GEMINI_API_KEY` for feature set `G`) in `.env` .

```
# 1. Build pipeline image & run the automated test suite (135 passed, 8 skipped)
docker compose --profile pipeline build pipeline
docker compose run --rm pipeline python -m pytest -q

# 2. Harvest DLI books & OCR (resumable, polite rate limit: 0.2s per request)
bash scripts/harvest_all.sh
docker compose run --rm pipeline python -m pipeline.ocr --workers 11
docker compose run --rm pipeline python -m pipeline.ocr --ingest
docker compose run --rm pipeline python -m pipeline.fulltext
docker compose run --rm pipeline python -m pipeline.quality
docker compose run --rm pipeline python -m pipeline.preprocess

# 3. Fit embeddings on train split, run clustering study, and evaluate once on val/test
docker compose run --rm pipeline python -m pipeline.features --fit train
docker compose run --rm pipeline python -m pipeline.study
docker compose run --rm pipeline python -m pipeline.evaluate --split val
docker compose run --rm pipeline python -m pipeline.evaluate --split test

# 4. Refit on all 13,358 usable books, export web data & 3-D map (PCA + optional UMAP)
docker compose run --rm pipeline python -m pipeline.features --fit all
docker compose run --rm pipeline python -m pipeline.study
docker compose run --rm pipeline python -m pipeline.sensitivity
docker compose run --rm pipeline python -m pipeline.export
docker compose run --rm pipeline python -m pipeline.galaxy    # writes galaxy.json & galaxy_umap.json
docker compose run --rm pipeline python -m pipeline.figures
docker compose up --build                                   # serves app at http://localhost:8000
```

Verification gates: Every number on the website is machine-checked against `web/data/facts.json` via `python scripts/check_claims.py` (0 unsourced numbers across 23 files). All static pages pass `sloplint.mjs` and `python -m pytest -q -p no:cacheprovider`.

3. Explaining Every Number in the Study

A. Why Per-Language Centring Matters (`centring_effect`)

Raw multilingual embeddings group books by **language/script first**, not topic. To align topics across scripts, we subtract each language's mean embedding vector and L2-normalise, then project to 100 PCA dimensions (`78.6%` variance explained). Measured on the 8,129 training books:

Metric (Train Books, n = 8,129)	Raw (Uncentred)	Language Removed (Centred)	Interpretation
Script NMI (Ward `k = 3`)	0.642	0.065	Cluster-to-script mutual information drops ~10×
Same-script 10-NN share	97.3%	62.8%	Raw neighbours are almost purely same-script
Cross-script 10-NN share	2.7%	37.2%	13.6× jump in cross-script topical neighbours

B. Clustering Quality & Stability (Why Silhouette Is 0.035)

- **Ward Hierarchical (`k = 3` , Euclidean):** Selected on train books via pre-registered fallback (selection rule required `NMI <= 0.5` and bootstrap `ARI >= 0.5`). On all 13,358 books: `silhouette = 0.035` (`0.038` on train), `script NMI = 0.019` (`0.017` on train), `era NMI = 0.017` . The three groups contain **6,152** books (46.1%, majority *literature*), **1,714** books (12.8%, majority *religion & philosophy*), and **5,492** books (41.1%, majority *religion & philosophy*).
- **Why Silhouette = 0.035 is expected:** Book topics form a continuous semantic manifold rather than separated spheres, so points near cluster borders have small margins. Within individual scripts, weighted mean silhouette is similarly `0.050` .
- **Bootstrap Stability (ARI = 0.260 all / 0.357 train):** Resampling books shifts soft cluster boundaries in a continuous space (`p10 = 0.214` , `p90 = 0.398` across 20 bootstraps). Adjacent `k` cuts stay close (`ARI = 0.789` vs `k=2` , `0.815` vs `k=4`).
- **Density Clustering (DBSCAN & HDBSCAN):** DBSCAN (`eps = 0.866` , `min_pts = 15`) finds `2` clusters and flags `30.7%` of books as noise (`silhouette = 0.006`). HDBSCAN (`min_cluster_size = 134`) leaves only the densest `5.9%` core (`94.1%` noise), where `silhouette = 0.282` and agreement with Ward on non-noise points is `ARI = 0.664` .

C. Held-Out Test Targets (n = 2,591 Unseen Test Books)

Target	Result (95% CI)	Baseline / Target	Status & Meaning
T1 Page-to-book agreement	75.2% (74.1%–76.3%)	35.9% chance / 80% target	Not Met Tested on 2,362 books (2.6% abstained on weak OCR)
T2 Subject accuracy (labelled)	53.0% (43.6%–61.5%)	38.5% script / +10 pp target	Met (+14.5 pp) 117 held-out books, `p = 0.0005` (Val was +4.2 pp)
T3 Bootstrap stability (ARI)	0.357 (train) / 0.260 (all)	0.60 target	Not Met Continuous manifold boundaries shift under resampling
Exploratory Catalogue classes	47.8% (44.9%–50.6%)	51.8% script baseline	-4.0 pp 1,168 test books; 3 coarse groups don't match shelf codes

D. Archivist Tools Performance

- **Catalogue Class Suggestion (`k = 25` distance-weighted vote):** `58.6%` Top-1 accuracy (95% CI `55.8%–61.5%`) and `91.6%` **Top-3 accuracy** (95% CI `90.1%–93.1%`) on 1,168 held-out books vs `47.8%` most-common baseline.
- **Edition Finder:** `100.0%` precision and `36.4%` recall (95% CI `8.3%–66.7%`) across 11 held-out edition groups (Euclidean distance threshold `0.0447`).
- **Scan Triage:** `AUC = 0.499` (95% CI `0.452–0.546`) on 341 books — no better than chance at predicting archive-vs-Tesseract OCR disagreement.

4. Why 3-D PCA Looks Like a Ball & What 3-D UMAP Changes

Why `Language removed` PCA looks uniform: A 3-D linear PCA projection (`web/data/galaxy.json`) only retains ~ `23%` of the 100-D variance (11% + 7% + 5%), collapsing local subject pockets into a spherical cloud.
How `Neighbourhoods (3-D UMAP)` fixes it: Computed on the `13,358 × 100` centred matrix (`n_neighbors = 15` , `min_dist = 0.1` , `metric = "cosine"` , `random_state = 2026`) into `web/data/galaxy_umap.json` , UMAP preserves local cosine neighbourhoods so topic islands and filaments separate visually, while `Language | Group` colouring and one-click sample chips highlight cross-script links.

Generated from verified repository artifacts (`web/data/facts.json` , `pipeline/config.py` , `README.md` , `pipeline/galaxy.py`) on 2026-11-15. Downloadable online at http://34.123.104.155:8000/grantha-reproducibility.pdf.