Pāṇḍitya: Visualizing Sanskrit Intellectual Networks

Tyler Neill
Brooklyn, NY
tyler.g.neill@gmail.com

Abstract

Pāṇḍitya is an interactive web-based visualization tool for mapping and exploring relationships between Sanskrit authors and their works. Built on the structured data of the *Pandit Prosopographical Database of Indic Texts*, it offers scholars and students an intuitive and extensible open-source interface for dynamically investigating commentarial networks, which are an essential aspect of Sanskrit intellectual history. Originally developed as an offline tool (*Pandit Grapher*), *Pāṇḍitya* has been re-imagined as an accessible online resource leveraging modern web technologies for everyday scholarly use. Beyond visualizing these networks, *Pāṇḍitya* also links to online Sanskrit e-texts through its sister project *SETI*, and may eventually be extended to illustrate additional phenomena such as parallel passages.

1 Introduction

1.1 Problem and Solution

Sanskrit intellectual history is vast and deeply intertextual, making it challenging to gain a holistic view of relationships between works and authors, even after years of study. *Pandit*—a digital humanities project that curates, organizes, and shares structured data on South Asian literary and intellectual history through an interactive, scholar-driven prosopographical database (Bronner and others, 2024)—provides a strong foundation for addressing this challenge. Until very recently (see § 1.3), *Pandit*'s wiki-like website lacked any intuitive, at-a-glance representation of these relationships. *Pāṇḍitya* fills this gap by layering on an interactive network visualization, enabling users to trace scholarly relationships with ease and engage more deeply with the material. By bridging structured data and human comprehension, *Pāṇḍitya* serves as both a research aid and a pedagogical tool, enabling more effective exploration of Sanskrit intellectual history while lowering barriers for newcomers.

1.2 Related Network Visualization Projects

Interactive visualizations of network data are increasingly common in the humanities, but authorial and commentarial relationships are rarely complex enough to necessitate such an approach. Likewise, network visualizations have not traditionally been used for organizing access to digital literary canons. Instead, most digital humanities projects have focused on other domains:

1. Social networks, often reconstructed through documented correspondence, family ties, or professional relationships, mirroring interest in modern social media analysis:
   - *Six Degrees of Francis Bacon*, which maps early modern Britain's (1500–1700) social networks through textual analysis of primary sources (Warren and others, 2016).
   - *Mapping the Republic of Letters*, which reconstructs Enlightenment intellectual networks through letter-writing archives (Edelstein and others, 2017).

for review submission

2. Citation networks in modern academic work (scientometrics):
    - *Connected Papers*, which positions papers in a 2D similarity space based on citation patterns and semantic analysis (Tarnavsky Eitan and others, 2025).
    - *VOSviewer* (Van Eck and Waltman, 2010) and *CitNetExplorer* (Van Eck and Waltman, 2014), which analyze large-scale scientific literature through co-citation, bibliographic coupling, and co-authorship patterns .
3. Computed intertextuality phenomena, such as parallel passages and topic modeling:
    - *Open Knowledge Maps*, which organizes academic papers into topic-based clusters rather than direct citation networks (Kraker and others, 2025).
    - *Paper Machines*, an open-source Zotero extension that enables researchers to analyze bibliographic metadata and full texts using topic modeling and visualization tools (Jo and Johnson-Roberson, 2012).
    - *The Viral Texts Project*, which maps reprinting networks in 19th-century English-language newspapers and magazines (Smith and others, 2015).
    - *BuddhaNexus*, which traces intertextual connections across Sanskrit, Pali, Tibetan, and Chinese using FastText embeddings, with a primary focus on Buddhist scriptures (Nehrdich, 2020).

Perhaps the most closely related project is non-academic in nature. *The Oracle of Bacon* is a playful exploration of movie actor networks based on the "six degrees of Kevin Bacon" concept, which posits that any actor in (America-centric) show business can be connected to Kevin Bacon within six hops or fewer (Reynolds and Tjaden, 2025). This project directly inspired the v1 tool *Pandit Grapher*, as evidenced by the latter's use of the phrase "bacon hops." Surprisingly, commercial streaming services, as well as subsidiary navigation platforms like IMDb and JustWatch, have yet to adopt similar visualizations for content browsing.

1.3 Project History

The first version of the project, *Pandit Grapher*, required users to install Python, manually execute scripts, and export data for visualization in *Gephi* (Neill, 2021). While potentially useful for technically proficient users, this process was inaccessible to most Sanskrit scholars.

To address this barrier, the second iteration, *Pāṇḍitya*, was developed as a fully online and interactive visualization tool. Built with *D3.js* for dynamic graph rendering and *Flask* as a backend framework, *Pāṇḍitya* significantly lowers the technical barrier, making this sort of interaction with Sanskrit scholarly networks more accessible and engaging. Users can explore and customize visualizations in real-time using only a web browser, adjusting parameters and filtering connections as needed. This transition from an offline, static workflow to an interactive web-based tool greatly enhances its potential as a reference and research instrument. At the same time, the v1 feature of exporting data for use with offline tools like *Gephi* will be retained—currently only on the backend, but soon also on the front end—as it is particularly useful for visualizing large and dense graphs with hundreds or thousands of nodes.

In late April 2025, the *Pandit* project independently introduced a graph visualization feature on its entity pages, similar in concept to *Pāṇḍitya*. Notable advantages of the *Pandit* implementation include: (1) broader coverage of entity types with options for targeted filtering; and (2) polished interface elements such as a collapsible sidebar, refined zoom controls, and helpful tooltip displays. Alongside this official implementation, *Pāṇḍitya* will continue in its current role as an independent, open-source platform well suited to experimentation, offering space for rapid prototyping, alternate feature sets, and exploratory visualization work. Mutual acknowledgment and ongoing collaboration between the two platforms can help ensure continued benefit to the broader community.

1.4 Name Derivation

The name *Pāṇḍitya* derives from *paṇḍita* ("scholar"), which is basis for the *Pandit* project's own name. *Pāṇḍitya*'s grammatical status as a *vṛddhi* derivative (meaning "scholarship") also symbolizes its creative derivation from the predecessor project. The full name, *Pāṇḍityatāraka*, can be taken to mean either "that which helps one cross (to the far shore of) Sanskrit scholarly learning" or "a tool for navigating *Pandit* data."

2 Data

References to files below correspond to the project's GitHub repository at `https://github.com/tylergneill/panditya`.

2.1 Source

Pāṇḍitya is built upon a structured subset of the *Pandit* database, derived from a snapshot taken on December 23, 2024. This source dataset contains 67,529 entities and 163 fields (`data/2024-12-23-pandit-entities-export.csv`).

2.2 Reduced Entity Model

The dataset was filtered to focus on works (`Content type==Work`), their authors (`Content type==Person`), and selected additional information: alternate names (`Aka` and author `Social Identifiers`), work `Discipline`, and basic dates (`Highest year` and `Lowest year`). Entirely omitted were entity types such as `Manuscript` (of which there are 7,532), manuscript `Extract` (5,911), and modern scholarship `Print` (35,686), along with their associated attributes. Additional fields within the `Work` and `Person` types, such as `Genre` classifications or various interpersonal relationships, respectively, were also excluded for now.

The reduced dataset is modeled with the following simplified Python class structure (see `data_models.py` for the full implementation):

```python
class Entity:
    def __init__(self, entity_id: str):
        self.id: str = entity_id
        self.type: str
        self.name: str
        self.aka: str
        self.highest_year: Optional[int]
        self.lowest_year: Optional[int]

class Work(Entity):
    def __init__(self, entity_id: str):
        super().__init__(entity_id)
        self.type: str = "work"
        self.author_ids: List[str]
        self.base_text_ids: List[str]
        self.commentary_ids: List[str]
        self.discipline: Optional[str]
        self.author_highest_year: Optional[int]
        self.author_lowest_year: Optional[int]

class Author(Entity):
    def __init__(self, entity_id: str):
        super().__init__(entity_id)
        self.type: str = "author"
        self.social_identifiers: Optional[str]
```

```
        self.work_ids: List[str]
        self.disciplines: Optional[str]
```

This streamlined approach improves usability while preserving essential scholarly connections. Users can still explore finer details through linked *Pandit* pages as needed.

2.3 Synthetic Attributes

Some clarification is needed for how certain of these attributes are newly constructed. Since the dataset only specifies a work's base text (via the field `Commentary on (work ID)`), its commentaries must be inferred by reversing this relation, iterating through all works to associate each with its corresponding commentaries. The resulting structure enables traversal of authorial and commentarial relationships in any direction, allowing for the construction of subgraphs that expand from arbitrary nodes.

`Discipline` is initially associated only with works—and even then, only sparsely. Based on these, a synthetic `disciplines` list is generated also for each author, as applicable, including counts of associated works per discipline (e.g., Maṇḍana Miśra's `disciplines` value is "Mīmāṃsā (3), Advaita Vedānta (1), Vyākaraṇa (1)").

Dates are tracked for both works and authors. When a work lacks its own date information, it may inherit the associated author's range, labeled accordingly (e.g., `author_highest_year`). In this way, the sparseness of date information for works can be partly overcome.

2.4 Obtaining and Processing Data

To support periodic updates to the underlying *Pandit* database, *Pāṇḍitya* is equipped with a simple Extract-Transform-Load (ETL) pipeline that processes *Pandit* data. However, the data must first be obtained from the *Pandit* database, which is less straightforward.

2.4.1 Exporting from Pandit Database

On the *Pandit* website's "Advanced Search" page, a "Download CSV" button allows users to download search results. When filtering by entity type via the left sidebar, this feature makes it appear possible to export arbitrarily large sets of entities, such as all Persons or all Works. However, such large requests do not currently complete on the project's production server. With the support of the *Pandit* team, work on *Pāṇḍitya* has so far proceeded on the basis of a full export initiated by a team member with access to the internal development server and manually transferred to me via Google Drive.

That said, automating updates via the production server still appears feasible through the following approach:

1. Use the lightweight JSON API by appending the `?_format=json` parameter to any node URL (e.g., `https://panditproject.org/node/89000?_format=json`).
2. Periodically query all relevant entities (Works and Persons) known from prior data, check the `changed` attribute's timestamps for recent updates, and update records accordingly.
3. Leverage the sequential nature of numerical identifiers to detect newly published entities.

Such automation has not yet been implemented at the time of writing.

2.4.2 Extract

Out of the original 67,529 rows and 163 columns, only the following were retained:

- 12,700 rows with `Content type` "Work", and 3,797 "Person" rows limited to individuals listed as authors of at least one work.
- 14 columns with a primary focus on authorial and commentarial relationships:
  - `ID`, `Title`, `Author (person IDs)`, `Authors (person)`, `Commentary on (work ID)`, `Commentary on (work)`, `Aka`, `Social identifiers`, `Discipline`, `Highest Year`, and `Lowest Year`, which are kept as-is.

- `Attributed author (person ID)` and `Attributed author (person)`, which are currently merged into `Author (person IDs)` and `Authors (person)`, respectively.
- `Content type`, which is ultimately dropped.

This filtering is automated with the script `utils/extract.py`, producing the output `data/2024-12-23-works-raw.csv`.

2.4.3 Manual Cleaning

Before transformation, minor manual cleaning was required to remove a few spurious entities. Details are documented in `data/manual_cleaning.md`, and the cleaned dataset is saved as `data/2024-12-23-works-cleaned.csv`.

2.4.4 Transform

The next step, implemented in `utils/transform.py`, converts the cleaned dataset into structured `Work` and `Author` objects, organizing them into an in-memory lookup table that reflects the entity model described in § 2.2.

2.4.5 Load

Finally, `utils/transform.py` saves the processed data in a human-readable and retrieval-efficient JSON format (`data/2024-12-23-entities.json`). This dataset can then be loaded by other code components using `utils/load.py`.

2.5 Component and Other Network Analysis

The module `utils/analyze.py` analyzes and categorizes network components, i.e., communities of connected nodes, within the dataset, offering insights into how works and authors interconnect. Table 1 summarizes the distribution of components, and full lists of component members are available at `https://panditya.info/notes/data`.

Component Type	Number of Components	Total Nodes
Isolated Nodes (single works only)	3,005	3,005
Small Communities (2–4 entities)	1,608	3,605
Medium Communities (5–9 entities)	90	565
Large Communities (10–25 entities)	24	344
Second-Largest Community	1	73
Central Community	1	8,905

Table 1: Summary of network component analysis.

A detailed discussion of these and other connection patterns is beyond the scope of this paper. However, the following key observations are noteworthy:

- The presence of a large central community based solely on accepted commentarial relationships underscores the fundamental role of commentaries in Sanskrit intellectual history.
- If additional intertextual phenomena, such as parallel passages, were included, many more so-called "isolated" works—especially those written by authors with only one or two extant texts—would be found to engage in broader intellectual discourse.
- This network structure may evolve as further updates to *Pandit* incorporate new philological findings.

Two additional practical takeaways from this analysis are:

- Isolated or nearly isolated items are relatively common.
- Conversely, participation in the central community is also widespread, making the visualization of subgraphs with 6–7+ hops impractical for many inputs.

Beyond this component analysis, `utils/analyze.py` also explores preliminary metrics such as centrality, influential nodes, and temporal patterns. These remain proofs of concept for now, but future refinements could significantly enhance the tool's analytical depth.

3 Web Application

Pāṇḍitya is built on a modular and scalable web architecture:

1. A *Flask* backend using `flask_restx` with Swagger-based API docs.
2. A REST API that serves entity metadata and builds graphs on demand.
3. A *D3.js* front end that dynamically renders and updates network visualizations in real time.
4. Version control for both code (*GitHub*) and containerized deployments (*Docker Hub*).
5. Deployment on a cloud server (*Digital Ocean*).

3.1 Flask App

The *Flask* app serves as the core backend, handling data requests and visualization processing:

- Loads entity data from the ETL pipeline.
- Exposes API endpoints (see § 3.2) used by the front end.
- Serves the main route (`/`) with HTML form and graph controls.
- Provides the `/view` route for external linking to specific graphs.
- Serves informational pages such as `/about`, `/notes/technical`, and so on.

3.2 Backend REST API

Key endpoints include:

- `GET /api/entities/<type>` – Retrieves `works`, `authors`, or `all` entities.
- `GET /api/entities/labels` – Maps ID numbers to human-readable labels.
- `POST /api/graph/subgraph` – Generates subgraphs from selected entities and hop counts using simple breadth-first traversal, with optional exclusion of specified nodes.

For users who wish to interact directly with the API, the following is also available:

- `/api/docs` – Interactive Swagger docs with example queries.

3.3 Use of D3.js

D3.js's `forceSimulation` models nodes as solid, mutually repelling objects connected by flexible links. Four adjustable forces determine the dynamic layout and can be tuned in-browser:

- `forceCollide` – Controls how nodes resist overlapping with local neighbors. Higher values increase virtual node size.
- `forceManyBody` (`.strength`) – Controls global node repulsion. Higher values increase repulsion strength.
- `forceLink` (`.distance`) – Controls the spacing of connected nodes. Higher values increase link distance.
- `forceCenter` (`.strength`) – Controls the tendency of nodes to return to the graph's center. Higher values increase centralization.

D3.js also provides built-in support for dragging and zooming, simplifying interactions. Additionally, it facilitates the implementation of *Pāṇḍitya*'s node context menus (see §3.7).

3.4 GitHub Repository and Local Deployment

For local development, users can follow these steps:

- Clone the repository from GitHub: `https://github.com/tylergneill/panditya`.
- Set up a virtual environment using Python 3.11+.
- Install dependencies from the provided requirements files: `requirements.txt`, `requirements_etl.in`, and `requirements_offline.in`.
- Start the server using the included Makefile: `make run`.

3.5 Versioned Deployment on Digital Ocean

Deployment is managed using Docker and Digital Ocean.

- New builds are created for both development and production releases (see GitHub PR titles and descriptions for versioning details); each production release is also marked with a corresponding Git tag.
- Images are pushed to Docker Hub and deployed as containers on the Digital Ocean server.
- *Nginx* and *Gunicorn* handle traffic management and load balancing.
- Daily backups of the Digital Ocean "Droplet" ensure system stability and data integrity.

The production deployment is accessible at `https://panditya.info`, running release 2.4.9 at the time of writing (Neill, 2025).

3.6 User Input Flow

Using the web form, users select works and/or authors via auto-complete drop-downs, which use IAST and are sorted in Sanskrit alphabetical order. These selections are grounded in *Pandit* data and disambiguated primarily by *Pandit* ID numbers.

Users should be aware of transliteration ambiguities. *Pāṇḍitya* currently relies on the mostly precomposed IAST inherited from the *Pandit* database. As a result, search results may vary when entering forms such as "Śaṃkara" (m with underdot), "Śaṁkara" (m with overdot), "Śaṅkara" (velar nasal), "Śaṃkara" (decomposed diacritics, i.e., S ´ am ¸ kara), or simply "samkara" or "sankara". The `Select2` JavaScript module used for dropdowns generally handles these variants well, and *Pāṇḍitya* itself supplements search with auxiliary information such as dates and alternate names, producing more verbose, disambiguated entries like "Śaṃkara (85218) [710] [Śaṃkarācārya, Śaṅkara ācārya]". Together, these measures are robust against most orthographic variation, with the exception of decomposed diacritics. In the future, support for a hub transliteration scheme such as SLP1 may improve consistency and search reliability across edge cases. When in doubt, users can consult *Pandit*'s native search interface to help identify the unambiguous ID number.

Once entities are selected, users then specify a hop count to determine graph expansion. Optionally, users can also exclude specific entities from expansion to reduce clutter in highly connected subgraphs.

For programmatic use cases, the `/view` route supports direct queries using these same inputs as URL parameters, e.g., Bhagavadgitābhāṣya, suppressing expansion on Bhagavadgītā and Śaṃkara: `https://panditya.info/view?works=88637&hops=2&exclude_list=85218,42214`.

3.7 Node Context Menu

After a graph is displayed, right-clicking a node reveals the following options:

1. `More info` – Lists additional fields when available: alternate names (`aka`), `social identifiers` (for authors), `dates`, and `discipline` (for works) or `disciplines` (for authors).
2. `View on` – Links to *Pandit* entity pages and relevant online e-text repositories (see § 4).
3. `Recenter` – Sets the selected node as the new center and regenerates the graph, expanding outward by 1–3 hops.
4. `Exclusions` – Collapses the selected node, with future options for full removal and re-expansion.

3.8 Force Controls

Sliders beneath the graph allow users to adjust each of the four `D3.forceSimulation` forces (see §3.3). Additionally, a "Freeze" toggle temporarily disables all forces, enabling manual rearrangement, which can facilitate close inspection and/or screenshots.

3.9 Screenshots

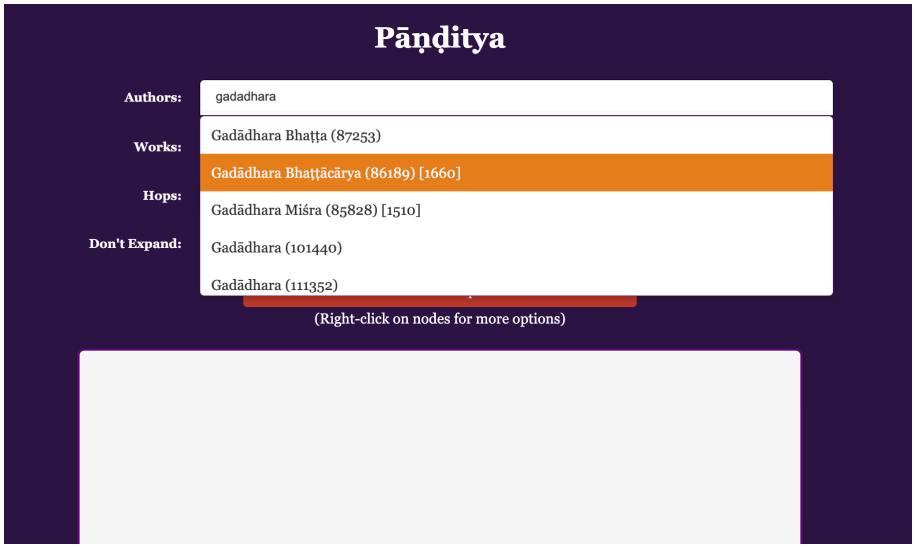

Figure 1: *Searching for an entity using auto-complete drop-down*

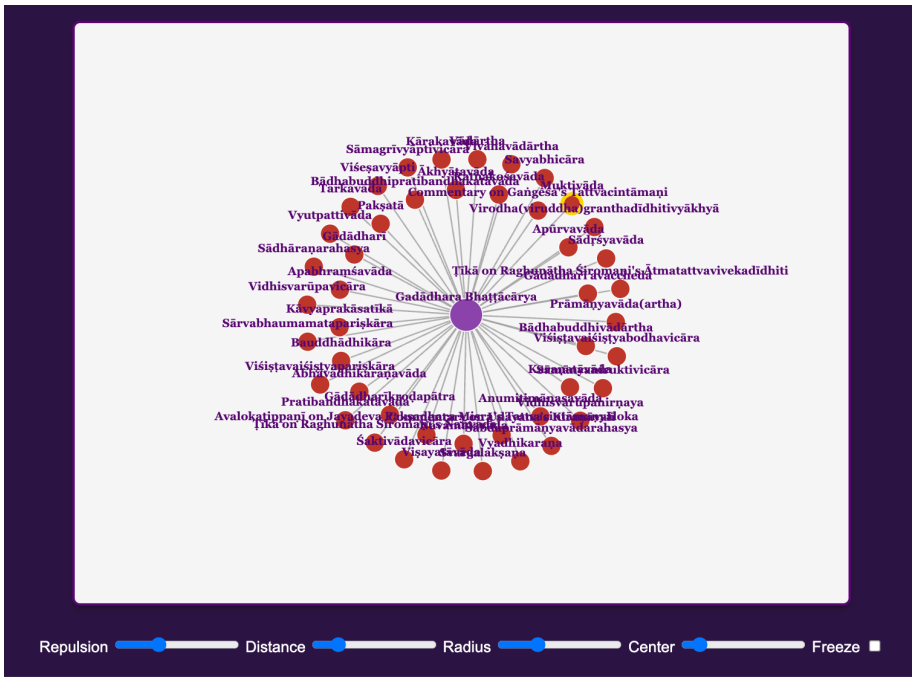

Figure 2: *Generating basic graph*

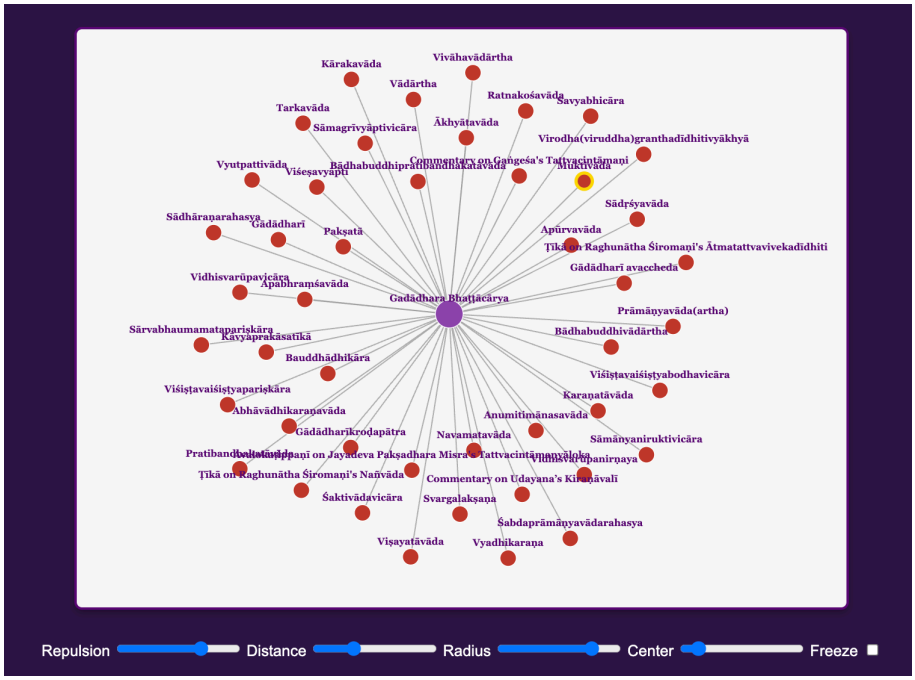

Figure 3: *Adjusting force controls*

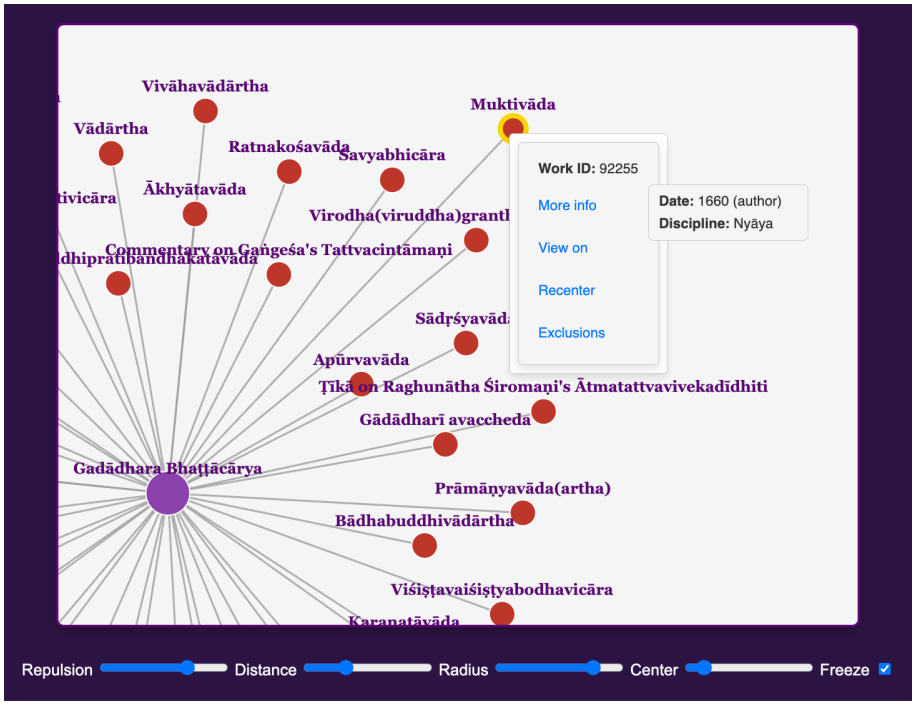

Figure 4: *Opening context menu with right-click ("More info")*

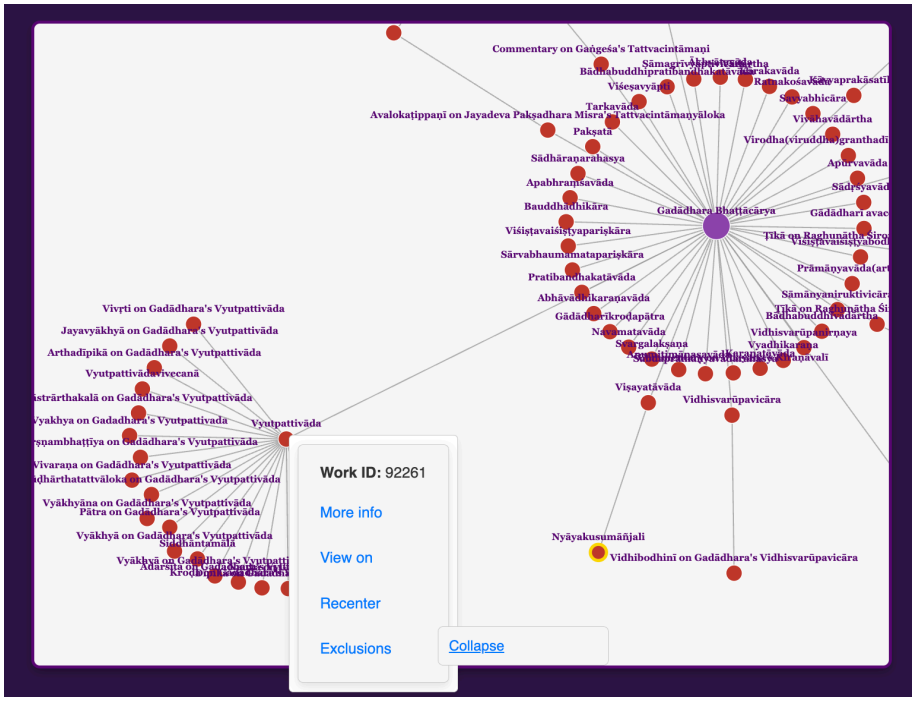

Figure 5: *Using context menu to collapse prolific node*

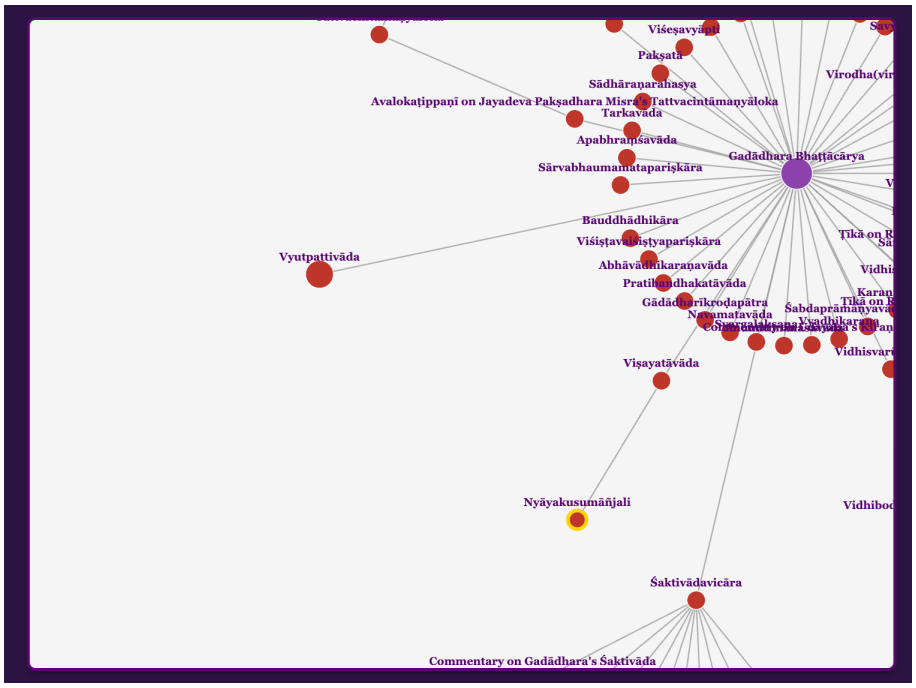

Figure 6: *New graph with collapsed node*

4 The Sanskrit E-Text Inventory (SETI)

In order to enable hyperlinks from *Pāṇḍitya* work node context menus to online Sanskrit e-texts, a separate but related effort, the *Sanskrit E-Text Inventory* (*SETI*), aggregates metadata from multiple repositories and aligns it with *Pandit* identifiers. Currently included repositories are *GRETIL*, *Digital Corpus of Sanskrit* (*DCS*), *SARIT*, *The Sanskrit Library* (and *TITUS*), sister projects *Vātāyana* and *Pramāṇa NLP*, and the *Muktabodha* collection of digitizations from the *Kashmir Series of Texts and Studies* (*KSTS*).

This complementary data layer surfaces throughout *Pāṇḍitya*: in the ETL pipeline's transform step, in the REST API (`/api/seti` routes), and in the D3 front end. For work nodes highlighted in gold, the context menu's "View on" option reveals available e-text collections, which are further differentiated into one to three levels of access, ranging from web reading platforms to raw GitHub data. Component collections and their contained e-texts can also be programmatically queried and compared via `/api/seti` GET routes: `by_collection`, `by_collection/overlap`, `by_collection/unique`, and `by_work`. In addition, the HTML route `/seti/by_collection/<collection>/visualize` behaves like `/view` (see § 3.6), returning a preloaded graph visualization in the browser—here, for entire collections at once. For example, `https://panditya.info/seti/by_collection/SARIT/visualize` shows the SARIT collection. Note, however, that only items present in both the source collection and the *Pandit* database can be visualized in *Pāṇḍitya*.

This integration of *Pāṇḍitya* and *SETI* constitutes a powerful navigation tool and highlights both the breadth of online Sanskrit resources and the structural role of shared identifiers. Further details, including a system design diagram, numerical coverage overview, and access to the public master spreadsheet, are available at `panditya.info/seti` and in the accompanying blog post.

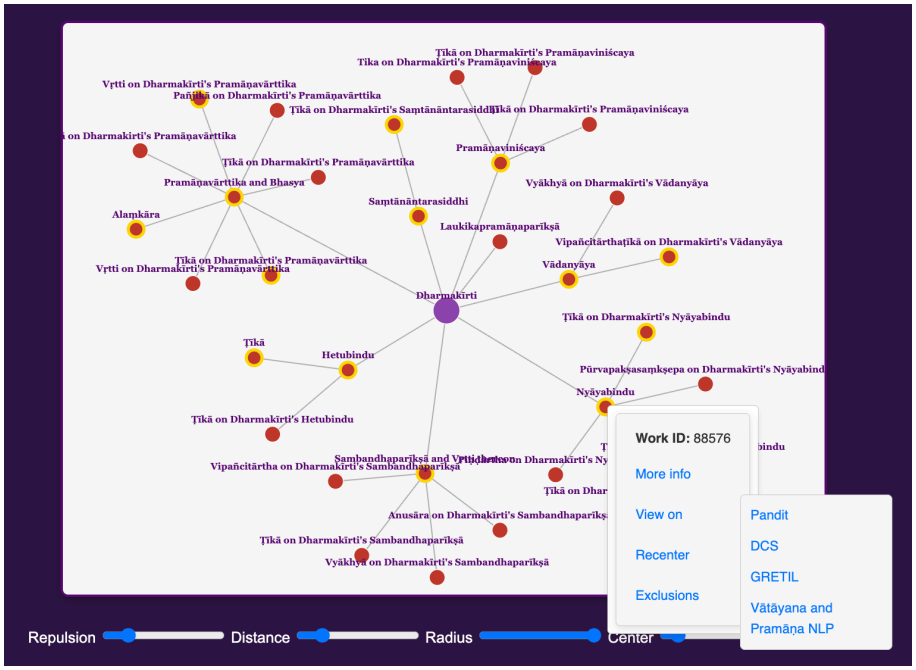

Figure 7: *Viewing Dharmakīrti e-texts via SETI and the "View on" context menu option*

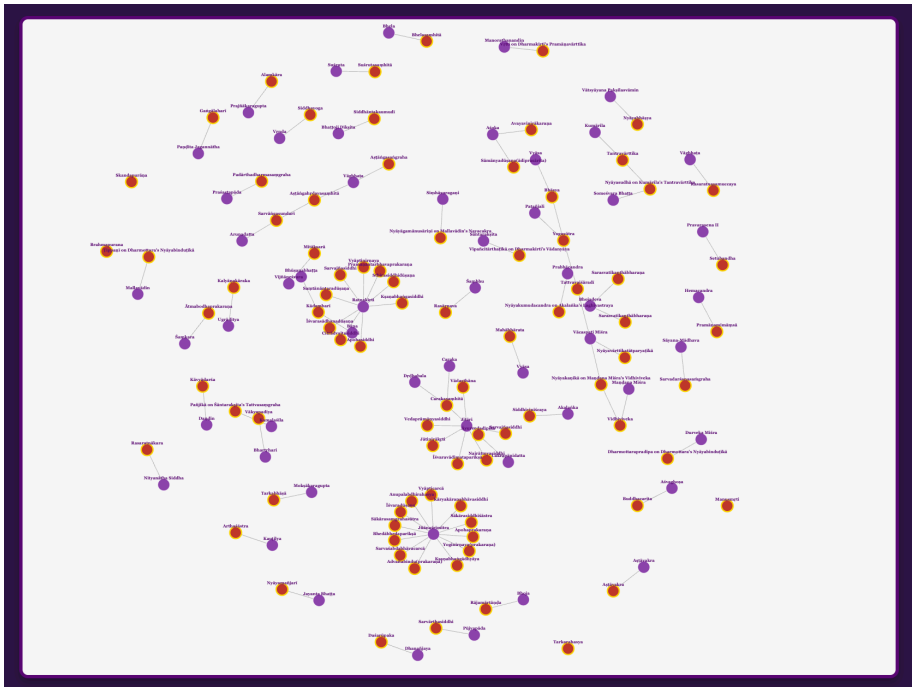

Figure 8: *Visualizing SARIT*

5 Conclusion and Future Directions

With its targeted use of interactive visualization, *Pāṇḍitya* bridges a tangible gap between structured data and human comprehension. It enhances access to the underlying *Pandit* dataset, empowering users to more effectively explore and understand Sanskrit intellectual relationships. Its three main use cases at present include:

1. A gamified reference tool for students and scholars, encouraging deeper engagement with Sanskrit works and authors.
2. A mechanism for improving *Pandit* data by highlighting inconsistencies and missing connections, while also fostering greater interest in the project.
3. An electronic catalog tool for finding and navigating to Sanskrit e-texts online.

Future expansions may include:

- Building additional features on top of *Pandit* data, such as quick search for related works by both discipline and date, or visualization of family trees.
- Feeding insights from *SETI* back into *Pandit* in the form of new entities and connections, thereby increasing visualization coverage within *Pāṇḍitya*.
- Illustrating intertextual network analysis, using weighted edges to represent textual interactions as computed by projects like *BuddhaNexus* (Nehrdich, 2020) and *Vātāyana* (Neill, 2022), with links to interactive intertextuality reports where available.

Beyond these enhancements, several known development priorities remain:

- Strengthening *Pandit*-*Pāṇḍitya* integration to streamline data refreshes.
- Improving navigation with more flexible input methods (e.g., transliteration options) and alternative modalities like alphabetical browsing.
- Implementing error-checking to prevent the creation of overly large subgraphs.
- Exposing v1 export functionality through the front-end interface.

With its rapid development cycle and adaptable design, *Pāṇḍitya* aims to serve as a vital tool for the Sanskrit scholarly community. It complements the *Pandit* project from which it derives, opening new paths for engaging with the vast intellectual heritage of the Sanskrit tradition.

Acknowledgments

Pāṇḍitya gratefully acknowledges *Pandit* as its source and follows it in adopting the Creative Commons BY-NC-SA 4.0 license, viewable at `https://creativecommons.org/licenses/by-nc-sa/4.0/deed.en`.

References

Yigal Bronner et al. 2024. Pandit Prosopographical Database of Indic Texts. Available at `https://www.panditproject.org/`. Accessed: 2025-04-30.

Dan Edelstein et al. 2017. Historical Research in a Digital Age: Reflections from the Mapping the Republic of Letters Project. *The American Historical Review*, 122(2):400–424, April. Published: 30 March 2017.

Guldi Jo and Chris Johnson-Roberson. 2012. Paper Machines. Available at `http://papermachines.org/`. Accessed: 2025-01-31.

Peter Kraker et al. 2025. Open Knowledge Maps. Available at `https://openknowledgemaps.org/`. Accessed: 2025-01-31.

Sebastian Nehrdich. 2020. A Method for the Calculation of Parallel Passages for Buddhist Chinese Sources Based on Million-scale Nearest Neighbor Search. *Journal of the Japanese Association for Digital Humanities*, 5(2):132–153.

Tyler G. Neill. 2021. Pandit Grapher. `https://doi.org/10.5281/zenodo.14768846`.

Tyler G. Neill. 2022. *Intertextual Readings of the Nyāyabhūṣaṇa on Buddhist Anti-Realism*. Dissertation, Universität Leipzig, December. Available at `https://nbn-resolving.org/urn:nbn:de:bsz:15-qucosa2-826296`.

Tyler G. Neill. 2025. Pāṇḍitya. `https://doi.org/10.5281/zenodo.15307376`.

Patrick Reynolds and Brett Tjaden. 2025. The Oracle of Bacon. Available at `https://oracleofbacon.org/`. Accessed: 2025-01-31.

David A. Smith et al. 2015. Computational Methods for Uncovering Reprinted Texts in Antebellum Newspapers. *American Literary History*, 27(3):417–445.

Alex Tarnavsky Eitan et al. 2025. Connected Papers. Available at `https://www.connectedpapers.com/`. Accessed: 2025-01-31.

N. J. Van Eck and L. Waltman. 2010. Software Survey: VOSviewer, a Computer Program for Bibliometric Mapping. *Scientometrics*, 84(2):523–538.

N. J. Van Eck and L. Waltman. 2014. Citnetexplorer: A New Software Tool for Analyzing and Visualizing Citation Networks. *Journal of Informetrics*, 8(4):802–823.

Chris Warren et al. 2016. Six Degrees of Francis Bacon: A Statistical Method for Reconstructing Large Historical Social Networks. *Digital Humanities Quarterly*, 10(3). Available at `http://www.sixdegreesoffrancisbacon.com/`. Accessed: 2025-01-31.