

Questionable analytical observations and what to do about them

Charles T. Gray & Hannah Fraser

October 11, 2025

Abstract

A critical challenge in data management is balancing complex data transformation with the need for analytic transparency for domain-knowledge validation. FAIR (findable, accessible, interoperable, reusable) data principles as *compliance* fall short when viewed as a checkbox exercise at publication. Living analyses evolve over time, and all analyses are iterative, requiring ongoing validation of data processing decisions. Contemporary data practices on the modern data stack provide robust tooling to validate data transformation, but in practice render data observability opaque to non-technical stakeholders. This creates a fundamental tension between engineering efficiency and analytic interpretability that undermines the core principles of FAIR data.

This document presents a novel tool-agnostic semantic layering methodology for analytic data pipeline design, demonstrated through provisioning of data comprising UN Sustainable Development Goal (SDG) priorities from surveys conducted in dam-affected communities in North India.

The central methodological contribution is a semantically-structured approach to data transformation that extends Data Build Tool’s engineering-focused patterns to prioritize interpretability and stakeholder communication. Unlike standard data engineering implementations that emphasize technical efficiency, this semantic layering methodology explicitly preserves analytic context, makes harmonization decisions transparent, and maintains conceptual clarity throughout the data pipeline.

The semantic layer architecture consists of four conceptually distinct stages: source base (raw data cleaning), source entities (region-specific preservation), semantic models (cross-source harmonization), and analytic models (analysis-ready datasets). This approach addresses the critical gap between technical data engineering practices and analysis methodology requirements, enabling automated observability while maintaining scholarly rigor. We argue FAIR is as relevant and urgent for enterprise data engineering as it is for research questions.

This semantic layering methodology offers a replicable framework for data management that bridges

the divide between engineering efficiency and analytic transparency, with applications across all domains requiring multi-source data integration and stakeholder communication.

Contents

1	Questionable analytical observations	4
1.1	Questionable research practices	4
1.2	What to do about questionable analytical observations	5
1.3	Case study: giveadam project	5
1.3.1	Minimal representation	6
1.4	Enterprise applications for data governance at scale	7
1.4.1	Modern data stack	7
2	About the giveadam data	8
3	Semantic data pipeline methodology	8
3.1	The semantic layering paradigm	9
3.2	Repository architecture supporting semantic design	10
3.3	Core innovation: Semantic transformation architecture	10
3.3.1	Why semantic layering transforms research data management	10
3.4	Automated observability for research transparency	11
4	FAIR principles implementation	12
5	Data lineage	12
6	Semantic layer implementation and observability	13
7	Use of NLP tools	13
8	Acknowledgements	13
A	FAIR principles implementation	13
A.1	Findable: Discovery and identification	13
A.2	Accessible: Retrieval and usability	14
A.2.1	SDG rankings dataset	14
A.2.2	Respondents dataset	15

56	A.2.3	SDG labels dataset	15
57	A.2.4	SDG7 analysis dataset	15
58	A.3	Interoperable: Integration and exchange	16
59	A.4	Reusable: Extension and adaptation	16
60	B	Data lineage	17
61	B.1	Source data and transformation overview	17
62	B.2	Pipeline architecture and processing	17
63	B.2.1	Data quality assurance	17
64	B.3	Visual data flow representation	18
65	B.4	Model catalog and validation	18
66	C	Semantic layer implementation and observability	18
67	C.1	Semantic layers as methodological framework	19
68	C.1.1	Methodological advantages of semantic layering	20
69	C.2	Interactive data exploration	20
70	C.3	Observability table generation	21
71	C.3.1	Automated metadata extraction	21
72	C.3.2	Model materializations	21
73	C.4	Data validation	22

74 List of Tables

75	1	Data Build Tool (DBT) models created by dbt_project, ordered by observability layer.	19
76	2	Analytic tests applied to dbt_project models in this layer.	22
77	3	Semantic tests applied to dbt_project models in this layer.	23
78	4	Source Entities tests applied to dbt_project models in this layer.	24
79	5	Base tests applied to dbt_project models in this layer.	24

80 List of Figures

81	1	Data architecture	6
82	2	Give a dam data	8
83	3	DBT documentation	12

84	4	UN SDG priorities by gender	27
85	5	Data lineage	28

1 Questionable analytical observations

Underestimating the complexity of data transformation is a well-recognised driver of garbage in, garbage out analytics in scientific research, industry, and public institutions. For example, gene name errors caused by Excel formatting issues are widespread—detected in 700 scientific publications (approximately 20 per cent of 3,500 publications analysed) [27]. More recently, the UK government’s COVID-19 tracking lost about 16,000 cases due to Excel row limits [17]. These examples illustrate how technical limitations and data handling errors can lead to significant analytical inaccuracies.

1.1 Questionable research practices

Broadly, the purpose of research and analytics is to discover something useful about the world; to derive meaningful inference from data. However, in reality there are so many kinks in the pipeline between the research question and the answer, that the usability and usefulness of the answer that is produced by that pipeline is uncertain. Though, research that passes peer review is often seen as authoritative and accurate. The ‘replication crisis’ movement challenged this, finding that only 1/3 – 1/2 of research, when repeated, finds the same results. This low replication rate has been attributed in part to the use of **questionable research practices** which make research findings appear more impactful and reliable than warranted by the methods used (Gould et al. in prep). Examples of questionable research practices include HARKing (Hypothesising after results are known), p-hacking (which includes everything from choosing to drop outliers after examining the impact on the results, to rounding p-values down to reach a threshold e.g. $p < 0.05$), and cherry picking (conducting multiple analyses but only presenting the ones with the significant/most publishable results) [13]. These practices are prevalent across the fields of science [26] are indicative of a huge problem in the research pipeline. However, the scope of QRP research has been limited to the analysis and write-up parts of the research pipeline. We argue that the focus on problematic research practices needs to include the earlier parts of the pipeline as well. In this paper we describe **questionable analytical observations** which occur between the point of data collection and the point of analysis, with flow on effects through the rest of the research pipeline. Questionable analytical observations arise when data transformation processes are not transparently validatable or interpretable to domain experts, leading to invisible garbage in, garbage out analytics. This is particularly problematic in multi-source data integration contexts, where harmonization decisions can significantly impact analytical outcomes.

Sharing code, materials and data is one proposed counter to questionable research practices. This is in line with FAIR principles (see below), facilitates true critical analysis of the analysis methods, and should allow someone to test the method by using the code and data to reproduce the original results. Currently, we are falling well short when it comes to code and data sharing estimates suggesting that across fields 0.5-27 per cent of articles share code and 2-79 per cent share data [11, 4, 22, 14, 10]. Further, even if this code and data is provided, there is only around a 30 per cent chance that the same answers can be reproduced by a new analyst[11, 14]. Clearly, there is a long way to go before an adequate level of transparency is achieved for the analysis component of the research pathway but at least the spotlight is on this issue. The parts of the pipeline between receiving rough-and-ready raw data and beginning the analysis have received almost no attention. In cases where the whole research pipeline is conducted by a single person, and that person is very diligent about documenting their code, methods and data, sometimes these steps are included. However, in other research contexts the research pipeline is carved up between researchers, and data engineers and curators. These circumstances require a premeditated structure to avoid questionable analytic observations that undermine the usefulness and reliability of their results.

1.2 What to do about questionable analytical observations

To address questionable analytical observations, we propose a methodological bridge (Section 3) between modern data engineering practices and analytical validation. This involves:

- **Semantic layering of data transformations** as shown in Figure 1 to preserve analytic contexts and make harmonization decisions explicit (Section 3.3);
- **Automated observability frameworks** that generate transparent documentation and validation reports from pipeline artifacts (Section C.3);
- **Stakeholder-centric communication** using conceptual rather than technical terminology to facilitate domain expert validation (Figures 5 and 3);
- **Iterative validation processes** that allow ongoing scrutiny of data processing decisions as analyses evolve over time.

1.3 Case study: giveadam project

This manuscript and accompanying GitHub repository `giveadam` present a case study implementing the proposed methodology on open science data (Section 2). The project analyzes UN Sustainable Development

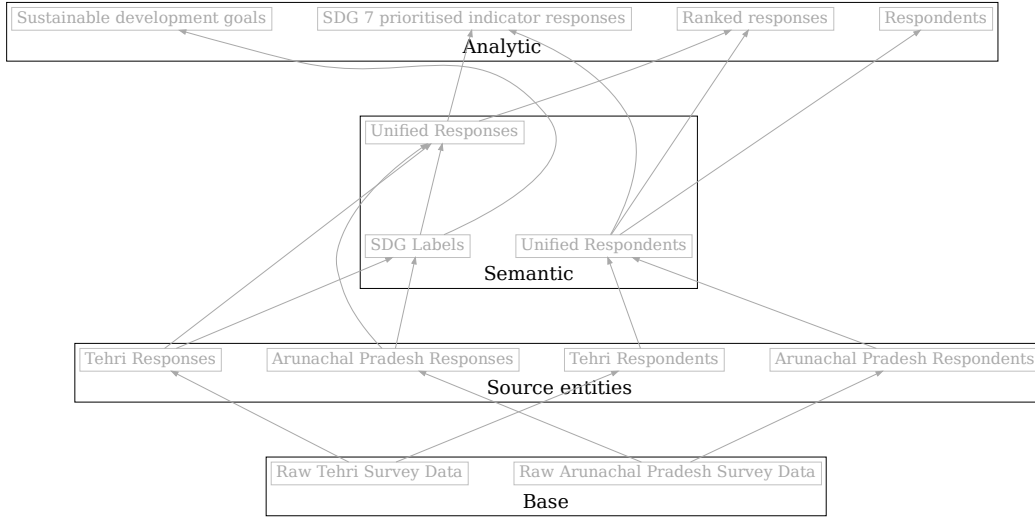


Figure 1: Data architecture: semantic layering from base to analytic models of **giveadam** data (described in Section 1.3), ordered by observability. This layering explicitly delineates between loading the data (base), extracting source-specific entities (source entities), harmonizing across sources (semantic), and preparing analysis-ready datasets (analytic). Each layer serves a distinct purpose in the data transformation pipeline, enhancing transparency and interpretability for domain experts. This layering provides a domain and context agnostic minimal test-driven development architecture checking unique keys for missingness and duplicates.

Goal (SDG) priorities from dam-affected communities in North India, using survey data collected by Garima Gupta from residents of Tehri (where a dam was constructed 20 years ago) and Arunachal Pradesh (where a dam is under development).

This case study demonstrates the methodology on small, analytic data where validation is tractable, but the principles are applicable to large-scale data contexts. To illustrate scalability, the methodology uses open source modern data stack tools standard in enterprise environments, such as Data Build Tool (dbt). The following sections detail the semantic layering methodology, repository architecture, and adherence to FAIR principles.

1.3.1 Minimal representation

This case study was chosen as it provides a minimal representation of two key challenges in data management:

- **Multi-source data integration:** The project integrates two regional survey datasets with differing formats and labeling conventions, requiring explicit harmonization decisions that impact analytical outcomes.

- **Stakeholder alignment:** Interoperation of three people with different roles and expertise: Garima Gupta (data collection and domain expertise); Charles T. Gray (data engineering and pipeline design); and Hannah Fraser (code review and validation). Crucially, it is Garima Gupta who possesses the domain knowledge to validate the data processing decisions, but she is not a data engineer. This necessitates a methodology that facilitates transparent communication and validation across technical and domain expertise boundaries. Similarly, Hannah Fraser is not a data engineer, but she is a domain expert and can validate the data processing decisions are interpretable.

1.4 Enterprise applications for data governance at scale

The case study described above was chosen so that the methodology is interpretable on data that can be validated by eye. However, at scale, in large data platforms simply opening datasets is not an option. Platforms at scale for enterprise data management and public good typically involve:

- hundreds, if not **thousands, of sources** many of which are provisioned in legacy formats (e.g. Excel, CSV, XML, JSON, SQL dumps);
- hundreds, if not **thousands, of databases** requiring aggregation with integrity;
- **billions of rows of living data** per table in each database equiring ongoing validation specific to **snapshot** or **incremental** data processing.

When working across legacy sources in the volatility of the technology landscape, data engineers typically have little more to work with than inscrutable table names and column headers. Only after many complex steps requiring advanced software engineering codebases is the data available in a way that is accessible to stakeholders with enough domain knowledge to validate that what has been extracted by, say, ETL (extract, transform, load) pipelines is the desired output.

1.4.1 Modern data stack

Advances in the modern data stack (a term to loosely define the 'collection of tools and cloud data technologies used to collect, process, store, and analyze data' [18]), such as dbt [5] and dagster [?] have unprecedented potential to address these challenges. However, there is a prevailing tendency for these tools to be oriented to interpretability by engineers, rendering the complexity of data processing — crucially the assumptions made — opaque to domain experts.

This creates a fundamental tension between engineering efficiency and analytic interpretability that undermines the core principles of FAIR data. The semantic layering methodology presented in Section 3

addresses this tension by reorienting data transformation pipelines around research concepts rather than technical convenience, enabling both computational efficiency and methodological rigor.

2 About the giveadam data

The data in this project aggregates two region-based surveys of UN SDG priorities from participants in Tehri and Arunachal Pradesh. Respondents ranked their top three UN SDG priorities, providing insights into development preferences in dam-affected communities. The data is stored in the `data/` directory of the giveadam repository (see Section 3.2 for complete repository structure). This project aggregates survey responses across regions using the semantic layering methodology (detailed in Section 3) to investigate the following research question:

What are the differences in SDG priorities between residents of Tehri, where a dam was constructed 20 years ago for hydroelectric power, and Arunachal Pradesh, where a dam is currently being developed?

Figure 2 shows a treemap of the top 3 UN SDG priorities from survey respondents in Tehri (dam constructed 20 years ago) and Arunachal Pradesh (dam under development), North India.

Figure 2: Treemap of top 3 UN SDG priorities from survey respondents in Tehri (dam constructed 20 years ago) and Arunachal Pradesh (dam under development), North India. Each rectangle represents a specific UN SDG, with its size proportional to the number of respondents who ranked it among their top three priorities. The treemap visually highlights the most and least prioritized SDGs in each region, providing insights into regional differences in development priorities.

By structuring the data in tidy format (one row per ranking by one respondent), the dataset enables flexible analysis across multiple dimensions of respondent characteristics and preferences.

3 Semantic data pipeline methodology

The core methodological innovation of the giveadam project is a semantic layering approach that reconceptualizes data transformation pipelines for research contexts. This methodology addresses a fundamental limitation in current data engineering practices: the tension between technical efficiency and research interpretability within the modern data stack paradigm [18]. Pipelines and the scale of 'big data' are becoming recognised as an integral component of reproducible analyses, however FAIRness is still framed from an extrinsic rather than intrinsic perspective [15]. Furthermore, the structure of code and data is becoming recognised as a critical factor in ensuring the long-term usability and reproducibility of research outputs [23].

The focus in this manuscript is on embedding FAIR principles intrinsically within the data transformation process itself in recognition that right across the contexts of enterprise, large-scale public good, as well as extensible research requires transparency and accessibility of data transformation over time — measured not in months, but *decades*. If we are to trust data we must embrace it evolves over time with interoperators changing and data sources being added and removed. Toward an any analytic goal, we are now in a landscape of not *data* but *platforms that provision data* a diversity of people whose needs evolve, maintained by engineers who work in a fragile ecosystem where layoffs are commonplace.

Data engineering communities such as the vibrant open-source community developing dbt [5] have made significant strides in democratizing data transformation through accessible tooling and best practices [6]. However, these tools are primarily oriented toward engineering efficiency, often at the expense of interpretability for non-technical stakeholders. This creates a fundamental tension between engineering efficiency and analytic interpretability that undermines the core principles of FAIR data.

Currently academic solutions to FAIR and enterprise engineering are evolving in disjunct worlds. We argue the solution lies in bridging the two worlds by reimagining robust techniques developed by engineers as FAIR to stakeholders. Duplicates, missingness, and scale are challenges in both research and enterprise contexts; there is no reason for advances in the modern data stack and FAIR research to be siloed.

3.1 The semantic layering paradigm

Traditional data engineering prioritizes performance optimization and code maintainability, often at the expense of conceptual clarity for non-technical stakeholders. The semantic layering methodology inverts this priority structure, organizing transformations around research concepts rather than technical convenience.

Furthermore, by prioritising a minimal test architecture that checks for uniqueness, duplicates, and missingness by semantic entities (responses and respondents in the *giveadam* case study discussed in Section 1.3) interpretable by domain experts, the semantic layering methodology ensures that data quality is validated in a manner aligned with analytic objectives as opposed to opaque technical metrics. Extracting these tests as observability provides a minimal viable data product that is extensible to attributes associated with the semantic entities as the analysis inevitably evolves through collaboration between data engineers and domain experts.

This approach ensures that:

1. **Research context is preserved** throughout the transformation pipeline.
2. **Harmonization decisions are explicit** and auditable by domain experts.
3. **Stakeholder communication** uses conceptual rather than technical terminology.

239 4. **Methodological transparency** is maintained without sacrificing automation.

240 3.2 Repository architecture supporting semantic design

241 The giveadam repository is organized into distinct functional areas:

- 242 • **raw_data/** — Original survey data files as provided by Garima Gupta
- 243 • **data/** — Published, analysis-ready datasets (pipeline outputs)
- 244 • **dbt_project/** — Data transformation pipeline with semantic layering
- 245 • **observability/** — Automated methodology documentation and quality reports
- 246 • **scripts/** — Data preparation and analysis scripts (R, Python)
- 247 • **vis-scripts/** — Visualization generation scripts

248 3.3 Core innovation: Semantic transformation architecture

249 The central methodological contribution lies in reimagining data transformation layers as conceptual re-
250 search stages rather than technical processing steps. Standard Data Build Tool implementations use staging
251 → intermediate → marts layers optimized for engineering workflows [6]. Layering data transformation is
252 commendable for maintainability, however these terms are inscrutable to domain experts and obscure the
253 analytic logic underpinning data processing.

254 The semantic layering methodology replaces these with four conceptually distinct stages (Figure 1) that
255 separate data transformations into layers interpretable to domain experts:

- 256 • **source_base/** — Raw data integrity and initial cleaning
- 257 • **source_entities/** — Context-preserving regional data entities
- 258 • **semantic/** — Explicit cross-source harmonization with documented assumptions
- 259 • **analytic/** — Research-question-specific datasets ready for analysis

260 3.3.1 Why semantic layering transforms research data management

261 Each semantic layer addresses specific research methodology requirements:

- 262 1. **Context preservation** — Source entities maintain regional survey characteristics, preventing prema-
263 ture harmonization that obscures methodological differences

- 264 **2. Transparent harmonization** — Semantic models explicitly document how different data sources are
265 reconciled (e.g., SDG labeling differences between regions)
- 266 **3. Assumption visibility** — Every transformation decision is captured and testable, enabling method-
267 ological scrutiny
- 268 **4. Stakeholder accessibility** — Layer terminology reflects research concepts (semantic, analytic) rather
269 than engineering jargon (staging, marts)
- 270 **5. Iterative extensibility** — New research questions can be addressed by extending the analytic layer
271 without disrupting upstream logic

272 This approach resolves the fundamental tension between automated data processing and research trans-
273 parency, enabling both computational efficiency and methodological rigor.

274 **3.4 Automated observability for research transparency**

275 The observability framework generates this methodology documentation automatically from pipeline arti-
276 facts, ensuring that:

- 277 • Documentation remains synchronized with actual transformations
- 278 • All data quality assumptions are explicitly validated and reported
- 279 • Research methodology is fully reproducible from code
- 280 • Stakeholders can verify data processing decisions without technical expertise

281 The complete automated documentation process includes both methodology generation and data pub-
282 lication. The validation tables in this document are generated through the automated process detailed in
283 Section C.3. Additionally, `scripts/publish_data.R` extracts all final analytic models from the dbt pipeline
284 and exports them as CSV files to the `data/` directory for publication and external use.

285 The validation tables presented in this document are high-level summaries. Engineers or researchers
286 requiring deeper pipeline investigation can use `dbt docs generate` and `dbt docs serve` for an interactive
287 exploration (e.g., Figure 3) of the complete data lineage (Figure 5), including detailed model specifications,
288 column-level lineage, and test results.

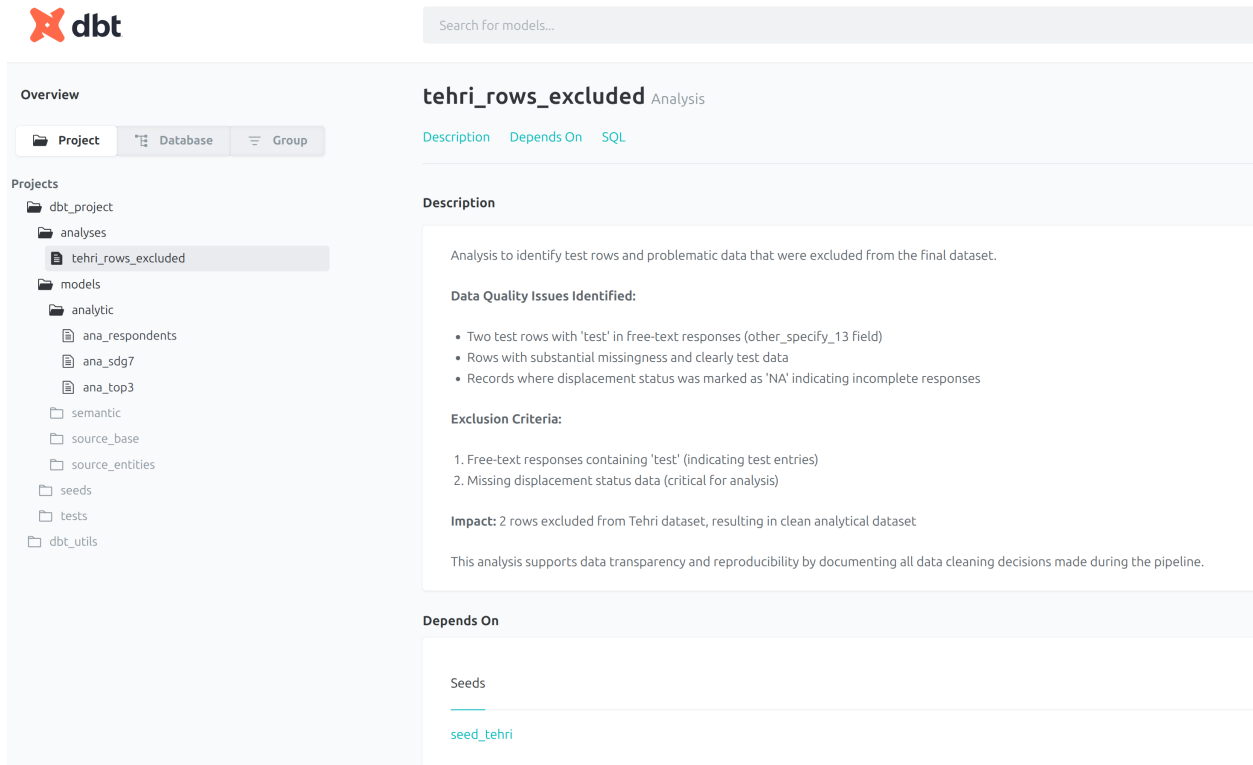


Figure 3: Screenshot from interactive dbt documentation showing the SQL logic excluding two test rows from the Tehri dataset. This exclusion is fully documented in `dbt_project/analyses/tehri_rows_excluded.sql` and reflects the methodological transparency principles outlined in Section 3. DBT documentation provides a user-friendly interface for engineers and stakeholders to verify data processing decisions without requiring technical expertise.

4 FAIR principles implementation

The giveadam project implements all four FAIR principles through its semantic layering methodology and automated observability framework. The semantic approach enhances traditional FAIR compliance by making research processes themselves findable, accessible, interoperable, and reusable. Complete technical details of the FAIR implementation, including dataset specifications, access protocols, and reusability frameworks, are provided in Appendix A.

5 Data lineage

The giveadam project pipeline transforms raw survey data through a systematic semantic layering approach, ensuring full traceability and methodological transparency. The complete data lineage, including technical pipeline details, model catalogs, validation frameworks, and directed acyclic graph visualizations, is documented in Appendix B.

6 Semantic layer implementation and observability

The observability framework operationalizes the semantic layering methodology through automated monitoring and validation at each conceptual stage. This implementation demonstrates how semantic design principles translate into technical infrastructure while maintaining research transparency. Complete implementation details, including technical infrastructure specifications, automated documentation processes, and validation frameworks, are provided in Appendix C.

7 Use of NLP tools

The giveadam project architecture, semantic data pipeline design, and all data transformations were conceived and implemented by Charles T. Gray. GitHub Copilot [7] was used for documentation editing and selected development operations tasks (supporting JSON manifest parsing scripts), but did not contribute to the fundamental design decisions, data modeling approach, or core analytical implementations. All code development, methodological innovations, and research insights represent the original work of the author.

8 Acknowledgements

The authors gratefully thanks Garima Gupta for collecting and sharing the original survey data from Tehri and Arunachal Pradesh, North India. The authors also acknowledge the open-source software community for developing the tools that made this project possible, including dbt, DuckDB, R, Python, and their associated libraries.

A FAIR principles implementation

The giveadam project implements all four FAIR principles [25] through its semantic layering methodology and automated observability framework detailed in Section 3. This section demonstrates how the semantic approach enhances traditional FAIR compliance by making research processes themselves findable, accessible, interoperable, and reusable.

A.1 Findable: Discovery and identification

Rich metadata and identifiers: The datasets are published with comprehensive metadata in `data/README.md`, including detailed column descriptions, data provenance, and research context. Each dataset has unique

identifiers (`respondents.csv`, `SDG_rankings.csv`, `SDG_labels.csv`, `SDG7_analysis.csv`) and is version-controlled in the GitHub repository `giveadam` at <https://github.com/softcloud/giveadam>.

Semantic documentation: Unlike standard data repositories, the semantic layering approach provides findable documentation at multiple conceptual levels. Researchers can locate relevant transformations by research concept (semantic models in Table 1) rather than technical implementation details. The four-layer architecture shown in Figure 5 enables discovery through conceptual navigation.

Automated cataloging: The observability framework generates searchable metadata from pipeline artifacts (Table 1), ensuring that all data transformations and quality checks are discoverable through the automated documentation system detailed in Section C.3.

A.2 Accessible: Retrieval and usability

Open access and standard protocols: Data are published in universally accessible CSV format [21] without authentication barriers. The complete repository is openly available via HTTPS and Git protocols [1], supporting both web browser access and programmatic retrieval through GitHub [8].

Human and machine readable: Column names follow interpretable conventions prioritizing domain understanding over technical convenience. The semantic layer architecture (Figure 5) ensures that data structure reflects research logic rather than processing efficiency.

Multiple access modalities: Researchers can access data through multiple pathways: direct CSV download, Git repository cloning [1], or programmatic URL access in R [20] or Python [19]. The interactive dbt documentation (`dbt docs serve`) [5] provides a web-based interface for exploring complete data lineage.

A.2.1 SDG rankings dataset

This dataset contains respondent rankings enriched with demographic metadata:

- **id_respondent:** Unique identifier for each respondent
- **rank:** Priority ranking (1=highest, 2=medium, 3=lowest priority)
- **sdg_number:** UN SDG number (1-17)
- **sdg_label:** Full name of the Sustainable Development Goal
- **age:** Age of respondent in years
- **gender:** Self-reported gender
- **displacement_status:** Dam impact classification

- **region:** Survey location (tehri, arunachal)

By enriching the responses with respondent metadata, we can analyse responses in the context of respondent demographics and characteristics. For example, in Figure 4, we can see the distribution of top 3 UN SDG priorities by gender.

A.2.2 Respondents dataset

- **id_respondent:** Unique identifier for each respondent
- **age:** Age of respondent in years
- **gender:** Self-reported gender (Male, Female, Prefer not to say)
- **displacement_status:** Impact classification related to dam construction
- **region:** Survey location (tehri, arunachal)

A.2.3 SDG labels dataset

This reference dataset provides standardized UN Sustainable Development Goal identifiers and labels:

- **sdg_id:** Standardized identifier (e.g., "SDG_1", "SDG_2")
- **sdg_number:** UN SDG number (1-17)
- **sdg_label:** Full name of the Sustainable Development Goal

A.2.4 SDG7 analysis dataset

This analysis-ready dataset focuses on energy priorities (SDG 7: Affordable and Clean Energy):

- **id_respondent:** Unique identifier linking to respondents table
- **region:** Survey location (tehri, arunachal)
- **age:** Age of respondent in years
- **gender:** Self-reported gender
- **displacement_status:** Dam impact classification
- **displacement_group:** Grouped displacement categories for analysis
- **sdg_7_chosen:** Binary indicator (1 = SDG 7 in top 3, 0 = not in top 3)

This dataset enables focused analysis of energy priorities across demographic groups and displacement categories, supporting research into how dam construction impacts community energy preferences.

A.3 Interoperable: Integration and exchange

Standard data formats and vocabularies: Data are published in CSV format [21] using UTF-8 encoding with standardized missing value representation (NA). Column naming follows consistent conventions across datasets, enabling seamless joining and integration across the semantic layers shown in Figure 5.

Semantic harmonization documentation: The semantic layer explicitly addresses interoperability challenges by documenting how disparate data sources are harmonized. For example, SDG labeling differences between regional surveys are reconciled with full documentation of mapping decisions in the semantic models (Table 1), enabling other researchers to understand and adapt the harmonization logic.

Modular pipeline architecture: The dbt project structure (Section 3.2) separates concerns across semantic layers, enabling selective reuse of transformation logic. Researchers can adopt the source entity patterns for context preservation while modifying semantic harmonization for different research domains. Each semantic layer (Table 1) implements specific interoperability functions that can be independently understood and modified.

Cross-tool compatibility: Tidy data principles ensure compatibility across analytical software. The semantic layer structure provides conceptual interoperability—researchers can understand and adapt the methodology regardless of their technical implementation preferences.

A.4 Reusable: Extension and adaptation

Comprehensive provenance and documentation: Complete methodology documentation enables confident reuse across research contexts. The automated observability system (Section C.3) ensures that documentation evolves with implementation, preventing methodology drift that undermines reusability.

Extensible semantic framework: The semantic layering methodology provides a reusable framework beyond this specific dataset. The four-layer architecture (source base → source entities → semantic → analytic) demonstrated in Figure 5 can be adapted for any multi-source research data integration challenge.

Quality assurance infrastructure: The validation framework (Table ??) provides reusable patterns for data quality verification across research contexts. These automated tests ensure that reused components maintain data integrity standards.

Licensing and attribution: Data are licensed under Creative Commons Attribution 4.0 International (CC BY 4.0) [3], enabling broad reuse with appropriate attribution. The license covers both datasets and

methodology, encouraging adaptation of the semantic layering approach.

Technical infrastructure reusability: The repository can be forked and the dbt pipeline extended for new data sources or research questions. The modular structure supports iterative development—researchers can extend the analytic layer for new questions without modifying upstream transformations.

Methodological transferability: The semantic layering approach addresses fundamental challenges in research data management that extend beyond this specific domain. The methodology’s emphasis on context preservation, transparent harmonization, and stakeholder communication applies to any research requiring multi-source data integration and methodological transparency.

B Data lineage

B.1 Source data and transformation overview

The giveadam project begins with two regional survey datasets stored in `raw_data/`. These datasets required integration across different formats: CSV files [21] from Arunachal Pradesh and Excel exports from the Kobo Toolbox [12] platform for the Tehri region.

To unify these disparate sources, we implemented a comprehensive data transformation pipeline using the Data Build Tool (dbt) [5, 16] within the `dbt_project/` directory. The complete repository architecture supporting this pipeline is detailed in Section 3.2.

B.2 Pipeline architecture and processing

The transformation pipeline consists of SQL scripts [2], automated tests, and comprehensive documentation that convert raw survey data into analysis-ready formats. A significant challenge involved extracting relevant columns from the complex Kobo export structure. We developed a dedicated R script [20, 24] (`scripts/tehri-cols.R`) to identify and map survey questions to data columns, with results documented in `figure_and_tables/tehri_columns.md` [9].

B.2.1 Data quality assurance

Our quality assurance process includes automated identification and exclusion of invalid responses. During processing, we identified and removed two test rows from the Tehri dataset that exhibited substantial missingness and clear indicators of being test data rather than genuine survey responses. This exclusion process is fully documented in `dbt_project/analyses/tehri_rows_excluded.sql` and reflects the methodological transparency principles outlined in Section 3.

B.3 Visual data flow representation

Figure 5 presents the directed acyclic graph (DAG) generated by dbt for the complete transformation pipeline. This visualization serves multiple purposes:

- **Dependency mapping:** Each node represents a data model or transformation step, while edges show the flow of dependencies between them
- **Layer visualization:** The graph clearly demonstrates the four-layer semantic architecture described in Section C.1
- **Quality assurance:** The DAG shows how data validation and testing are integrated throughout the transformation process
- **Reproducibility:** The complete lineage enables full reproduction of any analytic dataset from source data

This architectural approach ensures data integrity and methodological transparency while enabling efficient processing and clear stakeholder communication.

B.4 Model catalog and validation

Table 1 provides a comprehensive catalog of all dbt models created within the `dbt_project/`, organized by semantic layer. These models implement the semantic extension of dbt’s standard layering approach detailed in Section 3.3. Each model serves a specific role in the semantic transformation process, from preserving source context to enabling research-ready analysis.

The validation framework supporting these models is documented in Table ??, which details the automated quality checks applied at each transformation stage. This combination of systematic modeling and rigorous testing ensures both computational efficiency and methodological rigor.

C Semantic layer implementation and observability

The observability framework operationalizes the semantic layering methodology through automated monitoring and validation at each conceptual stage. This implementation demonstrates how semantic design principles translate into technical infrastructure while maintaining research transparency.

Table 1: Data Build Tool (DBT) models created by dbt_project, ordered by observability layer.

Model	Description
Analytic Models	
ana_respondents	Cleaned and transformed respondents data for analysis.
ana_sdg	UN SDG labels and numbers as identifiers.
ana_sdg7	Dataset reporting if SDG 7 was prioritised in top 3 ranking by respondents, with respondent metadata.
ana_top3	Top 3 analytical model for key metrics
Semantic Models	
sem_respondents	Each row represents one respondent in either the survey undertaken in Tehri or the survey undertaken in Arunachale Pradesh.
sem_sdg_labels	This is a transformation table to integrate SDG labelling across the two survey datasets.
sem_top3	Each row a ranking of top 3 SDG priorities, by one respondent from either the Tehri or Arunachal survey. In particular this step ensures that the SDG labels are harmonised across the two surveys.
sem_top3.arunachal.long	Long format of top 3 SDG rankings from Arunachal Pradesh respondents.
sem_top3.tehri.long	Each row represents a respondent's ranking of their top 3 UN Sustainable Development Goals (SDGs) for Tehri, India.
Source Entity Models	
se_respondents.arunachal	Each row represents a respondent from Arunachal Pradesh.
se_respondents.tehri	Each row represents a respondent from Tehri.
se_top3.arunachal	Each row represents a respondent's ranking top 3 UN Sustainable Development Goals (SDGs) for Arunachal Pradesh, India. These data are extracted in wide format from the original survey data preserving labelling for data validation.
se_top3.tehri	Each row represents a respondent's ranking of their top 3 UN Sustainable Development Goals (SDGs) for Tehri, India. These data are extracted in wide format from the original survey data preserving labelling for data validation.
Base Models	
base.arunachal	Raw data from survey in Arunachal Pradesh, collected by Dr Garima Gupta. Each row represents the survey responses of a single respondent. This step loads the data into the pipeline and establishes an identifier for each respondent. Only top 5 ranking and age and gender in these data, an extraction from the full arunachal dataset which is not available due to institutional restrictions. For this pipeline the top 3 rankings are extracted for analysis.
base.tehri	Raw data from survey in Tehri, collected by Dr Garima Gupta. Each row represents the survey responses of a single respondent. This step loads the data into the pipeline and establishes an identifier for each respondent. See 'raw_data/tehri.cols.md' for details on all questions included in this dataset.

C.1 Semantic layers as methodological framework

The four-layer semantic architecture serves as both a conceptual framework and technical implementation.

Each layer embodies specific research methodology principles:

- **Source Base Models:** Embody the principle of data integrity preservation. These foundational models maintain fidelity to original data sources while implementing only essential cleaning operations. By materializing as seeds, they create an immutable record of processed raw data, enabling complete methodological auditing.
- **Source Entities:** Operationalize the context preservation principle. Each regional dataset maintains its original structure and labeling conventions, preventing premature harmonization. This layer en-

ables validation of regional data characteristics and supports comparative analysis of data collection methodologies.

- **Semantic Models:** Implement the transparent harmonization principle. This layer explicitly documents how cross-source differences are resolved, such as reconciling disparate SDG labeling systems. All harmonization logic is encoded in SQL with accompanying documentation, making methodological decisions auditable and modifiable.
- **Analytic Models:** Realize the research-readiness principle. These models combine harmonized data with enriched metadata to directly support specific research questions. By materializing as tables, they optimize performance for iterative analysis while maintaining full lineage to upstream decisions.

C.1.1 Methodological advantages of semantic layering

This semantic architecture delivers specific methodological benefits that standard data engineering approaches cannot provide:

1. **Assumption explicitness:** Every harmonization decision is documented and testable
2. **Context preservation:** Regional and methodological differences are maintained until explicitly resolved
3. **Stakeholder communication:** Layer names and logic reflect research concepts rather than technical implementation
4. **Methodological auditability:** Complete transformation lineage enables scholarly review and replication
5. **Iterative extensibility:** New research directions can be accommodated without fundamental restructuring

C.2 Interactive data exploration

The validation tables presented in this document provide high-level summaries of data quality and pipeline structure. For comprehensive pipeline investigation, stakeholders can access the full interactive documentation using:

```
cd dbt_project/  
dbt docs generate  
dbt docs serve
```

This provides an interactive web interface showing detailed model specifications, column-level lineage, test results, and complete data flow visualization.

C.3 Observability table generation

The validation tables presented in this document (Tables 1 and ??) are automatically generated from dbt artifacts to ensure methodology documentation remains synchronized with the actual pipeline implementation.

C.3.1 Automated metadata extraction

The observability tables are generated through the following automated process:

1. **dbt artifacts generation** — Running `dbt build` produces `manifest.json` and `run_results.json` containing complete pipeline metadata
2. **Python extraction** — `create-obs-tables/get-obs-dat.py` extracts model descriptions and test results from JSON artifacts
3. **R formatting** — `create-obs-tables/obs-table.R` formats extracted data into LaTeX tables
4. **Document compilation** — LaTeX tables are included in this methodology document via `\input` commands

This automation ensures that any changes to model descriptions, test specifications, or pipeline structure are immediately reflected in the methodology documentation, maintaining complete transparency between implementation and documentation.

C.3.2 Model materializations

The semantic layers employ different dbt materializations optimized for their function:

- **Source base models** — Materialized as `seeds` (CSV files loaded directly into DuckDB)
- **Source entity models** — Materialized as `views` to preserve disk space while maintaining fast access for downstream models
- **Semantic models** — Materialized as `views` to enable flexible harmonization logic without storage overhead

- **Analytic models** — Materialized as **tables** to optimize query performance for research analysis and data export

The progression from views to tables reflects the increasing stability and query frequency of models as they approach the analytical layer. Source and semantic layers prioritize flexibility and maintainability through views, while analytic models prioritize performance through table materialization for repeated research queries.

C.4 Data validation

Table ?? lists the dbt tests applied to **dbt_project/** transformations, ordered by semantic layer. These tests document the data quality assumptions validated at each transformation step in the lineage (Figure 5). The tests ensure data integrity across the semantic pipeline and provide automated quality assurance for research reproducibility.

Table 2: Analytic tests applied to dbt_project models in this layer.

Test	Columns	Arguments	Result
ana_respondents			
accepted_values	region	tehri, arunachal	pass
custom	check_n.respondents	NA	pass
not_null	id.respondent	NA	pass
relationships	id.respondent	id.respondent, ref.ana.sdg7	pass
relationships	ana.sdg7.id.respondent	id.respondent, ref.ana.respondents	pass
relationships	id.respondent	id.respondent, ref.ana.top3	pass
relationships	ana.top3.id.respondent	id.respondent, ref.ana.respondents	pass
unique	id.respondent	NA	pass
ana_sdg			
accepted_values	ana.sdg7.gender	Male, Female, Prefer_not_to_say	pass
accepted_values	ana.sdg7.region	tehri, arunachal	pass
custom	check_n.sdgs	NA	pass
not_null	sdg.id	NA	pass
not_null	sdg.label	NA	pass
not_null	sdg.number	NA	pass
not_null	ana.sdg7.id.respondent	NA	pass
unique	sdg.id	NA	pass
unique	sdg.label	NA	pass
unique	sdg.number	NA	pass
unique	ana.sdg7.id.respondent	NA	pass
ana_top3			
not_null	id.respondent	NA	pass
	id.respondent	rank	pass
unique_combination_of_columns			

References

- [1] Scott Chacon and Ben Straub. *Pro Git*. Apress, 2nd edition, 2014.

Table 3: Semantic tests applied to dbt_project models in this layer.

Test	Columns	Arguments	Result
sem_respondents			
accepted_values	gender	Male, Female, Prefer_not_to_say	pass
accepted_values	region	tehri, arunachal	pass
not_null	id_respondent	NA	pass
	id_respondent	NA	pass
unique_combination_of_columns			
sem_sdg_labels			
not_null	sdg_arunachal	NA	pass
not_null	sdg_label	NA	pass
not_null	sdg_tehri	NA	pass
unique	sdg_arunachal	NA	pass
unique	sdg_tehri	NA	pass
sem_top3			
accepted_values	arunachal_long_rank	1, 2, 3	pass
accepted_values	tehri_long_rank	1, 2, 3	pass
accepted_values	rank	1, 2, 3	pass
not_null	arunachal_long_id_respondent	NA	pass
not_null	tehri_long_id_respondent	NA	pass
not_null	tehri_long_rank	NA	pass
not_null	tehri_long_sdg_tehri	NA	pass
not_null	id_respondent	NA	pass
not_null	sdg_label	NA	pass
not_null	sdg_number	NA	pass
	arunachal_long_id_respondent	NA	pass
unique_combination_of_columns			
	tehri_long_id_respondent	rank	pass
unique_combination_of_columns			
	id_respondent	rank	pass
unique_combination_of_columns			

[2] Donald D Chamberlin and Raymond F Boyce. Sequel: A structured english query language. In *Proceedings of the 1974 ACM SIGFIDET (now SIGMOD) Workshop on Data Description, Access and Control*, pages 249–264, Ann Arbor, Michigan, 1974. ACM.

[3] Creative Commons. Creative commons attribution 4.0 international license. <https://creativecommons.org/licenses/by/4.0/>, 2013. CC BY 4.0.

[4] Antica Culina, Ilona Van Den Berg, Simon Evans, and Alfredo Sánchez-Tójar. Low availability of code in ecology: A call for urgent action. *PLoS Biology*, 18(7):e3000763, 2020.

[5] dbt Labs. dbt core: Transform data in your warehouse. <https://github.com/dbt-labs/dbt-core>, 2024.

[6] dbt Labs. How we structure our dbt projects. <https://docs.getdbt.com/best-practices/how-we-structure/1-guide-overview>, 2024.

[7] GitHub, Inc. Github copilot. <https://github.com/features/copilot>, 2024.

Table 4: Source Entities tests applied to dbt_project models in this layer.

Test	Columns	Arguments	Result
se_respondents_arunachal			
accepted_values	gender	Male, Female, Prefer_not_to_say	pass
not_null	id_respondent	NA	pass
unique	id_respondent	NA	pass
	id_respondent	NA	pass
unique_combination_of_columns			
se_respondents_tehri			
accepted_values	gender	Male, Female, Prefer_not_to_say	pass
not_null	id_respondent	NA	pass
unique	id_respondent	NA	pass
	id_respondent	NA	pass
unique_combination_of_columns			
se_top3_arunachal			
not_null	id_respondent	NA	pass
not_null	ranked_sdg_first	NA	pass
not_null	ranked_sdg_second	NA	pass
not_null	ranked_sdg_third	NA	pass
	id_respondent	NA	pass
unique_combination_of_columns			
se_top3_tehri			
not_null	id_respondent	NA	pass
not_null	ranked_sdg_first	NA	pass
not_null	ranked_sdg_second	NA	pass
not_null	ranked_sdg_third	NA	pass
	id_respondent	NA	pass
unique_combination_of_columns			

Table 5: Base tests applied to dbt_project models in this layer.

Test	Columns	Arguments	Result
base_arunachal			
unique_combination_of_columns	id_respondent	NA	pass
	respondents	NA	pass
unique_combination_of_columns			
base_tehri			
not_null	id_respondent	NA	pass
unique	id_respondent	NA	pass
	id	NA	pass
unique_combination_of_columns			
unique_combination_of_columns	id_respondent	NA	pass

[8] GitHub, Inc. Github: Where the world builds software. <https://github.com>, 2024.

[9] John Gruber. Markdown. <https://daringfireball.net/projects/markdown/>, 2004.

[10] Daniel G Hamilton, Kyungwan Hong, Hannah Fraser, Anisa Rowhani-Farid, Fiona Fidler, and Matthew J Page. Prevalence and predictors of data and code sharing in the medical and health sciences: systematic review with meta-analysis of individual participant data. *bmj*, 382, 2023.

- [11] Tom E Hardwicke, Maya B Mathur, Kyle MacDonald, Gustav Nilsson, George C Banks, Mallory C Kidwell, Alicia Hofelich Mohr, Elizabeth Clayton, Erica J Yoon, Michael Henry Tessler, et al. Data availability, reusability, and analytic reproducibility: Evaluating the impact of a mandatory open data policy at the journal cognition. *Royal Society open science*, 5(8):180448, 2018.
- [12] Harvard Humanitarian Initiative. Kobo toolbox: Data collection tools for challenging environments. <https://www.kobotoolbox.org/>, 2024. Cambridge, MA: Harvard Humanitarian Initiative.
- [13] Leslie K John, George Loewenstein, and Drazen Prelec. Measuring the prevalence of questionable research practices with incentives for truth telling. *Psychological science*, 23(5):524–532, 2012.
- [14] Steven Kambouris, David P Wilkinson, Eden T Smith, and Fiona Fidler. Computationally reproducing results from meta-analyses in ecology and evolutionary biology using shared code and data. *Plos one*, 19(3):e0300333, 2024.
- [15] Jeremy Leipzig, Daniel Nüst, Charles Tapley Hoyt, Karthik Ram, and Jane Greenberg. The role of metadata in reproducible computational research. 2(9).
- [16] MotherDuck and dbt Labs. dbt-duckdb: dbt adapter for duckdb. <https://github.com/duckdb/dbt-duckdb>, 2024.
- [17] BBC News. Covid: Test and trace data glitch caused by excel spreadsheet limitations, 2020.
- [18] Amit Prakash. What defines the modern data stack and why you should care.
- [19] Python Software Foundation. *Python Language Reference*, 2024. Version 3.12.
- [20] R Core Team. R: A language and environment for statistical computing. <https://www.R-project.org/>, 2024.
- [21] Yakov Shafranovich. Common format and mime type for comma-separated values (csv) files. RFC 4180, 2005.
- [22] Victoria Stodden, Jennifer Seiler, and Zhaokun Ma. An empirical analysis of journal policy effectiveness for computational reproducibility. *Proceedings of the National Academy of Sciences*, 115(11):2584–2589, 2018.
- [23] Ana Trisovic, Katherine Mika, Ceilyn Boyd, Sebastian Feger, and Mercè Crosas. Repository Approaches to Improving the Quality of Shared Data and Code. 6(2):15.

- 579 [24] Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain
580 François, Garrett Grolemond, Alex Hayes, Lionel Henry, Jim Hester, et al. Welcome to the tidyverse.
581 *Journal of Open Source Software*, 4(43):1686, 2019.
- 582 [25] Mark D Wilkinson, Michel Dumontier, Ijsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton,
583 Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al.
584 The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3(1):160018,
585 2016.
- 586 [26] Yu Xie, Kai Wang, and Yan Kong. Prevalence of research misconduct and questionable research prac-
587 tices: A systematic review and meta-analysis. *Science and engineering ethics*, 27(4):41, 2021.
- 588 [27] Mark Ziemann, Yotam Eren, and Assam El-Osta. Gene name errors are widespread in the scientific
589 literature. *Genome Biology*, 17(1):177, 2016.

UN SDG priorities in Arunachal Pradesh and Tehri, by gender

Little difference between male and female respondents' priorities



Sized by the number of respondents who prioritised (in top 3) each UN SDG. One respondent chose 'Prefer not to say' and is excluded from this visualisation.

Figure 4: Treemap of top 3 UN SDG priorities from survey respondents in Tehri and Arunachal Pradesh, North India, by gender. Each rectangle represents a specific UN SDG, with its size proportional to the number of respondents of a particular gender who ranked it among their top three priorities. The treemap visually highlights the most and least prioritized SDGs in each region, providing insights into gender differences in development priorities.



Figure 5: Data Build Tool (dbt) directed acyclic graph (DAG) showing the semantic layer structure from source entities through analytic models.