ABSTRACT

Open data is a key resource for extracting indicators that support decision making in urban planning, environmental monitoring, and sustainable development. While indicators follow standardized definitions, open data remains fragmented, schema less, and inconsistent, making integration and computation challenging. This paper presents `SDG-KG`, a Knowledge Graph (KG) framework for the integration, mapping, and validation of indicators using multiple datasets. We illustrate our approach with the Sustainable Development Goals (SDGs), for which our framework structures textual definitions, guides data retrieval, and formalizes indicator computation. Using graph based modeling, we build both an `SDG Graph` and a `Metadata Graph`, resolve conflicts between sources, and generate query driven execution workflows. Experiments on six countries and two SDG indicators demonstrate the effectiveness, and interpretability of the proposed framework.

CCS CONCEPTS

• **Information systems** → **Graph-based database models**; • **Computing methodologies** → **Semantic networks**; • **Software and its engineering** → *Automatic programming*.

KEYWORDS

Knowledge Graph, Open Data, Data integration, SDG

10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Tracking progress is important for understanding how societies evolve. It helps measure economic growth, social development, and environmental impact. Indicators play a central role in this process. They provide concrete evidence to evaluate policies, identify gaps, and guide decision making. In urban planning, they reveal population growth and infrastructure needs. In environmental monitoring, they track pollution and resource consumption. For sustainable development, they measure progress towards global goals. Reliable and accurate indicators are essential for effective action and sustainable planning [17, 21].

Open data is a powerful resource for building these indicators. It is accessible, transparent, and available across many domains. Governments, organizations, and international agencies publish datasets on health, education, environment, and economic development [9, 26]. These datasets provide insights at global, national, and local levels. They enable policymakers to monitor progress and make informed decisions [3].

However, open data is often fragmented and inconsistent. It is stored in different formats like CSV, GeoJSON, SQL dumps, and APIs. It varies in granularity, spatial coverage, and update frequency. These differences make it hard to combine and analyze data effectively [1, 20]. Traditional methods struggle to harmonize these diverse datasets. There is a need to integrate and compute indicators across multiple sources.

In the context of sustainable development, the United Nations defined 232 indicators to monitor progress across 17 *Sustainable Development Goals* [13] (SDG). Many indicators, especially *Tier II* and *Tier III*, lack data or standard methods. Several national and international bodies encourage the use of open data to improve their coverage and computability. This requires finding relevant datasets, aligning schemas, and formalizing indicator logic.

A promising solution to this challenge is the use of Knowledge Graphs (KGs), which provides a flexible and semantically rich data model for integrating diverse sources while preserving relationships between entities [4, 8]. KGs have been widely adopted for data integration and semantic interoperability across disparate sources [14]. Additionally, Large Language Models (LLMs) have demonstrated strong capabilities in schema matching and entity linking, making them valuable for aligning heterogeneous datasets [16].

To understand how it works, consider the query shown in Figure 1. It focuses on SDG Indicator 11.2.1, which measures the share of population with access to low capacity public transport. The needed data exists in open platforms, but with different schemas and variable names. For example, `geometry` and `ArRGeopoint` describe the same thing but are not linked. Values like `bus` must be mapped to SDG terms such as `low capacity`. To handle this, we model the indicator in a knowledge graph. It defines concepts, attributes, and allowed values from official SDG metadata. Using LLMs, we align attributes and map values, even when formats or labels differ. Once the graph is populated, we can run queries that retrieve only the relevant data. This enables clean and accurate computation, even from fragmented and inconsistent sources.

In this paper, we present SDG-KG, a framework that supports the structured computation of SDG indicators from open and heterogeneous data sources. Our approach leverages a *dual knowledge graph model* that integrates both indicator semantics and dataset metadata, enabling automated query driven selection, alignment, and execution.

We summarize our contributions as follows:

- We introduce a dual-graph based framework that models indicators, data sources, and metadata to enable transparent and reproducible indicator computation from open data by generating source code indicators.
- We address three key challenges: schema mismatches across sources, conflicting or redundant datasets, and the absence of formal computation rules. We show how our dual graph architecture, combined with LLM-assisted mappings and a trust based filtering mechanism, addresses each.
- We evaluate SDG-KG on real world queries for SDG indicators (e.g., 11.2.1 and 11.6.2), using datasets from six countries. Results show accurate value computation, interpretable source selection, and robustness across metadata profiles.

The remainder of this paper is structured as follows: Section 2 reviews related work on data integration, indicator computation, and the use of knowledge graphs and language models. Section 3 introduces the key components of our modeling approach, including the SDG Graph and Metadata Graph. Section 2.2 presents the SDG-KG framework, detailing its architecture, data flow, and trust based source selection strategy. Section 5 reports the experimental evaluation on indicator computation, mapping quality, and metadata filtering. Section 6 concludes the paper.

2 RELATED WORK

2.1 Knowledge Graphs for Data Integration

Knowledge Graphs (KGs) are widely used for integrating heterogeneous data sources. They represent data as a set of entities and relationships, enabling semantic alignment across domains. KGs provide a flexible schema and support reasoning, making them useful for unifying structured and semi- structured data [8]. However, most KGs focus on entity level integration and do not capture the procedural or computational logic required for tasks like indicator computation.

Traditional integration methods struggle [5, 18] with schema heterogeneity and context loss. KGs address this by incorporating ontologies and shared vocabularies, which help map concepts across datasets [22]. This makes KGs effective for open data integration, especially when sources vary in format and semantics.

Virtual Knowledge Graphs (VKGs) extend this idea by providing an abstraction layer over existing databases without requiring data duplication [23]. Queries are rewritten dynamically into the sources, enabling access to distributed data through a unified semantic interface [2]. VKGs are increasingly used in industrial settings where data is too large or sensitive to move. Yet, they assume high metadata quality and lack mechanisms to rank or prioritize among multiple sources

Recent works also explore the combination of KGs with Large Language Models (LLMs) to improve schema matching and data discovery across sources. These hybrid systems leverage the structure of KGs and the contextual understanding of LLMs, offering promising results for open data integration [15]. All these recent techniques have not yet been applied to the studied domain because of the lack of structure in both open data and indicator documentation. It implies a high complexity for the computation of indicators or even the query itself that requires to be tuned manually.

2.2 Schema Matching and Computation

Knowledge Graphs are well suited for computing indicators, especially in the context of the SDGs. They can model indicators, data sources. Santos et al. [19] built a city level KG to structure urban mobility indicators and automate dashboard generation. Their work shows how KGs support dynamic computation from diverse inputs. An other approach [ANONYMIZED] combined KGs with Large Language Models (LLMs) to improve schema matching and entity linking. This approach enhances alignment between open datasets and SDG definitions, supporting scalable and accurate indicator computation. Together, these works show that KGs, especially when combined with LLMs, are a promising solution for modeling indicators and fragmented data.

Matching schemas across datasets is a core challenge in data integration. Traditional methods rely on rules or string similarity [18]. Recent work uses embeddings and language models to better capture semantic alignment [25]. In KGs, entity linking is used to match attributes and values to graph nodes. These techniques are critical when working with open data, where metadata is often missing or inconsistent [8].

While KGs and LLMs are effective for schema alignment, most approaches stop at the mapping stage and don't cover the full pipeline from source selection to indicator computation. They often ignore metadata like time or space, and rarely assess how mapping quality affects final results. There's still a gap in building end-to-end, explainable workflows.

3 PRELIMINARIES

Our work focuses on computing indicators using a KG based representation. In this section, we present the core modeling components that support this process. We define the structure of the indicator model, the representation of data sources, and the mapping.

3.1 SDG Graph

The Sustainable Development Goals (SDGs) are organized into a hierarchy of goals, targets, and indicators, each designed to measure global progress. While prior models like LinkedSDGs and SustainGraph [7] capture SDG concepts and target evolution, SDG Graph focuses on structuring indicators for computation using multiple and heterogeneous data sources from open data.

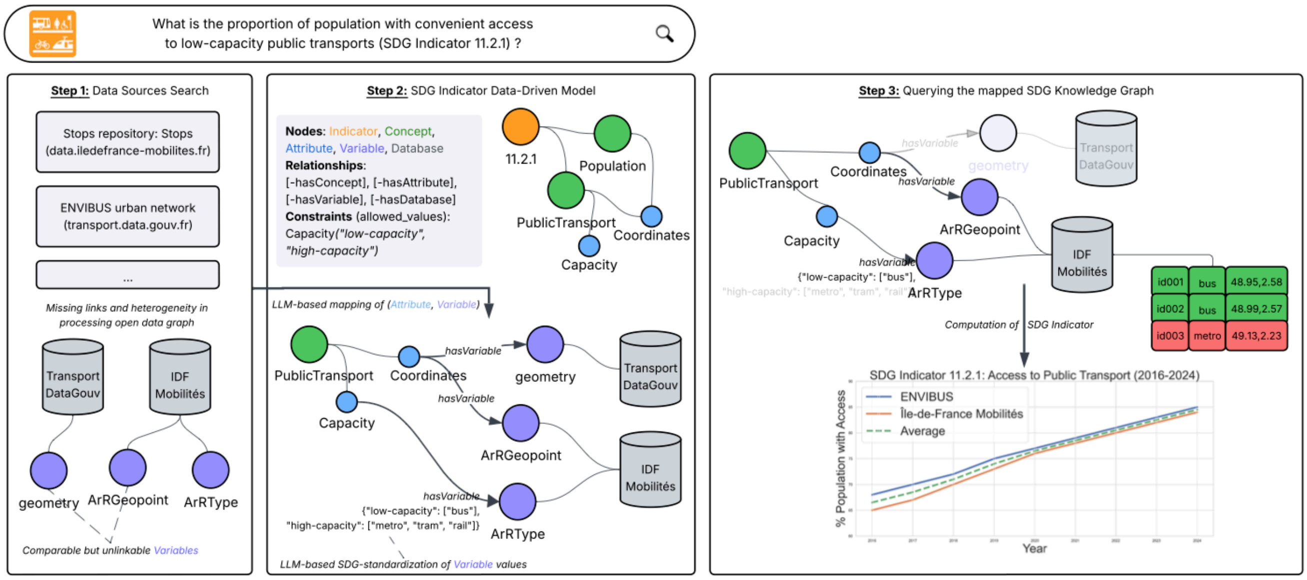

Figure 1: Computing SDG Indicator 11.2.1 from open data. Step 1: collect datasets. Step 2: model the indicator in a knowledge graph. Step 3: align data and compute the result.

The `SDG Graph` represents the hierarchical structure of the SDGs, where entities and relationships are explicitly defined following the United Nations (UN) *SDG Indicator Metadata Repository*. This structure is automatically generated as part of a data-driven approach and results from a previous expert-validated contribution.

Each node in the graph represents a well defined concept from the SDG documentation. Relationships describe how concepts are connected, from goals to indicators, and from indicators to variables needed for computation. The graph helps formalize indicator logic and connects it to data through the `Metadata Graph`.

The model consists of the following components:

- **Goals:** High level objectives that address global issues (*e.g., Sustainable Cities and Communities* - SDG 11).

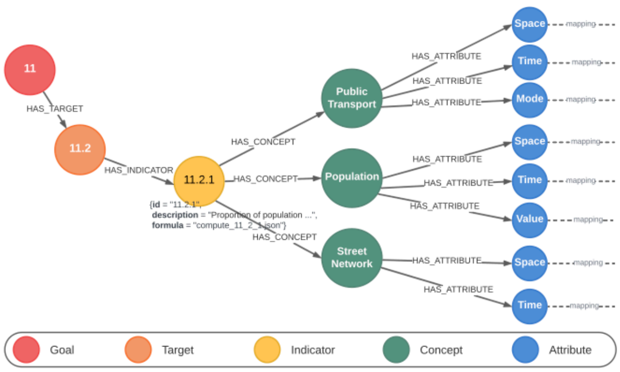

Figure 2: Sample of SDG Graph

- **Targets:** Specific aims within each goal that define measurable outcomes (*e.g., Provide access to safe, affordable, accessible, and sustainable transport systems*, Target 11.2).
- **Indicators:** Quantifiable metrics for evaluating progress toward each target (*e.g., Proportion of the population that has convenient access to public transport, by sex, age, and persons with disabilities*, Indicator 11.2.1).
- **Concepts:** Domain specific entities related to SDG measurement (*e.g., Population, Public Transport, Street Network*).
- **Attributes:** Descriptive properties of concepts used in calculations (*e.g.,* for *Population*: *Space, Time, Age Group, Sex*).

Concepts and attributes form the basis for computing an indicator. Each concept expects specific input data, which must be linked from real datasets. These inputs include spatial and temporal scopes, disaggregation dimensions, and value types. It worth noting that each concept node is linked to 1 *Space* node, 1 *Time* node and at least 1 *Value* node. This structure is a key structure for SDG computation which requires the alignment of data on space and time all along the process (Section 4.4).

Figure 2 illustrates a sample *SDG Graph*, highlighting the nodes and relationships between goals, targets, indicators, concepts, and attributes. To enable automatic computation, open dataset attributes are mapped to predefined SDG attributes. These mappings are used later in query generation and indicator calculation.

3.2 SDG Indicator Formula

To compute an indicator, the corresponding formula is embedded in the `SDG Graph`'s indicator node, represented in a structured JSON format that defines how involved concept nodes are transformed and aggregated. The *formula* is extracted from the UN Metadata

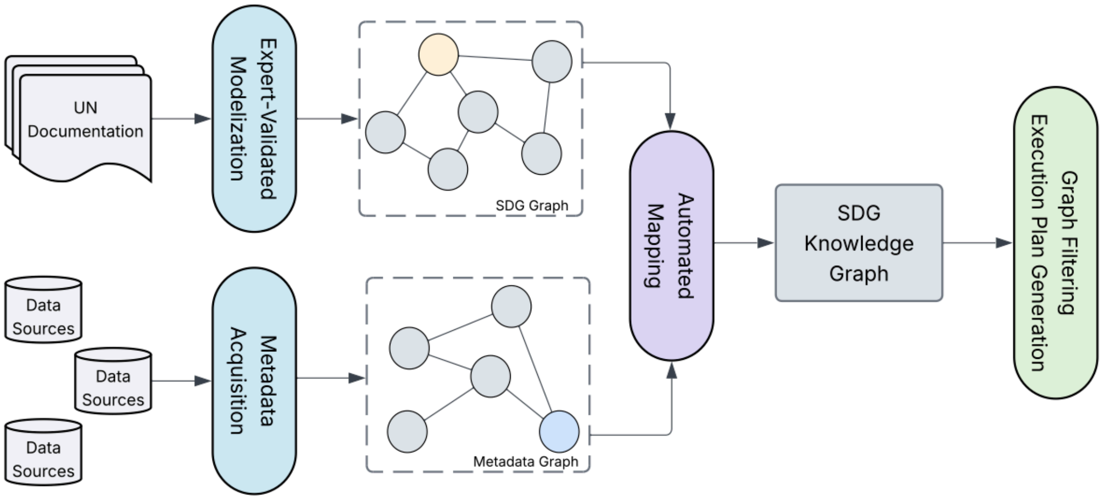

Figure 3: The SDG-KG's framework of our proposed method.

repository, where it is expressed in natural language or mathematical notation and must be encoded.

Take the following JSON as an example (refers to the *formula "compute_11_2_1.json"* of the indicator node in Figure 2):

```
[
  {"FilteredPop":{"$Filter":["Population","$Qc"]}},
  {"AccessPop":[
      "FilteredPop",
      {"$ComputeDistance": ["PublicTransport", "StreetNetwork"]}
  ]},
  {"RETURN": {"$div": ["AccessPop", "FilteredPop"]}}
]
```

Here, the JSON document is an ordered list of computation steps composed of documents with variables (keys), ending with a "RETURN" value. The "`FilteredPop`" variable represents the subset of population (concept node) linked to the use case's context $\$Q_c$ (gender, age, etc.). The "`AccessPop`" variable is the subset of this population with convenient access to public transport where the link between the population ("`FilteredPop`") and transports ("`$ComputeDistance`") is implied by the alignment of space and time values from each piece of data. The "`$ComputeDistance`" operation calculates the proximity between population clusters (extracted from the "`FilteredPop`" variable) and public transport stops using the "`StreetNetwork`". Finally, the "`RETURN`" statement computes the population proportion with access by dividing "`AccessPop`" by the total "`FilteredPop`". The formula guides the combination of concept nodes (string values in red) in order to capture any required attribute. Data linkage through space and time is implied (and treated during the execution plan generation Section 4.4).

4 PROPOSED METHOD

Our method connects open data to SDG indicators through structured modeling and automated alignment. It is based on a dual graph approach. The first graph captures indicator structure; the second describes datasets and their metadata. These graphs are combined and queried to compute the indicator.

Figure 3 presents an overview of the process. It starts by building the Metadata Graph from available datasets. Then, the SDG Graph is generated from official documentation. The two graphs are aligned using LLM-assisted mapping. The resulting graph is filtered based on the user query. Finally, the indicator is computed using a rule-based execution plan.

4.1 Metadata Graph

The Metadata Graph (`Metadata Graph`) describes the structure and content of each dataset [6]. It includes nodes for datasets, their attributes, and metadata. Each attribute is enriched with dimensions such as spatial and temporal coverage, granularity, source reliability, and completeness. The `Metadata Graph` follows *Open SDG*[1] format. Open SDG is a standard data structure for SDG reporting. It is used by number of countries and institutions to publish their indicator data. The format defines how each dataset sould be organized. It includes fields for indicator codes, dimensions, time periods, values and units. It also provides a way to include metadata such as definitions, sources and computation methods.

By following this structure, the `Metadata Graph` is compatible with official SDG data and makes it easier to connect new datasets.

This graph allows to make reasoning about data quality and relevance. It also enables flexible selection of sources based on user constraints. Similar metadata-driven approaches are used in prior works on data discovery and selection [12].

The `Metadata Graph` is designed to be queryable and extensible. For example, a user can retrieve all datasets that cover a specific region at a given time resolution. This structure also helps exclude sources that do not meet a minimum reliability threshold. For spatial and temporal descriptors, we rely on controlled vocabularies (*e.g.*, NUTS levels for regions). When these are missing, we extract them using LLM-based parsing from headers or descriptions.

[1] https://open-sdg.readthedocs.io/en/latest/data-format/

Table 1: Operators' Definition Involved in Algorithm 1

GetConcept	$\text{GetConcept}(D \in G^*, \mathcal{Q}_I) = R$
Filter	$\theta_{filter_c}(R) = \{r \in R \mid \sigma_c(r)\}$
ComputeDistance	$\theta_{CD_\tau}(R_1, R_2) = \{(s_i, t_i, v_i) \in R_1 \mid \exists (s_j, t_i, v_j) \in R_2 \text{ such that } \mathtt{dist}(s_i, s_j) \leq \tau\}$
Aggregation	$\theta_{\text{agg}}(R) = r', \text{ where } r' \text{ summarizes } R \text{ under } \oplus \mid \oplus : \mathcal{P}(R) \to \mathbb{R}^m, \oplus(R) = \text{agg}(v_1, v_2, \ldots, v_n)$
Align	$\theta_{Align}(R_1, R_2) = \{(s, t, (v_1, v_2)) \mid (s, t, v_1) \in R_1, (s, t, v_2) \in R_2\}$

4.2 Automated Mapping

Once both graphs are created, we map dataset attributes to SDG concepts. This step is partially automated using an LLM. It helps resolve naming mismatches and align semantics. For example, it maps `geometry` to `ArRGeopoint`, or `bus` to `low capacity`.

The mapping involves two levels. First, we align attributes to SDG concepts. Then, we align value vocabularies. The goal is to translate dataset specific codes into SDG compliant terms. LLMs have been shown effective for this task [11], especially when metadata is sparse. All mappings are stored in the graph, with scores (see below) and traceability. They can be reviewed and refined by experts if needed.

To ensure mapping quality, each link generated by the LLM is assigned a confidence score. This score is computed using cosine similarity between the embedding of the predicted label and that of the target concept in the SDG Graph. It captures how close the two terms are semantically, even if they differ lexically. For example, pop_acc_lc_500m is correctly matched to Proportion of population with access to low-capacity public transport within 500 meters. Low-confidence mappings (*e.g.*, $score < 0.7$) are flagged for expert review. This semi-automated process balances precision and scalability. In our experiments, LLMs outperform traditional string similarity methods, especially in multilingual cases or when values are abbreviated. This mapping step allows us to align the `SDG Graph` and `Metadata Graph`, producing the unified and executable graph: `SDG-KG`.

4.3 Graph Filtering

In order to answer a query from a given indicator, space and time parameters, we need to extract relevant parts of the `SDG-KG` to focus on proper open data values. The `SDG-KG` $\mathcal{G}$ helps to provide a set of solution subgraphs $\mathcal{G}_I \subset \mathcal{G}$ where the indicator node I (from use case Q_I), in the `SDG Graph`, is linked to source nodes $D \subset \mathcal{D}$ in the `Metadata Graph`.

The goal of this step is to select the most relevant graph $G^* \subseteq \mathcal{G}_I$ containing the best set of sources D among the available sources $\mathcal{D} \subset \mathcal{G}_I$. To achieve this, we propose the *Trust Score Metric (TSM)* that quantifies the alignment between a dataset's metadata and the user-defined query parameters.

Definition 4.1. Let $D \subseteq \mathcal{D} \subset \mathcal{G}_I$ be a dataset with metadata $D = \{D_s, D_t, D_c, D_r, D_p\}$ respectively representing its spatial, temporal, contextual granularity, reliability and completeness information. Let $Q_I = \{q_s, q_t, q_c\}$ be a query specifying spatial, temporal, and contextual constraints. The TSM of D regarding Q_I is defined as:

$$TSM(D, Q_I) = \frac{\sum_{i \in \{s,t,c\}} [w_i \cdot \text{Sim}(D_i, q_i)] + w_r \cdot u(D_r) + w_p \cdot v(D_p)}{\sum_{i \in \{s,t,c,r,p\}} w_i}$$

where:

- w_s, w_t, w_c, w_r, w_p represent user-defined normalized weights [10] controlling the importance of each factor, $\sum w_i = 1$,
- $\text{Sim}(D_i, q_i)$ is the similarity function for each dimension (spatial s, temporal t, contextual c - $q_i \in Q_I$),
- $u(D_r)$: Reliability score, assessing the trustworthiness of the dataset source (*e.g.*, governmental, NGO, crowdsourced data). $u(D_r) \in [0, 1]$,
- $v(D_p)$: Completeness score, indicating the proportion of expected records available in the dataset. $v(D_p) \in [0, 1]$.

The trust score serves as a ranking function over all possible dataset combinations. This is important when several datasets cover the same concept and parameters (space and time from $\mathcal{Q}_I$) but differ in quality or relevance. For example, two datasets may both contain population data, but only one may provide disaggregated information by age or sex. Our metric ensures that the selected subgraph G^* contains the most relevant and complete sources for computing the requested indicator.

The resulting TSM score is normalized between 0 and 1, where higher values indicate a closer match to the query. This formulation ensures that selected datasets for indicator computation best align with user-defined spatial, temporal, and contextual constraints with respect to the source reliability and completeness score.

The problem of finding the best candidate graph G^* can be formalized as the following:

$$G^* = \arg\max_{G \subset \mathcal{G}_I} \prod_{D \subseteq \mathcal{D} \cap G} \text{TSM}(D, Q_I) \quad (1)$$

4.4 Execution Plan Generation

Once the indicator node I and a use case Q_I are chosen, as well as the best subgraph solution $G^* \subseteq \mathcal{G}_I$, it requires to translate this subgraph into an execution plan that links open data sources D to the I node's formula computation f_I. The indicator's formula f_I given as a JSON document (Section 3.2) is composed of explicit (in the formula: *$Filter*, *$ComputeDistance*, *$Div*, etc.) and implicit operations (*Align*, *SUM*, etc.) applied on nodes (extracted by *GetConcept* and variables which structures the execution plan.

Each concept from the `SDG Graph` is linked to at least space (s), time (t) attributes and value nodes (v) (cf. Section 3.1). Thus, the use case Q_I is applied on each linked source to get corresponding data $R = [s, t, (v)]$.

Algorithm 1 Generated execution plan for Indicator 11.2.1

Input: A Solution Graph: G^*, Query Parameters $Q_I = \{q_s, q_t, q_c\}$

Step 1: Retrieve concepts from the graph

1: PT $\leftarrow$ GetConcept(G^*, "PublicTransport", q_s, q_t)

2: Pop $\leftarrow$ GetConcept(G^*,"Population", q_s, q_t)

3: SN $\leftarrow$ GetConcept(G^*, "StreetNetwork", q_s, q_t)

Step 2: Filter Population

4: FilteredPop $\leftarrow$ Filter(Pop, q_c)

Step 3: Compute Service Area & Population Within

5: ServiceArea $\leftarrow$ ComputeDistance(PT, SN, 500)

6: AccessPop $\leftarrow$ Align(FilteredPop, ServiceArea)

Step 4: Compute Indicator Value

7: FilteredPopV $\leftarrow$ Sum(GetAttribute(FilteredPop, "Value"))

8: AccessPopV $\leftarrow$ Sum(GetAttribute(AccessPop, "Value"))

RETURN (AccessPopV / FilteredPopV)

The execution plan θ_I is composed of operations $\theta \in \Theta$ to compute the indicator value (defined in Table 1). Each operator θ is defined as a transformation $\theta : R \rightarrow R$ and the execution plan is a composition of such operators:

$$\theta_{\mathrm{I}} = \theta_n \circ \theta_{n-1} \circ \cdots \circ \theta_1$$

Our approach aims at generating execution plans like Algorithm 1 relying on the `SDG-KG` graph. The execution plan is generated by applying the following steps:

- First each concept node is identified from Indicator nodes (graph traversal) and apply *GetConcept* to apply use case's space and time constraints on G^* (Step 1);
- For each key value pairs (except RETURN) generates a variable with the corresponding operations (Steps 2 and 3). This step is recursive (*e.g.,* lines 5 and 6). Notice that implied alignment operators (list of variables & operations) link data on related space areas s and time period t (Line 6 & Table 1);
- Finally, the RETURN step seeks for Value data (`GetAttribute`) with a graph traversal to mapped nodes in the `Metadata Graph`. By default a sum[2] of values is computed. The final computation formula is applied with previously computed variables.

The output of our approach is a source code like in Algorithm 1 that computes the indicator on a given use case Q_I.

Notice that the `GetAttribute` operation handles sources $D \subset G^*$ from the `Metadata Graph` with various capabilities (SQL, NoSQL, CSV). Implementation details are out of the scope this article.

Thanks to our approach composed of operators and the final execution plan, we can infer some properties: **compositionality** (each θ_i is closed over R, enabling modular and traceable pipelines), **reproducibility** (θ_I is fully determined by G^* and the indicator formula f_I), **inspectability** (intermediate results at each θ_i can be extracted for validation or adjustment).

5 EXPERIMENTS

In this section, we evaluate our framework by answering the following Research Questions (RQs):

[2]The number of raws in the Open SDG format depends on the level of granularity of geographic areas (national, regional, city, etc.)

6 CONCLUSION

In this work, we introduced SDG-KG, a knowledge graph based framework for computing SDG indicators from open and heterogeneous data. The approach relies on a dual graph model: one for indicator semantics and one for dataset metadata. It supports automatic mapping, trust based source selection, and indicator execution. Experiments on real world data show that SDG-KG produces accurate values, enables interpretable source filtering, and scales across diverse contexts. These results confirm that open data and

[5] https://databank.worldbank.org/source/sustainable-development-goals-(sdgs)

structured metadata can support scalable and explainable indicator computation.

Future work includes integrating domain specific statistical corrections, such as those applied by national agencies, to refine computed values. We also plan to support indicator computation at multiple spatial and temporal granularities that represent an underexplored aspect in current literature. Finally, we aim to model and explain the impact of major disruptive events (*e.g.*, pandemics, policy shifts) on indicator trends, using knowledge propagation within the graph.

REFERENCES

[1] Christian Bizer, Tom Heath, and Tim Berners-Lee. 2009. Linked Data—The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)* 5, 3 (2009), 1–22.

[2] Diego Calvanese, Benjamin Cogrel, Simon Komla-Ebri, Roman Kontchakov, Davide Lanti, Maria Rezk, Mariano Rodriguez-Muro, and Guohui Xiao. 2017. Ontop: Answering SPARQL queries over relational databases. In *Semantic Web*, Vol. 8. IOS Press, 471–487.

[3] Tim Davies and Stephen Walker. 2019. The state of open data: Histories and horizons. *African Minds* (2019).

[4] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, USA) *(KDD '14)*. ACM, 601–610. https://doi.org/10.1145/2623330.2623623

[5] Xin Luna Dong and Divesh Srivastava. 2015. *Motivation: Challenges and Opportunities for BDI.* Big Data Integration. Springer International Publishing, 1–29. https://doi.org/10.1007/978-3-031-01853-4_1

[6] Rebecca Eichler, Corinna Giebler, Christoph Gröger, Holger Schwarz, and Bernhard Mitschang. 2021. Modeling metadata in data lakes—A generic model. *Data Knowledge Engineering* 136 (2021), 101931. https://doi.org/10.1016/j.datak.2021.101931

[7] E. Fotopoulou, I. Mandilara, A. Zafeiropoulos, C. Laspidou, Gi. Adamos, P. Koundouri, and S. Papavassiliou. 2022. SustainGraph: A knowledge graph for tracking the progress and the interlinking among the sustainable development goals' targets. *Frontiers in Environmental Science* 10 (2022).

[8] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. De Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann. 2021. Knowledge Graphs. *ACM Comput. Surv.* 54, 4 (jul 2021), 37 pages.

[9] Marijn Janssen, Yannis Charalabidis, and Anneke Zuiderwijk. 2012. Open data as a tool for innovation and accountability: A typology of governance models. *Government Information Quarterly* 29, 4 (2012), 258–268.

[10] R. L. Keeney, H. Raiffa, and David W. Rajala. 1979. Decisions with Multiple Objectives: Preferences and Value Trade-Offs. *IEEE Transactions on Systems, Man, and Cybernetics* 9, 7 (1979), 403–403.

[11] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 1613, 15 pages.

[12] Fatemeh Nargesian, Erkang Zhu, Chen Pu, Renée J Miller, and Xin Luna Dong. 2018. Data lake management: challenges and opportunities. *Proceedings of the VLDB Endowment* 11, 12 (2018), 2022–2025.

[13] United Nations. 2015. Transforming our World: The 2030 Agenda for Sustainable Development. (2015).

[14] N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, and J. Taylor. 2019. Industry-scale knowledge graphs: lessons and challenges. *Commun. ACM* 62, 8 (July 2019), 36–43. https://doi.org/10.1145/3331166

[15] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Transactions on Knowledge and Data Engineering* (2024), 1–20. https://doi.org/10.1109/tkde.2024.3352100

[16] M. Parciak, B. Vandevoort, F. Neven, L. M. Peeters, and S. Vansummeren. 2024. Schema Matching with Large Language Models: an Experimental Study. In *TaDA Workshop at VLDB'24*. VLDB.org.

[17] László Pintér, Peter Hardi, and Peter Bartelmus. 2021. Use of Indicators in Strategic Environmental Assessments of Urban-Planning Instruments: A Case Study. *Sustainability* 13, 22 (2021), 12639.

[18] Erhard Rahm and Philip A Bernstein. 2001. A survey of approaches to automatic schema matching. *The VLDB Journal* 10, 4 (2001), 334–350.

[19] Henrique Santos, Victor Dantas, Vasco Furtado, Paulo Pinheiro, and Deborah L. McGuinness. 2017. From Data to City Indicators: A Knowledge Graph for Supporting Automatic Generation of Dashboards. In *The Semantic Web: 14th International Conference, ESWC'17* (Portorož, Slovenia). Springer-Verlag, 94–108. https://doi.org/10.1007/978-3-319-58451-5_7

[20] Patrick Schopf, Lorenz M. Hilty, and Bastian Wepner. 2020. Big Data Management: Data Governance Principles for Big Data and Open Data. *IEEE Access* 8 (2020), 193–201.

[21] Rajib Singh, Hrudaya N. Murty, Sunil K. Gupta, and Anil K. Dikshit. 2017. The Role of Indicator-Based Sustainability Assessment in Policy and Decision-Making. *Sustainability* 9, 6 (2017), 1018.

[22] Gerhard Weikum. 2021. Knowledge Graphs 2021: A Data Odyssey. *Proceedings of the VLDB Endowment* 14, 12 (2021), 3233–3238.

[23] Guohui Xiao, Linfang Ding, Benjamin Cogrel, and Diego Calvanese. 2019. Virtual Knowledge Graphs: An Overview of Systems and Use Cases. *Data Intelligence* 1, 3 (06 2019), 201–223. https://doi.org/10.1162/dint_a_00011

[24] Hui Yin, Amir Aryani, and Nakul Nambiar. 2024. Evaluating the Performance of Large Language Models for SDG Mapping (Technical Report). arXiv:2408.02201 [cs.LG] https://arxiv.org/abs/2408.02201

[25] Yunjia Zhang, Avrilia Floratou, Joyce Cahoon, Subru Krishnan, Andreas C. Müller, Dalitso Banda, Fotis Psallidas, and Jignesh M. Patel. 2023. Schema Matching using Pre-Trained Language Models. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. 1558–1571. https://doi.org/10.1109/ICDE55515.2023.00123

[26] Anneke Zuiderwijk, Marijn Janssen, Sunil Choenni, Ronald Meijer, and Rachel SC Alibaks. 2014. Innovation with open data: Essential elements of open data ecosystems. *Information Polity* 19, 1-2 (2014), 17–33.