AI-based knowledge graph construction and distributed storage for collaboration on the Sustainable Development Goals

Irene Kilanioti
NTUA
GR
eirinikoilanioti@mail.ntua.gr

George Angelos Papadopoulos
University of Cyprus
CY
papadopoulos.george@ucy.ac.cy

Abstract

The achievement of the Sustainable Development Goals (SDGs) is crucial for future generations. The plethora of SDG data available for analysis facilitates the tasks of practitioners that gather and assess SDG data, including intergovernmental organizations, government agencies and social welfare organizations. In this paper, we propose a framework that aspires to have a substantial impact for SDG practitioners: We propose AI-based construction of SDG knowledge graphs. AI-based methods along with dimensionality reduction undertake the task to semantically cluster new uncategorised SDG data and novel indicators and efficiently place them in the environment of a distributed knowledge graph store.

CCS Concepts

• **Information systems** → *Distributed storage*; *Ontologies*; • **Computing methodologies** → *Distributed algorithms*; **Natural language processing**; • **Applied computing** → *Computing in government*.

Keywords

Sustainable Development Goals ontology, distributed knowledge graphs, Hilbert Space Filling Curves, Artificial Intelligence

ACM Reference Format:
Irene Kilanioti and George Angelos Papadopoulos. 2024. AI-based knowledge graph construction and distributed storage for collaboration on the Sustainable Development Goals. In *13th Conference on Artificial Intelligence (SETN 2024), September 11–13, 2024, Piraeus, Greece.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3688671.3688736

1 Introduction

1.1 Motivation and Related work

The Sustainable Development Goals (SDGs) are a measurable international initiative in the framework of the United Nations (UN) 2030 Agenda for Sustainable Development, that aims to eradicate poverty, safeguard the environment and maintain peace and welfare. SDG ontology comprises substantially sustainable development goals, targets, indicators and data series for the quantification of their accomplishment, and the full taxonomy is accessible as linked open data at [8]. Depending on the grade of development of internationally established methodology and standards as well as regularity of production of relevant data, the afore-mentioned indicators are categorised into tiers.

The impact of actions of organizations that harvest and process SDG data needs a specific context to be perceived in its wholeness. Their pursued goals and undertaken actions can be thematically interweaved and mutually influenced: one agent may pool interrelated existing content from federated repositories or one undertaken action may affect currently ongoing activities of other agents. Knowledge graphs are ideal for manifold interpretations of the societal implications of actions that apply to each SDG goal, for association of contributions of actions to concrete SDG targets and quantification of the exerted influence. Concerning SDGs, knowledge graphs: i) support explainable decisions and insightful recommendations, ii) measure the influence of users and communities and iii) improve the user experience, as they facilitate extraction and organization of knowledge in a distributed manner and serve for distributed knowledge matching.

Challenges associated with the uptake of distributed knowledge graph technologies include automated knowledge graph construction, their efficient storage and use at scale [1]. Heretofore, SDG related systems have included in essence solely monitoring tools of SDGs data and metadata and mechanisms to enhance interoperability across independent information systems: UN [7] showcases use of mappings of terms to the UN Bibliographic Information System (UNIBIS) and the EuroVoc vocabularies.

Section 2 describes the methodology we followed (i) for AI-based SDG knowledge graph construction, and (ii) AI-based semantic similarity is used to store the SDG data. Then, section 3 describes a detailed case study in a distributed knowledge graph environment, that experimentally evaluates our algorithm. Dataset and experimental setup are thoroughly discussed. The results are presented and discussed in light of theory along with the actual impact of the work in Section 4. Section 5 summarizes the paper's contribution and discusses future extensions of our work.

2 Methodology

2.1 AI-based knowledge graph construction

We use the following AI models to construct SDG knowledge graphs:

- NLP-based Model using Named Entity Recognition (NER): The NLP-based model uses Named Entity Recognition (NER) to identify and classify entities within text (e.g., countries, indicators, and units) within the seriesDescription field. Whereas the model performs well in terms of identification of specific

SETN 2024, September 11–13, 2024, Piraeus, Greece

ACM ISBN 979-8-4007-0982-1/24/09
https://doi.org/10.1145/3688671.3688736

entities and their types, it requires labeled data for accurate evaluation.

- The Random Forest model is a supervised machine learning model that uses an ensemble of decision trees to classify text data. Features such as text length and the presence of specific tokens are used. The model is robust and less prone to overfitting, but may require a large amount of data to perform well.
- Gradient Boosting Classifier for Text Classification: The Gradient Boosting model is another supervised machine learning model that builds an ensemble of weak learners, typically decision trees, to create a strong classifier, focusing on reducing errors from previous models. It uses features similar to those in the Random Forest model.
- CNN-based Model for Text Classification: The Convolutional Neural Network (CNN) model is a deep learning model that uses convolutional layers to capture patterns in the text data. The model is trained to classify text based on features extracted from the seriesDescription.

2.2 A safe and efficient storage scheme based on SDG data

2.2.1 Similarity Assessment. We quantify the semantic textual similarity of each probe phrase, that is a candidate entrant-indicator, with existent SDG indicators. In this direction, we compute semantic textual similarity as calculated in Sentence-BERT (SBERT) [4], an approach that extracts and compares semantically meaningful sentence embeddings.

- Each word in the $sentence_i$ is preprocessed (e.g., tokenized).
- Each processed word in the sentence is encoded into vectors v_{ij} of 300 dimensions. Word2vec encodes similar words closer to each other in the vector space.
- To derive vector representation for $sentence_i$: average of such v_{ij} vector representations for $j = \{i, w\}$ is computed, where w is the number of words in the sentence.
- Sentence embeddings for all existent indicators in the SDG taxonomy are precalculated. They are assumed to be close in the 300 dimensional vector space if they are similar.
- Thus, computing cosine similary between the (300 dim) vector representation provides ideal score of 1, if the sentences are identical and score of 0, if the sentences are maximally dissimilar to each other.

Hilbert approximations for multidimensional data result in more efficient maintenance of local features as opposed to that achieved by linear ordering [2]. The next order Hilbert Space Filling (HSFC) curve comprises of four gyrated reiterations of the previous order curve. In the next repetition, quadrants are split up into four subquadrants each and so on. The line is repetitively folded in such a way that passes by successive neighboring points without intersecting itself and with infinite iterations of the curve construction algorithm it will not omit any point on a continuous plane. HSFCs are always bounded by the unit square, with Euclidean length exponentially growing with τ. Continuity of the curve ensures that affinity of bins on the unit interval signifies affinity in the unit square as well. Two points (x_1, y_1) and (x_2, y_2) with affinity in HSFC of order τ_1 depict affinity in HSFC of order $\tau_2 > \tau_1$ as well.

Algorithm 1: Algorithm for safe filtering of similarity search results among SDG data

Input: *app_id*, k, query q, distances $l_{\delta H}$, δ_H
Output: result_set S
Parameters: *indicator*, T=(x,y)
$\in$ N, Hilbert_Space_Filling_Curve *HSFC*

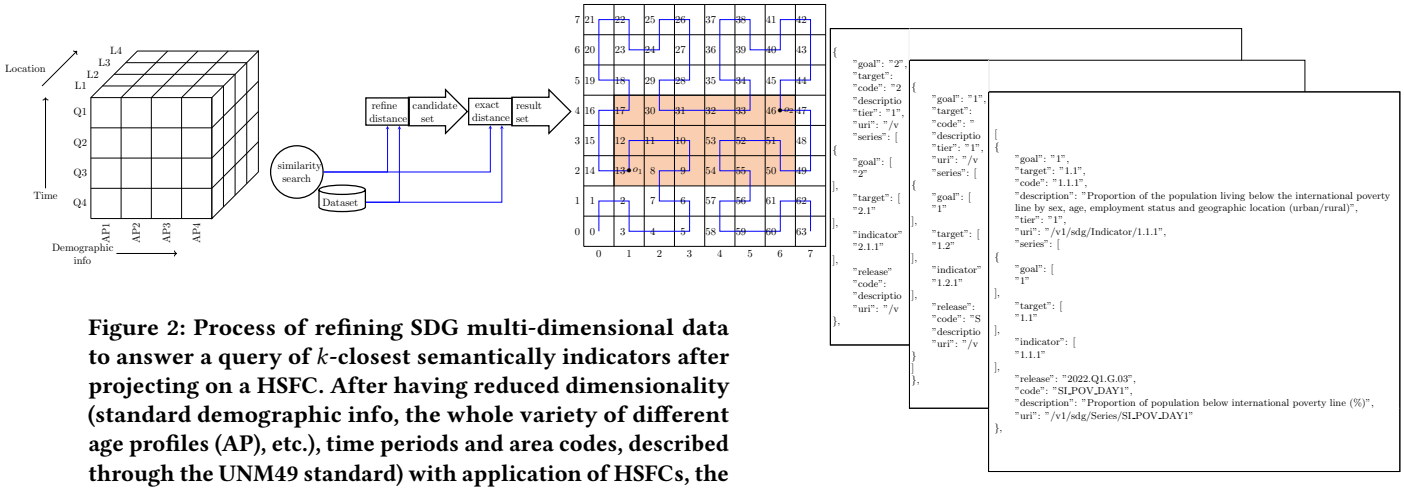
$S \leftarrow \emptyset$
$R_H \leftarrow$ranking(q, l_{δ_H})
$\epsilon \leftarrow$next value $\in R_H$
while $l_{\delta H}(q, \epsilon) \leq max_{\alpha \in S} \delta_H(q, \alpha)$ **do**
  if $|S| < k$ **then**
    $S \leftarrow S \cup \epsilon$
  else
    if $\delta_H(q, \epsilon) \leq \max_{\alpha \in S} \delta_H(q, \alpha)$ **then**
      $S \leftarrow S \cup \epsilon$
      $S \leftarrow S - \text{argmax}_{\alpha \in S} \delta_H(q, \alpha)$
    end if
  end if
  $\epsilon \leftarrow$next value $\in R_H$
  end
return S

Figure 1: Algorithm for safe filtering of similarity search results among SDG data

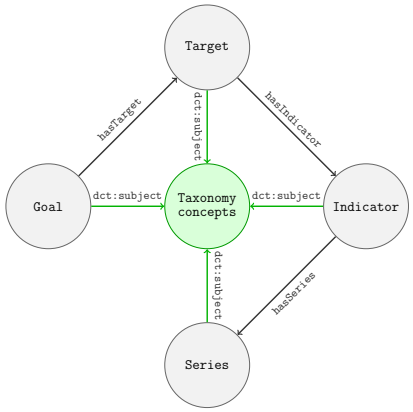
The first layer of the suggested distributed knowledge graph store will entail semantic representation of data. In the next layer, which acts as a substrate of the network topology, we split up the indexing area in semantically homogeneous areas through dimensionality reducing Hilbert Space Filling Curves (HSFC). Use of curves in this building block proves beneficial for preserving the neighbourhood property of concepts expressed by the indicators of an SDG target, as semantically related terms, more probable to respond to a user query, will be placed in the vicinity. In our suggestion linearization is implemented as an overlay upon existing two-dimensional search structures and the distributed file system, that ensures distribution and sharding that scale. Multidimensional queries upon the distributed knowledge graph can be mapped to two-dimensional queries, that range from the minimum to maximum linearization points of the initial query.

Retrieval of SDG data. The algorithm for matching k-semantically closest indicators is based on multi-step filtering and refinement, that consecutively removes irrelevant results and narrows the candidate set (Fig. 1). In order to optimally calculate distances, we use the algorithm proposed in [5], that performs optimally as far as the number of distance calculations is concerned, and modify it for HSFC representation. We create a ranking by means of the lower bound for a distance function among HSFC projections. Reranking takes place provided that the lower bound does nor surpass the k^{th}-nearest neighbor distance and the results are updated with objects of smaller distances [3].

The process of refining multi-dimensional data to answer a query of k-closest semantically indicators after projecting on a HSFC is depicted in Fig. 2. After having reduced dimensionality with application of HSFCs, the query for semantically similar indicators for target data of SDGs can be handled as a nearest neighbor search and implemented with a multi-step filter-and-refine approach [5]



(a) a



(b) b

Figure 2: Process of refining SDG multi-dimensional data to answer a query of k-closest semantically indicators after projecting on a HSFC. After having reduced dimensionality (standard demographic info, the whole variety of different age profiles (AP), etc.), time periods and area codes, described through the UNM49 standard) with application of HSFCs, the query for semantically similar indicators can be handled as a nearest neighbor search and implemented with a multi-step filter-and-refine approach for target data of SDGs. Creating a lower bound with a simple distance function filters out initially irrelevant results, and in the next step evaluation of results returned at the previous stage takes place with the use of the original distance function.

Figure 3: (a) IndicatorData dataset indicative excerpts for indicators 1.1.1, 1.2.1 and 2.1.1 (b) SDG schema of the dataset depicting structure of the UN SDG ontology.

[10] in an efficient way. The main idea is to filter at a later stage results falsely retrieved at first stage. Creating a lower bound with a simple distance function filters out initially irrelevant results, and in the next step evaluation of results returned at the previous stage takes place with the use of the original distance function. There are multiple properties describing each observation (data entry) and their Statistical Data and Metadata eXchange (SDMX)-standardized code equivalents are also provided. Dimensions (standard demographic info, the whole variety of different age profiles, etc.), time periods and area codes, described through the UNM49 standard are available in the dataset for each indicator from 2000 onwards [3].

3 Case Study

3.1 Dataset

We harvested an SDG dataset [2] of 2,21M. entries in total that includes all dimensions (standard demographic info, the whole variety of different age profiles, etc.), time periods and area codes, described through the UNM49 standard available for each indicator from 2000 onwards. We used the API of UN Statistics Division [9] with a set of scripts written in TypeScript and ran in the node.js environment. Dataset was particularly focused on indicators and list of all available SDG indicators was our starting point in the API, providing all available indicators in a self-contained response. Within the indicator related datasets, we collected 3 core datasets, while others were mostly redundant data provided for different data access or interpretation. Our dataset IndicatorData includes 169 targets, 248 indicators (with 13 replicated under two/three different targets), as well as 663 data series for the quantification of the SDGs' accomplishment [6]. The dataset includes series information and goal - target hierarchy with overall 663 series across 248 indicators (Fig. 3). The number of data entries per each indicator is 4150 after removal of 20% of top and tail outliers. There are multiple properties describing each observation (data entry) and their Statistical Data and Metadata eXchange (SDMX)-standardized code equivalents are also provided.

3.2 Experimentation Setup

The input dataset is the SDGindicator dataset. metaData file defined the schema of the KGs, whereas it was trained on 80 percent of data. 20 percent was used to evaluate the training, hence the dataset was split into 80:20 ratio. The parameters for the models appear in the Table 1.

We evaluate our algorithm in an experimental distributed environment over a key-value store of SDG data, that we collected. We use multiple servers and Hypertext Preprocessor (PHP) clients as APIs to handle cached values in a scheme built on Memcached, an optimized distributed hash map-based mechanism. Placement

Table 1: AI Model for the KG construction - parameters

Model	Parameter	Value
NLP-based using NER	model	Spacy's pre-trained model en_core_web_sm
Random Forest	max_depth	none
	n_estimators	100
	min_samples_split	2
	min_samples_leaf	1
	max_features	sqrt
Gradient Boosting	learning_rate	0.1
	max_depth	none
	n_estimators	100
	min_samples_split	2
	min_samples_leaf	1
	max_depth	3
Convolutional Neural Network	layers	Embedding, Conv1D, MaxPooling1D, GlobalMaxPooling1D, Dense
	epochs	5
	batch_size	32
	optimizer	Adam
	loss_function	Binary Crossentropy
	max_words	5000
	max_seq_len	250
	embedding_dim	100

Table 2: Setup settings in an experimental distributed environment over a key-value store of SDG data

Parameter	*Value*
Dataset	2,21M. entries
Number of servers	3
Number of virtual nodes per physical node	1
Queries	SELECT for similarity search
HSFC dimensions	2
HSFC order	3
Memcached server chunk size	1MB
Memcached server page size	40

of data with HSFCs is compared to default placement scheme of the prototype distributed cache mechanism in terms of response time for the executed SELECT queries and in terms of disk I/O. Experimental setup settings are described in Table 2.

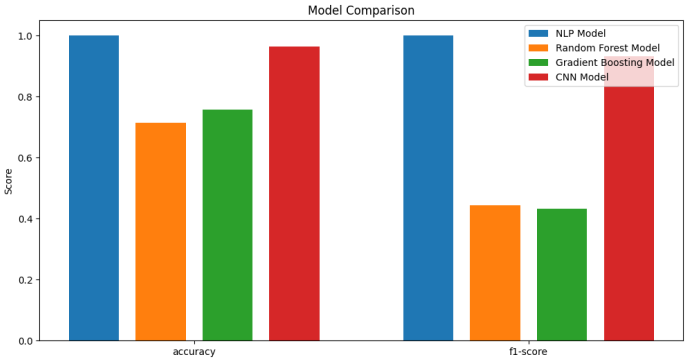


Figure 4: Comparative analysis of models

Model	Accuracy	F1-Score
NLP Model	1	1
Random Forest Model	0.685714	0.405405
Gradient Boosting Model	0.757143	0.433333
CNN Model	0.964286	0.933333

Table 3: Performance of AI models for the knowledge graph construction

4 Results and Discussion

- NLP-based Model using Named Entity Recognition (NER) depicts the best performance when it comes to identification of specific entities and their types in terms of accuracy and f1-score.
- The Random Forest model proves robust and less prone to overfitting, but requires a large amount of data to perform well. It underperforms in our case.
- Gradient Boosting Classifier is accurate and effective for structured data, although computationally demanding and can be sensitive to overfitting.
- CNN-based Model for Text Classification: The model is effective in capturing local dependencies and patterns in the text and depicts a very good performance in our case study, although it requires significant computational resources and a large amount of training data.

Visualization of produced knowledge graphs for each model is depicted in Fig. 5

We ran multiple sets of queries in an experimental distributed environment over a key-value store of SDG data with multiple servers and PHP clients as APIs to handle cached values in a scheme built on Memcached. After each set of queries the Memcached server was reset. We notice significant reduction in average response times for selection queries of combined indicators. Time difference between HSFC storage scheme and baseline distributed key-value store approach is more obvious in the case of disk I/O times (Global parameters used). There is also improvement in response times when HSFC mapping is loaded into Memcached keys directly, which is more obvious for combinations of sets of up to 4 indicators in our

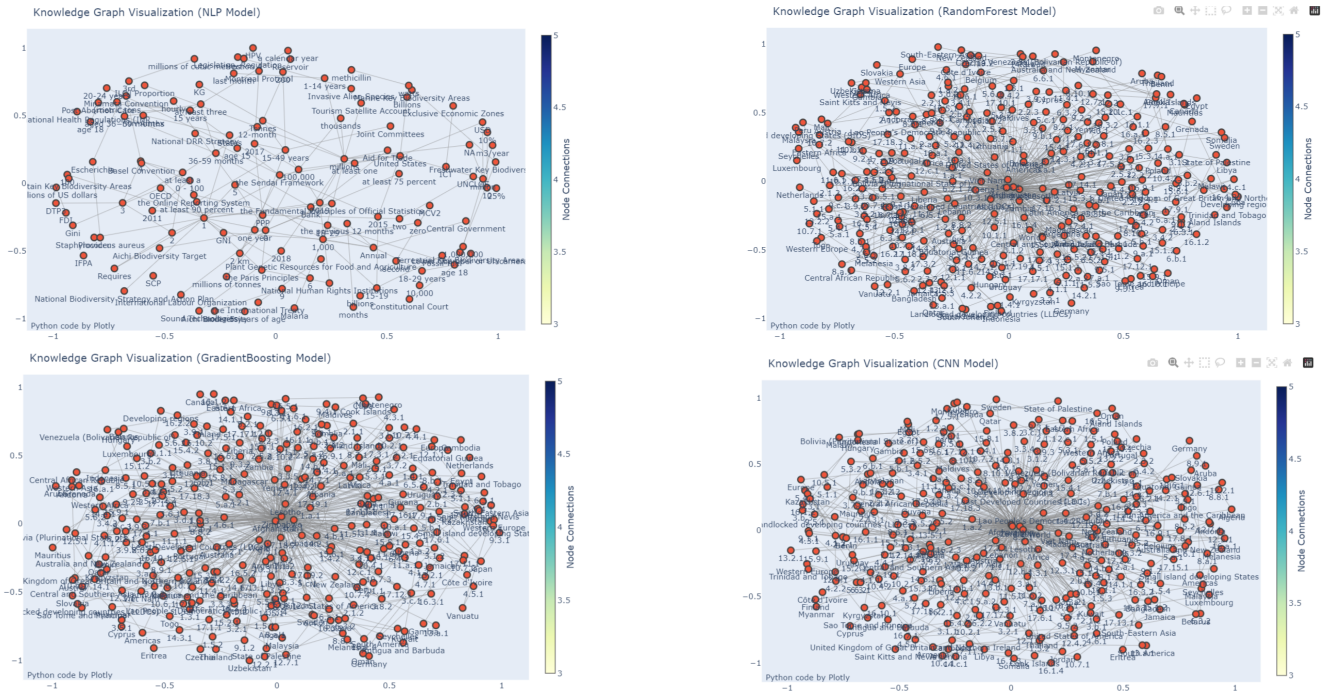
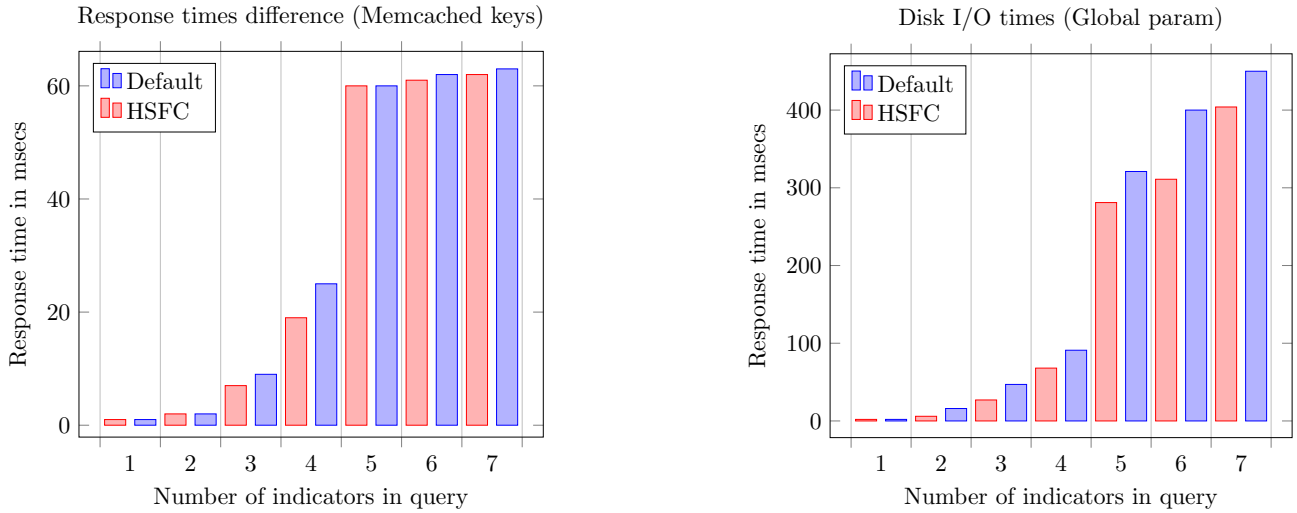


Figure 5: Knowledge graph visualization following various AI-based methods



(a) (b)

Figure 6: (a) Response time differences for HSFC mapping assigned to Memcached keys and (b) Disk I/O times for HSFC mapping passed as Globalparam. We notice significant reduction in average response times for selection queries of combined indicators. Time difference between HSFC storage scheme and baseline distributed key-value store approach is more obvious in the case of disk I/O times (Global parameters used). There is also improvement in response times when HSFC mapping is loaded into Memcached keys directly, which is more obvious for combinations of sets of up to 4 indicators in our setup.

setup (Fig. 6). The improvement in terms of memory response times can be further increased with further paging configuration, due to the nature of Memcached custom memory manager (slabs hold objects within specific ranges and slabs contain pages, split up in chunks) and the fact that a single indicator's entries reach up to 20MBs in our detailed dataset.

The practical impact of our work is that data retrieval times are reduced for semantically close data, that have not been categorised according to the prevailing SDG schema. Our approach empowers SDG knowledge graphs for causal analysis, inference, and manifold interpretations of the societal implications of SDG-related actions, as data are accessed in reduced retrieval times. The framework facilitates quicker measurement of influence of users and communities on specific goals and serves for faster distributed knowledge matching, as semantic cohesion is preserved.

Specifically, the suggested framework's impact on actions of organizations that harvest and process SDG datatakes is based on the consideration that the pursued goals and undertaken actions can be thematically interweaved and mutually influenced: one agent may pool interrelated existing content from federated repositories or one undertaken action may affect currently ongoing activities of other agents. Knowledge graphs are augmented with quicker similarity search, that reduces response times for manifold interpretations of the societal implications of actions that apply to each SDG goal, for association of contributions of actions to concrete SDG targets and quantification of the exerted influence. Concerning SDGs, knowledge graphs can quicker: i) support explainable decisions and insightful recommendations, ii) measure the influence of users and communities and iii) improve the user experience, as they facilitate extraction and organization of knowledge in a distributed manner and serve for quicker distributed knowledge matching.

5 Conclusions

Our approach empowers SDG knowledge graphs for causal analysis, inference, and manifold interpretations of the societal implications of SDG-related actions, as knowledge graphs are reliably constructed with AI and SDG data is accessed in reduced retrieval times. Our framework facilitates quicker measurement of influence of users and communities on specific goals and serves for faster distributed knowledge matching, as semantic cohesion of data is preserved. Our work aspires to support the collective effort to optimally harmonize sustainability goals, by proving useful to practitioners gathering and assessing SDG data.

References

[1] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A survey on knowledge graphs: representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems* 33, 2 (2022), 494–514. https://doi.org/10.1109/TNNLS.2021.3070843

[2] Irene Kilanioti and G. A. Papadopoulos. 2022. Best paper award.. An efficient storage scheme for Sustainable Development Goals data over distributed knowledge graph stores.. In *Proc. of 16th IEEE International Conference on Knowledge Graph (ICKG) '22*. Orlando, FL, USA.

[3] Irene Kilanioti and George A. Papadopoulos. 2023. A knowledge graph-based deep learning framework for efficient content similarity search of Sustainable Development Goals data. *Data Intelligence* (06 2023), 1–19. https://doi.org/10.1162/dint_a_00206 arXiv:https://direct.mit.edu/dint/article-pdf/doi/10.1162/dint_a_00206/2127019/dint_a_00206.pdf

[4] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[5] Thomas Seidl and Hans-Peter Kriegel. 1998. Optimal multi-step k-nearest neighbor search. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*. 154–165.

[6] UN. 2022. Global Indicator Framework after 2022 refinement - Global indicator framework for the Sustainable Development Goals and targets of the 2030 Agenda for Sustainable Development. https://unstats.un.org/sdgs/indicators/indicators-list/".

[7] UN. 2022. Linked SGD. https://linkedsdg.officialstatistics.org/.

[8] UN. 2022. SDG Taxonomy. http://metadata.un.org/sdg/.

[9] UN. 2022. SDGAPI. https://unstats.un.org/SDGAPI/swagger/.

[10] Cui Yu. 2002. *High-dimensional indexing: transformational approaches to high-dimensional range and similarity searches.* Springer.