IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 November 2023, accepted 4 December 2023, date of publication 7 December 2023,
date of current version 12 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3340984

RESEARCH ARTICLE

Instance-Based Lossless Summarization of Knowledge Graph With Optimized Triples and Corrections (IBA-OTC)

HAFIZ TAYYEB JAVED[1], KIFAYAT ULLAH KHAN[1,2], MUHAMMAD FAISAL CHEEMA[1], ASAAD ALGARNI[3], AND JEONGMIN PARK[4]

[1]School of Computing, FAST National University of Computer and Emerging Science (FAST-NUCES), Islamabad 44000, Pakistan
[2]College of Accountancy, Finance and Economics, Birmingham City Business School, Birmingham City University, B4 7BD Birmingham, U.K.
[3]Department of Computer Sciences, Faculty of Computing and Information Technology, Northern Border University, Rafha 91911, Saudi Arabia
[4]Department of Computer Engineering, Tech University of Korea, Siheung-si, Gyeonggi-do 15073, South Korea

Corresponding author: Jeongmin Park (jmpark@tukorea.ac.kr)

This work was supported by a grant of the Basic Science Research Program through the National Research Foundation (NRF) (2021R1F1A1063634) funded by the Ministry of Science and ICT(MSIT), Republic of Korea.

ABSTRACT Knowledge graph (KG) summarization facilitates efficient information retrieval for exploring complex structural data. For fast information retrieval, it requires processing on redundant data. However, it necessitates the completion of information in a summary graph. It also saves computational time during data retrieval, storage space, in-memory visualization, and preserving structure after summarization. State-of-the-art approaches summarize a given *KG* by preserving its structure at the cost of information loss. Additionally, the approaches not preserving the underlying structure, compromise the summarization ratio by focusing only on the compression of specific regions. In this way, these approaches either miss preserving the original facts or the wrong prediction of inferred information. To solve these problems, we present a novel framework for generating a lossless summary by preserving the structure through super signatures and their corresponding corrections. The proposed approach summarizes only the naturally overlapped instances while maintaining its information and preserving the underlying Resource Description Framework *RDF* graph. The resultant summary is composed of triples with positive, negative, and star corrections that are optimized by the smart calling of two novel functions namely *merge* and *disperse*. To evaluate the effectiveness of our proposed approach, we perform experiments on nine publicly available real-world knowledge graphs and obtain a better summarization ratio than state-of-the-art approaches by a margin of 10% to 30% with achieving its completeness, correctness, and compactness. In this way, the retrieval of common events and groups by queries is accelerated in the resultant graph.

INDEX TERMS Knowledge graph, semantic web, instance-based aggregation, super signature, optimized triples, optimized corrections.

I. INTRODUCTION

Knowledge Graphs, (KG), are heterogeneous and complex structures that represent various types of data and their relationships [6], [7]. Due to their heterogeneity and complexity, it is difficult for humans or even machines to perceive or even visualize them for analysis [8]. Such analysis requires data processing for retrieval, and correct inference for knowledge prediction. To minimize processing overhead, a summary graph is beneficial for aggregating concepts but every instance of it has own significance while aggregation. One such example is COVID-19 data [18] recently the world faced for predicting the next disease when a new instance added in the system. Thus, an aggregation of such concepts, requires its instance-based aggregation during summarization for correct inference. It also helps for future prediction of

The associate editor coordinating the review of this manuscript and approving it for publication was Alberto Cano.

those events that are dependent on newly joined concepts in KG. From aggregation with completion of summary graph, we are able to provide ease in visualization, extraction, fast in-memory processing, and even correct knowledge inference by saving computational time in comparison to KG.

In KG mining, several techniques have been developed for summarization. These techniques include bi-simulation [4], [7], [20], [32], co-occurrence [48], [50], and semantic similarity aggregation [44], [46], [47]. However, due to significance labels on the edges in KG, makes the summary graph lossy and inaccurate [1], [2]. As a result, a summary graph generates lossy compression as discussed in several studies [4], [20], [44], [46], [48], [50]. One technique [48] attempted to minimize the information loss using the in-lining mechanism for the resultant RDF graph but its prime concern was to preserve structure only, rather than preserving information and knowledge.

Recent studies, such as the ones present in [57], use semantic summarization as graph-based summarization (GBS) and query-based summarization (QBS) as a lossless compression. However, these approaches still aggregate individual events that require unique literal information in response to query retrieval in SPARQL for individual events. Thus, the composition of literals may be an accuracy tradeoff for lossy dependent data in case of no corrections. However, these approaches reduce the overall size of *KG*, they may lose some essential information for structural or data dependency [48], [50], [57]. Thus, a summary mechanism for KG needs to preserve information for correct knowledge inference while preserving its structure similar to its native one for its lossless aggregation. Moreover, it requires a reduction in size, and the presence of property existence while maintaining structural homomorphism between input and resultant graph. Such challenges are key challenges for summarizing KG. As a consequence, if the resultant summary preserves complete information in patches or levels, it leads KG to ensure its size reduction in each level with its completion of information and knowledge. In that case, it accelerates the SPARQL query for processing due to less number of triples during information preserving.

For lossless compression of KG while preserving complete information, it is necessary to identify natural overlapping events in a *KG*, these events facilitate aggregation of specific regions of the graph with minimum corrections. These corrections are: 1) rectification (negative corrections) [56] produce by computing valid information in case of lossy facts, 2) (positive corrections) consider when lossy facts are more than original facts, or 3) treats the same as input graph in case of disjoint events (star corrections) in a summary graph. In this way, the resultant graph ensures the completion of data for its dependency and similarity of structural homomorphism. In addition to it, it ensures each individual level in a multilevel graph which is lossless and guarantees its information about accuracy in a compact way.

In this paper, we present a novel lossless summarization approach of KGs, named Instance-based Aggregation with optimized Corrections and triples (IBA-OTC), which first identifies specific overlapping regions for aggregation. Secondly, different overlapping regions create multiple signatures for instance-based aggregation and ensure complete information preservation by maintaining the correction list. In this way, the proposed approach also reflects how the new information/knowledge is part of the resultant repository in case of aggregation. We demonstrate our proposed approach in Fig. (1) using a sample KG where we first generate 1). instance-based aggregation, then 2). optimize corrections and 3). triple optimization using two novel functions, *disperse* and *merge*. We perform experiments on 9 publicly available real-world KGs to obtain encouraging results. We also compare our proposed IBA-OTC with existing work in terms of 1). execution time and 2). compression ratio and obtain better performance.

The contributions of this paper are as follows.

- A mechanism to trace valid correction lists for each instance for its lossless aggregation of events.
- Mapping of natural merging events by finding specific overlapping regions of *KG* for its next upcoming event.
- Introduce ''merge'' and ''disperse'' for triple optimization in the resultant graph.
- Comparison with the state-of-the-art approaches for detailed analysis of summarized resultant graphs.

The structure of the paper is as follows. The background and related work will be discussed in section (II). In section (III), we present our problem definition and elaborate our objective on resultant summarized graph. Section (IV) presents our methodology and explains our approach using an illustrative example, along with a detailed explanation of the significance of each part of our three presented algorithms. Section (V) includes the validity proofs for our approach, which are demonstrated through three different properties. In section (VI), we conduct several experiments to show the summary behavior of our approach on various online available datasets. Finally, we conclude our paper in section (VII) and discuss the limitations and potential extensions of our approach in section (VIII).

II. RELATED WORK

In this section, we first discuss RDF graph summarization and categorize the existing work into four different classes. After providing an overview of these classes, we discuss some state-of-the-art approaches in detail as they are quite related to our work. We discuss these approaches in detail and compare our own approach with them.

The first class of work, focuses on preserving structure while doing summarization. The second class focuses on finding approximate patterns in knowledge graphs. The third class contains approaches for statistical metrics of KG and the last class contains hybrid methodologies for it.

A. STRUCTURAL METHODS

The approaches belonging to this class of work, concerns the preservation of graph structure for its summarization.

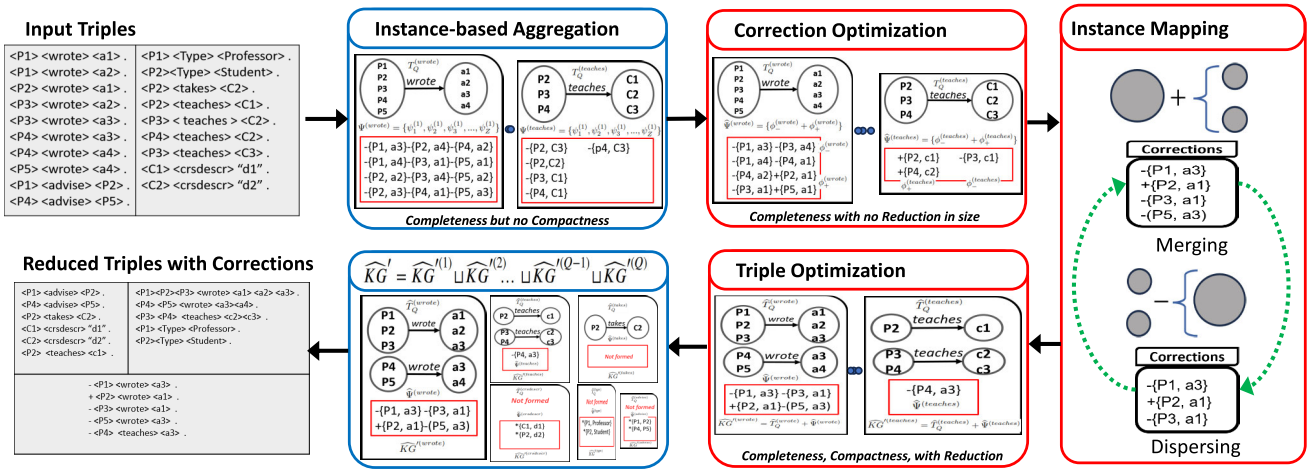

FIGURE 1. System diagram of the proposed solution. The steps shown in red color involve in the process of optimization. The green color represents the iterations.

Such methods [4], [7], [7] aim to structure visualization of RDF graphs. However, the resultant graph is not suitable for querying the information. References [15] and [17] proposed an approach to summarize large semantics graphs using namespaces. For this, the author used namespaces to create summary graphs of reduced size, for the sake of more meaningful visualization. In this summarization process, object literals are also reduced to their data type and the blank nodes. The main limitation of this approach likewise discussed in [29] is to visualize summary graphs to give an insight into the original large graph. Preserving information is still missing in this approach which we classified as a lossy data approach.

Recent studies presented SumMER [19], the first structural summarization technique using machine learning techniques for KGs. SumMER explores eight centrality measures and then uses machine learning techniques for optimally select most important nodes. After that, those nodes are linked to formulate a subgraph out from the original graph. In this approach, data completeness is still missing and it focuses only on visualizing in addition to the subgraph retrieval of the most significant nodes. Therefore, it is a lossy data compression. References [48] and [87] formally claimed another fine effort for summarizing RDF graphs by preserving structure only using the concept of "finding node level Co-Occurrence". Such effort for node merging criteria causes a reduction in size significantly with maintaining its property uniqueness but the main limitation of this work is a lossy compression keeping in view the information and knowledge in KG. Moreover, it provides an abstract summary for visualizing the RDF graph by preserving its structure. This summary is unable to answer several SPARQL queries likewise from the original graph. It uses intuition of transitive property for the inference of nodes merging. However, inlining mechanism also miss-preserves the information and inference in resultant KG. We question and solve this in our approach i.e. How to preserve structure and information while summarization.

B. PATTERN MINING METHODS

This line of works reveals frequent information or information regarding a similar entity for its summarizing task. In [5], [11], [12], [13], and [14], a framework is presented to model process data as a summary graph to discover concept hierarchies for entities based on both data objects and their interactions in summary graphs. For this, the authors presented a new language namely BP-SPARQL, for the explorative querying and understanding of summary graphs from various user perspectives. We understand that its architecture for querying, exploration, and analysis of process graphs is an extra overhead because resultant summary graph is not capable for querying traditional SPARQL language. More relevant to the structure and pattern, the authors in [16] presented a summarization method that takes into account both the graph structure and user query history. Specifically, the authors defined a mixed metric of node importance that captures both the structural importance and user query preferences. We categorize this method as a pattern mining summarization method because of finding individual node importance in KG. In this paper, two algorithms are proposed to extract summaries of a given RDF graph. The gap of lossless summarization still exists because query preferences lack some information in this summary graph. The authors in [22] extended the previous study, to speed up the evaluation of potentially interesting aggregates. Results show that their solution achieves significant execution time reductions while presenting output to close to the original. This approach is lossy and focuses only on interesting aggregated events.

Reference [31] proposed a novel summarization technique based on first-order logic rules. They formalized the problem to explain how the rules replace triples. Basically, it is a

top-down rule mining method to maximize the re-usability of cached results. Idea of their approach is subjective to us but the data dependent application still misses information of triples when replaced with rules. In similar lines, [3], [20], [32] presented to find Bi-Similarity among the nodes as node merging criteria which was further staggered in [15] and [87] for information mining. Its focus is to reduce the size by ignoring its structure which causes complexity increase in the graph. Therefore, the resultant graph is unable to retrieve information as original.

C. STATISTICAL METHODS

The aim of this class of work is to explore various qualitative measures for the statistical analysis of the graphs. References [34], [36], and [42] used the concept of bi-simulation that are applicable to different sub-graphs defined by different queries. Sometimes, an actual RDF graph is critical with the use of mixed and different vocabularies. Therefore, another scheme of study presented in [23], [24], [25], and [26] is to facilitate mainly the understanding of knowledge graphs with mixed vocabularies and not for retrieval or inference from its compact variant. However, [42] presented an idea of identifying super nodes in KG to reduce the data complexity. For this concern, it finds the most highlighted concepts and identifies the important nodes in a KG. This idea fascinates even for correct inference of knowledge but still lossy compression cannot be tolerated in linked data retrieval.

The authors in [38], [39], [40], [41], and [43] presented an approach for dynamic exploration with two novel concepts. One is responsible for granular information access while the other is for specifying several query nodes in KG. Notably, [43] is a sequel of [42] for identifying super nodes. Similar to it, a pioneering work already presented in [44] to analyze an approach based on knowledge pattern visualization, also supports query patterns even on unknown vocabularies. This approach is again lossy and lacks the correct inference that we are addressing in our approach. However, [44] explains the features that are important for the ontological point of KG. In addition, concept and relation ranking (CARRank) optimizes the weights of relations like a PageRank algorithm. One reason for summarization is the complex nature of linked data due to its diversity. Therefore, [46] presents this idea to formulate SPARQL queries across multiple heterogeneous data sources regardless of known or unknown to the user. The sole contribution of this work is to discuss queries and not for summary generation. Further, semantic summaries are understandable with the help of schema relations. In consequence, [49] presents RDF digest, a notion for automatically generating the schema and visualization of RDF graphs. This work presents two algorithms that uses both structure and meaning of the linked repository translated in RDF/RDFS format as summary graph. For that reason, it ignores missing information of RDF data which is not beneficial for lossless summarization.

D. HYBRID METHODS

In this class of work, we present a wide range of different RDF summarization techniques. As mentioned above, the recent contribution regarding RDF graph summarization lacks research on some key points like fast information retrieval, completion, and correctness from the resultant summary graph. Therefore, schema summary, structural refinement, and pattern identification does not cover these points. Below, we critique and discuss in detail some hybrid methods for summarization.

Reference [1] presents a survey of several summarization concepts with technical aspects and implementation. Its main aim is to enlighten several summarization methods for various usage scenarios. Although, an approach leading to lossless summarization, is still missing in this literature that's why [3] concentrates another survey on entity summarization literature on the previous state-of-the-art approaches. The main focus of this research is to present the first comprehensive survey of entity summarization in comparison to separately reviewing all methods.

Bi-simulation, a concept discussed earlier, compresses only the query part of a KG to give a notion in [34], [52], [53], [54], and [55] as an adaptive structure summary graph (ASSG). In relevance to graph size reduction, ASSG commits a resultant graph to ensure less number of nodes and edges with its adaptive ability to adjust its structure in connection to graph query. ASSG also commits lossy compression that has an impact on specific resultant queries. Moreover, a summary tool for visualization, LODSight, presented in [35] displays a typical combination of predicates and their types in a similar way. Due to certain reliability issues, the summary graph needs further work for its improvements. Therefore, [36], [65], [66], [67], [68] introduces a special variant of the database for answering basic graph patterns for better understanding of knowledge graph semantics. In particular, the focus of this research is to deploy the RDBMS engine which is still a trade-off to completely avoid preservation by generating its signature schema of the original graph. Moreover, [37] also critiques understanding datasets and the issue of understandability using the tool EXPLOD to facilitate the exploration of knowledge graph summaries among interlinked datasets.

Reference [7] presents the use of a structure index for the RDF graph. This is a structure-oriented approach for RDF data partitioning and query processing which lacks the information for summary generation of RDF graphs. In addition, [32] introduces two database styles for generating summary graphs. Although the dataset is not a linked data, their task specific to summarization shows it is relevant in KG summarization. The main limitation of this approach is generating one big homomorphic structural graph. Thus, our approach uses a similar idea for the identification of specific regions and multiple homomorphic summary graphs. Another study [20], investigates such approaches as summarizing big graphs. Reference [15]

is a recent contribution of summarizing large knowledge graphs using namespaces. Similar to the previous, Another recent contribution of compression is discussed in [27] and [89]. These approaches mainly focus on visualizing the graph, and not query retrieval. Visualizing such graphs only aims to give insight into the details of structure only in an abstract way. Further, [33], [79], [80], [81], [82] explains graph materializing to summarize reasoning from knowledge graphs. The idea behind the abstraction is based on equivalence classes with their refinements. This idea behind refinements is similar to our corrections but the resultant graph for this approach misses meta information of the classes that we consider in our approach. For lossless compression, an approach regarding the knowledge graph presented in [47] as rule-based compression (RB Compression), compresses the information by introducing new nodes and removes verbose triples in the knowledge base. Removing such triples may be beneficial but it requires the identification of missing information. In [48], [56], [93], [94], and [95], a KG summarization method is presented using the predicate-based Co-Occurrence as a hybrid model. The intuition behind this is to represent a KG in an entity-relationship (ER) style which is easy to comprehend and understand by the human. Although this model is a first fine effort regarding KG summarization and it misses the retrieval requests of several literal concepts. Therefore, this approach is a lossy data compression. Another recent effort [57] provides semantic summary of predicates especially grouped by queries but it is also lossy due to aggregation of literal information of *KG*.

Our approach is different from the existing studies and especially the aforementioned two recent approaches because we maintain a correction list (CL) with its three variants 1). *Positive*$(+ev)$, 2). *Negitive*$(-ev)$, and 3). *Star*$(*)$ to ensure completeness, and correctness in a compact way. We also compare our approach by considering the metrics like method of use, input graph type, and by focusing the problems of completeness, correctness, and accuracy. We present detailed comparison of existing studies in Table. (1). We compare our approach with state-of-the-art methodologies and consider our work as a first effort regarding RDF data aggregation with lossless information.

III. PROBLEM DEFINITION

Given a knowledge graph (KG) in triples, our objective is to create its lossless summary graph ($\widehat{KG}'$) to achieve 1) Compactness, 2) Correctness, and 3) Completeness. In this way, the resultant $\widehat{KG}'$ is a union composition of J super signatures ($\widehat{T_Q}$) where for every super signature $T_Q^{(j)} \in \widehat{T_Q}$ i.e $\widehat{T_Q} = T_Q^{(1)}, T_Q^{(2)}, \ldots, T_Q^{(J)}$ and their D corresponding corrections ($\widehat{\Phi}$) i.e $\widehat{\Phi} = \{\widehat{\Psi}^{(1)}, \widehat{\Psi}^{(2)}, \widehat{\Psi}^{(3)}, \ldots, \widehat{\Psi}^{(D)}\}$. Our task is to minimize each correction list $\widehat{\Psi}^{(d)}$ that generates their corresponding super signature by identifying only overlapping regions in KG.

IV. PROPOSED METHODOLOGY

We categorize our methodology into three modules. As a first step, we present instance-based aggregation approach for finding the specific overlapping regions of a KG. We then present a solution to compute corrections and minimize the correction list by performing correction optimization. Lastly, we ensure the reduction of triples in a resultant graph using triple optimization. We propose two novel functions *merge* and *disperse* for triple optimization. The detail of each module is explained below.

A. KNOWLEDGE GRAPH, A FORMAL DESCRIPTION

First, we describe the notations and symbols to represent a KG consisting of H number of facts. A fact may be a subject, an object or a literal. A set of subjects is represented as $S = \{s_1, s_2, s_3, \ldots, s_I\}$ which contains I number of subjects/objects. A set of N number of objects/literals is expressed as $S' = \{s'_1, s'_2, s'_3, \ldots, s'_N\}$ and a set of M number of literals is represented as $S'' = \{s''_1, s''_2, s''_3, , \ldots, s''_M\}$. A predicate set is represented as $P = \{p_1, p_2, p_3, , \ldots, p_J\}$. A single triple is (s_i, p_j, ω), where there are K number of ω and $\{\omega : \omega \in S' \quad \text{or} \quad \omega \in S''\}$. The entire triple resources of the input knowledge graph can be expressed as,

$$KG = \bigcup_{s_i \in S, p_j \in P, \omega} (s_i, p_j, \omega), \quad \{\omega : \omega \in S' \quad \text{or} \quad \omega \in S''\} \tag{1}$$

B. INSTANCE-BASED AGGREGATION

Our first step towards lossless summarization is the instance-based aggregation. We generate super signatures, based on each predicate separately.

1) LOCALITY SENSITIVE HASHING (LSH) BASED AGGREGATIONS

In our instance-based summarization of a KG, rather than performing pairwise similarity computations to obtain signatures, we adopt LSH that can identify similar triples with high accuracy. Similar, to the idea of [10] to group the nodes for generating the summary of an undirected graph. Using LSH, we generate instance-based signatures using hash codes matching from the neighborhood of each node. In this way, signatures with matching hash codes are grouped with each other. Thus, restricting similarity computations between similar nodes only, saving our computational cost in comparison to group-based aggregation like [57]. We apply LSH on a KG for (i) creating a minhash signature matrix and (ii) generating signatures of similar nodes. For a heterogeneous knowledge graph, KG, having j interacting with H different facts, contains subjects (S), subjects/objects (S') and objects/literals (S''). Both S and S' are mutually disjoint to S'' which is a superset of both sets. Therefore, the notation of single triple in KG is actually the LSH based super signature which is represented as:

$$T_Q^{(1)} = \mathsf{S}, p_1, \mathsf{W}, \tag{2}$$

TABLE 1. Comparison of existing approaches.

Methods	Type	Input	Addressed Problem	Complete	Correct	Accurate	Technique
[5]	Lossy	Process SPARQl Data	Query based Summary	✓	✗	✓	Summarized Process Data
[15]	Lossy	Graph	Graph based Summary	✗	✓	✓(Specific Queries)	Namespace Summarization
[16]	Lossless	Sub Graph	Query based Summary	✗	✓	✗	Node Significance Summarization
[19]	Lossy	Graph	Graph based Summary	✗	✓	✓(Specific Queries)	Structural Summarization
[22]	Lossy	Summary Graph	Summary for Training	✗	✗	✗	Training on RDF Summary
[24]	Lossless	Sub Graph	Query based Summary	✓	✓	✓	Summarization through Sampling
[33]	Lossy	Sub Graph	Query based Summary	✗	✓	✓	Entity Summarization
[54]	Lossy	RDF Graph	Visualization	✗	✓	✗	Structure Preserving
[63]	Lossy	RDF Graph	Graph Based Summary	✗	✓	Partially	Clustering same Predicates
[63]	Lossless	RDF Graph + Query	Query Based Summary	✓	✓	✓(Specified Queries)	Similar Predicates Retrieval
[69]	Lossy	RDF Graph	Resolving complexity	✗	✓	✓(Specified Queries)	Summarization for Query Optimization
[76]	Lossless	Graph	Graph based Summary	✗	✓	✓	Entity Summarization
[70]	Lossy	RDF + RDFS	K-Approximation Summary	✗	✓	✓(Specified Queries)	Efficiency in 'K' query Processing
[71]	Lossy	Sub Graph (RDF)	knowledge based Modeling	✗	✓	✓(Specified Queries)	Entity Summarization
[78]	Lossy	RDF Graph	Visualization	✗	✓	✓(Abstract)	Structure Preserving
[79]	Lossless	RDF Graph	Summarize Semantic Query	✓	✓	✓	Query based Summarization
Our Approach	Lossless	RDF Graph	Retrieval for all Queries	✓	✓	✓	Instance based Aggregation with Graph Correction (IBA-OC)
Our Approach	Lossless	RDF Graph	Retrieval for all Queries	✓	✓	✓	Instance based Aggregation Optimized Corrections (IBA-OT)

TABLE 2. Annotation table for all symbols used in proposed methodology.

Symbol	*Annotation*
KG	Knowledge Graph
s_i	An instance of Subject
p_j	An instance of Predicate
ω	An instance of Object
s	An instance of Super Subject
w	An instance of Super Object
T_Q	Super Signature
Φ	A naive correction list
$\psi_z^{(j)}$	z_{th} Correction for j_{th} predicate
Φ_-	Negative Correction List
Φ_+	Positive Correction List
Φ_*	Star Correction List
$\widehat{KG}$	Summarized knowledge graph after correction optimization
$\widehat{\Psi}$	Correction List after correction optimization
$\widehat{\Phi}$	Optimized Correction List
$\widehat{KG}'_m$	Instance of knowledge graph while merge
$\widehat{KG}'_d$	Instance of knowledge graph while disperse
t_r	Triple Ratio
t_a	Triple count in Actual graph
t_s	Triple count in Summarized graph
$\widehat{\Psi}_d$	Correction List after Triple Optimization
$\widehat{KG}'$	Resultant Summarized knowledge Graph

where s represents a super subject that may consist of multiple subjects while w is a super object that may also consist of a number of objects, *e.g.*, $\mathsf{s} = \{s_1, s_2, s_3\}$ and $\mathsf{w} = \{s'_1, s'_2, s'_3\}$. A super subject or super object is actually a super node consisting of a number of subjects or objects. In other words, we consider Q number of quotients in a KG, and the first two instances (a sub-graph), $KG^{(1)}$ and $KG^{(2)}$ are

$$KG^{(1)} = \bigcup_{s_i \in S, \omega} (s_i, p_1, \omega), \tag{3}$$

$$KG^{(2)} = \bigcup_{s_i \in S, \omega} (s_i, p_2, \omega), \tag{4}$$

For instance in an illustrative example, shown in Fig. (2), $KG^{(1)}$ is the instance representing "wrote" and $KG^{(2)}$ represents "teaches". All the quotients of knowledge graph are disjoint union with each other.

$$KG = KG^{(1)} \sqcup KG^{(2)} \sqcup KG^{(3)}, \ldots, \sqcup KG^{(Q-1)} \sqcup KG^{(Q)} \tag{5}$$

2) CORRECTION LIST (Φ).

We express a KG in (1). as,

$$KG = T_Q \cup \Phi \tag{6}$$

where T_Q is total number of super signatures.

$$T_Q = \{T_Q^{(1)}, T_Q^{(2)}, \ldots, T_Q^{(J)}\} \tag{7}$$

and Φ is the total number of correction lists

$$\Phi = \{\Psi^{(1)}, \Psi^{(2)}, \Psi^{(3)}, \ldots, \Psi^{(D)}\}. \tag{8}$$

For a specific predicate, the (6) becomes

$$KG^{(1)} = T_Q^{(1)} \cup \Psi^{(1)} \tag{9}$$

where $\Psi^{(1)}$ is the correction list for predicate p_1 and each correction in the correction list is represented as ψ. In this

way, a correction list may consist of Z number of corrections like, $\Psi^{(1)} = \{\psi_1^{(1)}, \psi_2^{(1)}, \psi_3^{(1)}, \ldots, \psi_Z^{(1)}\}$.

We introduce three different correction lists in the resultant knowledge graph $\widehat{KG}$. A correction list is a set of triples that is part of the super signature for a resultant summary graph, as shown in Fig. (1).

- Negative Correction List (Φ_-). In a negative correction list Φ_-, a negative correction ϕ_- indicates that a triple, (s_i, p_j, ω), is not part of KG with respect to its super signature, $T_Q = \mathtt{S}, p_1, \mathtt{W}$. More precisely, in a query retrieval process, a negative correction needs to be removed.
- Positive Correction List (Φ_+). In case of a positive correction list Φ_+, a positive correction ϕ_+ shows that a triple, (s_i, p_j, ω), is part of KG with respect to its super signature, $T_Q = \mathtt{S}, p_1, \mathtt{W}$. In a query retrieval process, a positive correction must need to be added.
- Star Correction List (Φ_*). In a star correction list Φ_*, a star correction ϕ_* points out the situation when both positive and negative corrections fail to produce a smaller number of triples compared to the actual ones. We represent these star corrections with the same as original triples in resultant summary graph.

The detailed process of our proposed instance-based aggregation and super node creation is presented in Algorithm 1. It requires a KG, in triples format having P different predicates and ensures all super signatures with the relevant negative corrections. Lines 1-4 ensures the graph contains triples with repeating predicates. Line 5-8 generates P different signatures using LSH, whereas line 9-12 generates its possible number of corrections for each super signature. The computational time for Instance-based aggregation is $kO(n)$ and it returns the super signatures $T_Q^{(J)}$, and naive corrections Ψ^-.

Algorithm 1 Instance-Based Aggregation and Super Node Creation

Require: A knowledge graph KG with a number of Triples $T = (s_i, p_j, \omega)$, with predicates $P = \{p_1, p_2, p_3, , \ldots, p_J\}$.
Ensure: The triples $T_Q^{(J)}$ and the correction list Ψ^-.
//editorialization
1: input a knowledge graph KG
2: **if** (count(T) ==($unique$(P))) **then**
3: return
4: **end if**
5: **for** $p_j \in P$ for triples T **do**
6: $T_Q^{(j)} \leftarrow LSH(s_i, p_j, \omega) \quad i \in \{1, 2, 3, \ldots, I\}$
7: $T_Q^{(j)} = \mathtt{S}, p_j, \mathtt{W}$
8: **end for**
9: **for** $p_j \in P$ for instance-based aggregated triples $T_Q^{(J)}$ **do**
10: **if** $(\forall s_i \in \mathtt{S}, p_j, \forall \omega_i \in \mathtt{W}) \notin KG$ **then**
11: $\Psi^- \leftarrow s_i, p_j, \omega_i$
12: **end if**
13: **end for**
14: **return** $T_Q^{(J)}, \Psi^-$

C. PROPOSED OPTIMIZATIONS FOR CORRECTIONS AND TRIPLES

The next step is the process of minimization which consists of two steps, 1). the corrections and the triple optimization.

1) CORRECTIONS OPTIMIZATION

In correction optimization, we deal with correction lists Φ and minimize the number of corrections in each individual correction list $\Psi^{(d)}$, $d \in \{1, 2, 3, \ldots D\}$. We reduce number of corrections in a correction list by introducing the different types of corrections types, as mentioned above. We, first, swap all those negative corrections with positive correction(s) for which the count of the positive corrections is lesser than count of the negative corrections. We then introduce a star correction which indicates that there are no benefits of aggregation at all. As a result, the original triples of the graph remain unchanged. In this way, the corrections are initially pure negative, which we distribute later in negative ϕ_-, positive ϕ_+ and star ϕ_* corrections. To this end, we have a mixture of corrections that ensures the correction minimization process for $\widehat{\Psi}^{(1)}$ for a specific predicate as

$$\widehat{\Psi}^{(1)} = \{\phi_-^{(1)} \cup \phi_+^{(1)} \cup \phi_*^{(1)}\} \tag{10}$$

After performing the correction optimization, the Equation 9 becomes,

$$\widehat{KG}^{(1)} = T_Q^{(1)} \cup \widehat{\Psi}^{(1)} \tag{11}$$

We update the correction list $\widehat{\Phi}$ for all available instances. As a result, the correction lists become

$$\widehat{\Phi} = \{\widehat{\Psi}^{(1)}, \widehat{\Psi}^{(2)}, \widehat{\Psi}^{(3)}, \ldots, \widehat{\Psi}^{(D)}\} \tag{12}$$

It is observed that the updated correction list is $\widehat{\Psi} < \Psi$. The discarded corrections are expressed as α. We present our approach to detail optimized correction list in Algorithm 2. It requires the output of proposed Algorithm 1. Its purpose is to replace negative corrections Ψ^- with positive corrections Ψ^+ that is beneficial for reduction and then update the correction list if the positive corrections of some events are lesser in number in comparison to its previous negative (-ve) corrections. Before the execution of this algorithm, each super signature has many Ψ^- corrections but after executing this algorithm, the correction list updates by removing α number of triples. In this way, updated events are Ψ^+ in the correction list. Note that, an event is either a part of any super signature either with positive corrections (+ve) or with negative corrections (-ve). Lines 1-3 generates the positive corrections for all of the super signatures in $O(n)$ time and updates by comparing the count of positive with negative. Lines 4-8 update the whole corrections list $O(k)$ times. We perform this using LSH to save our computational cost. Hence, the complete computational cost for checking

and updating is $O(kn)$. Lines 9-12 remove all invalid or duplicated corrections by updating KG in $O(k)$ times.

Algorithm 2 Correction Optimization

Require: $T_Q^{(J)}$, Ψ^-.
Ensure: The optimized corrections $\widehat{\Psi}^{(J)}$. //editorialization

```
for p_j ∈ P for instance-based aggregated triples T_Q^(J) do
  if (∀s_i ∈ S, p_j, ∀ω_i ∈ W) ∈ KG then
    Ψ+ ← s_i, p_j, ω_i
    for (∀s_i ∈ S) do
      if count(Ψ+) < count(Ψ−) then
        Ψ̂ ← Ψ+
      else
        Ψ̂ ← Ψ−
      end if
    end for
  end if
end for
return Ψ̂
```

2) TRIPLES OPTIMIZATION

In case of triple optimization, we perform the process of minimization by proposing two novel concepts of *merging* and *dispersing*. To this end, we have instance-based super signatures with all types of corrections, on which we perform *merging* or *dispersing*.

a: MERGING

In merging, we take a super signature and check the merging for every node in a super node, i.e. a super subject or super object. If recent merging of a node in a super node contains less or equal number of triples in summarized graph $\widehat{KG}^{\prime(1)}$ then the triples in an actual $KG^{(1)}$, with respect to that specific super signature, the process of merging is performed. More precisely, merging is beneficial only if nodes are combined together in such a way that the current count of triples is reduced in summary graph $\widehat{KG}^{\prime(1)}$ is less than or equal to the triples count in an actual graph $KG^{(1)}$ with respect to a specific super signature. It is to note that the total number of triples in the resultant sub-graph $KG^{(1)}$ contains its signatures and all corrections. Mathematically,

$$\widehat{KG}_m^{\prime(1)} = \widehat{T}_{Q,m}^{(1)} \cup \widehat{\Psi}^{(1)}. \tag{13}$$

b: DISPERSING

In case of dispersing, if current count of triples in a summarized graph $\widehat{KG}^{\prime(1)}$ becomes greater than the triples count in actual graph $KG^{(1)}$ from certain super signatures, merging process becomes failed and we perform dispersing instead. Mathematically,

$$\widehat{KG}_d^{\prime(1)} = \widehat{T}_{Q_d}^{(1)} \cup \widehat{\Psi_d}^{(1)}. \tag{14}$$

We perform merging and dispersing step by step triggered at once at a time based on the triple ratio, $t_r^{(1)} = \frac{t_s^{(1)}}{t_a^{(1)}}$, where $t_a^{(1)}$ is triple count of the actual sub-graph $KG^{(1)}$ and $t_s^{(1)}$ is triple count of the summarized sub-graph $\widehat{KG}^{\prime(1)}$ for some specific super signature. Thus, total number of discarded corrections is β. Mathematically,

$$\widehat{KG}^{\prime(1)} = \begin{cases} \widehat{KG}_m^{\prime(1)} & t_r^{(1)} \leq 1 \\ \widehat{KG}_d^{\prime(1)} & \text{otherwise} \end{cases} \tag{15}$$

Generally, we always perform merging first, if it fails then we try to disperse at a time. As a result, we get a new version of sub-knowledge graphs represented as $\widehat{KG}^{\prime(1)}$ with reduced number of triples

$$\widehat{KG}^{\prime(1)} = \widehat{KG}_m^{\prime(1)} \cup \widehat{KG}_d^{\prime(1)}. \tag{16}$$

It may also be expressed as,

$$\widehat{KG}^{\prime(1)} = \widehat{T}_Q^{(1)} \cup \widehat{\Psi}^{(1)}, \tag{17}$$

In the end, we obtain a summarized graph for all the instances as

$$\widehat{KG}' = \widehat{KG}^{\prime(1)} \sqcup \widehat{KG}^{\prime(2)} \sqcup \widehat{KG}^{\prime(3)} \ldots \sqcup \widehat{KG}^{\prime(Q-1)} \sqcup \widehat{KG}^{\prime(Q)} \tag{18}$$

The details of proposed mechanism about optimized triples is presented in Algorithm 3. It requires a KG and output of our Algorithm 2. It ensures the final resultant graph $\widehat{KG}'$ in a compact version. We call this as our cost function because it is an iterative process that finds the optimal region of the aggregated graph with minimum corrections. Line 1 indicates the outer loop for examining all super signatures and Line 2 indicates the inner loop to examine subjects and objects association of each super node in a KG. It performs the optimized assessment on each super signature with respect to its triple ratio t_r in an iterative manner. Lines 3-5 calculates and checks the triples ratio t_r. A super signature $T_Q^{(J)}$ splits into k multiple sub super signatures, if its triples ratio t_r exceeds one. Lines 6-10 perform merge and disperse function to create multi-super signatures with optimized correction list. We call this operation a dispersal of events that maps a new instance of KG by creating another super signature of the same predicate. Thus, it requires updating $\widehat{\Psi}$ by reducing β corrections.

D. EXAMPLE EXPLANATION OF THE PROPOSED ALGORITHMS USING A SAMPLE KNOWLEDGE GRAPH

In this section, we provide details of our proposed solution using a sample KG.

Given a KG, we summarize it in multi-instance KG_i where $i = 1, 2, 3, \ldots Q$. We aim to produce the summary instance of KG for property-specific SPARQL Query in addition to its refinements and preserving information in the summary graph. Here, $KG^{(1)}$ and $KG^{(2)}$ are,

$$Professor \xleftarrow{\text{type}} P_1 \xrightarrow{\text{advises}} P_2$$

The above scenario is sample sub-knowledge graph which contains two triples one depicting P_1 as Professor and the

Algorithm 3 Triples Optimization

Require: $KG, T_Q^{(J)}, \widehat{\Psi}$.

Ensure: The graph with optimized triples, $\widehat{KG}'_m$, $\widehat{KG}'_d$ and $\widehat{KG}'$.

```
for ∀p_j ∈ P in instance-based aggregated triples T_Q^(J) do
    for (∀s_i ∈ S, p_j, ∀ω_i ∈ W) ∈ K̂G do
        t_a = triple_count(KG)
        t_s = triple_count(K̂G')
        t_r = t_s / t_a
        if t_r ≤ 1 then
            K̂G'_m = merge(K̂G) //Single Super Signatures
        else
            K̂G'_d = disperse(K̂G) //Multiple Sub-Super Signatures
        end if
    end for
    K̂G' = K̂G'_m ∪ K̂G'_d
end for
K̂G' = K̂G'^(1) ⊔ K̂G'^(2) ⊔ K̂G'^(3) ... K̂G'^(Q)
return K̂G'
```

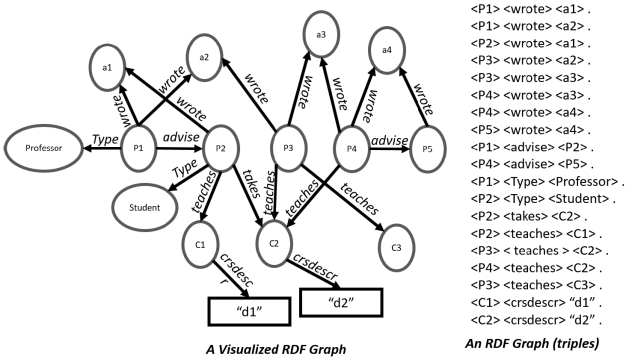

FIGURE 2. An example for illustration of heterogeneous graph.

other showing P_1 as a mentor of P_2. Therefore, two summary intakes are responsible for *type* and *advises* for this sub-graph separately, as show below.

$$KG^{(subgraph)} = KG^{(type)} \cup KG^{(advises)}$$

In the above, the triples of *type* and *advises* are independent so we need to preserve factful information about $'type'$ in KG_{type} and $'advises'$ in $KG_{advises}$ in two separate instances. Here, any instance of the graph is property-specific that responsible for its respective queries in $\widehat{KG}'$. Therefore, for every query: first, it retrieves from the property-specific instances, and second, it retrieves from the $\widehat{KG}'$.

Fig. (2) shows P_1, P_2, P_3, P_4, P_5 representing people who taught courses C_1, C_2, C_3 and wrote books a_1, a_2, a_3, and a_4, hence shows a heterogeneous KG. Visually, it is depicted in a graph format on the left while the right side represents its triple format. A triple is complete information of a subject interacting with an object with a relationship of its predicate. Our sample graph shows it contains 19 triples, 16 nodes and 19 triples.

In Fig. (3), $'wrote'$ is used for explaining the complete execution of our approach. Our proposed approach for corrections optimization, IBA-OC, facilitates query response for single super signatures. Similarly, proposed method for triples optimization, IBA-OT, responds for multiple sub-super signatures for its smart execution. Thus, a property 'wrote' retrieved from KG and producing KG_1 as its first instance. For this, SPARQL queries relevant to $KG^{(1)}$ save computational overhead by only identifying its instance. Moreover, any operation regarding aggregation also needs valid corrections in $\widehat{KG}'^{(1)}$ for its completeness, compactness, and correctness.

Below we present explanation of LSH-based aggregation, Correction optimization, and triples Optimization for mapping the natural merging events with minimum corrections. In below explanation, we summarize our sample KG having nineteen triples reduced into a summary having sixteen triples, as shown in Fig. (1). In this way, it ensures a lossless structure in a compact way in comparison to input KG.

1) APPLYING PROPOSED LSH-BASED AGGREGATIONS ON OUR SAMPLE KNOWLEDGE GRAPH

Our first step is the aggregation of triples having same predicates. For this, we use LSH for aggregation to efficiently identify the triples having same predicates. We aggregate the triples on the basis of *wrote*, *advises*, *teaches*, *takes*, *crsdescr*, and *type* as our six T_Q with thirty naive corrections ψ, in (6). Using, the aforementioned predicates, we segregate all the triples into groups, and then perform aggregation. The red shaded part in Fig. (3) shows the super signature of $'wrote'$ as $T_Q^{(wrote)}$ and its possible corrections as $\Psi^{(wrote)}$.

2) APPLYING PROPOSED CORRECTION OPTIMIZATIONS ON OUR SAMPLE KNOWLEDGE GRAPH

Our next step is correction optimization in order to reduce the number of negative corrections by converting them into positive (+ve), negative (−ve), and star (*) corrections. For this, we compute whether it is beneficial to convert triples into super signatures. If the answer is affirmative, we compute its super signature and find only respective minimum corrections either positive (+ve) or negative (−ve) in correction list. On the contrary, we choose the original triples as our star (*) correction ϕ_*. For our sample graph, we choose triples of *advises*, *crsdescr*, and *type* as star (*) correction. However, *wrote* and *teaches* use positive as well as negative (−ev) corrections $\widehat{\Psi}$. The yellow shaded part in Fig[3] shows the super signature of $'wrote'$ as $T_Q^{(wrote)}$ and its updated correction list as $\widehat{\Psi}^{(wrote)}$. Thus, the updated corrections are lesser than the naive corrections previously computed. i.e., $\widehat{\Psi}^{(wrote)} < \Psi^{(wrote)}$. The discarded number of corrections α are replaced with respective positive corrections $\phi_+^{(wrote)}$ of subject P_2 and P_5. Thus, the aggregated graph after correction

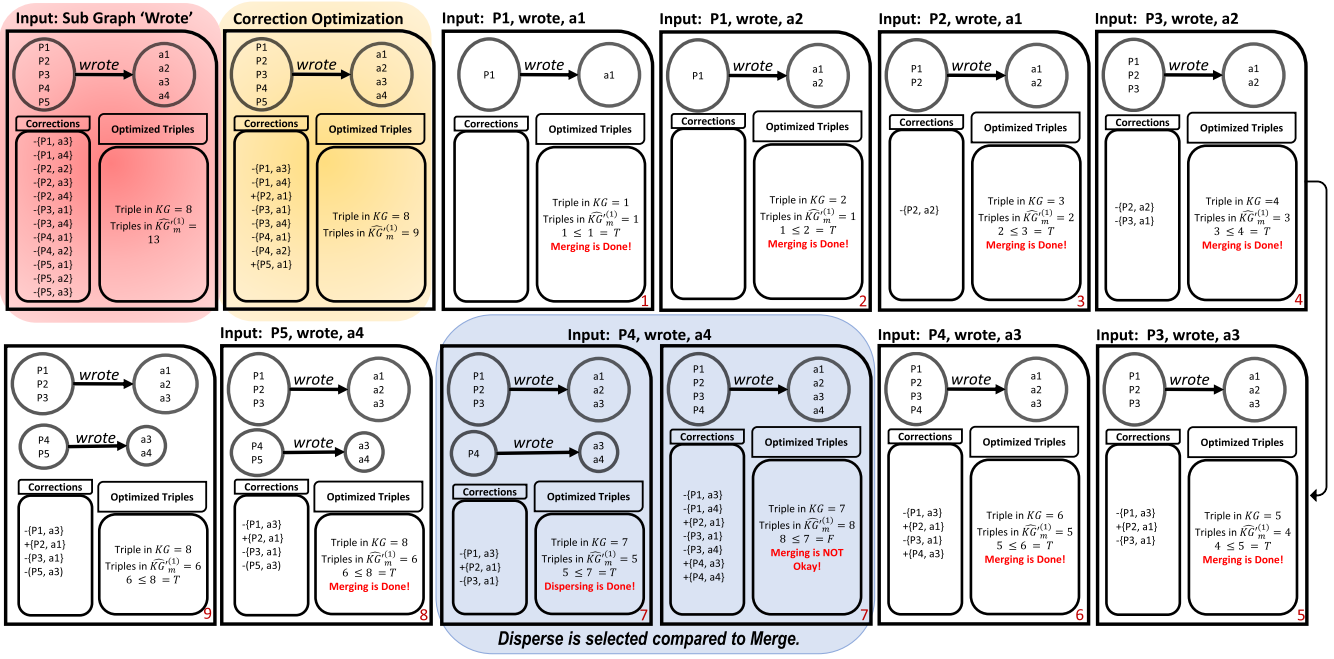

FIGURE 3. Complete execution of Correction Optimization, IBA-OC, and Triple optimization, IBA-OT, for instance 'wrote' in sample Knowledge Graph.

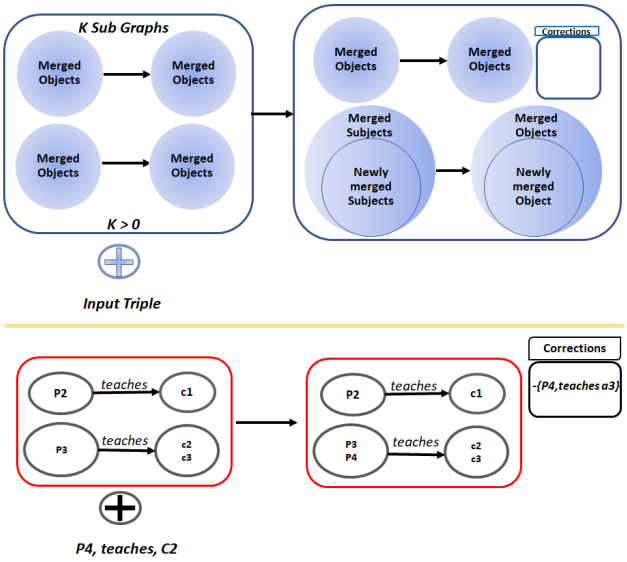

FIGURE 4. Illustration of proposed merge function for Correction Optimizations.

optimization shows completeness, and compactness in every aspect.

3) APPLYING PROPOSED TRIPLES OPTIMIZATIONS ON OUR SAMPLE KNOWLEDGE GRAPH

Lastly, we perform triple optimization and generate a resultant summary graph by identifying the specific region of aggregation from the graph with minimum corrections generation. To achieve this, we use our proposed *merge* and *disperse* functions for mapping events/triples into super signatures as triples optimization. Fig. (4) depicts the scenario where the merge is performed and Fig. (5) shows the dispersal of events for a specific quotient. It is an iterative process in which every triple, during merging, has to pass the criteria first. In case of failure, it performs disperse. It allows merging in all iterations except in seventh where the t_r does not meet the criteria and allows splitting of a single super signature into two sub super signatures for $'wrote'$. In consequence, after triple optimization, the resultant graph $\widehat{KG}^{\prime(wrote)}$ is the combination of two (02) sub super signature of wrote: $\widehat{T}_Q^{(wrote)}$ with its four (04) optimized corrections. Thus, $\widehat{\Psi}^{(wrote)}$ preserves less number of triples with complete information and knowledge. Note that the formation of multiple sub-signatures may depend on the sequence of execution but it ensures compactness.

V. KEY CONSTRAINTS FOR SUMMARIZING A KNOWLEDGE GRAPH AND THE PROPOSED SOLUTION

In this section, we discuss three C's (3C) key metrics of a summary graph of a KG. From the perspective of its correctness, a resultant summary graph is lossless if it satisfies the following:

- **Completeness:** For a given KG, its summary should ensure complete information accessible in comparison to the original graph regardless of type of aggregation operations during summarization.
- **Compactness:** The size after summarization of a KG should guarantee lesser number of triples in comparison

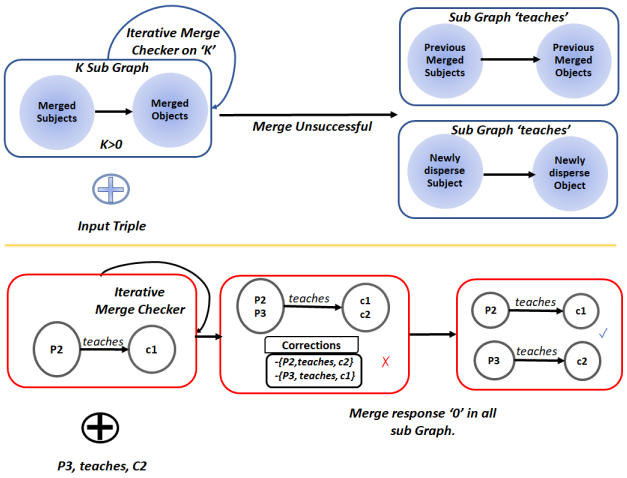

FIGURE 5. Illustration of proposed disperse function for triples optimization.

to the input graph. In this way, reduction of triples ensures less number of nodes, and edges as well.

- **Correctness:** It's important, that what is to be added or deleted in the resultant graph. Thus, a resultant graph should enable the removal of false information during retrieval and does not disturb the parent structure of input KG.

In consideration of the above constraints, below we show how IBA-OTC, our proposed solution, restricts the triples limit, ensures reduction in graph size, and behaves non-volatile in the formation of super signatures. We present the details of IBA-OTC satisfying the aforementioned constraints using the following theorems.

Theorem 1: (Given a KG, its resultant summary graph $\widehat{KG}'$ *restrict triples and ensure reduction while achieving its completeness during aggregation).*

proof: Lets given a subject s associated with predicate p to three different instances subjects/objects/literals. Optimized corrections restrict Φ and optimized triples restrict $\widehat{\Psi}$. Thus, if $KG^{(i)}$ associates to the original triples and $i \in T_Q^{(i)}$ and $j \in \Psi^{(i)}$ and $i \cup j > KG^{(i)}$ triples ratio t_r exceeds to 1. In this case, we consider star(*) corrections, hence it is unshrinkable.

In detail, aggregated triples with many corrections are only replaced by original triples or represented with positive and negative corrections in case of fewer numbers. Consider a sub-graph with n triples having all instances of literals/objects/subjects as different. Their association for aggregation needs $i \cup j$ number of corrections. The proposed approach is smart in a sense that it does not start aggregation and process of correction optimization because it restricts that if $i \cup j < n$ then only compute its signature and corrections. Therefore, the resultant triples count after optimized corrections does not exceed the original triples.

Theorem 2: (Given a KG, its resultant summary graph $\widehat{KG}'$ *ensures reduction in triples).*

Proof for Case 1: For any specific graph, any Q specific quotient sub-graph chooses its optimal selections for corrections. For that purpose, the *merge* triggers while promising lesser number of corrections. When the model perform *merge* for all instances. Using (15).

$$\widehat{KG}'^{(1)} = \widehat{KG}_m'^{(1)} t_r \leq 1$$

and we get from (14)
the resultant graph $\widehat{KG}'^{(1)}$ for an instance 1 we get.

$$\widehat{T}_Q^{(1)} \cup \widehat{\Psi}^{(1)} = \widehat{KG}_m'^{(1)}$$

if $\widehat{T}_Q^{(1)}$ is one, it means the predicate has one Super Signature and no further splitting is required throughout the complete iteration for a specific predicate. Therefore, we get no response from disperse part and we get from (14).

$$\widehat{KG}_m'^{(1)} = \widehat{T}_m^{(1)} \cup \widehat{\Psi}^{(1)}$$

In merge case when a single super signature for any predicate present in its resultant sub-graph then the updated correction list is $\widehat{\Psi} < \Psi$. The discarded number of corrections is expressed as α. It means the graph size reduces by α which ensures reduction of the resultant graph in comparison to the input sub-graph. From (11)., we get.

$$KG'^{(1)} < \widehat{KG}_m'^{(1)} - \alpha$$

and it is applicable to complete disjoint resultant graph $\widehat{KG}$. From (18)., we get.

$$= \widehat{KG}_m'^{(1)} - \alpha_1 \sqcup \widehat{KG}_m'^{(2)} - \alpha_2 \ldots \sqcup \widehat{KG}_m'^{(Q-1)} - \alpha_{Q-1} \sqcup \widehat{KG}_m'^{(Q)} - \alpha_Q$$

Thus a resultant graph contains Q different sub-graphs with only merging is reduced by.

$$= \bigcup_{p_i \in Q} (KG^{(i)} - [\alpha_1 + \alpha_2 + \alpha_3 \ldots \alpha_{Q-1} + \alpha_Q]$$

which is expressed as:

$$= \bigcup_{p_i \in Q} (KG^{(i)} - \sum_{i=1}^{Q} \alpha_i = \widehat{KG}'$$

Here, the resultant graph represents the union of all instances reduced by $\sum_{i=1}^{Q} \alpha_i$ in comparison to KG.

Proof for Case 2: For any specific graph, any Q specific quotient sub-graph chooses its optimal selections for corrections. For that purpose, the *disperse* triggers while promising a lesser number of corrections.

From Equation (15).

$$\widehat{KG}'^{(1)} = \widehat{KG}_d'^{(1)}$$

and Equation. [14] we get.

$$\widehat{KG}'^{(1)} = \widehat{T}_{Q_d}^{(1)} \cup \widehat{\Psi}_d^{(1)}$$

generally expressed as:

$$\widehat{KG}'^{(1)} = \sum_{d=1}^{D} \widehat{T}_{Q_d}^{(1)} \cup \widehat{\Psi}_d^{(1)}$$

it means $\widehat{T}_{Q_d}^{(1)}$ are multiple and split into d different sub-super signatures with its respective correction lists $\widehat{\Psi}_d^{(1)}$. In case of *disperse*, multiple super signatures for a single predicate first optimize by corrections and then optimize by triples. Thus, $\widehat{\Psi} < \Psi$ (*e.g.*$\widehat{\Psi}$ is reduced by α) and also $\widehat{\Psi}_d^{(1)} < \widehat{\Psi}$ (*e.g.*$\widehat{\Psi}_d^{(1)}$ is reduced by β) Therefore, (14) can be expressed as.

$$\widehat{KG}'^{(1)} = \sum_{d=1}^{D} \widehat{T}_{Q_d}^{(1)} \cup \widehat{\Psi}_d^{(1)} - [\alpha + \beta]$$

and the resultant graph $\widehat{KG}'$ from (18) can be expressed as.

$$= \bigcup_{i \in Q} (KG^{(i)} - [\sum_{i=1}^{Q} \alpha_i + \sum_{d=1}^{D} \beta_D]$$

This proves our resultant graph $\widehat{KG}'$ ensures lesser number of triples with our two-stage optimization process. It first optimizes Ψ into $\widehat{\Psi}$ and then optimizes into $\widehat{\Psi}_d^{(Q)}$ for all Q sub-quotients for splitting super signature into sub-super signatures. Moreover, it is also possible that a graph or sub-graph performs a mixture of merging and dispersing because it is our iterative process and we check every approachable instance during mapping for its suitable region.

Proof for Case 3: For any specific graph, any Q specific quotient sub-graph chooses its optimal selections for corrections. For that purpose, both *merge* and *disperse* trigger while promising less number of corrections.

The model ensures reduction but it is lesser in comparison to case 2. Therefore, we estimate how much the size of the graph increases in comparison to case 2. Note, If merging fails for all iterations we use disperse. In this way, super signatures are the same as original triples that occur seldom. On the other hand, for every successful *merge* in k (*e.g.*$k \leq Q$) merging cycle, it gains by the size of γ and the overall size of the graph is the additive factor. Hence, the resultant graph is increased by a factor of $\sum_{i=1}^{k} \gamma_i$. Therefore, (18). can be expressed as.

$$= \bigcup_{i \in Q} (KG^{(i)} - [\sum_{i=1}^{Q} \alpha_i + \sum_{d=1}^{D} \beta_D] + \sum_{i=1}^{k} \gamma_i$$

Generally, case 3 occurs because every graph has some overlapping and non-overlapping regions.

Theorem 3: (Given a KG, its super signatures of resultant graph $\widehat{KG}'$ are Non-Volatile in nature).

Case 1: Single super signature for each predicate is formed: In case T_Q are unique, the case is addressed in optimized correction and guarantees fewer number triples that save computational cost due to optimization when only perform *merge*.

TABLE 3. Details of the datasets used for experiments.

KG	$Triples$	s_i	p_j	ω
KEGG	110971	1106	13	7827
Affemetrix	804947	231	35	6607
Drug Bank	766921	341	57	10857
BSBM BenchMark	50117	1213	15	2978
DBpedia(Persondata)	6670072	2001	9	8175
DBpedia(Geo Coordinates)	320191	4000	4	10896
watdiv (10 M)	11878231	5597	85	13258
watdiv (100 M)	100131004	5597	85	13258
BSBM BenchMark (100 M)	100,001,402	9,201,350	411	3884140

Case 2:Multiple balanced signatures: For a heterogeneous graph it never happens. if the input graph is homogeneous, the size of Q signatures split in equal size with a fraction of b. The total time saved by the triples is bk computational cycles which is ideal for uniform computations at any region.

Case 3:Multiple but unbalanced signatures: It occurs several times, signatures with large number of corrections for such triples have higher b. The greater size of b leads to the denser distribution of any sub-graph. However, lesser b causes fewer fact aggregation with its optimized corrections.

VI. EXPERIMENTAL EVALUATION

In this section, we present the experimental evaluation of our approach. For the experiments, we used a system having 128 GB RAM with 6 core processors with Linux OS (Ubuntu). All the implementations are programmed in Python.

A. DATASETS

We perform experiments on 9 publicly available real-world knowledge graphs, the description of these datasets shown in Table [3]. It shows their triples as instances, unique subjects, predicates, and objects of each KG.

To observe the impact of KG size during summarization, we perform a series of experiments on each dataset, and show the results of each experiment in Table. (5). Our goal is to show the impact of reduction of $\widehat{KG}$ after correction optimization and $\widehat{KG}'$ after triple optimization with respect to input KG.

B. EVALUATION ON DATASETS USING COMPLETE KGS

We perform the same experiments on a large knowledge graph to further understand the effectivesness of our proposal. For this, we add watdiv benchmark and analyse the results on both variants (wdiv 10M, watdiv100 M). We observe that our results further validate our claim and yield similar reduction on large graphs. In addition, we also perform BSBM large data. The details of all such experiments are shown in Table (5). In a large dataset, multiple unbalanced super signatures are formed with large triples because multi sub-super signatures reduce the size with a higher fraction in comparison to quotient of KG.

TABLE 4. Experiments on real-world datasets with different size.

Dataset	KG	s_i	p_j	ω	$\widehat{KG}$	$(\widehat{KG}/KG)$*100	$\widehat{T}_Q$	$\widehat{\Phi}$	$\widehat{KG}'$	$(\widehat{KG}'/KG)$*100
KEGG.rn	500	37	13	292	460	**92.96**	16	370	386	**77.20**
	1000	76	13	548	926	**92.61**	21	676	697	**69.70**
	2000	143	13	1049	1859	**92.59**	23	1337	1360	**68.00**
	4000	292	13	2180	3710	**92.75**	28	2689	2717	**67.92**
	8000	569	13	4371	7433	**92.91**	32	5553	5585	**69.82**
	16000	1106	13	7827	14809	**92.56**	41	11111	11152	**69.70**
AFFEMETRIX	500	9	35	339	449	**89.8**	43	348	391	**78.22**
	1000	14	35	665	862	**86.2**	44	726	770	**77.00**
	2000	27	35	1062	1725	**86.25**	41	1513	1554	**77.71**
	4000	60	35	2031	3359	**83.97**	57	3041	3076	**76.91**
	8000	118	35	3665	6716	**83.95**	67	6110	6177	**77.22**
	16000	231	35	6607	13476	**84.22**	84	12367	12451	**77.82**
Drug Bank	500	13	57	292	463	**92.60**	64	382	446	**89.21**
	1000	29	56	666	890	**89.23**	62	833	895	**89.51**
	2000	55	58	1254	1725	**89.25**	67	1677	1744	**87.24**
	4000	89	61	2897	3740	**93.15**	75	3410	3485	**87.12**
	8000	170	57	5588	7327	**91.58**	77	6889	6966	**87.07**
	16000	341	57	10856	14982	**93.63**	97	13810	13907	**86.91**
BSBM	500	98	6	237	308	**61.6**	12	272	284	**56.81**
	1000	201	5	434	403	**40.3**	7	406	413	**41.31**
	2000	222	21	1085	1653	**82.65**	27	1567	1594	**71.17**
	4000	202	28	1736	2870	**71.75**	43	2726	2769	**69.22**
	8000	760	15	2024	3764	**47.05**	70	3528	3598	**44.97**
	16000	1213	15	2978	13768	**86.05**	28	10722	10750	**67.19**
DbPedia (Person Data)	500	69	9	362	358	**71.68**	22	317	339	**67.81**
	1000	152	9	610	704	**70.39**	28	651	679	**66.25**
	2000	262	9	1265	1417	**70.85**	35	1310	1345	**67.25**
	4000	517	9	2350	2830	**70.75**	55	2632	2687	**67.19**
	8000	10017	9	4350	5692	**71.15**	64	5396	5460	**68.25**
	16000	2001	9	8175	11360	**71.01**	82	10802	10844	**68.03**
DbPedia (Geo-Coordinates)	500	125	4	375	128	**25.6**	10	101	111	**22.06**
	1000	250	4	725	253	**25.3**	25	209	234	**23.45**
	2000	1000	4	2823	1003	**25.07**	37	420	459	**22.95**
	4000	1000	4	2823	1003	**25.07**	47	881	928	**23.21**
	8000	2000	4	5576	2006	**25.10**	111	1675	893	**22.33**
	16000	4000	4	10869	4013	**25.18**	216	3148	3364	**21.02**

C. A EVALUATION EXPLANATION ON EACH DATASET

Now we present experimental details on each dataset.

1) KEGG.RN

KEGG.rn contains 1.1 M triples having 13 unique predicates. Each subject is associated with several other objects with 13 unique predicates. Thus the predicates are repeating in triples. Some of the notable predicates are *schema#label*, *kegg#xProduct*, *title*, *identifier*,.

2) AFFEMETRIX

Our second dataset for evaluating our approach is AFFETMETRIX dataset and we choose a file $RT_U34.na32.annot$ from bulk for experiments. It contains 69 K triples about

TABLE 5. Table shows the reduction percentage of knowledge Graph with optimized correction and optimized triples in comparison to original knowledge graph.

Dataset	KG	s_i	p_j	ω	$\widehat{KG}$	$(\widehat{KG}/KG)*100$	$\widehat{T}_Q$	$\widehat{\Phi}$	$\widehat{KG}'$	$(\widehat{KG}'/KG)*100$
KEGG.rn	110971	8934	13	42136	102426	**92.30**	392	75878	76270	**68.73**
AFFEMETRIX	804947	14365	40	245362	791987	**98.39**	3078	612464	615542	**76.47**
Drug	766921	16027	69	2213984	695904	**90.73**	2361	644843	647204	**84.39**
BSBM	50117	3254	32	6749	40043	**91.87**	1246	37960	39206	**78.23**
DbPedia	6670072	1415	9	15426	4755761	**71.29**	728	4282125	4282853	**64.21**
DbPedia	320191	79834	4	214380	80143	**25.02**	10	75362	75372	**23.54**
wat div 10M	11878231	5597	85	13258	11618098	**97.81**	1725	9297742	9299467	**78.29**
wat div 100M	100131004	5597	85	13258	98138397	**98.01**	3254	74584330	74587584	**74.49**
BSBM 100M	100001402	9201350	411	3884140	97441366	**97.44**	2320	78908786	78911106	**78.91**

the information of several associated items with thirty-five 35 unique predicates. Some of the notable predicates are *location*, *process*, *function*, *scientific* − *name*, *annotation* − *date*, *id*, and *type*.

3) DRUG BANK

Drug Bank is our third data set for experiments. The information in this graph consists of several drug types used for special treatment. The subject of RDF graphs is Drug ids while predicates are the properties of different drug ids. The property can be its name, type, description, updated date, predicted solution for solubility, and information of use, etc. The objects are mostly literals or constant values which contain 151 M triples.

4) BSBM BENCHMARK

BSBM benchmark is our fourth dataset for experiments. The dataset contains the number of producers of the product with its type of different reviews and their reviewer's information. The subject of the RDF graph is *product*, *offers*, and *reviews*. Some of the notable properties are *comment*, *Publisher* − *date*, *lable*, *type* and etc. BSBM is the standard benchmark publically available in two variants. One contains 50K triples and the other contains 100M triples as a large dataset. We performed our experiments on both datasets to show the scalability of our approach.

5) DBPEDIA (PERSON DATA)

Another important dataset is DBpedia (person data) where persons are associated with (09) nine different properties among *name*, *type*, *description*, *Birthplace*, *Deathplace*, *Dathdate*, *Birthdate*, *givenname* and *surname*. It contains 6.6 M triples.

6) DBPEDIA (GEO-COORDINATES)

Our last dataset is Dbpedia (Geo-Coordinates). The dataset contains the information of different location points with their *name*, *longitude*, *latitude* and *type*. It contains 47.2 M triples of information available at different points. Points are repeated as objects of the RDF graph. As, each point is a different value, we consider cosine distance while mapping.

7) EVALUATION ON KEGG.RN

On evaluating IBA-OTC on our first dataset, With 4000 triples of *KG*, the size of the remaining graph after optimized corrections $\widehat{KG}$ is 92.75 % which is further optimized to 67.92 % using IBA-OT which is total 2717 triples remaining in the resultant graph $\widehat{KG}'$. Note that it is lossless. In 2717 triples, 28 are super signatures $\widehat{T}_Q$ while 2689 are its optimized corrections $\widehat{\Phi}$ of each individual. Some super signatures are not split into sub-super signatures. From 13 predicates, only six (06) predicates participate in splitting;

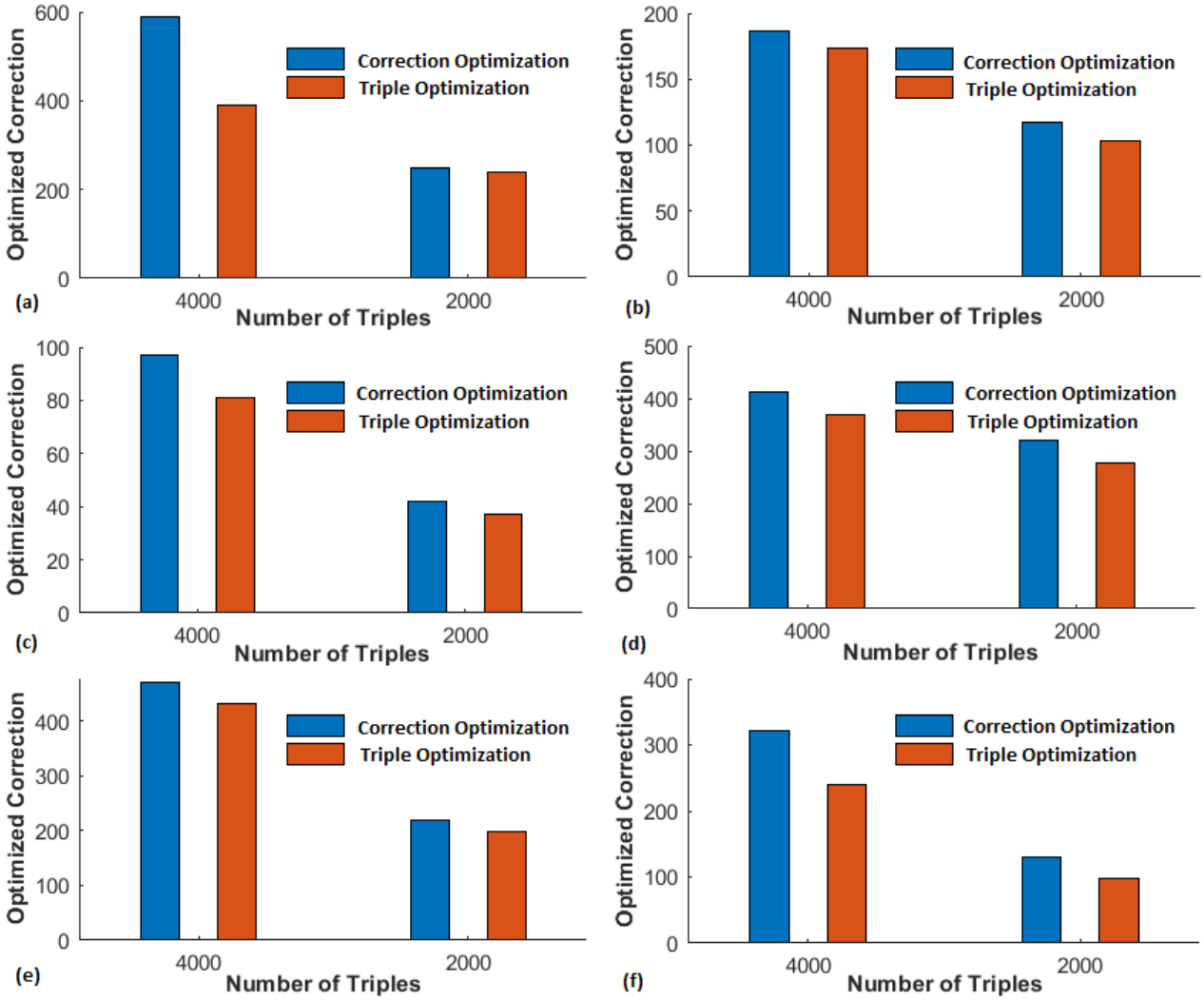

FIGURE 6. Correction Optimization and Triple Optimization (Left to Right) when a single predicate is selected. (a) KEGG.rn with predicate *Kegx-Product* (b) AFFEMETRIX with predicate *date annotation* (c) Drug Bank with predicate *update date* (d) BSBM with predicate *Product Feature* (e) DbPedia (Person Data) with predicate *birth place* (f) DbPedia (Geo-coordinates) with predicate *type*.

the rest of seven predicates only map into one super signature with many corrections.

8) EVALUATION ON AFFEMETRIX

On evaluation of IBA-OTC on this dataset, size of the remaining graph after optimized corrections $\widehat{KG}$ is 83.97% which is further optimized to 76.91 % using optimized corrections containing 3076 triples in $\widehat{KG}'$. The information aggregates to 57 super signatures $\widehat{T}_Q$ with its 3041 optimized corrections $\widehat{\Phi}$. Thus, a resultant graph is the union of $\widehat{T}_Q$ and $\widehat{\Phi}$. From 57 predicates, twenty-one 21 predicates with a single super signature do not participate in the formation of super facts, except 14 predicates.

9) EVALUATION ON DRUG BANK

To evaluate drug bank data, with 4000 triples of KG, $\widehat{KG}$ is 93.15 % which is further optimized to 87.12 % using triples optimization. The total number of triples are 3485 in $\widehat{KG}'$. In $\widehat{KG}'$, 75 are super signatures $\widehat{T}_Q$ with 3410 are the optimized corrections $\widehat{\Phi}$. It is observed from 75 predicates, thirty-four 34 predicates with a single super signature and twenty-seven 27 participate in the formation of super facts.

The reason behind many predicates not participating in compression, is actually their unique name against every drug that leads to more corrections when aggregating through *merge* or *disperse*. Moreover, due to the heterogeneity in nature of such graph, some properties may seldom generate super facts or weakly contribute to reduction of $\widehat{KG}'$ like *information of use*, *solution of* and *solubility*. However, strong candidates for merging are *type* and *updateddate* for the action *merge* and *merge* after *disperse* when *merge* fail.

10) EVALUATION ON BSBM BENCHMARK

For evaluating the BSBM benchmark, $\widehat{KG}$ is 71.75 % which is further optimized to 69.22 % using correction

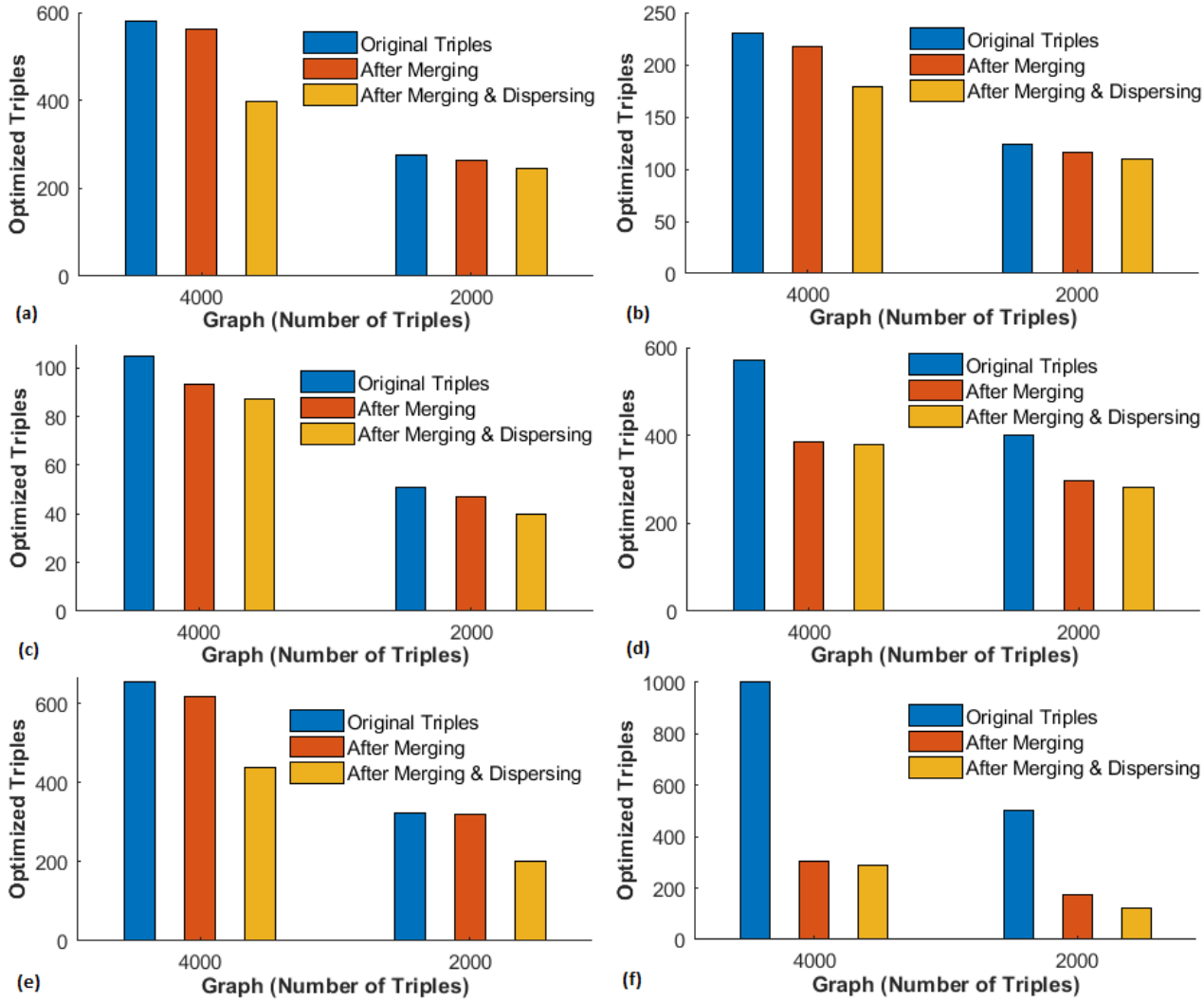

FIGURE 7. Merging and Dispersing(Left to Right). (a) KEGG.rn with predicate *Kegx-Product* (b) AFFEMETRIX with predicate *date annotation* (c) Drug Bank with predicate *update date* (d) BSBM with predicate *Product Feature*(e) DbPedia (Person Data) with predicate *birth place* (f) DbPedia (Geo-coordinates) with predicate *type*.

optimization. Thus, remaining triples in the resultant graph $\widehat{KG}'$ are 2769. From the $\widehat{KG}'$, 43 are super signatures $\widehat{T}_Q$ and 2726 are optimized corrections. From 43 predicates, only seven 7 participate in the super signatures that are factful process while the remaining predicates are with a single super signature.

11) EVALUATION ON DBPEDIA (PERSON DATA)

Another important dataset is DBpedia person data where people are associated with nine different properties. Some of nine properties are name, given name, surname, etc. Our approach reduces the size to 67.19%. The percentage reduction is 67. 19 % where the number of predicates is fixed in size. The formation of super facts $\widehat{T}_Q$ participates 4 out of 9. Such super facts show the close mapping of events with minimum $\widehat{\Phi}$.

12) EVALUATION ON DBPEDIA (GEO-COORDINATES)

Our last dataset for discussion and analysis is DBpedia (Geo-Coordinates) in which information about specific places is available. The graph contains information about several locations as subjects with four different properties. The property count is fixed. Therefore, the graph is homogeneous but the formation of super signatures and splitting them into multiple super signatures that are non-uniform after formation in resltant graph. It reduces the graph to approximately 75 % because several coordinates locations are repeated in the object and we consider cosine distance while mapping.

On evaluating IBA-OTC on Dbpedia (Geo-Coordinates), the resultant graph size is only 25.07% after performing correction optimization $\widehat{KG}$. It is further optimized with the slight reduction in IBA-OT as 23.21% in $\widehat{KG}'$. Only one predicate does not participate in super signature formation while the rest of three 3 are participating in the formation of

super signatures. The optimized correction against 47 super signatures $\widehat{T}_Q$ is 881 to make the resultant graph of 928 triples.

D. IMPACT OF PROPOSED OPTIMIZATIONS AND FUNCTIONS

We perform IBA-OC and IBA-OT to achieve completeness by reducing the number of triples in the system. Although, our naive approach increases the number of triples but provides lossless aggregation as signature of each predicate with its possible corrections. After a naive approach, we develop a model that restricts either fewer triples or considers the original triples of given KG. Reduction of triples is first achieved with correction optimization (I) and second correction optimization (II). Therefore, we discuss two more experiments to show the impact of reduction and optimization as follows.

a: CORRECTION OPTIMIZATION

Figure [6] shows the comparison of correction optimization (I) and correction optimization (II). Our model ensures the guarantee of reduction through our *merge* and *disperse* functions in the second type of optimization. We consider the triples count of 4000 and 2000 for experiments. It guarantees in reduction of triples before and after correction optimization.

b: MERGING AND DISPERSING

Fig. (7) shows the significance reduction of *merge* and *disperse*. It reveals that regardless of the dataset, it reduces by some fractional amount when information maps through our merge function and again guarantees reduction by a small fraction when it qualifies our *disperse* function. The number of triples in this experiment is the same as in Fig. (6). Thus, the reduction of disperse function creates single or multi sub-super signatures in the graph.

c: PREDICATE PARTICIPATION

To find about predicate participation, we perform another experiment to show in [8] the number of predicates involved in the formation of super signatures less in a fraction in comparison to not participating predicates. It is basically an ability of aggregating with minimum corrections. A higher chance of ability means a greater chance of predicate participation.

E. LIMITATION OF IBA-OTC

It is unable to distinguish RDF and RDFS triples. IBA-OTC treats the same for both triples. if *KG* contains both triples and it requires differentiation first before processing then it ignores the polarity of such triples. Also, our approach is for pre-processing for application related to retrival from KG. Currently this work only covers how to aggregate natural merging triples with minimum corrections that focus on the representation of any *KG* in $\widehat{KG}'$. Also, this model is not for retrieval of queries. However, the extension of this will cover query retrieval.

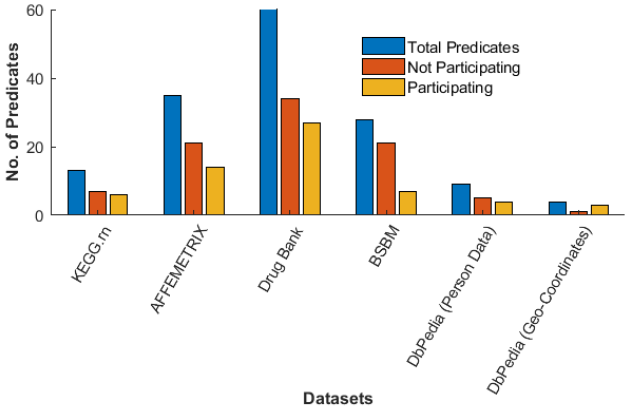

FIGURE 8. Predicates details of participating in mapping(Merge, Disperse) Process.

TABLE 6. Comparison of summary Ratio (SR)=($\widehat{KG}'$) / (*KG*)*100) on same datasets used by state-of-the-art approaches.

Datasets	No. of triples	[63] Lossy GBS	[53] Lossy GBS	Our Approach Lossless GBS	[63] Lossless QBS	Our Approach Lossless QBS+GBS
DBpedia	2,047	32.2 %	-	73 %	99 %	**96.3 %**
ESBM	6,584	33 %	-	88.1 %	99.8 %	**94.8 %**
WatDiv.10M	10,916,457	41.8 %	-	84.2 %	96.6 %	**94.2 %**
WatDiv.100M	108,997,714	57 %	-	86.5 %	97.4 %	**93.5 %**
DBLP	4657	-	96 %	90.7 %	-	**94.2 %**
Geonames	119416	-	97 %	91.2 %	-	**94.7 %**

F. COMPARISON WITH STATE-OF-THE-ART APPROACHES

We also perform experiments to compare against state-of-the-art methods. We perform these experiments on the same data mentioned by these approaches. Below, we discuss the details one by one.

a: SUMMARY RATIO WITH ITS COMPLETENESS

Table. (6) shows the summary ratio of the graph keeping in view its completeness. IBA-OTC produces lesser compression but it guarantees completeness in the resultant graph. However, it outperforms and commits more reduction in comparison to [57]. From analysis, our approach surely reduce significant number of triples with the assurance of it's 3C (Completeness, Compactness, and Correctness) as discussed above. It also shows that regardless the nature (homogeneous/heterogeneous) of any KG, our approach provides a multi-level lossless summary that remove the need of input graph to still keep in repository in secondary storage or even memory for saving computational time as well as its space efficiency.

b: COMPUTATIONAL TIME COMPARISON ON FORMATION OF SUPER SIGNATURE VS SUPER NODE

Table. (7) shows the time comparison of aggregation of instances for both GBS, and QBS. For all experiments, our

TABLE 7. Computational Time in seconds (s) for Instance-based aggregation in Comparison with state-of-the-art approaches.

Datasets	No. of triples	[63] GBS Aggregation $\widehat{T}_Q$	Our Approach GBS+QBS Aggregation+Correction $\widehat{T}_Q + \widehat{\Phi}$	[63] QBS Aggregation $\widehat{T}_Q$	Our Approach GBS+QBS Aggregation+Correction $\widehat{T}_Q + \widehat{\Phi}$
DBpedia	2,047	36.8 s	(22.21 + 8.24) s=**30.45 s**	2.8 s	(1.97 + 0.078) s =**2.057 s**
ESBM	6,584	34.2 s	(18.89 + 11.29) s =**30.18 s**	2.2 s	(1.61 + 0.089) s =**1.699 s**
WatDiv.10M	10,916,457	100.8 s	(75.92 + 15.43) s=**91.35 s**	14.2 s	(8.95 + 1.2356) s=**10.1856 s**
WatDiv.100M	108,997,714	787.6 s	(598.33 + 103.45) s=**701.78 s**	53.6 s	(41.21 + 4.215) s =**45.425 s**

approaches.

approach computes super signatures in a less computational time in comparison to [57] and corrections efficiently in comparison to [57]. Notably, the information is lossless and efficient due less computational cost.

VII. CONCLUSION

We developed a novel approach, IBA-OTC, based on aggregation and performed better lossless graph summarization of a KG in comparison to state-of-the-art work in the domain. We first established naive signatures of predicate-based aggregation. For such aggregation, we achieve 1). completeness. Next, we perform iterative optimization in two phases. We first identify positive corrections (+ve) and replace them with selective negative corrections. Also, we identify specific natural merging regions with minimum corrections of the graph by our two newly designed functions *merge* and *disperse*. After optimization of triples, our approach ensures KG's size reduction in a lossless manner. Moreover, we performed several experiments on large graphs to show the reduction of different KGs. We also validate the scalability test of our approach by conducting experiments on large scale datasets (wat div 100 M, BSBM 100M). At the end, we performed two more experiments to highlight the imapct of resultant summary graph in a lossless manner with comparisons of state-of-the-art approaches.

VIII. FUTURE WORK

Convolutional Neural Network (CNN) in deep learning (DL) facilitates fast analysis of heterogeneous structures [73], [74], [75], [76]. Such networks are repeatedly discussed in [30], [77], [78], and [83] with using DL techniques. We therefore, plan to work on a query retrieval model likewise discussed in [84], [85], and [86] for summarizing graphs $\widehat{KG}'$ and apply recurrent DL model for similarity measurement presented in [28], [30], [73], and [90] and use long short-term memory (LSTM) network for fast recognition of triples mapping. Aggregating such graphs with lossless saves computational costs for processing, in-memory visualization, and fast query retrieval using deep transfer learning and casual relational reasoning of input *KG* discussed in [51] and [92]. Due to its complexity, we further plan to investigate how to retrieve knowledge in a summary graph.

ACKNOWLEDGMENT

The authors would like to thank Dr. Ahmad Jalal for his support for his keen analytical and experimental observations.

REFERENCES

[1] Y. Liu, T. Safavi, A. Dighe, and D. Koutra, "Graph summarization methods and applications: A survey," *ACM Comput. Surv.*, vol. 51, no. 3, pp. 1–34, May 2019.

[2] Š. Čebirić, F. Goasdoué, H. Kondylakis, D. Kotzinos, I. Manolescu, G. Troullinou, and M. Zneika, "Summarizing semantic graphs: A survey," *VLDB J.*, vol. 28, no. 3, pp. 295–327, Jun. 2019.

[3] Q. Liu, G. Cheng, K. Gunaratna, and Y. Qu, "Entity summarization: State of the art and future challenges," *J. Web Semantics*, vol. 69, Jan. 2021, Art. no. 100647.

[4] A. Schätzle, A. Neu, G. Lausen, and M. Przyjaciel-Zablocki, "Large-scale bisimulation of RDF graphs," in *Proc. 5th Workshop Semantic Web Inf. Manage.*, Jun. 2013, pp. 1–8.

[5] A. Beheshti, B. Benatallah, H. R. Motahari-Nezhad, S. Ghodratnama, and F. Amouzgar, "BP-SPARQL: A query language for summarizing and analyzing big process data," in *Process Querying Methods*. Cham, Switzerland: Springer, 2021, pp. 21–48.

[6] V. Nebot and R. Berlanga, "Towards analytical MD stars from linked data," in *Proc. Int. Conf. Knowl. Discovery Inf. Retr.*, 2014, pp. 117–125.

[7] T. Tran, G. Ladwig, and S. Rudolph, "Managing structured and semistructured RDF data using structure indexes," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 9, pp. 2076–2089, Sep. 2013.

[8] K. Li, L. Ji, S. Yang, H. Li, and X. Liao, "Couple-group consensus of cooperative–competitive heterogeneous multiagent systems: A fully distributed event-triggered and pinning control method," *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 4907–4915, Jun. 2022, doi: 10.1109/TCYB.2020.3024551.

[9] M. Tasnim, D. Collarana, D. Graux, and M. E. Vidal, "Context-based entity matching for big data," in *Knowledge Graphs and Big Data Processing*, vol. 12072. Cham, Switzerland: Springer, 2020, pp. 122–146.

[10] K. U. Khan, W. Nawaz, and Y.-K. Lee, "Set-based approximate approach for lossless graph summarization," *Computing*, vol. 97, no. 12, pp. 1185–1207, Dec. 2015.

[11] Z. Zhang, L. Wang, W. Zheng, L. Yin, R. Hu, and B. Yang, "Endoscope image mosaic based on pyramid ORB," *Biomed. Signal Process. Control*, vol. 71, Jan. 2022, Art. no. 103261, doi: 10.1016/j.bspc.2021.103261.

[12] J. Zhang, C. Zhu, L. Zheng, and K. Xu, "ROSEFusion: Random optimization for online dense reconstruction under fast camera motion," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–17, Aug. 2021, doi: 10.1145/3450626.3459676.

[13] H. Tian, N. Huang, Z. Niu, Y. Qin, J. Pei, and J. Wang, "Mapping winter crops in China with multi-source satellite imagery and phenology-based algorithm," *Remote Sens.*, vol. 11, no. 7, p. 820, Apr. 2019, doi: 10.3390/rs11070820.

[14] H. Tian, J. Pei, J. Huang, X. Li, J. Wang, B. Zhou, Y. Qin, and L. Wang, "Garlic and winter wheat identification based on active and passive satellite imagery and the Google Earth Engine in northern China," *Remote Sens.*, vol. 12, no. 21, p. 3539, Oct. 2020, doi: 10.3390/rs12213539.

[15] A. R. S. L. da Costa, A. Santos, and J. P. Leal, "Large semantic graph summarization using namespaces," in *Proc. 11th Symp. Lang., Appl. Technol.*, 2022, pp. 1–9.
[16] J. Guo and Y. Wang, "Summarizing RDF graphs using node importance and query history," in *Proc. Int.Conf. Service Sci. (ICSS)*, May 2021, pp. 51–58.
[17] Y. Shen, N. Ding, H.-T. Zheng, Y. Li, and M. Yang, "Modeling relation paths for knowledge graph completion," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 11, pp. 3607–3617, Nov. 2021, doi: 10.1109/TKDE.2020.2970044.
[18] L. Cadorel and A. G. B. Tettamanzi, "Mining RDF data of COVID-19 scientific literature for interesting association rules," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. Intell. Agent Technol. (WI-IAT)*, Dec. 2020, pp. 145–152.
[19] G. Trouli, A. Pappas, G. Troullinou, L. Koumakis, N. Papadakis, and H. Kondylakis, "SumMER: Structural summarization for RDF/S KGs," *Algorithms*, vol. 16, no. 1, p. 18, Dec. 2022.
[20] S. Campinas, R. Delbru, and G. Tummarello, "Efficiency and precision trade-offs in graph summary algorithms," in *Proc. 17th Int. Database Eng. Appl. Symp.*, 2013, pp. 38–47.
[21] A. Generale, T. Blume, and M. Cochez, "Scaling R-GCN training with graph summarization," in *Proc. Companion Web Conf.*, Apr. 2022, pp. 1073–1082.
[22] M. Ioana and M. Mazuran, "Speeding up RDF aggregate discovery through sampling," in *Proc. 2nd Int. Workshop Big Data Vis. Explor. Anal.*, 2019.
[23] Q. She, R. Hu, J. Xu, M. Liu, K. Xu, and H. Huang, "Learning high-DOF reaching-and-grasping via dynamic representation of gripper-object interaction," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–14, Jul. 2022, doi: 10.1145/3528223.3530091.
[24] Y. Zheng, Y. Zhang, L. Qian, X. Zhang, S. Diao, X. Liu, J. Cao, and H. Huang, "A lightweight ship target detection model based on improved YOLOv5s algorithm," *PLoS ONE*, vol. 18, no. 4, Apr. 2023, Art. no. e0283932, doi: 10.1371/journal.pone.0283932.
[25] Z. Hu, L. Ren, G. Wei, Z. Qian, W. Liang, W. Chen, X. Lu, L. Ren, and K. Wang, "Energy flow and functional behavior of individual muscles at different speeds during human walking," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 294–303, 2023, doi: 10.1109/TNSRE.2022.3221986.
[26] G. Zhou and X. Liu, "Orthorectification model for extra-length linear array imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–10, 2022, doi: 10.1109/TGRS.2022.3223911.
[27] Q. Liu, H. Yuan, R. Hamzaoui, H. Su, J. Hou, and H. Yang, "Reduced reference perceptual quality model with application to rate control for video-based point cloud compression," *IEEE Trans. Image Process.*, vol. 30, pp. 6623–6636, 2021, doi: 10.1109/TIP.2021.3096060.
[28] F. Ahmad, "Deep image retrieval using artificial neural network interpolation and indexing based on similarity measurement," *CAAI Trans. Intell. Technol.*, vol. 7, no. 2, pp. 200–218, Jun. 2022, doi: 10.1049/cit2.12083.
[29] G. Zhou, Q. Wang, Y. Huang, J. Tian, H. Li, and Y. Wang, "True2 orthoimage map generation," *Remote Sens.*, vol. 14, no. 17, p. 4396, Sep. 2022, doi: 10.3390/rs14174396.
[30] X. Zhang, S. Wen, L. Yan, J. Feng, and Y. Xia, "A hybrid-convolution spatial–temporal recurrent network for traffic flow prediction," *Comput. J.*, vol. 2022, pp. 1–15, Nov. 2022, doi: 10.1093/comjnl/bxac171.
[31] R. Wang, D. Sun, and R. Wong, "RDF knowledge base summarization by inducing first-order horn rules," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, 2022, pp. 188–204.
[32] Y. Tian, R. A. Hankins, and J. M. Patel, "Efficient aggregation for graph summarization," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2008, pp. 567–580.
[33] B. Glimm, Y. Kazakov, T. Liebig, T.-K. Tran, and V. Vialard, "Abstraction refinement for ontology materialization," in *Proc. Int. Semantic Web Conf.* Cham, Switzerland: Springer, Oct. 2014, pp. 180–195.
[34] H. Zhang, Y. Duan, X. Yuan, and Y. Zhang, "ASSG: Adaptive structural summary for RDF graph data," in *Proc. ISWC*, 2014, pp. 233–236.
[35] M. Dudáš, V. Svátek, and J. Mynarz, "Dataset summary visualization with lodsight," in *Proc. Eur. Semantic Web Conf.* Cham, Switzerland: Springer, May 2015, pp. 6–40.
[36] F. Goasdoué, I. Manolescu, and A. Roatiş, "Efficient query answering against dynamic RDF databases," in *Proc. 16th Int. Conf. Extending Database Technol.*, Mar. 2013, pp. 299–310.
[37] S. Khatchadourian and M. P. Consens, "ExpLOD: Summary-based exploration of interlinking and RDF usage in the linked open data cloud," in *Proc. 7th Extended Semantic Web Conf.*, Crete, Greece, 2010, pp. 272–287.
[38] G. Zhou, H. Li, R. Song, Q. Wang, J. Xu, and B. Song, "Orthorectification of fisheye image under equidistant projection model," *Remote Sens.*, vol. 14, no. 17, p. 4175, Aug. 2022, doi: 10.3390/rs14174175.
[39] R. Zhang, L. Li, Q. Zhang, J. Zhang, L. Xu, B. Zhang, and B. Wang, "Differential feature awareness network within antagonistic learning for infrared-visible object detection," *IEEE Trans. Circuits Syst. Video Technol.*, p. 114, 2023, doi: 10.1109/TCSVT.2023.3289142.
[40] B. Cheng, D. Zhu, S. Zhao, and J. Chen, "Situation-aware IoT service coordination using the event-driven SOA paradigm. *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 2, pp. 349–361, Mar. 2016, doi: 10.1109/TNSM.2016.2541171.
[41] Y. Wang, N. Xu, A.-A. Liu, W. Li, and Y. Zhang, "High-order interaction learning for image captioning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4417–4430, Jul. 2022, doi: 10.1109/TCSVT.2021.3121062.
[42] A. Pappas, G. Troullinou, and G. Roussakis, "Exploring importance measures for summarizing RDF/S KBs," in *Proc. 14th Int. Conf. Semantic Web*. Cham, Switzerland: Springer, 2017, pp. 387–403.
[43] G. Troullinou, H. Kondylakis, K. Stefanidis, and D. Plexousakis, "Exploring RDFS KBs using summaries," in *Proc. 17th Int. Semantic Web Conf.* Cham, Switzerland: Springer, 2018, pp. 268–284.
[44] V. Presutti, L. Aroyo, A. Adamou, B. Schopman, A. Gangemi, and G. Schreiber, "Extracting core knowledge from linked data," Citeseer, 2011.
[45] G. Wu, J. Li, L. Feng, and K. Wang, "Identifying potentially important concepts and relations in an ontology," in *Proc. 7th Int. Semantic Web Conf.* Berlin, Germany: Springer, 2008, pp. 33–49.
[46] S. Campinas, T. E. Perry, D. Ceccarelli, R. Delbru, and G. Tummarello, "Introducing RDF graph summary with application to assisted SPARQL formulation," in *Proc. 23rd Int. Workshop Database Expert Syst. Appl.*, Sep. 2012, pp. 261–266.
[47] A. K. Joshi, P. Hitzler, and G. Dong, "Logical linked data compression," in *Proc. 10th Int. Conf. Semantic Web*. Berlin, Germany: Springer, 2013, pp. 170–184.
[48] F. Goasdoué, P. Guzewicz, and I. Manolescu, "RDF graph summarization for first-sight structure discovery," *VLDB J.*, vol. 29, no. 5, pp. 1191–1218, Sep. 2020.
[49] G. Troullinou, H. Kondylakis, E. Daskalaki, and D. Plexousakis, "Ontology understanding without tears: The summarization approach," *Semantic Web*, vol. 8, no. 6, pp. 797–815, Aug. 2017.
[50] Š. Čebirić, F. Goasdoué, and I. Manolescu, "query-oriented summarization of RDF graphs," *Proc. VLDB Endowment*, vol. 8, no. 12, pp. 2012–2015, Aug. 2015.
[51] Y. Liu, G. Li, and L. Lin, "Cross-modal causal relational reasoning for event-level visual question answering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 10, pp. 11624–11641, Feb. 2023, doi: 10.1109/TPAMI.2023.3284038.
[52] H. Liu, H. Yuan, J. Hou, R. Hamzaoui, and W. Gao, "PUFA-GAN: A frequency-aware generative adversarial network for 3D point cloud upsampling," *IEEE Trans. Image Process.*, vol. 31, pp. 7389–7402, 2022, doi: 10.1109/TIP.2022.3222918.
[53] C. Mi, S. Huang, Y. Zhang, Z. Zhang, and O. Postolache, "Design and implementation of 3-D measurement method for container handling target," *J. Mar. Sci. Eng.*, vol. 10, no. 12, p. 1961, Dec. 2022, doi: 10.3390/jmse10121961.
[54] M. Yang, H. Wang, K. Hu, G. Yin, and Z. Wei, "IA-Net: An inception–attention-module-based network for classifying underwater images from others," *IEEE J. Ocean. Eng.*, vol. 47, no. 3, pp. 704–717, Jul. 2022, doi: 10.1109/JOE.2021.3126090.
[55] X. Zhou and L. Zhang, "SA-FPN: An effective feature pyramid network for crowded human detection," *Int. J. Speech Technol.*, vol. 52, no. 11, pp. 12556–12568, Sep. 2022, doi: 10.1007/s10489-021-03121-8.
[56] M. Zheng, K. Zhi, J. Zeng, C. Tian, and L. You, "A hybrid CNN for image denoising," *J. Artif. Intell. Technol.*, vol. 2, no. 3, pp. 93–99, Apr. 2022.
[57] E. Niazmand, G. Sejdiu, D. Graux, and M.-E. Vidal, "Efficient semantic summary graphs for querying large knowledge graphs," *Int. J. Inf. Manage. Data Insights*, vol. 2, no. 1, Apr. 2022, Art. no. 100082.
[58] M. Zneika, C. Lucchese, D. Vodislav, and D. Kotzinos, "RDF graph summarization based on approximate patterns," in *Proc. Int. Workshop Inf. Search, Integr., Personalization*. Cham, Switzerland: Springer, 2016, pp. 69–87.

[59] J. Dolby, A. Fokoue, A. Kalyanpur, A. Kershenbaum, E. Schonberg, K. Srinivas, and L. Ma, "Scalable semantic retrieval through summarization and refinement," in *Proc. AAAI*, vol. 7, 2007, pp. 299–304.
[60] A. Alzogbi and G. Lausen, "Similar structures inside RDF-graphs," in *Proc. LDOW*, 2013.
[61] A. Khan, S. S. Bhowmick, and F. Bonchi, "Summarizing static and dynamic big graphs," *Proc. VLDB Endowment*, vol. 10, no. 12, pp. 1981–1984, Aug. 2017.
[62] S. Khatchadourian and M. P. Consens, "Understanding billions of triples with usage summaries," in *Semantic Web Challenge*, 2011.
[63] E. Niazmand, G. Sejdiu, D. Graux, and M. E. Vidal, "Efficient semantic summary graphs for querying large knowledge graphs," *Int. J. Inf. Manag. Data Insights*, vol. 2, no. 1, 2022, Art. no. 100082.
[64] S. Pouriyeh, M. Allahyari, K. Kochut, G. Cheng, and H. R. Arabnia, "ES-LDA: Entity summarization using knowledge-based topic modeling," in *Proc. 8th Int. Joint Conf. Natural Lang. Process.*, 2017.
[65] N. Zhou, J. Sun, S. Zhou, Z. Bai, L. Lu, Q. Chen, and C. Zuo, "Transport of intensity diffraction tomography with non-interferometric synthetic aperture for three-dimensional label-free microscopy," *Light, Sci. Appl.*, vol. 11, no. 1, p. 154, 2022, doi: 10.1038/s41377-022-00815-7.
[66] Y. Li, J. Qian, S. Feng, Q. Chen, and C. Zuo, "Deep-learning-enabled dual-frequency composite fringe projection profilometry for single-shot absolute 3D shape measurement," *Opto-Electron. Adv.*, vol. 5, no. 5, 2022, Art. no. 210021, doi: 10.29026/oea.2022.210021.
[67] Z. Liu, C. Wen, Z. Su, S. Liu, J. Sun, W. Kong, and Z. Yang, "Emotion-semantic-aware dual contrastive learning for epistemic emotion identification of learner-generated reviews in MOOCs," *IEEE Trans. Neural Netw. Learn. Syst.*, 2023, doi: 10.1109/TNNLS.2023.3294636.
[68] S. Li, H. Chen, Y. Chen, Y. Xiong, and Z. Song, "Hybrid method with parallel-factor theory, a support vector machine, and particle filter optimization for intelligent machinery failure identification," *Machines*, vol. 11, no. 8, p. 837, Aug. 2023, doi: 10.3390/machines11080837.
[69] D. Diefenbach and A. Thalhammer, "Pagerank and generic entity summarization for Rdf knowledge bases," in *Proc. 15th Int. Conf. Semantic Web*. Cham, Switzerland: Springer, 2018.
[70] S. Pouriyeh, M. Allahyari, Q. Liu, G. Cheng, H. R. Arabnia, M. Atzori, F. G. Mohammadi, and K. Kochut, "Ontology summarization: Graph-based methods and beyond," *Int. J. Semantic Comput.*, vol. 13, no. 2, pp. 259–283, Jun. 2019.
[71] P. Guzewicz and I. Manolescu, "Parallel quotient summarization of RDF graphs," in *Proc. Int. Workshop Semantic Big Data*, Jul. 2019.
[72] G. Vassiliou, G. Troullinou, N. Papadakis, K. Stefanidis, E. Pitoura, and H. Kondylakis, "Coverage-based summaries for RDF KBs," in *Proc. Eur. Semantic Web Conf.* Cham, Switzerland: Springer, 2021.
[73] Y. Zhuang, N. Jiang, and Y. Xu, "Progressive distributed and parallel similarity retrieval of large CT image sequences in mobile telemedicine networks," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–13, Jul. 2022, doi: 10.1155/2022/6458350.
[74] F. Wang, H. Wang, X. Zhou, and R. Fu, "A driving fatigue feature detection method based on multifractal theory," *IEEE Sensors J.*, vol. 22, no. 19, pp. 19046–19059, Oct. 2022, doi: 10.1109/JSEN.2022.3201015.
[75] C. Zong and Z. Wan, "Container ship cell guide accuracy check technology based on improved 3D point cloud instance segmentation," *Brodogradnja*, vol. 73, no. 1, pp. 23–35, Mar. 2022, doi: 10.21278/brod73102.
[76] J. Xu, S. Pan, P. Z. H. Sun, S. H. Park, and K. Guo, "Human-factors-in-driving-loop: Driver identification and verification via a deep learning approach using psychological behavioral data," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3383–3394, 2022, doi: 10.1109/TITS.2022.3225782.
[77] J. Xu, K. Guo, and P. Z. H. Sun, "Driving performance under violations of traffic rules: Novice vs. experienced drivers," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 4, pp. 908–917, Dec. 2022, doi: 10.1109/TIV.2022.3200592.
[78] S. Lu, Y. Ban, X. Zhang, B. Yang, S. Liu, L. Yin, and W. Zheng, "Adaptive control of time delay teleoperation system with uncertain dynamics," *Frontiers Neurorobotics*, vol. 16, Jul. 2022, Art. no. 928863, doi: 10.3389/fnbot.2022.928863.
[79] X. Liang, Z. Huang, S. Yang, and L. Qiu, "Device-free motion & trajectory detection via RFID," *ACM Trans. Embedded Comput. Syst.*, vol. 17, no. 4, pp. 1–27, Jul. 2018, doi: 10.1145/3230644.
[80] C. Liu, T. Wu, Z. Li, T. Ma, and J. Huang, "Robust online tensor completion for IoT streaming data recovery," *IEEE Trans. Neural Netw. Learn. Syst.*, 2022, doi: 10.1109/TNNLS.2022.3165076.
[81] J. Liu, C. Fan, Y. Peng, J. Du, Z. Wang, and C. Chu, "Emergent leader-follower relationship in networked multiagent systems," *Sci. China Inf. Sci.*, vol. 66, no. 12, Dec. 2023, doi: 10.1007/s11432-022-3741-3.
[82] J. Chen, Q. Wang, W. Peng, H. Xu, X. Li, and W. Xu, "Disparity-based multiscale fusion network for transportation detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18855–18863, Oct. 2022, doi: 10.1109/TITS.2022.3161977.
[83] H. Liu, Y. Xu, and F. Chen, "Sketch2Photo: Synthesizing photo-realistic images from sketches via global contexts," *Eng. Appl. Artif. Intell.*, vol. 117, Jan. 2023, Art. no. 105608, doi: 10.1016/j.engappai.2022.105608.
[84] J. Liu, Y. Zhou, J. Lu, R. Cai, T. Zhao, Y. Chen, M. Zhang, X. Lu, and Y. Chen, "Injectable, tough and adhesive Zwitterionic hydrogels for 3D-printed wearable strain sensors," *Chem. Eng. J.*, vol. 475, Nov. 2023, Art. no. 146340, doi: 10.1016/j.cej.2023.146340.
[85] Y. Zheng, X. Lv, L. Qian, and X. Liu, "An optimal BP neural network track prediction method based on a GA–ACO hybrid algorithm," *J. Mar. Sci. Eng.*, vol. 10, no. 10, p. 1399, Sep. 2022, doi: 10.3390/jmse10101399.
[86] Y. Zheng, P. Liu, L. Qian, S. Qin, X. Liu, Y. Ma, and G. Cheng, "Recognition and depth estimation of ships based on binocular stereo vision," *J. Mar. Sci. Eng.*, vol. 10, no. 8, p. 1153, Aug. 2022, doi: 10.3390/jmse10081153.
[87] X. Hu, Q. Kuang, Q. Cai, Y. Xue, W. Zhou, and Y. Li, "A coherent pattern mining algorithm based on all contiguous column bicluster," *J. Artif. Intell. Technol.*, vol. 2, no. 3, pp. 80–92, May 2022.
[88] J. Luo, Y. Wang, and G. Li, "The innovation effect of administrative hierarchy on intercity connection: The machine learning of twin cities," *J. Innov. Knowl.*, vol. 8, no. 1, Jan. 2023, Art. no. 100293, doi: 10.1016/j.jik.2022.100293.
[89] H. Liu, H. Yuan, Q. Liu, J. Hou, H. Zeng, and S. Kwong, "A hybrid compression framework for color attributes of static 3D point clouds," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1564–1577, Mar. 2022, doi: 10.1109/TCSVT.2021.3069838.
[90] Y. Zhuang, S. Chen, N. Jiang, and H. Hu, "An effective WSSENet-based similarity retrieval method of large lung CT image databases," *KSII Trans. Internet Inf. Syst.*, vol. 16, no. 7, pp. 1–12, 2022, doi: 10.3837/tiis.2022.07.013.
[91] W. Zheng and L. Yin, "Characterization inference based on joint-optimization of multi-layer semantics and deep fusion matching network," *PeerJ Comput. Sci.*, vol. 8, p. e908, Apr. 2022, doi: 10.7717/peerj-cs.908.
[92] S. Zhang, T. Li, S. Hui, G. Li, Y. Liang, L. Yu, D. Jin, and Y. Li, "Deep transfer learning for city-scale cellular traffic generation through urban knowledge graph," Presented at the KDD, New York, NY, USA, 2023, doi: 10.1145/3580305.3599801.
[93] J. Meng, Y. Li, H. Liang, and Y. Ma, "Single image dehazing based on two-stream convolutional neural network," *J. Artif. Intell. Technol.*, vol. 2, no. 3, pp. 100–110, Jun. 2022.
[94] J. Zhang, G. Ye, Z. Tu, Y. Qin, Q. Qin, J. Zhang, and J. Liu, "A spatial attentive and temporal dilated (SATD) GCN for skeleton-based action recognition," *CAAI Trans. Intell. Technol.*, vol. 7, no. 1, pp. 46–55, Mar. 2022, doi: 10.1049/cit2.12012.
[95] F. S. Hassan and A. Gutub, "Improving data hiding within colour images using hue component of HSV colour space," *CAAI Trans. Intell. Technol.*, vol. 7, no. 1, pp. 56–68, Mar. 2022, doi: 10.1049/cit2.12053.

HAFIZ TAYYEB JAVED is currently pursuing the Ph.D. degree with the FAST National University of Computer and Emerging Sciences (FAST-NUCES), Islamabad, Pakistan. He is also an Assistant Professor with the Computer Science Department, FAST-NUCES, CFD Campus, Pakistan. His research interests include knowledge graphs, data mining, and machine learning.

KIFAYAT ULLAH KHAN received the Ph.D. degree from Kyung Hee University, South Korea, and the M.S. degree from the University of Greenwich, U.K. He is currently a Senior Lecturer with the Department of Accountancy, Finance and Economics, Birmingham City Business School, Birmingham City University, Birmingham, U.K. His research interests include artificial intelligence, FinTech, databases, and deep learning.

MUHAMMAD FAISAL CHEEMA received the Ph.D. degree from Leipzig University, Germany, and the M.S. degree from the National University of Computer and Emerging Sciences, Islamabad, Pakistan. He is currently an Assistant Professor in deep learning. His research interests include deep learning, computer vision, knowledge graphs, data mining, and data and information visualization.

ASAAD ALGARNI received the Ph.D. degree in software engineering from North Dakota State University, USA. He is currently an Assistant Professor with the Department of Computer Sciences, College of Computing and Information Technology, Northern Borders University, Saudi Arabia. His research interests include software engineering, computer vision applications, and machine learning.

JEONGMIN PARK received the Ph.D. degree from the College of Information and Communication Engineering, Sungkyunkwan University, South Korea, in 2009. He is currently an Associate Professor with the Department of Computer Engineering, Tech University of Korea, South Korea. Before joining the Tech University of Korea, in 2014, he was a Senior Researcher with the Electronics and Telecommunications Research Institute (ETRI) and a Research Professor with Sungkyunkwan University. His research interests include high reliable autonomic computing mechanism and human oriented interaction systems.