International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 5
ISSN: 2088-5334

The Formal Graph of APRDF

Dewi Wardani [a,*], Maria Ulfah [b], Ardhi Wijayanto [a], Yessi Yunitasari [c]

[a] *Informatics Department, Universitas Sebelas Maret, Jl. Ir Sutami 36 A, Surakarta, 57126, Indonesia*
[b] *Information Technology Department, UIN Sunan Kalijaga, Jl. Marsda Adisucipto, Yogyakarta, 55282, Indonesia*
[c] *Information Technology Department, Universitas PGRI Madiun, Jl. Setia Budi No. 85, Madiun, 63118, Indonesia*
*Corresponding author: *dww_ok@uns.ac.id*

***Abstract*— A new alternative model for expressing more complex knowledge has been proposed as an attributed predicate RDF (APRDF). By handling attributes that represent any additional triples of the main triple, APRDF serves as a predicate. Therefore, the formal graph model of APRDF must be defined. Lastly, this work recommends that the APRDF's conventional diagram is a digraph-hypergraph mix. The previous formal graph of RDF is a hypergraph even though, visually intuitively, it is a digraph. It also contains inconsistency. The other new serialization needs to describe its formal model. Eventually, this work can provide the formal graph model of APRDF and maintain consistency. There have been a few definitions proposed. The direct impact of this formal model is that APRDF outperformed the other model significantly when retrieving complex queries within its formal graph. In querying, the initial implementation of the proposed formal graph takes an average of 62 milliseconds. Compared to the other graph models, the proposed formal graph can reduce query time by an average of 90,7 milliseconds on the BF-arch graph and 121,05 milliseconds on the naive/default graph. As the formal graph model is preserved, the attributed predicate of APRDF assumed will drive a new model in the retrieving process that more in using a predicate formed as a link in a graph. It will also be impacted in the mining process by more elaborate links/edges (link mining).**

***Keywords*—APRDF; digraph; graph model; hypergraph; RDF.**

Manuscript received 27 Nov. 2022; revised 31 Dec. 2022; accepted 16 Feb. 2023. Date of publication 31 Oct. 2023.

I. INTRODUCTION

APRDF [1] is a new model for expressing knowledge on the Semantic Web, but it must investigate two aspects: performance and formal representation. The semantic interpretation of it and the other fundamental issues of formalizing the APRDF formal graph must be completed. APRDF, like standard RDF, is represented as a graph. As a result, the APRDF formal graph can be formalized. The APRDF formal graph helps with some works that require graph theoretical issues, such as data integration and graph mining. The advantage of using a graphing approach to compose APRDF was valuable in previous work.

The RDF graph model is explained in a couple of publications. These RDF serializations, like previous semantic interpretations of N-Quad [2] and Named Graph [3], lack formal graphs. As a result, delivering the formal graph to APRDF at the beginning of the project is crucial. To summarize, the purpose of this effort and the contributions is to define the APRDF formal graph. A recent work [4] used an ad-joint ontology graph approach to measure ontology similarity but did not explain the formal ontology graph. As suggested in [5] with Formal Concept Analysis (FCA), defining a graph's lattice with a formal graph is also beneficial. More work shows the importance of the formal model [6][7].

The graph-based model RDF has been utilized to enhance a variety of tasks. The graph of an instance of RDF can be utilized to classify the sentience of review products, according to Santosh [8]. Using the graph concept to store, index and query vast volumes of provenance data was also advantageous, according to Chebot [9]. According to Kermani [10], the idea of a graph permits dynamically created protein ontologies. It is well understood that RDF is intuitively a graph. Despite this, only a few works explain its formal graph. Hayes et al. [11] that the formal graph can be derived as a bipartite graph as one of its principal works. Like the well-known work providing the formal graph, this work is more argumentation than experimentation. The other new model, [12] [13], does not include a formal graph. Providing a formal graph has also been introduced in several works. Wang et al. [14] define a formal graph to handle bijection functions to find edge magic within Unicyclic Graphs using a generating algorithm. To overcome the increasing unnecessary membership of the network in the antichain graph, the formal model of the bipartite graph was introduced [15]. To solve the matching problem, the bipartite graph is also formalized. Its formal model is based on a graph in the biology research area [16]. Because the vague graph is necessary, the theoretical definition of it in the telecommunication issue has been too [17]. Those works relied on a bipartite graph. In the meantime, our APRDF is more akin to a property hypergraph. Although [18] is not a hypergraph. Its property graph to unify data is similar to this work.

Several previous works show the importance of the formal graph model of RDF. The formal graph was proven needed in the retrieval process [19] [20] and valuable in storage management. Ontology as graph uses a graph model for ontology matching [21] [22], an essential step in data integration. Although it only works for simple graphs, the formal graph as a bipartite graph can represent the layer of the network [23]. A formal graph helps the ranking problem for a few definitions in work [24] by Hayes et al. [10]. The mapping from heterogeneous resources to a knowledge graph needs the graph model, too [25]. A formal graph is required to store complexity caused by reification and provenance [26]. The other work proposed the formal graph intuitively as a hypergraph [27], [28] instead of a digraph or bipartite graph, although it was for the same model, RDF. It is more complicated.

The recent work has added the semantic interpretation of APRDF [1]. The previous work on semantic interpretation only shows default RDF that consists of three elements. Further needed is the formal graph of APRDF. Therefore, this work provides it, as well as the main contribution of this work. This work will consider hypergraphs as well to formalize the APRDF. APRDF is more complex than default RDF. Therefore, considering hypergraph is worth equally instead of using hypergraph to formalize default RDF.

II. Material and Method

A. APRDF as Graph

An example, *"An author Jack wrote a new book, titled The Queen. CNN reported that the writing process was in London. The project's budget was USD100M. This information was submitted by Anna. The book project was led by Lina, begin on 2020-07-01 finished on 2020-12-21 then was led by Lena for 2 weeks. Jack wrote too the other book titled The King".* This example can be represented as APRDF, and its graph intuitively as below (author.ttl without prefix):

(e1) ex:Jack ex:write_1 ex:The Queen .
(e2) ex:write_1 ex:city_1 ex:London .
(e3) ex:write_1 ex:hasCost "US$100M" .
(e4) ex:write_1 ex:chief_1 ex:Anna .
(e5) ex:write_1 ex:chief_2 ex:Lena .
(e6) ex:write_1 ex:news_1 ex:CNN .
(e7) ex:chief_1 ex:beginDate "2020-0701"^^xsd:date .
(e8) ex:chief_1 ex:stopDate "2020-1221"^^xsd:date .
(e9) ex:chief_2 ex:officeTime "2"^^xsd:integer .
(e10) ex:news_1 ex:reportBy ex:Anna .
(e11) ex:Jack ex:write ex:The King .
ex:write_1 rdf:type ex:write .
ex:city_1 rdf:type ex:city

As that turtle (.ttl) is the serialization of the default format RDF/XML, including in using XML Schema Definition (XSD) data type e.g., date integer etc. The writing style of data type must start with "^^" in RDF.

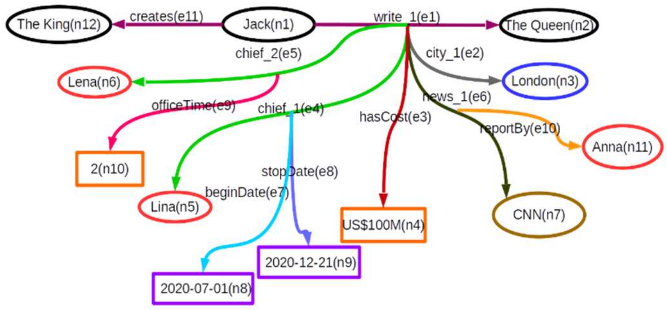

Fig. 1 The illustration of the graph of the author.ttl

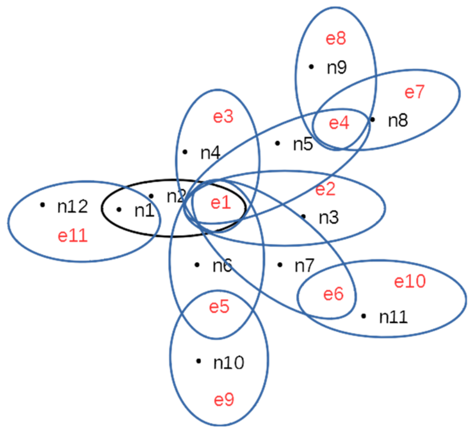

Fig. 2 The other illustration of the graph of the author.ttl

Graphs in Figures 1 and Figure 2 can be explained in a graph notation as below:

- V be a set of vertices, $V = \{v1 ,...,vx \}$. x is the number of node. In the case for author.ttl, $V =\{v1 ,.., v12 \}$.
- E be a set of edges, $E = \{e1 ,...,ey \}$. y is the number of edge. The set of edges of author.ttl is $E =\{e1 ,.., e11 \} = \{\{v1 ,v2 \}, \{\{v1 ,v2 \},n3 \}, \{\{e1 ,v3 \},n4 \}, \{\{e1 ,v3 \},n5 \}, \{\{e1 ,v3 \},n6 \}, \{\{e2 ,v5 \},n8 \}, \{\{e2 ,v5 \},n9 \}, \{\{e2 ,v6 \},n10 \}, \{\{e2 ,v7 \},n11 \}, \{\{v1 ,v12 \}\}$
- APRDF utilizes the design that end-point of edges can also be an edge [29]. A set of hyperedges: *"write_1"*, *"city_1"*, *"chief_1"*, *"chief_2"*, *"news_1"* and edges: *"hasCost"*, *"beginDate"*, *"stopDate"*, *"writeBy"*, *"officeTime"* and *"write"*.

The predicate *"write_1"* is an instance of *"write"* (predicate type). Instances are shaped at edges that own attributes. Similar to edges in property graphs, predicates may hold attributes in the APRDF model. Meanwhile, the predicate type is shaped if there are no attributes. The vertices with the arrow denote the pointed vertex as an object in the APRDF triple. It can be seen that we obtain two digraphs *{v1, v2}* and *{v1, v12}*. The others are hypergraphs. The formal graph of APRDF can be shaped as a digraph or hypergraph. They are based on the predicate type that is shaped by the edges.

B. The Formal Graph of APRDF.

Figure 2 depicts the above-mentioned directed property hypergraph explanation. Meanwhile, RDF in APRDF takes the form of a directed digraph. It explains how APRDF can be used to create both digraphs and hypergraphs. According to Hayes et al. [10], triples are hypergraphs. It should be noted that they are digraphs in reality. According to the proposed definitions, a digraph, "ex:Jack ex:write ex:The King" is hypergraph shaped.

Figure 2 depicts APRDF as digraphs *D* hypergraphs *H*. The vertices of an author.ttl triple is *"ex:Jack"* and *"ex:The King"*, and the edge is *"ex:write"*. Figure 1 shows how the predicate type *"ex:write"* can be shaped as a digraph edge. These circumstances inspired the proposed method for defining the APRDF formal graph.

The approach formalizes the graph model like the illustration, both as a hypergraph and a digraph. Figure 1 shows edge *"write_1"* is a hypergraph and edge *"hasCost"* is a digraph. This approach utilizes both edges and vertices to express P. Meanwhile, Hayes et al. [10] only utilize vertex to express P. It inconsistently defines that a predicate is shaped as a vertex and an edge in the same RDF dataset [11]. Figure 3 shows term "paints" are shaped in two forms: an edge and a vertex.

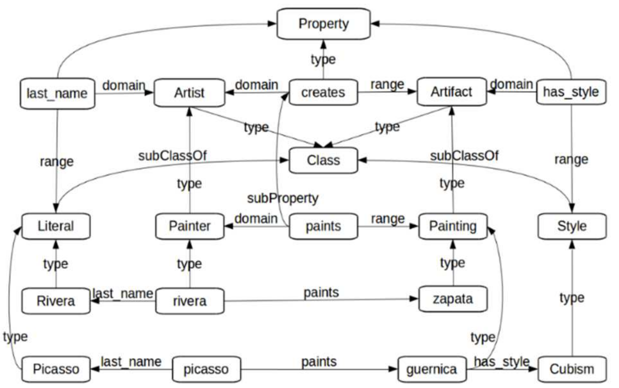

Fig. 3 The explanation of RDF inconsistency in [9]

This approach defines each element S, P and O in the same form as a vertex. It uses a surrogate vertex to hold all elements. Edge expresses the role of each vertex held by the surrogate vertices. The limitation of this approach is that the nature of the semantics of RDF is decreased. In the real world, predicates relate to two vertices. In default, a predicate shaped as an edge becomes vertices in this approach.

The summarization of the limitation of the current work in defining a formal graph of RDF as below:

- The use of different forms for the same element
- RDF is digraph-like. Unfortunately, inconsistency is created because the formal model only concerns the form of a digraph.
- As the solution to the first and second limitations, RDF is seen as a total hypergraph. It is a contradiction because RDF is digraph-like.
- The solution that sees RDF as the default hypergraph in the bipartite graph model has an impact in losing the semantic of the relationship (link), represented by the predicate. Therefore, in default, the predicate is formed as a node.

The proposed definition uses edges and vertices to express a predicate. The difference compared to the previous ones is that each predicate has a single presentation format in an RDF dataset. It is different from the note of the existing work that uses both forms for a single element. For example, a predicate *"hasCost"* is shaped as an edge in the entire APRDF. Then, a predicate *"write_1"* is shaped as a vertex in the entire APRDF. Therefore, the definition is consistent. The explanation is excluded from the inconsistency of this work [11]. The instances of the predicate in APRDF cause the formation of a hypergraph to be triggered. The instance is used to handle the initial triple attributes. Examples of the predicate are the various types of the predicate. The predicate type is also hypergraph-shaped as a result. Digraphs are a type of predicate without predicate instances. Gallo [15] proposed hypergraphs as connecting to multiple vertices or connecting to edges. As a result, APRDF is not only shaped as a digraph but also as a hypergraph. Each form is used under specific APRDF conditions. This feature is used in this approach. The following definitions apply to the proposed formal graph:

CONJECTURE 1. (Digraph and Hypergraph)
Triples of APRDF are hypergraphs formed or digraphs. APRDF's predicate types and instances are shaped like hypergraphs. Digraphs are shaped types of predicate types that do not have instances. *G* is a graph of the APRDF triple that contains the hypergraphs *H* and *D*, so $G = (H \cup D)$

CONJECTURE 2. (A digraph of APRDF)
D is digraphs in APRDF and ARDF is a set of triple RDF in APRDF. $D(ARDF) = (V, E, \wp)$, vertices $V = \{S, O\}$ and edges $E = \{P\ T\}$. $\wp: V x E \rightarrow \{s, p, o\}$ is a role of vertices that fulfill:

- $(\wp\ (V, E) = s) \Leftrightarrow (V \in startOf\ (E))$
- $p \Leftrightarrow E$
- $(\wp\ (V, E) = o) \Leftrightarrow (V \in stopOf\ (E))$.

CONJECTURE 3. (A directed hypergraph of APRDF)
H be hypergraphs in APRDF and TAPRDF be a set of APRDF triples. $H(TAPRDF) = (V, E, \wp)$ where vertices $V = \{S, P, O\}$ and edges $E = \{PT, PIT\}$. $\wp: NxE \rightarrow \{s, p, o\}$ is a role function of vertices that fulfill:

- $((\wp\ (V, E) = s) \Leftrightarrow (V \in startOf\ (E) \wedge (V \in S))$.
- $(\wp\ (V, E) = p) \Leftrightarrow (V \in startOf\ (E) \wedge (V \in P))$.
- $(\wp\ (V, E) = o) \Leftrightarrow (V \in stopOf\ (E) \wedge (V \in O))$.

III. RESULTS AND DISCUSSION

A. The Implementation of Formal Graph of APRDF

The implementation of this approach is shown in Figure 4. It can be explained as below:

- The predicate type *"beginDate"*, *"stopDate"*, *"reportedBy"* and *"officeTime"* are digraph shaped.
- Predicate type *"write"*, *"news"*, *"chief"* and *"city"* are hypergraphs shaped. They own instances that refer to those predicates.
- Predicate *"write_1"* is a predicate type's instance *"write"* and *"city_1"* is predicate type's instance *"city"* etc. They are also hypergraphs shaped.
- Predicate *"beginDate"*, *"stopDate"*, *"type"* and *"news"* are shaped as an edge in entire APRDF graph.

- Predicates *"write", "city", "write_1"* and *"city_1"* are shaped as a vertex in entire APRDF graph.

It has been seen that a predicate can be shaped as a vertex or an edge. Even though it can be shaped in two different shapes, the shapes are consistent for the entire APRDF graph. Hence, it is defined as consistent. The illustration of the author.ttl based on Hayes et al.'s approach is shown in Figure 5 as a comparison [10]. To complete the explanation of the formal graph model, the implementation of the formal graph on RDF is illustrated in Figure 6. By using the RDF of example.ttl in Figure 3, as below (in turtle format, without prefix):

:picasso	*:last_name*	*:Picasso .*
:Picasso	*:type*	*:Literal .*
:Rivera	*:type*	*:Literal .*
:Literal	*:subClassOf*	*:Class .*
:last_name	*:range*	*:Literal .*
:last_name	*:domain*	*:Artist .*
:rivera	*:type*	*:Painter .*
:Painter	*type*	*:Artist .*
:rivera	*:paints*	*:zapata .*
:zapata	*:type*	*:Painting .*
:Painting	*:type*	*:Artifact .*
:paints	*:range*	*:Painting.*
:Artist	*:type*	*:Class .*
:Artifact	*:type*	*:Class .*
:paints	*:subProperty*	*:creates .*
:creates	*:type*	*:Property .*
:creates	*:domain*	*:Artist .*
:creates	*:range*	*:Artifact .*
:last_name	*:type*	*:Property .*
:has_style	*:type*	*:Property .*
:has_style	*:range*	*:Style .*
:Style	*:subClassOf*	*:Class*
:Cubism	*:type*	*:Style .*
:picasso	*:paints*	*:guernica .*
:guernica	*:has_style*	*:Cubism .*
:guernica	*:type*	*:Painting .*

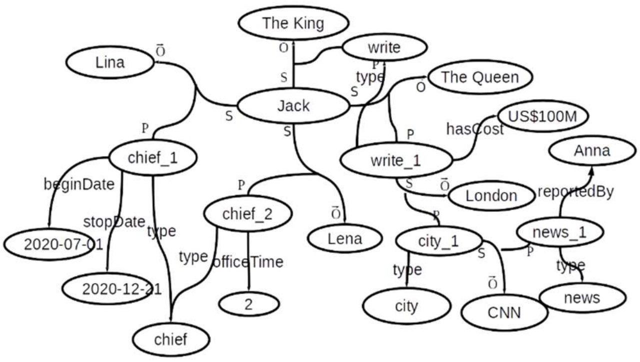

Fig. 4 The proposed definition in illustration

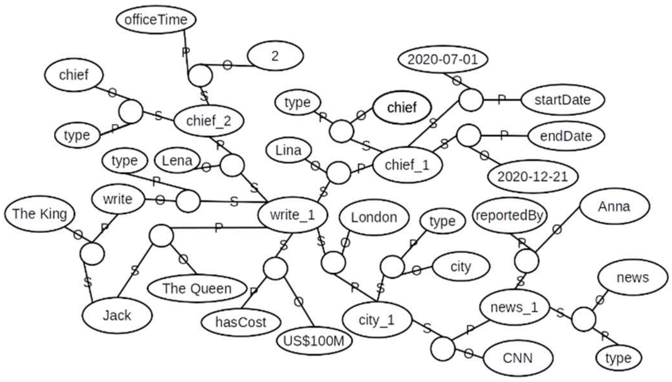

Fig. 5 The illustration of formal graph of author.ttl considering on Hayes et al. [10]

For simplicity and a more precise figure, the example.ttl only uses a part of RDF, which shows inconsistency.

:picasso	*:paints*	*:guernica .*
:rivera	*:paints*	*:zapata .*
:paints	*:domain*	*:Painter .*
:paints	*:range*	*:Painting.*
:paints	*:type*	*:Property .*
:paints	*:subPropertyOf*	*:creates .*
:rivera	*:type*	*:Painter .*
:zapata	*:type*	*:Painting .*

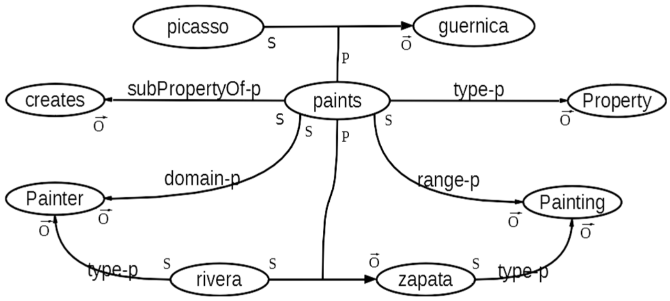

Fig. 6 The formal graph illustration of example.ttl based on the proposed definition.

Meanwhile, based on the approach of Hayes et al. [10], example.ttl is illustrated in Figure 7. The illustration based on Hayes et al. [10] in Figure 5 and Figure 7 represents all properties as a node. The term "type" is a predicate that relates the subject and predicate in the real world, which connects one to others [30]–[32]. The semantic of a predicate is a relationship [33]. In default, this semantic disappears because they are formed as a node. The same condition for the reserved term "domain", "range", etc. "paints" is formed as a node as well.

The illustration in Figures 4 and 6, which is based on the proposed graph model, keeps *"type", "domain"*, and *"range"* as a predicate which is formed as a link instead of a node. Those predicates are utilized in digraphs. "paints" is a predicate formed as a node and is utilized in the hypergraph. It is easier to notice that if a predicate is formed as a link, it is a part of a digraph. However, if a predicate is formed as a node, it is a part of the hypergraph. The information that a node represents a predicate will be noticed by its role.

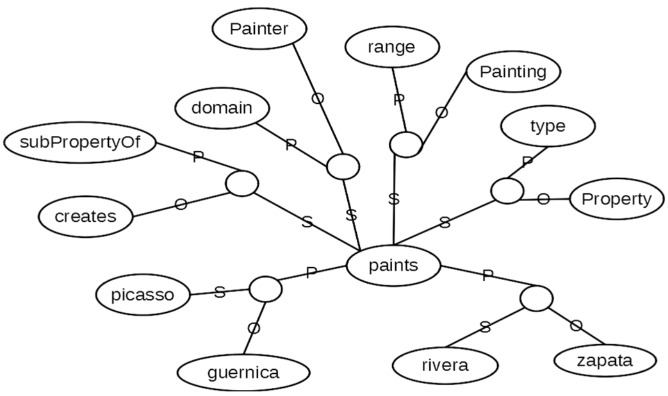

Fig. 7 The formal graph illustration of example.ttl based on the definition of Hayes et al [10]

B. The querying performance of Graph APRDF

In this experiment, we focus on examining the impact of property links/hyperedges in obtaining graph models on querying performance. Two real datasets are used in this experiment:

- The IMDB dataset is related to movies.
- The Mondial, the dataset is related to the country's landscape.

All the datasets were converted from a relational model into three graph models: the default/ naive model, BF-arch and the graph of APRDF. Table 1 describes the obtained mapping results of each dataset. Especially, APRDF is converted based on semantics [1] that has been evaluated, and the result can keep the semantics of data [34]. Figure 8 and Figure 9 are examples of a network formed as a consequence of Semantic Mapping on the IMDB and Mondial datasets. Figure 8 shows that the graph is identical to the APRDF Graph in Figure 1. Especially for the link *"aka_title"* and *"movie_link"*. All graph models were implemented in the Neo4j engine.

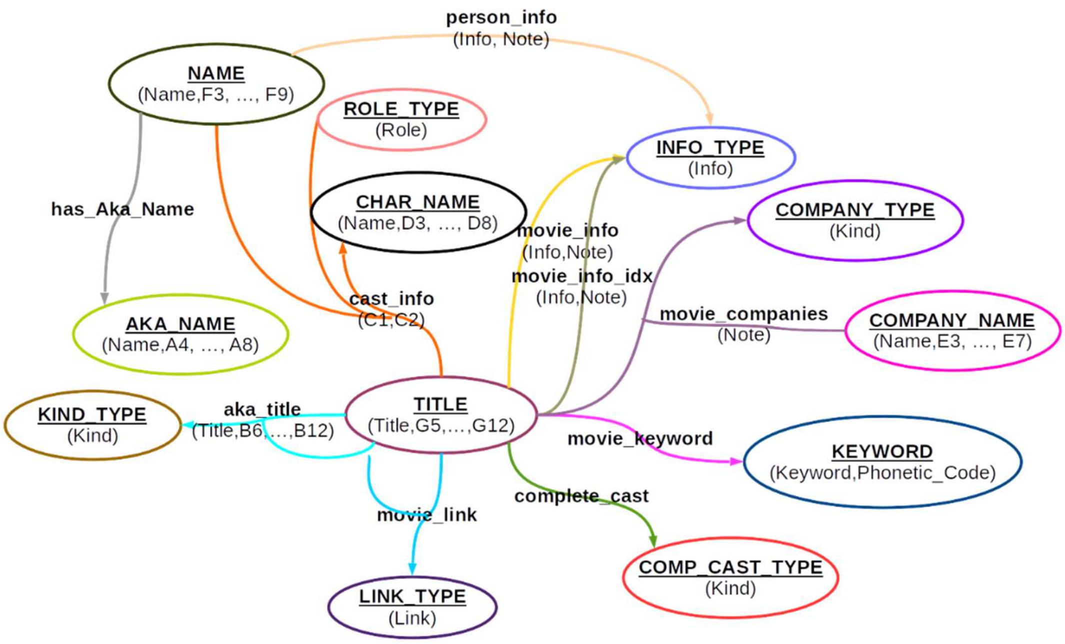

Fig. 8 The illustration of the result of APRDF Property Hypergraph on IMDB dataset

Therefore, all queries were written in Cypher. Each dataset contains ten queries. Table 2 shows an example of queries of both datasets. The APRDF formal graph model shows the direction of the link. Therefore, it is an advantage in the retrieval process.

TABLE I
THE DESCRIPTION OF OBTAINED THREE GRAPH MODELS

The dataset	Naive Model	BF-arch Hypergraph	APRDF Property Hypergraph
The IMDB	19 nodes, 28 links	12 original nodes, 2 BF-arc nodes 8 links, 1 pseudo-link	12 nodes, 8 links, 2 link property hypergraphs
The Mondial	26 nodes, 50 links	17 original nodes, 6 BF-arc nodes 13 links, 1 pseudo-link	16 nodes, 19 links, 6 link property hypergraphs

As a result, shown in Figure 10 and Figure 11, APRDF surpasses the other graph models. On average, the APRDF property hypergraph returns *81 ms* for IMDB queries, and *44,6 ms* for Mondial queries. It is followed by BF-arch, which returns an average of *115,9 ms* and *65,5 ms* for IMDB and Mondial, respectively. The weak performance is shown in the naive model with an average of *152 ms* for IMDB queries, and *90,1 ms* for Mondial queries.

TABLE II
THE EXAMPLE QUERIES

The dataset	The example query
The IMDB	The name of the firm associated with all movies that has a type as <t>. In addition, actors has alias <a>.
The Mondial	List of all nations and provinces bordering the Black Sea, as well as the rivers from the sea <s>that flow through those parts of the continent <ct>

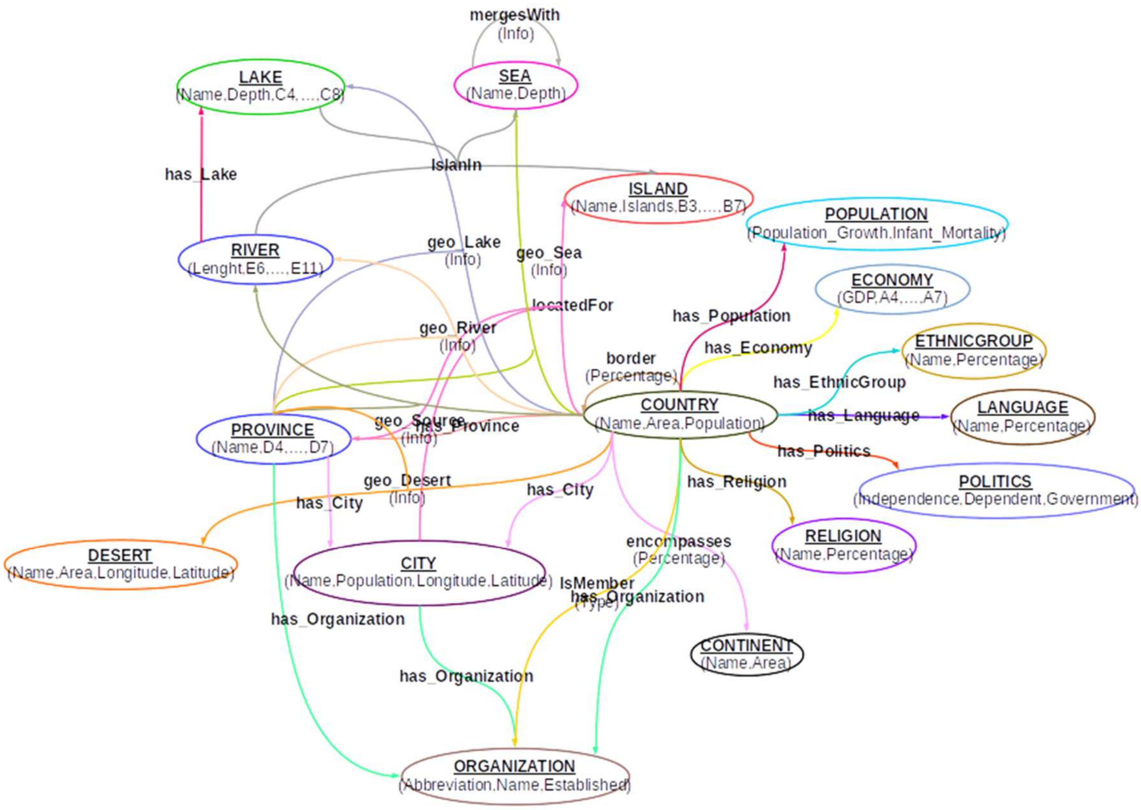

Fig. 9 The illustration of the result of APRDF Property Hypergraph on Mondial dataset

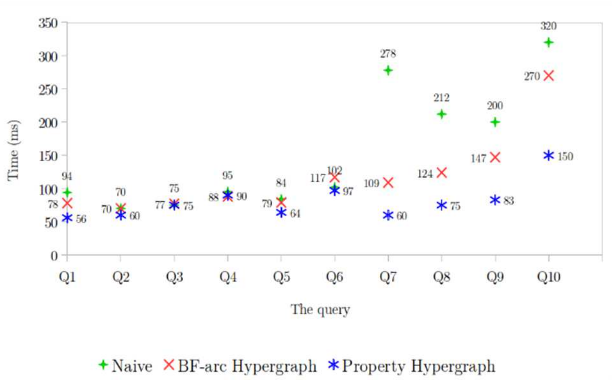

Fig. 10 The time's execution (in second) for IMDB dataset's queries

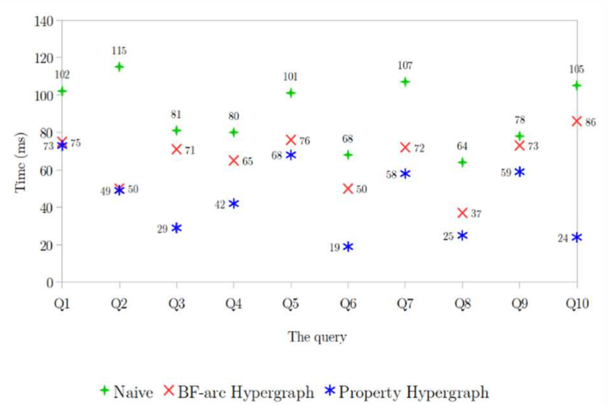

Fig. 11 The time's execution (in second) for Mondial dataset's queries

Table III shows the detail of query time on each graph model. The proposed formal graph model returns a much better number than the other models.

TABLE III
THE OBTAINED QUERY TIME

The dataset	Naive Model	BF-arch Hypergraph	APRDF Property Hypergraph
The IMDB	*152 ms*	*115,9 ms*	*81 ms*
The Mondial	*90,1 ms*	*65,5 ms*	*44,6 ms*

IV. CONCLUSION

We had previously assumed that the formal graph and the formal model of APRDF were connected. The formal graph model of APRDF has been proposed in this work. Additionally, it covers the default RDF. The graph approach aids comprehension of complex data, as previous research has demonstrated. Additional extensions of this work would be helpful in some services if an APRDF could be shaped like a graph. The benefits of this proper model affect the questioning presentation. In terms of performance, APRDF performs better than other models. The APRDF property hypergraph performs much more quickly than the average score. In the interim, different exhibitions were either indistinguishable from the typical score or essentially longer. The middle-level work in the future will look into whether or not this model can help steer information retrieval in a new direction. The retrieval procedure in triples APRDF places a greater emphasis on the predicate. We will evaluate its applicability in the subsequent middle layer, particularly in real-world application reasoning processes. Future research will focus on how the proposed ideas affect extended tasks like edge or graph mining.

ACKNOWLEDGMENT

We thank Universitas Sebelas Maret for funding this this research.

REFERENCES

[1] D. Wardani, "Complete W3C-Semantic's Interpretations of AP-RDF.," IAENG International Journal of Computer Science, vol. 49, no. 3, 2022

[2] G. Carothers, "RDF 1.1 N-Quads: A line-based syntax for RDF datasets," W3C Recomm., 2014.

[3] J. J. Carroll, C. Bizer, P. Hayes, and P. Stickler, "Named Graphs, Provenance and Trust," in Proceedings of the 14th International Conference on World Wide Web, New York, NY, USA, 2005, pp. 613–622.

[4] J. Tang, Y. Pan, Z. Wang, and L. Zhu, "Ontology Optimization Algorithm for Similarity Measuring and Ontology Mapping Using Adjoint Graph Framework.," Eng. Lett., vol. 26, no. 3, 2018.

[5] L. González and A. Hogan, "Modelling dynamics in semantic web knowledge graphs with formal concept analysis," in Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1175–1184.

[6] P. Monnin, M. Lezoche, A. Napoli, and A. Coulet, "Using formal concept analysis for checking the structure of an ontology in LOD: Example of DBpedia," in International Symposium on Methodologies for Intelligent Systems, 2017, pp. 674–683.

[7] M. Á. Rodríguez-García and R. Hoehndorf, "Inferring ontology graph structures using OWL reasoning," BMC Bioinformatics, vol. 19, no. 1, p. 7, 2018.

[8] D. T. Santosh and B. V. Vardhan, "Feature and sentiment based edgeed instance rdf data towards ontology based review categorization," in Proceedings of the World Congress on Engineering, 2015, vol. 1. A. Chebotko, J. Abraham, P. Brazier, A. Piazza, A. Kashlev, and S. Lu, "Storing, Indexing and Querying Large Provenance Data Sets as RDF Graphs in Apache HBase," in Services (SERVICES), 2013 IEEE Ninth World Congress on, 2013, pp. 1–8, doi: 10.1109/SERVICES.2013.32.

[9] M. H. Kermani, Z. Guessoum, and Z. Boufaida, "A Two-Step Methodology for Dynamic Construction of a Protein Ontology.," IAENG Int. J. Comput. Sci., vol. 46, no. 1, 2019.

[10] J. Hayes and C. Gutierrez, "Bipartite Graphs as Intermediate Model for RDF," in The Semantic Web – ISWC 2004, vol. 3298, SheilaA. McIlraith, D. Plexousakis, and F. van Harmelen, Eds. Springer Berlin Heidelberg, 2004, pp. 47–61.

[11] R. Soussi, "SPIDER-Graph: A Model for Heterogeneous Graphs Extracted from a Relational Database," in Conceptual Modeling, vol. 7532, P. Atzeni, D. Cheung, and S. Ram, Eds. Springer Berlin Heidelberg, 2012, pp. 543–552. S. Das, J. Srinivasan, M. Perry, E. I. Chong, and J. Banerjee, "A Tale of Two Graphs: Property Graphs as RDF in Oracle.," in EDBT, 2014, pp. 762–773.

[12] B. Wang and J. Li, "Edge-magic Total Labeling Algorithm of Unicyclic Graphs," structure, vol. 2, p. 2, 2021. K. A. Bhat, G. Sudhakara, and others, "Antichain Graphs.," IAENG International Journal of Applied Mathematics, vol. 51, no. 3, 2021.

[13] C. Qi and J. Diao, "A Bio-Inspired Algorithm for Maximum Matching in Bipartite Graphs.," IAENG International Journal of Computer Science, vol. 47, no. 1, 2020.

[14] L. Lin, M. Zhang, and L. Ma, "Solving the Telecommunication Network Problem using Vague Graph.," Engineering Letters, vol. 28, no. 4, 2020.

[15] P. Kaur and P. Nand, "Towards Transparent Governance by Unifying Open Data.," IAENG International Journal of Computer Science, vol. 48, no. 4, 2021.

[16] G. Gallo, G. Longo, S. Pallottino, and S. Nguyen, "Directed hypergraphs and applications," Discrete Appl. Math., vol. 42, no. 2–3, pp. 177–201, 1993.

[17] Y. Sun and J. Han, "Mining Heterogeneous Information Networks: A Structural Analysis Approach," SIGKDD Explor Newsl, vol. 14, no. 2, pp. 20–28, Apr. 2013.

[18] Y. Li, C. Shi, PhilipS. Yu, and Q. Chen, "HRank: A Path Based Ranking Method in Heterogeneous Information Network," in Web-Age Information Management, vol. 8485, F. Li, G. Li, S. Hwang, B. Yao, and Z. Zhang, Eds. Springer International Publishing, 2014, pp. 553–565.

[19] Z. Zheng, Y. Ding, Z. Wang, and Z. Wang, "A novel method of keyword query for RDF data based on bipartite graph," in 2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS), 2016, pp. 466–473.

[20] L. Nagy, T. Ruppert, and J. Abonyi, "Ontology-Based Analysis of Manufacturing Processes: Lessons Learned from the Case Study of Wire Harness Production," Complexity, vol. 2021.

[21] L. Zhao, Z. Liu, and J. Mbachu, "Highway alignment optimization: an integrated BIM and GIS approach," ISPRS International Journal of Geo-Information, vol. 8, no. 4, p. 172, 2019.

[22] Y. Li, Z. Jianhui, J. Liu, and Y. Hou, "Matching large scale ontologies based on filter and verification," Mathematical Problems in Engineering, vol. 2020.

[23] G. Honti and J. Abonyi, "Frequent Itemset Mining and Multi-Layer Network-Based Analysis of RDF Databases," Mathematics, vol. 9, no. 4, p. 450, 2021.

[24] A. A. Desouki, M. Röder, and A.-C. Ngonga Ngomo, "Ranking on Very Large Knowledge Graphs," in Proceedings of the 30th ACM Conference on Hypertext and Social Media, 2019, pp. 163–171.

[25] J. Rouces, G. De Melo, and K. Hose, "Addressing structural and linguistic heterogeneity in the Web," AI Communications, vol. 31, no. 1, pp. 3–18, 2018.

[26] B. Makni and J. Hendler, "Deep learning for noise-tolerant RDFS reasoning," Semantic Web, vol. 10, no. 5, pp. 823–862, 2019.

[27] H. Liu, D. Dou, R. Jin, P. Lependu, and N. Shah, "Mining Biomedical Ontologies and Data Using RDF Hypergraphs," in Machine Learning and Applications (ICMLA), 2013 12th International Conference on, Dec. 2013, vol. 1, pp. 141–146. doi: 10.1109/ICMLA.2013.31.

[28] G. Wu, J.-Z. Li, J.-Q. Hu, and K.-H. Wang, "System π: A native RDF repository based on the hypergraph representation for RDF data model," Journal of Computer Science and Technology, vol. 24, no. 4, pp. 652–664, 2009.

[29] G. Gallo, G. Longo, S. Pallottino, and S. Nguyen, "Directed hypergraphs and applications," Discrete Appl. Math., vol. 42, no. 2–3,pp. 177–201, 1993.

[30] Y. Sun and J. Han, "Mining Heterogeneous Information Networks: A Structural Analysis Approach," SIGKDD Explor Newsl, vol.14, no. 2, pp. 20–28, Apr. 2013.

[31] Y. Li, C. Shi, PhilipS. Yu, and Q. Chen, "HRank: A Path Based Ranking Method in Heterogeneous Information Network," in WebAge Information Management, vol. 8485, F. Li, G. Li, S. Hwang, B. Yao, and Z. Zhang, Eds. Springer International Publishing, 2014, pp. 553–565.

[32] Y. Zhou and L. Liu, "Activity-edge Centric Multi-label Classification for Mining Heterogeneous Information Networks," in Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 2014, pp. 1276–1285.

[33] P. Hayes and P. F. Patel-Schneider, "RDF 1.1 Semantics," W3C Recomm., vol. 25, 2014.

[34] D. Wardani, "The Evaluation of Semantic Mapping," Journal of Physics: Conference Series, vol. 1500, p. 012101, Apr. 2020, doi:10.1088/1742-6596/1500/1/012101.