VGStore: A Multimodal Extension to SPARQL for Querying RDF Scene Graph

Yanzeng Li[1], Zilong Zheng[2], Wenjuan Han[2] and Lei Zou[1,*]

[1]Wangxuan Institute of Computer Technology (WICT), Peking University, China
[2]Beijing Institute for General Artificial Intelligence (BIGAI), Beijing, China

Abstract

Semantic Web technology has successfully facilitated many RDF models with rich data representation methods. It also has the potential ability to represent and store multimodal knowledge bases such as multimodal scene graphs. However, most existing query languages, especially SPARQL, barely explore the implicit multimodal relationships like semantic similarity, spatial relations, etc. We first explored this issue by organizing a large-scale scene graph dataset, namely Visual Genome, in the RDF graph database. Based on the proposed RDF-stored multimodal scene graph, we extended SPARQL queries to answer questions containing relational reasoning about color, spatial, etc. Further demo (i.e., `VGStore`) shows the effectiveness of customized queries and displaying multimodal data.

Keywords

SPARQL, RDF, Multimodal, KBQA

1. Introduction

Over the recent years, we have witnessed an explosive growing trend on multimodal models due to the increasing computing power and massive multimodal datasets. Despite of inspiring performance that keeps updating on various multimodal benchmarks, the interpretability and reasonability have recently been challenged by researchers, namely, models are memorizing multimodal statistical features rather than understanding the joint information among them. For example, Visual Question Answering (VQA), a representative multimodal testbed for vision understanding, requires model to reason over images and answer questions. However, the current mainstream models still depend on end-to-end training by fitting input signals to ground truth answers, while neglecting the underlying visual relations and semantics.

To address these issues, we leverage an intrinsically explainable task, Knowledge Base Question Answering (KBQA), which aims to answer Natural Language Questions by referring to external Knowledge Base (KB). Semantic Parsing (SP)-based KBQA is a mainstream technique to solve the such QA problem via parsing a question into a KB query (such as SPARQL) [1]. The latest works are devoted to improving the performance of natural language understanding and parsing, while neglecting the expressiveness of KB queries, limiting the application of SP-based

ISWC'22: The 21st International Semantic Web Conference, October 23–27, 2022, Hangzhou, China
*Corresponding author.
liyanzeng@stu.pku.edu.cn (Y. Li); zlzheng@bigai.ai (Z. Zheng); hanwenjuan@bigai.ai (W. Han); zoulei@pku.edu.cn (L. Zou)

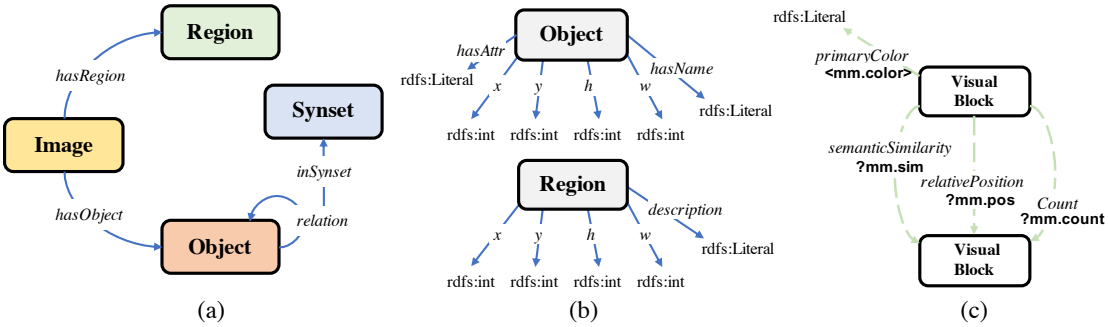

Figure 1: (a) The example of relations between classes in the ontology of visual genome. (b) The example of data properties of the class. (c) The implicit semantic relations between visual blocks.

methods in multimodal datasets. Although RDF has sufficient representation capability to describe multimodal data, the lack of multimodal semantic relationships in standard SPARQL has become a major challenge in applying SP-based KBQA methods to multimodal domains. Researchers have attempted to extend SPARQL for this purpose. For example, SPARQL-MM [2] proposed to use custom aggregation functions to access media fragments. However, previous works still suffer from limitations in extensibility and vague semantic representation, resulting in rare applications. Specifically, the custom aggregation functions introduced in SPARQL-MM are easy to understand by human beings but hard to understand and be expressed by the machine. This is because it brings significantly extra complexity to the query statement, e.g., if multimodal query statements in SPARQL-MM are used as query conditions, it inevitably leads to the union or nested query; however, the SP-based methods only support simple queries in the foreseeable future.

In this demo, we designed an ontology to organize the multimodal scene graph and store it with RDF. Furthermore, we implemented semantic multimodal SPARQL queries by extending the SPARQL engine, enabling the ability to answer questions related to multimodal information such as visual and spatial reasoning.

2. Storing Visual Genome with RDF

Visual Genome (VG) [3] is a large-scale dataset for fine-grained scene graphs, with rich annotations of images, regions, objects, as well as their relations[1]. A synset from WordNet [4] is introduced to link different scene graphs via the lexical relations between literals of the object relations. In addition, VG provided 1,445,332 relevant questions for 108,077 images, which are difficult to be answered by traditional SP-based KBQA methods because the SPARQL engine does not support the arithmetic opeartions needed to answer these multimodal questions.

For querying convenience, we formalize the elements of VG in RDF. Fig. 1(a) shows the designed ontology of RDF-stored VG (RDF-VG)[2]. Fig. 1(b) demonstrates properties of the defined classes in RDF-VG. The properties (x, y, h, w) determine the visual block of region or object

[1] http://visualgenome.org/VGViz/explore demonstrates the dataset.
[2] Due to space limitations, the detail of data processing and ontology organization are attached to the code repository.

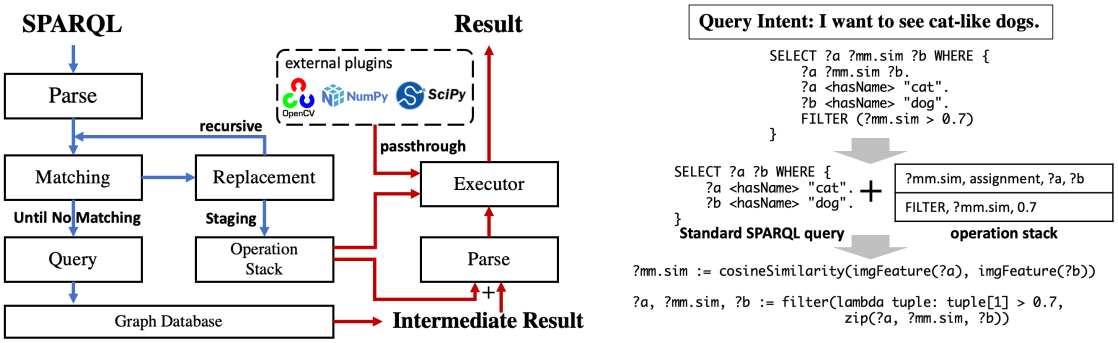

Figure 2: Left: The diagram of VGStore. Right: An example of VGStore handling the multimodal SPARQL query (Python-like pseudocode).

by tailoring image. We store RDF-VG in gStore [5], which is a graph-oriented RDF data management system supporting complex SPARQL queries on graph data.

3. Querying Multimodal Information via SPARQL

Traditional SPARQL engines (such as our backbone - gStore) cannot perform queries involving multimodality and thus cannot directly answer such questions (e.g. what color is this cat?). Therefore, we developed a VGStore extension based on standard SPARQL grammar and *py-parsing* [6] to parse custom predicates (as Fig. 1(c) shows) for arbitrary query patterns, enabling the ability of traditional graph databases to perform multimodal queries by passing through the extra computing requirements to third-party tools (e.g. OpenCV, Torch, etc.). The architecture of VGStore is shown in Fig. 2 (Left).

VGStore analyzes, matches, and replaces the clauses in the original SPARQL query that contain custom predicates. It recursively replaces all non-standard query patterns with the standard SPARQL syntax, and stores the replacement process in an operation stack temporarily. The standard SPARQL query can be handed over to the backbone graph database for execution.

Table 1
Part of the supported non-standard querying clauses. The ?a and ?b indicate the regular query variables.

Customized Triple Pattern	Description
?a **?mm.sim** ?b.	Represent the semantic similarity between ?a and ?b.
?a ⟨**mm.color**⟩ ?color.	Output the primary color of ?a to variable ?color.
?a **?mm.pos** ?b.	Represent the relative position relation between ?a and ?b.
?a **?mm.count** ?b.	Count the number of component ?b in image ?a.
FILTER(...)	Filter the results via customized variables, e.g., FILTER(?mm.sim > 0.5).
ORDER BY ...	Sort the results via customized variables, e.g., ORDER BY DESC(?mm.count).

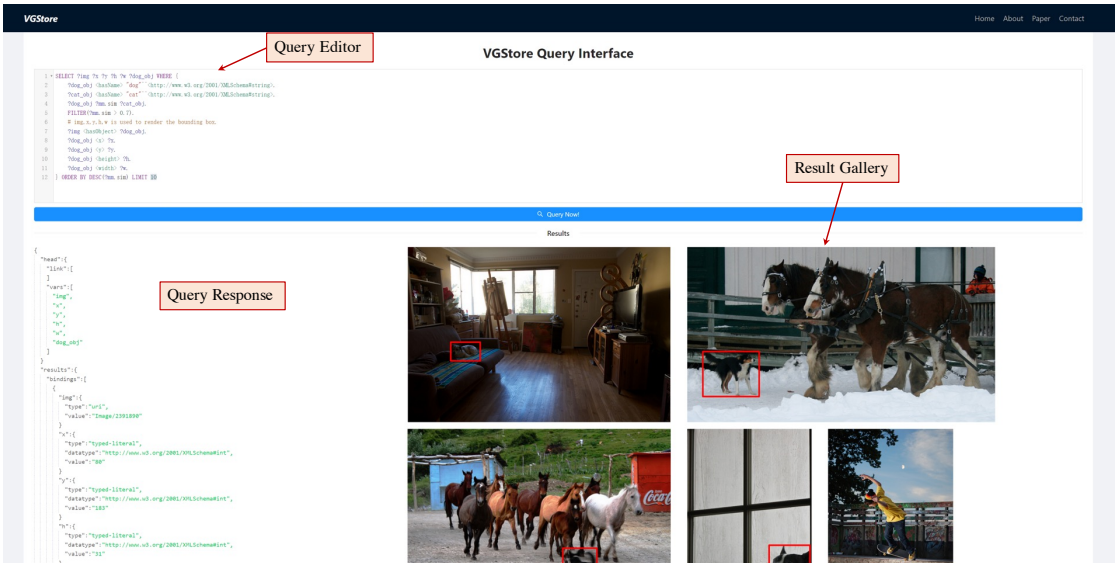

Figure 3: The interface in use for querying the RDF-stored visual genome scene graph with customized query pattern.

After getting the result, the inverse operation is successively performed according to the staged replacement operations in the stack. Finally, the intent of the original SPARQL query would be restored. Fig. 2 (Right) demonstrated how a multimodal query containing the non-standard custom predicate variable is to be executed. Table 2 illustrates part of the supported query patterns in `VGStore`, covering questions involving color, counting, and relative position in the VG question-answer dataset, which account for 15.0%, 11.4%, and 7.0% of all questions, respectively. Other simple questions (e.g., "What is this?") can be expressed and queried by native SPARQL directly, and the remaining non-factual questions (e.g., "What is this man's motivation?") or inference questions (e.g., "When was the picture taken?") are out of scope in this demonstration.

4. Discussion and Next Step

Although `VGStore` successfully supports multimodal queries by extending virtual predicates, it still has some limitations. `VGStore` is written in Python, which brings additional latency in runtime, and it is possible to reduce the performance loss by native support in the SPARQL engine. In addition, when the `VGStore` queries large data, and there are multiple extended query statements, it would cause severe performance problems. This drives us to schedule and parallelize the third-party tool calls.

`VGStore` currently only supports several basic query patterns (as listed in Table 1) specialized for the RDF-VG dataset, and does not adapt to other VQA datasets, nor does it support richer query patterns. Therefore, our next steps for improvements include supporting more query patterns and extending the applicability of `VGStore` to more graph databases with large-scale multimodal graphs.

5. Demonstration

This paper presented VGStore, an extension to SPARQL for querying multimodal information on RDF. The demo showcased the web user interface of VGStore for querying multimodal SPARQL on RDF-VG, as shown in Fig. 3. With this demo, we provided a potential solution to the main challenge of SP-based multimodal KBQA and laid the foundation of multimodal knowledge graph. Our code of RDF-VG builder, preliminary parser and frontend of demonstration are available at https://github.com/pkumod/VGStore.

Acknowledgments

This work was supported by NSFC under grant 61932001, U20A20174. The corresponding author of this paper is Lei Zou (zoulei@pku.edu.cn). We sincerely thank reviewers for their valuable comments and advises.

References

[1] M. Zhang, R. Zhang, Y. Li, L. Zou, Crake: Causal-enhanced table-filler for question answering over large scale knowledge base, in: Findings of the Association for Computational Linguistics: NAACL 2022, 2022, pp. 1787–1798.
[2] T. Kurz, S. Schaffert, K. Schlegel, F. Stegmaier, H. Kosch, Sparql-mm-extending sparql to media fragments, in: European Semantic Web Conference, Springer, 2014, pp. 236–240.
[3] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al., Visual genome: Connecting language and vision using crowdsourced dense image annotations, International journal of computer vision 123 (2017) 32–73.
[4] G. A. Miller, Wordnet: a lexical database for english, Communications of the ACM 38 (1995) 39–41.
[5] L. Zeng, L. Zou, Redesign of the gstore system, Frontiers of Computer science 12 (2018) 623–641.
[6] P. McGuire, Getting started with pyparsing, " O'Reilly Media, Inc.", 2007.