





SAP-KG: Synonym Predicate Analyzer Across Multiple Knowledge Graphs

Emetis Niazmand^{1,2}(✉)  and Maria-Esther Vidal^{1,2,3} 

¹ TIB Leibniz Information Centre for Science and Technology, Hannover, Germany
{Emetis.Niazmand, Maria.Vidal}@tib.eu

² Leibniz University of Hannover, Hannover, Germany

³ L3S Research Center, Hannover, Germany

Abstract. This demo paper presents SAP-KG, a knowledge graph agnostic tool, to illustrate the benefits of identifying the synonym predicates that provide complementary information; they are used for query rewriting to enhance query answer completeness. SAP-KG proposed a metric to compute the percentage of overlap between pairs of synonym predicate candidates and capture the most similar ones which can complement each other. We present a query processing technique that put in perspective the role of synonym predicates in query answer completeness. The demo code is available online in (<https://github.com/SDM-TIB/SAP-KG-ESWC2023Demo>) and can be run at (<https://mybinder.org/v2/gh/SDM-TIB/SAP-KG-ESWC2023Demo/main?labpath=SAP-KG.ipynb>).

Keywords: Knowledge Graphs · Synonym Predicates · Query Answer Completeness

1 Introduction

Community-maintained knowledge graphs, such as Wikidata [5] and DBpedia [3], have the potential to be incomplete due to the decentralized nature of their development and maintenance [1]. These community-maintained knowledge graphs are built collaboratively, enabling everyone to contribute and modify knowledge. Thus, predicates with different names that refer to the same thing can be added by different contributors. These predicates can be discovered as synonym based on different approaches; while they are precise in identifying synonym predicates, they cannot distinguish those with low overlap that represent complementary information. Moreover, the current approaches do not offer a query answering method to evaluate the completeness of answers after query rewriting utilizing identified synonym predicates. Acosta et al. [2] introduce a hybrid SPARQL engine that employs crowdsourcing to improve the completeness of query responses; while incorporating synonym predicates instead of crowd can reduce errors and uncertainty. We demonstrate SAP-KG, a knowledge graph-agnostic tool, to discover synonym predicates that provide complementary

information, and using them to reformulate the SPARQL queries to enhance query completeness. We illustrate the performance of SAP-KG, and show that rewriting queries based on synonym predicates can enhance answer completeness. Attendees will uncover the predicates with similar meanings but relating complementary entities in community-maintained KGs known as synonym predicates. They will also rewrite queries—which contain incomplete predicates—with their synonym ones to retrieve complete answers.

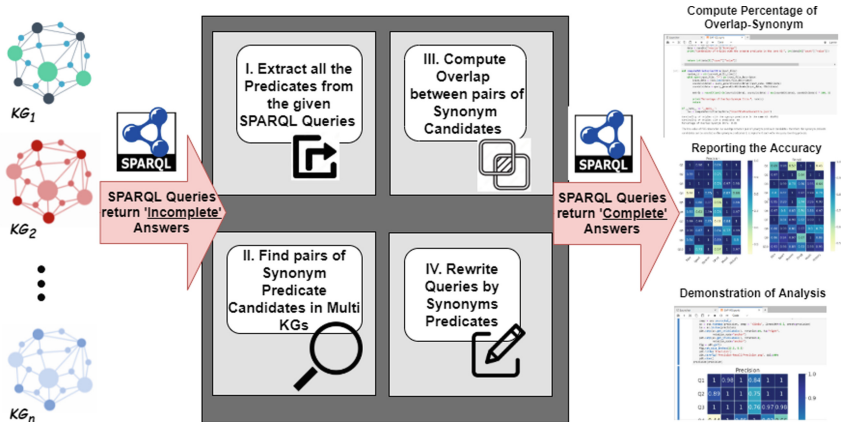


Fig. 1. The SAP-KG Architecture. An SPARQL query over multiple KGs is executed. After extracting predicates, synonym predicate candidates are discovered. The overlap between these candidates is computed. Synonym pairs with a low overlap are selected as synonym predicates to rewrite the query to retrieve the complete answers.

2 The SAP-KG Architecture

The input of SAP-KG is multiple knowledge graphs and SPARQL queries; these queries may retrieve incomplete results. SAP-KG outputs rewritten SPARQL queries based on synonym predicates to ones that retrieve complete answers. As seen in Fig. 1, SAP-KG comprises a step to extract predicates from an input query to find synonym candidates across multiple community-maintained KGs. We assume that the knowledge graphs contain all the links; otherwise, we can use tools such as Falcon 2.0 [4] as an extra step for joint entity and relation linking over the knowledge graphs. Among these candidates, the ones with lower overlap are selected to rewrite the query. For computing the overlap between pairs of synonym predicate candidates, the metric *Percentage of Overlap-Synonym (POS)* is defined. *POS* calculates the overlap between two predicates across multiple knowledge graphs. This metric indicates whether the equivalent predicate can provide additional information to complete the answers retrieved by a query.

Consider a knowledge graph G , a pair of predicates (P', P'') , and triple patterns $\mu(\cdot)$. The POS value is computed by the following formulation:

$$POS(P', P'', G) = \frac{\min(|\{\mu(P') | \mu(?s' P' ?o') \text{ in } G\}|, |\{\mu(P'') | \mu(?s'' P'' ?o'') \text{ in } G\}|)}{\max(|\{\mu(P') | \mu(?s' P' ?o') \text{ in } G\}|, |\{\mu(P'') | \mu(?s'' P'' ?o'') \text{ in } G\}|)} \times 100$$

The percentage of overlapping synonym predicates has a range between 0 and 100%, e.g., if the cardinality of triples with a specific predicate in KG_1 and the cardinality of triples with the synonym of that predicate in KG_2 are close to each other, then the POS value is close to 100%, otherwise the POS value is close to 0. Therefore, the POS value close to 100% describes that two predicates relate the same number of entities. On the other hand, the POS value close to 0 shows these predicates do not share the same entities, and can be considered as synonyms to complement each other. In the step of rewriting, the input query is transformed into an equivalent query that can produce more correct answers. The rewritten query that incorporates synonym predicates returns all the complete results. The aim is to rewrite the query with the minimum number of synonym predicates, while still returning the maximum number of correct answers. The naive tool rewrites queries with all possible synonym predicates, but POS metric help to select the synonym predicates that are most likely to return complete answers. Therefore, SAP-KG rewrites a query with a minimum number of synonym predicates that enhance answers completeness and return the maximum results.

3 Demonstration of Use Cases

Consider the following SPARQL queries in Fig. 2. The original query over Wikidata on the left side presents: *Retrieve name of children (wdt:P40), cause of death (wdt:P509), place of birth (wdt:P19), and parent (wdt:P8810) of Marella Agnelli (wd:Q3290404)? - (Retrieval date, Feb 2023)*, and on the right side the rewritten SPARQL query over Wikidata and DBpedia is shown.



Original SPARQL Query	Rewritten SPARQL Query
<pre> PREFIX wdt: <http://www.wikidata.org/prop/direct/> PREFIX wd: <http://www.wikidata.org/entity/> PREFIX dbr: <http://dbpedia.org/resource/> PREFIX dbo: <http://dbpedia.org/ontology/> PREFIX dbp: <http://dbpedia.org/property/> SELECT DISTINCT ?o ?o1 ?o2 ?o3 WHERE { wd:Q3290404 wdt:P40 ?o. wd:Q3290404 wdt:P509 ?o1. wd:Q3290404 wdt:P19 ?o2. wd:Q3290404 wdt:P8810 ?o3.} </pre> <div style="text-align: center;">  #Answer: 0 </div>	<pre> SELECT DISTINCT ?o ?o1 ?o2 ?o3 ?o4 ?o5 WHERE { SERVICE <https://query.wikidata.org/sparql> {wd:Q3290404 wdt:P40 ?o. wd:Q3290404 wdt:P509 ?o1. wd:Q3290404 wdt:P19 ?o2. wd:Q3290404 wdt:P25 ?o3.} SERVICE <https://dbpedia.org/sparql> {dbr:Marella_Agnelli dbo:child ?o4. dbr:Marella_Agnelli dbp:birthPlace ?o5.} </pre> <div style="text-align: center;">  #Answer: 8 </div>

Fig. 2. An original SPARQL query comprising four triple patterns executed over Wikidata does not retrieve any answers. The rewritten SPARQL query by detected synonym predicates from Wikidata and DBpedia retrieves eight answers.

By the time this demo was prepared, this query returns no answer. There are four predicates that cause the query to retrieve no result. Simply rewriting query to another query by considering all the synonym predicate candidates retrieve many results which may be incorrect or cannot complete the answers. Therefore, it needs a technique to select the synonym predicates among all discovered synonym predicate candidates with lower overlap for query rewriting. The rewritten SPARQL query returns eight answers.

Effects of *POS* Metric to Provide Complementary Synonym Predicates. A naive tool considers all possible synonym predicate candidates in rewriting queries; while the use of the *POS* metric enables the selection of synonym predicates that are most probable to provide comprehensive answers. As an example, the predicate *manner of death (wdt:P1196)* is a synonym candidate for the predicate *cause of death (wdt:P509)*. The *POS* metric is computed for the predicate candidate pairs. Since, the *POS* value of these synonym candidates is high (= 87.09%), they cannot be considered as synonym predicates for query rewriting. The high overlap shows these synonym candidates can not complement each other, and they do not lead to retrieve complete results. Thus, predicate *manner of death (wdt:P1196)* is not considered in query reformulation process.

Answer Completeness by Rewriting Queries with Synonym Predicates. For example, by having the predicate *place of birth (wdt:P19)* in the query, only *Florence* is returned as the answer. For predicate *place of birth (wdt:P19)* in Wikidata, there is at least one synonym predicate candidate in DBpedia as *dbp:birthPlace*; where the *POS* value for the above pair is equal to 1.23%. The low overlap value indicates that these synonym predicate candidates are complementary. By rewriting the query with the synonym predicate *dbp:birthPlace* in DBpedia, apart from *Florence*, also *Kingdom of Italy* is returned as the answer. The performance of our tool by running sixty queries over six domains *Person*, *Music*, *History*, *Film*, *Sport*, and *Drug* is shown in Fig. 3. The high value of precision in most of the queries indicates the completeness of answers of rewritten queries compared to the original ones.

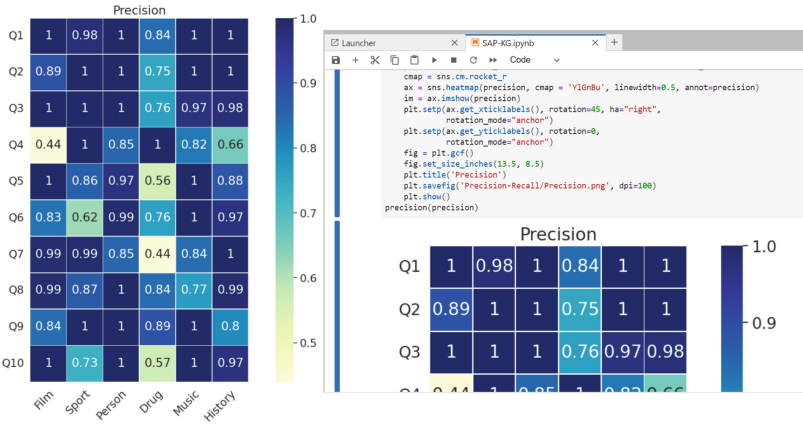


Fig. 3. The performance of SAP-KG where queries rewritten by synonym predicates.

4 Conclusions

We demonstrate SAP-KG and illustrate results that suggest that our proposed technique by considering synonym predicates may return complete answers when faced with queries containing incomplete predicates. Current approaches are not able to differentiate between synonym predicates that provide complementary information. Also, depending on distribution of synonym predicates in multiple community-maintained knowledge graphs, there will be different execution plans to rewrite the queries based on detected synonym predicates. The attendees will evaluate various queries and observe the crucial role of synonym predicates in the completeness of queries executed against Wikidata and DBpedia.

Acknowledgement. This work has been supported by the EU H2020 RIA project CLARIFY (GA No. 875160) and the project TrustKG-Transforming Data in Trustable Insights with grant P99/2020.

References

- Abiteboul, S.: Querying semi-structured data. In: Afrati, F.N., Kolaitis, P.G. (eds.) Database Theory - ICDT 1997, 6th International Conference, Delphi, Greece, 8–10 January 1997, Proceedings. LNCS, vol. 1186, pp. 1–18. Springer, Cham (1997). https://doi.org/10.1007/3-540-62222-5_33
- Acosta, M., Simperl, E., Flöck, F., Vidal, M.: Enhancing answer completeness of SPARQL queries via crowdsourcing. *J. Web Semant.* **45**, 41–62 (2017). <https://doi.org/10.1016/j.websem.2017.07.001>

3. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al., (eds.) *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007*, Busan, Korea, 11–15 November 2007. LNCS, vol. 4825, pp. 722–735. Springer, Cham (2007). https://doi.org/10.1007/978-3-540-76298-0_52
4. Sakor, A., Singh, K., Patel, A., Vidal, M.: FALCON 2.0: an entity and relation linking tool over WikiData. In: d’Aquin, M., Dietze, S., Hauff, C., Curry, E., Cudré-Mauroux, P. (eds.) *CIKM 2020: The 29th ACM International Conference on Information and Knowledge Management*, Virtual Event, Ireland, 19–23 October 2020, pp. 3141–3148. ACM (2020). <https://doi.org/10.1145/3340531.3412777>
5. Vrandečić, D., Krötzsch, M.: WikiData: a free collaborative knowledgebase. *Commun. ACM* **57**(10), 78–85 (2014). <https://doi.org/10.1145/2629489>