SAP-KG: Analysis of Synonym Predicates using Wikidata

Emetis Niazmand a,⁎ and Maria-Esther Vidal a,b

a TIB Leibniz Information Centre for Science and Technology, Leibniz University of Hannover, Germany
b L3S Research Center, Leibniz University of Hannover, Germany
E-mails: Emetis.Niazmand@tib.eu, Maria.Vidal@tib.eu

Abstract. Wikidata, as a community-maintained knowledge graph (KG), contains millions of facts; it may integrate different entities and relations with the same meaning. Contributors of community-maintained knowledge graphs can use new predicates which are similar in meaning to other predicates in the KG (a.k.a. synonym predicates). Detecting these synonym predicates plays a crucial role in interoperability and query answer completeness against community-maintained knowledge graphs. We tackle the problem of uncovering synonym predicates, and propose SAP-KG, a knowledge graph-agnostic approach, to uncover the predicates with similar meanings but relating complementary entities. SAP-KG comprises a set of metrics to describe and analyze synonym predicates; it resorts to Class-based Synonym Descriptions (CSDs) to capture the most important characteristics of the predicates of a knowledge graph. As a proof of concept, we evaluate SAP-KG over Wikidata and show the benefits of exploiting statements annotated with qualifiers, references, and ranks. Additionally, we present a query processing technique that put in perspective the role of synonym predicates in query answer completeness. We have empirically studied the distribution and percentage of overlapping synonym predicates in six domains in Wikidata. The highest percentage of synonyms has been detected in the Person domain at 86.66%, while Drug has the lowest percentage, i.e., 42.39%. These results provide evidence that community-maintained knowledge graphs enclose predicates that define the same real-world relationships.

Keywords: Knowledge Graphs, Wikidata, Synonym Predicates, Query Processing

1. Introduction

The Linked Open Data (LOD) cloud is a knowledge graph of publicly available RDF knowledge graphs that follow the Linked Data Principles [1]. LOD contains 1,512 RDF knowledge graphs with around 413,734,019,304 RDF triples by August 2021. The number of linked datasets has rapidly increased and share knowledge on various domains such as Wikidata [2] and DBpedia [3]. The growth of entities and properties in encyclopedic knowledge graphs impacts query management tasks like query processing. The community-maintained knowledge graphs, such as Wikidata, are flexible and interoperable. Since Wikidata’s web interface allows any user to edit it, multiple users can add properties with different names that refer to the same thing. Therefore, knowledge graphs may consist of many synonym properties.

Community-maintained knowledge graphs, such as Wikidata, have the potential to be incomplete due to the decentralized nature of their development and maintenance [4]. Since contributors may have varying levels of expertise and interest in different topics, some areas of the knowledge graph may not be as well-developed as others. Also, based on the Open World Assumption (OWA), if a relation is not part of some knowledge yet, then it cannot be

⁎Corresponding author. E-mail: Emetis.Niazmand@tib.eu.

1https://lod-cloud.net/lod-data.json
sure that this relation does not exist. Therefore, it is important to be aware of the potential incompleteness of such resources when using them for research or other purposes. This incompleteness can be enhanced by detecting synonym properties in KGs; they can be found at two levels, inside a knowledge graph or in other knowledge graphs. These predicates can be discovered as synonym based on different Natural Language Processing (NLP) methods such as Information Retrieval (IR) [5], Embeddings (Contextual [6], Word2Vec [7], RDF2Vec [8]), and Association Rule Mining [9, 10]. Albeit accurate in synonym predicate detection, existing approaches cannot discriminate from the synonym predicates that have low overlap and represent complementary information.

Our work improves upon previous approaches that utilize embeddings and rule mining to capture synonym predicates by taking into account overlap between predicates within a knowledge graph as well as across multiple knowledge graphs. There are few studies that examine the use of knowledge graph embedding models or association rule mining to address the problem of detecting synonym predicates in knowledge graphs, e.g., in [11], authors used a property of knowledge graph embeddings for detecting synonym predicates, such as TransH [12] and TransD [13]. Abedjan et al. [9] propose a method for discovering synonym predicate pairs in data that can be used to expand queries. Their method uses frequent item set mining-based techniques to identify synonymously used relationships. However, the experimental study reported by them is limited by a small dataset and may not be substitutes for all the properties in community-maintained knowledge graphs. Overall, current approaches are not able to differentiate between synonym predicates that provide complementary information. Additionally, they do not provide a query answering method to evaluate the completeness of query answers after query expansion with discovered synonym predicates. Acosta et al. [14] propose a hybrid SPARQL engine that leverages crowdsourcing to enhance query answer completeness. While innovative, incorporating knowledge of synonym predicates could further guide the crowd and reduce uncertainty [15]. Our approach relies on knowledge graph embeddings (e.g., Word2Vec and RDF2Vec) and determines relatedness between predicates based on similarity measures to discover synonym predicates that are complementary. Although our approach is knowledge graph-agnostic, we aim to detect the synonym predicates in Wikidata and make the community aware of the benefits of exploiting them in query processing.

Problem Statement and Proposed Solution. We tackle the problem of discovering synonym predicates in community-maintained knowledge graphs, e.g., Wikidata, and using them to reformulate the SPARQL queries to enhance query completeness. We describe Synonym Predicates in terms of Class-based Synonym Description (CSD), which captures synonym predicates at the level of classes. A CSD is presented to show how two predicates in knowledge graphs, at the level of classes, can be synonyms. In a CSD, diverse metrics are used to discover synonym predicates; they are calculated over all synonym predicates of a class in a knowledge graph and quantify the degree of similarity. SAP-KG is knowledge graph-agnostic; however, we show how it is able to capture and describe existing synonym predicates in Wikidata. We analyzed the distribution and percentage of overlapping synonym predicates across six domains in Wikidata. Person is the domain with the highest percentage, while Drug has the lowest scores. Additionally, we evaluate the performance of our proposed approach in discovering synonym predicates for query reformulation, resulting in answers that are more similar to those retrieved from a gold standard; this gold standard provides the complete answer to each query of the benchmark. Our findings demonstrate the benefits of SAP-KG, as synonym predicates required for enhancing answer completeness can be detected.

Contributions: In a nutshell, this paper makes the following contributions: a) Class-based Synonym Description (CSD), a KG description model to represent synonym predicates in knowledge graphs. b) SAP-KG, an analysis approach, that utilizes the CSD concept to select the synonym predicates that complete the answers in query processing. c) An empirical evaluation was conducted on six domains in Wikidata, and the results show that in the Person domain, the distribution of synonym predicates is 86.66%, while in the Drug domain, it is 42.39%. Furthermore, using the discovered synonym predicates to rewrite queries can result in a precision of 1.0 for all queries in SAP-KG. The recall of our approach is higher than 0.82% for 80% of the rewritten queries, which shows the completeness of the queries reformulated using synonym predicates with respect to the knowledge graph.

This paper is organized into five additional sections. Section 2 presents the preliminaries and motivates our work with an example. The related works are briefly described in section 3. Section 4 defines our approach, and section 6 reports and discusses the results of our empirical study. Finally, section 7 concludes and outlines our future work.
2. Preliminaries and Motivation

2.1. Preliminaries

RDF Knowledge Graphs. An RDF (Resource Description Framework) graph is a directed edge-labeled graph, \( G = (V, E, L) \), where \( V \) is a set of vertices represented as classes and entities, and \( E \) is a set of edges annotated with predicates in \( L \). Each edge in \( E \), represents an RDF triple, \( t = \langle s, p, o \rangle \), where \( s \), \( p \), and \( o \) correspond to a subject (the node at the start of the edge), a predicate (the edge that connects the subject to the object), and an object (the node at the end of the edge), respectively. Subjects can be URI and blank nodes, while predicates should be only in the form of URIs. Lastly, objects can be URIs, blank nodes, or literals.

Wikidata. In October 2012, Wikidata [2] was launched as a project of the Wikimedia Foundation with the purpose of providing a community-maintained encyclopedic knowledge graph. Wikidata is built collaboratively, enabling everyone to contribute and share knowledge. By the time this paper was prepared, Wikidata consists of almost 10,000 predicates with their descriptions. There are different types of predicates in Wikidata: a) functional predicates or predicates where each entity of the domain has exactly one object, e.g., mother(wdt:P25) or father(wdt:P22); and b) multivalued predicates where entities can have more objects, e.g., relative(wdt:P1038). However, in the most cases, an entity can have zero or several objects, e.g., child(wdt:P40) or spouse(wdt:P26). The main advantage of Wikidata compared with other community-maintained knowledge graphs is having statements annotated with values and reference information of the qualifier for predicates. These statements and qualifiers in Wikidata provide a more precise representation of data; thus, they can be used to support more accurate querying and reasoning over the data.

Knowledge Graph Embeddings. Given a directed edge-labeled graph \( G = (V, E, L) \); \( V \) is a set of nodes, \( L \) is a set of edge labels, and \( \top \) set of vectors. A knowledge graph embedding of \( G \) is a pair of mappings \( (\epsilon, \rho) \) such that [1]:

\[
\epsilon: V \rightarrow \top, \text{ i.e., } \epsilon(v) \text{ maps a node } v \text{ to a vector in } \top; \\
\rho: L \rightarrow \top, \text{ i.e., } \rho(l) \text{ maps a node } l \text{ to a vector in } \top.
\]

Knowledge graph embedding models transform nodes and edges in \( G \) into a low-dimensional continuous vector space that preserves the structure of \( G \). This symbolic depiction of a predicate can be encoded in a subsymbolic representation using embeddings computed by graph embedding methods [16], e.g., RDF2Vec [8] and Word2Vec [7].

Rule Mining Models. A mined rule \( r \) is a Horn clause of the form \( r: \text{Body} \Rightarrow \text{Head} \), where \( \text{Body} \) is a conjunction of predicate facts and \( \text{Head} \) is a predicate fact. All the variables in the \( \text{Head} \) are in terms of at least one predicate fact in the \( \text{Body} \); and every two predicate facts in the \( \text{Body} \) share at least one variable. In the context of knowledge graphs, there is a set of positive and negative predicates, where positive predicates are entailed by a knowledge graph; while negative predicates are defined according to a given assumption of completeness. An entailed fact of \( r \) is a positive entailed fact if \( \text{predicate}(\text{entity1}, \text{entity2}) \in E. \) Support: The support of the rule \( r \) corresponds to the number of positive entailed facts by the rule. Confidence: The confidence of the rule \( r \) corresponds to the ratio of the positive predicate facts of \( \text{Head} \) which are positive entailed facts based on the rule [17]. Partial Completeness in RDF Graph: Given a directed labeled-edge graph \( G = (V, E, L) \). Let \( E^+(r) \) be the positive predicate facts of \( r \). The Partial Completeness Assumption (PCA) [18] assumes that heuristic-based negative edges \( hE^-(r) \) are incomplete edges in \( G \), i.e., every edge \( \langle s, p, o' \rangle \) not in \( E^+(r) \), but the \( \langle s, p, o \rangle \) belongs to \( E^+(r) \). The PCA assumes that if there is an object for a subject with predicate in the knowledge graph, then that knowledge graph contains all the objects for that given subject and predicate. Mined rules are used to identify these heuristic-based negative edges. The Partial Completeness Assumption (PCA) confidence score of the rule \( r \) is the ratio of \( support(r) \) to the cardinality of the union of \( E^+(r) \) and \( hE^-(r) \).

---

2https://www.wikidata.org/wiki/Wikidata:Database_reports/List_of_properties/all - (Retrieval date, Jan 2023)  
3https://en.wikibooks.org/wiki/SPARQL/WIKIDATA_Qualifiers,_References_and_Ranks
The value of PCA confidence is between 0.0 to 1.0. A value of 0.0 shows that none of the entailed instantiations of predicate fact for r belongs to the KG. A value of 1.0 indicates that all the entailed instantiations of predicate fact for r belong to the KG. The low value of PCA confidence score leads to retrieve the incomplete answers. Example. The two depicted mined Horn rules have values of PCA confidence scores equal to 0.5 and 0.25, respectively. These rules suggest that if a person has a parent in Wikidata, they typically also have a father and/or mother. However, the low PCA confidence values, especially for the rule involving mother, indicate that many people in Wikidata have a parent but lack a father and/or mother. The SPARQL queries to compute the PCA confidence score are provided in the Appendix A.

2.2. Motivating Example

We motivate our work by providing an example to illustrate the presence of synonym predicates in Wikidata. The web interface of Wikidata allows contributors to modify this knowledge graph. Thus, the same properties with different names can be added by any users. This interoperability issue in Wikidata is shown in Figure 1, where there is a relation as parent (wdt:P8810) between Davi Lucca and Neymar, but there is no relation between them as father (wdt:P22). Since, predicates father (wdt:P22) and mother (wdt:P25) are subproperty of the predicate parent (wdt:P8810), they can be considered as synonym predicates. Moreover, there is an incorrect relation as mother (wdt:P25) and child (wdt:P40) between Davi Lucca and Rafaella Santos. In this example, by considering the predicates father and parent as synonyms, we can determine the father of Davi Lucca. Therefore, discovering the synonym predicates in Wikidata can help us to overcome the incompleteness in the knowledge graph [19] and retrieve the complete results.

3. Related Work

In recent years, knowledge graphs have become an increasingly popular way to represent large amounts of structured data. However, despite their usefulness, knowledge graphs are often incomplete. One promising approach involves discovering synonym predicates, which can help to uncover new relationships and connections within the data. In this section, we will review some of the existing literature on incompleteness in knowledge graphs and the use of synonym predicates for improving their completeness.

3.1. Approaches for Assessing and Addressing Incompleteness of KGs

Knowledge graphs based on Open-World Assumption (OWA) are quite incomplete. Incompleteness is a well-known problem in community-maintained knowledge graphs. Incompleteness occurs when a knowledge graph does not contain all the information about a particular entity or relationship that exists in the real-world, leading to retrieve incomplete results in query processing [4]. The study in [19] reports that in 69%-99% of entities in community-maintained knowledge graphs, at least one predicate that other entities in the same class have is missed. Incompleteness can have negative impacts on various applications that rely on knowledge graphs, such as question answering systems, recommendation engines, and intelligent agents. To address this problem, researchers have proposed sev-
Fig. 1. Motivating Example. This figure illustrates a small portion of Wikidata. The resource Davi Lucca is associated with the resource Neymar through predicate parent (wdt:P8810). However, there is no relation as father (wdt:P22) between them. Since father is subproperty of parent and male parent is equal to father, the resource Davi Lucca has Neymar as his father. Therefore, the predicates father and parent are synonym and can complement each other. Additionally, there is an incorrect relation as mother (wdt:P25) and child (wdt:P40) between Davi Lucca and Rafaella Santos. The pictures are taken from Wikipedia and Twitter. (Retrieval date, Jan 2023)

eral techniques [20], including data augmentation, entity resolution, knowledge graph enrichment, and crowdsourcing [14]. A notable study by Galárraga et al. [21] proposes an extension of the SPARQL query language to support completeness. While estimating incompleteness is a challenging task, several approaches employ different models to estimate it. For instance, Acosta et al. [14] leverages crowdsourcing to identify the parts of a query that retrieve missing answers, yielding incomplete results. Addressing the problem of incompleteness in knowledge graphs can lead to improvements in their overall quality. Developers and researchers can work to enhance the accuracy and consistency of the knowledge graph.

3.2. Association Rule Mining

Association Rule Mining [22] is a data mining technique used to identify and extract relationships or patterns between items in large datasets. The technique involves finding relationships between items in a dataset using two measures, support and confidence. Support measures the number of positive entailed facts, while confidence measures the number of positive entailed facts to the total number of positive predicate facts. For some applications, the standard measurements for support and confidence do not produce good results. PCA confidence is another measure that identifies much more productive Horn rules than the other measures by AMIE [18]. The mined Horn rules are of the form ?a (parent_P8810) ?b ⇒ ?a (father_P22) ?b, meaning that if a person has a father in Wikidata usually also has parent. The PCA confidence score for this mined Horn rule is 0.5. It suggests that while many people in Wikidata have a father (wdt:P22), a significant number of them lack a parent. Therefore, it can be inferred that the predicates father (wdt:P22) and parent (wdt:P8810) are incomplete with respect to the KG.

3.3. Synonym Predicates Discovery

To address the challenge of detecting synonymous predicates, several methods have been utilized, including lexical and semantic methods. These methods involve analyzing the text of predicates and related entities [23], utilizing embedding techniques to identify similarities among predicates [11], and employing frequent item set mining through aggregating positive and negative association rules at the statement level [9]. Moreover, there is a technique for de-
Fig. 2. Synonym Predicate Candidates. An example demonstrating the presence of synonym predicates in both Wikidata and DBpedia knowledge graphs. The resource Marella Agnelli cannot be associated to any objects through the predicate manner of death (wdt:P1196); however, there is a synonym for that predicate as cause of death (wdt:P509) which can be a complement for the predicate manner of death (wdt:P1196). (Retrieval date, Nov 2022)

Detecting synonym predicates in large knowledge graphs using association rule mining [10]. Additionally, knowledge-based methods, which use external knowledge sources, e.g., ontologies, can also be used to identify synonym predicates [24, 25]. Discovering synonym predicates enables to capture the inherent structure and relationships between entities in knowledge graphs, and can help to improve overall accuracy and completeness of the knowledge graph.

4. The SAP-KG Approach

This section presents the problem tackled in our proposed solution. First, we introduce concepts of synonym predicate candidates and synonym predicates. The synonym predicates in the same knowledge graph correspond to Intra-synonym, while Inter-synonym denote synonym predicates across various knowledge graphs. We introduce Class-based Synonym Description (CSD), as abstract descriptions of the synonym predicates in a knowledge graph.

**Synonym Predicate Candidates.** Given a knowledge graph $KG=(E,V,L)$, and predicates $p_i$ and $p_j$ in $E$, defined for the same class $C$. Predicates $p_i$ and $p_j$ are considered as synonym candidates, if they have the same subjects and the same objects. E.g., consider two triples $t_1 = \langle \text{wd:Q3290404}, \text{wdt:P509}, \text{wd:Q11085} \rangle$, $t_2 = \langle \text{wd:Q3290404}, \text{wdt:P1050}, \text{wd:Q11085} \rangle$. Since both triples have the same subject and object, they are considered equivalent. Their predicates, cause of death (wdt:P509) and medical condition (wdt:P1050), are considered as synonym predicate candidates but not synonym predicates because they cannot complement each other any further. Figure 2 shows an example of synonym predicate candidates. They can belong to the same knowledge graph (a.k.a. (Intra-synonym)), or be part of more than one KGs (a.k.a. (Inter-synonym)). Synonym predicate candidates are defined as:

$$s_i = s_j \land o_i = o_j \Rightarrow p_i \sim \text{candidate } p_j$$

Note that simply having the same subject and the same object is not enough to conclude that two triples include synonym predicates. Consider the other two triples with the same subject and object, $t_3 = \langle \text{wd:Q23882021}, \text{wdt:P19}, \text{wd:Q84} \rangle$ and $t_4 = \langle \text{wd:Q23882021}, \text{wdt:P20}, \text{wd:Q84} \rangle$. Since, the statement $st$, e.g., title for the predicates place of birth (wdt:P19) and place of death (wdt:P20) are not the same, then they are not con-
Fig. 3. **Class-based Synonym Description.** For each knowledge graph, the synonyms of predicates for each required classes are detected by existing techniques. Each Class-based Synonym Description (CSD) defines synonyms for predicates in the same data source or in other data sources. As an example, considering \( \text{dbo:child} \) in DBpedia as a synonym for \( \text{child(wdt:P40)} \) in Wikidata with POS value 3.1% returns all related entities to the resource Marella Agnelli that completes the answers during query processing. (Retrieval date, Nov 2022)

**Synonym Predicates.** Given synonym predicate candidates, \( p_i \) and \( p_j \) have the same subject and the same object. They have the same meaning but are defined differently in the knowledge graph. However, according to the definition of synonym predicate candidates, they are not actually synonym predicates because they cannot complement each other to provide additional information. However, consider two triples \( t_5 = \langle \text{wd:Q45864}, \text{wdt:P1196}, \text{wd:Q10737} \rangle \), \( t_6 = \langle \text{wd:Q45864}, \text{wdt:P509}, \text{wd:Q175111} \rangle \). Since both triples have the same subject and different object, the triples are not equivalent and synonym predicates manner of death(wdt:P1196) and cause of death(wdt:P509) can complement each other. So, they are synonym predicates. Selecting synonym predicates in KGs helps to complete the entities related to the specific resource. Therefore, we can conclude that one of the definitions of the synonym predicate is as follows:

\[
s_i = s_j \land o_i = o_j \land s_l = st_j \Rightarrow p_i \sim \text{candidate } p_j
\]

Moreover, we define a set of metrics in subsection 5.1 to describe synonym predicates. The idea is to rewrite queries using a minimum number of synonym predicates that maximize the completeness of the answer. Thus, we need to select from the list of synonym predicate candidates, the ones that correspond to synonym predicates.

**Class-based Synonym Description (CSD).** The concept of Class-based Synonym Description (CSD) is presented in this work. The definition of CSDs as a source description, equivalent to RDF-MTs [26], focusing on showing the synonym predicates of each classes in KGs. Each CSD represents the RDF classes, predicates, and synonym of predicates. This concept helps to define synonym predicates in different KGs. A Class-based Synonym Description (CSD) is a 3-tuple \( \langle KG, C, S DP \rangle \), where:

\( KG=\langle V, E, L \rangle \) is a knowledge graph;
Fig. 4. **Ideal Knowledge Graph.** a) A small portion of Wikidata and DBpedia as community-maintained knowledge graphs with a set of resources, entities, and predicates is presented. b) A part of Ideal knowledge graph, where all Intra- and Inter-synonym predicates by CSDs are described with `owl:sameAs` and `subproperty of (wdt:P1647)` between `relative (wdt:P1038)` and `child (wdt:P40)`.

C – is an RDF class such that the triple pattern `(?s rdf:type C)` is true in KG;

SDP – is a set of tuples (p, SD) such that p is a predicate with domain C, and SD is a set of tuples (p’, MSD) such that p’ is a synonym of p and MSD is a set of class-level metrics

A real example to describe the synonym predicates in KGs as CSDs can be seen in Figure 3. In this example, Wikidata and DBpedia are represented. The predicates which relate the resource Marella Agnelli to her children are detected as synonyms, `child (wdt:P40)` and `dbo: child`, respectively. The class-level metric scores, MSD, are calculated over all synonym predicates of a class in all knowledge graphs to describe the synonym predicates; Class-based Synonym Descriptions (CSDs) provides the required information to retrieve all related entities.

**Ideal Knowledge Graph.** Consider a knowledge graph $KG=(V, E, L)$ and an Ideal knowledge graph of $KG$, $KG_{Ideal} = (V', E', L')$ defined as two directed edge-labelled graphs with the same entities in $V$, where:

- $E'$ is a combining of $E^+$ and $E^-$;
- $E^+$ is a set of positive edges which correspond to the edges in $E$;
- $E^-$ is a set of negative edges corresponding to the triples $(s', p', o') \in (VxLxV) - E$

These negative edges $E^-$ are true in the Ideal knowledge graph ($KG_{Ideal}$) but because of the Open World Assumption (OWA), they are not included in $KG$. In the part of the Ideal knowledge graph depicted in Figure 4, all predicates and their synonyms can be found. The information of the `owl:sameAs`, `owl:equivalentProperty`, and `subproperty of (wdt:P1647)` is part of the Ideal $KG$. For instance, the predicate `child (wdt:P40)` is a subproperty of the predicate `relative (wdt:P1038)`. In the Ideal $KG$, each predicate is accompanied by all other predicates that share the same subjects and the same objects.

### 4.1. Problem Statement and Proposed Solution

**Problem Statement** Given a knowledge graph as a directed edge-labeled graph $KG=(V, E, L)$, where $V$ is a set of nodes and $E$ is a set of edges annotated with predicates in $L$. Each edge in $E$, represents an RDF triple structure of subject $s$, predicate $p$, and object $o$ such as $E=\{(s_1, p_1, o_1), (s_1, p_2, o_1), (s_1, p_3, o_2), \ldots\}$. There might be predicates (e.g., $p_1$, $p_2$, and $p_3$) that correspond to the same predicate in the Ideal knowledge graph. The problem is to identify Intra- and Inter- synonym predicates which are able to complement each other. **Proposed Solution** Our proposed approach resorts to Class-based Synonym Descriptions (CSDs) to capture the most important characteristics of the predicates. Our approach shows how CSDs enable to describe synonym predicates in Wikidata.
5. The SAP-KG Architecture

SAP-KG comprises three components: a) Incompleteness Detection; b) Synonym Predicates; and c) Query Rewriting. SAP-KG receives a SPARQL query, a knowledge graph, and an ontology as input. The output of SAP-KG is the enhanced query rewritten by synonym predicates with the complete results.

5.1. Incompleteness Detection

The input of this component is an SPARQL query that retrieves incomplete results, and the output is the number of predicates in triple patterns that cause the query to retrieve incomplete answers. Consider the original SPARQL query in Figure 6, that consists of four triple patterns. By calculating the value of PCA and the percentage of overlap between the predicates and their equivalents in the same or other knowledge graphs, we can determine whether these predicates may cause the query to return incomplete answers. In the following, a set of metrics described to capture the number of triples of each resource of class C for a specific predicate P and describe the synonym predicates based on classes. Percentage of Overlap-Synonym (POS) metric calculates the overlap between two predicates within a single knowledge graph or across multiple knowledge graphs. This metric indicates whether the equivalent predicate can provide additional information to complete the answers retrieved by a query. This overlap can be over domain POS-D or range POS-R. POS values range is between 0 and 100%, e.g., if the cardinality of triples with a specific predicate in Wikidata and the cardinality of triples with the synonym of that predicate in Wikidata or other knowledge graphs such as DBpedia 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. Consider two synonym predicate candidates child(wdt:P40) in Wikidata, and dbo:child in DBpedia, according to the example in Figure 5. Based on the following formula, the POS value is equal to 3.1% which means dbo:child is a synonym predicate for child(wdt:P40) and they can complete each other. The same for predicates child(wdt:P40) and relative(wdt:P1038) in Wikidata, with the POS value equal to 3.6%. The low value of POS indicates that these synonym predicate candidates are complementary as well.

**Definition 5.1 (Percentage of Overlap-Synonym (POS)).** Given a knowledge graph G, a pair of predicates (Pred₁, Pred₂), and the percentage of overlapping synonym predicates inside a knowledge graph or in a federation of knowledge graphs is as follows. μ(·) is also defined for triple patterns:

\[
\text{POS(Pred₁, Pred₂, G)} = \frac{\min\{|\mu(Pred₁)| \mu(?s₁, Pred₁ ?o₁) \text{ in } G\}, \{|\mu(Pred₂)| \mu(?s₂, Pred₂ ?o₂) \text{ in } G\}|}{\max\{|\mu(Pred₁)| \mu(?s₁, Pred₁ ?o₁) \text{ in } G\}, \{|\mu(Pred₂)| \mu(?s₂, Pred₂ ?o₂) \text{ in } G\}|} \times 100
\]

We define the same metrics with respect to the domain and range of each predicate as **Percentage of Overlap-Synonym in Domain (POS-D)** and **Percentage of Overlap-Synonym in Range (POS-R)**, respectively.

\[
\text{POS-D(Pred₁, Pred₂, G)} = \frac{\min\{|\mu(\mu(?s₁))| \mu(?s₁, Pred₁ ?o₁) \text{ in } G\}, \{|\mu(\mu(?s₂))| \mu(?s₂, Pred₂ ?o₂) \text{ in } G\}|}{\max\{|\mu(\mu(?s₁))| \mu(?s₁, Pred₁ ?o₁) \text{ in } G\}, \{|\mu(\mu(?s₂))| \mu(?s₂, Pred₂ ?o₂) \text{ in } G\}|} \times 100
\]

\[
\text{POS-R(Pred₁, Pred₂, G)} = \frac{\min\{|\mu(?o₁)| \mu(\mu(?o₁)) \mu(?s₁, Pred₁ ?o₁) \text{ in } G\}, \{|\mu(?o₂)| \mu(\mu(?o₂)) \mu(?s₂, Pred₂ ?o₂) \text{ in } G\}|}{\max\{|\mu(?o₁)| \mu(\mu(?o₁)) \mu(?s₁, Pred₁ ?o₁) \text{ in } G\}, \{|\mu(?o₂)| \mu(\mu(?o₂)) \mu(?s₂, Pred₂ ?o₂) \text{ in } G\}|} \times 100
\]
Fig. 5. **Percentage of Overlap-Synonym**: An example to show how the Percentage of Overlap-Synonym (POS) metric is computed over pairs of synonym predicate candidates inside a knowledge graph (Intra-synonym) or in other community-maintained knowledge graphs (Inter-synonym).

The high value of POS shows that the pairs of synonym predicate candidates are complete with respect to the knowledge graph; while the low overlap show they can be selected as the synonym predicates to complement each other.

The other metric is **Connectivity**, which computes the harmonic mean between Indegree and Outdegree. It shows how predicates are connected through their entities to other entities. There are two types of connectivity: Indegree, which shows the number of incoming predicates to the entities of a specific predicate, and Outdegree, which shows the number of outgoing predicates from the entities of a specific predicate. The range of Connectivity is between 0.0 and 1.0. For example, if the cardinality of predicates coming into/going out of the entities for pairs of synonym predicates are similar, the Connectivity value is close to 1.0. Otherwise, the Connectivity value is close to 0.0.

**Definition 5.2** (Connectivity - Indegree and Outdegree). Let $G$ be a knowledge graph, and $Pred$ be a specific predicate in $G$. The cardinality of predicates (?$p_0$) going to the entities related to predicate $Pred$ is calculated as Indegree, while the cardinality of predicates (?$p_0$) coming from the entities related to predicate $Pred$ is calculated as Outdegree using the following formula:

\[
\text{Indegree}(Pred, G) = ||\{ (?p_0, \mu(?p_0)) | \mu((?s_1 \text{ Predicate} ?o_1)) \text{ in } G \text{ and } \mu((?o_1 ?p_0 ?s_1)) \text{ in } G \text{ and } \mu(?p_0) = \text{Predicate}) \}|
\]

\[
\text{Outdegree}(Pred, G) = ||\{ (?p_0, \mu(?p_0)) | \mu((?s_1 \text{ Predicate} ?o_1)) \text{ in } G \text{ and } \mu((?s_1 ?p_0 ?o_1)) \text{ in } G \text{ and } \mu(?p_0) = \text{Predicate}) \}|
\]

After computing the cardinality of incoming and outgoing predicates from the entities, we can define Connectivity as an assessment metric, which incorporates both Indegree and Outdegree. The Connectivity is computed as follows:

\[
\text{Connectivity} = \frac{2 \times \text{Indegree} \times \text{Outdegree}}{\text{Indegree} + \text{Outdegree}}
\]
Fig. 6. **Running Example:** A SPARQL query comprising four triple patterns executed over Wikidata does not retrieve any answers. a) In the incompleteness detection phase, the values of PCA are calculated over the mined rules from Wikidata to detect which predicates lead to incomplete answers. The query has four incomplete predicates (NIP). b) Synonym predicate candidates are discovered using Word2Vec, RDF2Vec, and rule-based techniques. c) To select synonym predicates, the **POS** metric is computed, which is the value of overlapping synonym predicates at both levels of synonymous. Synonym pairs with a low overlap are selected as synonym predicates to reformulate the query. d) The query is rewritten to a set of expanded triple patterns using the minimum number of synonym predicates to retrieve the maximum answers.

5.2. Discovering Synonym Predicate Candidates

There are many techniques to discover synonym predicates in the graphs [9–11, 23–25, 27]. The process of determining the candidates of synonym predicates can be done at two levels: a) inside a knowledge graph to identify (Intra-synonym), and b) in a federation of knowledge graphs to uncover (Inter-synonym) predicates. In graph embedding techniques, these candidates sets are generated by converting predicates to the vectors to compute the similarity measure. As a proof of concept, three methods are used in the implementation of SAP-KG: Word2Vec, RDF2Vec, and rule-based.

5.3. Selecting Synonym Predicates

In this step, among the synonym predicates candidates, we select those that lead to provide complementary information in Wikidata. This selection is based on the value of our proposed metric **POS**, which indicates the percentage of overlap between synonym predicates candidates. Our running example, in Figure 6, presents the predicate manner of death (wdt:P1196) as a synonym candidate for the predicate cause of death (wdt:P509). But since the **POS** value (in domain or range) of these synonym candidates is high, they can not be considered as synonym predicates in the next step, 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. Therefore, the **POS** value (in domain or range) helps us to select the synonym predicates which are complementary.

5.4. Query Rewriting

This component is used to transform an input query into an equivalent query that can produce more correct answers. An example can be seen in Figure 6. The input is a SPARQL query and the output is a rewritten query that incorporates the synonym predicates detected in the previous step. There are various methods of query rewriting,
such as query expansion by finding available similar descriptions in the knowledge graph. The process involves parsing the original SPARQL query into its constituent parts, such as the WHERE clause, and then adding more triple patterns to the WHERE clause using synonyms of the predicates of the original query to retrieve additional results. The rewritten query is executed against the knowledge graph and the results are evaluated. Finally, the process entails rewriting the SPARQL query to include a union of similar triple patterns, which involves combining sets of triple patterns that share common subjects and objects. The rewritten query that incorporates synonym predicates returns all the complete results. The goal is to rewrite the query with the minimum number of synonym predicates, while still returning the maximum number of correct answers. The naive approach is to rewrite queries with all possible synonym predicates, but the metrics presented in this work 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. The original queries and their rewritten counterparts are provided in the Appendix B.

6. Empirical Evaluation

We evaluate the performance of SAP-KG in the detection of synonym predicates, and compare the observed outcomes with respect to Word2Vec [7], RDF2Vec [8], and rule-based [10] methods. This assessment is based on an Ideal knowledge graph KG_Ideal that comprises the synonym predicates existing in Wikidata triples in six domains Person, Music, History, Film, Sport, and Drug. The empirical evaluation aims to formulate the following research questions: RQ1) Is SAP-KG able to identify synonym predicate in Wikidata? RQ2) How frequent are synonym predicates in Wikidata? RQ3) Does SAP-KG outperform existing techniques of synonym predicate detection and enhance query completeness?

6.1. Experimental Configuration

Query Benchmark. We designed a benchmark of ten queries from six different domains by analyzing triple patterns answerable by Wikidata knowledge graph. The original queries have between 2 and 4 triple patterns. They are considered to evaluate whether they return complete answers by rewriting them with minimal synonym predicates over Wikidata. Both original queries and rewritten ones can be found in B.

Gold Standard. We created a gold standard as an Ideal knowledge graph to measure the correctness of our approach. It contains all the triples that should be in a knowledge graph. The gold standard is built by defining some SPARQL queries for each domain in the knowledge graph, which are rewritten with synonym predicates described by human. The gold standard consists of all entities that are associated not only with predicates, but also with their synonyms. Therefore, when querying the Ideal knowledge graph, it returns all possible complete answers.

Implementation. Experiments were run on a Windows 10 machine with an Intel i7-9850H 2.6 GHz CPU and 16 GB 1333 MHz DDR3 RAM. We implemented SAP-KG and related metrics in Python 3.7.5. Further, the public SPARQL endpoint of Wikidata is utilized for executing the decomposed subqueries.

Similarity Values of Synonym Predicates. Defining a threshold for similarity values to accept which predicates can be considered as synonym predicates. A threshold \( \tilde{\theta} \) where, \( \tilde{\theta} \) is a user-specified threshold is found to restrict the similarity values. Finding the proper thresholds for computing values of similarity can help to find the correct and relevant synonyms for predicates in the knowledge graph. In other words, it shows for which threshold of similarity value, which predicates can be selected as synonym and which cannot be. Percentiles defined by [28] are applied to calculate the value of such a threshold. The 95\textsuperscript{th} percentile represents the point at which 5\% of the similarity values exceeds that value assigned to the 95\textsuperscript{th} percentile category. Figure 7 depicts the percentile of similarity value. The 95\textsuperscript{th} percentile of the similarity measure is computed with a threshold \( \tilde{\theta} \) of 0.61. That means, only 5\% of the similarity values are higher than the threshold \( \tilde{\theta} \) of 0.61, and 95\% have values less than 0.61.

Metrics. In our experiment, the Connectivity metric computes the harmonic mean between Indegree and Outdegree described in subsection 5.1. This metric is used to compute the connectivity score to show the connectivity of predicates in six different domains in Wikidata. Another way to evaluate the accuracy of our approach is to compare the precision and recall values of the proposed approach with those of the Ideal knowledge graph. Precision: The car-
Similarity Value
95th Percentile

Fig. 7. Synonym Threshold. The figure displays the optimal value for accepting the similarity measure. The 95th percentile of the similarity measure is computed with a threshold of 0.61, which is found to be the most suitable threshold for the rest of the implementation. This means that only predicates with a similarity value greater than 0.61 are considered as candidates for synonym predicates.

Fig. 8. The Connectivity value for some pairs of synonym predicates over six domains in Wikidata are shown. As seen, the Connectivity value is not higher than 0.2. Due to the significant numerical difference in the cardinality of predicates that are coming into or going out of the entities for pairs of synonym predicates in Wikidata, the Connectivity value is expected to be close to 0.0. For example, consider the predicate pairs child(wdt:P40) and mother(wdt:P25). If an entity has fewer outgoing edges with predicate child(wdt:P40) than incoming edges with predicate mother(wdt:P25), then child(wdt:P40) and mother(wdt:P25) can be considered synonym predicates that substitute and complement each other. This proves that many of these pair of synonym predicates need to complete each other to increase the results in query processing. The experiment can also be performed on pairs of synonym predicates in Wikidata and DBpedia to demonstrate how predicates in DBpedia can complement predicates in Wikidata and vice versa.

6.2. Synonyms Predicates in Wikidata

The considered knowledge domains are about Person, Music, History, Film, Sport, and Drug. Figure 9 shows the distribution of synonyms by each predicate in six domains of Wikidata. The x-axis represents each predicate, and the y-axis represents the count of synonyms of predicates. The three color lines on the plots represent the three methods (Word2Vec, RDF2Vec, and rule-based) employed to detect synonym predicates. We observe that for the
Fig. 8. **Connectivity.** The Connectivity values for some pairs of synonym predicates over six domains in Wikidata. The Connectivity value is not high overall, and in the History and Drug domains, this value is much lower compared to the other domains, indicating that the entities with respect to the synonym predicates are incomplete. The synonym predicates can complement each other.

Fig. 9. **Synonym Distribution.** Boxplots were used to display the distribution of synonym predicates in Wikidata across six domains, including Person, Music, History, Film, Sport, and Drug. The synonym predicates were detected using three methods: Word2Vec, RDF2Vec, and rule-based methods. It is observed that predicates in the Person, Music, and Film domains have more synonyms than those in the History, Sport, and Drug domains. The most frequent number of synonyms for predicates in most domains is three.
three methods used, the distribution of synonyms for predicates in Person (Figure 9a), Music (Figure 9b), and Film (Figure 9d) domains is higher than other domains. In addition, predicates in History (Figure 9c), Sport (Figure 9e), and Drug (Figure 9f) domains have less number of synonym predicates compared with other domains. For example, for predicates in domain Drug mostly one synonym predicate can be found; while predicates in domain History have mostly two synonyms. Here, threshold 0.61 is considered for the similarity values. Table 1 summarizes the results depicted in Figure 9. The total percentage of predicates with synonyms for each domain shows that the domain of Person has the highest with 88.66%, while Drug has the lowest with 42.39%. Furthermore, predicates with three synonyms are more prevalent in the domains of Person, Music, and Film, whereas predicates in the Drug domain typically have only one synonym. The metrics in section 5 are proposed for pairs of synonym candidates in Wikidata or in other community-maintained knowledge graphs such as DBpedia to calculate the percentage of overlap of synonyms. Some of these values are reported in Table 2. As mentioned before, the range of value is between 0 and 100%. If the value of pairs of synonym candidates is close to 0, then these predicates can be considered as synonym in query rewriting. The value close to 100% refers that two synonym candidate predicates relate the same number of entities, and can not complement each other. Therefore, they are not considered as synonym predicates.

Table 1
The table describes the distribution of synonym predicates per domain in Wikidata. The threshold 0.61 has been considered for similarity value. The highest percentage of synonyms has been detected in the Person domain at 86.66%, while Drug has the lowest percentage, 42.39%.

<table>
<thead>
<tr>
<th>Wikidata’s Domain</th>
<th>#predicates</th>
<th>% total synonym</th>
<th>#frequent synonym</th>
<th>% frequent synonym</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>3,000*</td>
<td>86.66%</td>
<td>3</td>
<td>46.38%</td>
</tr>
<tr>
<td>Music</td>
<td>683</td>
<td>73.02%</td>
<td>3</td>
<td>44.6%</td>
</tr>
<tr>
<td>History</td>
<td>195</td>
<td>41%</td>
<td>2</td>
<td>56.25%</td>
</tr>
<tr>
<td>Film</td>
<td>1,000*</td>
<td>72.1%</td>
<td>3</td>
<td>71.7%</td>
</tr>
<tr>
<td>Sport</td>
<td>678</td>
<td>44.98%</td>
<td>2</td>
<td>51.47%</td>
</tr>
<tr>
<td>Drug</td>
<td>467</td>
<td>42.39%</td>
<td>1</td>
<td>60.6%</td>
</tr>
</tbody>
</table>

Table 2
Some pairs of synonym candidates with the POS value (in both domain POS-D and range POS-R). The highlighted values indicate that those pairs of synonyms can not be considered as synonyms, since they have high overlap.

<table>
<thead>
<tr>
<th>Pairs of Synonyms</th>
<th>POS</th>
<th>POS-D</th>
<th>POS-R</th>
</tr>
</thead>
<tbody>
<tr>
<td>wdt:P509 - wdt:P1196</td>
<td>87.09%</td>
<td>88.31%</td>
<td>25.92%</td>
</tr>
<tr>
<td>wdt:P8810 - wdt:P25</td>
<td>0.01%</td>
<td>0.01%</td>
<td>0.01%</td>
</tr>
<tr>
<td>wdt:P8810 - wdt:P22</td>
<td>0</td>
<td>0</td>
<td>0.01%</td>
</tr>
<tr>
<td>wdt:P19 - wdt:P1464</td>
<td>1.23%</td>
<td>1.24%</td>
<td>14.75%</td>
</tr>
<tr>
<td>wdt:P40 - wdt:P1038</td>
<td>3.69%</td>
<td>4.6%</td>
<td>4.29%</td>
</tr>
<tr>
<td>wdt:P509 - dbp:deathCause</td>
<td>6.58%</td>
<td>6.59%</td>
<td>87.98%</td>
</tr>
<tr>
<td>wdt:P8810 - dbo:mother</td>
<td>10.15%</td>
<td>8.44%</td>
<td>9.6%</td>
</tr>
<tr>
<td>wdt:P8810 - dbo:father</td>
<td>2%</td>
<td>1.65%</td>
<td>1.92%</td>
</tr>
<tr>
<td>wdt:P40 - dbo:relative</td>
<td>2.05%</td>
<td>2.21%</td>
<td>2.4%</td>
</tr>
<tr>
<td>wdt:P40 - dbo:child</td>
<td>3.1%</td>
<td>3.71%</td>
<td>4.24%</td>
</tr>
</tbody>
</table>

6.3. Accuracy of SAP-KG

We validate our approach by computing the accuracy (precision and recall values). Accuracy refers to how close the result of SAP-KG is to the Ideal knowledge graph. Table 3 reports the performance of SAP-KG approach compared with the techniques for detecting synonym predicates over 10 queries in Wikidata. The precision of all queries in SAP-KG is equal to 1.0 that indicates all the synonym predicates selected for reformulating the queries
are available in our Ideal knowledge graph. SAP-KG correctly detected the synonym predicates among all selected synonym predicates by Word2Vec, RDF2Vec, and rule-based techniques. Moreover, recall values in SAP-KG are also higher than other techniques. As seen, for queries 5, 6, and 8 the values for recall and precision are high in all techniques. Thus, many detected predicates in reformulation queries are linked less resources into the other entities or values. So, reformulating queries with these synonyms has no effect on the number of results. In addition, for queries 1, 2, 3, and 7 the values for precision are too low which means Word2Vec embedding discover many irrelevant predicates as synonym predicates for reformulating the queries.

Table 3

The performance of SAP-KG on Wikidata where the proposed metrics are applied to detect synonym predicates by Word2Vec, RDF2Vec, and rule-based techniques in order to select the synonym predicates is reported. In all queries, precision of SAP-KG is 1.0.

<table>
<thead>
<tr>
<th>Q</th>
<th>Word2Vec</th>
<th>RDF2Vec</th>
<th>Rule-based</th>
<th>SAP-KG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>1</td>
<td>0.1</td>
<td>0.42</td>
<td>0.3</td>
<td>0.42</td>
</tr>
<tr>
<td>2</td>
<td>0.07</td>
<td>0.44</td>
<td>1.0</td>
<td>0.44</td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>0.7</td>
<td>0.58</td>
<td>0.7</td>
</tr>
<tr>
<td>4</td>
<td>0.52</td>
<td>0.97</td>
<td>0.56</td>
<td>0.97</td>
</tr>
<tr>
<td>5</td>
<td>0.89</td>
<td>0.82</td>
<td>1.0</td>
<td>0.82</td>
</tr>
<tr>
<td>6</td>
<td>1.0</td>
<td>0.91</td>
<td>1.0</td>
<td>0.91</td>
</tr>
<tr>
<td>7</td>
<td>0.02</td>
<td>0.12</td>
<td>0.42</td>
<td>0.12</td>
</tr>
<tr>
<td>8</td>
<td>0.99</td>
<td>0.75</td>
<td>0.93</td>
<td>0.75</td>
</tr>
<tr>
<td>9</td>
<td>0.75</td>
<td>0.41</td>
<td>0.75</td>
<td>0.41</td>
</tr>
<tr>
<td>10</td>
<td>0.87</td>
<td>0.7</td>
<td>0.93</td>
<td>0.7</td>
</tr>
</tbody>
</table>

6.4. Enhancing Query Completeness Over Wikidata

We address the issue of enhancing query answering for incomplete community-maintained knowledge graphs like Wikidata. Our analysis reveals that Wikidata contains synonym predicates that can supplement each other. Our study demonstrates that identifying synonym predicates within a single knowledge graph Intra-synonym or across a federation of knowledge graphs Inter-synonym can enhance query answer completeness. The results presented in Table 3 demonstrate that queries rewritten using synonym predicates discovered in Wikidata retrieve answers that are more similar to those retrieved from the Ideal knowledge graph.

6.5. Discussion

Answer to RQ1. The Connectivity values indicate that at least six domains in Wikidata contain entities that are incomplete with respect to their predicates. Low values of Connectivity across all domains suggest that the number of outgoing edges for each entity with respect to predicate \( p_1 \) is lower than the number of incoming edges with respect to predicate \( p_2 \), i.e., \( p_1 \) and \( p_2 \) are complementary predicates. These findings motivate us to used synonym predicates from Wikidata during query processing.

Answer to RQ2. The distribution analysis of synonym predicates in Figure 9 and Table 1 reveals that Wikidata contains predicates that have synonyms. Domains such as Person, Music, and Film exhibit a higher number of discovered synonym predicates, while domains such as History, Sport, and Drug show fewer synonyms for their predicates. The domain Person exhibits the highest number of synonyms per predicate, with a maximum of seven synonyms for a single predicate. Table 2 provides the percentage of overlapping synonym predicates discovered within Wikidata, as well as between Wikidata and DBpedia. The high value of the POS metric (as well as the POS-D and POS-R metrics) indicates that the pairs of predicates are complete and do not need to complement each other.

Answer to RQ3. The results of our study are depicted in Table 3, demonstrate that detecting synonym predicates by SAP-KG approach outperforms other techniques, including embedding (Word2Vec and RDF2Vec) and the frequent item set. Our approach is particularly effective in detecting synonym predicates, which enhances query complete-
ness. The precision-recall values proves that our approach achieves high precision 1.0 for all recall values. These results suggest that the rewritten queries obtained by detecting synonym predicates yield answers that are similar to those retrieved from the Ideal knowledge graph KG_Ideal.

7. Conclusions and Future Work

This study highlights the problem of discovering synonym predicates in community-maintained knowledge graphs such as Wikidata. Because Wikidata’s web interface is open for any user to edit, multiple users can create predicates with different names that refer to the same thing. This means that a knowledge graph may contain numerous synonym predicates. This research presents a knowledge graph-agnostic approach to detect synonym predicates which can complete each other. To substitute and complement each other, certain pairs of synonym predicate candidates need to be selected as synonyms. These pairs of synonym predicates are identified using a Class-based Synonym Description (CSD) concept, which employs diverse metrics to capture synonyms at the class level and measure the overlap between predicates. Finally, these synonym predicates with low overlap can be utilized to rewrite queries and enhance query answer completeness. The empirical evaluation validates the presence of synonym predicates in Wikidata. As a result, 86.66% of the predicates in the Person domain have synonyms. Furthermore, the experimental results demonstrate that the SAP-KG approach achieves a high precision of 1.0, which is comparable to that of the Ideal knowledge graph. The main objective of this work is to identify synonym predicates that can complement predicates in Wikidata and improve answer completeness during query processing. We hope that our method will help researchers recognize the presence of numerous synonym predicates in Wikidata and leverage them to improve the quality of their research. In future work, we will define several execution plans for reformulating SPARQL queries, and aim to select the plan that maximizes the number of results while minimizing the number of synonym predicates used in the query engines.

Acknowledgements

This work has been partially supported by the EU H2020 RIA funded project CLARIFY (GA No. 875160) and EraMed project P4-LUCAT (GA No. 53000015). Furthermore, Maria-Esther Vidal is partially supported by the Leibniz Association in the program "Leibniz Best Minds: Programme for Women Professors", project TrustKG-Transforming Data in Trustable Insights with grant P99/2020.

References


Appendix A. Computing the Partial Completeness Assumption (PCA) in Wikidata

A.1. The SPARQL query to compute the PCA confidence score for the mined rule where parent (wdt:P8810) is in Body and father (wdt:P22) or mother (wdt:P25) is in Head.

1) PREFIX wdt: <http://www.wikidata.org/prop/direct/>
SELECT (xsd:float(?Support)/xsd:float(?PCABodySize) AS ?PCA)
WHERE {
    SELECT (COUNT(DISTINCT *) as ?Support) WHERE {
    }
    SELECT (COUNT(DISTINCT *) as ?PCABodySize) WHERE {
    }
}

2) PREFIX wdt: <http://www.wikidata.org/prop/direct/>
SELECT (xsd:float(?Support)/xsd:float(?PCABodySize) AS ?PCA)
WHERE {
    SELECT (COUNT(DISTINCT *) as ?Support) WHERE {
    }
    SELECT (COUNT(DISTINCT *) as ?PCABodySize) WHERE {
    }
}
Appendix B

B.1. List of entities and predicates used in this paper with their descriptions from Wikidata.

Table 4

<table>
<thead>
<tr>
<th>Wikidata Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wd:Q3290404</td>
<td>Marella Agnelli</td>
</tr>
<tr>
<td>wd:Q11085</td>
<td>Parkinson’s disease</td>
</tr>
<tr>
<td>wd:Q23882021</td>
<td>Charles Eugster</td>
</tr>
<tr>
<td>wd:Q84</td>
<td>London</td>
</tr>
<tr>
<td>wd:Q45864</td>
<td>John McAfee</td>
</tr>
<tr>
<td>wd:Q10737</td>
<td>suicide</td>
</tr>
<tr>
<td>wd:Q175111</td>
<td>hanging</td>
</tr>
<tr>
<td>wdt:P25</td>
<td>mother</td>
</tr>
<tr>
<td>wdt:P22</td>
<td>father</td>
</tr>
<tr>
<td>wdt:P1038</td>
<td>relative</td>
</tr>
<tr>
<td>wdt:P40</td>
<td>child</td>
</tr>
<tr>
<td>wdt:P26</td>
<td>spouse</td>
</tr>
<tr>
<td>wdt:P8810</td>
<td>parent</td>
</tr>
<tr>
<td>wdt:P1196</td>
<td>manner of death</td>
</tr>
<tr>
<td>wdt:P509</td>
<td>cause of death</td>
</tr>
<tr>
<td>wdt:P1050</td>
<td>medical condition</td>
</tr>
<tr>
<td>wdt:P19</td>
<td>place of birth</td>
</tr>
<tr>
<td>wdt:P20</td>
<td>place of death</td>
</tr>
<tr>
<td>wdt:P1647</td>
<td>subproperty of</td>
</tr>
<tr>
<td>wdt:P1464</td>
<td>category for people born here</td>
</tr>
<tr>
<td>wdt:P2175</td>
<td>medical condition treated</td>
</tr>
<tr>
<td>wdt:P274</td>
<td>chemical formula</td>
</tr>
<tr>
<td>wdt:P4844</td>
<td>research intervention</td>
</tr>
<tr>
<td>wdt:P276</td>
<td>location</td>
</tr>
<tr>
<td>wdt:P131</td>
<td>located in the administrative</td>
</tr>
<tr>
<td>wdt:P184</td>
<td>doctoral advisor</td>
</tr>
<tr>
<td>wdt:P21</td>
<td>sex or gender</td>
</tr>
<tr>
<td>wdt:P1066</td>
<td>student of</td>
</tr>
<tr>
<td>wdt:P66</td>
<td>composer</td>
</tr>
<tr>
<td>wdt:P170</td>
<td>creator</td>
</tr>
<tr>
<td>wdt:P495</td>
<td>country of origin</td>
</tr>
<tr>
<td>wdt:P166</td>
<td>award received</td>
</tr>
<tr>
<td>wdt:P39</td>
<td>position held</td>
</tr>
<tr>
<td>wdt:P103</td>
<td>native language</td>
</tr>
<tr>
<td>wdt:P136</td>
<td>genre</td>
</tr>
</tbody>
</table>

B.2. The original queries and the rewritten ones by discovered synonym predicates in six different domains in Wikidata.
Table 5

<table>
<thead>
<tr>
<th>Q</th>
<th>Original Query</th>
<th>Rewritten Query</th>
<th>Q</th>
<th>Original Query</th>
<th>Rewritten Query</th>
</tr>
</thead>
</table>