Enhancing Query Answer Completeness with Query Expansion based on Synonym Predicates

Emetis Niazmand
supervised by Prof. Dr. Maria-Esther Vidal
Leibniz Universität Hannover
TIB Leibniz Information Centre for Science and Technology
Hannover, Germany
Emetis.Niazmand@tib.eu

ABSTRACT

Community-based knowledge graphs are generated following hybrid approaches, where human intelligence empowers computational methods to effectively integrate encyclopedic knowledge or provide a common understanding of a domain. Existing community-based knowledge graphs represent essential sources of knowledge for enhancing the accuracy of data mining, information retrieval, question answering, and multimodal processing. However, despite the enormous effort conducted by contributing communities, community-based knowledge graphs may be incomplete and integrate duplicated data and metadata. We tackle the problem of enhancing query answering against incomplete community-based knowledge graphs by proposing an efficient query processing approach to estimate answer completeness and increase the results. It assumes that community-based knowledge graphs comprise synonym predicates that complement knowledge graph triples required to raise query answering completeness. The aim is proposing a novel query expansion method based on synonym predicates identified using embeddings built on a knowledge graph. Our preliminary analysis shows that our approach improves query answer completeness. However, queries can be expanded with some similar predicates which do not lead to complete answers. This shows that more work is required for query expansion with the minimum synonym predicates that maximize answer completeness.

CCS CONCEPTS

• **Information systems** → **Query optimization**; **Query planning**.

KEYWORDS

Knowledge Graph, Synonym Detection, Knowledge Graph Embedding, Query Expansion, Query Processing

ACM Reference Format:
Emetis Niazmand, supervised by Prof. Dr. Maria-Esther Vidal. 2022. Enhancing Query Answer Completeness with Query Expansion based on Synonym Predicates. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3487553.3524198

1 INTRODUCTION

Knowledge graphs have become a popular formalism for representing entities and their properties. Meanwhile, the size of knowledge graphs such as DBpedia [12] or WikiData [21] has increased constantly by growing the number of entities and their properties. For example, the number of total entities in DBpedia has passed from 78,412,721 (version 2015-04) to 135,355,098 (version 2016-10) [1]. Albeit enhancing the amount of represented structured knowledge, the frequent growth of entities modeled in encyclopedic knowledge graphs impacts on query management tasks like query processing. Nevertheless, knowledge graphs based on Open-World Assumption (OWA) can be incomplete by default, which leads query engines to retrieve incomplete answers. Information incompleteness is a main data quality issue that is aggravated by containing synonyms and duplicate data and metadata in the knowledge graphs; furthermore, entities and properties may have synonymous. Abedjan et al. [1], our baseline, discover synonym predicate pairs that substitute each other in the data and are good candidates for query expansions; it is built on top of synonymously used relationships by providing frequent item set mining-based techniques. Although the experimental study reported by Abedjan et al. shows the accuracy of mined synonym predicates, it is conducted over a small dataset, and does not reflect all the properties of community-based knowledge graphs. Also, they do not provide a query answering method to show the completeness of answers after expanding queries with synonym predicates. Moreover, due to the semi-structured nature of RDF data, there is no easy way to detect incompleteness with negative effects on query processing. Acosta et al. [2] present a hybrid SPARQL engine to enhance answer completeness via crowdsourcing. Although this work proposes a novel hybrid engine, knowledge about synonym predicates could also be used to guide the crowd, thus, reducing uncertainty [4]. We aim to devise query processing methods to overcome these limitations by exploiting knowledge about synonym predicates in knowledge graphs.

2 MOTIVATION

DBpedia comprises information about *Person*, *Music*, *Sport*, *Drug*, *Film*, and *History*. As other community-based knowledge graphs, DBpedia may suffer from incompleteness and integrate duplicated data and metadata. We motivate our work by considering a SPARQL

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France.
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524198

[1] Based on statistics published in http://downloads.dbpedia.org/

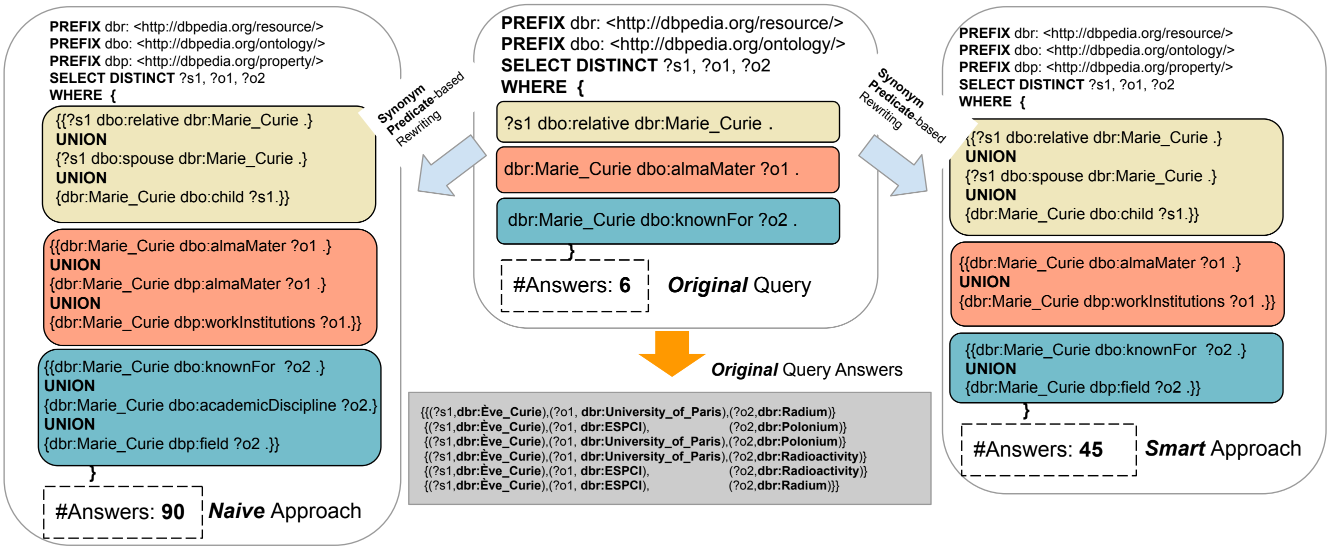

Figure 1: SPARQL query executed over DBpedia (version2016-10) and two queries generated by rewriting triple patterns using synonym predicates. The original SPARQL query consists of three triple patterns and retrieves six answers. The reformulated SPARQL query to a set of expanded triple patterns with a naive approach retrieves 90 answers. A reformulated SPARQL query to a set of expanded triple patterns with a smart approach retrieves 45 answers.

query over DBpedia (version 2016-10) as in Figure 1; it retrieves, for *dbr:Marie_Curie*, names of relatives, institutes, and research fields. This query consists of three triple patterns and returns six answers. As seen in our motivating example, Figure 1, the SPARQL query and the transformed ones with two different approaches, naive and smart, executed against DBpedia (version 2016-10) return different number of answers. Synonym predicates are present, and may not have the same number of instances. For example, both predicates *dbo:relative* and *dbo:child* are semantically similar to each other, but *dbo:relative* relates *dbr:Marie_Curie* to *dbr:Ève_Curie*, while *dbo:child* associates this resource with two entities *dbr:Ève_Curie* and *dbr:Irène_Joliot-Curie*. To enable answer completeness, both entities that represent Marie Curie's children should be part of *dbo:relative*, and not only one. Query expansion can be used to enhance answer completeness; it is the process of rewriting and transforming a query into other forms in order to improve the performance of information extraction. Query expansion may be done by finding available synonym descriptions in the knowledge graph, called representatives, and reformulating the query by these synonyms. A query expanded by a naive approach in Figure 1 includes all possible representatives for predicates; it returns 90 results. In this naive approach, each triple pattern is expanded by synonym predicates to return more answers. But many duplicated and incorrect answers can be found among of these 90 answers.

Therefore, the technique of expanding existing queries with similar descriptions does not always lead to retrieving the complete answers. In Figure 1, a smart approach is provided to expand the original query by detecting synonym predicates efficiently and consider only the representatives which return the complete answers without duplication in terms of semantic meaning in the results. As seen, the synonym predicates which returns incomplete and duplicate answers (e.g., *dbp:almaMater* and *dbo:academicDiscipline*) should not be part of the rewriting. Therefore, we aim at expanding the query with the minimal number of synonym predicates to increase answer completeness.

3 STATE OF THE ART

3.1 Query Answer Completeness

Several approaches have been developed to enhance the completeness of query answering. Initial works correspond to Motro [14] who formalizes the definition of partial completeness in terms of query completeness (QC) statements, which express that the answer of a query is complete. It can determine whether each answer of a user query is complete, or whether any subsets of it are complete. Therefore, a given query is complete whenever a set of other queries are complete. Later, Levy [13] extended this idea by expressing partial completeness of an incomplete database to show how to derive query answer completeness from them. Lastly, Razniewski and Nutt [18] proposed an approach based on the intersection of the statements introduced by Motro and Levy, expressed by selections on database tables, to specify complete parts of database tables. They showed how to adapt the operators of relational algebra to manipulate these completeness patterns to compute completeness patterns related to query answers. HARE [2], a hybrid SPARQL query engine, exploits a micro-task mechanism for enhancing the completeness of query answers using crowdsourcing. It uses a model to estimate the completeness of RDF datasets. HARE can identify the parts of a query that retrieve missing answers and yield incomplete results. However, using micro-task crowdsourcing may require a great deal of time, which may cost more effort and money.

We also aim at enhancing completeness of query answers in an efficient and effective manner, but our approach resorts to predicates in community-based knowledge graphs which are synonyms.

3.2 Query Expansion

Various techniques for expanding queries have been conducted with the aim of completing query answers by retrieving more relevant answers. Query expansion is done by reformulating queries with similar entities and properties. This is related to both record linkage [17], and ontology matching [20] where the aim is finding identical concepts. As an example, the technique Elbassuoni et al. [8] provided, is based on language modeling to find a match which is both relevant and close in spirit to a given entity or relation. Ghali et al. [7] proposed a probabilistic query expansion method to search within a set of candidate terms for the most relevant terms for the initial query to expand. The output is a set of terms that are candidates for expanding the query and their values of correlation with the whole query. In this paper, we focus on embedding models to discover candidates which are most relevant to the predicates in the query for query expansion.

3.3 Identifying Synonym Predicates

Most of the approaches to uncover synonyms for terms are based on natural language processing or information retrieval techniques. Among of these approaches, searching for co-occurrence of synonym candidates in unstructured data, such as web documents, is more common and requires natural language processing rules. For example, Baroni et al. [5] proposed an approach that calculates the ratio of co-occurrence of two terms, while the pairs of candidates are already available for their validations. They look for pairs of synonyms among pairs of unrelated terms, but without distinguishing between synonyms and other semantically related terms. Nevertheless, few researches have been done on detecting synonyms in knowledge graphs and RDF data. One example is the approach proposed by Abedjan et al. in [1] based on aggregating positive and negative association rules at statement level. With this approach, they discover overlaps between attribute values in an RDF data. They discover frequent item sets for each property consisting of object entities. Properties with high overlap with respect to their objects and with low overlap in their subjects are identified as synonym. The other work is [11] which provides a technique for detecting synonymous properties in large knowledge graphs by mining interpretable definitions of properties using association rule mining. In our baseline [1], the assumption is that properties never occur together for the same subject entities. We do not consider this assumption, while there are some properties in knowledge graphs with overlaps, but they are synonyms.

3.4 Knowledge Graph Embedding

Knowledge graph embedding provides techniques to be used in diversity applications [22] such as knowledge graph completion by predicting new triples in knowledge graphs, relation extraction, question answering, query expansion, finding synonymous relationships, etc. There are few works which consider knowledge graph embedding models to solve the problem of detecting synonym relationships. Kalo et al. [10] used a property of knowledge graph embeddings for detecting synonym relationships, such as TransH [23] and TransD [9] which are applied in this work as well. They showed how the representation of entities and relationships can be used to measure semantic similarity by applying distance metrics on the vectors. Similarly, our approach relies on knowledge graph embeddings (e.g., RDF2vec [19], TransH [23] and TransD [9]), and determines relatedness between predicates based on similarity measures computed on embeddings.

4 RESEARCH PROBLEM AND PROPOSED APPROACH

We define the problem of enhancing query answer completeness by query expansion.

Problem Statement. Given an RDF graph $G = (V, E, L)$ and a SPARQL query $Q = \{t_1, t_2, ..., t_n\}$ consisting of multiple RDF triple patterns t_i, $(i = 1, ...n)$. Consider an ideal RDF graph $G\_Ideal = (V', E', L')$, where all the triples and answers are in this graph. The problem of enhancing query answering by expanding Q as $Q' = \{t'_1, t'_2, ..., t'_n\}$ meets the following condition:

- Q' executed over G enhances answer completeness in comparison to the execution of Q against G. The set of entities and properties in Q' are semantically similar to Q.

$$[[Q]]^{\mathrm{G}} \subseteq [[Q']]^{\mathrm{G}} \subseteq [[Q]]^{\mathrm{G\_Ideal}} \tag{1}$$

Proposed Solution. We propose our solution to enhance the completeness of query answering by expanding queries based on detecting synonym properties exist in the knowledge graph. We aim at expanding queries with a minimal number of synonymous properties that maximize answer completeness. Therefore, we aim to formulate the following research questions for the thesis:

- **RQ1:** Is our approach able to retrieve complete answers using expanded queries with minimal synonym predicates?
- **RQ2:** How detecting synonymous properties for query expanding can improve query answer completeness?
- **RQ3:** How estimating the completeness in advance can help to reduce the cost by avoiding unnecessary expanding queries?

The general architecture for our approach is presented in Figure 2. It receives a SPARQL query and a knowledge graph and outputs the enhanced answer of the query. The architecture comprises four components: a) RDF Completeness Model; b) Detecting Synonym Predicates; c) Query Reformulation; and d) Query Engine.

RDF Completeness Model. This model estimates the completeness of a SPARQL query whenever it is evaluated against a knowledge graph. Since query expansion is an expensive task and does not always retrieve the complete answers, query expansion cost can be mitigated by estimating which queries retrieve complete answers in advance. The model resorts to the synonym predicates to assess a triple pattern completeness.

Detecting Synonym Predicates. Synonym predicates detector component for expanding the queries comprises two main tasks, generating candidate set and pruning them. Detecting synonymous predicates in RDF graphs can be done by several techniques, such as knowledge graph embeddings computation to find representatives for properties exist in the knowledge graph, association rules, and so on. There are many graph embeddings approaches to find

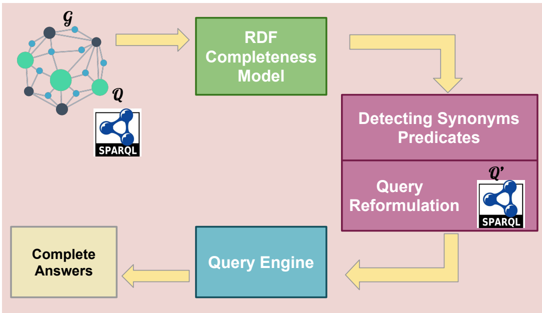

Figure 2: The general proposed architecture for expanding given a query Q as Q' and querying over an RDF knowledge graph G to retrieve complete answers.

representatives for properties, such as Graph2vec [15], sub2vec [3] (embed subgraphs), subgraph2vec [16], and RDF2vec [19]. Since, we deal with RDF knowledge graphs, using RDF2vec embeddings can help us to trace into the knowledge graph and find neighborhoods of entities and properties. In knowledge graph embedding techniques, the candidates sets are generated by converting properties to the vectors and find similarity and relatedness between them by measuring cosine similarity. In association rules technique, the synonymous properties are found by mining interpretable definitions of properties [11]. After generating candidate sets, we need to prune ones with low similarity values by grouping the most similar ones to stay in the same groups. Various methods can be used for grouping, such as Locality Sensitive Hashing (LSH), clustering, or community detection. The algorithm used here is based on Locality Sensitive Hashing; it hashes similar items into the same buckets [6].
Query Reformulation. The main idea of reformulating the SPARQL query to one with the union of similar triple patterns is rewriting the query by minimal representatives or synonymous predicates, which returns maximal results. The rewritten query is executed against a SPARQL endpoint and returns complete answers. We will devise heuristic- and cost-based query optimization techniques to generate efficient executions making use of the synonym predicates.

5 METHODOLOGY

Our work is based on the assumption that, while incomplete, community-based knowledge graphs comprise synonym predicates that complement each other. As a result, expanding queries with synonym predicates may yield increase in the answer completeness. We formulate three research questions to validate our hypotheses and conduct literature review to identify the related approaches from the state of the art. As a result, we devise an abstract architecture that aims to identify query plans able to increase answer completeness while reducing execution time. To answer the first research question, **RQ1**, the expansion query technique resorts to the RDF completeness model and novel query rewriting techniques to reformulate queries with a minimal number of synonym predicates. Regarding **RQ2**, our proposed approach exploits contextual knowledge during the identification of synonym predicates. We have studied frequent item set mining-based technique, as our baseline, and embedding knowledge graph techniques to discover the

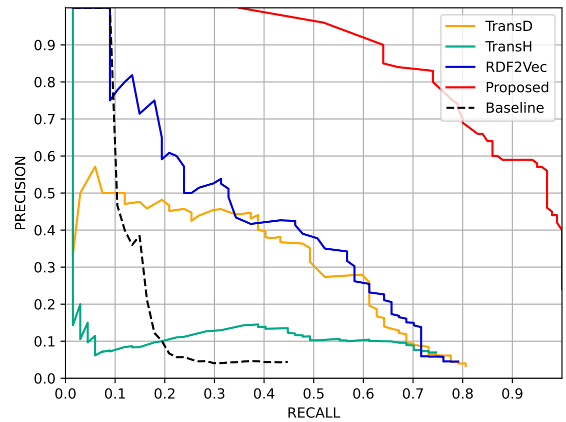

Figure 3: Precision-Recall-Curves with Cosine Similarity results on DBpedia to show the comparison between three embedding knowledge graph techniques (TransD, TransH, and RDF2vec), baseline with frequent item set mining-based technique, and our proposed technique.

performance of these techniques to discover synonymous predicates. Since query reformulation do not always return the complete answers, by applying the RDF completeness model in advance, we ensure whether the original SPARQL query returns the complete answer. We aim to identify efficient query plans and to answer **RQ3**. The behavior of our proposed approach is validated formally and empirically. At the formal level, we will demonstrate the correctness of query writings generated by the query expansion techniques, as well as their time and space complexity. We will conduct experimental studies on various community-based knowledge graphs to assess our research questions.

6 INITIAL EXPERIMENTAL STUDY

We report on the empirical assessment of the accuracy of our initial approach. We have created gold standards (i.e., ideal RDF graphs) and compute the precision and recall for each studied query.

6.1 Experimental Configuration

Benchmark. We conducted our evaluation over a total of 60 queries from six different domains in the DBpedia knowledge graph. The endpoint used to retrieve the results is based on DBpedia, version 2016-10. The considered knowledge domains are about *Music*, *Sport*, *Person*, *Drug*, *Film*, and *History*. For our evaluation, 10 queries for each knowledge domain are selected which do not return complete results due to incomplete portions of the knowledge graph. They are considered to evaluate whether they return complete answers by expanding with minimal synonym predicates over the RDF graph.
Baseline. We start with implementation of the `Range Content Filtering` and `Reversed Correlation Coefficient` as described in the baseline for synonym detection [1]. In this baseline, the assumption is that properties never occur together for the same subject entities, which is not considered in our experiments. While,

in community-based knowledge graphs, many entities and their triples can be inserted by different scientists or from different data sources. We also compare our approach with three knowledge graph embedding techniques (e.g., TransD, TransH, and RDF2vec). For embedding the properties in RDF graphs, the embedding approach RDF2vec is used. Different configuration have been applied to find the most relevant representatives to expand the queries. The RDF2vec configurations used to extract the embeddings are defined as the depth of the walks, the number of walks per entity, the walk strategy, and the vector size.

Metrics. Accuracy of the approach is assessed by computing and comparing the precision and recall values of the proposed approach by applying selected queries over different knowledge domains.

Implementation. Our approach is implemented using Python 3.7.5. The public SPARQL endpoint of DBpedia (version 2016-10) is utilized for executing the decomposed subqueries.

6.2 Discussion

The results of our study are depicted in Figure 3; it compares our proposed approach with the baseline (frequent item set) and embedding techniques (RDF2vec, TransD, and TransH). They suggest that our approach is able to enhance query completeness by detecting the synonym predicates efficiently compared with other approaches. The precision-recall curves in Figure 3, indicate that the proposed approach outperforms the baseline and other embedding techniques. For a recall from 0.3 to 0.8, the proposed approach achieves very high precision. This indicates that the answers retrieved from expanded queries by detected synonym predicates are more similar to the answers retrieved from the ideal RDF graph. However, the baseline achieves the highest precision at a recall less than 0.1, but then drops to a precision of 0.05. Moreover, we can observe that RDF2vec outperforms TransD and TransH in identifying synonym relationships. This suggests that contextual knowledge plays a crucial role in the identification of synonym predicates.

7 LESSONS LEARNED AND FUTURE WORK

Community-based knowledge graphs may suffer from incompleteness and integrate duplicated data and metadata. Therefore, the issue of the incompleteness of query answers should be addressed. Proposing a mechanism to retrieve complete information due to quality issues in community-based knowledge graphs has been essential for many years. Hence, providing a suitable query answering method to achieve answer completeness against incomplete knowledge graphs. The currently proposed completeness models do not consider similar and duplicated entities and properties existing in knowledge graphs. Enhancing query answer completeness with query expansion based on synonym predicates will be a solution to retrieve complete results. We will devise query processing techniques to exploit contextual knowledge from knowledge graphs to solve the problem efficiently.

ACKNOWLEDGMENTS

This work has been partially supported by the EraMed project P4-LUCAT (GA No. 53000015).

REFERENCES

[1] Ziawasch Abedjan and Felix Naumann. 2013. Synonym Analysis for Predicate Expansion, Vol. 7882. 140–154. https://doi.org/10.1007/978-3-642-38288-8_10

[2] Maribel Acosta, Elena Simperl, Fabian Flöck, and Maria-Esther Vidal. 2017. Enhancing answer completeness of SPARQL queries via crowdsourcing. *Web Semantics: Science, Services and Agents on the World Wide Web* 45 (07 2017). https://doi.org/10.1016/j.websem.2017.07.001

[3] Bijaya Adhikari, Yao Zhang, Naren Ramakrishnan, and B. Aditya Prakash. 2018. Sub2Vec: Feature Learning for Subgraphs. In *PAKDD*.

[4] Mohammad Allahbakhsh, Boualem Benatallah, Aleksandar Ignjatović, Hamid R. Motahari Nezhad, Elisa Bertino, and Schahram Dustdar. 2013. Quality Control in Crowdsourcing Systems: Issues and Directions. *Internet Computing, IEEE* 17 (03 2013), 76–81. https://doi.org/10.1109/MIC.2013.20

[5] Marco Baroni and Sabrina Bisi. 2004. Using cooccurrence statistics and the web to discover synonyms in a technical language. (01 2004).

[6] Mayur Datar, Piotr Indyk, Nicole Immorlica, and Vahab Mirrokni. 2004. Locality-sensitive hashing scheme based on p-stable distributions. *Proceedings of the Annual Symposium on Computational Geometry*. https://doi.org/10.1145/997817.997857

[7] Btihal El Ghali, Abderrahim El Qadi, Mohamed Ouadou, and Driss Aboutajdine. 2012. Probabilistic Query Expansion method using recommended past user queries. 406–411. https://doi.org/10.1109/INTECH.2012.6457806

[8] Shady Elbassuoni, Maya Ramanath, and Gerhard Weikum. 2011. Query Relaxation for Entity-Relationship Search, Vol. 6643. 62–76. https://doi.org/10.1007/978-3-642-21064-8_5

[9] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, 687–696. https://doi.org/10.3115/v1/P15-1067

[10] Jan-Christoph Kalo, Philipp Ehler, and Wolf-Tilo Balke. 2019. Knowledge Graph Consolidation by Unifying Synonymous Relationships.

[11] Jan-Christoph Kalo, Stephan Mennicke, Philipp Ehler, and Wolf-Tilo Balke. 2020. Detecting Synonymous Properties by Shared Data-Driven Definitions. In *The Semantic Web*, Andreas Harth, Sabrina Kirrane, Axel-Cyrille Ngonga Ngomo, Heiko Paulheim, Anisa Rula, Anna Lisa Gentile, Peter Haase, and Michael Cochez (Eds.). Springer International Publishing, Cham, 360–375.

[12] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. 2014. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal* 6 (01 2014). https://doi.org/10.3233/SW-140134

[13] Alon Y. Levy. 1996. Obtaining Complete Answers from Incomplete Databases. In *In Proc. of the 22nd Int. Conf. on Very Large Data Bases (VLDB'96*. 402–412.

[14] Amihai Motro. 1986. Completeness Information and Its Application to Query Processing. In *VLDB'86 Twelfth International Conference on Very Large Data Bases, August 25-28, 1986, Kyoto, Japan, Proceedings*, Wesley W. Chu, Georges Gardarin, Setsuo Ohsuga, and Yahiko Kambayashi (Eds.). Morgan Kaufmann, 170–178. http://www.vldb.org/conf/1986/P170.PDF

[15] Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. 2017. graph2vec: Learning Distributed Representations of Graphs. *CoRR* abs/1707.05005 (2017). arXiv:1707.05005 http://arxiv.org/abs/1707.05005

[16] Annamalai Narayanan, Chandramohan Mahinthan, Lihui Chen, Yang Liu, and Santhoshkumar Saminathan. 2016. subgraph2vec: Learning Distributed Representations of Rooted Sub-graphs from Large Graphs. (06 2016).

[17] Felix Naumann and Melanie Herschel. 2010. An Introduction to Duplicate Detection. In *An Introduction to Duplicate Detection*.

[18] Simon Razniewski, Flip Korn, Werner Nutt, and Divesh Srivastava. 2015. Identifying the Extent of Completeness of Query Answers over Partially Complete Databases. 561–576. https://doi.org/10.1145/2723372.2750544

[19] Petar Ristoski, Jessica Rosati, Tommaso Di Noia, Renato De Leone, and Heiko Paulheim. 2018. RDF2Vec: RDF graph embeddings and their applications. *Semantic Web* 10 (08 2018), 1–32. https://doi.org/10.3233/SW-180317

[20] Steffen Staab and Rudi Studer. 2003. *Handbook on Ontologies*. https://doi.org/10.1007/978-3-540-92673-3

[21] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[22] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering* PP (09 2017), 1–1. https://doi.org/10.1109/TKDE.2017.2754499

[23] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence* (Québec City, Québec, Canada) *(AAAI'14)*. AAAI Press, 1112–1119.