A Query Anonymization Approach using Ontology Mappings

Takuya Adachi[1] and Naoki Fukuta[2]

[1] Department of Informatics, Graduate School of Integrated Science and Technology, Shizuoka University
[2] College of Informatics, Academic Institute, Shizuoka University
{adachi.takuya.17@, fukuta@inf.}shizuoka.ac.jp

Abstract. In this paper, we propose an ontology matching mechanism for SPARQL query anonymization in a cooperative SPARQL query editing system. To provide a privacy-protection of SPARQL queries on the system, we proposed a mapping-based conversion of a query to a "semantically equivalent or very similar" query by using another ontologies. The mapping-based conversion uses ontology mappings to anonymize what the user is investigating in the query. Our mechanism uses two measures such as graph distance and semantic similarity, and also allows us to check anonymized queries whether it is a query according to user's preference by selecting the mappings from mapping candidates.

Keywords: SPARQL, ontology mapping, cooperative query editing, privacy protection, query anonymization

1 Introduction

A "query" is an intellectual resource among many on the vast web of multimedia content[9]. It is difficult for us to generate applicable queries for various resources since sometimes we have no particular knowledge about these resources. Also, structured query languages are quite expressive, yet they require an array of technical skills and knowledge on query language, syntax, and domain schema[20]. There are some approaches to support Linked Open Data retrievals from data resources, which are keyword-based, form-based and faceted search work[5][7]. These approaches would be helpful for us to retrieve various data from data resources as well as to solve elementary skills and knowledge. Other approaches have been also presented to utilize ontology mappings[15] in order to support the coding process of a SPARQL query for users who are unfamiliar to code it[8][12]. There approaches would be helpful for us to solve the complexity of domain-specific ontologies.

We focus on how to make complex queries such as federated queries across data resources and the single-stop entry point for Linked Open Data[3]. A possible idea is to support to have a help from another person. However, when we try

[3] LOD4ALL: https://lod4all.net/

to ask more people to involve, in some case it makes also necessary to keep the context of the query secret and private to avoid revealing what data are targeted to be searched when the person who would help is outside from the institution of the user or other reasons for privacy issues.

In this paper, we propose an ontology matching mechanism for SPARQL query anonymization in a cooperative SPARQL query editing system. We consider the requirement of anonymized queries and domain anonymization measures.

2 Motivation

We consider a possible scenario to use a cooperative SPARQL query editing system. A user would like to make a SPARQL query, which retrieve media genres and the count of genres for each year from media resources. However, the user does not have skills or knowledge about how to count items by using **GROUP BY** clause. It might be difficult for users to solve this issue by using some existing services. The user seeks a help from another person, then the user discusses the query on the cooperative SPARQL query editing system. In this case, it is a chance for another person to know what data are targeted to be searched. The user would like to keep the context of the query secret and private to avoid revealing them. We therefore try to anonymize SPARQL queries to discuss it and to keep the context of the query secret.

Figure 1 shows the workflow of the cooperative SPARQL query editing system. When a user input a SPARQL query, its comment, and some information on our system, our system would convert the query to an anonymized query and send another person. After a helper receives the anonymized query, the helper would revise and discuss it. Then our system reconverts the revised query to an original-domain query for the user.

3 MCHA SPAIDA

3.1 Overview

We are implementing a cooperative SPARQL query editing system, named MCHA SPAIDA[4], which is an extended version of our previously implemented system SPAIDA[2] for utilizing ontology mappings on SPARQL queries[1][2][4] which also includes anonymous helper mechanism MCHA for cooperatively editing and sophisticating queries. By collecting the processes of executing queries and checking results on this system, we aim to provide an environment for accumulating know-how of coding SPARQL queries in this system.

Figure 2 and Figure 3 show overviews of MCHA SPAIDA. A user makes an original query, that is to be reviewed and commented by anonymous helpers but it contains some information to be hidden from them. The user also describes

[4] An initial idea of this approach will be presented in [3].

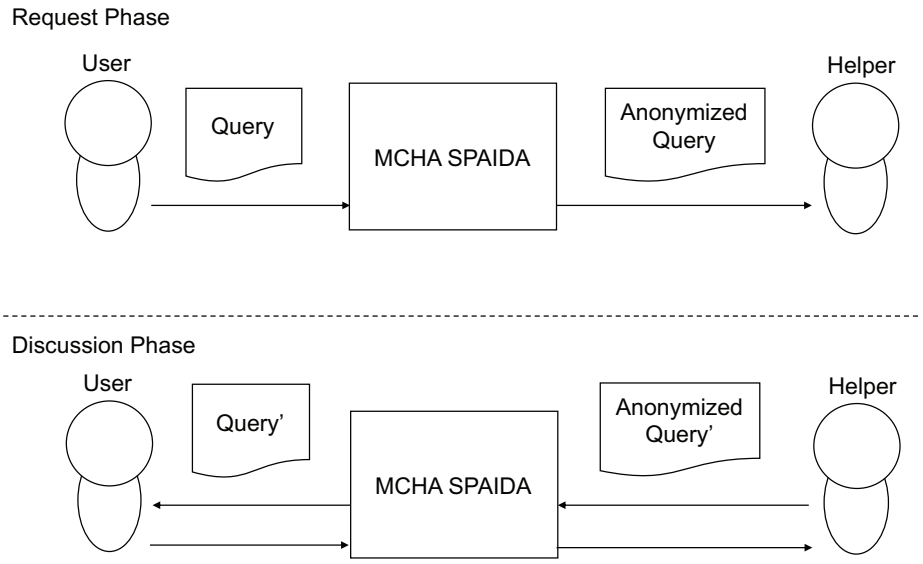

Fig. 1. The workflow of the cooperative SPARQL query editing system

a comment of this query, that is what the user would have for a support. After that, this system generates mapping candidates and anonymized queries (Figure 2). An anonymized query uses ontology mappings, that might be useful for us to anonymize SPARQL queries. Mapping candidates show us ontologies and anonymized measures. This system allows helpers to discuss a SPARQL query with the endpoint and its comment, and to help editing and enhancing a query (Figure 3). Here, the modified query can be transformed into the original form using these ontology mappings.

We are implementing a prototype system as a web application with SPARQL query editors and anonymous helpers. Figure 4 shows a system structure of MCHA SPAIDA. A user can browse the information by using a web application created by React, and MCHA SPAIDA is implemented using Scala and Play Framework, which is a web application framework.

3.2 Query Anonymization

When a user discusses a SPARQL query with SPARQL experts as helpers, our system could convert the query to an anonymized query by using an application of the on-the-fly mapping generation mechanism. This mechanism is MCHA (Mapping-based Conversion for Human-based query writing Assistance), which rather convert a query to another query which targets to completely different things while it tries to keep their attributes in the sense of complexity and structure of the output.

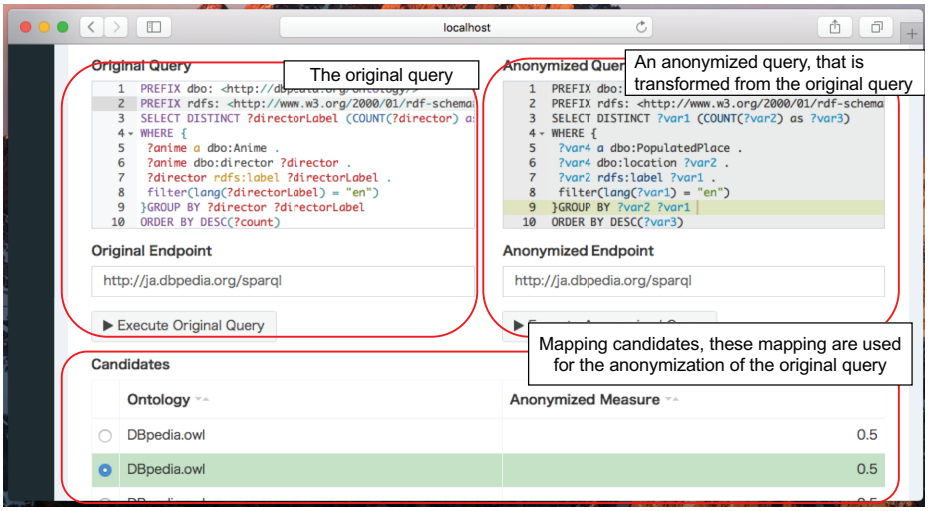

Fig. 2. An overview of MCHA SPAIDA (1)

In Wikidata SPARQL Logs[5], they describe the definition of anonymised query, that is the query reformatted and processed for reducing identifiability and this string is URL-encoded. The query strings were processed to remove potentially identifying information as far as possible, and to reduce spurious signals that could be used to reconstruct user traces[5]. Figure 5 shows an anonymised query example in Wikidata SPARQL Logs[5]. Note that, in query (c) we have removed the comments which have appeared in the original example in order to make it easier to read. However, this anonymization approach does not hide what the query is intended to search and the actual results to be searched.

Here, we consider the three requirements of anonymized queries. First requirement is to anonymize the targets of an original query. It is better to use heterogeneous ontologies in order to anonymize queries. Second requirement is to retrieve results that tend to resemble original queries when users execute them. Third requirement is to be easy to reflect edited queries to original queries as much as possible. Our MCHA mechanism would generate mapping data to convert original queries to anonymized query which satisfies these requirements.

In the initial execution, our system converts a query to an anonymized query in Figure 6. The original query could retrieve results from the endpoint. However, in this result, the anonymized query could not retrieve results from it. In this case, the anonymized query could not be useful for help editing and enhancing the query. Therefore, we would prepare an ontology matching mechanism that support the mapping-based conversion of a query to another query.

[5] Wikidata SPARQL Logs: https://iccl.inf.tu-dresden.de/web/Wikidata_SPARQL_Logs/en

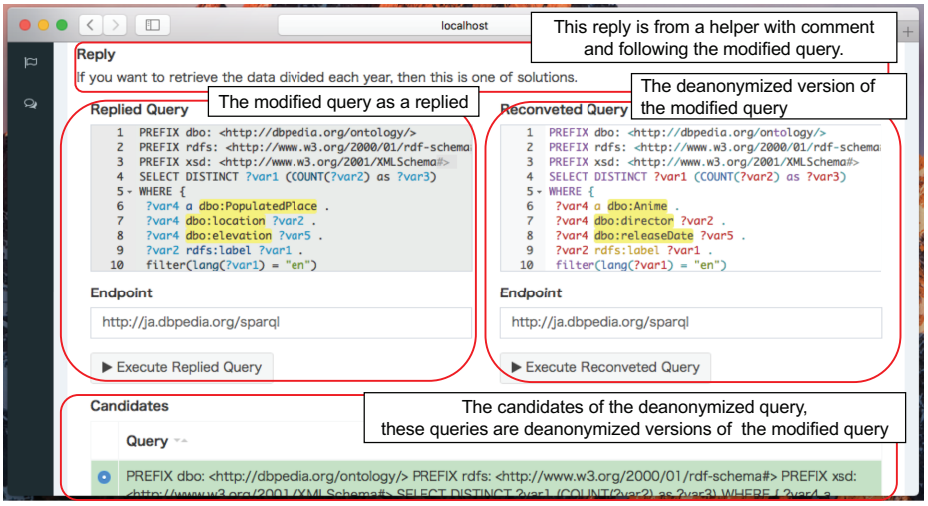

Fig. 3. An overview of MCHA SPAIDA (2)

4 MCHA Mechanism

We focus on ontology mapping and consider a mechanism to generate ontology mappings to convert queries to anonymize queries using terms in ontologies. The problem in generating ontology mappings for the query anonymization is to select mapping targets as domain ontologies. There are many candidate target ontologies (or Linked Open Data), actually there are over 200 available endpoints[6] and over 10,000 ontologies[7]. In SPARQL queries, it is possible to use a combination of multiple ontologies or schemas such as the federated query. Some

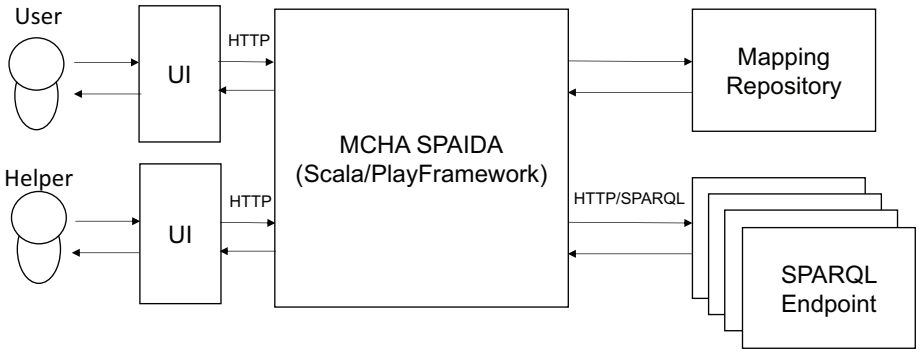

Fig. 4. System structure of MCHA SPAIDA

[6] SPARQL Endpoint Status: http://sparqles.ai.wu.ac.at
[7] Swoogle: http://swoogle.umbc.edu/2006/

```
SELECT DISTINCT ?city ?cityLabel ?mayor ?mayorLabel
WHERE
{
 BIND(wd:Q6581072 AS ?sex)
 BIND(wd:Q515 AS ?c)
 ?city wdt:P31/wdt:P279* ?c .
 ?city p:P6 ?statement .
 ?statement ps:P6 ?mayor .
 ?mayor wdt:P21 ?sex .
 FILTER NOT EXISTS { ?statement pq:P582 ?x }

 ?city wdt:P1082 ?population .
 SERVICE wikibase:label {
   bd:serviceParam wikibase:language "en" .
 }
}
ORDER BY DESC(?population)
LIMIT 10
```

(c) original query in Wikidata SPARQL Logs

```
SELECT DISTINCT ?var1  ?var1Label  ?var2  ?var2Label
WHERE {
  BIND (  <http://www.wikidata.org/entity/Q6581072>  AS  ?var3 ).
  BIND (  <http://www.wikidata.org/entity/Q515>  AS  ?var4 ).
  ?var1 ( <http://www.wikidata.org/prop/direct/P31> /
<http://www.wikidata.org/prop/direct/P279> *) ?var4 .
  ?var1  <http://www.wikidata.org/prop/P6>  ?var5 .
  ?var5  <http://www.wikidata.org/prop/statement/P6>  ?var2 .
  ?var2  <http://www.wikidata.org/prop/direct/P21>  ?var3 .
 FILTER (  (  NOT EXISTS  {
   ?var5  <http://www.wikidata.org/prop/qualifier/P582>  ?var6 .
 }
 )
) .
  ?var1  <http://www.wikidata.org/prop/direct/P1082>  ?var7 .
 SERVICE  <http://wikiba.se/ontology#label>  {
    <http://www.bigdata.com/rdf#serviceParam>
<http://wikiba.se/ontology#language>  "en".
  }
}
ORDER BY  DESC( ?var7 )
LIMIT 10
```

(d) anonymised query in Wikidata SPARQL Logs

Fig. 5. Anonymised query example in Wikidata SPARQL Logs [5]

```
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?genreLabel  (COUNT(?genre) as ?count )
WHERE {
        ?s a dbpedia-owl:Anime .
        ?s dbpedia-owl:genre ?genre .
        ?genre rdfs:label ?genreLabel .
}GROUP BY ?genre ?genreLabel
ORDER BY DESC(?count) LIMIT 10
```

(a) original query

```
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?genreLabel  (COUNT(?genre) as ?count )
WHERE {
        ?s a <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#Agent> .
        ?s <http://xmlns.com/foaf/0.1/primaryTopic> ?genre .
        ?genre rdfs:label ?genreLabel .
}GROUP BY ?genre ?genreLabel
ORDER BY DESC(?count) LIMIT 10
```

(b) anonymized query

Fig. 6. An example mapping-based conversion of a query to an anonymized query

ontologies also have complicated class and property structures. Since ontologies are used only a few in queries, considering how to retrieve subgraphs according to them, it might increase the number of combinations. By increasing the number of combinations, it is necessary to solve implementation problems in calculation and speed. There might be a trade-off between quality of anonymization and the implementation problems. Furthermore, the preference of anonymized queries and strength are different for each user, then it is necessary for users to show multiple anonymized queries.

4.1 Domain anonymization measures

We consider the measures of domain anonymization. We propose two domain anonymized measures, graph distance and semantic similarity.

The graph similarity has been studied with graph similarity search, whose goal is to retrieve relevant graphs given a user-specified, graph-structured query[11]. There have been some approaches for modeling and computation of similarity between graphs, such as graph edit distances[18], maximum common subgraphs[19], edge/feature misses[22], graph alignment[21]. MCHA mechanism would use Multi-Layer Index (ML-Index)[11], which is a multi-layered graph indexing approach to efficiently addressing the similarity search problem in graph databases. The sim-

ilarity search problem defined up the graph edit distance constraint. We apply graph edit distance using ontologies as well as directed labeled graphs.

Here, we briefly summarize the basic definitions of ML-Index[11] as follows. In [11], a graph g is defined as a 4-tuple $(V_g, E_g, l_g, \sum)$, where V_g is a vertex set; $E_g \subseteq V_g \times V_g$ is an edge set; $l_g : V_g \cup E_g \to \sum$ is a labeling function, where $\sum$ is the label set of vertices and edges. Here, as the definitions in [11], we would omit the subscript g in the notations when the context is clear. As explained in [11], a graph g can be modified by the following *graph edit operations*, which include (1) inserting a new, isolated vertex u; (2) inserting a new edge $e = (u, v)$ between existing vertices u and v; (3) deleting an isolated vertex u; (4) deleting an edge $e = (u, v)$; (5) changing the label $l(u)$ of the vertex u; (6) changing the label $l(e)$ of the edge e. So, in [11], the distance among two graphs is initially defined by the minimum editing operations from one graph to another graph, i.e., given two graphs g and g', g can be modified step-by-step to g', or vice versa, by a finite sequence of graph edit operations, the *minimum* number of which is referred to as the *graph edit distance* (GED) between g and g', denoted as $\mathrm{GED}(g, g')$. Here, the computation of graph edit distance is not an easy task because of its computational costs[11]. ML-Index is an efficient indexing approach to relax this computational cost issue by utilizing a well-designed hashing of the associated inverted index and its graph profile.

A semantic similarity (that is sometimes called as semantic relatedness) is an aggregate of the interconnections between two concepts, that is a slightly different notion form semantic distance, though the two terms are sometimes used interchangeably[6]. Semantic relatedness of entities has been heavily researched over the past couple of decades, that can identify two directions such as *corpus-based* and *graph-based*[10]. The corpus-based semantic relatedness models entities as multi-dimensional vectors that are computed based on distributional semantics techniques and word embeddings[14][17]. The graph-based semantic relatedness relies on a graph structured knowledge base such as WordNet[8] and DBpedia[9]. There also have been proposed the graph-based approaches, one is using object properties[13] and other one is based on instances[16]. MCHA mechanism first would apply the corpus-based approach, that uses Jaccard Index, since it is easier to understand for the users to know whether it can be measured semantic similarity or not.

4.2 Implementation

We are implementing our approach on our MCHA SPAIDA system. Figure 7 shows an overview of how the approach is actually used, with an example request and configuration in MCHA SPAIDA. MCHA SPAIDA allows us to adjust the semantic similarity between an original ontology and target ontologies by using the slider.

[8] Princeton University "About WordNet." https://wordnet.princeton.edu/ Princeton University. 2010.

[9] DBpedia: https://wiki.dbpedia.org/

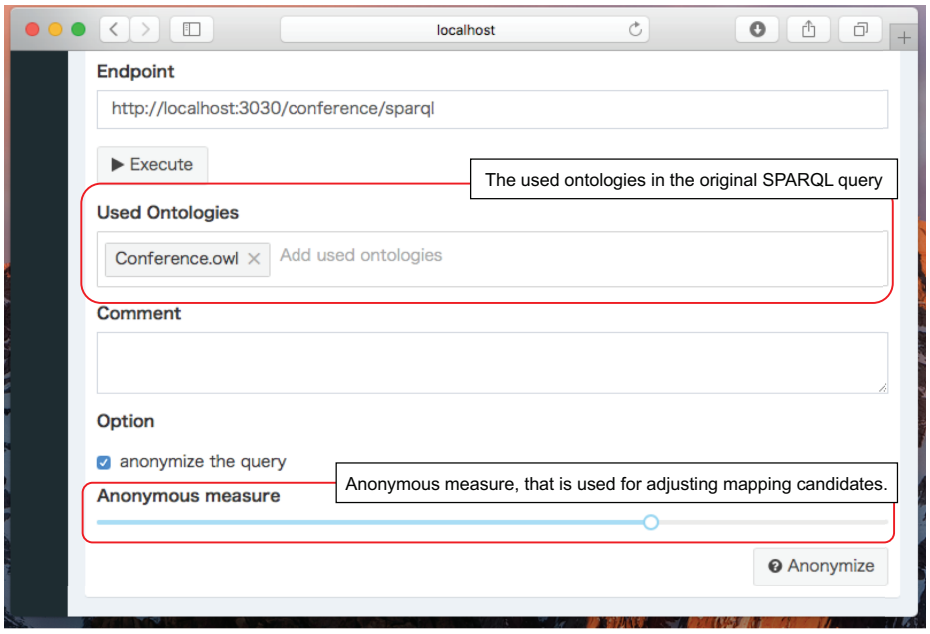

Fig. 7. An overview of the request and configuration in MCHA SPAIDA

Figure 8 shows a workflow of query anonymization in MCHA SPAIDA. When the user input a SPARQL query, ontology information, anonymous measure and others, then the mapping generator generates mapping candidates between used ontologies and stored ontologies in our system. After that, the query anonymizer converts the query to anonymized queries by using mapping candidates. Our system allows us to interactively check anonymized queries with mapping candidates, the user could choose one of the anonymized queries based on the user's preference.

5 Conclusion

In this paper, we proposed an ontology matching mechanism for SPARQL query anonymization in a cooperative SPARQL query editing system. We aim to a privacy-protection of SPARQL queries on the system, we proposed a mapping-based conversion of a query to a "semantically equivalent or very similar" query by using another ontologies. The mapping-based conversion uses ontology mappings to anonymize what the user is investigating in the query. Our mechanism uses two measures such as graph distance and semantic similarity, and also allows us to check anonymized queries whether it is a query according to user's preference by selecting the mappings from mapping candidates.

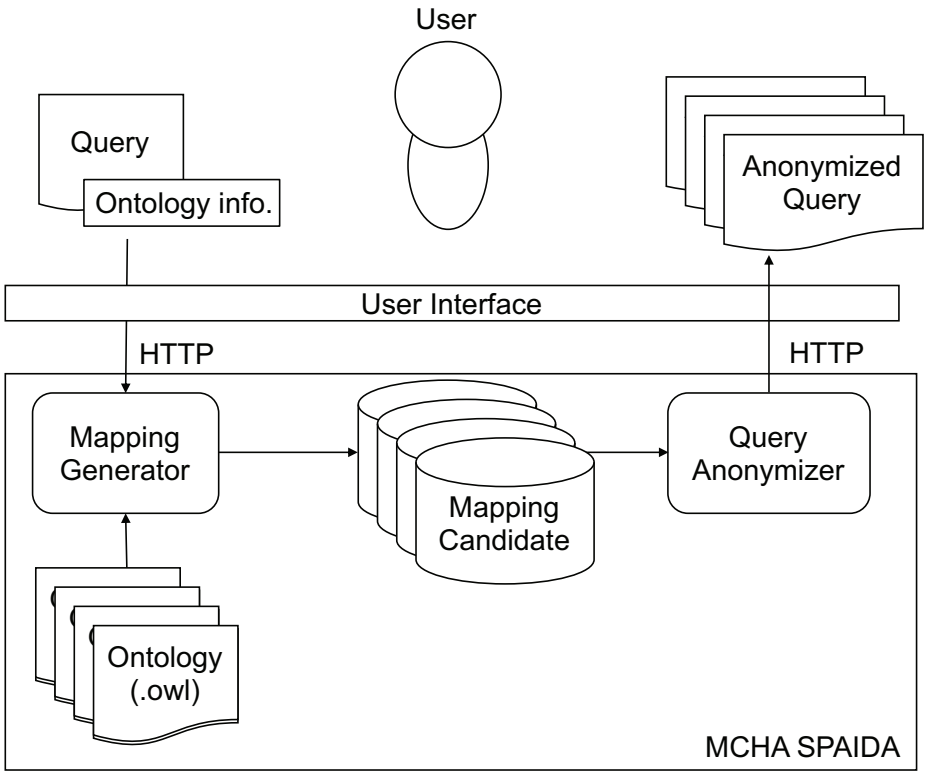

Fig. 8. A workflow of query anonymizaion in MCHA SPAIDA

Acknowledgments

The work was partly supported by the JST CREST JPMJCR15E1.

References

1. Adachi, T., Fukuta, N.: Toward Better Debugging Support on Extended SPARQL queries with On-the-fly Ontology Mapping Generation. In: Proc. of The 11th International Workshop on Ontology Matching (OM2016). pp. 239–240 (2016), (poster)
2. Adachi, T., Fukuta, N.: A Mapping-enhanced Linked Data Inspection and Querying Support System using Dynamic Ontology Matching. In: Proc. of 2nd International Workshop on Platforms and Applications for Social problem Solving and Collective Reasoning (PASSCR2017). pp. 1191–1194 (2017)
3. Adachi, T., Fukuta, N.: MCHA SPAIDA: A Cooperative Query Editor with Anonymous Helpers using Ontology Mappings. In: Proc. of The 13th International Workshop on Ontology Matching (OM2018) (2018), (poster), (to appear)
4. Adachi, T., Yamada, N., Fukuta, N.: Towards Better Query Coding Support Utilizing Ontology Mappings. In: Proc. of 1st International Workshop on Platforms and Applications for Social problem Solving and Collective Reasoning (PASSCR2016). pp. 96–99 (2016)
5. Arenas, M., Grau, B.C., Kharlamov, E., Marciuška, Š., Zheleznyakov, D., Jiménez-Ruiz, E.: SemFacet: Semantic Faceted Search over Yago. In: Proc. of the 23rd International Conference on World Wide Web (WWW2014). pp. 123–126 (2014)
6. Collins, A.M., Loftus, E.F.: A Spreading-Activation Theory of Semantic Processing. In: Psychological Review, vol. 82, pp. 407–428 (1975)
7. Ermilov, T., Moussallem, D., Usbeck, R., Ngomo, A.C.N.: GENESIS A Generic RDF Data Access Interface. In: Proc. of International Conference on Web Intelligence 2017 (WI2017). pp. 125–131 (2017)
8. Fujino, T., Fukuta, N.: Utilizing Weighted Ontology Mappings on Federated SPARQL Querying. In: Proc. of the 3rd Joint International Semantic Technology Conference (JIST2013) (2013)
9. Hamasaki, M., Goto, M.: QueryShare: Working Together to Facilitate Exploratory multimedia Searches without Skill in Creating. In: Proc. of the 13th International Symposium on Open Collaboration (OpenSym2017). pp. 12:1–12:9 (2017). https://doi.org/10.1145/3125433.3125463
10. Hulpuş, I., Prangnawarat, N., Hayes, C.: Path-based Semantic Relatedness on Linked Data and its use to Word and Entity Disambiguation. In: Proc. of the 14th International Semantic Web Conference (ISWC2015). pp. 442–457 (2015)
11. Liang, Y., Zhao, P.: Similarity Search in Graph Databases: A Multi-layered Indexing Approach. In: Proc. of 2017 IEEE 33rd International Conference on Data Engineering (ICDE2017). pp. 783–794 (2017). https://doi.org/10.1109/ICDE.2017.129
12. Makris, K., Bikakis, N., Gioldasis, N., Christodoulakis, S.: SPARQL-RW: Transparent Query Access over Mapped RDF Data Sources. In: Proc. of the 15th International Conference on Extending Database Technology (EDBT2012). pp. 610–613 (2012)
13. Mazuel, L., Sabouret, N.: Semantic Relatedness Measure Using Object Properties in an Ontology. In: Proc. of the 7th International Semantic Web Conference (ISWC2008). pp. 681–694 (2008)

14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In: Proc. of International Conference on Neural Information Processing Systems (NIPS2013). pp. 3111–3119 (2013)
15. Noy, N.F.: Ontology Mapping. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 573–590. Springer-Verlag Berlin Heidelberg (2009). https://doi.org/10.1007/978-3-540-92673-3
16. Passant, A.: dbrec — Music Recommendations Using DBpedia. In: Proc. of the 9th International Semantic Web Conference (ISWC2010). pp. 209–224 (2010)
17. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP2014). pp. 1532–1543 (2014)
18. Riesen, K., Fankhauser, S., Bunke, H.: Speeding Up Graph Edit Distance Computation with a Bipartite Heuristic. In: Proc. of International Workshop on Mining and Learning with Graph (2007)
19. Shang, H., Lin, X., Zhang, Y., Yu, J.X., Wang, W.: Connected Substructure Similarity Search. In: Proc. of the 2010 ACM SIGMOD International Conference on Management of Data. pp. 903–914 (2010). https://doi.org/10.1145/1807167.1807264
20. Soylu, A., Giese, M., Jimenez-Ruiz, E., Vega-Gorgojo, G., Horrocks, I.: Experiencing OptiqueVQS: a multi-paradigm and ontology-based visual query system for end users. Universal Access in the Information Society **15**(1), 129–152 (2016)
21. Tian, Y., McEachin, R.C., Santos, C., States, D.J., Patel, J.M.: SAGA: a subgraph matching tool for biological graphs. Bioinformatics **23**(2), 232–239 (2007). https://doi.org/10.1093/bioinformatics/btl571
22. Yuan, Y., Wang, G., Xu, J.Y., Chen, L.: Efficient distributed subgraph similarity matching. The VLDB Journal **24**(3), 369–394 (2015). https://doi.org/10.1007/s00778-015-0381-6