Knowledge Engineering and Data Science (KEDS)
Vol 3, No 1, July 2020, pp. 11–18

pISSN 2597-4602
eISSN 2597-4637

Query Rewriting with Thesaurus-Based for Handling Semantic Heterogeneity in Database Integration

I Made Riyan Adi Nugroho [a, 1, *], I Wayan Budi Sentana [b, 2]

[a] *Jurusan Teknik Elektro, Politeknik Negeri Bali*
Uluwatu St No.45, Jimbaran, South Kuta, Badung Regency, Bali 80361 Indonesia
[b] *Department of Computing, Macquarie University of Sydney*
Balaclava Rd, Macquarie Park NSW 2109, Australia
[1] *maderiyan@pnb.ac.id *;* [2] *i-wayan-budi.sentana@hdr.mq.edu.au*
** corresponding author*

ARTICLE INFO

Article history:
Received 2 March 2020
Revised 23 April 2020
Accepted 1 July 2020
Published online 17 August 2020

Keywords:
database integration
semantic heterogeneity
query rewriting
thesaurus

ABSTRACT

Nowadays, studies on handling semantic heterogeneity still become a challenge for researcher. Several methods have been used to solve these problems, one of which is query rewriting, implemented by rewriting a query into the latest one by using the selected schema. Semantic query rewriting needs a framework in order to identify the connection through the data schema sources. This line is used as a basis for scheme selection. Also, ontology is a model which often be used in these specific cases. The lack of ontology becomes a significant problem that usually seen. Therefore, this paper will describe an alternative framework in order to identify the link of semantic, which assisted by thesaurus.

I. Introduction

Integration of data sources is a process of combining two or more data resource so that the data which contained can be accessed simultaneously [1]. In the process of integrating data sources, data can be derived from different places or applications. Hence, its heterogeneously potential in format, structure, syntax, and semantic [2]. Heterogeneity can occur at the schema or instance data level [1]. This paper only focuses on semantic heterogeneity in both schema and instance data level. Semantic diversity at the schema level is related to name conflicts caused by synonyms, hyponyms, hypernym, and polysemy. On the other hand, semantic diversity at the data instance level only associated with a name conflict caused by synonyms.

Research on handling the diversity of data sources has long been done. Q*uery rewriting* becomes one of the methods that have been proposed [3]. This method contains a process of rewriting an original query to the new one by adjusting concepts or terminology which used in each data source [3]. There are several approaches in query rewriting, one of them is the ontology-based query rewriting [3][4][5][6][7][8][9]. On this method, ontology is used as a representation of the schema from any data source [3]. Moreover, query rewriting with ontology requires a global ontology as a mediator in identifying the data source schema [3]. In order to make global ontology, ontology reference is needed to identify the connection between existing concept [6]. It usually specific to a particular problem domain [6]. This kind of reference contains both concept and relation which refers to specific standard [6]. The main problem which usually seen is not all problem domains have a reference ontology [6]. In the domain of problem which have no ontology references, global ontologies created based on developer knowledge which potentially produce ambiguity [6].

This paper proposes a query rewriting method using a thesaurus to identify the scheme of a data source. In this step, a global scheme does not necessary. Thus, the identification is processed on *schema matching* by using the thesaurus and n-gram similarity. This process can be seen on Fig. 1.

https://doi.org/10.17977/um018v3i12020p11-18

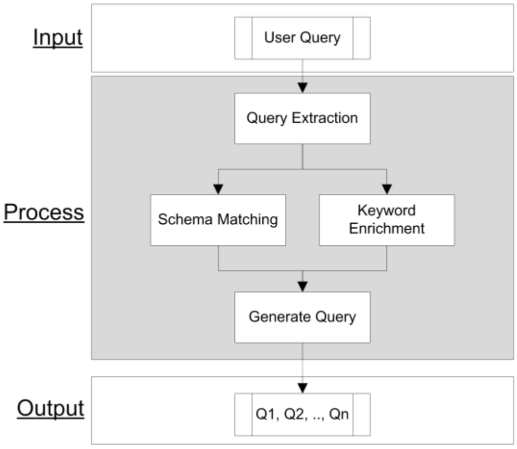


Fig. 2. The Proposed query rewriting process

This paper consists of several sections. The second section, methods describes *query extraction, schema matching*, *keyword enrichment*, and *generate a query*. The result and discussion explains about analysis and test result. The last one is the conclusion

II. Method

The query extraction process is needed to identify the schema of the querying user [10]. This process is made by dividing the querying user into three parts: domain scheme, property scheme, and keyword [3]. In the relational model, domains represent the name of the table, and attribute data is represented by attribute name and keyword, which represent data value. Both domain and property schema are processed at the *schema matching* stage. At the same time, the keyword is processed at the *keyword enrichment* phase. The example of *query extraction* results can be founded in Fig. 2.

Schema matching is needed in order to choose similar data resources with *user query* schema. The selection process is carried out by considering the similarity between semantics and syntax. The process consists of five stages: *schema extraction, get source schema, schema enrichment, string matching,* and *schema selection.* Fig. 3 is the proposed schema matching process.

Schema extraction is the pre-stage of schema matching. The purpose of this phase is to extract the schema from each data source. The schema extracted includes: the name of the data source, the table name, the table relation, attribute names, and attribute data type. The extracted schema is stored in a schema repository. Fig. 4 is an example of the schema extraction results.

Get source schema is a process of getting data source which produced by *extraction schema* stage. The obtained schema will then be calculated in order to find the syntactical similarity values with the schema generated from the enrichment one. The calculation performed on the *string matching* stage.

Enrichment schema is an enrichment process of *user query* outline that will be compared with the data source on the *string matching* process by adding synonyms, hyponyms, and hypernyms. The purpose of these three stages is to identify the data source, which has a semantic correlation. In this paper, the identification of synonyms, hyponyms, and hypernyms are identified by thesaurus. The selected thesaurus is WordNet.

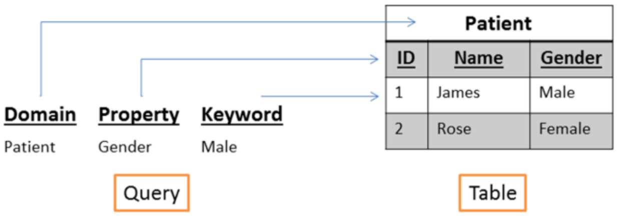


Fig. 2. Sample result of query extraction

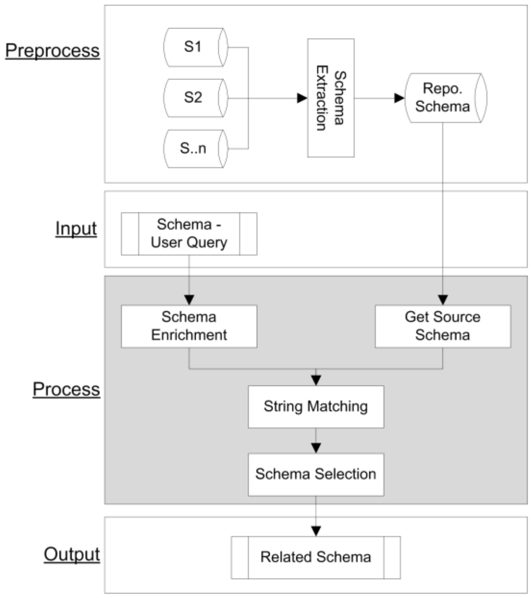


Fig. 3. The proposed schema matching process

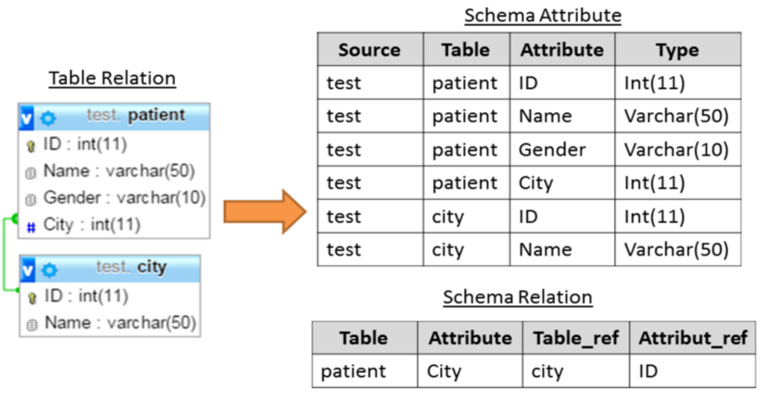


Table Relation

test. patient
ID : int(11)
Name : varchar(50)
Gender : varchar(10)
City : int(11)

test. city
ID : int(11)
Name : varchar(50)

Schema Attribute

Source	Table	Attribute	Type
test	patient	ID	Int(11)
test	patient	Name	Varchar(50)
test	patient	Gender	Varchar(10)
test	patient	City	Int(11)
test	city	ID	Int(11)
test	city	Name	Varchar(50)

Schema Relation

Table	Attribute	Table_ref	Attribut_ref
patient	City	city	ID

Fig. 4. Sample result of the Schema Extraction

The words in Wordnet are organized into a set of a synonym (synset) [11]. Each set closely related to other synset based on semantic relationships such as synonym, hyponym, hypernym, and antonym. A hierarchy tree can be founded from a synset correlation. Fig. 5 describes a synset connection.

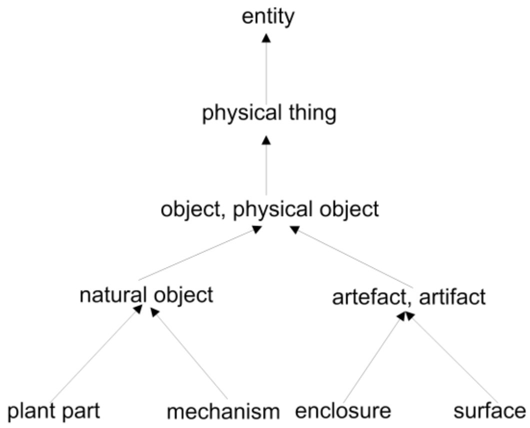


Fig. 5. Sample of the synset relation

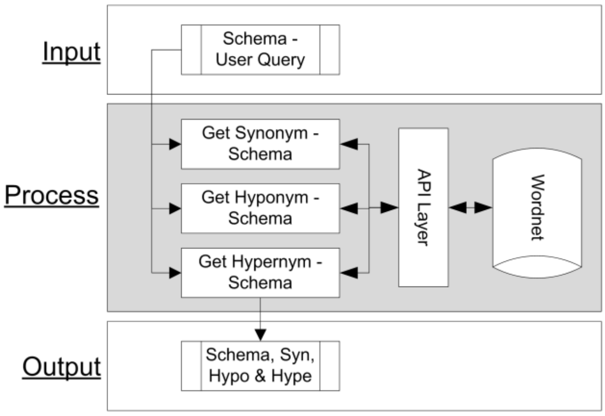


Fig. 6. The proposed schema of enrichment process

Table 1. Example of schema enrichment

Schema	Synonym	Hyponym	Hypernym
Patient	Sufferer	Inpatient, outpatient	Person, individual
Gender	Sex	Feminine, masculine	Category

A synonym can be identified by looking for a similar word located in a common *synset*. In addition, a hyponym can be founded by searching for an identic word that stands below it. Furthermore, hypernym can be seen by searching for words on it [12]. Fig. 6 is the proposed schema enrichment process. The sample results of enrichment schema, as shown in Table I.

After the data source schema has been obtained, and the user query successfully enriched, the following stage is *string matching*. *String matching* is a process to calculate the value of similarity between each scheme represented by a *string* [13]. This value is used as a basis for determining which schema that will be used as the *query*. The calculation is carried out between the domain scheme with table name as well as the property structure with the attribute name.

The string matching technique used in this paper is N-Gram Similarity. This method can be used in multiple string comparisons. By using this procedure, the typical number of n-gram can be counted as n character series between the string. In order to count the similarity of two strings, we can use the *Jaccard Coefficient* equation. Fig. 7 is an example of the n-gram similarity calculation [14].

$$Jaccard\ Coefficient\ (pattern, text) = \frac{|\text{ pattern} \cap \text{text }|}{|\text{ pattern} \cup \text{text }|} \tag{1}$$

Schema selection is a data source selection process in order to find the most appropriate structure with user query schema. This phase is carried out based on the highest similarity. Both *string matching* and *schema selection* are implemented consecutively, where the calculation and selection

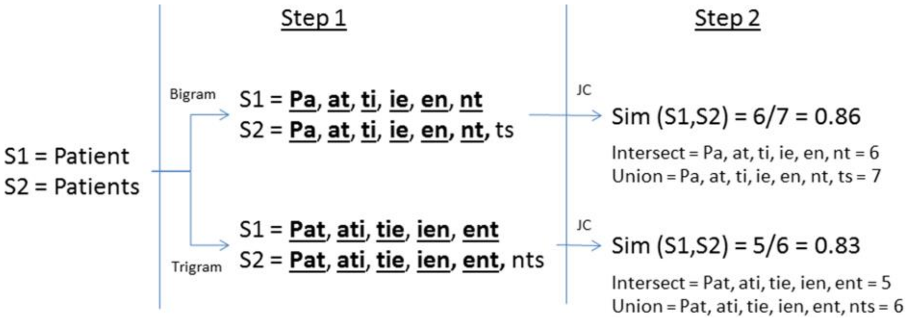
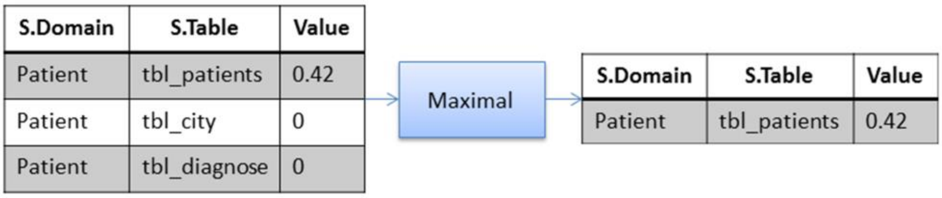

Step 1
Step 2
S1 = Patient
S2 = Patients
Bigram
S1 = Pa, at, ti, ie, en, nt
S2 = Pa, at, ti, ie, en, nt, ts
JC
Sim (S1,S2) = 6/7 = 0.86
Intersect = Pa, at, ti, ie, en, nt = 6
Union = Pa, at, ti, ie, en, nt, ts = 7
Trigram
S1 = Pat, ati, tie, ien, ent
S2 = Pat, ati, tie, ien, ent, nts
JC
Sim (S1,S2) = 5/6 = 0.83
Intersect = Pat, ati, tie, ien, ent = 5
Union = Pat, ati, tie, ien, ent, nts = 6

Fig. 7. Example calculation of the N-Gram Similarity



S.Domain	S.Table	Value
Patient	tbl_patients	0.42
Patient	tbl_city	0
Patient	tbl_diagnose	0

S.Domain	S.Table	Value
Patient	tbl_patients	0.42

Fig. 8. Sample results selection scheme

are made for the table and must first be done. Not only reducing the calculation of string matching, this process also decreasing the selection error made by homonym conditions. The example of *schema selection* is presented in Fig. 8.

The semantic heterogeneity on instance data level occurs due to entities differenciacy while it saved. This diversity contributes an impact on the completeness of data which are integrated. This problem is solved by keyword enrichment. This process is occurred by adding the synonym of keyword. The purpose of this additional is to integrate the information, not only based on the keyword which inputted but also followed by synonym of it.

The synonym identification is performed by thesaurus WordNet. This process followed by words recognition that are located in the corresponding synset as the keyword. Fig. 9 is the proposed keyword enrichment process.

Generate query is a process of query building in accordance with both schema and keyword, which generated in the process of matching schema and keyword enrichment [15]. In this research, the query is built in accordance with the terminology of SQL (Structured Query Language) language SELECT. In order to show the selected data, both SELECT and terminology must have contained in the SELECT order. While SELECT represents the attribute of the table name, in other cases, FORM represents the table name itself. Furthermore, WHERE, ORDER BY, GROUP BY, and HAVING are optional terminology that is representing the condition of data.

The main focus of query development concerns in three parts, such as SELECT, FROM, and WHERE. From *user query* perspective, SELECT represents the attribute schema. FORM represents the domain schema, as well as WHERE represents the keyword. Fig. 10 is an example of *generating query* results.

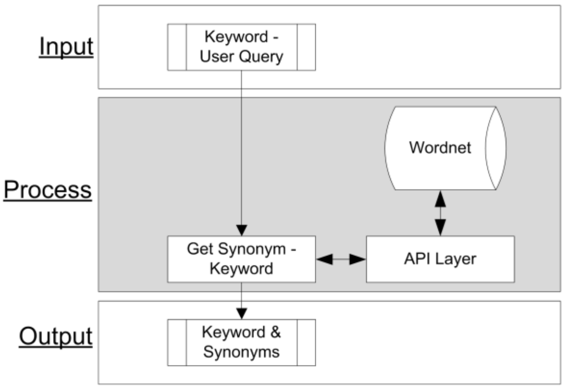
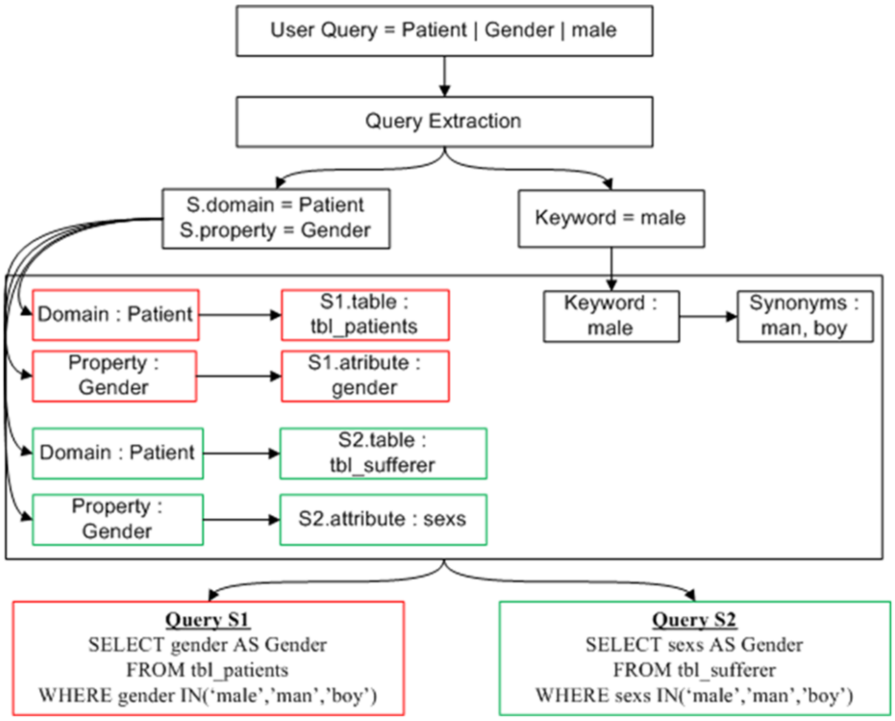


Fig. 9. The proposed of keyword enrichment process

Table 2. Sample result of keyword enrichment

Keyword	Synonym
Male	Man, Boy
Gender	Woman, Girl



```
SELECT gender AS Gender
FROM tbl_patients
WHERE gender IN('male','man','boy')
```

```
SELECT sexs AS Gender
FROM tbl_sufferer
WHERE sexs IN('male','man','boy')
```

Fig. 10. Sample results of the generate query process

III. Results and Discussions

In order to validate an offer, it is needed to build SQRe (*Semantic Query Rewriting*) tool and performed some experiments. SORe developed with CodeIgniter framework (PHP based) and using library NLTK (Python), which can be used to build API wordnet. Experiments were carried out by integrating two databases from different health information systems. Both data sources have semantic diversity at the schema level and instance data level. The first database scheme was shown in Fig. 11, and the second one can be seen in Fig. 12.

The test was finished by determining 5 query user and heterogeneity types of 2 data sources. Table III is showing the result of the test. The table showed that this model could handle the semantic heterogeneity in database integration, such as '*pria-lelaki', 'pasien-penderita', 'pekerjaan-profesi', 'kelamin-gender'*. However, query 3 and query 5 were failed. The failure of query 3 caused by the matching method couldn't handle a scheme which have more than two words, such as 'kode penyakit'. In addition to this, the limitation of data synset (in query 5) such as 'aktivitas-profesi', have made the word connection become unidentified.

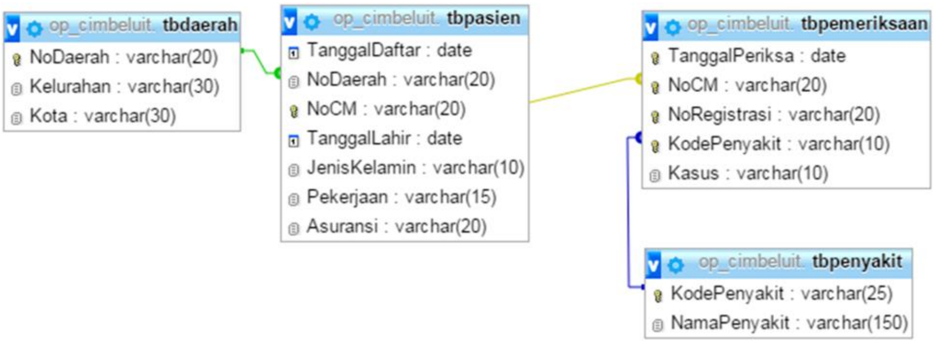


Fig. 11. Database schema 1

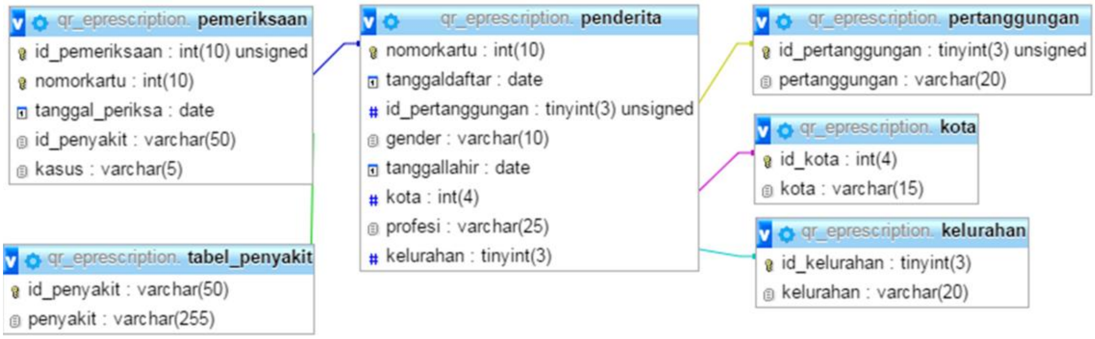

qr_eprescription. pemeriksaan
id_pemeriksaan : int(10) unsigned
nomorkartu : int(10)
tanggal_periksa : date
id_penyakit : varchar(50)
kasus : varchar(5)
qr_eprescription. penderita
nomorkartu : int(10)
tanggaldaftar : date
id_pertanggungan : tinyint(3) unsigned
gender : varchar(10)
tanggallahir : date
kota : int(4)
profesi : varchar(25)
kelurahan : tinyint(3)
qr_eprescription. pertanggungan
id_pertanggungan : tinyint(3) unsigned
pertanggungan : varchar(20)
qr_eprescription. kota
id_kota : int(4)
kota : varchar(15)
qr_eprescription. kelurahan
id_kelurahan : tinyint(3)
kelurahan : varchar(20)
qr_eprescription. tabel_penyakit
id_penyakit : varchar(50)
penyakit : varchar(255)

Fig. 12. Database schema 2

Table 3. Testing result

Type	Domain	Property	Keyword	Status
Query 1	pasien	pekerjaan	wiraswasta	
DB1	tbpasien	Pekerjaan	wiraswasta, wirausaha	success
DB2	penderita	profesi	wiraswasta, wirausaha	success
Query 2	Pasien	kelamin	pria	
DB1	tbpasien	JenisKelamin	pria, lelaki, jantan	success
DB2	penderita	gender	pria, lelaki, jantan	success
Query 3	penyakit	kode penyakit	I10	
DB1	tbpenyakit	KodePenyakit	I10	success
DB2	tabel_penyakit	penyakit	I10	fail
Query 4	pasien	kota	bandung	
DB1	tbpasien	Kota	bandung, pasang	success
DB2	penderita	kota	bandung, pasang	success
Query 5	pasien	aktivitas	buruh	
DB1	tbpasien	Pekerjaan	buruh, karyawan, pegawai, pekerja	success
DB2	penderita	tanggallahir	buruh, karyawan, pegawai, pekerja	fail

IV. Conclusion

This study has introduced an alternative method to handle semantic heterogeneity in the process of database integration with thesaurus-based query rewriting. Semantic heterogeneity at the schema data level is handled by identifying synonyms, hyponymy, and hypernym of each user query. The result of this identification then compared with each data source schema. Semantic heterogeneity at the instance data level handled by identifying synonyms of the keywords, and it will be used in keyword enrichment. Furthermore, the technique used in this schema comparison is n-gram similarity.

The proposed method can be optimized in further research. The reduction of synonym, hyponym, and hypernym can be minimized in order to simplify the calculation. Moreover, the election of schema can be added by metadata analysis and instance data from any data source. The process of schema election can collaborate with both metadata and instance data checking of any source schema. This process is expected can improve the speed as well as the accuracy of the query rewriting process.

Acknowledgement

This research was supported by Politeknik Negeri Bali and Macquarie University of Sydney. We thank our colleagues from both institutions. We thank everyone who contributed to the completion of this paper in one way or another. Hopefully, this research can be useful.

Declarations

Author contribution

All authors contributed equally as the main contributor of this paper. All authors read and approved the final paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] O. M. Tamer and V. Patrick, *Principles of Distributed Database System Third Edition*, vol. 91, no. 5. 2011.

[2] E. Rahm and H. H. Do, "Data Cleaning: Problems and Current Approaches," *Dol. Proc. ACM Int. Work. Data Warehous. Ol.*, pp. 49–56, 2007.

[3] A. Aslam, S. Khan, and K. Latif, "Semantic based query rewriting in heterogeneous sources," *Proc. - 4th IEEE Int. Conf. Emerg. Technol. 2008, ICET 2008*, pp. 292–297, 2008, doi: 10.1109/ICET.2008.4777517.

[4] Handoko and J. R. Getta, "Query decomposition strategy for integration of semistructured data," *ACM Int. Conf. Proceeding Ser.*, vol. 04-06-December-2014, pp. 459–463, 2014, doi: 10.1145/2684200.2684343.

[5] H. Imran and A. Sharan, "Thesaurus and Query Expansion," *Int. J. Comput. Sci. Inf. Technol.*, vol. 1, no. 2, pp. 89–97, 2009.

[6] K. Ramar and T. T. Mirnalinee, "A semantic web for weather forecasting systems," *2014 Int. Conf. Recent Trends Inf. Technol. ICRTIT 2014*, 2014, doi: 10.1109/ICRTIT.2014.6996127.

[7] F. L. R. Lopes, E. R. Sacramento, and B. F. Lóscio, "Using heterogeneous mappings for rewriting SPARQL queries," *Proc. - Int. Work. Database Expert Syst. Appl. DEXA*, no. iii, pp. 267–271, 2012, doi: 10.1109/DEXA.2012.58.

[8] Thantawi, Wicaksana, and W. Lily, "Query Rewriting Berbasis Semantik Menggunakan WordNet dan LCh pada Search Engine Google," in *Konferensi Nasional Sistem Informasi*, no. February, 2013.

[9] A. Shiri and C. Revie, "Query Expansion Behavior Within a Thesaurus-Enhanced Search Environment: A User-Centered Evaluation," *J. Am. Soc. Inf. Sci. Technol.*, vol. 57, no. July, pp. 462–478, 2006, doi: 10.1002/asi.

[10] H. Jayadianti, C. S. Pinto, L. E. Nugroho, and W. Widayat, "Solving Different Languages Problem (Portuguese, English and Bahasa Indonesia) In Digital Library With Ontology," *Proc. 7th ICTS*, vol. 7, pp. 197–202, 2013.

[11] Gunawan and A. Saputra, "Building synsets for Indonesian WordNet with monolingual lexical resources," *Proc. - 2010 Int. Conf. Asian Lang. Process. IALP 2010*, pp. 297–300, 2010, doi: 10.1109/IALP.2010.69.

[12] Gunawan and E. Pranata, "Acquisition of hypernymy-hyponymy relation between nouns for WordNet building," *Proc. - 2010 Int. Conf. Asian Lang. Process. IALP 2010*, pp. 114–117, 2010, doi: 10.1109/IALP.2010.70.

[13] G. Recchia and M. Louwerse, "A comparison of string similarity measures for toponym matching," *COMP 2013 - ACM SIGSPATIAL Int. Work. Comput. Model. Place*, no. July 2018, pp. 54–61, 2013, doi: 10.1145/2534848.2534850.

[14] N. H. Sulaiman and D. Mohamad, "A Jaccard-based Similarity Measure for Soft Sets," *IEEE Symp. Humanit. Sci. Eng. Res.*, pp. 634–651, 2012, doi: 10.4018/978-1-5225-0204-3.ch030.

[15] J. Wang, Y. Zhang, J. Lu, Z. Miao, and B. Zhou, "Query Processing for Heterogeneous Relational Data Integration," *Int. Conf. Intell. Comput. Integr. Syst.*, pp. 777–781, 2010.