

異種オントロジーマッピングでの信頼度を利用可能とした SPARQL クエリシステムの実現

藤野 敬久^{1,a)} 福田 直樹^{2,b)}

概要: セマンティックウェブの検索において、オントロジーの異種性は解決すべき課題である。ユーザが把握するオントロジーと検索対象のオントロジーは、異なるスキーマとして記述されていることがある。本論文で提案するシステムは、SPARQL クエリの中で、ユーザがマッピングの信頼度とそれに伴う結果の表示をコントロールすることを可能にする。提案するシステムでは、異種オントロジーマッピングが存在することを想定し、これらをユーザ自身でコントロールしながら、SPARQL クエリを変換する手法を取る。提案システムの特徴を、本システムと関連ツールとの比較により示す。

キーワード: SPARQL, 異種オントロジー, オントロジーマッピング, クエリ変換

1. はじめに

セマンティックウェブでは、データの持つ意味に基づいて検索を行うことができる。セマンティックウェブにおいて、RDF データを検索するための言語として、SPARQL がある。SPARQL の検索を行うことができるサービスとして、少なくとも 235 の SPARQL エンドポイントが存在する^{*1}。一般に、検索対象で既に用意されたオントロジーに基づいて、アプリケーション開発者は SPARQL クエリを生成する。例えば、Wikipedia の情報を RDF データとして提供する DBpedia[1] と呼ばれる SPARQL エンドポイントがある。このオントロジーは、クラスとして 320、オブジェクトプロパティとして 750、データタイププロパティとして 893、インスタンスとして 1,830,000 のデータが用意されている^{*2}。また、複数の SPARQL エンドポイントにアクセスするための手法も提案されている [2]。

しかしながら、アプリケーション開発者が、検索対象ごとに異なりうるオントロジーを理解し、それぞれの検索対象ごとにクエリを書くことは容易ではない。そのため、オ

ントロジー内の概念検索を行うための手法も提案されている [3]。これにより、アプリケーション開発者はシステムが求めるクエリを生成しやすくなる。私たちはこの問題に対し、アプリケーション開発者自身で検索するためのオントロジーを指定できるようにする手法を検討する。

検索対象で用意されているオントロジーと異なるオントロジーを利用して検索するためには、オントロジーマッピングが必要であるが、オントロジーマッピングを発見するのは非常に難しい課題である [4]。Makris ら [5] は、正確なオントロジーマッピングが与えられていることを前提として、指定したオントロジーを用いたクエリを、検索対象のオントロジーを用いたクエリへと書き換える研究を行なっている。しかし、オントロジーマッピングを発見することの難しさに加え、2つのオントロジー間に意味的な差異が存在する場合には、意味的な違いの欠落を回避して、マッピングを生成することはさらに難しい。そこで、私たちは、マッピングに信頼度が付与されていることとして、それらのマッピングを検索者自身で操作しながら、異なるオントロジーを持つ検索対象でも実行できることを目指す。クエリの中で検索者自身がマッピング操作を記述し、その記述に基づき、私達の提案するシステムにおけるクエリ変換エンジンが、検索対象で実行可能な SPARQL クエリへと変換する。

2. 異種オントロジーマッピングの利用

2.1 異種オントロジーマッピングと信頼度

オントロジーマッピングは、オントロジー研究の中で最

¹ 静岡大学大学院情報学研究科
Graduate School of Informatics, Shizuoka University,
Johoku, Hamamatsu, Shizuoka 432-8011, Japan

² 静岡大学情報学部
Faculty of Informatics, Shizuoka University,
Johoku, Hamamatsu, Shizuoka 432-8011, Japan

a) gs12033@s.inf.shizuoka.ac.jp

b) fukuta@cs.inf.shizuoka.ac.jp

*1 SPARQL エンドポイントの状況 (2012 年 9 月 23 日確認) -
<http://labs.mondeca.com/sparqlEndpointsStatus/>

*2 DBpedia 3.7 - <http://wiki.dbpedia.org/Ontology?v=181z>

も議論されている分野の一つであり、多くの研究がなされている [6]. OAEI(Ontology Alignment Evaluation Initiative)*³では、多くのオントロジーマッピングを導出するためのツールが提案されており [6], Alignment API[7]*⁴を利用することを勧めている. オントロジーマッピングを効率的に発見する手法も検討されている [8]. また, Alignment API とともに提案されている Alignment Format*⁵では、類似度などの指標を含められるように設計されている. 本研究では、このフォーマットを用いて、信頼度を含めることとした.

オントロジーマッピングを発見するための方法は多く考えられているが、状況に応じて効果的なものや適用可能なものは異なる [9]. 例えば、文字列ベースに基づく手法は、同じ言語で書かれたオントロジー間には、マッピングを発見しやすいと考えられる. また、インスタンスベースに基づく手法は、同一のインスタンスが判別できない状況では、発見が難しいと思われる. そこで、本研究では図1のように、複数のマッピングのセットが用意されていることを前提とし、ユーザがそれについて把握している場合には、適切に利用するマッピングを切り替えられることを想定している.

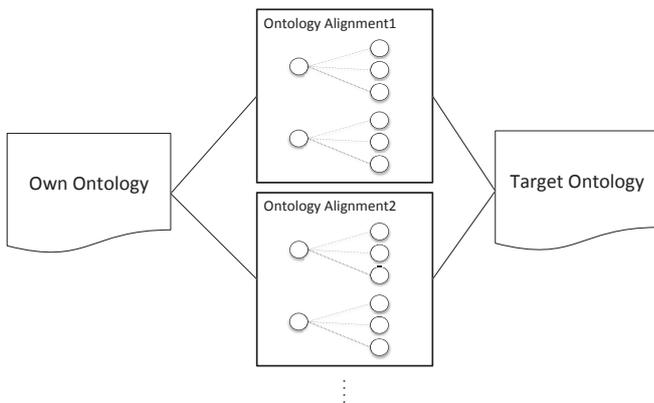


図 1 Heterogeneous Mapping Model in SPARQLoid

2.2 マッピングモデル

オントロジー間のマッピングは、図2のように表されるものとする. これは、図1のように、複数用意されていることを想定する. 図2では、1つのオントロジーにおける1つの概念やプロパティに、もう1つのオントロジーにおける複数の概念やプロパティに対応付けられる. このそれぞれのマッピングには、信頼度を $k[0-1]$ で含めている. この信頼度は、オントロジーマッピングを発見する様々なツールやメソッドを用いて導出されることを想定する.

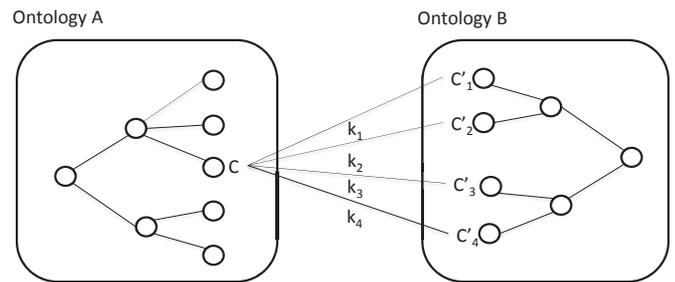


図 2 Basic Mapping Model in SPARQLoid

2.3 マッピング操作

本論文で提案するシステムでは、図1、図2のように定義されたマッピングを、SPARQL クエリを記述する際に、ユーザがランキングの指標や利用するマッピングの指定として利用することができる. ランキングの指標は RANKING 句、利用するマッピングの指定として THRESHOLD 句を新たに加えた. それぞれの操作は、表1、2のように、SPARQL クエリの中で指定できる. この記述の提案は文献 [10] および [11] で行った.

3. SPARQLoid

私たちが提案するシステムを SPARQLoid と呼ぶ. SPARQLoid では、異種オントロジーマッピングが存在する状況を想定し、それらをユーザ自身でコントロールし、利用するマッピングやランキングの指標を指定することが可能である.

3.1 実装モデル

SPARQLoid の実装では、マッピングデータを取得するための2つの実装モデルが考えられる. ここでは、それぞれ Embedding モデル、Referring モデルと呼ぶこととする. 図3に Embedding モデル、図4に Referring モデルとする.

Embedding モデルでは、変換されたクエリの中にマッピングデータを埋め込む. これにより、対象のエンドポイントにアクセス可能とする. 一方で、Referring モデルでは、クエリ変換の中で、対象エンドポイントとマッピングデータ両方にアクセスするようなクエリを生成する. この場合、変換されたクエリの中にマッピングデータは含まれない.

これらの実装モデルは、用途によって場合分けされるものであると考えられる. Embedding モデルでは、概念や信頼度といったマッピングデータを複数、直接埋め込むため、それらを導出するためのオーバーヘッドが必要である.

*³ <http://oaei.ontologymatching.org/>
 *⁴ <http://alignapi.gforge.inria.fr/>
 *⁵ <http://alignapi.gforge.inria.fr/format.html>

Expressions in RANKING	Ranking Criterion
A+B+C	sim=a+b+c
max(A)+max(B)+max(C)	sim=max(a)+max(b)+max(c), max(a) means maximum reliability degree when multiple mappings are available for a.
A=0.6,B=0.4,C=0.3	sim = a*0.6+b*0.4+c*0.3
A>B>C	a is used as the primary key for sorting, b is used as sub-key, and c is used as sub-subkey

表 1 Example Syntax of RANKING clause

Expressions in THRESHOLD	Filtering Criterion
A=0.3,B=0.2,C=0.4	Solutions using a > 0.3, b > 0.2, c > 0.4
max(A)=0.5, max(B)=0.4,max(C)=0.6	Solutions with max(a) > 0.5, max(b) > 0.4, max(c) > 0.6
A:3,B:3,C:3	At most top 3 relevant mappings will be used for A, B, and C

表 2 Example Syntax of THRESHOLD clause

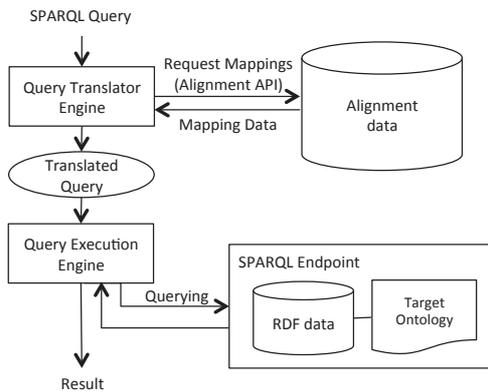


図 3 Embedding モデル

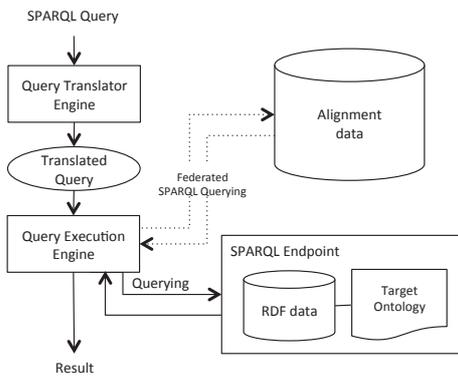


図 4 Referring モデル

また、変換されたクエリの中に、複数の概念や信頼度を含むため、変換されたクエリは冗長になり、可読性が落ちる可能性がある。一方で、Referring モデルでは、計算のオーバーヘッドは少なくなるが、検索対象のエンドポイントでクエリを実行したい場合などに、そのエンドポイントで外部へのアクセスを許していないようなときには、実行できないという点がある。

3.2 クエリ変換例

ここでは、クエリ変換の具体例を示す。以下に示される変換前のクエリには、RANKING、THRESHOLD、そして、検索するための SPARQL エンドポイントを記述している。ここでは、RANKING の指標として、利用されるクラスやプロパティに、重みを付けている。そして、利用するマッピングでは、クラスやプロパティに、閾値を設け、指定された値以上の信頼度を持つマッピングを利用して、クエリを変換する。

```
select ?song ?singer where {
  ?song rdf:type my:originalSong.
  ?song my:hassinger ?singer.
  FILTER isIRI(?singer)

  RANKING{my:hassinger=0.4,my:originalSong=0.6}
  THRESHOLD{my:hassinger=0.5,my:originalSong=0.4}
  ENDPOINT(http://localhost:2020/sparql)
}
LIMIT 100
```

ここで、Embedding モデルで変換されたクエリを示す。

```
SELECT ?song ?singer ?Similarity
WHERE
{
  {
    { ?song rdf:type tar:CoverOrOriginal.
      ?song tar:singer ?singer.
      BIND(0.4444444444444444e0 AS ?sim0)
      BIND(0.666666666666667e0 AS ?sim1)
    }
    UNION
    { ?song rdf:type tar:VocaloidOriginalSong.
      ?song tar:singer ?singer.
      BIND(0.4285714285714286e0 AS ?sim0)
      BIND(0.666666666666667e0 AS ?sim1)
    }
    BIND(( ( ?sim0 * 0.4e0 ) + ( ?sim1 * 0.6e0 ) )
    AS ?Similarity)
  }
  FILTER isIRI(?singer)
}
ORDER BY DESC(?Similarity)
LIMIT 100
```

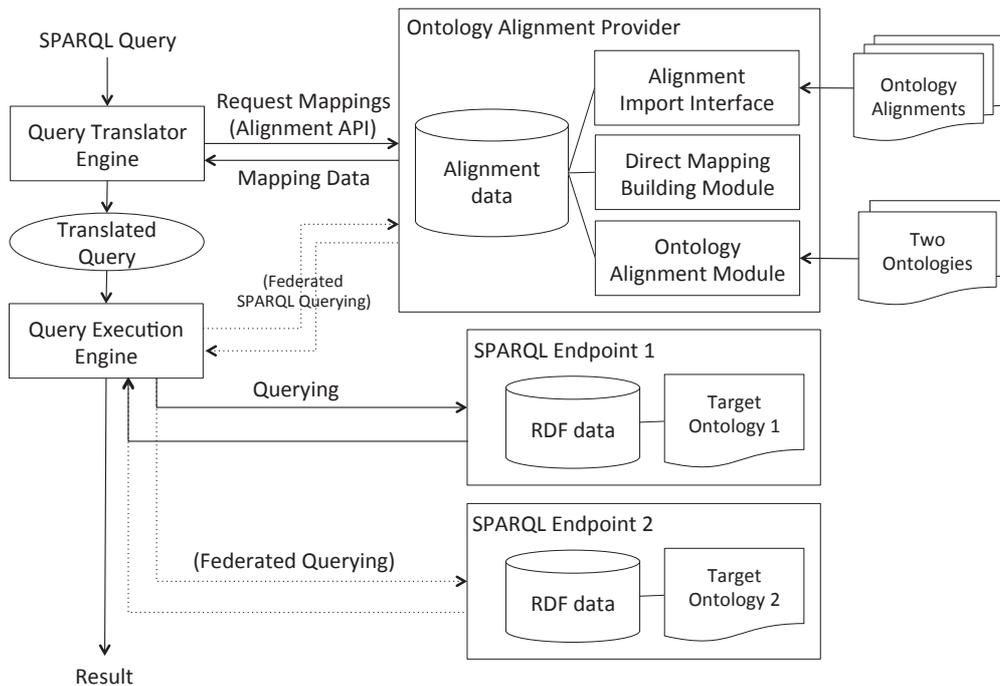


図 5 SPARQLoid Architecture

3.3 アーキテクチャ

図 5 に、SPARQLoid のアーキテクチャを示す。ユーザは、特別な構文を持った SPARQL クエリを入力として与える。このクエリの中には、RANKING 句や THRESHOLD 句などの、マッピングを操作するための記述が含まれている。クエリ変換は、Query Translator Engine で行われる。Query Translator Engine の中で、特別な構文を持った SPARQL クエリは、スタンダードな SPARQL クエリへと変換される。

Query Translator Engine の中で行われるクエリ変換は主に 2 つの実装モデルが考えられる。Embedding モデルでは、外部のアラインメントプロバイダから、マッピングデータを取得する。一方で、Referring モデルでは、それを行わず、実行エンジンの中で、マッピングデータを参照する。

Query Execution Engine は、Jena API を用いて SPARQLoid 内で用意したが、実際にはユーザが用いたいアプリケーションの中で利用したり、対象の SPARQL エンドポイントの中で行われる。SPARQLoid の中では、変換されたクエリの結果をユーザが確認できるようにするために、API を用いて用意した。

SPARQLoid の実装では、Jena API 2.6.4^{*6}、Alignment API 4.4 を利用した。

4. システムの実行情例

図 6 に、システムの実行情例を示す。SPARQLoid では、

^{*6} <http://jena.apache.org/>

ユーザは次のような操作を行う。まずはじめに、Setting タブにおいて、ユーザはクエリを記述するためのオントロジーを選択する (図 6(a))。次に、ユーザは、検索対象の SPARQL エンドポイントを選択する (図 6(b))。このエンドポイントごとに設定されているオントロジーをユーザは選択する (図 6(c))。そして、SPARQLoid は、(図 6(a)) と (図 6(c)) で設定されたオントロジー間のマッピングを検索する。オントロジーマッピングが発見されれば、ユーザはそれを選択でき、もし見つからない場合は、SPARQLoid に内蔵されたメソッドを用いて生成したものを利用できる (図 6(d))。Query タブにおいて、RANKING 句や THRESHOLD 句を用いて記述された SPARQL クエリを、ユーザは記述することができる (図 6(e))。Result タブでは、変換されたスタンダードな SPARQL クエリを確認することができる (図 6(f))。このクエリは、SPARQLoid に内蔵された Jena API の実行エンジンを用いて、検索対象に問合せが可能である (図 6(g))。また、他の SPARQL エンドポイントで直接利用することも考えられる。最後に、変換されたクエリを、指定されたプログラミング言語で利用しやすいように、ソースコードが表示される (図 6(h))。このソースコードは、ユーザのアプリケーションの中で、そのままコピーして利用することを想定している。

5. 関連研究

Makris らは、SPARQL-RW[5] を提案した。SPARQL-RW は、異なるデータセットがあったとしても、オントロジーマッピングに基づいてクエリを書き換えることにより、

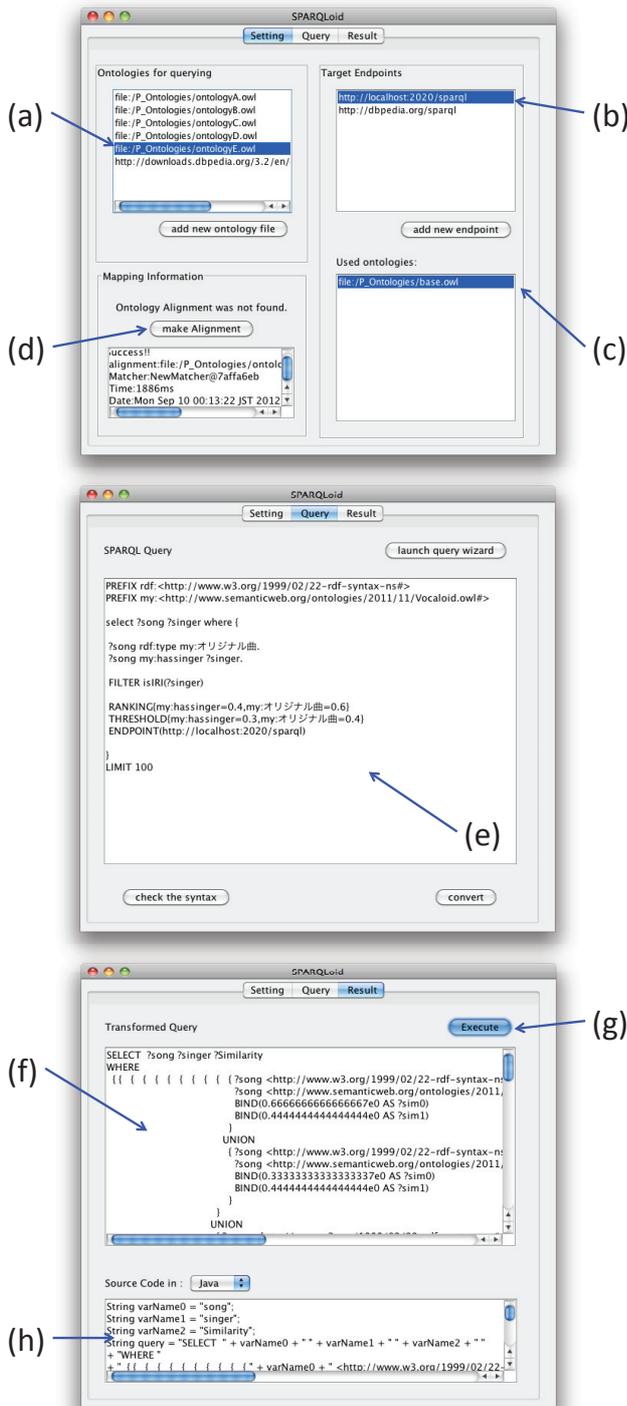


図 6 SPARQLoid screenshot

データアクセスを可能にするフレームワークである。これにより、ユーザはアプリケーション内で、異なるオントロジーで記述された異なるデータセットであったとしてもアクセスを行うことができる。SPARQL-RWでは、事前に正確なマッピングが定義されていることを前提としている。この点において、本論文の手法とは異なっている。また、SPARQL-RWにおけるマッピングのモデルでは、複雑な論理的マッピングが記述できるようになっている。複雑な

論理的マッピングとは、たとえば、1つのクラスが、対応するオントロジーの“1つのクラスと特定のプロパティを持つという条件をみたしたもの”に対応づけられるといった複雑な関係の記述が可能である。現在の SPARQLoid では、こういった複雑な論理的マッピングはサポートしていない。

Rivero らは、Mosto[12], [13] を提案した。Mosto では、ユーザが記述した SPARQL クエリでも実行可能なマッピングデータを生成可能である。これは、私たちのクエリ変換というアプローチとは異なっている。Mosto は、検索対象のデータを自身のクエリのオントロジーに合ったものに変換する。SPARQLoid では、検索対象のデータを変換するのではなく、クエリ自体を変換させることによって、アクセスを可能にした。SPARQLoid では、SPARQL の文法に新たな構文を与え、その構文をもとにして、スタンダードな SPARQL クエリを生成する。これは、ユーザが信頼度付きのマッピングを操作する必要があるためである。SPARQLoid では、ユーザによるマッピング操作のための記述のために、クエリ変換という手法を取っている。

Bizer らは、R2R[14] を提案した。R2R は、データ変換のために必要となるマッピングデータをウェブ上で提供したり、発見したりできるフレームワークである。R2R では、アプリケーションが必要とするオントロジー内の概念が既知でなかった場合、ウェブ上でマッピングデータを探し、自身のアプリケーションに合うようにデータ変換を行えることを想定している。Bizer らは、マッピングの質が多様であることを指摘し、マッピングの質にもアクセスできるようになっている。それにより、よりよい結果を導くマッピングを発見できる。これらは、ヒューリスティックな手法に基いて行われる。R2R で言及するマッピングの質は、SPARQLoid における信頼度に近いと考えられるが、それらをユーザ自身で能動的にランキングの指標にも用いるという点で異なっている。

6. 関連ツールとの比較

ここでは、SPARQLoid と既存の関連ツールとの比較を行う。本論文で比較とする対象は、SPARQLoid (本システム)、Mosto、SPARQL-RW、R2R である。表 3 に、その比較を示す。これらのツールは、それぞれがオントロジーの多様性についての問題を解決することを目的としており、解決するためにオントロジーマッピングを用いている。

SPARQLoid と SPARQL-RW は、クエリ変換という手段を取っており、Mosto と R2R は、データ変換という手段を取っている。他ツールとの比較における SPARQLoid の特長の 1 つは、マッピングにおける信頼度の操作をユーザ自身で記述することである。これによって、データ抽出における結果が異なってくる。R2R においても、マッピングの質について言及しているが、それをユーザ自身のクエ

	SPARQLoid	Mosto[12], [13]	SPARQL-RW[5]	R2R[14]
SPARQL クエリの生成	✓		✓	
マッピングにおける信頼度の利用	✓			✓
マッピングにおける信頼度の操作記述	✓			
データ変換		✓		✓
データ統合		✓		✓
マッピングの生成	*	✓		
複雑な論理的マッピングの利用		✓	✓	✓
マッピングの配布	**			✓
マッピングから他のデータへのリンク				✓
アプリケーション埋め込み性	✓		✓	✓
異種マッピングの利用	*			*

✓implemented, * partially implemented, ** under development

表 3 関連研究との比較

りで操作し、ランキングと利用マッピングを定義することでデータ抽出の結果を変化させることは想定していない。

一方で、SPARQLoid では、複雑な論理的なマッピングは利用できない。Mosto, SPARQL-RW, R2R では、それぞれ適用可能な範囲は異なるが、複雑な論理的なマッピングを利用可能としている。SPARQLoid では、クエリ変換の際にマッピングの信頼度を用いているため、複雑な論理的マッピングを実現することは容易ではない。SPARQLoid での、複雑な論理的マッピングの利用は今後の課題である。

7. 結論

本論文では、SPARQL クエリを用いてデータにアクセスする際に生じるオントロジーの異種性の問題に対し、オントロジー間のマッピングに信頼度を与えることにより、それらをユーザが操作できるようにしながら検索を行うことができるシステムを開発し、関連ツールとの定性的な比較を行った。一般に、クエリが用いているオントロジーと検索対象のオントロジーが異なる場合、ユーザは検索結果を得ることができない。この課題に対し、関連ツールでは、マッピングを用いてこのクエリを可能にする研究も行われているが、本論文ではマッピングを導出する難しさを考慮し、信頼度を用いて検索を可能にすることとした。本システムは、ユーザは自身が指定したオントロジーを用いて、マッピングの信頼度を考慮しながら、対象エンドポイントへ検索を行うことを可能とした。

参考文献

[1] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z.: DBpedia: A Nucleus for a Web of Open Data, *Proc. of the 6th International Semantic Web Conference (ISWC2007)*, Vol. 1, pp. 722–735, (2007).

[2] Quilitz, B., and Leser, U.: Querying Distributed RDF Data Sources with SPARQL, *Proc. of the 5th European semantic web conference on The semantic web (ESWC2008)*, pp. 524–538, (2008).

[3] 北河 祐作, 古崎 晃司, 溝口 理一郎: 多段階展開型オント

ロジー内概念検索システムの試作, 第 26 回人工知能学会全国大会, 3C1-OS-13a-2, (2012).

[4] Noy, N.: Ontology Mapping, Staab, S. and Studer, R.(Eds.), *Handbook on Ontologies*, pp. 573–590, (2009).

[5] Makris, K., Bikakis, N., Gioldasis, N., and Christodoulakis, S.: SPARQL-RW: Transparent Query Access over Mapped RDF Data Sources, *Proc. of the 15th International Conference on Extending Database Technology (EDBT2012)*, (2012).

[6] Euzenat, J., Ferrara, A., Hage, W., Hollink, L., Meilicke, C., Nikolov, A., Ritzke, D., Scharffe, F., Shvaiko, P., Stuckenschmidt, H., Šváb-Zamazal, O., Trojahn, C.: Final results of the Ontology Alignment Evaluation Initiative 2011*, *Proc. of the 6th International Workshop on Ontology Matching (OM 2011)*, (2011).

[7] Euzenat, J.: An API for ontology alignment, *Proc. of the 3rd International Semantic Web Conference (ISWC2004)*, pages 698–712, (2004).

[8] Seddiqui, M., Aono, M.: An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size, *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, Elsevier, Vol. 7, No.4, pp.344–356, (2009).

[9] Tordai, A., Ghazvinian, A., Ossenbruggen, J., Musen, M., and Noy, N.: Lost in Translation? Empirical Analysis of Mapping Compositions for Large Ontologies, *Proc. of the fifth International Workshop on Ontology Matching (OM 2010)*, at *ISWC 2010*, (2010).

[10] Fujino, T. and Fukuta, N.: A SPARQL Query Rewriting Approach on Heterogeneous Ontologies with Mapping Reliability, *Proc. of the IIAI International Conference on Advanced Applied Informatics (IIAI-AAI 2012)*, pp.230–235, (2012).

[11] 藤野 敬久, 福田 直樹: オントロジー変換を伴う SPARQL 検索のためのクエリ変換手法の検討, 第 26 回人工知能学会全国大会, 3C1-OS-13a-1, (2012).

[12] Rivero, C., Hernandez, I., Ruiz, D., and Corchuelo, R.: Mosto: Generating SPARQL Executable Mappings Between Ontologies, *Proc. of International Conference on Conceptual Modeling Demos and Posters*, (2011).

[13] Rivero, C., Hernandez, I., Ruiz, D., and Corchuelo, R.: Generating SPARQL Executable Mappings to Integrate Ontologies, *Proc. of International Conference on Conceptual Modeling*, pp. 118–131, (2011).

[14] Bizer, C. and Schultz, A.: The R2R Framework: Publishing and Discovering Mappings on the Web, *Proc. of the 1st International Workshop on Consuming Linked Data (COLLD 2010)*, (2010).