1N4-OS-10b-2in

マッピングの信頼度を利用した異種LODへの横断的SPARQL検索におけるデバッグ支援手法の検討

Toward a Better Debugging Support on Federated SPARQL Queries using Ontology Mappings with Reliability Degrees

藤野敬久[*1]　福田直樹[*1]
Takahisa Fujino　Naoki Fukuta

[*1]静岡大学大学院情報学研究科
Graduate School of Informatics, Shizuoka University

SPARQL is a standard query language for RDF data that are commonly used to represent and store Semantic Web data. There are a lot of SPARQL endpoints to retrieve and see the stored data by SPARQL queries. Although it greatly helps us query semantic data with ontologies, their diversity of ontologies make it difficult to query the data without understanding of their target ontologies. To solve this problem, several ontology mapping techniques have been investigated, which are trying to find and express a set of correspondences between components(e.g., concepts, etc.) of two ontologies. However, such ontology mappings are very difficult to be extracted. In our work, we have proposed SPARQLoid, which utilizes ontology mappings with reliability degrees that are produced by some ontology matchers. However, it is very difficult for users to find defects in such queries. In this paper, we discuss about possible debugging support approaches for building federated SPARQL queries using ontology mappings with reliability degrees.

1. はじめに

セマンティックウェブの技術のひとつにSPARQL[*1]がある．SPARQLは2008年にW3Cによって勧告されたRDFのクエリ言語である．SPARQLでは，情報の意味に基づいて検索を行うことができる．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266のSPARQLエンドポイントが存在する[*2]．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359，オブジェクトプロパティとして800，データタイププロパティとして975，インスタンスとして2,350,000のデータが用意されている[*3]．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[6]．2013年3月21日には，SPARQL 1.1が正式勧告となり，複数のエンドポイントのデータを組み合わせてクエリを行うことが可能となっている．

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．オントロジー内の概念検索を効果的に行うための手法[7]により，アプリケーション開発者はシステムの実装に必要なクエリの構成と準備が行いやすくなる．一方で，我々はこの課題に対し，アプリケーション開発者自身で検索に使用するオントロジーを自由に選択できるようにする手法を検討している．

検索対象で用意されているオントロジーとは異なるオントロジーを利用して検索処理を実行するためには，オントロジーマッピングが適用可能であるが，質の高いオントロジーマッピングを準備するのは非常に難しい課題である[5]．こうした課題を解決するため，精度，再現率の高いオントロジーマッピングを発見するための手法に関する競技会も開かれている[1]．オントロジーマッピングの手法では，マッピングを発見する際に類似度や距離などが利用される．我々は，そうした値を信頼度と呼ぶこととする．オントロジーマッピングの意味論を定義する研究[4]も行われており，マッピングの信頼度を示す値を利用することにも関心が集まっている．

Makrisら[3]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．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，我々は，マッピングに信頼度が付与されていることを前提として，それらのマッピングの信頼度情報を検索者自身がクエリ内で能動的に利用できるようにすることで，異なるオントロジーを持つ検索対象への効果的な検索クエリ処理を実行できることを目指してきた．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，我々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．そうしたことを実現するシステムとして，我々はSPARQLoid[2]を提案している．

信頼度を用いたオントロジーマッピングにおける検索では，その動作に不具合が見られたときに，例えば原因がクエリの記述そのものにあるのか，あるいは生成されたマッピングやデータそのもの等に起因するのかを把握することは容易ではない．特に，次の4つの点が課題と考えられる．1) バグがどんな条件によって起こるかが明らかでない．2) クエリ・マッピング・オントロジー・データなど，バグの原因となる理由が多岐にわたる．3) ユーザがマッピングのコントロールを行えない可能性がある．4) 適切なマッピングの閾値を設定できるとは限らない．本研究では，これらの課題を解決するための手法を検討する．特に，本論文では，1) および 2) のバグ発生の条件を検

連絡先: 藤野敬久, 静岡大学大学院情報学研究科, 〒 432-8011
静岡県浜松市中区城北 3-5-1, gs12033@s.inf.shizuoka.ac.jp

∗1 http://www.w3.org/TR/2013/REC-sparql11-query-20130321/

∗2 SPARQL エンドポイントの状況（2013 年 4 月 5 日確認） - http://labs.mondeca.com/sparqlEndpointsStatus/

∗3 DBpedia 3.8 - http://wiki.dbpedia.org/Ontology?v=181z

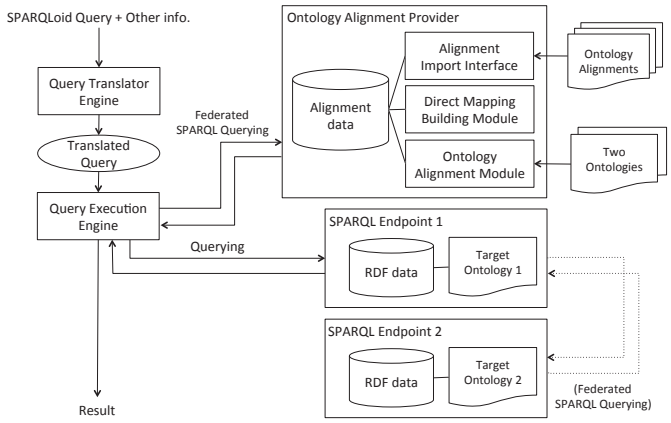

図 1: SPARQLoid Architecture

討するとともに，3) の一部として，4) での適切なマッピングの閾値の設定のための機構とユーザインターフェースを実際に実装することで検討を進める．

2. SPARQLoid

SPARQLoid は，信頼度を用いたオントロジーマッピングを利用する SPARQL クエリを生成するシステムである．そのクエリをユーザのアプリケーションの中で容易に利用できるようにする目的で，Java のソースコードの断片も生成可能としている．また，変換されたクエリを実際に動作させて，その検索結果を確認できる実行・テスト環境も用意している．

2.1 アーキテクチャ

図 1 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，標準的な SPARQL 1.1 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．図 1 では，Referring モデルを示している．

2.2 マッピングのモデル

図 2 に，SPARQLoid で扱うマッピングのモデルを示す．ユーザが把握するオントロジーの概念 1 つに対して，検索対象のオントロジーの概念複数に対応づけられる．また，それぞれに信頼度 k[0-1] が割り当てられる．これらのマッピングは，どのメソッドによって作られたかも明示されているものとする．このモデルは，alignment format[*4]，もしくは，その拡張である EDOAL[*5]によって記述される．これらは，オントロジーマッチングを行う際の出力の表現として用いられる．ここでのマッピングは，文献 [8] の手法に代表される既存のマッチング技術を用いて用意される．

∗4 http://alignapi.gforge.inria.fr/format.html
∗5 http://alignapi.gforge.inria.fr/edoal.html

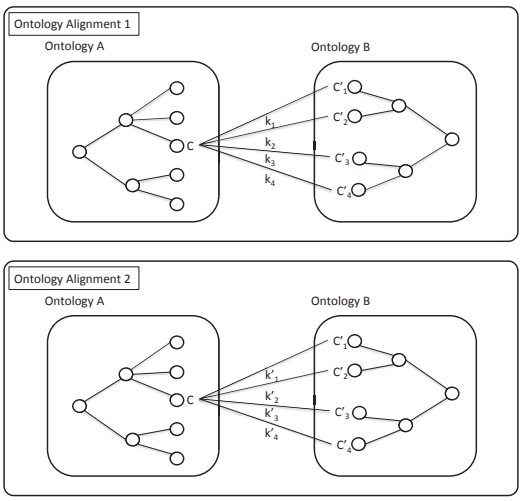

図 2: マッピングのモデル

3. 不具合の例と対策

ここでは，実際に SPARQLoid を用いた問合せの実例を挙げながら，SPARQLoid クエリにおける不具合発生の事例について述べる．ここでは，簡単のために，PLEFIX 等の記述は省いている．

3.1 閾値の設定による不具合

```
select ?x ?singer where {

?x rdf:type my: オリジナル曲.
?x rdf:type my: 初音ミク動画.
?x my:hassinger ?singer

RANKING{my:hassinger*0.4+my: オリジナル曲*0.6+my: 初音ミク動画*0.5}
THRESHOLD{my:hassinger=0.6,my: オリジナル曲=0.3,my: 初音ミク動画=0.1}
}
LIMIT 150
```

Listing 1: SPARQLoid query

```
select ?x ?singer where {

?x rdf:type my: オリジナル曲.
?x rdf:type my: 初音ミク動画.
?x my:hassinger ?singer

RANKING{my:hassinger*0.4+my: オリジナル曲*0.6+my: 初音ミク動画*0.5}
THRESHOLD{my:hassinger=0.7,my: オリジナル曲=0.3,my: 初音ミク動画=0.1}
}
LIMIT 150
```

Listing 2: SPARQLoid query

Listing 1 と Listing 2 に，SPARQLoid のための典型的なクエリを 2 つ示す．図 3，図 4 に，それぞれの実行結果を示す．Listing 1 と Listing 2 の違いは，THRESHOLD の値が異なる点のみである．Listing 1 での検索結果は，適切にランキングされた結果が得られる．しかし，Listing 2 の場合では，*my:hassinger* の閾値が高すぎてしまうため，マッピングが見つからず，適切な検索結果が得られない．

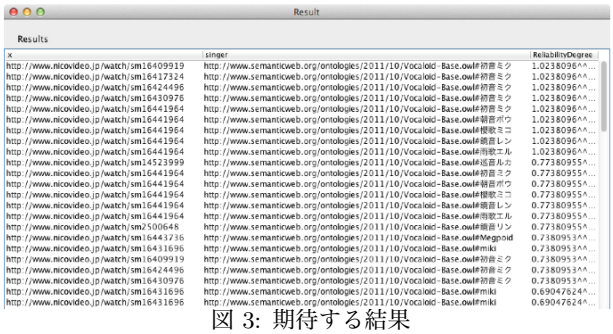

図 3: 期待する結果

図 4: 不具合を伴う結果

クエリ作成者は何が原因でこのような違いが起こるのかを，エンドポイントから返された結果のみから判断することは容易では無いと考えられる．特に，どの概念に対する閾値の設定を変えるべきなのかを判断することは，具体的な値を見てみなければ判断が難しい場合が多い．クエリ作成者がマッピングの情報を確認しながらクエリ作成を行うことができれば，これらの不具合発生やその原因に気づける可能性がある．

3.2 マッピングによる不具合

閾値を適切に指定できたとしても，それがクエリ作成者の期待する検索結果とはならない事例について述べる．クエリ作成者がクエリの中で設定する閾値に問題がなかったとしても，利用しようとするマッピングそのものの信頼性に問題があることがある．Listing 3 に示すクエリでは，*my:*神曲という概念が使われているが，検索対象にそれと高い信頼度で対応する概念へのマッピングは存在しない．図 5 にその実行結果を示す．一見それらしいデータが提示されているが，高い信頼度で対応する概念へのマッピングが存在しないため，検索対象を明確にクエリの中で表現することが困難になり，その検索結果がクエリ作成者にとって期待した結果となる可能性は低くなる．この例で用いたエンドポイントに直接クエリを行い，*my:*神曲のマッピングを抽出した結果を図 6 に示す．

```
select ?x ?singer where {

?x rdf:type my: 神曲.
?x my:hassinger ?singer

RANKING{my: hassinger*0.4+my: 神曲*0.6}
THRESHOLD{my: hassinger=0.2,my: 神曲=0.01}
}
LIMIT 15
```

Listing 3: SPARQLoid query

図 5: 不具合を伴う結果

図 6: *my:*神曲に対応するマッピングと信頼度

ここで用意されていたマッピングは，オントロジーマッチングで低い信頼度として出力されたものであった．この不具合を回避するためには，必要に応じて，マッピング生成手法を変更するなどが考えられる．しかしながら，こうした不具合はマッピングの情報のみに起因するものとは限らない．そうした場合，マッピングの情報を確認するだけでなく，その他の支援手法も必要になると考えられる．

3.3 その他の不具合

マッピングが存在していても，検索対象のデータそのものが存在しない場合がある．検索対象に概念は存在するが，そのインスタンスが存在しないという場合がそれに相当する．そうした場合には，マッピングの情報のみを提示したとしても，不具合は解決しない．検索対象にデータがどの程度存在しているのかをクエリ作成者に効果的に提示できるような機構を準備する必要があると考えられる．

さらに，検索対象のエンドポイントが動作していない場合やマッピングを提供するためのエンドポイントが動作していないなどの事象も，期待した検索結果が得られないことの原因になり得る．たとえば，エンドポイントへのアクセス過多による一時的なアクセス拒否などもこれに相当する．SPARQLoid では，エンドポイントが正常な反応を返さない場合には，クエリ変換の段階で警告を出す．しかしながら，アプリケーション開発者が自身のアプリケーションに変換されたクエリを埋め込んだ際に，そのような不具合が生じていることを検出することは容易ではない．特に，生成したクエリが Federated Query になっており，複数のエンドポイントにまたがるようなデータを取得する場合には，どこに問題が生じているのかを発見することは難しい．

3.4 不具合発見の支援

図 7 に，不具合発見の支援として我々が実装を進めている機構およびそのユーザインタフェースの例を示す．図 7 では，クエリの中に用いられた概念のマッピングをランキング形式で提示している．用意されているマッピングの数・閾値により利用可能なマッピングの数・検索対象に存在する対応する概念のインスタンス数を確認できるようにしている．このように，

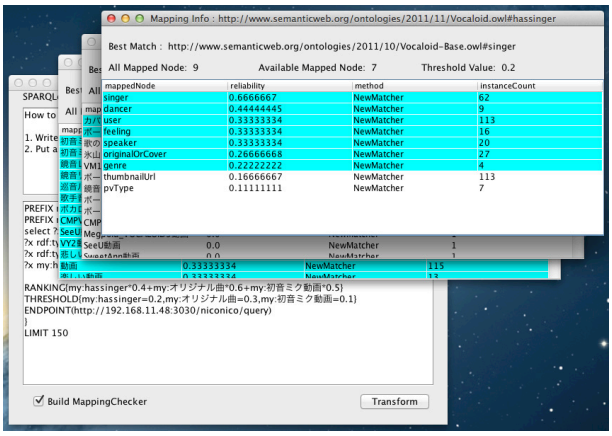

図 7: 対応する概念の提示

クエリ作成者が適切な閾値を指定するためには，マッピングの情報を確認できることが必要であると考える．閾値によって，対応する概念が存在しない場合や，対応する概念数が多すぎてしまう場合には，適切な検索結果が得られないことや，多くの計算時間が必要となるためである．

しかしながら，クエリの中で用いられる概念数が増えるにつれ，その確認や，適切な閾値を与えることが困難となる．デバッグ支援の方法としては，その他に，適切なマッチング手法の推薦，あるいは，利用可能なマッピングを効率的に発見する機構が必要になると考えられる．また，この支援例だけでは，データそのものを確認できないため，そのデータがクエリ作成者にとって抽出対象でない場合には，その不具合を発見することは難しい．

4. デバッグ支援システム構築に向けて

クエリの生成におけるデバッグ支援としては，SPARQLoid 内部での実装を進めている．マッピング情報の提示 (図 7) は，THRESHOLD 句をオブジェクト化したものを利用した．THRESHOLD 句のそれぞれのノードに対して，SPARQL query を生成し，マッピングを格納するエンドポイント・検索対象エンドポイントに対して問い合わせを行わせている．マッピング情報提示の流れについては，図 8 の通りである．

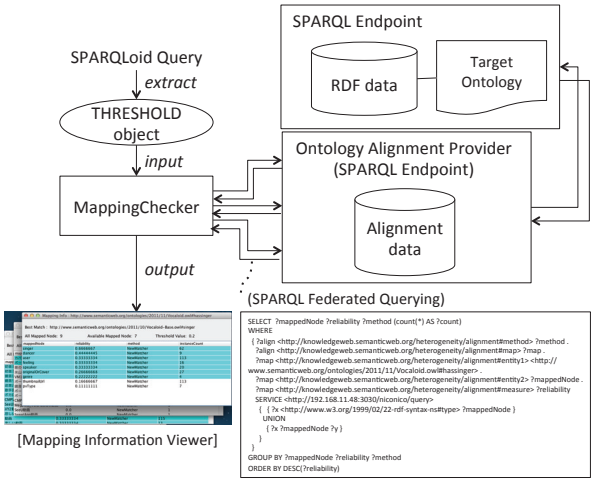

図 8: マッピング情報提示の流れ

しかしながら，3 章で述べたように，多様な不具合に効果的に対応できるようにするには，さらに手法や機構の開発を進める必要がある．クエリ作成時にエンドポイント内のデータ確認を容易にするシステムに加え，アプリケーション側でクエリを動作させた場合のクエリの振る舞いのトレース支援システムなどについてもその必要性を検討したい．

5. まとめ

本論文では，異種オントロジーを持つ LOD に対する，マッピングの信頼度を用いた SPARQL クエリのデバッグ支援の検討を行った．SPARQLoid といわれるシステムにおけるバグのパターンを複数提示し，そのバグを発見するための効果的な手法および可能な支援機構の実装方法について検討した．実際のアプリケーション開発にかかわるクエリ作成時にどんな不具合が起こる可能性が高いのか，どんな支援手法が効果的かなどを検証するための，実験・評価については，今後の課題である．

参考文献

[1] J. L. Aguirre, K. Eckert, J. Euzenat, A. Ferrara, W. R. van Hage, L. Hollink, C. Meilicke, A. Nikolov, D. Ritze, F. Scharffe, P. Shvaiko, O. Šváb Zamazal, C. Trojahn, E. Jiménez-Ruiz, B. C. Grau, and B. Zapilko. Preliminary results of the Ontology Alignment Evaluation Initiative 2012. In *Proc. of 7th Ontology Matching Workshop (OM2012), at International Semantic Web Conference (ISWC2012)*, 2012.

[2] T. Fujino and N. Fukuta. SPARQLoid - a querying system using own ontology and ontology mappings with reliability. In *International Semantic Web Conference (Posters & Demos) (ISWC2012)*, 2012.

[3] K. Makris, N. Bikakis, N. Gioldasis, and S. Christodoulakis. SPARQL-RW: transparent query access over mapped RDF data sources. In *Proc of the 15th International Conference on Extending Database Technology (EDBT2012*, pages 610–613, 2012.

[4] A. B. Manuel Atencia, J. Euzenat, C. Ghidini, and L. Serafini. A formal semantics for weighted ontology mappings. In *Proc. of the 11th International Semantic Web Conference (ISWC2012)*, pages 17–33, 2012.

[5] N. F. Noy. Ontology mapping. In S. Steffen and S. Rudi, editors, *Handbook on Ontologies*, pages 573–590. 2009.

[6] B. Quilitz and U. Leser. Querying distributed RDF data sources with SPARQL. In *Proc. of the 5th European Semantic Web Conference (ESWC2008)*, pages 524–538, 2008.

[7] 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 (JSAI2012), 2012.

[8] 野口 宙毅，藤野 敬久，福田 直樹. マッピング生成・更新機構を持った SPARQL 処理フロントエンドによる異種 LOD の横断的検索システムの試作. 第 27 回人工知能学会全国大会 (JSAI2013), 2013.