3C1-OS-13a-1

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

A Preliminary Proposal of SPARQL Query Rewriting with Ontological Translation

藤野敬久*1　福田直樹*2
Takahisa Fujino　Naoki Fukuta

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

*2静岡大学情報学部
Faculty 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 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 this paper, we present a preliminary idea and discussions about SPARQL query rewriting when ontology mappings are not complete. We show some examples about rewriting a SPARQL query with one ontology to another query with another ontology which can specify ordering by their mapping reliability.

1. はじめに

セマンティックウェブにおける検索のための技術として，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 は検索条件や検索結果に，RDFS や OWL などに基づいたオントロジーの概念を指定することができ，セマンティックウェブ技術の中で注目されている技術の 1 つである．現在でも，DBpedia をはじめ，200 を超えるデータセットがウェブ上で公開されており，SPARQL を実行することができる．こうしたサービスは SPARQL エンドポイントと呼ばれる．そして，複数の SPARQL エンドポイントなどにアクセスするための方法も提案されている [Ladwig 10].

ウェブ上に RDF データを公開しアクセスを可能にする SPARQL エンドポイントには，事前にオントロジーに基づいたデータが用意されている．例えば，DBpedia には 320 を超えるクラスや 1650 の異なるプロパティ*2が用意されている．これらのオントロジーを全て把握し，クエリを実行することは難しい．そのため，ユーザに既知のオントロジーが存在する場合には，そのオントロジーを用いて SPARQL エンドポイントにクエリを実行することが求められる．

一方で，SPARQL は検索条件や検索結果に，オントロジー内の概念を指定することができるが，一般に，クエリ内で指定したオントロジーの概念に基づかないデータは取得されない．そのため，異なるオントロジーが存在する状況では，クエリを実行し適切な結果を得るには，対象となる SPARQL エンドポイントで用いられるオントロジーを適切に参照したクエリを準備する必要がある．これを実現するためには，オントロジー間の対応関係を発見し，その対応関係を適切に表現できるようにする必要があり，オントロジーマッピングと呼ばれる研究領域において議論されている．しかし，自動的に高い精度のオントロジーマッピングを実現することは現時点でもまだ非常に難しい問題とされており，多くの研究が続けられている [Noy 09]. そのため，オントロジーマッピングには，間違いなどが含まれることを前提にして，何らかの工夫で適切な結果を取得できるようにする方法の実現には一定の合理性があると考えられる．

本稿では，これらを踏まえ，クエリを実行するユーザには既知のオントロジーが存在し，そのオントロジーの概念を用いて SPARQL エンドポイントにクエリを実行すると仮定し，既存のオントロジーマッピング技術を利用した際に生じうる不完全性を補完するための SPARQL クエリの変換手法を検討する．

2. 異種オントロジーによる SPARQL 検索

2.1 オントロジーマッピングのアプローチ

オントロジーマッピングとは，2 つのオントロジーにおける，コンポーネント間での一致セットである [Noy 09]. オントロジーマッピングは，2 つのオントロジー内の概念同士の対応関係を記述することによって，異なるオントロジーが存在するような状況においても，データの相互利用が可能になる．ここでは，オントロジーマッピングにおける，マッピングの発見とマッピングの表現について記述する．

オントロジーマッピングのためには，概念間の一致を発見する必要がある．概念間の一致の発見は，手動によるもの，自動によるもの，あるいは，これらを組み合わせることが考えられる．特に，自動によるものについては，高い精度のオントロジーマッピングを実現することは非常に難しい問題とされている [Noy 09].

概念間の一致を発見した後，それらをどう表現するかも議論すべき課題である．オントロジーマッピング技術を利用し，既知のオントロジーとは異なるオントロジーに基づくデータをクエリで取得するためには，オントロジーマッピングが記述されている必要がある．オントロジーマッピングの記述は，OWL や RDFS の語彙で記述する [Noy 09] こともできる．この場合，2 つのオントロジーは統一されている状況となる．また，マッピングを表現するための言語として，EDOAL(Expressive and Declarative Ontology Alignment Language)*3を用いる

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/rdf-sparql-query/

∗2 http://wiki.dbpedia.org/Ontology?v=181z

∗3 http://alignapi.gforge.inria.fr/edoal.html

ことで，概念間の一致の記述を，クラス間の 1:1 の対応関係などに比べて詳細に記述できる．具体的には，EDOAL を用いることで，クラス間のマッピングやプロパティ間のマッピングなどを，制約などを含めて詳細に記述することができる．例えば，図 1 のように，オントロジー A の「初音ミク動画」クラスを示すエンティティと，オントロジー B の「ボーカロイド動画」とそのインスタンスの singer プロパティの値が「初音ミク」であるエンティティ同士が一致であることを示すとすると，オントロジー B にはこの概念を示すクラスが直接は用意されない状況であっても，このような，すでに定義されたクラス間での対応関係を EDOAL により記述することができる．また，EDOAL では，概念の間のマッピングの信頼性 [0,1]（confidence）を加えることが可能になっている．

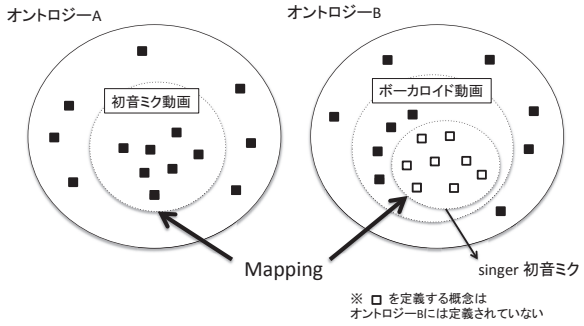

図 1: EDOAL での複雑なマッピングの例

2.2 SPARQL のクエリ変換

SPARQL におけるクエリ変換には，例えば，SPARQL を用いてデータを取得する際に，OWL の意味論に基づいて結果を取得する方法が提案されている [Kollia 11]．これは，一般に，SPARQL のクエリは RDF レベルで一致する結果を取り出すが，この研究では OWL の意味論に基づく検索を行うことについて検討している．そして，そのために SPARQL のクエリを BGP(Basic Graph Pattern) 単位で書き換える手段を取っている．その他にも，オントロジーマッピングが与えられている状況で，それらを利用して SPARQL クエリを書き換える研究も行われている．

[Makris 12] は，異なるオントロジーで書かれたデータセットに対し，クエリを書き換えることでアクセスすることを可能にしている．ここでは，異なるオントロジーは事前にマッピングされていることを前提としており，それらのマッピングが与えられている際に，クエリを書き換える方法を示している．また，オントロジーマッピングにおける表現力も高いものとなっている．例えば，クラスの表現力として，連言や選言を用いて，プロパティの値制約などを加えることが可能となっている．この研究でも同様に，BGP を書き換えることで，クエリの書き換えを可能にしている．

2.3 本論文で扱うマッピング表現

本論文では，オントロジーマッピングとして，概念間の対応を 1:N で表現し，それらの類似度をそれぞれ k[0,1] で表現する方法を用いる [藤野 12]．図 2 のように，検索対象のオントロジー（ベースオントロジー）とクエリが基づくオントロジー（個人オントロジー）があるとき，個人オントロジーのクラス 1 つに対し，ベースオントロジーの類似するクラスをそれぞれ対応付ける．このマッピングはオントロジーにおいて，名前付けされた概念間で行うものとする．なお，機械学習などでオントロジーマッピングを発見する際，概念間がどの程度似ているかという類似度 (Similarity) を扱うことが多い．こうした技術が，概念間の類似度として利用できることが想定される．

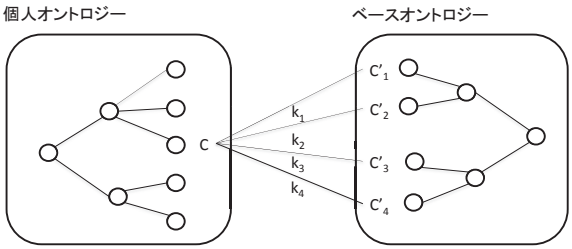

図 2: 本論文でのマッピング表現

オントロジーマッピングの研究では，EDOAL のような複雑な対応付け (complex mappings) を表現する言語や，それらを発見する試み [Ritze 10] はあるが，本論文では概念間の類似度まで含んだ対応付けのみを扱うところを，検討の出発点とする．ただし，概念間の類似度を含んだマッピングを用いる場合には，検索結果の再現率 (探したい対象の取りこぼしの少なさ) を重視するとクエリを変換した後の検索結果が増えすぎることから，それらを適切にランキングしたりフィルタしたりするための仕組みが必要となる．また，1 つのクエリで複数のオントロジーの概念を参照する場合には，クエリにおいてその概念類似度のどの部分を重視するのかといったことをクエリ内に記述できることが重要となる．

2.4 ユーザによるランキング項目の指定

ユーザが特定のノードに対する概念間類似度を重視した検索を行いたい場合には，項目並べ替えに用いる類似度計算の優先度をユーザのクエリの中に含められるようにすることが望ましい．例えば，SPARQL は，1 つのクエリの中で複数の概念を指定することができる．そのため，それらを満たす結果を表示する際に，複数の概念のどれに注目すべきかをユーザが指定できるようにする方法を考える．例えば，次のような SPARQL クエリがあったとする．

```
SELECT ?x
   ?x rdf:type my:ClassA .
   ?x rdf:type my:ClassB .
```

このようなクエリの場合，my:ClassA と my:ClassB のどちらの類似度をもとにランキングあるいはフィルタリングするのかが課題となる．仮に，ユーザが my:ClassA を検索の条件の主として想定している場合には，my:ClassA の類似度と my:ClassB の類似度では，my:ClassA の類似度を優先したい．一方で，それぞれのクラスを同程度の比重としてユーザが想定している場合には，my:ClassA と my:ClassB の類似度の和をもとに，ランキングしたい．

このように，ユーザが SPARQL のクエリを書いたとき，どの概念に対応する項目を優先的に抽出したいかについて，ユーザがクエリ内で指定できる仕組みが必要となる．SPARQL では，このような状況は必ずしも想定されないため，概念間の類似度の考慮およびその優先順位項目を記述するような構文は含まれていない．

3. 提案手法

3.1 ユーザによるランキング項目の表現

本論文では，ユーザがランキング項目を決定するための方法として，クエリの中にそれらの情報を含められるようにした．これは，SPARQL の文法には含まれないので，新たな構文を追加した．ここでは，ランキングの方法に加え，フィルタするための閾値の設定，検索対象のエンドポイントをユーザが決定するものにした．これらは，RANKING()，THRESHOLD()，ENDPOINT() として，クエリの中で表現する．RANKING() では，括弧内でクエリ内のノードを指定し，THRESHOLD() では，括弧内で概念の類似度によるフィルタのための類似度の下限を設定する．ENDPOINT() では，ユーザが検索を行いたい URL を指定する．例えば，次のように指定する．

```
SELECT ?x
   ?x rdf:type my:ClassA .
   ?x rdf:type my:ClassB .
   RANKING(max(my:ClassA)+max(my:ClassB))
   THRESHOLD(my:ClassA=0.2,my:ClassB=0.3)
   ENDPOINT(http://sample/endpoint)
```

このクエリは，my:ClassA,my:ClassB を個人オントロジーにおける概念とし，両者に属するインスタンス (?x) を検索するクエリである．このとき，?x における条件であるクラス（my:ClassA,my:ClassB）とベースオントロジーの対応するクラスとの類似度を含んだマッピングを参照することによって，結果を得る．このクエリでは，my:ClassA,my:ClassB のベースオントロジー内の対応するクラスとの類似度の最大値の和を検索結果のランキングの基準とし，それらの順にランキングする．すなわち，検索結果として得られる?x の属するベースオントロジーのクラスが，my:ClassA,my:ClassB と類似しているほど上位となる．クラスに属するインスタンスを検索するとき，個人オントロジー内の 1 つの概念に対し，ベースオントロジーの複数の概念とマッピングが生じるため，こうした最大値を取るような計算を含めている．また，THRESHOLD(my:ClassA=0.2,my:ClassB=0.3) を設定することにより，対応するベースオントロジーの概念との類似度が，それぞれ 0.2，0.3 以下の概念のみに属するインスタンスはフィルタされる．そして，ENDPOINT(http://sample/endpoint) により，検索対象のエンドポイントを指定している．

表 1，表 2 に，個人オントロジーにおける A,B,C という複数の考慮すべき概念項目が存在する状況を想定し，RANKING() と THRESHOLD() の表現方法を記述する．A とベースオントロジーの対応する概念との類似度を a とし，同様に B と b，C と c とする．ここでは，RANKING() と THRESHOLD() の指定について，いくつかのパターンを示す．

RANKING での表現	ランキング方法
A+B+C	sim=a+b+c
max(A)+max(B)+max(C)	sim=max(a)+max(b)+max(c)
A=0.6,B=0.4,C=0.3	sim = a*0.6+b*0.4+c*0.3
A ＞ B ＞ C	sort order=(a → b → c)

表 1: ランキング記述

THRESHOLD での表現	フィルタ方法
A=0.3,B=0.2,C=0.4	a ＞ 0.3, b ＞ 0.2, c ＞ 0.4 のみの結果を取得
max(A)=0.5, max(B)=0.4,max(C)=0.6	max(a) ＞ 0.5，max(b) ＞ 0.4, max(c) ＞ 0.6 のみの結果を取得

表 2: フィルタ記述

3.2 クエリ変換の手法

ここでは，クエリの変換の処理について，そのアルゴリズムの概観を示す．以下の関数 queryConversion(Q, Mapping) は，ユーザのクエリ（Q）とオントロジーマッピング（Mapping）を引数として与え，変換されたクエリ（Q'）を返すものである．getPattern(Q) において，クエリのパターンを解析する．そして，applyPattern(Q,RewritingPattern,Mapping) において，そのパターンとユーザのクエリ，マッピングを用いて，クエリを変換する．この変換では，SPARQL を部分的に変換していくことを示しており，必要に応じて再帰的に呼び出される．

```
function queryConversion( Q, Mapping )

 if(not conversionRequired(Q))
  return Q;
 else if( Q is single block ) then
  RewritingPattern = getPattern(Q);
  Q' = applyPattern(Q,RewritingPattern,Mapping);
 else
 foreach query block b in Q
 do
   b' = queryConversion( b, Mapping );
   add b' to Q'
 end foreach

 return Q'
```

4. 実データへの適用と予備的評価

4.1 準備

提案したクエリ変換手法の適用した際の性能などについての予備的な評価のために，個人オントロジーとベースオントロジーを実際に準備した．これらのオントロジーのドメインは，ボーカロイド音楽動画とし，Protégé 4.1 を用いて作成した．

ベースオントロジーについては，筆者自身が作成し，個人オントロジーについては，研究室のメンバー 8 人に作成依頼をした．エンドポイントに用いられるベースオントロジーは，クラス数が 83，オブジェクトプロパティが 9，データタイププロパティが 18，動画としてのインスタンスが約 120 であり，エンドポイント全体で約 8000 トリプル程度の規模である．個人オントロジーを制作した被験者は，オントロジー作成の経験が無かったため，事前に他のドメインのオントロジー制作についてのレクチャーを行い，その後，ボーカロイド音楽動画オントロジーの作成を依頼することとした．オントロジー制作についてのレクチャーは，Protégé の利用法を含めたオントロジー制作一般について，3 時間ほど行った．その後，被験者は，平均 6 時間ほどかけてオントロジーを作成し，その都度質問を受け付けた．オントロジーの作成依頼では，クラスのインスタ

ンスにあたる動画を 30 程度含めるよう依頼した．そして，エンドポイントには，実験のために被験者が定義したインスタンスを含めるように後から作成した．

本論文では，8 つの個人オントロジーのうち，本実験で問題なく使用が可能と判断された 5 つの個人オントロジーを利用し，共通のインスタンスが存在するクラス間にてマッピングを追加した．マッピングは合計 1460 であり，類似度については著者自らによって付与するものとした．

4.2 適用結果

ここでは，単一のクラスに属するインスタンスを検索するクエリ，積集合クラスに属するインスタンスを検索するクエリ，和集合クラスに属するインスタンスを検索するクエリについて図 3-5 に検索結果を示す．これは，個人オントロジーを用いて個人オントロジーに対してクエリした結果を正しい結果と定義し，その出現位置を示したものである．それぞれについて，(A) 概念間の類似度が高いクラスを用いた検索 (最大の類似度 ≧ 0.8，10 クエリの合計) と (B) 概念間の類似度が低いクラスを用いた検索 (最大の類似度≦ 0.6，10 クエリの合計) の 2 パターンの検索を行い，計 60 クエリを出力した．縦軸は出現したインスタンスの数を示しており，横軸はインスタンスが出現した位置を示している．インスタンスが出現した位置とは，ソートされた検索結果における，正しい結果の位置であり，番号が小さいほど上位である．

単一のクラスに属するインスタンスを検索するクエリと和集合クラスに属するインスタンスを検索するクエリでは，インスタンスの属するクラスの類似度（複数出現）の最大値をもとにランキングし，積集合クラスに属するインスタンスを検索するクエリでは，最大値の和とした．

これらの結果により，いずれの場合もクエリ結果に概念間類似度を反映することで目的とするインスタンスを見つけやすくなった．

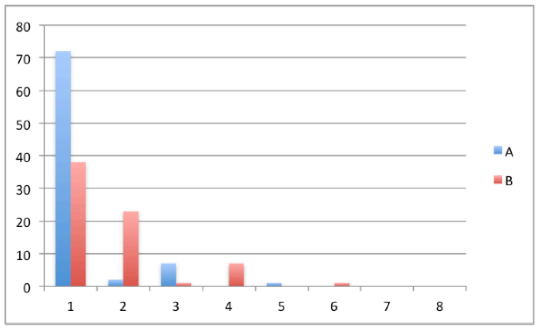

図 3: 単一のクラスに属するインスタンス検索

5. まとめ

本論文では，ユーザのクエリで参照しているオントロジーと検索対象のオントロジーが異なっている状況における，クエリ変換について検討した．クエリ変換では，参照しているオントロジーの概念と検索対象のオントロジーの概念との類似度が与えられていることを仮定し，それらの類似度を用いて，結果をランキング，類似度の低い結果をフィルタするための方法を用いた．本論文で示した実装では，SPARQL に一部の記述を加え，ランキング・フィルタを行い，概念の類似度をもとにした操作を行うことを可能にした．

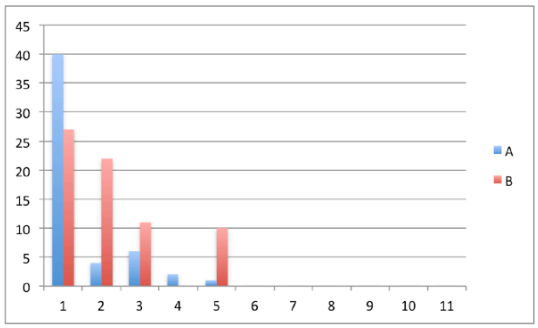

図 4: 積集合クラスに属するインスタンス検索

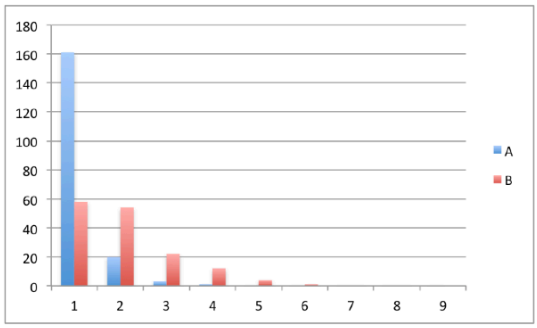

図 5: 和集合クラスに属するインスタンス検索

参考文献

[Ladwig 10] Ladwig, G., and Tran, T.: Linked Data Query Processing Strategies, Proc. 9th International Semantic Web Conference (ISWC2010), Vol. 1, pp. 453–469, (2010).

[Makris 12] Makris, K., Bikakis, N., Gioldasis, N., and Christodoulakis, S.: SPARQL-RW: Transparent Query Access over Mapped RDF Data Sources, 15th International Conference on Extending Database Technology (EDBT2012), (2012).

[Noy 09] Noy, N.: Ontology Mapping, Staab, S. and Studer, R.(Eds.), Handbook on Ontologies, pp. 573–590, (2009).

[Kollia 11] Kollia, I., Glimm, B., and Horrocks, I.: SPARQL Query Answering over OWL Ontologies, Proc. 8th Extended Semantic Web Conference(ESWC2011), Vol. 1, pp. 382–396, (2011).

[藤野 12] 藤野敬久，福田直樹:SPARQL を用いたセマンティックウェブ検索結果に対するパーソナライズドランキングの実現，情報処理学会第 74 回全国大会，(2012).

[Ritze 10] Ritze, D., Völker, J., Meilicke, C., and Šváb-Zamazal, O.: Linguistic Analysis for Complex Ontology Matching, Proc. The Fifth International Workshop on Ontology Matching(OM-2010), (2010).