1N4-OS-19a-5in2

オントロジーマッピングを用いたSPARQLクエリにおけるマッピングに関する詳細知識を必要としないシステムの試作

Toward a SPARQL Query System Utilizing Ontology Mappings without Detailed Knowledge on Mappings

足立拓也 *1　福田直樹 *2
Takuya Adachi　Naoki Fukuta

*1静岡大学情報学部情報科学科
Computer Science, Faculty of Informatics, Shizuoka University

*2静岡大学学術院情報学領域
Department of Informatics, Shizuoka University

SPARQLoid is an extended query language that allows SPARQL user to utilize weighted ontology mappings on the queries. Those queries want knowledge about detailed ontology and ontology mappings that are extended to specify reliability degrees to limit unnecessary of searches when using ontology mappings. SPARQLoid makes it easy to write such queries by using the extended query syntax that allows the query to control the order or limit of produced results based on the specified parameters. This paper presents some techniques to find the vocabulary mapping on the query, to show basis of parameter by the help of a word-relation vectorization technique (i.e., Word2vec). In our approach, users can write simple SPARLoid queries such as well-known SPARQL queries without special knowledge about detailed structure of mappings.

1. はじめに

セマンティックウェブの技術の 1 つとして，RDF 形式で記述されたデータを検索するクエリ言語である SPARQL がある．SPARQL は RDF 形式で記述されたオントロジーの詳細な知識を有していないと，クエリ記述をすることが容易ではない．詳細な知識を有しているオントロジーを用いて，他のオントロジーに対して SPARQL のクエリを発行する技術として SPARQLoid[1, 2, 3, 4] がある．SPARQLoid では，オントロジーマッピング [5] を効果的に利用するために拡張された SPARQL クエリの記述を行い，通常の SPARQL エンドポイント上で実行させることが可能な SPARQL クエリに変換することができる．SPARQLoid がオントロジーマッピングに基づく SPARQL クエリを発行するには 2 通りの方法があり，マッピングデータを外部のエンドポイントに格納しておき，クエリ実行時にそのマッピングを参照する方法を利用することがある．本研究では，ユーザがマッピングの詳細な情報を意識せずに，クエリを記述した場合にも，マッピングを効果的に利用可能なクエリへ拡張を行えるための機構の試作について述べる．

2. 試作システムの構成

2.1 試作システムの概要

本試作システムは，拡張構文で扱うパラメータの基準となる指標として，オントロジーマッピングで付与された確信度と Word2vec[6] で得られた語彙間の類似度を利用する．オントロジーマッピングで付与された確信度及びマッピング先の語彙が不明確である場合でも，Word2vec で得られた語彙間の類似度をある種のマッピングのように扱うことにより，どのような語彙と語彙がマッピングされているのかの把握の支援を行う．

例えば，Listing 1 に示す SPARQLoid クエリを記述した場合，オントロジーマッピングとして表 1 のようなマッピングが使われたとする．

このクエリの中に用いられている語彙である dbo:Person は，検索対象となるオントロジーには用意されているものではない．SPARQLoid では，こうしたクエリを行う際に語彙に対するマッピングを用いることにより，マッピングされた語彙である edam:data_1692 を用いたクエリへの変換を行う．マッピングを効果的に利用するために，THRESHOLD 句や CRITERIA 句，RANKING 句を記述し，フィルタリングやソートをすることができる．

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-
    schema#>
PREFIX edam: <http://edamontology.org/>

SELECT distinct ?person ?label
WHERE {
 ?person ?p dbo:Person .
 ?person rdfs:label ?label .
 THRSHOLD { dbo:Person >= 0.5 }
 CRITERIA ?c { dbo:Person * 100 }
 RANKING ?score { ?c }
}limit 100
```

Listing 1: SPARQLoid クエリ例

表 1: dbo:Person におけるマッピング例

MappedConcept	Confidence
edam:data_1692	0.51

表 1 にある概念 IRI edam:data_1692 のみでは，どのような概念を指すものかがわかりづらく，この概念を含んだマッピングにどの程度の確信度の閾値等をつけてクエリを行えば目的とするクエリ結果が得られるかが，容易にはわからない．そうした際に，本試作システムでは，Word2vec を用いて，後述する方法に基づいて”Person”と対応する edam:data_1692 に対する語彙の類似したものを提示することができる表 2 に”Person”に対する類似語彙とその類似度の結果を示す．このように，語彙と類似度を参考にしながらパラメータの調整を行える．

2.2 拡張構文のパラメータ

本試作では Word2vec の語彙間の類似度計算を用いて，オントロジーマッピングで付与された確信度を利用する拡張構文で扱うパラメータの基準となる指標の提示を行う．図 1 に

連絡先: 足立拓也，静岡大学情報学部情報科学科，
〒 432-8011 静岡県浜松市中区城北 3-5-1，
cs13007(at)s.inf.shizuoka.ac.jp

表 2: ”Person”に基づく類似度計算の結果

Words	Cosine Distance
Name	0.510069
bereits	0.492998
ein	0.486285
fordert	0.478758
bedeutet	0.474116

Word2vec を用いたオントロジーマッピング確信度のオンデマンド生成の例を示す．図 1 にあるように，オントロジーに含まれている語彙の IRI から抽出した文字列に対して，Word2vec を用いて語彙間の類似度を計算し，マッピングに確信度を追加的に付与する．オントロジーマッピングで付与された確信度と Word2vec で付与された語彙文字列間の類似度に基づく確信度の 2 つを相互参照できるようにすることにより，SPARQLoid クエリへのパラメータ調整の検討を容易にする．

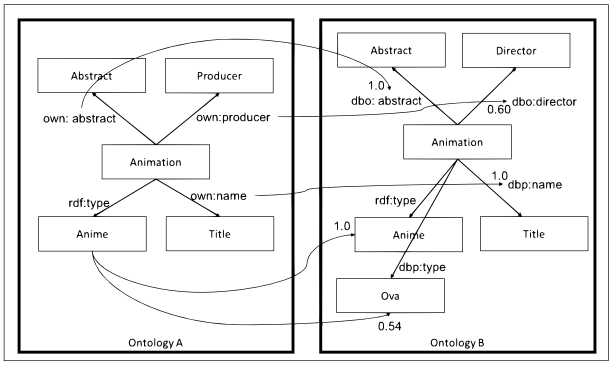

図 1: Word2vec を用いたオントロジーマッピング確信度のオンデマンド生成の例

SPARQLoid クエリでは，SPARQL クエリ記述の他に拡張された構文を記述する．拡張された構文ではマッピングに付与された確信度を利用するためのパラメータを記述できる．

例えば，THRESHOLD 句（閾値指定）における閾値を大きくすると，確信度が高いマッピングのみを検索の対象としたクエリ結果を取得することが可能である．また，閾値を小さくすると，確信度の低いマッピングも含めた，より広い範囲でのクエリ結果を取得することが可能である．

このような SPARQL, SPARQLoid クエリを記述する上での課題として，いくつか挙げられる．クエリ中の語彙を記述するためには，対象となるオントロジーの語彙を知らなければいけなく，ユーザが求めるクエリ結果を取得できるクエリを記述するためには，オントロジー内の語彙の関係を把握する必要がある．また，拡張構文に記述するパラメータに対して基準となる指標が必要であり，記述したクエリ中のどの語彙がマッピングされているのかを把握するためには，マッピングデータの詳細な知識を有している必要がある．

2.3 クエリ中の語彙の抽出

パラメータの指標として，ここまで述べたような Word2vec による類似語彙計算を行うためには，クエリ中にある語彙文字列表現の抽出が必要となる．本試作では，クエリ中にある語彙文字列表現の抽出を行うための機構を試作した．クエリ中に記述される語彙は，クエリ中への IRI の記述と名前空間接頭辞を用いて省略した IRI の記述の 2 通りある．これらの語彙の記述に対して，語彙の抽出を行い，語彙の提示を行う．

抽出された語彙の中からマッピングが施された語彙の探索については，マッピングデータを格納した SPARQL エンドポイントに対して，クエリ発行を行う手段を検討している．このようにして抽出された語彙に対して，Word2vec による語彙の類似度計算を行う．

2.4 試作システムの構成図

図 2 に試作システムの構成図を示す．

本試作システムでは，オントロジーを 2 つ選択し，システム上にて Alignment API を用いてマッピングデータを生成し，Apache Jena Fuseki[*1] を用いて選択したオントロジーや生成したマッピングデータを格納した SPARQL エンドポイントを起動する．クエリを記述する際に Word2vec によるパラメータの指標の提示を行い，記述した SPARQLoid クエリに対して，起動した SPARQL エンドポイントにクエリを発行する．また，キーワードを含む SPARQL クエリに対して，キーワードを補完し，SPARQL Endpoint Status[*2] から取得したエンドポイントから記述したクエリに適しているエンドポイントを探索する機構 [7] の導入を検討している．

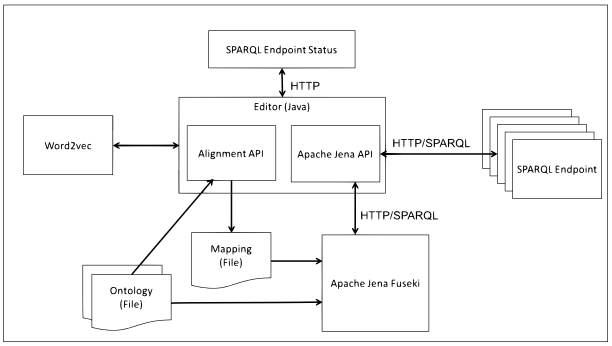

図 2: システムの構成図

3. Word2vec による語彙類似度取得

本研究では，オントロジーマッピングで得られたオントロジー間の語彙の関係のように，Word2vec を用いてオントロジー間の語彙の関係を得られることを想定している．

本試作で必要となるこれらの Word2vec による類似語彙計算のために，語彙の関係を Word2vec に学習させるための学習データの準備方法について検討する必要がある．

本節では，SPARQL Endpoint Status からエンドポイントのリストおよび可用性情報を取得し，リスト中の可用性の高いエンドポイントからトリプルを取得するクエリを実行し，取得したトリプルを基に学習させる方法を採る．また，それらから収集した語彙に対して類似度計算を行うことがどの程度可能であるかを次の手順により確かめる．

3.1 Word2vec の学習データの用意

本実験では，取得したトリプルに対して，以下のような学習データを用意した．学習データは無加工データを取得し，加工することにより，統一データ，IRI 省略データ，IRI 省略統一データを作成した．Listing 2, 3, 4, 5 にそれぞれのデータの例を示す．表 3 に学習データの詳細な情報を示す．

∗1 https://jena.apache.org/documentation/fuseki2/index.html
∗2 http://sparqles.ai.wu.ac.at

無加工データ

エンドポイントから得られたトリプルを加工無しに扱う．

```
<http://example/ontology/Animation> <http://
    www.w3.org/1999/02/22-rdf-ns#type> <http
    ://example/ontology/Anime>
<http://example/ontology/Animation> <http://
    example/ontology/name> "Anime"@en
```

Listing 2: 無加工データ例

統一データ

エンドポイントから得られたトリプルに対して，Subject に該当する語彙が一致するトリプルを 1 つの行として扱う．

```
<http://example/ontology/Animation> <http://
    www.w3.org/1999/02/22-rdf-ns#type> <http
    ://example/ontology/Anime> <http/example/
    ontology/name> "Anime"@en
```

Listing 3: 統一データ例

IRI 省略データ

エンドポイントから得られたトリプルに対して，IRI を省略して扱う．

```
Animation type Anime
Animation name "Anime"@en
```

Listing 4: IRI 省略データ例

IRI 省略統一データ

エンドポイントから得られたトリプルに対して，IRI を省略し，Subject に該当する語彙が一致するトリプルを 1 つの行として扱う．

```
Animation type Anime name "Anime"@en
```

Listing 5: IRI 省略統一データ例

表 3: 学習データの詳細

Data	Input file size	lines
無加工データ	1.69 GB	11089366
統一データ	1.46 GB	6528945
IRI 省略データ	401 MB	11089445
IRI 省略統一データ	326 MB	5832097

3.2 学習結果

表 4 に各学習データを Word2vec に学習させた出力データを示す．

Data には 3.1 節で述べた学習データ，Input size には用意した学習データのファイルサイズ，Output size には Word2vec による出力データのファイルサイズ，Vocab は Vocab size であり，Word2vec によって求められた語彙の大きさ，Words は words in file であり，Word2vec によって求められた学習データにある語彙の総数を示している．

表 4: Word2vec による出力データ 2

Data	Output file size	Vocab	Words
無加工データ	1.69 GB	639531	47542588
統一データ	1.46 GB	351649	37743909
IRI 省略データ	401 MB	332900	39065667
IRI 省略統一データ	326 MB	198128	29234500

3.3 IRI 語彙間の類似度の計算例

学習データを Word2vec を用いた学習させた結果を基に，類似度の計算を行った．

ユーザが記述すると予想される SPARQLoid クエリを用いることにより，記述されているクエリからマッピングされている語彙を抽出する．抽出した語彙に対して，学習データを用いて Word2vec による類似度計算を行い，取得可能か確認する．また，取得された語彙の中にマッピング先の語彙があるか確認する．

マッピングデータの生成は，Alignment API を利用した．Alignment API は文字列の類似度などに基づいてマッピング生成をしており，今回用いたクエリの語彙に対して Levenshtein Distance を用いた方法では，有効な類似度を取得することができなかった．

同様にマッピング生成に LogMap[8, 9, 10] を用いた場合も試した．LogMap は推論に基づく高性能なマッピング生成を実現している．

一例として，”University” と”College”がマッピングが施されており，”University”に基づく Word2vec の類似度計算によって”College”を統一データと IRI 省略データ，IRI 省略統一データから取得することができた．表 5 に”University”に基づく Word2vec の類似度計算の結果を示す．Cosine distance は語彙の類似度，Ranking は Word2vec の類似度を降順出力されている際の”College”の順位を表している．

表 5: ”University”に対する”College”の Word2vec 類似度と類似順位

Data	Cosine distance	Ranking
統一データ	0.532492	33
IRI 省略データ	0.470986	4
IRI 省略統一データ	0.434365	16

3.4 考察

それぞれの学習データでは，最大類似度の値に差が見られた．IRI 表現の語彙である類似語彙取得が少なく見られた．

これらの情報からだけでは，かならずしも Word2vec で得られたデータの特性はわからないが，Word2vec の学習手法と本研究が目指す手法に合った適切な学習データの生成・生成クエリの検討が必要であると考える．また，無加工データと統一データの出力結果には”,”や”(”，”)”といった記号が組み合わさった語彙があったほか，”University”と”university”といった大文字小文字の違う語彙で出力結果が異なるなどの場合が見られた．学習データの加工における適切な前処理の方法ついても検討が必要があると考える．

4. 試作システムの動作例

図 3 に Word2vec の動作例を示している．図 3 では，”http://www.w3.org/1999/02/22-rdf-syntax-ns#type”

という語彙に対して，実験で用いた Word2vec の学習データである無加工データを用いて語彙の類似度計算を行っている．

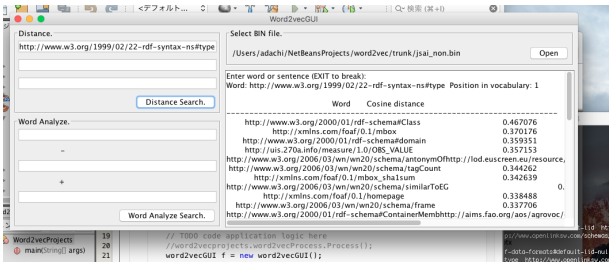

図 3: Word2vec による類似語彙の提示例

図 4 にクエリ中にある語彙を抽出する機構の動作例を示している．Listing 6 に図 4 で記述されているクエリを示す．このようなクエリが記述された際，本機構では IRI 表記で記述されている語彙を抽出を行うことができ，また接頭辞を用いた省略された語彙に対しても語彙の抽出を行うことができる．その際，語彙の統一を図るために，省略された語彙に対して IRI 表記にして抽出を行う．

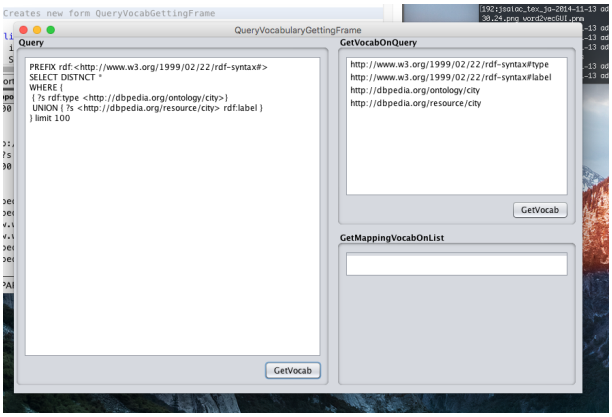

図 4: クエリ中の語彙情報抽出機構の動作例

```
PREFIX rdf:<http://www.w3.org/1999/02/22/rdf-
    syntas#>
SELECT DISTNCT *
WHERE {
  { ?s rdf:type <http://dbpedia.org/ontology/city
      >}
  UNION { ?s <http://dbpedia.org/resource/city>
      rdf:label }
} limit 100
```

Listing 6: 図 4 で記述したクエリ例

5. おわりに

本試作システムでは，オントロジーマッピングの詳細な情報を意識せずに記述したクエリをマッピングを利用するために拡張された SPARQL クエリへと記述するために，記述したクエリ中にあるマッピングされている語彙の探索機構，拡張構文に記述するためのパラメータの指標として語彙をベクトルとして表現する Word2vec を用いて提示する機構の試作を行った．今後の課題としては，たとえば Inants らのアプローチ [11] のような，複数のオントロジーマッピングを組み合わせてより複雑なマッピングを構成できる理論的枠組を導入した場合に対応可能な手法の検討が挙げられる．

謝辞

本研究の一部は，JST CREST の支援を受けたものである．

参考文献

[1] Fujino, T., Fukuta N.: A SPARQL Query Rewriting Approach on Heterogeneous Ontologies with Mapping Reliability, In: Proc. of the IIAI International Conference on e-Services and Knowledge Management (IIAI-ESKM2012), pp. 230-235, 2012

[2] Fujino, T., Fukuta, N.: SPARQLoid - a Querying System using Own Ontology and Ontology Mappings with Reliability, In: Proc. of the 11th International Semantic Web Conference (Posters & Demos) (ISWC2012), 2012

[3] Fujino, T., Fukuta, N.: Utilizing Weighted Ontology Mappings on Federated SPARQL Querying, In: Proc. of the 3rd Joint International Semantic Technology Conference (JIST2013), 2013

[4] Fujino, T., Fukuta, N: Utilizing Weighted Ontology Mappings on Federated SPARQL Querying, Lecture Notes in Computer Science,Vol.8388,pp.331–347,2014.

[5] Atencia M., Borgida A., Euzenat J., Ghidini C., Serafini L.: A Formal Semantics for Weighted Ontology Mappings, In: Proc. of the 11th International Semantic Web Conference (ISWC2012), 2012.

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean.: Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013.

[7] 足立拓也, 山田直希, 野口宙毅, 福田直樹.: オントロジーマッピングに基づく SPARQL クエリ記述支援システムの拡張機構の試作, 第 38 回セマンティックウェブとオントロジー研究会 (SIG-SWO38), 2016.

[8] Ernesto, R., Bernardo, G., Yujiao Z., Ian H.: Large-scale Interactive Ontology Matching: Algorithms and Implementation, In the 20th European Conference on Artificial Intelligence (ECAI 2012), 2012.

[9] Ernesto, R., Bernardo, G.: LogMap: Logic-based and Scalable Ontology Matching, In the 10th International Semantic Web Confernece (ISWC 2011), 2011.

[10] Ernesto, R., Christian M., Bernardo, G., Ian H.: Evaluating Mapping Repair Systems with Large Biomedical Ontologies, In 26th International Workshop on Description Logics (DL 2013), 2013.

[11] Armen Inants, Jerome Euzenat.: An Algebra of Qualitative Taxonomical Relations for Ontology Alignments, In: Proc. of the 14th International Semantic Web Conference (ISWC2015), 2015.