SIG-SWO-045-02

複数の大規模RDFデータセットを統合したキーワード検索

Keyword Search in Multiple Large RDF Datasets

山中 佑紀[1*]　兼岩 憲[1]
Yuuki Yamanaka[1]　Ken Kaneiwa[1]

[1] 電気通信大学大学院 情報理工学研究科 情報・ネットワーク工学専攻
[1] Department of Computer and Network Engineering, Graduate School of Informatics and Engineering, The University of Electro-Communications

Abstract:
セマンティック Web では，RDF で記述されたリンクトデータの規模が拡大しており，その活用が重要となっている．通常，RDF データの検索はクエリ言語 SPARQL を用いて詳細な問い合わせを実現する．しかし，SPARQL による検索は，クエリの記法に加え多くのデータセット固有の知識が利用者に要求され，このことがリンクトデータを統合して有効活用することを難しくしている．そこで本研究では，複数のキーワード間の関係性を導く検索と，複数の RDF データセットを統合して容易に検索するための同値関係プロパティの推論方法を提案する．

1　はじめに

現在，DBpedia [1] や Wikidata [2] といったハブデータセットを中心として，様々なドメインのリンクトデータが RDF(Resource Description Framework) [3, 4] を用いて公開されている．セマンティック Web [5] の進展に伴ってその数や量は増加しており，それらを活用して検索する重要度が増している．RDF データの検索手段に，クエリ言語 SPARQL があり，詳細な問い合わせを可能にする．

しかし，利用者は SPARQL クエリの記法を理解する必要があり，どのようなスキーマやリソース URI が異なるリンクトデータの記述に用いられるかという知識が必要となる．さらに，異なるデータセットを統合して多種多様なデータを検索してリンクトデータを活用するのは容易ではない．こういった理由でリンクトデータの幅広い活用を難しくしている．

一般の文書検索は，キーワードを入力する簡便な検索を広く用いる．これと同じように，RDF データのキーワード検索が考案されており，top-k 検索 [6]，k-NK 検索 [7]，K-FROST [8] などがある．これらの検索は，RDF グラフからキーワードに関連した部分グラフを出力する．しかし，部分グラフの検索は組み合わせが膨大なため，計算コストの増大が問題となる．また，先行研究では複数の RDF データセットを統合したキーワードの検索を扱っておらず，その実現には計算コストとデータ統合の複雑さのバランスが重要となる．

本研究では，最短 RDF パスによるキーワード検索を拡張した，集結パスによる検索方法を提案する．この集結パスの検索は，複数のリソースから到達できる共通のノードへのパスを見つける．リンクトデータは，外部リンクにより他のデータセットと意味的に繋がりをもつ．その一方でセマンティック Web では，同じリソースに対してデータセットごとに異なる URI が割り当てられる．これは Web の特性上避けられず，`owl:sameAs` などの同値関係プロパティによってリソースの同一性が明示される．本研究では，同値関係にあるリソース URI の集合を推論して 1 つのリソース URI にまとめることで，RDF グラフ上で集結パスによるキーワード検索を強化する．

本稿の構成は，次の通りである．まず 2 章で RDF グラフと RDF パスの定義を行い，RDF グラフ上におけるキーワード検索について述べる．3 章では，先行研究を発展させてキーワードからの集結パス検索について述べる．4 章では，同値関係プロパティの推論により RDF のデータセットを統合する手法について述べる．5 章では，同値類リソースを用いた RDF パス検索の評価実験を行い，6 章で結論と今後の課題を述べる．

2　RDF とキーワード検索

2.1　RDF グラフと RDF パス

本稿で用いる RDF グラフ･RDF パスの定義について述べる．U をリソース URI の集合，B を空白ノードの

*連絡先：電気通信大学大学院 情報理工学研究科
情報・ネットワーク工学専攻
〒 182-8585 東京都調布市調布ヶ丘 1-5-1
E-mail:yamanaka@sw.cei.uec.ac.jp

集合，L をリテラル (文字列) の集合とする．主語リソース $s \in U \cup B$，述語リソース $p \in U$，目的語リソース $o \in U \cup B \cup L$ の 3 つ組 (s, p, o) を RDF トリプルと呼び，`s p o .` と表記する．また，RDF トリプル (s, p, o) を，主語と目的語の位置を入れ替えて (o, p^{-1}, s) と記述したとき逆トリプルと呼ぶ．このとき，p を順プロパティ，p^{-1} を逆プロパティといい，p または p^{-1} を p^* で表す．RDF トリプルの有限集合 $G \subseteq (U \cup B) \times U \times (U \cup B \cup L)$ を RDF グラフと呼ぶ．

定義 2.1 (RDF ウォーク) *RDF* グラフ G 内の，$d-1$ 個の主語と目的語が一致する *RDF* トリプル列

$$(r_0, p_1^*, r_1), (r_1, p_2^*, r_2), \ldots, (r_{d-1}, p_d^*, r_d)$$

を長さ d の *RDF* ウォークと呼び，これを

$$(r_0, p_1^*, r_1, p_2^*, r_2, \ldots, r_{d-1}, p_d^*, r_d)$$

と $2d+1$ 個の要素列としても記述する．リソース r_0 を起点とする長さ *0* の *RDF* ウォークを (r_0) と記述する．また，*RDF* ウォークを構成するすべての主語と目的語の列 $(r_0, r_1, \ldots, r_d)$ を *RDF* ウォークのノード列と呼ぶ．

定義 2.2 (RDF パス) 閉路を含まない *RDF* ウォークを *RDF* パスと呼ぶ．特に，*RDF* パス $(r_0, p_1, r_1, p_2, r_2, \ldots, r_{d-1}, p_d, r_d)$ を順方向 *RDF* パス，*RDF* パス $(r_0, p_1^{-1}, r_1, p_2^{-1}, r_2, \ldots, r_{d-1}, p_d^{-1}, r_d)$ を逆方向 *RDF* パスと呼ぶ．順方向 *RDF* パスと逆方向 *RDF* パスを総称して単方向 *RDF* パスと呼ぶ．また，同一の起点と終点をもつ *RDF* パスのうち，長さが最も小さいものを最短 *RDF* パスと呼ぶ．

2.2 RDF グラフ上のキーワード検索

RDF グラフを対象としたキーワード検索は，キーワードに関連した部分グラフを抽出する問題である [7, 8]．2 つのキーワードを入力して，それぞれを含むリソース URI を起点と終点とする．RDF グラフ上における起点と終点の全ての組み合わせを結ぶ RDF パスを検索し，出力する．

濱松ら [8] は，最短の順方向 RDF パスのみに制限した上で，起点からの距離付き到達可能リストを利用した高速な探索アルゴリズムを提案している．

定義 2.3 (距離付き到達可能リスト) *RDF* グラフ内の任意のリソース r について，距離付き到達可能リスト A_r は，起点 r から順トリプルを辿り距離 $0, 1, \cdots, d$ で初めて到達できる目的語リソースの集合からなるリスト $A_r = (A_r[0], A_r[1], \cdots, A_r[d])$ である．ここで，$A_r[i]$ は r から最短距離 i 丁度で到達可能なリソースの集合なので，同じリソースは A_r 内に一度しか出現しない．ただし，$A_r[0] = \{r\}$ とする．

以上の到達可能リストは，述語の情報を保持せずに探索した目的語リソースのみを記憶する．RDF グラフ G のトリプル数を n としたとき，到達可能リストのリソース数は高々$O(n)$ である．そのため，計算量爆発を起こさずにパスを検索できる．検索過程で失った述語は到達可能リストに含まれる順方向 RDF パスのノード列を元に補完して最短 RDF パスを再構築する．

3 集結パスによるキーワード検索

2.2 節で述べたキーワード検索を発展させて，2 つ以上のキーワードによる集結パス検索について述べる．

3.1 集結パス集合

RDF グラフ G 内のリソース $s, c \in U \cup L$ に対し，s を起点とし c を終点とする最短順方向 RDF パスの集合を $\mathcal{P}(s, c)$ と表す．また，同様に最短逆方向 RDF パスの集合を $\mathcal{P}^{-1}(s, c)$ と表す．

定義 3.1 (集結パス集合) *RDF* グラフ G 内の要素数 *2* 以上のリソース集合 $R_k = \{s_1, s_2, \ldots, s_n\} \subset U \cup L$ に対して，以下を R_k の順方向の集結パス集合と呼ぶ．

$$\mathcal{P}(R_k) = \{\bigcup_{s_i \in R_k} \mathcal{P}(s_i, c) | {}^{\exists} c \in U \cup L, {}^{\forall} s_i \in R_k [\mathcal{P}(s_i, c) \neq \emptyset]\}$$

同様に $\bigcup_{s_i \in R_k} \mathcal{P}^{-1}(s_i, c)$ からは逆方向の集結パス集合 $\mathcal{P}^{-1}(R_k)$ が定義できる．$\mathcal{P}(R_k)$ と $\mathcal{P}^{-1}(R_k)$ の要素をそれぞれ順方向と逆方向の集結パスと呼び，R_k を起点ノード集合，c を共通ノードと呼ぶ．

例えば，$R_k = \{r_{k_1}, r_{k_2}, r_{k_3}\}$ としたとき，$\{(r_{k_1}, p, c), (r_{k_2}, p, r_1, p_1, c), (r_{k_2}, p, r_1, p_2, c), (r_{k_3}, p, r_2, p, c), (r_{k_3}, p, r_3, p, c)\}$ は c を共通ノードとした順方向集結パスであり，$\mathcal{P}(R_k)$ の要素である (図 1)．これは 3 つのリソース $r_{k_1}, r_{k_2}, r_{k_3}$ が共通点 c をもち，それぞれの c に対する関係性を表す．

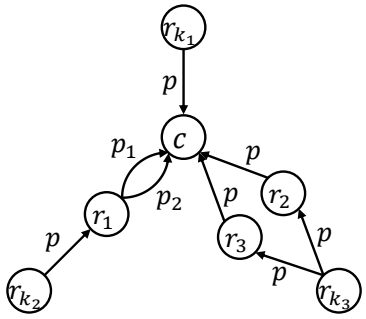

図 1: 集結パスの例

3.2 キーワードからの集結パス検索

集結パスを用いた RDF グラフのキーワード検索を以下のように定義する．まずキーワードに最も一致する RDF グラフ上のリソース URI を得るために次の過程が考えられる．

1. キーワードを入力する．
2. RDF グラフ内の `rdfs:label` や `skos:altLabel` 属性との文字列マッチングによってキーワードに近いリソース URI を検索し，出現頻度の高さや次数といった指標でソートして，`rdfs:comment` や `foaf:depiction` の説明文や画像などと一緒に提示する．
3. キーワードそれぞれに対してリソース URI を 1 つ選択する．

定義 3.2 (集結パス検索) *RDF* グラフ上の集結パス検索は，*2* つ以上のキーワードの集合 K を入力とし，K の要素それぞれに最も一致するリソース *URI* の集合 R_k から共通ノード c への集結パスを出力する．以下の手順より，リソース集合 R_k から集結パスを得る．

1. 集結パスを構成する順 (逆) 方向 *RDF* パスの最大長を d_{max} とする．$s_1, \ldots, s_n \in R_k$ に対して，最大距離 d_{max} のそれぞれの到達可能リスト A_{s_1}, $\ldots$, A_{s_n} を探索する．
2. すべての $s_i \in R_k$ に共通する要素 (共通点) として $R_c = \bigcap_{s_i \in R_k}(A_{s_i}[1] \cup \cdots \cup A_{s_i}[d_{max}])$ を求める．
3. すべての $s_i \in R_k$ を起点として，ある長さ $d(\leq d_{max})$ の各終点 $r \in R_c \cap A_{s_i}[d]$ への順 (逆) 方向 *RDF* パスを検索する．その際，到達可能リストからノード列 $(s_i, r_1, \ldots, r_{d-1}, r)$ を元に述語を補完して最短順 (逆) 方向 *RDF* パス $\mathcal{P}(s_i, r)$ $(\mathcal{P}^{-1}(s_i, r))$ を復元する．
4. 各共通点 $r \in R_c$ に対して，$\bigcup_{s_i \in R_k} \mathcal{P}(s_i, r)$ $(\bigcup_{s_i \in R_k} \mathcal{P}^{-1}(s_i, r))$ が R_k の集結パスとなる．

集結パス検索の例を次に示す．$R_k = \{$ `dbr-ja:`上杉謙信, `dbr-ja:`武田信玄, `dbr-ja:`伊達政宗, `dbr-ja:`明智光秀 $\}$ とする．R_k 内のリソースに共通する関係性を調べるため，$d_{max} = 2$ の集結パス集合を求める．

```
{(dbr-ja:上杉謙信, dbo:occupation, dbr-ja:大名),
 (dbr-ja:武田信玄, wdt:P39(公職), dbr-ja:大名),
 (dbr-ja:伊達政宗, wdt:P39(公職), dbr-ja:大名),
 (dbr-ja:明智光秀, wdt:P39(公職), dbr-ja:大名)}
```

この集結パスは R_k 中の 4 人の武将がすべて大名であったことを表す．

```
{(dbr-ja:上杉謙信, dbo:battle, dbr-ja:手取川の
戦い, dbo:commander, dbr-ja:織田信長),
(dbr-ja:武田信玄, dbo:child, dbr-ja:武田勝頼,
dbp-ja:妻, dbr-ja:織田信長),
(dbr-ja:伊達政宗, dbr-ja:主君, dbr-ja:徳川家康,
dbp-ja:主君, dbr-ja:織田信長),
(dbr-ja:明智光秀, dbp-ja:主君, dbr-ja:織田信長)}
```

この集結パスは，4 人の武将の共通点が何らかの形で織田信長と関わりをもつことを表す．

4 同値類リソースによるデータ統合

同じリソースに対して RDF データセットによって異なる URI が付けられる．このとき，`owl:sameAs` などにより，異なる 2 つの URI が同じリソースを意味するため，その同値性を推論してリソースを統合する．

4.1 同値関係プロパティと同値類リソース

複数のリンクトデータを統合するとき，同値関係プロパティで結ばれたリソースを別々のノードとして検索すると，本来の意味的な関係性に影響が生じる．`owl:sameAs` や `owl:equivalentProperty` は OWL [9] 推論で対応されるが，RDF パスなどの探索には反映されない．また，SKOS における `skos:exactMatch` や DBpedia Ontology における `dbo:wikiPageRedirects` のように，厳密ではない同値性を表すプロパティも存在する．本稿では「2 つのリソース URI が同一のリソースを表す」という意味のプロパティを同値関係プロパティと総称する．この同値関係プロパティから導かれる URI の集合と RDF パスを次のように定義する．

定義 4.1 (同値類リソース) *RDF* グラフ G 上のリソース *URI* を $r \in U$ とし，同値関係プロパティの集合を P_{eq} とする．このとき，r の同値類リソース $[r]_{eq}$ を，以下を満たす最小の集合として定義する．

1. $r \in [r]_{eq}$ である．
2. $r_1 \in [r]_{eq} (r_1 \neq r)$，$p^*_{eq} \in P_{eq}$ かつ $(r_1, p^*_{eq}, r_2) \in G$ ならば，$r_2 \in [r]_{eq}$ である．

RDF グラフ上では，リソース r を起点として同値関係プロパティで繋がれたリソース URI が r の同値類リソースに統合される．図 2 は，`dbr-ja:`電気通信大学における同値類リソースの例である．

定義 4.2 (同値類リソース上の RDF パス) あるリソース $s \in [r]_{eq}$ が存在するとき，(r) は長さ *0* の同値類 *RDF* パスである．あるリソース $s \in [r_1]_{eq}, p \in [q]_{eq}, o \in [r_2]_{eq}$ かつ *RDF* トリプル $(s, p^*, o) \in G$ が存在するとき，(r_1, q^*, r_2) は長さ *1* の同値類 *RDF* パスである．$d-1$ 個の主語と目的語が一致する長さ *1* の同値類 *RDF* パス列 $(r_0, q_1^*, r_1), (r_1, q_2^*, r_2), \ldots, (r_{d-1}, q_d^*, r_d)$ を長さ d の同値類 *RDF* パスと呼び，$(r_0, q_1^*, r_1, q_2^*, r_2, \ldots, r_{d-1}, q_d^*, r_d)$ と表す．

このような同値類RDFパスにより，簡潔なRDFパスで多くの関係性を抽出できる．例えば，起点に dbr-ja:織田信長，終点に dbr-ja:織田秀信 (信長の孫) を設定して最大長さ 2 までの順方向の同値類 RDF パスを検索すると，以下が得られる．

(dbr-ja:織田信長, dbo:child, dbr-ja:織田信忠,
dbo:child, dbr-ja:織田秀信)

このとき，RDF グラフは以下の RDF パスを含む．

(dbr-ja:織田信長, owl:sameAs, wd:Q171411,
wdt:P40, wd:Q2614349,
owl:sameAs^{-1}, dbr-ja:織田信孝,
dbp-ja:主君, dbr-ja:織田秀信)

これは，長さ 4 の非単方向 RDF パスである．同値類リソースにより，2 つの owl:sameAs で結ばれたリソース URI が 1 つにまとまる．また，次のプロパティ:P40 と dbo:child も同じ意味のプロパティとみなされる．

```
dbo:child owl:equivalentProperty wd:P40 .
wd:P40 wikibase:directClaim wdt:P40 .
```

この結果，長さ 4 の非単方向パスは，以下の長さ 2 に短くなった順方向の同値類 RDF パスとして検索される．

(dbr-ja:織田信長, dbo:child, dbr-ja:織田信孝,
dbp-ja:主君, dbr-ja:織田秀信)

5 実験

RDF パス検索において同値類リソースによる効果を評価するため，次の実験を行った．実験環境は，CPU:Intel Core i7 6800K，RAM:64GB(JVM Max Heap:56GB)，OS:Windows 10 Education である．無作為に選んだ，長さ 4 以下の順方向 RDF パスが存在する DBpedia Japanese 上の URI の組 1000 個に対し，同値類リソースの使用あり，なしと，大規模 RDF データ D_1(DBpedia Japanese [10] と DBpedia Ontology を統合，N-Triples 形式で 18.7GB)，D_2(D_1 に DBpedia を追加，69.3GB)，D_3(D_2 に Wikidata を追加，168GB) の組み合わせに対して長さ 4 以下の順方向 RDF パスを検索した．以下は，対象とした URI の組の例である．

(dbr-ja:勝浦オークワ，dbr-ja:大阪維新の会)
(dbr-ja:東映，dbr-ja:自由民主党_(日本))
(dbr-ja:名古屋山三郎，dbr-ja:徳川頼宣)
(dbr-ja:那須恵理子，dbr-ja:田村憲久)
(dbr-ja:洞爺湖町，dbr-ja:北海道)
(dbr-ja:道の駅みつまた，dbr-ja:国道 15 号)
(dbr-ja:ハルニレ，dbr-ja:イラクサ目)
(dbr-ja:イソウロウグモ，dbr-ja:サソリ)
(dbr-ja:この世界の片隅に，dbr-ja:感染列島)
(dbr-ja:功名が辻_(NHK 大河ドラマ)，dbr-ja:鶴瓶の家族に乾杯)

起点リソースは dbo:Organisation, dbo:Person, dbo:Event, dbo:MeanOfTransportation, dbo:Place, dbo:Species, dbo:Work のいずれかのインスタンスとし，終点リソースはそれと同じクラスのリソースとする．以下は，実験に用いた同値関係プロパティである．

1. owl:sameAs，skos:exactMatch，wd:P460：2 つのリソース URI の同値性を示すプロパティ
2. owl:equivalentProperty，wd:P1628：2 つのプロパティが同値であることを示すプロパティ
3. owl:equivalentClass，wd:P1709：2 つのクラスに含まれるリソースの集合が，実質的に同一であるような関係を示すプロパティ
4. dbo:wikiPageRedirects：Wikipedia 上で，主語をタイトルにもつページから述語をタイトルにもつページへのリダイレクトが張られていることを示す，DBpedia Ontology のプロパティ
5. wikibase:directClaim：Wikidata において，プロパティを代表する wd をプリフィクスにもつ Item と，Direct claim に実際に用いられる wdt をプリフィクスを持つプロパティとを結ぶプロパティ[1]

表 1: RDF パスの検索結果件数

同値類リソース	不使用			使用		
RDF データ	D_1	D_2	D_3	D_1	D_2	D_3
件数の平均	4.04	4.09	4.09	4.23	10.49	36.22
増加比の平均	-	1.02	1.02	1.02	2.86	11.73

[1] https://www.mediawiki.org/wiki/Wikibase/Indexing/RDF_Dump_Format

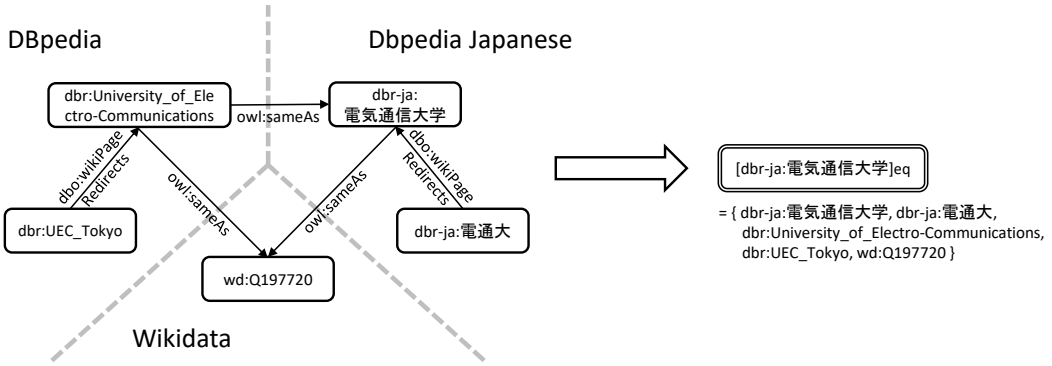

図 2: dbr-ja:電気通信大学と同値なリソースの統合

表 1 は，長さ 4 以下の順方向 RDF パスの検索結果件数の和の平均値と，同値類リソースを用いずに検索した結果の増加比の平均値を示す．同値類リソースを不使用のとき，RDF データを D_1 から D_2, D_3 へと増大しても，検索結果の件数にはわずかな変化しかない．一方，同値類リソースの使用は，データ増に応じて検索結果を増加させている．

表 2: 検索結果に含まれるノード数

同値類リソース	不使用			使用		
RDF データ	D_1	D_2	D_3	D_1	D_2	D_3
ノード数の平均	5.76	5.84	5.84	5.81	7.00	11.03
増加比の平均	-	1.02	1.02	1.00	1.21	1.86

表 2 は，それぞれの起点・終点に対する，検索された RDF パスに含まれる異なるノード数 (起点・終点を除いたノード総数) の和の平均値を示す．同値類リソースを不使用のときはノード数に違いがないが，使用した場合はデータ増に応じてノード数が増加する．これにより，同値類リソースは，複数の RDF データセットを統合するほど多くのノードを経由して意味的な関係性をより多く抽出できるといえる．

表 3: 実行時間

同値類リソース	不使用			使用		
RDF データ	D_1	D_2	D_3	D_1	D_2	D_3
時間の平均/ms	158.71	208.72	319.04	267.15	6058	45580
増加比の平均	-	1.38	2.36	1.50	30.81	399.84

表 3 は，RDF パス検索の実行時間を示す．同値類リソースは，データ増に応じて急激に実行時間を増加させる．これは，同値類リソースの保持によるオーバーヘッドであり，より効率的な検索方法が求められる．

6 結論と今後の課題

本研究では，最短 RDF パスの検索による RDF グラフ上のキーワード検索を拡張させて，2 つ以上のキーワードに対する，集結パス検索を提案した．また，異なるリソース URI の同値関係に対して，同値類リソースと同値類 RDF パスを定式化している．それにより，複数のリンクトデータを統合して，同値なリソース URI について透過的な推論を実現し，それが RDF パス検索にもたらす効果を実験した．

今後の課題としては，同値類リソースの統合処理の高速化が挙げられる．そのためには，RDF グラフ中の同値類リソースを効率的にまとめて処理できるようインデックスを再構築する方法が考えられる．さらに，リソース集合から高速に共通ノードを探索するアルゴリズムの設計を検討している．

参考文献

[1] Lehmann, Jens, et al. DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web – Interoperability, Usability, Applicability, Vol. 6, No. 2, pp. 167–195. IOS Press, 2015.

[2] Erxleben, Fredo, et al. Introducing Wikidata to the linked data web. In Proceedings of the 13th International Semantic Web Conference, pp. 50–65. Springer, 2014.

[3] Ora, Lassila, et al. (eds.) Resource Description Framework (RDF) model and syntax specification. W3C Recommendation, http://www.w3.org/TR/PR-rdf-syntax, 1999.

[4] 兼岩憲. RDF と RDF スキーマの推論. 人工知能学会論文誌, Vol. 26, No. 5, pp.473–481, 2011.

[5] セマンティック Web とリンクトデータ. 兼岩憲. コロナ社, 2017.

[6] Tran, Duc Thanh, et al. Top-k exploration of query candidates for efficient keyword search on graph-shaped (rdf) data. In Proceedings of the 25th International Conference on Data Engineering, pp. 405–416. IEEE, 2009.

[7] LIAN, Xiang, et al. k-nearest keyword search in RDF graphs. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 22, pp. 40–56, 2013.

[8] 浜松良樹, 兼岩憲. RDF グラフに対するキーワード検索の高速化と省メモリ化. 第 42 回セマンティックウェブとオントロジー研究会, 2017.

[9] Patel-Schneider, P. F., et al. (eds.) OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation, `https://www.w3.org/TR/owl-semantics`, 2004.

[10] Kato, Fumihiro, et al. Building DBpedia Japanese and linked data cloud in Japanese. In 2013 Linked Data in Practice Workshop, 2013.