Linked Open Data のための SPARQL クエリ共有システムの提案

Proposal of SPARQL Query Sharing System for Linked Open Data

濱崎 雅弘[1*]　後藤 真孝[1]
Masahiro Hamasaki[1]　Masataka Goto[1]

[1] 産業技術総合研究所
[1] National Institute of Advanced Industrial Science and Technology (AIST)

Abstract: We propose a SPARQL query sharing system for Linked Open Data. Recently, many and various datasets are published as a Linked Open Data (LOD). SPARQL is an RDF query language and it provides a powerful way to access LOD. However, it is not easy to utilize them because it requires not only techniques of SPARQL but also knowledge of datasets and vocabularies that they used for users. Therefore, we propose sharing SPARQL query as a solution of this problem. Sharing SPARQL query is a simple solution but it has an important role for LOD. In this paper, we describe a difficulty of utilizing LOD and introduce our prototype system for sharing SPARQL query.

1 はじめに

本稿では，Linked Open Data (LOD) のための SPARQL クエリ共有システムを提案する．LOD は様々な応用が期待されるが，膨大かつ多様なデータセットであるため，適切なクエリを作成するのは容易ではない．提案システムは，SPARQL クエリを共有し検索・推薦可能にすることで，一部の熟練ユーザが作成した SPARQL クエリの多くのユーザが利活用できるようにする．

本稿の構成は以下の通りである．まず 2 章にて，Linked Open Data の問題について，その原因と解決すべき課題について述べる．次に 3 章にて，提案システムの概要とプロトタイピングした Web アプリケーションを紹介する．4 章にて，SPARQL クエリ共有を実現するために不可欠な SPARQL クエリ間の類似度について議論する．5 章にて関連研究を述べ，6 章にて本稿をまとめる．

2 Linked Open Data 検索の課題

Linked Open Data (LOD) はオープンライセンスの元で公開された Linked Data である．Tim B. Lee は Linked Data を (1) 名前として URI を用いる，(2) 人々が HTTP で URI にアクセスできる，(3) URI にアクセスすると RDF で記述された有用な情報が提供される 1，(4) 他の URL が含まれており，さらなることを知ることができる，と定義している [1]．つまり Linked Data は機械可読な構造化データをアクセス可能にし，さらに，他のデータとリンクさせることが条件となる．これにより，そのデータ単体で，または，データ所有者ではできなかった価値創出が行われる可能性が生じる．特に他のデータとの組み合わせによる新たなサービスの実現（マッシュアップ）が注目されている．オープン化されたデータを Linked Data 化するプロジェクトも進められている．

LOD は機械可読なデータであるが，データが作られた出自を考えると，元のデータを Web データ化することが目的であり，ある特定のサービスで便利になるように作られたものではない．一般にデータベースは，あるサービスのために構造が定義（スキーマ設計）され，構造化データが蓄積される．そのため，サービスからのデータ利用は容易である．しかし LOD はデータ利用者ではなくデータ所有者（厳密にはデータそのもの）の都合に合わせて構造が定義され，構造化データが蓄積される．もちろん，様々な利用可能性が開かれているという点で，特定の利用者のためにデータ構造が設計されるのではなく，元データをより適切に共有するためにデータ構造が設計されることは正しい．しかしながらこの特徴ゆえに，Linked Open Data はアクセス可能・検索可能になっただけでは利活用は容易であるとはいえない．サービスにとって適した構造化がな

*連絡先：産業技術総合研究所
(〒 305-8568 茨城県つくば市梅園 1-1-1 中央第二)
E-mail: masahiro.hamasaki (at) aist.go.jp

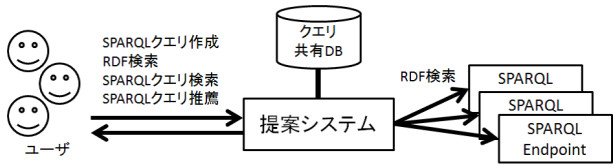

図 1: システム構成図

されているとは限らないため，ユーザはデータセットの構造をよく理解しクエリを工夫して作成しなくてはならない．

検索，とくに検索クエリの作成が困難な場合に，支援を行う技術は数多く提案されている．多くは検索対象となるデータをシステム側が把握することで，ユーザが目的とするクエリを予測し補完やサジェストを行うが，LOD はその性質上，検索対象となるデータをシステム側が把握することは困難である．また，クエリの補完やサジェストはユーザの目的を予測することで成立するが，検索ニーズは必ずしも予想可能なものだけではない．探索的検索 [2] のような，LOD の新しい利活用方法を発見するような検索行為を支援するには，予測というアプローチでは難しい．

そこで本研究では，SPARQL クエリをユーザ間で共有するアプローチを提案する．LOD のデータ構造が，データ所有者によって創発的に作られるように，それを利活用する SPARQL クエリもまた創発的に作ることを可能にする．SPARQL クエリはエンドユーザが自由に作成できるものであるが，ここでいう創発的に作るとは，ユーザ間で SPARQL クエリを共有することで，SPARQL クエリの再編集や派生クエリの作成によるクエリの洗練，拡張，発展を指す．次章では，SPARQL クエリの再編集や派生クエリの作成を可能にする，SPARQL クエリ共有システムについて述べる．

3 提案システム

3.1 システム概要

提案システムでは，ユーザ間で SPARQL クエリの共有を行う．システムは RDF データを持たず，他の SPARQL Endpoint への検索インタフェースとなる．システム構成を図 1 に示す．

ユーザがシステムに SPARQL クエリを入力すると，システムは SPARQL Endpoint に問い合わせ，得られた検索結果を表示する．ただしクエリの入力にあたっては，ユーザは SPARQL クエリ（とリクエストを投げる SPARQL Endpoint）だけでなく，クエリを説明するタイトル，説明文，タグ，コメントも入力する．システムは，*(a) SPARQL* **クエリ**に加えて，これらのユーザが入力した *(b)* **メタデータ**，さらに Endpoint から返ってきた *(c)* **検索結果**の合計三種類のデータをひとまとまりにしてデータベースに登録する．

図 2: クエリページの画面例

ユーザによって登録された SPARQL クエリは，**クエリページ**として他のユーザからも閲覧可能である．プロトタイプシステムのクエリページの例を図 **??**に示す．クエリページにはタイトルや説明文，タグなどのメタデータに加え，SPARQL クエリ本文とその検索結果が表示される．検索結果は毎回 SPARQL Endpoint に問い合わせているのではなく，クエリの更新や更新リクエストがない限りはキャッシュされたものが表示される．クエリによっては検索結果が膨大になるため，プロトタイプシステムでは SPARQL クエリに対して強制的に LIMIT 文を追加している．

3.2 SPARQL クエリの検索・推薦

ユーザは通常のキーワード検索によってクエリページを見つけることができる．クエリページ検索は，SPARQL クエリ，メタデータ，検索結果すべてに対する全文検索となる．よってタイトルや説明文，タグなどをキーワードで検索することもできるし，リソースの URI を入力して，それを用いたクエリやそれがヒットするクエリを検索することもできる．

ユーザが見つけたクエリは，ユーザ自身の目的に完全に一致しているとは限らない．提案システムではクエリの編集および派生クエリの作成が可能になってい

る．クエリの編集とは，現在閲覧中のクエリを直接編集することであり，派生クエリの作成とは，現在閲覧中のクエリをドラフトとして新しクエリを作成することである．プロトタイプではtwitter [1] が提供するOAuthによりユーザ認証を行い，認証済みユーザであれば誰でもクエリの作成，編集，派生クエリの作成が行える．

SPARQLクエリが誰でもアクセス可能な場所に置かれ，また，検索可能になったとしても，ユーザが目的のSPARQクエリに出会えるかどうかは確かではない．ユーザはクエリを検索するために必要なキーワードを知らないかもしれないし，そもそも探索的検索のように目的が不明瞭であるかもしれない．LODのような膨大かつ日々成長するデータセットに対しては，全データセットを把握できるのは原理的に不可能なため，このような曖昧な検索要求に対応することは重要であると考える．提案システムではSPARQLクエリ推薦機能によってこれを支援する．

SPARQLクエリ推薦機能とは，現在閲覧中のクエリページに対して関連するクエリページを推薦する機能である．プロトタイプシステムではメタデータによるつながりがあるクエリページと，派生関係によるつながりがあるクエリページを推薦している．どのようなクエリページ（SPARQLクエリ）が推薦されるべきかについては，次章にて議論する．

4 議論

提案システムではユーザ間でのSPARQLクエリ共有を実現するために，SPARQLクエリ推薦機能を持つ．本章では，どのようなSPARQLクエリが推薦されるとユーザにとって有用であるかについて議論する．

人気度に基づく推薦 ユーザ間でSPARQLクエリを共有することで得られるメリットの一つとして，クエリの人気度（利用頻度）が計算可能になることが挙げられる．より多くのユーザが閲覧したクエリ（クエリページ）は，典型性の高いお手本となるクエリであると考えられる．より多くの派生クエリが創られたクエリは，汎用性の高いテンプレート的なクエリであると考えられる．これらのクエリを推薦することは，特にSPARQLクエリの作成に不慣れなユーザに有用であると考えられる．

クエリの類似度に基づく推薦 ユーザが現在閲覧中のクエリに対して，SPARQLクエリ文そのものが類似するクエリを推薦する．情報推薦における基本的なアプローチであり，様々な手法が提案されている．これをSPARQLクエリ推薦に適用する場合，二種類の類似性が考えられる．

一つはクエリの言語レベルの類似性である．これは利用している関数や予約語の一致に基づく類似性である．この類似性を用いたクエリ推薦は，現在閲覧中のクエリが用いている関数や予約語の利用例を示すものといえ，特に初学者にとって有用であると考えられる．

もう一つはクエリで参照しているリソースの類似性である．日本語DBpediaの検索フォーム [2] で入力例として出ている「select distinct * where <http://ja.dbpedia.org/resource/東京都> ?p ?o . 」を例とすると,「<http://〜/東京都>」を利用しているクエリを推薦する．この類似性を用いたクエリ推薦は，現在閲覧中のクエリの拡張例を示すものといえる．例えば元クエリが東京都を出身地とするに人物名を集めていた場合,「東京都出身だが現在は東京都以外に住んでいる人」といった絞り込み条件を追加したクエリや,「東京都出身の人と結婚した人」といったRDFグラフを辿って異なるリソースを抽出してくるクエリなどが推薦される．これはユーザが興味のあるデータの周辺情報を提供していることになり，探索的検索にとって有用であると考えられる．

検索結果の類似度に基づく推薦 ユーザが現在閲覧中のクエリに対して，検索結果が類似するクエリを推薦する．これはつまり，違う言い方で同じものを得られているケースを推薦することになる．

クエリを作成するユーザからすると，クエリの改修案を示すものといえる．似たような結果を得られるが，クエリ本文がより簡潔であれば，より効率がよいクエリかもしれない．似ているがより多くの検索結果が得られるなら，網羅性が高いクエリとして参考になるかもしれない．

5 関連研究

5.1 セマンティックウェブ検索

セマンティックウェブおよびLinking Open Dataは複雑な構造を持つ膨大なデータの集まりである．SPARQLのような問い合わせ言語も開発されているが，事前にスキーマについての理解がなければ，適切なクエリを構築することができない．そこで，可視化やユーザインタフェースによってRDFデータの検索を支援する研究が多く存在する．

KieferらのiSPARQL [3] やRussellらのNITELIGHT [4] はグラフ図を描くことでSPARQLクエリを作成できる．直感的にSARQLクエリを作成できるが，事前に登録されたスキーマで定義された語彙をドラッグ＆ド

[1] http://twitter.com

[2] http://ja.dbpedia.org/sparql

ロップしてクエリ作成するため，様々な語彙が混じるLinking Open Dataでは問題が生じる．

Deligiannidisら [5] のPGVではRDFをグラフ図として可視化し，そのグラフを辿っていくことで目的のデータを発見する．このとき，膨大なリンクを持つノードがあるとデータが見にくくなるが，PGVでは着目した隣接ノードだけを固定して他の膨大な隣接ノードを動かしながら閲覧できる機能（Ferris-Wheel tequnique）や，Forwardリンクまたは Backwardリンクのみに限定して閲覧したりする機能を提供している．

Tummarelloら [6] のDeri Pipesは，RDFデータやSPARQLクエリを示すボックスを線でつないでいくことで，RDFデータのマッシュアップを容易に行うシステムである．Jarrarらの MashQLも同様のシステムであるが，クエリをツリー形式で記述することで，SAPQRLに関する知識無しでも利用できるのが特徴である．後藤らのDashSearchLD [7] も同じくSAPQRLに関する知識無しで利用できるLOD検索システムであり，検索結果ボックスをマウス操作で重ねることでAND検索を行い，さらにファセット検索を組み合わせることで対話的な検索を可能にする．

以上の研究はいずれも複雑なSPARQLクエリをどう作成するか，もしくは，SPARQLクエリを直接作成せずにSW/LOD検索を行うか，という点にフォーカスしている．これはつまりSPARQLクエリの作成が本質的に難しいことを示している．見方を変えると，これらの手法を用いて作成した複雑なクエリは，それ自体が有用なデータであるといえる．本稿で提案するアプローチは有用なデータであるクエリを共有するものであり，検索クエリ作成支援技術とは補完関係にある．

これらとは異なるLOD検索支援アプローチとして，Verborghらの Linked Data Fragment [8] がある．これは単純なSPARQLクエリと検索結果の一部をセットにしたフラグメントと呼ぶデータを事前に作成し，これを利用してSPARQLクエリを効率の良いものに書き換えることでエンドポイント側の負荷を軽減する．我々の提案システムでもSPARQLクエリと検索結果の一部を共有する仕組みを提供しており，Linked Data Fragmentのような拡張も可能であると考えられる．これは今後の課題である．

5.2 検索支援技術

検索を支援する技術は，セマンティックウェブ検索に限らず数多く提案されている．ここでは様々な検索支援技術を概観し，提案するアプローチとの違いを述べる．

クエリの修正を支援する技術は数多く提案されている．例えばテキストクエリの場合は，スペルミスを自動推定して修正したり [9][10]，別の語やフレーズが提案され，現在のクエリの代わりか追加する形で利用されたりする [11]．拡張していくことでより複雑なクエリの作成へと展開していくことも可能だが，実際には単純にクエリの置き換えとして利用されることが多い．

Query by Exampleは，例えば画像検索ならば画像 [12][13] を，楽曲検索なら音 [14] をクエリとして用いることで，言語情報に頼らない直感的な検索を可能にする．構造化データであるXQueryを入力すると同じXMLタグを含むXQueryを検索する手法 [15] も提案されている．検索対象コンテンツのタイプを選ばない強力な検索方法であるが，適切な例となるコンテンツを用意するのが難しい．

ファセット検索 [16][17][18] は利用可能なメタデータを逐次的に提示していくことで，事前知識のないユーザでもメタデータを利用した検索を可能にする．taしかし基本的にカテゴリやタグのようなメタデータを選択しながら検索するため，数値データや自由文を値として持つメタデータなどを利用するためには，やはり依然としてユーザがコンテンツやメタデータを熟知する必要がある．

適合性フィードバック (Relevance feedback) とは，入力されたクエリの検索結果を元に，クエリの修正を支援する技術である．具体的には追加クエリを提示したり，Query by Exampleとして使えるコンテンツを検索結果の中から見つけたりする [13]．クエリを修正するのではなく，リランキング [19] のように再検索ではなく検索結果の順序変更のためにフィードバックを利用する研究もある．適合性フィードバックはインタラクションによって複雑なクエリを作成する技術であるといえる．ただし，インタラクションの過程そのものがクエリとなるため，そのクエリを再利用したり他人と共有したりすることができない．

協調的検索 (Collaborative search) とは，複数人で協力して検索を行う仕組みである [20]．知り合いから適切なクエリを教えてもらうことは日常でもよくあることだが，それをオンライン上で行う仕組みである．グループ旅行のための情報収集など，同じ検索目的を持つメンバーからなるグループでの利用が想定されている [21]．これらは検索プロセスにおけるコミュニケーション支援が中心であり，複雑なクエリの作成を直接的に支援するものではない．また，同一の検索目的をもつ検索パートナーを持たない限り利用できないという問題もある．

以上のように，多くのクエリ作成支援技術は複雑なクエリを作ることを支援対象としておらず，また，クエリ作成が個人やグループで閉じている．ファセット検索やクエリ作成GUIは複雑なクエリの作成を可能にするが，事前知識のない一般ユーザにとっては難しい．適合性フィードバックは複雑な検索クエリをユーザに

見えない形で生成する．結果として，熟練ユーザは複雑なクエリを駆使して LOD を検索できるが，多くの一般ユーザはできない，という状況になる．本稿で提案するアプローチはユーザ間でクエリを共有することで複雑なクエリの生成・洗練させるという点で既存研究と異なっている．

6 おわりに

本稿では，Linked Open Data のための SPARQL クエリ共有システムを提案した．提案システムは，SPARQL クエリを共有し検索・推薦可能にすることで，ユーザが作成した SPARQL クエリの他の多くのユーザが利活用できるようにする．

Linked Open Data (LOD) は様々な応用が期待されるが，膨大かつ多様なデータセットであるため，適切なクエリを作成するのは容易ではない．しかも利用可能なデータセットは日々増えていくため，データセット全体を熟知するということは不可能である．よってユーザ全体でデータセット利用のための知識としての SPARQL クエリを共有することは LOD の利活用を推進するうえで強力な支援になると考える．

今後はプロトタイプシステムの開発を進め，Web サービスとして公開したい．実際に利用してもらうことで，ユーザがどのような SPARQL クエリを作成しているのか，どのような SPARQL クエリ推薦がユーザにとって有用であるのかを明らかにし，システムの改善を行っていきたい．

謝辞

本研究を行うにあたり議論していただいた国立情報学研究所 武田英明教授および LODAC Project のメンバーの皆様に感謝いたします．

参考文献

[1] Tim Bernares Lee. Linked data, 2009. http://www.w3.org/DesignIssues/LinkedData.html.

[2] Gray Marchionini. *Exploratory search: from finding to understanding*, Vol. 49 of *Communications of the ACM*, pp. 41–46. ACM, 2006.

[3] Christoph Kiefer, Abraham Bernstein, and Markus Stocker. The fundamentals of isparql: A virtual triple approach for similarity-based semantic web tasks. In *Proc. ISWC 2007*, 2007.

[4] A. Russell, P. R. Smart, D. Braines, and N. R. Shadbolt. Nitelight: A graphical tool for semantic query construction. In *Proc. SWUI 2008*, 2008.

[5] Leonidas Deligiannidis, Krys J. Kochut, and Amit P. Sheth. Rdf data exploration and visualization. In *Proc. CIMS '07*, pp. 39–46, 2007.

[6] Christian Morbidoni, Axel Polleres, and Giovanni Tummarello. Who the foaf knows alice? a needed step toward semantic web pipes. In *New Forms of Reasoning for the Semantic Web'07*, 2007.

[7] Takayuki Goto, Masahiro Hamasaki, and Hideaki Takeda. Dashsearch ld: Exploratory search for linked data. In *Proc. JIST 2012*, 2012.

[8] Ruben Verborgh, Miel Vander Sande, Pieter Colpaert, Sam Coppens, Erik Mannens, and Rik Van de Walle. Web-scale querying through linked data fragments. In *Proc. LODW 2014*, 2014.

[9] S. Cucerzan and E. Brill. Spelling correction as an iterative process that exploits the collective knowledge of web users. In *Proc. EMNLP 2004*, pp. 293–300, 2004.

[10] Mu Li, Muhua Zhu, Yang Zhang, and Ming Zhou. Exploring distributional similarity based models for query spelling correction. In *Proc. ACL 2006*, pp. 1025–1032, 2006.

[11] Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. Probabilistic query expansion using query logs. In *Proc. WWW 2002*, pp. 325–332, 2002.

[12] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Qian Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: the qbic system. *Computer*, Vol. 28, pp. 23–32, 1995.

[13] Yong Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 8, pp. 644–655, 1998.

[14] Asif Ghias, Jonathan Logan, David Chamberlin, and Brian C. Smith. Query by humming: musical information retrieval in an audio database. In *Proc. ACM Multimedia '95*, pp. 231–236, 1995.

[15] Daniele Braga, Alessandro Campi, and Stefano Ceri. Xqbe (xquery by example): A visual interface to the standard xml query language. *ACM Trans. Database Syst.*, Vol. 30, No. 2, pp. 398–443, 2005.

[16] Max L. Wilson, Paul Andre, and m.c. schraefel. Backward highlighting: enhancing faceted search. In *Proc. of UIST' 08*, pp. 235–238, 2008.

[17] m.c. schraefel, Max Wilson, Alistair Russell, and Daniel A. Smith. mspace: improving information access to multimedia domains with multimodal exploratory search. *Commun. ACM*, Vol. 49, No. 4, pp. 47–49, 2006.

[18] Marti A. Hearst. Clustering versus faceted categories for information exploration. *Commun. ACM*, Vol. 49, No. 4, pp. 59–61, 2006.

[19] 山本岳洋, 中村聡史, 田中克己. Rerank-by-example：編集操作の意図伝播によるウェブ検索結果のリランキング. 情報処理学会論文誌, Vol. 49, No. SIG7, pp. 16–28, 2008.

[20] Meredith Ringel Morris. A survey of collaborative web search practices. In *Proc. of CHI '08*, pp. 1657–1660, 2008.

[21] Gene Golovchinsky, Abdigani Diriye, and Jeremy Pickens. Designing for collaboration in information seeking. In *Proc. of HCIR '11*, 2011.