リンク構造解析を用いたLinked Open Dataに対するキーワード検索

奥村　彩水† 　天笠　俊之†† 　北川　博之††

† 筑波大学 システム情報工学研究科　〒305–8573 茨城県つくば市天王台 1–1–1
†† 筑波大学 システム情報系　〒305–8573 茨城県つくば市天王台 1–1–1
E-mail: †okumura@kde.cs.tsukuba.ac.jp, ††{amagasa,kitagawa}@cs.tsukuba.ac.jp

あらまし　Linked Open Data(LOD) という機械処理可能なデータを公開し共有する取り組みが普及している．LOD のデータは RDF という枠組みで記述され，データの問合せには SPARQL というクエリ言語を用いる．SPARQL を使用するには SPARQL 言語の習得と，LOD データ構造の理解が不可欠である．そこで，専門知識を必要とすることなく LOD データの検索を容易に行う方法として，キーワード検索を用いる．また，検索結果のランキングを行うにあたり，各ユーザの検索要求に合わせた結果を返したい．よって本研究では，適合フィードバックと，PageRank を拡張した ObjectRank を適用したキーワード検索手法について提案する．

キーワード　LOD，RDF，SPARQL，キーワード検索，ObjectRank，適合フィードバック

1. 序　　論

Linked Open Data（LOD）は，機械処理可能なデータを公開する取り組みである．この取り組みは企業や政府を中心に推進されている．データを公開して共有することでデータの二次利用など多様な運用が可能になる．LOD において構造化データの記述には，RDF（Resource Description Framework）(注1) が用いられる．RDF では，リソースの関係をトリプルと呼ばれる，主語，述語，目的語から構成される三つ組みの集合でグラフ構造を表現する．この RDF データに対して問合せを行うには，SPARQL(注2) というクエリ言語を用いる．

しかしながら，SPARQL クエリを記述するには SPARQL 言語についての学習が必要である．更に，問合せの対象となる LOD データの構造も理解していなければならない．LOD データのグラフ構造は一般に複雑であり，データ量も膨大であるため，特に後者は困難である．

この問題に対し，本研究では LOD に対するキーワード検索を提案する．キーワード検索を用いることで，LOD データに関する専門的な知識なしに検索を行うことが可能となる．また，検索結果のランキングを行うにあたり，各ユーザの検索要求に合わせたランキング結果を返したい．そこで，情報検索における手法の一つである適合フィードバックを用いる．本研究では更に，LOD データがグラフ構造であることを考慮し，ObjectRank [1] を利用したグラフ構造によるランキングと，それに対する適合フィードバックの適用についても議論する．

2. ObjectRank

ObjectRank は，データベース上のオブジェクトの重要度を評価するアルゴリズムである．代表的なリンク解析手法である PageRank を拡張した手法で，複数種類のノードやエッジを扱うことが可能である点が PageRank とは異なる．ObjectRank において，ノードとエッジはラベルの付与により種類が区別され，エッジは種類ごとに重みが付与される．

ObjectRank を用いてランク値を計算するにあたり，まず Authority Transfer Schema Graph（以下，Schema Graph とする）を構成する．これはノードおよびエッジの種類と，エッジの評価値を表す重みを定義したグラフである．続いて，Schema Graph に基づき，解析対象となる Authority Transfer Data Graph（以下，Data Graph とする）を構築する．Data Graph におけるエッジの重みは，そのエッジに付与されている重みをエッジの元ノードのもつ出次数で割った値となる．但し，その出次数は同種類のエッジに対して考えるものとする．

ノード v_i からノード v_j に対してエッジが存在する場合，a_{ij} にエッジ e_{ij} の重み w_{ij} を格納した遷移行列を $\boldsymbol{A}$ とする．このとき，ObjectRank による評価値 $\boldsymbol{r} = [r(v_1), \ldots, r(v_n)]^T$ は以下の式で求められる．

$$\boldsymbol{r} = d\boldsymbol{A}\boldsymbol{r} + \frac{1-d}{|V|}\boldsymbol{e}$$

ここで，d はダンピングファクタ，$\boldsymbol{e}$ は全ての要素が 1 の n 次元列ベクトルである．実際は，上記で求めた global ObjectRank に加えて検索キーワードを用いる keyword-specific ObjectRank を求め，両者を重み付き統合してスコア値を導出する．

3. 関 連 研 究

3.1 RDF データに対するキーワード検索

LOD や RDF のデータに対してキーワード検索を行う研究として，Lei ら [2] や Dass ら [10] によるキーワードの入力形式を指定されているものや，Pound ら [3] による入力にキーワードとクエリタイプの双方を用いて結果を返すものがある．しかしながら，これらの手法では検索結果がトリプル形式で出力され，データが部分的であり得られる情報としても限定されてしまう．

検索結果としてトリプルを拡張した形式で出力する手法と

(注1)：http://www.w3.org/TR/rdf11-concepts/
(注2)：http://www.w3.org/TR/rdf-sparql-query/

しては，Sinha ら [4] の研究がある．この研究では，RDF データにおいてドキュメントを作成し，そのドキュメントに対する検索システムを提案している．ドキュメントの構成としては，RDF のトリプルにおける主語がタイトル，述語と目的語のセットがコンテンツとなっている．但し，検索結果のランキングに関しては実装されていない．本研究では，この研究におけるドキュメントの構成をベースとしたキーワード検索を実装し，更に検索結果に対して LOD のグラフ構造を考慮したランキングを実装する．

3.2 LOD に対するランキング

LOD といったセマンティックウェブに対するランキングを行う研究として，一瀬ら [5] は DBpedia に対して PageRank を用いる手法を提案している．トリプルを対象とし，主語と目的語をノード，述語をエッジとして PageRank を適用している．しかしながら，PageRank は一種類のエッジやノードしか扱えず，同一エッジ内のみでしかランキングすることができない．

Mulay ら [6] の研究では，LOD におけるデータセット間の owl:sameAs リンクを考慮したランキングを提案している．データセット，リソース，トリプルのそれぞれのレイヤーにおいてスコアリングを行う手法である．この手法では，リソースを評価するにあたりデータセット間のリンク情報しか考慮しておらず，データセット内の関係性が無視されてしまう．

また，Pound ら [3] の TF-IDF を用いてランキングを行う研究がある．この手法では，同一エッジをもつリソースに対して IDF 値を計算する．つまり，RDF グラフの構造は考慮されておらず，データにおける重要なリンクがランキングに反映されない．

4. 提 案 手 法

提案手法の概要は以下の通りである．なお，基本的な手法は，既に提案した手法 [7] に基づいている．まず，検索に先立ち検索対象の LOD データからドキュメントの抽出を行う．抽出されたドキュメントから，ObjectRank を適用するために，Schema Graph の抽出及びそれに基づく Data Graph の抽出を行う．検索処理においては，ユーザから与えられた検索キーワードをもとに結果ドキュメントを検索し，ObjectRank 値によってランキングを行う．更に，得られた結果に対してユーザが適合，不適合の判定を行うことにより，適合フィードバック [8] を行い，検索結果を再計算する．

4.1 キーワード検索

LOD（RDF）データはトリプル $t = (s, p, o)$ の集合 T からなる．ここで s, p, o はそれぞれ主語，述語，目的語である．T に対するキーワード検索とは，ユーザによって与えられた検索キーワード集合 $K = \{k_1, k_2, \ldots, k_m\}$ に対し，適合するエンティティをランク付きで返す処理である．

まず，LOD データにおける検索対象となるエンティティを与える．一般に LOD データには，多様なエンティティが含まれる．本手法では，システム構築者等によって事前に検索対象となるエンティティが指定されているものとする．それには，クラスによる指定，SPARQL 問合せによる指定等を利用する．

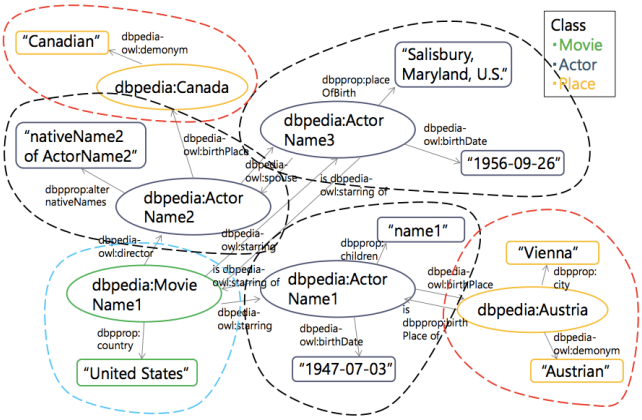

図 1　エンティティサブグラフの例

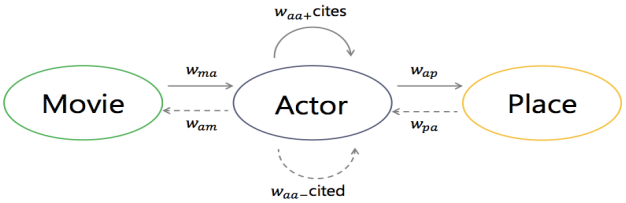

図 2　Schema Graph

検索対象となるエンティティが決まると，次に各エンティティに対応するサブグラフの抽出および特徴抽出を行う．まず，各検索対象エンティティの URI u_i を主語として持ち，リテラルを目的語に持つトリプル $(u_i, *, l_j)$ を探索する．これを**エンティティサブグラフ**と呼ぶこととする．次に，1) 各エンティティ（に対応する URI）u_i 中に出現する単語，および，2) リテラル l_j から単語抽出を行い，これを u_i の特徴語とする．本研究では，このエンティティサブグラフを対象としてキーワード検索を行う．

図 1を用いてエンティティサブグラフを説明する．この例では，`Movie`，`Actor` および `Place` が検索対象クラスとして指定される．その結果，`Movie` は一つ（水色点線内），`Actor` は三つ（黒点線内），`Place` は二つ（赤点線内）のエンティティサブグラフが抽出される．

4.2 Schema Graph の作成

ObjectRank 値の計算には，ノードおよび（重み付き）エッジの種類を規定した Schema Graph を作成する必要がある．このため，検索対象クラスをそれぞれ Schema Graph におけるノードとする．更に，LOD データの中にそのクラス間にエッジがある（トリプルが存在する）場合，Schema Graph におけるノード間に対応するエッジを作成する．この際，エッジの重みは，均等な値を初期値とし，後で説明する適合フィードバックによって動的に調整する．

図 2に図 1の場合の Schema Graph を示す．この例では，`Movie`，`Actor` および `Place` に対応するノードが作成され，各ノード間に対応するエッジが作成される．なお，`Actor` 同士の参照は自己ループとして表現される．また，ObjectRank の定

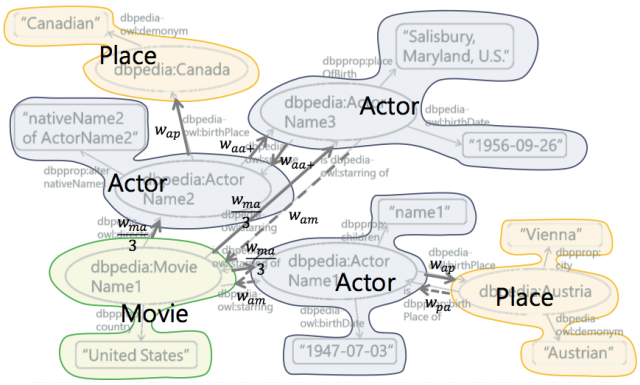

図 3　Data Graph

義に従い，片方向のエッジには必ず逆向きのエッジが作成される．

4.3　Data Graph の作成

Data Graph は，Schema Graph およびエンティティサブグラフを元に作成する．具体的には，エンティティサブグラフの各エンティティをノード集合として，LOD データから誘導される誘導部分グラフを用いる．このとき，各エッジの重みは，ObjectRank のアルゴリズムに従って Schema Graph 中のエッジの重みを元に算出される．

図 3に図 1から導出される Data Graph の例を示す．Data Graph におけるエッジの重みは，そのエッジに付与されている重みを，エッジの元ノードの持つ出次数で割った値となる．図 2 における `Movie` から `Actor` へのエッジを例に挙げる．`Movie` にあたる `dbpedia:Movie Name1` を含むエンティティサブグラフから，`Actor` にあたる `dbpedia:Actor Name1`，`dbpedia:Actor Name2` および `dbpedia:Actor Name3` の三つのエンティティサブグラフに対してエッジが張られている．よって，この三本のエッジの重みはそれぞれ $\frac{w_{ma}}{3}$ となる．

得られた Data Graph を基に作成した遷移確率行列 A を用いて各エンティティに対する評価値を計算し，ランキング結果を得る．

4.4　検索処理及び適合フィードバック

検索処理において，ユーザは検索キーワードを与え，システムはそれに基づきキーワードに適合するドキュメントを ObjectRank 値に基づいてユーザへ返却する．適合フィードバックとは，ユーザからのフィードバックを基に検索結果を改善していくアルゴリズムである．結果のうち適合しているものを用いて検索性能を上げる．適合フィードバックを行う方法の一つとして，Rocchio アルゴリズム [9] が有名である．これは，情報検索の代表的なモデルの一つであるベクトル空間モデルにおいて，適合情報を用いてクエリを修正していく手法である．ベクトル空間モデルでは，文書とクエリを単語の重み付きベクトルとして表す．ユーザの適合性判定から，クエリの単語ベクトルの重みを修正し，新しいクエリを用いて再検索を繰り返して検索結果を向上させるのである．クエリベクトル q の更新式は以下の通りである．

$$\boldsymbol{q_{n+1}} = \alpha \boldsymbol{q_n} + \beta \frac{1}{|D_r|} \sum_{\boldsymbol{d_j} \in D_r} \boldsymbol{d_j} - \gamma \frac{1}{|D_{nr}|} \sum_{\boldsymbol{d_j} \in D_{nr}} \boldsymbol{d_j}$$

ここで，α，β，γ は重み付け係数で，$|D_r|$ と $|D_{nr}|$ はそれぞれ適合文書数と不適合文書数である．本研究では，問合せの更新にこの手法を適用する．

本研究では，ObjectRank によるランキングを行っている．このため，ユーザからのフィードバックを利用した ObjectRank の改善についても検討する．具体的には，Schema Graph におけるエッジの重みを，ユーザからの適合・不適合判定に基づき調整する．基本的な考え方としては，各検索結果は，いずれか，または複数のクラスに所属している点に着目し，ユーザから適合（不適合）と判定されたクラスについて，Schema Graph における当該クラスへの入力辺の重みをより高く（低く）設定する．これによって，ユーザが適合（不適合）と判定したクラスのランキングをより高く（低く）することができると考えられる．具体的な更新式は以下の通りである．

$$w' = w * \alpha^m * (1/\beta)^n$$

ここで，w，w' は更新前，更新後の重み．α，β はそれぞれ適合数，不適合数に対する係数．そして m，n は適合数，不適合数である．

5.　予 備 実 験

提案手法であるエンティティベースでの ObjectRank によるランキング結果の検索キーワードに対する妥当性について，実際の LOD データに対して手法を適用することで検証を行った．本節ではその結果について述べる．

5.1　データセット

使用したデータは，Movie クラスのエンティティサブグラフ 949 件，Actor クラス 4,235 件，Place クラス 2,066 件の計 7,260 件である．データセットは，DBpedia が公開しているデータセットのうち，最新版であるバージョン 3.9 から，部分的に抽出したデータを使用した．Schema Graph，Data Graph は図 2，図 3と同様である．但し，Schema Graph(図 2) において，今回 `Actor` クラスが `Actor` クラスから参照されるようなエッジは存在しなかったため，w_{aa-} の値は 0 とする．

実験は，ランキング結果の上位 10 件を対象とし，以下二点に着目して行う．

- ObjectRank の有効性
- 適合フィードバックの有効性

検索キーワードには，キーワード 1「`godzilla`」，キーワード 2「`star wars`」を入力した．検索キーワードは，`Movie` クラスに含まれる単語である．また，重みの初期値として 0.33 を設定した．これは，今回の Schema Graph における一つのノードから出次するエッジの最大数が 3 であることによる．設定した重み w_0 を図 4に示す．

5.2　ObjectRank の有効性

ここでは，ObjectRank のリンク構造解析によるランキング結果が，有効であるか検証する．ユーザがキーワード 1 に関す

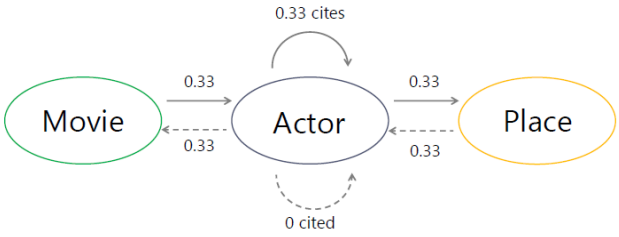

図 4　初期値の重み w_0

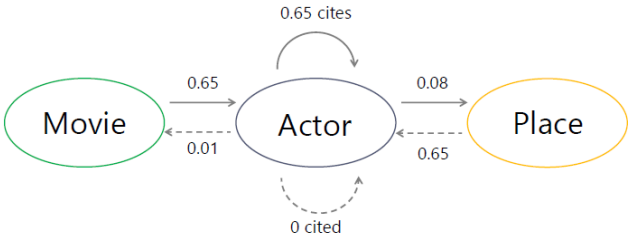

図 5　適合フィードバック適用後の重み w

るデータを検索したいと仮定する．このとき，ObjectRank によるランキング結果は図 6のようになった．ここで，エンティティラベルにおいて（M）=Movie クラス，（A）=Actor クラス，（P）=Place クラスである．図 6から，ランキング上位 10 件において全てのエンティティがキーワードと関連していることがわかる．キーワード 2 の場合では 10 件中 8 件がキーワードと関連していた．よって，ObjectRank によるランキングは有効であることがわかった．

順位	エンティティラベル	ObjectRank値	関連性
1	Godzilla: Final Wars(M)	4.4970e-05	○
2	Godzilla (1998 film) (M)	2.7969e-05	○
3	Godzilla, King of the Monsters! (M)	2.2013e-05	○
4	Japan (P)	1.8974e-05	○
5	Masami Nagasawa (A)	8.6844e-06	○
6	Hank Azaria (A)	8.5381e-06	○
7	Rei Kikukawa (A)	8.5269e-06	○
8	Roland Emmerich (A)	8.4663e-06	○
9	Don Frye (A)	8.4204e-06	○
10	Takashi Shimura (A)	8.2821e-06	○

図 6　ランキングの関連性

5.3　適合フィードバックの有効性

次に，上記で得られたランキング結果に，ユーザによる適合フィードバックを適用した結果の有効性について検証する．キーワード 1 を入力した際の ObjectRank によるランキング結果とそれに対するユーザの適合判定は図 7のようになった．ここで，ユーザは Actor クラスを検索しているものとする．図 8から，上位 4 件が不適合と判定されていることがわかる．

この適合情報から，重みの更新を行った．更新後の重みを図 5に示す．適合フィードバックを受けて，Movie クラスに向かうエッジが 0.33 から 0.01 に，Actor クラスに向かうエッジが 0.33 から 0.65 に Place クラスに向かうエッジが 0.33 から 0.08 に更新された．

順位	エンティティラベル	ObjectRank値	適合性
1	Godzilla: Final Wars(M)	4.4970e-05	×
2	Godzilla (1998 film) (M)	2.7969e-05	×
3	Godzilla, King of the Monsters! (M)	2.2013e-05	×
4	Japan (P)	1.8974e-05	×
5	Masami Nagasawa (A)	8.6844e-06	○
6	Hank Azaria (A)	8.5381e-06	○
7	Rei Kikukawa (A)	8.5269e-06	○
8	Roland Emmerich (A)	8.4663e-06	○
9	Don Frye (A)	8.4204e-06	○
10	Takashi Shimura (A)	8.2821e-06	○

図 7　ランキングの適合性

更新された重みを用いて再度 ObjectRank によるランキングを行った結果を図 8に示す．適合フィードバックを適用する前と後で，ユーザの適合判定が改善されていることがわかる．以上から，適合フィードバックによりユーザの意図する検索結果が，上位にランキングされるようになったことがわかった．これは，キーワード 2 を入力した際にも同等の結果が得られた．

	初期値			適合フィードバック適用後		
順位	エンティティラベル	ObjectRank値	適合性	エンティティラベル	ObjectRank値	適合性
1	Godzilla: Final Wars(M)	4.4970e-05	×	Roland Emmerich (A)	8.2175e-06	○
2	Godzilla (1998 film) (M)	2.7969e-05	×	Masami Nagasawa (A)	8.1619e-06	○
3	Godzilla, King of the Monsters! (M)	2.2013e-05	×	Hank Azaria (A)	8.1523e-06	○
4	Japan (P)	1.8974e-05	×	Godzilla: Final Wars (M)	8.0453e-06	×
5	Masami Nagasawa (A)	8.6844e-06	○	Don Frye (A)	8.0430e-06	○
6	Hank Azaria (A)	8.5381e-06	○	Takashi Shimura (A)	8.0075e-06	○
7	Rei Kikukawa (A)	8.5269e-06	○	Rei Kikukawa (A)	7.9217e-06	○
8	Roland Emmerich (A)	8.4663e-06	○	Raymond Burr (A)	7.8673e-06	○
9	Don Frye (A)	8.4204e-06	○	Chihiro Otsuka (A)	7.7999e-06	○
10	Takashi Shimura (A)	8.2821e-06	○	David Arnold (A)	7.7955e-06	○

図 8　適合フィードバック適用後のランキング

6.　結　　論

本研究では，LOD 上のデータに対し，ドキュメントを対象としたキーワード検索によって問合せを行う手法を提案した．また，キーワード検索を行う中で，ObjectRank と適合フィードバックを用いて検索結果をランキングした．LOD データの特徴であるリソースの多様性に適した，複数種類のノードやエッジを扱うことのできる ObjectRank を用いることで，既存手法に比べ高度なランキングを実現した．また，ObjectRank と適合フィードバックを兼ね合わせることで，ユーザの検索要求に合ったランキング結果を得られることがわかった．

今後の課題として，システムの実装を行うことと，より多いデータ数での実験を行うことが挙げられる．

7.　謝　　辞

本研究の一部は，共同研究費（富士通研究所 CPE27151），文科省“実社会ビックデータ利活用のためのデータ統合・解析技術の研究開発”，および，科研費（25240014）による．

文　　献

[1] Balmin, A. Hristidis, V. Papakonstantinou, and Y. ObjectRank:Authority-based keyword search in databases.

VLDB 2004.
[2] Yuangui Lei, Victoria Uren, and Enrico Motta. Semsearch:A Search Engine for the Semantic Web. EKAW 2006.
[3] Jeffrey Pound, Peter Mika, and Hugo Zaragoza. Ad-hoc Object Retrieval in the Web of Data. WWW 2010.
[4] Vineet Sinha and David R.Karger. Magnet: Supporting Navigation in Semistructured Data Environments. SIGMOD 2005.
[5] 一瀬詩織, 小林一郎, 岩爪道昭, 田中康司. DBpedia における SPARQL 検索結果のランキング手法. JSAI 2013.
[6] Kunal Mulay and P.Sreenivasa Kumar. SPRING: Ranking the results of SPARQL queries on Linked Data. COMAD 2011.
[7] 奥村彩水, 天笠俊之, 北川博之. Linked Open Data におけるグラフ構造を考慮したキーワード検索.
[8] R.B. Yates and B.R. Neto. Modern Information Retrieval. Addison Wesley 1999.
[9] J.J. Rocchio. Relevance feedback in information retrieval. 313-323, The Smart system - experiments in automatic document processing, Prentice Hall Inc.
[10] Ananya Dass, Aggeliki Dimitriou, Cem Aksoy, and Dimitri Theodoratos. Incorporating Cohesiveness into Keyword Search on Linked Data. WISE 2015.