

RDF リソースのリンク構造を考慮した SPARQL クエリ検索結果のランキング手法

一瀬 詩織^{†,a} 小林 一郎^{†,a} 岩爪 道昭^{‡,b} 田中 康司^{‡,b}

[†]お茶の水女子大学大学院 人間文化創成科学研究科 理学専攻 情報科学コース

[‡]独立行政法人 情報通信研究機構

a) { *ichinose.shiori, koba* }@*is.ocha.ac.jp* b) { *iwazume, tanaka* }@*nict.go.jp*

概要 近年, Web 上の共有資源として Linked Open Data(LOD) の公開が活発に行われ, Web 上に大きな Semantic Web データベースが形成されつつある. これに伴い, それらのデータベースから様々な目的に応じてデータを抽出するための手法が必要になってきている. 本研究では Semantic Web データベースからのデータ抽出手法として, データセット内のプロパティの出現頻度やトリプルの結びつきによるリンク情報を用い, SPARQL 問い合わせの取得結果をランキングする手法を提案する. 提案手法の評価を行うために 2 つのケースの SPARQL クエリ使用目的を設定し, LOD の中心的なデータセットである DBpedia を用いてそれぞれのケースに応じたクエリ検索結果のランキングを行った. またその結果を被験者実験により従来手法と比較することで, 本手法の有効性を示した.

キーワード DBpedia, SPARQL, Ranking, PageRank algorithm, Semantic Web

1 はじめに

近年の活発なオープンデータ活動により, Web 上に RDF 形式などで構造化されたデータベースが増加してきている. 例えば Wikipedia の構造情報を RDF 化した DBpedia[1] や, コンピュータサイエンスの主要な書誌をデータベース化した DBLP Bibliography Database¹, 政府の統計情報などを公開している Data.gov² など, 様々なデータが Semantic Web の構造化データとして公開されている. これらのデータはブラウジングや Swoogle[2] などの Semantic Web データのための検索エンジン, または SPARQL という問い合わせ言語を用いることで取得できる.

SPARQL クエリを用いた問い合わせでは RDF データセットに対し主語, 述語, 目的語からなるトリプルを最小単位とした条件を指定し, グラフパターンが合致したデータを取得する. この検索では構造化データからの単純な検索を目的としており取得した結果の順位付けは行われない. そのため条件と一致したデータが多量にあった場合, その中でどれが重要なものなのかは分からないという問題点がある.

本論文ではデータセット内のリンク構造やプロパティ出現頻度といったメタ情報, 検索で用いられる SPARQL クエリの構造情報を利用して, SPARQL 問い合わせで得られた検索結果をクエリに対する重要度によりランク付けする手法を提案する. 具体的には SPARQL の使用ケースとして「RDF トリプル (情報) を取得する場合」

「条件と一致したリソースを取得する場合」を想定し, 各ケースのクエリについて問い合わせ結果のランキングと被験者実験による有用性の評価を行った. 実験データには Linked Open Data で中心的な役割を果たし, 様々なリソースを持つ DBpedia のデータセットを用いた.

2 関連研究

本手法と同様に, Semantic Web リソースのランキングを行うことを目的とした手法は数多く提案されている. 特に PageRank[3] や HITS[4] といった Web ページの重み付けを行うアルゴリズムを応用して, データの構造情報からリソースの重み付けを行う試みには様々なものが存在する. Swoogle[2], ReConRank[5], ObjectRank[6] では PageRank アルゴリズムを Semantic Web データなどの構造化されたデータに適用した場合のアルゴリズムを提案している. Beagle[7] は ObjectRank のアルゴリズムを用いたローカルデータの重み付けを利用し, デスクトップ検索を行うデータ検索エンジンである. また SemRank[8] では, プロパティによるリソース同士の結びつきを利用した, Relation ベースの PageRank アルゴリズムを提案している.

特に本研究と同じく Semantic Web データセットを対象にクエリの問い合わせ結果をランキングする手法には, Bamba ら [9], Mulay ら [10] の研究がある.

Bamba らの研究 [9] では HITS アルゴリズム [4] を参考にしたアルゴリズムを用いてリソースの重み付けを行い, さらにクエリ検索結果のグラフ構造を用いて結果のスコアを計算する手法を提案している. Mulay らの研究 [10] ではデータセット, リソース, トリプルの三層構造

を用いたリンク構造の解析により、リソースの重み付けを行っている。ただし、利用している情報が SameAs リンクと関連した情報に限定されており、それ以外の様々なプロパティ情報は考慮されていない。

本研究では Bamba らの Semantic Web クエリ検索結果のランキングアルゴリズムを参考に、データベース内のリンク情報からリソースやプロパティの重み付けを行い、SPARQL クエリ検索結果のランキングを行う手法を提案する。

3 提案手法のフレームワーク

3.1 SPARQL の使用

SPARQL は RDF グラフからデータを抽出するためのクエリ言語である。SPARQL1.0 では SELECT, ASK, CONSTRUCT, DESCRIBE の 4 種類のクエリ形式を定めているが、本論文では、リソースを検索する際に一般に用いられる SELECT を用いたクエリ形式について取り扱う。SELECT 構文は SELECT 節と WHERE 節からなり、トリプルを最小単位としたグラフパターンの検索を行う。SELECT 節にはリソースやプロパティ等の取得したい要素を表す変数が入る。また、WHERE 節にはその要素が満たすべき複数の条件が記述される。

この研究では、SPARQL の使用目的として以下の 2 つの場合を想定した。

ケース 1. リソースから情報 (トリプル) を求めたい場合

例えばリソース ‘Tokyo’³ の情報を取得するなど、あるリソースを主語とするトリプルを取得する。図 2 のようにこの問い合わせでは、ある主語に対する複数の情報が検索される。

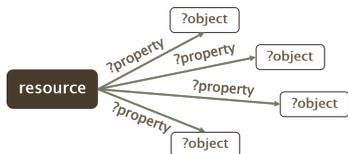


図 1 Search RDF triples from a resource

図 2 リソースからの RDF トリプルの検索

このときクエリは、?property と ?object を SELECT 節に取る、以下のような形式となる。

```
SELECT ?property ?object WHERE{
  <resource> ?property ?object .
}
```

ケース 2. 情報 (トリプル) と一致したリソースを求めたい場合

例えばクラス ‘dbo:City’⁴ に属するリソースなど、ある情報 (トリプル) と一致するリソースを取得する。図 3 のようにこの問い合わせでは、トリプルはリソースを絞り込むための条件として複数個設定される場合がある。

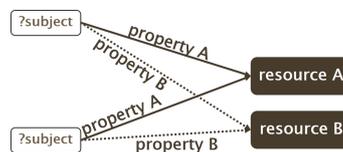


図 3 RDF トリプルからのリソースの検索

このときクエリは、?subject を SELECT 節に取った以下のような形式となる。

```
SELECT ?subject WHERE{
  ?subject <property> <object> .
}
```

プロパティと目的語はどちらも指定する場合もあれば、どちらか片方のみを指定する場合も考えられる。例えば目的語にアメリカの DBpedia 上のリソースである “http://dbpedia.org/resource/United_States” を取った場合、アメリカで生まれた人物やアメリカで出版された本など、様々な要素でアメリカと関係したリソースを取得することができる。

これらは SPARQL から情報を取得する際の一般的な使用ケースであり、その問い合わせを満足させる検索結果を得ることが SPARQL クエリのランキングには必要である。本研究では Bamba らの研究 [9] の評価アルゴリズムを参考に、クエリ検索結果を順位付けするための 2 つの手法を提案する。

3.2 プロパティ・リソースの重要度評価

本手法では検索結果の評価に、述語であるプロパティの重要度と、主語・目的語となるリソースの重要度の値とを用いる。本節ではこれらの重み付け指標の定義を行う。

3.2.1 グラフの定義

RDF データセット内のクラスとリソースをノード、リソース間のプロパティをエッジとし、各要素の重要度評価に用いる有向グラフ G, G' を定義する。

RDF データセットで定義されているすべてのトリプルの集合を T 、クラスとリソース、リテラルの集合を U 、プロパティの集合を P としたとき、データセット全体のグラフ G は $G = \langle U, P \rangle$ と定義される。また U からリテラルを除いた集合を R とし、データセットの

³‘.’ は http://dbpedia.org/resource/ の接頭辞である。

⁴‘dbo:’ は http://dbpedia.org/ontology/ の接頭辞である。

リソースグラフ G' を $G' = \langle R, P \rangle$ と定める。データセット内のトリプル t は $(s, p, o) \in T$ のように表す。ここで s は主語、 p は述語 (プロパティ)、 o は目的語である。例として $(u_1, p, u_2) \in T$ は主語が $u_1 \in U$ 、プロパティが $p \in P$ 、目的語が $u_2 \in U$ のトリプルを表す。グラフ G は、RDF データセット内に含まれるリソースとリソース間の関係のみによって構成されたグラフであり、RDF データセット外部のリソースとの関係は含まれていない。

3.2.2 プロパティの重要度評価

クエリ結果のランキングに際し、プロパティは極めて重要な意味を持つ。例えばそのリソースが作家であれば、作家の生まれた場所を意味する 'dbo:birthPlace' や影響を受けた人物を意味する 'dbo:influencedBy' などは重要な情報である。また大学であれば、設立年を意味する 'dbprop:established'⁵ やキャンパスのある場所を意味する 'dbprop:campus' などが主要な情報となると考えられる。このようにプロパティは、そのトリプルの主語が持つ情報が重要であるかないかの判断を可能にさせる役割を持つ。プロパティの重要度は主語になるリソースの属する 'dbo:Writer' や、'dbo:University' といったクラスに基づき決定されると考えられる。このようなリソース r が作家である、または大学である、などといったクラスの情報は、トリプル $(r, \text{rdf:type}^6, C)$ の目的語となるクラス C によって決定される。

このようなプロパティは特定のクラスに属するリソースを主語としたトリプルに多く出現する一方、データセット全体では出現頻度が低いものであると考えられる。あるリソースにとって特徴的であるプロパティをより高く評価するため、プロパティの評価手法として文書・単語の重要度評価によく用いられる TFIDF (Term Frequency, Inverted Document Frequency)[11] の概念に基づいた PFIPF (Property Frequency, Inverted Property Frequency) を定義した。

- *Property Frequency (PF)*

クラス内のすべてのトリプル数に対する、そのクラスに属するリソースについてプロパティが出現したトリプル数の割合を示す。これにより、そのクラスに属するプロパティの内、出現回数の多いプロパティをより高く評価することになる。あるリソースの属するクラスがデータセット内で定義されていない場合には、そのリソースはクラス 'owl:Thing'⁷ に属するものとして扱う。

- *Inverted Property Frequency (IPF)*

全クラス数に対してそのプロパティが出現したクラス数の割合の逆数に対数をとったものを示す。これは文書検索における IDF と同じく、一般的なプロパティに対するフィルタとして働く。多くのクラスで出現するプロパティは一般性が高いため、重要度が低くなる。

3.2.3 リソースの重要度評価

リソースの重要度評価には Web データの重要度評価によく用いられている PageRank アルゴリズムを用いた。DBpedia 上のすべてのリソース数を $|R|$ 、あるリソース $r \in R$ へ $(x, p, r) \in T$ のエッジを持つリソース x の集合を B_r 、リソース x から出るエッジの本数を c_x とし、以下の計算式に基づいた PageRank 値の計算を行った。

$$PR_r = \frac{1-d}{|R|} + d \sum_{x \in B_r} \frac{PR_x}{c_x} \quad (1)$$

PageRank アルゴリズムは一般的に有向グラフの重み付け指標としてよく用いられており、関連研究の多くの RDF 評価アルゴリズムでも基本的なとして用いられている。

リソースグラフに対してべき乗法を用いた PageRank 値の計算を行い、収束判定を PageRank 値ベクトルの差分が閾値より小さくなることを条件に収束判定を行った。安定した PageRank 値を得るため収束条件 $|PR_r^k - PR_r^{k-1}| < E^{-X}$ をおおよその収束状態を推測し、それぞれ $X = 5, 6, \dots, 14$ に設定して計算を行い、 $X = n$ と $X = n+1$ の場合における、PageRank 値によるリソースの順位変動を調べた。結果を図 4 に示す。

図 4 より、順位変動するリソース数は X の値が増加するにつれて大きく減少し、 $X = 10$ 付近では殆ど順位が安定していることが分かる。 $X = 10$ よりも値が大きい場合においては順位変動するリソース数は殆ど変化せず、むしろ 1000 位以上の大きな順位変動をするリソースの増加が見られたことから、 $X = 10$ の場合の PageRank 値を最も安定した値としてランキングに用いることとした。

3.3 ランキングアルゴリズム

この節では、本論文において提案する SPARQL クエリ検索結果のランキングアルゴリズムについて説明を行う。

SPARQL クエリ検索結果は、それぞれクエリの WHERE 節で示された条件に従い、リソースグラフ G の部分グラフを構成する。例えば図 3 のようにトリプルの条件を 2 つ指定してリソースを検索する場合、図 5 のように

⁵'dbprop:' は <http://dbpedia.org/property/> の接頭辞である。

⁶'rdf:' は http://www.w3.org/1999/02/22-rdf-syntax-ns# の接頭辞である。

⁷'owl:' は http://www.w3.org/2002/07/owl# の接頭辞である。

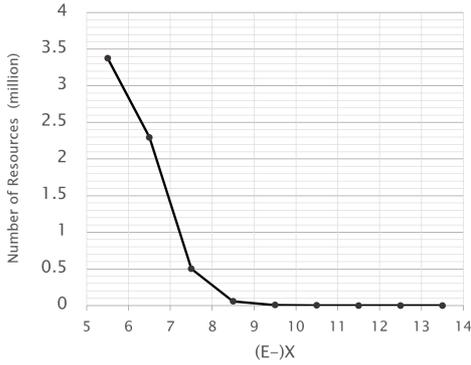


図4 X の値によるリソースの順位変動

個々の結果にはそれに対応する検索結果グラフが構成される。

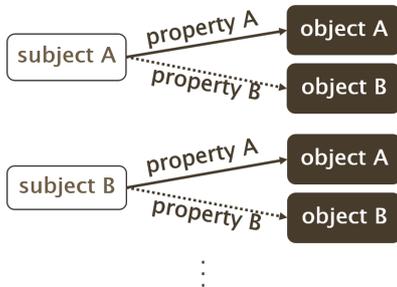


図5 リソース検索による検索結果グラフ

本研究では Bamba ら [9] のクエリ評価アルゴリズムを参考にし、基本概念として、検索結果の重要性を決める要素を以下のように定めた。

- リソース、プロパティの重要度: リソースやプロパティの重要度が高い場合、重要度の低い場合よりも結果の重要度は高くなる。
- グラフの大きさ: 検索結果グラフが大きい場合、グラフが小さい場合よりもクエリ条件に合った情報が多く含まれるため、検索クエリとの一致度は大きいと考えられる。
- ユーザの興味の強さ: SELECT 節で指定された変数は、ユーザが最も興味を持つアイテムである。逆にそれらのアイテム(リソースやプロパティ)からグラフ上で遠いアイテムに関しては、その距離に応じてユーザの関心は薄れていくと考えられる。

3.3.1 提案手法1: リソースとプロパティ重要度によるクエリの評価

先に定めた重要度の考え方から、評価アルゴリズムを以下のように定めた。decay はユーザの興味の強さに応じ、ノードやエッジの重みを調整するための変数である。

また scoreNode, scoreEdge はそれぞれノードとエッジのスコアを表す。

アルゴリズム 1:

1. decay = 1.0, scoreEdge = 0.0, scoreNode = 0.0 に初期化する。
2. Adj を SELECT 節で選択されたノードの集合で初期化する。この集合はユーザの興味の対象を表す。
3. Adj が空になるまで以下を繰り返す:
 - (a) ClassedEdges を Adj のノードのクラスと、ノードから伸びたエッジの集合とし、(c, e) で表す。
 - (b) $scoreNode(r) += \sum_{n \in Adj} Imp[n] \times decay$ ここで、Imp[n] はノード n の重要度を表す。(もしノードがラベルなどのリテラルだった場合、その重要度は 0 とする。)
 - (c) $scoreEdge(r) += \sum_{e \in ClassedEdges} PFIPF[c, e] \times decay$ ここで、PFIPF[c, e] は Property Frequency (PF) をクラス c, エッジ e の Inverted Property Frequency (IPF) で割った値とする。
 - (d) $decay = decay \times decayFactor$ (decayFactor < 1.0)
 - (e) Adj と隣接した、まだ訪れていないノードで Adj を初期化する。

本アルゴリズムの特徴は 3-(a) においてエッジの更新を行う際に、(c,e) のようにエッジの主語が属するクラス情報を用いた更新を行う点である。この更新情報により、3-(c) においては以下のようにエッジのスコアを更新する。

$$scoreEdge(r) += \sum_{e \in Edges} PFIPF[c, e] \times decay \quad (2)$$

IPF によりデータセット内での出現頻度が高いプロパティを高く、PF によってクラス内での出現頻度が高いプロパティを高く評価することで、主語のリソースにとって特徴的なプロパティを高く評価することができる。また decay を掛け合わせることで、ユーザの興味の対象から近い要素については高く、興味の対象から遠い要素については低い評価を与えている。これにより重要度が高く、ユーザの興味と近い検索結果を高く評価することができると考えられる。

3.3.2 提案手法2: トリプルによるクエリの評価

主語とプロパティ、目的語は3つで1つの情報を構成する。我々はノードとエッジを別々にはではなく、一つの情報であるトリプルの単位で評価したいと考えた。手法1ではリソースとプロパティの重要度をそれぞれ別のスコアとして評価しているが、手法2では両者の重要度を用いた評価式 TripleScore の導入を行う。

アルゴリズム 2:

1. $decay = 1.0$, $score = 0.0$ に初期化する.
2. Adj を, SELECT 節で選択されたノードを含んだトリプルの集合で初期化する.
3. Adj が空になるまで以下を繰り返す:
 - (a) $ClassedEdges$ を Adj のノードのクラスと, ノードから伸びたエッジの集合とし, (c, e) で表す.
 - (b) $score(r) += \sum_{t \in Adj} TripleScore[t] \times decay$
 - (c) $decay = decay \times decayFactor$ ($decayFactor < 1.0$)
 - (d) Adj と隣接した, まだ訪れていないトリプルで Adj を初期化する.

TripleScore は以下の式で定められる.

$$TripleScore = \frac{Imp(C_s) \times PFIPF(n_s, edge) \times Imp(n_o)}{linkNum(n_s) + linkNum(n_o) - 1} \quad (3)$$

n_s, n_o は, それぞれトリプルの主語, 目的語のノードを表す. また $linknum(n)$ はノード n から出るエッジの本数を表す. リンクの本数でスコアを分割しているのは, トリプル単位での評価により同じリソースを主語として, 目的語としてなど複数回評価する冗長を回避するためである. なお n_o がリソースでなくリテラルなどであった場合, 重要度にはすべてのリソース n の重要度の平均値を与える.

TripleScore の重要度は主語と目的語, 両者の間のプロパティの双方の重要度により決定する. 重要なトリプルがより高く評価されるため, 提案手法 1 よりも良いランキングを行うことが期待できる.

4 実験

4.1 実験仕様

本研究で設定した 2 つの目的について SPARQL クエリを設定し, Bamba らのアルゴリズムを採用したベースライン手法およびアルゴリズム 1, アルゴリズム 2 を用いて検索結果のランキングの比較を行った. ただし, Bamba らの研究 [9] では, リソースの重要度評価に HITS アルゴリズムを参考にした独自のアルゴリズムを用いているが, プロパティの重み付けなど, 自分での設定が必要であり再現が困難なことから, 本論文では PageRank アルゴリズムを用いた重要度評価を行った. 重みはすべてのアルゴリズムで, $decayFactor=0.5$, $nodeWeight=edgeWeight=0.5$ とした.

実験に使用したデータを以下の表 1 に示す.

リソースから情報を得る場合, リソースには ‘:Kyoto’ (京都), ‘:University_of_Tokyo’ (東京大学) を用いた. また, トリプルからリソースを得る場合については, 条

表 1 使用データ

データセット	DBpedia 3.8
総リソース数	9440897
総トリプル数	158373972

件として「クラス ‘dbo:University’ に属するリソース」「クラス ‘dbo:City’ に属するリソース」を設定した. 実際に用いたクエリを表 2 に示す.

クエリ 1, 2 はリソースからトリプルを検索する場合, クエリ 3, 4 はトリプルの条件からリソースを検索する場合のクエリである.

評価は人手での評価とし, 12 人の被験者が各アルゴリズムにおける結果について, 1 から 5 の範囲で評価値を付けた.

表 2 実験で使用されたクエリ

ID	クエリ
クエリ 1	SELECT ?property ?object WHERE{ :Kyoto ?property ?object.}
クエリ 2	SELECT ?property ?object WHERE{ :University_of_Tokyo ?property ?object.}
クエリ 3	SELECT ?subject WHERE{ ?subject rdf:type dbo:City. ?subject ?property ?object.}
クエリ 4	SELECT ?subject WHERE{ ?subject rdf:type dbo:University. ?subject ?property ?object.}

4.2 結果

被験者実験の評価値の平均を表 3, 4 に示す.

表 3 被験者実験による評価値の平均: トリプルの検索

リソース	ベースライン	アルゴリズム 1	アルゴリズム 2
:Kyoto (クエリ 1)	2.38	3.38	2.79
:University_of_Tokyo (クエリ 2)	2.58	3.58	3.63

表からトリプルの検索において, リソースを ‘:Kyoto’ に指定した場合はアルゴリズム 2, ‘:University_of_Tokyo’ に指定した場合にはアルゴリズム 3 が最も高く評価されたことが分かる. ただし, アルゴリズム 2 は 2 つのクエリにおいて評価値がほぼ同じであるが, アルゴリズム 3 はリソース ‘:Kyoto’ に対するクエリの評価値が低く, クエリによって評価値が異なる結果となった.

リソースの検索においては, アルゴリズム 3 が最も高く評価された. どちらのクエリにおいても, アルゴリズ

表 4 被験者実験による評価値の平均：リソースの検索

クラス	ベースライン	アルゴリズム 1	アルゴリズム 2
dbo:City (クエリ 3)	2.63	2.88	3.83
dbo:University (クエリ 4)	2.42	2.54	3.79

ム 1 と 2 の評価はほぼ同じであったが、アルゴリズム 2 の方が若干スコアが上回る結果となった。

4.3 考察

被験者実験においてアルゴリズム 1 の評価は常にベースラインの評価を上回った。特にクエリ 1, 2 におけるトリプルのランキングにおいて、他の手法よりも高い評価値を得ていることが分かる (表 3)。従って、アルゴリズム 1 でのプロパティの重み付けの変更は、SPARQL 検索結果のランキングに良い影響を与えていると言える。

アルゴリズム 2 はクエリ 3, 4 のリソースの検索においては他手法よりも上回る評価が得られた (表 4)。アルゴリズム 2 では検索結果の評価として TripleScore を導入したスコアリングを行っているが、これらのクエリに対しては有効に働いていると考えられる。

しかしながらクエリ 1, 2 のトリプルの検索においては、アルゴリズム 2 でのランキングはアルゴリズム 1 と同程度か、それよりも下回る評価となった (表 3)。アルゴリズム 2 を用いたランキングではトリプル単位で検索結果をランキングする TripleScore を導入しており、アルゴリズム 1 よりも、トリプルの要素の重みがそれぞれ影響しあう。そのため、プロパティまたはオブジェクトの重みが強く影響した結果が上位にランキングされ、トリプルのランキングにおいては安定した結果が得られなかった可能性が考えられる。

5 結論

本研究では SPARQL クエリの検索結果を対象とした、ランキングアルゴリズムを提案した。プロパティの重要度としてトリプルの主語のクラスに基づいた PFIPF を用いることにより、属するクラスによって異なったプロパティの重要度評価を可能にした。

被験者実験を行った結果、2 つの提案手法はどちらもベースラインの評価を上回り本研究において提案されたプロパティ評価手法がランキングに良い影響を与えたことを示した。

一方で、提案手法 1 を改善するために導入された提案手法 2 の被験者実験は一部では提案手法 1 の評価よりも良いとは言えない結果となった。提案手法 2 で導入された TripleScore の評価手法では提案手法 1 よりも結果の値が安定しないことを示唆している。

Linked Open Data のデータセットはこれからも増え

ていくと考えられ、データの検索手法もますます重要になってくると考えられる。本手法の SPARQL 検索結果のランキングも、そのようなデータ検索のアプローチの一つである。データセット内のリンク情報を用いた重み付け手法はシンプルでコストがかからないという点から有用な手法であると考えられる。

今後は TripleScore の算出に用いるリソースとプロパティの重要度がどのように TripleScore に影響を与えているかを調査し、より安定したトリプルの評価を行えるよう、アルゴリズムの改善を行ってきたい。

参考文献

- [1] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becler, R. Cyganiak, "DBpedia - a crystallization point for the web of data," Web Semant.,7, pp. 154-165, (2009)
- [2] Ding, L., Pan, R., Finin, T., et al.: Finding and ranking knowledge on the semantic web, Proc. of the 4th International Semantic Web Conference, pp. 156-170, 2005.
- [3] Page, L., Brin, S., Motwani, R., et al.: The pagerank citation ranking: Bringing order to the web, Technical report, Stanford University, 1998.
- [4] Kleinberg, J. M.: Authoritative sources in a hyperlinked environment, Journal of the ACM, 46(5), pp. 604-632, 1999.
- [5] Hogan, A., Harth, A. and Decker. S.: Reconrank: A scalable ranking method for semantic web data with context, Second International Workshop on Scalable Semantic Web Knowledge Base Systems, 2006.
- [6] Balmin, A., Hristidis., V., and Papakonstantinou, Y.: Objectrank: Authority-based keyword search in databases, Proc. of the Thirtieth International Conference on Very Large Data Bases (VLDB), vol. 30, pp.564-575, 2004.
- [7] Chirita, P. A., Costache, S., Nejd, W., et al.: Beagle++: Semantically enhanced searching and ranking on the desktop, Proc. of the 3rd European Semantic Web Conference, pp. 348-362, 2006.
- [8] Anyanwu, K., Maduko, A., Sheth, A.: Semrank: Ranking complex relationship search results of the semantic web," In WWW'05, pp. 117-127, 2005.
- [9] Bamba. B. and Soubata., S.: Utilizing resource importance for ranking semantic web query, SWDB 2004, pp. 185-198, 2004.
- [10] Mulay. K. and Kumar, P. S.: SPRING: Ranking the results of SPARQL queries on Linked Data, Proc. of the 17th International Conference on Management of Data (COMAD), 2011.
- [11] Salton, G. and cGill, M.: Introduction to Modern Information Retrieval, McGrawHill Book Company, 1984.