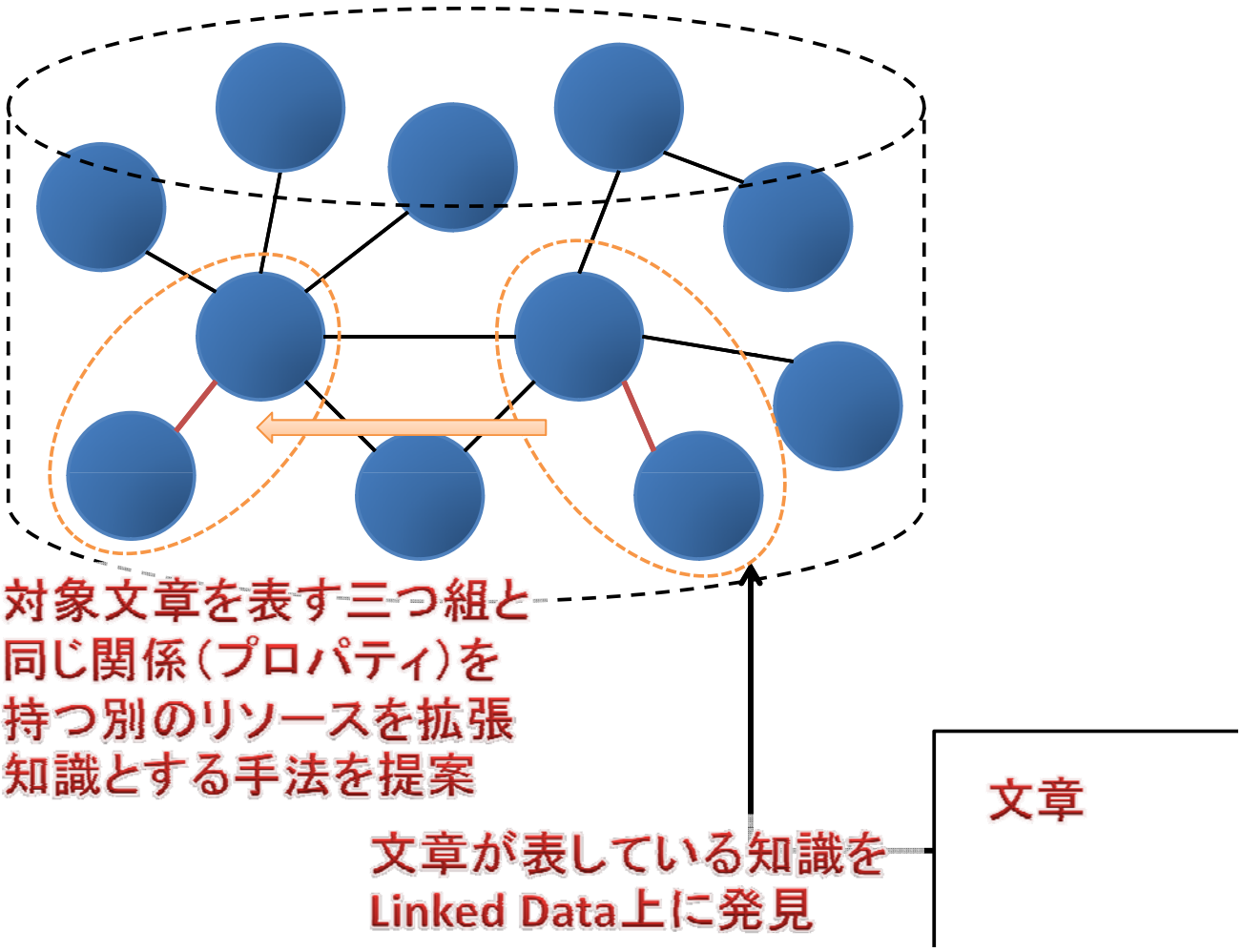


【入力】

All trains running within Scotland, including the local Glasgow trains, are operated by First ScotRail, who own the franchise as determined by the Scottish Government. Central Station and Queen Street Station are the two main railway terminals. Glasgow Central is the terminus of the 641.6-kilometre (398.7 mi) long West Coast Main Line from London Euston. All services to and from England use this station. Glasgow Central is also the terminus for suburban services on the south side of Glasgow, Ayrshire and Inverclyde, as well as being served by the cross city link from Dalmuir to Motherwell.

（http://en.wikipedia.org/wiki/Glasgowより）

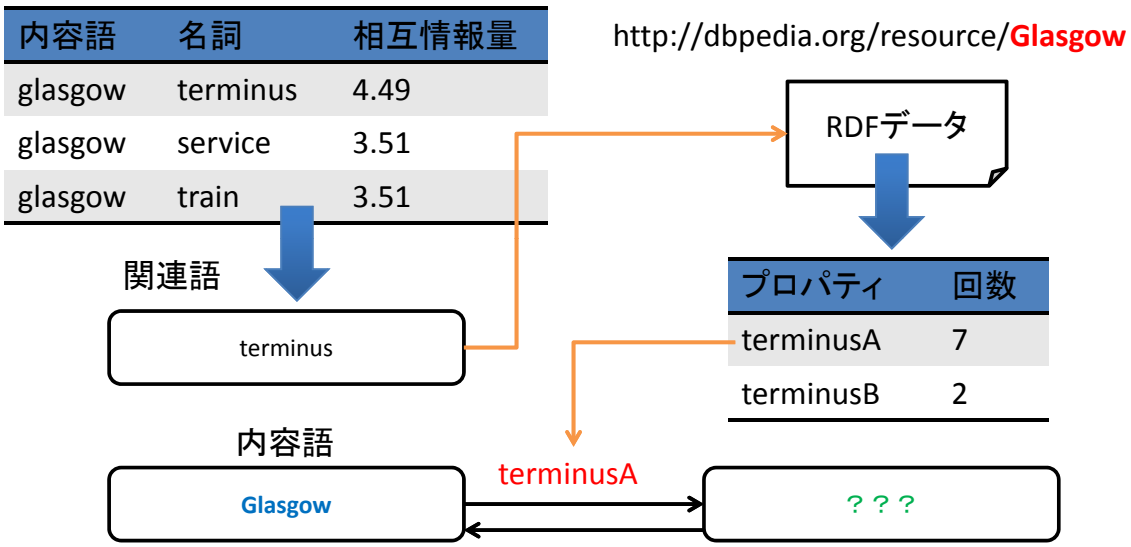
名詞	出現回数	不偏分散値
glasgow	4	4.96
station	3	98.12
terminus	2	400.00
service	2	1024.00
train	2	1600.00

内容語

Glasgow

参照RDFデータ：Dbpedia

http://dbpedia.org/resource/**Glasgow**




```
SELECT * WHERE {
{?Resource1 <P> <R>}
UNION
{<R> <P> ?Resource2}}
```

terminus a of M77_motorway is Glasgow.
terminus a of A89_road is Glasgow.
terminus a of A81_road is Glasgow.
...


```
SELECT * WHERE
{?Resource1 <P> ?Resource2}
```

結果が多量のため
内容語のクラスで絞り込む

```
SELECT * WHERE {
?Resource1または?Resource2 rdf:type <Resource type>.
?Resource1 <P> ?Resource2}
```

terminus a of A90_road is Edinburgh.
terminus a of A71_road is Edinburgh.
terminus a of A70_road is Edinburgh.
...

拡張知識

Linked Open Data を利用した情報拡張手法の提案 –DBpedia を例に–

大西可奈子
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

1 はじめに

近年，大容量かつ多様化する Web ドキュメントをどのようにして有効に扱うかが大きな課題となってきている．そこで，この問題の有効的な解決方法に成り得ると考えられるメタデータやセマンティック・ウェブの技術が，現在改めて注目されている．セマンティック・ウェブは，1998 年頃に Tim Berners-Lee 氏によって提唱された技術 [1] であり，従来の HTML では伝えきれなかった，語彙の意味なども記述できる XML、XML Schema、RDF、RDF Schema、OWL などを階層的な資源として構成される．. セマンティック・ウェブが注目を浴びる中，その技術の一つとして Tim Berners-Lee 氏によって新たに Linked Data[2][3] が提唱された．主要な Linked Data のいくつかとして，すべての国の地理情報，および 800 万の地名を Linked Data で記述した Geonames[4]，音楽のメタデータデータベースである MusicBrainz[5]，概念辞書である WordNet[6]，Dbpedia[7] などがある．DBpedia は，Wikipedia から構造化された情報を抽出し，その情報を Web で利用可能な RDF の形にして提供しているものである．抽出した語彙には、それぞれ URI が与えられており，その URI に語彙の概念や，固有名詞が持つ情報などが記述されている．本研究では，このようにして日々作られている Linked Data を利用して，Web 上に存在する膨大な情報の中からユーザの興味に応じた情報を提供する手法を提案する．なお，本研究では幅広い分野からユーザの興味に基づいた情報を提示することを目標としているため，情報資源には語彙の豊富な DBpedia を利用する．

2 関連研究

検索エンジンの開発において，Swoogle[8]，Watson[9]，SWME[10]，Sindice[11] など，Linked Data を利用した多くの研究がなされている．それ以外では，コンテンツを Linked Data と結び付けるアノテーション技術により，検索精度を従来よりも高める研究など数多く報告されている．例えば，BBC は自身のコンテンツを Linked Data で記述し，DBpedia や MusicBrainz とリンクさせるシステムを開発している [12]．また，対象コンテンツをビデオコンテンツに特化したものとして，ビデオデータのための意味検索を容易にするための手法を提案する研究 [13] や，動画の検索タスクに関連する画像を外部ソース（DBpedia，Flickr，Google Image）から自動で取得する [14] などがある．DBpedia Mobile[15] は，GPS 情報を用いて携帯にユーザの位置情報に加えて，その位置情報に関連する情報を DBpedia から取得しラベルやアイコンで表示する．これらの研究はいずれも実際に記述されている物・事（リソース）に対して，Linked Data を用いて，そのリソースに関する追加情報を取得することにより，検索を容易にしたり情報拡張を行うものである．本研究では，あるリソースに対する追加情報を取得するだけでなく，ある文章に記述されている二つのリソース間にある “関係” を取得し，ユーザに提示する．加えて，それらと同じ関係を持つ別のリソースを取得し，その情報もユーザに提示することにより，ユーザの興味に沿った情報拡張を行うことを目的とする．

3 情報拡張手法

本研究で提案する情報拡張手法を実現するシステムの概観を図 1 に示す．

ユーザが Web ページに記述されたある文章の中のある一部分に興味を持ち，その範囲を選択したと仮定する．システムは最初に，その選択された範囲の内容を最もよく表わしていると考えられる名詞をひとつ抽出する（図 1 中①）．本研究では，このような名詞を “内容語” と呼ぶ．この名詞は一つ目のリソースとなり，そのリソースに対する URI が参照される．例えば，その名詞が “Tokyo” だった場合，Tokyo に与えられている URI が指す RDF 内に記述されているデータを解析し，ユーザが興味をもった一部分より得られる知識の抽出を行う．

しかし，RDF は必ず Domain, Property, Range の三つ組で記述されているため，リソースが一つ決まっただけでは特定の知識を抽出することはできない．そこで，その名詞に対してのその他の名詞の関連の強さ，および重要度を求める（図 1 中②）．この時，重要度を数値化したものが最大となる語を内容語の “関連語” と呼ぶ．さらに，関連語について WordNet を用いて同義語を取得し（図 1 中③），それら同義語も関連語の一部として扱う．ここで，内容語と関連語の関係を見つけるために，内容語 URI が指す RDF データの中に関連語を含む知識を探す（図 1 中④）．その知識にしたがって，システムは RDF クエリ言語 SPARQL[16] クエリを自動作成し（図 1 中⑤），エンドポイントを通して Linked Data にアクセスし（図 1 中⑥），内容語と関連語に関係する知識の抽出を行う（図 1 中⑦）．ここで抽出される知識は RDF 言語で記述されているため，必ず “関係” を持つ．本研究では，その “関係” を持つもので，かつ内容語を含まないものを拡張情報としてユーザに提供する．ここでもシステムは，SPARQL クエリを自動作成し（図 1 中⑧），再度エンドポイントを通して Linked Data にアクセス，必要な知識の抽出を行う．

3.1 内容語抽出

本研究では，“何度も繰り返し言及される単語は重要な単語である” という仮説の下，ユーザが興味を持った文章を最もよく表わしている単語は，文章中において何度も繰り返し言及されている単語と考える．また，“重要な単語は一箇所に偏らず文章全体に現れる” という仮説を立て，文章中に頻出し，かつ文章全体に万遍なく出現している名詞が，ユーザが興味を持った文章 D を最もよく表わしている名詞，つまり内容語であると考える．

そこで，文章 D に含まれる名詞集合を $N = \{n_1, n_2, ..., n_i\}$ とする．選択された文章の形態素解析には openNLP[17] を用い，名詞（代名詞は含まない）のみを抽出し利用した．N は語の重複を許さない名詞の集合とする．また，ある名詞 n の文章 D における出現頻度を $f_D(n)$，ある名詞 n が選択された文章 D において，最初から数えて何文字目に出現したか（出現位置）を $pos(n) = \{p_1, p_2, ..., p_{f_D(i)}\}$ と表わす時，名詞 n の散らばりの程度 $W(n)$ を不偏分散を用いて以下の式で求める．

$$W(n) = \frac{1}{(f_D(n)-1)^{\alpha}} \sum_{i=1}^{n} (|p_{i+1} - p_i| - \bar{p})^2 \qquad (1)$$

これは，“単語 n が出現する間隔が，文章 D を単語 n の出現回数で割ったもの（単語間の距離の平均）に近ければ近いほど，ある単語 n が文章 D 上で最も均等に散らばっている” と考えられるためである．ここで，式 (1) において $\frac{1}{f_D(n)-1}$ を α 乗しているのは，前述した仮説 “頻出単語は重要である” により出現回数を強く考慮するためであり，α は経験的に 3 とする．なお，$\bar{p}$ は n が最も均等に分散した場合の単語間距離を表わしている．すなわち，文章 D に含まれる全単語数を X とすると，$\bar{p} = \frac{X}{f_D(n)}$ となる．$W(n)$ は名詞 n が広範囲かつ均等に出現している時，最小となる．したがって，$W(n)$ を最小とする名詞 n が文章 D を最もよく表わしている名詞と言える．本研究では，そのような名詞 n が文章 D の “内容語 ”となる．

3.2 関連語抽出

次に，内容語に対する関連語を見つける手法を示す．関連語とは，文章 D 中に出現し，内容語と強い関連を持ち，かつ重要だと思われる名詞のことである．本研究では，内容語との関連の度合いは相互情報量を用いて表し，重要度は単語の出現回数の平方根を用いて表す．したがって，内容語を n_k とする時，文章 D 中に現れる名詞 n_l の n_k に対する関連語らしさ $K(n_k, n_l)$ は以下のように表せる．

$$K(n_k, n_l) = I(n_k, n_l) \times \sqrt{f_D(n_l)} \qquad (2)$$

ここで，$I(n_k, n_l)$ は内容語 n_k と関連語候補 n_l の相互情報量であり，次の式で求められる．

$$I(n_k, n_l) = \log \frac{p(n_k, n_l)}{p(n_k)p(n_l)} \quad (n_k \neq n_l) \qquad (3)$$

式 (3) において，$p(n_k, n_l)$，$p(n_k)$，$p(n_l)$ はそれぞれ以下のように表わされる．

$$p(n_k, n_l) = \frac{f_D(n_k, n_l)}{X},\ p(n_k) = \frac{f_D(n_k)}{X},\ p(n_l) = \frac{f_D(n_l)}{X}$$

ここで X は文章 D 中の名詞の総数を表し，$f_D(n_k, n_l)$ は n_k と n_l が同時に一文中に出現する頻度を表す．これは，“ある二つの単語が一文中に同時に現れた時，その二単語は強い関係によって結ばれている可能性がある” と考えられるためである．したがって，$I(n_k, n_l)$ が大きいほど n_k と n_l は強い関係で結びついているとみなせる．

ここで，ある内容語 n_k について，その他のすべての名詞の $K(n_k, n_l)$ を求め，$K(n_k, n_l)$ について降順に並び変えたリストを N' とし，$K(n_k, n_l)$ を最大にする名詞 n_l を関連語と呼ぶ．

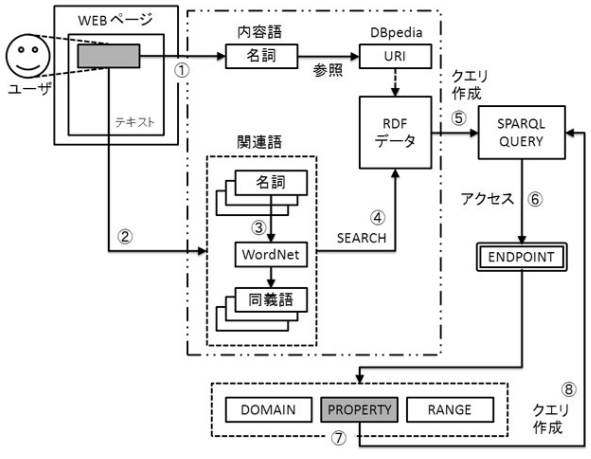


図 1: 提案手法過程

3.3 関連語の拡張

取得した関連語は WordNet の Synset を利用して同義語が取得される．Synset とは WordNet が提供する同義語の単語セットである．例えば，“Japan” という単語に対しては，{Japanese_Islands, Japanese_Archipelago, Nippon, Nihon} という同義語の集合が提供されている．本研究では，Synset に含まれるこれらの単語も関連語として扱う．

3.4 関係抽出

内容語と関連語の二つの名詞を基に，関係の抽出を行う．まず，行列の要素 $T_{ij}(i = \{0, 1, 2\}; 0 \leq j \leq m-1)$ が $\{0, 1\}$ をとる以下の行列を定義する．ここで i は 0, 1, 2 の値をとり，それらは Dmain，Property，Range をそれぞれ表わし，j は内容語 n に関して抽出された RDF データに記述された三つ組の内，関連語を含む三つ組の個数を示す．

$$T_{ij} = \begin{bmatrix} T_{00} & T_{01} & \cdots & T_{0m-1} \\ T_{10} & T_{12} & \cdots & T_{1m-1} \\ T_{20} & T_{23} & \cdots & T_{2m-1} \end{bmatrix} \qquad (4)$$

この行列は，内容語 n の RDF データに記述されている関連語を含むすべての三つ組に対して，Domain に関連語を含む場合は $T_{0j} = 1$，Property に関連語を含む場合は $T_{1j} = 1$，Range に関連語を含む場合は $T_{2j} = 1$ とする．関係の抽出は行列 T を使って表 1 に示すアルゴリズムに従って行われる．

この時，行列 T が以下の条件（表 1 中，条件 1 に相当）を満たす時（抽出した関連語が Property として多く記述されている場合に相当），

表 1: 情報拡張処理疑似コード

```
Input: 内容語リスト C = {c_1, ..., c_i}
Output: 拡張情報
for 内容語 ∈ C do
 関連語リスト作成 R = {r_1, ..., r_j}

 for 関連語 ∈ R do
  行列 T 作成
  if 行列 T が条件 1 を満たす then
  関連語を Property として知識抽出
  else if 行列 T が条件 2 を満たす then
  関連語を Domain または Range として知識抽出
  else if 関連語が Synset をもつ then
  同義語リスト作成 S = {s_1, ..., s_j}

  for 同義語 ∈ S do
   行列 T 作成
   if 行列 T が条件 1 を満たす then
   関連語を Property として知識抽出
   else if 行列 T が条件 2 を満たす then
   関連語を Domain または Range として知識抽出
  end

 end
end
if 知識が抽出されている then
内容語，関連語に対して拡張情報取得
exit
```

$(\sum T_{1j} \geq \sum T_{0j}) \wedge (\sum T_{1j} \geq \sum T_{2j})$

内容語をリソース，関連語を含むプロパティを関係として，知識の抽出を行う．

または，行列 T が以下の条件（表 1 中，条件 2 に相当）を満たす時（抽出した関連語がリソースとして多く記述されている場合に相当），

$(\sum T_{0j} \geq \sum T_{1j}) \wedge (\sum T_{0j} \geq \sum T_{2j}) \wedge (\sum T_{0j} < \alpha)$

または，

$(\sum T_{2j} \geq \sum T_{0j}) \wedge (\sum T_{2j} \geq \sum T_{1j}) \wedge (\sum T_{2j} < \alpha)$

内容語をリソース，関連語をもう一つのリソースとし，その二つのリソースの間にある関係を抽出する．

ここで，関連語が文章の特徴を表わさない単語だった場合，どの文章にも頻出する単語と考えられるため $\sum T$ は非常に大きな数値になると想定され，これは期待される単語の抽出がなされない．したがって，予備実験の結果を踏まえて，現在のところ α を経験的に 40 としたが，α は対象領域に依存して決められる．

$\sum T = 0$ となった場合で，関連語が同義語を持つ場合，同義語を新たな関連語として行列 T を求め直す．同義語を持たない場合，または，すべての同義語について $\sum T = 0$ となった場合，関連語らしさ $K(n_k, n_l)$ が n_l の次に大きい名詞について行列 T を求め直す．

すべての名詞 n について $\sum T = 0$ となった場合は，分散の程度 $W(n)$ が名詞 n の次に小さい名詞を新たに文章 D の内容語とし N' を求め直す．

3.5 クエリの作成

関係知識を Linked Data を用いて抽出するためのクエリは解析対象に基づいて自動生成される．なお，本研究では DBpedia へは，RDF クエリ言語 SPARQL を使い，エンドポイント `http://dbpedia.org/sparql` からアクセスする．

3.4 節に示したように，関連語を含む Property が解析対象であると判断された場合，ある文章から抽出される知識は “**リソース R（内容語）に対して，関連語を含む関係（Property）P を持つリソース**” になる．従って，文章 D から抽出できる知識は，SPARQL のコマンドで表現すると，

```
SELECT * WHERE {
{ ?Resource1 <P> <R> } UNION { <R> <P>
?Resource2 }}
```

で求められる．

一方，関連語を含む Domain または Range が解析対象であると判断された場合，文章 D から抽出される知識は “**リソース R（内容語）が、その他の関連語を含むリソース R’ との間に持つ関係（Property）**” である．従って，文章 D から抽出できる知識は，

```
SELECT * WHERE {
{<R> ?property <R'> } UNION { <R'> ?property
<R> }}
```

で求められる．

また，上記二つのクエリから抽出された知識が持つ関係を P とする時，新たに抽出される情報は “**関係 P を持つ Domain と Range の組**” である．したがって，文章 D に対して拡張できる情報は，

```
SELECT * WHERE {
?Resource1 <P> ?Resource2 }
```

で求められる．これにより，選択文章から抽出した知識がもつ関係を保存した，新たな知識を抽出することができる．

3.6 絞り込み

取得した知識が一定量以上の場合は，リソースのカテゴリによってユーザに提示する知識の絞り込みを行う．例えば，図 2 において，ある内容語を Resource1 とする時，関連語より関係 P が取得できるとする．この時，拡張できる情報は 3.5 節で述べたように，“関係 P を持つ Domain と Range の組” である．SPARQL クエリによって Linked Data から得られた結果リソースを，図 2 で示す R 群（グループ α とグループ β の和）とする．この R 群が多量であった場合，Resource1 のクラスを取得．図 2 においては Resource1 はクラス B に所属するため，結果リソースの内クラス B に所属するグループ α のみが抽出される．

3.7 表示

抽出された知識を表現する際，表示には対象プロパティ P のラベルを用いる．ラベルの取得には以下のクエリを用いる．

```
SELECT * WHERE { <P> rdfs:label ?Value }
```

このラベル L を用いて，抽出した知識以下の形式で表示する．

```
L of Resource1(domain) is Resource2(range).
```

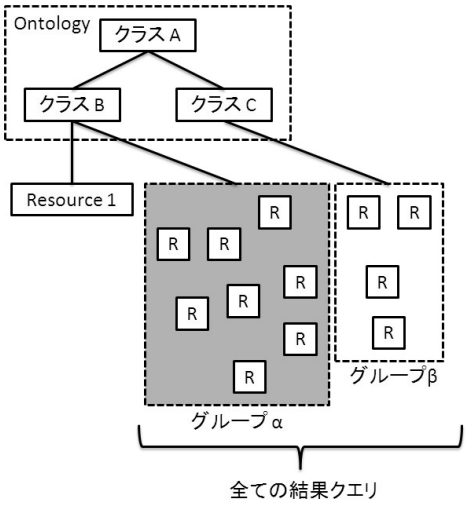


図 2: カテゴリによる絞り込み

表 2: 名詞の出現回数数

名詞	出現回数
glasgow	4
station	3
train	2
terminus	2
service	2

4 ケーススタディ

Glasgow（イギリス，スコットランドの都市）の交通ついて記述した以下のような文章があり，これを入力とする場合について具体的に説明する．

> *All trains running within Scotland, including the local Glasgow trains, are operated by First ScotRail, who own the franchise as determined by the Scottish Government. Central Station and Queen Street Station are the two main railway terminals. Glasgow Central is the terminus of the 641.6-kilometre (398.7 mi) long West Coast Main Line from London Euston. All services to and from England use this station. Glasgow Central is also the terminus for suburban services on the south side of Glasgow, Ayrshire and Inverclyde, as well as being served by the cross city link from Dalmuir to Motherwell.*
>
> （*http://en.wikipedia.org/wiki/Glasgow* より）

この文章の名詞の出現回数を表 2 にまとめる．

その他の名詞，scottish，government，scotland，queen，street，railway，terminal，scotrail，west，coast，line，euston，franchise，england，use，side，ayrshire，inverclyde，cross，city，link，dalmuir，motherwell の出現回数はそれぞれ 1 回である．

それぞれの名詞の不偏分散値は表 3 に示す通りである．数値は小数点以下第 3 位を切捨てとしている．

表 3: 名詞の不偏分散値

名詞	不偏分散値 $W(n)$
glasgow	4.96
station	98.12
terminus	400.00
service	1024.00
train	1600.00

表 4: glasgow との相互情報量

名詞	相互情報量 K
terminus	4.49
service	3.51
train	3.51

従って，内容語の第一候補として glasgow が抽出された．ここで，内容語 glasgow とその他の名詞の相互情報量を表 4 に示す．数値は小数点以下第 3 位を切捨てとしている．

なお glasgow と，scottish，government，scotland，scotrail，west，coast，line，euston，franchise，side，ayrshire，inverclyde，cross，city，link，dalmuir，motherwell の相互情報量はすべて 3.17 であった．

ここで，glasgow との相互情報量が最も高い terminus が関連語第一候補となる．そこで，terminus の同義語を WordNet を利用して取得する．state の同義語として，end_point，endpoint，termination，destination，terminal_figure，term，terminal，depot が取得され，これらすべてが関連語候補となった．

ここで，内容語 glasgow の RDF データを参照する．今回は参照 RDF データとして DBpedia を用いた．従って参照する URI は，`http://dbpedia.org/resource/Glasgow` となる．この glasgow についての RDF 記述には，関連語 terminus を含む一部の記述があった．その一部を N-Triples 記述を用いて以下に示す．

<http://dbpedia.org/resource/M80_motorway>
<http://dbpedia.org/property/terminusA>
<http://dbpedia.org/resource/Glasgow>.

<http://dbpedia.org/resource/A82_road>
<http://dbpedia.org/property/terminusA>
<http://dbpedia.org/resource/Glasgow>.

<http://dbpedia.org/resource/A81_road>
<http://dbpedia.org/property/terminusA>
<http://dbpedia.org/resource/Glasgow>.

<http://dbpedia.org/resource/A89_road>
<http://dbpedia.org/property/terminusA>
<http://dbpedia.org/resource/Glasgow>.

<http://dbpedia.org/resource/M77_motorway>
<http://dbpedia.org/property/terminusA>
<http://dbpedia.org/resource/Glasgow>.

表 5: nation を含む出現プロパティと出現数

プロパティ	出現数
http://dbpedia.org/property/terminusA	7
http://dbpedia.org/property/terminusB	2

<http://dbpedia.org/resource/A749_road>
<http://dbpedia.org/property/terminusB>
<http://dbpedia.org/resource/Glasgow>.

<http://dbpedia.org/resource/A77_road>
<http://dbpedia.org/property/terminusA>
<http://dbpedia.org/resource/Glasgow>.

<http://dbpedia.org/resource/
A80_road_%28Great_Britain%29>
<http://dbpedia.org/property/terminusA>
<http://dbpedia.org/resource/Glasgow>.

<http://dbpedia.org/resource/A74_road>
<http://dbpedia.org/property/terminusB>
<http://dbpedia.org/resource/Glasgow>.

プロパティに 9 回出現しているため，行列 T は，

$$T_{ij} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5)$$

となり，$(\sum T_{1j} \geq \sum T_{0j}) \wedge (\sum T_{1j} \geq \sum T_{2j})$ を満たす．

terminus を含む出現プロパティとその回数は，表 5 に示す通りである．

従って，まず http://dbpedia.org/property/terminusA についてクエリを作成する．

```
SELECT * WHERE {
{ ?Resource1
<http://dbpedia.org/property/terminusA>
<http://dbpedia.org/resource/Glasgow> }
UNION
{ <http://dbpedia.org/resource/Glasgow>
<http://dbpedia.org/property/terminusA>
?Resource2 }}
```

また，知識を表示するために利用するラベルを以下のクエリで抽出する．

```
SELECT * WHERE {
<http://dbpedia.org/property/terminusA>
rdfs:label ?hasValue}
```

これにより，http://dbpedia.org/property/terminusA のラベルが *terminus a* であるとわかる．よって，抽出される知識は以下のように表される．

terminus a of M80_motorway is Glasgow.
terminus a of A80_road_%28Great_Britain%29 is Glasgow.
terminus a of A82_road is Glasgow.
terminus a of A77_road is Glasgow.
terminus a of M77_motorway is Glasgow.
terminus a of A89_road is Glasgow.
terminus a of A81_road is Glasgow.

次に，プロパティhttp://dbpedia.org/property/terminusA を持つ別の知識を抽出するためのクエリが作成される．

```
SELECT * WHERE {?Resource1
<http://dbpedia.org/property/terminusA>
?Resource2}
```

結果が多量（現在は 2000 以上を多量と設定している）だったため，内容語 Glasgow の型に従って絞り込みを行う．まず，以下のクエリによって Glasgow の型が求められる．

```
SELECT * WHERE {
<http://dbpedia.org/resource/Glasgow>
rdf:type ?Resourec2}
```

これにより Glasgow は Settlement，PopulatedPlace，Place に属することがわかる．ここで，Settlement は PopulatedPlace のサブクラスであり，PopulatedPlace は Place のサブクラスである．そこで，http://dbpedia.org/property/terminusA の目的語の型が Place であるもののみを抽出する．Glasgow はプロパティterminusA の目的語となるので，

```
SELECT * WHERE {?Resource2 rdf:type
<http://dbpedia.org/ontology/Place>.
?Resource1 <http://dbpedia.org/property/terminusA>
?Resource2}
```

上記のクエリにより，新たな知識が抽出される．抽出された知識の一部を以下に示す．

terminus a of A8_road_%28Scotland%29 is Edinburgh.
terminus a of M8_motorway_%28Scotland%29 is Edinburgh.
terminus a of A702_road is Edinburgh.
terminus a of A7_road_%28Great_Britain%29 is Edinburgh.
terminus a of A90_road is Edinburgh.
terminus a of A71_road is Edinburgh.
terminus a of A70_road is Edinburgh.

Edinburgh とは Glasgow から電車で約一時間の距離にある，イギリス，スコットランドの首都である．すなわち，Glasgow の交通に関する文章から，まず Glasgow の交通知識が抽出でき，さらに Glasgow 以外の都市の交通知識が抽出できたことがわかる．

次に，プロパティhttp://dbpedia.org/property/terminusB について情報を抽出する．http://dbpedia.org/property/terminusA の時と同様に，

```
SELECT * WHERE { { ?Resource1
<http://dbpedia.org/property/terminusB>
<http://dbpedia.org/resource/Glasgow> }
```

```
UNION
{ <http://dbpedia.org/resource/Glasgow>
<http://dbpedia.org/property/terminusB>
?Resource2 }}
```

というクエリによって，Glasgow にといてプロパティ terminusB を持つ知識を抽出する．プロパティ terminusB のラベルも同様に求められる．従って，

terminus b of A74_road is Glasgow.
terminus b of A749_road is Glasgow.

が，Glasgow のプロパティ terminusB に関する知識である．同様に，プロパティ terminusB を持つその他の知識も抽出される．制約なしでは多量の結果が返ってきたため，同様の処理から以下のクエリが作成される．

```
SELECT * WHERE { ?Resource2 rdf:type
<http://dbpedia.org/ontology/Place>.
?Resource1
<http://dbpedia.org/property/terminusB>
?Resource2 }
```

これにより抽出された知識の一部を以下に示す．

terminus b of A4260_road is Oxford.
terminus b of Bundesautobahn_24 is Berlin.
terminus b of Motorway_1_%28Greece%29 is Athens.
terminus b of Japan_National_Route_289 is Iwaki,_Fukushima.
terminus b of A96_road is Inverness.
terminus b of Spadina_Expressway is North_York.

（以上に示す知識取得結果は 2012 年 1 月 23 日現在の Wikipedia，DBpedia，WordNet の資源に基づくものである．）

5 おわりに

本研究では，ユーザが興味をもった文章から，その文章の内容を最もよく表わしていると考えられる名詞（内容語と呼ぶ）を抽出し，その語に関連のある語と共に，Linked Data を用いて，その文章に関連のある情報を提供する情報拡張手法を提案した．内容語は本研究における知識抽出の基盤であり，知識抽出の際には内容語をインスタンスとする URI が参照される．知識源には DBpedia を用いた．さらに，内容語に強い関連を持つ名詞（関連語と呼ぶ）を取得する．取得された関連語が同義語を持つ場合，それらも関連語として扱う．同義語の抽出には WordNet の Synset を用いた．ここで，内容語と関連語について Linked Data を用いて二語間の関係を説明する情報をユーザに提示する．具体的には，内容語 URI が持つ RDF データの中に関連語を探し，それら二つのインスタンスの関係を探る．これはユーザが興味をもつ二つのインスタンスの間にある情報を提供したことを意味する．Linked Data を利用した情報取得は，SPARQL クエリを自動で作成し，エンドポイントにアクセスして行った．さらに，二つのインスタンスが持つ関係を持つ別のインスタンスを Linked Data を用いて取得し，それを拡張情報としてユーザに提示する．これはユーザが興味をもつ関係に関する情報を提供したことを意味する．またユーザに提供される情報は，DBpedia Ontology の内容語クラスに従って絞り込むことにより，よりユーザが好む情報を提供する．

今後の課題として，現在はユーザが興味を持つ文章の解析に，統計的な処理しか用いていない．そのため，例えば選択された文章が著しく短かった場合など，適切な内容語および関連語を見つけられないことがある．そのため，今後は係り受けなどの構文解析を導入する必要があると考える．また，現在は内容語と関連語の関係を見つけるのに，DBpedia の参照 RDF データに対して線形探索をしているにすぎない．そのため，内容語や関連語が RDF ファイルに含まれないような単語であった場合，有益な情報を取得することができない．今後はこのような場合でも，有益な情報を取得することが可能になるような手法に改良していくことが必要と考える．さらに，本研究で提案した手法の有用性の評価方法の検討と評価実験を行うことも重要であり，今後対処すべき課題として進めていくつもりである．

参考文献

[1] Berners-Lee, Semantic Web Road map(1998), http://www.w3.org/DesignIssues/Semantic.html

[2] Berners-Lee, T.: Design Issues: Linked Data (2006), http://www.w3.org/DesignIssues/LinkedData.html

[3] Bizer, C., Cyganiak, R., Heath, T.: How to publish Linked Data on theWeb (2007), http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/

[4] Geonames: http://www.geonames.org/

[5] MusicBrainz: http://musicbrainz.org/

[6] WordNet: http://wordnet.princeton.edu/

[7] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, Zachary Ives, “DBpedia: a nucleus for a web of open data”, ISWC’07/ASWC’07: Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference, November 2007.

[8] Li Ding, Tim Finin, Anupam Joshi, Rong Pan, R. Scott Cost, Yun Peng, Pavan Reddivari, Vishal Doshi, Joel Sachs, “Swoogle: a search and metadata engine for the semantic web”, CIKM ’04: Proceedings of the thirteenth ACM international conference on Information and knowledge management, November 2004.

[9] Watson: http://kmi-web05.open.ac.uk/WatsonWUI/

[10] SWSE: http://swse.deri.org/

[11] Eyal Oren, Renaud Delbru, Michele Catasta, Richard Cyganiak, Holger Stenzhorn, Giovanni Tummarello, "Sindice.com: a document-oriented lookup index for open linked data", International Journal of Metadata, Semantics and Ontologies, Volume 3 Issue 1, November 2008.

[12] Georgi Kobilarov, Tom Scott, Yves Raimond, Silver Oliver, Chris Sizemore, Michael Smethurst, Christian Bizer, Robert Lee, "Media Meets Semantic Web — How the BBC Uses DBpedia and Linked Data to Make Connections", ESWC 2009 Heraklion Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications, 2009.

[13] Waitelonis, J. Sack, H., "Towards Exploratory Video Search Using Linked Data", 11th IEEE International Symposium on Multimedia, San Diego, CA, pp.540 - 545, 2009.

[14] David Vallet, Ivan Cantador, Joemon M. Jose, "Exploiting external knowledge to improve video retrieval", MIR '10: Proceedings of the international conference on Multimedia information retrieval, March 2010.

[15] Christian Becker, Christian Bizer, "DBpedia Mobile: A Location-Enabled Linked Data Browser", 1st Workshop about Linked Data on the Web (LDOW2008), Beijing, China, 2008.

[16] SPARQL: http://www.w3.org/TR/rdf-sparql-query/

[17] OpenNLP: http://incubator.apache.org/opennlp/