Linked Dataを利用した情報拡張手法の開発

大西可奈子† 小林 一郎†

† 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
〒 112–8610 東京都文京区大塚 2–1–1
E-mail: †{onishi.kanako,koba}@is.ocha.ac.jp

あらまし 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．この問題を解決するために期待される技術として，Linked Data 等のセマンティック・ウェブ技術が広く注目され，この技術に基づいた多くのアプリケーションが開発されている．本研究では，Linked Data を利用し，ユーザの興味に関連する情報をユーザに提供することができる新しい情報拡張手法を提案する．具体的には，ユーザが選択した文書の一部から抽出した二つの語が持つ関係に関する情報を Linked Data を利用して見つけ，ユーザに提供する．また，我々の提案する技術は，ユーザの興味に応じた情報を見つける新しい手段というだけでなく，新しい Web ブラウジング手法となる．

キーワード セマンティックウェブ，Linked Data，情報拡張，DBpedia，WordNet

An Information Enhancement Technique by using Linked Data

Kanako ONISHI† and Ichiro KOBAYASHI†

† Ochanomizu University, Graduate School of Humanities and Sciences, Advanced Sciences
Ostuka 2–1–1, Bunkyo-ku, Tokyo, 112–8610 Japan
E-mail: †{onishi.kanako,koba}@is.ocha.ac.jp

Abstract Recently, it has been a big issue of how we effectively utilize the huge amount of Web documents. With the background of this issue, many fundamental technologies such as search engines, recommendation systems etc. have so far been developed. In particular, as the next promising technology, Semantic Web technology has been widely noticed and nowadays many application systems based on the technology have been getting developed. In this paper, we propose a new information extraction technique from the Web which can provide a user with the information related to user's interests by using the resources of Semantic Web framework, especially Linked Data in this study.

Key words Semantic Web，Linked Data，Information Enhancement，DBpedia，WordNet

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(注1)，音楽のメタデータデータベースである MusicBrainz(注2)，概念辞書である WordNet [4]，Dbpedia [5] 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

(注1)：http://www.geonames.org/
(注2)：http://musicbrainz.org/

手法を提案する．なお，本研究は，ユーザの興味に応じた情報を見つける新しい手段というだけでなく，新しい Web ブラウジング手法となることを目指す．

2. 関 連 研 究

検索エンジンの開発において Linked Data を利用した多くの研究がなされている．そのようなものに，Swoogle [6]，Watson(注3)，SWME(注4)，Sindicec [7] などが挙げられる．検索エンジン以外では，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も数多く報告されている．例えば BBC は，BBC 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 [8]．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，彼らは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した [9] 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 [10] などがある．また Bernhard らは，ユーザが簡易メモを入力すると，システムが潜在的に関連があると思われる DBpedia リソースをランク付けして提示するビデオアノテーション手法を提案した [11]．さらに，セマンティックデータのマッシュアップアプリケーションとして Linked Data を使う事例も報告されている．例えば，Aastrand らは，意味的な背景知識を DBpedia を通して利用することにより，コンテンツをより容易に分類する手法 [12] を提案した．また，DBpedia Mobile [13] 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．地理情報と Linked Data を用いた研究には他にも，沿岸エリアと失業率などの統計変数との間にある関係を分析した研究 [14] 等がある．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” に着目し，情報拡張を行う．また，同時に我々は，岩爪らによって提案されたリンクフリー・ブラウジング [15] のアイディアに基づいたブラウジング手法となることも目指す．

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，その名詞に対する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）．その知識に従って，システムは RDF クエリ言語 SPARQL(注5) 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．

また，ユーザが文章を広範囲に選択した後，その中に含まれるある一語をさらに選択した時（図 1 中 I），システムはその語を関連語として扱い（図 1 中 II），同様の手順で拡張情報を提供する．

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

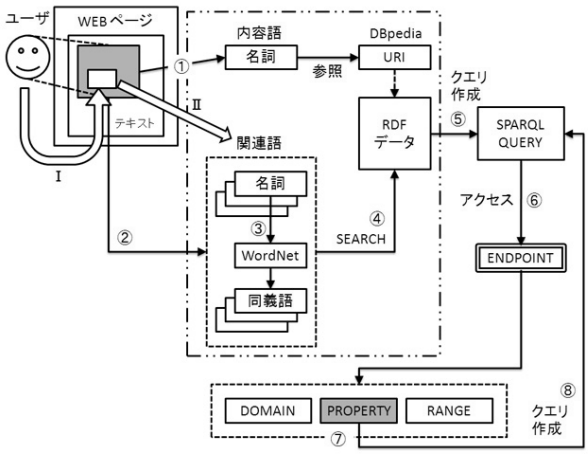

図 1　提案手法過程

(注3)：http://kmi-web05.open.ac.uk/WatsonWUI/
(注4)：http://swse.deri.org/
(注5)：http://www.w3.org/TR/rdf-sparql-query/

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，英語形態素解析器 openNLP(注6) を用い，名詞（代名詞を除く）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，文章 D 中の名詞 n の拡散の程度 $W(n)$ を不偏分散に基づき以下の式で求める．

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

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 を文章 D の “内容語 ”とする．

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．

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

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．

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 追加選択による関連語指定

ユーザが文章のある一部分を選択した後，その選択範囲に含まれる内容語以外の単語をさらに選択した場合，ユーザは，よりその単語に関する内容に興味があると考えられるため，選択された単語を最も関連語らしい単語として扱う．なお，この単語は名詞に限らない．また，追加選択はユーザにより任意で行われる．

3.5 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 $T_{ij}(i = \{0, 1, 2\}; 0 \leqq j \leqq m-1)$ 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 を使ってアルゴリズム 1 に従って行われる．

この時，行列 T が以下の条件（アルゴリズム 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），

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

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（アルゴリズム 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），

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

または，

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

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

(注6)：http://incubator.apache.org/opennlp/

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，以下，同じ手順が続く．

Algorithm 1 情報拡張手法アルゴリズム

```
for 内容語 ∈ C = {c_1, ..., c_i} 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 関連語が同義語を持つ 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 if
            end for
        end if
    end for
end for
if 拡張情報が存在する then
    内容語，関連語に対して拡張情報取得
end if
```

3.6 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，DBpedia のエンドポイント(注7)からアクセスする．

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 ?hasValue WHERE { <R> <P> ?hasValue }
```

(注7)：http://dbpedia.org/sparql

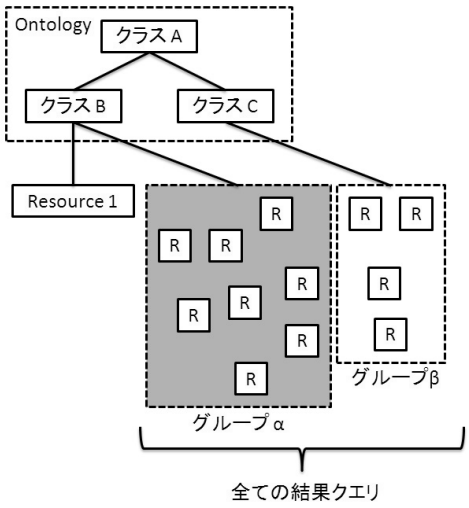

図 2　カテゴリによる絞り込み

または，

```
SELECT ?isValueOf WHERE { ?isValueOf <P> <R> }
```

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 ?property WHERE { <R> ?property <R'> }
```

または，

```
SELECT ?property WHERE { <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 ?isValue ?hasValue WHERE {
?isValue <P> ?hasValue }
```

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．

3.7 絞り込み

取得した知識が一定量以上の場合は，DBpedia 等のオントロジーやカテゴリに基づい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 “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

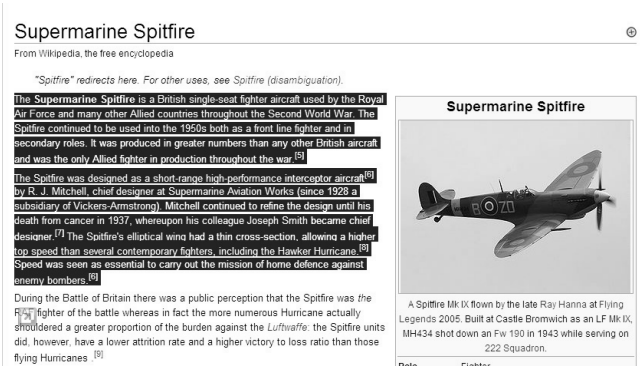

図 3　Wikipedia の Supermarine_Spitfire のページ

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．

3.8 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 ?hasValue WHERE {
<P> rdfs:label ?hasValue }
```

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 isValue(domain) is hasValue(range).

4. ケーススタディ

Wikipedia の Supermarine_Spitfire の項目(注8) において，図 3 に示す反転部分がユーザの興味を持ち選択した箇所とする．この時，内容語を抽出するために $W(n)$ を求めると，$W(n)$ 値の上位 10 項目は表 1 のようになる．

従って，分散値 $W(n)$ を最小とする “spitfire” が選択部分の内容語となる．なお，DBpedia の Spitfire の項目(注9) には Spitfire についての具体的な記述はない．そこで，本研究ではこのような場合にプロパティ `dbpedia-owl:wikiPageRedirects` 等を利用する．例えば，Spitfire.rdf(注10) には，

<rdf:Description
rdf:about=”http://dbpedia.org/resource/Spitfire”>
<dbpedia-owl:wikiPageRedirects rdf:resource=
”http://dbpedia.org/resource/Supermarine_Spitfire”/>

という記述がある．これにより，Spitfire に関する記述が Supermarine_Spitfire に示されていることがわかるため，内容語が “Spitfire” の時，参照 RDF ファイルは Supermarine_Spitfire.rdf(注11) となる．

表 1　$W(n)$ top 10

名詞	出現回数	$W(n)$
spitfire	4	10.546
fighter	4	47.953
supermarine	2	49.000
aircraft	3	107.125
british	2	729.000
war	2	1156.000
chief	2	1936.000
designer	2	1936.000
mitchell	2	3844.000
speed	2	4356.000

表 2　$K(n_k, n_l)$ top 10

内容語	名詞	$K(n_k, n_l)$
spitfire	fighter	6.673
spitfire	aircraft	5.575
spitfire	r	5.010
spitfire	air	4.317
spitfire	war	4.145
spitfire	speed	4.145
spitfire	supermarine	4.145
spitfire	british	4.145
spitfire	front	3.624
spitfire	line	3.624

4.1 追加選択がない場合

ここで，ユーザによる追加選択がない場合は，関連の強さおよび重要度を表わす $K(n_k, n_l)$ 値を求め，関連語を自動で抽出する．この例において $K(n_k, n_l)$ 値の上位 10 項目は表 2 のようになる．

従って，内容語 “spitfire” に対する関連語は，$K(n_k, n_l)$ を最大とする “Fighter/fighter” となる．そこでシステムは Supermarine_Spitfire.rdf 内の Domain，Property，Range の三つ組に，関連語 “fighter” を含むものを抽出する．以下にその一部を掲載する．

<rdf:Description
rdf:about=”http://dbpedia.org/resource/Spitfire_fighter”>
<dbpedia-owl:wikiPageRedirects
xmlns:dbpedia-owl=”http://dbpedia.org/ontology/”
rdf:resource=”http://dbpedia.org/resource/Supermarine_Spitfire”/>

<rdf:Description
rdf:about=”http://dbpedia.org/resource/Kampfgeschwader_200”>
<dbpedia-owl:aircraftFighter
xmlns:dbpedia-owl=”http://dbpedia.org/ontology/”
rdf:resource=”http://dbpedia.org/resource/Supermarine_Spitfire”/>

<rdf:Description

(注8)：http://en.wikipedia.org/wiki/Supermarine_Spitfire
(注9)：http://dbpedia.org/resource/Spitfire/
(注10)：http://dbpedia.org/data/Spitfire.rdf
(注11)：http://dbpedia.org/data/Supermarine_Spitfire.rdf

rdf:about=”http://dbpedia.org/resource/Supermarine_Spitfire”>
<dcterms:subject xmlns:dcterms=”http://purl.org/dc/terms/”
rdf:resource=”http://dbpedia.org/resource/
Category:British_fighter_aircraft_1930-1939”/>

なお，該当 RDF が関連語 “fighter” を含まない場合は，アルゴリズム 1 に従い，まず “fighter” の Synset である “combatant, battler, belligerent, scrapper, fighter_aircraft, attack_aircraft, champion, hero, paladin” が関連語として利用される．それらも該当 RDF に含まれなかった場合は，$K(n_k, n_l)$ の値が次に大きい “aircraft” が新たに関連語となる．内容語リソース RDF が，これらすべての関連語を一度も含まない場合，$W(n)$ 値が “fighter” の “spitfire” の次に高い “fighter” が内容語となり，“fighter” に対しての $K(n_k, n_l)$ 値を計算し直し，以下同じ手順が続く．

この例では，Supermarine_Spitfire.rdf 内の Domain，Property，Range の三つ組に関連語 “fighter” を含むものが 14 組であり，行列 T は以下のように表わされる．

$$T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

この行列は，条件 $(\sum T_{1j} \geq \sum T_{0j}) \wedge (\sum T_{1j} \geq \sum T_{2j})$ を満たす．ここで，関連語を含むプロパティとして `dbpedia-owl:aircraftFighter` が抽出される．したがって，内容語をリソース，関連語を含むプロパティ `dbpedia-owl:aircraftFighter` を関係として，知識の抽出を行う．すなわち，この文章から抽出される知識は “**Supermarine_Spitfire に対して関係 `aircraftFighter` を持つ Domain または Range**” である．この時 SPARQL クエリは，

```
SELECT ?isValue WHERE {
?isValue dbpedia-owl:aircraftFighter
dbpedia:Supermarine_Spitfire }
```

または，

```
SELECT ?hasValue WHERE {
dbpedia:Supermarine_Spitfire
dbpedia-owl:aircraftFighter ?hasValue }
```

となる．これらのクエリはエンドポイントを通り，いくつかの結果を得る．ここで，表示のためにプロパティのラベルを求める．SPARQL クエリは以下の通りである．

```
SELECT ?hasValue WHERE {
dbpedia-owl:aircraftFighter rdfs:label ?hasValue
}
```

これにより，プロパティ `aircraftFighter` のラベル “aircraft fighter” を取得する．結果，以下の知識（抽出されたものの一部を掲載）がユーザに提示される．

subsidiary of Vickers-Armstrong). Mitchell continued to refine the design until his death from cancer in 1937, whereupon his colleague Joseph Smith became chief designer.[7] The Spitfire's elliptical wing had a thin cross-section, allowing a higher top speed than several contemporary fighters, including the Hawker Hurricane.[8] Speed was seen as essential to carry out the mission of home defence against enemy bombers.[6]

図 4　追加選択イメージ

aircraft fighter of No._452_Squadron_RAAF is Supermarine_Spitfire.
aircraft fighter of No._80_Wing_RAAF is Supermarine_Spitfire.
aircraft fighter of No._303_Polish_Fighter_Squadron is Supermarine_Spitfire.
aircraft fighter of No._453_Squadron_RAAF is Supermarine_Spitfire.

次に拡張情報として，“**aircraftFighter という関係を持つリソースの組**” を得る．この質問に対する SPARQL クエリは，

```
SELECT ?isValue ?hasValue WHERE {
?isValue dbpedia-owl:aircraftFighter ?hasValue }
```

となる．クエリはエンドポイントを通じて，以下の知識（抽出されたものの一部を掲載）を取得し，ユーザに提示される．

aircraft fighter of Syrian_Air_Force is Mikoyan_MiG-29.
aircraft fighter of Royal_Malaysian_Air_Force is Mikoyan_MiG-29.
aircraft fighter of SFR_Yugoslav_Air_Force is Mikoyan_MiG-29.
aircraft fighter of Serbian_Air_Force_and_Air_Defence is Mikoyan_MiG-29.

4.2　追加選択がある場合

追加選択がある場合，すなわち，Wikipedia の Supermarine_Spitfire の項目において，ユーザがある広域範囲（図 3 に示す反転部分）を選択した後，さらにその文章に含まれるある単語を選択した場合，その単語を関連語として拡張情報を抽出する．

例えば，選択された単語が “speed”（図 4 に示す反転部分）の場合，これが関連語となる．内容語は同様に “spitfire” である．この例では，Supermarine_Spitfire.rdf 内の Domain，Property，Range の三つ組に関連語 “speed” を含むものが 2 組であり，行列 T は以下のように表わされる．

$$T = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}$$

この行列は，条件 $(\sum T_{1j} \geq \sum T_{0j}) \wedge (\sum T_{1j} \geq \sum T_{2j})$ を満たす．ここで，関連語を含むプロパティとして `dbpprop:maxSpeedMain` および `dbpprop:maxSpeedAlt` が抽出される．したがって，内容語をリソース，関連語を含むプロパ

ティdbpprop:maxSpeedMain または dbpprop:maxSpeedAlt を関係として，知識の抽出を行う．

すなわち，この文章から抽出される知識は “**Supermarine_Spitfire に対して関係** dbpprop:maxSpeedMain **または** dbpprop:maxSpeedAlt **を持つ Domain または Range**” である．

プロパティdbpprop:maxSpeedMain について，上記の知識を抽出するための SPARQL クエリは，

```
SELECT ?hasValue WHERE {
dbpedia:Supermarine_Spitfire dbpprop:maxSpeedMain
?hasValue}
```

または，

```
SELECT ?isValue WHERE {
?isValue dbpprop:maxSpeedMain
dbpedia:Supermarine_Spitfire}
```

となる．クエリはエンドポイントを通じて以下の知識を取得し，ユーザに提示される．

max speed main of Supermarine_Spitfire is 22680.0.

次に拡張情報として，“**maxSpeedMain という関係を持つリソースの組**” を得る．この質問に対する SPARQL クエリは，

```
SELECT ?isValue ?hasValue WHERE {
?isValue dbpprop:maxSpeedMain ?hasValue}
```

と表わせる．このクエリもエンドポイントを通じて，結果を得る．この例では，結果が非常に多量であったことからカテゴリで絞り込む．

そのため，まず以下の SPARQL クエリにより，Supermarine_Spitfire の dcterms:subject カテゴリを取得する．

```
SELECT ?hasValue WHERE {
dbpedia:Supermarine_Spitfire
dcterms:subject ?hasValue}
```

結果，以下の七つのカテゴリを得る．

Category:1938_introductions
Category:Propeller_aircraft
Category:Single-engine_aircraft
Category:Carrier-based_aircraft
Category:British_fighter_aircraft_1930-1939
Category:Supermarine_aircraft
Category:Low_wing_aircraft

ここで，これらカテゴリ毎に絞り込み結果を取得する．例えば，*Category:Propeller_aircraft* について絞り込む場合，“*Category:Propeller_aircraft* に属すリソースが持つ dbpprop:maxSpeedMain の値” を取得する．従って，SPARQL クエリは次のようになる．

```
SELECT ?isValue ?hasValue WHERE {
?isValue dbpprop:maxSpeedMain ?hasValue.
?isValue dcterms:subject
<http://dbpedia.org/resource/
Category:Propeller_aircraft> }
```

クエリはエンドポイントを通じて，以下の知識（抽出されたものの一部を掲載）を取得し，ユーザに提示される．

max speed main of Hawker_Hurricaneis 340 mph.
max speed main of Macchi_C.200 is 504.0.
max speed main of Macchi_C.205 is 640.0.
max speed main of Aichi_D3A is 430.0.

プロパティdbpprop:maxSpeedAlt についても同様の手法で情報が拡張される．

（以上に示す知識取得結果は 2011 年 2 月 14 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．またユーザが広範囲選択後，その文章中に含まれる別の単語を選択することにより，よりユーザの意図を汲んだ情報拡張も提案した．

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．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 において内容語が属するクラス，カテゴリ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，追加選択された単語，文章の利用方法の検討が挙げられる．現在は追加選択は一語しか受け付けていないが，例えば追加選択されたものが文章だった場合，係り受け解析等を行うことにより，よりユーザの興味に沿った情報拡張

ができるような手法を検討している．また，追加選択された単語・文章が，最初に選択された範囲外であった場合，追加選択された単語・文章をどのように利用するのかということも課題の一つである．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．

文　献

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

[2] Berners-Lee, T. 2006. Linked data–design issues–. In http://www.w3.org/DesignIssues/LinkedData.html.

[3] Bizer, C.; Cyganiak, R.; and Heath, T. 2007. How to publish linked data on the web. In http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/.

[4] Miller, G. A. 1995. Wordnet: a lexical database for english. Commun. ACM 38:39-41.

[5] Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: a nucleus for a web of open data. In Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference, ISWC'07/ASWC'07, 722-735. Berlin, Heidelberg: Springer-Verlag.

[6] Ding, L.; Finin, T.; Joshi, A.; Pan, R.; Cost, R. S.; Peng, Y.; Reddivari, P.; Doshi, V.; and Sachs, J. 2004. Swoogle: a search and metadata engine for the semantic web. In Proceedings of the thirteenth ACM international conference on Information and knowledge management, CIKM '04, 652-659. New York, NY, USA: ACM.

[7] Oren, E.; Delbru, R.; Catasta, M.; Cyganiak, R.; Stenzhorn, H.; and Tummarello, G. 2008. Sindice.com; a document;oriented lookup index for open linked data. Int. J. Metadata Semant. Ontologies 3:37-52.

[8] Kobilarov, G.; Scott, T.; Raimond, Y.; Oliver, S.; Sizemore, C.; Smethurst, M.; Bizer, C.; and Lee, R. 2009. Media meets semantic web ? how the bbc uses dbpedia and linked data to make connections. In Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications, ESWC 2009 Heraklion, 723-737. Berlin, Heidelberg: Springer-Verlag.

[9] Waitelonis, J., and Sack, H. 2009. Towards exploratory video search using linked data. In Proceedings of the 2009 11th IEEE International Symposium on Multimedia, ISM'09, 540-545. Washington, DC, USA: IEEE Computer Society.

[10] Vallet, D.; Cantador, I.; and Jose, J. M. 2010. Exploiting external knowledge to improve video retrieval. In Proceedings of the international conference on Multimedia information retrieval, MIR '10, 101-110. New York, NY, USA: ACM.

[11] Haslhofer, B.; Momeni, E.; Gay, M.; and Simon, R. 2010. Augmenting europeana content with linked data resources. In Proceedings of the 6th International Conference on Semantic Systems, I-SEMANTICS '10, 40:1-40:3. New York, NY, USA: ACM.

[12] Aastrand, G.; Celebi, R.; and Sauermann, L. 2010. Using linked open data to bootstrap corporate knowledge manage- ment in the organik project. In Proceedings of the 6th Inter- national Conference on Semantic Systems, I-SEMANTICS'10, 18:1-18:8.. New York, NY, USA: ACM.

[13] Christian, B., and Christian, B. 2008. Dbpedia mobile: A location-enabled linked data browser. In 1stWorkshop about Linked Data on the Web.

[14] de León, A.; Saquicela, V.; Vilches, L. M.; Villazón- Terrazas, B.; Priyatna, F.; and Corcho, O. 2010. Geographical linked data: a spanish use case. In Proceedings of the 6th International Conference on Semantic Systems, I- SEMANTICS '10, 36:1-36:3. New York, NY, USA: ACM.

[15] Iwazume, M.; Kaneiwa, K.; Zettsu, K.; Nakanishi, T.; Kidawara, Y.; and Kiyoki, Y. 2008. Kc3 browser: semantic mash-up and link-free browsing. In Proceeding of the 17th international conference on World Wide Web, WWW'08, 1209-1210. New York, NY, USA: ACM.