

# 固有表現抽出を用いた文化財 Web コンテンツの LOD 化支援システムの開発

## A Creation Support System for Linked Open Data from Cultural Assets Web Contents Using Named Entity Recognition

似内 勇太<sup>1\*</sup> 奥野 拓<sup>2</sup>  
Yuta Nitanaï<sup>1</sup> Taku Okuno<sup>2</sup>

<sup>1</sup> 公立はこだて未来大学大学院

<sup>1</sup> Graduate School of Future University Hakodate

<sup>2</sup> 公立はこだて未来大学

<sup>2</sup> Future University Hakodate

**Abstract:** Recently, historical contents have been digitized. In addition, they have been planned to publish in the format which is available for secondary use. Text of historical Web contents includes useful information such as period, events and the relevant person. This research adopts the LOD and supports publishing of useful information in the text of Cultural Assets Web contents. This research proposes to extract Web resources and to recommend RDF properties using the named entity recognition when the publisher converts data in the text into the LOD. This research develops the proposed system as a WordPress plugin. The proposed system shows extraction results and RDF properties by a tree-shaped structure. Users can make corrections in extraction results using GUI. The evaluation experiment compared the proposed system and the trial system which excluded the proposed method. As a result, mistakes of people who used the proposed system were less than mistakes of people who used the trial system. Working time of people who used the proposed system were shorter than working time of people who used the trial system. However, the necessity of adding the predicate and improvement of extracting accuracy were proved by the result.

## 1 はじめに

近年、市史や文化財などの歴史コンテンツのデジタル化が活発に行われている。例えば北海道では、北海道の文化財情報を公開している「北海道文化資源データベース」[1]、道南の文化財や博物館の情報を公開している「道南ブロック博物館施設等連絡協議会ブログ」[2]、函館の地域史資料を公開している「函館市史デジタル版」[3]など様々な Web コンテンツが公開されている。さらに、道南では文化財の情報を地図上にマッピングする Web コンテンツ「道南文化財ウェブマップ」の構築が公立はこだて未来大学と共同で進められている。また、「道南文化財ウェブマップ」で蓄積した文化財の情報を LOD 化することが計画されている。

現在、様々な種類のデータが LOD 化され、二次利用

可能な形式で公開されている。しかし、LOD で公開されている情報は、Web ページの表形式のデータや CSV 形式などの構造化データから変換したものが多く、文章はリテラルで公開されている。そのため、文章中の情報と関連しているデータを結びつけることができない。

歴史 Web コンテンツの文章には年代や場所、人名などの有益な情報が含まれており、それらの情報をリソースとして公開し、関連しているデータと結びつけることは重要である。本研究では歴史 Web コンテンツを公開する学芸員を対象に、歴史 Web コンテンツの一つである文化財 Web コンテンツの文章に含まれるデータの LOD 化を支援するシステムを開発する。

\*連絡先：公立はこだて未来大学大学院  
〒041-0803 北海道函館市亀田中野町 116 番地 2  
E-mail: g2114025@fun.ac.jp

## 2 文章中のデータを LOD 化する課題

文章中のデータを LOD 化する場合に主に三点の課題がある。一つ目は、文章中にはリソースとなり得る様々な単語が含まれていることである。文章中のデータを LOD 化するには、LOD 作成者が文章を読み、内容を把握した後にリソースを抽出する必要がある。そのため、リソースの抽出に多大なコストを要する。文章中のデータを LOD 化する支援ツールとして DBpedia Spotlight[4] がある。DBpedia Spotlight は、DBpedia で公開されているリソースが文章に含まれているかマッチングを行い、含まれていた場合は DBpedia の URI を文章にアノテーションする。しかし、DBpedia Spotlight は DBpedia で公開されていないリソースを抽出することはできない。

二つ目は、プロパティの選定作業である。様々な語彙が定義されており、どのプロパティが最適か判断することは LOD に関する知識がなければ難しい。また、新規にプロパティを定義する場合、RDFS などの知識が必要になる。

三つ目は、主語・述語・目的語の対応付けである。文章は非構造化データであるため、文章ごとに出現するプロパティに規則性はない。そのため、文章中のデータを LOD 化する場合、主語・述語・目的語のモデリングを文章ごとに行う必要がある。

本研究では、文章中の人名・地名などの固有表現を抽出する自然言語処理技術である、固有表現抽出を用いてリソースの抽出とプロパティの推薦を行う。また、LOD に関する知識が無くとも手軽にモデリングが行えるシステムを開発することで文章中のデータの LOD 化を支援する。

## 3 関連研究

文章中のデータを Linked Data 化する研究として、地域メディアの記事コンテンツを Linked Data 化した長野らの研究がある [5]。長野らは Linked Data 化するために記事コンテンツから人物、組織、場所、製品、イベント名を人手により抽出した。また、Linked Data 化したデータを活用して横浜の観光施設に関する記事を時系列で閲覧できる「ハマ経クロニクル」というサービスを開発した。

長野らは、Linked Data 化する際の固有表現抽出の難しさについて考察している。固有表現抽出を行った場合、「平成 16 年度第 1 回横浜観光プロモーションフォーラム認定事業」や「名取市図書館どんぐり子ども図書館」など名称が非常に長いものは、抽出が難しいと述べられている。文化財 Web コンテンツの文章から固有

表現抽出で文化財名を抽出する場合も同様の問題が発生する可能性があるため、本研究では文化財名を辞書に追加することで対応する。

固有表現抽出を用いて文章を Linked Data 化した田代らの研究がある [6]。田代らは、実世界の出来事を表す情報を「事象」とし、事象の主題、動作、動作の対象主など 13 種類を事象属性と定義した。そして、教師あり学習アルゴリズムである「条件付き確率場」(Conditional Random Fields. 以下、CRF) を用いて固有表現抽出を行い、事象属性を推定した。さらに CRF の出力から文節情報と品詞情報を手がかりとしたヒューリスティックルールを用いて Twitter とニュースサイトの記事から事象を抽出した。その後、抽出した事象と事象属性をもとに作成した RDF から事象ネットワークを作成し、事象間の関係性を可視化した。本研究では、田代らの研究を参考に新たに文化財の属性を定義し、CRF を用いて固有表現抽出を行う。

RDF のモデリングを手軽に行えるようにするために、森田らは視覚的に RDF と RDFS を編集できるエディタ MR<sup>3</sup> を開発した [7]。RDF を記述する初期段階ではプロパティの定義を行う必要があるため、RDF と RDFS を頻繁に編集する。そのため、森田らは同一画面上で RDF と RDFS の整合性を保ちながら編集を行えるようにした。また、MR<sup>3</sup> は、RDF グラフを使用し、視覚的にデータ間のつながりを考慮しつつ編集作業が行えるようになっている。本研究では森田らの研究を参考に、視覚的に RDF のモデリングを行うことができる GUI ツールの構築を行う。

## 4 固有表現抽出を用いた LOD 化支援

### 4.1 リソースの抽出

固有表現抽出を行う手法として、ルールベースで抽出を行う手法と機械学習を用いて抽出を行う手法がある。

ルールベースで固有表現抽出を行う場合、形態素解析された文章に対して、固有表現抽出用の辞書やパターンマッチルを用意し、抽出する [8]。ルールベースは人手によりルールの追加や調整を行うことで、最初は精度を上げやすい。しかし、辞書の作成コストが非常に高く、十分な被覆率を保つことも難しい。そのため、未知語に対して対処する方法が別途必要になる [9]。

機械学習で固有表現抽出を行う場合、正しくラベル付けされた教師データを用意し、教師データをもとにルールを自動で生成することで抽出を行う。機械学習の場合、教師データを作成するコストがかかるが、未知語にもある程度対応できる利点がある [9]。

本研究では、文化財名・地名・年・時代・人名・戦争名・出来事を文化財属性として定義し、文化財 Web コンテンツからリソースを抽出する。年の文化財属性が抽出された場合、その年を含む一文を出来事とし、「何年にどのようなことが起きたか」が書いてある文を抽出する。文化財名は文化財 Web コンテンツのタイトルや見出しに書かれ、文章中では指示語や普通名詞で書かれることが多い。そのため、文化財名を十分に学習できない可能性がある。加えて、3章で述べたように、長い名称を抽出する際に起こる問題は文化財名を抽出する場合も同様に発生する。文化財名は、長野らが抽出しようとしたイベント名と違い、新規名称が追加されることは少ない。さらに、国や市に登録されている場合が多いため、ある程度網羅されている可能性が高い。そこで、本研究では、文章に含まれる文化財名と出来事の抽出をルールベースで行い、文化財名と出来事以外の文化財属性の抽出を田代らの研究を参考に CRF を用いて機械学習で行う。

## 4.2 プロパティの推薦

本研究では、固有表現抽出により抽出されたリソースに付与されている文化財属性をもとに、プロパティを推薦する。しかし、文化財属性の意味は広義であるため、文化財属性に対応したプロパティを推薦したとしても、RDF で具体的なリソースの意味を表現することはできない。文章に地名が含まれていた場合、文化財が発見された場所や作られた場所などを表している可能性があるが、文化財属性を地名として抽出しただけでは、どのような意味を持つか定められない。そこで、本研究では文化財属性をより詳細に表現した述語を、主語の文化財属性と目的語の文化財属性の組み合わせによりユーザに推薦する。例えば、主語が「文化財名」で、抽出された文化財属性が「年」の場合、「発見された年」と「作られた年」をユーザに推薦する。

述語に対応したプロパティを用いて RDF を生成する。プロパティには、広く一般的に用いられている三つの語彙と本研究で定義した語彙のプロパティを用いる。用いた語彙は、schema.org, Dublin Core Metadata Initiative Metadata Terms, Friend of a Friend (FOAF) の三つである。「発見された場所」や「作られた場所」など、既存の語彙で定義されていないプロパティを本研究で新たに定義する。文化財属性をクラス、述語をプロパティとして定義した。文化財属性の組み合わせとプロパティの対応関係を表 1 に示す。dc と dcterms は Dublin Core Metadata Initiative Metadata Terms, schema は schema.org, foaf は Friend of a Friend, cul は本研究で定義した語彙を表す接頭辞である。

表 1: 文化財属性の組み合わせと推薦するプロパティ

目的語の文化財属性	主語の文化財属性	述語	プロパティ
文化財名	文化財名	発見された場所	cul:discoverPlace
		所在地	schema:address
		-	rdfs:seeAlso
地名	文化財名	発見された場所	cul:discoverPlace
		作られた場所	cul:createPlace
		所在地	schema:address
	地名	地名の旧名	cul:oldPlaceName
	人名	生まれた場所	cul:birthPlace
	戦争名	戦地	cul:warPlace
	-	地名	cul:placeName
人名	文化財名	製作者	dcterms:creator
		発見者	cul:discoverer
		設計者	cul:designer
		寄贈者	schema:contributor
		使用者	cul:user
	人名	師	cul:master
	-	弟子	cul:pupil
-	人名	foaf:name	
戦争名	-	戦争名	cul:war
時代	文化財名	発見された時代	cul:discoverEra
		作られた時代	cul:createEra
		-	cul:era
年	文化財名	発見された年	cul:discoverDate
		作られた年	dcterms:date
	人名	生年	schema:birthDate
		没年	schema:deathDate
	戦争名	開戦年	cul:warStartDate
		終戦年	cul:warEndDate
	-	年	dc:date
出来事	-	出来事	schema:description

## 5 固有表現抽出実験

### 5.1 実験方法

北海道文化資源データベースで公開されている文化財の説明からリソースを抽出する実験を行った。北海道文化資源データベースは北海道庁が公開している Web サイトであり、データに信頼性があるため、実験に使用した。北海道文化資源データベースの文化財カテゴリの函館と道南圏のカテゴリから、文化財の説明文を Web スクレイピングにより収集した。文化財名をルールベースで抽出するために、同様のカテゴリから文化財名を収集し、形態素解析エンジンである MeCab の辞書に追加した。

本研究では CRF による処理を行うツールキットの CRF++[10] を使用して固有表現抽出を行った。教師

データを作成するために、収集した 670 文を、MeCab により形態素に分割した。また、形態素解析の結果の単語の品詞情報も学習データとして活用した。各形態素に対して文化財属性を IOB2 タグ [11] の形式でタグ付けし、教師データを作成した。作成した教師データと CRF++ を用いて抽出器を作成し、リソースの抽出を行った。

本研究では、交差検定を用いて抽出の正確性を検証した。教師データを 5 分割したうち、4 つを教師データとし、残りをテストデータとした。5 分割したデータの組み合わせを変えて検証を繰り返し、適合率と再現率から F 値の平均を求めることで評価を行った。

## 5.2 結果・考察

実験の結果から再現率は 60.5%、適合率は 88.4%、F 値は 71.8% となった。年代や時代、戦争名など文章中に多く含まれる語は高い確率で抽出できた。しかし、名字だけが書かれていた場合や「フランク・ロイド・ライト」など外国人の名前が書かれていた場合に、人名の抽出に失敗していた。名字だけの人名や外国人名が教師データにあまり含まれていないため、学習を十分に行えなかったことが原因であると考えられる。この問題に対して、人名辞書などを追加して対応する必要がある。

今回は、一つのサイトに対して実験を行ったため、書き方や文章内の表現など、ある程度の規則性があった。しかし、執筆者ごとに書き方や文章の癖など様々な規則性があり、より多くの規則性を学習する必要がある。そのため、複数の文化財 Web コンテンツから文章を収集し教師データを追加する必要がある。

## 6 WordPress プラグインの実装

本研究では、リソースの抽出とプロパティの推薦を行い、その結果を GUI で修正できる WordPress のプラグインを開発した。WordPress はオープンソースの CMS であり、世界の 22% を越える Web サイトで利用され、CMS ソフトウェアとしての世界シェアは約 60% となっている。提案システムを WordPress のプラグインとして開発することで、WordPress で公開している文化財 Web コンテンツに適用でき、文化財コンテンツを作成する作業と同時に、抽出と修正作業ができる利点がある。また、WordPress はセマンティックウェブや Web 標準などを意識して開発されており、カスタムフィールドを用いることで、簡単にメタデータを入力し蓄積することができる。

公開される文化財コンテンツは、文化財名が記事のタイトルとなり、一つの記事で一つの文化財が紹介される

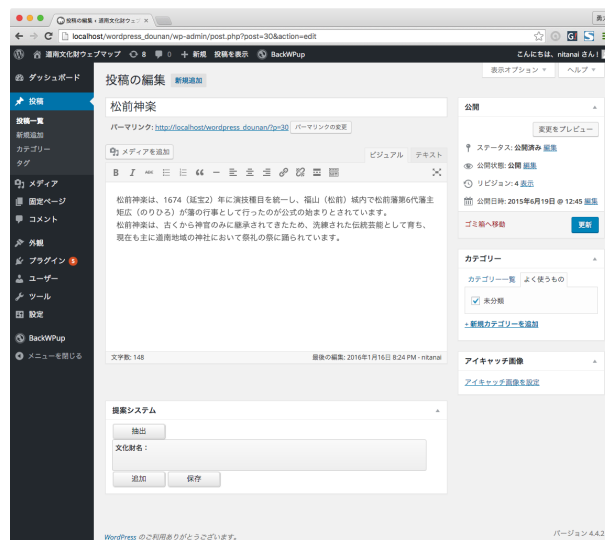


図 1: 文化財 Web コンテンツの投稿画面

ことを想定している。開発するプラグインは、リソースの抽出とプロパティの推薦の他に、リソースの主語と目的語を自由に変更できる編集機能と DBpedia Japanese のリソースとマッチングを行う機能を提供する。

ユーザが文化財のタイトルと説明文を入力した後に、抽出ボタンを押すとリソースが抽出される。道南の文化財の一つである松前神楽を投稿した画面は図 1 のようになる。図 1 の下部にある抽出ボタンを押した場合、抽出結果を図 2 のようにツリー構造で表す。タイトルの文化財名がルートノードとなり、抽出されたリソースはルートノードの文化財を説明する内容となる。また、親ノードが主語、子ノードが目的語を表している。出来事は、出来事が起きた年を親ノードとして子ノードに追加される。ノードの内容は上から、文化財属性、述語、抽出されたリソース、DBpedia Japanese のリソースである。例えば、「松前神楽の作られた年は 1674 年である」という文を提案システムで表現した場合、図 2 の「1674」と書かれているノードになる。

抽出後は文化財のタイトルがルートノードになるが、生年や没年などのプロパティの主語は文章中に登場した人物となることがあり、主語が文化財名とは限らない。そこで、文化財に関係した人物の情報なども適切に表現できるように、抽出されたリソースの主語を文化財名以外に自由に変更できる機能を実装した。子ノードに設定したいノードを、親ノードに設定したいノードの右下に移動させることでツリーの親子関係を変更できる。

リソースの抽出に漏れがあった場合、図 2 の下にある追加ボタンを押すことでルートノードの下にノードが追加される。逆に不要なリソースが抽出された場合、ノードの右上にある削除ボタンを押すことで削除できる。

図 2: WordPress プラグインの抽出結果の例

ユーザが修正を完了し、記事を公開したタイミングと同時に RDF を生成する。生成のアルゴリズムについて述べる。最初にルートノードを「http://ホスト名/cul/抽出されたリソース」の形式で URI に変換する。ホスト名はプラグインを適用した文化財 Web コンテンツのホスト名である。ノードの内容を RDF に変換する際、ノードが持つ文化財属性によって変換方式を変える。文化財属性が出来事以外の場合、抽出結果を URI で記述し、表 1 の述語に対応したプロパティを用いて変換する。文化財属性が出来事であった場合は、リテラルとして扱う。リテラルを主語にすることは出来ないため、出来事に対して子ノードがあった場合は、削除する。

DBpedia Japanese の URI が入力されていた場合、または子ノードがあった場合は、クラスを用いて RDF に変換する。DBpedia Japanese の URI と子ノードの情報はクラス内に記述する。最後に抽出もとの文章全文を schema:description のプロパティを用いてルートノードの下に追加する。松前神楽の文章の抽出結果である図 2 のツリー構造から、RDF を生成した例を図 3 に示す。

## 7 提案システムの評価実験

### 7.1 実験方法

提案手法の有効性を評価するため、被験者に提案システムと実験システムを操作してもらい、自動生成された RDF の正確性と操作時間を比較した。生成された RDF にどの程度抽出結果の修正作業の誤りがあった

```
<rdf:RDF
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dc = "http://purl.org/dc/elements/1.1/"
  xmlns:schema = "http://schema.org/"
  xmlns:foaf = "http://xmlns.com/foaf/0.1/"
  xmlns:cul = "http://donan-museums.jp/property/"

  <rdf:Description rdf:about="http://localhost/cul/松前神楽">
    <dc:title>松前神楽</dc:title>

    <cul:Year rdf:resource="http://localhost/cul/1674">
      <cul:createYear>1674</cul:createYear>
      <rdfs:seeAlso rdf:resource="http://ja.dbpedia.org/page/1674年" />
      <schema:description>
        松前神楽は、1674 (延宝2) 年に演技種目を統一し、
        福山 (松前) 城内で松前藩第6代藩主矩広 (のりひろ) が藩の行事とし
        て行ったのが公式の始まりとされています。
      </schema:description>
    </cul:Year>

    <cul:Person rdf:resource="http://localhost/cul/矩広" >
      <dc:terms:creator>矩広</dc:terms:creator>
      <rdfs:seeAlso rdf:resource="http://ja.dbpedia.org/page/松前矩広" />
    </cul:Person>

    <schema:description>
      松前神楽は、1674 (延宝2) 年に演技種目を統一し、福山 (松前) 城内で松前藩第6代藩
      主矩広 (のりひろ) が藩の行事として行ったのが公式の始まりとされています。松前神楽
      は、古くから神官のみに継承されてきたため、洗練された伝統芸能として育ち、現在も
      主に道南地域の神社において祭礼の祭に踊られています。
    </schema:description>

  </rdf:Description>
</rdf:RDF>
```

図 3: 生成された RDF の例

かによって、提案システムの有効性を評価する。加えて、誤りを分類し、誤りの原因について考察する。また、計測した各システムの操作時間を比較することで作成コストの削減について評価する。実験では、6章で説明した提案システムと、提案システムからリソース抽出とプロパティの推薦機能を除いた実験システムを用いた。

実験システムでは、抽出ボタンを押した場合、記事のタイトルの文化財名をルートノードに表示する。また、文化財名と同一の URI が DBpedia Japanese で公開されていた場合、ルートノードに表示する。

実験に使用する文化財 Web コンテンツの題材として、文化財属性の変更、述語の選択など全ての操作が発生するように三種類の文章を用意した。被験者には、用意した全ての題材に対して、提案システムと実験システムのどちらかを使用し操作を行ってもらった。

被験者にシステムの説明と抽出して欲しい文化財属性を説明し、システムの操作方法についてわからない部分があった場合のみ、質問を受け付けた。

提案システムと実験システムの両方とも、タイトルと説明文が入力されている状態から操作を始める。提案システムを操作する被験者は、まず記事のタイトルと説明文の内容を確認し、抽出ボタンを押す。被験者は表示された抽出結果に対して、文化財属性の変更、述語の選択、DBpedia Japanese の URI の確認、ノードの追加・削除・移動を行う。実験システムを操作する被験者は、実験システムにはリソース抽出機能が無いため、タイトルと説明文の内容を確認した後、どの単語をリソースとするか選定し、ノードの追加を行う。実



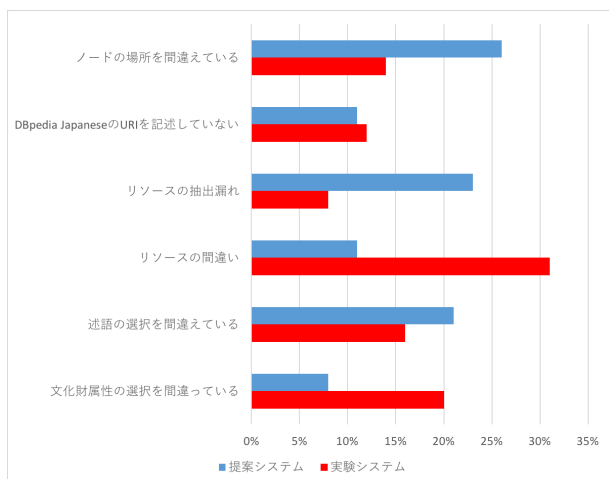


図 4: 修正作業の誤りの内容

験システムでは、プロパティの推薦機能がないため、被験者は表 1 の述語全てから一つを選択する。それ以外は提案システムと同様の操作を行う。操作後、被験者に対してアンケートを取った。アンケートでは、被験者に LOD や WordPress についての知識とシステムを操作して気づいた点について質問した。

## 7.2 結果・考察

被験者 10 名のうち、5 名に提案システム、残りの 5 名に実験システムを操作してもらった。修正作業の誤りの数は、提案システムの方が 36 個、実験システムの方が 52 個となった。提案システムの方が、実験システムと比べ誤りが少なく、正確に RDF 化された。誤りの分類とその割合を図 4 に示す。

提案システムの誤りの内容で一番数が多かったのが、ノードの移動ミスであった。これは、ある被験者が出来事の下に人名などのリソースを入力していたため、この修正作業の誤りが多くなった。ノードによる主語・述語・目的の表現は説明したが、一人の被験者の操作結果がこの内容の誤りの 6 割を占めた。

次に多かった提案システムの誤りの内容はリソースの抽出漏れである。固有表現抽出で抽出できたリソースに関しては問題無かったが、抽出できなかったリソースを手作業で追加する作業はあまり行われなかった。逆に実験システムは、提案システムと比べ抽出の漏れは少なかった。これは被験者の文章への理解度が原因だと考えられる。実験システムでは文章を十分に読む必要があるが、提案システムの場合、リソースが自動で抽出されるため、抽出されたリソースが文章中でどのような内容で記述されているか確認するだけで済む。そのため、文章全体を十分に読まなくなり、固有表現抽出で漏れてしまったリソースに気づきにくくなると考

えられる。今回は、事前に文化財の説明文を入力した状態で実験を始めた。しかし、文章の執筆者と抽出結果を修正する人物は本来は同一となる。同一人物であれば、リソースの修正作業を行う前から文章の内容を把握しているため、リソースの追加の漏れは少なくなると考えられる。

提案システムに比べ実験システムで多かった修正作業の誤りは、リソースの抽出間違いであった。提案システムでは人名だけが正しく抽出されていたが、実験システムでは「松前藩第 6 代藩主矩広」のように人名と役職が一つのリソースとして抽出されていることがあった。また、出来事が文章ではなく「建設」とだけ書かれている場合があり、リソースとリテラルの抽出はあまり正確に行われなかった。このことからリソースの自動抽出により、抽出の誤りを防ぐことができたと考えられる。

提案システムと実験システムの両方で述語の選択を間違えていることがあった。これは述語が十分に定義できていなかったことが原因だと考えられる。選択で間違っていた部分は、抽出されたリソースに対して適切な述語がないノードであった。また、アンケートで「実験の最中に選択肢にない述語に気がきましたら記述してください」と質問したところ、「焼失した年」や「和暦」が無いという回答があった。このことから、再度述語の調査と定義を行う必要性が判明した。

各システムの操作時間の平均は、提案システムは 15 分 09 秒、実験システムは 29 分 05 秒となった。提案システムの方が実験システムと比べ 13 分 54 秒短縮され、作業時間を 47.8%削減することができた。このことから、提案システムにより文章中のデータを RDF 化するコストを削減できたと考えられる。

## 8 まとめ

本研究では、文化財 Web コンテンツを対象に、文章中のデータの LOD 化を支援するシステムを構築した。文章中のデータを LOD 化する際に課題となる、リソースの抽出とプロパティの選定のコストを削減するために、固有表現抽出を用いたリソースの抽出とプロパティの推薦を提案した。そして、提案手法をシステム化した WordPress プラグインの開発を行った。リソースの抽出とプロパティの推薦の機能を除いた実験システムを用いて、提案システムと比較実験を行った。検証の結果、提案手法を適用しなかったシステムと比較し、修正作業の操作誤りが少なく RDF 化できることが示唆された。また、操作時間を比較した結果、実験システムに比べ提案システムの操作時間は半分程度となり、RDF 化するコストを削減できた。今後は、抽出精度向上と述語の追加を行う。

## 参考文献

- [1] 北海道文化資源データベース,  
<http://www.northerncross.co.jp/bunkashigen/>
- [2] 道南ブロック博物館施設等連絡協議会ブログ,  
<http://dounan.exblog.jp/>
- [3] 『函館市史』デジタル版,  
[http://archives.c.fun.ac.jp/hakodateshishi/shishi\\_index.htm](http://archives.c.fun.ac.jp/hakodateshishi/shishi_index.htm)
- [4] Pablo N. Mendes, Max Jakob, Andrs Garca-Silva, Christian Bizer:DBpedia Spotlight: Shedding Light on the Web of Documents, Proceedings of the 7th International Conference on Semantic Systems, pp.1-8 (2011)
- [5] 長野伸一, 川村隆浩, 小林巖生, 杉浦裕樹: 地位メディア情報を活用した Linked Data サービスの試作評価, 人工知能学会全国大会, pp.1-2 (2014)
- [6] 田代和浩, 王冕, 越川兼地, 西村悟史 他: Linked Data を用いたソーシャルメディア×マスメディアの比較実験, 人工知能学会全国大会, pp.1-4 (2013)
- [7] Morita Takashi, Izumi Noriaki, Fukuta Naoki, Yamaguchi Takahiro, A Graphical RDF-Based Meta-Model Management Tool, IEICE transactions on information and systems E89-D(4), pp.1368-1377 (2006)
- [8] 竹元義美, 福島俊一, 山田洋志: 辞書およびパターンマッチルールの増強と品質強化に基づく日本語固有表現抽出, 情報処理学会論文誌, pp.1580-1591 (2001)
- [9] 福島健一, 鍛冶伸裕, 喜連川優: コーパスからの固有表現辞書の自動構築, 知識ベースシステム研究会, pp.19-24 (2007)
- [10] 工藤拓: CRF++: Yet Another CRF toolkit,  
<http://crfpp.googlecode.com/svn/trunk/doc/index.html>
- [11] Erik F. Tjong Kim and Jorn Veenstra: Representing Text Chunks, In Proceedings of the 9th European Chapter of the Association for Computational Linguistics, pp. 173179 (1999)