

固有表現抽出を用いた歴史コンテンツの LOD 化支援

似内 勇太

公立はこだて未来大学大学院
システム情報科学研究科

奥野 拓

公立はこだて未来大学
システム情報科学部

近年、歴史コンテンツは書籍での公開だけではなく、Web 上でも発信されており、二次利用可能な形式で公開することも検討されている。歴史コンテンツの文章には年代や出来事、関連のある人物などの有益な情報が含まれている。本研究では、二次利用可能な形式として LOD を採用し、文章に含まれる有益な情報の公開を支援する。固有表現抽出を用いて、文章に含まれるデータを 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を支援する方法を提案する。実験では歴史コンテンツの一つである文化財コンテンツの文章から固有表現抽出を行い、抽出精度の評価を行う。

Support for the Creation of Linked Open Data from Historical Contents Using Named Entity Recognition

Yuta Nitandai

Graduate School of Systems
Information Science
Future University Hakodate

Taku Okuno

School of Systems
Information Science
Future University Hakodate

Recently, historical contents are published on the Web as well as books. In addition, they are considered to be published in the format which is available for secondary use. Text of historical contents includes useful information such as age, events and the relevant person. This research adopts the LOD and supports publishing of useful information in the text. This research proposes to extract resources and to recommend RDF properties using named entity recognition when the publisher converts data in the text to the LOD. Extraction accuracy of named entity recognition is evaluated by the experiments which extract named entities from the text of cultural heritage contents.

1. 歴史コンテンツの現状

近年、市史や文化財などの歴史コンテンツは書籍での公開だけではなく、Web 上での発信も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発信している「北海道文化資源データベース」、道南の文化財や博物館の情報を発信している「道南ブロック博物館施設等連絡協議会ブログ」、函館の地域史資料を発信している「函館市史デジタル版」など様々な Web サイトが公開されている[1][2][3]。

歴史コンテンツのデータを二次利用可能な形式で公開することが検討されている[2][3]。現在、Web API などの技術を用いて、様々なデータが二次利用可能な形式で公開されている。しかし、それらのデータは、Web ページの表形式データを CSV 形式などの構造化データに変換したものが多く、文章に含まれる非構造化データはあまり公開されていない。歴史コンテンツの文章には年

代や出来事、関連のある人物などの有益な情報が含まれており、それらのデータは二次利用可能な形式で公開する上では重要なデータである。そこで、本研究では歴史コンテンツの文章に含まれるデータを、二次利用可能な形式で公開することを促進するために支援を行う。

2. Linked Open Data

本研究では、二次利用可能な形式として、Web 上にデータを公開する仕組みである LOD (Linked Open Data) を用いる。Web 上のデータを LOD 化することで、事物に関するデータや事物同士の関係を機械が処理しやすい形式で公開することができる。そのため、アプリケーション開発への活用などデータの二次利用が期待できる。また、LOD 化したデータを Web 上で共有し、相互に関連付けることができるため、新たな知識を発見することができ、データの有用性が高まる。

LOD は標準形式の一つとして RDF (Resource Description Framework) を採用している。RDF は存在しているものの全てをリソースとして扱う。RDF は主語・述語・目的語の三つの要素でリソースの関係を表現する。主語はリソース、述語は主語と目的語の関係、目的語はリテラルや他のリソースを表す。述語はプロパティと呼ばれ、RDFS (RDF Schema) を用いて別途定義し、URI で参照可能にする。RDF では、リソースを記述する際に、クラスを用いることで、リソースの分類を表現することができる。RDF は URI を用いてリソースを表現するため「ピリカ遺跡」と「美利河遺跡」のように文化財名に表記揺れがある場合や町名の「森」と人名の「森」がある場合でも、一意に定めることができる。以上の理由から、歴史コンテンツを公開する際の二次利用可能な形式として LOD を採用した。本研究では歴史コンテンツの一つである文化財コンテンツの文章を対象に LOD 化支援を行う。

3. 文章中のデータを LOD 化する課題

構造化データを LOD 化する場合、LOD 作成者が構造化データに含まれるデータから公開するデータの項目を決定し、主語と目的語の関係を決定する。その後、データ項目に適したプロパティを選定する。公開するデータを RDF で記述することで LOD 化することができる。文章に含まれるデータを LOD 化する上で主に二点の課題がある。

一つ目は、文章に含まれるリソースとなり得る様々な単語を手で抽出する必要があることである。文章は非構造化データであるため、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。文章からリソースを抽出するツールとして DBpedia Spotlight が公開されている [4]。DBpedia Spotlight は Wikipedia の情報を LOD 化した 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う。文章に既に公開されているリソースが含まれていた場合、図 1 の下線が引かれた部分のように文章に DBpedia の URI をアノテーションする。これにより、リソースの関連情報へのリンクをたどることができるなどの利点がある。しかし、このようなツールを用いて文章からリソースを抽出したとしても、主語と目的語の関係の対応付けやプロパティの選定、RDF の記述は人手で行う必要がある。

二つ目は、構造化データの LOD 化と比べ、プロパティの選定作業が多くなることである。LOD 化する際に、広く一般的に用いられているプロパティを利用することで、アプリケーション内で



図 1 DBpedia Spotlight のデモ画面

Fig.1 The demonstration of DBpedia Spotlight.

RDF データのプロパティを修正することなく、同じプロパティを使用している LOD と連携することができる。そのため、既存のプロパティを調べた上で、そのプロパティを使用するか、新規にプロパティを定義するかを決める必要がある。しかし、プロパティは様々な組織で公開されており、どのプロパティを用いるのが最適かを判断することが手間になってしまう。

本研究では、固有表現抽出を用いて上記の課題を解決する。固有表現抽出とは、文章から人名・地名・組織名などの固有名詞を抽出する自然言語処理の技術である。

4. 関連研究

文章に含まれるデータを LOD 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 化するために人物、組織、場所、製品、イベントの五つの情報を記事コンテンツから人手で抽出した。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」と呼ばれる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。地域メディアに対して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が存在し、抽出が難しいことが述べられている。歴史コンテンツの文章から固有表現抽出を行う場合も同様の問題が発生する可能性があるため、対処する必要がある。

固有表現抽出を用いて文章を LOD 化する関連研究として、ソーシャルメディアとマスメディアから固有表現抽出を行い、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、事象間の意味を表現する主題、動作と動作の対象主などを事象属性

と定義した。固有表現抽出を用いて、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 データから事象ネットワークを作成し、事象と事象の関係性を可視化した。これにより、各メディアでどのような観点から意見が主張されているかなど論点の比較ポイントが提示できるようになった。本研究では、田代らの研究を参考に固有表現抽出を行う。

5. 固有表現抽出を用いた LOD 化支援

本研究では、リソースの抽出とプロパティの推薦を行い、文化財コンテンツの LOD 化を支援する。

5.1 リソースの抽出

固有表現抽出を用いて、文章に含まれるリソースを特定し、抽出する。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二つの手法がある。

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ールを用意し、固有表現の有無を判別する。ルールベースは人手で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、ルールベースは対象とする固有表現の種類が多くなるにつれて、抽出精度は伸びなくなる。また、未知語に対応できない欠点がある。

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作成することで抽出を行う。機械学習は教師データを作成するコストがかかるが、ルールベースに比べ、未知語にもある程度対応できる利点がある。

本研究では、文章に含まれる文化財名の抽出をルールベースで行い、文化財名以外の固有表現は機械学習を用いて抽出する。機械学習を用いた固有表現抽出を行うために、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)を用いる。抽出するリソースを歴史属性として定義する。定義した歴史属性を表 1 に示す。本研究では、歴史コンテンツの文章をもとに教師データを作成し、機械学習を行うことで文章に含まれるリソースを抽出する抽出器を作成する。

5.2 プロパティの推薦

抽出の結果にはリソースに歴史属性が付与されて表示される。本研究では、抽出されたリソースに付与されている歴史属性をもとに、プロパティを推薦する。提案システムでは、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

表 1 歴史属性一覧

Table 1 The list of historical properties.

属性名	例
文化財名	松前祇園ばやし
地名	松前町
年	1789
時代	縄文時代
人物	太刀川善吉
戦争名	箱館戦争

語彙のプロパティを推薦する。一つ目は、検索の結果に詳細な情報を反映するために必要な構造化データのフォーマットを標準化した schema.org である[7]。二つ目は、Web や文書の作者、タイトルなどの書誌情報をまとめた Dublin Core Metadata Initiative Metadata Terms である[8]。三つ目は、人物に関する情報をまとめた Friend of a Friend (FOAF) である[9]。これらの語彙で定義されていない、文化財を表現するプロパティを本研究で新たに定義する。

提案システムは、固有表現抽出の結果に付与されている歴史属性をユーザに提示する。歴史属性に対応するプロパティを表 2 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。

歴史属性に対応するプロパティの意味は広義である。そのため、文化財コンテンツに含まれる歴史属性の内容を、より詳細に説明することが望ましい。本研究では、その内容を説明するプロパティを、補足情報としてユーザに推薦する。補足情報とプロパティの対応関係を表 3 に示す。補足情報の推薦の例として、主語が文化財名で、抽出された歴史属性が「年」の場合、補足情報は「発見された年」と「作られた年」を推薦する。ユーザが補足情報を選択した場合、システムは補足情報に対応するプロパティを用いて RDF データを作成する。補足情報を選択しなかった場合、上記で述べた通りシステムは歴史属性に対応するプロパティを用いて RDF データを作成する。

5.3 提案システム

本研究では、固有表現抽出を行う抽出器を作成した後に、リソースの抽出とプロパティを推薦する WordPress のプラグインを開発する。WordPress はオープンソースの CMS であり、セマンティック Web や Web 標準などを意識して開発されている[10]。WordPress では入力項目を柔軟に設定でき、Web ページの内容を構造化できるカスタムフィールドと呼ばれる機能を提供してい

表2 歴史属性とプロパティの対応関係
Table 2 The correspondence of historical properties to RDF properties.

歴史属性	プロパティ
文化財名	<code>rdfs:seeAlso</code>
地名	<code>cul:address</code>
年	<code>dc:date</code>
時代	<code>cul:era</code>
人物	<code>foaf:name</code>
戦争名	<code>cul:war</code>

表3 補足情報とプロパティの対応関係
Table 3 The correspondence of supplementary information to RDF properties.

歴史属性	推薦する補足情報	プロパティ
文化財名	発見された場所	<code>cul:discoverPlace</code>
	所在地	<code>schema:address</code>
地名	作られた場所	<code>cul:createPlace</code>
	地名の旧名	<code>cul:oldNamePlace</code>
	生まれた場所	<code>cul:birthPlace</code>
	戦地	<code>cul:warPlace</code>
年	作られた年	<code>dcterms:date</code>
	生年	<code>foaf:birthday</code>
	没年	<code>schema:deathDate</code>
	発見された年	<code>cul:discoverDate</code>
	開戦年	<code>cul:warStartDate</code>
	終戦年	<code>cul:warEndDate</code>
時代	発見された時代	<code>cul:discoverEra</code>
	作られた時代	<code>cul:createEra</code>
人名	製作者	<code>dcterms:creator</code>
	発見者	<code>cul:discoverer</code>
	設計者	<code>cul:designer</code>
	寄贈者	<code>schema:contributor</code>
	使用者	<code>cul:user</code>
	師	<code>cul:master</code>
	弟子	<code>cul:pupil</code>

る。また、カスタムフィールドを用いて構造化した情報を RDF データに変換し、公開することが可能である。提案システムをプラグインとして開発することで、WordPress を使用して文化財コンテンツを作成している途中で、文化財の説明文からリソースを抽出し、構造化することができる利点がある。

公開される文化財コンテンツの内容は、文化財名が記事の主題で、一つの記事で一つの文化財が紹介される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、リソースの主語を変更する機能と他のリソースとのマッチング機能を提供する。

抽出が完了した段階では、記事の主題を主語として表示する。しかし、生年や没年などのプロパティの主語は人物であり、主語が文化財名とは限らない。提案システムでは、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ようにする。リソースの主語が主題の文化財名以外の場合、クラスを用いて記述する。クラスを用いることで主語をより細かく分類することができる。本研究で定義したクラスを表4に示す。例として、人物の生年や没年が文章に含まれていた場合、システムで「`cul:Person`」クラスを使用し、人物を主語として、生年や没年をRDFで記述する。

抽出されたリソースがDBpediaで公開されているか、自動でマッチングを行う。RDFのクエリ言語であるSPARQLを用いてDBpediaのRDFデータにアクセスし、リソースの有無を確認する。リソースが存在していた場合、マッチング結果をユーザに提示する。

開発するプラグインの文化財コンテンツ投稿画面を図2に示す。また、北海道茅部郡森町の指定文化財である「行幸柳」についての記事が投稿された場合を想定した、プラグインの使用手順を以下に示す。

1. ユーザは通常の記事の投稿と同様に記事の主題と文化財の説明文を入力する。
2. ユーザは入力が終わった後、説明文の入力フォームの下部にある抽出ボタンを押す。
3. システムは説明文に対して固有表現抽出を行う。抽出されたリソースがDBpediaで公開されているか、自動でマッチングを行う。
4. 図2の抽出ボタンの下部に、3の結果を図3のようにユーザに提示する。抽出結果を四角のセクションで分け、主語と目的語の関係をツリー構造で表現する。セクション内に歴史属性、抽出された固有表現、DBpediaとのマッチング結果、補足情報を表示する。
5. ユーザは提示された結果から、抽出の誤りの修正と補足情報の選択を行い、記事を投稿する。
6. 記事が投稿されると同時にシステム側でデータの整形を行い、リソースをURIで記述することでRDF化する。

「行幸柳」の説明文をRDF/XMLに変換した例を図4に示す。文化財情報を「`cul:CulturalAssets`」で表現する。「行幸柳」を「`http://ドメイン名/パス/文化財名`」の規則でURI化し`rdf:about`属性

表 4 主語とクラスの対応関係

Tabel 4 The correspondence of subjects to classes.

主語	クラス
文化財名	cul:CulturalAssets
地名	cul:Area
時代	cul:Era
人物	cul:Person
戦争名	cul:War

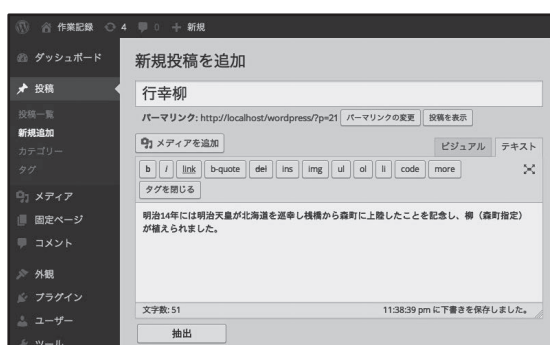


図2 文化財コンテンツ投稿画面

Figure 2 The post screen of cultural heritage contents.

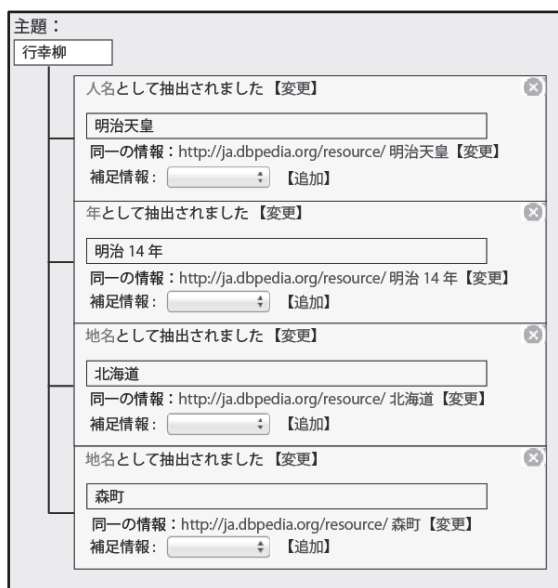


図3 抽出とマッチング結果の表示画面

Figure 3 The display screen of extraction and matching results.

で表現する．リソースの名前を「dc:title」で表現する．その後、ユーザが修正した抽出結果とマッチング結果をシステムが記述する．年の値が単調増加する西暦の方が二次利用可能な形式に適しているため、明治14年のように和暦で年が記述されていた場合、RDFデータにする際にシステムで西暦に変換する．DBpediaにリソースが存在していた場合は、リソースの関連情報を示す「rdfs:seeAlso」を用いて記述する．主語が文化

```
<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:cul = "http://cul-museums.jp/property/">

  <cul:CulturalAssets rdf:about=
    "http://cul-museums.jp/culture/行幸柳">
    <dc:title>行幸柳</dc:title>
    <rdfs:seeAlso rdf:resource="http://ja.dbpedia
.org/resource/明治天皇" dc:title=明治天皇/>
    <cul:createDate>1884</cul:createDate>
    <cul:createPlace>
      <cul:Area>
        <rdfs:seeAlso rdf:resource=
          "http://ja.dbpedia.org/resource/森町(北海道)" />
        <rdfs:label>森町(北海道)</rdfs:label>
      </cul:Area>
    </cul:createPlace>
  </cul:CulturalAssets >
</rdf:RDF>
```

図4 生成されるRDFデータ

Figure 4 The generated RDF data.

財名以外の場合と、ユーザが補足情報を選択し、DBpediaにリソースが存在していた場合は、図4の「cul:Area」要素のように表4で定義したクラスを用いて詳細にリソースを記述する．

6. 固有表現抽出の予備実験

予備実験として北海道文化資源データベースの文章に含まれるリソースを抽出する実験を行った．実験では教師データを作成し、条件付き確率場による処理を行うツールキットの CRF++ を使用して固有表現抽出を行った[11]．

北海道文化資源データベースの文化財カテゴリから、無作為に抽出した 411 文をもとに教師データを作成した．教師データに変換するために、抽出した文章を、形態素解析エンジンである MeCab を用いて形態素に分割した[12]．形態素に分割した後に、各形態素に対して歴史属性のタグを付加し、教師データを作成した．付加するタグの形式は IOB2 タグを用いた．IOB2 タグのラベルである、B(begin)は固有表現の開始時点、I(inside)は固有表現の内部、O(outside)は固有表現以外の形態素を表現する．また、IOB2 タグに属性情報を付与し、どの属性に分類されるかを定義する．

4章で述べた、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際にも発生する．文化財コンテンツは Web ページのタイトルや見出し

に文化財名が書かれ、文章中では文化財名は指示語や普通名詞で書かれることが多く、文化財名を十分に学習できない。それに加え、「求福山山車の人形その他付属品」や「旧笹浪家住宅主屋及び土蔵」など長い文化財名があり、機械学習を行っても正確に抽出できない可能性がある。本研究では、北海道文化資源データベースで公開されている文化財名を MeCab の辞書に追加することで対処した。

本研究では、F 値を用いて抽出精度の評価を行った。全ての固有表現のうち、抽出した固有表現の割合を示す適合率と、抽出した固有表現のうち、正しく抽出した固有表現の割合を示す再現率から F 値を算出する。評価に使用するためのテストデータは北海道文化資源データベースから教師データに含まれていない 36 文をランダムに抽出した。抽出した文章に対して MeCab を用いて形態素に分割することでテストデータを作成した。

7. 実験結果・考察

実験の結果から再現率は 48.9%、適合率は 87.2%、F 値は 62.6% となった。抽出の結果、文化財名以外に関しては、抽出で漏れている語が多いことが判明した。今回の実験では、411 文という少ない数の教師データで抽出器を作成したため、学習が十分に行われず、再現率が低くなった可能性がある。今後は、抽出精度の向上のため、北海道文化資源データベース以外の Web サイトからも文章を収集し、教師データを増やす必要がある。

8. まとめ

本研究では、歴史コンテンツの文章に含まれるデータの LOD 化促進のために、固有表現抽出を用いた文章の LOD 化支援について提案した。文章に含まれるデータを 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作業に対して固有表現抽出を用いて支援するシステムを提案した。提案システムはユーザが、歴史コンテンツである文化財コンテンツを Web に公開する際に、入力した文化財の説明文に対して固有表現抽出を行い、リソースの抽出とプロパティの推薦を行う。文化財コンテンツに対して抽出の予備実験を行った結果、抽出精度は 62.6% となった。今後、抽出精度の向上と提案したシステムの開発を行い、LOD 化支援の方法について有効性の評価を行う。

参考文献

- 1) 北海道：北海道文化資源データベース，入手先〈<http://www.northerncross.co.jp/bunkashigen/>〉（参照2014-09-11）。
- 2) 道南ブロック博物館施設等連絡協議会：道南ブロック博物館施設等連絡協議会ブログ，入手先〈<http://dounan.exblog.jp/>〉（参照2014-09-11）。
- 3) 函館市中央図書館：函館市史デジタル版，入手先〈http://www.lib-hkd.jp/hensan/hakodateshishi/shishi_index.htm〉（参照2014-09-11）。
- 4) DBpedia Spotlight，入手先〈<https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki>〉（参照2014-09-11）。
- 5) 長野伸一，川村隆浩，小林巖生ほか：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，越川兼地ほか：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) Schema.org initiative：schema.org，入手先〈<http://schema.org/>〉（参照2014-09-11）。
- 8) Dublin Core Metadata Initiative：Dublin Core Metadata Initiative (DCMI)，入手先〈<http://dublincore.org/>〉（参照2014-09-11）。
- 9) Dan Brickley and Libby Miller：The FOAF Project，入手先〈<http://www.foaf-project.org/>〉（参照2014-09-11）。
- 10) WordPress Foundation：WordPress › Blog Tool, Publishing Platform, and CMS，入手先〈<https://wordpress.org/>〉（参照2014-09-11）。
- 11) 工藤拓：CRF++：Yet Another CRF toolkit，入手先〈<http://crfpp.googlecode.com/svn/trunk/doc/index.html?source=navbar>〉（参照2014-09-11）。
- 12) 工藤拓：MeCab：Yet Another Part-of-Speech and Morphological Analyzer，入手先〈<http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html#thanks>〉（参照2014-09-11）。