

非構造化記述を含む文化財書誌に対する
Linked Data化手法の開発
—古典籍書誌データへの適用とその評価—

2018年3月

佐賀大学大学院工学系研究科
システム創成科学専攻

吉賀 夏子

**A Method for Converting Bibliographical
Notes into Linked Data – Case Studies in
Historical Rare Books in Edo Era –**

March 2018

**Department of Science and Advanced Technology
Graduate School of Science and Engineering
Saga University**

Natsuko Yoshiga

概要

古文書などの日本の文化財に関する多数のデジタルアーカイブがオンライン公開されつつある。これらアーカイブの書誌の記述には、文化財を構成する部品の詳細、文化財の成立、他の文化財および人物との関係などを示す重要な記述が含まれている。これらの書誌データのほとんどは、日本語を読める読者を対象としており、簡素化された自然言語で記載されている。しかし、このような記述様式では、プログラムを使って他のオンライン公開された書誌データを含めて横断的に検索することはできない。

プログラムを使って横断的に検索できることを、「機械可読」と呼ぶ。こうした書誌データを機械可読にするためには、書籍の部品や来歴を記述するデータ構造を定義するとともに、コレクションおよび対象分野で使用される語彙を整備し、横断検索にふさわしいデータ形式、例えば Linked Data へ変換することが必要である。これまで、こうした作業は、知識科学、プログラミング知識および対象文化財についての専門知識が必要とされ、具体的な変換手法も確立されていないため、データ整備を困難としてきた。

本論文では、文化財書誌の具体例として、江戸時代以前の書籍である古典籍の書誌データを用いて、非構造化データを構造化データに可能な限り自動的に変換する手法について提案する。また、提案手法の有効性および汎用性を二つの規模の異なる古典籍書誌データで比較検証する。その結果は、提案手法が、様々な書誌データに対して適用可能な汎用性の高い手法であることを示す。この手法により、対象領域の専門家が Linked Data を構築する負担を減らし、Linked Data の内容確認に注力できる。また、Linked Data 技術により、書誌データの属性および値に外部組織で定義されている情報を接続することが可能となり、横断検索により理解を深めることを支援できる。

Abstract

A large number of digital archives on Japanese cultural heritages such as historical rare books has been published online. Bibliographic descriptions in those archives contain important information about parts and provenances of books, relations to other books or persons and so on. Most of those bibliographic descriptions are written in concise natural texts which are aimed to be read by human users. In other words, those archives do not have data structure suitable for analyses by computer programs.

A word “Machine-readable data” means data which can be cross-searched with external data by computer programs. To convert conventional bibliographic data into machine-readable ones such as Linked Data, we need to define suitable data structure which describes parts and provenances of a book, and collect relating vocabularies. Particularly in cultural heritage archives, converting bibliographic data into Linked Data requires handworks by specialists of the book’s domain. Costs of those handworks prevent us to construct cultural heritage archives in Linked Data formats.

In this thesis, I propose a semi-automatic method to convert unstructured data into structured ones in a case of bibliographic data of Japanese historical rare books as an example of cultural heritages. By applying the method to two bibliographic data of historical rare books in different sizes, the method shows high performance and high versatility which can be applicable to various bibliographic data. Employing this method, specialists of the domain can focus their attention on quality of Linked Data, rather than constructing them. Moreover, by connecting attributes and values of bibliographic data to external information through technologies in Linked Data, archives will enhance their values and availability.

目次

第 1 章	序論	1
1.1	目的	1
1.2	背景	3
1.2.1	インターネット普及に伴う文化財資料が果たす役割の変化	3
1.2.2	文化財分野で Semantic Web 実現が難しい理由	6
1.2.3	本論文の位置付け	9
1.3	論文の構成	9
第 2 章	Linked Data 化への基本技術	10
2.1	Semantic Web の考え方	10
2.2	Uniform Resource Identifier によるリソースの明示	12
2.2.1	概要	12
2.2.2	実用的な URI を作るためのガイドライン	12
2.2.3	URI の参照解決方法	13
2.2.3.1	サーバレスポンスの制御による方法	13
2.2.3.2	フラグメント識別子を用いる方法	14
2.2.4	URI の接頭辞表記法	15
2.3	Resource Description Framework の仕組み	16
2.3.1	概要	16
2.3.2	概念の構造化を行う RDF スキーマ	19
2.3.2.1	リソースの分類を表現する基本クラス	19
2.3.2.2	ノード間の関係を表現する基本プロパティ	20
2.3.2.3	空白ノード	22
2.3.3	RDF でのシリアル化形式	22

2.4	Linked Data による知識の連携	23
2.4.1	Linked Data の 4 原則	23
2.4.2	外部リンクへの接続方法	24
2.4.2.1	文化財書誌につながるメタデータ	25
2.4.2.2	Linked Date に用いる汎用語彙の選択	26
2.4.3	Web 空間でのオントロジー	28
2.4.3.1	知的処理の研究概要	28
2.4.3.2	知的処理, オントロジーおよび Semantic Web の融合	29
2.4.3.3	ウェブ・オントロジー言語 OWL の概要	30
2.4.3.4	ウェブ・オントロジー言語 OWL の語彙	31
2.4.3.5	文化財オントロジーの概要	34
2.4.4	RDF ストアおよび SPARQL 言語の概要	39
2.4.4.1	RDF ストアの概要	39
2.4.4.2	RDF ストアの実装	39
2.4.4.3	SPARQL 言語の概要	40
2.5	文化財書誌の解析に必要な自然言語処理	41
2.5.1	自然言語処理技術の概要	41
2.5.2	形態素解析手法の種類と特徴	42
2.5.2.1	形態素単位の手法	42
2.5.2.2	文字単位の手法	43
2.5.3	固有表現抽出	44
2.5.3.1	概要	44
2.5.3.2	固有表現の定義	44
2.5.4	検索システムの評価指標 F 値	44
第 3 章	非構造化書誌データから Linked Data への自動変換手法	47
3.1	提案手法の概要	47
3.1.1	既存の古典籍書誌データの Linked Data 化が困難な理由	47
3.1.2	古典籍の読み解きに求められる正規化	49
3.1.3	提案手法の概要	49

3.1.4	使用データセット	50
3.2	固有表現抽出	52
3.3	固有表現と外部情報資源の接続	66
3.4	書誌オントロジーの定義および構築	68
3.4.1	基本オントロジーの構築	69
3.4.1.1	クラスの定義	69
3.4.1.2	オブジェクト・プロパティの定義	69
3.4.1.3	データ・プロパティの定義	72
3.4.1.4	アノテーション・プロパティの定義	73
3.4.2	用語オントロジーの構築	73
3.5	提案オントロジーに対応する Linked Data への変換	74
第 4 章	提案手法の有効性	78
4.1	提案手法の利点	79
4.2	注記項目から固有表現抽出を行う意義	79
4.3	MeCab による単語切り出し精度	81
4.4	固有表現抽出精度	82
4.5	スクリプト群の汎用性	83
4.6	抽出単語の外部リンクへの接続の容易さ	84
4.7	生成した Linked Data の評価	84
4.8	提案手法の適用可能性	85
第 5 章	先行研究に対する本研究の位置付け	86
第 6 章	まとめ	88
	参考文献	91
付 録 A	RDF ストアの利用	108
A.1	MetaBridge を用いた URI 参照解決	108

付録B ソースコード一覧	112
B.1 単語リストの例	113
B.2 Mecab ユーザ辞書生成	116
B.3 Linked Data 化スクリプト群	130

目次

2.1	接頭辞の例	16
2.2	RDF トリプル模式図	17
2.3	適合率および再現率についての図解	46
3.1	Linked Data 生成自動化の手順	50
3.2	自動スクリプト, データ提供者, コンテンツ自体の専門家による Linked Data 生成過程での変換作業の分担	52
3.3	Mecab ユーザ辞書作成の流れ	57
3.4	スクリプト B2 (人名推定) の仕組みおよび具体例	60
3.5	データクリーニングの具体例	62
3.6	Linked Data 化された書誌データ (表 3.1) の一部	63
3.7	URI 付与ルール具体例	65
3.8	簡易な古典籍書誌オントロジー	69
3.9	古典籍書誌オントロジーの基になる, 基本オントロジーのクラス構成	70
3.10	クラス同士の関係を定義した古典籍書誌オントロジー	71
3.11	オブジェクト・プロパティの構成 (左) およびその定義 (右) [1]	71
3.12	データ・プロパティの設定例	72
3.13	オントロジー表現の例	74
3.14	スクリプト G (書誌オントロジー生成) および H (Linked Data 生成) の流れ	75
3.15	古典籍書誌オントロジー上に Linked Data 化した <i>Ichiba</i> をインポートした際のスクリーンショット	76
3.16	古典籍書誌オントロジー OWL データ上に Linked Data 化した <i>Ichiba</i> をインポートした結果	77

目次

A.1 ブラウザでのメタデータ情報取得画面（古典籍）	110
A.2 ブラウザでのメタデータ情報取得画面（著者）	110
A.3 ブラウザでのメタデータ情報取得画面（外題）	111

表 目 次

3.1	既存の古典籍書誌データの例	48
3.2	自動化スクリプトおよびその入出力の概要	51
3.3	古典籍書誌データの固有表現クラスおよび固有表現の例	56
3.4	固有表現クラス別のユーザ辞書搭載単語数	58
4.1	提案手法を適用したコレクションのサンプルレコード数および全レコード数	78
4.2	<i>Ichiba</i> および <i>Kokubun</i> に対する Linked Data 化にかかる時間	79
4.3	固有表現抽出を行う項目	80
4.4	<i>Ichiba</i> および <i>Kokubun</i> における全レコード数および注記有りレコード数	81
4.5	各固有表現毎の、全項目中に現れる固有表現数に対する注記中のその割合	81
4.6	<i>Ichiba</i> および <i>Kokubun</i> に対する MeCab の単語切り出し精度	82
4.7	<i>Ichiba</i> のスクリプト (表 3.2) および固有表現クラス別固有表現抽出精度 (F 値)	83
4.8	<i>Kokubun</i> の固有表現クラス別固有表現抽出精度 (F 値)	84
4.9	書誌データ毎の固有表現クラス別固有表現出現数に対する接続リンク数の割合	85
B.1	<i>Ichiba</i> および <i>Kokubun</i> 解析用データベース sites に格納されたテーブル	112

ソースコード目次

2.1	RDF トリプルのソースコード例	18
2.2	SPO 文による全 RDF データの抽出	40
3.1	ユーザ辞書未搭載および搭載の MeCab での解析結果	57
3.2	CSV 形式で表現した由来情報データベース上のデータ (一部)	61
3.3	<i>Ichiba</i> 書誌項目に関するデータ	64
3.4	CSV 形式で表現した典拠データベース 1 上のデータ (一部)	66
3.5	CSV 形式で表現した典拠データベース 2 上のデータ (一部)	67
A.1	Apache2.2 系 Web サーバ用リダイレクト設定	108
A.2	佐賀大学電子図書館貴重書コレクションウェブサイトから MetaBridge の <i>Ichiba</i> 用 SPARQL エンドポイントへのクエリ発行用 PHP スクリ プト	109
B.1	固有表現クラス「時」の単語リストの例	113
B.2	固有表現クラス「部品」の単語リストの例	113
B.3	固有表現クラス「ロール」の単語リストの例	113
B.4	固有表現クラス「専門用語」の単語リストの例	114
B.5	NDLA の SPARQL エンドポイントから人名典拠データを抽出するス クリプト	114
B.6	<i>Ichiba</i> の書名, 編著者名をキーワードとして NDL サーチから関連単 語の自動収集例	116
B.7	ユーザ辞書生成スクリプト集	124
B.8	3MB 以上の XML を読み込み MeCab 用 CSV などのフォーマットに 変換可能なスクリプト	127
B.9	MeCab 設定ファイル (mecabrc)	128
B.10	コスト値生成ファイル	129

B.11 書誌属性リストの例	130
B.12 メインスクリプト (A, B, Cを連動)	130
B.13 スクリプト B1 (時クラス抽出)	135
B.14 スクリプト B1 (題クラス抽出1)	141
B.15 スクリプト B2 (題クラス抽出2)	143
B.16 スクリプト B1 (用語クラス抽出)	143
B.17 スクリプト B1 (場所クラス抽出)	145
B.18 スクリプト B1 (人名クラス抽出)	146
B.19 スクリプト B2 (人名クラス推定)	148
B.20 スクリプト D および E (データクリーニングおよび URI 付与スクリ プトを連動)	150
B.21 スクリプト F (外部リンク探索および典拠データベース登録)	162
B.22 基本オントロジー (手作業による作成)	176
B.23 スクリプト G (用語オントロジー生成)	201
B.24 スクリプト H (Linked Data 生成)	208

第1章

序論

1.1 目的

Web 上には、多数の文化財の書誌データおよび原本画像が、国内外の様々な規模の組織により公開されている。これらの公開情報は、その注記事項に至るまで、文化財の研究者が文化財成立の手掛かりを把握するための重要な情報となる。さらに、文化財の位置づけを知るためには、対象の文化財のみではなく、その文化財に関する文化財あるいは文化財とその成立に影響を与えた人とその人の役割、時、場所などの情報を知ること重要である [2][3]。

書誌データの記載事項を分析可能にする手段として、図書館、博物館などの書誌提供機関は、**Linked Data**[4]化を試みている。理想的な書誌データの Linked Data 化とは、従来の書誌データから書誌の読み解きに必要な人名、時、地名、専門用語などのキーワード（以下、固有表現 [5] と呼ぶ。）を抽出し、固有表現に対し Web で一意の事物を指す **Uniform Resource Identifier (URI)** [6] を紐付けた正規化データに変換するとともに、プログラムで分析できる形式とすることである。ここでの正規化とは、注釈などの補足的な記述を値から分離するなどの過程により、各属性に適した型のデータとするとともに表記揺れを無くすことである。また、書誌属性とその値も全て URI で表現することで、属性同士、固有表現と属性の関係を知識構造のデータとして利用可能になる。さらに、URI で定義された属性や固有表現には同義の外部組織の URI を紐付けられる。URI を関連付けることにより、Linked Data 化した書誌データは、元書誌に付加された外部情報と関連付けられる。このように Web 空間で共有および再利用可能なデータとすることにより、書誌利用者は、Web

上の関連情報を横断的かつ機械的に分析することが可能となる。

既存の書誌データから Linked Data を作成するには、単語間の区切りが無い日本語からの固有表現抽出、人名や件名の名寄せ、データ項目および固有表現の語彙整備と内容確認、さらに文が示す事物間の関係の定義といった作業が必要である。これまで、こうした作業の大半は、文化財コレクション自体の内容に精通した専門家の助けを借りて手作業で行う必要があった [7]。その書誌データの構造は、研究者が対象の文化財のみではなく、その文化財に関係する周辺の文化財あるいは文化財とその成立に影響を与えた人、人の役割、時、場所などの単語から関連性を見つけ出すことが可能な構造であるとともに Web 公開可能でなければならない。こうした作業を手作業で行うのは、多大なコストを必要とし、このことが書誌データの Linked Data 化を妨げてきた [8]。

本論文では、従来から指摘されていたコストに関する課題を解決し、文化財書誌データの Linked Data 化を実現するために、可能な限り自動的な手法を提案する。また、本手法を江戸時代の古典籍からなるコレクションに適用し、その有効性を示す。本手法は、対象コレクションの詳細に依存しない汎用性の高い手法であり、対象コレクションに応じた専門家による固有表現収集や定義、および内容確認による若干の手作業や調整により対応可能である。

従来、Linked Data 化を目的とした固有表現抽出では、一般に形態素解析ツールや文脈パターン認識を用いた手法が用いられてきた。本研究では、それらの手法を、古典籍で使用される単語や書誌学用語に対して拡張する。これらの手法を有効に活用するために、あらかじめ、目録記述規則および Web から研究対象周辺の予備知識を収集する。また、実際の書誌データを構造化する枠組みとして、古典籍を構成する概念およびそれに紐づく固有表現やデータ項目の関係に関する機械可読な構造的知識（以下、オントロジー [9][10] と呼ぶ。）を定義する。さらに、本手法での Linked Data 化の効率を検証するために、小規模および国立機関のコレクションのサンプルデータに本手法を適用して、固有表現抽出精度や外部組織へのリンクの容易さなどを分析する。

提案手法で作成した固有表現の定義および相互の関連は、構造化データとして再利用可能となるため、軽微な調整で、類似分野の他コレクションの Linked Data 化

に応用できる。また，Linked Data の構造は簡易な形式であり，対象となる古典籍自体の専門家も，構造化データを容易に編集可能である。

1.2 背景

1.2.1 インターネット普及に伴う文化財資料が果たす役割の変化

我々の周りには，過去の歴史や文化の経緯を伝えるものとして，長期間保存されている有形・無形の事物が存在する。中でも，人による創造物で，歴史的，文化的，または学術的価値が高いとされているもの，人々の生活の推移の理解のため欠くことのできないものなどは，文化財 [11] と呼ばれている。多種多様でありかつ大量に存在する文化財に対し，十分な情報を整理して資料化し，未来に向けていかに効率的に伝えるかは，博物館，図書館，文書館などの MLA 機関 (Museum, Library and Archives¹) で大きな課題となっている [12][13]。

例えば，書誌データを作成する場合，記述対象の選定，画像の撮影基準，書誌データの公開範囲など，多くの専門的事項についてあらかじめ取り決める必要がある [7]。これらの書誌記述の基準とその書誌データの多くは，組織外の人々に書誌データを公開することを主たる目的とせず，記録を主目的とし，組織ごとの方針で作成され，運用されている。

1990 年代以降にインターネットが普及し，我々の生活に深く浸透し始めた。まず，米国が，1993 年に情報通信による教育および産業の発展を目指して，全米情報基盤 (NII, National Information Infrastructure[14]，別名：情報スーパーハイウェイ) 構想に着手した。その後，日本でも，2000 年代前半にインターネットの急速な普及が始まると，様々な産業分野で Web 上を用いたビジネスが進展した。個人でも，モバイル機器が普及し，SNS (Social Networking Service, 社会ネットワーク) の利用が活発になった。これに伴い，ビジネスに個人データを利活用する動きが加速した。

文化財関係組織においては，資料群を電子化し，データベースにすることで，インターネットを通じて誰もが文化財資料にアクセスできる情報基盤の整備が世界的

¹博物館 (美術館)，図書館，文書館のような，文化的，歴史的な情報資源の収集，保存，提供を行う組織を指す。

に進められた [15]. 文化財分野のアーカイブ²では, Web 空間で流通可能な資料, 特に, 対象文化財そのものの状態が記載されている書誌データ, 画像, 動画, 音声データの公開が進められた. 例えば, 2017 年 10 月の時点で, 欧州の Europeana[17] が 5323.7 万件, 米国の Digital Public Library of America (DPLA) [18] が 1782.0 万件, 日本古典籍総合目録 [19] が 59.3 万件, 国立国会図書館デジタルコレクション [20] が 58.4 万件のデータを公開している³.

このような文化財書誌データと画像類の Web 公開は, 人々に文化財の存在とその意義を展示し, 関心を高めることに貢献している. また, 文化財を研究対象とする研究者にとっては, Web アーカイブに設置されている検索システムを用いて研究対象となりうる資料の所在を簡便かつ速やかに調べることが可能となる. その結果, 文化財アーカイブは, 文化財の研究者が研究対象そのもののみでなく, その研究周辺分野に関連する知識を広範に探索し, 理解する手がかりを得る手段としての役割を担うものになった [13].

2010 年代後半の現在は, 身の回りの電化製品がデータを集積し, インターネット で活用可能にする Internet of Things (IoT, モノのインターネット) が徐々に普及しつつある. 通信機器を用いて, 過去の活動をデータとして収集し, 活動パターンを自動的に分析, 学習を繰り返すことで, 適切な情報やサービスに還元する Artificial Intelligence (AI, 人工知能) 技術が, ビジネス面のみではなく, 我々の生活をも大きく変えようとしている. このような社会の変革は, 第四次産業革命 [21] と呼ばれる. また, 実社会と Web 空間相互でデータを収集する基盤が整い, 蓄積されたデータを解析することで大きな社会的価値を生み出していく社会は, データ駆動型社会と呼ばれている.

データ駆動型社会への対応は, 文化財関連データの世界でも始まっている. 文化財関連データをインターネットを介して活用するための, 様々な取り組みが進められている. 例えば, 国レベルでは, 2012 年に総務省が知のデジタルアーカイブ [22][23] を提言し, そのガイドライン [24] を示している. これは, 国全体で MLA 機関が所蔵する知的情報を Web 上で共有できる基盤 (知識インフラ) を構築することを目的

²デジタルデータ, すなわち対象文化財の画像などとメタデータ [7] を付与したもの (デジタル二次資料) を指す [16].

³書誌データ数はいずれも概数.

とする。

また、石川 [25] は、アーカイブに蓄積されている多様な情報で、研究者の「知の構造化」を支援するのがこれからの MLA 機関の重要な役割であり、さらにその支援はこれらの館が連携することで実現すると指摘している。具体的には、各 MLA 機関が、個別に保有する書誌データを統合して検索可能にすることで、研究者が求める情報へのアクセスを容易にする。また、史料から歴史的事項の 5W1H（いつ、誰または何が、どこで、なぜ、どのように）を集積し提供することで、物事の成り立ちの解明を裏付ける知識を補完する。さらに、5W1H の出来事の参考資料となる背景知識、例えば、人名とその来歴、地名、時、職名あるいは役割名などの予備知識を引用可能にする。

実際に、このような知の構造化が実現すると、より大局的かつ研究分野を超えた多様な視点で史実を俯瞰することが可能となるため、文化系統学 [26][27]、歴史知識学 [28][2] あるいは総合資料学 [29] と呼ばれる、人文系史料に対してコンテンツの周辺情報までを研究対象に含めた定量的な研究が加速すると考えられる。例えば、歴史史料から戦国大名の同盟または敵対関係に関するキーワードを抽出し、時間軸に沿って人間関係ネットワークを可視化することが可能となる [30]。この例では、研究対象のデータベースが、史料の要約となるテキストデータ、時、人名、職位、地名、備考などの属性から構成されており、それらに対応する値も正規化されている [31]。

一方、Web 公開されている書誌データの多くは、一見、属性と値に分けて記述されていても、属性の中に詳細属性が内包される、あるいは値に注釈が付加される、同一のものが複数の表記を持つなど、そのままの形でデータ分析に活用することが困難な場合が散見される。加えて、各属性の語彙定義および属性同士の関係定義についても、それらが凡例や記述規則として参照可能であっても、他のコレクションに用いられている属性との同義性などが曖昧な場合がある。

このように、データ分析に利用可能な書誌データを構成するためには、データの構造化と適切な正規化が必要である。こうした作業には、構造化と正規化に詳しい情報系の専門家と、対象に詳しい歴史的文献の専門家の共同作業が必要である。しかしながら、多くの手作業が必要とされ、そのことが構造化・正規化の妨げとなっ

てきた [32][33].

1.2.2 文化財分野で Semantic Web 実現が難しい理由

Web 技術は、素粒子実験に関わる研究者のために、Tim Berners-Lee によって開発された。当初はテキストデータを公開することが中心であったが、画像などのマルチメディアに対応するとともに、動的に情報を構成する技術が開発されてきた。これらの技術は、人が情報を閲覧することを前提に開発されてきた [34]。このままの形では、ソフトウェアを用いて自動的に構造化された情報を取得することは困難である。

文化財書誌データでは、さらに特殊な事情がある。文化財書誌データの場合、対象分野の専門家あるいは学芸員などが観察した結果である手書きの記録票（台帳）には、多くの文化財で共通する題名、作成者、作成年などの基本属性に加えて、作品分野に特有な属性および注記事項が記述されている [35]。現在、オンライン公開されている文化財データベース上では、基本属性は多くの文化財で共通して設定される属性であるため、属性は独立して設定されており、それに対応する値の多くは正規化された値として入力されている。一方、それ以外の文化財固有の詳細属性、書籍を例にするならば、本文の著者以外の、挿絵や奥付などの作者、作成年などの属性はあらかじめ設定されていない。そのため、本文の著者名の属性欄にそれらの値を入力し、その前後に記号や括弧書きなどを記す、あるいは、注記に入力するという方法を取ることになる。結果的に、これらの記録票に記載されたすべての情報が、文化財研究者にとって重要であるため、正規化されていない値が入力される、あるいは、書誌データベースの注記欄には多くのテキスト文が記載されることになる [36]。通常、注記欄の内容は自然文で非構造的に記述されるため、機械可読の形で情報を取り出すことが困難となる。

1998 年に Tim Berners-Lee は、**Semantic Web**[37] を提唱した。Semantic Web とは、プログラムによりデータ構造を読み出すことができる（「機械可読」という）Web サービスおよびその連携を指す。この語は、従来の非構造的なテキスト文を解析処理可能な構造化データに変え、そのデータを人間の暮らしをより良くするために活用しようという呼びかけの象徴としても使用される。この Semantic Web の考

え方に沿って、2006 年に Web 上で物事概念を取り扱う方法として、一意の識別子である URI による Linked Data モデルが提案された [38]。Linked Data 技術により、URI 化された概念同士がサービスを跨いで結合され、Web 上に公開された物事の構造化と機械可読性が向上することが期待された。

一方、構造化された概念を記述するための具体的な方法として、オントロジー工学と呼ばれる、人工知能研究の一分野が研究されてきた [39]。人工知能におけるオントロジーとは、ある分野において人間が得た知識を機械処理可能なデータで表現したものであり、オントロジー工学では、オントロジーをコンピュータが処理可能な計算量で、かつ形式的な構文で、推論可能な論理を記述する手法を取り扱う。Semantic Web が提唱されて以来、物事概念構造までをデータ化する技術として、記述論理を含むオントロジー工学が注目された (2.4.3.1 節参照.)。

2004 年には、Web 上での概念や概念関係の記述方法、概念に関する推論システムの実装を課題としたオントロジー記述方法が標準化された [40]。すなわち、Web 上でのオントロジー記述に対応する Web オントロジー言語 (Web Ontology Language, OWL) [9] が World Wide Web Consortium (W3C) [41] から公告された。これにより、知識体系を機械可読にして Web 上に公開することが加速された。

文化財分野においては、先進的な MLA 機関にて、長い年月を掛けて、所蔵コレクションの分野に対応したオントロジーの提案と実装が行われてきた。例えば、書籍の分野では、国際図書館連盟 (The International Federation of Library Associations and Institutions, IFLA) [42] が、Functional Requirements for Bibliographic Records Entity-Relationship (FRBR_{ER}) [43] を策定している。これは、書籍から創作表現を抜き出し、主題を分析可能にする形式オントロジーである。FRBR_{ER} を用いた例として、具体的な書籍名の上位概念である著作 (2.4.3.5 節参照.) を検索可能とした WorldCat [44] がある。国立国会図書館では、人名および件名典拠情報である Web NDL Authorities (NDLA) [45] を一般公開している。NDLA は、世界の典拠情報を統合する Virtual International Authority File (VIAF) [46] に接続されている。

一方、書籍以外の文化財では、Europeana が文化財の成立過程、すなわち、知識構造の表現に Europeana Data Model (EDM) [47] を採用している。これは、Lightweight Information Describing Objects (LIDO) [48]、Encoded Archival Description (EAD)

[49], Metadata Encoding & Transmission Standard (METS) [50]などで記述された, ヨーロッパの MLA 機関が所蔵する多様な文化財メタデータを Semantic Web 技術で統合するオントロジーである.

その他にも, Wikipedia[14]に登録された記事を構造化知識で提供する DBpedia[51], 文化財用語の典拠情報である The Art & Architecture Thesaurus (AAT) [52]など, Linked Data のハブとなり得る多くの汎用的オントロジーが構築されている.

上記のような, 比較的大規模で汎用的なオントロジーが利用可能となった結果, これらを利用して, 地域や個人などが所蔵する小規模のコレクションについて言及した, より具体化したオントロジーと Linked Data を構築することができる. つまり, ある文化財に関する普遍的なルールは, データハブとなる公的機関の汎用オントロジーに一任することができる基盤が整い始めている.

しかし, Semantic Web 技術活用に向けた文化財書誌データの機械可読化は, 専門家との協働, 各組織で取り扱われている概念の擦り合わせ, 構築にかかるコストなど, 課題が山積している [10][53][54]. 例えば, Europeana, Digital Public Library of America (DPLA) [18], 国立国会図書館など, 比較的規模の大きい組織では, 数十万から数千万点規模の書誌データを Linked Data 化済みとしているが, 実際には, 題名や作者など書誌項目に入力された基本情報を正規化, 構造化, あるいは名寄せなどの整備を行わずに, グラフ形式へ自動変換したにすぎない. また, 書誌利用者が知りたい情報は, 注記中の自然文にも多く存在するが, それは利用者が手入力するキーワードを用いた全文検索でしか抽出できない. さらに, 注記の自然文を正規化せずにグラフ形式のデータに変換しても, 表形式のデータの検索と比べて, グラフパターンの抽出に加え, 各ノード内の全文検索を行う分, 検索時間がより長く掛かる. つまり, 中途半端な Linked Data では, アプリケーションによる機械的処理が困難であり, グラフ形式のデータが持つ利点を生かせない.

文化財の書誌データには, 非構造化データを含む注記欄にも無視できない情報が含まれているため, Linked Data やオントロジー技術を用いる以前に, そこから属性と値を可能な限り自動的に取り出す作業が必要となる. 加えて, 現実には, 特定の文化財の専門用語に関する, 共有可能なオントロジーが存在しない場合がほとんどである. よって, 所蔵するコレクションから専門用語や固有名詞を抜き出して, オ

ントロジーを構築することが必要になるが、これについても可能な限りコストを抑えた方法を用いる必要がある。

1.2.3 本論文の位置付け

Semantic Web 技術を活用した文化財情報の公開を行うには、実在する書誌データから、文化財の理解に必要な語の抽出および定義を容易に行えるような技術開発が必要である。本論文では、Semantic Web 志向による文化財書誌データの機械可読化技術の視座から解決を試みる。そのために、文化財書誌データの例として、実在する古典籍書誌データをオントロジーに沿って構造化 (Linked Data 化) する。また、Linked Data 化は、可能な限り人手によらない方法を提案する。最終的には、3 章で示す手法を基にして、多種多様な館が所有する文化財書誌データから抽出された Linked Data が現実的コストで形作られることを目指す。提案手法で生成された Linked Data は、Web で文化財専門家をはじめとした多くの人々に共有可能となると同時に、その内容吟味にも注力されるため、信頼度の高いデータとなる。加えて、その周辺の、大小様々な規模のコレクションの構造化に、既に構築されたデータを再利用できる。その結果、文化財の研究に Linked Data 化された書誌データが欠かせないものとなり、多様な視点からの分析を促進する。

1.3 論文の構成

本論文の構成を以下に示す。

第 1 章では、本研究の目的および背景について述べる。

第 2 章では、本研究の基盤となる技術について説明する。

第 3 章では、非構造化書誌データからの Linked Data 化について述べる。

第 4 章では、3 章で示す手法の有効性を検証する。

第 5 章で、先行研究に対する本研究の位置付けについて述べる。

第 6 章で、まとめおよび今後の展開について述べる。

第2章

Linked Data化への基本技術

本章では、3章で用いる基本技術およびその関連知識について述べる。古典籍書誌情報、特に注記に含まれる情報を Linked Data へと変換するための基本技術についてまとめる。機械可読な Web である Semantic Web およびそのデータ実装としての RDF (Resource Description Framework), Semantic Web を相互に接続する Linked Data, そしてテキストから構造を読み出す自然言語処理がテーマである。

2.1 Semantic Web の考え方

European Organization for Nuclear Research (CERN) の技術者であった Tim Berners-Lee は、CERN に世界中から集まる研究者が実験データを共有する仕組みとして、Web 技術を 1989 年に開発した [55]。Web は、単にテキストを提供するだけでなく、画像等の他の媒体の提供が可能であるとともに、他のページとのハイパーリンクによって、インターネットに情報のネットワークを構築することを可能とした。2000 年代以降、企業をはじめとする Web サイトが増加し、インターネットのクライアントアプリとして拡大した。

Web ページの多くは、人が閲覧することを前提としている。つまり、自然言語で記述されている。世界中にある Web サイト (2017 年 11 月現在, 12 億 8800 万以上と推定されている¹⁾) から必要な情報を探し出すには、Google などの検索エンジンを使うことになる。検索エンジンは Web サイトを robot と呼ばれる仕組みで巡回し、全文検索によってキーワードを抽出し、蓄えている。ユーザの検索には、このキー

¹<http://www.internetlivestats.com/total-number-of-websites/> より

ワードに基づき、検索サイト独自の順位付けによって、検索結果を返してくる。そのような検索結果には、以下のような課題がある [56].

- 検索結果が多数であっても、必要なものが得られるか、上位に示されるかはわからない。
- 得られた結果の信頼性が曖昧である。
- 対象を示すキーワードが複数ある場合、効率的に和集合を得ることが困難である。

検索エンジンは、キーワードに類似した文字列を含むファイルを見つけることはできるが、キーワードとファイルの中身との関係を読み取るわけではない [57]. このような不便さは、Web 上の情報量とともに増加している。さらに、Web が企業活動に広く利用されるため、検索結果に数多く表示されることを目的としたキーワードの埋め込みも行われている。この結果、キーワードとそれを含む Web サイトの関係が曖昧になっている。

1998 年、Tim Berners-Lee は、ファイルの内容についても機械可読にした情報網を **Semantic Web** と呼び、それを Web 空間で築くための原則と道筋を示した [34]. その原則は以下にしめすものである [58].

1. あらゆる事物を Unified Resource Identifier (URI) で識別可能にする。
2. ある事物に関する情報の記述は断片的でよい。
3. 事物に関する全ての記述には、その記述が行われる状況 (Context) がある。アプリケーションが記述の信頼性を評価するためには、この状況に関するデータが必要である。
ここでの信頼性とは、信じるに値するかということである。言い換えると、ある事物が正しいという記述はしない。
4. 多様なコミュニティーが記述した情報は、矛盾点を指摘し合いながらも、お互いうまく融合することで自律的に進化できるものとする。
5. 可能な限り簡単に事物を記述する。不要な標準化、制約を避ける。

Semantic Web の起源は、1956 年、Richard H. Richens による自然言語を機械翻訳するための解析器の研究で発明された意味ネットワーク (Semantic Network) に遡る [59]。意味ネットワークは、人間の知識を表現するデータ構造の一つであり、人間の生み出した概念をノード、概念間の関係をエッジとしたグラフ型モデルを採る。このグラフ型モデルは、Web のハイパーリンクの仕組みと類似し、高い親和性がある。

Web 技術の標準化団体である World Wide Web Consortium (W3C) は、インターネット上に機械可読データの Web を形成するために、先に述べた Semantic Web の原則に基づき、要素技術および仕様を策定した。主要な技術について以下の節に説明する。

2.2 Uniform Resource Identifier によるリソースの明示

2.2.1 概要

実世界の事物 (リソース²) を Web 上で取り扱うには、まず、それらに URI[6] を付与する。URI とは、リソースを一意に指すために付与される識別子である。URI は、ある事物を一意に指し示す識別子である Uniform Resource Name (URN) [6] としての役割と、その事物について記述したファイルの置き場所である Uniform Resource Locator (URL)³ としての役割を持つ。Internationalized Resource Identifier (IRI, 国際化資源識別子) は URI を多言語対応させたものである [60]。URI では、ASCII 文字セットのみが使用を許されているが、IRI では Universal Character Set (Unicode) を含むことができる。

2.2.2 実用的な URI を作るためのガイドライン

安定して永続的に参照解決されるような実用的 URI を組み立てるのは、非常に難しいことである [61]。そこで、幾つかのガイドライン [62] が W3C によって示されている。その概略を以下に示す。

²実際には事物の概念を指す。

³URI, URN, URL の説明は、RFC3986[6] の 1.1.3 節を参照のこと。

- 短くて簡単にすること.
- 可能な限り長く残り得る URI にすること. 動的な要素 (プログラム依存の拡張子, パス名, 分類名など) を組み込まない.
- 管理しやすい URI にすること. 人間にとってもドメイン内での意味がわかりやすい実体名を採用する.

URI 決定後は, 以下に示す参照解決方法を利用して, クライアント側の要求に応じて適切なファイルを提供する.

2.2.3 URI の参照解決方法

Web 空間では, HTTP プロトコル [63] を用いてサーバとクライアント間でデータを送受信する. 実在するリソースを指す URI をサーバに要求する場合も, その値を HTTP プロトコルにより参照可能でなくてはならない. よって, リソースについて記述したファイルの在り処を参照可能にする必要がある.

しかし, リソース記述ファイルは, RDF (3.6 節に後述.), HTML などの形式やバージョンが異なる複数のものが存在する可能性がある. そこで, リソースとその記述ファイルの混同を防ぐために, Web サーバ上でコンテンツネゴシエーション (Content Negotiation) と呼ばれる処理を行い, 要求に合わせたリソースを提供する. 具体的には, 303 URI あるいはハッシュ URI と呼ばれる技術を要求するデータの規模, アプリケーション処理効率などを考慮しながら用いる [62]. これら二つの URI 参照解決方法について, 以下に述べる.

2.2.3.1 サーバレスポンスの制御による方法

Web 上で, 実在するリソースのやり取りを仮想的に行う方法の一つに, クライアント側の要求するデータ形式が RDF か HTML かを確かめて, サーバのレスポンスを制御する方法 (303 URI) がある. クライアント側が URI および返信に RDF データをリクエストする場合, サーバは HTTP ステータスコード “303 See Other” とリソースについて記述されているファイルの URI をクライアント側に返信する. 続い

て、クライアントはサーバから送られた URI を GET リクエストし、求める RDF データをサーバから受信する。

上記のように、サーバとクライアント間で2回の送受信が発生する。また、サーバに URI 要求に対応するレスポンスとリダイレクトの設定、並びに、実際に受信可能なファイルの URI 指定をあらかじめ行う必要がある。一方、リソースそのものの URI の記述は、例えば、

```
http://example.com/SagaUniversity
```

と、拡張子無しの簡易な URI で設定しておくことができる⁴。そして、この URI に対応するデータを

```
http://example.com/SagaUniversity.rdf
```

```
http://example.com/SagaUniversity.html
```

などとして、クライアントの要求に応じて送信可能にする。この方法は、次節のハッシュ URI による方法と比べると、サーバが保持する巨大なリソース情報からクライアントが求めるリソースの情報のみを部分的に提供できる利点がある。

2.2.3.2 フラグメント識別子を用いる方法

HTML の id 属性, name 属性のように、URL には元々ファイル中の特定の場所 (フラグメント) を指定する機能がある [6]。フラグメント識別子 (Fragment Identifier) は、# (ハッシュ記号) を語彙 (実体名) の先頭に付加したものである。

クライアントが基底部分⁵とフラグメント識別子を繋げたリソース⁶の URI についての RDF データを要求すると、サーバはフラグメント部分を除いた URI に対する HTTP レスポンスを返す。この方法による URI 参照解決はハッシュ URI と呼ばれる。例えば、

⁴example.com, example.net, example.org など、“example.” が付くドメインは、仕様書の作成用として RFC6761[64] にてあらかじめ予約されている。

⁵例えば、架空のリソースである `http://example.com/SagaUniversity#ComputerNetworkCenter` のうち、リソースのある場所を表す `http://example.com/` の部分を指す。

⁶例えば、`http://example.com/SagaUniversity#ComputerNetworkCenter` のうち、SagaUniversity を指す。

`http://example.com/SagaUniversity#ComputerNetworkCenter`

にアクセスすると、サーバは

`http://example.com/SagaUniversity`

の部分のみを見てレスポンスする。クライアントが RDF 形式でのデータを要求している場合、サーバは

`http://example.com/SagaUniversity.rdf`

を返信する。

SagaUniversity に関する語彙集である

`http://example.com/SagaUniversity.rdf`

には、フラグメントで表現していた `ComputerNetworkCenter`, `Library` などの語彙定義を含む。RDF データを扱うアプリケーションは、

`http://example.com/SagaUniversity.rdf`

を走査してフラグメントの `ComputerNetworkCenter` についての語彙定義を取り出すことができる。

フラグメントを付加した URI は、フラグメントが指すリソースに関する情報を一意のファイルとして直接には指定せず、リソースの識別子として都合が良い。その代わりに、クライアントは指定した形式で対象に関する記述を含む RDF ファイルをダウンロードすることができる。サーバとクライアント間の送受信も一度で済む。一般的に、あるドメイン（分野）で使用される各語彙のトリプル数（2.3.1 参照。）は多くて千単位と小規模である [65]。そのため、語彙集を一括でダウンロード可能な方が、サーバとの通信回数は一度で済み、アプリケーションの処理効率が良い。

2.2.4 URI の接頭辞表記法

URI を作る際には、Web 空間で基底部

`http://example.com/`

が一意であり，かつ，基底部中に実体 `SagaUniversity` が一意に存在していれば，

`http://example.com/SagaUniversity`

は一意の識別子として通用する．基底部は，あらかじめデータの先頭に定義された接頭辞で簡略的に表記される場合がある．接頭辞は名前空間 (namespace) と呼ばれる．

本研究に使用する接頭辞の例を以下に示す．

接頭辞	名前空間
dc	<code>http://purl.org/dc/elements/1.1/</code>
dcndl	<code>http://ndl.go.jp/dcndl/terms/</code>
dcterms	<code>http://purl.org/dc/terms/</code>
ndctype	<code>http://ndl.go.jp/ndctype/</code>
foaf	<code>http://xmlns.com/foaf/0.1/</code>
frbroo	<code>http://www.cidoc-crm.org/rdfs/FRBR2.0-draft-B.rdfs#</code>
ichiba_tenseki	<code>http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#</code>
rdf	<code>http://www.w3.org/1999/02/22-rdf-syntax-ns#</code>
skos	<code>http://www.w3.org/2004/02/skos/core#</code>
xsd	<code>http://www.w3.org/2001/XMLSchema#</code>

図 2.1: 接頭辞の例.

接頭辞と実体の間は「:」(コロン) を用いて区切る．例えば，

`http://example.com/SagaUniversity`

は，`http://example.com/`の部分を接頭辞 `ex` と定義した場合，`ex:SagaUniversity` と表記する．

2.3 Resource Description Framework の仕組み

2.3.1 概要

Web 空間で Semantic Web 志向のデータネットワークを形成するためには，Resource Description Framework (RDF) モデル [66] を用いることが推奨されている．

RDF とは、属性および値の関係を有向グラフで表現するための W3C[41] 標準データモデルである (図 2.2).

RDF トリプル例

<http://id.ndl.go.jp/auth/ndlna/00271618>とされる人物の情報

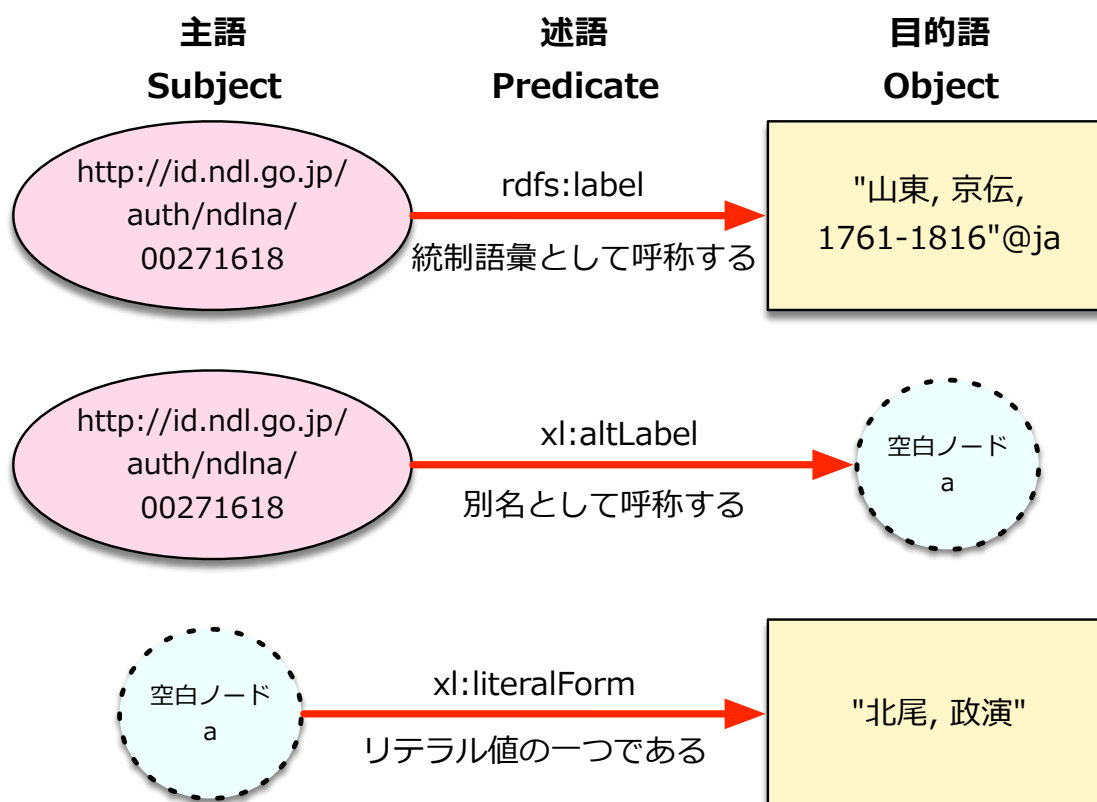


図 2.2: RDF トリプル模式図 [67].

図 2.2 で, 実在した江戸時代の人物について, 国立国会図書館の典拠データ (NDLA[45]) でどのように表現しているかを簡略的に示す [67]. RDF の最小単位はトリプル (Triple, Triplet) と呼ばれ, 情報資源間の関係を表現する. 全てのトリプルは主語, 述語, 目的語のセットで記述される. 複数のトリプルを繋ぐことで, グラフ型のデータ空間が構成される.

図 2.2 は RDF で記述するとソースコード 2.1 の通りとなる.

ソースコード 2.1: RDF トリプルのソースコード例. <<http://id.ndl.go.jp/auth/ndlna/00271618>> の Turtle[68] ファイルから抜粋.

```
1 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
2 @prefix xl: <http://www.w3.org/2008/05/skos-xl#>.
3 @prefix ndl: <http://ndl.go.jp/dcndl/terms/>.
4
5 <http://id.ndl.go.jp/auth/ndlna/00271618>
6   rdfs:label "山東, 京伝, 1761-1816";
7   xl:altLabel [xl:literalForm "北尾, 政演";ndl:transcription "キタオ,
   マサノブ"@ja-Kana,"Kitao, Masanobu"@ja-Latn]
```

Web 上のリソースの表現に RDF を使ったデータ構造を採用することで, リソース同士の接続を容易に行える. ノードを一意的識別子である URI, 例えば,

<<http://id.ndl.go.jp/auth/ndlna/00271618>>

とすれば, そのノードに別のリソースの URI を接続して, 関係を表現できる. このように, RDF によるデータモデルでは, 主語および目的語に相当するリソースの関係を簡潔なトリプルで表現する.

RDF を採用する利点は, 主語や目的語に割り当てられた URI に外部リソースを接続することで, 単一の Web サーバ上で取得可能な情報のみでなく, Web 上に散在する関連情報を収集できる点にある (2.4.2 節参照.). さらに, 述語に割り当てる URI に, 外部組織で定義されている語彙 (2.4.2.2 節参照.) で使用されている URI を紐付けて代わりに定義すること, 述語に定義域および値域 (2.3.2 節参照.) を定義することなどが, 一般的な表形式のデータモデルと比較して簡潔に行えるようになる.

そして, 組織を超えてリソースが属するクラスおよびその上位関係, あるプロパティを挟む主語と述語になるべきクラスあるいはデータ型が決定され知識構造が形成されると, 新しいリソースの知識構造についても既存の知識構造を用いて論理的に推定することが可能となる.

2.3.2 概念の構造化を行う RDF スキーマ

RDF には、リソース同士の関係を構造的に記述するための定義（以下、スキーマ (schema) と呼ぶ。）が提供されている。RDF スキーマ (RDF Schema, RDFS[69]) は、トリプルの主語および目的語となるノードのクラス、主語と目的語を繋ぐ述語を指すプロパティ、クラスあるいはプロパティ同士の親子関係などを定義するための基本機能を提供する。また、主語となるノードのクラス範囲である定義域、述語となるノードのクラス範囲あるいはリテラル値の型である値域を定義する。

RDF スキーマ以外に知識表現に必要な機能は、OWL[9] (オントロジー言語) に用意されている語彙や Dublin Core Metadata Initiative (DCMI) の DCMI Metadata Terms[70] などの汎用語彙 (2.4.2.2 参照.) をリソースの RDF データに追加して表現する。

2.3.2.1 リソースの分類を表現する基本クラス

RDF で表現される全ての事物はリソース (**Resource**) と呼ばれる。RDF スキーマ [69] では、あるリソースの分類を示すために事物の抽象的な概念であるクラスを用いる。URI が指し示すリソース自身も概念であるためクラスである。クラスは内包関係にあるサブクラスを持つ。

RDF スキーマの機能により、あるリソースがどのクラスに属するかという関係を明示することで、RDF で表現されるリソースの潜在的な意味と、別のリソースとの意味上の関係をデータで表現することが可能となる。

rdfs:Resource RDF で表現されるすべてのリソースの親クラスである。

rdfs:Class リソースの分類を表すクラス。具体的な分類名は **rdfs:Class** のサブクラスとして定義される。

rdfs:Literal データ中に存在する直接の値を指すリテラル値 (直値) が所属するクラス。 **rdfs:Literal** は **rdfs:Class** のサブクラスである。

rdfs:Datatype 数、文字列、日付などのリテラル値の型 [71] が所属するクラス。 **rdfs:Datatype** は **rdfs:Class** のサブクラスである。 **rdfs:Datatype** は **rdfs:Literal** の属性である。

rdf:langString `rdf:langString`⁷は言語タグ付き文字列値のクラスである。また、`rdfs:Datatype` のインスタンスで、`rdfs:Literal` のサブクラスである。

rdf:HTML (参考⁸) `rdf:HTML` は HTML リテラル値のクラスである。HTML リテラル値とは、値を HTML 形式でマークアップしたものであり、W3C DOM4[72] 仕様に依存する。`rdfs:Datatype` のインスタンスとして指定され、`rdfs:Literal` のサブクラスである。

rdf:XMLLiteral (参考) `rdf:XMLLiteral` は `rdf:HTML` 同様、XML リテラル値に対して用意されたクラスである。

rdf:Property リソースの属性のクラスである。

2.3.2.2 ノード間の関係を表現する基本プロパティ

RDF プロパティ [71] は、トリプルにおける述語にあたる。言い換えると、主語と目的語の関係を表す概念である。プロパティは `rdfs:subPropertyOf` を用いてサブプロパティを表現できる。また、プロパティは、オブジェクト・プロパティ (Object Properties)、データ・プロパティ (Data Properties) およびアノテーション・プロパティ (Annotation Properties) に分別される。

- オブジェクト・プロパティ：トリプルの目的語にリソースを持つプロパティ
- データ・プロパティ：トリプルの目的語にデータ上にある直接の値 (Individuals, 直値) を持つプロパティ
- アノテーション・プロパティ：トリプルの主語に、表記名あるいは補足説明などを追加するプロパティ

オブジェクト・プロパティおよびデータ・プロパティの定義用主要プロパティ

⁷クラスであるが、小文字である。

⁸RDF Schema 1.1[69] のような仕様で規定されているのではなく、参考として規定に記述されている情報には、参考 (Non-Normative) と記されている。

rdfs:range `rdfs:range` は, `rdf:Property` のインスタンスである.

`P rdfs:range C` というトリプルがあるとき, ある述語 `P` につながる目的語のクラスはクラス `C` と定義する.

rdfs:domain `rdfs:domain` は, `rdf:Property` のインスタンスである.

`P rdfs:domain C` という構文があるとき, ある述語 `P` で示される主語のクラスはクラス `C` と定義する.

rdf:type `R rdf:type C` という構文があるとき, リソース `R` はクラス `C` 型であると宣言する. つまり, 主語が目的語で示されるクラスのインスタンスであることを示す.

rdfs:subClassOf `C1 rdfs:subClassOf C2` とすると, クラス `C1` はクラス `C2` のサブクラスであると定義する. クラス `C1` はクラス `C2` の全てのプロパティに加え, 新規のプロパティを持つクラスである. 「is-a」関係とも呼ばれる.

rdfs:subPropertyOf `P1 rdfs:subPropertyOf P2` とすると, プロパティ `P1` はプロパティ `P2` のサブクラスであると定義する. 主語で示すプロパティは, 目的語で示すプロパティのサブクラスである. プロパティ `P1` はプロパティ `P2` の性質を全て持ち, かつ新しい性質が加わることで, `P2` の性質がより詳細になったものである.

アノテーション・プロパティとして使用する主要プロパティ

rdfs:label 主語で示すリソースには, 目的語で示す表記名 (人間が読める名前) を持つ.

rdfs:comment 主語で示すリソースは, 目的語に示す記述 (人間が読める文) を持つ. リソースが持つ概念を説明することによって, 当該リソースの利用を円滑にする役目がある.

上記以外にも, クラスおよびプロパティに使用されることがある語彙についての規定がある. 詳細は, RDF Schema 1.1[69] にて公開されている.

2.3.2.3 空白ノード

空白ノード (Blank Nodes) は、明確な URI を設定できないノードを設定したい場合に用いる。例えば、名前が特定されていない人物を主語として、その住所、ある特定の人物と友人関係であること、ある作品に大きな影響を与えたことなどの関連情報を目的語として表現する必要がある場合、主語に空白ノードを用いることで、関連情報を表現可能となる。

別の例として、図 2.2 では、主語は

`http://id.ndl.go.jp/auth/ndlna/00271618`

とする人名 URI, 述語は別名を表す `x1:altLabel`, 目的語が空白ノードである。その空白ノードにリテラル値 `x1:literalForm` である「北尾, 政演」および読みを表す属性 `ndl:transcription` として「“キタオ, マサノブ”@ja-Kana, “Kitao, Masanobu”@ja-Latn」がそれぞれ接続されている。つまり、NDLA の RDF データでは、別名とそれに対応する読みがある場合、空白ノードに一旦まとめて、それを人名 URI に接続するという方法が取られている。

2.3.3 RDF でのシリアル化形式

RDF でデータを公開する際は、用途に応じて様々なシリアル化形式を使用できる。シリアル化とは、RDF のようなグラフデータを、直列的なデータ形式に変換することである。具体的なシリアル化形式には以下の種類が存在する [66]。

RDF/XML W3C によって標準化された形式 [73]。XML 構文を利用する。現実的には、アプリケーションの多くはデータ量を節約し、文法が単純で、人間にとっても参照しやすいと考えられている Turtle や JSON 形式を利用する。RDF データ配布側は、歴史的経緯により、この標準形式のデータをダウンロード可能にしている。

Turtle 群 (N-Triples, Turtle, TriG, N-Quads) トリプル単位で RDF データを記述する [68]。トリプルの形に合わせて記述するので、人間にとっても理解しやすい。また、終了タグがないため、RDF/XML よりデータ容量を節約できる。

JSON-LD JSON 形式 [74][75] は, 多くのプログラミング言語がインポート可能な軽量データ形式としてサポートしている. JSON-LD[76] では, 外部組織の属性を使用可能である.

RDFa RDFa[77] は, HTML と RDF を同時に記述できる方法として開発されている. 初めに開発された RDFa Core[78] 構文は, 高機能な構造化データを構築可能なものである. しかし, 高機能の代わりに構文理解とコンテンツへのデータマークアップにコストが掛かる. そこで, より単純な仕様である Microformats[79], microdata[80] が開発された. 2012 年に W3C 勧告 [81]⁹となった RDFa Lite 1.1[82] は, RDFa Core を簡略化したものである.

2.4 Linked Data による知識の連携

2.4.1 Linked Data の 4 原則

人あるいはアプリケーションがデータを見つけた時, 他の関連するデータも取得できるように, 外部組織の URI をリンクしたデータを Linked Data[4][65] と呼ぶ. Semantic Web は, 異なるデータセット同士につながりを持たせた Linked Data による Web である.

2.1 節で示す Semantic Web の原則と同様に, Linked Data を Web で公開する方法にも原則が設けられている [38].

1. 事物の名前には, URI を用いること.
2. 人々が事物の名前を参照できるように, HTTP URI を使うこと.
3. 誰かがある URI 参照した時には, W3C 標準技術 (RDF, SPARQL) を用いて有用な情報を提供すること.
4. 人々がより多くの情報を見つけられるように, 提供する URI には, 他の URI を接続すること.

⁹Web 技術の国際的な標準規格化に向けた批准プロセスの一つ. 現実には, 検討およびテストは終了しており, 実用化の段階に入っている技術.

インターネット標準の HTTP URI を用いれば、URI スキームで他の URI との衝突を避けて該当するリソースに到達可能である。また、HTTP プロトコルにより、人々はウェブページで当該 URI の内容を参照できる。URI から URI を説明するウェブページへのアクセス方法については、2.2.3 節で説明している。URI についての機械処理可能な構造化データは、RDF で提供する (3.6 節)。加えて、RDF データをシリアル化したもの (2.3.3 節) や、可能であれば、SPARQL エンドポイント¹⁰を Web に公開することで、相互運用性が高まる。そして、URI には、多くの人が利用していて、信頼性が高く、安定した運用が行われると見込まれる、外部の URI¹¹へ可能な限り接続させる (2.4.2 節)。

2.4.2 外部リンクへの接続方法

2.4.1 で示した Linked Data の原則によれば、Linked Data で提供する URI には、ある URI に関連する情報をリッチにするために、外部組織の URI を接続させることが推奨されている [65]。

具体的には、ある組織の URI を主語として、その述語と目的語を別組織の URI で繋ぐ。外部 URI との接続の種類を以下に挙げる [65]。

同一性を明言する接続 ある二つの URI が指し示すリソースが同一である、あるいは完全に同じ意味であることを示すには、それらの URI を

```
http://www.w3.org/2002/07/owl#sameAs (owl:sameAs)
```

で繋いだトリプルを作る [83]。

owl:sameAs は、オントロジー言語 OWL による関係プロパティの一つである (2.4.3.3 節)。

ある特定の関係を明言する接続 二つの URI を特定の関係で繋ぐ場合は、Web で広く使用および管理されている DCMI Metadata Terms [70] や SKOS [84]、FOAF [85] などの再利用可能な汎用メタデータ (詳細は 2.4.2.1 参照。) を優先的に用いる。

¹⁰Web 上に公開されている RDF データベースとその検索サービス。

¹¹データハブと呼ばれる。

用語定義のための接続 データの表現に使われるクラスや属性 (用語) の定義を行う。この場合、定義自体を外部組織で管理されている語彙 (属性とその記述規則) を使い、RDF で記述することで、アプリケーションが意味を理解し機械的に処理可能な情報を取得できるようにする¹²。

外部の語彙や URI を自分の RDF データに接続すればするほど、異なるデータセットとの統合が上手くいくようになる。

2.4.2.1 文化財書誌につながるメタデータ

メタデータ (metadata) とは、データに関するデータを指す [7]。例えば、文化財の題名、作成年、作成者など、対象物に関して各組織で決めた属性とその値を記録したデータを指す。属性とその値の組で登録されるため、メタデータは RDF で表現可能であり、構造化データでもある。

また、誰でも Web から取得し利用可能な語彙¹³と、独自語彙¹⁴との意味上の関係を設定することで、外部組織のメタデータを含む横断検索が可能なデータセットを作成できる。

谷口 [7] によれば、文化財を対象にしたメタデータは用途により、三つの種類に大別される。記述対象の受け入れ機関、文化的背景、技術レベル、記述にかかるコストなどにより、属性の種類およびその粒度は様々である。

記述メタデータ (Descriptive metadata) 情報資源の発見、選択、入手を可能にするメタデータ。MLA 機関で利用者が文化財を検索するために閲覧する書誌情報は、ほぼ全て文化財について記述した、記述メタデータである。

管理メタデータ (Administrative metadata) 管理メタデータは、機関がリソースを管理するための情報について記述したメタデータである。

技術メタデータ デジタル資源の長期保存を目的に、再生形式、必要なソフトウェアのバージョンなどの技術情報を記述する。

¹²自分自身を自分の枠組みで記述可能な性質を自己記述性 (Self-Describing) [86] と呼ぶ。

¹³汎用メタデータスキーマとも呼ばれる。

¹⁴ここでは、自組織で定義しているデータセットの語彙を指す。

権利メタデータ 記述対象の著作権許諾, 利用条件などの権利および利用手続きに関する情報を記述する.

保存メタデータ 長期保存を目的として施した処置, 加工などの記録を記述する.

構造メタデータ (**Structural metadata**) 記述対象がシリーズで構成されている場合, 親子関係, シリーズの全数などの情報などを記述する.

メタデータ提供者は, 上に挙げたメタデータの用途および文化財の分野に合わせて, Linked Data に採用する汎用語彙を選択する必要がある.

2.4.2.2 Linked Date に用いる汎用語彙の選択

RDF は, リソースを記述するための形式的枠組みである. リソースそのものに関する特定の関係性を表現するには, 語彙を用いて行う [65]. 語彙もリソースを表現するための概念体系の一種であり, オントロジー [87] である.

また, リソースの集合であるデータセット自身についての記述も必要である. 管理および構造メタデータに該当する記述が相当する. 具体的には, データセットの来歴, 著作権 (利用条件), 作成日時, バージョンに関する記述である.

多様な文化財メタデータの表現を, 独自の語彙に加えて, 汎用的な語彙やオントロジーでも行うことで, データは冗長になるが, 専門的で多くの人にとって理解困難な傾向のある文化財メタデータが, より分かりやすく, 機械的処理の可能なものとなる.

以下に, 文化財分野で広く使用されている汎用語彙を示す.

DCMI Metadata Terms (通称: Dublin Core) 可能な限り幅広い領域のリソースを記述および再利用を可能にする語彙 [70]. DCMI (Dublin Core Metadata Initiative) が語彙の管理を行う. DCMI で次に挙げる語彙セット以外の全ての属性, クラス, エンコーディング規則¹⁵, DCMES (Dublin Core Metadata Element Set, Version 1.1, 通称: シンプル DC)¹⁶, DCMI-TYPE (DCMI Type

¹⁵<http://purl.org/dc/terms/>

¹⁶<http://purl.org/dc/elements/1.1/>

Vocabulary)¹⁷, DCAM (DCMI Abstract Model)¹⁸の4つの語彙セットで構成されている。国立国会図書館¹⁹, 国立公文書館²⁰, 東京国立博物館²¹など, 国内で多くの書誌を所蔵する館では, カバーする文化財分野を超えて, リソースの題名, 作者, 作成日などの基本的な属性に DCMI Metadata Terms を用いている [88][89][90].

SKOS SKOS (Simple Knowledge Organization System[84]) は, シソーラス, 分類表, 件名標目表, タキソノミ (Taxonomy) などの知識組織化体系 (分類体系) と呼ばれる概念体系を RDF で構築することでリソースの表現を行う際に用いる語彙である。SKOS の基本は, 概念に URI 付与し, 該当する呼称を `rdfs:label`²² 同様にラベル付けすることである。その際, `skos:prefLabel`, `skos:altLabel`, `skos:hiddenLabel` でラベルの使用優先順位を制御できる。また, 二つの異なる概念関係が, 広義である, 狭義である, 関連性があるという, 階層的表現が可能である。加えて, 概念関係の定義自体の記録 (`skos:historyNote`) や注釈 (`skos:editorialNote`) 用途の語彙も用意されている。

FOAF FOAF (Friend of a friend) [85] は, リソースとしての人物およびその団体を表現する語彙である。記述対象となる人物の URI, ウェブサイト, メールアドレス, 所属組織, 知り合いなどを記述する。FOAF 利用により, ネットワーク上の人間関係を記述し, 分析可能となる。

VOID VOID (Vocabulary of Interlinked Datasets) [91] は, RDF データセット自体のメタデータを記述する語彙である。また, 2.4.2.1 節の管理メタデータに相当する, データセットの利用許諾, 構造について記述するための語彙が用意されている。データセット (Dataset) は, ある組織が意図を持って公開, 管理, あるいは配布している RDF トリプルの集合である。また, 異なるデータセット間の関係を示すトリプル (主語と目的語のデータセットが異なるトリプル) をリンクセット (Linkset) と呼ぶ。このリンクセット自体にも URI を付けて,

¹⁷<http://purl.org/dc/dcmitype/>

¹⁸<http://purl.org/dc/dcam/>

¹⁹<http://www.ndl.go.jp>

²⁰<http://www.archives.go.jp>

²¹<http://www.tnm.jp/>

²²<http://www.w3.org/2000/01/rdf-schema#label>

`void:Linkset` クラスのインスタンスと宣言できる。つまり、データセットのトリプル全体の定義とは別に、他のデータセットとの関係を示すトリプルに限定したデータセットの定義ができる。リンクセットを利用可能にすると、例えば、リンク発見フレームワーク Silk[92] を採用したアプリケーションが、異なるデータセット間に貼られたリンクを効率的に抽出できる。

2.4.3 Web 空間でのオントロジー

2.4.3.1 知的処理の研究概要

人工知能研究では、コンピュータを用いて知的処理を行う試みが長年行われてきた。知的処理とは、主に、蓄積した知識あるいは経験で得たデータを将棋や囲碁の対戦、自動運転、画像認識、自然言語の解釈、明示されていない情報を組み合わせた推論など、人間では当たり前のように行われているが、コンピュータには難しいとされている処理を指している [93]。このような知的処理をコンピュータが行うには、人間が知覚した概念を解釈する過程である認知、自然言語処理、ロボット工学、Commonsense reasoning（日々の活動から得た経験則を用いて未知の局面に対する判断を行う能力を再現する人工知能研究の一分野 [94].）などの課題解決を必要とする。これらの課題のうち、主に認知や自然言語処理について明らかにする研究分野に、意味ネットワーク (Semantic Network)、フレーム (Frame)、記述論理などがある。

まず、意味ネットワーク [59] により、概念 (Node, ノード) と概念間の関係 (Edge, エッジ) によるラベル付けされた有向グラフで知識を表現する手法が確立された。次に、概念を表すノードを構成する下位の概念について属性スロットおよび属性値で表現するフレーム手法 [95] が提案された。しかし、意味ネットワークとフレームによる知識表現は、人間にも理解の容易な表現であるが、個体と集合の違いなど、より厳密で明確な概念およびその関係定義が困難である²³。そこで、KL-ONE[96] などの意味ネットワークを拡張した知識表現手法が開発された。KL-ONE では、意味ネットワークやフレームに不足していた、概念間の包含関係 (Subsumption)、集合とそれを構成する個体 (Instance) の関係を表現可能とした。また、エッジにロール

²³[40] の 1.1.1 節「知識表現の研究」より

という属性値を追加し、各エッジにつながる概念の種類および個体数を制約可能にした。

現在は、意味ネットワーク、フレーム、KL-ONE などの思想を継承し、プログラミング可能な範囲での概念の形式的な知識表現と推論の方法論を提案する記述論理 [97]、自然言語を機械可読な情報へ変換する UNL (Universal Networking Language[98]) などによる知識表現が開発されている [40]。これらは、Semantic Web を基にした知識の共有と再利用を指向するもので、高度な推論を追求する述語論理 [99] による知識表現とは異なる方向性をもつ。

2.4.3.2 知的処理、オントロジーおよび Semantic Web の融合

オントロジー (Ontology) とは、元々、哲学分野の用語である。日本語では「存在論」と称され、この世に存在するものはなぜ存在するのかということを議論する学問分野とされている [10]。1990 年代に、Gruber らによってオントロジーという言葉が人工知能分野でも用いられるようになると、人工知能および知的処理研究における意味合いは、2.4.3.1 節で示した知的処理、すなわち、ある概念をコンピュータで形式的に表現する手法に加え、データベースとして蓄積した知識を共有・再利用するシステムを研究する分野としての意味合いとなった [100]。人工知能分野でのオントロジーとは、人間が理解した、ある世界についての物事を体系的に書き記したもの [39][101]、あるいは、「普遍性を有する概念を関係付ける枠組み」 [40] とされている。知的処理技術を含むオントロジーは、概念の形式的表現や論理的処理を Web 空間で行うことを目指す Semantic Web の目標と合致している。

Semantic Web および Linked Data 構築のために策定されたオントロジーである **Web** オントロジー (Web Ontology) は、W3C による 2004 年の定義 [102] によると、ある特定の分野を意味するドメイン (domain) の知識を記述し、他者と共有して知識理解を深めるための語彙集と定義されている。また、RDF スキーマによるリソース (概念) 間の関係を拡張する表現およびその枠組みとみなされている [4]。2012 年の定義 [103] では、ドメインおよびそれに関連する利用者の組織で共有されている用語を形式化する語彙であると定義されている。すなわち、Web 空間でのオントロジーは、全ての用語を他の用語との関係で表現することを形式化する語彙であると

定義されている²⁴.

加えて、オントロジーの記述内容には、事実をありのままに述べるのみで、述べた事実以外にも真が存在する可能性（開世界仮説：Open world assumption）がある [9]. この点で、2.1 節で示した Semantic Web 実現に向けた原則に沿っており、言及している事以外は全て偽とする閉世界仮説（Close world assumption）を採るリレーショナルデータベースとは異なる²⁵.

さらに、Semantic Web の下では、あるトピックに関して構築されたオントロジーは Web 空間内に分散して存在すると考える。よって、当該トピックの表現を拡張するために、別のオントロジーを取り込んで利用することも考慮されている [10].

以上から、オントロジー構築技術は、様々な文化および言語を背景に持つコレクション特有のリソースおよびその知識に対しても、機械可読データとしての記述とその共有を Web 空間で可能とする糸口となり得る。

2.4.3.3 ウェブ・オントロジー言語 OWL の概要

オントロジー記述言語である OWL (OWL Web Ontology Language) [103] は、全て RDF で記述される。主に、知識構造の表現を行う際に、2.3.2 節の RDF スキーマのみでは不足する場合に、OWL で規定されているより柔軟な表現で補完する。その際、オントロジーが記述されたデータの共有を円滑に行うために、RDF/XML 構文でのシリアル化が必須と W3C で取り決められている。

OWL による知識表現は基本的に以下の概念で構成されている [9].

公理 (Axioms) OWL オントロジーによる基本的命題。

エンティティ (Entities) 実世界の事物を参照するために使用される構成要素。クラス (Classes, 2.3.2.1 節参照.), 属性 (プロパティ, Properties, 2.3.2.2 節参照.), 個体 (Individuals) に大別される。

OWL の世界では、個体は何らかのクラスに必ず属しているとされている。つまり、個体は、何らかのクラスのインスタンス (Instance) である。文化財書誌上ではメタデータが個体に相当する。個体を定義する際に用いられる主なプロ

²⁴以降、本研究でのオントロジーは全て 2012 年に W3C 定義のオントロジーを意味する。

²⁵リレーショナルデータベースの「リレーショナル」は閉世界においての関係性を指す。

パーティは, 2.3.2.2 節の `rdf:type`, 2.4.3.4 節の `owl:sameAs`, `owl:AllDifferent` である.

式 (**Expressions**) 簡易な記述から複雑な記述を構成するような, エンティティの組み合わせ. オブジェクト指向言語のコンストラクタのように, オブジェクトの基本構成を宣言する式を, エンティティの組み合わせで表現できる.

実際のオントロジー構築では, 基本構成を以下の通りとする.

- ヘッダ (Header)
- エンティティの定義
 - クラス定義
 - プロパティ定義
 - 個体の定義

記述対象オントロジーのヘッダには, バージョン, 作成者, 作成日時などのメタデータおよび外部オントロジーの名前空間 (Namespaces) と対応する略記号を記述する.

また, `owl:import` の記述により, 別途作成したオントロジーや Linked Data を対象オントロジーに追加する機能がある. 特に, 外部組織で作られた Linked Data を参照可能となったオントロジーでは, 実際に外部組織で用いられている個体の記述が, インポート先のオントロジーの枠組みに当てはまるかを確認できる. そして, 整合性が確認された新たなオントロジーでは, 推論により導出した新たな公理や制約を利用可能となる. すなわち, 異なる知識構造の融合, その融合による推論から導き出された新たな公理, それらの公理の再利用が可能となる. 実例として, 本研究で作成したオントロジーに Linked Data をインポートした結果を 3.5 節図 3.16 に示す.

2.4.3.4 ウェブ・オントロジー言語 OWL の語彙

実際にオントロジーを記述する際には, RDF スキーマに加え, より具体的なエンティティの振る舞いを記述する必要がある. なぜなら, 開世界を基にした OWL で

は、あるプロパティを定義するとき、その主語と目的語はどんな種類のリソースや値でもよいことになるためである。したがって、通常は、エンティティの定義に公理 (Axioms) および制約 (Restrictions), すなわち対象分野の常識を追加して、その振る舞いを制御する。加えて、現実に応じた制約を組み合わせることで、高精度な推論が可能となる。

OWL でより具体的な公理および制約を宣言するためには、あらかじめ用意されている以下の語彙を用いる。本論文では、3 節で作成するオントロジーに使用した語彙を中心に説明する。

OWL 構文など詳しい情報は、[9] の他、[10], [104] に記載されている。

クラスおよび個体に関する性質の明示

owl:sameAs 二つの個体が等価であると宣言する。

owl:differentFrom 主語の個体は目的語の個体とは異なると宣言する。

owl:AllDifferent 列挙した個体が全て異なるものであると宣言する。

owl:equivalentClass 二つの異なるクラスが等価であると宣言する。

owl:someValuesFrom 主語のクラスは目的語で指定したクラスあるいは個体を一つ以上含むと宣言する。

owl:allValuesFrom 主語のクラスは目的語で指定した全てのクラスあるいは個体を含むと宣言する。

owl:oneOf 主語のクラスは目的語で列挙した個体のリスト中から一つの個体を持つと宣言する。

owl:disjointWith 主語と目的語のクラスは共通する個体を持たないと宣言する。

owl:unionOf 複数のクラスに共通する特徴を持つ個体からつくられる新しいクラスを定義する。

owl:intersectionOf ある個体が属するクラスが、指定する複数のクラスのいずれかに該当する場合、それらの個体の集合を新しいクラスとして定義する。

owl:complementOf 主語のクラスは目的語で指定したクラスに所属する個体の集合を含まないことを宣言する.

プロパティに関する性質の明示

owl:equivalentProperty 二つの異なるプロパティが等価であると宣言する.

owl:InverseFunctionalProperty 例えば, ウェブ上で一意とされる書籍のシリアル番号など, あるデータ値から主語が特定できるプロパティであると明示する.

owl:FunctionalProperty ある主語に対応する目的語が一意に定まるような関係を明示する. 例えば, 「山の合言葉は川である」と合言葉が決められている場合が相当する.

owl:inverseOf あるプロパティの意味とは反対の意味を持つプロパティであると宣言する. 例えば, 夫婦関係では「夫がいる」の対となる意味は「妻がいる」が相当することを明示する.

owl:SymmetricProperty 主語と目的語を入れ替えても関係が成立するプロパティ. 例えば, 「親子である」「夫婦である」という関係は, 主語と目的語を入れ替えても意味が通る.

owl:TransitiveProperty 継承関係が通用するプロパティであることを明示する. 例えば, あるプロパティが主語 P と目的語 Q のトリプル, 主語 Q と目的語 R のトリプルを成立させている時, 主語 P と目的語 R のトリプルでも当該プロパティが成立する時に用いる.

owl:cardinality あるプロパティに繋がる目的語 (個体) の出現回数を宣言する.

owl:minCardinality あるプロパティに繋がる目的語 (個体) の最低出現回数を宣言する.

owl:maxCardinality あるプロパティに繋がる目的語 (個体) の最大出現回数を宣言する.

オントロジーは、公理、制約およびエンティティを用いて形式的に表現した知識の集合とも言える。ただし、知識の集合が整合性のある集合かどうかは、論証 (Reasoners) ツールを用いて検証する必要がある。

2.4.3.5 文化財オントロジーの概要

文化財の書誌記述では、人、時代、場所、等の諸概念が、ある作品にどのように関与しているかという視点で書誌を記述することが求められている。図書館および文化財関連機関では、作品構造に対応する概念の関連を構造化データで表現したものを書誌オントロジーとして Web 上に公開している。

FRBR_{ER} 書籍の分野では、FRBR_{ER}[43] と呼ばれる、書籍から創作表現を抜き出し、主題を分析可能にする形式オントロジーが、国際図書館連盟 (IFLA[42]) によって策定されている。1961 年に、パリ原則 [105] では、利用者が求める書籍の所在を発見し、内容を識別するために、図書資源、著作、著者、主題概念を軸にした目録規則を制定した。FRBR_{ER} は、このパリ原則をより詳細に展開したモデルである [7]。具体的には、利用者が求める機能要件を分析し、書誌データから読み取れる事物の概念を抽出し、その関連をモデル化する。FRBR_{ER} では、事物を実体 (Entity²⁶)、事物の持つ特徴を属性 (Attribute)、別の実体との関係を関連 (Relationship) と呼ぶ。三つのグループに大別した実体名を以下に示す [43][106]。

- 第 1 グループ：知的・芸術的な創作である実体クラス群: 著作 (work), 表現形 (expression), 体現形 (manifestation), 個別資料 (item)
- 第 2 グループ: 実際に製作や頒布、管理を行う実体クラス群: 個人 (person), 家族 (Family) および団体 (corporate body)
- 第 3 グループ：主題に関する付加的な実体クラス群: 概念 (concept), 物 (object), 出来事 (event), 場所 (place)

²⁶オントロジーでの Entity とは定義が異なり、FRBR ではリソースに該当する概念が Entity である (2.4.3.3 節参照.)。

FRBR_{ER} を書誌記述に採用する目的は、ある著作をどのような言語、表現方法、記録媒体、版、実物資料で利用者に配布するのかを分けて記述することで、利用者の求める著作につながる資料へ適切にアクセス可能にすることである。そのため、FRBR_{ER} モデルを基にする書誌記述は、以下に示す第 1 グループの実体に関する記述を中心としている [43].

- 著作 (work) : 個別の知的・芸術的創造 (a distinct intellectual or artistic creation) であり、手元にある資料 (item) に強く結びつく上位の概念を指す。例えば、具体的な資料には「源氏物語・若菜上」「源氏小鏡」「あさきゆめみし」など、様々な表現や形態をした作品が多数含まれており、それらは「源氏物語」と呼ばれる著作に結びついている。
- 表現形 (expression) : 著作の知的・芸術的実現 (the intellectual or artistic realization of a *work*) とされている。すなわち、著作「源氏物語」をどのような表現で我々に示しているかということである。書籍では日本語版、英語版の他、映画、絵巻あるいは楽曲などの表現方法が表現形に該当する。
- 体現形 (manifestation) : 著作の表現形の物理的な具体化 (the physical embodiment of an *expression* of a *work*) とは、作品のタイトルとその利用形態である。例えば、「源氏物語・若菜上」というタイトルおよび古活字版の書籍、あるいは「源氏物語 千年の謎」というタイトルおよび DVD-ROM という情報が該当する。
- 個別資料 (item) : 体現形の単一の例示 (a single exemplar of a *manifestation*) とされている。言い換えると、上記体現形で示す作品タイトルを入手した時に、実際に手に取れる、あるいはダウンロードできるコピーそのものを指す。

以上の第 1 グループの実体群に関する記述に加え、著作権を持つ人、著作に関連する事物に関する記述を行うことで、利用者の資料発見を支援する。

この FRBR_{ER} を用いたメタデータ構築例として、主に書籍タイトルを著作レベルで統制可能とした WorldCat[44] がある。国内では、日本の古典著作を実

際に同定した事例 [107] がある。また、日本目録規則 (NCR) 2018 年版 (仮称) [108] において、FRBR_{ER} を基盤とすることが予定されている。

CIDOC CRM 文化財の分野では、CIDOC (The International Committee for Documentation) Conceptual Reference Model (CIDOC CRM [109]) と呼ばれる形式オントロジーが国際博物館会議 (International Council of Museums, ICOM [110]) によって策定されている。CIDOC CRM は、文化財中に隠れた各実体に関わる出来事を明らかにするオントロジーである [111]。CIDOC CRM の拡張として、Europeana [17] の Europeana Data Model (EDM) [47] および東京国立博物館のミュージアム資料情報構造化モデル [112][113] があり、それぞれの組織で実装されている。これらのオントロジーは、CIDOC CRM で可能な記述範囲を実際の業務および資料の特性に対応する範囲に限定して独自開発したモデルである。

CIDOC CRM では文化財を記述するために必要であると考えられるクラスおよび関係プロパティ群が揃っており、実体の粒度および構造をデータ設計者の目的に合わせて自由に調節できる。しかし、言い換えると、実体構造化の自由度が高いゆえに、EDM やミュージアム資料情報構造化モデルのように、データ設計者が書誌記述の範囲を書誌利用者の要求に合わせて適切に設定しなければ、実用的な粒度でのメタデータ取得および記述が困難となる。

FRBR_{OO} CIDOC CRM の策定後、書籍の分野では、2008 年に FRBR_{ER} と文化財分野の CIDOC CRM を融合した Functional Requirements for Bibliographic Records Object-Oriented (FRBR_{OO}) [114] が構築された。FRBR_{OO} を構成するクラスは、文化財の記述に必要な広範な概念を対象にしている。その汎用性および構造化自由度は高く、クラス間の継承関係も記載されている。一冊の書籍から読み取れる創作表現について、それがどの部品かを特定しない FRBR_{ER} に対し、FRBR_{OO} では、挿絵、題字等の書籍を構成する個々の創作を部品と捉え、各部品レベルでの主題の分析および表現に関わる出来事を記述することが可能である。また、作品内容の受け皿となる媒体に関して、FRBR_{ER} の機能に加え、CIDOC CRM のクラスを用いて、従来困難であった版や媒体の違いに関する作品記述を詳細に行える。

FRBR_{OO} は、マンガオントロジー [115], スロベニアの民族音楽および舞踏に関する EthnoMuse[116], ポーランドデジタル図書館連盟の文化財書誌群 [117], 同人創作物 [118], 古典籍書誌 [119] にて利用が試みられているが, モデルの複雑さ [120] がネックとなり, これらの例では本格運用および普及には至っていない。一方, 書籍, 新聞および雑誌など, 同名タイトルでシリーズとなる作品あるいは逐次刊行物などの記述については, FRBR_{ER} のみでは記述が困難であることが指摘されていた。そこで, FRBR_{OO} の拡張である PRESS_{OO}[121] の開発および運用が IFLA (国際図書館連盟) により進められ, 2017 年 3 月に IFLA 標準として公開された。

その後, Europeana Metadata Model (EDM[47]), Bibliographic Framework as a Web of Data (BIBFRAME[122]) など, CIDOC CRM や FRBR_{OO} で培った, 作品中の出来事を記述するというアイデアを継承しながら, 概念モデル自体の規模をより単純化し利便性を高めたモデルが考案された。

EDM (Europeana Data Model) EDM は, 元々 Europeana で扱う様々な文化財メタデータを統合するために, CIDOC CRM を基に構築されたオントロジーである。Europeana の目的は, EU 各国の MLA 組織で所蔵する文化財書誌を統合することである。具体的には, 文化財書誌情報の Linked Data 化や書誌データの分析に利用可能な API の積極的利用を Web 空間で促進している。また, 書誌の再利用を可能にするため, クリエイティブ・コモンズ (Creative Commons[123]) を活用し, 権利関係の整備を進めている。また, オンライン文化資源の再利用条件を明確にするため, Deutsche Digitale Bibliothek[124] (ドイツデジタル図書館) とともに RightsStatements.org[125] を創設している²⁷。EDM では, Semantic Web の考え方を基本とし, 収集したメタデータを Linked Data 化することを目的としている。その方針を以下にまとめる²⁸。

- 文化財そのものと, その情報を記録したものに関する記述を明確に分離する。

²⁷文化財資源の現実的利用には, Linked Data 化などの技術的課題と機械可読化を含めた著作権に関する課題をクリアする必要がある。

²⁸“Reminder on EDM requirements and design principles” [47] より。

- 同一の文化財に対して複数の記述が存在し、その内容に矛盾があってもよい。
- 多様なデータ構造, 異なるオントロジーとの互換性を考慮する。
- EDMで書誌情報を連携するために必要なメタデータスキーマを提供する。
- 統制用語 (典拠情報) を含めた文化財周辺情報を提供する。

また, EDMでは, Linked Data 化による文化財周辺知識の拡充のみではなく, 文化財にまつわる出来事を記述する枠組み (Event-centric approach²⁹) を設けている。文化財そのものを記述する枠組み (Object-centric approach) に加え, 出来事を記述することにより, 書誌利用者は文化財の成立過程を把握可能となる。

EDMは, EUのMLA機関の他, DPLA (米国デジタル公共図書館) で採用されている。

BIBFRAME BIBFRAMEは, 2012年に米国議会図書館 (Library of Congress, LC) によって公開された。このモデルは, FRBR_{ER}と同様に, 実体同士の関連を記述することに加え, Linked Dataの要素を取り入れたもので, 実体, 実体同士の関係および属性にURIを用いて識別する。モデルを構成する実体は, 以下に示す著作 (Creative Work), インスタンス (Instance), 典拠 (Authority), アノテーション (Annotation) とされ, 作品内容とその表現の分析に重点を置くFRBR_{ER}とは異なり, 作品情報の簡易なLinked Data化に重点を置いたモデルとなっている。

- 著作 (Creative Work): 所蔵作品と強い関連性のある概念を指す。FRBR_{ER}の著作と等しい。
- インスタンス (Instance): 著作を具体化した作品のタイトルで, FRBR_{ER}の体現形と等しい。
- 典拠 (Authority): 著作およびインスタンスに関連する概念で, 例えば, 人名, 場所, 件名など, FRBR_{ER}の第2および第3グループに該当する実体群を指す。

²⁹[47]の5.3節“Event-centric approach”より

- アノテーション (Annotation³⁰) : BIBFRAME では, 上記の著作, インスタンスおよび典拠とは別に, これらの実体群に所蔵資料に関連する Web 上の多様な情報資源を接続するための仕組みが設けられている. アノテーションの主な目的は, BIBFRAME の枠組みで作成した書誌データに, 書誌を作成した機関以外のサードパーティーが付加情報を追加できるようにすることである. 例えば, 対象書誌に対する消費者レビューや批評家によるコメントなどがアノテーションに該当する.

2.4.4 RDF ストアおよび SPARQL 言語の概要

2.4.4.1 RDF ストアの概要

RDF データは, RDF ストアと呼ばれる専用のデータベースに格納することができる. RDF ストア (トリプルストア, Triplestore) に対しては, SPARQL (SPARQL Protocol and RDF Query Language) 言語 [126][127] を用いてトリプルを検索できる. 人やアプリケーションは, HTTP による SPARQL エンドポイントに REST (Representational State Transfer[128]) 機能を利用してクエリ (検索, 問い合わせ) を送信し, 結果を受信することができる. クエリは, グラフパターンの組み立てにより行われ, 結果はその組み立てパターンに合ったトリプルの提示で返される.

SPARQL エンドポイントでは, 特に大規模データセットに対する検索の際, 分析に必要なグラフパターンのみを抽出できる. また, 異なるデータセットを複数選択し, 横断検索する機能が提供されている [129]. さらに, 既存のノードを新しいプロパティで結んだトリプルをテンプレートにして, 新しいトリプルを生成し, RDF ストアに追加することができる. この方法で新しく構築されたトリプルは, 元のデータセット中には存在しないリンクであり, それらを含めたトリプルから単一のデータセットからは予想できない結果を得る可能性がある.

2.4.4.2 RDF ストアの実装

RDF ストアの実装に用いることができるアプリケーションには, Apache Jena[130], RDF4j[131], OpenLink Virtuoso[132] などウェブサーバに設置するもの, Javascript な

³⁰[122] の “Annotating the BIBFRAME model” より.

どを用いるウェブブラウザ用プラグインの `rdfstore-js`[133], SaaS³¹形態の `MetaBridge` [134] など様々な形態で存在する.

例えば, `RDF4j` は, 導入が比較的容易な Java の API である. ウェブサーバに `RDF4j` をインストールすると, RDF データの構築, 蓄積, 論証, SPARQL エンドポイントの提供が可能となる. 一般に, SPARQL エンドポイントでは, HTTP REST 形式でクエリとその結果を送受信する. そのため, アプリケーションの言語に依存することなく, 目的のデータ形式で RDF データを取得し, 処理を実行することができる.

Web サーバを管理できない場合も, 内容と RDF データ量が, 当該 SaaS の想定している利用形態と合致していれば RDF ストアおよび SPARQL エンドポイント提供サービスを利用できる可能性が高い. 本研究では, SaaS の `MetaBridge` を, 佐賀大学附属図書館の Web サーバ `http://www.dl.saga-u.ac.jp` 上に存在している `Ichiba` の URI に対する参照解決先に利用している. 参考として, 具体的な設定を付録 A に示す.

2.4.4.3 SPARQL 言語の概要

SPARQL のバージョンは 2017 年 10 月時点で 1.1 である³². 言語に関する具体的な説明は, [127][126] に掲載されているため省略する.

SPARQL 構文は, 名前空間の宣言と SQL に類似のクエリ文で構成されている. クエリ文は, クエリ実行者が求める RDF グラフパターンの集合である. グラフパターンとして, 主語 (Subject), 述語 (Predicate), 目的語 (Object) で構成されるトリプル文を列挙する.

ソースコード 2.2 に全トリプルを抽出するグラフパターンを示す.

ソースコード 2.2: SPO 文による全 RDF データの抽出.

```
1 select * where {?s ?p ?o} limit 10
```

³¹SaaS とは, Software as a Service の略で, 必要な機能 (ウェブブラウザを用いるアプリケーションなど) のみを提供するサービス形態.

³²以降, 説明は, SPARQL1.1 を前提とする.

ソースコード 2.2 の 7 行目は、「?s ?p ?o の形に並んだグラフを 10 行選べ」という意味である。where で囲まれた「?s ?p ?o」が、一般に SPO 文と呼ばれるグラフパターンの一例である。? は変数記号である。つまり、全てのデータは ?s ?p ?o の形に並んだトリプルであるため、全ての RDF データが結果となる。その際、「limit 10」と記述して、返す RDF データの数を 10 とする。もちろん、この部分は省略可能だが、実際に SPARQL エンドポイントで limit の無い SPO 文を実行するとサーバに負荷を掛けるため、通常は、limit を付けて取得結果の量を抑えるべきである。

他にも、より複雑なクエリを試すことができる。例えば、国立国会図書館 NDLA の SPARQL エンドポイント³³では、実例 [135] と共に、実際にクエリを試すことができる。

2.5 文化財書誌の解析に必要な自然言語処理

2.5.1 自然言語処理技術の概要

新聞、ウェブサイト、SNS、コールセンターに寄せられる質問、議事録、論文など、人の手には負えない膨大なテキストが世の中に存在し、日々生成されている。テキストを機械的に解析し、我々の生活に有効活用するには、コンピュータでの機械処理が欠かせない。

特に、我が国の文化財書誌データの多くは、日本語で記述されている。その内容には、多くの専門用語が含まれており、専門用語の定義や関係が文化財の専門家でなければ明確にならない場合もある。さらに、日本語は単語間に区切りがない。その上、主語や目的語、代名詞などが人間にとって自明の場合は、省略される場合がある。以上の理由から、日本語で書かれた文の解析には自然言語処理技術が必要である。

自然言語処理とは、電子化された、データベースでの検索用に正規化などが行われる前の、自然文のテキスト（自然言語）に対し、その内容の理解を機械処理によって行うことを目的とした研究領域である [136]。

自然言語処理は、以下の技術で構成される³⁴。

³³Web NDL Authorities SPARQL Endpoint : <http://id.ndl.go.jp/auth/ndla/?query=>

³⁴[136] から引用。

形態素解析 (**Morpheme analysis, Morphemic analysis**) テキスト文を単語に分割し, 各単語の品詞を特定する精度を高める技術.

固有表現抽出 (**Named entity extraction, NE**) 単語の意味クラスを同定する精度 (2.5.3 節参照) を高め, 未知の固有表現を推定する技術.

構文解析 (係り受け解析, **Syntactic analysis**) 単語列を文節にまとめ, 文節間の係り受け関係を特定する.

述語項構造解析 (**Predicate Argument Structure Analysis**) 文中では, 格情報 (主語や目的語) がしばしば省略される. 述語に対応する格情報が何かを特定する. 代名詞や指示詞の推定も含まれる.

語義曖昧性解消 (**Word-sense disambiguation**) 文中の単語が多義語の場合, 文中に対応した適切な語義を特定する.

同義・含意関係認識 (**Textual entailment recognition, TE**) 単語は異なるが, 同じ情報を含む言い回し同義関係として認識する技術.

本研究では, 書誌中の非構造化テキスト文に対し自然言語処理技術を適用する. 対象テキスト文は, 書誌の注記に記入されており, 句読点に区切られた句または文が一トリプルに対応する. よって, 形態素解析および固有表現技術を用いて, ほぼ全ての文あるいは句からのトリプル生成が行える.

2.5.2 形態素解析手法の種類と特徴

単語間に区切りのない日本語, 中国語, 韓国語などの言語は, テキスト文から単語を抽出する技術が必要不可欠である. 形態素解析手法には, 形態素単位の手法と文字単位の手法がある [137].

2.5.2.1 形態素単位の手法

あらかじめ単語の辞書を準備し, それを基に文を形態素 (文中で意味を持つ最小単位) に分解する. 辞書上の単語には, 単語名, コスト値, 品詞, 読みなどが登録

されている。コスト値とは、形態素の品詞同士が隣接する可能性を数値化したものである。形態素解析器には、先頭と末尾を含む全ての品詞の組み合わせに対応するコスト値のデータベースを持っている。例えば、文の先頭には、一般的に、助詞より名詞が位置する確率が高い。よって、先頭が助詞になるコスト値は、先頭が名詞になるコスト値よりも高くなる。品詞同士のコスト値が低いほど、隣同士に並ぶ可能性が高い。

形態素解析器は、辞書を基に単語分割を実行し、文頭から文末まで並ぶべき形態素の経路候補を作成する。次に、その候補の中から、品詞の隣接可能性データベースと、搭載辞書上のコスト値を照会して、最もコスト値の低い経路を見つけて結果を提示する。

ただし、ある分野に適応する辞書、コスト値のデータベースの分野適応とそれに必要なコーパス（正解と確認済みの解析結果）などの用意に大きなコストがかかることがある。

日本語を対象とした形態素単位の手法を用いる解析器には、MeCab[138][139]、JUMAN[140]、Chasen[141] などがある。上記の形態素間のコスト値設定の取り扱いが、各解析器によって異なる。

2.5.2.2 文字単位の手法

この手法は、MeCab、JUMAN などの形態素解析器の苦手な分野への適用と大量のコーパスの準備を軽減するために開発された手法である。この手法を用いた解析器に、Kytea[142][143] がある。

まず、解析器が、各文字間の単語境界を判定する。必要に応じて、特定分野の単語分割データ、品詞正解ラベルのついたデータなど、学習データを与える。この手法の利点は、形態素単位の手法と比べて、単語分割の失敗に対する対応を容易に行える点である。形態素単位の手法では、一部の単語分割に失敗した場合、搭載辞書中のコスト値を微調整し、再度辞書を構成し直す、コーパスを用意してコスト値を再編成するなどの工数がかかる。一方、この手法では、単語境界の誤りを、簡易な記号でマークアップしたデータを専用ツールキットで修正できる [144]。

2.5.3 固有表現抽出

2.5.3.1 概要

書誌, ウェブサイト, 新聞など, 人が取り扱うテキスト中には, 5W1H を読み取る上で重要な人名, 地名, 時などの単語が含まれている. このような, テキスト中のキーワードとなり得る単語は, 固有表現と呼ばれる. 固有表現抽出は, コンピュータが, 人のようにテキスト中から要点を抜き出すための自然言語処理技術の一分野である [5]. 固有表現抽出は, 質問応答システム, 機械翻訳, テキスト文の要約を行う際の要素技術として用いられている.

固有表現をテキスト文から抽出することで, 単なる単語抽出とは異なる利便を受けられる. 単語の組み合わせを固有表現とすることで, 特定の意味を持つ固有表現をより適切に抽出可能である. 例えば, 年月日の組み合わせをパターン化し, このパターンで出現する単語群を固有表現にできる.

2.5.3.2 固有表現の定義

どのような単語, あるいは単語の組み合わせを固有表現とするかは, 対象テキスト文の分野に依存する. 少なくとも, テキストから 5W1H の要素を汲み取れるように, 人名・組織名, 地名, 時, 専門用語に該当する単語を固有表現とする. 分野に特化したテキスト文からの情報抽出技術の向上と実用化を目指した MUC[5] では, 特に有用な出来事に関する情報をテキスト文から抽出するために, 固有表現を ORGANIZATION, PERSON, LOCATION, DATE, TIME, MONEY, PERCENT としている. 文化財分野であれば, 既存の書誌データや凡例, 目録記述規則を比較し, 共通して使用されている項目から固有表現のクラスを決定する. 本研究で決定した固有表現クラスは, 3 章の表 3.4 で示す.

2.5.4 検索システムの評価指標 F 値

3 章での自然文に形態素解析ツールによる単語区切り, 固有表現抽出プロセスなどパターン認識や検索システムなどの精度評価指標の一つである F 値を用いる. F 値はあるシステム (精度を求める対象) の図 2.3 および以下に示す適合率 (Precision)

と再現率 (Recall) の調和平均を採る. F 値が 1 に近づくほど検索精度が高い. F 値を求めるには, 事前に検索対象中の正解および不正解データ数をカウントする必要がある.

適合率 適合率は, あるシステムの全抽出解において, システム抽出した正解の割合である. つまり, システム抽出解にどれだけのノイズ (不正解) が含まれているかを示す.

再現率 再現率は, 全正解において, システムが抽出した正解の割合である. つまり, システムがどれだけ正解を網羅しているかを示す.

F 値は適合率 P および再現率 R の割合の平均 (調和平均) を採るため,

$$F = \frac{2PR}{P + R} \quad (2.1)$$

となる.

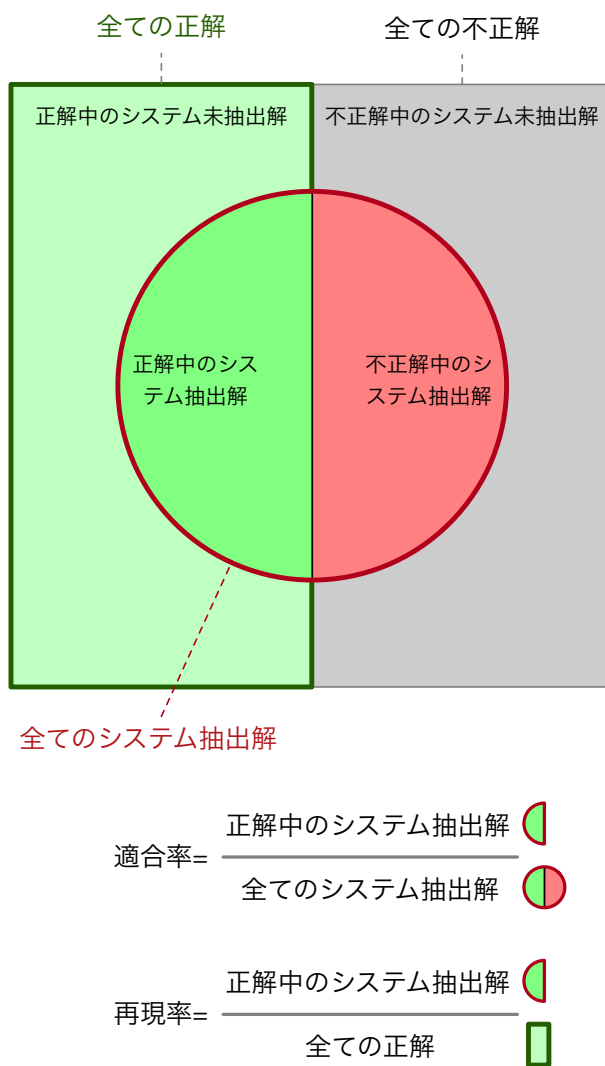


図 2.3: 適合率および再現率についての図解 [145].

第3章

非構造化書誌データから Linked Data への自動変換手法

3.1 提案手法の概要

本研究では、1.1 節で示したように、文化財書誌データを Linked Data に変換する際に掛かる多大なコストを可能な限り抑える手法を、実例をもって提案する。

本章、4 章および 5 章は、主要論文 [146] の内容を基に、適宜詳細な説明を付加したものである。

3.1.1 既存の古典籍書誌データの Linked Data 化が困難な理由

現在 Web 上で閲覧可能な古典籍書誌データの一例を、表 3.1 に示す。

表 3.1 の書誌データは、書誌項目 (属性) と値から構成されるメタデータである。このメタデータは、Linked Data として直接利用できない。その理由を以下に挙げる。

- 各書誌項目に入力されているメタデータには、「安政四年序」の「序」のような言葉が「安政四年」に追加されている。この文を人が読めば「安政四年」に「序」が書かれたと分かるが、アプリケーションによる自動処理には適さない。
- 編著者の値は、「(二世) 蓬萊山人作」である。この項目は人名を入力する場であるため、正規化すると「(二世) 蓬萊山人」であるべきである。同様に、刊行・書写年次の値は「安政四年序」であるが、「安政四年」であるべきであり、「序」は余分である。

表 3.1: 既存の古典籍書誌データの例. ([146] 表 2 から引用)

書誌項目	値
整理番号	45
分類	5-1-(2)-(6)-(6)
分類1	文学
分類2	国文
分類3	小説
分類4	-
分類5	洒落本
書名	青楼心得艸
読み	セイロウココロエグサ
書型	-
巻冊	-
編著者	(二世) 蓬萊山人作
刊行・書写年次	安政四年序
西暦	1857
巻写	写
注記	安政四年一月蓬萊山人序。 天明五年刊「息子部屋」ノ改題本ナリ。
印記	1
文庫	市場
他	-

- 注記項目は、簡略化された自然文で記述されている。一番目の文には、安政四年一月に「蓬萊山人」と呼ばれる人物が「序」を記述したと書かれている。蓬萊山人とは、おそらく「(二世) 蓬萊山人」を指すと推測されるが、国立国会図書館のNDLA[45]で人名典拠データを調査すると、蓬萊山人と呼ばれる人物は三人存在する。そのため、正確には誰なのか、専門家による明示が必要である。

上記に示す通り、表 3.1 のような古典籍書誌を読み解くには、書誌学、古典籍および歴史学に関する基礎知識を前提に、適切な単語抽出および単語の意味の確認を行う必要がある。そのため、古典籍書誌メタデータを機械可読化するには、古典籍分野の基礎知識を搭載した正規化を行う必要がある。

3.1.2 古典籍の読み解きに求められる正規化

本論文で必要となる正規化をまとめると、以下の通りとなる [1].

1. 書籍を記述する情報を、属性と値の組にする。各属性は書誌に凡例があればそれを用いる。各属性に対し、言葉の意味を明確に定義した上、URI を識別子として発行し、Web で一意に参照可能とする。さらに、各属性には適切な型を持つ値を設定する。特に、注記に含まれていた部品に関する書誌情報を属性と値の組として取り出す。
2. 古典籍書誌データの採録では、資料の解読からは値が確定されず、データ採録（提供）者の注釈・解釈が含まれていることがある [33]。そのため、書誌データに「?」、括弧、「～か?」等の注釈的記号・記述が含まれていることがある。これらの注釈的記号・記述を値から分離する。
3. 人名および分類名に対して、表記ゆれを抑制するため、日本国内であれば、国立国会図書館などの公的機関で発行されている典拠データの標目 URI を紐付ける。

3.1.3 提案手法の概要

表 3.1 のような既存の古典籍書誌データに対し、3.1.2 節の正規化を行い、Linked Data へ、可能な限り低コストで変換する手法の概要を図 3.1 に示す。

本手法は、以下に示す三つの手順で取得したデータ（表 3.2 の A, D, F および G の出力データ）を Linked Data 生成スクリプト（表 3.2H, 3.5 節）に入力することで得られる。手順の中には調整や確認といった若干の処理を必要とするものもある。

1. 固有表現抽出（3.2 節）
2. 外部機関で作成された標目 URI（以下、外部リンクと呼ぶ。）の整備（3.3 節）
3. 構造化データ（オントロジー）生成（3.4 節）

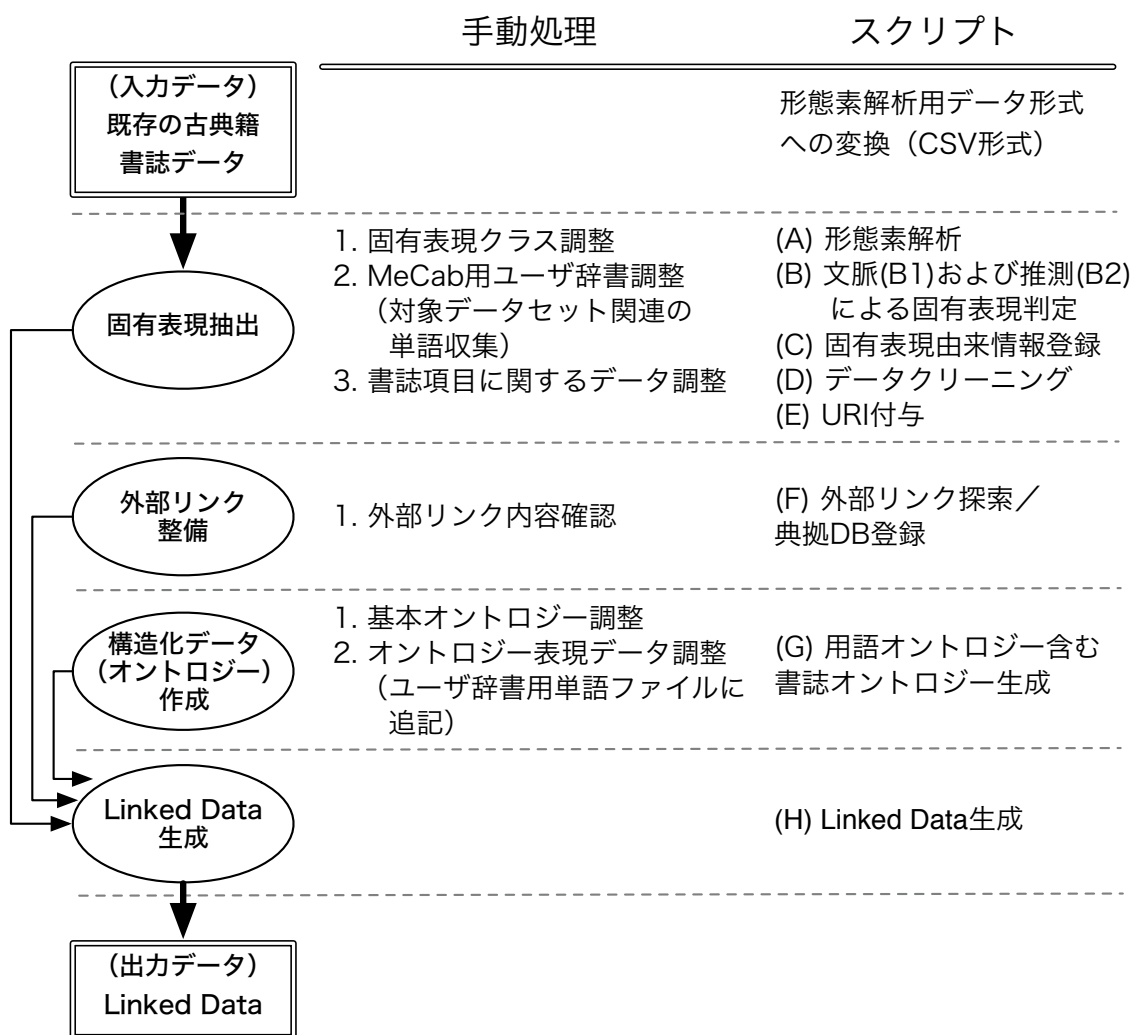


図 3.1: Linked Data 生成自動化の手順 [146].

提案手法の実行にあたり、書誌データを表 3.2 のスクリプトで実行し、Web 上に公開するデータ提供者、固有表現抽出状況、外部リンクおよびオントロジー表現データの手作業による内容確認は文化財の専門家が分担して行う。

スクリプト、データ提供者、コンテンツ自体の専門家による変換作業の分担について図 3.2 に別途示す。

3.1.4 使用データセット

古典籍とは、江戸時代以前に作られた書籍を指す。提案手法は、佐賀大学附属図書館所蔵の市場直次郎コレクション [147] の古典籍書誌データセット [148] (以下、

表 3.2: 自動化スクリプトおよびその入出力の概要. 図 3.1 に実行位置を示す [146].

略称	用途	入出力
A	形態素解析	(入) 書誌データ (出) 形態素解析結果
B	文脈 (B1) および推測 (B2) による固有表現 判定	(入) 同上 (出) 固有表現クラス由来情報 (CSV)
C	固有表現由来情報登録	(入) 同上 (出) 固有表現クラス由来情報 (関係データベース)
D	データクリーニング	(入) 同上 (出) 不要データを除去した由来情報
E	固有表現への URI 付与	(入) 同上 (出) 基底 URI と由来情報から整形した URI
F	外部リンク探索および 典拠データベース*登録	(入) 固有表現表記名および英語名 (出) 外部リンクの URI および統制語
G	書誌オントロジー生成	(入) 基本オントロジーおよびオントロジー 表現データ (出) 書誌オントロジー
H	Linked Data 生成	(入) A の書誌データ, D の固有表現由来 情報, F の典拠情報および G の書誌 オントロジー (出) Linked Data

CSV: コンマ区切りのデータ. (Comma Separated Values)

*外部組織が蓄積する一意に識別可能な単語情報を格納するデータベース.

Ichiba と記す.) を対象に開発してきたものである [8][1]. 当コレクションには, 古典籍の他に, 扇面, 短冊, 掛け軸, 色紙など様々な文化財が含まれている. また, 本研究で取り扱う書誌メタデータの記述対象となる古典籍の大半は, 江戸時代の日本で出版されている. 提案手法により Linked Data 化したデータセット¹および SPARQL エンドポイント²を公開している.

文化財は多種多様な事物を対象としている. 古典籍書誌を対象にする理由は, 比較的多数の件数が Web 上で公開され, 閲覧およびダウンロードの容易な分野であること, 異なるコレクション書誌間の属性が類似していることである. さらに, 古典

¹<http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/tensekiInstances.rdf>

²<https://www.metabridge.jp/sparql?lu=cwuboswnkghofpjoizlg>

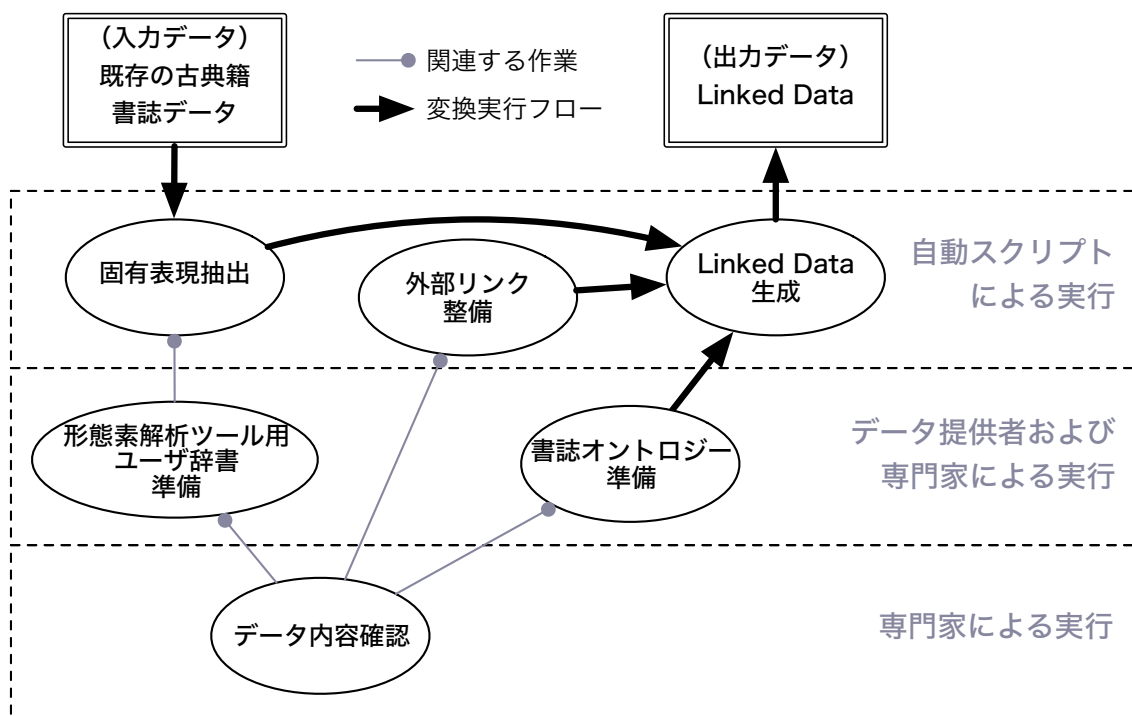


図 3.2: 自動スクリプト，データ提供者，コンテンツ自体の専門家による Linked Data 生成過程での変換作業の分担.

籍は，モノとしての文化財とともに書籍あるいは文書としての一面を有している。そのため，古典籍書誌は，MLA 連携を考慮した Linked Data 化を行う際に，汎用性の高い一例となる。以上の理由から，本研究では，文化財書誌の代表例として古典籍書誌を選択する。

3.2 固有表現抽出

固有表現抽出 (2.5.3 節) では，表 3.1 のような古典籍書誌データ全体に含まれる単語の中で，古典籍の読み解きに有用な単語群を抽出する。それらの単語群を固有表現，単語群のカテゴリーを固有表現クラスと呼ぶ。固有表現抽出プロセスの目的は，ある古典籍について記述したメタデータの属性と値を正規化し，構造化することである。

既存の古典籍書誌データの Linked Data 化が困難な理由 (3.1.1 節) で示したように，古典籍の場合，現在流通している書籍の書誌に準じ，古典籍の題名，著者名，刊行年など，一部の項目が独立して記述される一方，その他の詳細項目は注記に自然

文で記述されている。また、独立項目に対応する値も正規化されていない場合がある。そこで、固有表現抽出を行うことで、古典籍書誌データに対応する正規化(3.1.2節)を同時に行うことができる。

国立などの大規模組織が所蔵する古典籍の記述方法は、国立国会図書館の日本目録規則 [149]、国文学研究資料館 [19] の日本古典籍書誌レコード作成要領 [150]、アメリカ議会図書館の目録規則 [151] にて Web 上で公開されている。

例えば、国立国会図書館の日本目録規則の「古.0.3.2 各書誌的事項の情報源」および「古.0.4 記述すべき書誌的事項」によると、古典籍書誌の主な情報源は、タイトル、責任表示(著者、書写者、書肆名^{しよし}など)、出版・頒布情報、形態、シリーズおよび注記とされる。特に、記述すべき書誌的事項とされるタイトル、責任表示、出版・頒布に関する情報源は、表紙、見返し、序、跋(後書き)^{ぼつ}など、書籍の一部とすることが明示されている。また、「古.7.3.1 (タイトルに関する注記)」「古.7.3.2 (責任表示に関する注記)」においても、データ提供者は、これらの部品情報を、別途定められた略号を用いて注記することが推奨されている。

また、国内最大の古典籍データベース [19] を運営する国文学研究資料館 [19] の利用案内 [152] では、目録凡例の一例が参照できる。利用案内の「書名の種別」および「注記(書誌注記)」には、古典籍で注目すべき事柄を略記号を用いて表現していることが示されている。

〔「利用案内」 [152] から抜粋〕

書名の種別

- ‘内’ 巻首題
- ‘目’ 目録題(目録冒頭)
- ‘目中’ 目録中に記された題
- ‘扉’ 扉題
- ‘扉裏’ 扉裏に記された題
- ‘封’ 封面題
- ‘尾’ 尾題(本文末尾)
- ‘見’ 表紙見返しに記された題
- ‘裏見’ 裏表紙見返しに記された題

- ‘外’ 外題 (表紙、表紙題簽)
 - ‘序首’ 序首題 (古典籍総合目録データの場合、データソースに「序」とあるものを含む)
 - ‘跋首’ 跋首題 (古典籍総合目録データの場合、データソースに「跋」とあるものを含む)
 - ‘凡’ 凡例冒頭に記された題
 - ‘柱’ 柱刻題
 - ‘奥中’ 奥書の中に記された題
 - ‘序中’ 序中に記された題
 - ‘跋中’ 跋中に記された題
 - ‘刊’ 刊記に記された題
 - ‘裏表’ 裏表紙に記された題
 - ‘袋’ 近世版本等の書袋に記された題
 - ‘帙’ 帙ならびに箱等に記された題
 - ‘X’ その他 (耳題、欄外題、喉に記された書名、書根字 (小口書)、極札・極書に記された書名等)
-
- ‘A’ 主として巻首題
 - ‘B’ 広義の内題 (外題に対する内題の意)
 - ‘C’ 主として広義の内題
 - ‘D’ 外題あるいは巻首題
 - ‘E’ 外題・巻首題以外
 - ‘G’ 別名
 - ‘H’ 種別不明
 - ‘I’ 主として外題
 - ‘J’ 巻首題以外
 - ‘K’ 主として広義の内題に題簽の角書を冠す
 - ‘L’ 『国書総目録』の書名
 - ‘M’ 明細書・ターゲットの書名 (マイクロ資料目録データ)

‘N’ 当館で標目書名として定めた書名（和古書目録データ）

‘Z’ 書名不明・記載なし

注記（書誌注記）

〈系〉（系統注記）資料の系統に関することがら

〈著〉（著者に関する注記）

〈版〉（出版に関する注記）

〈写〉（書写に関する注記）

〈序〉（序跋注記）

〈形〉（形態注記）装丁、匡郭、行数、題簽の大きさ、料紙、その他

〈奥〉（奥書・識語注記）

〈書〉（書き入れ・校合注記）

〈伝〉（伝来注記）資料の旧蔵者とその蔵書印に関する注記

〈叢〉（叢書注記）叢書の細目について単独で記入を作成する場合、叢書に関する表示を示します。〔例〕〈叢〉群書類従 和書之部三の内。

〈般〉（一般注記）各データ項目の補足説明、その資料の特徴、性質等を示すことがら、その他上記の注記の種別に入らない注記

〈注〉（一括記入）上記の種別に振り分けをしないで、まとめて記入する場合

〈内〉（内容細目）

〈備〉（備考）データ作成者による注記

書名の種別とは、古典籍の題名の種類である。古典籍では、破損や汚れ等で、書籍名や著者、刊行年、現代書籍にて必須となっている事項が確定しない場合に、書名の候補として幾つかの部品にある文字列を採用する。つまり、古典籍には題が複数存在する可能性があり、部品の情報とともに部品上の識別可能な文字列が書誌に記載されている。注記に関しても、由来や管理、書籍の特徴などの補足事項とともに、序跋、奥書などの部品に関する事項を表す項目が含まれている。

このように、古典籍の分析に必要な事項の多くが部品名に対応して記載される。以降、これらの書籍の一部を示す用語を部品と呼ぶ。

本研究では、目録規則に用いられている部品名を抽出する。その部品名を固有表

現とする。同時に、人名、時、場所、専門用語、書誌データの内容、形状などの客観情報および成立状況など、研究者が求める周辺情報に関連する、部品名以外の固有表現クラスを特定する。

固有表現クラス名と特定した固有表現の一部を表 3.4 (文献 [1] の表 3 に専門用語および分類の項を追加) に示す。

表 3.3: 古典籍書誌データの固有表現クラスおよび固有表現の例 [146].

固有表現クラス	定義	固有表現の例
時	時を表す語	安政, 四年, 春, 卯月
地名	場所を表す語	江戸, 京都, 大阪
人名	人名・団体名を表す語	梅暮里谷峨, 栄久堂
題	作品名を表す語	息子部屋, 青楼心得艸
部品	書誌記述対象となる語	表紙, 序, 跋, 見返し
ロール	人名の役割を表す語	作, 画, 翻訳, 編, 写
専門用語	上記以外の専門用語	解題本, 折本, 後印
分類	分類名を表す語	文学, (分類名の) 洒落本

固有表現のデータを準備した後、書誌データからの固有表現抽出を実行する。具体的には、注記などの非構造化テキストに対し、句読点などの記号で一文単位に区切った後、MeCab[138] による形態素解析を行う (表 3.2A)。

その際、MeCab 用基本辞書の Unidic[153] に加えてユーザ辞書を用いる。ユーザ辞書には、固有表現クラスに分けた単語リストから作成する (付録 B.1, B.2, B.3 に例示.)。

時節用語 (付録 B.1) (二十四節気および雑節 [154], 月の名称および別名 [155], 十千十二支 [156], 年号 [157]), 人名典拠 [158] の標目および別名データ³を、固有表現クラスに分けた単語リストに登録する (図 3.3)。また、部品名 (付録 B.2), ロール名 (付録 B.3), 書誌学用語等の専門用語 (付録 B.4), 分類名を、古典籍総合目録データベースの「利用案内」[152] および *Ichiba* の凡例から抽出し追加する。さらに、*Ichiba*

³抽出スクリプト例を付録 B.5 に示す。本スクリプトの実行で 32 時間で約 100 万件 (別名込み) の XML データが抽出可能。相手サーバに負担を掛けないように注意。また、200MB 以上の XML ファイルから MeCab 辞書用データへの変換には、XMLReader (ストリーム方式) の PHP ライブラリが必要 (付録 B.8)。

の書名, 編著者名をキーワードとして自動的に収集した関連書誌データ (付録 B.6) から, 固有表現に相当する候補単語を抽出し追加する (表 3.4) [1].

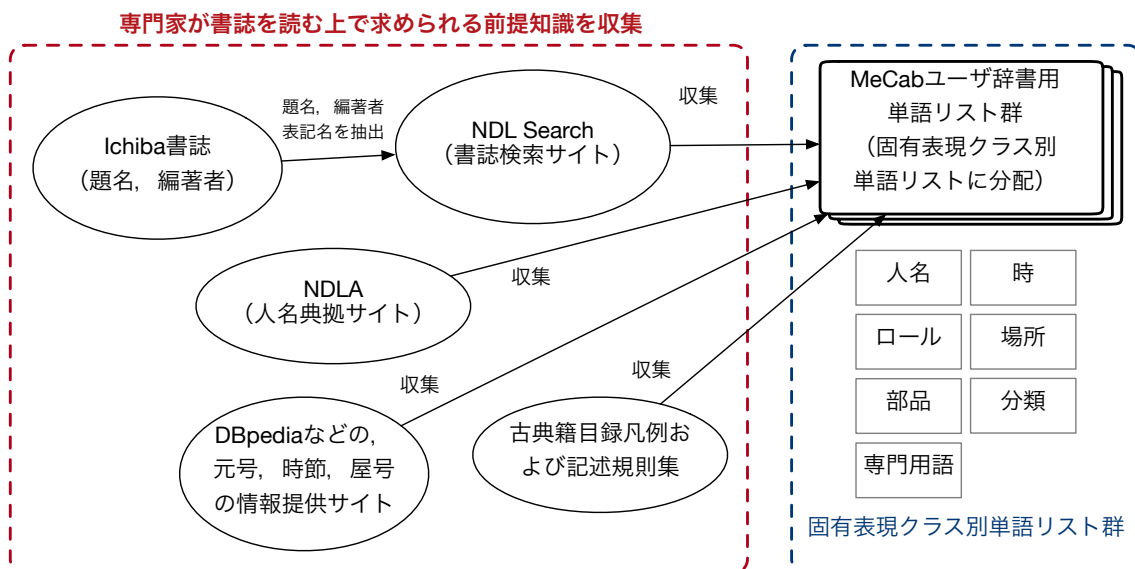


図 3.3: Mecab ユーザ辞書作成の流れ.

次に, 単語リストから形態素解析用ユーザ辞書を生成する. 形態素解析用ユーザ辞書の作成は, MeCab での手順 [159] に沿って行う. 本研究では, 単語リストから MeCab ユーザ辞書生成スクリプト群 (付録 B.7, B.9, B.10) をあらかじめ用いた.

また, MeCab 実行時に固有表現クラス名を識別可能とする解析用フラグを, ユーザ辞書作成用 CSV ファイルに付加する. 加えて, 辞書に登録する単語が基本辞書上の単語より優先的に判定されるように, 各単語に設定されている MeCab 累積コスト値を適宜調整する. 部品, ロールおよび専門用語クラスの単語リストはオントロジー表現データの入力にも流用する (3.4 節).

こうした準備の後, 形態素解析を実行し, 同定された単語に対して固有表現クラスの判定を行う (付録 B.12). まず, 各単語の解析結果に出現する固有表現クラス名のフラグで判定する. 具体的には, ソースコード 3.1 のような解析結果から, 各行末尾の「ICHIBA_DATETERMS」, 「ICHIBA_JINMEI」などの固有表現クラス名のフラグを判定材料にする (ソースコード 3.1 中の「ユーザ辞書搭載の解析結果」を参照).

表 3.4: 固有表現クラス別のユーザ辞書搭載単語数.

固有表現クラス	単語リストレコード数*
(参考) Mecab 基本辞書	1649863
時 (時節)	1312
地名	124
人名	765329
題	556
部品	85
ロール名	58
専門用語	199
分類	461

*2017 年 10 月時点での総数

ソースコード 3.1: ユーザ辞書未搭載および搭載の MeCab での解析結果.

```

1 //ユーザ辞書未搭載の解析結果
2 安政 名詞,固有名詞,一般,*,*,*,アンセイ,安政,安政,アンセー,安政,アンセー,
   固,*,*,*,*
3 四 名詞,数詞,*,*,*,*,ヨン,四,四,ヨン,四,ヨン,和,*,*,*,*
4 年 名詞,普通名詞,助数詞可能,*,*,*,ネン,年,年,ネン,年,ネン,漢,*,*,*,*
5 一 名詞,数詞,*,*,*,*,イチ,一,一,イチ,一,イチ,漢,*,*,*,*
6 月 名詞,普通名詞,助数詞可能,*,*,*,ガツ,月,月,ガツ,月,ガツ,漢,*,*,*,*
7 蓬莱 名詞,普通名詞,一般,*,*,*,ホウライ,蓬莱,蓬莱,ホーライ,蓬莱,ホーライ,
   漢,*,*,*,*
8 山人 名詞,普通名詞,一般,*,*,*,サンジン,山人,山人,サンジン,山人,サンジン,
   漢,*,*,*,*
9 序 名詞,普通名詞,一般,*,*,*,ジョ,序,序,ジョ,序,ジョ,漢,*,*,*,*
10
11 //ユーザ辞書搭載の解析結果
12 安政 名詞,普通名詞,一般,*,*,*,安政,安政,,安政,,和,*,*,*,*,ICHIBA_DATETERMS
13 四 名詞,数詞,*,*,*,*,ヨン,四,四,ヨン,四,ヨン,和,*,*,*,*
14 年 名詞,普通名詞,助数詞可能,*,*,*,ネン,年,年,ネン,年,ネン,漢,*,*,*,*
15 一 名詞,数詞,*,*,*,*,イチ,一,一,イチ,一,イチ,漢,*,*,*,*
16 月 名詞,普通名詞,助数詞可能,*,*,*,ガツ,月,月,ガツ,月,ガツ,漢,*,*,*,*
17 蓬莱山人 名詞,固有名詞,人名,一般,*,*,*,蓬莱山人,蓬莱山人,,蓬莱山人,,固,*,*,*,*,
   ICHIBA_JINMEI
18 序 名詞,普通名詞,一般,*,*,*,序,序,,序,,漢,*,*,*,*,ICHIBA_TFB

```

フラグで判定不可能な単語群に対して、固有表現クラス別に簡易な文脈判定 (B1) および推測 (B2) スクリプト (表 3.2B) を実行する。スクリプト B1 および B2 については、コレクションの書誌内容あるいは記述形式に対応した調整が必要となる場

合がある。 *Ichiba* の B1 スクリプト (文脈判定) の場合, 時クラスでは, 年号など時節用語, もしくは, 数字と年月日など時の単位の組み合わせ, 「同年」などの省略的記述が文中に出現した場合に該当単語群を「時」と判定する (付録 B.17)。 題クラスでは, 鉤括弧に括られた文字列を「題」と判定する (付録 B.14)。 人名クラスでは, 事前に対象分野および時代に使用された屋号 (屋, 堂, 亭など) あるいは呼称 (氏, 山人, 主人など) が文中に出現した場合に直前の単語を含めて「人名」と判定する [8] (付録 B.18)。

その他のクラスについては, スクリプト B1 を実行しない。

人名については, 書誌利用者ユーザ辞書に登録可能な情報が少なく, 固有表現抽出精度が他クラスより低い [1]。 そのため, パターン解析で判定できない単語に対し, 人名を推測する B2 スクリプトを利用して可能な限り抽出する (付録 B.19)。 例えば *Ichiba* 注記では, 一文に含まれる単語数が通常 10 語以下と簡潔で, 一文につき一つの出来事を含む。 また, 表 3.1 注記「安政四年一月蓬萊山人序。」のように, 部品名の直前に人名が並ぶ文体が多い。 そこで, 一つの注記文から, 時, 部品, ロールなどの判定済み単語および助詞, 記号類などの除外可能な MeCab の品詞を差し引き, 残った単語群が部品名の直前にあればそれを人名と判定する [1] (図 3.4)。

例1) 嘉永四年立秋廓遊舎文盛序。

／はMeCabによる単語区切り位置.

1	MeCab単語分割	嘉永／四／年／立秋／廓／遊／舎／文／盛／序／。
2	日付・専門用語類の同定	嘉永四年／立秋／廓／遊／舎／文／盛／序
3	部品・ロール名が文末にある文章に対し, 人名推定	嘉永四年／立秋／廓／遊／舎／文／盛／序 (E) ← (S)
4	人名ラベル付与	嘉永四年／立秋／廓遊舎文盛／序
5	クリーニング	嘉永四年立秋／廓遊舎文盛／序

*パターンマッチ 条件1: 「の」以外の助詞ではない・

条件2 「の」が単語の両端にある時は人名にせず探索続行.

条件3: 記号ではない.

条件4: 他の固有表現ではない.

例2) 大坂和多屋平兵衛

／はMeCabによる単語区切り位置.

1	MeCab単語分割	大坂／和／多屋／平兵衛
2	地名・MeCabによる人名の同定	大坂／和／多屋／平兵衛
3	屋号**・称号***が文中にある文章に対し, 人名推定	大坂／和／多屋平兵衛 (E) ← (S)
4	人名ラベル付与	大坂／和多屋／平兵衛
5	クリーニング	大坂／和多屋平兵衛

**屋号: 屋, 堂, 亭, 軒, 家, 社, 庵など

***称号: 翁, 氏, 散人, 山人, 産人, 主人など

図 3.4: スクリプト B2 (人名推定) の仕組みおよび具体例 [1].

スクリプト B1 および B2 を実行後、固有表現クラス名、レコード番号、項目番号、文に割り振った番号、文中の単語位置情報、単語および URI の項目で構成される固有表現の由来情報を、固有表現由来情報データベースに自動登録（表 3.2C）する。由来情報の URI については、図 3.6 の基底 URI にレコード番号、項目番号、文に割り振った番号、単語位置情報をハイフンで連結して組み立てる。

URI の構成は、2.2.2 節で示した通り、人にとっても理解しやすいことが推奨されている [65]。本手法では、具体的な単語位置を上記の方法で表現することで、特定の固有表現とレコード、属性、値の位置への対応が明確となる。

ソースコード 3.2: CSV 形式で表現した由来情報データベース (PostgreSQL) 上のデータ (一部)。左から、固有表現クラス名略記、レコード番号、項目番号、文に割り振った番号、文中の単語位置情報 (開始位置)、文中の単語位置情報 (終了位置)、単語、URI を表す。

1	Date,0,17,0,0,4,安政 4年 1月,http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/000-17-0-0-4
2	Person,0,17,0,5,5,蓬萊山人,http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/000-17-0-5-5
3	TFB,0,17,0,6,6,序,http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/000-17-0-6-6
4	Date,0,17,1,0,2,天明 5年,http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/000-17-1-0-2
5	Role,0,17,1,3,3,刊,http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/000-17-1-3-3
6	PNoun,0,17,1,4,7,息子部屋,http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/000-17-1-4-7
7	Term,0,17,1,9,9,改題本,http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/000-17-1-9-9

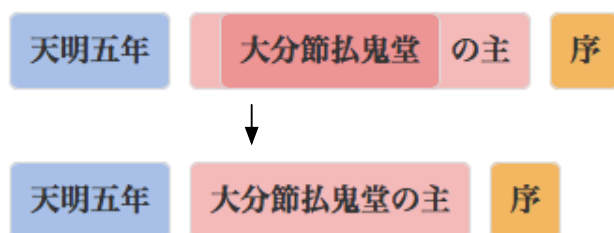
さらに、由来情報データベースに格納されている単語位置情報 (ソースコード 3.2) を利用して、単語の重複や連続、誤りなどを検出し、一つの単語に再構成するデータクリーニング (表 3.2D, 付録 B.20)⁴を実行する。

データクリーニング後、書誌データを図 3.6⁵に示す Linked Data にするために、元の書誌データの項目および値に加え、抽出した全ての固有表現および固有表現間の

⁴文献 [1] におけるクリーニングスクリプト。

⁵図 3.6 中の基底 URI は <http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/> である。

例1) 固有表現判定箇所が重複した部分はまとめる.



例2) 時クラスの固有表現が連続して並ぶ場合連結する.

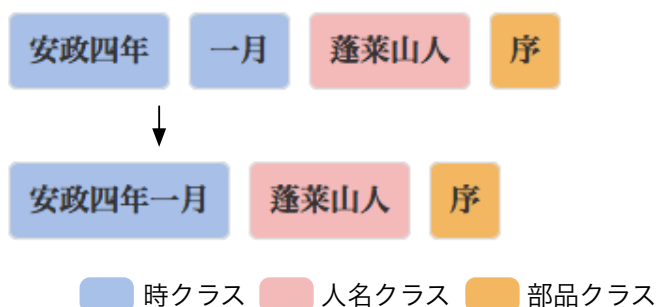


図 3.5: データクリーニングの具体例.

関係, 固有表現クラスに対し, 一意の URI を付与する.

そこで, 各固有表現に対し, 表 3.2E のスクリプト (付録 B.20) 内に明示.) で基底 URI⁶および固有表現由来情報に基づく URI (図 3.6 参照) を図 3.7 生成し, 由来情報データベースに登録する.

固有表現クラス自体の URI, 固有表現クラス間の関係を表す URI, 書誌データ項目を表す URI は, 基本オントロジー作成時にオントロジー作成ツール⁷で作成する. 部品, ロールおよび専門用語については, 3.4 節で用語オントロジーを生成する際 (表 3.2G) に, 各固有表現クラス名とリスト上の番号を組み合わせた URI を自動生成する (付録 B.23). これらの対象コレクションに関する URI が外部情報資源の URI と等価であればスクリプト F (典拠データベース蓄積, 3.3 節参照.) 実行時に紐付ける (表 3.2F).

また, 書誌データの各項目に対応する, 項目名, 固有表現クラス名, Dublin Core[70] などの標準メタデータスキーマなどをリストにした書誌項目に関するデータを作成

⁶Ichiba の基底 URI は <http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/> である.

⁷本研究では Protégé[160] を使用した.

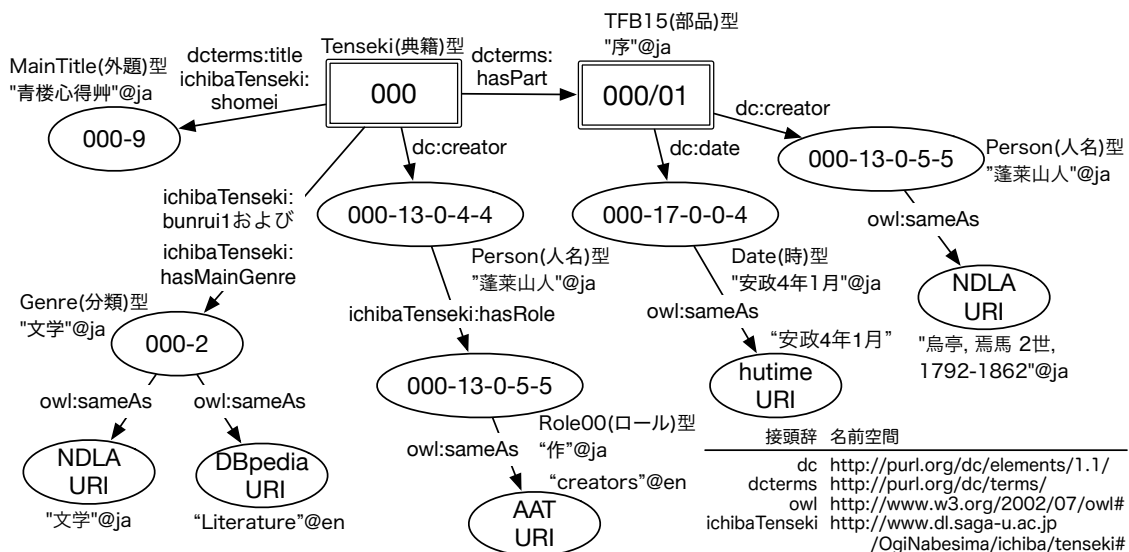


図 3.6: Linked Data 化された書誌データ (表 3.1) の一部 [146]. 「000」「000-13-0-4-4」などを基底 URI 末尾に付けることで URI となる. また, ダブルクォート内の文字列は URI に対応する識別用ラベル名をそれぞれ表す.

する (ソースコード 3.3). 書誌項目に関するデータは, 3.5 節の Linked Data 生成 (付録 B.24) 時に利用する.

ソースコード 3.3: *Ichiba* 書誌項目に関するデータ.

```
1 整理番号,seiribangou,dcterms:identifier,dataTypeNone,mainField,,,
2 分類,bunrui,,dataTypeNone,mainField,,,Genre
3 分類1,bunrui1,,,mainField,mecab,,,Genre
4 分類2,bunrui2,,,mainField,mecab,,,Genre
5 分類3,bunrui3,,,mainField,mecab,,,Genre
6 分類4,bunrui4,,,mainField,mecab,,,Genre
7 分類5,bunrui5,,,mainField,mecab,,,Genre
8 順位,juni,,,mainField,,,
9 画像公開,gazoukoukai,,,mainField,,,画像を公開許可,
10 書名,shomei,dcterms:title,,,mainField,,,代表書名,MainTitle
11 読み,yomi,dcndl:Transcription,,,mainField,,,書名の読み仮名,MainTitle
12 書型,shokei,dcterms:extent,,,mainField,,,典籍の大きさ,
13 巻冊,kansatsu,,,mainField,,,典籍の構成,
14 編著者,henchosha,dcterms:creator,,,mainField,mecab,,,Person
15 刊行・書写年次,kankou_nen,dcterms:date,,,mainField,mecab,年号で表現された日
    付,Date
16 西暦,seireki,dcterms:issued,,,mainField,,,西暦で表現された日付,Date
17 刊写,kansha,,,mainField,,,印刷および書写の別,
18 注記,chuki,dcterms:description,,,mainField,mecab,項目に当てはまらない詳細な
    記載や付記,
19 印記,inki,,,dataTypeNone,mainField,,,
20 文庫,bunko,,,mainField,,,コレクション名,Collection
21 他,hoka,,,dataTypeNone,mainField,,,
```

固有表現例

例1

安政四年一月

例2

蓬萊山人

例3

序

固有表現に対応するURI例

例1	Date: "安政4年1月" http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/000-17-0-0-4
例2	Person: "蓬萊山人" http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/000-17-0-5-5
例3	TFB: "序" A. http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/000-17-0-6-6 B. http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#TFB15

図 3.7: URI 付与ルール具体例. 例 1 は時クラス, 例 2 は人名クラスの固有表現にそれぞれ付与する URI, 例 3A は書誌レコード中に出現した固有表現としての URI, 例 3B は *Ichiba* 語彙中の部品名「序」を表現する URI.

3.3 固有表現と外部情報資源の接続

固有表現として抽出した単語に対して, DBpedia, The Getty Vocabularies[52], VIAF[46], NDLA などの外部組織で定義されている Linked Data 形式の単語情報に紐付ける. Linked Data 形式の単語情報には, Web で一意となる識別子として URI (外部リンク) が紐づく. そこで, 事前に用意した二種類の典拠データベース (ソースコード 3.4 およびソースコード 3.5) に抽出した固有表現に関連する典拠情報を蓄積する. 外部リンクの候補を集積する典拠データベース 1 には, 抽出した固有表現, スクリプト (表 3.2F) で自動探索した外部リンクおよび表記名を蓄積する.

加えて, 人名については, 同一の表記で異なる人物を指す可能性があるため, 典拠データベース 2 には, 抽出した固有表現, 固有表現の位置を表す URI, 外部リンク, 外部リンクと同じリソースを指す VIAF, 注記, 採用フラグを蓄積し, 抽出した固有表現に接続する外部リンクを特定し登録する. 採用フラグとは, 特定の固有表現 URI に対応する外部リンクが合っているかを目視で確認するためのフラグである. 採用フラグにより, 表記名の候補となる URI が複数存在する, あるいは, URI が存在しても確定ができない場合は, Linked Data への出力を制御することができる (ソースコード 3.5).

ソースコード 3.4: CSV 形式で表現した典拠データベース 1 (PostgreSQL) 上のデータ (一部). 左から, *Ichiba* 上の固有表現, 自動探索で取得した NDLA の URI, 統制語, 注記, 典拠データベース名, 固有表現クラスを表す.

```

1 為永春水,http://id.ndl.go.jp/auth/ndlna/00272104,為永春水 2世,,ndla,Person
2 一荷堂半水,http://id.ndl.go.jp/auth/ndlna/00312152,一荷堂半水,,ndla,Person
3 一筆庵,http://id.ndl.go.jp/auth/ndlna/00529592,一筆庵主人,,ndla,Person
4 一筆庵,http://id.ndl.go.jp/auth/ndlna/00269943,池田,英泉,1790-1848,古今秀
  句落し噺,ndla,Person
5  ...
6 安永3年1月,http://datetime.hutime.org/calendar/1001.1/month/安永3年1月,
  安永3年1月,,hutime,Date
7 安永3年8月,http://datetime.hutime.org/calendar/1001.1/month/安永3年8月,
  安永3年8月,,hutime,Date
8 安永8年,http://datetime.hutime.org/calendar/1001.1/year/安永8年,安永8
  年,,hutime,Date
9  ...
10 京,http://www.wikidata.org/entity/Q740246,京都,,dbpediaJ,Place
11 京都,http://www.wikidata.org/entity/Q740246,京都,,dbpediaJ,Place
12 広島,http://www.wikidata.org/entity/Q240429,広島,,dbpediaJ,Place
13 江戸,http://www.wikidata.org/entity/Q215646,江戸,,dbpediaJ,Place

```

```

14 ...
15 目首,http://www.wikidata.org/entity/Q1456936,Table_of_contents,,
    dbpediaE,TFB
16 目録,http://www.wikidata.org/entity/Q11581608,目録,,dbpediaJ,TFB
17 目録,http://www.wikidata.org/entity/Q250233,Catalog,,dbpediaE,TFB
18 標題紙,http://www.wikidata.org/entity/Q1339862,Title_page,,dbpediaE,TFB
19 ...
20 作者,http://www.wikidata.org/entity/Q3480291,作者,,dbpediaJ,Role
21 作者,http://www.wikidata.org/entity/Q14326950,Creator,,dbpediaE,Role
22 共著,http://www.wikidata.org/entity/Q3480291,作者,,dbpediaJ,Role
23 画,http://www.wikidata.org/entity/Q11578372,画,,dbpediaJ,Role
24 述,http://www.wikidata.org/entity/Q5273935,Dictation,,dbpediaE,Role
25 ...
26 刊本,http://www.wikidata.org/entity/Q11396303,刊本,,dbpediaJ,Term
27 黄表紙,http://www.wikidata.org/entity/Q3196335,黄表紙,,dbpediaJ,Term
28 朱印,http://www.wikidata.org/entity/Q11521518,朱印,,dbpediaJ,Term
29 書物,http://www.wikidata.org/entity/Q11515178,書物,,dbpediaJ,Term
30 ...
31 人情本,http://id.ndl.go.jp/auth/ndlsh/00576398,人情本,,ndla,Genre
32 人情本,http://dbpedia.org/resource/Ninjobon,人情本,,dbpediaJ,Genre
33 人情本,http://www.wikidata.org/entity/Q1100782,人情本,,dbpediaJ,Genre

```

ソースコード 3.5: CSV 形式で表現した典拠データベース 2 (PostgreSQL) 上のデータ (一部). 左から, *Ichiba* 上の固有表現, *Ichiba* 書誌上の URI, 自動探索で取得した NDLA の URI, VIAF の URI, 注記, 採用フラグを表す.

```

1 為永春水,http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki
    /115-17-0-35-35,http://id.ndl.go.jp/auth/ndlna/00272103,http://viaf.
    org/viaf/108681979,,t
2 為永春水,http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki
    /115-13-0-0-0,http://id.ndl.go.jp/auth/ndlna/00272103,http://viaf.
    org/viaf/108681979,,t
3 一荷堂半水,http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki
    /221-17-0-0-0,http://id.ndl.go.jp/auth/ndlna/00312152,http://viaf.
    org/viaf/29474440,,t
4 一荷堂半水,http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki
    /221-13-0-0-0,http://id.ndl.go.jp/auth/ndlna/00312152,http://viaf.
    org/viaf/29474440,,t
5 一荷堂半水,http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki
    /211-17-0-0-0,http://id.ndl.go.jp/auth/ndlna/00312152,http://viaf.
    org/viaf/29474440,,t
6 一荷堂半水,http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki
    /210-17-0-0-0,http://id.ndl.go.jp/auth/ndlna/00312152,http://viaf.
    org/viaf/29474440,,t
7 一筆庵,http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki
    /013-17-0-3-3,http://id.ndl.go.jp/auth/ndlna/00428743,http://viaf.
    org/viaf/257843733,,t

```


その後、典拠データベース上の単語定義情報を手作業で確認する。⁸

3.4 書誌オントロジーの定義および構築

3.2節で特定した固有表現クラスは、古典籍書誌の内容を構成する概念の集合である。オントロジー言語 OWL[9] を用いて、それらの関係を構造化することで、機械的処理が可能となる。

古典籍の読み解きを目的とした周辺情報の検索においては、書籍の構成部品に紐づく情報が求められる [162] ため、本研究では、目録規則における書誌記述対象である部品に注目して構造化する。そのため、EDM[47], CIDOC CRM[109], FRBR_{oo}[114] など、作品構成を作品成立にまつわる出来事として記述可能な文化財用形式オントロジーの考え方を基に、古典籍世界での書誌学のルールに従って簡易な書誌オントロジーを定義する。これにより、例えば、古典籍自体の著作情報とは別に、部品の表紙に表紙画の作者名、作者のロール、題名、作成日、作成地など、部品自体の著作情報を書誌データとして登録可能となる。

提案書誌オントロジー (図 3.8) は、古典籍と固有表現クラスおよび各メタデータ項目の関係を記述する基本オントロジーと、部品、ロールおよび専門用語について詳細な関係を記述する用語オントロジーを組み合わせた構成とする。基本オントロジーは、古典籍とその部品の関係の他に、代表書名となる外題と題名の候補である内題の関係のような、古典籍分野特有の関係を考慮しながら、オントロジー構築ツールを用いて作成する。本研究では、オントロジー構築ツール (以下、ツールと省略。) として Protégé[160] を用いる。本ツールにより、*Ichiba* の語彙情報⁹を構成する。具体的には、OWL でエンティティと呼ばれる知識表現に必要な構成要素である、クラス (Class)、クラス同士の関係 (Object Properties, オブジェクト・プロパティ)、個体¹⁰ (Individuals)、クラスと個体の関係 (Data Properties, データ・プロ

⁸探索自体は容易であるが、人名、地名などの固有名詞には表記揺れや別名が多く、特定の自動化は困難であり [32], 専門家による同定が必要である。適切な典拠データの同定を支援するために、例えば、Open Calais[161] のように、候補となる典拠に対して関連する周辺情報から確率を求めるような機械学習的仕組みの導入も選択肢のひとつとなる。

⁹*Ichiba* 語彙情報は <http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/index.rdf> からダウンロード可能である。

¹⁰あるクラスのインスタンスを OWL では個体と呼ぶ ([83] および 2.4.3.3 節参照)。

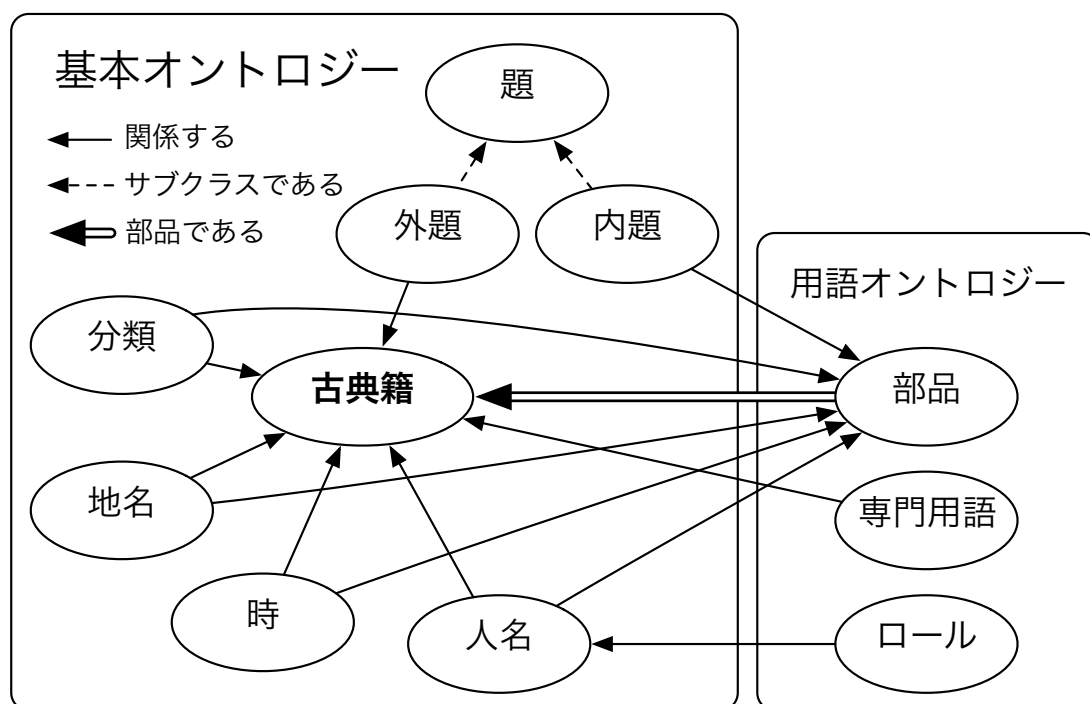


図 3.8: 簡易な古典籍書誌オントロジー [146].

ロパティ), 各クラス, オブジェクト・プロパティ, 個体などに繋がるリンクおよび補足事項 (Anotation properties, アノテーション・プロパティ) を記述する.

3.4.1 基本オントロジーの構築

3.4.1.1 クラスの定義

Ichiba の知識表現に必要な語彙となる概念をクラスとして表現する. ツール中にクラス定義を入力した結果を図 3.9 に示す. 図 3.9 では, 古典籍は本のサブクラスと定義する. 次に, 固有表現クラスであるロール, 人名, 分類, 場所, 専門用語, 時, 部品, 題を語彙中に追加する. さらに, 古典籍の世界では, 題のサブクラスに内題および外題があるため, これらも追加する.

3.4.1.2 オブジェクト・プロパティの定義

クラス同士の関係は, 図 3.10 および図 3.11 に示す関係となるように, ツールのオブジェクト・プロパティの項に定義する. その際, 古典籍の目録記述対象となる部



図 3.9: 古典籍書誌オントロジーの基になる, 基本オントロジーのクラス構成.
品, 外題, 内題の関係を明示する.

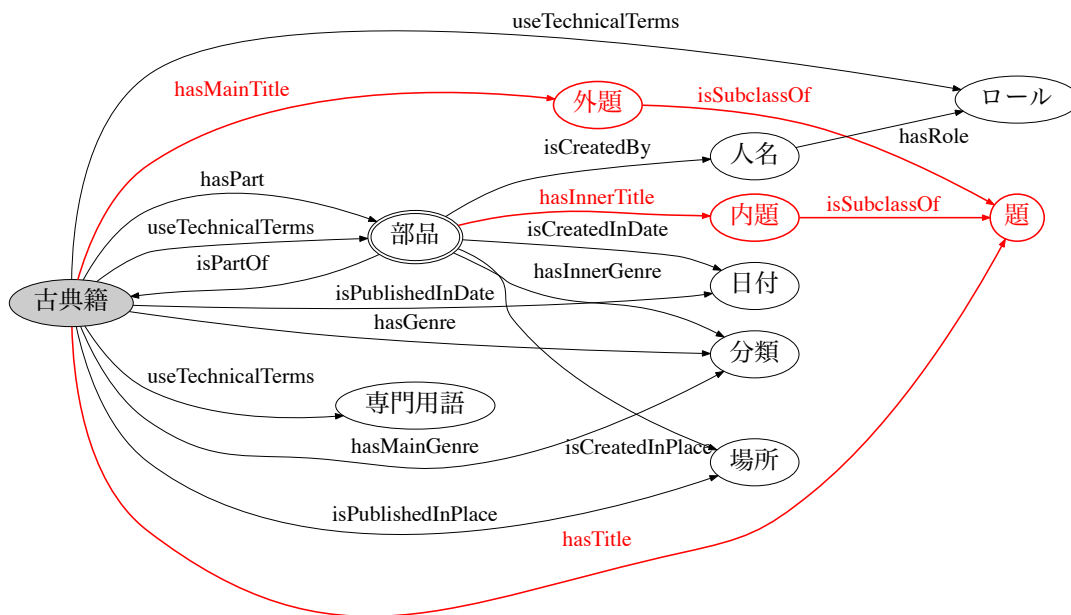


図 3.10: クラス同士の関係を定義した古典籍書誌オントロジー [1]. 赤字部分は外題および内題の関係を示す.

<ul style="list-style-type: none"> ■ hasPart ■ hasRole ■ useTechnicalTerms ▼ ■ isPublished <ul style="list-style-type: none"> ■ isPublishedInPlace ■ isPublishedBy ■ isPublishedInDate ▼ ■ hasTitle <ul style="list-style-type: none"> ■ hasInnerTitle ■ hasMainTitle ▼ ■ hasIdentifier <ul style="list-style-type: none"> ■ hasMainURI ■ hasSubURI ▼ ■ hasGenre <ul style="list-style-type: none"> ■ hasMainGenre ■ hasInnerGenre ■ isPartOf ■ isCollectionOf ▼ ■ isCreated <ul style="list-style-type: none"> ■ isCreatedBy ■ isCreatedInPlace ■ isCreatedInDate 	<table border="1"> <tr><td>hasPart</td><td>A hasPart Bで古典籍Aは部品Bを持つ.</td></tr> <tr><td>hasRole</td><td>A hasRole Bで人名Aは役割Bを持つ.</td></tr> <tr><td>useTechnicalTerms</td><td>A use TechnicalTerms Bで古典籍Aは専門用語Bを持つ.</td></tr> <tr><td>isPublishedInPlace</td><td>A isPublishedInPlace Bで古典籍Aは場所Bで出版される.</td></tr> <tr><td>isPublishedBy</td><td>A isPublishedBy Bで古典籍Aは人名Bにより出版される.</td></tr> <tr><td>isPublishedInDate</td><td>A isPublishedInDate Bで古典籍Aは時Bに出版される.</td></tr> <tr><td>hasTitle</td><td>A hasTitle Bで古典籍Aは題Bを持つ.</td></tr> <tr><td>hasInnerTitle</td><td>A hasInnerTitle Bで部品Aは内題Bを持つ.</td></tr> <tr><td>hasMainTitle</td><td>A hasMainTitle Bで古典籍AはBの外題を持つ.</td></tr> <tr><td>hasGenre</td><td>A hasGenre Bで古典籍Aは主分類名Bを持つ.</td></tr> <tr><td>hasMainGenre</td><td>A hasMainGenre Bで古典籍Aは分類名Bを持つ.</td></tr> <tr><td>hasInnerGenre</td><td>A hasInnerGenre Bで部品Aは分類名Bを持つ.</td></tr> <tr><td>isPartOf</td><td>A isPartOf BでAは古典籍Bの部品である.</td></tr> <tr><td>isCreatedBy</td><td>A isCreatedBy Bで部品Aは人名Bによって作成される.</td></tr> <tr><td>isCreatedInPlace</td><td>A isCreatedInPlace Bで部品Aは場所Bで作成される.</td></tr> <tr><td>isCreatedInDate</td><td>A isCreatedInDate Bで部品Aは時Bに作成される.</td></tr> </table>	hasPart	A hasPart Bで古典籍Aは部品Bを持つ.	hasRole	A hasRole Bで人名Aは役割Bを持つ.	useTechnicalTerms	A use TechnicalTerms Bで古典籍Aは専門用語Bを持つ.	isPublishedInPlace	A isPublishedInPlace Bで古典籍Aは場所Bで出版される.	isPublishedBy	A isPublishedBy Bで古典籍Aは人名Bにより出版される.	isPublishedInDate	A isPublishedInDate Bで古典籍Aは時Bに出版される.	hasTitle	A hasTitle Bで古典籍Aは題Bを持つ.	hasInnerTitle	A hasInnerTitle Bで部品Aは内題Bを持つ.	hasMainTitle	A hasMainTitle Bで古典籍AはBの外題を持つ.	hasGenre	A hasGenre Bで古典籍Aは主分類名Bを持つ.	hasMainGenre	A hasMainGenre Bで古典籍Aは分類名Bを持つ.	hasInnerGenre	A hasInnerGenre Bで部品Aは分類名Bを持つ.	isPartOf	A isPartOf BでAは古典籍Bの部品である.	isCreatedBy	A isCreatedBy Bで部品Aは人名Bによって作成される.	isCreatedInPlace	A isCreatedInPlace Bで部品Aは場所Bで作成される.	isCreatedInDate	A isCreatedInDate Bで部品Aは時Bに作成される.
hasPart	A hasPart Bで古典籍Aは部品Bを持つ.																																
hasRole	A hasRole Bで人名Aは役割Bを持つ.																																
useTechnicalTerms	A use TechnicalTerms Bで古典籍Aは専門用語Bを持つ.																																
isPublishedInPlace	A isPublishedInPlace Bで古典籍Aは場所Bで出版される.																																
isPublishedBy	A isPublishedBy Bで古典籍Aは人名Bにより出版される.																																
isPublishedInDate	A isPublishedInDate Bで古典籍Aは時Bに出版される.																																
hasTitle	A hasTitle Bで古典籍Aは題Bを持つ.																																
hasInnerTitle	A hasInnerTitle Bで部品Aは内題Bを持つ.																																
hasMainTitle	A hasMainTitle Bで古典籍AはBの外題を持つ.																																
hasGenre	A hasGenre Bで古典籍Aは主分類名Bを持つ.																																
hasMainGenre	A hasMainGenre Bで古典籍Aは分類名Bを持つ.																																
hasInnerGenre	A hasInnerGenre Bで部品Aは分類名Bを持つ.																																
isPartOf	A isPartOf BでAは古典籍Bの部品である.																																
isCreatedBy	A isCreatedBy Bで部品Aは人名Bによって作成される.																																
isCreatedInPlace	A isCreatedInPlace Bで部品Aは場所Bで作成される.																																
isCreatedInDate	A isCreatedInDate Bで部品Aは時Bに作成される.																																

図 3.11: オブジェクト・プロパティの構成 (左) およびその定義 (右) [1]

3.4.1.3 データ・プロパティの定義

データ・プロパティの定義とは、実存するデータセットをオントロジーに適用する際に、そのデータセットを構成する属性、例えば、表 3.1 の書誌項目がどの固有表現クラスに紐づくかをあらかじめ設定することである。加えて、書誌項目に対応する直値のデータ型も設定する。

その設定例を図 3.12 に示す。この設定例では、*Ichiba* の項目の一つ「刊行・書写年次」は時クラスであり、登録されている直値（例えば、安政 4 年）のデータ型は `rdfs:Literal` と定義する。

The screenshot displays the configuration for the data property `刊行・書写年次` (Publication/Scripture Year) in an ontology editor. The left sidebar shows a hierarchy of properties under `owl:topDataProperty`, with `刊行・書写年次` selected. The main area is divided into several panels:

- Annotations:** Shows two annotations for the property:
 - `rdfs:label` [language: ja] with the value `刊行・書写年次`.
 - `rdfs:comment` [language: ja] with the value `年号で表現された日付`.
- Usage:** Shows the property's usage, including a note that it is used in `isPublishedInDate` with a range of `xsd:anyURI`.
- Description:** Shows the property's characteristics and relationships:
 - Functional:** A checkbox that is currently unchecked.
 - Equivalent To:** No equivalent properties are listed.
 - SubProperty Of:** Lists `刊行年` (Publication Year) as a subproperty.
 - Domains (intersection):** Lists `時` (Time) as the domain.
 - Ranges:** Lists `rdfs:Literal` as the range.

図 3.12: データ・プロパティの設定例.

3.4.1.4 アノテーション・プロパティの定義

各エンティティには、外部リンクおよび補足事項を関連付けることができる。ツール上のアノテーション・プロパティの項目には、関連付けに使用する汎用語彙を登録する。汎用語彙には、RDFS (2.3.2 節) および 2.4.2.2 節に挙げた DCMI Metadata Terms, SKOS などを使用できる。

また、関連付けを行う際に、`rdfs:label` でリテラル型値を接続する場合は、データ型 (Datatype) および言語を値に付加する。

3.4.2 用語オントロジーの構築

次に、表 3.4 に示す部品、ロール、専門用語の固有表現クラスが内包する固有表現群について、その定義に加え、等価および包含の関係を用語オントロジーとして定義する。その際、各固有表現にオントロジー表現データと呼ぶ、固有表現の関係を表現した略記号を便宜的に付加したデータを、3.2 節で作成した MeCab ユーザ辞書用単語リストに追加する。オントロジー表現データは、固有表現表記、関係記号 (後述の「=」「:」「C」.)、検索性英語表記から成る。なお、固有表現表記および検索性英語表記については、等価な語彙定義を持つ外部リンクを探索し、典拠データベース 1 に格納するスクリプト F (表 3.2F) で利用する。

図 3.13 でオントロジー表現データの例を示す。例えば、固有表現「序」では、あらかじめ序の URI を TFB15¹¹ と定義する。この TFB15 は、固有表現「序中」と等価であるため、等号「=」を付加する。後に、等号が付加された固有表現は、等号の左側の固有表現、すなわち序と等価として登録する。序と題の複合語である「序題」の場合は、序と題の複合語と等しいことを等号とコロン「:」で = (序:題) と表現する。序題は、書誌学のルールで「内題」(クラス名の英語表記は「TitleOfWork」) のサブクラスとなるため、包含記号「C」を用いる。

オントロジー表現データおよび外部リンク情報を作成後、書誌オントロジー生成スクリプト G (表 3.2G, 付録 B.23) を実行する。スクリプト G にオントロジー表現データを入力すると、各固有表現に対し、クラス型、基本オントロジー (B.22 節

¹¹TFB は Terminology for bookbinding (製本用語) の略。序の URI は <http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#TFB15> に示す。

オントロジー表現	オントロジー表現データをスクリプトGで変換した結果										
序,,preface	<pre><owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#TFB15"> <rdfs:subClassOf rdf:resource="http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#TFB"/> <rdfs:label xml:lang="ja">序</rdfs:label> <rdfs:label xml:lang="en">preface</rdfs:label> <owl:sameAs rdf:resource="http://vocab.getty.edu/aat/300055032"/> <owl:sameAs rdf:resource="http://www.wikidata.org/entity/Q670787"/> </owl:Class></pre>										
序中,=序,	<pre><owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#TFB20"> <rdfs:subClassOf rdf:resource="http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#TFB"/> <owl:equivalentClass rdf:resource="http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#TFB15"/> <rdfs:label xml:lang="ja">序中</rdfs:label> </owl:Class></pre>										
序題, =(序:題) c内題, title from preface	<pre><owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#TFB21"> <owl:equivalentClass> <owl:Class> <owl:intersectionOf rdf:parseType="Collection"> <owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#TFB15"> <owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#TitleOfWork"/> </owl:intersectionOf> </owl:Class> </owl:equivalentClass> <rdfs:label xml:lang="ja">序題</rdfs:label> <rdfs:label xml:lang="en">title from preface</rdfs:label> </owl:Class></pre> <table style="margin-left: auto; margin-right: 0;"> <thead> <tr> <th>接頭辞</th> <th>名前空間</th> </tr> </thead> <tbody> <tr> <td>rdf</td> <td>http://www.w3.org/1999/02/22-rdf-syntax-ns#</td> </tr> <tr> <td>rdfs</td> <td>http://www.w3.org/2000/01/rdf-schema#</td> </tr> <tr> <td>owl</td> <td>http://www.w3.org/2002/07/owl#</td> </tr> <tr> <td>xml</td> <td>http://www.w3.org/XML/1998/namespace/</td> </tr> </tbody> </table>	接頭辞	名前空間	rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	rdfs	http://www.w3.org/2000/01/rdf-schema#	owl	http://www.w3.org/2002/07/owl#	xml	http://www.w3.org/XML/1998/namespace/
接頭辞	名前空間										
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#										
rdfs	http://www.w3.org/2000/01/rdf-schema#										
owl	http://www.w3.org/2002/07/owl#										
xml	http://www.w3.org/XML/1998/namespace/										

図 3.13: オントロジー表現の例 [146].

参照.)¹²で設定したクラス間の関係定義, 対象の固有表現と等価である外部リンクについての情報を読み込み, OWL で記述した用語オントロジー (B.23 節参照.)¹³を自動生成する. その後, 基本オントロジーと用語オントロジーを合成して, 書誌オントロジー全体を出力する¹⁴. これにより, 元の書誌データから, 提案書誌オントロジー中に記載されているデータ構造に準拠した Linked Data に変換可能となる.

合成済み書誌オントロジーの構文およびオントロジーの整合性は, Protégé で確認する.

3.5 提案オントロジーに対応する Linked Data への変換

最後に, 表 3.1 の書誌データに 3.4 節の提案オントロジーを適用し, RDF 形式のデータに自動変換する (表 3.2H, 付録 B.24). スクリプト H に書誌データを入力すると, 提案オントロジーで定義した古典籍同士, 古典籍と固有表現, 固有表現同士などの関係定義, スクリプト E (固有表現への URI 付与) および F (外部リンク探

¹²OWL ファイルは /source/ichiba/basicOntology.owl に配置.

¹³OWL ファイルは /source/ichiba/tensekiOntology.owl に配置.

¹⁴生成結果は <http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/index.rdf> から取得可能. また, /source/ichiba/bibliographicOntology.owl に配置.

索および典拠データベース登録)にて生成した URI 情報, 書誌項目に関するデータ上に記載されている標準メタデータを参照して, 書誌データ項目と値の関係を示すデータの組 (トリプル) を出力する.

なお, 生成した *Ichiba* の Linked Data¹⁵は, 古典籍書誌オントロジー (RDF/XML 形式データ)¹⁶上にインポートして, 個体レベルでの整合性を確認することができる (図 3.14, 図 3.15, 図 3.16 および `/source/ichiba/tensekiIndividuals.owl` を参照).

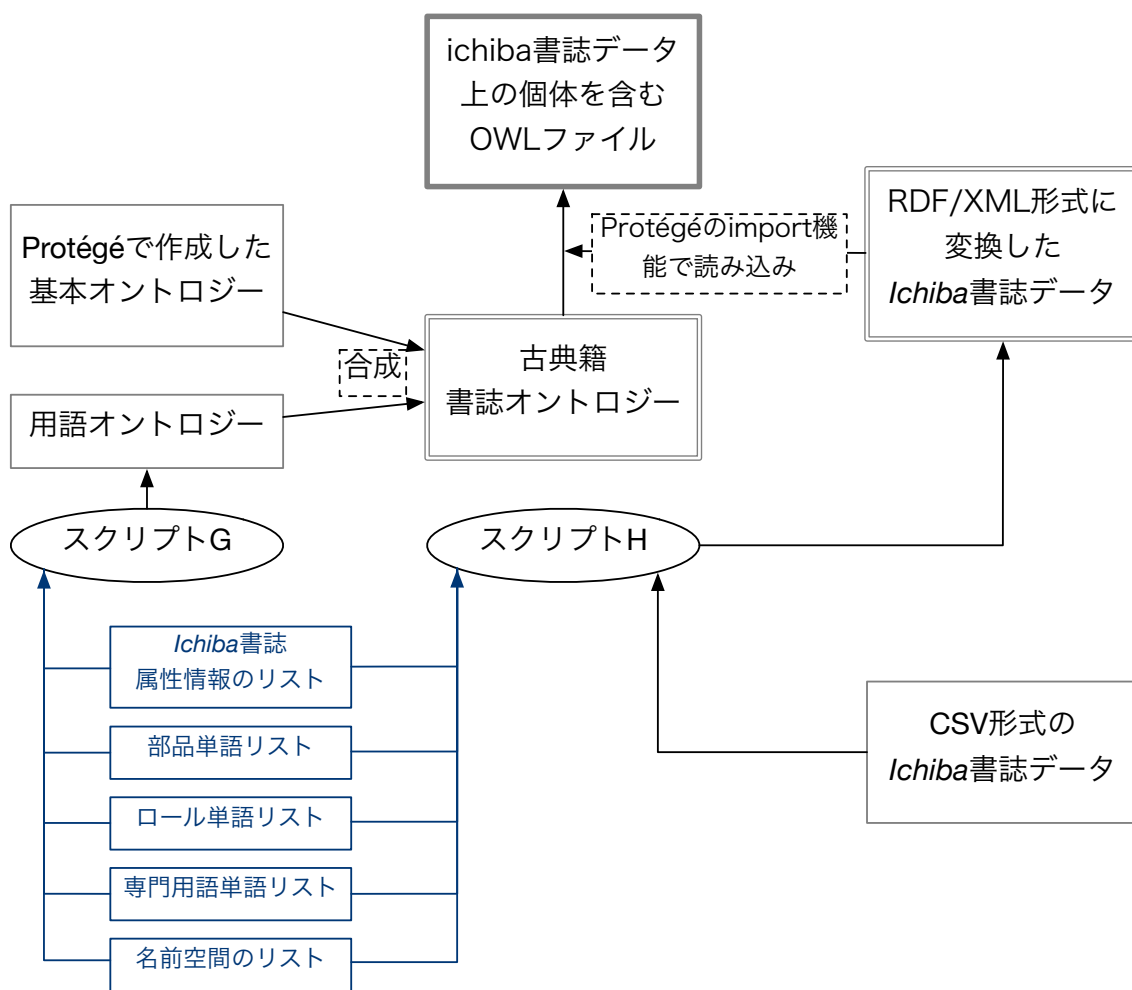


図 3.14: スクリプト G (書誌オントロジー生成) および H (Linked Data 生成) の流れ.

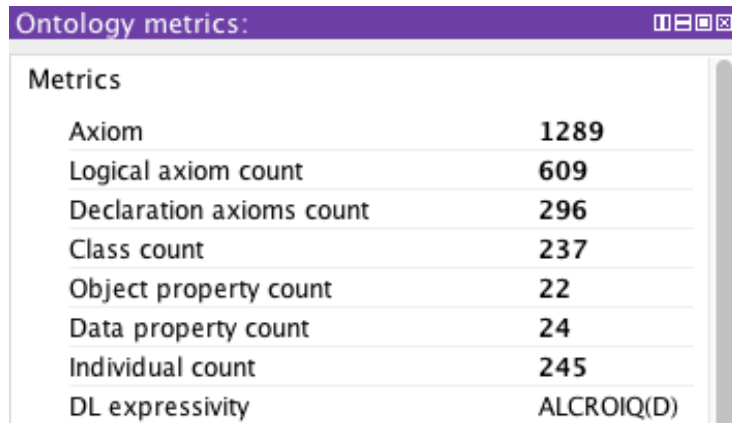
¹⁵<http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/tensekiInstances.rdf> からダウンロード可能.

¹⁶<http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/index.rdf> からダウンロード可能.

第 3 章. 非構造化書誌データから Linked Data への自動変換手法



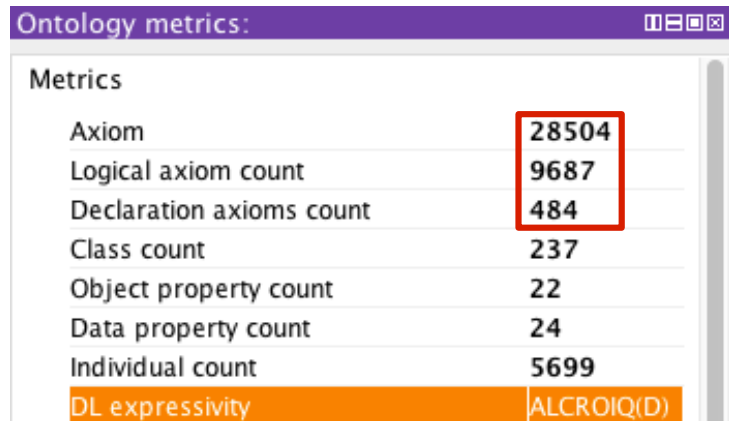
図 3.15: 古典籍書誌オントロジー上に Linked Data 化した *Ichiba* をインポートした際のスクリーンショット。



The screenshot shows a window titled 'Ontology metrics:' with a table of metrics. The table has two columns: the metric name and its count. The metrics listed are Axiom (1289), Logical axiom count (609), Declaration axioms count (296), Class count (237), Object property count (22), Data property count (24), Individual count (245), and DL expressivity (ALCROIQ(D)).

Metric	Count
Axiom	1289
Logical axiom count	609
Declaration axioms count	296
Class count	237
Object property count	22
Data property count	24
Individual count	245
DL expressivity	ALCROIQ(D)

Linked Dataをインポート前の*Ichiba*書誌オントロジー



The screenshot shows the same 'Ontology metrics:' window after Linked Data import. The table of metrics is updated. The 'Axiom' count has increased to 28504, 'Logical axiom count' to 9687, and 'Declaration axioms count' to 484. These three values are highlighted with a red rectangular box. The 'Individual count' has increased to 5699. The 'DL expressivity' is still ALCROIQ(D).

Metric	Count
Axiom	28504
Logical axiom count	9687
Declaration axioms count	484
Class count	237
Object property count	22
Data property count	24
Individual count	5699
DL expressivity	ALCROIQ(D)

Linked Dataをインポート後の*Ichiba*書誌オントロジー

図 3.16: 古典籍書誌オントロジー OWL データ上に Linked Data 化した *Ichiba* をインポートした結果、赤枠内の公理数が増加した。

第4章

提案手法の有効性

本節では、小規模な古典籍書誌データ *Ichiba* (佐賀大学附属図書館所蔵の市場直次郎コレクション, 典籍の部) および大規模な古典籍書誌データの *Kokubun* (国文学研究資料館所蔵の日本古典籍データセット [163] (表 4.1)) に提案手法を適用し, Linked Data 化について提案手法の有効性を検証する。解題は, 古典籍の解説であり, 注記と記載内容が異なるため, 本稿では形態素解析の対象としない。

表 4.1: 提案手法を適用したコレクションのサンプルレコード数および全レコード数。

コレクション名	サンプルレコード数	全レコード数 (2017年11月)
<i>Ichiba</i>	222	222
<i>Kokubun</i>	981	約 593,600

検証は, 以下の項目について, 人手のみで Linked Data 化を行う場合 (以下, 人手と呼ぶ。) と比較した場合に, 本手法による Linked Data 化は妥当であるかの観点から行う。

1. 提案手法の利点
2. 注記項目から固有表現抽出を行う意義
3. MeCab による単語切り出し精度
4. 固有表現抽出精度
5. スクリプト群の汎用性

6. 抽出単語の外部リンクへの接続の容易さ
7. 生成した Linked Data の評価
8. 提案手法の適用可能性

なお、自動化処理にあたり、*Kokubun* と *Ichiba* 間で共通の語彙が多いため、MeCab ユーザ辞書 (3.2 節) およびオントロジー表現データ (3.4 節) は、*Ichiba* のデータを基に追加して作成した。

4.1 提案手法の利点

古典籍書誌から Linked Data を人手で作成する作業では、Linked Data 化作業に携わる人が古典籍の書誌学、目録記述規則および周辺知識に加え、データ変換に関連する情報処理技術にも精通することが前提となる。提案手法では、人手で管理していた専門的知識をデータ化し、煩雑な情報処理を可能な限り自動化することで、一定の質を保つ Linked Data を生成できる。その結果、古典籍の専門家は確認作業に注力できる。

表 4.2: *Ichiba* および *Kokubun* に対する Linked Data 化にかかる時間。

作業	<i>Ichiba</i>	<i>Kokubun</i>
形態素解析, 固有表現抽出, データクリーニング (スクリプト A-E)	約 3 分	約 8 分
外部リンク探索 (スクリプト F) および手動での外部リンク選択	約 2 日	約 2 日
Linked Data 自動生成 (スクリプト H)	約 2 秒	約 5 秒

4.2 注記項目から固有表現抽出を行う意義

Ichiba および *Kokubun* の書誌項目で、固有表現抽出を行う項目を図 4.3 に示す。これらの書誌項目は、記録すべき項目に書誌学的な慣例が存在するため、類似している。

表 4.3: 固有表現抽出を行う項目 [146].

<i>Ichiba</i>	固有表現クラス	<i>Kokubun</i>
刊行・書写年次 西暦 注記	時	書写年 注記
注記	場所	書肆（書肆名に続いて記載） 注記
編著者 注記	人名	記載著者名表記 記載著者名よみ 記載著者名他等 注記
書名 読み 注記	題	統一書名 統一書名よみ 記載書名表記 記載書名よみ 注記
注記	部品	記載書名種別 注記
注記	ロール	記載著者名役割 注記
注記	専門用語	注記
分類 分類1 分類2 分類3 分類4 分類5	分類	オープンデータ分類 書誌種別

分類以外の固有表現クラスに注目する（図 4.3）。*Kokubun* の項目は、*Ichiba* より粒度が細かく、題（内題に相当する記載書名）、ロールおよび部品に相当する項目が独立して設けられている。加えて、*Kokubun* の注記項目にも、*Ichiba* と同様に様々な固有表現が記載されている。また、*Ichiba* および *Kokubun* とともに、登録された書誌の半数以上に注記の記載がある（表 4.4）。

表 4.5 に、全項目中に対する注記中の固有表現出現数の割合を書誌データおよび固有表現クラス毎に示す。*Kokubun* には、場所、題（記載書名）、部品、ロールに対する独立した項目があるため、*Ichiba* に比べて、注記に記載されている固有表現は少ない。また、専門用語および分類を除く固有表現クラス毎に固有表現出現割合

表 4.4: *Ichiba* および *Kokubun* における全レコード数および注記有りレコード数. カッコ内は全レコード中の出現割合 [146]

書誌データ	全レコード数	注記有りレコード数
<i>Ichiba</i>	222	210 (94.59%)
<i>Kokubun</i>	981	645 (65.75%)

表 4.5: 各固有表現毎の, 全項目中に現れる固有表現数に対する注記中のその割合 [146].

固有表現クラス	<i>Ichiba</i>	<i>Kokubun</i>
時	53.94% (130 / 241)	33.08% (171 / 517)
場所	98.37% (121 / 123)	4.39% (36 / 820)
人名	67.35% (427 / 634)	38.53% (1040 / 2699)
題	72.93% (598 / 820)	12.18% (902 / 7405)
部品	91.29% (346 / 379)	27.52% (1213 / 4407)
ロール	38.99% (124 / 318)	29.20% (535 / 1832)
専門用語	100.00% (134 / 134)	97.85% (683 / 698)
分類	0.00% (0 / 875)	0.00% (0 / 2274)

の平均を取ると, *Ichiba* の場合には 70.48%, *Kokubun* の場合には 24.15% の書誌が, 注記に固有表現を含んでいる.

4.3 MeCab による単語切り出し精度

MeCab 搭載の評価ツールを用いて, *Ichiba* および *Kokubun* の注記文に対する単語切り出し精度を計測する (表 4.6). *Kokubun* は, MeCab 基本辞書 Unidic[153] のみを用いた解析精度が *Ichiba* に比べて高い. その理由は, *Kokubun* の注記文が, 国立国会図書館などと同様に, 目録規則にて採録方法が明確に示されており, *Kokubun* の表記揺れは *Ichiba* よりも少ないことから, 単語切り出しが容易となるためである [150].

表 4.6: *Ichiba* および *Kokubun* に対する MeCab の単語切り出し精度 [146].

書誌名	コーパス単語数	F 値	
		ユーザ辞書無	ユーザ辞書有
<i>Ichiba</i>	5406	0.594	0.863
<i>Kokubun</i>	21730	0.840	0.993

4.4 固有表現抽出精度

表 3.2 で示した各スクリプトについて、固有表現クラスごとに抽出精度を測定した結果を表 4.7 および表 4.8 に示す。ただし、時クラスのクラス判定では、例えば、「安政」「4年」「1月」など、連続した単語群で時の意味を成す場合、「安政4年1月¹」で正解とする。

A として示した MeCab ユーザ辞書の作成による抽出精度は、両コレクションの全てのクラスで上昇した。特に、題、部品、ロールおよび専門用語クラスの抽出精度は 0.8 以上となった。一方、時、場所および人名クラスでは、時、人名および題に適用した B 群を適用することで、B 群未適用である場所クラスの判定精度も上昇した。*Ichiba* 人名クラスに適応した B1 および B2 により、判定精度は 0.429 から 0.518 に改善した。ただし、人名クラス用 B1 は *Ichiba* に対して有効である一方、*Kokubun* では精度の上昇は見られなかった。

D のデータクリーニングの実行では、時および人名クラスの単語群で、解析により形態素に分割されていた氏と名、年月日などが適切な固有表現にまとめられるため、検索精度は両コレクション共に上昇した。文字列の一括訂正についても、抽出精度を高める一助となった。例えば、*Kokubun* の場所クラスでは一文字の単語を多数判定し、適合率を低下させていたが、注記文中では全て誤りであるため一括して削除することで、検索適合率が高まり抽出精度が上昇した。

¹漢数字は時クラス用スクリプト B で算用数字に変換済み。

表 4.7: *Ichiba* のスクリプト (表 3.2) および固有表現クラス別固有表現抽出精度¹ (F 値) [146].

固有表現クラス	N ²	スクリプト名 (表 3.2 参照)			
		A	A+B1	A+B1+B2	A+B1+B2+D
時 ³	0.000	0.018	0.282	0.282	0.895
場所	0.579	0.685	0.768	0.768	0.768
人名	0.213	0.365	0.429	0.518	0.716
題	0.000	0.812	0.952	0.952	0.952
部品	0.869	0.911	0.911	0.911	0.911
ロール	0.698	0.800	0.838	0.838	0.838
専門用語	0.701	0.904	0.908	0.908	0.908

¹ 分類は固有表現抽出の対象外.

² N はユーザ辞書無しでの精度.

³ 時クラスは複数単語の組み合わせを正解としている.

4.5 スクリプト群の汎用性

表 3.2 に示すスクリプト群については, B1 (文脈判定), B2 (人名推定) および D (データクリーニング) を除き, 図 3.1 の手順およびスクリプトの構成を変更せずに, 各コレクションに対し若干の調整を必要とする. 本稿で比較対象とした *Kokubun* は, コレクションの分野, 時代, 書名, 著者など *Ichiba* と重複しているため, *Kokubun* 用ユーザ辞書およびオントロジー表現データに用いる単語リストは *Ichiba* で作成したリストに新たな単語を追加することで作成する.

時クラスに対する B1 (文脈判定) の適用では, 両コレクションとも同じスクリプトを用いる. *Kokubun* の題クラス判定用 B1 では, 注記文中に内題や外題を含む題クラスに相当しない題などが記載されるため, 他のクラスを判定するための材料として用いる. 人名クラス判定用 B1 に対しては, コレクションに合わせて文脈判定用の記述パターンに変更を加える. B2 (人名推定) は, 部品の前に人名が位置するパターンを探索するが, *Kokubun* の注記文には該当するものがないため適用しない.

このように, B1 および B2 は, 各コレクションでの書誌記述内容および目録記述規則に合わせて固有表現の抽出を試みる場合の補助手段として用いる. 仮に, 時クラス以外の判定で B1 および B2 の適用を省いても, 図 3.1 で提案する他のスクリプ

表 4.8: *Kokubun* の固有表現クラス別固有表現抽出精度¹ (F 値) [146].

固有表現クラス	スクリプト名 (表 3.2 参照)			
	N ²	A	A+B1	A+B1+D
時 ³	0.000	0.236	0.620	0.883
場所	0.050	0.533	0.571	0.973
人名	0.060	0.760	0.748	0.775
部品	0.609	0.990	0.990	0.990
ロール	0.618	0.981	0.981	0.981
専門用語	0.698	0.973	0.973	0.973

¹ 題および分類は注記からの固有表現抽出の対象外.

² N はユーザ辞書無しでの精度.

³ 時クラスは複数単語の組み合わせを正解としている.

ト群は実行可能である.

D (データクリーニング) は, 固有表現由来情報データベースに対し, 単語の不要な重複や分割の調整および訂正を行う. これらも, 軽微なスクリプトの変更で対応可能である.

4.6 抽出単語の外部リンクへの接続の容易さ

抽出単語の外部リンクへの接続の容易さを表 4.9 に示す. 3.3 節で述べた通り, 単語に対応する外部リンクの候補を Web から自動抽出することが容易な一方で, 外部リンクへの接続を確定することは, 書誌提供者の知識量に依存する. 本研究では, 古典籍に詳しくない著者が同定を行っているため, 専門知識の必要な人名, 部品, 専門用語, 分類の単語への接続リンク数が少ない (表 4.9). 一方, 時, 場所およびロールについては, hutime[164] および DBpedia などで Linked Data を取得可能な単語が増えているため, 同定は困難ではない.

4.7 生成した Linked Data の評価

提案手法によって非構造的な文から切り出された固有表現群は, 2.4 節で構成した書誌オントロジーの下, 自動的に Linked Data 化することができる. つまり, 部品

表 4.9: 書誌データ毎の固有表現クラス別固有表現出現数¹に対する接続リンク数の割合. カッコ内は接続リンク数および固有表現出現数 [146].

固有表現クラス	<i>Ichiba</i>	<i>Kokubun</i>
時	91.70% (221 / 241)	95.94% (496 / 517)
場所	81.30% (100 / 123)	84.15% (690 / 820)
人名	44.64% (283 / 634)	11.67% (315 / 2699)
題 ²	- (- / 820)	- (- / 7405)
部品	76.78% (291 / 379)	31.00% (1366 / 4407)
ロール	94.97% (302 / 318)	85.59% (1568 / 1832)
専門用語	96.27% (129 / 134)	33.95% (237 / 698)
分類	79.09% (692 / 875)	42.26% (961 / 2274)

¹ 全項目中の固有表現出現数.

² 現時点は古典籍著作典拠データにアクセス不可である.

や来歴を含む書誌情報を整合的に Linked Data とすることができる.

4.8 提案手法の適用可能性

前述のように, 提案手法は *Ichiba* を対象として構築を行った. この手法を *Kokubun* へ適用する段階では, 辞書用単語追加, 文脈・推測による固有表現判定およびデータクリーニングスクリプトに対し, *Kokubun* の注記規則に合わせて修正を行った. このように, 提案手法は対象コレクションに対する軽微な調整で適用することができる.

第5章

先行研究に対する本研究の位置 付け

古典籍書籍データの主に注記に含まれる，序や跋などの部品に関する情報を半自動的に Linked Data 化する手法を提案した．二つの規模の異なるコレクションに対して適用することで，その有効性を示した．コレクションへの対応の差は，注記の記述規則に応じたスクリプトの変更と形態素解析用辞書の追加であった．

3.2節の固有表現抽出手法は，非構造化テキスト文から固有表現となる単語を抽出する先行研究を参考にしている．例えば，川村と大須賀[165]は，自動でTwitter上の日本語ツイートあるいはニュース文からトリプルおよび固有表現抽出を行い，Linked Open Data化するWeb APIの開発を行った．その際，Subject（主語），Object（目的語），Action（述語），Location（場所），Time（時間）に加えて，Modifier（修飾句），Because（因果関係），Other（その他）の8種類を関係ラベル（固有表現クラス）と定め，文を構成する単語の属するクラスを判定した．自然文の解析には，係り受け解析器 CaboCha[166] および形態素解析ツール MeCab を用いた．MeCab のユーザ辞書には，研究対象文の分野に近い単語を収集したものを利用した．また，コーパスからある程度の法則性を見出し，類似パターンを基に単語がどの固有表現クラスに該当するか分類した．その法則性には，固有表現クラスに連続する助詞の並びをルール化したものを作成し，判定材料に用いた．また，固有表現クラスの不明な単語について，WordNet Japanese[167]上の類義語を探索し，クラス推定を行った．

他方，Ulicny の研究[168]では，ある固有表現 URI とそれ以外の URI との類似度を比較し，等価であれば owl:sameAs で紐付けるための判定材料として，Thomson

Reuter 社のニュース記事などの英語テキスト文から、抽出した単語の周辺にある単語との共起性を測定し、確率で判定する機械学習の手法を提案した。具体的には、テキスト文を自動で RDF データに変換するサービスとして、Open Calais[161] を利用している。

3.4 節の書誌オントロジーの構築では、オントロジーを基にした Linked Data の参考例として、EDM[47] を文化財書誌データに適用した Europeana[17] プロジェクトがある。EDM は、文化財オントロジー CIDOC CRM[109] を基に、多様なアーカイブ資料を統合しながら、作品を構成する部品と出来事の両方についての記述を行う。また、図書資料情報を提供する WorldCat.org[169] では、元々 FRBR_{ER} を基にしていたが、現在は、Linked Data の形式への対応を目的として、schema.org[170] による語彙定義と BIBFRAME[122] を基にした資料間の関係明示を行う [171]。

しかし、近年では、Linked Data の作成自体は数多く行われている一方で、上記汎用オントロジーあるいは独自ドメインオントロジーの利用例は少なく、Linked Data に使用されている語彙を構造化したデータが公開されていないことが課題となっている [172]。

そこで、これまで著者は、*Ichiba* を事例に、古典籍の構成に対応する部品情報を取得可能な Linked Data の作成を可能な限り自動化する手法を提案してきた。本研究では、研究対象として *Ichiba* に加えて、*Kokubun* を用いて自動化を実行し、提案手法が別の古典籍書誌データでも実行可能であることを示した。その結果、提案手法が手作業によるコストの最小化に有効であることを示した。ただし、語彙情報の正確性は、古典籍の専門家の協力によって担保される。

以上に示した提案手法に対する検証から、古典籍書誌に見るような書誌記述の視点、すなわち、本体とその部品に関する記述を行う文化財書誌に対して、提案手法による Linked Data 化は有効である。特に、本論文にて扱ったコレクションと類似分野では、一度作成した MeCab ユーザ辞書、OWL 形式の語彙情報の再利用が可能であるため、軽微な追加作業を通じて、他のコレクションの分析に応用することが可能である。

第6章

まとめ

本論文のまとめおよび展望について述べる。著者は、本論文を含む関連研究を通じて、多様な文化財の価値を高めるために、情報技術がどのように寄与できるのか、その方法論の一つを、実在する文化財書誌データを用いて提案してきた。

第1章では、文化財書誌の Linked Data 化が推進されている背景には、世の中に蓄積されている膨大なデータを我々の生活やビジネスに利活用する機運の高まりがあることを示した。しかし、従来の書誌データは、人が読むことを前提としており、アプリケーションなどで機械的処理することは考慮されていないため、昨今のデータ活用に対する需要を満たせない。そこで、本研究では、データの機械可読化およびその構造化を Web 上で実現することを目指した Semantic Web と呼ばれる概念と関連技術を用いて、文化財書誌データを Linked Data に変換する手法の確立を目的とすることを述べた。

第2章では、Web 空間で知識の共有を行うための基本概念である Semantic Web について、歴史的経緯を述べた上で、知識の構造化を行う上で重要な概念となる RDF, URI, オントロジー, 自然言語処理など、文化財書誌を実用的な Linked Data にする観点から実装に必要な概念について概説した。

第3章および第4章は、非構造化テキストによる記述が行われている文化財書誌の代表例として、Web 公開中の古典籍書誌データを採用し、それを実際に提案手法を用いて Linked Data に変換した上、本手法の有効性を、元の書誌データに含まれる情報の再現性、生成した Linked Data の整合性、変換に掛かる時間、生成あるいは手作業で作成したデータの再利用可能性などの観点から検証した。

第5章では、提案手法を確立する上で特に参考にした先行研究を紹介するとともに、それらの研究に対する本研究の位置付けを示した。

最後に、少し広い観点から、本研究の意義についてまとめる。

自然言語処理の精度向上 提案手法における自然言語処理では、古典籍をはじめとした文化財書誌に対応するよう先行研究の手法を拡張した。また、古典籍に対する書誌中の注記という短文で単純な構造である対象に処理を制限することで、テキストの構造をある程度制限することができた。さらに、使用される用語も制限することができた。その結果、高い精度で形態素解析を実施することができた。このことにより、構造が制限された文を対象とすれば、事前に使われる用語を収集することで、自然文からでも高度な技術を用いることなく情報を高い精度で取得できることを示した。つまり、自然文解析の適用範囲に関する可能性を拡げることができた。

Semantic Web 技術の活用 古典籍の書誌は、現代書籍の書誌よりも柔軟でかつ複雑な構造を必要としている。通常の関係データベースでは、注記のようなテキスト部分を残さざるを得ない。Semantic Web 技術は、古典籍の書誌のような複雑な対象のデータ化に非常に適していることを示すことができた。

古典籍の横断的分析 古典籍の書籍を機械可読な形式に実用的な時間で変換することができた。このデータを分析することで、書籍間の関係を人名や書誌を介して横断的に分析することが可能となる。国文学研究資料館などの大規模コレクション書誌が機械可読となれば、古典籍研究に新たな視点を提供することが期待できる。

文化財データの整理 古典籍以外に、道具などの有形の、あるいは演芸のような無形の文化財は、書籍よりも柔軟なデータ構造を必要とするであろう。それらのデータ化は、関係データベースを用いたのでは、多くの部分がデータ化されない可能性がある。Semantic Web 技術の柔軟な構造は、これらのデータ化に対応可能であることが予想される。さらに、機械可読となることで、統計的な分析手法が拡大し、比較研究の可能性が広がるであろう。

今後は、提案手法を原型として、世の中の9割を占めるとも言われている、非構造化データを伴う文化財書誌データを対象にした機械可読データへの変換技術の改良と、文理を問わず多様な分野の有識者による協働がMLAコミュニティを中心として発展していくことに貢献したいと考えている。すなわち、多くの有識者から賛同を得られる Linked Data が少ないコストで形作られるようになれば、災害の記録、判例、行政文書など、データ化されるのを待つ古い記録の掘り起こしが一層推進され、人文学分野での新たな価値の発見あるいは創造につながっていくと期待する。

参考文献

- [1] 吉賀夏子, 渡辺健次, 只木進一. 古典籍書誌注記文からの作品構造および関連人物のつながりを明らかにする周辺情報抽出. 人文科学とコンピュータ研究会 (CH) , 第 2016-CH-109 巻. 情報処理学会, 1 2016.
- [2] 保立道久. 歴史のひろば: 歴史データベースの将来と歴史知識学: コンピュータはただの便利な道具か. 歴史評論, Vol. 702, pp. 80–89, 2008.
- [3] 相田満. 和漢古典学のオントロジ. 勉誠出版, 2007.
- [4] Tom Heath and Christian Bizer. *Linked data: Evolving the web into a global data space*. Morgan & Claypool Publishers, 2011.
- [5] Ralph Grishman and Beth Sundheim. Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pp. 466–471, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.
- [6] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform resource identifier (uri): Generic syntax. <https://tools.ietf.org/html/rfc3986>, 1 2005. 2017-04-28 参照.
- [7] 谷口祥一. メタデータの「現在」: 情報組織化の新たな展開. 勉誠出版, 初版, 1 2010.
- [8] 吉賀夏子, 只木進一. 古典籍データ中の非正規記述からの周辺情報抽出とその構造化. じんもんこん 2016 論文集, 第 2016 巻, pp. 147–152, dec 2016.
- [9] Owl 2 web ontology language primer (second edition). <http://www.w3.org/TR/owl2-primer/>, 12 2012. 2017-10-10 参照.
- [10] 将来型文書統合システム標準化調査研究委員会, 小町祐史, 大野邦夫, 須栗裕樹, 山田篤. オントロジ技術入門: an introduction to ontology technology: ウェブオントロジと OWL. セマンティック技術シリーズ. 東京電機大学出版局, 2005.

- [11] 文化庁. 文化財保護法 (昭和二十五年法律第二百十四号). http://elaws.e-gov.go.jp/search/elawsSearch/elaws_search/lsg0500/detail?lawId=325AC1000000214&openerCode=1#A, 1950. 2017-10-10 参照.
- [12] 菅野育子. 欧米における図書館、文書館、博物館の連携—cultural heritage sector としての図書館—. <http://current.ndl.go.jp/ca1644>, 12 2007. 2017-10-10 参照.
- [13] 石川徹也, 根本彰, 吉見俊哉. つながる図書館・博物館・文書館 : デジタル化時代の知の基盤づくりへ. 東京大学出版会, 2011.
- [14] Wikipedia. High performance computing act of 1991 — wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=High_Performance_Computing_Act_of_1991&oldid=802420932, 2017. 2017-10-10 参照.
- [15] 影山幸一. 連載【デジタルアーカイブ羅針盤】. <http://www.infocom.co.jp/das/column/column1/column1.html>, 1 2010. 2017-10-10 参照.
- [16] 岡本真, 柳与志夫. デジタル・アーカイブとは何か : 理論と実践. 勉誠出版, 2015.
- [17] 欧州委員会. Europeana. <https://www.europeana.eu/portal/en>, 11 2008. 2018-01-04 参照.
- [18] Inc. Digital Public Library of America. Digital public library of america. <https://dp.la>. 2017-10-10 参照.
- [19] 国文学研究資料館. 日本古典籍総合目録データベース. <http://base1.nijl.ac.jp/~tkoten>, 2006. 2017-11-10 参照.
- [20] 国立国会図書館. 国立国会図書館デジタルコレクションについて. <http://dl.ndl.go.jp/ja/intro.html>. 2017-10-10 参照.
- [21] 内閣府政策統括官. 日本経済2016－2017 第2章 新たな産業変化への対応第1節 第4次産業革命のインパクト. http://www5.cao.go.jp/keizai3/2016/0117nk/n16_2_1.html, 1 2017. 2017-10-16 参照.
- [22] 総務省. 知のデジタルアーカイブ—社会の知識インフラの拡充に向けて—. http://www.soumu.go.jp/main_content/000156248.pdf, 3 2012. 2017-10-16 参照.

- [23] 文化庁. 文化関係資料のアーカイブに関する有識者会議中間とりまとめ. http://www.bunka.go.jp/seisaku/bunkashingikai/kondankaito/bunka_archive/pdf/torimatome.pdf, 8 2014. 2017-10-16 参照.
- [24] 総務省. デジタルアーカイブの構築・連携のためのガイドライン. http://www.soumu.go.jp/main_content/000153595.pdf, 3 2012. 2017-10-16 参照.
- [25] 石川徹也. 学術活動支援のための知の構造化（「つながる図書館・博物館・文書館：デジタル化時代の知の基盤づくりへ」より抜粋）, 第6章. 東京大学出版会, 2011.
- [26] 中尾央, 三中信宏. 文化系統学への招待: 文化の進化パターンを探る. 勁草書房, 2012.
- [27] G. Kubler. *The Shape of Time: Remarks on the History of Things*. Yale paperbound. Yale University Press, 1962.
- [28] 横山伊徳, 石川徹也. 歴史知識学ことはじめ. 勉誠出版, 2009.
- [29] 人間文化研究機構国立歴史民俗博物館. 総合資料学の創成に向けて. <http://www.metaresource.jp>, 2016. 2017-10-16 参照.
- [30] M. Itoh and M. Akaishi. Visualization for changes in relationships between historical figures in chronicles. In *2012 16th International Conference on Information Visualisation*, pp. 283–290, July 2012.
- [31] 赤石美奈, 伊藤直之, 箱石大, 石川徹也. 編年型データ解析ツールの開発. 第24回人工知能学会全国大会論文集, 1B1-2, 2010.
- [32] 山田太造, 遠藤珠紀, 荒木裕行, 井上聡, 久留島典子. 前近代日本史史料から人名を集める. *じんもんこん 2016 論文集*, pp. 159–164, 12 2016.
- [33] 吉賀夏子, 渡辺健次, 只木進一. 書誌学的情報およびデータ入力を考慮した貴重書書誌オントロジーの構築. 第28回人工知能学会全国大会 (JSAI2014), No. 1G5-OS-19b-1, 5 2014. 2017-12-01 参照.
- [34] Tim Berners-Lee. Semantic web road map. <https://www.w3.org/DesignIssues/Semantic.html>, 10 1998.
- [35] 佐々木利和, 湯山賢一. 博物館資料論. 放送大学教材, No. 1554590-1-1211. 放送大学教育振興会, 改訂新版, 2012.

- [36] 吉賀夏子, 渡辺健次, 只木進一. 貴重書書誌の注記から抽出したメタデータによるオントロジー構築および書誌・美術関連 linked data と連携した検索システム構築. デジタル図書館, No. 45, pp. 3–9, 2013.
- [37] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 2001.
- [38] Tim Berners-Lee. Linked data. <https://www.w3.org/DesignIssues/LinkedData.html>, 7 2006.
- [39] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, Vol. 5, No. 2, pp. 199–220, 4 1993.
- [40] 兼岩憲, 人工知能学会. 記述論理と Web オントロジー言語. 知の科学. オーム社, 2009.
- [41] The World Wide Web Consortium (W3C). The world wide web consortium (w3c). <http://www.w3.org>. 2017-10-10 参照.
- [42] The international federation of library associations and institutions (ifla). <https://www.ifla.org>.
- [43] 和中幹雄, 古川肇, 永田治樹 (訳). 書誌レコードの機能要件. 日本図書館協会, 2004.
- [44] Rick Bennett, Brian F. Lavoie, and Edward T. O'Neill. The concept of a work in worldcat: an application of frbr. *Library Collections, Acquisitions, & Technical Services*, Vol. 27, pp. 45–59, 2003.
- [45] 国立国会図書館. Web ndl authorities. <http://id.ndl.go.jp/auth/ndla>, 2011. 2017-10-18 参照.
- [46] OCLC. バーチャル国際典拠ファイル. <http://viaf.org>, 2010. 2017-04-07 参照.
- [47] Antoine Isaac. Edm primer. http://pro.europeana.eu/files/Europeana_Professional/Share_your_data/Technical_requirements/EDM_Documentation/EDM_Primer_130714.pdf, July 2013. 2017-04-10 参照.
- [48] International Council of Museums ICOM. Lido - lightweight information describing objects. <http://network.icom.museum/cidoc/working-groups/lido/>, 2010. 2017-10-17 参照.

- [49] The Library of Congress. Encoded archival description official site. <http://www.loc.gov/ead/>, 10 2017. 2017-10-17 参照.
- [50] The Library of Congress. Metadata encoding & transmission standard. <http://www.loc.gov/standards/mets/>. 2017-10-17 参照.
- [51] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, et al. Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 2014.
- [52] The Getty Research Institute. The getty vocabularies. <http://vocab.getty.edu>. 2017-04-07 参照.
- [53] 山縣友紀, 古崎晃司, 今井健, 大江和彦, 溝口理一郎. 疾患知識統合に向けた異常状態オントロジーの linked data 化. 人工知能学会論文誌, Vol. 31, No. 1, pp. LOD-A_1-15, 2016.
- [54] 加藤文彦. Linked data 作成支援ツールの現状と課題. 第 24 回セマンティックウェブとオントロジー研究会, No. 03, pp. 1-4, 2011.
- [55] Timothy J Berners-Lee. Information management: A proposal. <https://www.w3.org/History/1989/proposal.html>, 3 1989. 2018-01-04 参照.
- [56] D. Allemang and J. Hendler. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Elsevier Science, 2011.
- [57] Grigoris Antoniou and Frank Van Harmelen. *A semantic web primer*. MIT press, 2004.
- [58] Marja-Riitta Koivunen. W3c semantic web activity by marja-riitta koivunen. <https://www.w3.org/Talks/2001/1102-semweb-fin/>, 2001.
- [59] F. Lehmann. *Semantic Networks in Artificial Intelligence*. Elsevier Science Inc., New York, NY, USA, 1992.
- [60] M. Duerst and M. Suignard. Internationalized resource identifiers (iris). <https://tools.ietf.org/html/rfc3987>, 1 2005. 2017-10-18 参照.
- [61] W3C. Cool uris don't change. <https://www.w3.org/Provider/Style/URI.html>. 2017-10-18 参照.

- [62] W3C Interest Group. Cool uris for the semantic web. <https://www.w3.org/TR/cooluris/>, 12 2008. 2017-10-18 参照.
- [63] Fielding and et al. Rfc 2616: Hypertext transfer protocol - http/1.1. <http://www.ietf.org/rfc/rfc2616.txt>, 6 1999. 2017-10-18 参照.
- [64] M. Krochmal S. Cheshire. Special-use domain names. <https://tools.ietf.org/html/rfc6761>, 2 2013. 2017-10-18 参照.
- [65] 武田英明 (監訳) . *Linked Data*. 近代科学社, 2 2013.
- [66] Guus Schreiber, VU University Amsterdam, Yves Raimond, and BBC. Rdf 1.1 primer. <http://www.w3.org/TR/rdf11-primer/>, 2 2014. 2017-12-07 参照.
- [67] 吉賀夏子. 文化財の分析を支援する書誌オントロジーの構築—実在する貴重書メタデータへの適用とその評価—. 修士論文, 佐賀大学工学研究科, 2 2015.
- [68] David Beckett, W3C Tim Berners-Lee, W3C Eric Prud'hommeaux, and Lex Machina Inc Gavin Carothers. Rdf 1.1 turtle. <http://www.w3.org/TR/turtle/>, 2 2014. 2017-12-20 参照.
- [69] Rdf schema 1.1. <http://www.w3.org/TR/rdf-schema/>, 2 2014. 2017-10-10 参照.
- [70] DCMI. Dublin core® metadata initiative (dcmi). <http://dublincore.org>, 1995. 2017-12-20 参照.
- [71] Richard Cyganiak, David Wood, and Markus Lanthaler. Rdf 1.1 concepts and abstract syntax. <http://www.w3.org/TR/rdf-concepts>, 2 2014. 2017-10-18 参照.
- [72] Anne van Kesteren, Aryeh Gregor, Ms2ger, Alex Russell, and Robin Berjon. W3c dom4. <http://www.w3.org/TR/dom/>, 2 2014. 2017-10-18 参照.
- [73] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible markup language (xml) 1.0 (fifth edition). <https://www.w3.org/TR/xml/>, 11 2008. 2017-10-18 参照.
- [74] ECMA International. Standard ecma-404 the json data interchange format. <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>, 10 2013. 2017-10-18 参照.

- [75] ECMA International. Introducing json. <http://www.json.org>. 2017-10-18 参照.
- [76] Manu Sporny and Markus Lanthaler Gregg Kellogg and. Json-ld 1.0. <https://www.w3.org/TR/json-ld/>, 1 2014. 2017-10-18 参照.
- [77] Ivan Herman, Ben Adida, Manu Sporny, and Mark Birbeck. Rdfa 1.1 primer - third edition. <https://www.w3.org/TR/rdfa-primer/>, 3 2015. 2017-10-18 参照.
- [78] Ben Adida, Mark Birbeck, Shane McCarron, Ivan Herman, and et al. Rdfa core 1.1 - third edition. <http://www.w3.org/TR/rdfa-core/>, 3 2015. 2017-10-18 参照.
- [79] Microformats. <http://microformats.org/>. 2017-10-18 参照.
- [80] Ian Hickson. Html microdata. <http://www.w3.org/TR/microdata/>, 10 2013. 2017-10-18 参照.
- [81] Charles McCathie Nevile. World wide web consortium process document. <https://www.w3.org/Consortium/Process/>, 3 2017. 2017-10-18 参照.
- [82] Manu Sporny. Rdfa lite 1.1 - second edition. <http://www.w3.org/TR/rdfa-lite/>, 3 2015. 2017-10-18 参照.
- [83] W3C. Owl web ontology language guide. <https://www.w3.org/TR/owl-guide/>, 2 2004. 2017-10-18 参照.
- [84] W3C Working Group. Skos simple knowledge organization system primer. <http://www.w3.org/TR/skos-primer/>, August 2009. 2017-10-18 参照.
- [85] Dan Brickley. The foaf project. <http://www.foaf-project.org>, 2000. 2017-10-18 参照.
- [86] Noah Mendelsohn. The self-describing web. <http://www.w3.org/2001/tag/doc/selfDescribingDocuments>, 2 2009. 2017-10-18 参照.
- [87] Thomas R. Gruber. *Ontology (Entry in the Encyclopedia of Database Systems)*. Springer-Verlag, 9 2008. 2018-01-04 参照.
- [88] 国立国会図書館. 国立国会図書館ダブリンコアメタデータ記述 (dc-ndl2011年12月版) 第一部 ndl metadata terms. <http://dl.ndl.go.jp/view/download/>

- digidepo_8295098_po_ndl-term201112.pdf?contentNo=2, 12 2011. 2017-11-19 参照.
- [89] 国立公文書館. デジタルアーカイブ・システム標準仕様書. http://www.archives.go.jp/law/pdf/da_100118.pdf, 3 2009. 2017-10-18 参照.
- [90] 研谷紀夫, 北岡タマ子, 高橋英一, 三橋徹. 文化資源のデジタル化に関するハンドブック 詳細版. 東京大学大学院情報学環・凸版印刷株式会社共同研究プロジェクト, 第2版, 11 2011.
- [91] Keith Alexander, Richard Cyganiak, Michael Hausenblas, and Jun Zhao. Describing linked datasets with the void vocabulary. <https://www.w3.org/TR/void/>, 3 2011. 2017-10-18 参照.
- [92] Robert Isele, Anja Jentzsch, Christian Bizer, Julius Volz, and Petar Petrovski. Silk the linked data integration framework. <http://silkframework.org>. 2017-12-20 参照.
- [93] Elaine Rich and Kevin Knight. *Artificial Intelligence*. Artificial Intelligence Series. McGraw-Hill, 1991.
- [94] Wikipedia. Commonsense reasoning. https://en.wikipedia.org/wiki/Commonsense_reasoning, 1 2018.
- [95] Marvin Minsky. A framework for representing knowledge. <https://dspace.mit.edu/bitstream/handle/1721.1/6089/AIM-306.pdf?sequence=2>, 1974.
- [96] Ronald J Brachman and James G Schmolze. An overview of the kl-one knowledge representation system. *Cognitive science*, Vol. 9, No. 2, pp. 171–216, 1985.
- [97] Ronald J Brachman and Hector J Levesque. The tractability of subsumption in frame-based description languages. In *AAAI*, Vol. 84, pp. 34–37, 1984.
- [98] UNDL Foundation. Universal networking language. <http://www.undl.org>, 2010. 2018-01-04 参照.
- [99] Alan G Hamilton. *Logic for mathematicians*. Cambridge University Press, 1988.

- [100] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, Vol. 43, No. 5-6, pp. 907–928, 1995.
- [101] 溝口理一郎. オントロジー研究の基礎と応用. 人工知能学会誌, Vol. 14, No. 6, pp. 977–988, nov 1999.
- [102] Jeff Heflin. Owl web ontology language use cases and requirements. <https://www.w3.org/TR/webont-req/>, 2 2004. 2017-10-18 参照.
- [103] W3C. OWL 2 Web Ontology Language Document overview (second edition). <http://www.w3.org/TR/owl-overview>, 2012. 2017-12-02 参照.
- [104] 神崎正英. ウェブ・オントロジー言語 owl. <http://www.kanzaki.com/docs/sw/webont-owl.html>, 2004. 2017-10-10 参照.
- [105] Eva Verona. *Statement of principles, adopted at the International Conference on Cataloguing Principles, Paris, October 1961*, Vol. xviii. IFLA Committee on Cataloguing, annotated ed. with commentary and examples edition, 1971.
- [106] 蟹瀬智弘. 所蔵目録からアクセスツールへ rda (resource description and access) が拓く新しい情報の世界. 情報管理, Vol. 56, No. 2, pp. 84–92, 2013.
- [107] 宮田洋輔, 上田修一, 谷口祥一, 横山幸雄, 鴫田拓哉, 向當麻衣子. Frbr における「著作」実体としての日本の古典著作: Frbr 研究会の取り組み ii. 2011 年日本図書館情報学会春季研究集会発表要綱, pp. 63–66, 2011.
- [108] 日本図書館協会. 日本目録規則 (ncr) 2018 年版 (仮称) . <http://www.jla.or.jp/committees/mokuroku//tabid/643/Default.aspx>. 2017-12-10 参照.
- [109] Nick Crofts, Martin Doerr, Tony Gill, Stephen Stead, and Matthew Stiff. Definition of the cidoc conceptual reference model. http://cidoc-crm.org/docs/cidoc_crm_version_6.2.pdf, March 2015. 2017-04-10 参照.
- [110] International council of museums. <http://icom.museum>. 2017-12-10 参照.
- [111] Dominic Oldman and CRM Labs. The cidoc conceptual reference model (cidoc-crm): Primer. http://www.cidoc-crm.org/docs/CRMPprimer_v1.1.pdf, July 2014. 2017-12-01 参照.
- [112] 村田良二. ミュージアム資料情報構造化モデルの開発. 人文科学とコンピュータシンポジウム, pp. 63–70, 12 2005.

- [113] 村田良二. 博物館におけるコレクション情報の組織化：情報標準と東京国立博物館の事例. *情報管理*, Vol. 59, No. 9, pp. 577–586, 2016.
- [114] Chryssoula Bekiari, Martin Doerr, Patrick Le Bœuf, and Pat Riva. Definition of frbr_{oo}: A conceptual model for bibliographic information in object-oriented formalism. http://www.ifla.org/files/assets/cataloguing/FRBRoo/frbroo_v_2.4.pdf, November 2015. 2017-04-10 参照.
- [115] 孫外英, 永森光晴, 杉本重雄. オブジェクト指向 frbr を基礎としたマンガオントロジーの設計. *デジタル図書館*, No. 38, pp. 3–13, 3 2010.
- [116] Gregor Strle and Matija Marolt. The ethnomuse digital library: conceptual representation and annotation of ethnomusicological materials. *International Journal on Digital Libraries*, Vol. 12, No. 2/3, pp. 105–119, 2012.
- [117] Cezary Mazurek, Krzysztof Sielski, Justyna Walkowska, and Marcin Werla. From marc21 and dublin core, through cidoc crm: first tenuous steps towards representing library data in frbr_{oo}. *CIDOC 2012*, 2012.
- [118] 安田つくし, 三原鉄也, 永森光晴, 杉本重雄. Frbroo に基づいた同人創作物のアーカイブ構築のためのメタデータの作成. *じんもんこん 2015 論文集*, 第 2015 巻, pp. 255–262, dec 2015.
- [119] 吉賀夏子, 渡辺健次, 只木進一. 貴重書メタデータの設計図としての書誌オントロジーを適用した linked data. 第 35 回セマンティックウェブとオントロジー研究会, No. SIG-SWO-035-08, pp. 1–9, 3 2015.
- [120] Patrick Le Bœuf. Transforming FRBR into FRBR_{oo}. <http://old.cidoc-crm.org/docs/Transforming%20FRBR%20into%20FRBRoo.pdf>, 5 2013. 2018-01-04 参照.
- [121] PRESS_{oo} Review Group IFLA. Definition of pressoo: A conceptual model for bibliographic information pertaining to serials and other continuing resources. https://www.ifla.org/files/assets/cataloguing/PRESSoo/pressoo_v1-3.pdf, 3 2017. 2018-01-04.
- [122] アメリカ議会図書館. Bibliographic framework initiative. <http://www.loc.gov/bibframe/>. 2017-05-01 参照.
- [123] Creative Commons. Creative commons. <http://creativecommons.org>, 2001. 2017-11-19 参照.

- [124] Deutsche digitale bibliothek. <https://www.deutsche-digitale-bibliothek.de>. 2017-12-10 参照.
- [125] Europeana and Digital Public Library of America. Rightsstatements.org. <http://rightsstatements.org/en/>. 2017-12-10 参照.
- [126] B. DuCharme. *Learning SPARQL*. O'Reilly Media, 2011.
- [127] The W3C SPARQL Working Group. Sparql 1.1 overview. <http://www.w3.org/TR/sparql11-overview/>, 3 2013.
- [128] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.
- [129] Eric Prud'hommeaux, Carlos Buil-Aranda, Andy Seaborne, Axel Polleres, Lee Feigenbaum, and Gregory Todd Williams. Sparql 1.1 federated query. <https://www.w3.org/TR/sparql11-federated-query/>, 3 2013. 2017-10-18 参照.
- [130] The Apache Software Foundation. Apache jena. <https://jena.apache.org>, 2011.
- [131] Eclipse RDF4J. Eclipse rdf4j (sesame). <http://rdf4j.org>.
- [132] OpenLink Software. Openlink virtuoso. <https://virtuoso.openlinksw.com>, 2015. 2017-10-18 参照.
- [133] Antonio Garrote. rdfstore-js. <https://github.com/antoniogarrote/rdfstore-js>, 2011. 2017-10-18 参照.
- [134] メタデータ基盤協議会. Metabridge. <http://metabridge.jp>, 2013. 2017-12-10 参照.
- [135] 国立国会図書館. Web ndl authorities について. <http://id.ndl.go.jp/information/about/>, 2011. 2017-10-18 参照.
- [136] 健太郎乾, 聡関根. 自然言語処理技術の現状と展望-エラー分析プロジェクトを通して- : 2. 自然言語処理技術の概要. 情報処理, Vol. 57, No. 1, pp. 6–9, dec 2015.
- [137] 信介森, 伸裕鍛冶, 有吾村脇, いつみ齊藤. 自然言語処理技術の現状と展望-エラー分析プロジェクトを通して-[基礎技術] 3.1 形態素解析. 情報処理, Vol. 57, No. 1, pp. 10–11, dec 2015.

- [138] 工藤拓. Mecab: Yet another part-of-speech and morphological analyzer (ver. 0.996). <http://taku910.github.io/mecab/>, 2013. 2017-04-07 参照.
- [139] 工藤拓, 山本薫, 松本裕治. Conditional random fields を用いた日本語形態素解析. 情報処理学会研究報告自然言語処理 (NL), Vol. 2004, No. 47 (2004-NL-161), pp. 89–96, 2004.
- [140] 京都大学大学院情報学研究科. 日本語形態素解析システム juman. <http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN>, 1 2012. 2017-10-18 参照.
- [141] 奈良先端大学. Chasen – 形態素解析器. <http://chasen-legacy.osdn.jp>, 2007. 2017-10-18 参照.
- [142] Graham Neubig. Kytea. <http://www.phontron.com/kytea/index-ja.html>, 1 2012. 2017-10-18 参照.
- [143] Graham Neubig, 中田陽介, 森信介. 点推定と能動学習を用いた自動単語分割器の分野適応. 言語処理学会第 16 回年次大会 (NLP2010), Vol. 3, , 2010.
- [144] Graham Neubig. Kytea を使った単語分割の分野適応. <http://www.phontron.com/kytea/active-ja.html>, 2 2011. 2017-10-18 参照.
- [145] Wikipedia. Precision and recall. https://en.wikipedia.org/wiki/Precision_and_recall. 2017-11-10 参照.
- [146] 吉賀夏子, 只木進一. 古典籍書誌データ構造に対応した linked data への半自動変換. 情報処理学会論文誌, Vol. 59, No. 2, pp. 257–266, 2 2018.
- [147] 井上敏幸 (編). 市場直次郎コレクション目録. 佐賀大学附属図書館・地域学歴史文化研究センター, 2007.
- [148] 佐賀大学附属図書館. 佐賀大学貴重書コレクション. <http://www.cc.saga-u.ac.jp/OgiNabesima/>, 2001. 2017-04-02 参照.
- [149] 国立国会図書館. 『国立国会図書館「日本目録規則 1987 年版改訂 3 版」和古書適用細則』(2012 年 1 月). <http://warp.da.ndl.go.jp/info:ndljp/pid/9484238/www.ndl.go.jp/jp/library/data/wakosho201201.pdf>, 1 2012. 2017-04-07 参照.
- [150] 増井ゆう子 (国文学研究資料館). 国文学研究資料館和古書目録データベースの作成. <http://www.nijl.ac.jp/pages/event/seminar/images/wakosyodb.pdf>, 1 2012. 2017-10-10 参照.

参考文献

- [151] Isamu Tsuchitani and Manae Fujishiro. Descriptive cataloging guidelines for pre-meiji japanese books. http://www.eastasianlib.org/cjm/jrb/2011_japaneseRareBooksCatalogingGuidelines.pdf, 2011. 2017-04-07 参照.
- [152] 国文学研究資料館. 利用案内 (日本古典籍総合目録データベース) . <http://base1.nijl.ac.jp/~tkoten/howto.html>, 12 2006. 2018-01-04 参照.
- [153] 国立国語研究所. Unidic (バージョン 2.1.2). http://pj.ninjal.ac.jp/corpus_center/unidic/, 2013. 2017-04-06 参照.
- [154] 国立天文台. こよみ用語解説. <http://eco.mtk.nao.ac.jp/koyomi/faq/24sekki.html>, 1994. 2017-01-01 参照.
- [155] みんなの知識委員会. みんなの知識ちょっと便利帳. <http://www.benricho.org/koyomi/tuki.html>, 2002. 2017-11-01 参照.
- [156] ウィキペディア日本語版. 干支. <https://ja.wikipedia.org/wiki/干支>. 2015年12月30日 (水) 11:04 の版, 2017-12-01 参照.
- [157] ウィキペディア日本語版. 元号一覧 (日本) . [https://ja.wikipedia.org/wiki/元号一覧_\(日本\)](https://ja.wikipedia.org/wiki/元号一覧_(日本)). 2015年12月14日 (月) 09:46 の版, 2017-12-01 参照.
- [158] 国立国会図書館. Web ndl authorities. <http://id.ndl.go.jp/auth/ndla>, 2011. 2017-12-02 参照.
- [159] 工藤拓. オリジナル辞書/コーパスからのパラメータ推定. <http://taku910.github.io/mecab/learn.html#eval>, 2013. 2017-04-07 参照.
- [160] Stanford Center for Biomedical Informatics Research. Protégé. <http://protege.stanford.edu>, 4 2013. 2017-04-07 参照.
- [161] Thomson Reuter. Open calais. <http://www.opencalais.com>. 2017-11-10 参照.
- [162] 橋口侯之介. 江戸の本屋と本づくり 続和本入門. 平凡社, 2011.
- [163] 国文学研究資料館. 日本古典籍データセット. <http://codh.rois.ac.jp/pmjt/>, 11 2016. 2017-04-07 参照.
- [164] 関野樹 (総合地球環境学研究所) . Hutime. <http://www.hutime.org>, 2012. 2017-04-24 参照.
- [165] 川村隆浩, 大須賀昭彦. Text2lod. 人工知能学会論文誌, Vol. advpub, , 2016.

- [166] 工藤拓, 松本裕治. Cabocha/南瓜: Yet another japanese dependency structure analyzer. <https://taku910.github.io/cabocha/>. 2017-11-10 参照.
- [167] Francis Bond, Takayuki Kuribayashi, Hitoshi Isahara, Kyoko Kanzaki, Kiyotaka Uchimoto, Kow Kuroda, Masao Utiyama, Darren Cook, Asuka Sumida, and Kentaro Torisawa. Japanese wordnet. <http://compling.hss.ntu.edu.sg/wnja/>. 2017-11-10 参照.
- [168] Brian Ulicny. Constructing knowledge graphs with trust. In *METHOD 2015: 4th International Workshop on Methods for Establishing Trust of (Open) Data*, October 2015.
- [169] Online Computer Library Center. Worldcat.org. <http://www.worldcat.org>. 2017-05-01 参照.
- [170] W3C Schema.org Community Group. schema.org. <http://schema.org/>. 2017-05-09 参照.
- [171] Online Computer Library Center. Data strategy. <http://www.oclc.org/en/worldcat/data-strategy.html>. 2017-12-20 参照.
- [172] 高間康史, 矢部彩佳, 石川博. サンプルングに基づく構造推定を用いた lod 視覚的分析支援システム. 人工知能学会論文誌, Vol. 32, No. 1, pp. WII-B.1-11, 2017.
- [173] F. Yergeau. Utf-8, a transformation format of iso 10646. <https://tools.ietf.org/html/rfc3629>, 11 2003. 2018-01-04 参照.

研究業績

主要論文

1. 吉賀夏子, 只木進一. 古典籍書誌データ構造に対応した Linked Data への半自動変換. 情報処理学会論文誌, Vol. 59, No. 2, pp. 257-266. 情報処理学会, 2018, 2. (査読有)

本研究関連の論文

1. 吉賀夏子, 渡辺健次, 只木進一. 古典籍書誌注記文からの作品構造および関連人物のつながりを明らかにする周辺情報抽出. 情報処理学会人文科学とコンピュータ研究会 (CH) , 2016-CH-109, No. 3, Vol. 109. 情報処理学会, 2016, 1.
2. 吉賀夏子, 只木進一. 古典籍データ中の非正規記述からの周辺情報抽出とその構造化. じんもんこん 2016 論文集, pp. 147-152. 情報処理学会, 2016. 12. (査読有)
3. Natsuko Yoshiga and Shin-Ichi Tadaki. Semi-automated Generation of Linked Data from Unstructured Bibliographic Data for Japanese Historical Rare Books. 14th International Conference on Digital Preservation (iPRES2017), <https://ipres2017.jp/wp-content/uploads/13.pdf>, Kyoto, Japan. 2017. 9. (査読有)

その他

1. Natsuko Yoshiga, Kenzi Watanabe, Shin-ichi Tadaki. A Visual Search System with Semantic Web Technologies on Digital Archives for Historical Documents. International Conference on Convergence Content (ICCC 2012), pp. 23-24, 2012. 12.

2. 吉賀夏子, 渡辺健次, 只木進一. 貴重書デジタルアーカイブの書誌オントロジーおよび Semantic Web 技術を活用した検索システムの構築. 第 27 回人工知能学会全国大会 (JSAI2013), 1N3-OS-10a-4in, 富山県富山市, 2013. 6.
(備考) 2013-06-06 インタラクティブセッションで同大会学生奨励賞を受賞.
3. 吉賀夏子, 渡辺健次, 只木進一. 貴重書書誌の注記から抽出したメタデータによるオントロジー構築および書誌・美術関連 Linked Data と連携した検索システム構築. デジタル図書館, No. 45, pp. 3-9, 2013.
4. 吉賀夏子, 渡辺健次, 只木進一. 書誌学的情報およびデータ入力を考慮した貴重書書誌オントロジーの構築. 第 28 回人工知能学会全国大会 (JSAI2014), 1G5-OS-19b-1, 愛媛県松山市, 2014. 5.
5. 吉賀夏子. 文化財の分析を支援する書誌オントロジーの構築 —実在する貴重書メタデータへの適用とその評価—. 佐賀大学工学研究科修士論文. 2015. 2.
6. 吉賀夏子, 渡辺健次, 只木進一. 貴重書中の部品を記述できるオントロジーに基づき Linked Data 化したメタデータを用いた人名ネットワーク構築の試み. 第 29 回人工知能学会全国大会 (JSAI2015), 1G4-1in, 北海道函館市, 2015. 5.
7. 吉賀夏子, 渡辺健次, 只木進一. 貴重書メタデータの設計図としての書誌オントロジーを適用した Linked Data. 第 35 回セマンティックウェブとオントロジー研究会, SIG-SWO-035-08, pp. 1-9. 人工知能学会, 2015. 3.

謝辞

本研究を遂行するにあたり、佐賀大学大学院工学研究科教授只木進一先生は、貴重な時間を割いて、細やかな研究指導を行ってくださいました。広島大学大学院教育学研究科教授渡辺健次先生には、およそ一年間お忙しい中ご指導いただきました。また、佐賀大学総合情報基盤センター准教授大谷誠先生には、ubiquitousな研究環境を整備していただきました。佐賀大学地域学・歴史文化研究センター准教授伊藤昭弘先生には、共同研究の機会を与えていただきました。研究対象の市場直次郎コレクションの取扱いに際しては、佐賀大学附属図書館の方々に、権利上の手続き、資料貸し出しなどの便宜を図っていただきました。

そして、佐賀大学総合情報基盤センターの業務と研究の両立には、同センターおよび総務部情報企画管理課全員の方々のご理解とご協力が不可欠でした。私の父である帆足章二氏は、美術品商としての視点から見た文化財についてのお話、学会へ行く際の子供の一時預かりなど様々な面で支援していただきました。娘の吉賀まりさん、息子の吉賀太郎くんには、大きな病気や怪我もなく、元気に過ごしてもらうことで、研究生活を支援していただきました。

この場で皆様方に感謝いたします。

付録 A

RDF ストアの利用

A.1 MetaBridge を用いた URI 参照解決

本研究では *Ichiba* を Linked Data として Web 上に公開するため、管理の容易な大学外組織の共用 SPARQL サーバを利用する。そこで、RDF ストア (2.4.4 節) の一つである MetaBridge を用いた URI 参照解決の手順を紹介する。

本研究では、*Ichiba* を管理するサーバ (<http://www.dl.saga-u.ac.jp/>) が URI リクエストを受けると、以下に示す二種類のレスポンスを行う。

1. URI にハッシュタグがある場合、*Ichiba* 語彙定義ファイルである、<http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/index.rdf> をクライアントに返す。(例えば、<http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#hasRole> 等。)
2. URI にハッシュタグがない場合、HTTP ステータスコード 303 を返すとともに、MetaBridge の RDF ストアにリクエストした URI の情報を要求するプログラムを実行するファイル (ソースコード A.2) にリダイレクトする (ソースコード A.1)。この実行ファイルは、要求された URI から RDF ストアへのクエリ結果を返す。

ソースコード A.1: Apache2.2 系 Web サーバ用リダイレクト設定。

```
1 //Web サーバ上のプログラムにリダイレクトする
2 RewriteEngine on
```

```
3 RewriteRule ^tenseki/([0-9]+)(.*) http://www.dl.saga-u.ac.jp/
  OgiNabesima/ichiba/tenseki/data/index.php?q=$1$2 [R=303,L]
```

ソースコード A.2: 佐賀大学電子図書館貴重書コレクションウェブサイト [148] から MetaBridge の *Ichiba* 用 SPARQL エンドポイントへのクエリ発行用 PHP スクリプト.

```
1 <?php
2 $url = $_SERVER['REQUEST_URI'];
3 #print $_SERVER['REQUEST_URI'];
4 $q = htmlspecialchars(strip_tags(trim($_GET['q'])));
5 $sparql_query = "
6 PREFIX skosxl: <http://www.w3.org/2008/05/skos-xl#>
7 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
8 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
9 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
10 PREFIX owl: <http://www.w3.org/2002/07/owl#>
11 PREFIX dcterms: <http://purl.org/dc/terms/>
12 PREFIX dc: <http://purl.org/dc/elements/1.1/>
13 PREFIX ichiba_tenseki: <http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/
  tenseki#>
14 PREFIX frbroo: <http://www.cidoc-crm.org/rdfs/FRBR2.0-draft-B.rdfs#>
15 PREFIX dcndl: <http://ndl.go.jp/dcndl/terms/>
16 PREFIX ndltype: <http://ndl.go.jp/ndltype/>
17 ";
18 $sparql_query .= "select * where {<http://www.dl.saga-u.ac.jp/
  OgiNabesima/ichiba/tenseki/\".$q.\"> ?p ?o}";
19 #print $sparql_query;
20 $metabridge = "https://www.metabridge.jp/api/sparql/G0000138mb?lang=
  default&lu=cwuboswnkghofpjoizlg&format=html&query=".urlencode(
  $sparql_query);
21 #print $metabridge;
22 header("Location: ".$metabridge."");
23 ?>
```

上記スクリプトでは、クライアントがリクエストした URI を SPARQL クエリに変換し、MetaBridge の RDF ストアに問い合わせしている。

このクエリでは、例えば、<http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100> の部品情報を取得できる (図 A.1, 図 A.2, 図 A.3)。

付録 A . RDF ストアの利用

図 A.1: ブラウザでのメタデータ情報取得画面. <http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100> (古典籍) での結果.

p	o
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#Tenseki
http://purl.org/dc/terms/extent	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-11
http://purl.org/dc/terms/creator	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-13
http://purl.org/dc/elements/1.1/creator	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-13-0-0-0
http://purl.org/dc/terms/identifier	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-0
http://purl.org/dc/terms/title	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-9
http://purl.org/dc/terms/description	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-17
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#bunrui1	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-2
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#bunrui2	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-3
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#bunrui3	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-4
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#chuki	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-17
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#gazoukukai	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-8
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#hasMainTitle	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-10
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#hasMainTitle	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-9
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#hasMainURI	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#henchosha	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-13
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#kankou_nen	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-14
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#kansatsu	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-12
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#kansha	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-16
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#seiribangou	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-0
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#shokei	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-11
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#shomei	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-9
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#useTechnicalTerms	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-17-0-1-1
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#useTechnicalTerms	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-13-0-1-1
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#yomi	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-10
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#hasMainGenre	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-2
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#hasMainGenre	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-3
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#hasMainGenre	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-4
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#isCollectionOf	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba
http://purl.org/dc/terms/date	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-14
http://ndl.go.jp/dcndl/terms/Transcription	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-10

図 A.2: ブラウザでのメタデータ情報取得画面. <http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-13-0-0-0> (著者) での結果.

p	o
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#Person
http://www.w3.org/2000/01/rdf-schema#label	"藤田彪"@ja
http://www.w3.org/2000/01/rdf-schema#label	"藤田東湖"@ja
http://www.w3.org/2002/07/owl#sameAs	http://id.ndl.go.jp/auth/ndlna/00269832
http://www.w3.org/2002/07/owl#sameAs	http://viaf.org/viaf/27862907
http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#hasRole	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-13-0-1-1

図 A.3: ブラウザでのメタデータ情報取得画面. <http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/100-9> (外題) での結果.

p	o
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#MainTitle
http://www.w3.org/2000/01/rdf-schema#label	"弘道館記述義"@ja

付録B

ソースコード一覧

本研究で作成したソースコードを掲載する。動作環境は、CentOS release 6.9 (Final), PostgreSQL 9.4, PHP 5.5.32 である。

本論文付録の *Ichiba* 解析ソースコードは、付属の CD-ROM 中の `/source/ichiba/` 下に配置する。 *Kokubun* 解析ソースコードは `/source/kokubun/` 下に配置する。ただし、ソースコード内では、ディレクトリ構成の記述を変更していない。

(注意) ソースコードの正規表現で用いている語に、文字化けが生じていることがある。対応する電子ファイルでは問題なく表示できる。全ファイルの文字コードは UTF-8[173], 改行コードは LF (Line Feed) 形式である。

また、データベース sites には、以下のテーブル (表 B.1) が格納されている。データベース sites のバックアップは、 `/source/20171218sites.backup` にテキスト形式で配置する。

表 B.1: *Ichiba* および *Kokubun* 解析用データベース sites に格納されたテーブル。

テーブル名	コメント	関連スクリプト
data-all	Mecab 解析済 <i>Ichiba</i> 全単語データ	スクリプト C, D
middledata	<i>Ichiba</i> 固有表現由来情報	スクリプト C, D
correctiondata	middledata の修正データ (手作業)	
headings	<i>Ichiba</i> 典拠データベース 1	スクリプト F
ichibaviaf	<i>Ichiba</i> の典拠データベース 2	スクリプト F
kokubunall	Mecab 解析済 <i>Kokubun</i> 全単語データ	スクリプト C, D
kokubunindex	<i>Kokubun</i> 固有表現由来情報	スクリプト C, D
kokubuncorrectiondata	<i>Kokubun</i> 修正データ (手作業)	
headingskokubun	<i>Kokubun</i> 典拠データベース 1	スクリプト F
kokubunviaf	<i>kokubun</i> の典拠データベース 2	スクリプト F

B.1 単語リストの例

単語リスト全文は/source/ichiba/words 下に配置する.

ソースコード B.1: 固有表現クラス「時」の単語リストの例 (dateterms.csv).

```
1 //左から,固有表現,読みを表す.ただし,読みは参考のための付加データで,固有表現抽出ス  
   クリプトには使用しない.  
2  
3 睦月,むつき  
4 夷鐘,いしょう  
5 寅月,いんげつ  
6 祝月,いわいづき  
7 王春,おうしゅん  
8 開春,かいしゅん  
9 ...
```

ソースコード B.2: 固有表現クラス「部品」の単語リストの例 (tob.csv).

```
1 //左から,固有表現,オントロジー表現,英語の固有表現,参考情報を表す.  
2  
3 題,#Title  
4 内題,#TitleOfWork,  
5 外題,#MainTitle,  
6 解題,,introductions  
7 開題,=解題,  
8 巻頭,,opening  
9 巻首題,=(巻頭:題)|C内題,caption titles  
10 目首,,table of contents  
11 目録,,catalog  
12 目次題,=(目首:題)|C内題,title from table of contents  
13 目録題,=(目録:題)|C内題,title from catalog  
14 巻末,,End of text  
15 尾,=巻末,  
16 帙,,a wrap around case,A wrap-around case usually made of blue cloth  
   over boards or thick paper to contain paper-bound volumes (satsu  
   冊). Closed by means of bone pegs.  
17  
18 ...
```

(英文は [151] から引用.)

ソースコード B.3: 固有表現クラス「ロール」の単語リストの例 (role.csv).

```
1 //左から,固有表現,オントロジー表現,英語の固有表現,参考情報を表す.  
2  
3 作,,creator  
4 著,,author,本文作者  
5 共著,⌋著,co-author,複数の本文作者  
6 編,,editor  
7 編者,=編  
8 選  
9 ...
```

ソースコード B.4: 固有表現クラス「専門用語」の単語リストの例 (terms.csv).

```
1 //左から,固有表現,オントロジー表現,英語の固有表現,参考情報を表す.  
2  
3 版,,edition  
4 新版,⌋版  
5 再版,⌋版,reprints  
6 再板,=再版  
7 再版本,=再版|⌋本  
8 両面刷  
9 謄写版,⌋版  
10 ...
```

ソースコード B.5: NDLA の SPARQL エンドポイントから人名典拠データを抽出するスクリプト (ndlasrap.php).

```
1 <?php  
2 //NDLA に登録されている人名を抽出  
3 //サーバに負荷を掛けないように途中でスリープする  
4 //接続失敗に備えて抽出回数をカウントする  
5  
6 // URL to a SPARQL endpoint.  
7 $url='http://id.ndl.go.jp/auth/ndla';  
8 $offsetnum = 0;  
9 $allquerycount = 706808;  
10 (int)$count = $allquerycount / 100;  
11  
12 for ($i = $count; $i > 0; $i--) {  
13  
14 if ($i % 100 == 0) {
```

```
15 | print "-----sleep time-----\n";
16 | sleep(10);
17 | }
18 |
19 | // The SPARQL query to run.
20 | $sparql_pre="
21 | PREFIX dc:<http://purl.org/dc/elements/1.1/>
22 | PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
23 | PREFIX foaf:<http://xmlns.com/foaf/0.1/>
24 | PREFIX dcterms:<http://purl.org/dc/terms/>
25 | PREFIX owl:<http://www.w3.org/2002/07/owl#>
26 | PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
27 | PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
28 | PREFIX skos:<http://www.w3.org/2004/02/skos/core#>
29 | PREFIX rda:<http://RDVocab.info/ElementsGr2/>
30 | PREFIX xl:<http://www.w3.org/2008/05/skos-xl#>
31 | PREFIX dcndl:<http://ndl.go.jp/dcndl/terms/>
32 | ";
33 |
34 | $sparql_cmd = "
35 | SELECT DISTINCT ?uri1 ?fname ?yomi
36 | ";
37 |
38 | $sparql_cmd2 = "
39 | WHERE {
40 | ?uri1 foaf:primaryTopic ?uri2 .
41 | ?uri2 rdf:type foaf:Person .
42 | ?uri2 foaf:name ?fname .
43 | optional {
44 | ?uri1 xl:prefLabel [ xl:literalForm ?heading ].
45 | ?uri1 xl:prefLabel [ dcndl:transcription ?yomi ].
46 | }
47 | filter (lang(?yomi) = 'ja-Kana').
48 | }
49 | ";
50 |
51 | $offsetnum = 100 * ($count - $i);
52 | $sparql_offs = 'offset '.$offsetnum;
53 |
54 | $sparql = $sparql_pre.$sparql_cmd.$sparql_cmd2.$sparql_offs;
55 | $sparql = str_replace(array("\r\n", "\r", "\n"), '', $sparql);
56 | $param = $url.'?query=' . urlencode($sparql).'&output=xml';
57 | $datafile = '/home/yoshiga/analysis/ndldata.xml';
58 | #print $param;
59 |
60 | if (!( $fp = fopen($param, 'rb') )) {
61 |     trigger_error("リンクを読み込めません。");
```



```
62  exit;
63 } else {
64  $data = stream_get_contents($fp);
65  fclose($fp);
66
67  if (mb_strlen($data) < 1) {
68    trigger_error("データがありません。");
69    exit;
70  }
71
72  if (!(($fpw = fopen($datafile, 'a')))) {
73    trigger_error("データ書き込みファイルを開けません。");
74    exit;
75  } else {
76    $data = strip_tags($data, '<result><binding><uri><literal>');
77    fwrite($fpw, $data);
78    fclose($fpw);
79
80    print "count---- ".$i." ".$sparql_offs."\n";
81    sleep(2);
82  }
83 }
84 }
85 ?>
```

B.2 Mecab ユーザ辞書生成

ソースコード B.6: *Ichiba* の書名, 編著者名をキーワードとして NDL 検索から関連単語の自動収集例 (expertXmltoMecab.php).

```
1 <?php
2 //expertXmltoMecab.php
3 //Ichiba タイトルが NDL にないか探索し,
4 //あれば MeCab 辞書用 CSV にフォーマットする.
5 //関連する人名, 出版者(書誌), タイトル, 出版日なども抽出
6
7 // ~~~ef^bd^a1-^ef^be^9f <== preg 系正規表現中の左記文字列は,
8 //tex では文字化けするので, ソースコードを直接参照してください.
9
10 $record = file("/home/yoshiga/analysis/ichibaoriginalwuri.csv",
    FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
```

```
11 $role = file("/home/yoshiga/analysis/role.csv",
12             FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
13 foreach ($role as $val) {
14     $pattern[] = "/"(.trim($val)."$)/u";
15 }
16 #print_r($pattern);
17
18 $i = 0;
19 while ($i < count($record)) {
20     $persons = array();
21     $titles = array();
22     $place = array();
23     $date = array();
24     $subjects = array();
25     $materials = array();
26
27     $xml = new XMLReader();
28     $xml->open("/home/yoshiga/analysis/ichiba_tenseki_rdf/
29             ichiba_tenseki_rec".str_pad($i, 3, "0", STR_PAD_LEFT).".rdf");
30
31     while ($xml->read()) {
32         switch ($xml->nodeType) {
33             case (XMLREADER::ELEMENT):
34                 if ($xml->name == "numberOfRecords") {
35                     if ($xml->readString() == "0") {
36                         break(2);
37                     }
38                 }
39                 if ($xml->name == "dcndl_simple:dc") {
40                     $node = $xml->expand();
41                     $dom = new DomDocument();
42                     $dom->preserveWhiteSpace = false;
43                     $n = $dom->importNode($node, true);
44                     $sdom = simplexml_import_dom($n);
45
46                     $sdom->registerXPathNamespace("dcndl_simple", "http://ndl.
47                             go.jp/dcndl/dcndl_simple/");
48                     $sdom->registerXPathNamespace("dc", "http://purl.org/dc/
49                             elements/1.1/");
50                     $sdom->registerXPathNamespace("dcterms", "http://purl.org/
51                             dc/terms/");
52                     $sdom->registerXPathNamespace("dcndl", "http://ndl.go.jp/
53                             dcndl/terms/");
54                     $sdom->registerXPathNamespace("foaf", "http://xmlns.com/
55                             foaf/0.1/");
```

```
51     $identifier = $sdom->xpath("dc:identifier");
52     $title = $sdom->xpath("dc:title");
53     $partTitle = $sdom->xpath("dcndl:partTitle");
54     $person = $sdom->xpath("dc:creator");
55     $publisher = $sdom->xpath("dc:publisher");
56     $publicationPlace = $sdom->xpath("dcndl:publicationPlace");
57     $issuedDate = $sdom->xpath("dcterms:issued");
58     $materialType = $sdom->xpath("dcndl:materialType");
59     $subject = $sdom->xpath("dc:subject");
60
61     #print $title[0]."\n\n";
62     #var_dump($publisher);
63
64     //$title
65     foreach ($title[0] as $t) {
66         $t2 = preg_replace($pattern, '', $t);
67         if ($t2 != $t) {
68             $persons[] = $t2;
69         } else {
70             $titles[] = $t;
71         }
72     }
73
74     //$partTitle
75     foreach ($partTitle as $pT) {
76         if (preg_match_all('/\[([^\s\., \,]+)\]/u', $pT, $mpT)) {
77             foreach ($mpT[1] as $mpTval) {
78                 if (preg_match_all('/\[([^\(\() \)]+\) \]/u',
79                     $mpTval, $mpTperson) &&
80                     preg_match('/^[^ ~^^ef^^bd^^a1-^^ef^^be^^9
81                         f 0 1 2 3 4 5 6 7 8 9](.+)/u', $mpTperson[1][0]) &&
82                     $mpTperson[1][0] != "")
83                 ) {
84                     $persons[] = preg_replace($pattern, '', $mpTperson
85                         [1][0]);
86
87                     $mpT2 = preg_replace('/\[([^\(\() \)]+\) \]/u',
88                         '', $mpTval);
89                     #print "test=".$mpT2."\n";
90                     if ($mpT2 != "" && preg_match('/^[^ ~^^ef^^bd^^a1-^^
91                         ef^^be^^9f 0 1 2 3 4 5 6 7 8 9](.+)/u', $mpT2)) {
92                         if (preg_match('/\[([^\(\() \)]+\) \]/u', $mpT2, $m1) ||
93                             preg_match('/\[([^\(\() \)]+\) \]/u', $mpT2, $m2)) {
94                             if (isset($m1[1])) {
95                                 $persons[] = preg_replace($pattern, '', $m1[1]);
96                             }
97                         }
98                     }
99                 }
100             }
101         }
102     }
103 }
```

```
92         if (isset($m2[1])) {
93             $persons[] = preg_replace($pattern, '', $m2[1]);
94         }
95     } else {
96         $mpT22 = preg_replace($pattern, '', $mpT2);
97         if ($mpT22 != $mpT2) {
98             $persons[] = $mpT22;
99         } else {
100             $titles[] = $mpT2;
101         }
102     }
103 }
104 }
105 }
106 }
107
108 // $person
109 if (isset($person[0])) {
110     foreach ($person[0] as $p) {
111         $p2 = preg_replace('/[ \/\](.+)$/u', '', $p);
112         $persons[] = preg_replace($pattern, '', $p2);
113     }
114 }
115
116 // $publisher
117 if (isset($publisher[0][0])) {
118     $pub = (string)$publisher[0][0];
119     $j = 0;
120     while ($j < count($publisher)) {
121         $persons[] = $pub;
122         $j++;
123     }
124 }
125
126 // $publicationPlace
127 if (isset($publicationPlace[0][0])) {
128     $pplace = (string)$publicationPlace[0][0];
129     $j = 0;
130     while ($j < count($publicationPlace)) {
131         $place[] = $pplace;
132         $j++;
133     }
134 }
135
136 // date
137 if (isset($issuedDate[0][0])) {
138     $idate = (string)$issuedDate[0][0];
```

```
139         $j = 0;
140         while ($j < count($issuedDate)) {
141             $date[] = $idate;
142             $j++;
143         }
144     }
145
146     //subject
147     if (isset($subject[0][0])) {
148         $subj = (string)$subject[0][0];
149         $j = 0;
150         while ($j < count($subject)) {
151             $subjects[] = mb_convert_kana($subj, "a");
152             $j++;
153         }
154     }
155
156     //materialType
157     if (isset($materialType[0][0])) {
158         $mtty = (string)$materialType[0][0];
159         $j = 0;
160         while ($j < count($materialType)) {
161             $materials[] = $mtty;
162             $j++;
163         }
164     }
165
166     }
167 }
168 }
169
170 $xml->close();
171
172 # $ref = "ICHIBA_EXPERT,ichiba_tenseki_rec".str_pad($i, 3, "0",
173         STR_PAD_LEFT).".rdf";
174 $ref = "ICHIBA_EXPERT";
175
176 //Type の確認とフィルター
177 sort($titles);
178 $titles2 = array_unique($titles, SORT_STRING);
179
180 sort($persons);
181 $persons2 = array_unique($persons, SORT_STRING);
182
183 foreach ($persons2 as $pval) {
184     $pval2 = preg_replace('/[ \| - \, \(\) ]/u', '', $pval);
```

付録 B . ソースコード一覧

```

184     if (trim($pval2) != "" && isset($pval2) && preg_match('/^[^ -~~~
        ef^^bd^^a1-^^ef^^be^^9f\,0 1 2 3 4 5 6 7 8 9「『\s](.+)/u',
        $pval2) && preg_match('/^(〔第他〕|(^ほか))([~0-90 1 2 3 4 5 6
        7 8 9一二三四五六七八九十]+)/u', $pval2)) {
185     if (strpos($pval2, "/") !== false) {
186         $pval3 = explode("/", $pval2);
187         $k = 0;
188         while ($k < count($pval3)) {
189             $personsAll[] = $pval3[$k].",,,,名詞,固有名詞,人名,一
                般,*,*,,.$pval3[$k].",,,$pval3[$k].",,,固,*,*,*,*,".
                $ref;
190             $k++;
191         }
192     } else {
193         if ($pval2 != "") {
194             $personsAll[] = $pval2.",,,,名詞,固有名詞,人名,一般,*,*,,.
                $pval2.",,,$pval2.",,固,*,*,*,*,".$ref;
195         }
196     }
197 }
198 }
199
200 if (isset($place)) {
201     sort($place);
202     $place2 = array_unique($place, SORT_STRING);
203
204     foreach ($place2 as $plval) {
205         if ($plval != "" && preg_match('/^[^ -~~~ef^^bd^^a1-^^ef^^be^^9
            f 0 1 2 3 4 5 6 7 8 9](.+)/u', $plval)) {
206             $plval2 = preg_replace('/[ \| - \[\] ]/u', '', $plval);
207             if (strpos($plval2, "/") !== false) {
208                 $plval3 = explode("/", $plval2);
209                 $k = 0;
210                 while ($k < count($plval3)) {
211                     $placeAll[] = $plval3[$k].",,,,名詞,固有名詞,地名,一
                        般,*,*,,.$plval3[$k].",,,$plval3[$k].",,,
                        固,*,*,*,*,".$ref;
212                     $k++;
213                 }
214             } else {
215                 if ($plval2 != "") {
216                     $placeAll[] = $plval2.",,,,名詞,固有名詞,地名,一般,*,*,,.
                        $plval2.",,,$plval2.",,固,*,*,*,*,".$ref;
217                 }
218             }
219         }
220     }

```

付録 B . ソースコード一覧

```

221 }
222
223 foreach ($titles2 as $tval) {
224     #if ($tval != "" && preg_match('/^[^ -~ef^bd^a1-~ef^be^9
        f 0 1 2 3 4 5 6 7 8 9](.+)/u', $tval)) {
225         $tval2 = preg_replace('/[ \| - 「 」 『 』 〈 〉 ]/u', '', $tval);
226         if ($tval2 != "" && preg_match('/^[^ -~ef^bd^a1-~ef^be^9
            f 0 1 2 3 4 5 6 7 8 9](.+)/u', $tval2) && preg_match('/^(〔第
                他〕|(^ほか))([^-90 1 2 3 4 5 6 7 8 9一二三四五六七八九十]+)/u',
                    $tval2)) {
227             $titlesAll[] = $tval2.",,,,名詞,固有名詞,一般,*,*,*,, ".$tval2
                .",, ".$tval2.",,固,*,*,*,*, ".$ref;
228         }
229     }
230
231 if (isset($date)) {
232     sort($date);
233     $date2 = array_unique($date, SORT_STRING);
234
235     foreach ($date2 as $dval) {
236         if ($dval != "" && preg_match('/^[0-9]+/u', $dval)) {
237             $dateAll[] = $dval.",,,,名詞,固有名詞,一般,*,*,*,, ".$dval
                .",, ".$dval.",,固,*,*,*,*, ".$ref;
238         }
239     }
240 }
241
242 if (isset($subjects)) {
243     sort($subjects);
244     $subjects2 = array_unique($subjects, SORT_STRING);
245
246     foreach ($subjects2 as $sval) {
247         $sval2 = preg_replace('/[ \| / \, ]/u', "/", $sval);
248         if (strpos($sval2, "/") !== false) {
249             $sval3 = explode("/", $sval2);
250             $k = 0;
251             while ($k < count($sval3)) {
252                 $subjectAll[] = $sval3[$k].",,,,名詞,固有名詞,一般,*,*,*,, ".
                    $sval3[$k].",, ".$sval3[$k].",,固,*,*,*,*, ".$ref;
253                 $k++;
254             }
255         } else {
256             if ($sval != "") {
257                 $subjectAll[] = $sval.",,,,名詞,固有名詞,一般,*,*,*,, ".$sval
                    .",, ".$sval.",,固,*,*,*,*, ".$ref;
258             }
259         }

```

```
260     }
261 }
262
263 if (isset($materials)) {
264     sort($materials);
265     $materials2 = array_unique($materials, SORT_STRING);
266
267     foreach ($materials2 as $mval) {
268         if ($mval != "") {
269             $materialAll[] = $mval.",,,,名詞,固有名詞,一般,*,**,,$mval
                .",,$mval.",,固,*,**,*,$ref;
270         }
271     }
272 }
273
274 $i++;
275 }
276
277 #print_r($personsAll);
278 #print_r($placeAll);
279 #print_r($dateAll);
280 #print_r($subjectAll);
281 #exit;
282
283 sort($personsAll);
284 $personsAll2 = array_unique($personsAll, SORT_STRING);
285
286 foreach ($personsAll2 as $paval) {
287     print $paval."\n";
288 }
289
290 sort($placeAll);
291 $placeAll2 = array_unique($placeAll, SORT_STRING);
292
293 foreach ($placeAll2 as $placeval) {
294     print $placeval."\n";
295 }
296
297 //固有名詞が人名にある場合は固有名詞に登録しない(人名優先)
298 sort($titlesAll);
299 $titlesAll2 = array_unique($titlesAll, SORT_STRING);
300
301 foreach ($titlesAll2 as $taval) {
302     print $taval."\n";
303 }
304
305 sort($dateAll);
```


付録 B . ソースコード一覧

```
306 $dateAll2 = array_unique($dateAll, SORT_STRING);
307
308 foreach ($dateAll2 as $dateval) {
309     print $dateval."\n";
310 }
311
312 sort($subjectAll);
313 $subjectAll2 = array_unique($subjectAll, SORT_STRING);
314
315 foreach ($subjectAll2 as $subjval) {
316     print $subjval."\n";
317 }
318
319 sort($materialAll);
320 $materialAll2 = array_unique($materialAll, SORT_STRING);
321
322 foreach ($materialAll2 as $mtval) {
323     print $mtval."\n";
324 }
325
326 //単語群が典拠データ、ichiba データに存在しないか検査
327
328 //ファイルをuserdicdir 以下にリダイレクトで生成する
329 //~mecab_work/userdicdir/expertXml.o
330
331 ?>
```

ソースコード B.7: ユーザ辞書生成スクリプト集 (makeMecabDic.txt).

```
1 //参考用Mecab 辞書生成スクリプト集
2 //CentOS6.9のシェル上で実行します.
3 //php で自動実行するには、system()関数などを使います.
4
5 //Mecab 辞書用 csv ファイルの作成
6
7 //入力:単語リスト
8 //出力:コスト値が空のコストファイル(CSV形式)
9 php /home/yoshiga/mecab-work/userdicdir/makeCostFile.php;
10 php /home/yoshiga/mecab-work/userdicdir/makeCostFile_date.php;
11 php /home/yoshiga/mecab-work/userdicdir/makeCostFile_title.php;
12 php /home/yoshiga/mecab-work/userdicdir/makeCostFile_place.php;
13
14
15 //コスト値が入ったコストファイル生成スクリプト
```

付録 B . ソースコード一覧

```
16 //モデル値は別途ダウンロードします。
17 //入力：コスト値が空のコストファイル
18 //出力：コスト値が自動入力されたコストファイル
19 /usr/local/libexec/mecab/mecab-dict-index -m /home/yoshiga/mecab-
    work/unidic-relearn/model/model.def -d /home/yoshiga/mecab-work/
    unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/jinmeiCost
    .csv -f utf-8 -t utf-8 -a /home/yoshiga/mecab-work/userdicdir/
    jinmei.csv;
20
21 /usr/local/libexec/mecab/mecab-dict-index -m /home/yoshiga/mecab-
    work/unidic-relearn/model/model.def -d /home/yoshiga/mecab-work/
    unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/termsCost.
    csv -f utf-8 -t utf-8 -a /home/yoshiga/mecab-work/userdicdir/
    termsall.csv;
22
23 /usr/local/libexec/mecab/mecab-dict-index -m /home/yoshiga/mecab-
    work/unidic-relearn/model/model.def -d /home/yoshiga/mecab-work/
    unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/
    datetermCost.csv -f utf-8 -t utf-8 -a /home/yoshiga/mecab-work/
    userdicdir/dateterms.csv;
24
25 /usr/local/libexec/mecab/mecab-dict-index -m /home/yoshiga/mecab-
    work/unidic-relearn/model/model.def -d /home/yoshiga/mecab-work/
    unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/
    ichibajinmeirecordCost.csv -f utf-8 -t utf-8 -a /home/yoshiga/
    mecab-work/userdicdir/ichibajinmeirecord.csv;
26
27 /usr/local/libexec/mecab/mecab-dict-index -m /home/yoshiga/mecab-
    work/unidic-relearn/model/model.def -d /home/yoshiga/mecab-work/
    unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/verbsCost.
    csv -f utf-8 -t utf-8 -a /home/yoshiga/analysis/words/verbs.csv;
28
29 /usr/local/libexec/mecab/mecab-dict-index -m /home/yoshiga/mecab-
    work/unidic-relearn/model/model.def -d /home/yoshiga/mecab-work/
    unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/expertCost
    .csv -f utf-8 -t utf-8 -a /home/yoshiga/mecab-work/userdicdir/
    expertXml.o;
30
31 /usr/local/libexec/mecab/mecab-dict-index -m /home/yoshiga/mecab-
    work/unidic-relearn/model/model.def -d /home/yoshiga/mecab-work/
    unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/
    ichibaTitlesCost.csv -f utf-8 -t utf-8 -a /home/yoshiga/mecab-
    work/userdicdir/ichibaTitlesall.csv;
32
33 /usr/local/libexec/mecab/mecab-dict-index -m /home/yoshiga/mecab-
    work/unidic-relearn/model/model.def -d /home/yoshiga/mecab-work/
    unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/
```

```

    placetermsCost.csv -f utf-8 -t utf-8 -a /home/yoshiga/mecab-work/
    userdicdir/placeterms.csv;
34
35
36 //Mecab ユーザ辞書生成スクリプト
37 //入力:コスト値が自動入力されたコストファイル
38 //出力:ユーザ辞書(.dic ファイル)
39 /usr/local/libexec/mecab/mecab-dict-index -d /home/yoshiga/mecab-
    work/unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/
    jinmei.dic -f utf8 -t utf8 /home/yoshiga/mecab-work/
    userdicdirModel/jinmeiCost.csv;
40
41 /usr/local/libexec/mecab/mecab-dict-index -d /home/yoshiga/mecab-
    work/unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/
    ichibaTerms.dic -f utf8 -t utf8 /home/yoshiga/mecab-work/
    userdicdir/termsCost.csv;
42
43 /usr/local/libexec/mecab/mecab-dict-index -d /home/yoshiga/mecab-
    work/unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/
    ichibaDateTerms.dic -f utf8 -t utf8 /home/yoshiga/mecab-work/
    userdicdir/datetermCost.csv;
44
45 /usr/local/libexec/mecab/mecab-dict-index -d /home/yoshiga/mecab-
    work/unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/
    ichibaJinmeiRecord.dic -f utf8 -t utf8 /home/yoshiga/mecab-work/
    userdicdir/ichibajinmeirecordCost.csv;
46
47 /usr/local/libexec/mecab/mecab-dict-index -d /home/yoshiga/mecab-
    work/unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/
    ichibaVerbs.dic -f utf8 -t utf8 /home/yoshiga/mecab-work/
    userdicdir/verbsCost.csv;
48
49 /usr/local/libexec/mecab/mecab-dict-index -d /home/yoshiga/mecab-
    work/unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/
    ichibaVerbs.dic -f utf8 -t utf8 /home/yoshiga/mecab-work/
    userdicdir/seed/verbsCost0514.csv;
50
51 /usr/local/libexec/mecab/mecab-dict-index -d /home/yoshiga/mecab-
    work/unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/
    ichibaExpert.dic -f utf8 -t utf8 /home/yoshiga/mecab-work/
    userdicdir/expertCost.csv;
52
53 /usr/local/libexec/mecab/mecab-dict-index -d /home/yoshiga/mecab-
    work/unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/
    ichibaTitles.dic -f utf8 -t utf8 /home/yoshiga/mecab-work/
    userdicdir/ichibaTitlesCost.csv;
54

```

```

55 /usr/local/libexec/mecab/mecab-dict-index -d /home/yoshiga/mecab-
    work/unidic-relearn/ -u /home/yoshiga/mecab-work/userdicdir/
    places.dic -f utf8 -t utf8 /home/yoshiga/mecab-work/userdicdir/
    placetermsCost.csv;
56
57 /usr/local/libexec/mecab/mecab-dict-index -d /home/yoshiga/mecab-
    work/unidic-relearn/ -u /home/yoshiga/usercorpus/kokubun.dic -f
    utf8 -t utf8 /home/yoshiga/mecab-work/userdicdir/kokubunTermsCost
    .csv;
58
59
60 //ユーザ辞書作成後辞書設定ファイルの変更
61 //MeCab 設定ファイルを書き換えます.
62 sudo vi /usr/local/etc/mecabrc;

```

通常、XML の読み込みは 3MB 以上になると処理速度が極端に低下する。

ソースコード B.8: 3MB 以上の XML を読み込み MeCab 用 CSV などのフォーマットに変換可能なスクリプト (ndlaXmltoMecab.php).

```

1 <?php //xml 高速変換
2 $xml = new XMLReader();
3 $xml->open("ndladata.xml");
4
5 $fname = '';
6 $yomi = '';
7 $uri = '';
8
9 while ($xml->read()) {
10     switch ($xml->nodeType) {
11         case (XMLREADER::ELEMENT):
12             if ($xml->name == "result") {
13                 $node = $xml->expand();
14                 $dom = new DomDocument();
15                 $dom->preserveWhiteSpace = false;
16                 $n = $dom->importNode($node, true);
17                 $sdom = simplexml_import_dom($n);
18
19                 $fname_arr = $sdom->xpath("binding[@name='fname']");
20                 $fname = str_replace(",", "", $fname_arr[0]->literal);
21                 $fname = str_replace(" ", "", $fname);
22                 $yomi_arr = $sdom->xpath("binding[@name='yomi']");
23                 $yomi = str_replace(",", "", $yomi_arr[0]->literal);
24                 $uri_arr = $sdom->xpath("binding/uri");

```

```

25     $uri = $uri_arr[0];
26     #print $fname.", ".$yomi.", ".$uri."\n";
27
28     $ref = ", ".$uri;
29     print $fname.",,,,名詞,固有名詞,人名,一般,*,*,".$yomi.", ".$yomi
        .", ".$fname.", ".$yomi.", ".$fname.", ".$yomi.", 固,*,*,*,*".
        $ref."\n";
30     #print_r($sdom);
31     #print_r($fname);
32     #print "-----\n";
33 }
34 }
35 }
36 $xml->close();
37 ?>

```

ソースコード B.9: MeCab 設定ファイル (mecabrc).

```

1 ;
2 ; Configuration file of MeCab
3 ;
4 ; $Id: mecabrc.in,v 1.3 2006/05/29 15:36:08 taku-ku Exp $;
5 ;
6 dicdir = /usr/local/lib/mecab/dic/unidic
7
8 ;maindic for analysis
9 ;dicdir = /home/yoshiga/mecab-work/userdicdir/final
10 ;dicdir = /home/yoshiga/mecab-work/unidic-relearn
11
12 ;all loaded modules for Ichiba
13 ;userdic = /home/yoshiga/mecab-work/userdicdir/jinmei.dic,/home/
        yoshiga/mecab-work/userdicdir/ichibaTerms.dic,/home/yoshiga/mecab
        -work/userdicdir/ichibaDateTerms.dic,/home/yoshiga/mecab-work/
        userdicdir/ichibaJinmeiRecord.dic,/home/yoshiga/mecab-work/
        userdicdir/ichibaVerbs.dic,/home/yoshiga/mecab-work/userdicdir/
        ichibaExpert.dic,/home/yoshiga/mecab-work/userdicdir/ichibaTitles
        .dic,/home/yoshiga/mecab-work/userdicdir/places.dic
14
15 ;Kokubun loaded modules
16 ;userdic = /home/yoshiga/mecab-work/userdicKokubun/kokubun.dic,/home
        /yoshiga/mecab-work/userdicdir/jinmei.dic,/home/yoshiga/mecab-
        work/userdicdir/ichibaTerms.dic,/home/yoshiga/mecab-work/
        userdicdir/ichibaDateTerms.dic,/home/yoshiga/mecab-work/
        userdicdir/ichibaJinmeiRecord.dic,/home/yoshiga/mecab-work/

```

```

    userdicdir/ichibaExpert.dic,/home/yoshiga/mecab-work/userdicdir/
    ichibaTitles.dic,/home/yoshiga/mecab-work/userdicdir/places.dic
17
18 ; output-format-type = wakati
19 ; input-buffer-size = 8192
20
21 ; node-format = %m\n
22 ; bos-format = %S\n
23 ; eos-format = EOS\n

```

ソースコード B.10: コスト値生成ファイル (makeCostFile_date.php).

```

1 <?php
2 //home/yoshiga/analysis/dateterms.
   csv を dateterms_original としてコピーしておく。
3 $data1 = file("/home/yoshiga/analysis/words/dateterms.csv",
   FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
4 $data2 = file("/home/yoshiga/analysis/words/gengo.txt",
   FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
5 $data = array_merge($data1, $data2);
6
7 foreach ($data as $val) {
8   $valcomma = explode(",", $val);
9   print $valcomma[0];
10  $list[] = $valcomma[0].",,,,名詞,普通名詞,一般,*,*,*,",
   mb_convert_kana(@$valcomma[1], "C").",".$valcomma[0].",".
   $valcomma[0].",".mb_convert_kana(@$valcomma[1], "C").",".
   $valcomma[0].",".mb_convert_kana(@$valcomma[1], "C").",
   和,*,*,*,*,ICHIBA_DATETERMS";
11 }
12
13 //セキュリティにより、"/home/yoshiga/analysis/words/dateterms.csv"を讀
   み込みできないので、下記の場所にファイルを置く
14 $datafile = '/home/yoshiga/mecab-work/userdicdir/dateterms.csv';
15 $dataval = '';
16 if (is_writable($datafile)) {
17   if (!($fpw = fopen($datafile, 'wb'))){
18     trigger_error("データ書き込みファイルを開けません。");
19   } else {
20     foreach ($list as $dataval) {
21       fwrite($fpw, $dataval."\n");
22     }
23     fclose($fpw);
24   }

```

```

25 } else {
26     trigger_error("ファイルを開ける状態ではありません。");
27 }

```

ソースコード B.11: 書誌属性リストの例 (fieldDataIchiba.txt).

```

整理番号
1 ,seiribangou,dcterms:identifier,dataTypeNone,mainField,,,
2 分類,bunrui,,,dataTypeNone,mainField,,,Genre
3 分類1,bunrui1,,,mainField,mecab,,,Genre
4 分類2,bunrui2,,,mainField,mecab,,,Genre
5 分類3,bunrui3,,,mainField,mecab,,,Genre
6 分類4,bunrui4,,,mainField,mecab,,,Genre
7 分類5,bunrui5,,,mainField,mecab,,,Genre
8 順位,juni,,,mainField,,,
9 画像公開,gazoukoukai,,,mainField,,,画像を公開許可,
10 書名,shomei,dcterms:title,,,mainField,,,代表書名,MainTitle
11 読み,yomi,dcndl:Transcription,,,mainField,,,書名の読み仮名,MainTitle
12 書型,shokei,dcterms:extent,,,mainField,,,典籍の大きさ,
13 巻冊,kansatsu,,,mainField,,,典籍の構成,
14 編著者,henchosha,dcterms:creator,,,mainField,mecab,,,Person
15 刊行・書写年次,kankou_nen,dcterms:date,,,mainField,mecab,年号で表現され
    た日付,Date
16 西暦,seireki,dcterms:issued,,,mainField,,,西暦で表現された日付,Date
17 刊写,kansha,,,mainField,,,印刷および書写の別,
18 注記,chuki,dcterms:description,,,mainField,mecab,項目に当てはまらない詳
    細な記載や付記,
19 印記,inki,,,dataTypeNone,mainField,,,
20 文庫,bunko,,,mainField,,,コレクション名,Collection
21 他,hoka,,,dataTypeNone,mainField,,,

```

B.3 Linked Data化スクリプト群

ソースコード B.12: メインスクリプト (A, B, Cを連動, main.php).

```

1 <?php //mecabを使った文章の分割
2 include_once '/home/yoshiga/analysis/function.php';
3 include_once '/var/www/html/tsr/dbsetting.php';

```

```
4
5 $field = file("fieldDataIchiba.txt", FILE_IGNORE_NEW_LINES |
6     FILE_SKIP_EMPTY_LINES);
7
8 $sentence = array();
9 $mtextarray = array();
10
11 //////////////////////////////////////
12 //固有表現タグ付け
13 //mecab フラグ*が立っている行を見つける
14 $fldflag = array();
15 $i = 0;
16
17 foreach($field as $fldKey => $fldstr) {
18     if (mb_strpos($fldstr, "mainField") !== false) {
19         #print $fldKey."\n";
20         if (mb_strpos($fldstr, "mecab") !== false) {
21             $fldflag[] = $fldKey - $i;
22         }
23     } else {
24         $i++;
25     }
26 }
27 #print_r($fldflag); exit;
28
29 for($rn=0; $rn<count($record); $rn++) {
30     $value = explode(',', $record[$rn]);
31     #print_r($value);
32
33     foreach($value as $key => $val) {
34         $sentence_pre = preg_replace('/。| /', '@@', $val);
35         $sentence_pre = mb_ereg_replace('[@]{2}$|\n', '', $sentence_pre
36     );
37         $sentence[$rn][] = explode('@@', $sentence_pre);
38     }
39     #print_r($sentence); exit;
40
41     for ($snum = 0; $snum < count($sentence[$rn]); $snum++) {
42         for ($osnum = 0; $osnum < count($sentence[$rn][$snum]); $osnum++)
43             {
44                 if (array_search($snum, $fldflag) !== false) { //国文研注記は 21
45                     市場は 18
46                     //Mecab オブジェクトを生成
47                     //https://github.com/rsky/php-mecab
```



```
45     //Copyright (c) 2006-2015 Ryusuke SEKIYAMA. All rights
        reserved.
46     $mecab = new MeCab_Tagger();
47     $mtext = $mecab->parse(trim($sentence[$rn][$snum][$osnum]));
        //解析実行
48     $mtext = str_replace("EOS", '', $mtext);
49     } else {
50     $mtext = trim($sentence[$rn][$snum][$osnum]);
51     }
52     $mtextarray[$rn][$snum][$osnum] = $mdatavals[] = explode("\n",
        trim($mtext));
53     }
54     }
55 }
56
57 #print_r($sentence); exit;
58 #print_r($mtextarray); exit;
59
60 ///////////////////////////////////////////////////////////////////
61 //全ての単語の情報を抽出
62 $type = "ALL";
63
64 for ($i = 0; $i < count($mtextarray); $i++) {
65     $rn = $i;
66     for ($e = 0; $e < count($mtextarray[$i]); $e++) {
67         $snum = $e;
68         for ($f = 0; $f < @count($mtextarray[$i][$e]); $f++) {
69             $osnum = $f;
70             for ($j = 0; $j < count($mtextarray[$i][$e][$f]); $j++) {
71                 $tangoarray = explode("\t", $mtextarray[$i][$e][$f][$j]);
72                 $element = '';
73                 $element = @str_replace(",", ":", $tangoarray[1]);
74                 $data_all[] = $type.", ".$rn.", ".$snum.", ".$osnum.", ".$j.", ".
                    $j.", ".$tangoarray[0].", ".$element;
75             }
76         }
77     }
78 }
79
80 //全データをファイルに格納
81 $data = $data_all;
82 $datafile = '/home/yoshiga/analysis/data-all.csv';
83 $dataval = '';
84 if (is_writable($datafile)) {
85     if (!($fpw = fopen($datafile, 'wb'))) {
86         trigger_error("データ書き込みファイルを開けません。");
87     } else {
```

```
88     foreach ($data_all as $dataval) {
89         fwrite($fpw, $dataval."\n");
90     }
91     fclose($fpw);
92 }
93 } else {
94     trigger_error("ファイルを開ける状態ではありません。");
95 }
96
97 //データベースに格納
98 if (!$iddel = sql("DELETE from \"data-all\";")) {
99     trigger_error("中間データを消去できませんでした。");
100 } else {
101     $copyingdata = file($datafile, FILE_IGNORE_NEW_LINES |
102         FILE_SKIP_EMPTY_LINES);
103     if (!$idmdcopy = sqlCopy('\"data-all\"', $copyingdata, ',')) {
104         trigger_error("中間データを登録できませんでした。");
105     }
106 }
107 //////////////////////////////////////////////////
108 //Mecab 解析結果ファイル
109
110 $mdatafile = 'mtextdata.txt';
111
112 if (is_writable($mdatafile)) {
113     if (!$fpm = fopen($mdatafile, 'wb')) {
114         trigger_error("データ書き込みファイルを開けません。");
115     } else {
116         foreach ($mdatavals as $mdataval) {
117             foreach ($mdataval as $mdata) {
118                 fwrite($fpm, $mdata."\n");
119             }
120         }
121         fclose($fpm);
122     }
123 } else {
124     trigger_error("ファイルを開ける状態ではありません。");
125 }
126
127 $toki_tango = array();
128 $etc_tango = array();
129 $propernoun_tango = array();
130 $propernoun_tango2 = array();
131 $tob_tango = array();
132 $role_tango = array();
133 $terms_tango = array();
```

```
134 $all_tango1 = array();
135 $all_tango2 = array();
136 $place_tango = array();
137 $jinmei_tango = array();
138 $jinmei_tango2 = array();
139 $unknown = array();
140
141 //元号、年月、数詞、助数詞に関する単語データの抽出
142 require_once 'analysisDate.php';
143 rsort($toki_tango);
144 rsort($etc_tango);
145
146 //題(括弧内の文字列)と部品用語の抽出
147 require_once 'analysisPnoun.php';
148 require_once 'analysisPnoun2.php';
149 require_once 'analysisParts.php';
150 rsort($propernoun_tango);
151 rsort($propernoun_tango2);
152 rsort($tob_tango);
153 rsort($role_tango);
154 rsort($terms_tango);
155
156 //抽出した単語配列を連結
157 $all_tango1 = array_merge($toki_tango, $etc_tango,
    $propernoun_tango2, $tob_tango, $role_tango, $terms_tango);
158
159 //人名の抽出
160 require_once 'analysisJinmei.php';
161 rsort($jinmei_tango);
162
163 //抽出した単語配列を連結
164 $all_tango2 = array_merge($all_tango1, $jinmei_tango,
    $propernoun_tango);
165
166 //地名の抽出
167 require_once 'analysisPlace.php';
168 rsort($place_tango);
169
170 //抽出した単語配列を連結
171 $all_tango = array_merge($all_tango2, $place_tango);
172
173 //文脈(屋号)人名検出
174 require_once 'analysisJinmeiYago.php';
175
176 //推測で人名を探索
177 require_once 'analysisJinmei2.php';
178 rsort($jinmei_tango2);
```

付録 B . ソースコード一覧

```
179
180 //中間データをファイルに格納(固有表現由来情報)middledata.csv
181 $data = array_merge($all_tango, $jinmei_tango2);
182 sort($data, SORT_NATURAL);
183 $data = array_unique($data, SORT_STRING);
184 #print_r($data); exit;
185
186 $datafile = '/home/yoshiga/analysis/middledata.csv';
187 $dataval = '';
188 if (is_writable($datafile)) {
189     if (!(($fpw = fopen($datafile, 'wb')))) {
190         trigger_error("データ書き込みファイルを開けません。");
191     } else {
192         foreach ($data as $dataval) {
193             fwrite($fpw, $dataval."\n");
194         }
195         fclose($fpw);
196     }
197 } else {
198     trigger_error("ファイルを開ける状態ではありません。");
199 }
200
201 //中間データをデータベースsites にアップロード
202 if (!(($iddel = sql("DELETE from middledata;")))) {
203     trigger_error("中間データを消去できませんでした。");
204 } else {
205     $copyingdata = file($datafile, FILE_IGNORE_NEW_LINES |
206         FILE_SKIP_EMPTY_LINES);
207     if (!(($idmdcopy = sqlCopy('middledata', $copyingdata, ','))) {
208         trigger_error("中間データを登録できませんでした。");
209     }
210 }
211 //スクリプトD および E へ進む。
212 include_once "main-datacontrol.php";
213 for ($i = 0; $i<10; $i++) { echo "\007";} //解析終了のベル
214
215 ?>
```

ソースコード B.13: スクリプト B1 (時クラス抽出, analysisDate.php).

```
1 <?php //mecab を使った文章の分割: 日時の検出
2 //function kan2num(): 漢数字からアラビア数字に変換する関数(時を表す数詞の
   み変換する)
```

```
3
4 $josushi = 0;
5 $sushi_josushi = 0;
6 $josushi_sushi = 0;
7 $gengostr = '';
8 $josushistr = '';
9 $flag = 0; //ダブリ検出用フラグ
10
11 //ICHIBA_DATETERMS
12 $dterms = file("/home/yoshiga/analysis/words/dateterms.csv",
13     FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
14 foreach ($dterms as $dval) {
15     $dlist = explode(",", $dval);
16     $dateterms[] = $dlist[0];
17 }
18 $gengo = file("/home/yoshiga/analysis/words/gengo.txt",
19     FILE_IGNORE_NEW_LINES);
20 #print_r($fldflag);
21 #print_r($mtextarray); exit;
22
23 $rn = $snum = $osnum = 0;
24 for ($i = 0; $i < count($mtextarray); $i++) {
25     $rn = $i;
26     for ($e = 0; $e < count($fldflag); $e++) {
27         $snum = $fldflag[$e];
28         for ($f = 0; $f < @count($mtextarray[$i][$snum]); $f++) {
29             $osnum = $f;
30             for ($j = 0; $j < count($mtextarray[$i][$snum][$f]); $j++) {
31                 //各文中に出現する助数詞的単語は本当に助数詞か判定する
32                 $uri = "http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki
33                     /".str_pad($rn, 3, 0, STR_PAD_LEFT)."-" . $snum . "-" . $osnum
34                     . "-";
35                 $jp = $j - 1;
36                 $jp2 = $jp - 1;
37                 $jn = $j + 1;
38                 $temparray = array();
39                 $sushi = array();
40
41                 $tangoarray = @explode("\t", $mtextarray[$i][$snum][$f][$j]);
42                 $tangoarrayp = @explode("\t", $mtextarray[$i][$snum][$f][$jp
43                     ]);
44                 $tangoarrayn = @explode("\t", $mtextarray[$i][$snum][$f][$jn
45                     ]);
46                 $tangoarrayp2 = @explode("\t", $mtextarray[$i][$snum][$f][
47                     $jp2]);
```

```
43
44 //元号チェック
45 if ((@mb_strpos($tangoarray[1], "名詞,") != false) &&
46     array_search($tangoarray[0], $gengo) > 0){
47     $flag = 1;
48     $gengostr = $tangoarray[0];
49     $josushistr = "";
50
51     $josushi_sushi++;
52     $tempnum = $jn;
53
54     $pos = 0; //数詞の位置
55
56     if (mb_strpos($tangoarrayn[0], "元年") != false) {
57         $toki_tango[] = "Date",".$rn.", ".$snum.", ".$osnum.", ".$j
58             .", ".$jn.", ".$gengostr."元年.", ".$uri.$j."-".$jn;
59     } elseif (mb_strpos($tangoarrayn[0], "元") != false) {
60         $toki_tango[] = "Date",".$rn.", ".$snum.", ".$osnum.", ".$j
61             .", ".$jn.", ".$gengostr."元年.", ".$uri.$j."-".$jn;
62     } else {
63         while ($tempnum < count($mtextarray[$i][$snum][$f])) {
64             if (mb_strpos($mtextarray[$i][$snum][$f][$tempnum], "数
65                 詞") != false) {
66                 $temparray = explode("\t", $mtextarray[$i][$snum][$f][
67                     $tempnum]);
68                 $sushi[$pos] = $temparray[0];
69                 $pos++;
70                 $tempnum++;
71             } else {
72                 break;
73             }
74         }
75
76         //出力
77         //助数詞「年」が省略されている場合、年を追加して出力
78         $truedatenum = kan2num(implode($sushi),3); //出力
79         if (count($sushi) > 0 && $truedatenum > 0 && $truedatenum
80             < 100) {
81             $pos = $pos + 1 + $j;
82             $josushistr = "年";
83             $toki_tango[] = "Date",".$rn.", ".$snum.", ".$osnum.", ".$j
84                 .", ".$pos.", ".$gengostr.$truedatenum.$josushistr
85                 .", ".$uri.$j."-".$pos;
86         }
87     }
88 } elseif ((strpos($tangoarray[0], "同年") != false || strpos(
89 $tangoarray[0], "同月") != false) && strpos($tangoarray[1], "名
```

```
    詞,普通名詞") != false) {
81     } elseif (((mb_strpos($tangoarray[0], "同年") != false ||
        mb_strpos($tangoarray[0], "同月") != false) && mb_strpos(
        $tangoarray[1], "名詞,普通名詞") != false) || (mb_strpos(
        $tangoarray[0], "同") != false && array_search(
        $tangoarrayn[0], $dterms) > 0)) {
82     $flag = 1;
83     //前の文を遡って年を調べる
84     $tokiall = count($toki_tango);
85     $tokilast = $tokiall - 1; // $toki_tango の最後尾位置
86     $tokiprev = $tokilast; //
        $toki_tango の最後尾位置から前の値をみるための変数
87     while ($tokilast >= 0 && $tokiprev >= 0) {
88         if (mb_strpos($toki_tango[$tokiprev], "年") != false ||
            mb_strpos($toki_tango[$tokiprev], "月") != false ||
            mb_strpos($toki_tango[$tokiprev], "日") != false) {
89             $vals = explode(',', $toki_tango[$tokiprev]);
90             $toki_tango[] = "Date,".$rn.", ".$snum.", ".$osnum.", ".$j
                .",0,".$vals[6].", ".$uri.$j."-0"; //出力
91             break;
92         } else {
93             if ($tokiprev >= 0) { $tokiprev--; } else { break; }
94         }
95     }
96     #} elseif (strpos($tangoarray[0], "同") != false &&
        array_search($tangoarrayn[0], $dterms) > 0) {
97
98     } elseif (mb_strpos($tangoarrayp[0], "の") != false &&
99
100     mb_strpos($tangoarrayp[1], "助詞") != false &&
101     (mb_strpos($tangoarray[0], "年") != false || mb_strpos(
        $tangoarray[0], "月") != false)) {
102     $toki_tango[] = "Date,".$rn.", ".$snum.", ".$osnum.", ".$j.",
        $j,".$tangoarray[0].", ".$uri.$j."-".$j;
103     $toki_tango[] = "Date,".$rn.", ".$snum.", ".$osnum.", ".$jp.",
        $jp,".$tangoarrayp[0].", ".$uri.$jp."-".$jp;
104     $toki_tango[] = "Date,".$rn.", ".$snum.", ".$osnum.", ".$jp2
        .", $jp2,".$tangoarrayp2[0].", ".$uri.$jp2."-".$jp2;
105
106     } elseif (@mb_strpos($tangoarray[1], ",助数詞") != false ||
        mb_strpos($tangoarray[0], "肆") != false || mb_strpos(
        $tangoarray[0], "冊") != false) {
107     $josushi++;
108     $josushistr = $tangoarray[0];
109
110     //ichiba では「肆」を助数詞的に用いている。
111     if ($jp > -1 && mb_strpos($tangoarrayp[1], ",数詞") !=
```

```
112         false) {
113         //数詞+助数詞(十二年etc.)のパターンがある場合
114         $sushi_josushi++;
115         $tempnum = $jp;
116
117         $pos = 0; //数詞の位置
118
119         while ($tempnum >= 0) {
120             $temparray = explode("\t", $mtextarray[$i][$snum][$f][
121                 $tempnum]);
122             if (mb_strpos($temparray[1], "数詞") != false) {
123                 $sushi[$pos] = $temparray[0];
124                 $pos++;
125                 $tempnum--;
126             } else {
127                 $tempnum = $tempnum + 1;
128                 break;
129             }
130         }
131
132         //出力
133         if (count($sushi) > 0) {
134             krsort($sushi);
135             $truedatenum = kan2num(implode($sushi),3);
136             if ($tempnum < 0) {
137                 $tempnum = 0;
138             }
139
140             if ($josushistr == "年" && $truedatenum > 0 && $flag !=
141                 1) {
142                 $toki_tango[] = "Date, ".$rn.", ".$snum.", ".$osnum.", ".
143                     $tempnum.", ".$j.", ".$truedatenum.$josushistr.", ".
144                     $uri.$tempnum."-".$j; //出力
145             } elseif ($josushistr == "月" && $truedatenum > 0 &&
146                 $truedatenum < 13) {
147                 $toki_tango[] = "Date, ".$rn.", ".$snum.", ".$osnum.", ".
148                     $tempnum.", ".$j.", ".$truedatenum.$josushistr.", ".
149                     $uri.$tempnum."-".$j; //出力
150             } elseif ($josushistr == "日" && $truedatenum > 0 &&
151                 $truedatenum < 32) {
152                 $toki_tango[] = "Date, ".$rn.", ".$snum.", ".$osnum.", ".
153                     $tempnum.", ".$j.", ".$truedatenum.$josushistr.", ".
154                     $uri.$tempnum."-".$j; //出力
155             } else {
156                 if ($josushistr != "年" && $josushistr != "月" &&
157                     $josushistr != "日") {
158                     $etc_tango[] = "ETC, ".$rn.", ".$snum.", ".$osnum.", ".

```



```

        $tempnum.", ".$j.", ".implode($sushi).$josushistr
        .", ".$uri.$tempnum."-".$j;
147     }
148   }
149 }
150 } elseif ($jn < count($mtextarray[$i][$snum][$f]) &&
        mb_strpos($tangoarrayn[1], "数詞") !== false) {
151   //助数詞+数詞(巻二etc.)
152   $josushi_sushi++;
153   $tempnum = $jn;
154
155   while ($tempnum < count($mtextarray[$i][$snum][$f])) {
156     if (mb_strpos($mtextarray[$i][$snum][$f][$tempnum], "数
        詞") !== false) {
157       $temparray = explode("\t", $mtextarray[$i][$snum][$f][
        $tempnum]);
158       $sushi[] = $temparray[0];
159       $tempnum++;
160     } else {
161       break;
162     }
163   }
164
165   if (count($sushi) > 0) {
166     $etc_tango[] = "ETC, ".$rn.", ".$snum.", ".$osnum.", ".$j
        .", ".$tempnum - 1).", ".$josushistr. implode($sushi
        ).", ".$uri.$j."-".$tempnum - 1); //出力
167   }
168 } else {
169 }
170 } else {
171 if (@mb_strpos($tangoarray[1], "_DATETERMS") !== false ||
172 array_search($tangoarray[0], $dateterms) > 0) &&
173 mb_strpos($tangoarray[1], "人名") === false
174 ) {
175   //時節の単語は他の時節の単語と隣接していると仮定
176   $el = explode(",", end($toki_tango));
177   reset($toki_tango);
178
179   if ($el[1] == $rn && $el[2] == $snum && $el[3] == $osnum
        && $j == $el[5] + 1) {
180     $toki_tango[] = "Date, ".$rn.", ".$snum.", ".$osnum.", ".$j
        .", ".$j.", ".$tangoarray[0].", ".$uri.$j."-".$j;
181   }
182
183   if (mb_strpos($tangoarrayp[0], "の") !== false &&
184   mb_strpos($tangoarrayp[1], "助詞") !== false) {
```

```

185         //助詞「の」
186         $toki_tango[] = "Date",".$rn.",".$snum.",".$osnum.",".$jp
           .",".$jp.",".$tangoarrayp[0].",".$uri.$jp."-".$jp;
187
188         //「の」の前の単語も時節を表わすと仮定
189         $toki_tango[] = "Date",".$rn.",".$snum.",".$osnum.",".
           $jp2.",".$jp2.",".$tangoarrayp2[0].",".$uri.$jp2
           ."-".$jp2;
190
191         $toki_tango[] = "Date",".$rn.",".$snum.",".$osnum.",".$j
           .",".$j.",".$tangoarray[0].",".$uri.$j."-".$j;
192     }
193 }
194 }
195 }
196 }
197 }
198 }
199 ?>

```

ソースコード B.14: スクリプト B1 (題クラス抽出 1, analysisPnoun.php).

```

1 <?php //mecab を使った文章の分割: 括弧内の固有名詞(題)の検出
2 $uri = '';
3
4 $tango = array_merge($toki_tango, $etc_tango);
5
6 for ($i = 0; $i < count($mtextarray); $i++) {
7     $rn = $i;
8     for ($e = 0; $e < count($fldflag); $e++) {
9         $snum = $fldflag[$e];
10        $nokakkoflag = 0;
11        for ($f = 0; $f < @count($mtextarray[$i][$snum]); $f++) {
12            $osnum = $f;
13            for ($j = 0; $j < count($mtextarray[$i][$snum][$f]); $j++) {
14                $tangoarray = @explode("\t", $mtextarray[$i][$snum][$f][$j]);
15                $uri = "http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki
           /".str_pad($rn, 3, 0, STR_PAD_LEFT)
16        ."-".$snum."-".$osnum."-";
17
18        //括弧内の文字列抽出
19        if (@mb_strpos($tangoarray[1], "括弧開") !== false &&
           preg_match('/「|^ef^bd^a2|『/′', $tangoarray[0])){
20            $fj = 0;

```

```
21     $fj = $j + 1;
22     $fj2 = $fj;
23     $propernoun = '';
24     $ftango = array();
25
26     //開き括弧の位置を記憶
27     $countkakko = 0;
28
29     while ($fj < count($mtextarray[$i][$snum][$f])) {
30         $ftango = explode("\t", $mtextarray[$i][$snum][$f][$fj]);
31         if (mb_strpos($ftango[1], "括弧閉") !== false &&
32             preg_match('/\| |^ef^bd^a3| \|', $ftango[1])) {
33             $propernoun_tango[] = "PNoun,".$rn.",".$snum.",".$osnum
34                 .",".$j.",".$fj.",".$propernoun.",".$uri.$j."-".$fj;
35             break;
36         } else {
37             $propernoun .= $ftango[0];
38         }
39         $fj++;
40     }
41
42     //○から始まるタイトル
43     if (@mb_strpos($tangoarray[1], "○") !== false) {
44         $fj2 = 0;
45         $fj2 = $j + 1;
46         $propernoun2 = '';
47         while ($fj2 < count($mtextarray[$i][$snum][$f])) {
48             $ftango2 = explode("\t", $mtextarray[$i][$snum][$f][$fj2
49                 ]);
50             if (mb_strpos($ftango2[0], "○") === false && mb_strpos(
51                 $ftango2[0], "(") === false && mb_strpos($ftango2[0],
52                 " ") === false) {
53                 $propernoun2 .= $ftango2[0];
54                 $fj2++;
55             } else {
56                 break;
57             }
58         }
59         if ($fj2 < 0) {
60             $fj2 = 0;
61         }
62         $propernoun_tango2[] = "PNoun2,".$rn.",".$snum.",".$osnum
63             .",".$(j + 1).",".$(fj2 - 1).",".trim($propernoun2).",".
64             $uri.$(j + 1)."-".$(fj2 - 1);
65     }
```

付録 B . ソースコード一覧

```
61     }
62   }
63 }
64 }
65 ?>
```

ソースコード B.15: スクリプト B2 (題クラス抽出 2, analysisPnoun2.php).

```
1 <?php //mecab を使った文章の分割：括弧内の固有名詞および部品の検出
2 $uri = '';
3
4 $tango = array_merge($toki_tango, $etc_tango);
5
6 for ($i = 0; $i < count($mtextarray); $i++) {
7   $rn = $i;
8   for ($e = 0; $e < count($fldflag); $e++) {
9     $snum = $fldflag[$e];
10    $nokakkoflag = 0;
11    for ($f = 0; $f < @count($mtextarray[$i][$snum]); $f++) {
12      $osnum = $f;
13      for ($j = 0; $j < count($mtextarray[$i][$snum][$f]); $j++) {
14        $tangoarray = @explode("\t", $mtextarray[$i][$snum][$f][$j]);
15        $uri = "http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki
16          /".str_pad($rn, 3, 0, STR_PAD_LEFT)
17          . "-". $snum . "-". $osnum . "-";
18        if (@mb_strpos($tangoarray[1], "_TITLE") !== false) {
19          $propernoun_tango2[] = "PNoun2, ". $rn . ", ". $snum . ", ". $osnum
20            . ", ". $j . ", ". $j . ", ". $tangoarray[0] . ", ". $uri . $j . "-". $j;
21        }
22      }
23    }
24  }
25 ?>
```

ソースコード B.16: スクリプト B1 (用語クラス抽出, analysisParts.php).

```
1 <?php //mecab を使った文章の分割：括弧内の固有名詞および部品の検出
2 $uri = '';
```

```

3 $keyFromList = '';
4
5 $tobBase = file("/home/yoshiga/analysis/words/tob.csv",
6     FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
7 $i = 0;
8 $tobArray = array();
9 while (count($tobBase) > $i) {
10     $tobArray = explode(",", $tobBase[$i]);
11     $tob[$i] = $tobArray[0];
12     $i++;
13 }
14 $roleBase = file("/home/yoshiga/analysis/words/role.csv",
15     FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
16 $i = 0;
17 $roleArray = array();
18 while (count($roleBase) > $i) {
19     $roleArray = explode(",", $roleBase[$i]);
20     $role[$i] = $roleArray[0];
21     $i++;
22 }
23 $termsBase = file("/home/yoshiga/analysis/words/terms.csv",
24     FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
25 $i = 0;
26 $termsArray = array();
27 while (count($termsBase) > $i) {
28     $termsArray = explode(",", $termsBase[$i]);
29     $terms[$i] = $termsArray[0];
30     $i++;
31 }
32 for ($i = 0; $i < count($mtextarray); $i++) {
33     $rn = $i;
34     for ($e = 0; $e < count($fldflag); $e++) {
35         $snum = $fldflag[$e];
36         $nokakkoflag = 0;
37         for ($f = 0; $f < @count($mtextarray[$i][$snum]); $f++) {
38             $osnum = $f;
39             for ($j = 0; $j < count($mtextarray[$i][$snum][$f]); $j++) {
40                 $tangoarray = @explode("\t", $mtextarray[$i][$snum][$f][$j]);
41                 //用語チェック
42                 if ((array_search($tangoarray[0], $tob) !== false ||
43                     @mb_strpos($tangoarray[1], "_TOB") !== false) &&
44                     checkKeywords($i, $snum, $f, $j, $j, $etc_tango) == 0){
45                     $keyFromList = array_search($tangoarray[0], $tob);
46                     $uri = "http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/

```

```

        tenseki/".str_pad($rn, 3, 0, STR_PAD_LEFT)."-"."$snum
        ."-"."$osnum."-"."$j."-"."$j;
45     $tob_tango[] = "TFB,"."$rn."."$snum."."$osnum."."$j."."
        $j."."$tangoarray[0]."."$uri; //出力
46     }
47
48     if (array_search($tangoarray[0], $role) !== false &&
        checkKeywords($i, $snum, $f, $j, $j, $etc_tango) == 0){
49         $keyFromList = array_search($tangoarray[0], $role);
50         $uri = "http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/
        tenseki/".str_pad($rn, 3, 0, STR_PAD_LEFT)."-"."$snum
        ."-"."$osnum."-"."$j."-"."$j;
51         $role_tango[] = "Role,"."$rn."."$snum."."$osnum."."$j
        ."."$j."."$tangoarray[0]."."$uri; //出力
52     }
53
54     if (array_search($tangoarray[0], $terms) !== false &&
        checkKeywords($i, $snum, $f, $j, $j, $etc_tango) == 0){
55         $keyFromList = array_search($tangoarray[0], $terms);
56         $uri = "http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/
        tenseki/".str_pad($rn, 3, 0, STR_PAD_LEFT)."-"."$snum
        ."-"."$osnum."-"."$j."-"."$j;
57         $terms_tango[] = "Term,"."$rn."."$snum."."$osnum."."$j
        ."."$j."."$tangoarray[0]."."$uri; //出力
58     }
59     $keyFromList = '';
60     $uri = '';
61 }
62 }
63 }
64 }
65 ?>

```

ソースコード B.17: スクリプト B1 (場所クラス抽出, analysisPlace.php).

```

1 <?php //mecab を使った文章の分割: 地名の検出
2 $placeterms1 = file("/home/yoshiga/analysis/words/kokubun-place.csv
    ", FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
3 $placeterms2 = preg_replace('/[<> ]/u', '', $placeterms1);
4 $placeterms = array_merge($placeterms1, $placeterms2);
5 #print_r($placeterms); exit;
6 $placeuri = '';
7
8 $rn = $snum = $osnum = 0;

```

```
9 for ($i = 0; $i < count($mtextarray); $i++) {
10     $rn = $i;
11     for ($e = 0; $e < count($fldflag); $e++) {
12         $snum = $fldflag[$e];
13         for ($f = 0; $f < @count($mtextarray[$i][$snum]); $f++) {
14             $osnum = $f;
15             for ($j = 0; $j < count($mtextarray[$i][$snum][$f]); $j++) {
16                 $tangoarray = explode("\t", $mtextarray[$i][$snum][$f][$j]);
17                 $element = @explode(",", $tangoarray[1]);
18
19                 //抽出単語を除外した文字列の抽出
20
21                 if (checkKeywords($i, $snum, $f, $j, $j, $all_tango2) == 0) {
22                     //地名チェック
23                     if (@mb_strpos($tangoarray[1], "地名") !== false ||
24                         array_search($tangoarray[0], $placeterms) !== false){
25                         $placeuri = "http://www.dl.saga-u.ac.jp/OgiNabesima/
26                             ichiba/tenseki/".str_pad($rn, 3, 0, STR_PAD_LEFT)
27                 ."-". $snum."-". $osnum."-";
28
29                         $place_tango[] = "Place,".$rn.", ".$snum.", ".$osnum.", ".$j
30                             .", ".$j.", ".$tangoarray[0].", ".$placeuri.$j."-". $j; //
31                             出力
32                     }
33                 }
34             }
35         }
36     }
37 }
```

ソースコード B.18: スクリプト B1 (人名クラス抽出, analysisJinmeiYago.php).

```
1 <?php //mecabを使った文章の分割
2 //不明な文字列の抽出
3 $uri = '';
4 $yago = file("/home/yoshiga/analysis/words/yago.csv",
5     FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
6
7 //あとで人名(呼称)をmecabに登録
8 $pattern = "/" . implode("", $yago) . "翁氏)|(散人)|(山人)|(産人)|(主人)/
9     u";
10
11 $rn = $snum = $osnum = 0;
```

```
10 for ($i = 0; $i < count($mtextarray); $i++) {
11     $rn = $i;
12     for ($e = 0; $e < count($fldflag); $e++) {
13         $snum = $fldflag[$e];
14         for ($f = 0; $f < @count($mtextarray[$i][$snum]); $f++) {
15             $osnum = $f;
16             for ($j = 0; $j < count($mtextarray[$i][$snum][$f]); $j++) {
17                 //人名チェック4:屋号を探す
18                 $uri = "http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki
19                     /".str_pad($rn, 3, 0, STR_PAD_LEFT)."-$snum."-$osnum
20                     ."-";
21                 $tangoarray = explode("\t", $mtextarray[$i][$snum][$f][$j]);
22
23                 if (preg_match($pattern, $tangoarray[0])) {
24                     $cnt = 0;
25                     $jne = $j + 1;
26                     $jpr = $j - 1;
27                     $unitwordp = array();
28                     $unitwordn = array();
29                     $unitwordpKRsort = '';
30
31                     while ($jpr >= 0 && checkKeywords($i, $snum, $f, $jpr, $j,
32                         $all_tango) == 0) {
33                         $tangoarrayp = explode("\t", $mtextarray[$i][$snum][$f][
34                             $jpr]);
35
36                         if (checkKeywords2($i, $snum, $f, $jpr, $j, $all_tango1)
37                             > 0) {
38                             break;
39                         } elseif (
40                             preg_match('/[^^ef^^bd^^a5・○／、()] /u', $tangoarrayp
41                                 [0])
42                         ) {
43                             break;
44                         } elseif (
45                             mb_strpos($tangoarrayp[1], "助詞") !== false && (
46                                 mb_strpos($tangoarrayp[0], "の") === false &&
47                                 mb_strpos($tangoarrayp[0], "や") === false)
48                         ) {
49                             break;
50                         } else {
51                             #print "(屋号)".$i.", ".$snum.", ".$f.", ".$($jpr + 1).", ".
52                                 $j.", ".$tangoarrayp[0].$tangoarray[0]."\n";
53                             $unitwordp[$cnt] = $tangoarrayp[0];
54                             $jpr--;
55                             $cnt++;
56                         }
57                     }
58                 }
59             }
60         }
61     }
62 }
```



```

48     }
49
50     krsort($unitwordp);
51     $unitwordpKRsort = implode("", $unitwordp);
52
53     if (trim($unitwordpKRsort) != "") {
54         $jinmei_tango2[] = "Person",".$rn.",".$snum.",".$osnum
                    .",".$jpr + 1).",".$j.",".$unitwordpKRsort.
                    $tangoarray[0].",".$uri.($jpr + 1)."-".$j;
55     }
56     } else {
57         //UNKnown は無視 2016-06-07
58         /*if (checkKeywords($i, $snum, $f, $j, $all_tango) == 0) {
59             $unknown[] = $type.",".$rn.",".$snum.",".$osnum.",".$j
                    .",".$j.",".$tangoarray[0].",".$uri;
60         }*/
61     }
62 }
63 }
64 }
65 }
66 ?>

```

ソースコード B.19: スクリプト B2 (人名クラス推定, analysisJinmei2.php).

```

1 <?php //mecab を使った文章の分割: 人名の検出 2
2 $rn = $snum = $osnum = 0;
3 for ($i = 0; $i < count($mtextarray); $i++) {
4     $rn = $i;
5     for ($e = 0; $e < count($fldflag); $e++) {
6         $snum = $fldflag[$e];
7         #print "fldflag: ".$snum."\n";
8         for ($f = 0; $f < @count($mtextarray[$i][$snum]); $f++) {
9             $osnum = $f;
10            for ($j = 0; $j < count($mtextarray[$i][$snum][$f]); $j++) {
11                $tangoarray = explode("\t", $mtextarray[$i][$snum][$f][$j]);
12                $element = @explode(",", $tangoarray[1]);
13                $uri = "http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki
                    /".str_pad($rn, 3, 0, STR_PAD_LEFT)."-".$snum."-".$osnum
                    ."-";
14
15                //人名チェック 2:文末に部品名, 人名がその直前に来るパターン
16                $jq = 0;
17                $jq = $j - 1;

```

```
18     $jqf = $jq + 1;
19
20     if ($jq >= 0) {
21         $tangoarrayq = explode("\t", $mtextarray[$i][$snum][$f][$jq
22             ]);
23     }
24     $unitwordrev = array();
25     $item = 0;
26
27     if ((array_search($tangoarray[0], $tob) !== false ||
28         array_search($tangoarray[0], $role) !== false)) {
29         while ($jq >= 0 && checkKeywords($i, $snum, $f, $jq, ($j -
30             1), $all_tango) == 0) {
31             $tangoarrayq = explode("\t", $mtextarray[$i][$snum][$f][
32                 $jq]);
33
34             if (checkKeywords2($i, $snum, $f, $jq, ($j - 1),
35                 $all_tango1) > 0) {
36                 break;
37             } elseif (preg_match('/[^^ef^^bd^^a5・()、*]/u',
38                 $tangoarrayq[0])) {
39                 break;
40             } elseif (@mb_strpos($tangoarrayq[1], "助詞") !== false &&
41                 $jq == $j - 1
42             ) {
43                 print "(助詞)".$i.", ".$snum.", ".$f.", ".$jq.", ".$j -
44                     1).", ".$tangoarrayq[0]."\n";
45                 break;
46             } elseif (@mb_strpos($tangoarrayq[1], "助詞") !== false &&
47                 @mb_strpos($tangoarrayq[0], "の") === false &&
48                 mb_strpos($tangoarrayq[0], "ゃ") === false
49             ) {
50                 break;
51             } else {
52                 $unitwordrev[$item] = $tangoarrayq[0];
53                 $jq--;
54                 $item++;
55             }
56         }
57     }
58
59     krsort($unitwordrev);
60     $unitwords = implode("", $unitwordrev);
61
62     if (trim($unitwords) != "" && ($jq + 1 <= $j - 1)) {
63         $jinmei_tango2[] = "Person, ".$rn.", ".$snum.", ".$osnum
64             .", ".$jq + 1).", ".$j - 1).", ".$unitwords.", ".$uri.(
65             $jq + 1)."-".$j - 1);
66     }
```

```

56         }
57
58     }
59 }
60 }
61 }
62 }
63 ?>

```

ソースコード B.20: スクリプト D および E (データクリーニングおよび URI 付与スクリプトを連動, main-datacontrol.php).

```

1 <?php //スクリプトD: main.php で作成したデータから不要なデータを除去する
2 include_once '/home/yoshiga/analysis/function.php';
3 include_once '/var/www/html/tsr/dbsetting.php';
4
5 function dataControl0() {
6     include_once '/home/yoshiga/analysis/function.php';
7     include_once '/var/www/html/tsr/dbsetting.php';
8     //検出した人名をチェックする
9     //2016-09-21 fieldnum 変更
10    $id = sql("select * from middldata where type = 'Person' and
11             fieldnum = 17 order by record, sentence, wordstart, wordend,
12             oid;");
13
14    $delarr = array();
15
16    //不要な重複データを削除する
17    $iddel3 = sql("delete from middldata as m1 where exists (
18    select * from middldata as m2 where
19    m1.type = 'Person' and
20    m1.record = m2.record and
21    m1.fieldnum = 17 and
22    m1.fieldnum = m2.fieldnum and
23    m1.sentence = m2.sentence and
24    m1.wordstart = m2.wordstart and
25    m1.wordend = m2.wordend and
26    m1.word = m2.word and
27    m1.oid > m2.oid
28    );");
29
30    #@print_r($delarr);

```

```
30     return count($delarr);
31 }
32
33 function dataControl1() {
34     include_once '/home/yoshiga/analysis/function.php';
35     include_once '/var/www/html/tsr/dbsetting.php';
36     //検出した人名をチェックする
37     //2016-09-21 fieldnum 変更
38     $sid = sql("select * from middledata where type = 'Person' and
39         fieldnum = 17 order by record, sentence, wordstart, wordend
40         ;");
41     #$sid = sql("select * from middledata where type = 'Person' order
42         by record, fieldnum, sentence, wordstart;");
43
44     $delarr = array();
45
46     //不要な重複データを削除する
47     for ($i = 0; $i < pg_num_rows($sid); $i++) {
48
49         //○を除去(ichibaのみ)
50         if (pg_fetch_result($sid, $i, 'word') == "○") {
51             $delarr[] = pg_fetch_result($sid, $i, 'record').",".
52                 pg_fetch_result($sid, $i, 'sentence').",".pg_fetch_result(
53                     $sid, $i, 'wordstart').",".pg_fetch_result($sid, $i, '
54                     wordend').",".pg_fetch_result($sid, $i, 'word');
55         }
56
57         //隣り合う行が同じレコードで同じ文
58         if (@pg_fetch_result($sid, $i, 'record') == @pg_fetch_result($sid
59             , ($i + 1), 'record') &&
60             @pg_fetch_result($sid, $i, 'sentence') == @pg_fetch_result($sid,
61                 ($i + 1), 'sentence')) {
62
63             //前の行のwordの方が文字数少ない
64             if (@pg_fetch_result($sid, $i, 'wordstart') > @pg_fetch_result
65                 ($sid, ($i + 1), 'wordstart') &&
66                 @pg_fetch_result($sid, $i, 'wordend') < @pg_fetch_result($sid,
67                     ($i + 1), 'wordend')
68             ) {
69                 $delarr[] = pg_fetch_result($sid, $i, 'record').",".
70                     pg_fetch_result($sid, $i, 'sentence').",".pg_fetch_result
71                         ($sid, $i, 'wordstart').",".pg_fetch_result($sid, $i, '
72                         wordend').",".pg_fetch_result($sid, $i, 'word');
73
74             } elseif (@pg_fetch_result($sid, $i, 'wordstart') <
75                 @pg_fetch_result($sid, ($i + 1), 'wordstart') &&
76                 @pg_fetch_result($sid, $i, 'wordend') > @pg_fetch_result($sid,
```

```
        ($i + 1), 'wordend')
63     ) {
64
65     //後の行のwordの方が文字数少ないか同じ
66     $delarr[] = pg_fetch_result($id, ($i + 1), 'record').",".
        pg_fetch_result($id, ($i + 1), 'sentence').",".
        pg_fetch_result($id, ($i + 1), 'wordstart').",".
        pg_fetch_result($id, ($i + 1), 'wordend').",".
        pg_fetch_result($id, ($i + 1), 'word');
67
68     } else {
69     if (@pg_fetch_result($id, $i, 'wordstart') ==
        @pg_fetch_result($id, ($i + 1), 'wordstart')) {
70     if (mb_strlen(pg_fetch_result($id, $i, 'word')) <
        mb_strlen(pg_fetch_result($id, ($i + 1), 'word'))) {
71     $delarr[] = pg_fetch_result($id, $i, 'record').",".
        pg_fetch_result($id, $i, 'sentence').",".
        pg_fetch_result($id, $i, 'wordstart').",".
        pg_fetch_result($id, $i, 'wordend').",".
        pg_fetch_result($id, $i, 'word');
72     } else {
73     $delarr[] = pg_fetch_result($id, ($i + 1), 'record
        ').",".pg_fetch_result($id, ($i + 1), 'sentence
        ').",".pg_fetch_result($id, ($i + 1), 'wordstart
        ').",".pg_fetch_result($id, ($i + 1), 'wordend
        ').",".pg_fetch_result($id, ($i + 1), 'word');
74     }
75     } elseif (@pg_fetch_result($id, $i, 'wordend') ==
        @pg_fetch_result($id, ($i + 1), 'wordend')) {
76     if (mb_strlen(pg_fetch_result($id, $i, 'word')) <
        mb_strlen(pg_fetch_result($id, ($i + 1), 'word'))) {
77     $delarr[] = pg_fetch_result($id, $i, 'record').",".
        pg_fetch_result($id, $i, 'sentence').",".
        pg_fetch_result($id, $i, 'wordstart').",".
        pg_fetch_result($id, $i, 'wordend').",".
        pg_fetch_result($id, $i, 'word');
78     } else {
79     $delarr[] = pg_fetch_result($id, ($i + 1), 'record
        ').",".pg_fetch_result($id, ($i + 1), 'sentence
        ').",".pg_fetch_result($id, ($i + 1), 'wordstart
        ').",".pg_fetch_result($id, ($i + 1), 'wordend
        ').",".pg_fetch_result($id, ($i + 1), 'word');
80     }
81     } else {
82
83     }
84     }
```

```

85     }
86 }
87
88 //データベースから削除
89 if (isset($delarr)) {
90     foreach ($delarr as $val) {
91         $value = explode(",", $val);
92         $iddel = sql("delete from middledata where type = 'Person'
                       and fieldnum = 17 and record = ".$value[0].
                       " and sentence = ".$value[1].
                       " and wordstart = ".$value[2].
                       " and wordend = ".$value[3].
                       " and word = '".$value[4]."' ;");
93     }
94 }
95 #@print_r($delarr);
96
97 return count($delarr);
98
99 }
100
101 function dataControl2() {
102     //main.php で作成したデータから不要なデータを除去する 2
103     include_once '/home/yoshiga/analysis/function.php';
104     include_once '/var/www/html/tsr/dbsetting.php';
105
106     //検出した人名をチェックする
107     $sid = sql("select * from middledata where type = 'Person' and
                fieldnum = 17 order by record, sentence, wordstart, wordend
                ;");
108
109     //隣接する人名をつなげる
110     for ($i = 0; $i < pg_num_rows($sid); $i++) {
111         if (pg_fetch_result($sid, $i, 'record') == @pg_fetch_result($sid,
                                ($i + 1), 'record') &&
112             pg_fetch_result($sid, $i, 'sentence') == @pg_fetch_result($sid, (
                                $i + 1), 'sentence') &&
113             (pg_fetch_result($sid, $i, 'wordend') + 1) == pg_fetch_result(
                                $sid, ($i + 1), 'wordstart')) {
114
115             //スクリプトE (URI 付与)
116             $uri = "http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki
                    /".str_pad(pg_fetch_result($sid, $i, 'record'), 3, 0,
                                STR_PAD_LEFT)."-".pg_fetch_result($sid, $i, 'fieldnum
                    ')."-".pg_fetch_result($sid, $i, 'sentence')."-".
                    pg_fetch_result($sid, $i, 'wordstart')."-".pg_fetch_result(
                                $sid, ($i + 1), 'wordend');
117
118             $sidins = sql("insert into middledata values (

```

```
119         'Person',
120         ".pg_fetch_result($id, $i, 'record').",
121         ".pg_fetch_result($id, $i, 'fieldnum').",
122         ".pg_fetch_result($id, $i, 'sentence').",
123         ".pg_fetch_result($id, $i, 'wordstart').",
124         ".pg_fetch_result($id, ($i + 1), 'wordend').",
125         '".pg_fetch_result($id, $i, 'word').pg_fetch_result($id, ($i
126         + 1), 'word')."',
127         '". $uri."'
128     );");
129
130     if ($idins) {
131         //インサートできたら古いデータを削除
132         $iddel1 = sql("delete from middledata where record = ".
133             pg_fetch_result($id, $i, 'record')." and fieldnum = 17
134             and sentence = ".pg_fetch_result($id, $i, 'sentence')."
135             and wordstart = ".pg_fetch_result($id, $i, 'wordstart
136             ')." and wordend = ".pg_fetch_result($id, $i, 'wordend
137             ')." and word = '".pg_fetch_result($id, $i, 'word')."'
138             ;");
139
140         $iddel2 = sql("delete from middledata where record = ".
141             pg_fetch_result($id, ($i + 1), 'record')." and fieldnum
142             = 17 and sentence = ".pg_fetch_result($id, ($i + 1), '
143             sentence')." and wordstart = ".pg_fetch_result($id, ($i
144             + 1), 'wordstart')." and wordend = ".pg_fetch_result($id
145             , ($i + 1), 'wordend')." and word = '".pg_fetch_result(
146             $id, ($i + 1), 'word')."' ;");
147     } else {
148         return false;
149     }
150 }
151
152 return true;
153 }
154
155 function dataControl2Date() {
156     //main.php で作成したデータから不要なデータを除去する 2
157     include_once '/home/yoshiga/analysis/function.php';
158     include_once '/var/www/html/tsr/dbsetting.php';
159
160     //検出した日付をチェックする
161     $sid = sql("select * from middledata where type = 'Date' and
162         fieldnum = 17 order by record, sentence, wordstart, wordend
163         ;");
164
165     //隣接する日付をつなげる
```

```
151     for ($i = 0; $i < pg_num_rows($id); $i++) {
152         if (pg_fetch_result($id, $i, 'record') == @pg_fetch_result($id,
153             ($i + 1), 'record') &&
154             pg_fetch_result($id, $i, 'sentence') == @pg_fetch_result($id, (
155                 $i + 1), 'sentence') &&
156             (pg_fetch_result($id, $i, 'wordend') + 1) == pg_fetch_result(
157                 $id, ($i + 1), 'wordstart')) {
158
159             //スクリプトE (URI 付与)
160             $uri = "http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki
161                 /".str_pad(pg_fetch_result($id, $i, 'record'), 3, 0,
162                 STR_PAD_LEFT)."-" . pg_fetch_result($id, $i, 'fieldnum
163                 ')."-" . pg_fetch_result($id, $i, 'sentence')."-" .
164                 pg_fetch_result($id, $i, 'wordstart')."-" . pg_fetch_result(
165                 $id, ($i + 1), 'wordend');
166
167             $idins = sql("insert into middledata values (
168                 'Date',
169                 ".pg_fetch_result($id, $i, 'record').",
170                 ".pg_fetch_result($id, $i, 'fieldnum').",
171                 ".pg_fetch_result($id, $i, 'sentence').",
172                 ".pg_fetch_result($id, $i, 'wordstart').",
173                 ".pg_fetch_result($id, ($i + 1), 'wordend').",
174                 '".pg_fetch_result($id, $i, 'word').pg_fetch_result($id, ($i
175                     + 1), 'word')."',
176                 '". $uri."'
177             );");
178
179             if ($idins) {
180                 //インサートできたら古いデータを削除
181                 $iddel1 = sql("delete from middledata where record = ".
182                     pg_fetch_result($id, $i, 'record')." and fieldnum = 17
183                     and sentence = ".pg_fetch_result($id, $i, 'sentence')."
184                     and wordstart = ".pg_fetch_result($id, $i, 'wordstart
185                     ')." and wordend = ".pg_fetch_result($id, $i, 'wordend
186                     ')." and word = '".pg_fetch_result($id, $i, 'word')."
187                     ";");
188
189                 $iddel2 = sql("delete from middledata where record = ".
190                     pg_fetch_result($id, ($i + 1), 'record')." and fieldnum
191                     = 17 and sentence = ".pg_fetch_result($id, ($i + 1), '
192                     sentence')." and wordstart = ".pg_fetch_result($id, ($i
193                     + 1), 'wordstart')." and wordend = ".pg_fetch_result($id
194                     , ($i + 1), 'wordend')." and word = '".pg_fetch_result(
195                     $id, ($i + 1), 'word')."
196                     ";");
197             } else {
198                 return false;
199             }
200         }
201     }
```



```
177     }
178   }
179 }
180 return true;
181 }
182
183 function dataControl3() {
184   include_once '/home/yoshiga/analysis/function.php';
185   include_once '/var/www/html/tsr/dbsetting.php';
186
187   //検出した人名をチェックする
188   $id = sql("select * from middledata where type = 'Person' and
            fieldnum = 17 order by record, sentence, wordstart, wordend
            ;");
189
190   //文字の端が重複
191   $count = 0;
192
193   for ($i = 0; $i < pg_num_rows($id); $i++) {
194     $str = "";
195     $strlen = 0;
196
197     if (pg_fetch_result($id, $i, 'record') == @pg_fetch_result($id,
            ($i + 1), 'record') &&
198         pg_fetch_result($id, $i, 'sentence') == @pg_fetch_result($id, (
            $i + 1), 'sentence')) {
199       if (pg_fetch_result($id, $i, 'wordend') == pg_fetch_result(
            $id, ($i + 1), 'wordstart')) {
200         //形態素解析基本データ呼出
201         $ida = sql("select * from \"data-all\"
202                   where record = ".pg_fetch_result($id, $i, 'record')." and
203                   sentence = ".pg_fetch_result($id, $i, 'sentence')." and
204                   wordstart = ".pg_fetch_result($id, $i, 'wordend')."
205                   order by record, sentence, wordstart;");
206
207         $strlen = mb_strlen(pg_fetch_result($ida, 0, 'word'));
208
209         $str = pg_fetch_result($id, $i, 'word').mb_substr(
            pg_fetch_result($id, ($i + 1), 'word'), $strlen);
210         $count++;
211
212       } elseif (pg_fetch_result($id, $i, 'wordend') >
            pg_fetch_result($id, ($i + 1), 'wordstart')) {
213         //n 単語重複
214         $n = pg_fetch_result($id, $i, 'wordend') - pg_fetch_result(
            $id, ($i + 1), 'wordstart');
215
```

```
216 //形態素解析基本データ呼出
217 $ida = sql("select * from \"data-all\"
218 where record = ".pg_fetch_result($id, $i, 'record')." and
219 sentence = ".pg_fetch_result($id, $i, 'sentence')." and
220 wordstart >= ".(pg_fetch_result($id, $i, 'wordend') - $n)."
      and
221 wordend <= ".pg_fetch_result($id, $i, 'wordend')."
222 order by record, sentence, wordstart;");
223
224 for ($j = 0; $j < pg_num_rows($ida); $j++) {
225     $strlen += mb_strlen(pg_fetch_result($ida, $j, 'word'));
226 }
227
228 $str = pg_fetch_result($id, $i, 'word').mb_substr(
      pg_fetch_result($id, ($i + 1), 'word'), $strlen);
229 $count++;
230
231 } else {
232
233 }
234
235 #print $str."\n\n";
236
237 //スクリプトE (URI 付与)
238 $uri = "http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki
      /".str_pad(pg_fetch_result($id, $i, 'record'), 3, 0,
      STR_PAD_LEFT)."-".pg_fetch_result($id, $i, 'fieldnum
      ')."-".pg_fetch_result($id, $i, 'sentence')."-".
      pg_fetch_result($id, $i, 'wordstart')."-".pg_fetch_result(
      $id, ($i + 1), 'wordend');
239
240 if ($str != "") {
241     $idins = sql("insert into middledata values (
242     'Person',
243     ".pg_fetch_result($id, $i, 'record').",
244     ".pg_fetch_result($id, $i, 'fieldnum').",
245     ".pg_fetch_result($id, $i, 'sentence').",
246     ".pg_fetch_result($id, $i, 'wordstart').",
247     ".pg_fetch_result($id, ($i + 1), 'wordend').",
248     '$str.',
249     '$uri.'
250     );");
251 }
252 }
253 }
254
255 //重複データを除去
```

```
256     $iddel3 = sql("delete from middledata as m1 where exists (  
257     select * from middledata as m2 where  
258     m1.type = 'Person' and  
259     m1.record = m2.record and  
260     m1.fieldnum = 17 and  
261     m1.fieldnum = m2.fieldnum and  
262     m1.sentence = m2.sentence and  
263     m1.wordstart = m2.wordstart and  
264     m1.wordend = m2.wordend and  
265     m1.word = m2.word and  
266     m1.oid > m2.oid  
267     );");  
268  
269     return $count;  
270 }  
271  
272 function dataControl3Date() {  
273     include_once '/home/yoshiga/analysis/function.php';  
274     include_once '/var/www/html/tsr/dbsetting.php';  
275  
276     //検出した人名をチェックする  
277     $sid = sql("select * from middledata where type = 'Date' and  
                fieldnum = 17 order by record, sentence, wordstart, wordend  
                ;");  
278  
279     //文字の端が重複  
280     $count = 0;  
281  
282     for ($i = 0; $i < pg_num_rows($sid); $i++) {  
283         $str = "";  
284         $strlen = 0;  
285  
286         if (pg_fetch_result($sid, $i, 'record') == @pg_fetch_result($sid,  
                ($i + 1), 'record') &&  
287         pg_fetch_result($sid, $i, 'sentence') == @pg_fetch_result($sid, (  
                $i + 1), 'sentence')) {  
288             if (pg_fetch_result($sid, $i, 'wordend') == pg_fetch_result(  
                $sid, ($i + 1), 'wordstart')) {  
289                 //形態素解析基本データ呼出  
290                 $ida = sql("select * from \"data-all\  
291                 where record = ".pg_fetch_result($sid, $i, 'record')." and  
292                 sentence = ".pg_fetch_result($sid, $i, 'sentence')." and  
293                 wordstart = ".pg_fetch_result($sid, $i, 'wordend')."  
294                 order by record, sentence, wordstart;");  
295  
296                 $strlen = mb_strlen(pg_fetch_result($ida, 0, 'word'));  
297
```

```
298     $str = pg_fetch_result($id, $i, 'word').mb_substr(  
        pg_fetch_result($id, ($i + 1), 'word'), $strlen);  
299     $count++;  
300  
301 } elseif (pg_fetch_result($id, $i, 'wordend') >  
        pg_fetch_result($id, ($i + 1), 'wordstart')) {  
302     //n 単語重複  
303     $n = pg_fetch_result($id, $i, 'wordend') - pg_fetch_result(  
        $id, ($i + 1), 'wordstart');  
304  
305     //形態素解析基本データ呼出  
306     $ida = sql("select * from \"data-all\"  
307     where record = ".pg_fetch_result($id, $i, 'record')." and  
308     sentence = ".pg_fetch_result($id, $i, 'sentence')." and  
309     wordstart >= ".(pg_fetch_result($id, $i, 'wordend') - $n)."  
        and  
310     wordend <= ".pg_fetch_result($id, $i, 'wordend')." "  
311     order by record, sentence, wordstart;");  
312  
313     for ($j = 0; $j < pg_num_rows($ida); $j++) {  
314         $strlen += mb_strlen(pg_fetch_result($ida, $j, 'word'));  
315     }  
316  
317     $str = pg_fetch_result($id, $i, 'word').mb_substr(  
        pg_fetch_result($id, ($i + 1), 'word'), $strlen);  
318     $count++;  
319  
320 } else {  
321  
322 }  
323  
324 #print $str."\n\n";  
325  
326 //スクリプトE (URI 付与)  
327 $uri = "http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki  
        /".str_pad(pg_fetch_result($id, $i, 'record'), 3, 0,  
        STR_PAD_LEFT)."-".pg_fetch_result($id, $i, 'fieldnum  
        ')."-".pg_fetch_result($id, $i, 'sentence')."-".  
        pg_fetch_result($id, $i, 'wordstart')."-".pg_fetch_result(  
        $id, ($i + 1), 'wordend');  
328  
329 if ($str != "") {  
330     $idins = sql("insert into middledata values (  
331     'Date',  
332     ".pg_fetch_result($id, $i, 'record').",  
333     ".pg_fetch_result($id, $i, 'fieldnum').",  
334     ".pg_fetch_result($id, $i, 'sentence').",
```

```
335     ".pg_fetch_result($id, $i, 'wordstart').",
336     ".pg_fetch_result($id, ($i + 1), 'wordend').",
337     ".$str.",
338     ".$uri."
339 );");
340 }
341
342 if ($idins) {
343     $idDelData = sql("delete from middledata where
344     type = 'Date' and
345     record = ".pg_fetch_result($id, $i, 'record')." and
346     fieldnum = ".pg_fetch_result($id, $i, 'fieldnum')." and
347     sentence = ".pg_fetch_result($id, $i, 'sentence')." and
348     wordstart = ".pg_fetch_result($id, $i, 'wordstart')." and
349     wordend = ".pg_fetch_result($id, $i, 'wordend').";");
350
351     $idDelData = sql("delete from middledata where
352     type = 'Date' and
353     record = ".pg_fetch_result($id, $i, 'record')." and
354     fieldnum = ".pg_fetch_result($id, $i, 'fieldnum')." and
355     sentence = ".pg_fetch_result($id, $i, 'sentence')." and
356     wordstart = ".pg_fetch_result($id, ($i + 1), 'wordstart')."
357         and
358     wordend = ".pg_fetch_result($id, ($i + 1), 'wordend').";");
359 }
360 }
361
362 //重複データを除去
363 $iddel3 = sql("delete from middledata as m1 where exists (
364     select * from middledata as m2 where
365     m1.type = 'Date' and
366     m1.record = m2.record and
367     m1.fieldnum = 17 and
368     m1.fieldnum = m2.fieldnum and
369     m1.sentence = m2.sentence and
370     m1.wordstart = m2.wordstart and
371     m1.wordend = m2.wordend and
372     m1.word = m2.word and
373     m1.oid > m2.oid
374 );");
375
376     return $count;
377 }
378
379 function dataControlPlace() { //地名
380     include_once '/home/yoshiga/analysisKokubun/function.php';
```

```
381     include_once '/var/www/html/tsr/dbsetting.php';
382     $delarr = array();
383
384     $sid = sql("select * from kokubunindex where type = 'Place' and
                fieldnum = 17 order by record, sentence, wordstart, wordend
                ;");
385
386     $place = file("/home/yoshiga/analysisKokubun/words/kokubun-place.
                csv", FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
387
388     for ($i = 0; $i < pg_num_rows($sid); $i++) {
389         if (array_search(pg_fetch_result($sid, $i, 'word'), $place) ===
                false) {
390             $delarr[] = @pg_fetch_result($sid, $i, 'record').".".
                @pg_fetch_result($sid, $i, 'sentence').".".@pg_fetch_result
                ($sid, $i, 'wordstart').".".@pg_fetch_result($sid, $i, '
                wordend').".".@pg_fetch_result($sid, $i, 'word');
391         }
392     }
393     print_r($delarr);
394
395     if (isset($delarr)) {
396         foreach ($delarr as $val) {
397             $value = explode(",", $val);
398             if (!$iddel = sql("delete from kokubunindex where type = '
                Place' and fieldnum = 17 and record = ".$value[0]." and
                sentence = ".$value[1]." and wordstart = ".$value[2]." and
                wordend = ".$value[3]." and word = '".$value[4]."'");)) {
399                 die("dataControlPlace has trouble.");
400             }
401         }
402     }
403     return count($delarr);
404 }
405
406 //データベース処理連結
407 $c = 0;
408 $c2 = 0;
409
410 do {
411     if (dataControl0()) {
412         print "0-done!\n";
413     }
414
415     do {
416         $c = dataControl1();
417     } while ($c > 0);
```

```

418     print "1-done!\n";
419
420     if (dataControl2()) {
421         print "2-done!\n";
422     }
423
424     if (dataControl2Date()) {
425         print "2Date-done!\n";
426     }
427
428     $c2 = dataControl3();
429     print "(.$c2.) 3-done!\n";
430
431     $c2Date = dataControl3Date();
432     print "(.$c2Date.) 3Date-done!\n";
433
434     dataControlPlace();
435
436 } while($c2 > 0);
437
438 //人名で1文字と判定しているところを除去
439 #$_idup_ichiba = sql("update middledata set type = 'UNK' where type
        = 'Person' and length(word) = 1;");
440
441 //
        ichiba 限定処理：「収」（収ム）がMecab で処理できないためデータベース上で消す
442 $_idup_ichiba2 = sql("update middledata set type = 'UNK' where type
        = 'Person' and (word = '収' or word = 'ヲ収');");
443 $_iddel_ichiba = sql("delete from middledata where type = 'Person'
        and word = '息子部屋;");
444 ?>

```

ソースコード B.21: スクリプト F（外部リンク探索および典拠データベース登録, findHeading.php).

```

1 <?php
2 //典拠データ探索モジュール群
3 //固有表現クラス別に外部の典拠データベースを選んで外部リンク候補を取得.
4
5 //分類名
6 function findGenreNDLA ($tango) {
7     $filteredVal = "";

```

```
8   $array = array();
9
10  $sparqlPre="
11  PREFIX dc:<http://purl.org/dc/elements/1.1/>
12  PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
13  PREFIX foaf:<http://xmlns.com/foaf/0.1/>
14  PREFIX dcterms:<http://purl.org/dc/terms/>
15  PREFIX owl:<http://www.w3.org/2002/07/owl#>
16  PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
17  PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
18  PREFIX skos:<http://www.w3.org/2004/02/skos/core#>
19  PREFIX rda:<http://RDVocab.info/ElementsGr2/>
20  PREFIX xl:<http://www.w3.org/2008/05/skos-xl#>
21  PREFIX dcndl:<http://ndl.go.jp/dcndl/terms/>
22  ";
23
24  $sparqlc = "
25  SELECT DISTINCT ?uri1 ?name
26  ";
27
28  $sparqlc2 = "
29  WHERE {
30  ?uri1 skos:inScheme <http://id.ndl.go.jp/auth#topicalTerms> .
31  { { ?uri1 xl:prefLabel [ xl:literalForm ?name ]. }
32  UNION
33  { ?uri1 xl:altLabel [ xl:literalForm ?name ]. }
34  UNION
35  { ?uri1 rdfs:label ?name .}
36  }
37  filter regex(?name, '^".$tango."$');
38  }";
39
40  $url = 'http://id.ndl.go.jp/auth/ndla';
41  $sparql = $sparqlPre.$sparqlc.$sparqlc2;
42  $sparql = str_replace(array("\r\n", "\r", "\n"), '', $sparql);
43  $param = $url.'?query='.urlencode($sparql).'&output=json';
44  #print $param."\n";
45
46  $array = json_decode(file_get_contents($param), true);
47  #print_r($array);
48
49  for ($j=0; $j<count($array['results']['bindings']); $j++) {
50  $headings = $array['results']['bindings'][$j]['name']['value'];
51  $uri = $array['results']['bindings'][$j]['uri1']['value'];
52  $checked = 'ndla';
53  $filteredVal = $tango." ".$uri." ".$headings." ".$checked;
54  }
```



```
55     return $filteredVal;
56 }
57
58 function findGenreDBPedia ($tango) {
59     $filteredVal = "";
60     $array = array();
61
62     $sparqlPre="
63     PREFIX dc:<http://purl.org/dc/elements/1.1/>
64     PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
65     PREFIX foaf:<http://xmlns.com/foaf/0.1/>
66     PREFIX dcterms:<http://purl.org/dc/terms/>
67     PREFIX owl:<http://www.w3.org/2002/07/owl#>
68     PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
69     PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
70     PREFIX skos:<http://www.w3.org/2004/02/skos/core#>
71     PREFIX rda:<http://RDVocab.info/ElementsGr2/>
72     PREFIX xl:<http://www.w3.org/2008/05/skos-xl#>
73     PREFIX dcndl:<http://ndl.go.jp/dcndl/terms/>
74     ";
75
76     //DBPedia Japanese 探索
77     $sparql_cmd = "select DISTINCT * ";
78
79     $sparql_cmd2 = "
80     WHERE {
81     <http://ja.dbpedia.org/resource/\".$tango.> owl:sameAs ?sameUri .
82     }";
83
84     $sparql = $sparqlPre.$sparql_cmd.$sparql_cmd2;
85     $sparql = str_replace(array("\r\n", "\r", "\n"), '', $sparql);
86
87     $url='http://ja.dbpedia.org/sparql';
88     $param = $url.'?query='.urlencode($sparql).'&output=json';
89     #print $param."\n";
90
91     $array = json_decode(file_get_contents($param), true);
92     #print_r($array);
93
94     $headings = '';
95     $uriPre = array();
96     for ($j=0; $j<count($array['results']['bindings']); $j++) {
97         if ($array['results']['bindings'][$j]['sameUri']['value'] != "")
98             {
99                 $uriPre[] = $array['results']['bindings'][$j]['sameUri']['value
100                 '];
101             }
102     }
```

```
100 }
101
102 foreach ($uriPre as $uri) {
103     $checked = 'dbpediaJ';
104     if (strpos($uri, "/dbpedia.org") !== false || strpos($uri, "
        wikidata") !== false) {
105         $filteredVal[] = $tango.".$uri."."$tango."."$checked;
106     }
107 }
108
109 print_r($filteredVal);
110
111 return $filteredVal;
112 }
113 /*
114 findGenreDBPedia ("黄表紙");
115 exit;
116 */
117
118 //人名
119 function findPerson () {
120     include_once '/var/www/html/tsr/dbsetting.php';
121
122     $sparqlPre="
123     PREFIX dc:<http://purl.org/dc/elements/1.1/>
124     PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
125     PREFIX foaf:<http://xmlns.com/foaf/0.1/>
126     PREFIX dcterms:<http://purl.org/dc/terms/>
127     PREFIX owl:<http://www.w3.org/2002/07/owl#>
128     PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
129     PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
130     PREFIX skos:<http://www.w3.org/2004/02/skos/core#>
131     PREFIX rda:<http://RDVocab.info/ElementsGr2/>
132     PREFIX xl:<http://www.w3.org/2008/05/skos-xl#>
133     PREFIX dcndl:<http://ndl.go.jp/dcndl/terms/>
134     ";
135
136     $idPerson = sql("select * from correctiondata where type = 'Person
        ' order by word");
137
138     for ($i=0; $i<pg_num_rows($idPerson); $i++) {
139         //NDLA 探索
140         $tango = trim(pg_fetch_result($idPerson, $i, 'word'));
141         #$tango = '蓬莱山人';
142         #print $tango."\n";
143
144         $sparql_cmd = "
```

```
145     SELECT DISTINCT ?uri1 ?name
146     ";
147
148     $sparql_cmd2 = "
149     WHERE {
150     ?uri1 foaf:primaryTopic ?uri2 .
151     ?uri2 rdf:type foaf:Person .
152     { { ?uri1 xl:prefLabel [ xl:literalForm ?name ]. }
153     UNION
154     { ?uri1 xl:altLabel [ xl:literalForm ?name ]. }
155     UNION
156     { ?uri2 foaf:name ?name .}
157     }
158     filter regex(?name, ".$tango."").
159     }";
160
161     $url = 'http://id.ndl.go.jp/auth/ndla';
162     $sparql = $sparqlPre.$sparql_cmd.$sparql_cmd2;
163     $sparql = str_replace(array("\r\n", "\r", "\n"), '', $sparql);
164     $param = $url.'?query='.urlencode($sparql).'&output=json';
165     #print $param."\n";
166
167     $array = json_decode(file_get_contents($param), true);
168
169     for ($j=0; $j<count($array['results']['bindings']); $j++) {
170     $headings = $array['results']['bindings'][$j]['name']['value'];
171     $uri = $array['results']['bindings'][$j]['uri1']['value'];
172     $saltword = "";
173     $checked = 'ndla';
174
175     //DB用データ入力
176     $idIns = sql("
177     insert into headings values (
178     ".$tango.",
179     ".$uri.",
180     ".$headings.",
181     ".$saltword.",
182     ".$checked.",
183     'Person',
184     ''
185     );");
186     }
187 }
188 }
189
190 #findPerson();
191
```

```
192 //時
193 function findDate () {
194     include_once '/var/www/html/tsr/dbsetting.php';
195     include_once '/home/yoshiga/analysis/kan2num.php';
196     $idDate = sql("select * from correctiondata where type = 'Date'
197         order by word");
198
199     for ($i=0; $i<pg_num_rows($idDate); $i++) {
200         //hutime 探索
201         $tangoVariation = 0;
202         $tangoPre = trim(pg_fetch_result($idDate, $i, 'word'));
203
204         //時節用語をとりあえず省略する
205         $datePattern = "/([0-9年月]+)$ /u";
206         $tango = preg_replace($datePattern, '', $tangoPre);
207
208         if (@mb_strpos($tango, "年") !== false) {
209             if (@mb_strpos($tango, "月") !== false) {
210                 $tangoVariation = 1;
211             } else {
212                 $tangoVariation = 2;
213             }
214         } else {
215             $tangoVariation = 0;
216         }
217
218         //単語バリエーションによる場合分け
219         Switch ($tangoVariation) {
220             case 1: //年月
221                 $uri = 'http://datetime.hutime.org/calendar/1001.1/month/'.
222                     $tango;
223                 break;
224             case 2: //年
225                 $uri = 'http://datetime.hutime.org/calendar/1001.1/year/'.
226                     $tango;
227                 break;
228             default:
229                 $uri = '';
230         }
231
232         #print $tangoPre." : ".$uri."\n";
233
234         //DB用データ入力
235         $headings = $tango;
236         $altword = "";
237         $checked = 'hutime';
238     }
239 }
```

```
236     $idIns = sql("
237     insert into headings values (
238         '$tangoPre.',
239         '$uri.',
240         '$headings.',
241         '$saltword.',
242         '$checked.',
243         'Date',
244         ''
245     );");
246 }
247 }
248 #findDate();
249
250 //場所
251 function findPlace() {
252     include_once '/var/www/html/tsr/dbsetting.php';
253     $sparqlPre="
254     PREFIX dc:<http://purl.org/dc/elements/1.1/>
255     PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
256     PREFIX foaf:<http://xmlns.com/foaf/0.1/>
257     PREFIX dcterms:<http://purl.org/dc/terms/>
258     PREFIX owl:<http://www.w3.org/2002/07/owl#>
259     PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
260     PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
261     PREFIX skos:<http://www.w3.org/2004/02/skos/core#>
262     PREFIX rda:<http://RDVocab.info/ElementsGr2/>
263     PREFIX xl:<http://www.w3.org/2008/05/skos-xl#>
264     PREFIX dcndl:<http://ndl.go.jp/dcndl/terms/>
265     ";
266
267     $idPlace = sql("select * from correctiondata where type = 'Place'
268         order by word");
269
270     for ($i=0; $i<pg_num_rows($idPlace); $i++) {
271         //DBPedia Japanese 探索
272         $tango = trim(pg_fetch_result($idPlace, $i, 'word'));
273         #print $tango."\n";
274
275         $sparql_cmd = "select DISTINCT * ";
276
277         $sparql_cmd2 = "
278         WHERE {
279         <http://ja.dbpedia.org/resource/" . $tango . "> owl:sameAs ?sameUri .
280         }";
281
282         $sparql = $sparqlPre.$sparql_cmd.$sparql_cmd2;
```

```
282     $sparql = str_replace(array("\r\n", "\r", "\n"), '', $sparql);
283
284     $url='http://ja.dbpedia.org/sparql';
285     $param = $url.'?query='.urlencode($sparql).'&output=json';
286     #print $param."\n";
287
288     $array = json_decode(file_get_contents($param), true);
289     #print_r($array);
290
291     $uri = '';
292     for ($j=0; $j<count($array['results']['bindings']); $j++) {
293         $uriPre = $array['results']['bindings'][$j]['sameUri']['value
294             '];
295
296         if (strpos($uriPre, "http://www.wikidata.org/entity/") !==
297             false) {
298             $uri = $uriPre;
299         }
300     }
301
302     $arrayUri[] = $tango."".$uri;
303 }
304
305 $arrayUri2 = array_unique($arrayUri);
306
307 foreach ($arrayUri2 as $value) {
308     $token = explode(",", $value);
309     $tango = $token[0];
310     $uri = $token[1];
311     $headings = $tango;
312     $saltword = "";
313     $checked = 'dbpediaJ';
314
315     //DB用データ入力
316     $idIns = sql("
317         insert into headings values (
318             '".$tango."',
319             '".$uri."',
320             '".$headings."',
321             '".$saltword."',
322             '".$checked."',
323             'Place',
324             ''
325         );");
326 }
```

```
327 #findPlace();
328
329 //部品, ロール, 専門用語の探索
330 function findParts() {
331     include_once '/home/yoshiga/analysis/function.php';
332     include_once '/var/www/html/tsr/dbsetting.php';
333
334     $sparqlPre="
335     PREFIX dc:<http://purl.org/dc/elements/1.1/>
336     PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
337     PREFIX foaf:<http://xmlns.com/foaf/0.1/>
338     PREFIX dcterms:<http://purl.org/dc/terms/>
339     PREFIX owl:<http://www.w3.org/2002/07/owl#>
340     PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
341     PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
342     PREFIX skos:<http://www.w3.org/2004/02/skos/core#>
343     PREFIX rda:<http://RDVocab.info/ElementsGr2/>
344     PREFIX xl:<http://www.w3.org/2008/05/skos-xl#>
345     PREFIX dcndl:<http://ndl.go.jp/dcndl/terms/>
346     ";
347
348     $jwordsTob = file('/home/yoshiga/analysis/words/tob.csv',
349         FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
350     $jwordsRole = file('/home/yoshiga/analysis/words/role.csv',
351         FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
352     $jwordsTerm = file('/home/yoshiga/analysis/words/terms.csv',
353         FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
354
355     //TOB=Terminology of Bookbinding, TFB, Terminology for Bookbinding
356     , 製本用語
357     $typeArray = array("TOB"=>$jwordsTob, "Role"=>$jwordsRole, "Term
358         "=>$jwordsTerm);
359
360     foreach($typeArray as $typeKey => $typeVal) {
361
362         $ontNodeProperty1 = array();
363         $ontkey = $ontkeyPart = "";
364         $node = $property = "";
365
366         foreach($typeVal as $value) {
367             $token = array();
368             $ontNodeProperty = array();
369             $tango = "";
370             $uri = "";
371             $headings = "";
372             $saltword = "";
373             $use = "";
```

```

369     $node = "";
370
371     $token = @explode(",", $value);
372
373     $tango = $token[0];
374     #print $tango."\n";
375
376     //オントロジー表現 $token[1]
377     if (@$token[1] != "") {
378         preg_match_all('/[=]([^\C\\(\)]+)/u', $token[1], $match1);
379         preg_match_all('/[C][\(\]?([^\C\\(\)]+)[\)]?/u', $token[1],
380             $match2);
381
382         #print_r($match1);
383
384         //owl:sameAs
385         $ontNodeProperty = @explode(":", $match1[1][0]);
386         $node = $ontNodeProperty[0]; //序
387         #$property = @$ontNodeProperty[1]; //題
388     }
389
390     if ($node == "") {
391         $headings = $tango;
392     } else {
393         $headings = $node;
394     }
395
396     /*
397     print $headings."\n";
398     print "node: ".$headings."\n";
399     print "type: ".$typeKey."\n";
400     */
401
402     //URI 探索
403     //dbpedia Japanese を調べる
404     //dbpedia 本家を調べる
405     $filteredValueDbpedia は dbpedia 日英を合わせたデータ
406     $enWord = @str_replace(" ", "_", trim($token[2]));
407     #print ucfirst($enWord)."\n";
408
409     $sparql_cmd = "select DISTINCT * ";
410
411     $sparql_cmd2 = "
412     WHERE {
413     <http://ja.dbpedia.org/resource/" . $headings . "> owl:sameAs ?
414         sameUri .
415     }";

```



```
413     $sparql_cmd3 = "  
414     WHERE {  
415     <http://dbpedia.org/resource/".ucfirst($enWord)."> owl:sameAs ?  
         sameUri .  
416     }";  
417     #print $tango[2]."\n";  
418  
419     //AAT を探索  
420     print "AAT: ".$enWord."\n";  
421     if ($enWord != "") {  
422         $sparqlAAT = "http://vocab.getty.edu/sparql.xml?query=select+  
             distinct+*+where+%7B%0D%0A++%3Fs+rdf%3Atype+skos%3AConcept  
             +.%0D%0A++%3Fs+rdfs%3Alabel+%3Fo+.%0D%0A++filter+regex  
             %28%3Fo%2C+%22".$enWord."%22%29+.%0D%0A%7D&_implicit=false  
             &implicit=true&_equivalent=false&equivalent=true&_form=%2  
             Fsparql";  
423  
424         //AAT にはサーバから直接 sparql できない  
425         $ch = curl_init();  
426         curl_setopt($ch,CURLOPT_URL,$sparqlAAT);  
427         curl_setopt($ch,CURLOPT_HEADER,FALSE);  
428         curl_setopt($ch,CURLOPT_RETURNTRANSFER,TRUE);  
429         curl_setopt($ch,CURLOPT_FOLLOWLOCATION,TRUE);  
430  
431         //ブラウザWindows IE11 としてアクセス  
432         curl_setopt($ch,CURLOPT_USERAGENT,'Mozilla/5.0 (Windows NT  
             6.3; WOW64; Trident/7.0; Touch; rv:11.0) like Gecko');  
433         $xmlAAT = curl_exec($ch);  
434         curl_close($ch);  
435  
436         $arrayAAT = simplexml_load_string($xmlAAT);  
437         #print_r($arrayAAT);  
438         foreach($arrayAAT->results as $arrValAAT) {  
439             foreach($arrValAAT->result as $resAAT) {  
440                 if ($resAAT->binding->uri != "") {  
441                     $filteredValueDbpedia[] = $tango." ".$resAAT->binding->  
                         uri." ".$headings." "AAT"." ".$typeKey;  
442                 }  
443             }  
444         }  
445     }  
446  
447     //NDLA を探索  
448     $sparqlNDLAc = "  
449     SELECT DISTINCT ?uri1 ?name  
450     ";  
451
```

```
452     $sparqlNDLAc2 = "  
453     WHERE {  
454     ?uri1 skos:inScheme <http://id.ndl.go.jp/auth#topicalTerms> .  
455     { { ?uri1 xl:prefLabel [ xl:literalForm ?name ]. }  
456     UNION  
457     { ?uri1 xl:altLabel [ xl:literalForm ?name ]. }  
458     UNION  
459     { ?uri1 rdfs:label ?name .}  
460     }  
461     filter regex(?name, '". $tango. "').  
462     }";  
463  
464     $urlNDLA = 'http://id.ndl.go.jp/auth/ndla';  
465     $sparqlNDLA = $sparqlPre.$sparqlNDLAc.$sparqlNDLAc2;  
466     $sparqlNDLA = str_replace(array("\r\n", "\r", "\n"), '',  
467     $sparqlNDLA);  
468     $paramNDLA = $urlNDLA.'?query='.urlencode($sparqlNDLA).'&output  
469     =json';  
470     #print $param."\n";  
471  
472     $arrayNDLA = json_decode(file_get_contents($paramNDLA), true);  
473     #print_r($arrayNDLA);  
474  
475     for ($j=0; $j<count($arrayNDLA['results']['bindings']); $j++) {  
476     $headings = $arrayNDLA['results']['bindings'][$j]['name']['  
477     value'];  
478     $uri = $arrayNDLA['results']['bindings'][$j]['uri']['value  
479     '];  
480     $altword = "";  
481     $checked = 'ndla';  
482     $filteredValueDbpedia[] = $tango.", ".$uri.", ".$headings.", ".  
483     $checked.", ".$typeKey;  
484     }  
485  
486     //GND を探索  
487     }  
488 }  
489  
490 @print_r($filteredValueDbpedia);  
491 //headings テーブルに入力  
492 foreach($filteredValueDbpedia as $insVal) {  
493     $insToken = @explode(",", $insVal);  
494     #print_r($insToken);  
495     $idIns = sql("insert into headings values (  
496     '$insToken[0].',  
497     '$insToken[1].',  
498     '$insToken[2].',
```

```
494     '$saltword.',
495     '$insToken[3].',
496     '$insToken[4].',
497     '$use.'
498     );");
499 }
500 return $filteredValueDbpedia;
501 }
502 #findParts();
503
504 //WikidataID に接続している外部リンクを探索
505 function checkRelatedConcept($wikidataUriArray) {
506     include_once '/home/yoshiga/analysis/function.php';
507     include_once '/var/www/html/tsr/dbsetting.php';
508
509     $searchUri = array();
510
511     foreach($wikidataUriArray as $value) {
512         $token = explode(",", $value);
513         $id = "";
514         $aatNum = "";
515         $ndlNum = "";
516         $gndNum = "";
517
518         //wikidata の id から aat などの典拠データに繋げる
519         $id = trim($token[1], " \t\n\r\0\x0B\"'");
520         #print $id."\n";
521
522         //AAT
523         $queryAat = "https://query.wikidata.org/sparql?query=".urlencode
            ('select ?aat where { <' . $id . '> wdt:P1014 ?aat .}');
524         $aat = file_get_contents($queryAat);
525         $aatid = new SimpleXMLElement($aat);
526         #print_r($aatid);
527
528         $broaderUri = 'http://vocab.getty.edu/aat/300220751';
529         $topUri = 'http://vocab.getty.edu/aat/300000000';
530
531         foreach($aatid->results->result as $aatid2) {
532             foreach(@$aatid2->binding->literal as $aatNum) {
533                 #print_r($aatid2);
534                 $searchUri[] = $token[0].",".$id.","."http://vocab.getty.edu/
                    aat/" . $aatNum . "," . $token[2] . "," . "AAT" . "," . $token[4];
535             }
536         }
537
538         //NDLA
```

```
539 $queryNdl = "https://query.wikidata.org/sparql?query=".urlencode
      ('select ?ndl where { <' . $id . '> wdt:P349 ?ndl .}');
540 $ndl = file_get_contents($queryNdl);
541 $ndlid = new SimpleXMLElement($ndl);
542 #print_r($ndlid);
543 foreach($ndlid->results->result as $ndlid2) {
544     foreach(@$ndlid2->binding->literal as $ndlNum) {
545         #print_r($ndlid2);
546         $searchUri[] = $token[0] . " . " . "http://id.ndl.go.jp/auth/ndlna
              /" . $ndlNum . " . " . $token[2] . " . " . "NDLA" . " . " . $token[4];
547     }
548 }
549
550 //GND
551 $queryGnd = "https://query.wikidata.org/sparql?query=".urlencode
      ('select ?gnd where { <' . $id . '> wdt:P227 ?gnd .}');
552 $gnd = file_get_contents($queryGnd);
553 $gndid = new SimpleXMLElement($gnd);
554 #print_r($gndid);
555 foreach($gndid->results->result as $gndid2) {
556     foreach(@$gndid2->binding->literal as $gndNum) {
557         #print_r($gndid2);
558         $searchUri[] = $token[0] . " . " . $id . " . " . "http://d-nb.info/gnd/" .
              $gndNum . " . " . $token[2] . " . " . "GND" . " . " . $token[4];
559     }
560 }
561
562 //VIAF
563 $queryViaf = "https://query.wikidata.org/sparql?query=".urlencode
      ('select ?viaf where { <' . $id . '> wdt:P214 ?viaf .}');
564 $viaf = file_get_contents($queryViaf);
565 $viafid = new SimpleXMLElement($viaf);
566 #print_r($viafid);
567 foreach($viafid->results->result as $viafid2) {
568     foreach(@$viafid2->binding->literal as $viafNum) {
569         #print_r($viafid2);
570         $searchUri[] = $token[0] . " . " . $id . " . " . "https://viaf.org/viaf
              /" . $viafNum . " . " . $token[2] . " . " . "VIAF" . " . " . $token[4];
571     }
572 }
573
574 #print_r($searchUri);
575 }
576
577 sort($searchUri);
578 array_unique($searchUri);
579
```

```
580 //headings テーブルに入力
581 $saltword = "";
582 $use = "";
583
584 foreach($searchUri as $insVal) {
585     $insToken = @explode(",", $insVal);
586     #print_r($insToken);
587     $idIns = sql("insert into headings values (
588         '$insToken[0].'",
589         '$insToken[2].'",
590         '$insToken[3].'",
591         '$saltword."',
592         '$insToken[4].'",
593         '$insToken[5]."',
594         '$use."'
595     );");
596 }
597 return $searchUri;
598
599 }
600
601 //WikidataUriArray の抽出例
602 $WikidataUriArray = array
603 (
604     0 => "解題,http://www.wikidata.org/entity/Q11630951,解題,dbpediaJ
        ,TOB",
605     1 => "開題,http://www.wikidata.org/entity/Q11630951,解題,dbpediaJ
        ,TOB",
606     2 => "巻頭,http://www.wikidata.org/entity/Q411189,Opening,
        dbpediaE,TOB",
607     3 => "目首,http://www.wikidata.org/entity/Q1456936,
        Table_of_contents,dbpediaE,TOB",
608     4 => "目録,http://www.wikidata.org/entity/Q11581608,目録,dbpediaJ
        ,TOB",
609     5 => "目録,http://www.wikidata.org/entity/Q250233,Catalog,
        dbpediaE,TOB",
610     6 => "序,http://www.wikidata.org/entity/Q670787,Preface,dbpediaE,
        TOB",
611 );
612 #WikidataUriArray = findParts();
613 #checkRelatedConcept($WikidataUriArray);
614 ?>
```

ソースコード B.22: 基本オントロジー (手作業による作成, basicOntology.owl).

```

1 <?xml version="1.0"?>
2 <rdf:RDF xmlns="http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/
   tenseki#"
3   xml:base="http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki"
4   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
5   xmlns:owl="http://www.w3.org/2002/07/owl#"
6   xmlns:ichibaTenseki="http://www.dl.saga-u.ac.jp/OgiNabesima/
   ichiba/tenseki#"
7   xmlns:xml="http://www.w3.org/XML/1998/namespace"
8   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
9   xmlns:dcterms="http://purl.org/dc/terms/"
10  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
11  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
12  xmlns:dc="http://purl.org/dc/elements/1.1/">
13 <owl:Ontology rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/
   ichiba/tenseki#">
14   <owl:versionIRI rdf:resource="http://www.dl.saga-u.ac.jp/
   OgiNabesima/ichiba/tenseki#/2.0"/>
15   <rdfs:comment>Copyright 2017 Saga University Library
16 This data were normalized by Natsuko Yoshiga @ Saga University,
   Japan. </rdfs:comment>
17 </owl:Ontology>
18
19
20
21 <!--
22 ////////////////////////////////////////////////////////////////////
23 //
24 // Annotation properties
25 //
26 ////////////////////////////////////////////////////////////////////
27 -->
28
29
30
31
32 <!-- http://purl.org/dc/elements/1.1/title -->
33
34 <owl:AnnotationProperty rdf:about="http://purl.org/dc/elements
   /1.1/title"/>
35
36
37
38 <!-- http://purl.org/dc/terms/creator -->
39

```

```

40     <owl:AnnotationProperty rdf:about="http://purl.org/dc/terms/
41         creator"/>
42
43
44     <!-- http://purl.org/dc/terms/description -->
45
46     <owl:AnnotationProperty rdf:about="http://purl.org/dc/terms/
47         description"/>
48
49
50     <!-- http://purl.org/dc/terms/extent -->
51
52     <owl:AnnotationProperty rdf:about="http://purl.org/dc/terms/
53         extent"/>
54
55
56     <!-- http://purl.org/dc/terms/references -->
57
58     <owl:AnnotationProperty rdf:about="http://purl.org/dc/terms/
59         references"/>
60
61
62     <!-- http://purl.org/dc/terms/title -->
63
64     <owl:AnnotationProperty rdf:about="http://purl.org/dc/terms/title
65         "/>
66
67
68     <!-- http://www.w3.org/1999/02/22-rdf-syntax-ns#type -->
69
70     <owl:AnnotationProperty rdf:about="http://www.w3.org/1999/02/22-
71         rdf-syntax-ns#type"/>
72
73
74     <!-- http://www.w3.org/2002/07/owl#sameAs -->
75
76     <owl:AnnotationProperty rdf:about="http://www.w3.org/2002/07/owl#
77         sameAs"/>
78
79

```

```

80     <!-- http://www.w3.org/2004/02/skos/core#Concept -->
81
82     <owl:AnnotationProperty rdf:about="http://www.w3.org/2004/02/skos
83         /core#Concept"/>
84
85
86     <!-- http://www.w3.org/2004/02/skos/core#broader -->
87
88     <owl:AnnotationProperty rdf:about="http://www.w3.org/2004/02/skos
89         /core#broader"/>
90
91
92     <!-- http://www.w3.org/2004/02/skos/core#narrower -->
93
94     <owl:AnnotationProperty rdf:about="http://www.w3.org/2004/02/skos
95         /core#narrower"/>
96
97
98     <!-- http://www.w3.org/2004/02/skos/core#related -->
99
100    <owl:AnnotationProperty rdf:about="http://www.w3.org/2004/02/skos
101        /core#related"/>
102
103
104    <!--
105    //////////////////////////////////////
106    //
107    // Datatypes
108    //
109    //////////////////////////////////////
110    -->
111
112
113
114
115    <!-- http://www.w3.org/2001/XMLSchema#gYear -->
116
117    <rdfs:Datatype rdf:about="http://www.w3.org/2001/XMLSchema#gYear
118        "/>
119
120
121    <!--

```



```

122 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
123 //
124 // Object Properties
125 //
126 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
127 -->
128
129
130
131
132 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      hasGenre -->
133
134 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#hasGenre">
135   <rdfs:domain>
136     <owl:Class>
137       <owl:unionOf rdf:parseType="Collection">
138         <rdf:Description rdf:about="http://www.dl.saga-u.
              ac.jp/OgiNabesima/ichiba/tenseki#TOB"/>
139         <rdf:Description rdf:about="http://www.dl.saga-u.
              ac.jp/OgiNabesima/ichiba/tenseki#Tenseki"/>
140       </owl:unionOf>
141     </owl:Class>
142   </rdfs:domain>
143   <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Genre"/>
144 </owl:ObjectProperty>
145
146
147
148 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      hasInnerGenre -->
149
150 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#hasInnerGenre">
151   <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#hasGenre"/>
152   <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#TOB"/>
153   <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Genre"/>
154   <rdfs:comment xml:lang="ja">内題の分類</rdfs:comment>
155 </owl:ObjectProperty>
156
157
158

```

```

159 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      hasInnerTitle -->
160
161 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#hasInnerTitle">
162 <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#hasTitle"/>
163 <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Tenseki"/>
164 <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#TitleOfWork"/>
165 <owl:propertyDisjointWith rdf:resource="http://www.dl.saga-u.
      ac.jp/OgiNabesima/ichiba/tenseki#hasMainTitle"/>
166 <rdfs:comment xml:lang="ja">A hasInnerTitle
      B で古典籍 A は内題 B がある。</rdfs:comment>
167 </owl:ObjectProperty>
168
169
170
171 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      hasMainGenre -->
172
173 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#hasMainGenre">
174 <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#hasGenre"/>
175 <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Tenseki"/>
176 <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Genre"/>
177 <rdfs:comment xml:lang="ja">古典籍自体の分類</rdfs:comment>
178 </owl:ObjectProperty>
179
180
181
182 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      hasMainTitle -->
183
184 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#hasMainTitle">
185 <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#hasTitle"/>
186 <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Tenseki"/>
187 <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#MainTitle"/>
188 <rdfs:range>

```

```

189         <owl:Restriction>
190             <owl:onProperty rdf:resource="http://www.dl.saga-u.ac.
                jp/OgiNabesima/ichiba/tenseki#shomei"/>
191             <owl:cardinality rdf:datatype="http://www.w3.org/2001/
                XMLSchema#nonNegativeInteger">1</owl:cardinality>
192         </owl:Restriction>
193     </rdfs:range>
194 </rdfs:range>
195     <owl:Restriction>
196         <owl:onProperty rdf:resource="http://www.dl.saga-u.ac.
                jp/OgiNabesima/ichiba/tenseki#yomi"/>
197         <owl:cardinality rdf:datatype="http://www.w3.org/2001/
                XMLSchema#nonNegativeInteger">1</owl:cardinality>
198     </owl:Restriction>
199 </rdfs:range>
200 <rdfs:comment xml:lang="ja">A hasInnerTitle
    Bで古典籍 A は外題 B がある。</rdfs:comment>
201 </owl:ObjectProperty>
202
203
204
205 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
    hasMainURI -->
206
207 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#hasMainURI">
208 <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#hasURI"/>
209 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
    FunctionalProperty"/>
210 <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#Tenseki"/>
211 <owl:propertyDisjointWith rdf:resource="http://www.dl.saga-u.
    ac.jp/OgiNabesima/ichiba/tenseki#hasSubURI"/>
212 <rdfs:comment xml:lang="ja">A hasMainURI Bで古典籍 A は主 URI
    Bを持つ。</rdfs:comment>
213 </owl:ObjectProperty>
214
215
216
217 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
    hasPart -->
218
219 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#hasPart">
220 <owl:inverseOf rdf:resource="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#isPartOf"/>

```

```

221     <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      FunctionalProperty"/>
222     <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Tenseki"/>
223     <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#TOB"/>
224 </owl:ObjectProperty>
225
226
227
228 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      hasRole -->
229
230 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#hasRole">
231     <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Person"/>
232     <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Role"/>
233     <rdfs:comment xml:lang="ja">A hasRole B で A は B の役目を持つ。
      </rdfs:comment>
234 </owl:ObjectProperty>
235
236
237
238 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      hasSubURI -->
239
240 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#hasSubURI">
241     <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#hasURI"/>
242     <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      InverseFunctionalProperty"/>
243     <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Tenseki"/>
244     <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#TOB"/>
245     <rdfs:comment xml:lang="ja">A hasSubURI B で古典籍 A は副 URI
      B を持つ。 </rdfs:comment>
246 </owl:ObjectProperty>
247
248
249
250 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      hasTitle -->
251

```

```

252 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#hasTitle">
253 <rdfs:domain>
254 <owl:Class>
255 <owl:unionOf rdf:parseType="Collection">
256 <rdf:Description rdf:about="http://www.dl.saga-u.
    ac.jp/OgiNabesima/ichiba/tenseki#TOB"/>
257 <rdf:Description rdf:about="http://www.dl.saga-u.
    ac.jp/OgiNabesima/ichiba/tenseki#Tenseki"/>
258 </owl:unionOf>
259 </owl:Class>
260 </rdfs:domain>
261 <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#Title"/>
262 </owl:ObjectProperty>
263
264
265
266 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#hasURI
    -->
267
268 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#hasURI">
269 <rdfs:domain>
270 <owl:Class>
271 <owl:unionOf rdf:parseType="Collection">
272 <rdf:Description rdf:about="http://www.dl.saga-u.
    ac.jp/OgiNabesima/ichiba/tenseki#TOB"/>
273 <rdf:Description rdf:about="http://www.dl.saga-u.
    ac.jp/OgiNabesima/ichiba/tenseki#Tenseki"/>
274 </owl:unionOf>
275 </owl:Class>
276 </rdfs:domain>
277 </owl:ObjectProperty>
278
279
280
281 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
    isCollectionOf -->
282
283 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#isCollectionOf">
284 <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#Tenseki"/>
285 <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#Collection"/>
286 <rdfs:comment xml:lang="ja">A isCollectionOf B で作品(各古典籍

```

```

287     などの個体)A はコレクション(ichiba)B を持つ。 </rdfs:comment>
288 </owl:ObjectProperty>
289
290
291 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
292     isCreated -->
293
294 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
295     OgiNabesima/ichiba/tenseki#isCreated">
296     <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
297         OgiNabesima/ichiba/tenseki#TOB"/>
298     <rdfs:range>
299         <owl:Class>
300             <owl:unionOf rdf:parseType="Collection">
301                 <rdf:Description rdf:about="http://www.dl.saga-u.
302                     ac.jp/OgiNabesima/ichiba/tenseki#Date"/>
303                 <rdf:Description rdf:about="http://www.dl.saga-u.
304                     ac.jp/OgiNabesima/ichiba/tenseki#Person"/>
305                 <rdf:Description rdf:about="http://www.dl.saga-u.
306                     ac.jp/OgiNabesima/ichiba/tenseki#Place"/>
307             </owl:unionOf>
308         </owl:Class>
309     </rdfs:range>
310     <rdfs:comment xml:lang="ja">部品の作成日、作者、作成地</rdfs:
311         comment>
312 </owl:ObjectProperty>
313
314
315 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
316     isCreatedBy -->
317
318 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
319     OgiNabesima/ichiba/tenseki#isCreatedBy">
320     <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
321         OgiNabesima/ichiba/tenseki#isCreated"/>
322     <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
323         OgiNabesima/ichiba/tenseki#TOB"/>
324     <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
325         OgiNabesima/ichiba/tenseki#Person"/>
326     <rdfs:comment xml:lang="ja">A isCreatedBy B で部品 A(TOB)は
327         B によって作られた。 </rdfs:comment>
328 </owl:ObjectProperty>

```

```

320 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      isCreatedInDate -->
321
322 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#isCreatedInDate">
323 <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#isCreated"/>
324 <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#TOB"/>
325 <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Date"/>
326 <rdfs:comment xml:lang="ja">A isCreatedInDate B で部品 A(TOB)は
      B(日付)に作られた。</rdfs:comment>
327 </owl:ObjectProperty>
328
329
330
331 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      isCreatedInPlace -->
332
333 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#isCreatedInPlace">
334 <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#isCreated"/>
335 <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#TOB"/>
336 <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Place"/>
337 <rdfs:comment xml:lang="ja">A isCreatedInPlace B で部品 A(TOB)
      はB(場所)で作られた。</rdfs:comment>
338 </owl:ObjectProperty>
339
340
341
342 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      isPartOf -->
343
344 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#isPartOf">
345 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      InverseFunctionalProperty"/>
346 <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#TOB"/>
347 <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Tenseki"/>
348 </owl:ObjectProperty>
349

```

```

350
351
352 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      isPublished -->
353
354 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#isPublished">
355 <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Tenseki"/>
356 <rdfs:range>
357 <owl:Class>
358 <owl:unionOf rdf:parseType="Collection">
359 <rdf:Description rdf:about="http://www.dl.saga-u.
      ac.jp/OgiNabesima/ichiba/tenseki#Date"/>
360 <rdf:Description rdf:about="http://www.dl.saga-u.
      ac.jp/OgiNabesima/ichiba/tenseki#Person"/>
361 <rdf:Description rdf:about="http://www.dl.saga-u.
      ac.jp/OgiNabesima/ichiba/tenseki#Place"/>
362 </owl:unionOf>
363 </owl:Class>
364 </rdfs:range>
365 <rdfs:comment xml:lang="ja">古典籍本体の発行者、刊行日、刊行地</
      rdfs:comment>
366 </owl:ObjectProperty>
367
368
369
370 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      isPublishedBy -->
371
372 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#isPublishedBy">
373 <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#isPublished"/>
374 <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Tenseki"/>
375 <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Person"/>
376 <rdfs:comment xml:lang="ja">A isPublishedBy B で古典籍 A は B(人
      名)によって作られた。B は書肆名。 </rdfs:comment>
377 </owl:ObjectProperty>
378
379
380
381 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      isPublishedInDate -->
382

```



```
383 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/  
    OgiNabesima/ichiba/tenseki#isPublishedInDate">  
384 <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/  
    OgiNabesima/ichiba/tenseki#isPublished"/>  
385 <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/  
    OgiNabesima/ichiba/tenseki#Tenseki"/>  
386 <rdfs:range>  
387   <owl:Restriction>  
388     <owl:onProperty rdf:resource="http://www.dl.saga-u.ac.  
        jp/OgiNabesima/ichiba/tenseki#kankou_nen"/>  
389     <owl:qualifiedCardinality rdf:datatype="http://www.w3.  
        org/2001/XMLSchema#nonNegativeInteger">1</owl:  
        qualifiedCardinality>  
390     <owl:onDataRange rdf:resource="http://www.w3.org/2001/  
        XMLSchema#anyURI"/>  
391   </owl:Restriction>  
392 </rdfs:range>  
393 <rdfs:range>  
394   <owl:Restriction>  
395     <owl:onProperty rdf:resource="http://www.dl.saga-u.ac.  
        jp/OgiNabesima/ichiba/tenseki#seireki"/>  
396     <owl:qualifiedCardinality rdf:datatype="http://www.w3.  
        org/2001/XMLSchema#nonNegativeInteger">1</owl:  
        qualifiedCardinality>  
397     <owl:onDataRange rdf:resource="http://www.w3.org/2001/  
        XMLSchema#anyURI"/>  
398   </owl:Restriction>  
399 </rdfs:range>  
400 <rdfs:comment xml:lang="ja">A isPublishedInDate  
    Bで古典籍 A は日付 B に刊行された。</rdfs:comment>  
401 </owl:ObjectProperty>  
402  
403  
404  
405 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#  
    isPublishedInPlace -->  
406  
407 <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/  
    OgiNabesima/ichiba/tenseki#isPublishedInPlace">  
408 <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/  
    OgiNabesima/ichiba/tenseki#isPublished"/>  
409 <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/  
    OgiNabesima/ichiba/tenseki#Tenseki"/>  
410 <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/  
    OgiNabesima/ichiba/tenseki#Place"/>  
411 <rdfs:comment xml:lang="ja">A isPublishedInPlace  
    Bで古典籍 A は場所 B で刊行された。B は発行書肆に紐づく地名。</
```

```

412         rdfs:comment>
413     </owl:ObjectProperty>
414
415
416     <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
         useTechnicalTerms -->
417
418     <owl:ObjectProperty rdf:about="http://www.dl.saga-u.ac.jp/
         OgiNabesima/ichiba/tenseki#useTechnicalTerms">
419         <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
         OgiNabesima/ichiba/tenseki#Tenseki"/>
420         <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
         OgiNabesima/ichiba/tenseki#Role"/>
421         <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
         OgiNabesima/ichiba/tenseki#TOB"/>
422         <rdfs:range rdf:resource="http://www.dl.saga-u.ac.jp/
         OgiNabesima/ichiba/tenseki#Term"/>
423         <rdfs:comment xml:lang="ja">A useTechnicalTerms
         BでAはBの専門用語を使う。</rdfs:comment>
424     </owl:ObjectProperty>
425
426
427
428     <!--
429     //////////////////////////////////////
430     //
431     // Data properties
432     //
433     //////////////////////////////////////
434     -->
435
436
437
438
439     <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#bunko
         -->
440
441     <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
         OgiNabesima/ichiba/tenseki#bunko">
442         <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
         OgiNabesima/ichiba/tenseki#Collection"/>
443         <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
         schema#Literal"/>
444         <rdfs:comment xml:lang="ja">コレクション名</rdfs:comment>
445         <rdfs:label xml:lang="ja">文庫</rdfs:label>
446     </owl:DatatypeProperty>

```

```

447
448
449
450 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#bunrui
      -->
451
452 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#bunrui">
453   <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#genre"/>
454   <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Genre"/>
455   <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
      schema#Literal"/>
456   <rdfs:label xml:lang="ja">分類</rdfs:label>
457 </owl:DatatypeProperty>
458
459
460
461 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      bunrui1 -->
462
463 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#bunrui1">
464   <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#genre"/>
465   <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Genre"/>
466   <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
      schema#Literal"/>
467   <rdfs:label xml:lang="ja">分類 1</rdfs:label>
468 </owl:DatatypeProperty>
469
470
471
472 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      bunrui2 -->
473
474 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#bunrui2">
475   <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#genre"/>
476   <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Genre"/>
477   <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
      schema#Literal"/>
478   <rdfs:label xml:lang="ja">分類 2</rdfs:label>

```

```

479     </owl:DatatypeProperty>
480
481
482
483     <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
         bunrui3 -->
484
485     <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
         OgiNabesima/ichiba/tenseki#bunrui3">
486         <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
         OgiNabesima/ichiba/tenseki#genre"/>
487         <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
         OgiNabesima/ichiba/tenseki#Genre"/>
488         <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
         schema#Literal"/>
489         <rdfs:label xml:lang="ja">分類 3</rdfs:label>
490     </owl:DatatypeProperty>
491
492
493
494     <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
         bunrui4 -->
495
496     <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
         OgiNabesima/ichiba/tenseki#bunrui4">
497         <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
         OgiNabesima/ichiba/tenseki#genre"/>
498         <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
         OgiNabesima/ichiba/tenseki#Genre"/>
499         <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
         schema#Literal"/>
500         <rdfs:label xml:lang="ja">分類 4</rdfs:label>
501     </owl:DatatypeProperty>
502
503
504
505     <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
         bunrui5 -->
506
507     <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
         OgiNabesima/ichiba/tenseki#bunrui5">
508         <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
         OgiNabesima/ichiba/tenseki#genre"/>
509         <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
         OgiNabesima/ichiba/tenseki#Genre"/>
510         <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
         schema#Literal"/>

```

```
511     <rdfs:label xml:lang="ja">分類 5</rdfs:label>
512 </owl:DatatypeProperty>
513
514
515
516 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#chuki
517 -->
518 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
519   OgiNabesima/ichiba/tenseki#chuki">
520   <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
521     schema#Literal"/>
522   <rdfs:comment xml:lang="ja">項目に当てはまらない詳細な記載や付
523     記</rdfs:comment>
524   <rdfs:label xml:lang="ja">注記</rdfs:label>
525 </owl:DatatypeProperty>
526
527
528 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#date
529 -->
530 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
531   OgiNabesima/ichiba/tenseki#date">
532   <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
533     OgiNabesima/ichiba/tenseki#Date"/>
534   <rdfs:label xml:lang="ja">刊行年</rdfs:label>
535 </owl:DatatypeProperty>
536
537
538 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
539   gazoukougai -->
540 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
541   OgiNabesima/ichiba/tenseki#gazoukougai">
542   <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
543     schema#Literal"/>
544   <rdfs:comment xml:lang="ja">画像を公開許可</rdfs:comment>
545   <rdfs:label xml:lang="ja">画像公開</rdfs:label>
546 </owl:DatatypeProperty>
547
548 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#genre
549 -->
```

```

547 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#genre">
548 <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#Genre"/>
549 <rdfs:label xml:lang="ja">分類</rdfs:label>
550 </owl:DatatypeProperty>
551
552
553
554 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
    henchosha -->
555
556 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#henchosha">
557 <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#Person"/>
558 <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#Literal"/>
559 <rdfs:label xml:lang="ja">編著者</rdfs:label>
560 </owl:DatatypeProperty>
561
562
563
564 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#hoka
    -->
565
566 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#hoka">
567 <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#Literal"/>
568 <rdfs:label xml:lang="ja">他</rdfs:label>
569 </owl:DatatypeProperty>
570
571
572
573 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#inki
    -->
574
575 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#inki">
576 <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#Literal"/>
577 <rdfs:label xml:lang="ja">印記</rdfs:label>
578 </owl:DatatypeProperty>
579
580
581

```

```

582 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#juni
      -->
583
584 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#juni">
585   <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
      schema#Literal"/>
586   <rdfs:label xml:lang="ja">順位</rdfs:label>
587 </owl:DatatypeProperty>
588
589
590
591 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      kankou_nen -->
592
593 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#kankou_nen">
594   <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#date"/>
595   <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Date"/>
596   <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
      schema#Literal"/>
597   <rdfs:comment xml:lang="ja">年号で表現された日付</rdfs:comment>
598   <rdfs:label xml:lang="ja">刊行・書写年次</rdfs:label>
599 </owl:DatatypeProperty>
600
601
602
603 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      kansatsu -->
604
605 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#kansatsu">
606   <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
      schema#Literal"/>
607   <rdfs:comment xml:lang="ja">典籍の構成</rdfs:comment>
608   <rdfs:label xml:lang="ja">巻冊</rdfs:label>
609 </owl:DatatypeProperty>
610
611
612
613 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#kansha
      -->
614
615 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#kansha">

```

```

616     <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
        schema#Literal"/>
617     <rdfs:comment xml:lang="ja">印刷および書写の別</rdfs:comment>
618     <rdfs:label xml:lang="ja">刊写</rdfs:label>
619 </owl:DatatypeProperty>
620
621
622
623 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
        seireki -->
624
625 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
        OgiNabesima/ichiba/tenseki#seireki">
626     <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
        OgiNabesima/ichiba/tenseki#date"/>
627     <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
        OgiNabesima/ichiba/tenseki#Date"/>
628     <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#
        gYear"/>
629     <rdfs:comment xml:lang="ja">西暦で表現された日付</rdfs:comment>
630     <rdfs:label xml:lang="ja">西暦</rdfs:label>
631 </owl:DatatypeProperty>
632
633
634
635 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
        seiribangou -->
636
637 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
        OgiNabesima/ichiba/tenseki#seiribangou">
638     <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
        OgiNabesima/ichiba/tenseki#Tenseki"/>
639     <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
        schema#Literal"/>
640     <rdfs:comment xml:lang="ja">佐賀大学附属図書館の所蔵番号</rdfs:
        comment>
641     <rdfs:label xml:lang="ja">整理番号</rdfs:label>
642 </owl:DatatypeProperty>
643
644
645
646 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#shokei
        -->
647
648 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
        OgiNabesima/ichiba/tenseki#shokei">
649     <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-

```



```

        schema#Literal"/>
650     <rdfs:comment xml:lang="ja">典籍の大きさ</rdfs:comment>
651     <rdfs:label xml:lang="ja">書型</rdfs:label>
652 </owl:DatatypeProperty>
653
654
655
656 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#shomei
        -->
657
658 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
        OgiNabesima/ichiba/tenseki#shomei">
659     <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
        OgiNabesima/ichiba/tenseki#title"/>
660     <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
        OgiNabesima/ichiba/tenseki#Title"/>
661     <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
        schema#Literal"/>
662     <rdfs:comment xml:lang="ja">代表書名</rdfs:comment>
663     <rdfs:label xml:lang="ja">書名</rdfs:label>
664 </owl:DatatypeProperty>
665
666
667
668 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#title
        -->
669
670 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
        OgiNabesima/ichiba/tenseki#title">
671     <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
        OgiNabesima/ichiba/tenseki#Title"/>
672     <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
        schema#Literal"/>
673     <rdfs:label xml:lang="ja">題</rdfs:label>
674 </owl:DatatypeProperty>
675
676
677
678 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#yomi
        -->
679
680 <owl:DatatypeProperty rdf:about="http://www.dl.saga-u.ac.jp/
        OgiNabesima/ichiba/tenseki#yomi">
681     <rdfs:subPropertyOf rdf:resource="http://www.dl.saga-u.ac.jp/
        OgiNabesima/ichiba/tenseki#title"/>
682     <rdfs:domain rdf:resource="http://www.dl.saga-u.ac.jp/
        OgiNabesima/ichiba/tenseki#Title"/>

```

```

683     <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
        schema#Literal"/>
684     <rdfs:comment xml:lang="ja">書名の読み仮名</rdfs:comment>
685     <rdfs:label xml:lang="ja">読み</rdfs:label>
686 </owl:DatatypeProperty>
687
688
689
690 <!--
691 ////////////////////////////////////////////////////////////////////
692 //
693 // Classes
694 //
695 ////////////////////////////////////////////////////////////////////
696 -->
697
698
699
700
701 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#Book
        -->
702
703 <owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/
        ichiba/tenseki#Book">
704     <rdfs:label xml:lang="ja">本</rdfs:label>
705     <skos:Concept rdf:resource="http://id.ndl.go.jp/auth/ndlsh
        /00573378"/>
706     <skos:Concept rdf:resource="https://www.wikidata.org/wiki/
        Q571"/>
707 </owl:Class>
708
709
710
711 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
        Collection -->
712
713 <owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/
        ichiba/tenseki#Collection">
714     <rdfs:label xml:lang="ja">コレクション名</rdfs:label>
715 </owl:Class>
716
717
718
719 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#Date
        -->
720
721 <owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/

```

```

    ichiba/tenseki#Date">
722   <rdfs:label xml:lang="ja">時</rdfs:label>
723 </owl:Class>
724
725
726
727 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#Genre
    -->
728
729 <owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/
    ichiba/tenseki#Genre">
730   <rdfs:label xml:lang="ja">分類</rdfs:label>
731 </owl:Class>
732
733
734
735 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
    MainTitle -->
736
737 <owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/
    ichiba/tenseki#MainTitle">
738   <rdfs:subClassOf rdf:resource="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#Title"/>
739   <owl:disjointWith rdf:resource="http://www.dl.saga-u.ac.jp/
    OgiNabesima/ichiba/tenseki#TitleOfWork"/>
740   <dcterms:references rdf:resource="http://www.fumikura.net/
    other/biblio.html"/>
741   <rdfs:comment xml:lang="ja">書物の表紙に記載されている書名のこ
    。表題・標題・題目ともいう。直接表紙に記された「打付外題」と、題簽に
    書名が記された「貼外題」との二種類がある。のうち墨書された題簽を「
    書題簽」というが、一般こに板本の大部分には印刷された「摺題簽」が用
    いられている。題簽には短冊型と方型のものがあり、多くは外周に子
    持枠などの線が施される。漢籍の伝統を踏まえた書物は表紙の左肩に、
    物語の系統を継ぐものには中央に貼られることが多い。枠の中には書名
    や巻数などが摺られるが、二冊以上の場合には字体を楷書から行草書へ
    と次第に崩すことがある。凝った例としては芝居絵本や浄瑠璃本などが
    あり、中広の短冊簽に絵や板元名・座元名などを入れたりした。これに
    影響を受けた草双紙では絵入の大きな「絵題簽」が使われるようになる。
    また目録などを記した「副題簽」を加える場合もある。しかし題簽は時の
    経過とともに糊が利かなくなり剥がれやすくなるので、時代を上るほど
    残存しているものは少ない。とりわけ初期の草双紙などは題簽以外に何
    一つ書誌事項が記されていないので、題簽が残存しているものは資料価
    値も高い。ただし後から別本の題簽が貼られたものも存在するので、外
    題に加えられた改変には注意を要する。</rdfs:comment>
742   <rdfs:isDefinedBy xml:lang="ja">Copyright (C) 1999-2012 Gen
    TAKAGI</rdfs:isDefinedBy>
743   <rdfs:label xml:lang="en">title from cover</rdfs:label>
```

```
744     <rdfs:label xml:lang="ja">外題</rdfs:label>
745 </owl:Class>
746
747
748
749 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#Person
    -->
750
751 <owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/
    ichiba/tenseki#Person">
752     <rdfs:label xml:lang="ja">人名</rdfs:label>
753 </owl:Class>
754
755
756
757 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#Place
    -->
758
759 <owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/
    ichiba/tenseki#Place">
760     <rdfs:label xml:lang="ja">場所</rdfs:label>
761 </owl:Class>
762
763
764
765 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#Role
    -->
766
767 <owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/
    ichiba/tenseki#Role">
768     <rdfs:label xml:lang="ja">ロール</rdfs:label>
769 </owl:Class>
770
771
772
773 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#TOB
    -->
774
775 <owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/
    ichiba/tenseki#TOB">
776     <rdfs:label xml:lang="ja">部品</rdfs:label>
777     <skos:related rdf:resource="http://vocab.getty.edu/aat
        /300055027"/>
778 </owl:Class>
779
780
781
```

```

782 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      Tenseki -->
783
784 <owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/
      ichiba/tenseki#Tenseki">
785   <rdfs:subClassOf rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Book"/>
786   <rdfs:label xml:lang="ja">古典籍</rdfs:label>
787   <skos:Concept rdf:resource="http://vocab.getty.edu/page/aat
      /300266304"/>
788 </owl:Class>
789
790
791
792 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#Term
      -->
793
794 <owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/
      ichiba/tenseki#Term">
795   <rdfs:comment xml:lang="ja">古典籍特有の専門用語</rdfs:comment>
796   <rdfs:label xml:lang="ja">専門用語</rdfs:label>
797 </owl:Class>
798
799
800
801 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#Title
      -->
802
803 <owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/
      ichiba/tenseki#Title">
804   <rdfs:label xml:lang="ja">題</rdfs:label>
805 </owl:Class>
806
807
808
809 <!-- http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#
      TitleOfWork -->
810
811 <owl:Class rdf:about="http://www.dl.saga-u.ac.jp/OgiNabesima/
      ichiba/tenseki#TitleOfWork">
812   <rdfs:subClassOf rdf:resource="http://www.dl.saga-u.ac.jp/
      OgiNabesima/ichiba/tenseki#Title"/>
813   <dcterms:references rdf:resource="http://www.fumikura.net/
      other/biblio.html"/>
814   <rdfs:comment xml:lang="ja">書物の表紙に記載されている書名のこ
      と。表題・標題・題目ともいう。直接表紙に記された「打付外題」と、題簽に
      書名が記された「貼外題」との二種類がある。のうち墨書された題簽を「

```

書題簽」というが、一般こに板本の大部分には印刷された「摺題簽」が用いられている。題簽には短冊型と方型のものがあり、多くは外周に子持枠などの線が施される。漢籍の伝統を踏まえた書物は表紙の左肩に、物語の系統を継ぐものには中央に貼られることが多い。枠の中には書名や巻数などが摺られるが、二冊以上の場合には字体を楷書から行草書へと次第に崩すことがある。凝った例としては芝居絵本や浄瑠璃本などがあり、中広の短冊簽に絵や板元名・座元名などを入れたりした。これに影響を受けた草双紙では絵入の大きな「絵題簽」が使われるようになる。また目録などを記した「副題簽」を加える場合もある。しかし題簽は時の経過とともに糊が利かなくなり剥がれやすくなるので、時代を上るほど残存しているものは少ない。とりわけ初期の草双紙などは題簽以外に何一つ書誌事項が記されていないので、題簽が残存しているものは資料価値も高い。ただし後から別本の題簽が貼られたものも存在するので、外題に加えられた改変には注意を要する。

```

815 【参考資料】長沢規矩也『図解古書目録法』汲古書院、昭和 49年。中野三敏『書誌学談義
      江戸の板本』岩波書店、一九九五年。</rdfs:comment>
816     <rdfs:isDefinedBy xml:lang="ja">Copyright (C) 1999-2012 Gen
      TAKAGI</rdfs:isDefinedBy>
817     <rdfs:label xml:lang="ja">内題</rdfs:label>
818   </owl:Class>
819 </rdf:RDF>
820
821
822
823 <!-- Generated by the OWL API (version 4.2.6.20160910-2108) https://
      github.com/owlcs/owlapi -->

```

スクリプト G で生成する用語オントロジーは `tensekiOntology.owl` である。書誌オントロジーは基本オントロジーの最後尾に用語オントロジーの内容をペーストして作成する。Ichiba の書誌オントロジーは `source/bibliographicOntology.owl` および `http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/index.rdf` からダウンロード可能である。

ソースコード B.23: スクリプト G (用語オントロジー生成, `dataOntology.php`).

```

1 <?php //オントロジー表現データからRDF データを作成する
2 #if(function_exists("array_column")) {print "true"; exit;}
3 include_once "/home/yoshiga/analysis/function.php";
4 include_once '/var/www/html/tsr/dbsetting.php';
5
6 $field = file("/home/yoshiga/analysis/fieldDataIchiba.txt",

```

```

    FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
7 $jwordsClass = file('/home/yoshiga/analysis/words/class.csv',
    FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
8 $jwordsTFB = file('/home/yoshiga/analysis/words/tob.csv',
    FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
9 $jwordsRole = file('/home/yoshiga/analysis/words/role.csv',
    FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
10 $jwordsTerms = file('/home/yoshiga/analysis/words/terms.csv',
    FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
11 $nss = file('/home/yoshiga/analysis/words/namespaces.csv',
    FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
12 #print_r($nss); exit;
13
14 $baseUri = 'http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki#';
    //ichiba 語彙定義ファイル URI
15 $version = "2.0";
16
17 $patterns = array();
18 $replacements = array();
19 $patternsHttp = array('/\//', '\./');
20 $replacementsHttp = array('\/', '\. ');
21 $nsshort = '';
22
23 //スキーマ定義
24 /*
25 $rdf = '<?xml version="1.0" encoding="utf-8"?>'\n";
26 $rdf .= "<rdf:RDF \n";
27
28 //namespace を生成する
29 foreach ($nss as $valns) {
30     $ns = explode(',', $valns);
31     $rdf .= ' xmlns:'.$ns[0].'="' . $ns[1].'"'\n";
32     $patterns[] = "/" . preg_replace($patternsHttp, $replacementsHttp,
        $ns[1])."/";
33     $replacements[] = $ns[0].": ";
34 }
35 $rdf .= ">\n";
36 $rdf .= "<owl:Ontology rdf:about=\"\".$baseUri.\">\n";
37 $rdf .= "<owl:versionIRI rdf:resource=\"\".$baseUri.\"/".$version
    .\""/>\n";
38 $rdf .= "<rdfs:comment>Copyright ".date('Y')." Saga University
    Library. These data were normalized and restructured by Natsuko
    Yoshiga. </rdfs:comment>\n";
39 $rdf .= "</owl:Ontology>\n\n";
40 */
41
42 ///////////////////////////////////////////////////

```

```

43 //TFB, Role, Terms のオントロジー生成
44 $parts = array("TFB" => $jwordsTFB, "Role" => $jwordsRole, "Term" =>
    $jwordsTerms);
45 ///////////////////////////////////////////////////語彙URI を定義
46 $basewords = $partswords = array();
47
48 foreach($parts as $partName => $jwords) {
49     $key = 0;
50     $val = array();
51     foreach ($jwords as $key => $val) {
52         $basewords[$key] = explode(',', $val);
53         $partsArray[] = $partName.str_pad($key, 2, 0, STR_PAD_LEFT).",".
            $basewords[$key][0].",".@$basewords[$key][1].",".@$basewords[
            $key][2];
54         $partsArrayCol[] = array('pn' => $partName.str_pad($key, 2, 0,
            STR_PAD_LEFT), 'word' => $basewords[$key][0], 'ont' =>
            @$basewords[$key][1], 'labelen' => @$basewords[$key][2]);
55     }
56 }
57 #print_r($partsArray);
58 #print_r($partsArrayCol);
59 #print_r(array_column($partsArrayCol, 'word'));
60 #print_r(array_column($partsArrayCol, 'ont'));
61 #exit;
62 foreach($partsArray as $partsVal) {
63     $partswords = explode(",", $partsVal);
64
65     preg_match_all('/[=] [\(\]?([^\=C\(\)]+)\ [\)]?/u', $partswords[2],
        $match1); //sameAs
66     preg_match_all('/[C] [\(\]?([^\=C\(\)]+)\ [\)]?/u', $partswords[2],
        $match2); //subClassOf
67     preg_match_all('/[#] ([a-zA-Z]+)$/u', $partswords[2], $match3); //別
        Class
68     #print_r($match2);
69     #superClass = array();
70     #superClass = explode("|", $partswords[2]);
71
72     $class = "";
73     if (@$match3[0][0] != "") {
74         //#で始まるClass
75         $class = $match3[1][0];
76         #rdf .= "<owl:Class rdf:about=\"\".$baseUri.$class.\"\"/>\n";
77     } else {
78         #rdf .= "<owl:Class rdf:about=\"\".$baseUri.$partswords[0].\"\"/>\n
            ";
79
80     #print_r($match2);

```



```
81     $ont = array();
82     $ontkey = $ontkeyPart = "";
83     $node = $property = "";
84     $key = $key2 = $keyS = 0;
85     #print $match1[1][0].", ".$node.", ".$property.", ".$match2[1][0]."\n";
86
87     //owl:sameAs
88     if (@$match1[1][0] != "") {
89         //プロパティ
90         $ont = @explode(":", $match1[1][0]);
91         $node = $ont[0]; //序
92         $property = @$ont[1];
93
94         //owl 組み立て
95         if ($node != "") {
96             if ($property != "") {
97                 Switch($property) {
98                     case "題":
99                         //ある題が内題か外題かの判定
100                        if ($match2[1][0] == "外題") {
101                            $titletype = $baseUri."MainTitle";
102
103                        } elseif ($match2[1][0] == "内題") {
104                            $titletype = $baseUri."TitleOfWork";
105                        } else {
106                            $titletype = $baseUri."Title";
107                        }
108                        $needle = trim($node);
109                        $key = array_search($needle, array_column($partsArrayCol
110                            , 'word'), true);
110                        $rdf .= "\t<owl:equivalentClass>\n";
111                        $rdf .= "\t\t<owl:Class>\n";
112                        $rdf .= "\t\t\t<owl:intersectionOf rdf:parseType=\"
113                            Collection\">\n";
113                        $rdf .= "\t\t\t\t<owl:Class rdf:about=\"".$baseUri.
114                            $partsArrayCol[$key]['pn'].\""/>\n";
114                        $rdf .= "\t\t\t\t<owl:Class rdf:about=\"".$titletype
115                            .\""/>\n";
115                        $rdf .= "\t\t\t</owl:intersectionOf>\n";
116                        $rdf .= "\t\t</owl:Class>\n";
117                        $rdf .= "\t</owl:equivalentClass>\n";
118                        break;
119                    case "著":
120                        $needle = trim($node);
121                        $key = array_search($needle, array_column($partsArrayCol
122                            , 'word'), true);
```

```
122     $rdf .= "\t<rdfs:subClassOf rdf:resource=\"\".$baseUri.  
        preg_replace('/[0-9]{2}$/', '', $partswords  
        [0]).\""/>\n";  
123     $rdf .= "\t<owl:equivalentClass rdf:resource=\"\".  
        $baseUri.$partsArrayCol[$key]['pn'].\""/>\n";  
124     break;  
125     default:  
126     $needle = trim($node);  
127     $key = array_search($needle, array_column($partsArrayCol  
        , 'word'), true);  
128     $needle2 = trim($property);  
129     $key2 = array_search($needle2, array_column(  
        $partsArrayCol, 'word'), true);  
130  
131     $rdf .= "\t<owl:equivalentClass>\n";  
132     $rdf .= "\t\t<owl:Class>\n";  
133     $rdf .= "\t\t\t<owl:intersectionOf rdf:parseType=\"  
        Collection\">\n";  
134     $rdf .= "\t\t\t\t<owl:Class rdf:about=\"\".$baseUri.  
        $partsArrayCol[$key]['pn'].\""/>\n";  
135     $rdf .= "\t\t\t\t<owl:Class rdf:about=\"\".$baseUri.  
        $partsArrayCol[$key2]['pn'].\""/>\n";  
136     $rdf .= "\t\t\t</owl:intersectionOf>\n";  
137     $rdf .= "\t\t</owl:Class>\n";  
138     $rdf .= "\t</owl:equivalentClass>\n";  
139     break;  
140 }  
141 } else {  
142 //superclass が別途指定されている時  
143 $superClass = "";  
144 $needle = trim($node,"|");  
145 $key = array_search($needle, array_column($partsArrayCol, '  
        word'), true);  
146  
147 if (@$match2[1][0] != "") {  
148     $needleS = trim($match2[1][0]);  
149     if (!(($keyS = array_search($needleS, array_column(  
        $partsArrayCol, 'word'), true)))) {  
150         //本しかないので  
151         $superClass = $baseUri."Book";  
152     } else {  
153         $superClass = $baseUri.$partsArrayCol[$keyS]['pn'];  
154     }  
155  
156     $rdf .= "\t<rdfs:subClassOf rdf:resource=\"\".$superClass  
        .\""/>\n";  
157     $rdf .= "\t<owl:equivalentClass>\n";
```

```

158     $rdf .= "\t\t<owl:Class>\n";
159     $rdf .= "\t\t\t\t<owl:intersectionOf rdf:parseType=\"
      Collection\">\n";
160     $rdf .= "\t\t\t\t\t\t<owl:Class rdf:about=\"\".$baseUri.
      $partsArrayCol[$key]['pn']\">\n";
161     $rdf .= "\t\t\t\t\t\t<owl:Class rdf:about=\"\".$superClass
      .\">\n";
162     $rdf .= "\t\t\t\t</owl:intersectionOf>\n";
163     $rdf .= "\t\t\t</owl:Class>\n";
164     $rdf .= "\t</owl:equivalentClass>\n";
165   } else {
166     //sameAs となる単語の番号を探す
167     $rdf .= "\t<rdfs:subClassOf rdf:resource=\"\".$baseUri.
      preg_replace('/[0-9]{2}$/', '', $partswords[0])\">\n";
168     $rdf .= "\t<owl:equivalentClass rdf:resource=\"\".$baseUri
      . $partsArrayCol[$key]['pn']\">\n";
169   }
170 }
171 } else {
172   //superclass の表示
173   //オントロジー表現の部分がCで始まる時
174   if (@$match2[1][0] != "") {
175     $needleS2 = trim($match2[1][0]);
176     if (!(($keyS2 = array_search($needleS2, array_column(
      $partsArrayCol, 'word'), true))) {
177       //本しかないので
178       $superClass2 = $baseUri."Book";
179     } else {
180       $superClass2 = $baseUri.$partsArrayCol[$keyS2]['pn'];
181     }
182
183     $rdf .= "\t<rdfs:subClassOf rdf:resource=\"\".$superClass2
      .\">\n";
184   }
185 }
186 } else {
187   //オントロジー表現の部分がCで始まる時
188   if (@$match2[1][0] != "") {
189     $needleS2 = trim($match2[1][0]);
190     if (!(($keyS2 = array_search($needleS2, array_column(
      $partsArrayCol, 'word'), true))) {
191       //本しかないので
192       $superClass2 = $baseUri."Book";
193     } else {
194       $superClass2 = $baseUri.$partsArrayCol[$keyS2]['pn'];
195     }

```

```
196
197     $rdf .= "\t<rdfs:subClassOf rdf:resource=\"\".$superClass2
      .\"\"/>\n";
198 } else {
199     $rdf .= "\t<rdfs:subClassOf rdf:resource=\"\".$baseUri.
      preg_replace('/[0-9]{2}$/', '', $partswords[0]).\"\"/>\n";
200 }
201 }
202
203 $rdf .= "\t<rdfs:label xml:lang=\"ja\">\".$partswords[1].\"</rdfs:
      label>\n";
204 if ($partswords[3] != "") {
205     $rdf .= "\t<rdfs:label xml:lang=\"en\">\".$partswords[3].\"</rdfs
      :label>\n";
206 }
207
208 //典拠データを探す
209 $idHeadings = sql("select * from headings where word = '".
      $partswords[1]."'");
210 if (pg_num_rows($idHeadings) > 0) {
211     for ($n=0; $n<pg_num_rows($idHeadings); $n++) {
212         $headingsUri = pg_fetch_result($idHeadings, $n, 'uri');
213         if ($headingsUri != "") {
214             $rdf .= "\t<owl:sameAs rdf:resource=\"\".$headingsUri.\"\"/>\n
              n";
215         }
216     }
217 }
218 $rdf .= "</owl:Class>\n\n";
219 }
220 }
221
222 # $rdf .= "</rdf:RDF>\n";
223
224 print $rdf;
225 #exit;
226
227 //データ書き込み
228 $rdfDatafile = '/home/yoshiga/analysis/tensekiOntology.owl';
229 if (is_writable($rdfDatafile)) {
230     if (!($fpw = fopen($rdfDatafile, 'wb'))) {
231         trigger_error("データ書き込みファイルを開けません。");
232     } else {
233         fwrite($fpw, $rdf);
234         fclose($fpw);
235     }
236 } else {
```

```
237 trigger_error("ファイルを開ける状態ではありません。");
238 }
239 ?>
```

スクリプト H で生成した *Ichiba* 書誌の Linked Data は、<http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki/tensekiInstances.rdf> でダウンロード可能である。

ソースコード B.24: スクリプト H (Linked Data 生成, dataStructure.php).

```
1 <?php //RDF トリプル化
2 include_once '/home/yoshiga/analysis/function.php';
3 include_once '/var/www/html/tsr/dbsetting.php';
4
5 $baseUrl = "http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba/tenseki";
6 $baseSchema = "ichibaTenseki";
7 $field = file("/home/yoshiga/analysis/fieldDataIchiba.txt",
8 FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
9 $tobList = file("/home/yoshiga/analysis/words/tob.csv",
10 FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
11 $roleList = file("/home/yoshiga/analysis/words/role.csv",
12 FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
13 $termList = file("/home/yoshiga/analysis/words/terms.csv",
14 FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
15
16 foreach($tobList as $headTob) {
17 $arrayTob = explode(',', $headTob);
18 $tob[] = $arrayTob[0];
19 }
20
21 foreach($roleList as $headRole) {
22 $arrayRole = explode(',', $headRole);
23 $role[] = $arrayRole[0];
24 }
25
26 foreach($termList as $headTerm) {
27 $arrayTerm = explode(',', $headTerm);
28 $term[] = $arrayTerm[0];
29 }
30
31 foreach ($field as $valfield){
32 $arrayField = explode(',', $valfield);
33 if ($arrayField[4] == "mainField") {
```

```
30     $mainField[] = $valfield;
31     $fieldNames[] = $arrayField[1];
32     $metadata[] = $arrayField[2];
33 }
34
35 //type データ作成
36 $fieldTypes[] = $arrayField[7];
37 }
38
39 $csvData = file("/home/yoshiga/analysis/ichibaoriginal.csv",
40     FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
41 $nss = file('/home/yoshiga/analysis/words/namespaces.csv',
42     FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
43
44 $rdfDatafile = '/home/yoshiga/analysis/tensekiInstances.rdf';
45
46 $voc = ""; //$field から読み込んだ語彙
47 $vocab = array(); //$voc を配列化するための変数
48 $ins = $instr = $instrUri = "";
49 $instance = array(); //$voc を配列化するための変数
50
51 foreach ($mainField as $voc) {
52     $vocab[] = explode(',', $voc);
53 }
54
55 foreach ($csvData as $ins) {
56     $instance[] = explode(',', $ins);
57 }
58
59 $schema = '<?xml version="1.0" encoding="utf-8"?>'\n";
60 $schema .= '<rdf:RDF'\n";
61
62 //namespace を生成する
63 foreach ($nss as $valns) {
64     $ns = explode(',', $valns);
65     $schema .= 'xmlns:'. $ns[0]. '='. $ns[1]. ' '\n";
66 }
67
68 #print $schema; exit;
69
70 //属性のトリプル化
71 $sentence = array();
72 $mtextarray = array();
73 $rdfDataaval = $schema.">\n\n";
74
75 for($record=0; $record<count($csvData); $record++) { //レコード別の処理
76     $headingsPerson = "";
```

```
75
76 //mecab 解析済みデータから部品を抽出
77 $id = sql("select * from correctiondata where record = ".$record."
           order by record, fieldnum, sentence, wordstart, wordend;");
78 $idParts = sql("select * from correctiondata where record = ".
                $record." and type = 'TFB' order by record, fieldnum, sentence,
                wordstart, wordend;");
79
80 //主URI の記述
81 $mainUri = $baseUrl."/".str_pad($record, 3, 0, STR_PAD_LEFT);
82 $rdfDataVal .= "<rdf:Description rdf:about=\"\".$mainUri.\">\n";
83 $rdfDataVal .= "\t<rdf:type rdf:resource=\"\".$baseUrl.\"#Tenseki
                \"/>\n";
84 $rdfDataVal .= "\t<ichibaTenseki:hasMainURI rdf:resource=\"\".
                $mainUri.\">\n";
85
86 //isCollectionOf
87 $rdfDataVal .= "\t<ichibaTenseki:isCollectionOf rdf:resource=\"
                http://www.dl.saga-u.ac.jp/OgiNabesima/ichiba\">\n";
88
89 //主URI と各フィールドの関係
90 for($fn=0; $fn<count($mainField); $fn++) {
91     if ($instance[$record][$fn] != "") {
92         if ($metadata[$fn] != "") {
93             $mt = $metadata[$fn];
94             $rdfDataVal .= "\t<ichibaTenseki:.$fieldNames[$fn].\" rdf:
                resource=\"\".$baseUrl."/".str_pad($record, 3, 0,
                STR_PAD_LEFT)."-".$fn.\">\n";
95         } else {
96             $mt = "ichibaTenseki:.$fieldNames[$fn];
97         }
98         $rdfDataVal .= "\t<\".$mt.\" rdf:resource=\"\".$baseUrl."/".
                str_pad($record, 3, 0, STR_PAD_LEFT)."-".$fn.\">\n";
99     }
100 }
101
102 //部品構成
103 if (pg_num_rows($idParts) > 0) {
104     for ($cp=0; $cp<pg_num_rows($idParts); $cp++) {
105         $rdfDataVal .= "\t<ichibaTenseki:hasPart rdf:resource=\"\".
                $mainUri."/".str_pad($cp, 2, 0, STR_PAD_LEFT).\">\n";
106         $rdfDataVal .= "\t<ichibaTenseki:hasSubURI rdf:resource=\"\".
                $mainUri."/".str_pad($cp, 2, 0, STR_PAD_LEFT).\">\n";
107     }
108 }
109
110 //useTechnicalTerms
```

```

111 $idAllTerms = sql("select * from correctiondata where record = ".
    $record." and (type = 'TFB' or type = 'Role' or type = 'Term')
    order by record, fieldnum, sentence, wordstart, wordend;");
112 if (pg_num_rows($idAllTerms) > 0) {
113     for ($cu=0; $cu<pg_num_rows($idAllTerms); $cu++) {
114         $rdfDataval .= "\t<ichibaTenseki:useTechnicalTerms rdf:resource
            =\"\".pg_fetch_result($idAllTerms, $cu, 'uri')."\"/>\n";
115     }
116 }
117
118 //コメント
119 if (isIchibaComment($mainUri) == true) { $rdfDataval .=
    findIchibaComment($mainUri); }
120 $rdfDataval .= "</rdf:Description>\n\n";
121
122 //フィールド別の処理
123 for($fldnum=0; $fldnum<count($mainField); $fldnum++) {
124     $instr = $instance[$record][$fldnum]; //インスタンスそのもの(リテラ
        ル)
125
126     if($instr == "") {
127         continue;
128     } else {
129         $instrUri = $baseUrl."/".str_pad($record, 3, 0, STR_PAD_LEFT
            )."-".$fldnum;
130
131         Switch ($fldnum) {
132             //Identifier
133             case 0:
134                 $rdfDataval .= "<rdf:Description rdf:about=\"\".$instrUri
                    .\"\">\n";
135                 $rdfDataval .= "\t<rdfs:label>\".$instr.</rdfs:label>\n";
136                 if (isIchibaComment($instrUri) == true) { $rdfDataval .=
                    findIchibaComment($instrUri); }
137                 $rdfDataval .= "</rdf:Description>\n\n";
138                 break;
139             //Genre 型
140             case 1:
141                 $rdfDataval .= "<rdf:Description rdf:about=\"\".$mainUri
                    .\"\">\n";
142                 $rdfDataval .= "\t<ichibaTenseki:hasGenre rdf:resource=\"\".
                    $instrUri.\"\"/>\n";
143                 $rdfDataval .= "</rdf:Description>\n\n";
144
145                 $rdfDataval .= "<rdf:Description rdf:about=\"\".$instrUri
                    .\"\">\n";
146                 $rdfDataval .= "\t<rdfs:label>\".$instr.</rdfs:label>\n";

```



```

147         if (isIchibaComment($instrUri) == true) { $rdfDataval .=
148             findIchibaComment($instrUri); }
149         $rdfDataval .= "</rdf:Description>\n\n";
150         break;
151     case 2:
152     case 3:
153     case 4:
154     case 5:
155     case 6:
156         $rdfDataval .= "<rdf:Description rdf:about=\"\".$mainUri
157             .\"\">\n";
158         # $instrUri = $baseUrl."/".str_pad($record, 3, 0,
159             STR_PAD_LEFT)."-$fldnum;
160
161         //objectProperty
162         $rdfDataval .= "\t<ichibaTenseki:hasMainGenre rdf:resource
163             =\"\".$instrUri.\"\"/>\n";
164         $rdfDataval .= "</rdf:Description>\n\n";
165
166         //owl:samaAs を挿入 (テーブル headings から)
167         $idHeadings = sql("select * from headings where word = '".
168             $instr."'");
169
170         $rdfDataval .= "<rdf:Description rdf:about=\"\".$instrUri
171             .\"\">\n";
172         $rdfDataval .= "\t<rdf:type rdf:resource=\"\".$baseUrl.\"#\".
173             $vocab[$fldnum] [7].\"\"/>\n";
174         if (pg_num_rows($idHeadings) > 0) {
175             for ($n=0; $n<pg_num_rows($idHeadings); $n++) {
176                 $headingsGenreUri = pg_fetch_result($idHeadings, $n, '
177                     uri');
178                 $headingsGenreHeadings = pg_fetch_result($idHeadings, $n
179                     , 'headings');
180                 if ($headingsGenreUri != "") {
181                     $rdfDataval .= "\t<owl:sameAs rdf:resource=\"\".
182                         $headingsGenreUri.\"\"/>\n";
183                 }
184             }
185         }
186         $rdfDataval .= "\t<rdfs:label xml:lang=\"ja\">\".$instr.\"</
187             rdfs:label>\n";
188         if (isIchibaComment($instrUri) == true) { $rdfDataval .=
189             findIchibaComment($instrUri); }
190         $rdfDataval .= "</rdf:Description>\n\n";
191         break;
192
193     //Title の記述

```

```

182     case 9:
183     case 10:
184         $rdfDataval .= "<rdf:Description rdf:about=\"\".$mainUri
            .\">\n";
185         $rdfDataval .= "\t<ichibaTenseki:hasMainTitle rdf:resource
            =\"\".$instrUri.\"/>\n";
186         $rdfDataval .= "</rdf:Description>\n\n";
187
188         //インスタンスそのものの記述
189         $rdfDataval .= "<rdf:Description rdf:about=\"\".$instrUri
            .\">\n";
190         if ($vocab[$fldnum][7] != "") {
191             $rdfDataval .= "\t<rdf:type rdf:resource=\"\".$baseUrl
            .\"#\".$vocab[$fldnum][7].\"/>\n";
192         }
193         $rdfDataval .= "\t<rdfs:label xml:lang=\"ja\">\".$instr.\"</
            rdfs:label>\n";
194         if (isIchibaComment($instrUri) == true) { $rdfDataval .=
            findIchibaComment($instrUri); }
195         $rdfDataval .= "</rdf:Description>\n\n";
196         break;
197
198         //Person の正規化処理
199     case 13:
200         $idPerson = sql("
201             select * from correctiondata
202             where record = ".$record." and fieldnum = ".$fldnum." and
            type = '".$fieldTypes[$fldnum]."'
203             order by record, fieldnum, sentence, wordstart, wordend;"
204         );
205
206         $rdfDataval .= "<rdf:Description rdf:about=\"\".$instrUri
            .\">\n";
207         if ($vocab[$fldnum][7] != "") {
208             $rdfDataval .= "\t<rdf:type rdf:resource=\"\".$baseUrl
            .\"#\".$vocab[$fldnum][7].\"/>\n";
209         }
210         $rdfDataval .= "\t<rdfs:label xml:lang=\"ja\">\".$instr.\"</
            rdfs:label>\n";
211         if (isIchibaComment($instrUri) == true) { $rdfDataval .=
            findIchibaComment($instrUri); }
212         $rdfDataval .= "</rdf:Description>\n\n";
213
214         for ($i=0; $i<pg_num_rows($idPerson); $i++) {
215             $instrUri2 = pg_fetch_result($idPerson, $i, 'uri');
216             $instrPerson = pg_fetch_result($idPerson, $i, 'word');
217             $instrPersonSen = pg_fetch_result($idPerson, $i, '

```

```
218         sentence');
219     $idOther = sql("
220     select * from correctiondata
221     where record = ".$record." and fieldnum = ".$fldnum."
222           and type != '". $fieldTypes[$fldnum]."'
223           and sentence = ".$instrPersonSen."
224     order by record, fieldnum, sentence, wordstart, wordend
225           ;"
226 );
227 $instrOther = @pg_fetch_result($idOther, 0, 'word');
228 $instrOtherUri = @pg_fetch_result($idOther, 0, 'uri');
229 $instrOtherType = @pg_fetch_result($idOther, 0, 'type');
230
231 //古典籍と編著者内の人名との関係
232 $rdfDataval .= "<rdf:Description rdf:about=\"".$mainUri
233               ."\"><br>\n";
234 $rdfDataval .= "\t<dc:creator rdf:resource=\"".$instrUri2
235               ."\"/>\n";
236 $rdfDataval .= "</rdf:Description>\n\n";
237
238 //owl:samaAs を挿入 (テーブル headings から)
239 $idHeadings = sql("select * from ichibaviaf where \"check
240                 \" = 't' and ichiba = '". $instrUri2."'");
241
242 if (pg_num_rows($idHeadings) > 0) {
243     for ($k=0; $k<pg_num_rows($idHeadings); $k++) {
244         $rdfDataval .= "<rdf:Description rdf:about=\"".
245                       $instrUri2."\">\n";
246         $rdfDataval .= "\t<rdf:type rdf:resource=\"".$baseUrl
247                       ."#".$vocab[$fldnum][7]."\"/>\n";
248         $headingsPersonUri = pg_fetch_result($idHeadings, $k,
249         'ndla'); //NDLA uri
250         $headingsPersonViaf = pg_fetch_result($idHeadings, $k,
251         'viaf');
252
253         if ($headingsPersonUri != "") {
254             $rdfDataval .= "\t<owl:sameAs rdf:resource=\"".
255                             $headingsPersonUri."\">\n";
256             $rdfDataval .= "\t<owl:sameAs rdf:resource=\"".
257                             $headingsPersonViaf."\">\n";
258             $idHeadingsPerson = sql("select * from headings
259                                     where uri = '". $headingsPersonUri."'");
260             if (pg_num_rows($idHeadingsPerson) > 0) {
261                 $headingsPersonHeadings = pg_fetch_result(
262                 $idHeadingsPerson, 0, 'headings');
263                 $rdfDataval .= "\t<rdfs:label xml:lang=\"ja\">".
```

```

                $headingsPersonHeadings."</rdfs:label>\n";
251     }
252   }
253   if (isIchibaComment($instrUri2) == true) { $rdfDataval
        . = findIchibaComment($instrUri2); }
254   $rdfDataval . = "</rdf:Description>\n\n";
255   }
256 }
257
258 //ロールの記述
259 if ($instrOtherUri != "") {
260   $rdfDataval . = "<rdf:Description rdf:about=\"".
        $instrUri2."\">\n";
261   $rdfDataval . = "\t<ichibaTenseki:hasRole rdf:resource
        =\"".$instrOtherUri."\"/>\n";
262   $rdfDataval . = "</rdf:Description>\n\n";
263
264 //ロールの記述、典拠
265 $rdfDataval . = "<rdf:Description rdf:about=\"".
        $instrOtherUri."\">\n";
266
267 if ($instrOtherType == "Role") {
268   $keyRole = array_search($instrOther, $role);
269   $rdfDataval . = "\t<rdf:type rdf:resource=\"".$baseUrl
        ."#Role".str_pad($keyRole, 2, 0, STR_PAD_LEFT
        )."\"/>\n";
270 }
271
272 if ($instrOtherType == "TFB") {
273   $keyTob = array_search($instrOther, $tob);
274   $rdfDataval . = "\t<rdf:type rdf:resource=\"".$baseUrl
        ."#TFB".str_pad($keyTob, 2, 0, STR_PAD_LEFT
        )."\"/>\n";
275 }
276
277 $idHeadingsRole = sql("select * from headings where word
        = '".$instrOther."'");
278
279 if (pg_num_rows($idHeadingsRole) > 0) {
280   for ($l=0; $l<pg_num_rows($idHeadingsRole); $l++) {
281     $headingsRoleUri = pg_fetch_result($idHeadingsRole,
        $l, 'uri');
282     if ($headingsRoleUri != "") {
283       $rdfDataval . = "\t<owl:sameAs rdf:resource=\"".
        $headingsRoleUri."\"/>\n";
284     }
285   }

```

```

286     }
287
288     $rdfDataval .= "\t<rdfs:label xml:lang=\"ja\">".
        $instrOther."</rdfs:label>\n";
289     if (isIchibaComment($instrOtherUri) == true) {
        $rdfDataval .= findIchibaComment($instrOtherUri); }
290     $rdfDataval .= "</rdf:Description>\n\n";
291     }
292 }
293 break;
294
295 case 14:
296     //Date 型
297     $rdfDataval .= "<rdf:Description rdf:about=\"\".$instrUri
        .\">\n";
298     $rdfDataval .= "\t<rdf:type rdf:resource=\"\".$baseUrl."#
        Date\"/>\n";
299     $rdfDataval .= "\t<rdfs:label xml:lang=\"ja\">".$instr."</
        rdfs:label>\n";
300     if (isIchibaComment($instrUri) == true) { $rdfDataval .=
        findIchibaComment($instrUri); }
301     $rdfDataval .= "</rdf:Description>\n\n";
302
303     $idDate = sql("
304         select * from correctiondata
305         where record = ".$record." and fieldnum = ".$fldnum." and
            type = '".$fieldTypes[$fldnum]."'
306         order by record, fieldnum, sentence, wordstart, wordend;"
307     );
308
309     if (pg_num_rows($idDate) > 0) {
310         for ($i=0; $i<pg_num_rows($idDate); $i++) {
311             $instrUri2 = pg_fetch_result($idDate, $i, 'uri');
312
313             //古典籍と刊行年との関係
314             $rdfDataval .= "<rdf:Description rdf:about=\"\".$mainUri
                .\">\n";
315             $rdfDataval .= "\t<dc:date rdf:resource=\"\".$instrUri2
                .\"/>\n";
316             $rdfDataval .= "</rdf:Description>\n\n";
317
318             $rdfDataval .= "<rdf:Description rdf:about=\"\".
                $instrUri2.\">\n";
319             $idDate2 = sql("select * from correctiondata where uri =
                '".$instrUri2."'");
320             $idHeadingsDate = sql("select * from headings where uri
                != '' and type = '".$fieldTypes[$fldnum]."' and word

```

```

321         = '.pg_fetch_result($idDate2, 0, 'word').''");
322     if (pg_num_rows($idHeadingsDate) > 0) {
323         for ($m=0; $m<pg_num_rows($idHeadingsDate); $m++) {
324             $headingsDateUri = pg_fetch_result($idHeadingsDate,
325                 $m, 'uri');
326             #print $headingsDateUri."\n";
327             if ($headingsDateUri != "") {
328                 $rdfDataval .= "\t<owl:sameAs rdf:resource=\"".
329                     $headingsDateUri."/>\n";
330             }
331         }
332     }
333     $rdfDataval .= "\t<rdfs:label xml:lang=\"ja\">".
334         pg_fetch_result($idDate2, 0, 'word')."</rdfs:label>\n";
335     $rdfDataval .= "</rdf:Description>\n\n";
336 }
337 }
338 break;
339
340 case 15:
341     //Date 型
342     $rdfDataval .= "<rdf:Description rdf:about=\"".$instrUri
343         ."\n">\n";
344     $rdfDataval .= "\t<rdf:type rdf:resource=\"".$baseUrl."#
345         Date\"/>\n";
346     $rdfDataval .= "\t<rdfs:label>".$instr."</rdfs:label>\n";
347     if (isIchibaComment($instrUri) == true) { $rdfDataval .=
348         findIchibaComment($instrUri); }
349     $rdfDataval .= "</rdf:Description>\n\n";
350     break;
351
352 case 19:
353     //Collection 型
354     $rdfDataval .= "<rdf:Description rdf:about=\"".$instrUri
355         ."\n">\n";
356     $rdfDataval .= "\t<rdf:type rdf:resource=\"".$baseUrl."#".
357         $vocab[$fldnum][7]."\n\"/>\n";
358     $rdfDataval .= "\t<rdfs:label>".$instr."</rdfs:label>\n";
359     $rdfDataval .= "\t<owl:sameAs rdf:resource=\"http://www.dl.
360         saga-u.ac.jp/OgiNabesima/ichiba\"/>\n";
361     if (isIchibaComment($instrUri) == true) { $rdfDataval .=
362         findIchibaComment($instrUri); }
363     $rdfDataval .= "</rdf:Description>\n\n";
364     break;
365
366 default:

```

```
356 //上記タイプ以外の記述
357 $rdfDataval .= "<rdf:Description rdf:about=\"\".$instrUri
    .\">\n";
358 if ($vocab[$fldnum][7] != "") {
359     $rdfDataval .= "\t<rdf:type rdf:resource=\"\".$baseUrl
        .\"#\".$vocab[$fldnum][7].\">\n";
360 }
361 $rdfDataval .= "\t<rdfs:label xml:lang=\"ja\">\".$instr.</
    rdfs:label>\n";
362 if (isIchibaComment($instrUri) == true) { $rdfDataval .=
    findIchibaComment($instrUri); }
363 $rdfDataval .= "</rdf:Description>\n\n";
364 }
365 }
366 }
367
368 //////////// 副URI の記述
369 $partnum = 0;
370 $partSchema = '';
371
372 for ($i=0; $i<pg_num_rows($id); $i++) {
373     if (mb_strpos(pg_fetch_result($id, $i, 'type'), 'TFB') != false)
374     {
375         $rdfDataval .= "<rdf:Description rdf:about=\"\".$mainUri.\"/\".
            str_pad($partnum, 2, 0, STR_PAD_LEFT).\">\n";
376
377         //部品クラス名を$tob から検索
378         $keyTob = array_search(pg_fetch_result($id, $i, 'word'), $tob);
379
380         //ただしtob 以外の type の時は$tobList を参照する
381         if (mb_strpos($tobList[$keyTob], "#") != false) {
382             $t = explode(",", $tobList[$keyTob]);
383             $typeFromKeyTob = $baseUrl.$t[1];
384         } else {
385             $typeFromKeyTob = $baseUrl."#TFB".str_pad($keyTob, 2, 0,
                STR_PAD_LEFT);
386         }
387
388         $rdfDataval .= "\t<owl:sameAs rdf:resource=\"\".pg_fetch_result(
            $id, $i, 'uri').\">\n";
389         $rdfDataval .= "\t<rdf:type rdf:resource=\"\".$typeFromKeyTob
            .\">\n";
390         $rdfDataval .= "\t<dcterms:partOf rdf:resource=\"\".$mainUri
            .\">\n";
391         $rdfDataval .= "\t<ichibaTenseki:isPartOf rdf:resource=\"\".
            $mainUri.\">\n";

```

```
392 $rdfDataVal .= "\t<rdfs:label xml:lang=\"ja\">".pg_fetch_result
    ($id, $i, 'word')."</rdfs:label>\n";
393 $rdfDataVal .= "\t<dcterms:references rdf:resource=\"".$mainUri
    ."-".pg_fetch_result($id, $i, 'fieldnum')."\"/>\n";
394 if (isIchibaComment($mainUri."/".str_pad($partnum, 2, 0,
    STR_PAD_LEFT)) == true) { $rdfDataVal .= findIchibaComment(
    $mainUri."/".str_pad($partnum, 2, 0, STR_PAD_LEFT)); }
395 $rdfDataVal .= "</rdf:Description>\n\n";
396
397 $idSentence = sql("select * from correctiondata where record =
    ".$record." and fieldnum = ".pg_fetch_result($id, $i, '
    fieldnum')." and sentence = ".pg_fetch_result($id, $i, '
    sentence')." order by sentence, wordstart, wordend;");
398
399 if (pg_num_rows($idSentence) > 0) {
400     for ($j=0; $j<pg_num_rows($idSentence); $j++) {
401         $partObject = "";
402         Switch (pg_fetch_result($idSentence, $j, 'type')) {
403             case 'Date':
404                 $partSchema = "dc:date";
405                 $typeName = 'Date';
406                 $partObject = "ichibaTenseki:isCreatedInDate";
407                 break;
408             case 'PNoun':
409                 $partSchema = "dc:title";
410                 $typeName = 'Title';
411                 break;
412             case 'PNoun2':
413                 $partSchema = "dc:title";
414                 $typeName = 'TitleOfWork';
415                 break;
416             case 'Place':
417                 $partSchema = "dc:place";
418                 $typeName = 'Place';
419                 $partObject = "ichibaTenseki:isCreatedInPlace";
420                 break;
421             case 'Person':
422                 $partSchema = "dc:creator";
423                 $typeName = 'Person';
424                 $partObject = "ichibaTenseki:isCreatedBy";
425                 break;
426             default:
427                 $partSchema = "";
428                 break;
429         }
430
431         if ($partSchema != "") {
```



```
432     $uriToken = '';
433     $literalUriArray = array();
434     $literalUri = '';
435
436     $pattern = $mainUri."-";
437     $uriToken = str_replace($pattern, "", pg_fetch_result(
438         $idSentence, $j, 'uri'));
439     $literalUriArray = explode("-", $uriToken);
440     $literalUri = $mainUri."-".$literalUriArray[0]."-".
441         $literalUriArray[1]."-".$literalUriArray[2]."-".
442         $literalUriArray[3];
443
444     $rdfDataval .= "<rdf:Description rdf:about=\"".$mainUri
445         ."/".str_pad($partnum, 2, 0, STR_PAD_LEFT).\">\n";
446     $rdfDataval .= "\t<".$partSchema." rdf:resource=\"".
447         $literalUri.\">\n";
448     if ($partObject != "") { $rdfDataval .= "\t<".$partObject
449         ." rdf:resource=\"".$literalUri.\">\n"; }
450     if (isIchibaComment($mainUri."/" .str_pad($partnum, 2, 0,
451         STR_PAD_LEFT)) == true) { $rdfDataval .=
452         findIchibaComment($mainUri."/" .str_pad($partnum, 2, 0,
453         STR_PAD_LEFT)); }
454     $rdfDataval .= "</rdf:Description>\n\n";
455
456     //部品の典拠
457     if ($typeName == "Person") {
458         $idHeadingsParts = sql("select * from ichibaviaf where
459             \"check\" = 't' and ichiba = '".$literalUri."'");
460         if (pg_num_rows($idHeadingsParts) > 0) {
461             $rdfDataval .= "<rdf:Description rdf:about=\"".
462                 $literalUri.\">\n";
463             for ($n=0; $n<pg_num_rows($idHeadingsParts); $n++) {
464                 $idHeadingsPartsHeadings = sql("select * from
465                     headings where uri = '".pg_fetch_result(
466                         $idHeadingsParts, $n, 'ndla')."');
467                 if (pg_num_rows($idHeadingsPartsHeadings) > 0) {
468                     $tenkyoName = pg_fetch_result(
469                         $idHeadingsPartsHeadings, 0, 'headings');
470                 }
471                 $tenkyoUri = pg_fetch_result($idHeadingsParts, $n, '
472                     ndla');
473                 $tenkyoUri2 = pg_fetch_result($idHeadingsParts, $n,
474                     'viaf');
475                 $rdfDataval .= "\t<rdfs:label xml:lang=\"ja\">".
476                     $tenkyoName."</rdfs:label>\n";
477                 $rdfDataval .= "\t<owl:sameAs rdf:resource=\"".
478                     $tenkyoUri.\">\n";
479             }
480         }
481     }
482 }
```

```
461         $rdfDataval .= "\t<owl:sameAs rdf:resource=\"".
462             $tenkyoUri2."/>\n";
463     }
464     if (isIchibaComment($literalUri) == true) {
465         $rdfDataval .= findIchibaComment($literalUri); }
466     $rdfDataval .= "</rdf:Description>\n\n";
467     }
468     } else {
469     $idHeadingsParts = sql("select * from headings where uri
470         != '' and type = '". $typeName.'" and word = '".
471         pg_fetch_result($idSentence, $j, 'word')."");
472     if (pg_num_rows($idHeadingsParts) > 0) {
473     $rdfDataval .= "<rdf:Description rdf:about=\"".
474         $literalUri."\">\n";
475     for ($n=0; $n<pg_num_rows($idHeadingsParts); $n++) {
476         $tenkyoName = pg_fetch_result($idHeadingsParts, $n,
477             'headings');
478         $tenkyoUri = pg_fetch_result($idHeadingsParts, $n, '
479             uri');
480         $rdfDataval .= "\t<rdfs:label xml:lang=\"ja\">".
481             $tenkyoName."</rdfs:label>\n";
482         $rdfDataval .= "\t<owl:sameAs rdf:resource=\"".
483             $tenkyoUri."\">\n";
484     }
485     if (isIchibaComment($literalUri) == true) {
486         $rdfDataval .= findIchibaComment($literalUri); }
487     $rdfDataval .= "</rdf:Description>\n\n";
488     }
489     }
490     }
491     $partnum++;
492 }
493 }
494 //hasInnerTitle
495 $idInnerTitle = sql("select * from correctiondata where record =
496     ".$record." and type = 'PNoun2' order by record, fieldnum,
497     sentence, wordstart, wordend;");
498 if (pg_num_rows($idInnerTitle) > 0) {
499     for ($cit=0; $cit<pg_num_rows($idInnerTitle); $cit++) {
500         $rdfDataval .= "<rdf:Description rdf:about=\"". $mainUri."\">\n
501             ";
502         $rdfDataval .= "\t<ichibaTenseki:hasInnerTitle rdf:resource
503             =\"".pg_fetch_result($idInnerTitle, $cit, 'uri')."\">\n";
```

```
494     $rdfDataval .= "</rdf:Description>\n\n"; }
495 }
496
497 //Term
498 $idTerm = sql("select * from correctiondata where record = ".
    $record." and type = 'Term' order by record, fieldnum, sentence
    , wordstart, wordend;");
499 if (pg_num_rows($idTerm) > 0) {
500     for ($ct=0; $ct<pg_num_rows($idTerm); $ct++) {
501         $idTermH = sql("select * from headings where word = '".
            pg_fetch_result($idTerm, $ct, 'word')." and type = 'Term
            ';");
502         if (pg_num_rows($idTermH) > 0) {
503             for ($ch=0; $ch<pg_num_rows($idTermH); $ch++) {
504                 $rdfDataval .= "<rdf:Description rdf:about=\"".
                    pg_fetch_result($idTerm, $ct, 'uri')."\">\n";
505                 $rdfDataval .= "\t<owl:sameAs rdf:resource=\"".
                    pg_fetch_result($idTermH, $ch, 'uri')."\"/>\n";
506                 $rdfDataval .= "</rdf:Description>\n\n";
507             }
508         }
509     }
510 }
511 }
512
513 //ichiba上の単語に関する全情報
514 $id2 = sql("select * from correctiondata where type != 'ETC' AND
    type != 'UNK' order by record, fieldnum, sentence, wordstart,
    wordend;");
515 for ($i=0; $i<pg_num_rows($id2); $i++) {
516     $uri = pg_fetch_result($id2, $i, 'uri');
517     $label = pg_fetch_result($id2, $i, 'word');
518     $type = pg_fetch_result($id2, $i, 'type');
519     $fieldNum = pg_fetch_result($id2, $i, 'fieldnum');
520
521     Switch ($type) {
522         case 'Date':
523         case 'Place':
524         case 'Term':
525         case 'Person':
526         case 'Genre':
527             $type = $baseUrl."#".$type;
528             break;
529         case 'PNoun':
530             if ($fieldNum == 17) {
531                 $type = $baseUrl."#"."TitleOfWork";
532             } else {
```

```

533     $type = $baseUrl."#".'Title';
534   }
535   break;
536 case 'PNoun2':
537     $type = $baseUrl."#".'TitleOfWork';
538     break;
539 case 'Role':
540     $type = $baseUrl."#".'Role'.str_pad(array_search($label, $role
541       ), 2, 0, STR_PAD_LEFT);
542     break;
543 case 'TFB':
544     //部品クラス名を$tob から検索
545     $keyTob2 = array_search(pg_fetch_result($id2, $i, 'word'), $tob
546       );
547
548     //ただしtob 以外の type の時は$tobList を参照する
549     if (mb_strpos($tobList[$keyTob2], "#") !== false) {
550       $t2 = explode(",", $tobList[$keyTob2]);
551       $type = $baseUrl.$t2[1];
552     } else {
553       $type = $baseUrl."#TFB".str_pad($keyTob2, 2, 0, STR_PAD_LEFT
554         );
555     }
556
557     #$type = 'TFB'.str_pad(array_search($label, $tob), 2, 0,
558       STR_PAD_LEFT);
559     break;
560 default:
561 }
562 $rdfDataval .= "<rdf:Description rdf:about=\"".$uri.">\n";
563 $rdfDataval .= "\t<rdf:type rdf:resource=\"".$type."/>\n";
564 $rdfDataval .= "\t<rdfs:label xml:lang=\"ja\">".$label."</rdfs:
565   label>\n";
566 if (isIchibaComment($uri) == true) { $rdfDataval .=
567   findIchibaComment($uri); }
568 $rdfDataval .= "</rdf:Description>\n\n";
569 }
570
571 $rdfDataval .= "<rdf:Description rdf:about=\"http://www.dl.saga-u.ac
572   .jp/OgiNabesima/ichiba\">\n";
573 $rdfDataval .= "\t<rdf:type rdf:resource=\"".$baseUrl."#Collection
574   \"/>\n";
575 $rdfDataval .= "\t<foaf:homepage rdf:resource=\"http://www.dl.saga-u
576   .ac.jp/OgiNabesima\"/>\n";
577 $rdfDataval .= "\t<rdfs:label xml:lang=\"ja\">市場直次郎コレクション</
578   rdfs:label>\n";
579 $rdfDataval .= "\t<dcterms:rightsHolder rdf:resource=\"http://www.

```

```
lib.saga-u.ac.jp\"/>\n";
570 $rdfDataval .= "</rdf:Description>\n\n";
571
572
573 $rdfDataval .= "</rdf:RDF>\n";
574
575 //データ書き込み
576 #print $rdfDataval; exit;
577 if (is_writable($rdfDatafile)) {
578     if (!$fpw = fopen($rdfDatafile, 'wb')) {
579         trigger_error("データ書き込みファイルを開けません。");
580     } else {
581         fwrite($fpw, $rdfDataval."\n");
582         fclose($fpw);
583     }
584 } else {
585     trigger_error("ファイルを開ける状態ではありません。");
586 }
587
588 ?>
```