# 日本語 Wikipedia オントロジーの
# 自動構築と評価

2013 年度

玉川 奨

学位論文　博士(工学)

# 日本語 Wikipedia オントロジーの<br>自動構築と評価

2013 年度

慶應義塾大学大学院理工学研究科

玉川 奨

# 要旨

大規模オントロジーは，データ統合などの情報基盤として期待されているが，オントロジーの手動構築には，コストと保守に大きな課題を抱えている．その課題を解決するために，フリーテキストからのオントロジー自動構築が試みられてきたが，自然言語理解に限界があり，実用レベルに到達しないことから，近年，半構造情報を有する情報資源からオントロジーを自動的に構築する方法が注目されてきた. 以上の背景から, 本論文では, Web 上のオンライン百科事典である日本語版 Wikipedia の有する半構造情報から，インスタンスの抽出, 概念抽出, 概念間の上位下位関係, インスタンス間の意味的関係（プロパティ），プロパティの定義域と値域，プロパティタイプなど，オントロジーとして重要な情報を抽出する方法を検討し，大規模汎用オントロジー（日本語 Wikipedia オントロジーと呼ぶ）を自動構築し，日本語 Wikipedia オントロジーの領域オントロジー構築支援可能性と Linked Open Data のハブの観点から，その有用性を評価する．

以下に本論文の構成を示す.

はじめに，第 1 章において，本研究の背景，問題，目的について述べる．

第 2 章では，本研究の関連技術として，オントロジーの定義および具体例，オントロジー構築方法論，Wikipedia，Linked Open Data について述べると共に，それらの関連研究についても述べる．

第 3 章では，日本語版 Wikipedia から概念および概念間の関係（is-a 関係，クラスーインスタンス関係，プロパティ定義域，プロパティ値域，プロパティ上位下位関係，インスタンス間関係，その他の関係）を抽出することで，日本語 Wikipedia オントロジーを自動構築する手法の提案と各手法の評価について述べる．

第 4 章では，日本語 Wikipedia オントロジーの領域オントロジー構築支援としての評価について述べる．また，Linked Open Data としての設計と公開，Linked Open Vocabularies との連携による日本語語彙構築手法の提案と評価, 検索支援ツール WiLD の設計と評価により，Linked Open Data のハブとしての評価について述べる．これらの評価から日本語 Wikipedia オントロジーの有用性を示す.

最後に第 5 章では，本論文のまとめと今後の課題および展望について述べる．

# Title: Building up Japanese Wikipedia Ontology with Semistructured Information

## Abstract:

Large-scale ontologies are expected to work as an information infrastructure for information services, such as information retrieval and data integration. Because it takes many costs for human experts to build and maintain ontologies, much attention has been come to the work on automatic ontology construction from free text. However, natural language processing still has much limitation to free text and so the work has not been in practice yet. Thus more attention moves to automatic ontology construction from semi-structured information resources, such as Wikipedia.

This dissertation discusses how to extract important information to compose ontologies from Japanese Wikipedia (Japanese Wikipedia Ontology). They include instances, classes, super-sub relationships between classes, properties between instances, property domains and ranges, and property types. Furthermore, Japanese Wikipedia Ontology has been evaluated from the following points: how much to support for human experts to build up domain ontologies and how much it works as Japanese Linked Open Data Cloud Hub.

This dissertation has the following structure.

Chapter 1 describes backgrounds and goals of this research.

Chapter 2 explains what ontologies in information science are, and shows us ontology development process and environment, Wikipedia, and Linked Open Data.

Chapter 3 discusses how to extract important information to compose Japanese Wikipedia Ontology methods with extraction metrics, such as precision.

Chapter 4 evaluates how much Japanese Wikipedia Ontology support human experts to build up domain ontologies in the field of hydroelectricity and how much Japanese Wikipedia Ontology works as Japanese Linked Open Data Cloud Hub.

Chapter 5 wraps up lessons learned from building Japanese Wikipedia Ontology with case studies and shows us what future issues are.

# 目次

# 図目次

# 表目次

# 第1章　序論

## 1.1　背景と目的

近年，次世代 Web の候補の一つとして，セマンティック Web [1, 2]が多くの企業および研究者から注目を集めている．セマンティック Web は，ソフトウェアが意味理解可能な辞書に基づき，Web コンテンツにソフトウェア可読なメタデータを付与することによって，ソフトウェアが Web コンテンツの意味を理解し, 推論することを可能にしようという試みであり，メタデータを記述した機械可読目録として W3C (World Wide Web Consortium)[1] により標準化されているのがオントロジーである．現在米国では政府機関および民間企業において，データ統合，情報検索，情報共有などをはじめ様々な分野で，オントロジーを利用したソリューションが提供され始めている．特に，大規模なオントロジーの構築は情報検索やデータ統合において有用であり，日本語の大規模オントロジーとしては，日本語 WordNet [3]や日本語語彙大系 [4]などが存在している．しかし，これらは手動で構築されており，構築コストが大きい．オントロジーの手動構築には，膨大な時間がかかり，オントロジーの保守や更新が困難という問題がある．そこで近年，オントロジー工学のコミュニティは，オントロジー開発コストを削減するために，オントロジー学習 (Ontology Learning)とも呼ばれる，（半）自動的にオントロジーを構築する手法，方法論，アルゴリズム，ツールなどの研究開発に取り組んできた．特に，フリーテキストからのオントロジー学習に関しては，機械学習，知識獲得，自然言語処理，情報検索など，様々な専門分野の手法を組み合わせた手法がこれまで数多く提案されている[5].

一方，ユーザ参加型の大規模な半構造化情報資源が広がりをみせている．中でも情報鮮度・語彙網羅性の優れた百科事典 Wikipedia[2]がその代表例である．Wikipedia は Wiki ベースのオンライン百科事典であり，日本語版 Wikipedia は 2013 年 10 月現在，87 万を超える記事が存在する[3]．これは EDR 電子化辞書 [6]が持つ日本語登録数の 3 倍を上回っている．Wikipedia のような知識形態は「集合知」とも呼ばれ，一般的な概念から最新の技術動向に関する記事まで幅広い分野の記事が網羅されており，膨大なコンテンツ量が存在する．Wikipedia のデータは GFDL (GNU Free Documentation License) [7]のライセンスの下にフリーで公開され，SQL [8]や XML (Extensible Markup Language) [9]の形式でダウンロードすることができる．このような特色を持つ Wikipedia は，半構造情報資源であるため, フリーテキストに比べ, 構造情報資源であるオントロジーとのギャップが小さく，大規模で汎用的なオントロジー構築のためのコーパスとして非常に注目されており，現在

[1] http://www.w3.org
[2] http://ja.wikipedia.org
[3] http://ja.wikipedia.org/wiki/Wikipedia:日本語版の統計

Wikipedia からオントロジーを構築する様々な研究が行われている [10]．しかしながら，Wikipedia はユーザ参加型という性質上，厳密な体系化が行われていないため，Wikipedia からのオントロジー学習にも多くの課題が存在している．

また，構造情報の利用方法として，セマンティック Web の研究分野では，各 Web サイトで公開されている政府，科学，写真，音楽などのデータベースを RDF (Resource Description Framework) [11]化して連携する，LOD (Linked Open Data) [12]が注目を集めている．LOD では，各 RDF データベース間を相互にリンクするためのハブとして，英語版 Wikipedia から自動構築した DBpedia [13]と呼ばれるオントロジーおよび RDF データが活用されている．

本論文では日本語版 Wikipedia を情報資源として，大規模で汎用的なオントロジーを自動構築し，構築したオントロジー（日本語 Wikipedia オントロジー）の有用性を評価する．第一に日本語版 Wikipedia の有する半構造情報から日本語 Wikipedia オントロジーを自動構築する手法の提案と評価を示す．第二に日本語 Wikipedia オントロジーの領域オントロジー構築支援可能性と LOD ハブの観点から，日本語 Wikipedia オントロジーの有用性を示す．

## 1.2 日本語 Wikipedia オントロジーの自動構築

大規模オントロジーは，データ統合などの情報基盤として期待されおり，日本語の大規模オントロジーとしては，日本語 WordNet [3]や日本語語彙大系 [4]などが存在している．しかし，これらは手動で構築されており，オントロジーの手動構築には，膨大な時間がかかり，保守や更新が困難という問題がある．その課題を解決するために，フリーテキストからのオントロジー自動構築が試みられてきたが，自然言語理解に限界があり，実用レベルに到達しないことから，近年，半構造情報を有する情報資源からオントロジーを自動的に構築する方法が注目されてきた．その情報資源として，Web 上の百科事典である Wikipedia は語彙網羅性，即時更新性に優れており，半構造情報資源であることからフリーテキストと比べてオントロジーとのギャップが小さいため，Wikipedia を利用した研究は多い．しかしながら，Wikipedia はユーザ参加型という性質上，厳密な体系化が行われていないため，Wikipedia からのオントロジー構築にも多くの課題が存在している．特に日本語版 Wikipedia に適用可能なオントロジー構築手法はほとんど提案されていない．

以上より，本論文では，日本語版 Wikipedia から概念および概念間の関係（is-a 関係，クラスーインスタンス関係，プロパティ定義域，プロパティ値域，インスタンス間関係，その他の関係）を抽出することで，大規模で汎用的なオントロジー（日本語 Wikipedia オントロジー）を学習する手法の提案と各手法での評価を行う．

## 1.3 日本語 Wikipedia オントロジーの評価

領域オントロジーとは，特定の領域（法律やビジネスなど）に存在する概念とその間の関係を定義したものであり，ソフトウェアが RDF コンテンツを理解する際に，辞書的な役割を果たす. しかしながら, 領域オントロジーの構築と保守には多大なコストがかかる. そのため，多くの研究は，知識工学，自然言語処理，データマイニングなどの技術を用いて，テキストや汎用オントロジーなどの既存情報資源から（半）自動的に領域オントロジーを構築している[14, 15].

加えて，構造情報の利用方法として，セマンティック Web の研究分野では，各 Web サイトで公開されている政府，科学，写真，音楽などのデータベースを RDF 化して連携する，LOD が注目を集めている．各データベース間の情報を繋げることで，情報を容易に引き出してくる事が可能であり，これにより多くのアプリケーションやサービスでデータを簡単に参照し，利用することができる．海外の LOD では，各 RDF データベース間を相互にリンクするためのハブとして，英語版 Wikipedia から自動構築した DBpedia と呼ばれるオントロジーおよび RDF データが活用されている．

一方，LOD の語彙に着目した LOV (Linked Open Vocabularies) [16]という取り組みも存在している．LOV は，LOD の各データベースで使用されている関係名となる語彙を集めて，語彙の検索を可能にすることで，新たな LOD を構築する際に語彙の再利用を促す取り組みである．しかしながら，LOD を構築する際に，新たに語彙を作ってしまう方が，目的に合致する語彙を見つけてくるよりもはるかに容易であり，標準語彙と呼ばれる，既に普及している一部の語彙を除いて，再利用されているケースは少ない．加えて，国内では Linked Open Vocabularies に相当する取り組みがまだ存在しておらず，日本語の標準語彙というものがないため，今後さらに国内の LOD が普及する際に，LOD 構築者にとって障壁となりうる．

以上により，本論文では領域オントロジー構築支援可能性と LOD ハブの観点から，日本語 Wikipedia オントロジーの有用性を評価する．

## 1.4 論文の構成

以降，本論文の構成は次のとおりである．第 2 章では，本研究の関連技術として，オントロジーの定義および具体例，オントロジー構築方法論，Wikipedia，Linked Open Data について述べると共に，それらの関連研究についても述べる．第 3 章では，日本語版 Wikipedia から概念および概念間の関係（is-a 関係，クラスーインスタンス関係，プロパティ定義域，プロパティ値域，インスタンス間関係，その他の関係）を抽出することで，大規模で汎用的なオントロジー(日本語 Wikipedia オントロジー)を学習する手法の提案と各手法での評価を述べる．第 4 章では，日本語 Wikipedia オントロジーの領域オントロジ

一構築支援としての評価を述べる．また，Linked Open Data としての設計と公開，Linked Open Vocabularies との連携による日本語 Linked Open Data のための日本語語彙構築手法の提案と評価，検索支援ツール WiLD の設計と評価により，Linked Open Data ハブとしての評価を述べる．これら 2 点の評価から日本語 Wikipedia オントロジーの有用性を示す．最後に第 5 章では，本論文のまとめと今後の課題および展望について述べる．

# 第2章　関連研究

## 2.1　概要

本章では，本論文に関連する技術および関連研究を述べる．はじめに，2.2 節では，オントロジーの定義を述べ，代表的な汎用オントロジーとオントロジーの応用例について述べる．2.3 節では，Wikipedia と Wikipedia を用いたオントロジー構築についての既存研究を述べる．2.4 節では，Linked Open Data と Linked Open Vocabularies について述べ，いくつかの代表的な取り組みを紹介する．

## 2.2　オントロジー

セマンティック Web では，Web 上のリソースにメタデータを付与し，計算機がそれを理解して推論を行うなど，Web そのものを知識ベースとして扱えるようにすることが大きな目標であり，各種関連技術の仕様策定が行われている．現在，オントロジーは，メタデータを記述した機械可読目録として W3C により標準化されており，セマンティック Web の核となる要素となっている．以下では，オントロジーの定義と技術的な仕様を述べ，代表的な汎用オントロジーとオントロジーの応用例を述べる．

### 2.2.1　オントロジーの概要

オントロジーは，元々は哲学の世界における用語であり，その意味は「存在に関する体系的な理論（存在論）」である．世の中に存在するすべてのものを系統立てて説明するといいうことを目指したものである．一方，人工知能の立場からのオントロジーとしては，「概念化の明示的な仕様書（explicit specification of conceptualization）」という定義がなされている．ここでいう概念化とは，対象（世界）において興味を持つ概念と概念間の関係とを指している．オントロジーの構成物として，溝口理一郎著の書籍 [17]では以下の構成物があると定義されている．

- 対象世界から基本概念を切り出した結果としての「概念」の集合
- 概念の is-a 関係
- is-a 関係以外で必要となる概念間の関係
- 抽出した概念と関係の定義，あるいは意味制約の公理化

しかしながら，オントロジーは決定的な構築論が存在しないため，構築に膨大なコストがかかる．

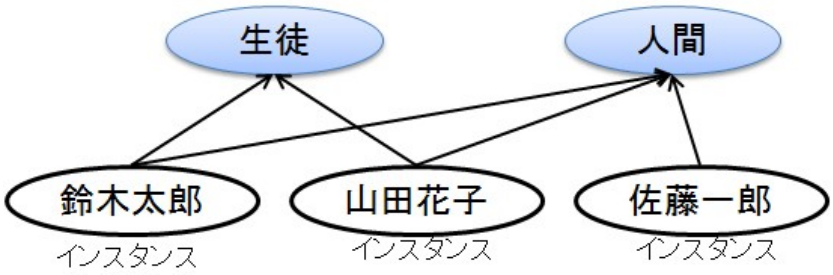



図 2.1 クラスーインスタンス関係の例

## 2.2.2 オントロジーの構成

- **クラス（概念）**

オントロジーにおいては，インスタンス（実体）の集合にふるラベルを「概念」という．従って，概念化とは，その集合にラベルをつけることを意味する．オブジェクト指向におけるクラスに近い要素であるが，オントロジーの概念は状態や振る舞いをひとまず考慮しない．例えば，研究室に在籍する各メンバーをインスタンスとした場合，このメンバー全体に「生徒」というラベルをつけることができる．もちろん，ラベルは他のもの（大学生，人間など）をいくらでもつけることができ，これら全てをオントロジーにおける概念と考えることができる．

- **is-a 関係**

is-a 関係（もしくは Kind-of 関係）とは，日本語で言えば，汎化ー特化関係といえる．言葉のとおり，「A is a B」といえる関係を指す．このとき，B の方がより抽象的な概念となり，両者の上位関係として，B が上位概念となり，A が下位概念となる．より抽象的な概念が上位に位置し，より具体的な概念が下位概念に位置することになる．ただし，概念と実体のインスタンス関係も「A is a B」と言えてしまうため，「A is a B」となる関係が全て is-a 関係とは言えない．オブジェクト指向における継承関係とほぼ同様の意味づけをできるが，何を継承するのかは場合による．図 2.2 に，is-a 関係の具体例を示す．

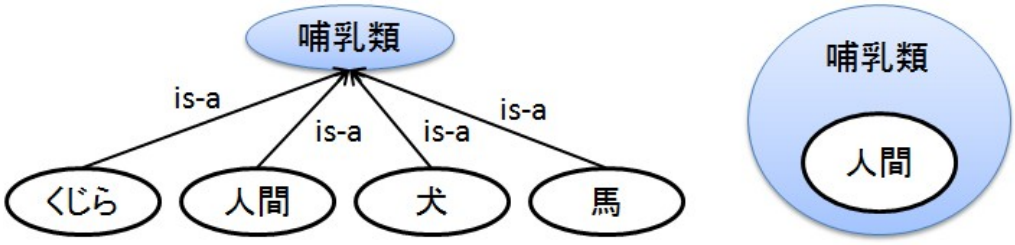



図 2.2 is-a 関係の例

例えば，「人間 is a 哺乳類」であり，人間と哺乳類の間には is-a 関係が成り立ち，哺乳類が上位概念，人間が下位概念である．概念を集合に付けられたラベルと考えると，哺乳類というラベルを振られたインスタンスの集合は，人間というラベルを振られたインスタンス集合より大きい．つまり哺乳類⊃人間といえる．カバーしているインスタンスの集合が大きい方を，小さい方から見た場合，一般化といい，ラベル内のインスタンスの中から，特定のサブセットを取り出すことが，特殊化である．

- **その他の関係**

その他の関係として定義できるものは無数にある．例えば現実世界における人間の親子関係や順番の前後関係等をオントロジーで表現でき，必要であれば，どんな関係を定義しても良い．

- **意味制約の公理化**

オントロジーでは，インスタンスの集合の組み合わせにおける和集合や積集合を概念として定義することができる．このような論理的組み合わせによる概念定義の他に，関係を利用して，意味制約から概念を定義することもできる．例えば，本名という関係を 1 つ持ち，少なくとも 1 つの親という関係を持っている概念を人間として定義できる．論理的組み合わせや意味制約から公理を形成し，概念を定義でき，これをクラス公理という．図 2.3 に公理と関係制約の概念図を示す．図 2.3 の左では，倫理的組み合わせにより，哺乳類という概念かつ草食動物という概念の積集合を草食哺乳類という概念として定義していおり，右では哺乳類という概念に含まれるインスタンスのうち，植物という概念に含まれるインスタンスを食料としているインスタンスを草食哺乳類という概念として定義している．

## 2.2.3 オントロジーの役割

オントロジーの使用場面とその効用として,溝口理一郎著の書籍から「合意を得る手段」，「暗黙情報と明示化」，「再利用と共有」，「知識の体系化」，「標準化」，「メタモデル的機能」，「統合的効用」の 7 つについて述べる．

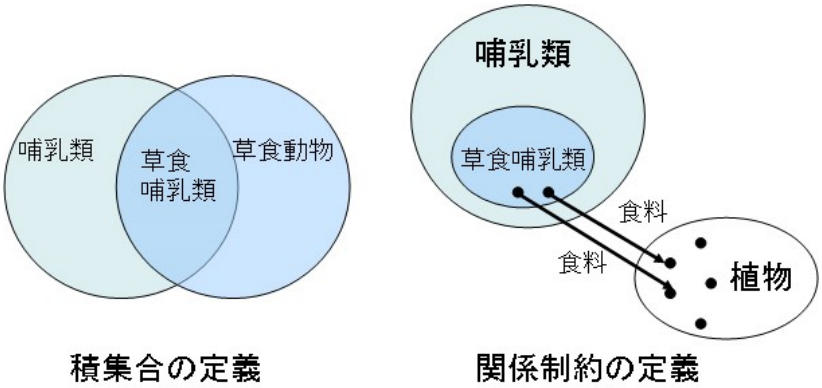



図 2.3 オントロジーにおける公理と関係制約の例

- **合意に達する手段**

他人と知識に関する合意のような，細部にわたる合意に達することは，容易ではない．オントロジーとは知識そのものとは異なり，知識を体系化したものであり，対象世界の骨格を明示化し，対象世界をよりシンプルにとらえたものといえる．それゆえ，オントロジーを媒介とすることで，複雑な事象をシンプルに捉えることができ，合意形成において有効となる．

- **暗黙情報を明示化**

対象世界の概念化とは，通常無意識のうちに仮定し，前提としている概念を明示化することである．知識ベースはもちろん，一般にソフトウェアは何らかの概念化に基づいているが，その概念化に関する情報は多くの場合暗黙的である．オントロジーはまさにこのような暗黙知識を記述しようとするものであり，その暗黙情報を明示化する役割を持っている．

- **再利用と共有**

知識の共有と再利用は，知識の前提となる概念が暗黙的であるため，困難である．特に，専門家の経験則などは多様な概念の集合であり，主観的要素が強いものも多く，共有と再利用は難しい．オントロジー構築の際に，知識を構成する基本概念に立ち戻って，対象世界を客観的な存在として考察することによって，そのような知識を構成する基本概念を同定するため，物事や対象の成り立ちを基本から検討し，共有・再利用可能な知識を見出す糸口を与えることができる．

- **知識の体系化**

人間は，情報の整理や前述したような再利用や共有のために知識を体系化するが，それは人間が理解するための体系化であり，コンピュータには理解できない．コンピュータ上で知識の体系化ができるとすれば非常に有用性が高いことは間違いない．知識の体系化にとって，最も重要なことは，関係する対象世界を構成している概念を明確化することと知識を記述するための共通語彙を定めることである．オントロジーはその両方を与えることができ，さらにコンピュータで処理可能である．

- **標準化**

オントロジーは，少なくともあるコミュニティで共有させることを目指して開発される．このことが示唆するように，オントロジーに含まれる概念や語彙は共有性が高く，知識の標準化へとつながる．

- **メタモデル的効力**

オントロジーとは，人工システムを構築する際にビルディングブロックとして用いられる基本概念と語彙の体系とも定義することができる．オントロジーの概念から，その概念に含まれるインスタンスを生成し，ある事象のモデルを構築するというオントロジー利用

プロセスを考えると，オントロジーはモデル構築に必要な基本概念とガイドラインを提供する効力があるといえる．

- **統合的効用**

以上，述べてきたオントロジーの効用を眺めてみると，オントロジーがいかに有用であるかが見てとれる．オントロジーにより，通常暗黙となっている基本的な概念が明示化され,それを共有することにより知識の根元となる概念も明示化することができ,複数の人々の間の合意形成，知識の標準化，事象のモデル構築に役立つ．構築されたモデルは，透明度の高い，共有することができる規範的なモデルとなる．オントロジーは，これらのことを実現する可能性を持っていて，知識マネジメントに貢献することができる．

### 2.2.4 オントロジーの分類

オントロジーは概念化の構造形式と概念化の対象という 2 つの特徴により分類することができる．以下順に，溝口理一郎著の書籍より本論文の内容に沿うように一部改編して引用する．

- **概念化の構造形式による分類**

  - Terminological Ontology（用語オントロジー）

    辞書のように,あるドメインの知識を表現するために使われる用語を体系化したものである．医療分野におけるこの種類のオントロジーの例としては，UMLS（Unified Medical Language System:1993）の意味ネットワークが挙げられる．

  - Information Ontology（情報オントロジー）

    データベースの記憶構造を体系化したもので，例として，データベースの概念スキーマが挙げられる．また，患者の医療記録をモデル化するためのフレームワーク（枠組）である PEN&PAD model の Level1 は，医療分野における情報オントロジーの典型的な例である．Level1 において，そのモデルは患者の基本的な観察について記録するためのフレームワークを提供する．

  - Knowledge Modeling Ontology（知識モデルオントロジー）

    知識を体系化したものであり，情報オントロジーと比較すると，このオントロジーは大抵,より豊かな内部構造を持っている.さらにこのオントロジーはしばしば,記述すべき知識が特殊な場合に使われる．知識ベースシステム開発の分野に限定すると,知識モデルオントロジーは,多くの研究者から注目を集めている.PEN&PAD model の Level2 の記述が,医療分野におけるこのオントロジーの例である.Level2 において，Level1 での観察結果は，意思決定プロセスの記述のために系統的分類がなされる．

- **概念化の対象による分類**

・ Application Ontology（アプリケーションオントロジー）

アプリケーションオントロジーは,特定のアプリケーションで要求される知識をモデル化するために，必要となる全ての定義を含む．一般的に，アプリケーションオントロジーはドメインオントロジーと汎用オントロジーを総合したものとなる．さらに，アプリケーションオントロジーはタスク固有の拡張を含む場合がある．また，アプリケーションオントロジーはそれ自体再利用可能ではない．再利用可能な部分は，特定のアプリケーションの為に微調整されたオントロジーライブラリーから，いくつかの theory を選択することによって得られる場合がある．

・ Domain Ontology（領域オントロジー）

日本語では領域オントロジーと訳されており,特定の専門領域についての知識を明確に定義したものである．現在の知識工学の方法論では，ドメインオントロジーとドメイン知識の間に明確な区別をする．ドメイン知識は，あるドメインにおける実際の状況を記述する（例，胸の痛みはアテローム性動脈硬化症の兆候である）のに対して,ドメインオントロジーは,ドメイン知識の構造と内容に制限を与える(例，病気には兆候がある)．

・ Generic Ontology（汎用オントロジー）

ドメインオントロジーと類似しているが,汎用オントロジーで定義される概念は,多くの分野にわたっており，一般的と考えられている．一般的に，汎用オントロジーは状態，出来事，仮定，行動，部分などの概念を定義する．ドメインオントロジーの中の概念は，しばしば汎用オントロジー中の概念を特殊化したものとして定義される．もちろん，汎用オントロジーは全ての分野の概念化を網羅的に列挙したわけではないため，汎用オントロジーとドメインオントロジーの間の境界は曖昧である．しかしながら，その区別は直感的に意味のあるものであり，ライブラリ構築の際に役立つ．また，このオントロジーが最初に研究されたオントロジーである．その後，ドメインオントロジーやアプリケーションオントロジーといったものが研究されていった．

・ Representation Ontology（表現オントロジー）

対象世界に立ち入らずにフレームワークのみを提供するオントロジーであり,このオントロジーはドメイン固有性を持たない．言葉のシンタックスを定義するようなものである．ドメインオントロジーと汎用オントロジーは，表現オントロジーによって提供されるプリミティブを使って記述される．この種類のオントロジーの例として，ontolingua [18]で使われるフレームオントロジーがあげられる．表現オントロジーは，オントロジーのためのオントロジーであり，そういう意味からメタオ

ントロジーとも呼ばれる．

### 2.2.5 オントロジー記述言語

オントロジー工学の考え方が定まっていくと同時に，オントロジーを記述するための言語が登場した．ここでは，W3C によって策定された OWL (Web Ontology Language) [19] 及びそのバージョンアップ版である OWL2 [20]について解説する．OWL は，前身である DAML と OIL という二つのオントロジー記述言語を統合し，改訂することによって完成した．この OWL は DAML+OIL [21]と同様に，XML によるリソース記述用のフレームワークである RDF の拡張として提供されている．図 2.4 にセマンティック Web の技術的階層図（レイヤーケーキ）[4]を示す．OWL は，セマンティック Web の技術階層に組み込まれており，セマンティック Web の核として期待されている．

以下では，OWL と OWL を支える技術である URI/IRI，XML，RDF，RDFS について説明する．

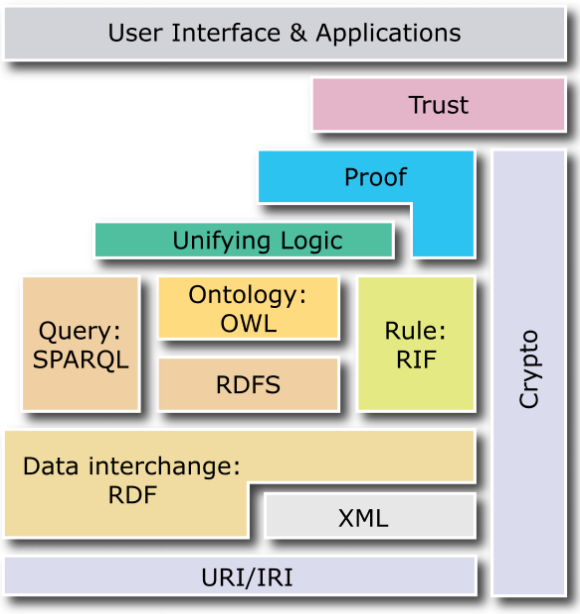



図 2.4 セマンティック Web のレイヤーケーキ

（出所）http://www.w3.org/2007/03/layerCake.png

[4]出典: W3C Semantic Web Activity, http://www.w3.org/2001/sw/

- **URI/IRI (Uniform Resource Identifier / Internationalized Resource Identifier)**

URI [22]は Web 上のリソース（あらゆるもの）を一定の書式の下で識別するために定められた識別子である．具体例としては，Web サイトにおける http://～で始まる URL (Uniform Resource Locator)が URI の一種と言える．URI は URL のような識別子と考えてよいが，URI としては，リソースが実際に Web 上にあるかどうかは問われない．つまり，HTML 文書や画像のような Web 上に置いておけるリソースでない人や物に対しても一意の名前を付け，Web 上で識別することができる．

IRI [23]に関しては多国語を使える URI である．Unicode の文字レパートリを使えるようにした URI であり，例えば以前ではエスケープ文字で表現されていた日本語での URI が日本語そのままで識別可能となる．

- **XML (Extensible Markup Language)**

XML [9]は文字列である文書にデータの意味や構造を表現できるようにするためのマークアップ言語兼マークアップ言語のメタ言語である．マークアップ言語はタグと呼ばれる特殊文字を用いて，文章に構造を埋め込むことができる言語で，例えば HTML もマークアップ言語に相当する．XML を単純に言語として，そのまま意味や構造を付け加えるために使用することができるが，本来 XML はメタ言語のため，意味や構造の種類や決まりを記述することもできる．

- **RDF (Resource Description Framework)**

RDF [11]は Web 上にリソースを表現するためのフレームワークである．RDF では「主語(subject)－述語(predicate)－目的語(object)」の三つ組み（トリプル）により，リソースとリソースの関係情報を表現する．トリプルは XML の表記に従い，タグで入れ子にすることで表現される．また，リソースは上述の URI によって識別されるものと，URI を持たない空白ノードがあり，述語部分は URI により表現されている．結果として，トリプルは「リソース－URI－リソース」とあらわすことができる．このトリプルを複数定義することで，ネットワーク構造のリソース集合とその関係を記述することができる．

- **RDFS (RDF Vocabulary Description Language: RDF Schema)**

RDFS [24]は RDF に基づき，トリプルにおけるリソースのカテゴリや述語の定義をするための語彙を提供する．オブジェクト指向において，インスタンスを生成するためのクラスを定義することと同等な意味を持つ．RDFS では，リソース集合の外延であるクラス(rdfs:Class)と，述語の定義であるプロパティ(rdfs:Property)が提供される．また，RDFS ではこのクラスもしくはプロパティ同士で継承関係を定義できる．RDFS ではこの継承関係と,プロパティにおける値域定義域,またラベルやコメントのみしか提供していないが，クラスや継承関係を定義できるため，RDFS はライトウェイトなオントロジー記述言語と言える．

- **OWL (Web Ontology Language)**

OWL [19]は RDF 形式の記述方法によってオントロジーを記述するために策定されたオントロジー記述言語である. OWL では RDF トリプルの集合としてオントロジーが記述され，OWL で記述されたオントロジーには以下の 4 つの構成要素を含む.

(1) オントロジー・ヘッダ
(2) クラスを定義するクラス公理
(3) プロパティを定義するプロパティ公理
(4) 個体(Individual)：クラスのインスタンスによる事実の記述

RDFS では基本的なクラスとプロパティ，また継承を定義していたが，オントロジーの構成物を全て記述するには表現形式が不十分である．OWL ではクラスの論理的組み合わせによる新たなクラスの定義や，プロパティによる制約されたクラスの定義，また，プロパティの特性を定義できる．また，オブジェクト指向におけるクラスーインスタンス関係のように，あるクラスにおけるインスタンスである実体(Individual)が定義できる．このような OWL の特徴は OWL によって記述したオントロジーを機械的に処理によって推論などを行うことを目的に作られている.

また，OWL は記述論理の厳密性の違いにより，DL，Full，Lite の三つのサブセットが用意されている．以下に述べるこれらサブセットは，オントロジーを利用する状況によって使い分けることが望まれる.

- OWL Full

  OWL サブセットの中では最大の表現力を持ち，OWL で提供される全ての語彙を用いて制約無くオントロジーの記述ができる. 複雑なクラス定義が可能であるが，推論における計算の完全性，決定可能性は保証されない．複雑なオントロジーを機械可読な形式で記述したいと言う場合に OWL Full の使用が望まれる.

- OWL DL

  記述論理に対応して作られた, OWL サブセットであり, DL は Description Logic の略である．語彙としては OWL Full と同じものを使用できるが，記述論理に基づいた決定可能性を保証するために，記述するための制約がある．機械的な推論を目的としたオントロジーでは OWL DL の使用が望まれる.

- OWL Lite

  Full や DL で用意された語彙の一部が使用できない，OWL Lite であるが，その分簡単に，単純な制約のみのオントロジーを記述することができる．形式が複雑でないため，オントロジーを利用したソフトウェアなどが実装しやすい.

以下に 4 つの構成要素の簡単な説明を示す.

### (1) オントロジー・ヘッダ

ヘッダは owl:Ontology 要素として記述し，バージョン情報と他のオントロジーのインポートを示す．さらに，OWL 以外の RDF 要素を埋め込む事ができる．

### (2) クラス公理

概念であるクラスは owl:Class 要素によって表現し, 次の要素でクラス公理を構成する．

- rdfs:subClassOf
  参照クラスのサブクラスとして，クラス間の必要条件（部分公理）を記述する．オントロジーにおける is-a 関係にあたる．

- owl:disjointWith
  参照クラスとは分離している（共通インスタンスがない）というクラス間の必要条件を記述する．

- owl:equivalentClass
  参照クラスと同じインスタンスを持つクラスというクラス間の必要十分条件(完全公理）を記述する．

- owl:oneOf
  インスタンスとなる個体を全て列挙することで必要十分条件（完全公理）を記述する．

- クラス式の組み合わせ
  匿名クラス（owl:Restriction）をつくり，クラス名，クラスの列挙，プロパティの制約条件，もしくはこれらの論理的組み合わせによって owl:Class に結びつけて公理を記述する．

### (3) プロパティ公理

プロパティは，オントロジーでの is-a 関係以外の関係を定義する部分になる．クラス公理でのプロパティの制約は，あるプロパティがそのクラスと共に用いられる際のローカルな制約を定義するが, プロパティ要素はそのプロパティそのものをグローバルに定義する．

プロパティには，個体（オブジェクト）を別の個体（オブジェクト）と関連づける個体値型プロパティと, オブジェクトをデータ型値に結びつけるデータ値型プロパティがあり，両者はそれぞれ owl:ObjectProperty 要素, owl:DatatypeProperty 要素で定義する. また，特別なプロパティとしてオントロジーの管理情報を記述する owl:OntologyProperty, オントロジーの注釈に用いる owl:AnnotationProperty がある．OWL でのプロパティは，必ずこの 4 つのどれかのタイプを持たなければならない．

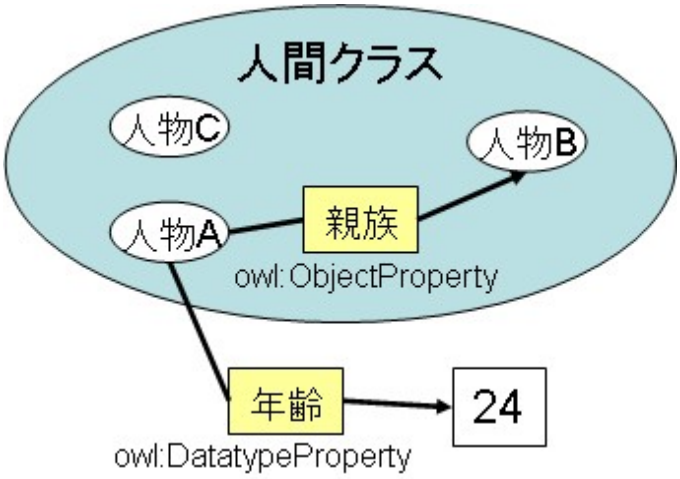



図 2.5 owl:ObjectProperty と owl:DatatypeProperty の例

図 2.5 に owl:ObjectProperty と owl:DatatypeProperty の一例を示す．図 2.5 では，人間クラスのインスタンスである人物 A は親族プロパティにより，インスタンスである人物 B と関連付けられており，さらに，年齢プロパティによりデータ型である 24 と結びついている．

個体値型プロパティ,データ値型プロパティの基本的な公理は次の構成要素で記述する．

- rdfs:subPropertyOf
  参照プロパティのサブプロパティ

- rdfs:range
  プロパティの値域である．プロパティの目的語は，参照クラスのインスタンスである．

- rdfs:domain
  プロパティの定義域である．プロパティの主語は，参照クラスのインスタンスである．

- owl:equivalentProperty
  参照プロパティと同じインスタンス（主語，目的語リソースの組み合わせ）を持つ．

- owl:inverseOf
  参照プロパティと反対の関係を表現する．

図 2.5 を例にすると,「親族」プロパティの主語であるインスタンス「人物 A」は「人間」クラスに属している．そのため，rdfs:domain は「人間」クラスとなる．同様に目的語であるインスタンス「人物 B」も「人間」クラスのため，rdfs:range も「人間」クラスとなる．「親族」プロパティのサブプロパティとしては，「家族」，「兄弟」，「親」などが考えら

れる.

OWL では，プロパティの論理的な性質（タイプ）を示すことで，その関係を利用した推論などを可能にする．以下の 4 つのタイプが存在する.

- owl:TransitiveProperty
  推移関係プロパティ.「子孫」プロパティのように, P(x,y)と P(y,z)が真なら P(x,z)も真であるという関係が推移していくプロパティ

- owl:SymmetricProperty
  対称関係であるプロパティ.「夫婦」プロパティのように，P(x,y)⇔P(y,x)が成り立つプロパティ

- owl:FunctionalProperty
  関数関係プロパティ.「本名」のように，値が唯一に定まるプロパティ

- owl:InverseFunctionalProperty
  逆関数関係プロパティ.「ISBN」のように，その値から主語が特定できるようなプロパティ

**(4) 個体による事実の記述**

クラスやプロパティの公理は，用語集や推論などを行うためのルール集のような役割を果たし, これを用いて, 実際に存在するものを具体的に描くのがインスタンスとなり, OWL では，インスタンスは必ず何かのクラスに属する.

## 2.2.6 オントロジー構築支援ツール

前項で紹介した OWL 等のオントロジー記述言語は人間が記述でき，機械が読める特徴がある．しかし，いくらマークアップ言語として見通しが良く構造化されていても，URI とタグの羅列となってしまう平文をそのまま人間が読み解いたり記述したりすることは難しい．特にオントロジーが大規模になってくるほど，この問題は顕著になり，オントロジーを簡単に構築するツールの必要性が高まる．現在までに多数のオントロジー構築ツールが開発されたが，ここでは，OWL オントロジー構築ツールにおいて幅広く利用されている Protégé [25]と，半自動オントロジー構築ツール DODDLE-OWL [26]を紹介する.

- **Protégé**

Protégé [25]はスタンフォード大学で開発された Java ベースのオントロジー構築ツールである．OWL におけるクラスの階層定義と，プロパティの階層定義，及び実体(Individual)の定義をグラフィカルに記述していくことができる．Protégé は OWL 専用のオントロジー構築ツールではないが, 特に OWL に適合して改良されてきており, OWL DL

におけるクラス公理やプロパティ制約を分かりやすく記述することができる．また，Protégé はプラグインに対応しており，現在までにグラフィカルツールなど，様々なプラグインが開発されている．

- **DODDLE-OWL**

DODDLE-OWL [26]は Protégé とは趣向が違い，オントロジーを半自動で構築することを目的としている．入力は，対象ドメインの専門文書等で，自然言語文を入力とすることができ，ユーザとの対話的半自動構築によって，最終的に OWL 形式のドメインオントロジーを出力することができる．DODDLE-OWL は図 2.7 のように 6 つのモジュールから成り立っている．

入力モジュールでは，専門文書から形態素解析などの自然言語処理を駆使して用語を抽出する．こうして抽出された用語はオントロジー構築モジュールにおいて，WordNet 等の汎用オントロジーと照らし合わせることで概念階層が構築される．一方で，専門文書から相関ルールなどを用いて概念間の関係を抽出し，概念対集合とする．概念階層と概念対集合は，オントロジー洗練モジュールにおいて，視覚化モジュールを通してユーザに提示され，ここで，階層の修正を行うことで，オントロジーを完成させる．完成したオントロジーは変換モジュールによって，OWL 形式のオントロジーに変換され出力される．

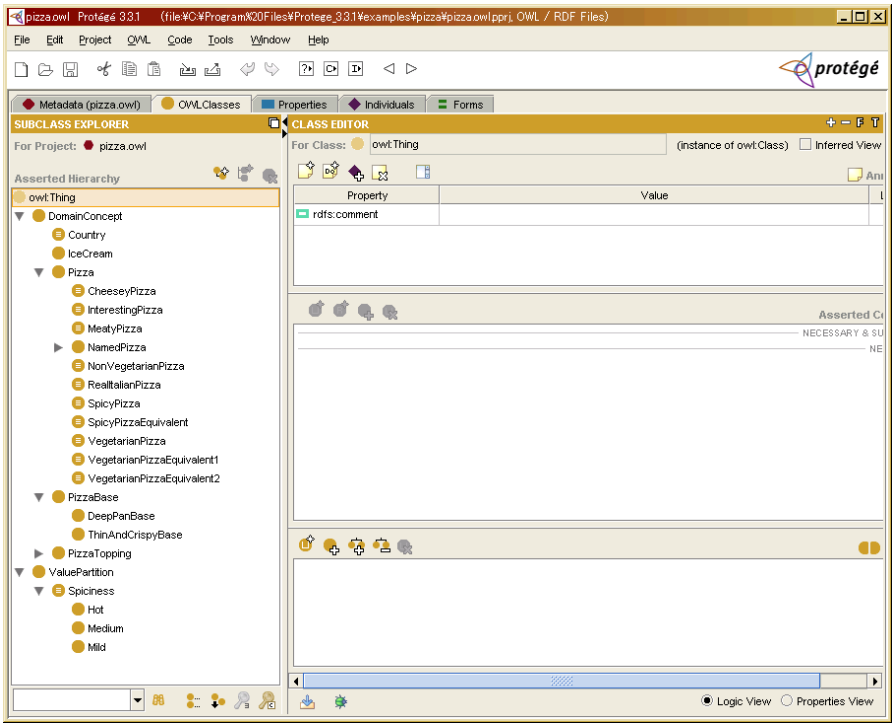



図 2.6 Protégé のクラス階層画面

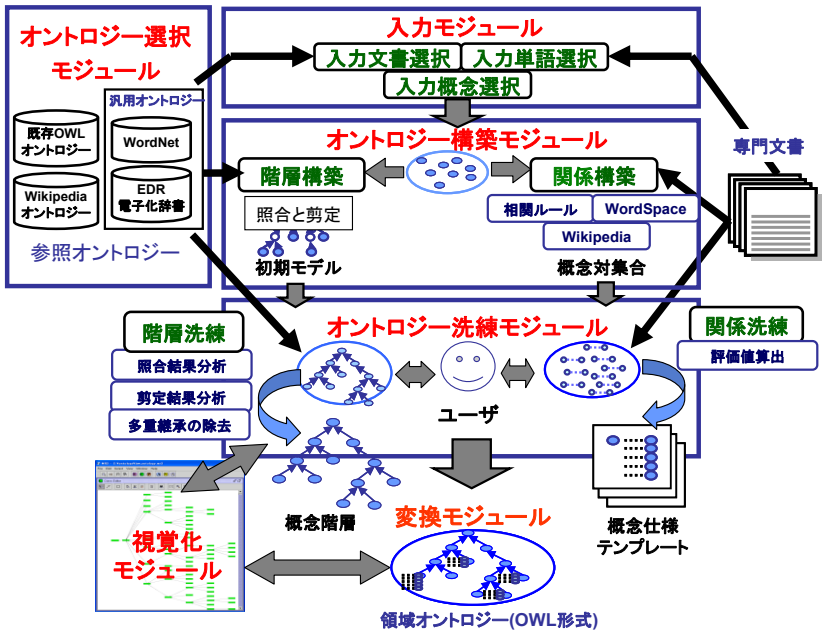



図 2.7 DODDLE-OWL の構成

（出所）DODDLE プロジェクト[5] 基本設計，DODDLE-OWL のシステムフロー

## 2.2.7 汎用オントロジー

自然言語理解の研究分野では，電子化辞書（MRD: a Machine Readable Dictionary)の開発が精力的に行われており，オントロジーというと電子化辞書を指す場合が多い．よく知られている電子化辞書としては，WordNet（プリンストン大学），EDR 電子化辞書（情報通信研究機構），日本語語彙大系（NTT コミュニケーション科学基礎研究所）などがある．電子化辞書の特徴として，定義される概念が一般的かつ多くの分野にわたっている点があげられる．そのため，電子化辞書は汎用オントロジーとしてとらえることができる．以下では，概念階層構造が整っていることから広く使われている WordNet，階層構造としての is-a 関係だけでなく他の概念関係子もサポートしている EDR 電子化辞書，日本語に特化することで最大規模の概念を有する日本語語彙大系について概略を述べる．

[5] http://doddle-owl.sourceforge.net/ja/

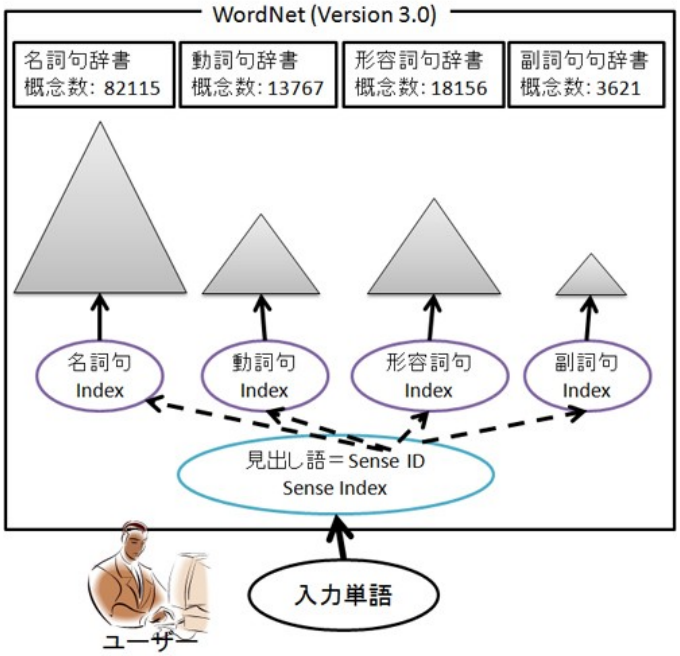



図 2.8 WordNet の概観

表 2.1 WordNet (version 3.0) の辞書サイズ

| 辞書名 | 見出し（語彙）数 | 意味（概念）数 |
|---|---|---|
| 名詞句辞書 | 117, 798 | 82, 115 |
| 動詞句辞書 | 11, 529 | 13, 767 |
| 形容詞句辞書 | 21, 479 | 18, 156 |
| 副詞句辞書 | 4, 481 | 3, 621 |
| 合計 | 155, 287 | 117, 659 |

- **WordNet**

WordNet [3] (version 3.0) は，図 2.8 に示すように，名詞句辞書，動詞句辞書，形容詞句辞書，副詞句辞書から構成されており，総計約 15 万の語彙を保持している．各々の辞書に記録されている見出し数および概念数を表 2.1 に示す．

同じ概念を意味するいくつかの単語見出しが，同じ概念 ID によって一つの概念にまとめられており，この集合を synset (synonym set) と呼ぶ．WordNet 内では，この synset を単位として階層・定義の記述が成されている．

名詞句辞書と動詞句辞書のみが階層構造を持ち，一部の概念 ID には，反対概念の概念 ID，part of，member of，substance of 関係の概念 ID なども与えられている．

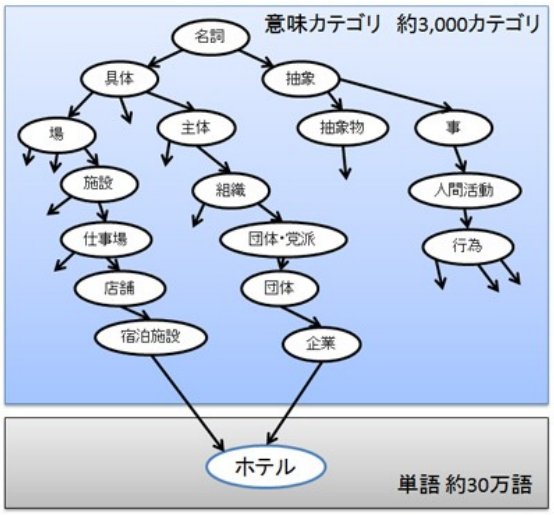



図 2.9 日本語語彙大系の意味カテゴリと単語（ホテル）の対応関係の例

- **EDR 電子化辞書**

EDR 電子化辞書[6] は，単語辞書，対訳辞書，概念辞書，共起辞書，専門用語辞書（情報処理），EDR コーパスから構成され，日本語単語辞書は約 27 万語，概念辞書は約 40 万概念が収録されている．単語辞書は，見出し情報，文法情報，意味情報，運用・その他の情報から構成されており，意味情報には，概念辞書の各概念ノードを識別するための概念識別子が割り当てられ，単語辞書と概念辞書を結合する働きを持っている．一方，概念辞書には，多重継承を許す概念階層関係を定義した概念体系辞書と，agent（動作主体），object（対象），goal（目標），implement（道具・手段），cause（原因），place（場所），scene（場面），a-object（属性を持つ対象）という 8 種類の概念関係子による概念間関係を定義した概念記述辞書がある．各概念は，主に，概念識別子，概念見出し，概念の説明を持つ．

- **日本語語彙大系**

日本語語彙大系 [4]は約 3,000 種の意味カテゴリと約 30 万語の単語から構成されており，意味カテゴリは名詞，固有名詞，用言という 3 つのルート意味体系から階層構造により構成され，各単語は意味カテゴリを持つ．図 2.9 に，日本語語彙大系の意味カテゴリと単語（ホテル）の対応関係の例を示す．

## 2.2.8 オントロジーの応用例

オントロジーの応用は幅広い．現在，米国では，政府機関および民間企業において，データ統合，情報検索，情報共有などをはじめ様々な分野で，オントロジーを利用したソリューションが提供され始めている．応用の対象として，ソフトウェア開発，インフラストラクチャ，情報システム，ナレッジシステム，行動システムなどが挙げられる．

本項では，オントロジーの応用例として，データ統合，自然言語検索，ソーシャルブックマークへの応用について述べる．

- **データ統合への応用**

機械に対して共通理解を提供するオントロジーをデータ統合に応用する事例が多く存在し，実際のビジネスシーンの中でオントロジーを利用したソリューションが登場している．例えば，オラクル社は企業データの統合の技術として RDF とオントロジーを利用したデータベース製品を開発している．各企業組織または業界から抽出したデータ・スキーマに基づき作成されたオントロジーを利用し，様々なアプリケーション固有のデータ・スキーマを統合する技術を提案している [27].

図 2.10 が表すように，オントロジーが異機種間のデータソースへの問い合わせとアプリケーション固有のスキーマを一致させる．オントロジーによるデータモデル管理は，ファイルベースまたは特殊データベースによるアプローチにはない大きな利点を持つ．主な 5 つを以下にまとめる．

・ 総所有コストの削減

セマンティック・アプリケーションは，他のアプリケーションと組み合わせることができ，データを中央に保存して企業レベルで配置できるので，所有コストが削減される．企業データベース以外では，サービス指向型アーキテクチャ（SOA：Service Oriented Architecture)によって，クライアント側のソフトウェアのデスクトップへのインストールや，データの個別管理等をする必要がなくなる．

・ 保守や更新によるリスクを低下

RDF および OWL モデルは，既存の組織データ，XML，空間的情報，およびテキスト文書とともに，企業の DBMS に直接統合できる．その結果，結合されたスケーラブルでセキュアな高性能アプリケーションの実現が可能となる．既存の IT リソースを使用する任意のサーバプラットフォームにこれらのアプリケーションを配置し，管理できる．

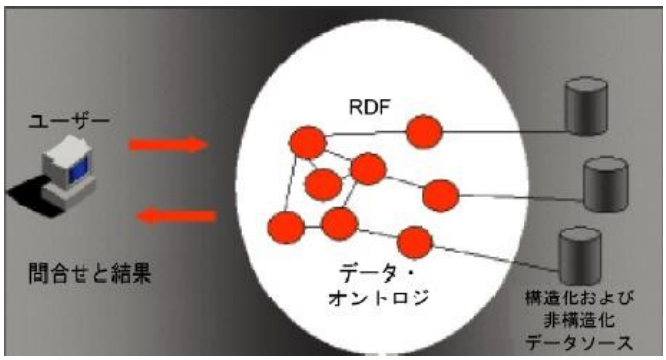



図 2.10 エンタープライズ統合のワークフロー

- 高い価値

  インターネットを使用して，より多数のユーザが，実質的な追加コストなしに，組織のアプリケーションにアクセスできる．そのため，ミッションクリティカルな情報にアクセスする必要のあるすべてのユーザは年間 365 日，1 日 24 時間いつでも情報にアクセスできる．

- パフォーマンスとセキュリティ

  マルチテラバイトの RDF データベースを管理し，ミッションクリティカルなセマンティックデータモデルに対して，データベースのセキュリティ，スケーラビリティおよびパフォーマンスの提供が可能となる．

ビジネス情報，科学的データ，政府文書，電子メール・メッセージ，および Web コンテンツの増加が止まらない現状では，データを統合し，ビジネス情報のエンタープライズリポジトリから新しい意味や価値，情報を得る多くの機会が存在する．企業，科学者，政府アナリストは，構造化および非構造化データの異機種間ソースへのアクセスを試みるシステムの構築を始めている．現在までは，これらのシステムにはそのようなドメイン間の統合を可能にするように構造化されたものは存在しなかった．データ統合は，異なるドメインおよびアプリケーションの領域に，具体的なメリットを提供する．米国では以下に示す領域でのケーススタディが盛んに行われている．

- エンタープライズ・データ統合
- ドメイン・データ・アグリゲーション
- コンテキスト・アグリゲーション/ナレッジ管理
- 企業向け検索

以下，情報集約型ナレッジワークの自動化やセマンティック・インフラに含まれるセマンティック Web 関連のソリューションとして，複雑なデータの統合を行った航空宇宙局（NASA）の事例 [28]を紹介する．全米 11 ヶ所に宇宙センターおよび研究機関などを抱える NASA では，毎日膨大な量のデータが生成されている．しかし，同局では，これら 11 機関によって生成されるデータをひとつに集合させるといった中央集中型のデータ構造を採用しておらず，データ統合が非常に複雑となっている．また，同局のデータは，異なるデータベースに保管されており，データ･フォーマットが統一されていないため，データの検索が困難であり，見つかりにくいデータなどは何度も作成されるなど，データ重複の原因となっていたという．こうした状況を改善するために，同局は現在，既存データソースを利用して効率的なデータ管理を行っていくために，同局内のグループやプロジェクトに対して，セマンティック Web 技術の RDF やオントロジーの利用を推進し，NASA 全体におけるデータの統合を進めた．同局では，地球科学分野における情報の発見，利用，共有を促進するために大規模なオントロジー「SWEET（Semantic Web for Earth and

Environmental Terminology)」が開発され，既に複数のプロジェクトによって使用されているほか，JAVA を使ったセマンティック・ブラウザ・アプリケーションである「mSpace」や「jSpace」などのユーザインタフェースも開発した．NASA の最高技術責任者（CTO）である Andrew Schain 氏によると，同局のチーフエンジニア室（Office of Chief Engineer）では，セマンティック Web の研究開発に取り組んでいる Clark & Parsia 社と協力し，4 つの異なるデータベースの情報を RDF でエンコードし，ブラウズした情報を表示するユーザインタフェース「jSpace」を構築した．ユーザは，同ユーザインタフェースを利用することによって，4 つの異なるデータベースにある情報を自由にブラウズすることが可能となった．jSpace ブラウザは，異なるデータベースの情報を包括的に検索し，その結果を人物（People），機関（Organizations），プロジェクト（Projects），スキル（Skills）の 4 つのフィールド（通称：POPS）にあわせて表示できるように，それらに適合した情報を導き出すことができる．

- **自然言語検索（質問応答システム）への応用**

RDF で表現している Triple と呼ばれる主語－述語－目的語の関係を持つデータの検索を，自然言語から変換したクエリにより行う研究が多くなされている．例えば，AquaLog [29]などがある．この研究において自然言語検索の仕組みは，「自然言語による質問→クエリトリプルの生成→オントロジーと互換性のあるトリプル生成→答え」という流れで構成されている．図 2.12 が，その概要である．

例えば，「日本の首都はどこですか？」というクエリに対して，日本→[首都]→（答え）というクエリトリプルを生成し，データベースを対応させることで答えを返す．

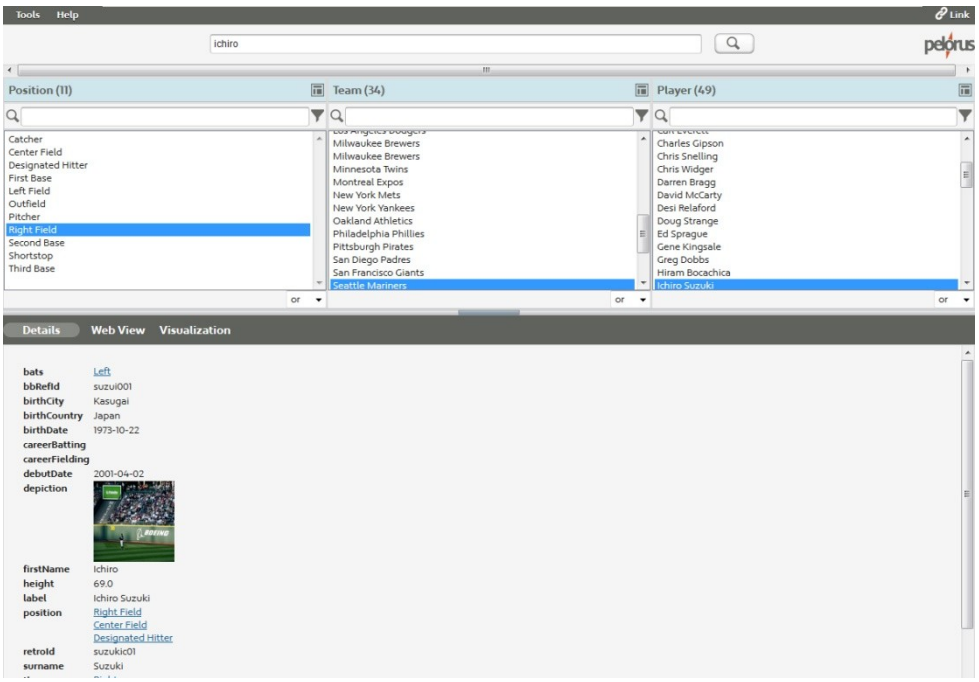



図 2.11 jSpace ブラウザの検索結果の例

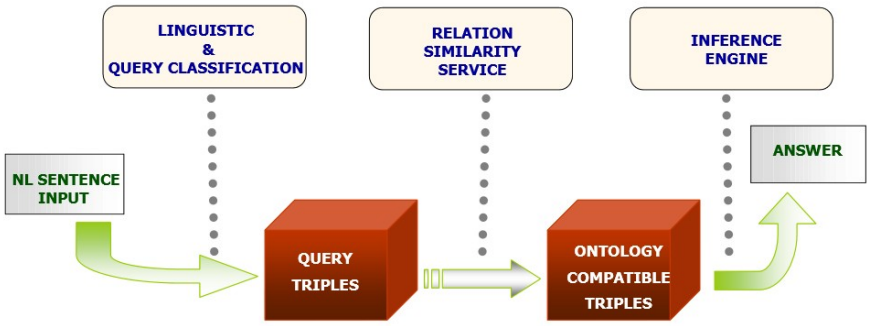



図 2.12 AquaLog の RDF トリプルを用いた自然言語検索の仕組み

（出所）AquaLog: An Ontology-Portable Question Answering System for the Semantic Web [29], p.548

実際に WolframAlpha[6]という自然言語と独自の知識ベースを用いて質問応答システムとして公開されているサービスも存在する.

WolframAlpha は自然言語の質問から，知識ベースの構造化データ内の答えと関連する情報を検索し，出力する．例えば，「2005 年にローマ教皇は何歳か？（How old was Bishop of Rome in 2005?)」という問いに対して，Google による検索結果は Wikipedia のフランシスコ教皇に関する記事である．一方，WolframAlpha による回答は「how old」というフレースから「age」を認識し，「Bishop of Rome」から「Pope Francis」を認識する．さらに，2005 年時点での年齢を算出し「68 years」という結果を出力する．また，「国民 1 人あたりの国内総生産が 21 番目に大きい国は？（What is the twenty-first country by GDP per capita?)」という検索文に対して，Google では，Wikipedia の国の国内総生産順リスト（一人当り為替レート)の記事が出力されるが，WolframAlpha では，GDP から capita を割った値のうち，21 番目の国である日本が出力され，その計算結果である「$46,720」という値も出力される.


図 2.13 WolframAlpha

[6] http://www.wolframalpha.com/

このように，現在 RDF データベースやオントロジーに対する自然言語検索技術に関する多くの研究がなされている．自然言語によるクエリは RDF のトリプル群に変換され，回答結果を出す際に必要になるのが検索対象となるデータと，そのデータが持つ関係も含めたメタデータである．領域オントロジーは，検索分野において専門的な概念体系を持つ参照情報として応用することができる．

- **ソーシャルブックマークへの応用**

領域オントロジーの応用例としてソーシャルブックマークサービスへの応用を紹介する．現在，セマンティックなタグ付けが可能である Faviki[7]というサービスが存在する．

Faviki は基本的にはソーシャルブックマークであり，はてなブックマークや del.icio.us などと似ているが，タグが Wikipedia の項目名と連動している点が異なる．参照しているデータベースは関連研究で述べた DBpedia をベースとして，タグ間の意味関係を用いたセマンティックなタグ付を実現している．つまり Faviki は，自由で乱雑な語彙が付与されやすいタグシステムに対して Wikipedia のフィルターを介すことで語彙の統制を行い，DBpedia が提供するオントロジーを利用し意味関係を抽出している．タグ間において関係が定義されているため，従来のソーシャルブックマークサービスよりも意味的な検索，整理が可能となる．本研究での構築対象である領域オントロジーは，より専門性に特化したコミュニティでのタグシステムへの利用へ応用することができると考えられる．

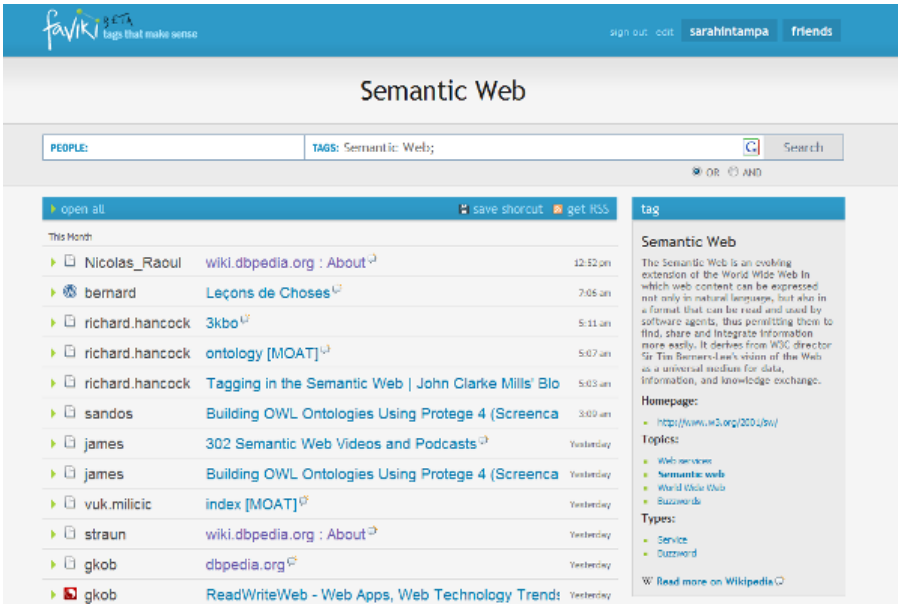



図 2.14 Faviki

[7] http://faviki.com/

## 2.3 Wikipedia

### 2.3.1 Wikipedia の概要

Wikipedia[8]は誰もが無料で自由に編集に参加できるオンライン百科事典であり，日本語版 Wikipedia は 2013 年 10 月現在，既に 87 万を超えるページ数が存在する[9]．Wikipedia のような知識形態は「集合知」とも呼ばれ，語彙の現在使用されている意味の定義が掲載されており，既存の辞書よりも真の意味を表すという考え方も広まってきている．Wikipedia は一般的な概念から最新の技術動向に関する記事まで幅広い分野の記事が網羅されていて膨大なコンテンツが存在し，さらに記事内の単語それぞれから対応したページへのリンクや言語リンク，関連項目のページへのリンクなど，Wikipedia 内の各ページ間でのハイパーリンクも充実している．Wikipedia ではカテゴリ階層，Infobox といった構造フォーマットを利用してこの膨大な量のコンテンツを整理している．Wikipedia のデータは記事本文，リンク構造などは GFDL (GNU Free Documentation License) [7]のライセンスの下にフリーで公開され，SQL や XML の形式でダウンロードすることができる．

### 2.3.2 Wikipedia の利点

Wikipedia はオントロジー構築の観点から見て有用な点が多いコーパスである．以下にその点を 3 つ述べる．


図 2.15 Wikipedia のトップページ

8 http://ja.wikipedia.org
9 http://ja.wikipedia.org/wiki/Wikipedia:日本語版の統計

**(1) URL による語彙の一意性確立**

URL によって語彙の一意性が確立されている点は, Wikipedia の大きな特徴の一つである．電子辞書では，通常一つの見出し語が一つのページに割り当てられており，その中で複数の意味について詳述される．一方，Wikipedia では一つの URL（ページ）に一つの概念が割り当てられており, 多義性が URL によって解決されている点が大きな特徴である．たとえば,「Football」は強いコンテキスト依存を持つ単語であり，アメリカンフットボールを示す場合もサッカーを示す場合もある．Wikipedia では，これら二つの概念は別々のページで管理されており，

”http://en.wikipedia.org/wiki/American_Football”， ”http://en.wikipedia.org/wiki/Football_%28soccer%29”という別々の URL が割り当てられている．

**(2) 辞書更新の即時性**

従来の辞書では，一般的な語からトップダウン的に追加されていくのが通常であり，一般的でない語や専門的な語は辞書に追加されるのが遅れる．もしくはいつまでも登録されないのが一般的である．しかし，Wikipedia では，インターネットを通じてリアルタイムに記事が編集・アップロードされ，リンクが構築されていくため，極めて即時性が高い．例えば，ある企業から最新の技術の発表があった数時間後には，エントリが生成され，その説明や詳細なスペック，画像などが他の語へのリンク付きで公開されたというケースもある．このような新しい概念に対する網羅性の高さはコーパスとしてみたときの重要な特徴の一つである．

**(3) コンテンツの網羅性**

従来，WWW を自然言語処理のコーパスとして利用する場合，その探索空間が膨大すぎることから，解析内容が発散もしくは偏ってしまうという問題があった．これを回避するためにはクローリングの方法を工夫するか大規模な並列システムを構築しなければならなかった．これに対し，Wikipedia は，一般的な概念から最新の技術動向に関する記事まで幅広い分野の記事が網羅されており，膨大なコンテンツ量が存在するものの，WWW の探索空間に比較するとそのリンク構造はサイト内で閉じられており，現実的な時間での解析が可能となる．

### 2.3.3 Wikipedia のデータ

Wikipedia を構成するデータとその構造のうち，主要なものを説明する．

- **記事ページ**

Wikipedia の構成単位としては最も主要なものである．電子辞書でいう見出し語の一つを記述しているページである．一つのページに一つの概念が割り当てられており，多義性が URL によって解決されている．図 2.16 に記事ページの例を示す．


図 2.16 記事ページの例

- **Infobox**

記事ページの中には，Infobox と呼ばれる構造を持つページもある．Infobox は，その概念の基本的な情報をテーブル形式でまとめたもので，動物，果物，国など種類ごとにテンプレートが存在する．図 2.17 は Infobox を持つ記事ページと Infobox である．


図 2.17 Infobox を持つ記事ページ（左）と Infobox（右）の例


図 2.18 カテゴリページ（左）とカテゴリ階層の概念図（右）の例

- **カテゴリとカテゴリ階層**

カテゴリには記事ページが割り当てられ，記事ページの分類・整理の役割を果たしている．記事ページは複数のカテゴリに属している場合もある．カテゴリ自体も親カテゴリ，子カテゴリが割り当てられ，ネットワークを形成しているが，ある部分だけを一種の木構造をとみなし，カテゴリ階層と捉えることができる．

- **一覧ページ**

記事ページの中には，「～の一覧」というタイトルのページがあり，そのページには，記事のタイトルをクラスと見立てた場合にそのインスタンスとなるものが項目として列挙されている．一覧ページは主に記事タイトル，その項目（インスタンス），インスタンスを整理・分別している目次から成り，図 2.19 のような形態をしている．Wikipedia が多くの分野の知識をカバーしているのと同様，一覧ページも数多くの分野のものが存在する．数多く存在する一覧ページは本研究でも着目している構造化情報であり，豊富なインスタンスの情報を抽出できる可能性を持っている．

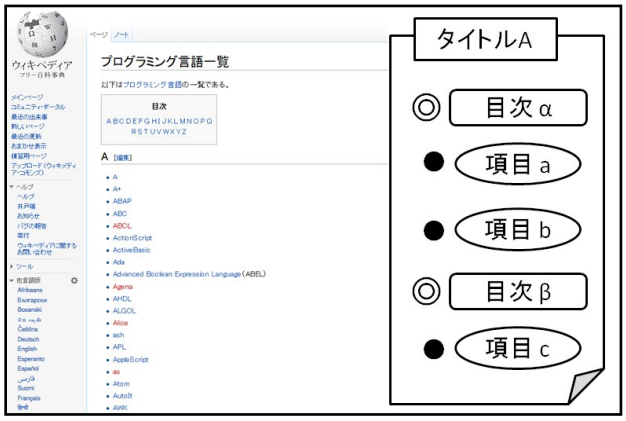



図 2.19 一覧ページ（左）とその概念図（右）の例

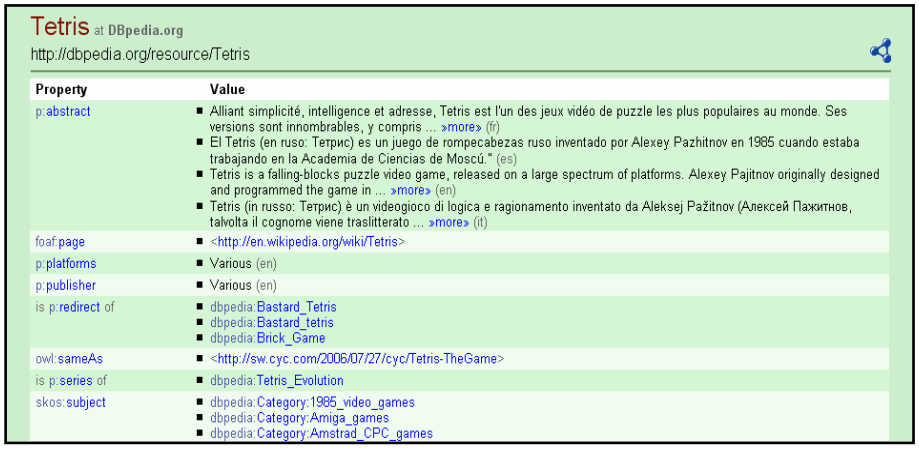



図 2.20 DBpedia の記事の例

## 2.4 Wikipedia 関連研究

現在，Wikipedia を情報資源としてオントロジーの構築を行っている研究は多い．本節では，国内外での，いくつかの代表的な研究を紹介する．

### 2.4.1 DBpedia

DBpedia [13]は, RDF を基盤とした記事タイトルについての膨大な量のメタデータベースを構築した．主に英語 Wikipedia の Infobox に着目しているが，外部リンクや所属カテゴリも応用している．また，記事のアブストラクトに関しては主要 11 言語で抽出している．しかし，抽出した情報は何もフィルタリングされておらず，ノイズも大量に含まれてしまっている．図 2.20 は DBpedia の一概念を表すページの一部である．

なお，DBpedia は多言語に対応している．2013 年 9 月時点での DBpedia の持つ言語別のデータ量を表 2.2 に示す．

表 2.2 2013 年 9 月時点の DBpedia のデータ

| 言語 | アブストラクトの数 | 言語 | アブストラクトの数 |
|---|---|---|---|
| English | 4,004,000 | Polish | 961,000 |
| Dutch | 1,405,000 | Swedish | 957,000 |
| German | 1,368,000 | Russian | 954,000 |
| French | 1,315,000 | Japanese | 825,000 |
| Italian | 980,000 | Portuguese | 736,000 |
| Spanish | 965,000 | Chinese | 653,000 |

DBpedia では，一意に定義した膨大な数の個体を他の RDF データベースの個体と結びつけることで Web 上に存在するデータの意味付けされたネットワークを構築している. 現在も様々な RDF データが公開されては DBpedia とのリンクが構築され, DBpedia はいわば Web 上のインスタンスデータのハブとなっている．このように，公開された RDF データで他のデータベースと結合されているものは LOD (Linked Open Data)と呼ばれ，現在のセマンティック Web 研究において一つの大きな流れとなっている.

DBpedia は非常に大規模なデータベースであるが，手動構築した 170 のクラスと 720 のプロパティを利用し，Infobox の構造をそのまま抽出している．手動構築のプロパティと Infobox からのプロパティは分離しており，Infobox からのプロパティの多くはオントロジー内で統合されていない．さらに，日本語固有の Wikipedia の記事には対応しておらず，日本語 LOD のハブとして利用するために十分とはいえない. 本家 DBpedia が英語版 Wikipedia のテンプレートをベースに構築しているのに対し，日本語版 Wikipedia を対象とし，独自でマッピング作業を行なっている DBpedia Japanese[10]も存在しているが，日本語特有のクラス階層を持っていない.

### 2.4.2 YAGO（Yet Another Great Ontology）

YAGO [30]は，概念階層部分として英語版汎用オントロジーの WordNet を利用し，末端のクラスに英語 Wikipedia に存在する膨大な量のインスタンス情報を付加したオントロジーである.

どの記事をどの WordNet クラスのインスタンスとするかの判断は，その記事がどの Conceptual Category と呼ばれるカテゴリに属しているかで決めている．Conceptual Category は，経験則に基づいた簡単なカテゴリ名の構文解析から定義している．ここで提案されている手法は英語においてのみ適用可能であり，所属カテゴリを利用してさまざまなプロパティを手動で定義し，記述している．YAGO は Wikipedia のインスタンス情報を主に利用している点が特徴である．図 2.21 は YAGO の一部である.

YAGO は関係の種類数としては is-a 関係も含めて 15 種しかなく，プロパティを設けているが，手動で 170 種程度であり，プロパティの定義域や値域については扱っていなかった.

[10] http://ja.dbpedia.org/

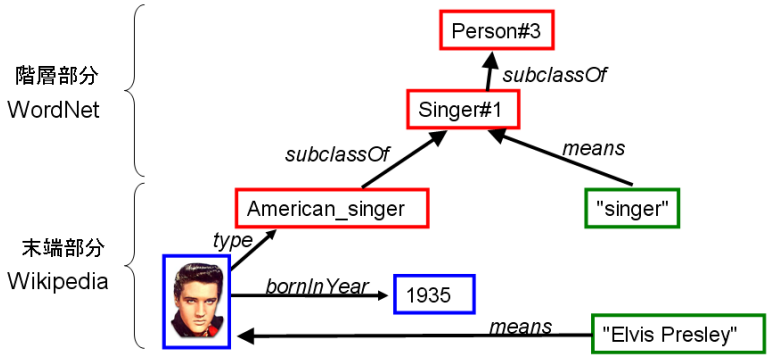



図 2.21 YAGO における階層関係の構築の例

YAGO2 および YAGO2s [31]では YAGO の知識ベースの拡張として，これまでの WordNet に Wikipedia のカテゴリを付加してオントロジーの拡張を行うだけでなく，GeoNames[11]などの Wikipedia 以外の情報資源を用いて，時空間的情報を抽出する事で，さらなるオントロジーの拡張を目指している．これら時空間的情報は wasBornOnDate や isLocatedIn といった関係を定義し，インスタンスとつないでおり，非階層関係となっている．非階層関係に着目し，時空間も含めた高度なオントロジーを構築しているが，これらの関係は手動で定義されており，プロパティの定義域や値域についても手動で定義されている．

### 2.4.3 Wikipedia からの上位下位関係抽出

Ponzetto ら [32]は，Wikipedia カテゴリから上位下位概念関係の抽出を試みている．手法としては，カテゴリリンクに以下のようなメソッドを適用することによって主に関係を抽出している．

- Category network cleanup
  Wikipedia 独自のノイズを取り除く
- Refinement link identification
  “Y X” - “X by Z”というカテゴリリンクを“X by Z” is-refined-by “Y X”と定義
- Syntax-based methods
  カテゴリ名の head（主要部）と modifier（修飾部）のマッチで分類

  British Computer Scientists is-a Computer Scientists

  Crime Comic not-is-a Crime（is-a ではないカテゴリ分けを指摘）

[11] http://www.geonames.org

- Connectivity-based methods
  複数形の head を持つカテゴリとそのサブカテゴリを is-a で結ぶ

Wikipedia カテゴリだけでなく，記事の自然言語文やハイパーリンクから機械学習により，上位階関係を抽出する研究もある．Wei ら [33]は，Wikipedia 記事内のハンパーリンクを，自動的に 13 次元の特徴ベクトルにマッピングし，Wikipedia の構造情報から抽出したトレーニングデータを基に，分類器を生成している．分類器はドメインごとの特徴を備えており，ドメイン固有の上位下位関係を発見する事が可能である．実際に，いくつかのドメインに分類器を適用した結果，辞書と構文パターンによるアプローチに比べ，パフォーマンスの向上が見られる．

### 2.4.4 Wikipedia の Infobox を用いた意味関係抽出

Wu ら [34]は，Wikipedia の Infobox が持つテンプレートに着目し，Infobox テンプレートを WordNet のクラス階層に写像することで，is-a 関係を構築している．写像した Infobox テンプレートが持つ各プロパティは is-a 関係により継承される．各プロパティが Infobox テンプレートという定義域を持ち，継承される点で高度なオントロジーと言えるが，プロパティ自体は Infobox からの情報のみであり，さらにプロパティのタイプについては検討していない．

Xu ら [35]は，Wikipedia の Infobox からトリプルを抽出する際に，欠けてしまった要素間のリンクを発見し，補完する手法を提案している．DBpedia などの Infobox からのプロパティ抽出では，Wikipedia の記者に依存し，Infobox 内にハイパーリンクをつけていない，余分な注釈を入れているなどの理由により，トリプルを抽出できないことが多い．本手法は各プロパティの値となる部分の特徴を重みとして取得し，学習することで，プロパティの値となる要素を予測し，欠けてしまった要素を補完している．

### 2.4.5 日本語版 Wikipedia を用いた研究

日本語版 Wikipedia を用いて日本語語彙体系を拡張する研究も行われている[36, 37]．柴木らは [37], 日本語版 Wikipedia を用いて日本語語彙体系を拡張する研究を行っている．日本語語彙大系を上位階層とし，日本語版 Wikipedia のカテゴリと対応付けることで is-a 階層を構築し，さらに Wikipedia の見出し語に着目し，記事からインスタンスを抽出することで汎用オントロジーを構築する手法を提案している．本論文と同様に，後方文字列照合を用いて精度の高い is-a 階層とインスタンスを抽出しているが，非階層関係については言及されていない．

隅田らは [38], Wikipedia の記事構造に機械学習によるフィルタリングを用いることで，大規模な上位下位関係にある単語ペアの獲得を行っている．獲得された単語ペアにおける

上位下位関係の精度は高いが独立しており，本論文のように階層構造になっていない．また，クラスやインスタンスの区別もされていない．

### 2.4.6 関連研究の総括

クラスーインスタンス関係および階層関係に焦点が当てられたものが多く，プロパティを含むオントロジーを構築している研究は少ない．また，プロパティを含むオントロジーも Wikipedia の Infobox のみに着目しており，他の構造を利用したプロパティ構築研究は少なく，プロパティのタイプやプロパティ間の関係にまで着目した研究は見られない．さらに，日本語版 Wikipedia からのオントロジー構築研究については，非階層関係の抽出に焦点を絞った研究は少ない．

## 2.5 Linked Open Data

LOD (Linked Open Data)とは今まで互いに関連していなかったデータ同士を Web の仕組みを利用することによって繋げ，データをつなげることに対する敷居を低くするための試みである．LOD は Semantic Web における URIs と RDF を利用することにより，１つ１つのデータ，情報，そして知識をつなげるために実践的な方法の１つである．ここでは Linked Open Data 概要と現状について述べる．

### 2.5.1 Open Government Data の始まり

LOD はデータを開示しようという世界的な流れの中で生まれ, 政府の保有する大量のデータを如何にして使いやすい形で公開できるかという活動から始まった．この活動を Open Government[12]と呼ぶ．何層にも重なっているデータ，特定の領域についてごく詳しく述べられているデータをどのように公開したら良いかというのはデータの開示という側面からは１つの大きな問題である．そこでセマンティック Web の技術を利用してオープンに結び付けられたデータを作ろうというのが，LOD のビジョンである．

2007 年，30 の Open Government の団体がアメリカ合衆国カリフォルニア州の Sebastopol で会合を行い，Open Government Data Principles [39]を作成した．ここで定められた 8 つの原則は次のとおりである．

[12] http://www.whitehouse.gov/open

(1) データは完全でなければならない.
(2) データは一次情報でなければならない.
(3) データは直ちに開示されなければならない.
(4) データはアクセス可能でなければならない.
(5) データは機械処理可能でなければならない.
(6) データは平等にアクセス可能である.
(7) データの形式は専売的ではないものが望ましい.
(8) データはライセンスフリーでなければならない.

これらは Open Government Data Principles の世界標準となっており, オーストラリア, ニュージーランド, 欧州, 北アメリカで活動が始まった. 今日ではアジア, 南アメリカ, アフリカでも Open Government Data Principles の活動が広まりつつある.

### 2.5.2 Open Data から Linked Open Data へ

Open Government Data は情報とデータをオープンにしてそれらの再利用を高める役割がある. Open Government Data を具体的な事例としつつ, より汎用的に色々なデータを Open Government Data のような形で公開し再利用していくことはできないだろうかという議論が進んだ.

Open Data の恩恵を十分に受けるには, 情報とデータを新しい知識を生み出すコンテキストにおき, かつ, 魅力的なサービスやアプリケーションが存在する必要がある. LOD の活動を促進させるためには, 各企業や特定のドメインの中で閉じられているデータを最利用する際の情報管理と情報の統合のメカニズムが重要になる.

Tim Berners-Lee は 2010 年のワシントン DC での Gov2.0 Expo において Open Government Data 及び LOD のためのプレゼンテーションを行った. 2006 年に公開された Linked Data Design Issue [40]において, Linked Data の条件として以下の 4 つが述べられている.

(1) モノの名前に URI を利用する
(2) HTTP URI で名前を参照できる
(3) URI を参照した時, 関連する情報を提供できるように, Web 標準の技術（RDF/SPARQL 等）を使う
(4) より多くの関連情報を引き出せるよう, 外部データへのリンク（URI）を含める

また, Michael Hausenblas がそのプレゼンテーションの核となる部分をまとめ, 5 つ星モデルを提唱した. 表 2.3 に 5 つ星モデルを示す.

表 2.3 Linked Data のための 5 つ星

| | |
|---|---|
| ★ | 情報はオープンなライセンスにおいてweb上で公開されている．（フォーマットは問わない） |
| ★★ | 情報は構造化されたデータとして公開されている，（例えば，スキャンされた画像の代わりにExcelを用いる） |
| ★★★ | 専有的でないフォーマットが利用されている．（例えば，Excelの代わりにCSVを用いる） |
| ★★★★ | 利用者が個々のデータを選ぶためにURI識別子が利用されている． |
| ★★★★★ | データは所与のコンテキストのデータと結び付けられている． |

現在も LOD は広がり続けており，様々な領域において増々重要な位置を占めるようになってきている．LOD は既に多くの著名な組織，プロダクト，そしてサービスのために利用されており，ポータルサイトやプラットフォームなどのインターネットをベースとしたサービスやシステムを構築するために活用されている．

図 2.22 は, Linked Open Data の近年の成長の様子を示した Linked Open Data Cloud[13] である．総データセット数は 295 である．

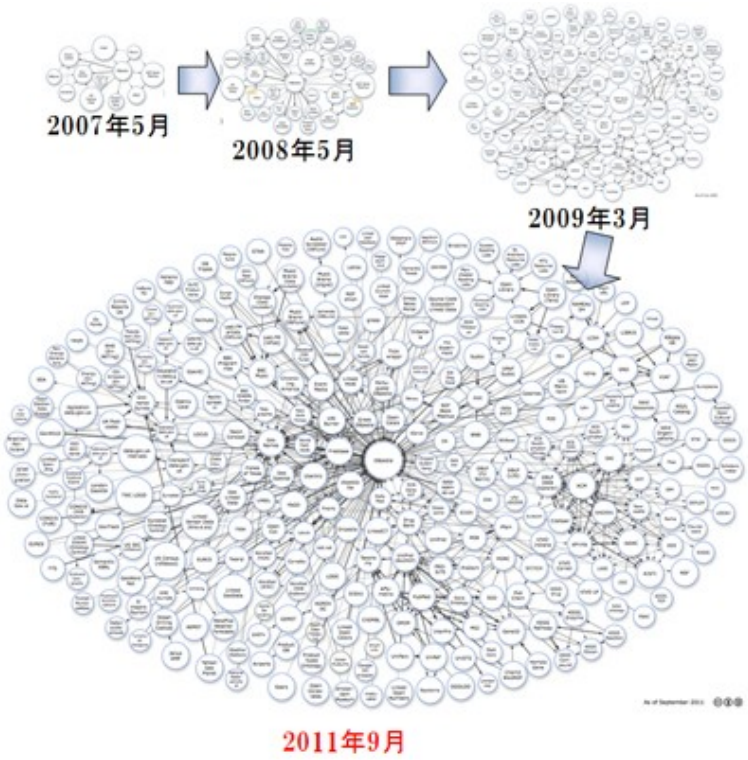



図 2.22 近年の Linked Open Data の広がり

（出所）http://lod-cloud.net/versions/2011-09-19/lod-cloud.html

[13] http://lod-cloud.net/

### 2.5.3 日本における Linked Open Data の現状

アメリカや欧州においては LOD の活動が広がり，国内における様々な普及活動の効果もあり，普及し始めている．日本における LOD の活動を紹介し，現状をまとめる．

- **DBpedia Japanese**[14]

前述した DBpedia の日本語版である. 本家 DBpedia は英語版 Wikipedia のテンプレートをベースに構築しているため，日本語特有の記事については対応していなかった．DBpedia Japanese では，日本語版 Wikipedia を対象とし，独自でマッピング作業を行なうことで，日本語特有の記事にも対応可能であるが，日本語特有のクラス階層を持っていない．

- **CiNii（国立情報学研究所）**[15]

論文のデータベースである CiNii でも RDF データを公開している. CiNii とは論文や図書・雑誌などの学術情報で検索できるデータベース・サービスのことである．RDF で取得できる情報としては，論文情報，著者情報，図書・雑誌情報，図書館情報がある．

- **LOD チャレンジ Japan**[16]

日本国内での LOD に関するコンテストであり，第 1 回大会は 2011 年末から 2012 年初めにかけて開催され，日本国内初の取組みであった．このチャレンジでは一般の応募者から LOD 活用のためのアイデア，及び，実際の LOD データ，LOD を利用したアプリケーションを募集している．2012 年に第 2 回，2013 年に第 3 回と毎年開催されている．

- **LODAC**[17]

国立情報学研究所が保有しているデータを LOD 化するプロジェクトが LODAC [42]である． LODAC プロジェクトでは学術情報の LOD 化を目的としおり，博物館情報を対象に LOD 化を行なっている．情報源としては 14 館の博物館資料，及び，日本美術シソーラス[18]，国指定文化財データベース[19]，文化遺産オンライン[20]，DBpedia Japanese を利用している

- **NDLSH（国立国会図書館）**[21]

国立国会図書館典拠データ検索サービスである Web NDL Authorities が公開されてい

14 http://ja.dbpedia.org/
15 http://ci.nii.ac.jp/
16 http://lod.sfc.keio.ac.jp/challenge2012/
17 http://lod.ac/
18 http://www.tulips.tsukuba.ac.jp/jart/mokuji/index.html
19 http://www.bunka.go.jp/bsys/index_pc.asp
20 http://bunka.nii.ac.jp/Index.do
21 http://id.ndl.go.jp/auth/ndla

る．このサービスは国立国会図書館の典拠データを一元的に検索・閲覧・ダウンロードすることができる．収録データは日次で更新されており，国立国会図書館件名標目表(National Diet Library Subject Headings: NDLSH)に基づいている．件名標目とは，目録を検索する際の手がかりとして資料の主題をことばで表現したものである．Web NDL Authorities においては NDLSH の収録範囲に該当する全件データを RDF/XML・TSV の形式でダウンロードすることが可能となっている．

図 2.23 は Linked Open Data Initiative[22]がまとめた 2013 年 10 月時点の日本語 LOD クラウドである．総データセット数は 2008 年時点の LOD クラウドと同程度であり，今後のさらなる普及が期待できる．

### 2.5.4 Linked Open Vocabularies

LOV (Linked Open Vocabularies)[23] [16]は，LOD の語彙に着目した取り組みである．各 LOD で使用されている語彙を集めて，語彙の検索を可能にすることで，新たな LOD を構築する際に語彙の再利用を促す取り組みである．図 2.24 は LOV における名前空間の集合であり，2013 年 10 月時点で 370 の名前空間から語彙が登録されており，これらには FOAF (Friend of a Friend) [43]や SKOS (Simple Knowledge Organization System) [44]といった代表的な語彙を含んでいる．

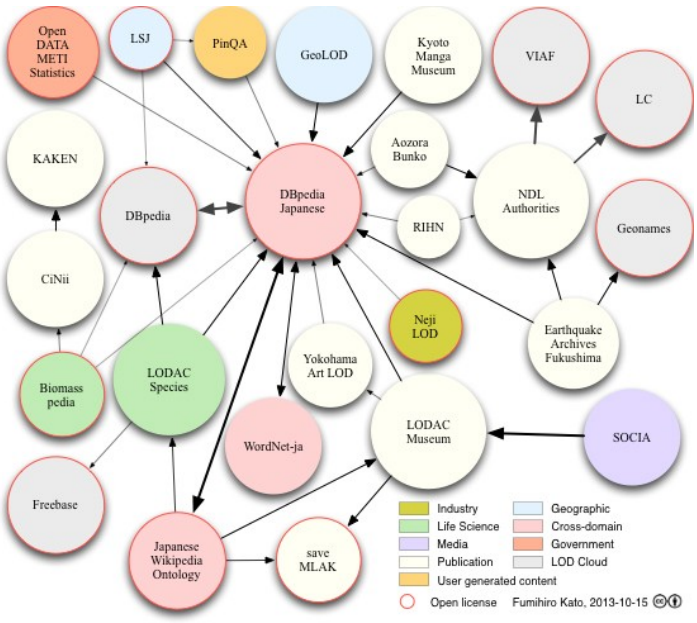



図 2.23 日本版 LOD クラウド

(出所) 日本語版 Linked Data クラウド図, http://linkedopendata.jp/?p=411

[22] http://linkedopendata.jp
[23] http://lov.okfn.org/dataset/lov/

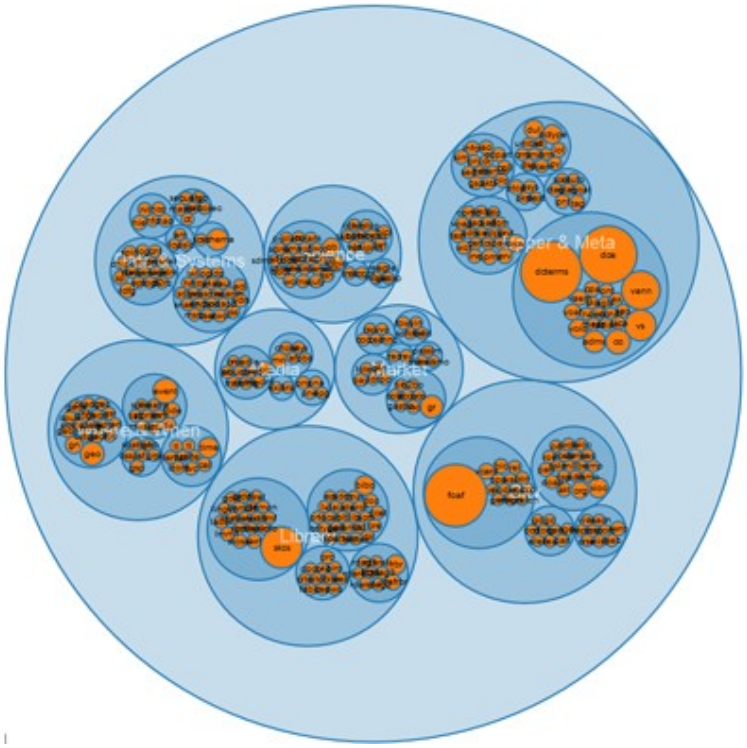



図 2.24 Linked Open Vocabularies 名前空間の全体像

（出所）Linked Open Vocabularies (LOV), http://lov.okfn.org/dataset/lov/

しかしながら，LOD を構築する際に，新たに語彙を作成する方が，目的に合致する語彙を見つけてくるよりもはるかに容易である．そのため，標準語彙と呼ばれる，既に普及している一部の語彙を除いて，再利用されているケースは少ない．加えて，国内では LOV に相当する取り組みがまだ存在しておらず，日本語の標準語彙というものがないため，今後さらに国内の LOD が広がることを想定すると，LOD 構築者にとって障壁となりうる．

- **FOAF (Friend of a Friend)**

Friend of a Friend[24] [43]は友達の友達・・・という人物間の繋がりをメタデータとして表現することで，人物の属性や関係を理解可能にする試みである．名前や年齢といった一般的な属性だけでなく，関心領域，ホームページなどといった，人を描写するための様々な語彙を定義している．

- **gr (GoodRelations)**

Good Relations[25] [45]は e-コマースのための語彙を定義している．商品や製品のブランドや店舗の所在地や営業日などの属性が定義されており，商品レビューの検索や分析の際に有用なメタデータを付加することができる．

[24] http://www.foaf-project.org/
[25] http://www.heppnetz.de/projects/goodrelations/

- **gn(GeoNames)**

GeoNames[26] [46]は地理や位置情報のための語彙を定義している．所在地や座標だけでなく，近隣施設や隣接施設などのメタデータを付加することで，施設情報の空間的な検索が可能になる．

- **schema (schema.org)**

schema.org[27] [47]は Google，Yahoo!，Microsoft の 3 社が共同でセマンティック Web を導入しやすい環境作りのために発足したイニシアティブである．HTML に埋め込むことで，検索エンジンがマークアップを通してより質の高い検索結果を返せるようになることを目指している．大分類として 7 つのクラスがあり，それぞれが幾つかの下位クラスをもっている．これらのクラスを定義域として非常に多岐にわたる語彙があり，広く汎用的に利用可能な語彙となっている．このため，HTML だけでなく LOD の語彙としても広く利用されている．

## 2.6 まとめ

本章では，はじめにオントロジーの定義を述べ，代表的な汎用オントロジーとオントロジーの応用例について述べた．また，Wikipedia と Wikipedia を用いたオントロジー構築についての既存研究を述べた，最後に Linked Open Data と Linked Open Vocabularies について述べ，国内外のいくつかの代表的な取り組みを紹介した．

オントロジーの有用性が高まる一方，いくつかのオントロジー構築支援ツールは存在しているが，オントロジーの手動構築にかかるコストは大きな課題となっている．その課題を解決するため，フリーテキストからのオントロジー自動構築が試みられてきたが，自然言語理解に限界があり，実用レベルに到達しない．そのため，半構造情報を有する情報資源からオントロジーを自動的に構築する方法が提案されており，その情報資源として Wikipedia は大きな注目を集めている．しかしながら，Wikipedia からのオントロジーの自動構築に関する研究は，クラス階層構築に焦点を当てているものが多く，プロパティの定義域・値域を含めたクラススキーマ階層を持つような質の高いオントロジーを構築する研究は少ない．

セマンティック Web の分野でのオントロジーの応用に関しては，Linked Open Data が注目を集めており，様々な領域の RDF のデータベースが共有・公開されている．しかしながら，国内では欧米における DBpedia のようなハブが確立されておらず，データベース数も依然少ない．加えて，Linked Open Vocabularies のような取り組みが存在していないため，今後国内 LOD が普及する上で大きな課題となっている．

[26] http://www.geonames.org
[27] http://schema.org/

# 第3章　日本語Wikipediaオントロジーの自動構築

本章の内容は，文献[48, 49]に基づいている．

## 3.1　概要

大規模なオントロジーの構築は情報検索やデータ統合において有用であり，日本語の大規模オントロジーとしては日本語WordNet [3]や日本語語彙大系［4］などが存在している．しかし，これらは手動で構築されており，構築コストが大きい．オントロジーの手動構築には，膨大な時間がかかり，間違いを起こしやすく，オントロジーの保守や更新が困難という問題がある．そこで，近年，オントロジー工学のコミュニティは，オントロジー開発コストを削減するために，オントロジー学習（Ontology Learning）と呼ばれる，（半）自動的にオントロジーを構築する手法，方法論，アルゴリズム，ツールなどの研究開発に取り組んできた．特に，フリーテキストからのオントロジー学習に関しては，機械学習，知識獲得，自然言語処理，情報検索など，様々な専門分野の手法を組み合わせた手法がこれまで数多く提案されている［5]．しかしながら，非構造情報資源であるフリーテキストと構造情報資源であるオントロジーの間のギャップは大きく，高精度で，大規模なオントロジーを構築することは困難であるのが現状である．

一方，近年，Web 上の百科事典である Wikipedia が，新たな情報資源として注目を集めている［10]．Wikipedia は語彙網羅性，即時更新性に優れており，半構造情報資源であることからフリーテキストと比べてオントロジーとのギャップが小さい．そのため，Wikipedia からのオントロジー学習研究が近年，盛んに行われている．しかしながら，Wikipedia はユーザ参加型という性質上，厳密な体系化が行われていないため，Wikipedia からのオントロジー学習にも，多くの課題が存在している．加えて，Wikipedia を用いたオントロジー構築の多くはクラス階層構築に焦点を当てており，プロパティの定義域・値域を含めたクラススキーマ階層を構築する研究は少ない．

そこで本論文では，日本語版 Wikipedia をリソースとして，概念および概念間の関係を抽出する事で，大規模で汎用的であり，クラススキーマ階層を持つオントロジーを自動構築する手法の提案を行う．

以降，本章の構成は次のとおりである．3.2 節では，日本語 Wikipedia オントロジーの概要を示す．3.3 節では，日本語 Wikipedia オントロジーの自動構築手法を説明する．3.4 節で各関係抽出手法についての実験結果と考察と，オントロジー全体の評価と考察を述べる．最後にまとめと今後の課題について述べる．

## 3.2 日本語 Wikipedia オントロジーの概要

日本語 Wikipedia オントロジーは日本語版 Wikipedia から, 以下に示す関係とタイプを抽出し自動構築される. ただし, () 内は, 抽出した関係に対応する, OWL [19], RDFS[24], RDF [11], 日本語 Wikipedia オントロジーで定義した語彙(クラス及びプロパティ) を示す. 図 3.1 に日本語 Wikipedia オントロジーの概略図を示した.

(1) is-a 関係(rdfs:subClassOf)
(2) クラス-インスタンス関係(rdf:type)
(3) プロパティ名とトリプル(以下のプロパティタイプを含む)
  (A) オブジェクトプロパティ(owl:ObjectProperty)
  (B) データタイププロパティ(owl:DatatypeProperty)
  (C) 対称関係プロパティ(owl:SymmetricProperty)
  (D) 推移関係プロパティ(owl:TransitiveProperty)
  (E) 関数関係プロパティ(owl:FunctionalProperty)
  (F) 逆関数関係プロパティ(owl:InverseFunctionalProperty)
(4) プロパティ定義域(rdfs:domain)
(5) プロパティ値域(rdfs:range)
(6) プロパティ上位下位関係(rdfs:subPropertyOf)
(7) 上位下位関係(jwo:hyper)
(8) 関連語・同義語(jwo:nearly)
(9) 動詞とプロパティの関係(jwo:verb)

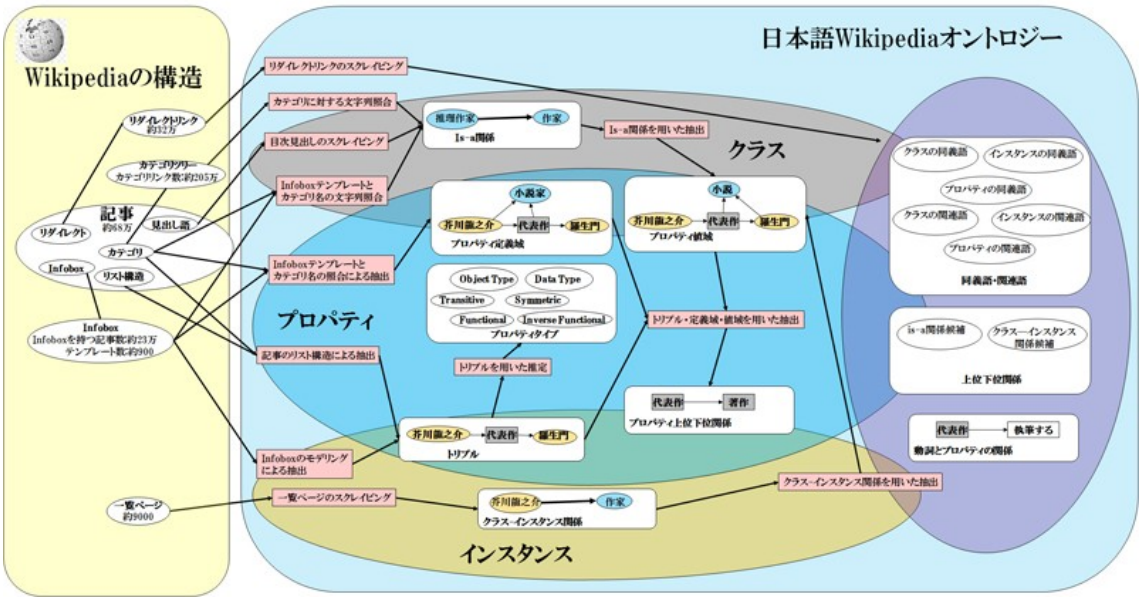



図 3.1 日本語 Wikipedia オントロジーの概略図

# 3.3 日本語 Wikipedia オントロジー構築手法

## 3.3.1 is-a 関係の抽出

Wikipedia には，記事の分類を目的とした階層的なカテゴリが存在する．しかし，下位カテゴリと上位カテゴリとの関係は，性質の継承という観点から捉えた is-a 関係ではない場合が多々見られる．実際にカテゴリ階層がどの程度 is-a 関係を持っているかを知るために，日本語版 Wikipedia のカテゴリ関係のダンプデータから 1,000 個の標本を抽出し，下位カテゴリと上位カテゴリがどのような関係になっているのか予備実験を行った．その結果，is-a 関係は 50.9%であった．誤りの例としては，アニメキャラクターとその声優が階層構造になっているものや，オリンピックメダリストーオリンピックといった人物と事象が階層関係になっているものなどが多く見られた．また，クラスーインスタンス関係も全体の 10.1%であり，「スーパーマリオ」や「SONY」といった「ゲーム作品」や「企業」クラスのインスタンスも存在した．

以上の予備実験の結果から，Wikipedia カテゴリ階層をそのまま is-a 関係に利用する事は困難である．そのため，本論文では is-a 関係を以下の 3 つの手法により構築する．

(1) カテゴリ階層に対する文字列照合
(2) カテゴリ名と Infobox テンプレートの照合
(3) 目次見出しのスクレイピング

**(1) カテゴリ階層に対する文字列照合**

あるカテゴリから相対的に下位に存在するサブカテゴリは，増加した記事を細分化するために作成されるという性質から，上位カテゴリの名称を含む複合語で形成される場合が多い．例えば「原子力ー原子力発電所」や「ソフトウェアーフリーソフトウェア」といった階層である．前者は性質の継承という観点からみた is-a 関係としては不適切な関係であるが，後者は is-a 関係に相当する関係となっている．本論文では，is-a 関係を抽出するためのカテゴリ階層の複合語に対する文字列照合として，「後方文字列照合」と「前方文字列照合部除去」を行う．

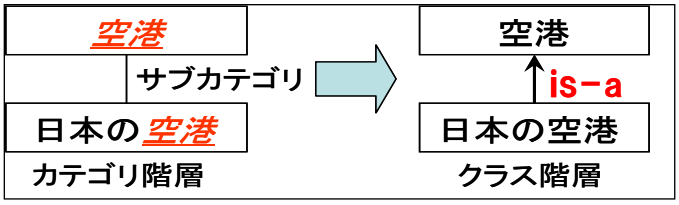



図 3.2 後方文字列 照合

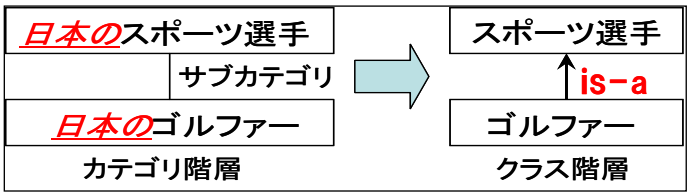



図 3.3 前方文字列字照合部除去

- **後方文字列照合**

後方文字列照合とはカテゴリ階層を構成する親カテゴリ名と子カテゴリ名とを比較し，子カテゴリ名が“任意の文字列＋親カテゴリ名”となっているものを抽出する手法である．図 3.2 の例では“空港”という文字列の後方文字列照合により，“日本の空港” is-a “空港”という関係を得ることができる．この手法は，Ponzetto ら [32]によって実践されている手法である．また，この手法は 1 世代の親－子カテゴリリンクだけではなく，N 世代離れたカテゴリリンクにまで適用することが可能である．例としては，“心理学”－“精神医学”－“分析心理学”といった親－孫のリンクからは“分析心理学” is-a “心理学”というリンクを抽出できる．

また，2 世代のカテゴリリンクまで検索の対象を広げて文字列マッチングを適用し，クラス階層の抽出を行った．

- **前方文字列照合部除去**

前方文字列照合部除去とは親カテゴリ名と子カテゴリ名とを比較し，親カテゴリ名と子カテゴリ名で“任意の文字列＋の”という部分が先頭から一致しているものを抽出，照合部を除去する手法である．図 3.3 の例では“日本の”という前方文字列照合部を除去することにより，“ゴルファー” is-a “スポーツ選手”という関係を得ることができる．この手法は，文字列の重複に依存しない is-a 関係を取得できる点が大きな利点である．この手法も N 世代離れたカテゴリリンクにまで適用することが可能である．

### (2) カテゴリ名と Infobox テンプレートの照合

Infobox は，テーブルを利用して Wikipedia の記事の属性（Wikipedia では主に“項目”と呼ばれている）と属性値を整理して表示しているもので，記事の中にしばしば掲載されている．ここで使用される項目が，ドメインごとにある程度フォーマット化されているということが大きな特徴である．例えば「Java」の記事に掲載されている Infobox には“開発者”や“プラットフォーム”などの項目とそれぞれに対応する値が記述されており，この“開発者”や“プラットフォーム”という項目は，Infobox のテンプレート“プログラミング言語”で定められている．

本手法は，各 Infobox の持つ抽象的なテンプレート名と，領域によっては多くの具体的な概念を持つカテゴリ名との関係に着目する．テンプレート名とカテゴリ名の照合を行い is-a 関係を抽出する．is-a 関係の抽出手順は以下の①～④の通り行う．

① カテゴリとテンプレートの情報を MySQL に格納
② カテゴリ名とテンプレート名の単純文字列照合
③ 照合したカテゴリ以下に存在するサブカテゴリ名と，照合したテンプレートを持つ記事が所属する全てのカテゴリ名とのマッチング
④ マッチングによって得られたサブカテゴリ名をテンプレート名と is-a 関係が成り立つとして抽出

図 3.4 に，Infobox テンプレートとカテゴリ名の照合による is-a 関係抽出の具体例を示す．図 3.4 は，「楽器」テンプレート，「楽器」テンプレートを用いて作成した Infobox を持つ「ピアノ」と「フルート」記事，「ピアノ」と「フルート」記事が所属するカテゴリ，カテゴリツリーとそれらの間の関係を示している．まず，「楽器」テンプレート名とカテゴリツリーの照合を行い，楽器カテゴリを同定する．次に，楽器カテゴリのサブカテゴリ名と「ピアノ」および「フルート」記事が属するカテゴリ名を照合する．その結果，「鍵盤楽器」is-a「楽器」，「ピアノ」is-a「鍵盤楽器」，「木管楽器」is-a「楽器」，「フルート」is-a「木管楽器」の 4 つの is-a 関係が抽出できる．ここで，「ピアノ」is-a「鍵盤楽器」と「フルート」is-a「木管楽器」は，文字列照合では抽出できない is-a 関係である．以上の手順を行うことによって，3·2·1(1)で述べた文字列の特性を利用した「カテゴリ階層に対する文字列照合」では抽出することのできなかった is-a 関係を抽出できる．それに伴い，正しくない is-a 関係を多く持つ Wikipedia カテゴリツリーの洗練が可能になると考えられる．

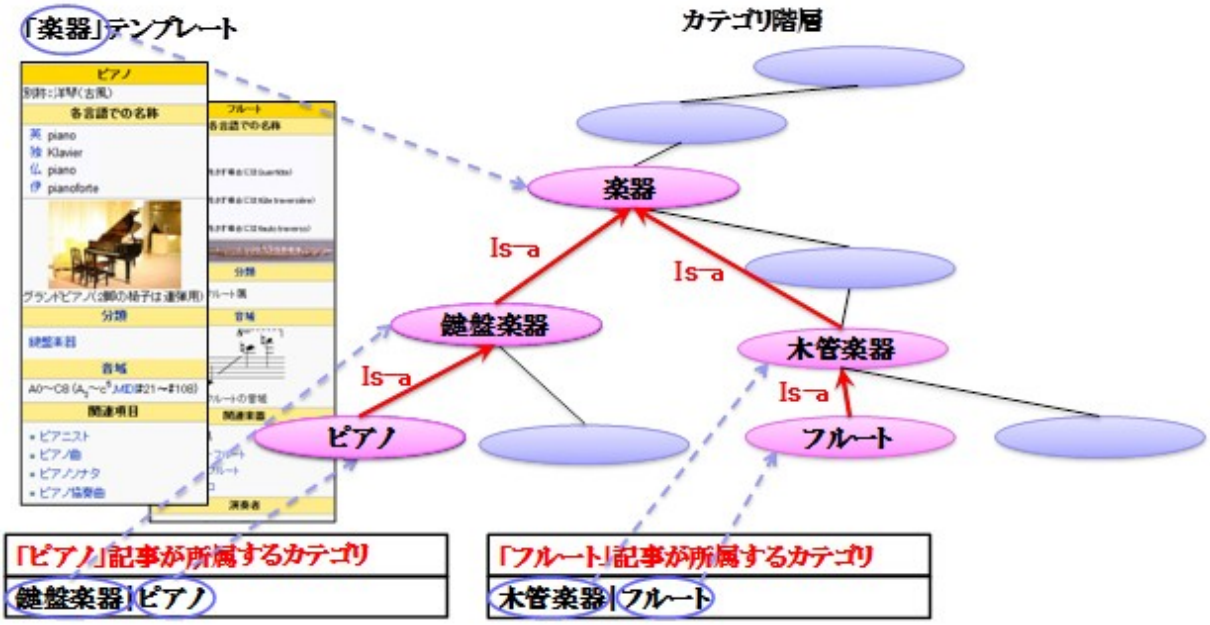



図 3.4 Infobox テンプレートとカテゴリ名の照合

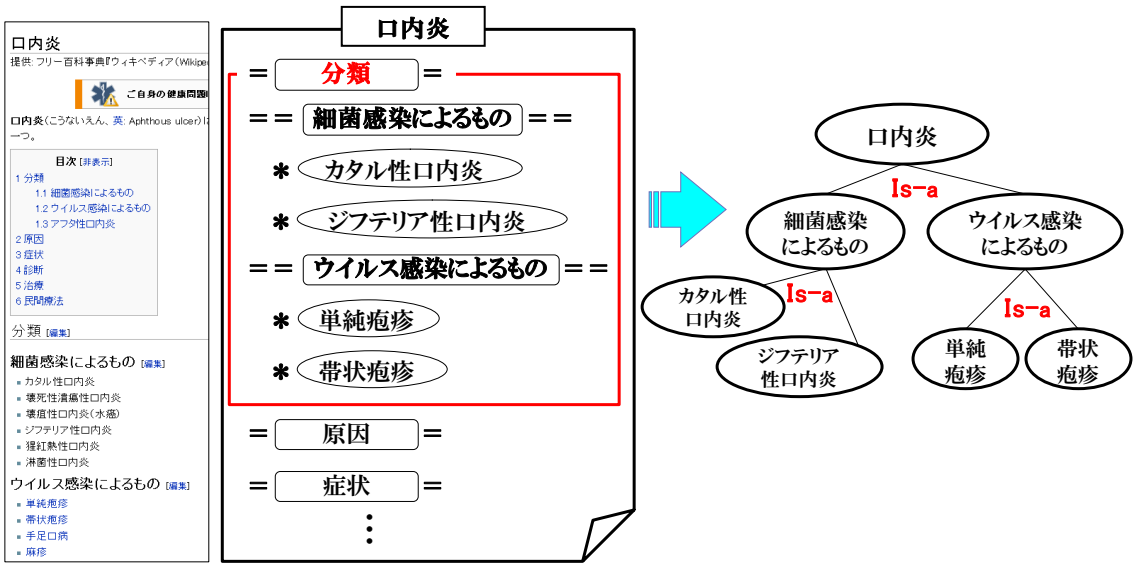



図 3.5 目次見出しのスクレイピングによる is−a 関係の抽出

### (3) 目次見出しのスクレイピング

Wikipedia の記事には目次が存在する．この目次となる見出し語に着目し柴木らはインスタンスの抽出を行っていた [37]．本手法では，カテゴリ階層以外からの is-a 関係抽出法として「目次見出しのスクレイピング」を用いて，Wikipedia 記事の見出し構造から is-a 関係の抽出を試みる．Wikipedia 各記事の目次は階層化された記述が見られるが，その上位と下位の関係は必ずしも is-a の関係になっているわけではない．例えば「口内炎」という記事は“分類－細菌感染によるもの” という目次が存在し，“細菌感染によるもの”を見出し語として“カタル性口内炎”，“壊死性潰瘍性口内炎” といった項目が箇条書きされている．一方，「カクテル」という記事は“カクテルの用語－カクテルの用具”という目次が存在し，“カクテルの用具”を見出し語として“シェイカー”，“マドラー” といった項目が箇条書きされている．前者はクラスの is-a 関係だが，後者はクラスーインスタンス関係となっている．そこで，本論文では目次が「分類」や「種類」という語を含む場合は is-a 関係となりやすいことに着目し，「分類」や「種類」という語を含む目次を持つ記事をスクレイピングすることで is-a 関係の抽出を行う．具体的な手法を表しているのが図 3.5 である．「口内炎」記事は“分類”という単語が目次となっており，そこから階層化された記述が見られ，“ウイルス感染” と“細菌感染”という分類を弁別属性として，“カタル性口内炎” などが箇条書きされている．これらは口内炎を最上位の概念として is-a 階層を構築している．

## 3.3.2 クラス-インスタンス関係の抽出

Wikipedia は物事のリストが記述された記事，一覧記事をもつ．例えば，「言語の一覧」には世界の言語のリストが記述されている．文章表現を工夫したり細かな事実を確認した

りする必要がないために，一覧記事の執筆者は非常に多いと考えられる．このため情報量は豊富であり，かつ記述形式がある程度統一されているため，一覧記事から大規模なインスタンスを収集することが可能である．

しかし，一覧記事はインスタンスを収集するためには不要な情報を含んでいる．本実験では一覧記事から不要な情報を取り除くための手法としてスクレイピングを行うことによって，インスタンスの収集を試みる．ダンプデータの pages-articles.xml は全記事の xml テキストファイルであり，図 3.6 が示すようになっている．

本項では，一覧記事のソースに対してスクレイピングを行う．以下(1)～(7)で，スクレイピングの具体的な手法を解説する．

**(1) 大まかな不要情報の除去**

図 3.6 の a の page タグ title タグを利用して，一覧記事のテキスト以外を除去し，title タグ部分も除去する．一覧記事では d のように ‘＊’または‘＃’から始まる行（以下，‘＊’行と呼ぶ）にインスタンスが記述されており，c のように‘=’で囲まれた部分にはインスタンスを分類する単語が記述されている（本論文ではこれを目次見出しと呼ぶ）．この c，d を残し，b の‘＊’や‘=’以外から始まる行を除去する．図 3.6 中の“[[ ]]”は，Wikipedia の内部リンクを表している．

一覧記事の中には，‘＊’や‘＃’を利用した箇条書きによる記述ではなく，テーブル形式でインスタンスを列挙している記事がある．例としては“内閣総理大臣の一覧”がある．この記述形式は多数存在し，インスタンスがどのように列挙されているかのパターン化が難しい．本実験では，テーブル形式でインスタンスが記述されている一覧記事からの抽出は行わない．

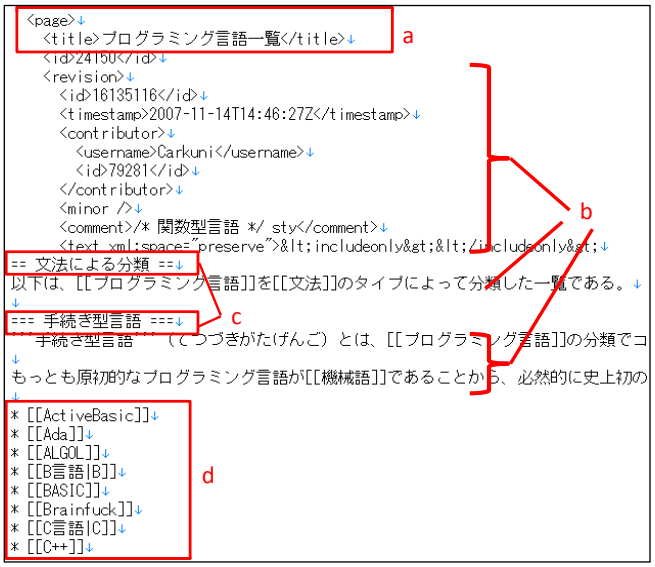



**図 3.6 一覧記事ソーステキストの一部**

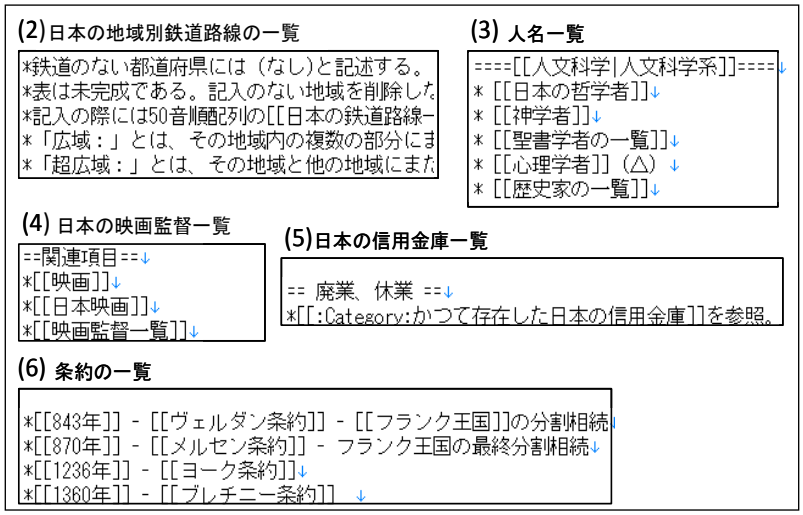



図 3.7 一覧記事の不要な情報の例

以下からは図 3.7 に示される例のような，(1)で抽出した‘＊’行に存在する，正しいインスタンスを含んではいない行を除去していく．以下(2)～(6)は，図 3.7 の例に対応している．

### (2) 一覧記事の説明に使用される‘＊’行を除去

一覧記事の多くは，インスタンスを列挙する前に一覧の内容の説明を加えたり，類似した一覧記事へのリンクを紹介したりしている．図 3.7 のように，それらの記述の中で箇条書きを用いる場合も，‘＊’が使用されていることになる．この行の中には記事名に対するインスタンスは記述されていないため，除去する必要がある．このような‘＊’行は，タイトルと一つ目の目次見出しの間にある場合がほとんどであるため，その位置にある‘＊’行を除去すればよい．

### (3) 「＊～一覧」と同じ目次見出しの下位にある‘＊’行を除去

図 3.7 を見ると，“日本の哲学者”は“人名”のインスタンスではないといったように，‘＊’行がタイトルに対するインスタンスになっていないことがわかる．一覧記事中には閲覧者の利便性向上のために関連の高いページへのリンクが列挙されている部分があり，これが‘＊’行として記述されてしまっている場合がある．このような‘＊’行の特徴は，同じ目次見出しに属する‘＊’行のどれかに“～一覧”という文字列を含んでいることである．図 3.7 でも，“聖書学者の一覧”という‘＊’行が同じ目次見出し以下に含まれていることがわかる．このように“～一覧”という文字列を含む‘＊’行を特定し，それが属する目次見出しに属する行をすべて除去する．

### (4) 不要な目次見出し下位の‘＊’行を除去

ある特定の目次見出しに属する‘＊’行は，タイトルに対するインスタンスとしては誤

りである．そのような目次見出しの条件は“関連”，“外部”，“備考”，“参考”，“related”，“凡例”，“カテゴリ”，“出典”，“特記事項”を含むことである．これらのキーワードを含む目次見出しに属する‘＊’行はすべて除去する．

**(5) 不要な‘＊’行を除去**

抽出した‘＊’行の中に未だ存在する不適な行で，行単位でパターン化できるものも存在する．不適な行のパターンとしては，「“＊[[Wikipedia:”から始まる」，「“＊[[:Category:”から始まる」，「“＃REDIRECT”から始まる」，「“＊人名　あ行”のように，五十音のインデックスをもつ」，「“＊[[～一覧]]”」，「“＊関連項目”」がある．このパターンに当てはまる‘＊’行を除去する．

**(6) 不要な年号記述部分を除去**

(7)で述べるが，インスタンスは‘＊’や‘＃’の直後に配置されていることが収集可能な条件である．図 3.7 のような年号の記述は除去し，そのような配置に修正する．

**(7) ‘＊’行からのインスタンスの抽出**

(7)までのスクレイピングで，ほぼ全ての‘＊’行の中には適切なインスタンスが記述されているという状態になった．最後に，‘＊’行の中でどの部分がインスタンスを表す文字列であるかを特定する六つのパターンを作成し，これに従ってインスタンス以外の部分をスクレイピングし，最終的に記事名で表されるクラスとインスタンスの残し，インスタンスを収集する．

一覧記事では‘＊’や‘＃’の直後にインスタンスが配置されている行が圧倒的に多い．逆に‘＊’や‘＃’の直後ではない箇所にインスタンスが配置されている場合，その箇所の特定は難しい．また，‘＊’や‘＃’の直後といっても何文字目までがインスタンスの 1 語を表しているかの特定も難しい．このため，図 3.8 が示す六つのパターンでは，基本的に‘＊’や‘＃’の直後のリンク記号“[[ ]]”に着目してインスタンス文字列を特定し，抽出を行う．

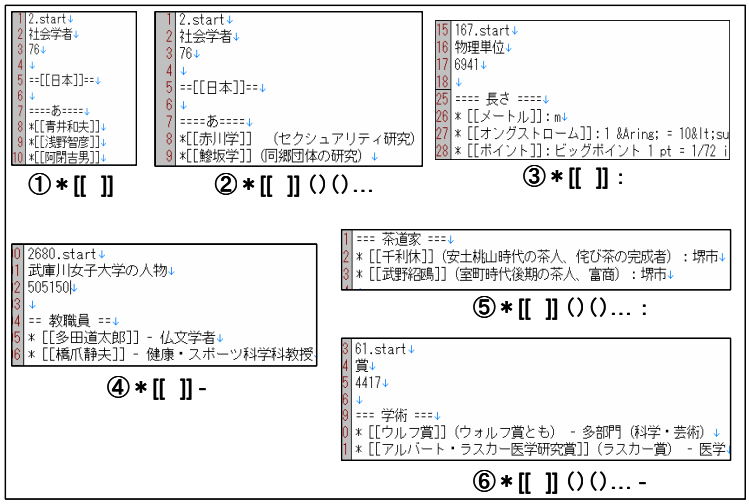



図 3.8 ‘＊’行中でインスタンス箇所を特定するパターン

### 3.3.3 プロパティ名の抽出

本項では，以下の 2 つの手法を用いて，Wikipedia からプロパティ名の抽出を行う．

(1) Infobox のモデリングによるプロパティ名抽出
(2) 記事のリスト構造からのスクレイピングによるプロパティ名抽出

**(1) Infobox のモデリングによるプロパティ名抽出**

Infobox を有する「記事－項目－値」という三つ組は，「インスタンス－プロパティ名－プロパティの値」という三つ組と捉えることができる．そのため，Wikipedia ダンプデータから直接トリプルとして記事タイトルごとのプロパティ名を抽出できるが，いくつかの問題点が存在する．まず，Media Wiki 書式から Infobox を表示するための構造上の問題がある．Infobox には記事の種類ごとにテンプレートが存在し，かつ英語 Wikipedia のテンプレートを利用できる．これは Media Wiki を用いて日本語と英語で完全互換性をとっており，そのため記事の執筆者が簡単に編集できるための措置であるが，ダンプデータからトリプルを抽出する際には英語表記と日本語表記でプロパティ名が別のものとなってしまう．図 3.9 の例では，記事ソース内には「Genre」という単語が述語になっているが，実際の記事では「ジャンル」に変換される．このため，記事ソースから直接 Infobox トリプルを抽出すると，“ジャンル”プロパティではなく，“Genre”プロパティとして抽出してしまう．次に，全てのプロパティの値をリテラルとして抽出してしまうと，プロパティのとる値がデータ値となるのかインスタンスとなるのかの区別が出来ず，プロパティタイプがわからないという問題がある．図 3.9 の例では，ジャンルや開発元プロパティの値は owl:ObjectProperty によりインスタンスと関連付けるべきであるが，人数プロパティの値は owl:DatatypeProperty によりリテラルと関連づけるべきである．

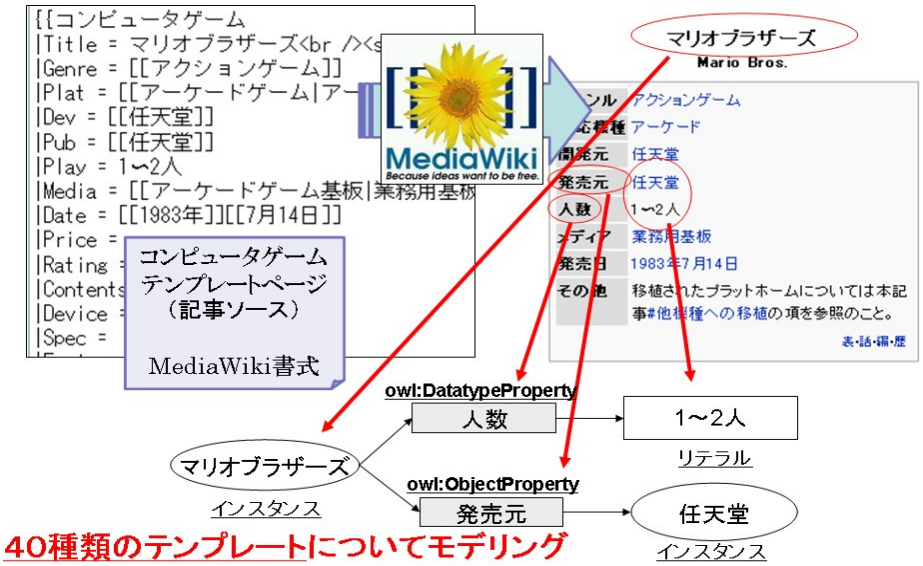



図 3.9 Infobox トリプルからのプロパティ名抽出の一例

以上 2 つの問題点に対応するため, 40 種類の Infobox テンプレートにモデリングを行い, プロパティ名を抽出する．ここでモデリングとは，各テンプレートのプロパティの目的語がインスタンスになるかリテラルになるか，また，目的語がリテラルになる場合にはそのデータ型を記述することを意味する．この際用いる Infobox テンプレート数を 40 と指定したのは, 2009 年 10 月の Wikipedia ダンプデータにおいて, Infobox の総数約 20 万 2000 個に対し, 出現頻度が高かった上位 40 種類の Infobox テンプレートで約 14 万 6000 個（約 72%）の Infobox のモデリングが行えたためである．

**(2) 記事のリスト構造からのスクレイピングによるプロパティ名抽出**

多くの Wikipedia の記事はリスト構造を有している. 本手法はこのリスト構造に着目し, 記事名－リスト構造の見出し語－リスト構造の各値をトリプルと据えてプロパティ名を抽出する．この際に，各記事が属するカテゴリを照合し，カテゴリごとに多く含まれている見出し語を収集する．これにより，記事が属するカテゴリをプロパティの定義域として抽出することが可能となる．ここで，リスト構造の各値とは Wikitext において“＊”から始まる箇条書き文である．抽出の手順を(a)～(d)に示す.

(a) ダンプデータから記事ごとにカテゴリと見出し語を抽出
(b) (a)で抽出したデータから，各カテゴリの見出し語の出現頻度を測定
(c) (b)から出現頻度が少ないものを除去(今回は 5 以下を除去した)
(d) (c)で得た見出し語をプロパティ名として，記事毎にリスト構造の各値を抽出

図 3.10 に, 記事のリスト構造のスクレイピングによるプロパティ名抽出の具体例を示す.

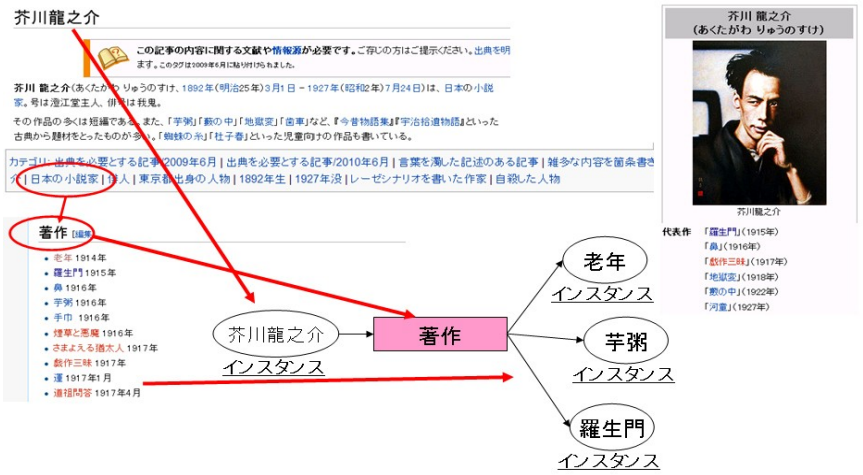



図 3. 10 記事のリスト構造からのプロパティ名抽出の一例

図 3.10 は”芥川龍之介”記事から見出し語“著作”をプロパティ名として，リスト構造の各値である“老年”，“羅生門”，“芋粥”などのプロパティ値を抽出している．プロパティ値“羅生門”は，前述した Infobox のモデリングによる抽出法から抽出可能であるが，“老年”や“芋粥”は，本手法により抽出可能な値である．本手法は，Infobox からは抽出できないプロパティ名だけでなく，プロパティの値をトリプルとして抽出することもできる．

## 3.3.4 プロパティ定義域の抽出

3.3.3 項で述べた抽出法で得たトリプルにおける主語は, 記事名をインスタンスとして据えていた．そのため，主語である記事が属するカテゴリを調べることで，プロパティの定義域を定義できる可能性がある．図 3.11 は，記事「Ruby」が属するカテゴリが Infobox トリプルにおける「設計者」プロパティの定義域として定義できる可能性があることを表している．本手法では以下に述べる手順でプロパティ定義域の抽出を行う．

まず, Infobox テンプレート名を, Infobox が持つ各プロパティの定義域として抽出する．次に，3.3.1 項(2)で述べた「カテゴリ名と Infobox テンプレートの照合」により得た is-a 関係として正しいサブカテゴリを，テンプレートの持つ各プロパティの定義域として対応付ける．さらに，テンプレートで定義されていないプロパティの定義域抽出を試みる．

実際に記事に記載された Infobox に登場するプロパティは，テンプレートで定義されているプロパティ以外のものが使用されるケースが多数存在する．例えば,「有機化合物」というテンプレートで定義されているプロパティは「構造式」,「形状」,「沸点」など合計 21 あるが，実際の記事に掲載されている Infobox のソースから収集したプロパティは，「揮発性」,「臭気」,「蒸気圧」などテンプレートで定義されていないものが多く存在し，合計 33 のプロパティを持つ．

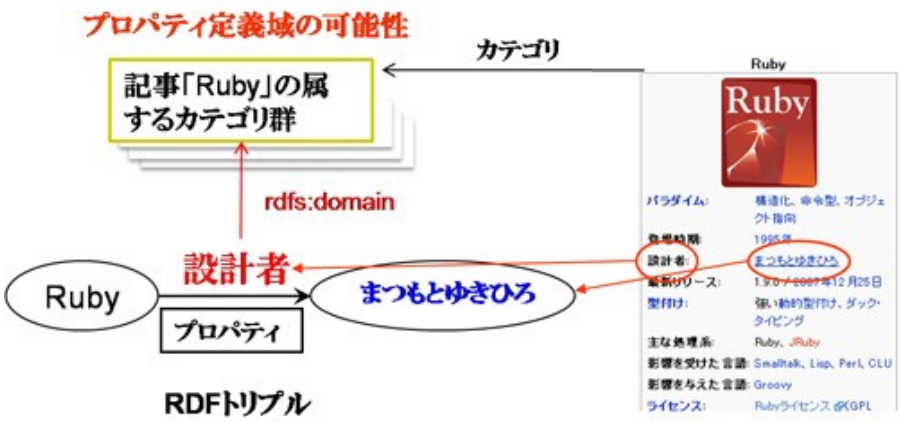



図 3.11 プロパティ定義域と記事が属するカテゴリの対応例

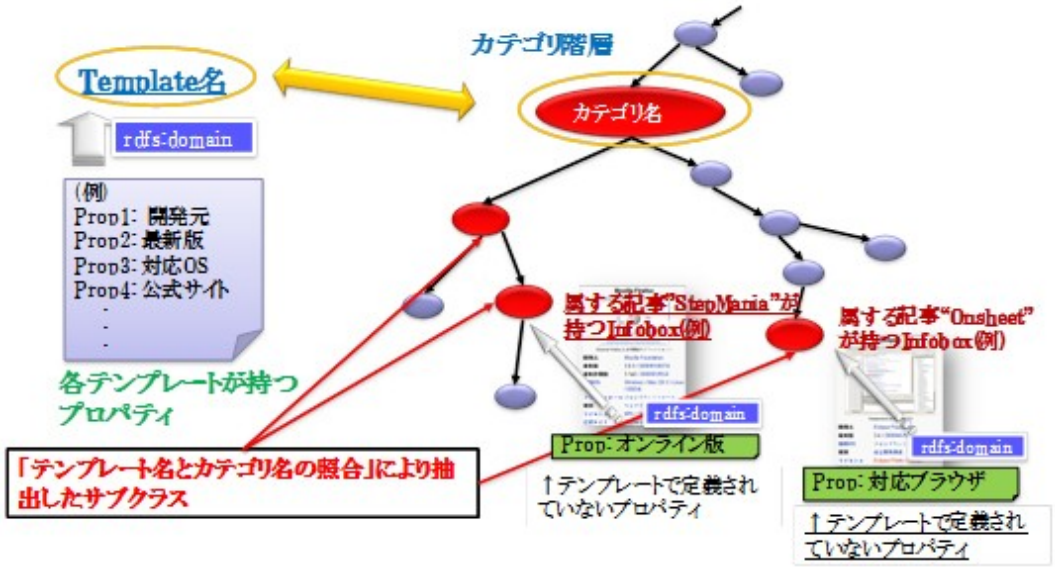



図 3.12 テンプレートで定義されていないプロパティ定義域の抽出

そこで，本提案手法では，3.3.1 項(2)で述べた手法により得たサブカテゴリと，そのカテゴリに属する記事が持つ Infobox テンプレートで定義されていないプロパティとの関係に着目する．Infobox テンプレートで定義されていないプロパティは 3.3.1 項(2)で述べた手法により得たサブカテゴリのいずれかをドメインとする可能性があり，これらを対応付けることで，各プロパティの定義域として最上位の概念であるテンプレート名が得られるだけではなく，より具体化し，ドメインに特化したプロパティおよび定義域の抽出が可能になると考えられる．図 3.12 がプロパティ定義域の抽出手法の全体像である．3.3.1 項(2)で述べた手法により得たサブカテゴリと，この記事が属するカテゴリを比較することで，新たに限定した定義域が抽出できる．

### 3.3.5 プロパティ値域の抽出

3.3.3 項の手法により抽出した各プロパティから，プロパティ値域の抽出を試みる．Infobox トリプルにおいて主語となるインスタンス名は記事名と対応し，その記事が持つ Infobox の元となる Infobox テンプレート名をプロパティ定義域とみなすことができるため，定義域の定義は比較的容易であった．しかし，プロパティ値域は目的語となるインスタンスが記事名とは断定できず，定義域のように全てのプロパティについて定義することは難しい．そこでプロパティ値域の抽出には，以下の 2 つの手法を用いる．

(1) クラスーインスタンス関係からの抽出
(2) is-a 関係からの抽出

**(1) クラスーインスタンス関係からの抽出**

まず，3.3.3 項で抽出したトリプルの目的語（インスタンス）に着目する．Wikipedia の性質上，ある単語が既存記事名と対応する場合には該当記事にリンクされている場合が多く，とりわけ Infobox トリプルにおける値に既存記事名が含まれている場合には，該当記事にリンクされている可能性が高い．また，日本語 Wikipedia オントロジーでは，記事名はインスタンス名に対応している．そこで，Infobox トリプルにおけるプロパティ値に含まれる既存記事へのリンク（アンカーテキスト）と日本語 Wikipedia オントロジーにおけるインスタンス名を文字列照合し，照合したインスタンスのタイプ（クラス）をプロパティ値域として抽出する．

**(2) is-a 関係からの抽出**

次に，先の手法では抽出できないプロパティ値域を抽出するために，前述した手法と同様に，トリプルの目的語となるインスタンスに着目し，インスタンス名と同名の記事が属するカテゴリ名と日本語 Wikipedia オントロジーにおいて既知である is-a 関係のクラス名との文字列照合を行い，照合したクラスを値域として抽出する．さらに抽出したクラスの is-a 関係における最上位概念も値域として抽出する．これは，値域として定義されたクラスがプロパティごとに複数存在するため，今後，上位概念に統合する際の指標となる．

図 3.13 がプロパティ値域の抽出法の一例である．“開発元”プロパティの値である“任天堂”はクラスーインスタンス関係において“日本の企業”クラスに属するため，これを値域として抽出する．さらに，“任天堂”記事が属するカテゴリと日本語 Wikipedia オントロジーにおける is-a 階層を照合し，カテゴリと照合したクラスとその最上位概念となる Infobox テンプレート名（この例では“会社”）を値域として抽出する．

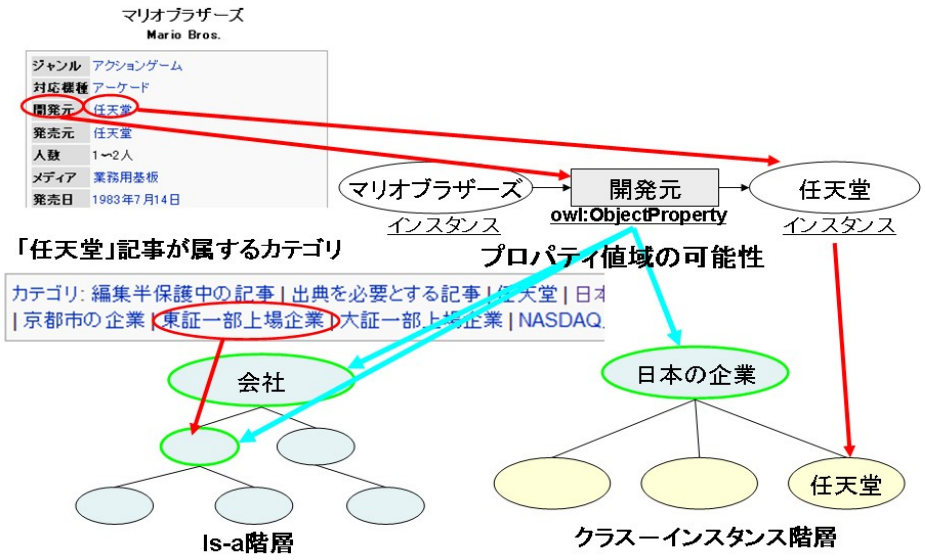



図 3.13 プロパティ値域の抽出の一例

## 3.3.6 プロパティ上位下位関係の抽出

Wikipedia の Infobox が記事の概要を表しているという Wikipedia の特徴に着目し，3.3.3 項(1)で Infobox から抽出したプロパティ名と 3.3.3 項(2)でリスト構造から抽出したプロパティ名の上位下位関係の抽出を試みる．まず，トリプルの主語となるインスタンスごとにリスト構造から抽出した各プロパティの値と Infobox から抽出した各プロパティの値を照合し，プロパティの値が少なくとも 1 つ存在していた場合に，リスト構造から抽出したプロパティ名を Infobox から抽出したプロパティ名の上位プロパティ候補として抽出する．次に，先ほど抽出したプロパティ候補の上位プロパティと下位プロパティの定義域と値域を照合し，どちらのプロパティにも同じ定義域と値域が存在していた場合にプロパティの上位下位関係として抽出する．

図 3.14 がプロパティ上位下位関係の抽出の一例である．トリプルの主語である“芥川龍之介”は 3.3.3 項(2)で抽出した“著作”プロパティと，その値である“老年”，“羅生門”，“芋粥”等を持っており，さらに 3.3.3 項(1)で抽出した“代表作”プロパティと，その値である“羅生門”，“鼻”等を持っている．そのため，上位プロパティとして“著作”，下位プロパティとして“代表作”というプロパティ上位下位関係候補を得る．次に，これらの定義域と値域を照合すると，どちらも定義域として“作家”，値域として“日本の小説”を持っている．このため，プロパティの上位下位関係として“著作－代表作”という関係が抽出できる．

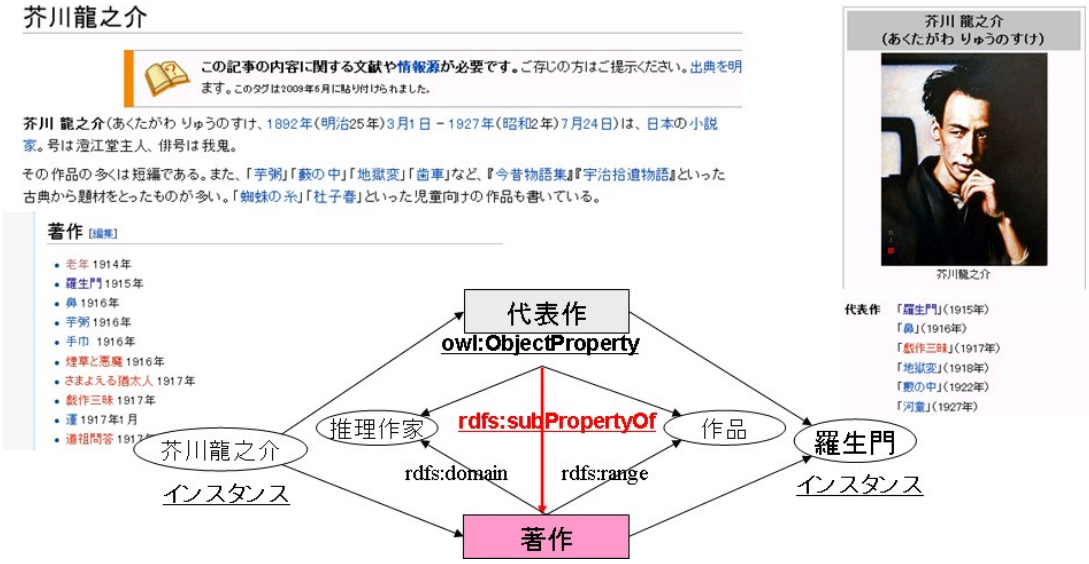



図 3. 14 プロパティ上位下位関係の抽出の一例

## 3.3.7 プロパティタイプの推定

3.3.3 項(1)の手法で抽出したプロパティは Infobox のモデリングにより，オブジェクトプロパティとデータタイププロパティの分類がなされている．本手法ではオブジェクトプロパティとデータタイププロパティのプロパティタイプに加え，3.3.3 項で抽出したトリプルを用いて，以下の 4 つのプロパティタイプの推定を行う．

(1) 対称関係プロパティ(owl:SymmetricProperty)
(2) 推移関係プロパティ(owl:TransitiveProperty)
(3) 関数関係プロパティ(owl:FunctionalProperty)
(4) 逆関数関係プロパティ(owl:InverseFunctionalProperty)

ここで対称関係プロパティとは，主語 X(n)－プロパティ P(n)－目的語 Y(n)となるプロパティ P(n)が存在した場合に，Y(n)－P(n)－X(n)も成り立つプロパティであり，推移関係プロパティとは，主語 X(n)－プロパティ P(n)－目的語 Y(n)，主語 Y(n)－プロパティ P(n)－目的語 Z(n)，となるプロパティ P(n)が存在した場合に，X(n)－P(n)－Z(n)も成り立つプロパティであり，関数関係プロパティとは，プロパティ P(n)について全ての主語 X から目的語 Y(n)が 1 つに決まるプロパティであり，逆関数関係プロパティとは，P(n)について全ての目的語 Y から主語 X(n)が 1 つに決まるプロパティである．

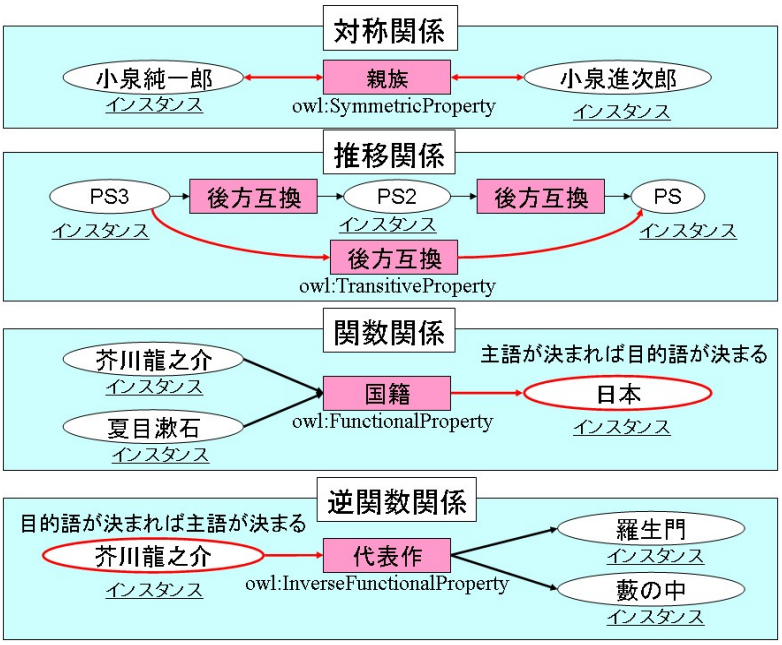



図 3.15 プロパティタイプの抽出の一例

### (1) 対称関係プロパティ(owl:SymmetricProperty)

はじめに，対称関係プロパティの推定を行う．3.3.3 項で抽出した各プロパティ P(n)の主語であるインスタンス X(n)とプロパティの値であるインスタンス Y(n)を取り出し，プロパティ P(n)において Y(n)－P(n)－X(n)も成り立っていた場合にプロパティ P(n)を対称関係プロパティの候補として抽出する．さらに，プロパティ P(n)の全トリプル数 A と対称関係として抽出したトリプル数αの割合から対称関係プロパティの推定を行う．図 3.15 において，“親族”プロパティは，そのトリプルである“小泉純一郎－親族－小泉進次郎”と“小泉進次郎－親族－小泉純一郎”という対称関係が成立している．

### (2) 推移関係プロパティ(owl:TransitiveProperty)

次に，推移関係プロパティの推定を行う．各プロパティ P(n)の主語であるインスタンス X(n)とプロパティの値であるインスタンス Y(n)を取り出し，さらに，インスタンス Y(n)とプロパティの値であるインスタンス Z(n)を取り出す．このプロパティ P(n)において X(n)－P(n)－Z(n)も成り立っていた場合にプロパティ P(n)を推移関係プロパティの候補として抽出する．さらに，プロパティ P(n)の全トリプル数 A と推移関係として抽出したトリプル数αの割合から推移関係プロパティの推定を行う．図 3.15 において，“後方互換”プロパティは推移関係プロパティであり，そのトリプルとして“PS3－後方互換－PS2”と，“PS2－後方互換－PS”が成り立つとき，“PS3－後方互換－PS”も成り立つ．

### (3) 関数関係プロパティ(owl:FunctionalProperty)

同様に関数関係プロパティと逆関数関係プロパティの推定を行う．各プロパティ P(n)の主語 X(n)と目的語 Y(n)を取り出し，プロパティ P(n)において，全ての主語 X から目的

語 Y(n)が特定できるとき，このプロパティ P(n)を関数関係プロパティとして抽出する．図 3.15 において，“国籍”プロパティは関数関係プロパティであり，そのトリプルとして，“芥川龍之介－国籍－日本”や“夏目漱石－国籍－日本”など，全トリプルにおいて，プロパティの主語が決まれば，プロパティの値が特定できる．

**(4) 逆関数関係プロパティ(owl:InverseFunctionalProperty)**

関数関係プロパティと同様に，全ての目的語 Y から主語 X(n)が特定できるとき，このプロパティ P(n)を逆関数関係プロパティとして抽出する．図 3.15 では，“代表作”プロパティは逆関数関係プロパティであり，そのトリプルとして，“芥川龍之介－代表作－羅生門”や“芥川龍之介－代表作－藪の中”など，全トリプルにおいて，プロパティの値が決まれば，プロパティの主語が特定できる．

## 3.3.8 jwo 語彙関係の抽出

ここまで抽出した大規模オントロジー構築のための関係に加え，LOD としての有用性を高めるため，下記の 3 つの関係を抽出する．これら 3 つの関係はこれまでの手法に比べ，曖昧で誤りも多い．しかしながら，LOD として公開した場合に，検索やデータの対応付けの際に指標として利用可能である．

(1) 上位下位関係 (jwo:hyper)
(2) 関連語・同義語(jwo:nearly)
(3) 動詞とプロパティの関係(jwo:verb)

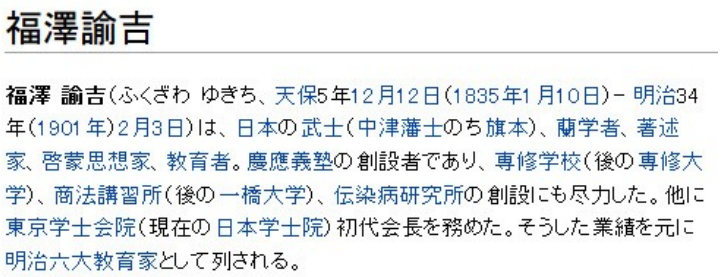
**(1) 上位下位関係の抽出**

3.3.1 項と 3.3.2 項で，クラス及びインスタンスを明確に定義しており，上位下位関係を is-a 関係とクラスーインスタンス関係に分類していた．しかしながら，上位のクラスを持たない記事も多く存在しているため，新たに記事のアブストラクトから上位下位関係を抽出し，jwo:hyper 語彙により関係を定義する．実際の抽出手順は次のとおりである．

(1) Wikipedia 記事の最初の段落をアブストラクトとして抽出
(2) いくつかのパターンから記事名を下位語とする上位下位関係を抽出
(3) jwo:hyper を語彙として関係を定義

図 3.16 は福澤諭吉の記事のアブストラクトである．多くの Wikipedia の記事には図のように「記事名(よみ、生年- 没年)」は、上位語 1、上位語 2・・・」という記述が見られる．このようなパターンから記事名を下位語として上位下位の関係を構築する．

結果として，「福澤諭吉」記事から「著述家」「蘭学者」，「トヨタ自動車」記事から「自動車メーカー」，「吾輩は猫である」記事から「長編小説」などを上位語として抽出した．


福澤諭吉

**福澤 諭吉**(ふくざわ ゆきち、天保5年12月12日(1835年1月10日) - 明治34年(1901年)2月3日)は、日本の武士(中津藩士のち旗本)、蘭学者、著述家、啓蒙思想家、教育者。慶應義塾の創設者であり、専修学校(後の専修大学)、商法講習所(後の一橋大学)、伝染病研究所の創設にも尽力した。他に東京学士会院(現在の日本学士院)初代会長を務めた。そうした業績を元に明治六大教育家として列される。

**図 3.16 福澤諭吉記事のアブストラクト**

### (2) 関連語・同義語の抽出

Wikipedia にはリダイレクトという機能が存在する．これは，あるページを表示した際に同義語のページへ自動的にリンクさせる機能である．リダイレクト元の記事名とリダイレクト先の記事名との関係は同義語の関係にあり，Wikipedia のリダイレクト情報を利用することで同義語の抽出が可能となる．実際に，Wikipedia ダンプデータから 313,527 のリダイレクトリンクを抽出し，3.3.1 項と 3.3.2 項の手法で得たクラスおよびインスタンスの同義語として，約 10 万の語彙を得た．表 3.1 に正しく抽出した同義語の例を，表 3.2 に誤って抽出した同義語の例を示す．

抽出した同義語から 1,000 個の標本抽出を行い，同義語の正解率の区間推定を行った．その結果，正解率の 95%信頼区間は，67.0 ± 2.90%だった．リダイレクトリンクから直接，クラスおよびインスタンスにおける同義語を高精度に抽出できないことがわかる．同義語としての精度が低いため，よりゆるいリソース間をつなぐ語彙として jwo:nearly を用いて関係を定義する．また，Infobox から直接抽出した Infobox プロパティと日本語 Wikipedia オントロジー独自のプロパティの関係も jwo:nearly 語彙により定義する．

結果として，「福澤諭吉」と「福沢諭吉」，「スティーヴジョブス」と「スティーブジョブズ」，「国籍」プロパティと「nationality」プロパティなどを関連語・同意語の関係として抽出した．

**表 3.1 正しく抽出した同義語の例**

| クラス名・インスタンス名 | 同義語 |
|---|---|
| ソフトウェア工学 | ソフトウエア工学 |
| イギリス | 英国 |
| 国際連合 | UN |
| 横浜ベイスターズ | 太洋ホエールズ |
| アメリカ特殊作戦軍 | SOCOM |

表 3.2 誤って抽出した同義語の例

| クラス名・インスタンス名 | 同義語 | 誤りの内容 |
|---|---|---|
| アイドル | 男性アイドル | is-a 関係 |
| ビール | 非熱処理ビール | is-a 関係 |
| イタリアの戦車 | L5/30 | クラスーインスタンス関係 |
| 警察 | 警察力 | has-a 関係 |
| 社会科学部 | 社会科学科 | has-a 関係 |

**(3) 動詞とプロパティ関係の抽出**

プロパティトリプルを用いて，Wikipedia 記事内の文章から同一の目的語が出現する文に注目し，その文中の動詞を抽出する．これにより，プロパティと意味的に近い動詞が抽出できる可能性があり，今後プロパティの表記揺れ問題の対策に利用できる．本関係は jwo:verb 語彙により表記する．例えば，日本語 Wikipedia オントロジーの「周辺情報」プロパティを含むトリプルの目的語は文中で「位置する」「隣接する」といった動詞と共に出現することが多い．こうしたプロパティと動詞を jwo:verb により対応付ける．結果として，先の「周辺情報」プロパティと「位置する」「隣接する」，「発売元」プロパティと「発売する」「販売する」，「掲載誌」プロパティと「掲載する」などを抽出した．

## 3.3.9 抽出した関係の洗練

本項では，3.3.2 項，3.3.3 項で抽出した以下の 2 つの関係を洗練することで，精度の向上を行う．

(1) クラスーインスタンス関係の洗練
(2) プロパティ定義域・値域の洗練

**(1) クラスーインスタンス関係の洗練**

3.3.2 項で述べたように，クラス―インスタンス関係は一覧記事のスクレイピングにより構築している．本手法によって抽出したクラス名は一覧記事名となるため，例えば，“芥川龍之介” インスタンスは“日本の小説家” クラスに属していることとなる．本手法は多くのクラス―インスタンス関係を抽出することが可能になるが，“日本の小説家”，“アメリカの小説家”といった，クラス階層にハイブランチ構造を生じさせる問題がある．事前実験として，Wikipedia ダンプデータから抽出した 10,854 の一覧記事のうち，「日本の」からはじまる記事は 624 であった．このような『国名や地域名＋格助詞「の」＋クラス名』となるクラスは多く，これらがハイブランチ構造を生む要因となっている．ハイブランチ構造によりプロパティ定義域・値域の洗練の際に，問題が生じるため，まずこの除去を行う．実際の除去の手順は次のとおりである．

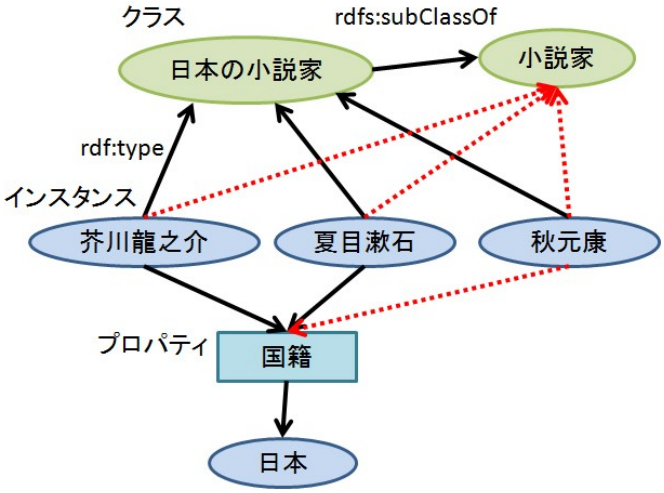



図 3.17 クラスーインスタンス関係の洗練の一例

(1) クラス―インスタンス関係のクラス名に注目し，格助詞「の」が含まれるクラス名を抽出
(2) (1) で抽出したクラスに含まれるインスタンスのうちプロパティの値が格助詞「の」の前方部となっているプロパティを抽出
(3) (2) から出現頻度が少ないものを除去(今回は 5 以下を除去した)
(4) 格助詞「の」の後方部を新たなクラス―インスタンス関係として抽出
(5) プロパティとプロパティの値を持たないインスタンスは抽出した関係を補完

図 3.17 は，本手法の一例である．“日本の小説家” クラスには “芥川龍之介”，“夏目漱石”，“秋元康” など多くのインスタンスが属している．まずクラス名の格助詞「の」に注目し，クラスに属するインスタンスのプロパティの値に “日本” が含まれるプロパティを抽出する．多くのインスタンスは “国籍” プロパティを持っており，その値は “日本” になっている．そこで，クラス名から日本を除去し，新たに “小説家” クラスのインスタンスとして定義する．さらに，これまでの日本の小説家クラスのインスタンスのうち “国籍” プロパティとその値 “日本” を持っていないインスタンス(この例では “秋元康” インスタンス) にその関係を補完する．

**(2) プロパティ定義域・値域の洗練**

日本語 Wikipedia オントロジーの多くのプロパティ定義域はリーフとなるクラスに偏っているという問題がある．これは，プロパティ抽出をインスタンス(記事名) をベースに行っていることに起因する．インスタンスは主にリーフクラスに属するため，各記事がもつプロパティはリーフクラスに直接定義されてしまう．例えば，野球選手である “イチロー” というインスタンスは日本語 Wikipedia オントロジーにおいて “日本のプロ野球選手” というクラスに属しているため，“イチロー” (および他の日本のプロ野球選手) が持つ「国籍」や「ポジション」や「年度別打撃成績」といったプロパティは，“日本のプロ野球選手”

クラスを定義域として持つ．同様に，“日本のサッカー選手” クラスのインスタンスが持つ「国籍」や「生年月日」や「ポジション」といったプロパティは“日本のサッカー選手”クラスを定義域とし，“小説家” クラスのインスタンスが持つ「国籍」「生年月日」「処女作」「受賞歴」といったプロパティは“小説家” クラスを定義域として持つ．しかし，「生年月日」や「国籍」といったプロパティは本来“人物” クラスに定義されるべきものである．そして“人物” クラスにそれらが定義できれば，クラス階層を利用して上位クラスからプロパティ継承を用いることで，“人物” クラスの下位にあるクラスは“人物” クラスのプロパティセットを継承することができる．そこで，プロパティを持つインスタンスとクラス―インスタンス関係を用いて，各プロパティをクラスに紐付けし，親子クラス及び兄弟クラスに紐付けされたプロパティを参照する．これにより，定義域を上位クラスに統合(リフトアップ) が可能になり，先の問題を解消する．しかしながら，本手法の問題として，is-a 階層のハイブランチ構造により，リフトアップがうまくいかないことがあった．そこで, 本手法を 3.3.9(1)の手法を用いて新たに抽出した定義域・値域に適用することで，リフトアップ精度をあげるとともに，これまで行っていなかった値域にも洗練を行う．図 3.18 がプロパティ定義域・値域の洗練の一例である．

## 3.4 実験と考察

本節では，はじめに 3.3 節で提案した各手法に関する実験と評価・考察を述べる．次に，日本語 Wikipedia オントロジー全体の評価と考察を述べる．本実験における実験環境を表 3.3 に示す．なお，本章の実験は，2010 年 11 月時点の Wikipedia ダンプデータをダウンロードして行った．

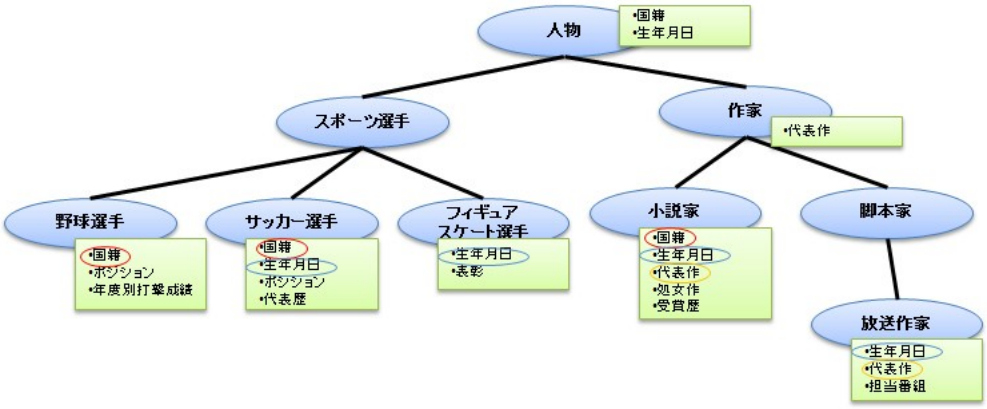



図 3. 18 プロパティ定義域・値域の洗練の一例

表 3.3 実験環境

| CPU | Intel Core i7 2.80GHz |
|---|---|
| メモリ | 16.0GB |
| OS | Windows 7 |
| 開発言語 | Java |
| DBMS | MySQL 5.0 |

## 3.4.1 is-a 関係の抽出結果と考察

**(1) カテゴリ階層に対する文字列照合による is-a 関係の抽出結果と考察**

実験の結果，総カテゴリ数 91,316 個のカテゴリを持つカテゴリ階層から後方文字列照合によって 7,971 個，前方文字列照合部除去によって 4,587 個，計 12,558 個の is-a 関係を抽出した．抽出した 12,558 個の母集団の中から 1,000 個の標本を抽出し，正誤を判定した．その結果から式①[56]を利用して真の正解率の 95%信頼区間を算出すると，93.1 ± 1.51%という結果が得られた．式①において $N$ は母数，$n$ は標本数，$\hat{p}$は真の正解率の推定量であり，正解の標本数を総標本数で割ったものである．表 3.4 および表 3.5 にそれぞれ後方文字列照合，前方文字列照合部除去で抽出されたリンクの例を提示する．表 3.6 は誤りの例とその内容を表している．

$$\left[\quad \hat{p}-1.96\sqrt{\left(1-\frac{n}{N}\right)\frac{\hat{p}(1-\hat{p})}{n-1}} \quad , \quad \hat{p}+1.96\sqrt{\left(1-\frac{n}{N}\right)\frac{\hat{p}(1-\hat{p})}{n-1}} \quad \right] \quad \cdots①$$

表 3.4 後方文字列照合で抽出した is-a 関係の例

| 親クラス | 子クラス |
|---|---|
| 俳優 | アトラクション俳優 |
| 高速道路 | 各国の高速道路 |
| 高速鉄道 | 台湾高速鉄道 |
| 魚介料理 | 日本の魚介料理 |
| 魚類 | 軟骨魚類 |
| 鳥類 | 絶滅鳥類 |

表 3.5 前方文字列照合部除去で抽出した is-a 関係の例

| 親クラス | 子クラス |
|---|---|
| 食品メーカー | 製パン業者 |
| 武器 | 刀剣 |
| 麺料理 | 焼きそば |
| 輸送機器 | 自動車 |

表 3.6 文字列照合で抽出した is-a 関係の誤りの例

| 親クラス | 子クラス | 間違いの内容 |
|---|---|---|
| グローバリゼーション | 反グローバリゼーション | 反・非などを含む |
| 文庫 | 富士見ミステリー文庫 | クラスーインスタンス |
| 高速道路 | ジャンクション | Has-a関係 |
| 経済 | 企業 | 抽象的な語が親 |
| 教育の歴史 | 旧制教育機関 | 抽象的な語が親 |
| 文化 | アニメ作品 | 抽象的な語が親 |
| 歴史 | 政治 | 抽象的な語が親 |
| 社会 | 事件 | 抽象的な語が親 |
| 地理 | 建築物 | 抽象的な語が親 |

全体的な正解率としては良い結果を得た．表 3.4，表 3.5 より，後方文字列照合では複合語からなる is-a 関係を抽出できており，前方文字列照合部除去では文字列に依存しない is-a 関係の抽出ができていることがわかる．しかし汎用オントロジーとしての階層の規模としてはまだ小さい．次に誤りの内容について考察する．表 3.6 の 1 つ目の誤りは，「反」や「非」などの否定語が子クラスの先頭にくる場合に，後方文字列照合により誤った is-a 関係を抽出した例を示している．表 3.6 の 2 つ目の誤りは，後方文字列照合により，クラスーインスタンス関係を誤って抽出した例を示している．Wikipedia では，有名なインスタンスはカテゴリ化され，クラスに相当するカテゴリと下位カテゴリ・上位カテゴリの関係が構築されることがある．そのような場合，後方文字列照合により，is-a 関係ではなくクラス-インスタンス関係を誤って抽出することがある．表 3.6 の 3 つ目の誤りは，Has-a 関係を誤って抽出した例を示している．表 3.6 の 4 つ目以降の誤りは，「経済－企業」のように抽象的なクラスが親クラスとなっている場合に，誤った is-a 関係を抽出した例を示している．抽象的なクラスとは，Wikipedia カテゴリ階層の上位に存在するカテゴリに相当する．日本語版 Wikipedia のカテゴリ階層は，上位オントロジーのように物ごとの厳密な分類がなされておらず，「学問」，「技術」，「自然」，「社会」，「地理」，「人間」，「文化」，「歴史」に「総記」を加えた 9 の「主要カテゴリ」がルートカテゴリとなっている．これらのルートカテゴリとその直下のカテゴリの間には，is-a 関係として不適切な関係が多く存在する．抽象的なクラスを親クラスに持つ場合に，誤った is-a 関係を抽出した理由は，

Wikipedia では，上記で示した 9 つのルートカテゴリは分類の基幹となっているため，ルートカテゴリに修飾語を追加したカテゴリが数多く Wikipedia カテゴリ中に存在するためである．例えば，表 3.6 の 4 つ目の誤りの例では，「日本の経済」（ルートカテゴリ「経済」に「日本の」という修飾語を追加したカテゴリ）と「日本の企業」という上位・下位カテゴリの関係から，前方文字列照合部除去により「経済」is-a「企業」という誤った is-a 関係を抽出している．

**(2) カテゴリ名と Infobox テンプレートの照合による is-a 関係の抽出結果と考察**

Wikipedia ダンプデータから 889 種類の Infobox テンプレートおよび 212,419 の Infobox を抽出した．表 3.7 に頻出 Infobox テンプレートおよびその Infobox テンプレートを用いて作成された Infobox を掲載している記事数を示す．また，3.3.1 項(2)で述べた手法を用いた結果を以下の表 3.8 にまとめる．

表 3.8 に示すように，結果として 3,782 の is-a 関係が抽出できた．テンプレート名とカテゴリ名が照合したカテゴリ（以下，本稿ではルートカテゴリと呼ぶ）は 216 存在し，各ルートカテゴリにおいて Infobox を持つ記事が属するカテゴリ群と照合したサブカテゴリを持つルートカテゴリは 157 であった．3 割程度のルートカテゴリから is-a 関係が抽出できなかった原因の大部分は，ルートカテゴリの中に「オリンピック[国名]選手団」（[国名]には，「アメリカ」や「日本」などが入る）など，サブカテゴリを 1 つも持たないルートカテゴリが存在したからである．Infobox を持つ記事が属するカテゴリ群と照合した 157 のルートカテゴリにおける全サブカテゴリ数は 50,107 個であった．しかし，「国」，「経済」，「解剖学」の 3 つのルートカテゴリで 42,468 個を占めていた．この原因として，「国」，「経済」は概念として抽象的であり，Wikipedia では「各国の人物」などの多くの国に関係するカテゴリが下位カテゴリとして記述される傾向があるためである．「解剖学」については人体の部位の他，解剖学に用いられるコンピュータ機器等の多くの関係する概念がカテゴリとして記述されているためだと考えられる．上記 3 つのルートカテゴリから抽出した is-a 関係数は 518 個であり，ルートカテゴリの全サブカテゴリからの抽出率は非常に低い．

**表 3.7 Infobox テンプレート名と掲載記事数**

| | |
|---|---|
| 基礎情報 会社 | 13,717 |
| Infobox Album | 11,984 |
| 駅情報 | 11,363 |
| 生物分類表 | 8,517 |
| ActorActress | 8,103 |
| サッカー選手 | 7,750 |
| Single | 6,861 |
| Baseball Player | 6,672 |
| Musician | 6,645 |

表 3.8 Infobox テンプレート名とカテゴリ名の照合結果

| Wikipediaカテゴリの数 | Infoboxテンプレートの種類 | テンプレート名とカテゴリ名の照合数 | is-a関係として抽出されたサブカテゴリ数 |
|---|---|---|---|
| 91,316 | 889 | 216 | 3,782 |

表 3.9 にカテゴリ名と Infobox テンプレートの照合による is-a 関係抽出結果の一部を示す．なお，表 3.9 の再現率は，is-a 関係が正しく成り立つと人手によって判断した各ルートカテゴリ以下のサブカテゴリを正解集合として算出した． また，「国」，「経済」，「解剖学」の 3 つのルートカテゴリについては，上記で述べたように，サブカテゴリ数が多く，再現率の算出が困難なため，表 3.9 の評価結果には反映していない．

抽出した is-a 関係から 1,000 個の標本抽出を行い，式①を用いて，正解率の区間推定を行った．正解率の 95%信頼区間は 93.2 ± 1.34%であり，9 割以上の精度で is-a 関係が抽出できた．「楽器」を例に挙げると， 3.3.1 項(1)で述べた文字列照合による手法では抽出できない「ピアノ」や「トランペット」などの下位概念が抽出できている．先の「国」，「経済」，「解剖学」の 3 つのルートカテゴリを含めた場合の正解率は 95.6 ± 1.09%であり，こちらも 9 割以上の精度で is-a 関係が抽出できている．再現率に関しては 68.7%という結果を得た．「日本の温泉地」のように，抽出した is-a 関係が正解集合と完全一致したケースもあるが，「新聞」のように，正解の is-a 関係がサブカテゴリ以下に 38 存在しているにも関わらず，抽出した is-a 関係は 4 であったケースもあった．再現率が低くなった理由として，Wikipedia 全記事数に対して，Infobox を持つ記事数が 3 割程度しかないことが挙げられる．全体の記事に対して，Infobox を持つ記事が少ないため，Infobox を持つ記事が属するカテゴリを網羅的に獲得することができず，is-a 関係の抽出漏れが発生していると考えられる．また，カテゴリツリーは正しい is-a 関係を多数含むものの，性質の継承という観点から捉えた際，is-a 関係とは呼べないその他の関係も同時に多く含んでいる．先ほど述べた「国」，「経済」，「解剖学」の 3 つのルートカテゴリおける全サブカテゴリについて，1,000 の標本を抽出し，is-a が成立する割合を人手により調べた結果，約 7.2%であった．サブカテゴリを多く持つルートカテゴリは，そのほとんどが間違って記述された下位カテゴリから派生したもので占めていることがわかる．したがって，提案手法を用いてより多くの is-a 関係を抽出し，再現率を高めるためには，洗練された階層を持つカテゴリに対して，それぞれのカテゴリに属する記事の Infobox を増やすことが効果的であると考えられる．そのためには，Wikipedia の記事の編集において，記事に与えるべきカテゴリと Infobox が完全に独立している現状を変える必要がある．例えば，ユーザが記事に対して属するカテゴリを追加する際に関連する Infobox を追加，また，Infobox を追加した際には関連するカテゴリを追加，といったようなカテゴリと Infobox の自動連携など，両者の対応関係を増加させる仕組みの検討が必要である．

表 3.9 カテゴリ名と Infobox テンプレートの照合により抽出した is-a 関係の評価

| ルートカテゴリ名 | 抽出したサブカテゴリ数 | 抽出したサブカテゴリのis-a正答率 | 正しいis-aと判断された数 | 再現率 |
|---|---|---|---|---|
| サッカークラブ | 99 | 1 | 106 | 0.93 |
| 有機化合物 | 76 | 1 | 83 | 0.92 |
| テニス選手 | 73 | 1 | 80 | 0.91 |
| 日本の温泉地 | 48 | 1 | 48 | 1 |
| サッカー選手 | 144 | 0.99 | 150 | 0.96 |
| 無機化合物 | 15 | 0.86 | 17 | 0.88 |
| ・・・ | ・・ | ・・ | ・・ | ・・ |
| 平均 | 17.82 | 0.93 | 26.21 | 0.68 |

### (3) 目次見出しのスクレイピングによる is-a 関係の抽出結果と考察

Wikipedia ダンプデータから目次に「分類」「種類」が含まれる記事 10,124 記事においてスクレイピングを行った結果， 83,288 個の is-a 関係を抽出した．抽出した is-a 関係のルートの概念数は 6,370 個, リーフ数は 55,081 個, 全概念数は 73,837 個であった. 表 3.10 に抽出した is-a 関係の例を示す．表 3.11 に下位概念数が多いルート概念の例を示す.

表 3.10 目次見出しのスクレイピングで抽出した is-a 関係の例

| 親クラス | 子クラス |
|---|---|
| コケ植物 | ゼニゴケ植物門 |
| 木材パルプ | N 材 |
| 医療用ロボット | 介護ロボッ |
| 哲学 | 論理哲学 |

表 3.11 下位概念数が多いルート概念の例

| ルート概念 | ルート概念に含まれる下位概念の数 |
|---|---|
| カード | 1,176 |
| ゲーム | 131 |
| 北西太平洋岸のインディアン | 91 |
| ニベ科 | 66 |
| ハタネズミ属 | 47 |

表 3.12 目次見出しから抽出した is-a 関係の誤りの例

| ルート概念 | リーフ概念までのis-a 階層構造 |
|---|---|
| 阪神電気鉄道 | 阪神電気鉄道－分類について-前期大型車 |
| 戦争 | 戦争－自衛権の容認理由 |
| 暗号 | 参考：コードの例－例1 |
| ピアノ | ピアノ－その他 |
| 図書館 | 日本の国立図書館－国立国会図書館 |
| 新潟県 | 新潟県－経済－情報インフラ |

抽出した is-a 関係から 1,000 個の標本抽出を行い，式(1) を用いて，正解率の区間推定を行った．その結果，正解率の 95%信頼区間は，72.6 ± 2.74%であった．正解率が他の is-a 関係抽出法に比べ低くなっている原因としては，目次見出しから抽出した is-a 階層には，「その他」などのコンテキストに依存したクラスが含まれているためである．例えば，「ピアノ－その他」といった is-a 関係が目次見出しから抽出できるが，ここでいう「その他」は「その他のピアノ」の省略であり，コンテキストを考慮して，補完を行うことにより，妥当な is-a 関係に洗練できる場合もある．そこで新たに，抽出した is-a 階層構造のルート概念からリーフ概念までを 1 個の標本として，1,000 個の標本抽出を行い，正解率の区間推定を行った．その際，「その他」などコンテキストに依存するクラスについては，補完を行うことで正しいクラスに修正した上で評価を行った．正解率の 95%信頼区間は，86.1 ± 2.13%であった．3.3.1 項(1)及び 3.3.1 項(2)で述べた手法では得ることのできない「パーソナルコンピューターデスクトップパソコン」や「寿司－巻き寿司－太巻」といった is-a 階層関係が抽出できている．

また，抽出した概念は表 3.10 を見ると分かるように，遊戯に関するもの，インディアンに関するもの，生物の分類体系に関するものが多かった．カードが多くなっている理由として，カードゲームに関する概念も含まれているためと考えられる．インディアンや生物分類に関する概念が多い理由は，Wikipedia のアメリカ州の先住民族の記事や生物に関する記事は綺麗な階層的記述が多いため，今回の抽出法と相性がよく，比較的多い概念を抽出できたからである．

表 3.12 に目次見出しから抽出した is-a 関係の誤りの例を表す．誤りは，スクレイピングのルールが不足していることによるものが大部分を占めていた．表 3.10 の 1 つ目と 2 つ目は，「分類」「種類」は含まれていたが，その後に説明文が箇条書きされていたために，誤って抽出した is-a 関係の例である．表 3.10 の 3 つ目と 4 つ目は，箇条書きから is-a 関係を抽出するためのスクレイピングのルールが不足していたために，誤って抽出した is-a 関係の例である．スクレイピングのルールの不足以外に，表 3.10 の 5 つ目のようにクラスーインスタンス関係や 6 つ目のように抽象的な概念を含むために，誤って is-a 関係を抽出するケースもあった．

したがって，提案手法による is-a 階層関係の正解率を上げるためにはより厳密なスクレイピングルールを追加する必要があると考えられる．しかし，is-a 関係の抽出数としては 3.4.1 項(1)及び 3.4.1 項(2)で抽出できた is-a 関係数のおよそ 5 倍の関係が抽出できており，規模としては非常に大きいといえる．

## 3.4.2 クラスーインスタンス関係の抽出結果と考察

Wikipedia ダンプデータから抽出した 8,796 の一覧記事に対してスクレイピングを行い，クラス―インスタンス関係の抽出を行った．取得したインスタンスは 323,024 個，一覧記事の記事名から生成したクラス数は 2,902 個，クラスーインスタンス関係数は 421,989 であった．また，抽出したクラスーインスタンス関係から 1,000 個の標本抽出および正誤判定を行い，式①を用いて正解率の区間推定を行った．その結果，正解率の 95%信頼区間は，97.2 ± 1.02%と高精度であった．表 3.13 に正しく抽出したクラスーインスタンス関係の例を示す．表 3.14 に誤って抽出したクラスーインスタンス関係の例を示す．

インスタンスを多く持つクラスの例としては，日本の声優クラスが 3,658 個，日本の漫画家クラスが 2,854 個，日本の男優クラスが 2,321 個といったように，人物のインスタンス数が圧倒的に多い．これは Wikipedia 一覧記事が人物のコンテンツを特に多く持つということをよく反映している結果である．しかし，表 3.13 に示す例のように，人物以外のクラスーインスタンス関係も数多く抽出できている．例えば，日本の峠クラスのインスタンスは 3,180 個， アメリカ海軍駆逐艦クラスのインスタンスは 2,144 個，抽出できた．

**表 3.13 正しく抽出したクラスーインスタンスの例**

| クラス | インスタンス |
|---|---|
| 楽器 | ラッパ |
| 推理作家 | 松本清張 |
| 映画監督 | ジョージ・ルーカス |
| 国鉄・JRの車両形式 | クキ1000 |
| プログラミング言語 | Java |

**表 3.14 インスタンスの誤りの例**

| クラス | インスタンス |
|---|---|
| 言語 | :en:Ngumba language |
| 千葉県の神社 | 熊野神社（坂田） |
| 国際競技連盟 | 相撲 |
| 世界一 | ハヤブサ |
| スポーツ競技 | 野球 |

誤りは,スクレイピングのルールが不足していることによるものが大部分を占めていた.表 3.14 の 1 つ目と 2 つ目は，Wikipedia の言語リンクを表す“:(言語コード):”という記述を除去するスクレイピングのルールおよび“()” の注釈を除去するスクレイピングのルールが不足していたために生じた誤りの例である.表 3.14 の 3 つ目と 4 つ目は,“*”や“#”ではじまる行の中のどの部分がインスタンスを表しているかを特定するためのスクレイピングのルールが不足していたために生じた誤りの例である．スクレイピングのルールの不足以外にも，表 3.14 の 5 つ目の誤りのように is-a 関係を誤って抽出するケースも見受けられた.

## 3.4.3 プロパティ名の抽出結果と考察

**(1) Infobox トリプルからのプロパティ名の抽出結果と考察**

Wikipedia のダンプデータから 3.3.3 項(1)で提案した手法により，7,137 のプロパティ名と 1,962,411 のトリプルを抽出し，Infobox を持つ記事のうちトリプルの主語として抽出したインスタンス数は 171,190 であった．表 3.15 に Infobox トリプルから抽出したプロパティ名のうち，利用頻度が高い上位 5 つのプロパティ名を示す.

表 3.15 より,利用頻度が高いプロパティ名の多くは owl:ObjectProperty となっている.この原因として，英語記述からの変換過程において変換が十分ではなく，抽出できなかったものが多いことや，モデリングが不十分だったためにスクレイピングが適切に行えなかったことなどが考えられる. owl:DatatypeProperty の例としては“生年月日”プロパティ，“リリース”プロパティ，“資本金”プロパティ，“身長”プロパティなどがあった．これらの owl:DatatypeProperty は，Infobox テンプレートの利用頻度が高い，人物，Album，Single，会社，駅などに多く見られる.

**表 3.15 Infobox から抽出した，利用頻度が高い上位 5 つのプロパティ名**

| プロパティ名 | トリプル数 | プロパティのタイプ |
|---|---|---|
| 所在地 | 59,751 | owl:ObjectProperty |
| 本社所在地 | 36,373 | owl:ObjectProperty |
| 出身地 | 30,042 | owl:ObjectProperty |
| 生年月日 | 25,108 | owl:DatatypeProperty |
| ジャンル | 22,239 | owl:ObjectProperty |

全 7,137 のプロパティ名のうち，モデリングを行った 40 個の Infobox テンプレートから 313 のプロパティ名について，owl:ObjectProperty と owl:DatatypeProperty の分類ができた．これら 313 のプロパティ名を持つトリプル数は 1,329,549 であった．上記モデリングにより，67.8%のトリプルを分類できた．誤りの多くはスクレイピングミスであり，特にプロパティ値に URL が記述されている際のスクレイピングミスが多かった．また，モデリングにより分類できたプロパティ名であっても，“主要株主”プロパティの値のように，複数の値が混在するために誤りが生じるケースも見られた．（例えば，SONY の場合には，“主要株主”プロパティの値として，「Moxley and Company，日本トラスティ・サービス信託銀行（株）（信託口），State Street Bank and Trust Company」を抽出したが，これらは，3 つのトリプルに分けて抽出すべきである．）

**(2) 記事のリスト構造からのプロパティ名の抽出結果と考察**

Wikipedia のダンプデータから 3.3.3 項(2)で提案した手法により，3,980 のプロパティ名と 2,919,470 のトリプルを抽出し，トリプルの主語として抽出したインスタンス数は 233,247 個であった．表 3.16 に，記事のリスト構造から抽出したプロパティ名のうち，利用頻度が高い上位 5 つのプロパティ名を示す．

先の手法では抽出できない表 3.16 のような多くのトリプルを持つプロパティ名を抽出できた．2,919,470 のトリプルから 1,000 個の標本を抽出し，正解率の区間推定を行った．その結果，正解率の 95%信頼区間は，92.5±1.63%であった．誤りの多くはスクレイピングミスであり，リスト構造の各行に多くの情報が記述されている場合に誤ったトリプルを抽出している．表 3.16 のプロパティ名を例にとると，“収録曲”プロパティは歌手のアルバムやシングルの記事に多く見られるが，これらには収録曲以外にも作詞者や作曲者，リリース年といった情報も記載されている場合が多く，“収録曲”プロパティの値として作詞者や作曲者，リリース年が取れてしまっていた．しかし，記事が属するカテゴリごとに，こうした構造はいくつかに絞られるため，より詳細なリスト構造のルールを追加することで，これらの誤りを取り除く事ができる可能性がある．

表 3.17 に，3.3.3 項(1)と 3.3.3 項(2)の両方の手法により抽出したプロパティ数，トリプル数，主語となるインスタンス数，トリプルの正解率を示す．

**表 3.16 記事のリスト構造から抽出した，利用頻度が高い上位 5 つのプロパティ名**

| プロパティ名 | トリプル数 |
|---|---|
| スタッフ | 136, 033 |
| キャスト | 102, 617 |
| テレビドラマ | 70, 839 |
| 映画 | 69, 690 |
| 収録曲 | 66, 841 |

表 3.17 2 つの手法により抽出したプロパティ数，トリプル，主語となるインスタンス数，トリプルの正解率

| 手法 | プロパティ数 | トリプル数 | インスタンス数 | 正解率 |
|---|---|---|---|---|
| Infoboxトリプルから | 7,137 | 1,962,411 | 171,190 | 95.2 ± 1.33% |
| 記事のリスト構造から | 3,980 | 2,919,470 | 233,247 | 92.5 ± 1.63% |
| 2つの手法 | 10,769 | 4,867,882 | 319,742 | 94.3 ± 1.44% |

表 3.17 より，Infobox からの抽出法と記事のリスト構造からの抽出法を合わせ，重複を除外すると，10,769 のプロパティ名について，4,867,882 ものトリプルが抽出できている．全 4,867,882 のトリプルの正解率は 94.3 ± 1.44%であり，記事のリスト構造からの抽出法により，Infobox からの抽出法に比べ，多少正解率は下がったものの，プロパティ数として約 1.5 倍，トリプル数として約 2.5 倍も増加している．さらに，Infobox からの抽出法と記事のリスト構造からの抽出法により，重複を除外すると，319,742 個のインスタンスをトリプルの主語として抽出しており，Infobox を持たない 148,552 個の記事をトリプルの主語であるインスタンスとして追加できている．

## 3.4.4 プロパティ定義域の抽出結果と考察

3.3.3 項の手法により抽出した 10,769 のプロパティ名に対して，3.3.4 項で提案した手法を行った結果，9,486 のプロパティ定義域が抽出できた．Infobox から抽出したプロパティ名は Infobox テンプレートを定義域として持つため，全てのプロパティ名について定義域を定義でき，さらにリスト構造から抽出したプロパティ名のうち，1,888 のプロパティ名について定義域を定義できた．全体として，8,831 のプロパティ名について定義域を定義できたため，82%のプロパティ名が定義域を持つ事になる．9,486 のプロパティ定義域から 1,000 個の標本を抽出し，式①を利用して正解率の 95%信頼区間を算出した．その結果，正解率の 95%信頼区間は，94.8 ± 1.22%だった．表 3.18 にプロパティ名とプロパティ定義域のうち，主語となるインスタンスを多く持つ上位 5 つのプロパティ名と定義域を示す．

表 3.18 プロパティ名とプロパティ定義域の例

| プロパティ名 | プロパティ定義域 | 主語となるインスタンス数 |
|---|---|---|
| スタッフ | テレビ番組 | 26,251 |
| キャスト | ドラマ | 21,140 |
| スタッフ | ドラマ | 10,871 |
| 施設 | 道の駅 | 10,088 |
| 著書 | 文学 | 9,299 |

表 3.18 より，プロパティ定義域が複数定義されている場合に，それらの共通上位クラスが定義域として定義されていない問題があることがわかる．例えば，表 3.18 では“スタッフ”プロパティは“テレビ番組”クラスを定義域として持つが，“スタッフ”プロパティは“ドラマ”クラスも定義域として持っており，この他にも“ラジオ番組”，“野球チーム”など多くのクラスを定義域として持っている．このような複数の定義域を持つプロパティ名は多く存在しており，より上位のクラスに統合するべきである．

## 3.4.5 プロパティ値域の抽出結果と考察

**(1) クラスーインスタンス関係からの抽出法による結果と考察**

3.3.5 項(1)のクラスーインスタンス関係からの抽出法により，4,007 のプロパティ名について値域を定義でき，プロパティ名と値域の関係数は 14,053 であった．14,053 のプロパティ値域の関係数から 1,000 個の標本を抽出し，式①を利用して正解率の 95%信頼区間を算出した．その結果，正解率の 95%信頼区間は，88.3 ± 1.92%だった．表 3.19 に利用頻度が高い上位 5 つのプロパティ名と値域を示す．

表 3.19 より，“テレビアニメ”プロパティの値域として“深夜アニメ”や“キャスト”プロパティの値域として“日本の男優”などドメインに特化した値域が見られる．“国籍”プロパティの値域として“島国”が抽出されている理由は，日本語版 Wikipedia には日本人の記事が多く，これらの人物の多くは国籍として日本を持っており，さらに日本語 Wikipedia オントロジーのクラスーインスタンス関係において，日本というインスタンスが島国というクラスに属しているためである．誤りの例としては，“国籍”プロパティの値域として“世界各国の著作権保護期間”や“民族衣装”といったクラスが抽出されていたことが挙げられる．これは，Wikipedia の「世界各国の著作権保護期間」や「民族衣装」の一覧記事の記述において国名が箇条書きされており，クラスーインスタンス関係抽出において誤った関係を抽出してしまったことが原因である．プロパティ値域の定義における誤りの多くは，クラスーインスタンス関係定義の誤りから生じているため，クラスーインスタンス関係の精度を上げることで値域の精度も上がると考えられる．

**表 3.19 クラスーインスタンス関係を用いたプロパティ値域抽出法により抽出した利用頻度が高い値域の例**

| プロパティ名 | 利用インスタンス数 | プロパティ値域 |
|---|---|---|
| テレビアニメ | 23,195 | 日本の漫画作品 |
| キャスト | 20,633 | 日本の男優 |
| テレビアニメ | 15,956 | 深夜アニメ |
| キャスト | 12,821 | 日本の女優 |
| 国籍 | 11,569 | 島国 |

表 3.20 is-a 関係を用いたプロパティ値域抽出法により抽出した値域の例

| プロパティ名 | 利用インスタンス数 | プロパティ値域 |
|---|---|---|
| キャスト | 44,766 | 存命人物 |
| キャスト | 43,532 | 日本の俳優 |
| 出演者 | 22,175 | 存命人物 |
| 映画 | 16,370 | 日本の映画作品 |
| 出演者 | 12,236 | 日本の俳優 |

**(2) is-a 関係からの抽出法による結果と考察**

次に，3.3.5 項(2)の is-a 関係からの抽出法を用いて，値域の抽出を行った．3,234 のプロパティ名について値域を定義でき，35,946 のプロパティ名と値域の関係を抽出した．35,946 のプロパティ値域の関係数から 1,000 個の標本を抽出し，式①を利用して正解率の 95%信頼区間を算出した．その結果，正解率の 95%信頼区間は，92.1 ± 1.65%だった．表 3.20 に is-a 関係を用いたプロパティ値域抽出法により抽出したプロパティ値域のうち，トリプルを多く持つプロパティ名の上位 5 つを示す．

利用頻度が高い値域の殆どが“キャスト”プロパティなどの目的語として俳優を中心とした人物をインスタンスにするもの，“所属事業者”プロパティなどの目的語として鉄道駅に関するインスタンスを持つもの，“国籍”プロパティなどの目的語として国名をインスタンスとするものであり，そのため，値域も国，人物，鉄道関係のクラスとなるものが多い．しかし，クラスーインスタンス関係を用いた抽出法では抽出できない，“在籍チーム”プロパティの値域として“サッカークラブ”のような，より抽象的な値域が抽出されている事が特徴である．

誤りの例としては，“優勝回数”プロパティや“宿泊施設数”プロパティの値域として“数学に関する記事”が抽出されていた．これはモデリングが不十分なために生じた誤りであり，本来はプロパティタイプが owl:DatatypeProperty になるため，値域はリテラル(rdfs:Literal)となる．さらに，“収録曲”プロパティの値域として“存命人物”が定義されていた．これは 4.3.3 項(2)で述べたリスト構造のルール不足により，誤って抽出したトリプルが原因であり，このような大本のトリプルが原因となり抽出してしまった誤った値域は is-a 関係からの抽出だけでなく，クラスーインスタンス関係からの抽出にも見られる．さらに，定義域と同様の問題も存在する．表 3.20 では“キャスト”プロパティは“日本の俳優”クラスを値域として持つが，“キャスト”プロパティはこの他にも“イギリスの俳優”，“日本出身の人物”など多くのクラスを値域として持っている．定義域と同様に，今後は上位クラスへの統合を検討する必要があるが，“出演者”プロパティの値域である“存命人物”などあまりに抽象的すぎる概念になってしまっているものもあり，オントロジーとしての利用を考慮した際に，どのレベルまで統合するべきなのかも併せて検討する必要がある．


図 3.19 出現数 n と上位下位関係数及び正答率

## 3.4.6 プロパティ上位下位関係の抽出結果と考察

3.3.6 項で提案した手法により，2,322 の上位下位関係の候補を抽出し，そのうち定義域と値域が共通であるものは 1,387 であった．1,387 のプロパティ上位下位関係について，手作業ですべてのプロパティ上位下位関係の正誤を測定した結果，正答率は 57.5%であった．そこで，それぞれの上位下位関係について，上位下位関係が出現する記事数を数え，出現数と正答率の関係を計測した．図 3.19 に出現数 n 以上となる上位下位関係数及び正答率を示す．図 3.19 より，出現数 n と上位下位関係数は反比例をしているが，正答率は出現数が 18 以上の時に最も高く，75.7%となっている．表 3.21 に上位下位関係の例を示す．

“キャスト”や“スタッフ”といったテレビや映画に関するプロパティ上位下位関係が非常に多かった．また，“祭神－主祭神”や“作品－代表作”のような“主”や“代表”とった語を含む関係も多い．さらに，“関連会社－主要子会社”のように“関連”という語を含む関係も多い．

表 3.21 プロパティ上位下位関係の例

| 上位プロパティ名 | 下位プロパティ名 | 出現頻度 |
|---|---|---|
| キャスト | 出演者 | 2,082 |
| スタッフ | 監督 | 1,919 |
| スタッフ | 脚本 | 1,514 |
| 祭神主 | 祭神 | 237 |
| 関連会社 | 主要子会社 | 227 |

誤りの例としては，“主要株主－主な株主”のように，同じ意味となるプロパティ名を上位下位として抽出してしまっているものが最も多い．これは，Infobox に羅列された情報が記事内でも同義の見出し語として出現しており，プロパティ名抽出の際に別々のプロパティ名として抽出してしまったためである．また，“学科－学部”のように上位と下位が逆となっているものもあった．これは，そもそもの Infobox のテンプレートに学科という項目が無いために，記者は新たに学科項目を追加するのではなく，学部項目に学科を列挙するケースが多く，このため Infobox からのプロパティ名抽出の際に“学部”プロパティの値として各学科が抽出されており，それが影響している誤りである．

## 3.4.7 プロパティタイプの抽出結果と考察

3.3.7 項で提案した手法により，3.4.3 項で抽出した 4,867,882 のトリプルを用いて，10,769 のプロパティ名からプロパティタイプの推定を行う．

**(1) 対称関係プロパティの推定結果と考察**

はじめに，対称関係プロパティの推定を行った．3.3.7 項(1)で提案した手法により，全トリプルから対称関係が成立するトリプルのペアは 10,927 組であった．このトリプルから，415 の対称関係プロパティの候補を抽出した．415 の対称関係プロパティ候補を手作業ですべて正誤判定した結果，正答率は 45.1%であった．さらに，全トリプル数と対称関係として抽出したトリプル数の割合から対称関係プロパティの推定を行う．今回は既に手作業により対称関係プロパティを抽出しているが，この作業は次回以降のプロパティタイプの自動推定の際の指標となる．図 3.20 にトリプルが含まれる割合 x 以上となる対称関係プロパティ数及び正答率を示す．

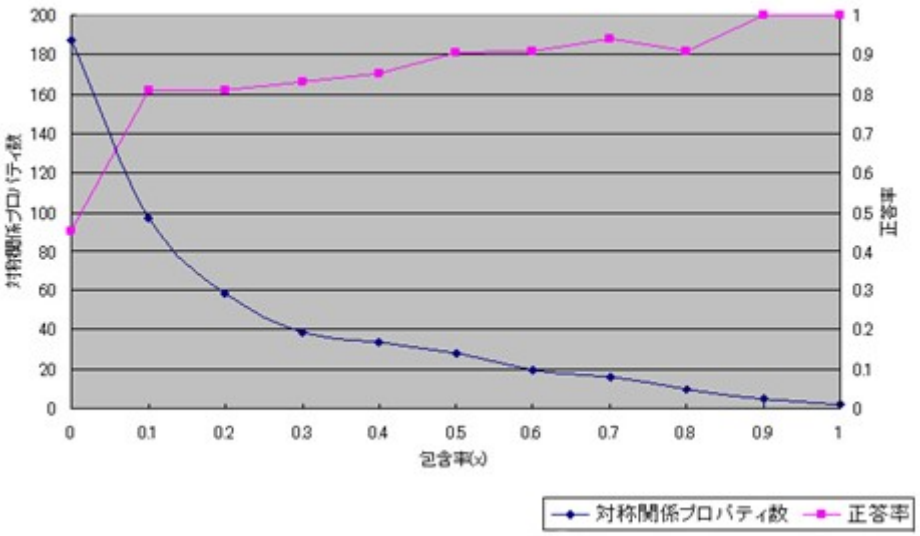



図 3.20 包含率 x と対称関係プロパティ数及び正答率

表 3.22 対称関係プロパティとその対称関係数，全トリプル数，包含率の一例

| 対称関係プロパティ | 対称関係数 | 全トリプル数 | 包含率 |
|---|---|---|---|
| 類似の表彰記章 | 30 | 30 | 1 |
| 隣接する星座 | 486 | 510 | 0.95 |
| 相方 | 198 | 360 | 0.55 |
| 関連学校 | 108 | 258 | 0.42 |
| 接続道路 | 2,720 | 11,016 | 0.25 |

図 3.20 より，包含率が 0.1 程度に増えると正答率は 8 割程度に上がるが，それ以降は包含率を増やしても正答率はあまり上がらないことがわかる．プロパティ数は包含率に反比例して減ることがわかる．表 3.22 に抽出した対称関係プロパティとその対称関係数，全トリプル数，包含率の一例を示す．

表 3.22 より，最も包含率が高かったものは“類似の表彰記章”プロパティであった．このプロパティは表彰記章記事に存在するプロパティであり，“消防庁長官表彰功績章－類似の表彰記章－消防庁長官表彰特別功労章”といったトリプルを作っている．本手法によって抽出した対称関係プロパティの多くは“隣接する正座”，“関連学校”，“接続する道路”のような“隣接”，“接続”，“関連”といった語を含むプロパティが多い．しかし，“相方”や“姉妹校”，“親族”のような上記の語を含まないプロパティも抽出されている．

さらに，包含率 0.5 以上のとき，抽出した 34 の対称関係プロパティについて，トリプルの補完を行った．ここで，トリプルの補完とは，対称関係プロパティを持つトリプルにおいて，一方向のトリプルから両方向のトリプルを定義する事である．この結果，新たに 55,887 のトリプルを抽出した．例えば，“奈良公園－周辺情報－奈良国立博物館”，“おとめ座－隣接する星座－うみへび座”のようなトリプルを補完できた．どちらのトリプルも 3.3.3 項の手法では一方向のトリプルしか抽出できておらず，実際の Wikipedia の「奈良公園」記事には周辺情報として奈良国立博物館は存在しておらず，「奈良国立博物館」記事にのみ存在している．「おとめ座」記事も同様である．

**(2) 推移関係プロパティの推定結果と考察**

次に，推移関係プロパティの推定を行った．3.3.7 項(2)で提案した手法により，全トリプルから推移関係が成立するトリプルは 340 組であった．この際，対称関係プロパティとなるものは除外している．このトリプルから，210 の推移関係プロパティの候補を抽出し，手作業ですべて正誤判定を行ったが，推移関係プロパティと思われるプロパティを見つける事ができなかった．包含率が最も高いものでも，わずか 3 割ほどしかなく，誤りの中には対称関係プロパティとなりうる“関連”や“隣接”といった語を含むプロパティも多い．推移関係数が最も多かったものは“トレーナー”プロパティで，その値は 54 であった．“トレーナー”プロパティはプロレスラーやボクサーといった格闘家全般の記事に見られるプ

ロパティであり，複数の人物に教わっている場合が多いために推移関係プロパティ候補として抽出されているが，“トレーナー”プロパティは必ずしも推移関係であるわけではないため，誤りである．

推移関係プロパティとなりうるものとして，“登山ルート”，“行程”といったプロパティが存在した．“登山ルート”プロパティは山岳記事に存在するプロパティであり，本来は“小蓮華山－三国境－鉢ヶ岳”のようなルートとして抽出するべきものであるが，3.2.1 項(2)の手法によって抽出したトリプルはそれぞれ分離しており，正式にプロパティ名を定義するならば“登山ルートでの通過点候補”のようなプロパティである．この場合，トリプルは推移しており，推移関係プロパティとなりうる．“行程”プロパティも同様である．

3.3.3 項の手法により抽出したプロパティには推移プロパティが存在しないという結果となった．このような結果の背景の 1 つとして，リスト構造や Infobox の構造からのプロパティ名抽出の限界が言えるのではと考えている．今回の手法により推移関係を抽出する場合は，推移関係となる少なくとも 3 つのトリプルが抽出されていなければならない．そのため，Wikipedia 記事内で Infobox もしくはリスト構造によりこれらの情報が同一のプロパティ名として，網羅されていなければならないが，こうした網羅された情報は非常に少ない．実際に，3.3.7 項の例として示した“後方互換”の場合，Wikipedia の Infobox 内でこの後方互換はその他の記事にもいくつか見られるが，トリプルとして 3 つのインスタンス間で網羅されているのは“PS”，“PS2”，“PS3”の組み合わせのみであり，“ゲームボーイ”，“ゲームボーイカラー”，“ゲームボーイアドバンス”も後方互換であるが，ゲームボーイカラー記事に“後方互換”の項目が存在しない．このため，今回の手法では推移関係プロパティとして抽出できなかった．推移関係プロパティを抽出するためには，プロパティ名を洗練し同一のものを統合する，より記事内部の構造化されていない部分に踏み込んだプロパティ抽出を試みるなどの対応が必要である．

**(3) 関数関係プロパティの推定結果と考察**

次に，関数関係プロパティの推定を行った．3.2.5 項(3)で提案した手法により，全トリプルから関数関係が成立するトリプルは 185,700 であった．このトリプルから，トリプルを 1 つしか持たないプロパティ名を除外した関数関係プロパティ候補は 2,267 であった．2,267 の関数関係プロパティ候補を手作業ですべて正誤判定した結果，正答率は 54.3%であった．関数関係プロパティの例として，最もトリプル数が多かったものは“投球・打席”プロパティであった．これは野球選手記事に存在するプロパティである．“都道府県”プロパティは市町村記事に，“毛色”プロパティは馬の記事に，“築城主”プロパティは城の記事に存在するプロパティである．これらはすべて関数関係となっており，プロパティの値としてインスタンスを唯一つ持っている．誤りの殆どは，実際には owl:DatatypeProperty となるべきプロパティであり，インスタンスではなく，リテラルとして値を持つべきプロパティであった．例えば，“総試合数”プロパティや“泉温”プロパティ，“着工年”プロパティが関数関係プロパティとして抽出してしまったが，これらは本来

owl:DatatypeProperty となるべきプロパティであり，3.3.3 項(1)の手法で owl:DatatypeProperty か owl:ObjectProperty に分類できなかったために，誤りとして影響を及ぼしている．

**(4) 逆関数関係プロパティの推定結果と考察**

最後に，逆関数関係プロパティの推定を行った．全トリプルから逆関数関係が成立するトリプルは 47,295 であった．このトリプルから，トリプルを 1 つしか持たないプロパティ名を除外した逆関数関係プロパティ候補は 3,670 であった．3,670 の逆関数関係プロパティ候補を手作業ですべて正誤判定した結果，正答率は 22.4%であった．非常に低い正答率となってしまっている理由として，関数関係プロパティと同様に owl:DatatypeProperty となるべきプロパティが抽出されてしまっていることが言え，この誤りが最も多かった．さらに，プロパティ名抽出の際のプロパティ名の定義が不十分である事も言える．プロパティの表記ゆれの問題に起因し，例えば，“主な作品”プロパティは人物全般に存在するプロパティであり，このプロパティは逆関数プロパティではないが，表記ゆれのプロパティ名として“おもな作品”プロパティも存在する．“おもな作品”プロパティはトリプルとしての抽出数が少なく，不幸にも全てのトリプルが逆関数関係となっていた．そのため，“主な作品”プロパティと同義であるはずの“おもな作品”プロパティは逆関数関係として抽出してしまっていた．このようなプロパティ名の表記ゆれや，先の推移関係プロパティの抽出の際の“登山ルート”プロパティのようなプロパティ名の定義が曖昧なために，トリプル数が少なく，逆関数関係として抽出してしまう誤りも多く，今後は，プロパティ名の表記の問題の対策をとる必要がある．正当な逆関数関係プロパティの例としては“主な所属アーティスト”，“主な所属タレント”，“収録作品タイトル”，“同州出身の有名人”などである．“主な所属アーティスト”や“主な所属タレント”プロパティは音楽会社や芸能事務所記事に存在するプロパティである．“収録作品タイトル”プロパティは DVD 記事や短編集記事に，“同州出身の有名人”は州記事に存在するプロパティである．“収録作品タイトル”プロパティとは別に“収録作品”プロパティや“同州出身の有名人”プロパティとは別に“出身有名人”プロパティも存在しており，正当な逆関数関係プロパティでもプロパティ名の表記ゆれの問題が垣間見える．

## 3.4.8 抽出関係の洗練

本項では，3.4.2，3.4.3 で抽出した以下の 2 つの関係を洗練することで，精度の向上を行う．

(1) クラスーインスタンス関係の洗練
(2) プロパティ定義域・値域の洗練

表 3.23 クラスーインスタンス関係の洗練結果の一例

| 元のクラス名 | 洗練後のクラス名 | 関係数 | 属するインスタンスの一例 |
|---|---|---|---|
| 日本の漫画作品 | 漫画作品 | 3,622 | ドラゴンボール，ONE PIECE |
| 日本の漫画家 | 漫画家 | 3,592 | 鳥山明，手塚治虫 |
| 日本のラジオパーソナリティ | ラジオパーソナリティ | 3,144 | 山谷親平，中村鋭一 |
| 東京大学の人物 | 人物 | 2,888 | 夏目漱石，鳩山邦夫 |
| 早稲田大学の人物 | 人物 | 2,605 | 福原愛，江戸川乱歩 |

**(1) クラスーインスタンス関係の洗練**

3.4.2 項で抽出したのクラス―インスタンス関係を使用し，3.3.8 項(1)で提案した手法により，378 のクラスと 131,235 の関係を洗練した．表 3.23 に洗練したクラス名のうち関係数が多い上位 5 つのクラスを示す．最も多くインスタンスを持つクラスは“日本の漫画作品”であった．これは漫画作品のうちアニメ化されたものの多くは“放送国”プロパティとその値“日本”をもつためである．このような国名や地名が格助詞「の」の前に来ているものは非常に多く，“日本”，“東京都”，“アメリカ合衆国”などがある．しかし，そのほかにも“東京大学”，“早稲田大学”などの学校名や“平安時代”，“戦国時代”などの時代名も多い．さらに，プロパティのトリプルとして新たに 12,051 の関係を補完した．トリプルの多くは“ビリー・ジョエル―国籍―アメリカ合衆国”や“江戸橋―都道府県―東京都”など，クラス名と同様に国名や地名が値となるものが多かった．しかし，“t.A.T.u.―ジャンル―ポピュラー音楽”や“FRONT MISSION―対応機種―プレイステーション”といったものも存在する．しかしながら，本手法は格助詞「の」に注目しているため，それ以外のクラス名については抽出できない点や格助詞「の」を含んでいても，トリプルの値としてその前方部分が完全一致しないため取りこぼす問題などがある．例えば，“NHK のアナウンサー”クラスは格助詞「の」を持ち，“NHK のアナウンサー”クラスに属するインスタンスは“放送局”プロパティを持っているが，その値は“NHK 山口放送局”などであり，NHK と完全一致しないため，本手法では洗練できない．手法を改良し，洗練数を増やすことが今後の課題といえる．

**(2) プロパティ定義域・値域の洗練**

3.3.8 項(2)で提案した手法をプロパティ定義域・値域に適用した．本手法を適用することで，定義域については，“党首”プロパティの定義域が洗練前は“日本の政党”，“台湾の政党”，“宗教政党”などであったのに対し，洗練後は“政党”クラスに，“国籍”プロパティや“身長”プロパティの定義域は“人物”クラスにリフトアップしている．値域についても，定義域に比べ非常に分散しているが，“接続道路”プロパティの値域が“道路”クラスに，“付属校”プロパティの値域が“幼稚園”クラスや“小学校”クラスにリフトアップしている．しかしながら，閾値としての兄弟クラスの占める割合を変えることでリフトア

ップの値は大きく変わってしまう．例えば，“著作”プロパティの定義域は“小説家”クラスなどの上位クラスである“著作者”クラスが妥当であるが，兄弟クラスが定義域としてすべて含まれるものは自動構築である日本語 Wikipedia オントロジーでは少ないため，兄弟クラスのうち定義域・値域として占める割合を閾値として設定している．そのため，この値が高ければあまりリフトアップが起こらず，低ければ先の例で言うと“人物”クラスにまでリフトアップされてしまうことがある．

図 3.21 は兄弟クラスに占める割合を変えた際のプロパティ定義域・値域の洗練結果である．ここで兄弟クラスが占める割合を変化させると，例えば，“背番号”プロパティの定義域が“野球選手”クラスであり，“野球選手”クラスの上位クラスに“スポーツ選手”，兄弟クラスに“テニス選手”があった場合，割合が 0.5 以上であればリフトアップは行わないが，0.5 より低い場合はリフトアップが行われ，“背番号”プロパティの定義域は“スポーツ選手”となる．なお，洗練前の定義域の関係数は 67,652，値域の関係数は 54,567 であった．図を見ると，定義域の減少率が値域に比べ高いことが分かる．値域は定義域に比べ同じプロパティ名でも値の概念が広く分散していることが主な原因である．日本語 Wikipedia オントロジーでのプロパティトリプルの主語は主に記事名に対応付けされており，必ず定義域を持つのに対し，値域は記事を持たないものも多い．そのため，抽出が不十分で，クラスーインスタンス関係や is-a 関係に定義されず，概念が分散してしまっていることが考えられる．閾値を低く設定すれば定義域で 5 割程度，値域で 6 割程度，関係数を減少させる事が可能であるが，先のような問題が生じてしまう．高く設定すれば，減少率は下がってしまうが，比較的この問題は除外できる．ただし，全く無くすということはできない．例えば“著名な出身者” プロパティの値域は Wikipedia に記事がある人物はまず間違いなく著名な人物であるので，値域が“人物”クラスの下位クラス全域に分散しており，“人物”クラスにまでリフトアップしてしまう．

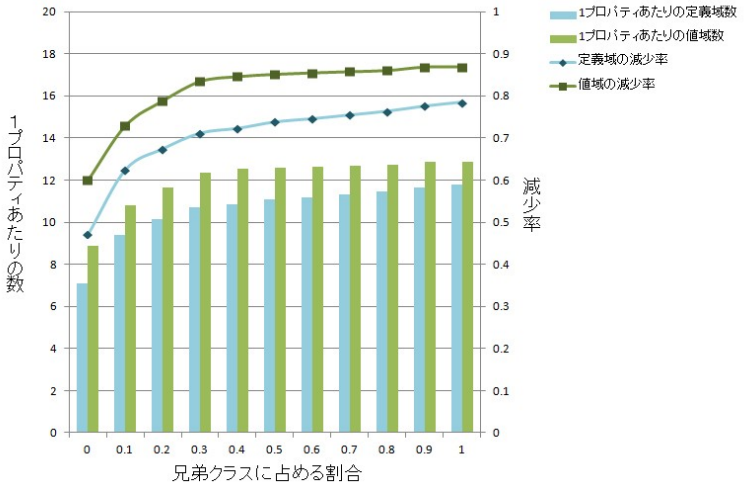



**図 3.21 プロパティ定義域・値域の洗練結果**

## 3.5 日本語 Wikipedia オントロジーの全体像

本節では，日本語 Wikipedia オントロジーの全体像について述べる．

表 3.24 に構築した日本語 Wikipedia オントロジーのクラス数，プロパティ数，インスタンス数を示す．表 3.25 に日本語 Wikipedia オントロジーの is-a 関係，クラス―インスタンス関係，トリプル，プロパティ定義域，同義語の各関係数および正解率の 95%信頼区間を示す．

**表 3.24 日本語 Wikipedia オントロジーのクラス数，プロパティ数，インスタンス数**

| クラス数 (owl:Class) | プロパティ数 (owl:Object/DatatypeProperty) | インスタンス数 |
|---|---|---|
| 87,159 | 10,769 | 323,024 |

**表 3.25 日本語 Wikipedia オントロジーの関係数と正解率**

| 関係の種類 | | 関係数 | 正解率 |
|---|---|---|---|
| is-a関係 (rdfs:subClassOf) | 全てのis-a関係 | 93,322 | 76.30% |
| | 文字列照合 | 12,558 | 93.1±1.51% |
| | Infoboxテンプレート照合 | 3,782 | 95.6±1.09% |
| | 目次見出し | 83,288 | 72.6±2.74% |
| クラスーインスタンス関係 | | 421,989 | 97.2±1.02% |
| トリプル | 全てのトリプル | 4,867,882 | 94.3 ± 1.44% |
| | Infoboxからの抽出 | 1,962,411 | 95.2 ± 1.33% |
| | リスト構造からの抽出 | 2,919,470 | 92.5 ± 1.63% |
| プロパティ定義域(rdfs:domain) | | 9,486 | 94.8 ± 1.22% |
| プロパティ値域(rdfs:range) | | 40,262 | 90.4 ± 1.81% |
| | クラスーインスタンス関係からの抽出 | 14,053 | 88.3 ± 1.92% |
| | is-a関係からの抽出 | 35,946 | 92.1 ± 1.65% |
| プロパティ上位下位 | | 1,387 | 57.5% |
| jwo語彙関係 | 上位下位関係(jwo:hyper) | 274,363 | - |
| | 関連語・同義語(jwo:nearly) | 258,853 | - |
| | 動詞とプロパティの対応 (jwo:verb) | 63,670 | - |
| 全関係 | | 6,031,214 | - |

表 3.24 および表 3.25 より，87,159 個のクラスについて，93,322 もの is-a 関係を抽出することができた．目次見出しからの is-a 関係抽出手法は，約 83,000 もの is-a 関係が抽出できているものの，正解率は約 72%と低く，is-a 関係全体の正解率を下げている．それ以外の手法により抽出した is-a 関係数は約 16,000，正解率は 90%以上で，高精度となっている．is-a 関係全体の正解率を上げるためには目次見出しからの is-a 関係抽出精度を上げる必要があり，今後の課題である．

次に，クラス階層のルートとなっている各クラス数とルートから全てのリーフのクラスへのパスを調べた．全ルートクラス数は 7,211，リーフ数は 65,721 であり，抽出したパスの本数は 257,313 本であった．構造全体の階層の深さの平均は約 5.83 本であった．さらにオントロジー全体を見渡すために，各ルートクラスについて派生するリーフの分布を測定した．横軸にルートクラスを，縦軸にクラスの階層の深さを取ったものが図 3.22 である．

図 3.22 を見ると分かるように，1 つのツリーに集約せず，小さなツリーが散在してしまっている．特に深い階層ができているツリーがいくつか見られるが，これは Wikipedia 主要カテゴリがルートとなっているツリーである．これらのことからも上位概念や中間概念が不足していることが分かる．

クラスーインスタンス関係は 421,989 もの関係を抽出し，インスタンス数も 323,024 と多く，また正解率も 97.2%と高い．しかし，これらは一覧記事から抽出したインスタンスであり，Wikipedia の記事数が現在，約 64 万記事あることを考えるとさらに多くのインスタンスを抽出できる可能性がある．

表 3.26 に構築した日本語 Wikipedia オントロジー内のプロパティタイプ別，プロパティ数，正答率，トリプル数を示す．


図 3.22 オントロジーの階層の深さとルートの関係

**表 3.26 日本語 Wikipedia オントロジーのプロパティタイプ別，プロパティ数，正答率，トリプル数**

| 種類 | プロパティ数 | 正答率 | トリプル数 |
|---|---|---|---|
| 全プロパティ | 10,769 | - | 4,867,882 |
| owl:DatatypeProperty | 214 | - | 416,803 |
| owl:ObjectProperty | 99 | - | 912,746 |
| owl:SymmetricProperty | 415 | 45.1% | 21,854 |
| owl:TransitiveProperty | 210 | 0% | 1,020 |
| owl:FunctionalProperty | 2,267 | 54.3% | 185,700 |
| owl:InverseFunctionalProperty | 3,670 | 22.4% | 47,295 |

表 3.25 および表 3.26 より，10,769 のプロパティ名を抽出することができ，トリプル数としては 4,867,882 ものトリプルを抽出できている．リスト構造からのトリプルの抽出精度は Infobox からの抽出に比べ低いものの，約 2 倍ものトリプルを抽出できており，全体としても約 94%と高精度で抽出できている．

プロパティ定義域は，9,486 の関係を 8,831 のプロパティ名について定義できており，82%のプロパティ名は定義域を持っていることとなる．プロパティ値域は，2 つの手法からあわせて 49,262 の関係を抽出でき，5,120 のプロパティ名について定義できており，48%のプロパティ名は値域を持っていることとなる．どちらも正解率は 90%程であり，高精度となっているが，半分以上のプロパティ名には値域を定義できておらず，定義されていない値域の定義が今後の課題である．さらに，定義域と値域が定義されているプロパティ名についても，複数の定義域や値域を持つものもあり，それらをどのように統合していくかも今後の課題である．

また，57.5%と精度は低いものの，1,387 のプロパティ上位下位関係を抽出しており，プロパティ間の上位下位関係の抽出は今までにない試みである．

さらに，プロパティタイプについてはこれまでの owl:Object/DatatypeProperty に加え，新たに，対称関係(owl:SymmetricProperty)，推移関係(owl:TransitiveProperty)，関数関係(owl:FunctionalProperty)，逆関数関係(owl:InverseFunctionalProperty)の推定を行った．そのままの抽出結果では精度は高くないものの，トリプルの包含率により絞り込む事により，特に対称関係プロパティは 8 割以上の精度で抽出できており，これらの更なる精度向上が今後の課題と言える．

## 3.6 まとめ

本章では，日本語版 Wikipedia を情報資源として，日本語版 Wikipedia から概念および概念間の関係（is-a 関係，クラスーインスタンス関係，プロパティ定義域，プロパティ値域，プロパティ上位下位関係，インスタンス間関係，その他の関係）を抽出し，自動構築により大規模かつ汎用的な日本語 Wikipedia オントロジーの構築手法の提案とその評価を行った．Wikipedia は，is-a 関係やクラスーインスタンス関係だけでなく，プロパティに着目する事で，多くの非階層な関係を抽出できる有用な情報資源であることを示すことができた．

提案手法の各パートに対して実験・評価をした結果，Wikipedia に対して提案手法を適用することで，is-a 関係，クラスーインスタンス関係，インスタンス間関係を高精度で抽出し，大規模で汎用的な日本語 Wikipedia オントロジーを構築することが可能であることがわかった．また，プロパティ定義域，プロパティ値域，プロパティ上位下位関係などのプロパティ関係を構築できたことで，クラススキーマ階層を構築できたと言え，上位下位関係のみのオントロジーや，手動でプロパティを構築しているオントロジー，プロパティ定義域・プロパティ値域を持たないオントロジーなど，他の関連研究より質の高いオントロジーの半自動構築ができたと言える．

今後の課題として，プロパティタイプなどの十分に抽出できなかった部分について，改善し，より精度の高いオントロジーの構築を目指す．また，Wikipedia において，本文には有用な情報が多くあり，このような構造化されていない部分から，オントロジー構築のためのルールを自動生成することで，さらなる規模の拡大は可能であり，今後の課題である．

# 第4章　日本語Wikipediaオントロジーの評価

## 4.1　概要

領域オントロジーは，特定の領域（法律やビジネスなど）に存在する概念とその間の関係を定義したものであり，ソフトウェアが RDF コンテンツを理解する際に，辞書的な役割を果たす．しかしながら，領域オントロジーの構築と保守には専門家を交えたインタビューなどを行うことで概念を列挙し，関係を定義するというプロセスを伴う．そのため，多大なコストがかかる．そこで，多くの研究は，知識工学，自然言語処理，データマイニングなどの技術を用いて，テキストや汎用オントロジーなどの既存情報資源から（半）自動的に領域オントロジーを構築している[14, 15]．日本語 Wikipedia オントロジーは汎用オントロジーであるが，特定の領域に存在する概念や関係を再利用することで，構築コストを削減できる可能性があり，領域オントロジー構築のための情報資源になりうる．

加えて，セマンティック Web の研究分野では，近年，各 Web サイトで公開されている政府，科学，写真，音楽などのデータベースを RDF 化して連携する，LOD (Linked Open Data)が注目を集めている．各データベース間の情報を繋げることで，情報を容易に引き出してくる事が可能であり，これにより多くのアプリケーションやサービスでデータを簡単に参照し，利用することができる．海外の LOD では，各 RDF データベース間を相互にリンクするためのハブとして，英語版 Wikipedia から自動構築した DBpedia [13]と呼ばれるオントロジーおよび RDF データが活用されている．

一方，LOD の語彙に着目した LOV (Linked Open Vocabularies) [16]という取り組みも存在している．各 LOD で使用されているプロパティを集めて，語彙の検索を可能にすることで, 新たな LOD を構築する際に語彙の再利用を促す取り組みである. しかしながら，LOD を構築する際に，新たにプロパティを作ってしまう方が，目的に合致するプロパティを見つけてくるよりもはるかに容易であり，標準語彙と呼ばれる，既に普及している一部の語彙を除いて，再利用されているケースは少ない．加えて，国内では Linked Open Vocabularies に相当する取り組みがまだ存在しておらず，日本語の標準語彙というものがないため, 今後さらに国内の LOD が広がるために, LOD 構築者にとって障壁となりうる．

以上により，本論文では大規模で汎用的なオントロジーである日本語 Wikipedia オントロジーを, 領域オントロジー構築支援としての利用および LOD ハブとしての利用という 2 つの視点から評価を行い，その有用性を示す．

以降，本章の構成は次のとおりである．4.2 節では，いくつかの領域に限定し，各領域のクラス，インスタンス，プロパティの関係を示す事で定性的に日本語 Wikipedia オント

ロジーの領域オントロジー構築支援としての有用性を評価する．4.3 節では，日本語 LOD としての設計と公開方法を述べる．4.4 節では，日本語 Wikipedia オントロジーのプロパティと Linked Open Vocabularies の語彙の対応付けによる日本語語彙構築手法を述べる．4.5 節では，4.4 節で述べた手法の結果と考察，代表的な LOD ハブである DBpedia との比較, 検索支援ツール WiLD の設計と評価により, 日本語 Wikipedia オントロジーの LOD ハブとしての有用性を示す．最後にまとめと今後の課題について述べる．

## 4.2 領域オントロジー構築支援

既存の汎用オントロジーとの比較と，いくつかの領域に限定し，各領域のクラス，インスタンス，プロパティの関係を示す事で，定性的に日本語 Wikipedia オントロジーの領域オントロジー構築支援としての有用性の評価を行った．限定した領域は水力発電，人物(作家クラス)，土地(都市クラス)，抽象物(過去など)である．水力発電領域については実際に専門家から意見を頂いている．

### 4.2.1 汎用オントロジーとの比較

日本語 Wikipedia オントロジーと代表的な既存汎用オントロジーである，日本語語彙体系および日本語 WordNet との比較評価を行った．表 4.1 に日本語 Wikipedia オントロジーと既存汎用オントロジーの比較例を示す．表 4.1 中の「Wiki」は本論文で構築した日本語 Wikipedia オントロジーを，「W」は日本語 WordNet を，「N」は日本語語彙体系をそれぞれ表している．また，「Path」はいくつかの主要クラスからリーフまでのパスを表している．

日本語 Wikipedia オントロジーの特徴として，表 4.1 の例の「ジャズ・ギタリスト」や「イギリスの空対空ミサイル」クラスのように，特定分野に特化して詳細なクラス階層を定義している点があげられる．さらに，日本語 Wikipedia オントロジーのクラスは膨大なインスタンスを持っている．これらは，他の既存汎用オントロジーにはない，日本語 Wikipedia オントロジーの特徴である．

表 4.1 の「事物－人物」の関係のように，構築した日本語 Wikipedia オントロジーは他と比べて上位概念が不足していることが分かる．これは，Wikipedia のカテゴリが 9 種の主要カテゴリから分類されているために構築したオントロジーもこの主要カテゴリをルートとした階層関係になっているためである．また，「人物」から「ギタリスト」クラスまでのパスの深さが浅いことから，中間概念が不足していることもわかる．

表 4.1 オントロジー比較の例

| クラス | オントロジー | Path |
|---|---|---|
| 人(音楽家) | Wiki | 事物－人物－音楽家－演奏家<br>－ギタリスト－ジャズ・ギタリスト |
| | W | ものー全般－生き物－生物－人－エンターテイナー<br>－公演者－ミュージシャン－ギタリスト |
| | N | 名詞－具体－主体－人－人職業地位役割<br>－人職業－人専門的技術的職業－芸術家－音楽家 |
| 無生物(兵器) | Wiki | 文化と歴史－出来事－政治－行政－軍事－兵器－航空兵器<br>－空対空ミサイルーイギリスの空対空ミサイル |
| | W | もの－全般－出土品－機器－機器－装甲－ミサイル－空対空ミサイル |
| | N | “兵器”の類はなし |
| 抽象物(過去) | Wiki | “過去”は未定義 |
| | W | 属性－時－古－過去 |
| | N | 名詞－抽象－抽象的関係－時間－非暦日－現在過去未来－過去 |

## 4.2.2 水力発電領域

知識マネジメントに特化したオントロジー構築ツール General knowlEdge Navigator (GEN)により構築した水力発電所領域のオントロジー[55]と，日本語 Wikipedia オントロジーの水力発電領域について，専門家による評価を行った．図 4.1 に GEN の設備オントロジーの一部を示す．また，図 4.1 中の水系，発電機，水車，変電所，発電所の各概念について，日本語 Wikipedia オントロジーから得た概念を図 4.2 に示す．

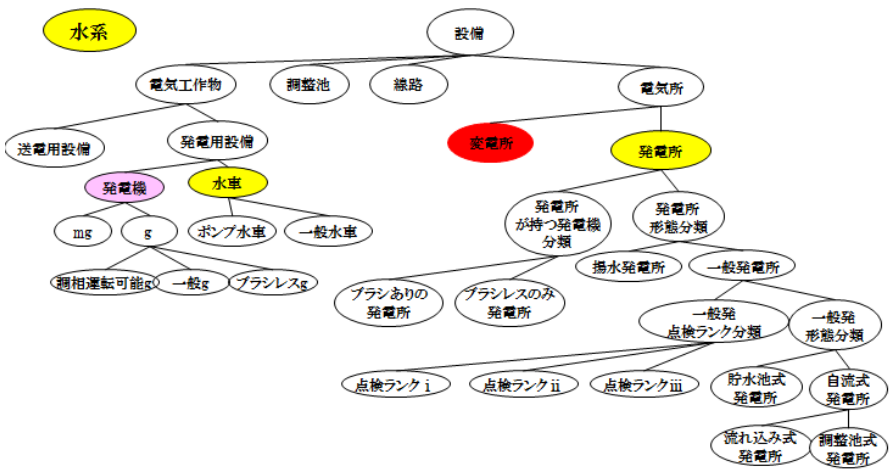



図 4.1 GEN の設備オントロジーの一部

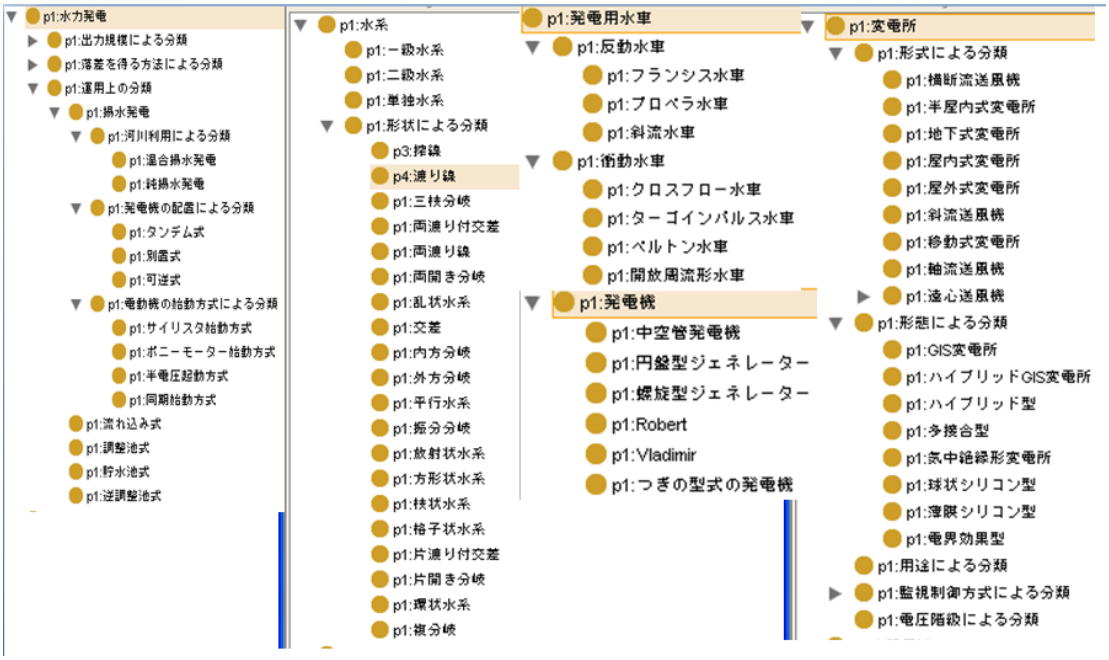



**図 4.2 日本語 Wikipedia オントロジーの水力発電領域に関する概念**

以下に専門家による各概念での評価を記す.

- 水力発電

  おおむね良くできている．ただし，発電機の配置による分類には疑問が残る.

- 発電用水車

  よくできている.

- 水系

  国土交通省政令に決められている分類に従っている．形状による分類は，電力会社ではあまり使っていない.

- 発電機

  専門家の聞いたことの無い単語ばかりでよく分からない.

- 変電所

  形式による分類において，送風機関係が入っているのはおかしい．地上か地下かという分類と，送風機の形式は同一の軸で扱うものでない.

  形態による分類において，いろいろな概念が混ざっている．GIS は変電所の中の遮断機の形式としてよく出てくる．XX シリコン型は，小さな変圧器に関するもので，電力会社の変電所の用語ではない.

以上の評価から，おおむね良くできているという評価をいただいた．一部概念において

は不明な点や，専門家にとって違和感がある分類はあるが，今回評価をいただいた半分以上の概念については再利用が可能であると考えられる．

### 4.2.3 人物領域

図 4.3 に作家クラスのインスタンスである“芥川龍之介”と“夏目漱石”に関するクラス，インスタンス，プロパティ関係の一部を示す．

多くの関係が定義されている事が分かる．人物ドメインのプロパティの多くは owl:ObjectProperty となっており，プロパティの値がインスタンスとなるので，さらにその先の関係へと連結されている．図 4.3 の例では，“夏目漱石”は“門下生”として“芥川龍之介”を持っており，さらに，“芥川龍之介”の“親族”である“芥川也寸志”は“作曲家”である事がわかる．このようにインスタンスとインスタンスのつながりから，“作家”クラスから離れている“作曲家”へ関係が繋がっているように，様々な関係がネットワーク構造によって広がっている事が分かる．

さらに，“親族”プロパティと“配偶者”プロパティや，“代表作”プロパティと“著作”プロパティのようにプロパティの上位下位関係が成り立っている．

しかし，“日本”クラスと“日本”インスタンスが顕在しているように，クラスでありインスタンスである概念が存在している，そもそもの抽出の誤り，などが原因のために，定義域や値域が妥当でないものも見られ，今後の課題といえる．

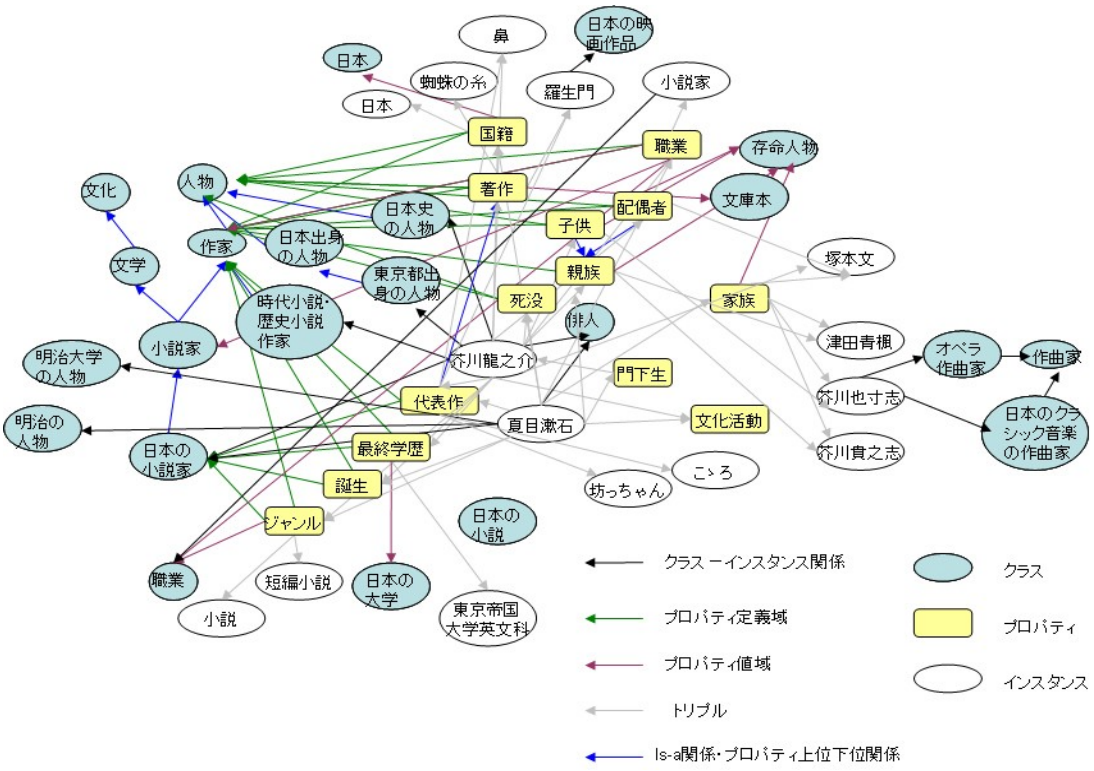



図 4. 3 人物(作家クラス)領域の一部

### 4.2.4 都市領域

図 4.4 に都市クラスのインスタンスである“パリ”に関するクラス，インスタンス，プロパティ関係の一部を示す.

人物ドメインに比べ，土地ドメインは人口や面積など owl:DatatypeProperty となるプロパティが多く存在しており，関係のつながりは少ない．しかし，“姉妹都市・提携都市”プロパティや“スポーツ”プロパティのように，インスタンスと結びつくプロパティを持っており，全くつながりが無いわけではない．土地ドメインは多くがインスタンスでありクラスとなっており，さらに“パリ”の場合は“フランスの音楽学校”クラスに地名と同名のインスタンスが存在しているために，is-a 関係として誤った関係が多い.

### 4.2.5 抽象的な概念の領域

“過去”や“現在”などの抽象的な概念を日本語 Wikipedia オントロジーで探したが，見つける事ができなかった．この理由として，具体物に比べ，抽象物は評価基準や数量的な定義が難しく，Wikipedia の記事として，あまり詳細に記述されない事や，人物などのようにクラスーインスタンス関係として表現する事が難しい事がいえる.

“時間”という抽象物を見つけたが，日本語 Wikipedia オントロジー内では“単位”クラスのインスタンスとして定義されており，抽象的な概念としての“時間”とは少し違っていた. 日本語 Wikipedia オントロジーは中間概念や上位概念が欠落している傾向があり，このような抽象物は上位概念に多く存在するため，如何に上位概念と中間概念を補完するかが今後の課題といえる.

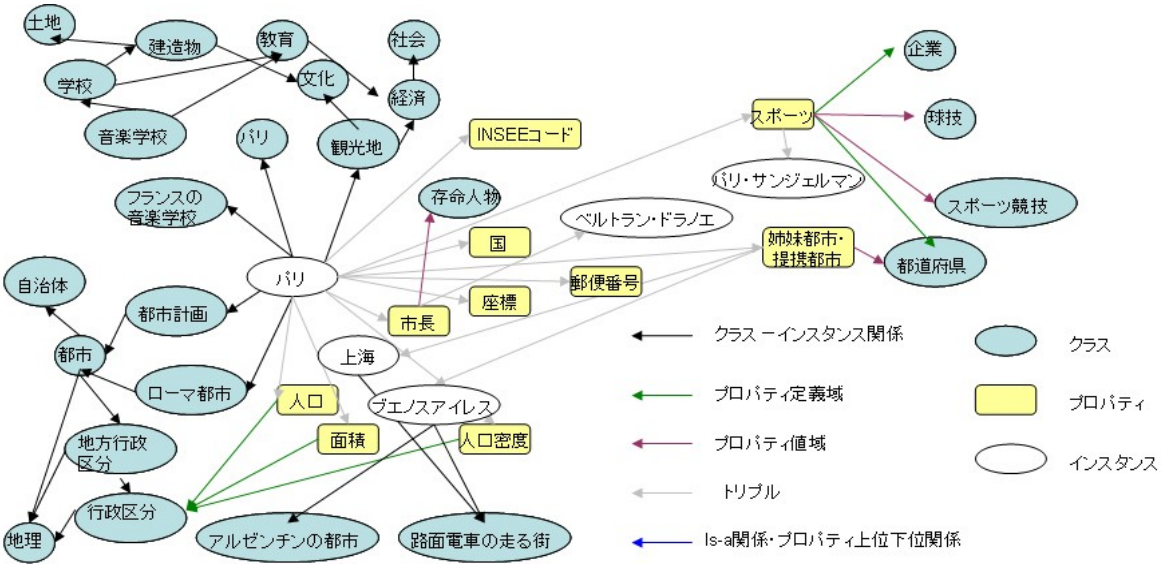



図 4. 4 土地(都市クラス)領域の一部

## 4.3 日本語 Wikipedia オントロジーLinked Open Data

日本語 Wikipedia オントロジーを日本語 Linked Open Data ハブとして利用するため，LOD として設計と公開を行う．本節では LOD 設計と公開方法及び日本語 Wikipedia オントロジーのプロパティと Linked Open Vocabularies の語彙の対応付けによる日本語語彙構築手法を述べる．

### 4.3.1 日本語 Wikipedia オントロジーLOD の設計と公開

図 4.5 にシステムの概要図を示す．LOD として公開するにあたり，RDF ストアとして Virtuoso28を利用しており，SPARQL クエリは Virtuoso を通して結果が返ってくる．各リソースのウェブページとデータはできるだけ，メモリ及びキャッシュに保存することで高速に表示するようにしている．下記の全ての機能は，日本語 Wikipedia オントロジープロジェクトページより利用することができる．

日本語 Wikipedia オントロジープロジェクトページ: http://wikipediaontology.org/

2013 年 10 月時点の日本語 Wikipedia オントロジーの最新バージョンと統計情報を図 4.6 に示す．

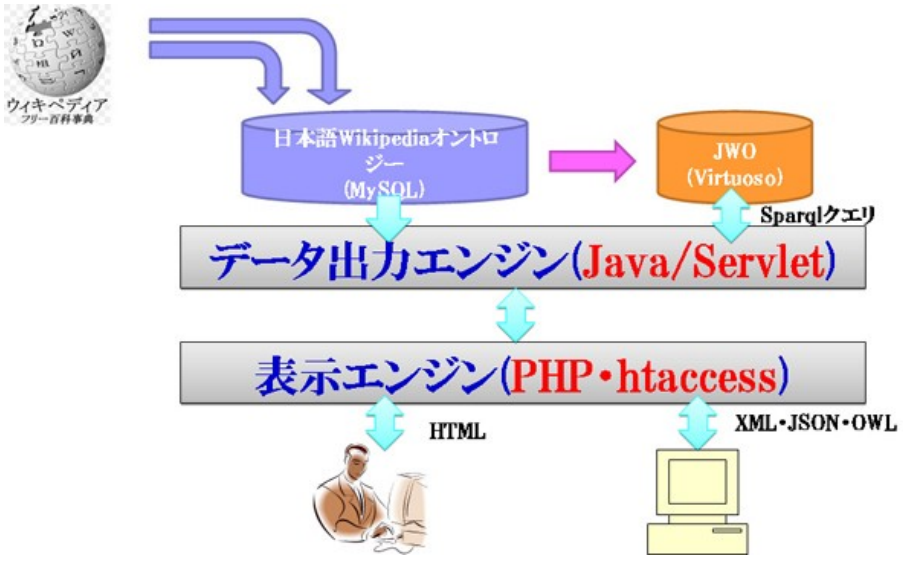



図 4.5 日本語 Wikipedia オントロジーLOD のシステム概要図

[28] http://virtuoso.openlinksw.com/

| 統計情報 | |
|---|---|
| クラス数 | 162,407 |
| インスタンス数 | 1,867,071 |
| プロパティ数 | 25,266 |
| クラスを持つインスタンス数 | 700,104 |
| is-a関係数(rdfs:subClassOf) | 58,954 |
| タイプの数(rdf:type) | 1,013,926 |
| 定義域関係数(rdfs:domain) | 30,533 |
| 値域関係数(rdfs:range) | 72,308 |
| プロパティ上位下位関係数(rdfs:subPropertyOf) | 303 |
| 上位下位関係数(jwo:hyper) | 274,363 |
| 関連語・同義語(jwo:nearly) | 258,853 |
| 動詞とプロパティの対応数(jwo:verb) | 63,670 |
| プロパティトリプル数 | 10,064,292 |
| Infoboxトリプル数 | 3,006,812 |
| 外部への参照数(owl:sameAs) | 1,048,957 |

**図 4.6 日本語 Wikipedia オントロジー統計情報（20130530 版）**

日本語 Wikipedia オントロジーLOD の設計にあたり，以下の 3 点に注意した．これらは Linked Data Design Issue 及び Linked Data のための 5 つ星の原則に則っている．

(1) URI の定義
(2) 他の LOD との関連付け
(3) SPARQL エンドポイントの公開

**(1) URI の定義**

外部からの参照を可能にするため，日本語 Wikipedia オントロジーの全てのクラス・インスタンス・プロパティについて，HTTP URI を付加する．表 4.2 に HTTP URI の一覧を示す．

各リソースは「/」以下に日本語もしくは URI エンコードされた日本語を入力することでアクセス可能である．301 リダイレクトにより，ブラウザからのアクセスは「page」へ，アプリケーションからのアクセスは「data」へアクセスする．現在選択できるデータの種類は rdf，owl，rdf/json，json の 3 種類である．例えば，

「http://www.wikipediaontology.org/instance/福澤諭吉」へブラウザからアクセスした場合は「http://www.wikipediaontology.org/pages/instance/福澤諭吉」へリダイレクトされる．福澤諭吉の URI エンコードである

「%E7%A6%8F%E6%BE%A4%E8%AB%AD%E5%90%89」へアクセスした場合も同様である．

表 4.2 日本語 Wikipedia オントロジーURI

| リソース | | URI |
|---|---|---|
| インスタンス | URI | http://www.wikipediaontology.org/instance/ |
| | ページ | http://www.wikipediaontology.org/pages/instance/ |
| | データ | http://www.wikipediaontology.org/data/instance/ |
| クラス | URI | http://www.wikipediaontology.org/class/ |
| | ページ | http://www.wikipediaontology.org/pages/class/ |
| | データ | http://www.wikipediaontology.org/data/class/ |
| プロパティ | URI | http://www.wikipediaontology.org/property/ |
| | ページ | http://www.wikipediaontology.org/pages/property/ |
| | データ | http://www.wikipediaontology.org/data/property/ |
| Infoboxプロパティ | URI | http://www.wikipediaontology.org/infobox/ |
| | ページ | http://www.wikipediaontology.org/pages/infobox/ |
| | データ | http://www.wikipediaontology.org/data/infobox/ |

### (2) 他の LOD との関連付け

日本語 Wikipedia オントロジーのインスタンスと DBpedia Japanese[29], LODAC[30], 青空文庫[31], saveMLAK[32]のリソースの関連付けを行う．日本語 Wikipedia オントロジー内のインスタンスと各 LOD のリソースの文字列照合を行い，完全照合した場合に owl:sameAs によって対応付けを行う．表 4.3 に関連付けの一例を示す．

表 4.3 他の LOD リソースとの関連付けの一例

| 日本語Wikipedia オントロジーURI | 関連先URI |
|---|---|
| http://www.wikipediaontology.org/instance/福澤諭吉 | http://ja.dbpedia.org/resource/福澤諭吉 |
| http://www.wikipediaontology.org/instance/福澤諭吉 | http://www.aozora.gr.jp/index_pages/person296.html |
| http://www.wikipediaontology.org/instance/吾輩は猫である | http://www.aozora.gr.jp/cards/000148/card789.html |
| http://www.wikipediaontology.org/instance/ギアナウズラ | http://lod.ac/species/ギアナウズラ |
| http://www.wikipediaontology.org/instance/慶應義塾普通部 | http://savemlak.jp/wiki/慶應義塾普通部 |
| http://www.wikipediaontology.org/instance/東京都立大島高等学校 | http://savemlak.jp/wiki/東京都立大島高等学校 |
| http://www.wikipediaontology.org/instance/落穂拾い | http://lod.ac/id/497029 |

[29] http://ja.dbpedia.org/
[30] http://lov.okfn.org/dataset/lov/
[31] http://www.aozora.gr.jp/
[32] http://lov.okfn.org/dataset/lov/

```
SELECT ?prop ?value
WHERE
{
   <http://www.wikipediaontology.org/instance/福澤諭吉> ?prop ?value.
}
```

**URL**

**http://www.wikipediaontology.org/query/?q=SELECT+%3Fprop+%3Fvalue%0D%0AWHERE%0D%0A{%0D%0A+++%3Chttp%3A%2F%2Fwww.wikipediaontology.org%2Finstance%2F%E7%A6%8F%E6%BE%A4%E8%AB%AD%E5%90%89%3E+%3Fprop+%3Fvalue.%0D%0A}%0D%0A%0D%0A&type=xml&limit=100**

図 4.7 SPARQL クエリの一例

### (3) SPARQL エンドポイントの公開

SPARQL エンドポイントは「http://www.wikipediaontology.org/query/」である．図 4.7 上部のような SPARQL クエリを入力する場合，図 4.7 下部のような URL にアクセスすることで，xml 形式でデータを得ることができる．

ブラウザを利用し，表示した場合の一例を図 4.8 に示す．リソース名(図 4.8 では「福澤諭吉」インスタンス)を主語とした「主語-述語-目的語」のトリプルが一覧で表示される．

また，図 4.9 のように検索ページから概念の検索が可能である．入力語に完全一致するリソース，部分一致するリソース，入力語を目的語とするリソースを関連候補として，順に表示している．

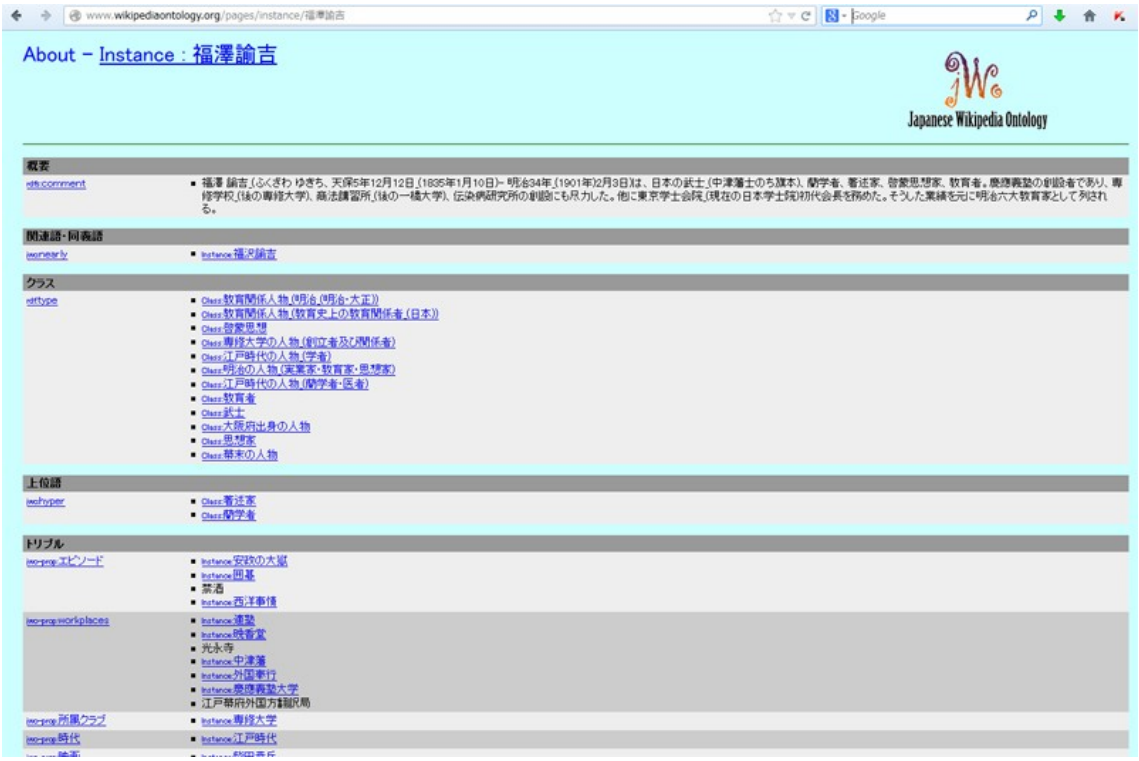



図 4.8 HTTP ページの一例(福澤諭吉インスタンス)


図 4.9 検索実行結果の一例

## 4.4 日本語 Wikipedia オントロジーからの日本語語彙構築

本節では，日本語 Wikipedia オントロジーのプロパティと Linked Open Vocabularies の語彙の対応付けによる日本語語彙抽出手法を述べる．

### 4.4.1 Linked Open Vocabularies からのプロパティ抽出

Linked Open Vocabularies には 2013 年 10 月の時点で 370 の語彙が登録されている．これらの全ての語彙について，以下の手順でプロパティの抽出を行う．データベースとして Virtuoso を使用し，全てのデータ取得は SPARQL クエリを通して行っている．

(1) RDF/XML，Turtle，N-Triple，N3 のいずれかの形式でスキーマを取得
(2) (1)で取得したファイルをデータベースに入力
(3) (2)で入力したデータから SPARQL クエリにより，
rdf:type が rdf:Property，owl:ObjectProperty，owl:DatatypeProperty，owl:TransitiveProperty，owl:SymmetricProperty，owl:FunctionalProperty，owl:InverseFunctionalProperty のいずれかになるプロパティを抽出

ここで取得したプロパティを用いて，日本語 Wikipedia オントロジープロパティとの対応付けから日本語語彙を構築する．

### 4.4.2 日本語 Wikipedia オントロジープロパティとの対応付け

日本語 Wikipedia オントロジーに存在する 10,769 のプロパティと 4.4.1 項で抽出したプロパティの対応付けを行う．しかしながら，これらのプロパティを手作業で対応付けることは困難である．そこで，プロパティの数に比べ，それらの定義域及び値域の数のほうが少なく，さらに日本語 Wikipedia オントロジーのプロパティが定義域と値域を持っていることに着目し，以下の手順で対応付けを行う．

(1) 定義域もしくは値域を持つプロパティを抽出
(2) (1) で抽出したクラス名と日本語 Wikipedia オントロジーのクラス名を対応付け
(3) 日本語 Wikipedia オントロジーのプロパティのうち，定義域もしくは値域が照合されたプロパティを日本語語彙の候補として抽出

上記(2) のクラス名の対応付けは，手作業で，英語であるクラス名を日本語に翻訳し，日本語 Wikipedia オントロジーのクラスに同様のクラスが存在する場合は owl:sameAs によりリンクするという処理を行う．図 4.10 に日本語 Wikipedia オントロジーのプロパティとの対応付けの具体例を示す．schema.org の“Person”クラスと日本語 Wikipedia オントロジーの“人物”クラス，schema.org の“Country”クラスと日本語 Wikipedia オントロジーの“国” クラスの対応付けを行うことで，定義域として“Person”クラスを持ち，値域として“Country”クラスを持つ schema.org の“nationality”プロパティの候補として日本語 Wikipedia オントロジーの“国籍”プロパティを抽出している．

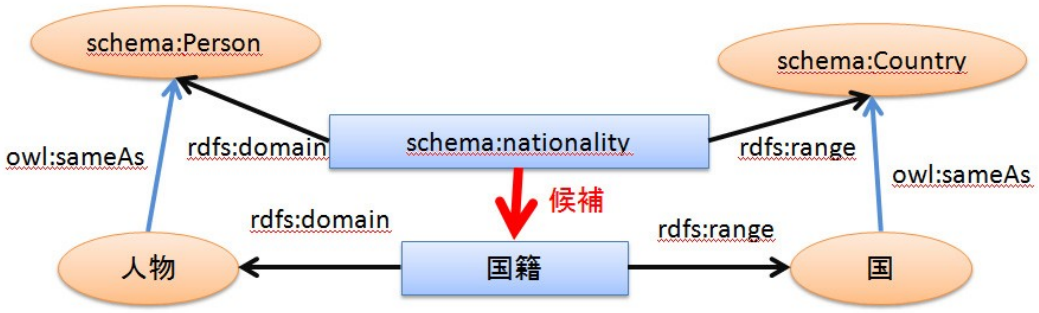



図 4.10 日本語 Wikipedia オントロジーのプロパティと語彙の対応付けの一例

表 4.4 Linked Open Vocabularies に存在するタイプごとのプロパティ数

| タイプ(rdf:type) | プロパティ数 |
|---|---|
| rdf:Property | 4,756 |
| owl:ObjectProperty | 7,190 |
| owl:DatatypeProperty | 7,034 |
| owl:TransitiveProperty | 116 |
| owl:SymmetricProperty | 135 |
| owl:FunctionalProperty | 654 |
| owl:InverseFunctionalProperty | 54 |

## 4.5 日本語 Wikipedia オントロジーLinked Open Data の評価

本節では，4.4 節で述べた手法の結果と考察，代表的な LOD ハブである DBpedia との比較, 検索支援ツール WiLD の設計と評価により, 日本語 Wikipedia オントロジーの LOD ハブとしての有用性を示す.

### 4.5.1 日本語 Wikipedia オントロジーからの日本語語彙構築結果と考察

**(1) Linked Open Vocabularies からの日本語語彙構築結果と考察**

4.4.1 項の結果，341 の語彙から 19,939 のプロパティを取得した．表 4.4 に各タイプのプロパティ数を示した．owl:ObjectProperty と owl:DatatypeProperty はほぼ同数であることがわかる.

**(2) 日本語 Wikipedia オントロジープロパティとの対応付け結果と考察**

- **クラスの対応付け結果と考察**

19,939 の語彙のうち，定義域もしくは値域を持つ語彙は 15,579 であった．これらの定義域および値域となるクラスは 3,287 であり，3,287 のクラスから各語彙の名前空間を除外し，クラス名のみとした場合，2,739 のクラス名が存在した．この 2,739 のクラス名のうち, 575 のクラスについて日本語 Wikipedia オントロジーのクラスと対応付けを行った. 表 4.5 が対応付けの一例である.

表 4.5 日本語 Wikipedia オントロジークラスとクラス名の対応付けの一例

| クラス名 | 日本語Wikipediaオントロジークラス | 名前空間の例 |
|---|---|---|
| Person | 人物 | http://schema.org/Person |
| Organization | 組織 | http://xmlns.com/foaf/0.1/Organization |
| Country | 国 | http://schema.org/Country |
| CreativeWork | 作品 | http://schema.org/CreativeWork |
| ProductOrService | 製品，サービス | http://purl.org/goodrelations/v1#ProductOrService |

表 4.5 の“Person”，“Country”，“Organization”のようなクラスは単語を翻訳するだけで，日本語 Wikipedia オントロジー内にもクラスが存在するため，対応付けが容易であった．しかし，“CreativeWork”のようなただ翻訳しただけでは存在しないものや，“ProductOrService”のような日本語 Wikipedia オントロジーでは複数のクラスを合わせたものも存在していた．しかしながら，これらは抽象度の高いクラスであり，日本語 Wikipedia オントロジーにも似たようなクラスが存在しているため，対応付けが可能である．対応付けが難しい例として，

“ http://bonubase.com/dicom/dicom.owl#SequenceItem.IonMachineVerificationSequence”などが挙げられる．ヘルスケアに関する語彙である Healthcaremetadata - DICOM ontology[33]において，

“http://purl.org/healthcarevocab/v1#RecordedRangeModulatorSequence”プロパティなどの定義域となるクラスであるが，汎用的な日本語 Wikipedia オントロジーには似たようなクラスは存在していないため，対応付けができない．さらに，データ品質管理に関する語彙である The Data Quality Management Vocabulary[34]に出現する

“http://purl.org/dqm-vocabulary/v1/dqm#DataQualityScore”などの単語だけではどのクラスと対応付けすべきなのかわからないクラス名も対応付けが難しい．このようなクラスは専門性が高い語彙に多く存在し，日本語 Wikipedia オントロジーは汎用的なオントロジーであり，専門性の高いプロパティやクラスが存在していないケースが多く，対応付けが困難である．

- **語彙候補の抽出結果と考察**

対応付けを行ったクラスを用いて，定義域・値域から日本語 Wikipedia オントロジーのプロパティを日本語語彙の候補として抽出した結果，15,579 のプロパティのうち，4,094 のプロパティについていずれかの日本語 Wikipedia オントロジープロパティを候補として推薦できた．約 26%のプロパティは候補を持つことになる．そのうち，903 のプロパティがタイプとして owl:ObjectProperty を持つものであった．全てのプロパティ候補数は

[33] http://purl.org/healthcarevocab/v1
[34] http://purl.org/dqmvocabulary

158,950 であり, 1 つのプロパティあたり約 39 個の候補が出現していることになる. 今回, プロパティのタイプが owl:DatatypeProperty になるものについては値域を考慮していない. 日本語 Wikipedia オントロジーの owl:DatatypeProperty となるプロパティはデータ型について詳細に定義しておらず, 全てリテラルとなっている. そのため, owl:DatatypeProperty については値域で分類できない. そこで, 定義域が照合されれば候補として抽出するようにした. その結果, 非常に多くの候補が出現してしまっている. 実際に owl:ObjectProperty となるプロパティについてだけ見てみると, 全てのプロパティ候補数は 3,938 であり, 1 つの語彙あたり約 4 個の候補が出現している.

表 4.6 が候補の一例である. 人物に関するプロパティが多い. 人物に関する語彙は定義域が“Person”となっているケースが多く, 日本語 Wikipedia オントロジーの“人物”クラスと対応付けされるため, 候補も多くなってしまっている. “http://schema.org/relatedTo”は, 単語のみでは非常に広い意味を持つが, schema.org 語彙において, relatedTo は人物間の関係性を示すプロパティであり, 定義域・値域ともに“Person”となるため,“関連人物”プロパティのような正しい候補が抽出できている.

タイプとして owl:ObjectProperty を持つ 903 のプロパティについて, 正答なプロパティが候補として出現しているものは 680 個であった. 75.3%のプロパティについて正答なプロパティを定義できることになる. 表 4.7 にいくつかのプロパティでのタイプが owl:ObjectProperty となるプロパティ数, 候補としての出現数, 正答なプロパティ数を示した.

**表 4.6 プロパティの日本語語彙候補の一例**

| プロパティ | 候補 |
|---|---|
| http://schema.org/manufacturer | 製造メーカー<br>所属<br>所属団体 |
| http://schema.org/relatedTo | 関連人物<br>関連<br>配偶者 |
| http://purl.org/dc/elements/1.1/publisher | 製作者<br>原作者<br>スタッフ |
| http://swrc.ontoware.org/ontology#publication | 著書<br>著作<br>共著書 |
| http://purl.org/goodrelations/v1#isSimilarTo | 後継製品<br>関連商品 |

表 4.7 日本語 Wikipedia オントロジークラスとクラス名の対応付けの一例

| 語彙 | プロパティ数 | 候補出現数 | 正答数 |
|---|---|---|---|
| http://purl.org/dc/terms/ | 14 | 1 | - |
| http://xmlns.com/foaf/0.1/ | 34 | 28 | 22 |
| http://www.w3.org/2003/01/geo/wgs84_pos# | 1 | 1 | 1 |
| http://www.geonames.org/ontology# | 16 | 6 | 3 |
| http://purl.org/goodrelations/v1# | 50 | 11 | 8 |
| http://purl.org/ontology/mo/ | 98 | 53 | 23 |
| http://schema.org/ | 113 | 62 | 53 |
| http://www.w3.org/2004/02/skos/core# | 6 | 1 | 1 |
| http://purl.uniprot.org/core/ | 51 | 8 | 2 |

表 4.7 を見ると，汎用的語彙である schema.org や人物に関する語彙である foaf，音楽に関する語彙である MusicOntology [50]などは多くのプロパティを候補として抽出している．これは，日本語 Wikipedia オントロジーが日本語版 Wikipedia から構築したオントロジーであり，Wikipedia は音楽や人物情報が多く含んでいることが理由と考えられる．しかし，汎用的な部分については Wikipedia もプロパティを多く含んでいるため，schema.org の半分以上のプロパティと対応付けできており，正答率も高い．

しかしながら，Uniprot Core Ontology [51]のような専門性が高くなる語彙については，日本語 Wikipedia オントロジーにクラスやプロパティが存在しておらず，候補としての抽出数，正答率ともに低い結果となっており，このような専門性の高い語彙をいかに日本語化するかが今後の課題と言える．

地理情報の Geo Names [46]や WGS84 Geo Positioning [52] はタイプが owl:DatatypeProperty となるものが多いため，プロパティ数が少なくなっている．また，DCMI MetadataTerms [53]についてはほとんどのプロパティが値域を定義しているが，定義域を定義していないため，候補として抽出できなかった．

**(3) schema.org 語彙との比較評価**

schema.org 語彙の上位カテゴリである CreativeWork，Event，Intangible，Organization，Person，Place，Product の 7 つの領域について今回構築した日本語語彙と比較評価を行った．表 4.8 に，schema.org 語彙の各領域におけるプロパティと構築した日本語語彙プロパティの比較例を示した．構築は 4.5.1 項(2)で抽出した候補の中から手作業で選択した．

“CreativeWork”，“Organization”，“Person”の領域は半数以上が日本語語彙化できている．schema.org 語彙の中には，あまり利用頻度が高くないと思われるプロパティも多く存在しており，各領域において比較的一般的であろうと思われるプロパティについてはほとんどが日本語語彙化できたと考えている．

表 4.8 schema.org 語彙の各領域と構築した日本語語彙の比較例

| 領域 | schema.org | 日本語語彙 |
|---|---|---|
| CreativeWork | author | 著者 |
| | award | 賞 |
| | creator | 製作者 |
| | など50 プロパティ | など29 プロパティ |
| Event | startDate | 開始日 |
| | location | 場所 |
| | など12 プロパティ | など4 プロパティ |
| Intangible | 0 プロパティ | 0 プロパティ |
| Organization | location | 所在地 |
| | founder | 設立者 |
| | foundingDate | 設立年月日 |
| | など33 プロパティ | など16 プロパティ |
| Person | affiliation | 所属 |
| | children | 子供 |
| | nationality | 国籍 |
| | など48 プロパティ | など32 プロパティ |
| Place | address | 郵便番号 |
| | openingHoursSpecification | 開園時間 |
| | など19 プロパティ | など4 プロパティ |
| Product | manufacturer | 製造メーカー |
| | isRelatedTo | 関連製品 |
| | releaseDate | 発売日 |
| | など19 プロパティ | など4 プロパティ |

日本語語彙化できなかったプロパティについては，日本語 Wikipedia オントロジーに似たようなプロパティが存在していない，プロパティは存在するが定義域や値域を定義していないという理由があるため，このようなプロパティについては今後，日本語 Wikipedia オントロジー内のプロパティを手作業，もしく新たな手法の追加による自動構築により，プロパティ数を増やすことで対応する必要がある．

### 4.5.2 DBpedia との比較評価

日本語 Wikipedia オントロジー内のプロパティと Linked Open Data において代表的なハブとして利用されている DBpedia との比較評価を行った. DBpedia は多言語 Wikipedia 記事を対象に RDF データベースを構築しているが，本実験における比較対象の DBpedia のデータは日本語版 Wikipedia 記事からの Infobox プロパティのデータセットを利用した. 表 4.9 に DBpedia と日本語 Wikipedia オントロジーの比較結果を示した.

本手法で抽出したトリプル数は DBpedia に比べ約 200 万多い事が分かる．しかし，プロパティ数に関しては DBpedia より 700 程度多いだけである. この理由として, DBpedia は“wikiPageUsesTemplate”プロパティのような独自のプロパティ名を持っていることに加え，DBpedia が Infobox の wiki 記述から直接トリプルを抽出している事が言える. このため，多くのプロパティ名は英語表記や省略された形で抽出されており，直接的にプロパティ名からでは意味を理解できないプロパティを多く含んでいる．日本語 Wikipedia オントロジーでは Infobox からのプロパティ名抽出の際にモデリングを行い，wiki 記述を実際の表記の形に変換する処理を行っているため，日本語のプロパティ数は多くなっている．しかし，この処理により変換できず，抽出できなかったトリプルも多く含んでおり，Infobox からの抽出法からのトリプルのみを比較すると，日本語 Wikipedia オントロジーは 1,962,411 と，DBpedia に比べ 100 万も少なくなっている．実際に，全プロパティ名から半角英数を取り除いた日本語表記のみで構成されているプロパティ数は日本語 Wikipedia オントロジーが 8,447 に対し，DBpedia は 5,056 程度となっている．さらに，主語となるインスタンス数も日本語 Wikipedia オントロジーは DBpedia に比べ 2.4 倍であり，より多くの記事名をインスタンス化し，プロパティを付加できている.

さらに，定性的な評価として，いくつかのインスタンスにおける関係の比較を行った. 図 4.11 にクラス階層の比較，表 4.10 に同義語の比較，表 4.11 にプロパティ比較の結果の一例を示す. また, 表 4.11 中のプロパティ名の後に“*”が付いているものは ObjectProperty, “+”は DatatypeProperty を表している.

**表 4.9 日本語 Wikipedia オントロジーと DBpedia の比較結果**

| 関係の種類 | 日本語Wikipediaオントロジー | DBpedia |
|---|---|---|
| プロパティ数 | 10,769 | 10,034 |
| トリプル数 | 4,867,882 | 2,840,553 |
| プロパティ定義域(rdfs:domain) | 9,486 | - |
| プロパティ値域(rdfs:range) | 5,120 | - |
| 主語となるインスタンス数 | 319,742 | 133,999 |

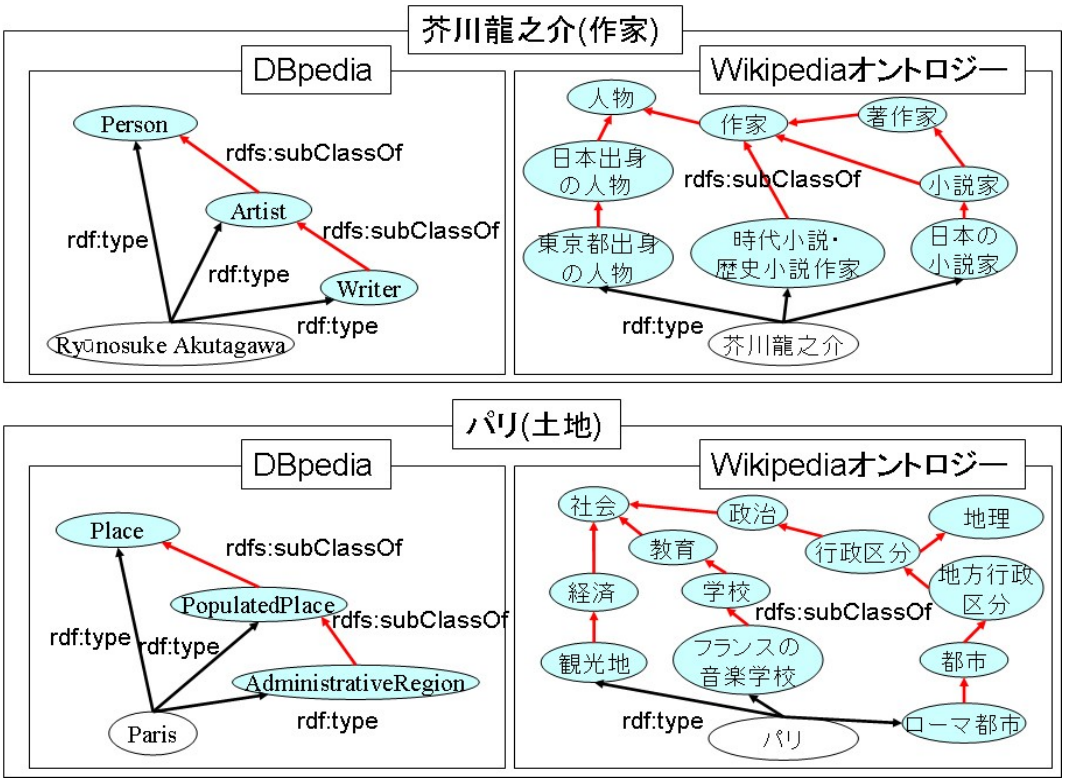



**図 4.11 日本語 Wikipedia オントロジーと DBpedia のクラス階層比較例**

**表 4.10 日本語 Wikipedia オントロジーと DBpedia の同義語比較例**

| 概念 | DBpedia | 日本語Wikipediaオントロジー |
|---|---|---|
| 芥川龍之介<br>(作家) | Ryunosuke Akutagawa<br>Chokodo Shujin<br>Kappa (short story) | 芥川竜之介<br>河童忌 |
| パリ<br>(土地) | City of Paris<br>Parisian (person)<br>Paris (France) | パリ県<br>Paris<br>巴里 |

図 4.11 により，DBpedia はどちらのインスタンスも全てのクラスをタイプとして持っており，オントロジーのクラスーインスタンス関係として冗長であると言える．また，日本語 Wikipedia オントロジーの方が中間概念や下位概念がより詳細に定義されているが，誤った関係も見られる．

表 4.11 日本語 Wikipedia オントロジーと DBpedia のプロパティ比較例

| 概念 | DBpedia | | 日本語 Wikipedia オントロジー | |
|---|---|---|---|---|
| | 項目 | 値の例 | 項目 | 値の例 |
| 人物<br>芥川龍之介<br>(作家) | Genre* | 短編小説 | ジャンル* | 短編小説 |
| | notable works | 「羅生門」(1915 年) など | 代表作* | 羅生門など |
| | birth place* | 日本, 東京 | 国籍* | 日本 |
| | children | 芥川比呂志（長男）など | 子供* | 芥川比呂志など |
| | relations | 芥川麻実子（孫）など | 親族* | 芥川麻実子など |
| | death date+ | 1927-07-24 | 死没+ | 1927 年 7 月 24 日 |
| | birth date+ | 1892-03-01 | 誕生+ | 1892 年 3 月 1 日 |
| | その他<br>のプロパティ | wikiPageUsesTemplate,<br>imagesize など<br>6 プロパティ 6 トリプル | その他<br>のプロパティ | 著作, 家族など<br>7プロパティ 63 トリプル |
| 無生物<br>パリ<br>(土地) | sans | 都市圏：11,840,000 | 人口+ | 11,840,000 人 |
| | km2 | 都市圏：14,518 | 面積+ | 14,518km2（10,540ha） |
| | alt maxi | 130m | 標高 | 最高:130m |
| | alt mini | 28m | 標高 | 最低:28m |
| | maire | ベルトラン・ドラノエ | 市長 | ベルトラン・ドラノエ |
| | cp | 75001 - 75020、75116 | 郵便番号 | 75001 - 75020、75116 |
| | その他<br>のプロパティ | xPrecipMm,xSun,region など<br>22 プロパティ 69 トリプル | その他<br>のプロパティ | 国,姉妹都市・提携都市<br>スポーツ, 自治体間連合<br>17 トリプル |

DBpedia は明確な同義語を定義していないが，Wikipedia のリダイレクトリンクのプロパティが存在する．表 4.10 により，DBpedia は Wikipedia のリダイレクトリンクをそのまま利用しているため，誤った関係も多い．日本語 Wikipedia オントロジーも Wikipedia のリダイレクトリンクを利用しているが，抽出の際にオントロジー内のクラスやインスタンスと照合処理を行っているため，誤った関係が少なくなっている．このため，関係数として芥川龍之介の場合, DBpedia が 9 つのリダイレクトがあるのに対し, 日本語 Wikipedia オントロジーは 2 つしかない．さらにパリについては，DBpedia が 30 に対し，日本語 Wikipedia オントロジーが 3 つとなっている．

表 4.11 より，プロパティ名について，DBpedia は言語対応がなされておらず，一見して何を意味するのかがわかりにくくなっている．各トリプルとプロパティ数に関して，DBpedia は独自のプロパティを多く含んでおり，特に土地に関しては yearSun など詳細な DatatypeProperty が存在しているが，代わりに日本語 Wikipedia オントロジーでは人物の著作や家族，土地の姉妹都市やスポーツなどのプロパティが存在している．プロパテ

ィタイプごとに見ていくと，DatatypeProperty について，人物に関して誕生日や没日といった DatatypeProperty は DBpedia も日本語 Wikipedia オントロジーも同じである．しかし，土地に関しては先に述べたような詳細な DatatypeProperty が存在している．また，日本語 Wikipedia オントロジーではモデリングが不十分であったため“標高”プロパティとして最高，最低の値が見られるが，DBpedia ではそれぞれが別のプロパティ名として定義されている. ObjectProperty については日本語 Wikipedia オントロジーの方が詳細であり，特に，人物に関しては“子供”，“配偶者”，“代表作”といったプロパティが ObjectProperty として定義されており，値もリテラルではなくインスタンスとして関係付けられている．さらに，日本語 Wikipedia オントロジーには“家族”プロパティが存在し，この“家族”プロパティと“子供”や“配偶者”プロパティの間には上位下位の関係が作られている事も特徴である．

DBpedia に比べ，日本語 Wikipedia オントロジーは同義語・クラス階層についてはより詳細であり，さらにプロパティについて，DatatypeProperty は部分的に少ない所があるものの，ObjectProperty は非常に豊富な関係を定義している．DBpedia が海外において，Linked Open Data のハブとして利用されていることをふまえると，日本語 Wikipedia オントロジーは Linked Open Data のハブとして十分に利用できるといえる．

### 4.5.3 日本語 Wikipedia オントロジーLinked Open Data を利用したアプリケーション

本項では日本語 Wikipedia オントロジーLOD を利用した一例として実装した，日本語 Wikipedia オントロジーと Linked Open Data を連携させて検索することができるツール WiLD（Wikipedia Linked Data Application）を紹介する．実装は主に日本語 Wikipedia オントロジーの検索 API モジュールと検索インタフェースモジュールに分かれており，それぞれ Java Servlet，Adobe Flash Builder をベースとして実装している．WiLD は実際に以下の URL で使用することができる．

WiLD (Wikipedia Linked Data Application): http://wild.wikipediaontology.org

#### (1) WiLD の機能

WiLD は日本語 Wikipedia オントロジー内のデータを参照し，検索するだけでなく，外部の Linked Open Data を参照し，ユーザインタフェースにより，ユーザに閲覧しやすい形式で表示する．図 4.12 が WiLD のシステムアーキテクチャである．

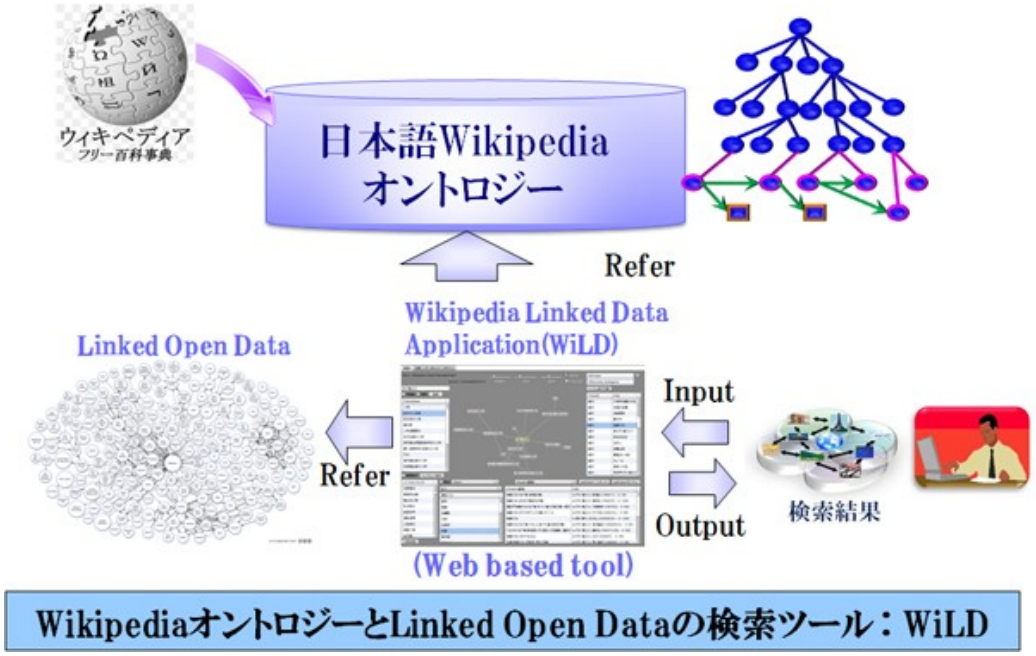



図 4.12 WiLD のシステムアーキテクチャ

WiLD は，ユーザインタフェースを通して，ユーザから入力されたキーワードを参照モジュールに渡す．参照モジュールは日本語 Wikipedia オントロジーの関連する概念を取得し，取得した概念をユーザインタフェースが再びユーザが閲覧しやすい形式にデータを編集して表示する．この際に，取得した概念と事前に登録してある Linked Open Data が関連付け可能な場合はユーザに外部 Linked Open Data の参照が可能であることを示し，ユーザは Linked Open Data 連結モジュールを通してより詳細な情報を得ることができる．以下が WiLD の主な特徴である．

- **日本語 Wikipedia オントロジーの検索機能**
  - グラフ表示機能
    - インスタンスとその所属クラスをノードとし，それを結び付けるグラフ
    - インスタンス，プロパティ，プロパティの値を結び付けるグラフ
  - クラス表示機能
    - ルートクラスから検索した概念が所属するクラスへのツリー表示
    - インスタンスの所属クラス一覧の表示
  - トリプルデータ表示機能
    - インスタンスを主語とするトリプルデータの表示
  - インスタンス一覧表示機能
    - 選択したクラスのインスタンス一覧の表示
  - プロパティ表示機能
    - 選択したクラスを定義域とするプロパティ表示

- プロパティの値域やタイプによる絞込み

- **日本語 Wikipedia オントロジーのインスタンス間の比較機能**
  - インスタンス間のプロパティ，クラスの比較表示
  - 比較結果のグラフ表示

- **Linked Open Data の連結機能**
  - 外部 Linked Open Data の登録
  - 日本語 Wikipedia オントロジーの概念と Linked Open Data の連結による検索
  - 日本語 Wikipedia オントロジーの概念と Linked Open Data の連結による比較

図 4.13 が WiLD のユーザインタフェースである．

### (2) WiLD でのデータ検索

書籍販売を行っている大規模な書籍販売サイトである Amazon を例に実際にデータの検索を行う．Amazon では，書籍の検索方法としてキーワード検索の他，「社会・政治･法律」「ノンフィクション」「歴史・地理」といったカテゴリ検索をカバーしている．しかしながら，これ以外には提供されておらず，ユーザは Amazon がサポートした検索軸を利用する他ない．ここで WiLD を利用するとどのような検索ができるのかを検証する．

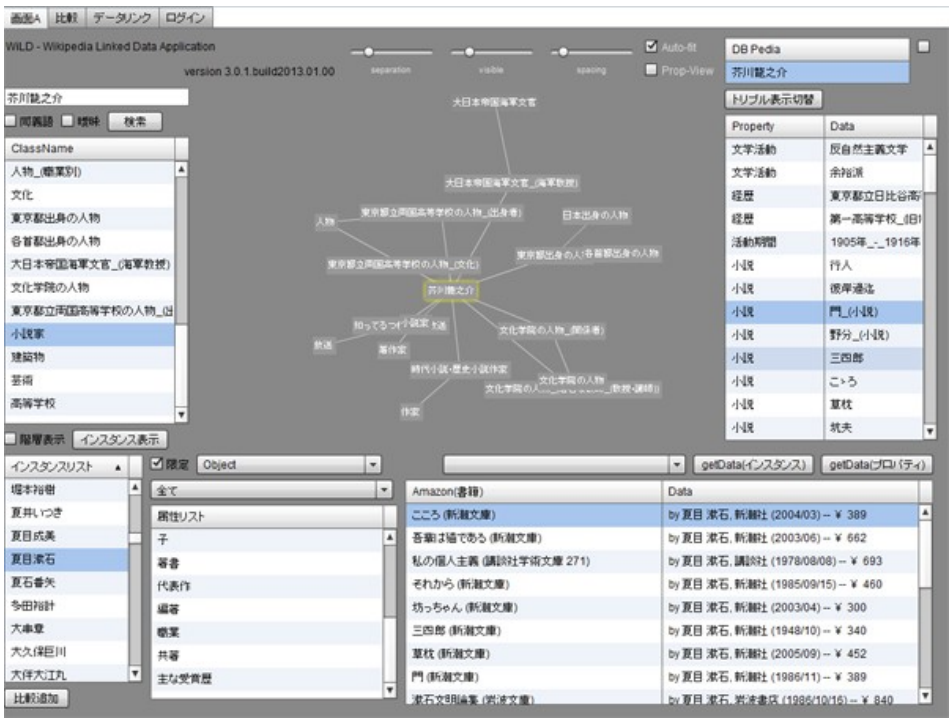



図 4.13 WiLD のユーザインタフェース

検索は以下の流れで行われる．それぞれの手順は図 4.14～図 4.19 に対応している．

① 検索キーワードボックスにキーワードを入力する. 例では作家である「芥川龍之介」を入力し，検索を実行している．(図 4.14)

② ①により，日本語 Wikipedia オントロジーから芥川龍之介に関係する情報が出力される．ここでは，芥川龍之介というインスタンスが属する「著作家」「小説家」「時代小説・歴史小説作家」といったクラスがクラスリストに表示される．さらに画面右側のトリプルリストに芥川龍之介を主語とするトリプル一覧が表示される．(図 4.15)

③ クラスリストから,「時代小説・歴史小説作家」と「小説家」クラスを選択し，インスタンス表示を行うと,「時代小説・歴史小説作家」かつ「小説家」である作家名がインスタンスリストに表示される．(図 4.16)

④ ここで,「著作」プロパティー蜘蛛の糸というトリプルを選択する．Amazon API が外部 Linked Open Data として既に「著作」プロパティと関連付けられているため，ドロップダウンリストに Amazon API が表示される．(図 4.17)

⑤ Amazon API を選択し，getData(プロパティ)ボタンを押す事で,「蜘蛛の糸」に関する書籍を Amazon から検索し，結果を出力する．(図 4.18)

⑥ さらに，インスタンスリストから「森鴎外」を選択し，getData(インスタンス)ボタンを押す事で,「森鴎外」に関する書籍を Amazon から検索し, 結果を出力する．(図 4.19)

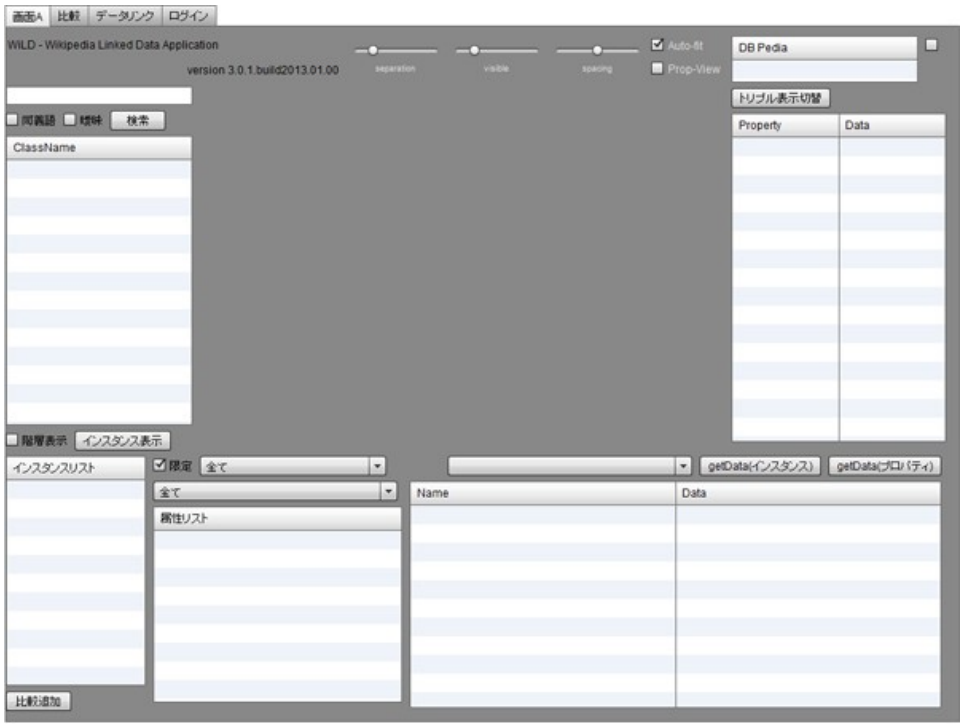



図 4.14 WiLD の検索インタフェース画面

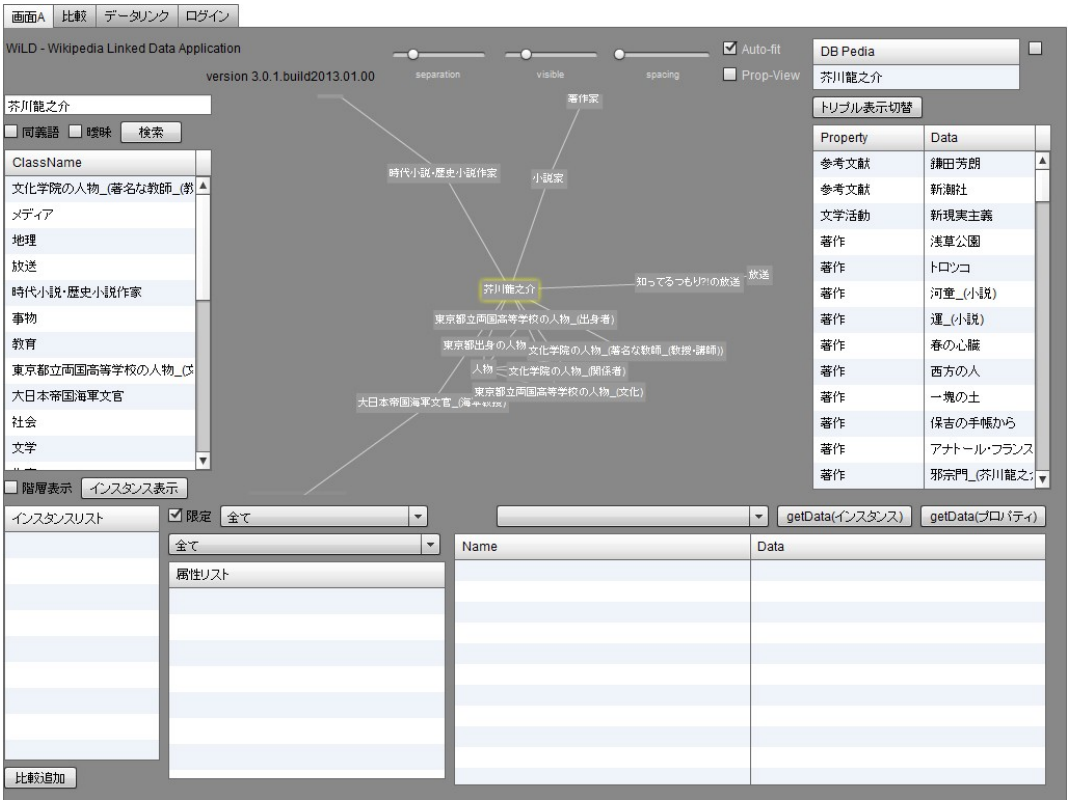



図 4.15 検索結果画面 1

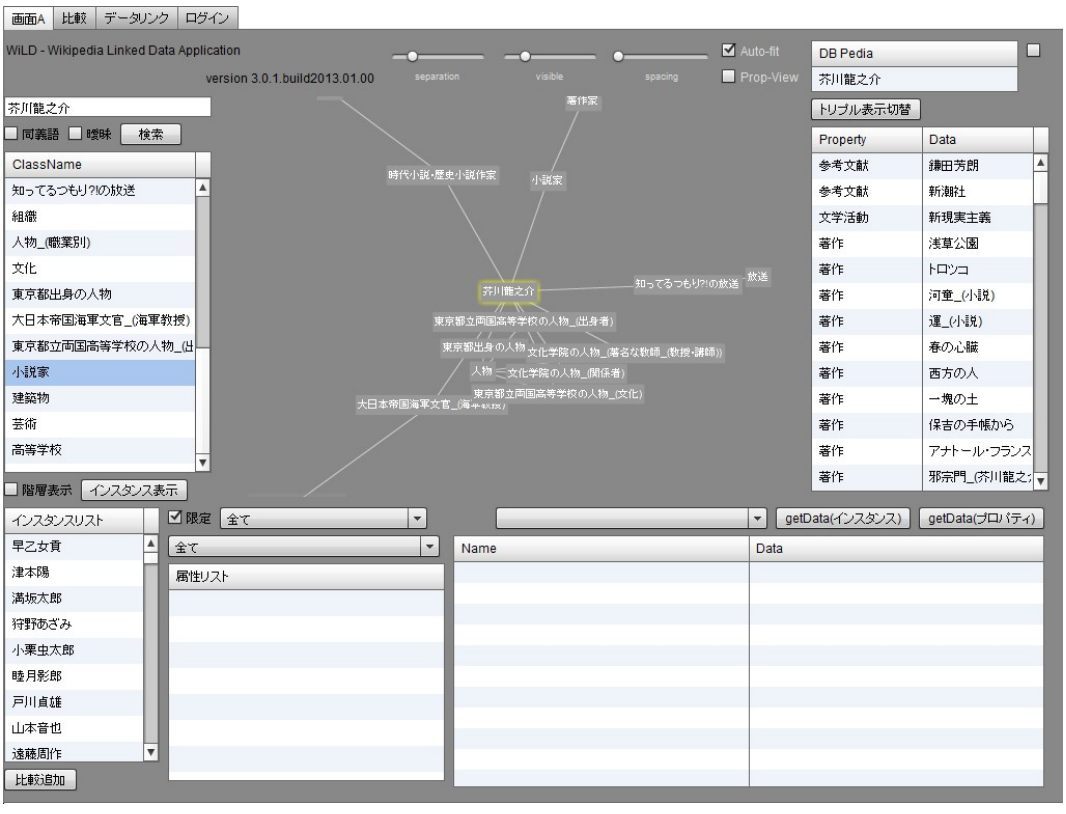



図 4.16 検索結果画面 2

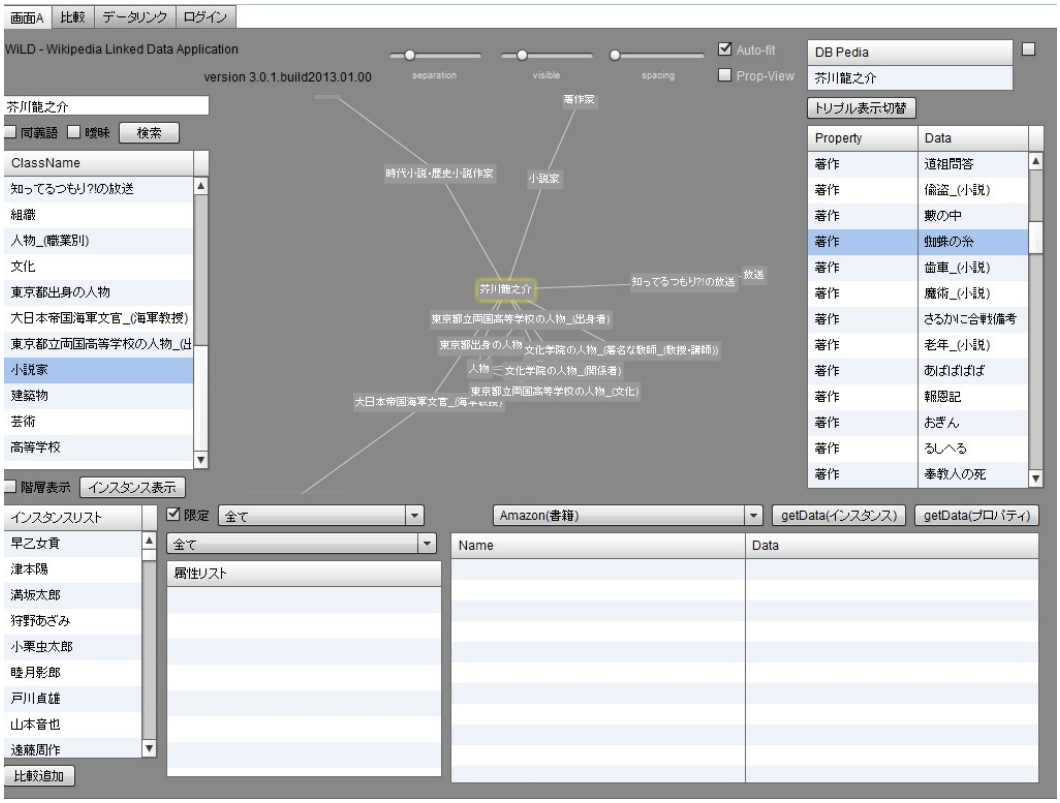



図 4.17 検索結果画面 3

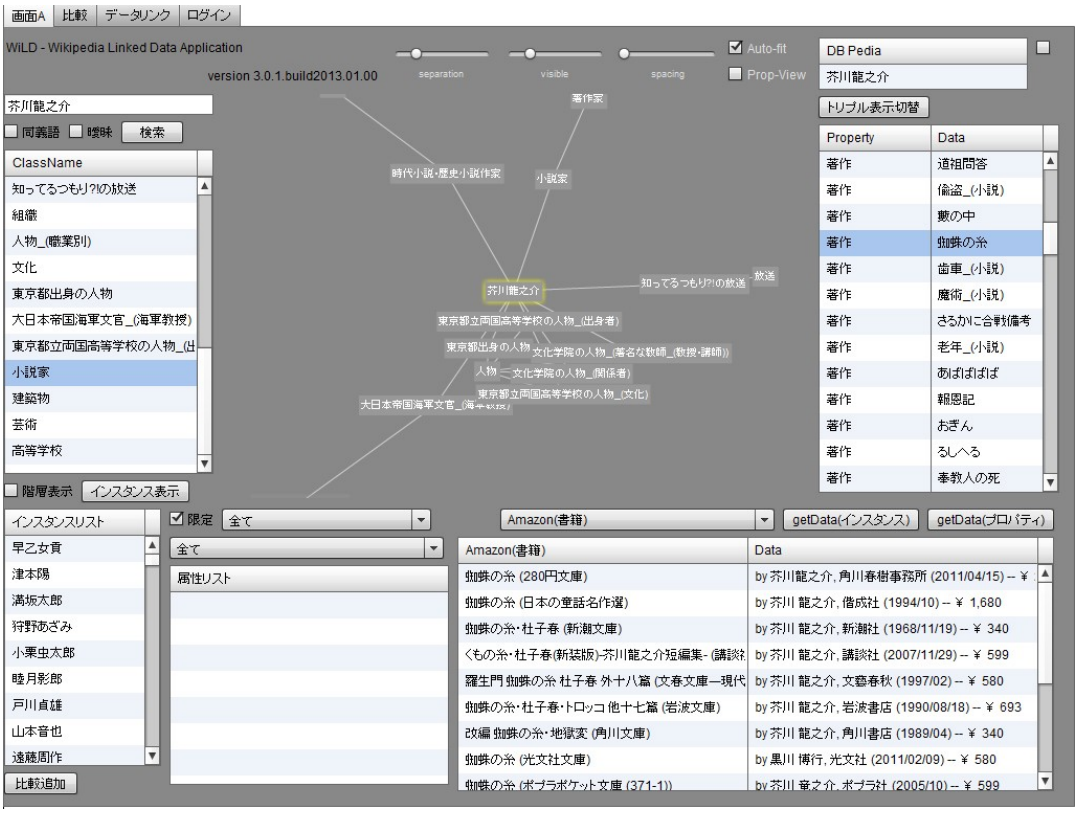



図 4.18 検索結果画面 4

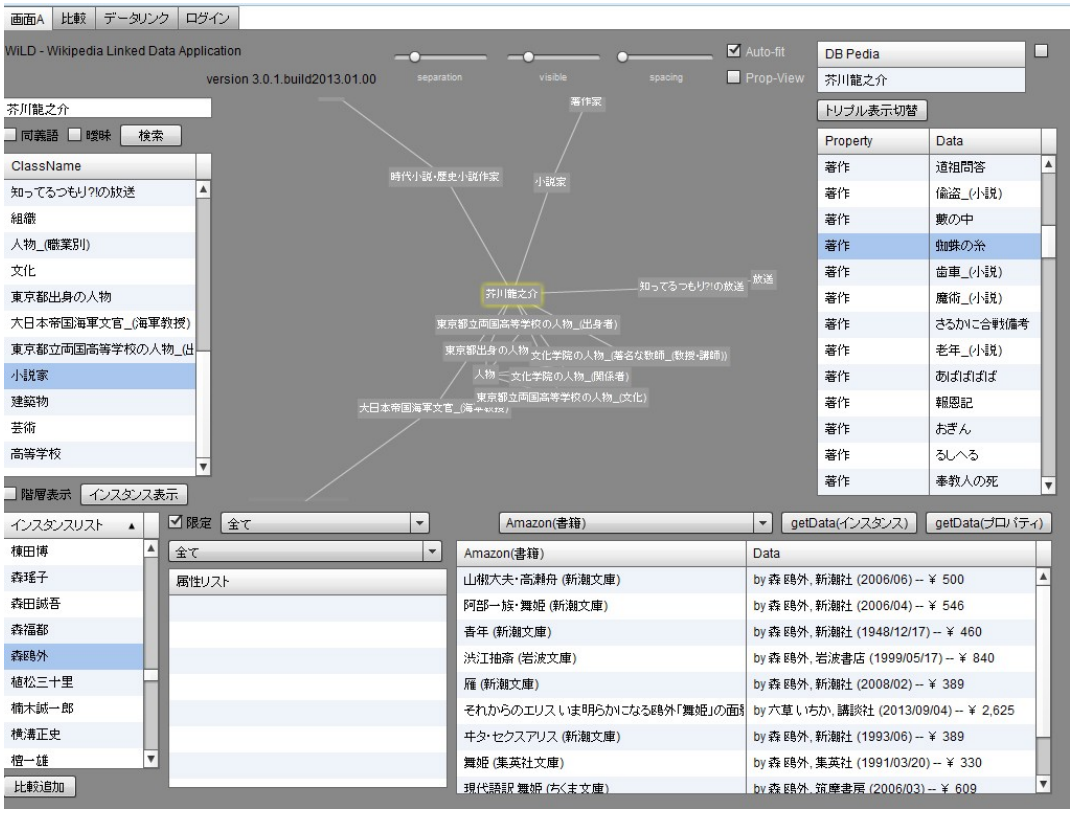



**図 4.19 検索結果画面 5**

今回の例では，「芥川龍之介」という入力から，「時代小説・歴史小説作家かつ小説家」という共通点を持つ別の作家の「森鴎外」を連想し，それらの作家の書籍を検索することが可能となっていることがわかる．作家の持つ属性に応じて関連する他の作家を特定するという検索が可能になり，Amazon の従来の検索機能を拡張した検索ができるようになっている．他にも生まれた年代が近い作家，出身地別の作家，同じ賞を取得した作家など，様々な関連を持った作家を検索することも可能である．また，プロパティの値から連想的にデータをたどる事もでき，クラスによる階層関係だけではなく非階層関係による連想を支援する．

書籍検索以外の例として，レストラン検索にも応用することができる．日本語 Wikipedia オントロジーとグルメ情報に関する Linked Open Data を連携させることによって，以下のような検索を行うことができる．

・日本語 Wikipedia オントロジー側
入力概念：郷土料理
→「郷土料理」のインスタンスを出力
→インスタンスの中から「ちゃんぽん」を選択

・グルメ系 Linked Open Data 側
レストランのジャンルプロパティの値に「ちゃんぽん」が含まれるのレストランを検索
→ちゃんぽんが食べられる料理店の一覧を出力

以上の検索プロセスにより，漠然とした「郷土料理」というキーワードから，「ちゃんぽん」というより詳細な料理を出力し，その料理を提供しているレストランを検索することができる．

さらに，値域やプロパティの値から検索を行う例を示す．

・日本語 Wikipedia オントロジー側
入力概念：松阪市
→「日本の地方公共団体」のインスタンスとプロパティを出力
→インスタンスと「名産品」プロパティを選択

・グルメ系 Linked Open Data 側
キーワードとして名産品の値(例えば「松坂牛」)を持つレストランを検索
→名産品のレストラン一覧を出力

以上のように，日本語 Wikipedia オントロジーLOD と外部 LOD を連結することで，連想検索支援に利用できる．テキストベースの知識では知り得ない新たな気付きを提供することができるという点で，ビジネスや教育といった分野において日本語 Wikipedia オントロジーは有用であると考えられる．

### (3) WiLD でのデータ分析

日本語 Wikipedia オントロジーは，トリプルにより，多様なインスタンスの情報が蓄積されている．これらは単一のインスタンスの情報として利用できるだけでなく他の情報とも容易に連結できる．以下の例では，トリプルから抽出したデータを属性ごとに連結し，比較分析を行っている．

- **定性的データの比較分析**

① キーワードとして，「織田信長」を入力する．織田信長が属するクラスがクラスリストに出力されるので，「軍事指揮官」かつ「日本の神_(日本人)」のインスタンスを出力する．さらに，出力したインスタンスリストから，「豊臣秀吉」を選択し，比較追加ボタンを押す．同様に「徳川家康」を選択し，比較追加ボタンを押す．(図 4.20)

② 左上の比較タブを選択し，選択ビューに移行する．このビューでは先ほど選択した，インスタンスがリストに追加されている．比較ボタンを押すとクラス及びイ

ンスタンスンお比較結果が表示される．(図 4.21)

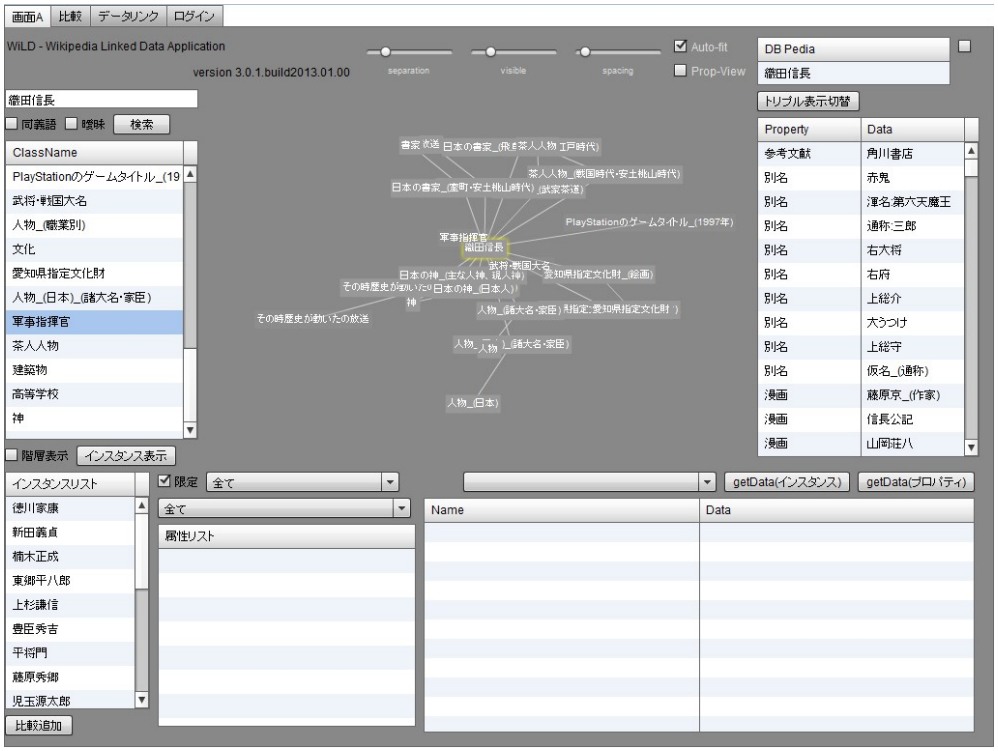



図 4.20 検索結果画面 6

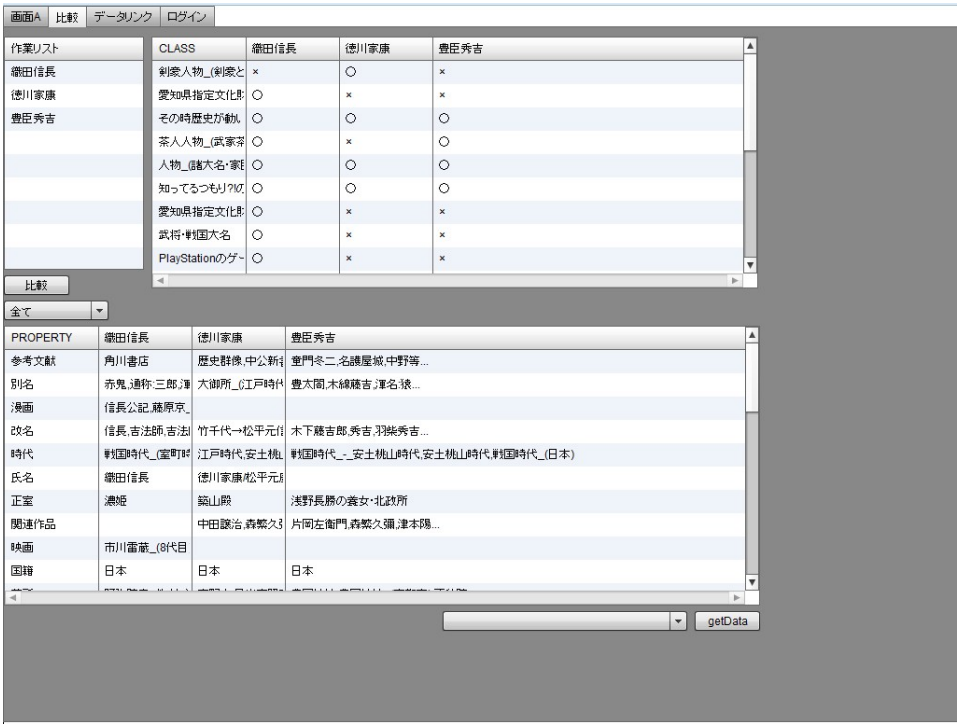



図 4.21 検索結果画面 7

- **定量的データの比較分析**

① 入力キーワードとして，「利根川」が与えられたとする．先ほどと同様の操作で，利根川が属するクラスがクラスリストに出力されるので，「日本の川」のインスタンスを出力する．さらに，出力したインスタンスリストから，「信濃川」を選択し，比較追加ボタンを押す．同様に「石狩川」を選択し，比較追加ボタンを押す．この時，属性リストには日本の一級河川を定義域に持つプロパティが表示され，値域やプロパティタイプによって絞り込むこともできる．(図 4.22)

② 左上の比較タブを選択し，選択ビューに移行する．比較ボタンを押すとクラス及びインスタンスの比較結果が表示される．DatatypeProperty を選択し，その中から「延長」プロパティをダブルクリックする．定量的なデータがグラフとして表示される．(図 4.23)

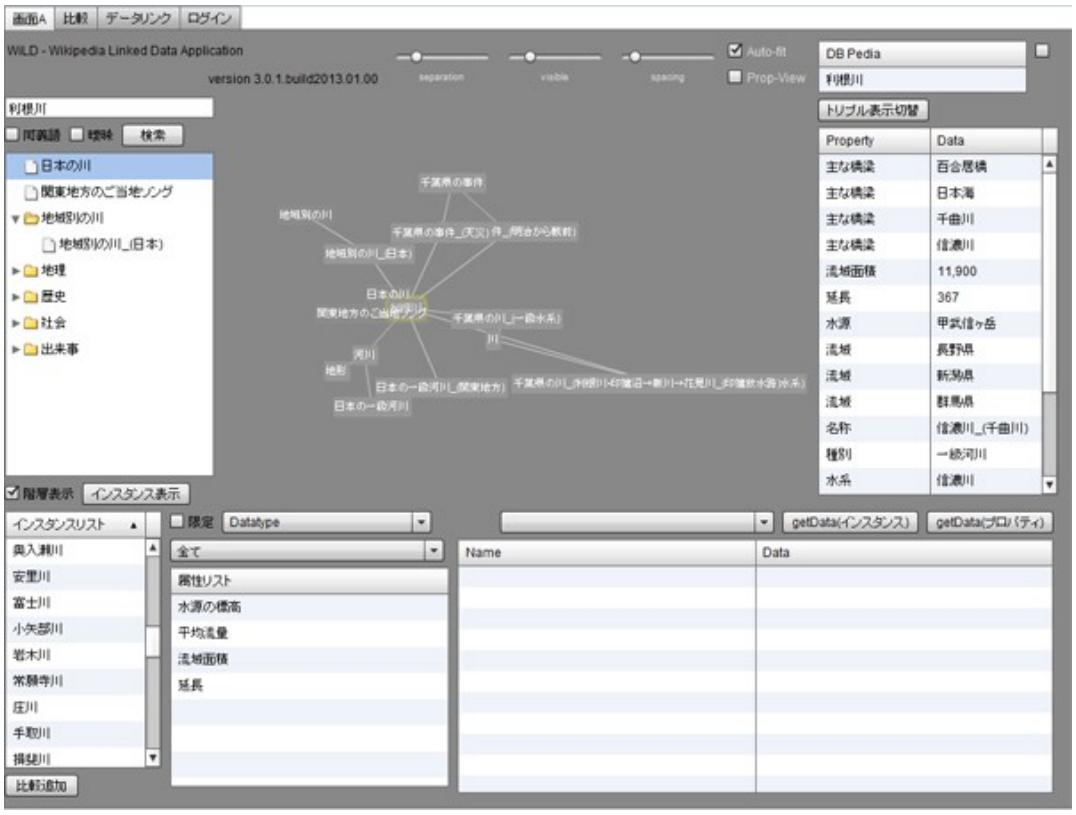



図 4.22 検索結果画面 8

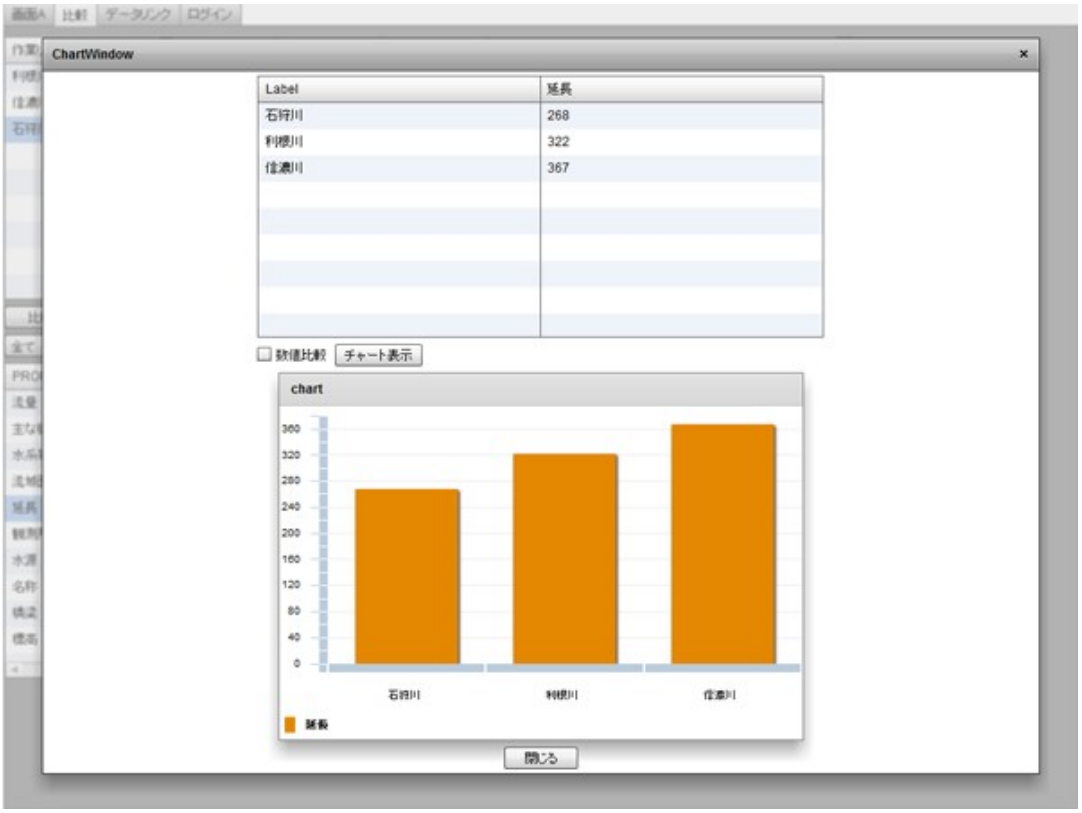



**図 4.23 検索結果画面 9**

今回の例では，あるクラスによってカテゴリ分けされたインスタンスを比較することが可能となっていることがわかる. 単純に個別検索によって収集したデータの比較ではなく，rdf:type と rdfs:subClassOf を利用することによって，必要に応じた同種のインスタンスに絞り込んだうえで比較分析することができている．rdfs:domain を参照することによって，どのクラスのインスタンスがどのような属性を持つかということを特定できることも重要な点である.

図 4.24 に一般的な概念間の比較に関するプロセスと WiLD を利用した比較に関するプロセスの一例を示す.

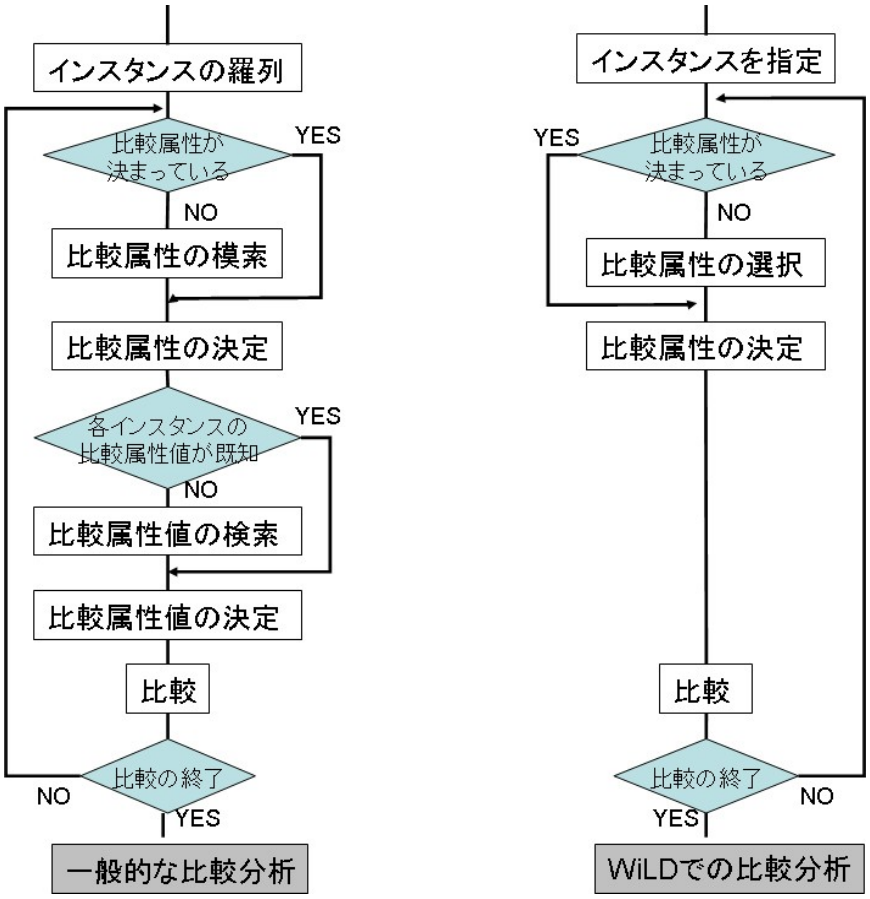



**図 4.24 一般的な比較分析プロセスと WiLD における比較分析プロセスの一例**

図 4.24 の例では比較対象のインスタンスの一部もくしは全部が既知である条件での比較を行っている．先の定性的データ分析のように「織田信長」「豊臣秀吉」「徳川家康」というインスタンス全てが既知である状況は必ずとは言えない．現実では，調査をしながら検索対象を決定する場合も考えられが，この場合も一般的にはインスタンスを網羅する所からはじまるのに対し，WiLD では単一のインスタンスを指定し，クラスを指定する事である程度絞り込んだ条件での比較が可能となる．ここから比較プロセスが始まる．まず，比較する属性を決定しなければならない．一般的には，「生まれた年」「生存していた時代」「家臣の数」などには，各インスタンスを検索しつつ比較できそうな属性に絞込み，決定する．さらに，決定した属性から属性値を決定するが，この際にも属性値が既知ではない場合検索しなければならない．WiLD では先に指定したクラスによって，属性は絞り込まれており，日本語 Wikipedia オントロジー内での情報に限るならば，属性を指定した段階で属性値が決定し，自動的に比較結果を表示する．また，WiLD では属性値から連想的に概念をたどって行くことが容易であり，この連想を通じて比較分析者は更に分析を展開していく事が可能になると考える．

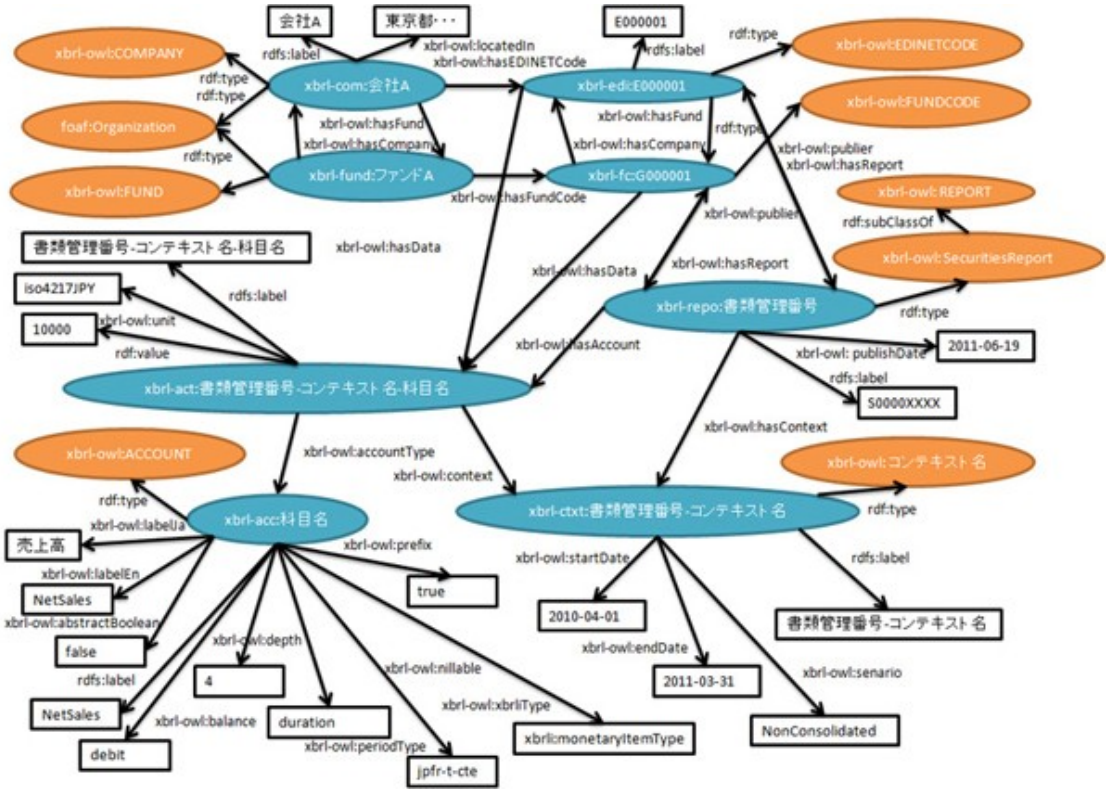



図 4.25 XBRL Linked Open Data のモデル

### (4) 企業会計情報 Linked Open Data との連携による分析

WiLD では，外部 Linked Open Data と連携することで分析支援を行うことができる．例として，XBRL Linked Open Data [54]と連携することで，企業間の会計情報の分析を行う. XBRL Linked Open Data は, 金融庁 EDINET から取得した企業の会計情報を基に，自動的に RDF グラフを構築し，Linked Open Data として公開したものである．図 4.25 に XBRL Linked Open Data のモデルを示す.

会計情報は組織名や売上高，営業利益といった科目情報があるが，日本語 Wikipedia オントロジーも組織の情報といくつかの科目プロパティを持っている．そのため，連携が可能である．XBRL Linked Open Data は，以下の URL から利用することができる.

XBRL Linked Open Data: http://xbrl-lod.org

① 入力キーワードとして，「トヨタ自動車」が与えられたとする．定量的データの分析と同様の操作で, トヨタ自動車が属するクラスがクラスリストに出力されるので，「自動車メーカー」のインスタンスを出力する．さらに，出力したインスタンスリストから，自動車メーカーを選択し，比較追加ボタンを押す．(図 4.26)

② 左上の比較タブを選択し，選択ビューに移行する．比較ボタンを押すとクラス及びインスタンスの比較結果が表示される．DatatypeProperty を選択し，その中から「売上高」プロパティを選択する. 既に売上高プロパティと XBRL Linked Open

Data の売上高プロパティを対応付けているため，XBRL Linked Open Data から詳細な数値データを取得し，グラフとして表示される．(図 4.27)

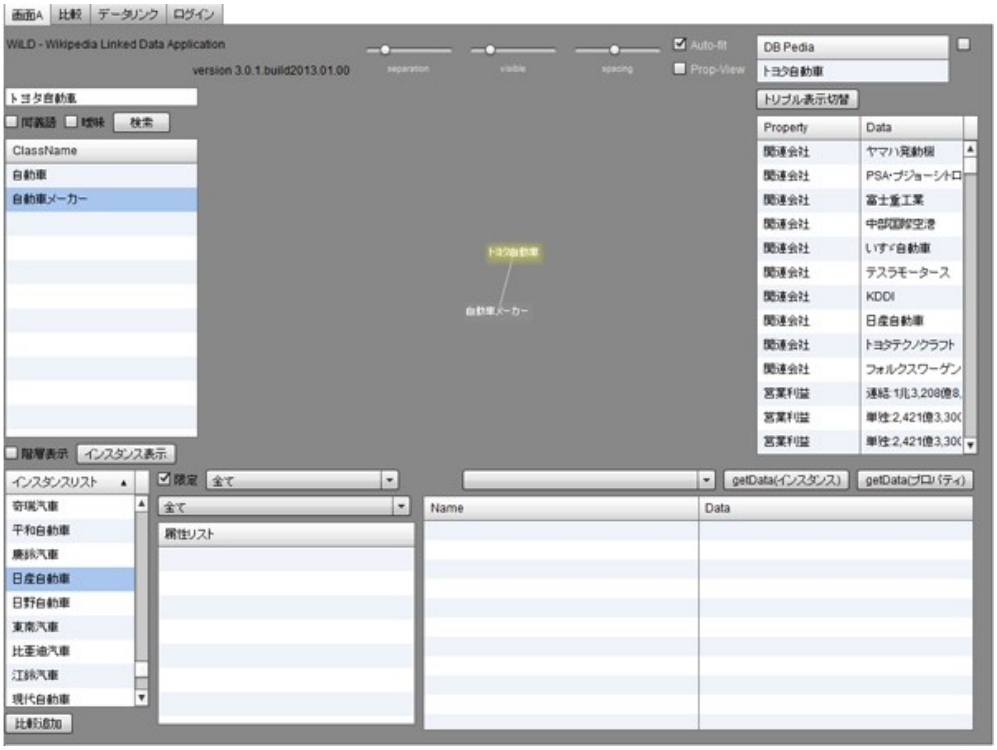



図 4.26 検索結果画面 10

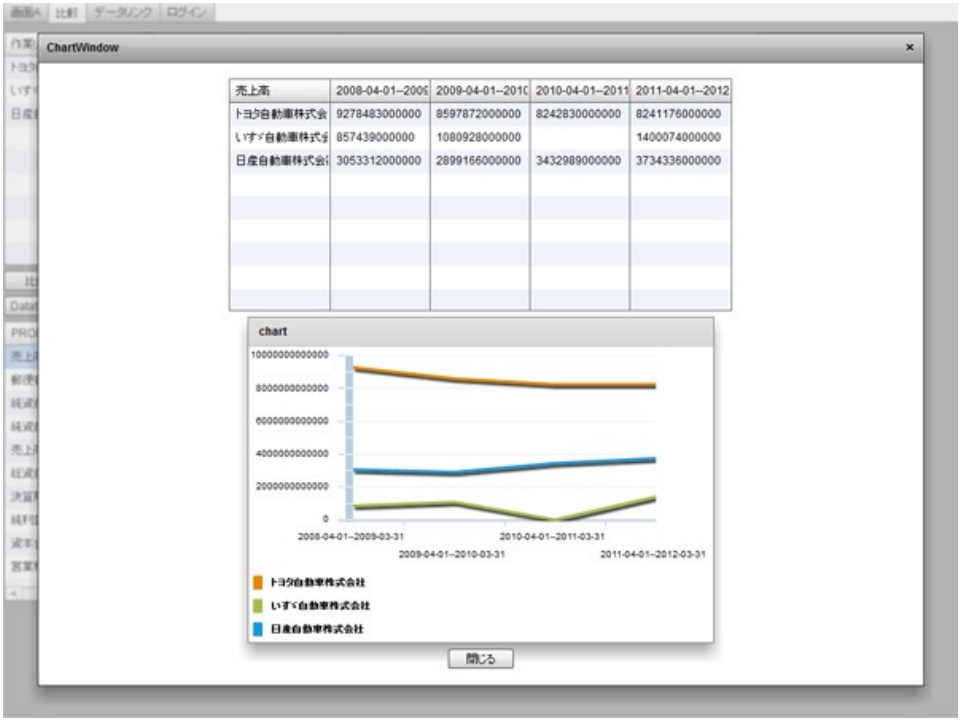



図 4.27 検索結果画面 11

## 4.6 まとめ

本章では大規模で汎用的なオントロジーである日本語 Wikipedia オントロジーを, 領域オントロジー構築支援としての評価および LOD ハブとしての評価という 2 つの視点から有用性の評価を行った.

日本語 Wikipedia オントロジーのいくつかの領域において，クラス，インスタンス，プロパティおよびそれらの関係を評価した結果，人物や土地などの具体物がインスタンスとなる領域では日本語 Wikipedia オントロジーは領域オントロジー支援として有用である. しかしながら，時間や研究分野などの抽象的な領域ではトリプルを生成することができていないため，領域オントロジー支援として利用することは難しいことがわかった.

また，日本語 Wikipedia オントロジーを LOD 化し，DBpedia との比較および日本語 Wikipedia Linked Open Data を利用したアプリケーションの評価を行った結果，日本語 Wikipedia オントロジーは国内 LOD のハブとして有用であることがわかった．さらに，Linked Open Vocabularies に存在する各語彙が持つプロパティの対応付けによる日本語語彙の半自動構築手法により，日本語語彙として，日本語 Wikipedia オントロジーのプロパティを用いることで，LOD 構築のための支援としての利用可能性を示すことができた.

今後の課題として，領域オントロジーとしての利用が難しい抽象的な領域において，日本語 Wikipedia オントロジーの概念を増やす必要があり，Wikipedia において，抽象的な記事はフリーテキストにより書かれている場合が多く，自然言語処理技術等を利用して抽出する必要がある．また，LOD を中心とした日本語 Wikipedia オントロジーの更なる利用法の検討も今後の課題である.

# 第5章　結論

本章では，本論文のまとめと今後の課題および展望について述べる．

本論文では，半構造化情報資源として，日本語版 Wikipedia を用いて，概念および概念間の関係を抽出する事で，大規模で汎用的なオントロジー(日本語 Wikipedia オントロジー)の自動構築手法の提案とその評価を行った．また，日本語 Wikipedia オントロジーの領域オントロジー構築支援としての評価と Linked Open Data のハブとしての評価により，構築した日本語 Wikipedia オントロジーの有用性を示した．

日本語 Wikipedia の様々な構造から is-a 関係(rdfs:subClassOf)，クラス-インスタンス関係(rdf:type)，プロパティ名とトリプル，プロパティ定義域(rdfs:domain)，プロパティ値域(rdfs:range)，プロパティ上位下位関係(rdfs:subPropertyOf)，上位下位関係(jwo:hyper)，関連語・同義語(jwo:nearly)，動詞とプロパティの関係(jwo:verb)を抽出し，関係を統合することで, 1300 万以上もの関係を持った非常に大規模で汎用的なオントロジーを自動構築することができた．加えて，日本語 Wikipedia オントロジーは，クラス，インスタンス，プロパティのオントロジー3 要素を持つだけでなく，プロパティ定義域，プロパティ値域，プロパティ上位下位関係とプロパティタイプを持つ．これはオントロジー記述言語 OWL の仕様に則ったオントロジーであり，自動構築したオントロジーの質としては非常に高いものである．さらに，手動構築されている WordNet や日本語語彙大系との比較により，日本語 Wikipedia オントロジーは他の汎用的なオントロジーには無い詳細な概念を多く持っており，汎用的なオントロジーとしての有用性を示すことができた．

日本語 Wikipedia オントロジーの利用については, 日本語 Wikipedia オントロジーのいくつかの領域でクラス, インスタンス, プロパティおよびそれらの関係を評価することで，具体物がインスタンスとなる領域では日本語 Wikipedia オントロジーは領域オントロジー支援として有用であり，再利用可能であることを示すことができた．また，日本語 Wikipedia オントロジーを Linked Open Data 化し，既存の代表的な Linked Open Data ハブである DBpedia と比較した結果，日本語 Wikipedia オントロジーは DBpedia には無い関係を多く持っており，国内 Linked Open Data ハブとして有用であることを示すことができた．加えて，外部 Linked Open Data との連携による日本語 Wikipedia オントロジーLinked Open Data の有用性を示すために，日本語 Wikipedia オントロジーと Linked Open Data を連携させて検索することができるツール WiLD（Wikipedia Linked Data Application）を実装した．外部 Linked Open Data として，XBRL Linked Open Data を用いて日本語 Wikipedia オントロジー内の科目情報と，XBRL Linked Open Data の科目情報を関連付けることで，分析支援ツールとして利用可能であった．日本語 Wikipedia オントロジーのプロパティと Linked Open Vocabularies の語彙の対応付けによる日本語語彙構築手法の提案と評価については, schema.org に代表される汎用的な語彙に関しては日

本語 Wikipedia オントロジーとの対応付けが可能であり，日本語 Wikipedia オントロジーのプロパティを用いることで，LOD 構築のための支援としての利用可能性を示すことができた．しかしながら，専門性が高い語彙については，日本語 Wikipedia オントロジーとの対応付けが十分にできず，このような専門性が高い概念を Wikipedia から抽出することが今後の課題である．

日本語 Wikipedia オントロジー構築についての今後の課題として，日本語 Wikipedia オントロジーに不足している抽象的な概念や専門性が高い概念の構築がある．抽象的な概念や専門性が高い概念は構造化することが比較的難しく，Wikipedia の本文中にその説明が書かれていることが多い．本文の多くはフリーテキストによる記述のため，本文からの情報抽出のためには自然言語処理技術の利用が必要不可欠である．フリーテキストからのオントロジー構築に関する研究は多く存在しているが，辞書に依存することが多く，あまり成果を上げられていない．フリーテキストからのオントロジー構築は高いハードルがあるため，既に抽出している日本語 Wikipedia オントロジーの概念を再利用し，Wikipedia 本文中のフリーテキストからオントロジー構築のためのルールを抽出し，ルールを基にオントロジーを再構築するといった手法が可能なのではないかと考えている．

日本語 Wikipedia オントロジーの利用についての今後の課題として，Linked Open Data を中心とした日本語 Wikipedia オントロジーの利用法を検討していく必要がある．より実用性を考慮したオントロジーとするためには，質の向上と規模の拡大が必要不可欠であり，先に述べたような概念の抽出による規模の拡大だけでなく，各種手法による抽出結果の精度の向上も同時に行わなければならない．

最後に，今後の展望として，日本語 Linked Open Data の更なる普及のために Linked Open Data 構築支援環境の整備が考えられる．現在の Linked Open Data の公開には構築者の技術，知識，手法に依存しているところが多く，ハードルが高いと言える．Linked Open Data の再利用や構築支援ツールの普及など，ハードルを下げる取り組みが必要である．

DBpedia Japanese のサービスが開始され，現在，日本語 Wikipedia オントロジーは DBpedia Japanese とリンクし合う形で，組み込まれている．DBpedia Japanese は DBpedia Japanese に無い概念を日本語 Wikipedia オントロジーから補完し，日本語 Wikipedia オントロジーは，自動構築のために日本語 Wikipedia オントロジーに不足している安定性や精度を DBpedia Japanese から補完している．このように，共に補完し合う形で，Linked Open Data ハブとして共存は可能であり，DBpedia と YAGO の関係と似ていると言える．今後はより，サービスとしての有用性を意識して，国内の Linked Open Data 普及のための構築支援に取り組んでいきたい．

# 参考文献


[1] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web,” Scientific American, pp.34-43, 2001.

[2] N. Shadbolt, T. Berners-Lee, and W. Hall, “The Semantic Web Revisited,” IEEE Intelligent Systems, vol.21, no.3, pp.96-101, 2006.

[3] F. Bond, H. Isahara, S. Fujita, K. Uchimoto, T. Kuribayashi, and K. Kanzaki, “Enhancing the Japanese WordNet,” Proceedings of the 7th Workshop on Asian Language Resources, 2009.

[4] 池原悟，宮崎正弘，白井諭，横尾昭男，中岩浩己，小倉健太郎，大山芳史，林良彦，“日本語語彙大系”，岩波書店，1997.

[5] P. Buitelaar, P. Cimiano, and B. Magnini, “Ontology Learning from Text: Methods, Evaluation and Applications,” Frontiers in Artificial Intelligence and Applications Series, IOS Press, Vol. 123, 2005.

[6] T. Yokoi, “The EDR Electronic Dictionary,” Communications of the ACM, Vol.38, No.11, pp.42-44, 1995.

[7] Free Software Foundation, “GNU Free Documentation License,” 2008,

http://www.gnu.org/licenses/fdl-1.3.en.html

[8] ISO/IEC 9075:2008, “Information technology--Database languages--SQL,” 2008.

[9] T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, and F. Yergeau, “Extensible Markup Language (XML) 1.0 (Fifth Edition),” W3C Recommendation, 2008,

http://www.w3.org/TR/xml/

[10] 中山浩太郎，伊藤雅弘，Erdmann, M.，白川真澄，道下智之，原隆浩，西尾章治郎，“Wikipedia マイニング 近未来チャレンジキックオフ編”，人工知能学会論文誌，Vol. 24，No. 6，pp.549--557，2009.

[11] F. Manola and E. Miller, “RDF Primer,” W3C Recommendation, 2004,

http://www.w3.org/TR/rdf-primer/

[12] T. Berners-Lee, “Design Issue: Linked Data,” 2009,

http://www.w3.org/DesignIssues/LinkedData

[13] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, “DBpedia: A Nucleus for a Web of Open Data,” 6th International Semantic Web Conference, Vol. 4825, pp.722-735, 2007.

[14] A. Maedche and S. Staab, “Discovering Conceptual Relations from Text,” Proceedings of the ECAI2000, pp.321-325, 2000.

[15] D. Faure and C. Nedellec, “Knowledge Acquisition of Predicate Argument Structures from Technical Texts Using Machine Learning: The System ASIUM,”

Proceedings of the EKAW, vol.1621, pp.329-334, Springer, 1999.

[16] P. Y. Vandenbussche, B. Vatant, and L. Rozat, “Linked Open Vocabularies,” Atelier Qualite et Robustesse pour le Web de Donnees IC 2011, 2011.

[17] 溝口理一郎，“オントロジー工学”，人工知能学会編集，オーム社，2005.

[18] T. R. Gruber, “Ontolingua: A Mechanism to Support Portable Ontologies,” Knowledge Systems Laboratory, Stanford University, 1991.

[19] M.K. Smith, C. Welty, and D.L. McGuinness, “OWL Web Ontology Language Guide,” W3C Recommendation, 2004, http://www.w3.org/TR/owl-guide/

[20] W3C OWL Working Group, “OWL 2 Web Ontology Language Document Overview (Second Edition),” W3C Recommendation, 2012,

http://www.w3.org/TR/owl2-overview/

[21] D. Connolly ,F. Harmelen, I. Horrocks, D. McGuinness, P. F. Patel-Schneider, and L. A. Stein, “Annotated DAML+OIL Ontology Markup,” 2001, http://www.w3.org/TR/daml+oil-walkthru/

[22] T. Berners-Lee, R. Fielding, and L. Masinter, “RFC 3986: Uniform Resource Identifier (URI): Generic Syntax,” IETF, 2005, http://www.ietf.org/rfc/rfc3986.txt

[23] M. Duerst and M. Suignard, “RFC 3987: Internationalized Resource Identifiers (IRIs),” IETF, 2005, http://www.ietf.org/rfc/rfc3987.txt

[24] D. Brickley and R. Guha, “RDF Vocabulary Description Language 1.0: RDF Schema,” W3C Recommendation, 2004, http://www.w3.org/TR/rdf-schema/

[25] H. Knublauch, R.W. Fergerson, N.F. Noy, and M.A. Musen, “The Prot´eg´e OWL Plugin: An Open Development Environment for Semantic Web Applications,” Proceedings of the Third International Semantic Web Conference, pp.229-243, 2004, http://protege.stanford.edu/

[26] T. Morita, N. Fukuta, N. Izumi, and T. Yamaguchi, “DODDLEOWL: Interactive Domain Ontology Development with Open Source Software in Java,” IEICE Transactions on Information and Systems, Vol.E91-D, No.4, pp.945-958, 2008.

[27] X.Lopez and S. Das, “Oracle Database 11g Semantic Technologies Semantic Data Integration for the Enterprise,” an Oracle White Paper, 2009.

[28] R. M. Keller, D. C. Berrios, R. E. Carvalho, D. R. Hall, S. J. Rich, I. B. Sturken, K. J. Swanson, and S. R. Wolfe, “SemanticOrganizer: A Customizable Semantic Repository for Distributed NASA Project Teams,” International Semantic Web Conference 2004, pp. 767-781, 2004.

[29] V. Lopez, M. Pasin, and E. Motta, “AquaLog: An Ontology-Portable Question Answering System for the Semantic Web,” The Semantic Web: Research and Applications, pp.546-562, 2005.

[30] F. M. Suchanek, G. Kasneci, and G. Weikum, “YAGO: A Large Ontology from Wikipedia and WordNet,” Journal of Web Semantics, Volume 6 Issue 3, pp. 203-217, 2008.

[31] J. Biega, E. Kuzey, F. M. Suchanek, “Inside YAGO2s: a transparent information extraction architecture,” Proceedings of the 22nd international conference on World Wide Web companion, pp. 325-328, 2013.

[32] S. P. Ponzetto and M. Strube, “Deriving a Large Scale Taxonomy fromWikipedia,” Proceedings of the 22nd national conference on Artificial intelligence, Vol. 2, pp.1440-1447, 2007.

[33] B. Wei, J. Liu, J. Ma, Q. Zheng, W. Zhang, B. Feng, “Motif-based Hyponym Relation Extraction from Wikipedia Hyperlinks,” Proceedings of the 19th international conference on Neural Information Processing, Vol. 5, pp.610-619, 2012.

[34] F. Wu and D. S. Weld, “Automatically Refining the Wikipedia Infobox Ontology,” Proceedings of the 17th international conference on World Wide Web, pp.635-644, 2008.

[35] M. Xu, Z. Wang, R. Bie, J. Li, C. Zheng, W. Ke, M. Zhou, “Discovering Missing Semantic Relations between Entities in Wikipedia,” Proceedings of the 12th International Semantic Web Conference, pp. 673-686, 2013.

[36] 小林暁雄，増山繁，関根聡，“日本語版ウィキペディアのカテゴリー階層に着目した日本語 WordNet 上位下位意味体系の拡張手法”, 電子情報通信学会論文誌 D, J95-D, 6, pp.1356-1368, 2012.

[37] 柴木優美，永田昌明，山本和英，“日本語語彙大系を用いた Wikipedia からの汎用オントロジー構築”，情報処理学会研究報告，自然言語処理研究会報告，2009-NL-194-4, pp. 1-8，2009.

[38] 隅田飛鳥，吉永直樹，島澤健太郎，“Wikipedia の記事構造からの上位下位関係抽出”，自然言語処理学会，Vol.16，No.3，pp. 3-24，2009.

[39] Open Government Working Group, “8 Principles of Open Government Data,” 2007,

http://www.opengovdata.org/home/8principles

[40] T. Berners-Lee, “Linked Data - Design Issues”, 2006,

http://www.w3.org/DesignIssues/LinkedData

[41] R. Cyganiak and J. Anja, “The Linking Open Data cloud diagram,” 2011,

http://lod-cloud.net/

[42] 武田英明, 嘉村哲郎, 加藤文彦, 大向一輝, 高橋徹, 上田洋, “日本における Linked Data の普及にむけて”，第 25 回人工知能学会全国大会論文集， NO.3E3-OS20-9，2011.

[43] D. Brickley and L. Miller, “FOAF Vocabulary Specification 0.98,” 2010,

http://xmlns.com/foaf/spec/

[44] A. Miles and D. Brickley, “SKOS Core Guide,” 2005,

http://www.w3.org/TR/swbp-skos-core-guide/

[45] M. Hepp, “GoodRelations Language Reference V 1.0,” 2011,

http://www.heppnetz.de/ontologies/goodrelations/v1

[46] GeoNames Team, “GeoNames Ontology V 3.1,” 2012,

http://www.geonames.org/ontology/documentation.html

[47] W3C Web Schemas group, “http://schema.org/docs/documents.html,” 2011,

http://schema.org/docs/documents.html

[48] 玉川奨，桜井慎弥，手島拓也，森田武史，和泉憲明，山口高平，“日本語 Wikipedia からの大規模オントロジー学習”，人工知能学会論文誌，Vol.25，No. 5，pp. 623-636，2010.

[49] 玉川奨，森田武史，山口高平，“日本語 Wikipedia からプロパティを備えたオントロジーの構築”，人工知能学会論文誌，Vol.26，No.4，pp. 504-517，2011.

[50] F. Giasson and Y. Raimond, “Music Ontology Specification,” 2007,

http://motools.sourceforge.net/doc/musicontology.html

[51] M. Magrane and UniProt Consortium, “UniProt Knowledgebase: a hub of integrated protein data,” Journal of Biological Databases and Curation, PMC3070428, 2011.

[52] D. Brickley, “WGS84 Geo Positioning: an RDF vocabulary,” 2009, http://www.w3.org/2003/01/geo/wgs84_pos#

[53] DCMI Usage Board, “DCMI Metadata Terms,” 2012,

http://dublincore.org/documents/dcmi-terms/

[54] 鈴木健太，玉川奨，山口高平，“大規模会計 Linked Data のためのシステムアーキテクチャ”， 第 26 回人工知能学会全国大会論文集，3C2-OS-13b-7，2012.

[55] M. Okabe, A. Yoshioka, K. Kobayashi, and T. Yamaguchi, "Organizational Knowledge Transfer Using Ontologies and a Rule-Based System," IEICE Transactions on Information and Systems, Special Section on Knowledge Based Software Engineering, Vol.E93-D, No.4,pp.763-773, 2010.

[56] 得丸英勝，“統計工学ハンドブック”，培風館，1987.

# 学位論文に関連する論文および口頭発表

## 定期刊行誌掲載論文（主論文に関連する原著論文）

1. 玉川 奨，森田 武史，山口 高平，“日本語 Wikipedia からプロパティを備えたオントロジーの構築”，人工知能学会論文誌，Vol.26，No.4，pp. 504-517，2011.
2. 玉川 奨, 桜井 慎弥, 手島 拓也, 森田 武史, 和泉 憲明, 山口 高平, “日本語 Wikipedia からの大規模オントロジー学習”，人工知能学会論文誌，Vol.25，No.5，pp. 623-636，2010

## 定期刊行誌掲載論文（その他の論文）

なし

## 国際会議論文（査読付きの full-length papers）

1. *K. Kagawa, S. Tamagawa, and T. Yamaguchi, “An automatic sameAs link discovery from Wikipedia,” The 3rd Joint International Semantic Technology Conference, 2013 (to appear)
2. *S. Tamagawa, T. Morita, and T. Yamaguchi, “Extracting Property Semantics from Japanese Wikipedia,” Proceedings of the 8th international conference on Active Media Technology, pp. 357-368, 2012
3. *T. Morita, Y. Sekimoto, S. Tamagawa, and T. Yamaguchi, “Building up a Class Hierarchy with Properties from Japanese Wikipedia,” Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology, Volume 01, pp. 514-521, 2012
4. *S. Kobayashi, S. Tamagawa, T. Morita, and T. Yamaguchi, “Intelligent humanoid robot with Japanese Wikipedia ontology and robot action ontology,” Proceedings of the 6th international conference on Human-robot interaction, pp. 417-424, 2011
5. *S. Tamagawa, S. Sakurai, T. Tejima, T. Morita, N. Izumi, and T. Yamaguchi, “Learning a Large Scale of Ontology from Japanese Wikipedia,” Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Vol. 1, pp. 279-286, 2010
6. *S. Kobayashi, S. Tamagawa, T. Morita, and T. Yamaguchi, “Human Robot Interaction Based on Wikipedia Ontology and Robot Action Ontology,” International Joint Conference on Knowledge-Based Software Engineering 2010, pp. 119-132, 2010

## その他の国際会議発表

なし

## 国内学会発表


1. *玉川 奨，香川 宏介，森田 武史，山口 高平，“日本語 Wikipedia オントロジーの Linked Open Data への取り組み”，第 27 回人工知能学会全国大会論文集，1N4-OS-10b-3，2013.
2. *玉川 奨，香川 宏介，森田 武史，山口 高平，“日本語 Wikipedia オントロジーの構築と利用”, 人工知能学会, 第 29 回セマンティックウェブとオントロジー研究会, 2013.
3. *玉川 奨，“日本語 Wikipedia オントロジーの構築と利用”，Wikimedia Conference Japan 2013，2013.
4. *森田 武史，玉川 奨，山口 高平，“オントロジーアライメントを用いた日本語 Wikipedia オントロジーと日本語 WordNet の統合”，人工知能学会学会，第 28 回セマンティックウェブとオントロジー研究会，2012.
5. *玉川 奨，森田 武史，山口 高平，“日本語 Wikipedia からのクラススキーマ階層の自動構築と利用”，第 26 回人工知能学会全国大会論文集，2C1-NFC-2-1，2012.
6. 鈴木 健太，*玉川 奨，山口 高平，“大規模会計 Linked Data のためのシステムアーキテクチャ”， 第 26 回人工知能学会全国大会論文集，3C2-OS-13b-7，2012.
7. *森田 武史, 玉川 奨, 山口 高平, “日本語 Wikipedia オントロジーと日本語 WordNet の統合”， 第 26 回人工知能学会全国大会論文集，1I2-R-4-6，2012.
8. *森田 武史，関本 有佳，玉川 奨，山口 高平，“日本語 Wikipedia からのプロパティ付きクラス階層の構築と評価”，人工知能学会学会，第 26 回セマンティックウェブとオントロジー研究会，2011.
9. *玉川 奨，関本 有佳，森田 武史，山口 高平，“日本語 Wikipedia からプロパティを備えたオントロジーの構築”，第 25 回人工知能学会全国大会論文集，2J3-NFC2-5，2011.
10. *玉川 奨, 桜井 慎弥, 手島 拓也, 森田 武史, 和泉 憲明, 山口 高平, “日本語 Wikipedia インフォボックスからのプロパティ自動抽出”, 第 24 回人工知能学会全国大会論文集，2I3-NFC4-3，2010.
11. *玉川 奨, 桜井 慎弥, 手島 拓也, 森田 武史, 和泉 憲明, 山口 高平, “日本語 Wikipedia からの大規模オントロジー学習”，人工知能学会，第 4 回フレッシュマンのための人工知能交流会，2010.
12. *森田 武史，桜井 慎弥，玉川 奨，和泉 憲明，山口 高平，“日本語 Wikipedia オントロジーの構築および検索システムの実装”，情報システム学会，第 5 回全国大会・研究発表大会，2009.

## その他

1. 玉川 奨，香川 宏介，森田 武史，山口 高平，“大規模 Linked Open Data のための日本語語彙の構築”，人工知能学会論文誌，Vol.29，No.4，2014 ，投稿中

# 謝辞

本論文を執筆するにあたり,多くの方々から多大なるご指導およびご助言を賜りました.

本研究を行う契機と環境を与えて下さり，研究の全過程を通じて，常に温かく適切な御指導を頂いた，山口高平先生に心から感謝いたします.

本論文の副査を快諾していただき，本論文の執筆にあたり，有益な御助言および御指導を頂いた，櫻井彰人先生，鈴木秀男先生，萩原将文先生に厚く御礼申し上げます.

本研究を進めるにあたって，適切な御指導および御助言を頂いた，青山学院大学社会情報学部森田武史先生に深く感謝いたします.

産業技術総合研究所和泉憲明先生，山口研究室手島拓也氏，桜井慎弥氏，関本有佳氏，香川宏介氏には本研究を進めるにあたって，多くの有益なコメントをいただきました．ここに感謝いたします.

研究活動全般に渡って，御援助下さった山口研究室の学生諸氏に感謝いたします.

また，本研究の一部は，日本学術振興会特別研究員(DC2)時代に行ったものであり，JSPS 科研費 12J00003 の助成を受けております．ここに感謝いたします.

最後に，快く大学院への進学を許可してくれたことをはじめ，博士課程在学中において経済的，精神的に支えてくれた両親，そして妻へ感謝の言葉を送りたいと思います.