

Wikipedia を活用したセマンティック Web コンテンツの半自動構築および Web API の実装

代表研究者 森田 武史 慶應義塾大学大学院理工学研究科 特別研究助教

1 はじめに

現在の Web コンテンツは、人間が理解することを目的として構築されているため、ソフトウェアによる理解は困難である。一方、次世代 Web の候補の一つであるセマンティック Web は、ソフトウェアが意味理解可能な辞書（オントロジー）に基づいて、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 コンテンツを理解し、推論することを可能にしようという試みである。セマンティック Web の実現により、Web サービスの自動的な連携、Web 上の異種データの分析、アプリケーションを横断したデータ統合および再利用などが可能となる。しかしながら、ソフトウェアが Web コンテンツを理解するためのセマンティック Web 標準技術である、RDF、RDFS、OWL により記述されるセマンティック Web コンテンツは、現状では人手によって構築されており、多大なコストを要している。

一方、近年、ハイパーリンクやフィードを活用した半構造情報資源が広がりを見せている。中でも即時更新性・語彙網羅性に優れたオンライン百科事典 Wikipedia がその代表例であり、セマンティック Web コンテンツ構築のためのリソースとして注目を集めている。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、セマンティック Web コンテンツへ直接結び付けることは難しい。

そこで、本研究では、Wikipedia からセマンティック Web コンテンツを半自動的に構築する手法を提案し、様々なソフトウェアから活用できるように Web API として実装することを目的とする。本研究により提供される Web API は、Web サービスにおける入出力データの意味づけ（セマンティック Web サービス）、ソーシャルタギングにおけるタグへの意味づけ（セマンティックソーシャルタギング）、Web ページへのアノテーション（セマンティックアノテーション）、領域オントロジー構築支援などに活用可能な、セマンティック Web ひいては電気通信関連技術における意味基盤となるものである。網羅性の高い Wikipedia から構築されたセマンティック Web コンテンツが、上記で示したような様々なアプリケーションから活用されることによって、アプリケーションを横断したデータ統合、分析、再利用などが容易になることが期待できる。

本研究報告の構成は次のとおりである。2 章では、関連研究を紹介する。3 章では、日本語 Wikipedia オントロジーと日本語 WordNet の統合について述べる。4 章では日本語 Wikipedia オントロジー検索システムの構成について述べる。5 章では、日本語 Wikipedia オントロジー Web サービス、6 章では、日本語 Wikipedia オントロジー検索インタフェースについて、それぞれ述べる。最後に、6 章で本研究報告をまとめる。

なお、紙面の都合上、日本語 Wikipedia から構築したオントロジー（日本語 Wikipedia オントロジー）構築手法については、[1] を参照していただきたい。

2 関連研究

海外の Wikipedia からセマンティック Web コンテンツを構築する主な研究として、DBpedia[2] と YAGO[3] があげられる。

DBpedia は Wikipedia の半構造情報資源を RDF データに変換することによって、大規模な RDF データベースを構築している。RDF データ構築に利用している主なデータソースは、英語 Wikipedia の Infobox や外部リンク、カテゴリといった半構造情報資源である。英語 Wikipedia を核として、Wikipedia の多言語リンクを活用することで、多言語の RDF データを構築している点も特徴的である。2011 年 1 月現在、97 の言語、約 6 億 7200 万の RDF トリプルを構築しており、世界最大規模の RDF データベースとなっている。また、Linked Open Data のハブとしても広く利用されている。しかしながら、英語 Wikipedia を核にしていることもあり、予備実験として日本語 Wikipedia から構築したセマンティック Web コンテンツのラベル (rdfs:label プロパティの値) と DBpedia のラベルを照合したところ、カバー率は 20%程度であった。つまり、本研究により構築するセマンティック Web コンテンツは DBpedia を補うことが可能であり、日本語 Wikipedia からセマンティック Web コンテンツを構築する意義はあると考えられる。

YAGO は、Conceptual Category と呼ばれるカテゴリをクラスとして利用し、WordNet を拡張している。Conceptual Category とは、カテゴリ名の head 部分が複数形になっている英語 Wikipedia のカテゴリのことである。例えば、“American singers of German origin”カテゴリは、head 部分が“singers”のように複数形になっているため、Conceptual Category といえる。YAGO では、このようにして特定された Conceptual Category に属する記事をインスタンスとしてクラスに付加している。インスタンスに関しては、BORNINYEAR や LOCATEDIN といったプロパティを用いて RDF トリプルを記述し、非階層関係も抽出している。YAGO は大規模なインスタンスの収集を可能にしているが、その手法は英語 Wikipedia に特化した手法である。本研究では、日本語 Wikipedia に対応した手法により、なるべく言語に依存しない汎用的なセマンティック Web コンテンツ構築手法を試みている。例えば、本研究では、一覧記事からのクラス-インスタンス関係構築を試みているが、一覧記事は日本語 Wikipedia だけでなく、英語 Wikipedia でも“Listing Page”として存在しており、言語に依存することなくクラス-インスタンス関係を抽出できることが確認できている。

国内では、東京大学 知の構造化センターの中山らのグループが Wikipedia マイニングと呼ばれる Wikipedia を解析対象とし、有用な知識を抽出する手法を提案している[4]。現状では、主に、Wikipedia における記事のリンク間関係を分析し、関連関係を同定することにより、大規模なシソーラス辞書を構築している。本研究調査では、単に関連があるというだけではなく、より厳密に関係名にあたるプロパティの同定やクラス-インスタンスの厳密な区別など、より正確なオントロジーおよび RDF データの構築を目指している。そのため、現状では規模はそれほど大きくはないが、今後様々な手法を導入し、規模を拡大したいと考えている。

3 日本語 Wikipedia オントロジーと日本語 WordNet の統合

Wikipedia からオントロジーにおけるクラス階層を構築するために、カテゴリおよびカテゴリ階層がしばしば利用される。しかしながら、カテゴリは記事を分類するために作成されるため、記事の分類に寄与しない抽象的なカテゴリなどは、わずかにしか作成されていない。そのため、カテゴリ階層から構築したクラス階層は、上位層において抽象的なクラスが不足している。この欠点を補うために、本研究では日本語 WordNet[5] と日本語 Wikipedia オントロジーの統合を試みる。日本語 WordNet は、独立行政法人情報通信研究機構 (NICT) が開発した日本語の意味辞書であり、人手で作成された英語の意味辞書である Princeton WordNet 3.0 に準じているため、抽象的な概念 (クラス) が数多く含まれている。

抽象的な概念を導入するメリットとして、Web アプリケーション開発者がセマンティック Web コンテンツを活用する際に、曖昧な要求 (例:「人物」に関するクラスとそのインスタンスデータが取得したいなど) に対応できる点があげられる。本研究では、日本語 Wikipedia から構築したセマンティック Web コンテンツと日本語 WordNet を統合するために、オントロジーアライメントの技術を適用する。

日本語 Wikipedia オントロジーと日本語 WordNet の統合を行うための準備として、はじめに、同義語候補抽出とその活用方法について検討を行った。[6] より、オントロジーアライメントにおける概念の類似性の尺度として、「語類似度」、「語リスト類似度」、「概念階層類似度」、「構造類似度」の主に 4 種類の手法がすべて有効であることが示された。「語類似度」は、他の 3 種類の手法の基礎となるが、「語類似度」を求めするためには、概念の同義語が必要となる。

そこで、日本語 Wikipedia におけるリダイレクトリンクと曖昧さ回避のためのページを利用して、同義語候補抽出を試みた。リダイレクトとは、ユーザが Web ブラウザを介してある URL にアクセスした時に、別の URL に転送する機能である。Wikipedia では、記事の別名を表す URL にアクセスした時に、その記事の正式名を表す URL に転送する機能をリダイレクトと呼ぶ。また、リダイレクト元からリダイレクト先へのリンクをリダイレクトリンクと呼ぶ。例えば、「福沢諭吉」(別名)を表す記事の URL¹にアクセスした場合、「福澤諭吉」(正式名)を表す記事の URL²にリダイレクトされる。Wikipedia のダンプデータ[7]には、リダイレクトリンクをまとめたテーブルが用意されているため、そこから、リダイレクト元を別名、リダイレクト先を正式名として、同義語候補セットを抽出した。

さらに、曖昧さ回避のためのページも同義語候補抽出のために用いた。曖昧さ回避のためのページとは、記事の別名が多義語となっている場合に、それらを判別するために用意されたページである。例えば、「スピード」は、トランプゲーム、アイドルグループ、アメリカ映画など、複数の意味を表す多義語であるため、

¹ <http://ja.wikipedia.org/wiki/福沢諭吉>

² <http://ja.wikipedia.org/wiki/福澤諭吉>

先に述べたリダイレクトリンクとして、一意にリダイレクト先を決定することができない。そのため、「スピード」を表す記事の URL にアクセスした場合には、曖昧さ回避のためのページが表示され、その中から、「スピード (トランプ)」、「SPEED」、「スピード (映画)」などをユーザが選択できるようになっている。曖昧さ回避のためのページは、「曖昧さ回避」カテゴリに属しているため、そのカテゴリに属する記事を Wikipedia のダンプデータから抽出し、記事のタイトルを別名、記事内で箇条書きとして列挙されたリンクのアンカーテキストを正式名として、Web スクレイピングすることにより、同義語候補セットを抽出した。

以上 2 種類の方法で抽出した同義語候補セットを統合し、Wikipedia から構築したセマンティック Web コンテンツにおける、同義語候補セットとした。各方法で抽出した同義語候補セット数を表 1 に示す。なお、Wikipedia のダンプデータは、2010 年 6 月時点のものを利用している。

表 1: リダイレクトリンクおよび曖昧さ回避のためのページから抽出した同義語候補セット数

(1) リダイレクトリンクから抽出した同義語候補セット数	197,915
(2) 曖昧さ回避のためのページから抽出した同義語候補セット数	127,581
(1) と (2) を統合した同義語候補セット数	291,709

ここで、日本語 Wikipedia オントロジーと日本語 WordNet のアライメントを含めて、抽出した同義語候補セットについて、以下の 3 つの活用方法を検討した。

- 3-1 日本語 Wikipedia オントロジーにおけるクラスおよびインスタンスの同義語定義
- 3-2 日本語 Wikipedia オントロジーにおけるインスタンス間の owl:sameAs 関係定義
- 3-3 日本語 Wikipedia オントロジーと日本語 WordNet のアライメント

以下では、それぞれについて、説明する。

3-1 日本語 Wikipedia オントロジーにおけるクラスおよびインスタンスの同義語定義

日本語 Wikipedia オントロジーにおけるインスタンス名は、基本的には Wikipedia における記事の正式名に対応している³。そこで、インスタンス名（正確にはインスタンスの URI におけるローカル名）と各同義語候補セットにおける正式名について、完全文字列照合を行い、照合した同義語候補セットをインスタンスの同義語として定義した。また、日本語 Wikipedia オントロジーにおけるクラス名は、基本的には Wikipedia におけるカテゴリ名に対応している⁴。Wikipedia におけるカテゴリ名には、別名は定義されていないが、有名な記事はカテゴリ化される特徴が Wikipedia にはあることから、カテゴリ化された記事に別名が存在する場合には、それを対応するカテゴリの別名とみなすことができると考えられる。以上より、クラスおよびインスタンス名と同義語候補における正式名について、完全文字列照合を行った。照合結果は、以下のとおりである。

- 同義語候補における正式名とクラス名の完全文字列照合結果
 - ・ 日本語 Wikipedia オントロジーにおけるクラス数：45,149
 - ・ 照合数：9,739
- 同義語候補における正式名とインスタンス名の完全文字列照合結果
 - ・ 日本語 Wikipedia オントロジーにおけるインスタンス数：655,633
 - ・ 照合数：168,101

以上の結果より、完全文字列照合のみでは、照合できないクラスおよびインスタンスが多数あることがわかった。そのため、完全文字列照合のみではなく、文字列の類似度を測る手法を適用する必要がある。文字列の類似度を測る手法は、[8] で多数紹介されており、ライブラリとして実装も公開されているため、こ

³一部、一覧記事や Infobox における項目の値から抽出したインスタンスについては、別名がインスタンス名となる場合がある。これについては、「3-2 日本語 Wikipedia オントロジーにおけるインスタンス間の owl:sameAs 関係定義」で対応策を示す。

⁴一部、前方文字列照合部除去により獲得したクラスについては、Wikipedia のカテゴリ名と一致しないクラスも存在する ([1] 参照)。

れらの中から適切な手法を選択し、類似度がある一定の閾値よりも大きい同義語候補セットの正式名とクラスおよびインスタンス名を同一とみなすことで、より多くの同義語定義ができると考えられる。

3-2 日本語 Wikipedia オントロジーにおけるインスタンス間の owl:sameAs 関係定義

Wikipedia オントロジーでは、一覧記事や Infobox から抽出したインスタンス名が正式名ではなく、別名となる場合がある。一覧記事の場合、正式名か別名かをページ更新者が意識することなく、項目を記載できることがこの原因である。Wikipedia は人間向けに作られているため、リダイレクトにより、最終的に目的とする記事が Web ブラウザを介して表示されれば問題ない。そのため、Wikipedia では、ページ更新者の利便性を考慮して、一覧記事における各項目について、対応する記事にリンクを張る際に、厳密にアンカーテキストが別名か正式名かの確認を行っていない。Infobox における項目の値が Wikipedia の記事へのリンクとなっている場合にも、同様の問題が生じている。このことは、同一インスタンスを表しているにもかかわらず、正式名と別名の URI が別々に定義される可能性があることを示している。図 1 に示すように、クラス-インスタンス関係とインスタンスに関する RDF トリプルを統合する際に、問題となる。図 1 では、一覧記事からは「福沢諭吉」インスタンスが、Infobox からは「福澤諭吉」インスタンスがそれぞれ抽出されている。これらを機械的に統合しようとしても、異なるインスタンスと見なされて、統合できないという問題が生じる。

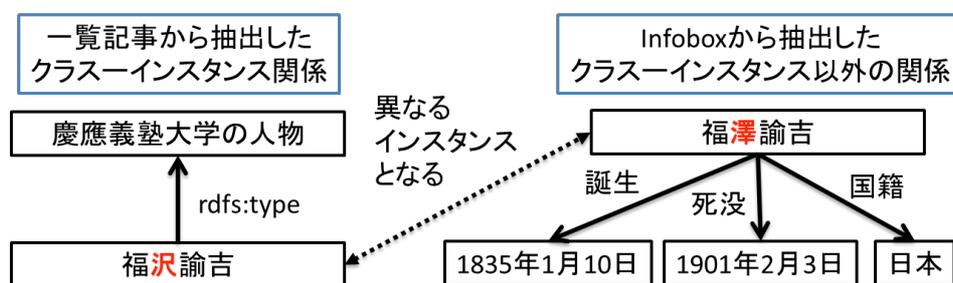


図 1: 同一インスタンスについて正式名と別名の URI が別々に存在する例

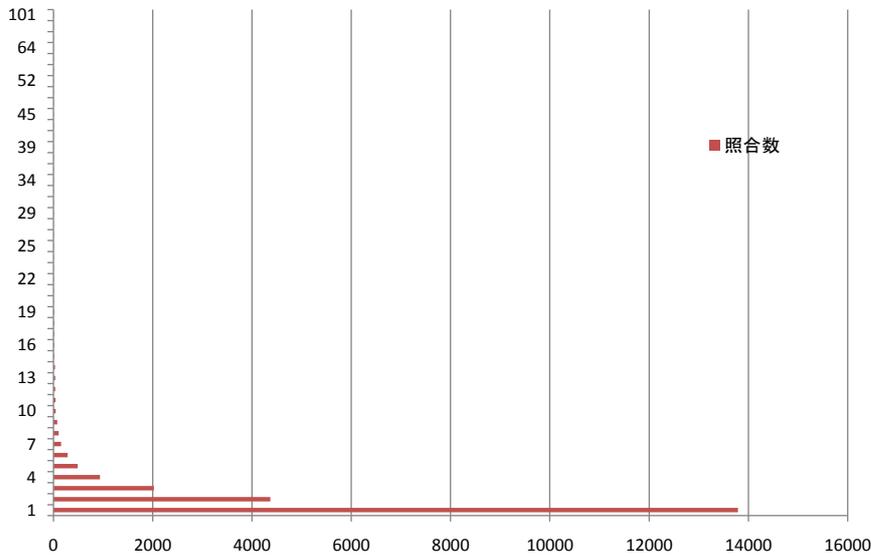
以上の問題を解決するために、同義語候補セットの利用を検討した。

同義語候補と Wikipedia における全記事名を利用して、別名の URI と正式名の URI を owl:sameAs 関係で定義する (または、別名の URI を正式名の URI に置換する)。以下の手順によりインスタンス間に owl:sameAs 関係を定義する。

1. 全インスタンス名と同義語候補セットにおける正式名を完全文字列照合し、正式名を持ちかつ同義語を持つインスタンスのセット (Irs) を獲得する。
2. 全インスタンス名と Wikipedia における全記事名を完全文字列照合し、正式名を持つインスタンスのセット (Ir) を獲得する。
3. 全インスタンスのセットを I とする時に、別名を持つインスタンスのセット ($I_a = I - Ir$) を獲得する。
4. 別名のインスタンスに対応する正式名のインスタンスリストのセットを獲得する。具体的には、 I_a の各要素 $ia_1 \dots ia_n$ について Irs の各要素 $irs_1 \dots irs_n$ の同義語セット $Sirs_1 \dots Sirs_n$ と照合を行う。Irs 要素における同義語セットのいずれかの要素と照合した I_a の要素と該当する Irs の要素とを対応づける。

以上より、基本的には owl:sameAs 関係をインスタンス間に定義することができるが、別名を持つインスタンスに対応する正式名を持つインスタンスが複数ある場合には、多義性解消を行う必要がある。図 2 に、owl:sameAs 関係定義の実験結果を示す。

- |I| (全インスタンス数) = 655,633
- |Irs| (正式名を持ちかつ同義語を持つインスタンス数) = 168,101
- |Ia| (別名を持つインスタンス数) = 22,544



1	13788	24	1
2	4369	25	3
3	2024	26	4
4	937	27	5
5	487	29	1
6	284	31	1
7	153	32	3
8	104	34	3
9	77	36	1
10	48	37	3
11	41	39	1
12	35	40	2
13	35	42	3
14	31	45	1
15	16	49	1
16	13	51	1
17	16	52	1
18	15	54	1
19	13	63	1
20	7	64	1
21	2	84	1
22	5	93	1
23	3	101	1

図 2: Wikipedia から抽出した同義語候補における別名と Wikipedia オントロジーにおけるインスタンスの照合結果

図 2 より, 8,756 個のインスタンスについては, 二つ以上の正式名を持つインスタンスと照合しているため, 多義性解消を行う必要があり, 今後の課題である. 以下に, 「ソルトレイク」が正式名「ソルトレイクシティ」または「グレートソルト湖」の別名になっている例を示す. 下線は正式名, 太字は別名を表している. また, 「229871」と「282247」は, 同義語候補セットに割り振った識別子をそれぞれ表している.

例: ソルトレイク, 229871, 282247

229871, ソルトレイクシティ, Salt Lake City, **ソルトレイクシティ**, **ソルトレイク・シティ**,
ソルトレーク市, ソルトレークシティ, **ソルトレイク市**, **ソルトレークシティ**市,
ソルトレーク・シティ, **ソルトレークシティ**, **ソルトレイク**

282247, グレートソルト湖, **グレートソルトレイク**, **ソルトレイク**

3-3 日本語 Wikipedia オントロジーと日本語 WordNet のアライメント

日本語 Wikipedia オントロジーにおけるクラス同義語定義を利用して, 日本語 Wikipedia オントロジーと日本語 WordNet のアライメントを行った. アライメントは以下の手順で行った.

1. 日本語 Wikipedia オントロジーにおけるルートクラス (3,082) (正式名とその同義語セット) と日本語 WordNet (ver. 1.1)における名詞の synset (82,115) (同義語セット) について, [6] を参考にし, 文字列に基づく 4 種類の手法 (「プレフィックス」, 「サフィックス」, 「編集距離」, 「n グラム」) を適用し, 類似度の高い 5 つの synset を求めた. 日本語 Wikipedia オントロジーにおけるクラスの正式名と日本語 WordNet における synset が照合した場合には, 2 倍の重みをつけた. また, 「n グラム」については, 「3 グラム」で実験を行った.
2. 同一類似度の synset と照合した場合には, synset の ID で昇順にソートした.
3. 最も類似度の高い synset と Wikipedia オントロジーのルートクラスを対応づけた.

表 2 に日本語 Wikipedia オントロジーにおけるルートクラスと日本語 WordNet における synset のアライメント評価結果を示す. 本評価実験では, 3,082 個のルートクラスの中から 300 クラスをランダムに抽出し, 人手により, 日本語 WordNet における synset とどのような関係で対応付けがなされたかを評価した.

表 2: 日本語 Wikipedia オントロジーにおけるルートクラスと日本語 WordNet における synset のアライメント評価結果

	数	割合
sameAs	52	0.173
isa	90	0.300
isa-reverse	20	0.067
sibling	21	0.070
failure	101	0.337
error	16	0.053
すべて	300	1

表 2における「sameAs」は同値関係として対応付けがなされたことを意味する。「isa」は日本語 WordNet における synset を親クラス、日本語 Wikipedia オントロジーにおけるルートクラスを子クラスとした対応付けがなされたことを意味する。「isa-reverse」は、日本語 Wikipedia オントロジーにおけるルートクラスを親クラス、日本語 WordNet における synset を子クラスとした対応付けがなされたことを意味する。「sibling」は、日本語 Wikipedia オントロジーにおけるクラスと日本語 WordNet における synset が兄弟関係として対応づけられたことを意味する。「failure」は誤って対応付けられたことを意味する。「error」は日本語 Wikipedia オントロジーにおけるルートクラスの中で、「過去ログページ」や「経」など、意味をなさないクラスとの対応付けが行われたことを意味する。

表 3から表 8に、各関係についての具体例を示す。日本語 Wikipedia オントロジーにおけるクラスの正式名および日本語 WordNet の synset の ID は太字で示している。

表 3: sameAs 関係の具体例

日本語 Wikipedia オントロジーのルートクラス (正式名および同義語セット)	日本語 WordNet の synset (ID および同義語セット)
シオニズム, シオニスト, シオニズム運動, シオニスト運動	06663463-n , zionism, シオニズム
アルコール	07884567-n , alcoholic_beverage, 気違水, アルコホル, 御神酒, アルコホル, intoxicant, お酒, 気違い水, alcohol, 気狂い水, 酒, アルコホル飲料, 酒類, inebriant, alcoholic_drink
水雷艇, はやぶさ, T-26, 第一号型, T-27, パリフィーマス, 千鳥	04017807-n , mosquito_boat, mosquito_craft, 魚雷艇, motor_torpedo_boat, pt_boat
非営利団体, 非営利組織, AAM, JAM, NPO	01137597-n , not-for-profit, 非営利的, 非営利, nonprofit_organization, nonprofit, 非営利団体
人類学	06143546-n , 人類学, anthropology

表 4: isa 関係の具体例

日本語 Wikipedia オントロジーのルートクラス (正式名および同義語セット)	日本語 WordNet の synset (ID および同義語セット)
デンマーク海軍艦艇	02858304-n , ボート, 小舟, 艇, 舟艇, バッテラ, 短艇, boat, 舟, 小船, 猪牙, バッテラ
岐阜県保育所	03165466-n , day_care_center, day_nursery, 保育所, 託児所, デイクケアセンター
ガンダムシリーズの世界観	06211389-n , weltanschauung, world_view, 世界観, 宇宙観
民族衣装	02728440-n , apparel, ウェア, 御召し物, 御衣, ドレス, 装束, 衣紋, ベベ, 御召し, 衣裳, お召し, 着衣, 着類, 御召物, お召し物, ウェア, 着物, 御召, 衣料, 衣服, 衣類, 衣装, おべべ, 洋服, コスチューム, お召, 服, clothes, お召しもの, お召物, dress, 狭衣, 衣, アパレル, wearing_apparel, 御召しもの, 着り物, 被服

対戦型格闘ゲームキャラクター	04620216-n, 心柄, 資, 人柄, 性情, character, 質, 品性, 根性, 地, 稟性, 性合, 本質, 人となり, 性合い, 根, 人品, fiber, 性格, 性根, キャラクター, 素質, キャラクタ, 性, 気質, 気っ風, 性質, 気だて, fibre, 人格
----------------	--

表 5: isa-reverse 関係の具体例

日本語 Wikipedia オントロジーの ルートクラス (正式名および同義語セット)	日本語 WordNet の synset (ID および同義語セット)
教育学, ロート	00889760-n, 音楽教育学, music_lesson
作品, ネタ, DOG, 創造物, 芸術作品, ウル	04601690-n, 芸術作品, 芸術品, 美術品, work_of_art
神, Deities, 神業, ゴッド, GOD, 日本の神, 天主, エロヒーム, 神祇, 神々, エロヒム, 天, 神様, 神族, ブラン, クリエイター, 守護神	09560061-n, cupid, 愛の神, amor
害虫, 虫害	01314388-n, 病害生物, pest, 病害虫
ネコ, ニャンニャン, トムキヤット, 猫ドア, ニャンコ, ポッポ, 子猫, スー, マンチキン, イエネコ, ニャーニャー, ねこ, キヤット, エジプシャン, キヤッツ, ボンベイ, CAT, ギルバート, 猫, ホームズ	02123597-n, シヤムネコ, siamese_cat, siamese, シヤム猫

表 6: sibling 関係の具体例

日本語 Wikipedia オントロジーの ルートクラス (正式名および同義語セット)	日本語 WordNet の synset (ID および同義語セット)
ラトビア人, ラトヴィア人	02379908-n, アラビア人, arabian, arab
NHL チーム	08079613-n, club, ball_club, nine, 球団, プロ野球チーム, クラブ, チーム, 野球クラブ, ナイン, baseball_club
公文書館, 文書館	03660664-n, 寄託図書館, ライブラリー, 文庫, depository_library, ライブラリ, library, 図書館, 文庫
地域自治区, 区	09039260-n, iraqi_kurdistan, クルド人自治区
ソロモン諸島の州	09092497-n, pine_tree_state, マツの州, me, 松の州, メーン州, メーン, maine

表 7: failure の具体例

日本語 Wikipedia オントロジーの ルートクラス (正式名および同義語セット)	日本語 WordNet の synset (ID および同義語セット)
隋末唐初の群雄	01321230-n, 雄, 牡, オス, male
ラグビーワールドカップ	03147509-n, カップ, cup, コップ
海賊	11110917-n, alfred_louis_kroeber, クローバー, alfred_kroeber, kroeber
年代別アニメソング (エンディング曲)	00303748-n, ハンググライディング, hang_gliding
宮崎県高等学校の廃校	08276720-n, 学院, スクール, 学校, school, 学園

表 8: error の具体例

日本語 Wikipedia オントロジーの ルートクラス (正式名および同義語セット)	日本語 WordNet の synset (ID および同義語セット)
試行 2	13743269-n, deuce, 両, 2, 2, two, ii, 式
col	09249034-n, gap, col, 鞍部
経	05476256-n, 脳神経, cranial_nerve
Peri_GR	02988486-n, CD-R, cd-wo, cd-r, CD-WO, compact_disc_write-once, compact_disc_recordable, シーディーアール
過去ログページ	06256697-n, ペイジ, page, ページ

図 3に日本語 Wikipedia オントロジーにおけるルートクラスと日本語 WordNet における synset のアライメント結果を視覚化した画像の一部を示す。視覚化には Graphviz⁵を用いた。図 3において楕円は日本語 WordNet における synset (1 行目は ID, 2 行目以降は最大 5 個の同義語), 青色の矩形は日本語 Wikipedia オントロジーにおけるルートクラス (1 行目は正式名, 2 行目以降は最大 5 個の別名), 赤色の点線はアライメントにより対応づけられた関係を示している。

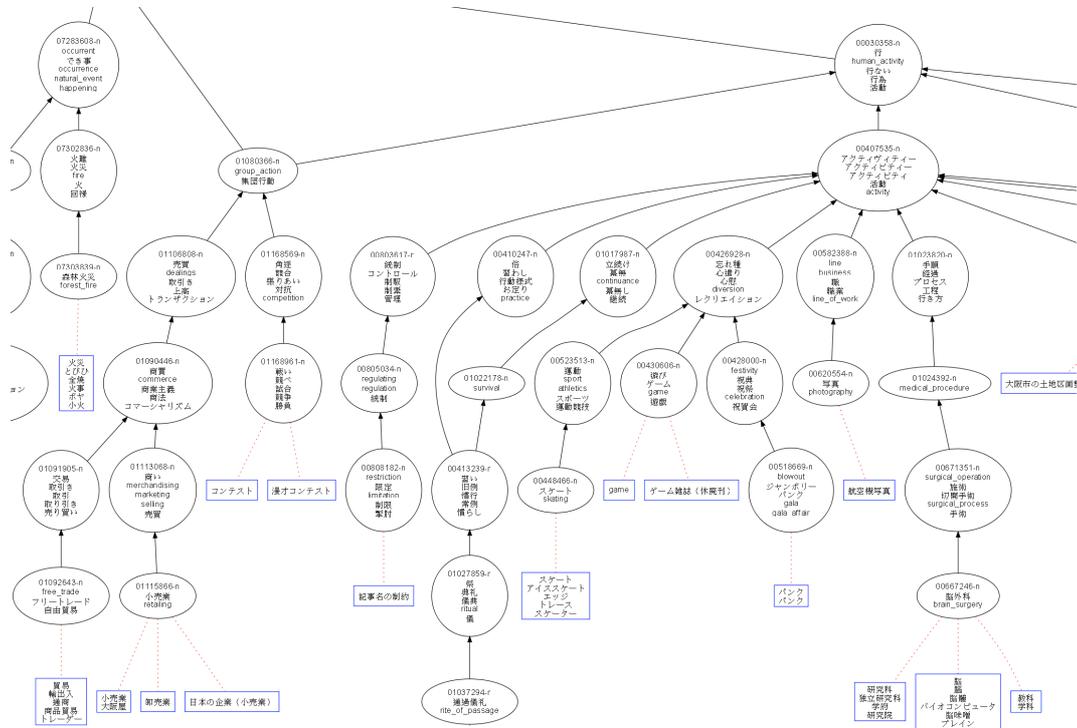


図 3: 日本語 Wikipedia オントロジーにおけるルートクラスと日本語 WordNet における synset のアライメント結果の視覚化

表 2より, 文字列に基づく 4 種類の手法を用いたアライメント結果として, 約 37%は isa 関係(isa および isa-reverse), 約 34%は照合失敗, 約 17%は同値関係, 約 7%は兄弟関係, 約 5%はエラーとなることがわかった。今後は, ユーザとのインタラクションを通して関係を同定可能なツールの研究開発を行う予定である。また, より高精度に日本語 Wikipedia オントロジーと日本語 WordNet を統合できるように, 照合に失敗した原因を分析し, [8] [9]などを参考に, その他のオントロジーアライメント手法の適用を検討していきたい。

3-4 同義語候補洗練の課題

抽出された同義語候補セットが同義語としてどの程度正しく抽出されているかを確認するために, 特にリダイレクトリンクから抽出した同義語候補セットから 1000 個の標本抽出を行い正誤判定を行ったところ, 7割程度の正解率であった。このことから, 抽出された同義語候補には, 多くの誤りが含まれているため, 同義語候補の洗練を行う必要があると考えられる。

同義語候補の洗練方法として, 既存のシソーラスおよび汎用言語オントロジーの活用を検討した。既存のシソーラスとしては, 国立国会図書館件名標目表 (NDSLH) [10]に着目した。NDSLHは, 国立国会図書館の目録を検索するために維持・管理している件名標目表である。提供するデータ構造としては, 同義語, 上位語, 下位語, 関連語などがあり, 2010年6月末頃からRDF/XML形式のデータが提供されはじめたことと, 研究目的・非営利目的の利用がしやすいことから, 利用を検討した。以下の予備実験では, 2010年6月末時点のデータを用いた。NDSLHの主なデータ数は以下のとおりである。

- 1,294,666 トリプル
- 96,736 の件名標目

⁵ <http://www.graphviz.org/>

また、既存の汎用言語オントロジーとしては、日本語 WordNet[5] に着目した。こちらも、NDLSH と同様、研究目的・非営利目的の利用がしやすいことから、利用を検討した。以下の予備実験では、日本語 WordNet 1.0 を用いた。日本語 WordNet 1.0 に収録された synset 数や単語数、語義数は次のとおりである。

- 56,741 概念 (synset 数)
- 92,241 words 語
- 157,398 語義 (synset と単語のペア)
- 135,692 定義
- 48,276 事例

以上を踏まえて、NDLSH と日本語 WordNet を利用した Wikipedia オントロジーにおける同義語候補の洗練に関する予備実験を行った。実験手順として、はじめに、NDLSH と日本語 WordNet から同義語セットを抽出した。NDLSH は、RDF 形式のデータを提供しているため、図 4 に示す SPARQL クエリにより、代表語（正式名）と別名をそれぞれ抽出した。NDLSH の RDF モデルについては、[11] を参考にした。

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX xl: <http://www.w3.org/2008/05/skos-xl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?prefLabel ?altLabel
WHERE {
    ?x xl:prefLabel [xl:literalForm ?prefLabel];
        xl:altLabel [xl:literalForm ?altLabel].
}
```

図 4: NDLSH から代表語と別名を抽出するための SPARQL クエリ

日本語 WordNet は、SQLite 形式でデータを提供しているため、SQLite JDBC Driver[12] を用いて、synsetID と synset を同義語セットとして抽出した。英語版の WordNet では、synset に含まれる語が、使用頻度に基づいてランク付けされているため、ある synset における代表語（正式名）が明確である。しかし、日本語 WordNet ver. 1.0 では、使用頻度やランクに関する情報が欠落していたため、日本語 WordNet から抽出した同義語セットについては、代表語（正式名）は不明であった。

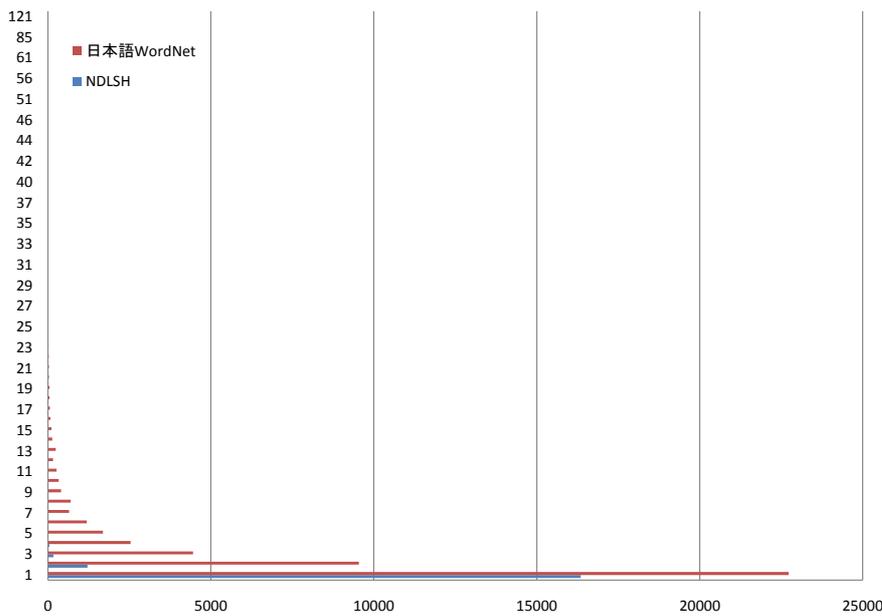
NDLSH と日本語 WordNet から抽出した同義語セット数は以下のとおりである。

- NDLSH から抽出した同義語セット数： 22,748
- 日本語 WordNet から抽出した同義語セット数： 95,883

次に、Wikipedia のリダイレクトリンクおよび曖昧さ回避のためのページから抽出した同義語候補セット (candidate_for_wikipedia_synonyms) と NDLSH および日本語 WordNet の同義語セット (ndlsh_synonyms および jpwn_synonyms) を完全文字列照合した。candidate_for_wikipedia_synonyms の各同義語候補セットについて、ndlsh_synonyms と jpwn_synonyms の全同義語セットと照合を行い、1 つ以上共通の同義語を持つ ID を対応付ける。同義語セット同士の照合は、完全文字列照合とした。

図 5 に、Wikipedia から抽出した同義語候補と日本語 WordNet および NDLSH の同義語との完全文字列照合結果を示す。図 5 より、NDLSH の同義語と 1 つ以上照合した Wikipedia の同義語候補セット数は、17,808 であった。また、日本語 WordNet の同義語と 1 つ以上照合した Wikipedia の同義語候補セット数は、45,570 であった。Wikipedia から抽出した全同義語候補セット数は、291,709 であることから、完全文字列照合のみでは、網羅率は低いといえる。その他、意味が一意に定まっていないため、多義性解消を行う必要もある。

今回は、すべての同義語候補セットに対して実験を行ったが、Wikipedia オントロジーにおけるクラスとインスタンスのみについて、照合を行った場合の網羅率についても検証する必要がある。さらに、NDLSH と日本語 WordNet に照合しなかった Wikipedia 固有の同義語候補について、どのように正誤判定を行うべきかも今後の課題である。



	NDLISH	日本語WordNet
1	16350	22730
2	1213	9538
3	167	4452
4	40	2536
5	13	1687
6	7	1191
7	4	648
8	4	699
9	1	401
10	1	333
11	0	266
12	2	158
13	2	238
14	2	134
15	0	109
16	0	79
17	0	58
18	0	46
19	0	51
20	0	30
21	1	33
22	0	24
23	0	17
24	1	14
25	0	14
26	0	7
27	0	7
28	0	2
29	0	14
30	0	7
31	0	8
32	0	9
33	0	1
34	0	5
35	0	1
36	0	2
37	0	2
38	0	1
40	0	1
41	0	1
42	0	1
43	0	1
44	0	1
45	0	1
46	0	1
47	0	1
51	0	2
53	0	1
56	0	1
57	0	1
61	0	1
83	0	1
85	0	1
94	0	1
121	0	1

図 5: Wikipedia から抽出した同義語候補と日本語 WordNet および NDLISH の同義語の完全文字列照合結果

4 日本語 Wikipedia オントロジー検索システムの構成

図 6 および図 7 に日本語 Wikipedia オントロジー (JWO) 検索システムとそのサーバの構成を示す。JWO 検索システムは、JWO 検索インタフェース、JWO Web サービス、JWO データベース (DB) の大きく分けて 3 つの部分から構成される。

JWO 検索インタフェースは、ExtJS ライブラリ [13] を用いて実装されている。JWO Web サービスは、Apache Wicket [14]、Jena Semantic Web Framework for Java [15]、ActiveObjects [16]、Memcached Java Client [17]、TDB - A SPARQL Database for Jena [18]、H2 DB [19] を用いて実装し、Servlet コンテナ Jetty [20] 上で動作している。JWO DB は、Jena および TDB の API を用いてあらかじめ OWL 形式の日本語 Wikipedia オントロジーおよびインスタンスを TDB 形式で格納している。

JWO 検索インタフェースでは、はじめにユーザが検索キーワードを入力すると、キーワードに関連する日本語 Wikipedia オントロジー内のクラス、プロパティ、インスタンスを識別する URI に変換する。JWO Web サービスでは、クライアントプログラムから問い合わせのあった URI に関連するステートメントを JWO DB に問い合わせ、JWO 検索インタフェースに、RDF/XML や JSON データを返す。JWO 検索インタフェースは、JWO Web サービスから返された JSON データを元に、表形式や木構造で Wikipedia オントロジーのクラス、プロパティ、インスタンスにおけるステートメントやクラス階層関係を表示する。また、ソースコードとして RDF/XML データを表示することもできる。

5 章および 6 章では、JWO Web サービスと JWO 検索インタフェースについて、それぞれ説明する。

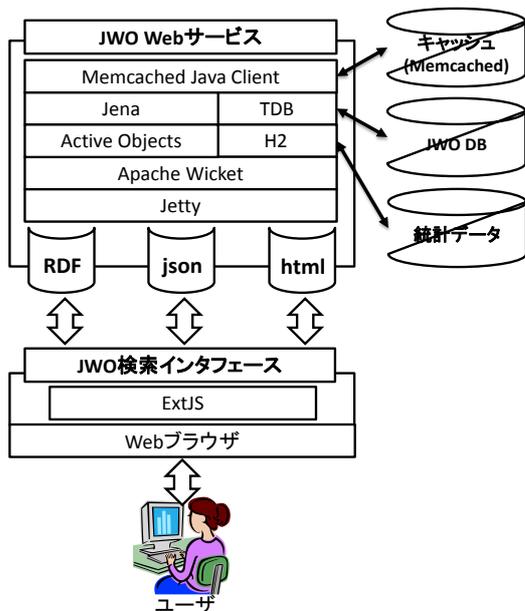


図 6: 日本語 Wikipedia オントロジー検索システムのシステム構成

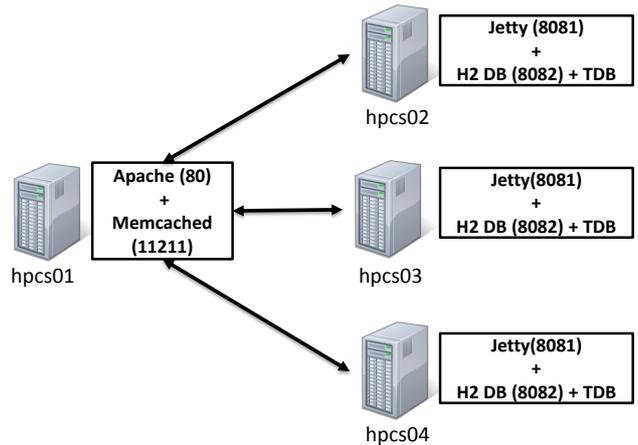


図 7: 日本語 Wikipedia オントロジー検索システムにおけるサーバの構成

5 日本語 Wikipedia オントロジー Web サービス

JWO Web サービスは以下に示す仕様に従い、日本語 Wikipedia オントロジー内のクラス、プロパティ、インスタンスに関連するデータを RDF/XML や JSON 形式で提供する。一般的には、SPARQL エンドポイントを用意し、クライアントプログラム内で SPARQL クエリを構築し、SPARQL エンドポイントにクエリを送信することにより、RDF データにおける特定のトリプルが取得できる。しかしながら、クラスのインスタンス、兄弟クラス、下位クラス、プロパティの定義域および値域、インスタンスのタイプなど、比較的頻繁に多くのセマンティック Web アプリケーション開発者によって送信されるであろう SPARQL クエリについては、あらかじめ、Web API としてシステムが提供することにより、開発コストが低減できると考えられる。また、SPARQL エンドポイントは、RDF/XML または XML 形式で結果を返すが、近年、クライアント側のプログラムに JavaScript を利用する場合が増加しつつあり、その際には、JSON 形式で結果が得られると開発がより容易となる。以上より、本研究では SPARQL エンドポイントを提供すると同時に、開発者が比較的頻繁に送信するであろう SPARQL クエリについては、あらかじめ、Web API の形式で同等の結果を提供することとした。なお、JWO Web サービスの設計にあたっては、オントロジー検索エンジン Swoogle が提供している Web サービス⁶を参考にした。

5-1 共通 URL パラメータ

JWO Web サービスでは、表 9 に示す共通 URL パラメータを利用できる。

start と limit パラメータの使用例として、以下の URL にアクセスすることで、「慶應義塾大学の人物」クラスのインスタンスに関する 500 番目～600 番目までのステートメントを取得することができる。

http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/慶應義塾大学の人物.rdf?start=500&limit=100&search_option=instances_of_class

search_option パラメータの使用例として、「日本」をラベルに含むクラスを完全一致、部分一致、前方一致、後方一致で検索した結果を取得するための URL を表 10 に示す。

⁶ http://swoogle.umbc.edu/index.php?option=com_swoogle_manual&manual=search_overview

表 9: 共通 URL パラメータ

URL パラメータ	説明
search_target	「uri」または「label」を指定可能。「uri」を指定した場合には、リソース URI のローカル名を検索対象とし、「label」を指定した場合には、「rdfs:label」プロパティの値を検索対象とする。
search_option	完全一致 (exact_match), 部分一致 (any_match), 前方一致 (starts_with), 後方一致 (ends_with) 検索オプションのいずれかを指定できる。「search_target」パラメータの値を「uri」に指定した場合には、「exact_match」のみ指定可能。
inference_type	推論の種類を指定する。現状では「rdfs」のみ指定可能。
start	start パラメータで指定した位置を開始ステートメントとして、limit パラメータで指定したステートメント数を返す。
limit	一度に返すステートメント数を指定する。
version	検索対象の Wikipedia オントロジーのバージョンを指定。現状では、「2010_11_14」と「2010_02_09」の 2 種類のみ指定可能。

表 10 search_option パラメータの使用例

search_option パラメータ値	使用例
exact_match (完全一致)	http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/日本.rdf?search_target=label&search_option=exact_match
any_match (部分一致)	http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/日本.rdf?search_target=label&search_option=any_match
starts_with (前方一致)	http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/日本.rdf?search_target=label&search_option=starts_with
ends_with (後方一致)	http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/日本.rdf?search_target=label&search_option=ends_with

5-2 クラスの取得

以下の形式の URI にアクセスすると、クラスに関連するステートメントを取得することができる。

[BASE_URI]class/[page|data]/クラス名.[rdf|n3|nt|json|jsonp]?search_option=[exact_match|sibling_classes|sub_classes|properties_of_domain_class|properties_of_range_class|instances_of_class|path_to_root_class|inverse_statements]

以下の注意点は、プロパティとインスタンスについても同様である。

- [BASE_URI]は、「http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/」を表す。
- page を指定すると HTML 形式で、data を指定すると RDF/XML, N3, N-Triples, JSON, JSONP のいずれかの形式でデータを取得可能。
- 拡張子により取得データ形式を変更可能。現状では、RDF/XML(rdf), Notation3(n3), N-Triples(nt), JSON(json), JSONP(jsonp)の形式に対応している (括弧内は拡張子を表す)。
- 拡張子を省略した場合は RDF/XML 形式でデータを取得可能。

「search_option」パラメータの値を指定することにより、取得するオントロジー要素を変更できる。表 11 に、クラスに関連するオントロジー要素を取得する際の「search_option」パラメータの値を示す。

以下では、クラスに関連するオントロジー要素を取得する URL の具体例および SPARQL クエリテンプレートを示す。また、日本語 Wikipedia オントロジーで主に利用している名前空間接頭辞と名前空間を表 12 に、プロパティを表 13 に示す。

表 11: クラスに関連するオントロジー要素を取得する際の「search_option」パラメータ値

クラスに関連するオントロジー要素	search_option パラメータ値
検索クラスを主語リソースとするステートメント	exact_match
検索クラスの兄弟クラス	sibling_classes
検索クラスの下位クラス	sub_classes
検索定義域クラスのプロパティ (検索クラスを定義域として持つプロパティ)	properties_of_domain_class
検索値域クラスのプロパティ (検索クラスを値域として持つプロパティ)	properties_of_range_class
検索クラスのインスタンス	instances_of_class
検索クラスからルートクラスまでのパス	path_to_root_class
検索クラスを目的語リソースとする (逆関係) ステートメント	inverse_statements

表 12: 日本語 Wikipedia オントロジーで主に利用している接頭辞と名前空間

接頭辞	名前空間
wikiont_class	http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/
wikiont_property	http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/property/
wikiont_instance	http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/instance/
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
owl	http://www.w3.org/2002/07/owl#
foaf	http://xmlns.com/foaf/0.1/

表 13: 日本語 Wikipedia オントロジーで主に利用しているプロパティ

プロパティ	意味
rdfs:label	クラスまたはプロパティのラベルを表す.
rdf:type	インスタンスのタイプ (属するクラス) を表す.
foaf:page	Wikipedia 記事ページまたはカテゴリページへのリンクを表す.
rdfs:subClassOf	クラスの上位・下位関係を表す.

(1) 検索クラスを主語リソースとするステートメントの取得 URL 例

慶應義塾大学の人物クラスを主語リソースとするステートメントを取得する URL
http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/慶應義塾大学の人物?search_option=exact_match

図 8の\$RESOURCE を検索クラス (ユーザが検索対象としたクラス) に置換することにより, 検索クラスを主語リソースとするステートメントのプロパティおよびプロパティ値が取得できる.

```
SELECT ?p ?o
WHERE {
  $RESOURCE ?p ?o .
}
```

図 8: 検索クラスのプロパティおよびプロパティ値を取得する SPARQL クエリテンプレート

(2) 検索クラスの兄弟クラスの取得 URL 例

慶應義塾大学の人物クラスの兄弟クラスを取得する URL
http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/慶應義塾大学の人物?search_option=sibling_classes

図 9の\$CLASS_NAME を検索クラスに置換することにより, 検索クラスの兄弟クラスが取得できる.

```

SELECT ?c
WHERE
{
  ?c rdfs:subClassOf ?supTargetCls.
  wikiont_class:$CLASS_NAME rdfs:subClassOf ?supTargetCls.
}

```

図 9: 検索クラスの兄弟クラスを取得する SPARQL クエリテンプレート

(3) 検索クラスの下位クラスを取得 URL 例

人物クラスの下位クラスを取得する URL

http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/人物?search_option=sub_classes

図 10の\$CLASS_NAME を検索クラスに置換することにより，検索クラスの下位クラスが取得できる。

```

SELECT ?c
WHERE
{
  ?c rdfs:subClassOf wikiont_class:$CLASS_NAME.
}

```

図 10: 検索クラスの下位クラスを取得する SPARQL クエリテンプレート

(4) 検索定義域クラスのプロパティの取得 URL 例

バレーボール選手クラスを定義域として持つプロパティを取得する URL

http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/バレーボール選手?search_option=properties_of_domain_class

図 11の\$CLASS_NAME を検索クラスに置換することにより，検索クラスを定義域として持つプロパティが取得できる。

```

SELECT ?p
WHERE
{
  ?p rdfs:domain wikiont_class:$CLASS_NAME.
}

```

図 11: 検索定義域クラスのプロパティを取得する SPARQL クエリテンプレート

(5) 検索値域クラスのプロパティの取得 URL 例

バレーボールクラスを値域として持つプロパティを取得する URL

http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/バレーボール選手?search_option=properties_of_range_class

図 12の\$CLASS_NAME を検索クラスに置換することにより，検索クラスを値域として持つプロパティが取得できる。

```

SELECT ?p
WHERE
{
  ?p rdfs:range wikiont_class:$CLASS_NAME.
}

```

図 12: 値域クラスのプロパティを取得する SPARQL クエリテンプレート

(6) 検索クラスのインスタンスの取得 URL 例

慶應義塾大学の人物クラスのインスタンスを取得する URL

`http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/慶應義塾大学の人物?search_option=instances_of_class&limit=50&offset=0`

図 13 の \$CLASS を検索クラスに置換することにより、検索クラスのインスタンスおよびそのラベルが取得できる。また、\$LIMIT は共通 URL パラメータの limit の値、\$OFFSET は共通 URL パラメータの start の値により、それぞれ置換される。

```
SELECT ?instance ?label
WHERE
{
  ?instance rdf:type $CLASS;
  rdfs:label ?label.
}
ORDER BY ?label
LIMIT $LIMIT
OFFSET $OFFSET
```

図 13: 検索クラスのインスタンスを取得する SPARQL クエリテンプレート

(7) 検索クラスからルートクラスまでのパスの取得 URL 例

慶應義塾大学の人物クラスからルートクラスまでのパスを取得する URL

`http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/慶應義塾大学の人物?search_option=path_to_root_class`

検索クラスからルートクラスまでのパスについては、Jena[15] を用いて、プログラム中で RDF トリプルを生成するため、SPARQL クエリテンプレートは利用していない。

(8) 検索クラスを目的語リソースとする（逆関係）ステートメントの取得 URL 例

慶應義塾大学の人物クラスを目的語リソースとするステートメントを取得。

`http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/慶應義塾大学の人物?search_option=inverse_statements`

図 14 の \$RESOURCE を検索クラスに置換することにより、検索クラスを目的語とするステートメントの主語リソースおよびそのプロパティを取得できる。

```
SELECT ?s ?p
WHERE {
  ?s ?p $RESOURCE .
}
```

図 14: 逆関係ステートメントを取得する SPARQL クエリテンプレート

5-3 クラス一覧の取得

以下の形式の URI にアクセスすると、クラス一覧を取得することができる。（現状では、JSON 形式のみに対応している。）

`[BASE_URI]class_list?start=[]&limit=[]&order_by=[name_asc, name_desc, instance_count_asc, instance_count_desc]&keyword=[クラス名]&search_option=[exact_match|any_match|starts_with|ends_with]`

order_by パラメータは、クラス一覧を取得する際のソート順序を指定するために用いられる。order_by パラメータの値には、表 14 に示す 4 つの値を指定することができる。

表 14: order_by パラメータ値の種類

order_by パラメータの値	説明
name_asc	クラス名 昇順
name_desc	クラス名 降順
instance_count_asc	クラスのインスタンス数 昇順
instance_count_desc	クラスのインスタンス数 降順

keyword パラメータに指定したクラス名またはクラス名の部分文字列と search_option より, 完全, 部分, 前方, 後方一致検索を行う。

クラス一覧をインスタンス数降順で取得する例を以下に示す。

http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class_list?search_option=exact_match&order_by=instance_count_desc&start=0&limit=50

5-4 クラスのインスタンス一覧の取得

以下の形式の URI にアクセスすると, class パラメータに指定したクラスのインスタンス一覧を取得することができる。(現状では, JSON 形式のみに対応している)

[BASE_URI]class_list?start=[]&limit=[]&class=[クラス名]

5-5 プロパティの取得

以下の形式の URI にアクセスすると, プロパティに関連するステートメントを取得することができる。

[BASE_URI]property/[page|data]/プロパティ.[rdf|n3|nt|json|jsonp]?search_option=[exact_match|domain_classes_of_property|range_classes_of_property|inverse_statements]

「search_option」パラメータの値を指定することにより, 取得するオントロジー要素を変更できる。表 15 に, プロパティに関連するオントロジー要素を取得する際の「search_option」パラメータの値を示す。

表 15: プロパティに関連するオントロジー要素を取得する際の「search_option」パラメータ値

プロパティに関連するオントロジー要素	search_option パラメータ値
検索プロパティを主語リソースとするステートメント	exact_match
検索プロパティの定義域	domain_classes_of_property
検索プロパティの値域	range_classes_of_property
検索プロパティを目的語リソースとする(逆関係)ステートメント	inverse_statements

以下では, プロパティに関連するオントロジー要素を取得する URL の具体例および SPARQL クエリテンプレートを示す。

(1) 検索プロパティを主語リソースとするステートメントの取得 URL 例

ジャンルプロパティを主語リソースとするステートメントを取得する URL

http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/property/data/ジャンル?search_option=exact_match

検索プロパティ (ユーザが検索対象としたプロパティ) を主語リソースとするステートメントを取得するための SPARQL テンプレートは, 図 8 に示したテンプレートと同様である。

(2) 検索プロパティの定義域の取得 URL 例

ジャンルプロパティの定義域を取得する URL

http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/property/data/ジャンル?search_option=domain_classes_of_property

図 15の\$PROPERTY_NAME を検索プロパティに置換することにより, 検索プロパティの定義域が取得できる.

```
SELECT ?c
WHERE
{
  wikiont_property:$PROPERTY_NAME rdfs:domain ?c.
}
```

図 15: 検索プロパティの定義域を取得する SPARQL クエリテンプレート

(3) 検索プロパティの値域の取得 URL 例

ジャンルプロパティの値域を取得する URL

http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/property/data/ジャンル
?search_option=range_classes_of_property

図 16の\$PROPERTY_NAME を検索プロパティに置換することにより, 検索プロパティの値域が取得できる.

```
SELECT ?c
WHERE
{
  wikiont_property:$PROPERTY_NAME rdfs:range ?c.
}
```

図 16: 検索プロパティの値域を取得する SPARQL クエリテンプレート

(4) 検索プロパティを目的語リソースとする (逆関係) ステートメントの取得 URL 例

現在のところ, 日本語 Wikipedia オントロジー内には, 検索プロパティを目的語リソースとするステートメントは存在していないと思われる.

5-6 プロパティ一覧の取得

以下の形式の URI にアクセスすると, プロパティ一覧を取得することができる. (現状では, JSON 形式のみに対応している)

[BASE_URI]property_list?start=[]&limit=[]&order_by=[name_asc, name_desc, instance_count_asc, instance_count_desc]&keyword=[プロパティ名]&search_option=[exact_match|any_match|starts_with|ends_with]

order_by パラメータは, プロパティ一覧を取得する際のソート順序を指定するために用いられる. order_by パラメータの値には, 表 16に示す 4つの値を指定することができる.

表 16: order_by パラメータ値の種類

order_by パラメータの値	説明
name_asc	プロパティ名 昇順
name_desc	プロパティ名 降順
instance_count_asc	プロパティをもつインスタンス数 昇順
instance_count_desc	プロパティをもつインスタンス数 降順

keyword パラメータに指定したプロパティ名またはプロパティ名の部分文字列と search_option より, 完全, 部分, 前方, 後方一致検索を行う.

プロパティ一覧をプロパティを持つインスタンス数降順で取得する例を以下に示す.

http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/property_list?search_option=exact_mate
h&order_by=instance_count_desc&start=0&limit=50

5-7 プロパティを持つインスタンス一覧の取得

以下の形式の URI にアクセスすると、property パラメータに指定したプロパティを持つインスタンス一覧を取得することができる。(現状では、JSON 形式のみに対応している)

[BASE_URI]property_list?start=[]&limit=[]&property=[プロパティ名]

5-8 インスタンスの取得

以下の形式の URI にアクセスすると、インスタンスに関連するステートメントを取得することができる。

[BASE_URI]instance/[page|data]/インスタンス名.[rdf|n3|nt|json|jsonp]?
search_option=[exact_match|types_of_instance|inverse_statements]

「search_option」パラメータの値を指定することにより、取得するオントロジー要素を変更できる。表 17 に、インスタンスに関連するオントロジー要素を取得する際の「search_option」パラメータの値を示す。

表 17: インスタンスに関連するオントロジー要素を取得する際の「search_option」パラメータ値

インスタンスに関連するオントロジー要素	search_option パラメータ値
検索インスタンスを主語リソースとするステートメント	exact_match
検索インスタンスのタイプ	types_of_instance
検索インスタンスを目的語リソースとする(逆関係)ステートメント	inverse_statements

以下では、インスタンスに関連するオントロジー要素を取得する URL の具体例および SPARQL クエリテンプレートを示す。

(1) 検索インスタンスを主語リソースとするステートメントの取得 URL 例

福澤諭吉を主語リソースとするステートメントを取得する URL

http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/instance/data/福澤諭吉?search_option=exact_match

検索インスタンス(ユーザが検索対象としたインスタンス)を主語リソースとするステートメントを取得するための SPARQL クエリテンプレートは、図 8 に示したクエリテンプレートと同様である。

(2) 検索インスタンスのタイプの取得 URL 例

福澤諭吉のタイプを取得する URL

http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/instance/data/福澤諭吉?search_option=types_of_instance

図 17 の \$INSTANCE を検索インスタンスに置換することにより、検索インスタンスのタイプおよびラベルが取得できる。取得したタイプリストはラベル名でソートする。

```
SELECT ?type ?label
WHERE
{
  $INSTANCE rdf:type ?type.
  ?type rdfs:label ?label.
}
ORDER BY ?label
```

図 17: 検索インスタンスのタイプを取得する SPARQL クエリテンプレート

(3) 逆関係のステートメントの取得 URL 例

福澤諭吉を目的語リソースとするステートメントを取得する URL

http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/instance/data/福澤諭吉?search_option=inverse_statements

検索インスタンスを目的語とするステートメントを取得する SPARQL クエリテンプレートは、図 14 に示したクエリテンプレートと同様である。

5-9 インスタンス一覧の取得

以下の形式の URI にアクセスすると、インスタンス一覧を取得することができる。(現状では、JSON 形式のみに対応している)

```
[BASE_URI]instance_list?start=[]&limit=[]&order_by=[name_asc, name_desc]&keyword=[インスタンス名]&search_option=[exact_match|any_match|starts_with|ends_with]
```

order_by パラメータは、インスタンス一覧を取得する際のソート順序を指定するために用いられる。order_by パラメータの値には、表 18 に示す 2 つの値を指定することができる。

表 18: order_by パラメータ値の種類

order_by パラメータの値	説明
name_asc	インスタンス名 昇順
name_desc	インスタンス名 降順

keyword パラメータに指定したインスタンス名またはインスタンス名の部分文字列と search_option より、完全、部分、前方、後方一致検索を行う。

5-10 複数のタイプ(rdf:type)を持つインスタンスの取得

以下の URI にアクセスすると、複数のタイプ(rdf:type)を持つインスタンス(複数のクラスに属するインスタンス)を取得できる。type=[クラス名]を URL パラメータとして複数指定することで、インスタンスをタイプで絞り込むことができるようになっている。以下の例では、「慶應義塾大学の人物」クラスと「神奈川県出身の人物」クラスの両方に属するインスタンス一覧を取得する。

RDF/XML データ

[http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/q.rdf?type=\[クラス名\]&type=\[クラス名\]...](http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/q.rdf?type=[クラス名]&type=[クラス名]...)

例: http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/q.rdf?type=慶應義塾大学の人物&type=神奈川県出身の人物

JSON データ

[http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/q.json?type=\[クラス名\]&type=\[クラス名\]...](http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/q.json?type=[クラス名]&type=[クラス名]...)

例: http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/q.json?type=慶應義塾大学の人物&type=神奈川県出身の人物

5-11 推論モデルを用いたクラスおよびインスタンスの取得

RDFS 推論ルールにより導出されたステートメントの集合に対して、クラスおよびインスタンスを取得することができる。

例として、「アナウンサー」クラスのインスタンス取得について説明する。JWO には「アナウンサー」クラスのインスタンスが直接定義されていないため、以下の URI にアクセスしても「アナウンサー」クラスのインスタンスを取得することはできない。

```
http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/アナウンサー.rdf?search_option=instances_of_class&limit=50&offset=0
```

しかしながら、「アナウンサー」クラスの下位クラスには「日本のアナウンサー」クラスなどのインスタンスが定義されたクラスが存在し、RDFS 推論ルールを用いることで、これら下位クラスで定義されたインスタンスも「アナウンサー」クラスのインスタンスであることを導出できる。上記の「アナウンサー」クラスを表す URL に「inference_type=rdfs」オプションを追加することで、RDFS 推論ルールを用いて導き出したステートメントを取得することができ、結果として「アナウンサー」クラスのインスタンスを取得することができる。

```
http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/data/アナウンサー.rdf?inference_type=rdfs&search_option=instances_of_class&limit=50&offset=0
```

5-12 リソースの URI とデータの URI の区別について

クラス、プロパティ、インスタンスを識別するための URI (リソースの URI) と RDF/XML や JSON 形式などのデータを表す URI は区別している。現状では、以下の形式でリソースの URI を定義している。

クラスの URI:

```
http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/[クラス名]
```

プロパティの URI:

```
http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/property/[プロパティ名]
```

インスタンスの URI:

```
http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/instance/[インスタンス名]
```

現在、Request Header の Accept の値によって、Web ブラウザからのアクセスかエージェント (プログラム) からのアクセスかを識別し、人間向けの HTML ページまたはエージェント向けの RDF/XML や JSON データを返すようにしている。

例えば、[http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/\[クラス名\]](http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/[クラス名]) に Web ブラウザからアクセスすると、[http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/page/\[クラス名\].html](http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/page/[クラス名].html) にリダイレクトされる。同様に、

[http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/instance/\[インスタンス名\]](http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/instance/[インスタンス名]) に Web ブラウザからアクセスすると、

[http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/instance/page/\[インスタンス名\].html](http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/instance/page/[インスタンス名].html) にリダイレクトされる。アクセス時の Request Header の Accept の値に "text/html" が含まれている場合には HTML ページにリダイレクトされるようになっている。

また、Request Header の Accept の値に "application/rdf+xml" が含まれている場合には RDF/XML データに、"application/json" が含まれている場合には JSON (テーブル用) データにリダイレクトされるようになっている⁷。

☒ 18 に、Java 言語を用いて HTML、RDF/XML、JSON 形式のデータを取得するためのサンプルプログラムを示す。

⁷厳密には、<http://hpcs01.comp.ae.keio.ac.jp/> で JWO Web サービスのプログラムは動作しているため、http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/ からリダイレクトにより http://hpcs01.comp.ae.keio.ac.jp/wikipedia_ontology/ にページ遷移している。将来的に JWO Web サービスのプログラムの場所は変更になる可能性がある。また、リソースの URI についても、試験的に設定しているため、将来変更になる可能性がある。

```

public static void main(String[] args) throws Exception {
    String keyword = "慶應義塾大学の人物";
    //String keyword = "福沢諭吉";
    String ns = "http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/class/";
    //String ns = "http://www.yamaguti.comp.ae.keio.ac.jp/wikipedia_ontology/instance/";
    keyword = URLEncoder.encode(keyword, "UTF-8");
    URL url = new URL(ns + keyword);
    URLConnection urlConnection = url.openConnection();
    //urlConnection.setRequestProperty("Accept", "text/html");
    urlConnection.setRequestProperty("Accept", "application/rdf+xml");
    //urlConnection.setRequestProperty("Accept", "application/json");
    BufferedReader reader = new BufferedReader(new InputStreamReader(urlConnection.getInputStream(), "UTF-8"));
    while (reader.ready()) {
        System.out.println(reader.readLine());
    }
}

```

図 18: Java 言語を用いて HTML, RDF/XML, JSON 形式のデータを取得するためのサンプルプログラム

5-13 データ構造

(1) JSON (テーブル形式) のデータ構造

```

{"statement": [{"subject": "[主語 URI]", "predicate": "[述語 URI]", "object": "[目的語 URI]"},
               {"subject": "[主語 URI]", "predicate": "[述語 URI]", "object": "[目的語 URI]"},
               ...], "numberOfStatements": [ステートメント数]}

```

図 19: JSON (テーブル形式) のデータ構造

※ numberOfStatements 属性の値となるステートメント数は, limit と start を指定した時に返されるデータのステートメント数とは異なっている. 常に, あるクラスまたはインスタンスに関連するすべてのステートメント数を返すようになっている. 日本語 Wikipedia オントロジー検索システムにおける検索パネルで表示ステートメント数を指定した場合にも, 全体としてのステートメント数を把握できるようにするために numberOfStatements 属性は用意されている.

(2) JSON (ツリー形式) のデータ構造

```

[{"id": "[クラス URI]", "text": "[クラスラベル] ([インスタンス数]) ",
  "children": [{"id": "[サブクラス URI] または [インスタンス URI]",
               "text": "[クラスラベル] または [インスタンスラベル]", "leaf": true, "iconCls": "icon-instance"}],
  "leaf": false, "iconCls": "icon-class"}, ...]

```

図 20: JSON (ツリー形式) のデータ構造

children 属性の値として, クラスまたはインスタンスの配列を指定することで, 階層構造を定義している. leaf 属性は, そのノードが葉ノードの場合に true となり, そうでない場合には false となる. iconCls には, ノードを表示するために必要な CSS クラスを指定する. 日本語 Wikipedia オントロジー検索システムでは, クラスを表示するための CSS クラスとして icon-class クラスを, インスタンスを表示するための CSS クラスとして, icon-instance クラスを用意している. スタイルシートは以下の URL から取得できる.

http://hpcs01.comp.ae.keio.ac.jp/wikipedia_ontology/my_resources/mystyle.css

5-14 SPARQL による検索

以下の URL に q パラメータの値として UTF-8 でエンコードした SPARQL クエリを与えると, SPARQL による検索結果を得ることができる.

http://hpcs01.comp.ae.keio.ac.jp/wikipedia_ontology/sparql

q 以外に指定可能な URL パラメータを表 19 に示す.

表 19: SPARQL クエリによる検索で指定可能な URL パラメータ

URL パラメータ	説明
inference_type	推論器の種類を指定する. 現状では「rdfs」のみ指定可.
output	「xml」, 「txt」, 「json」を指定できる.
version	「2010_02_09」と「2010_11_14」の2種類のみ指定可.

例として、「福澤諭吉」をラベルとして持つインスタンスとそのタイプを XML 形式で得るための URL を以下に示す.

```
http://hpcs01.comp.ae.keio.ac.jp/wikipedia_ontology/sparql?q=PREFIX+rdf:++3Chttp://www.w3.org/1999/02/22-rdf-syntax-ns%23%3E%0APREFIX++rdfs:++3Chttp://www.w3.org/2000/01/rdf-schema%23%3E%0A%0ASELECT+%3Finstance+%3Fclass%0AWHERE%0A{%0A+++3Finstance+rdf:type+%3Fclass;%0A+++rdfs:label+%22%E7%A6%8F%E6%BE%A4%E8%AB%AD%E5%90%89%22.%0A}&output=xml&inference_type=none
```

6 日本語 Wikipedia オントロジー検索インタフェース

6-1 JWO 検索インタフェースの概要

JWO 検索インタフェースは、ステートメントウィンドウ、クラス一覧ウィンドウ、プロパティ一覧ウィンドウ、インスタンス一覧ウィンドウ、SPARQL ウィンドウ、ブックマークウィンドウ、履歴ウィンドウ、ソースコードウィンドウから構成される。以下では、各ウィンドウについて説明する。

6-2 ステートメントウィンドウ

ステートメントウィンドウでは、クラス、プロパティ、インスタンスに関するステートメントを JWO から検索することができる。図 21 にステートメントウィンドウのスクリーンショットを示す。「キーワード」テキストフィールドに、クラス、プロパティ、インスタンスのラベルまたは URI におけるローカル名を入力し、検索対象（クラス、プロパティ、インスタンス）を選択し、検索ボタンを押すと、ウィンドウ下部に関連するステートメント（主語、述語、目的語）のリストが表示される。主語、述語、目的語がリソースの場合には、リンクが張られ、それらを選択すると、選択したリソースを主語リソースとするステートメントリストが表示される。クラス、プロパティ、インスタンスごとに指定できる検索オプションは異なり、以下に示すオプションを指定することができる。

(1) クラスの検索オプション

- 完全一致
- 兄弟クラス
- 下位クラス
- 定義域クラスのプロパティ
- 値域クラスのプロパティ
- クラスのインスタンス
- ルートクラスまでのパス
- 逆関係のステートメント

(2) プロパティの検索オプション

- 完全一致
- プロパティの定義域
- プロパティの値域
- 逆関係のステートメント

(3) インスタンスの検索オプション

- 完全一致
- インスタンスのタイプ
- 逆関係のステートメント

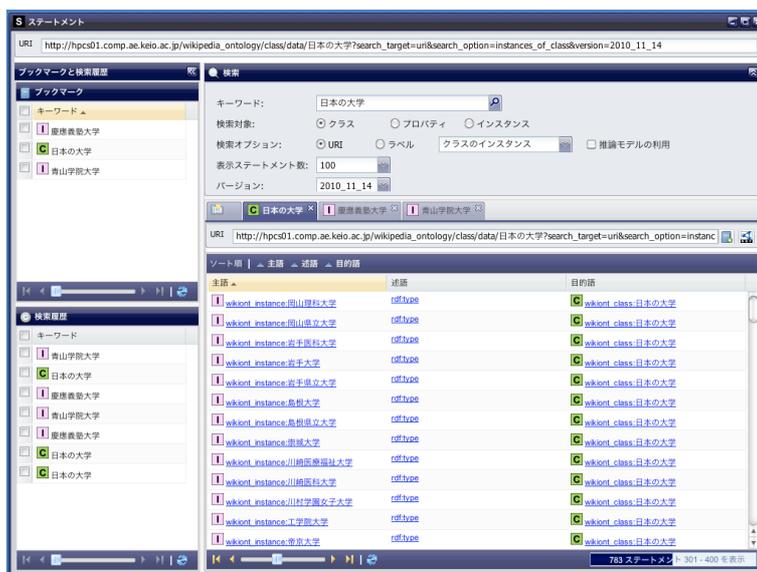


図 21: ステートメントウィンドウ

6-3 クラス一覧ウィンドウ

クラス一覧ウィンドウでは、JWO に含まれる全クラスを、名前またはクラスのインスタンス数の昇順または降順でソートして、閲覧することができる。また、クラス名 (URI のローカル名) の完全一致、部分一致、前方一致、後方一致でクラスを検索することができる。図 22にクラス一覧ウィンドウのスクリーンショットを示す。図 22画面左側のクラスリストの中からクラスを選択すると、選択クラスを定義域として持つプロパティ、選択クラスを値域として持つプロパティ、選択クラスのインスタンスが図 22画面右側に表示される。さらに、表示されたプロパティまたはインスタンスを選択すると、ステートメントウィンドウに選択したプロパティまたはインスタンスを主語リソースとするステートメントリストが表示される。

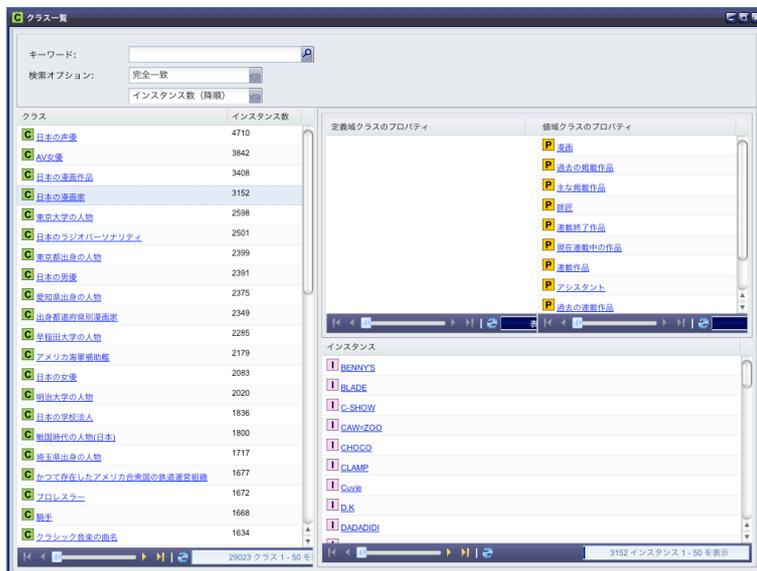


図 22: クラス一覧ウィンドウ

6-4 プロパティ一覧ウィンドウ

プロパティ一覧ウィンドウでは、JWO に含まれる全プロパティを、名前またはそのプロパティを利用しているインスタンス数の昇順または降順でソートして、閲覧することができる。また、プロパティ名 (URI のローカル名) の完全一致、部分一致、前方一致、後方一致でプロパティを検索することができる。図 23にプロパティ一覧ウィンドウのスクリーンショットを示す。図 23画面左側のプロパティリストの中からプロパティを選択すると、選択プロパティの定義域および値域、選択プロパティを持つインスタンスが図 23画面右側

に表示される。さらに、表示されたクラスまたはインスタンスを選択すると、ステートメントウィンドウに選択したクラスまたはインスタンスを主語リソースとするステートメントリストが表示される。

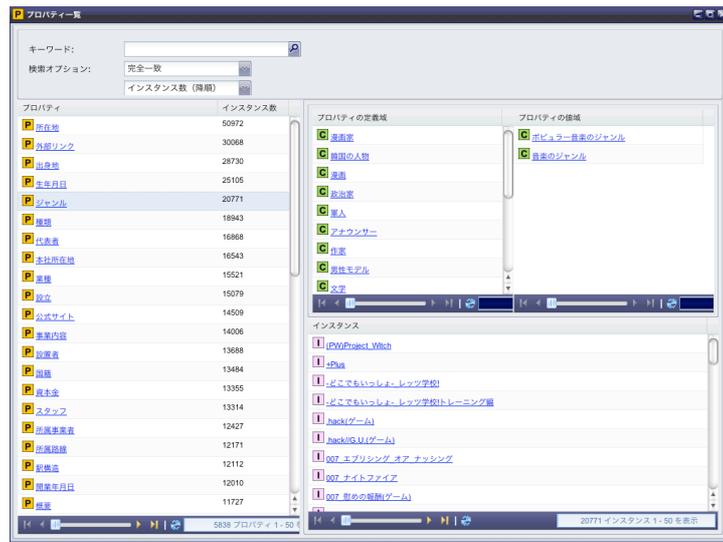


図 23: プロパティ一覧ウィンドウ

6-5 インスタンス一覧ウィンドウ

インスタンス一覧ウィンドウでは、JWO に含まれる全インスタンスを、名前の昇順または降順でソートして、閲覧することができる。また、インスタンス名 (URI のローカル名) の完全一致、部分一致、前方一致、後方一致でインスタンスを検索することができる。図 24にインスタンス一覧ウィンドウのスクリーンショットを示す。図 24画面左側のインスタンスリストの中からインスタンスを選択すると、選択インスタンスのタイプが図 24画面右側に表示される。さらに、表示されたクラス (選択インスタンスのタイプ) を選択すると、ステートメントウィンドウに選択したクラスを主語リソースとするステートメントリストが表示される。

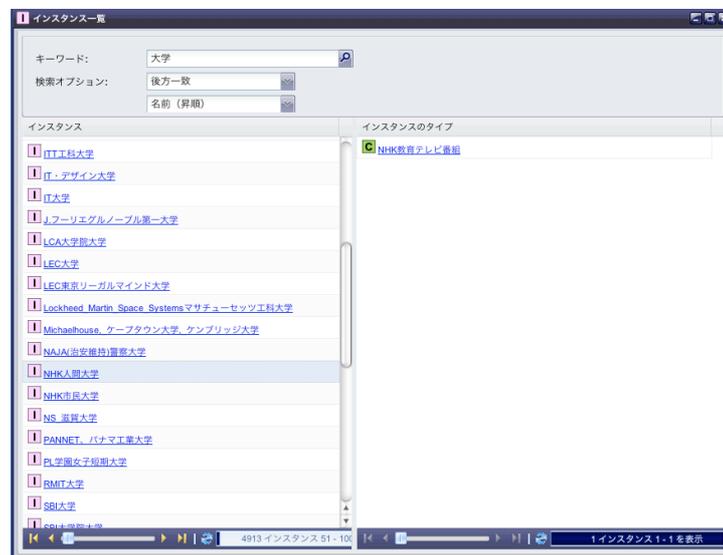


図 24: インスタンス一覧ウィンドウ

6-6 SPARQL ウィンドウ

SPARQL ウィンドウでは、ユーザが SPARQL クエリをその説明と共に登録し、クエリ結果を表示することができる。図 25に SPARQL ウィンドウのスクリーンショットを示す。図 25右側下部の「登録クエリ」タブでは、「織田信長はいつ生まれた？」という問いに対応する SPARQL クエリを登録し、図 25左側下部の「クエリ結果」タブでは、その SPARQL クエリ結果を表示している。クエリ結果に JWO のクラス、プロパティ、インス

タンスが含まれている場合には、リンクが張られ、それらを選択するとステートメントウィンドウに、選択したクラス、プロパティ、インスタンスを主語リソースとするステートメントリストが表示される。

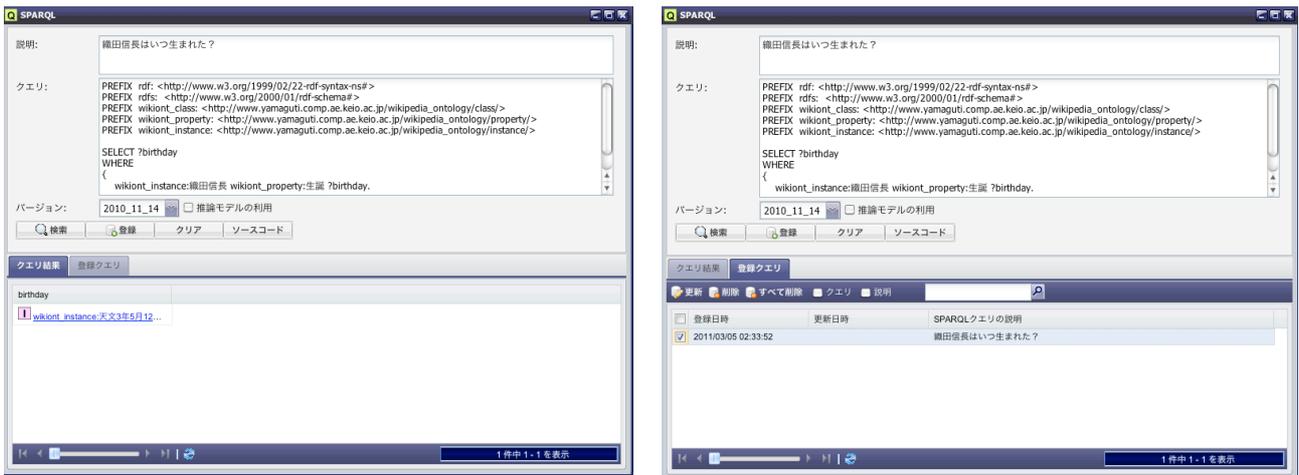


図 25: SPARQL ウィンドウ

6-7 ブックマークウィンドウ

ブックマークウィンドウは、ステートメントウィンドウで検索した結果を保存しておき、後から参照することを可能にする。ステートメントウィンドウ下部の各タブ領域上部の URI テキストフィールド右側にブックマークを追加するためのボタンがある。このボタンを選択すると、そのタブに表示されているステートメントリストを検索する際に指定した検索オプションなどをブックマークとして保存できる。図 26にブックマークウィンドウのスクリーンショットを示す。

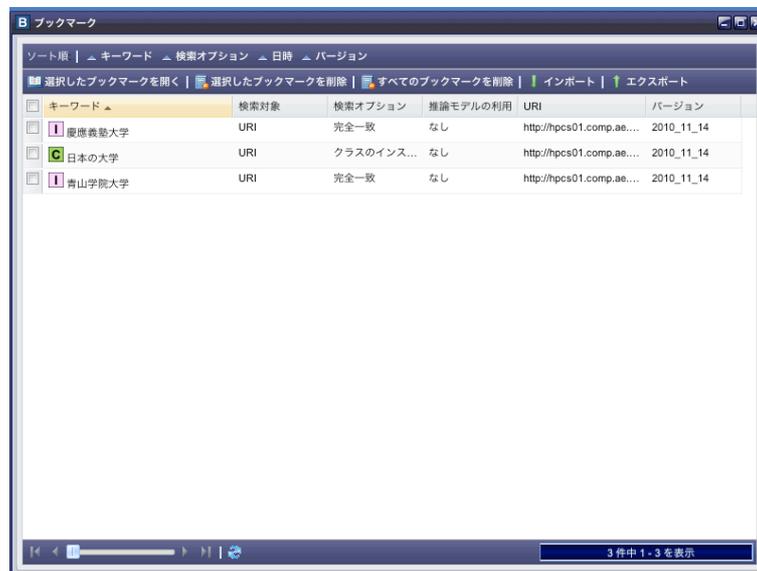


図 26: ブックマークウィンドウ

6-8 履歴ウィンドウ

履歴ウィンドウは、ステートメントウィンドウで検索した履歴を保存しておき、後から参照することを可能にする。また、選択した履歴をブックマークに追加することができる。図 27に履歴ウィンドウのスクリーンショットを示す。

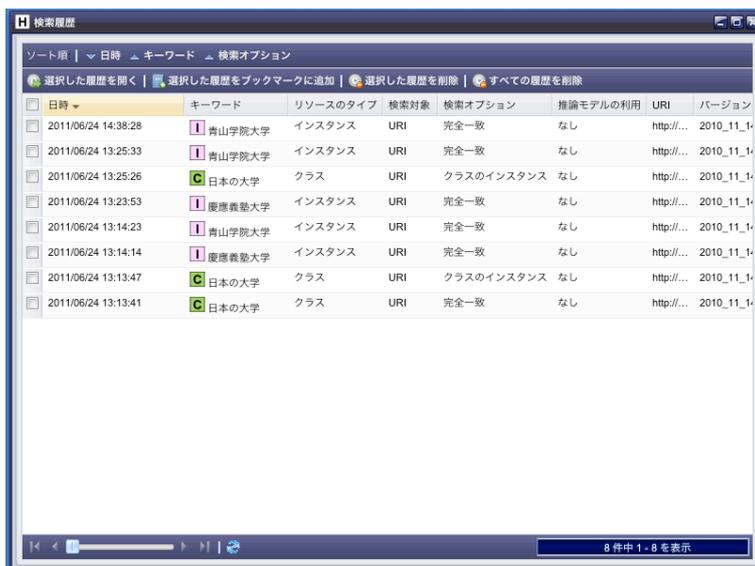


図 27: 履歴ウィンドウ

6-9 ソースコードウィンドウ

ソースコードウィンドウには、ステートメントウィンドウで検索した結果を RDF/XML 形式で表示することができます。ステートメントウィンドウ下部の各タブ領域上部の URI テキストフィールド右側にソースコードを表示するためのボタンがある。このボタンを選択すると、そのタブに表示されているステートメントリストが RDF/XML 形式でソースコードウィンドウに表示される。図 28にソースコードウィンドウのスクリーンショットを示す。

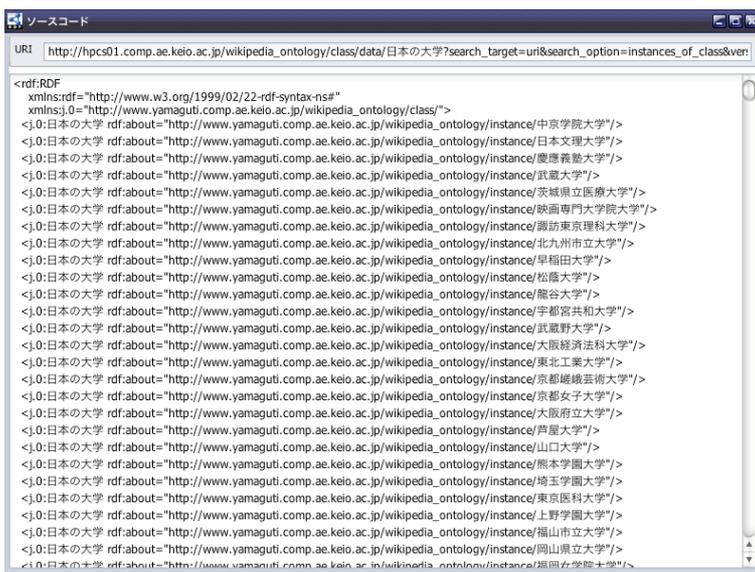


図 28: ソースコードウィンドウ

7 まとめ

本研究では、ソフトウェアが意味理解可能な Web (セマンティック Web) を実現するための基盤として日本語 Wikipedia から半自動構築したオントロジー (日本語 Wikipedia オントロジー) に着目した。日本語 Wikipedia オントロジーにおけるクラス階層の上位層には、抽象的なクラスが不足している問題があることから、日本語 Wikipedia オントロジーと日本語 WordNet の統合を試みた。日本語 WordNet との統合には未だ多くの課題が存在するが、日本語 Wikipedia オントロジーにおけるクラスに同義語を定義し、それらを活用して、文字列に基づくオントロジーアライメント手法を適用し、その限界と今後の研究の方向性を明らかにすることができた。

また、日本語 Wikipedia オントロジーを様々なソフトウェアから活用できるように Web サービスを実装し

た。本 Web サービスにより、日本語 Wikipedia オントロジーにおけるクラス、プロパティ、インスタンスに関連するステートメント、クラスのインスタンス、兄弟クラス、下位クラス、プロパティの定義域および値域、インスタンスのタイプなど、様々な関係をセマンティック Web アプリケーションが Web API から取得可能になった。さらに、本 Web サービスをセマンティック Web アプリケーション開発者が容易に検索できるように、検索インタフェースも実装した。本 Web サービスは、Web サービスにおける入出力データの意味づけ（セマンティック Web サービス）、ソーシャルタギングにおけるタグへの意味づけ（セマンティックソーシャルタギング）、Web ページへのアノテーション（セマンティックアノテーション）、領域オントロジー構築支援などへの活用が期待される。

今後は、開発した日本語 Wikipedia オントロジー Web サービスおよび検索インタフェースを活用して、セマンティック Web アプリケーションを研究開発していく予定である。また、より高精度に日本語 Wikipedia オントロジーと日本語 WordNet を統合するための手法やツールの研究開発も行っていきたい。

なお、本研究で開発した日本語 Wikipedia オントロジー Web サービスおよび検索インタフェースは、[21]にて、試験的に公開している。今後、SourceForge.jp などのオープンソース・ソフトウェアを公開する Web サイトなどで、広く研究開発成果を公開していく予定である。

【参考文献】

- [1] 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 論文特集「2009 年度全国大会近未来チャレンジ」 Vol. 25 No.5 pp.623-636 (2010)
- [2] Soren Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, Zachary Ives: DBpedia: A Nucleus for a Web of Open Data, 6th International Semantic Web Conference, Vol. 4825, pp. 722-735 (2007)
- [3] Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum: Yago: a core of semantic knowledge, Proceedings of the 16th international conference on World Wide Web, pp. 697-706 (2007)
- [4] 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)
- [5] 日本語 WordNet: <http://nlpwww.nict.go.jp/wn-ja/>
- [6] 市瀬 龍太郎, オントロジーマッピングに有効な特徴の抽出, 第 22 回人工知能学会全国大会, 2E1-1, 2008
- [7] Wikipedia ダンプデータ: <http://download.wikimedia.org/jawiki/>
- [8] SimMetrics: <http://www.dcs.shef.ac.uk/~sam/stringmetrics.html>
- [9] Euzenat, J., Shvaiko, P., Ontology Matching. Springer. Berlin; Heidelberg (2007)
- [10] 国立国会図書館件名標目表 (NDLSH): <http://id.ndl.go.jp/auth/ndlsh/>
- [11] NDLSH の RDF モデル: <http://id.ndl.go.jp/auth/docs/rdf>
- [12] SQLite JDBC Driver: <http://www.xerial.org/trac/Xerial/wiki/SQLiteJDBC>
- [13] ExtJS ライブラリ: <http://extjs.co.jp/products/js/>
- [14] Apache Wicket: <http://wicket.apache.org/>
- [15] Jena Semantic Web Framework for Java: <http://jena.sourceforge.net/>
- [16] ActiveObjects: <https://activeobjects.dev.java.net/>
- [17] Memcached Java Client: <http://github.com/gwhalin/Memcached-Java-Client>
- [18] TDB - A SPARQL Database for Jena: <http://openjena.org/TDB/>
- [19] H2 DB: <http://www.h2database.com/html/main.html>
- [20] Jetty: <http://jetty.codehaus.org/jetty/>
- [21] 日本語 Wikipedia オントロジー検索インタフェース & Web APIs : http://hpcs01.comp.ae.keio.ac.jp/wikipedia_ontology/

〈 発 表 資 料 〉

題 名	掲載誌・学会名等	発表年月
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	2011 年 6 月
Intelligent Humanoid Robot with Japanese Wikipedia Ontology and Robot Action Ontology	HRI2011 (6th International Conference on Human-Robot Interaction)	2011 年 3 月
日本語 Wikipedia オントロジーと動作オン トロジーを利用した人型ロボットの対話と 動作の融合	HAI シンポジウム 2010 (Human-Agent Interaction Symposium 2010)	2010 年 12 月
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	2010 年 11 月
Learning a Large Scale of Ontology from Japanese Wikipedia	2010IEEE/WIC/ACM International Conference on Web Intelligence	2010 年 9 月
Human Robot Interaction Based on Wikipedia Ontology and Robot Action Ontology	International Joint Conference on Knowledge-Based Software Engineering 2010	2010 年 8 月
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	2010 年 6 月