2J3-NFC2-5

日本語Wikipediaからプロパティを備えたオントロジーの構築

Building up Ontologies with Many Properties from Japanese Wikipedia

玉川 奨[*1] Susumu Tamagawa　関本 有佳[*1] Yuka Sekimoto　森田 武史[*2] Takeshi Morita　山口 高平[*1] Takahira Yamaguchi

[*1]慶應義塾大学 Keio University　[*2]青山学院大学 Aoyama Gakuin University

Here is discussed how to extract property definitions automatically from Japanese Wikipedia which are name, type(owl:ObjectProperty, owl:DatatypeProperty, owl:SymmetricProperty, owl:TransitiveProperty, owl:FunctionalProperty, owl:InverseFunctionalProperty), domain(rdfs:domain), range(rdfs:range) and hyponymy Relationship(rdfs:subPropertyOf).

1. はじめに

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 や日本語語彙大系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，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，ツールなどの研究開発に取り組んできた．特に，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く，情報資源として注目されている．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．

我々はこれまで，日本語 Wikipedia における様々なリソース (カテゴリツリー，一覧記事，リダイレクトリンク，Infobox) から，概念および概念間の関係 (Is-a 関係，クラス－インスタンス関係，プロパティ定義域，プロパティ値域，同義語，インスタンス間関係) を抽出し，高精度かつ大規模な汎用オントロジー (以下，日本語 Wikipedia オントロジー) を学習する手法を提案してきた [玉川 10]．しかし，プロパティ定義において，いくつかの課題が残っていた．そこで本稿では，これまでの手法に加えて，プロパティ名とトリプル数の増加を図るために，記事のリスト構造のスクレイピングを行う．次に，過去に提案した Infobox からのプロパティ抽出法 [玉川 10] により抽出したプロパティ名と今回新たに抽出したプロパティ名を照合し，プロパティ間の上位下位関係を抽出する．抽出したトリプルから，プロパティ関係として対称関係，推移関係，関数関係，逆関数関係の推定を試みる．さらに，プロパティ定義域の洗練をするために，プロパティ定義域の親クラスと兄弟クラスを参照し，定義域のリフトアップを行う．

2. 関連研究

DBpedia[Auer 07] は，Wikipedia の半構造情報を RDF に変換することによって，大規模なデータベースを構築している．リソースとしては主に，英語 Wikipedia の Infobox や外部リンク，所属カテゴリといった半構造情報を利用している．これらは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．

YAGO2[Johannes 10] 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，Wikipedia と GeoNames から自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．

2.1 Wikipedia オントロジーのプロパティ定義

日本語 Wikipedia オントロジーのプロパティ定義は以下の関係とタイプから構築される．本稿では以下の関係とタイプの抽出に加え，プロパティ定義域のリフトアップを行うことで，定義域の洗練を行う．

1. プロパティ名
2. プロパティ定義域 (rdfs:domain)
3. プロパティ値域 (rdfs:range)
4. プロパティ上位下位関係 (rdfs:subPropertyOf)
5. プロパティタイプ
   (a) オブジェクト型プロパティ (owl:ObjectProperty)
   (b) データ型プロパティ (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:InverseFunctional Property)

連絡先: 玉川 奨，山口高平，慶應義塾大学理工学研究科
{s_tamagawa,yamaguti}@ae.keio.ac.jp

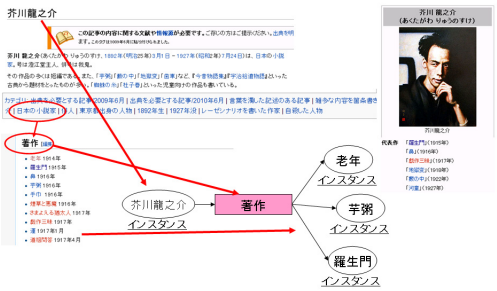

図 1: 記事のリスト構造からのプロパティ名抽出の一例

2.2 記事のリスト構造からのプロパティ名とトリプル抽出

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．図 1 の例では，“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 Infobox トリプルからの抽出法でも抽出可能だが，“老年” や “芋粥” は，記事のリスト構造からの抽出により抽出可能な値であり，本手法は，Infobox トリプルからは抽出できないプロパティ名だけでなく，トリプルを抽出することもできる．

2.3 プロパティ上位下位関係抽出

Infobox が記事の概要を表しているという Wikipedia の特徴に着目し，Infobox から抽出したプロパティ名と 2.2 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 2 の例では，主語である “芥川龍之介” はリスト構造から抽出した “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 Infobox から抽出した “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．

2.4 プロパティタイプの推定

Infobox より抽出したプロパティは，オブジェクト型とデータ型の分類がなされていた．本手法ではこれに加え，対称関係プロパティ，推移関係プロパティ，関数関係プロパティ，逆関数関係プロパティの推定を行う．

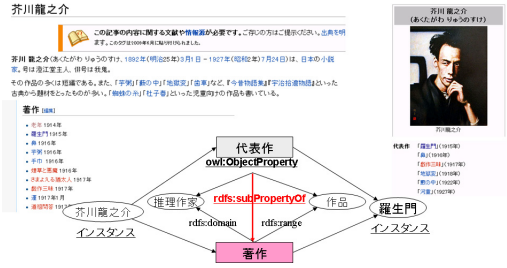

図 2: プロパティ上位下位関係の抽出の一例

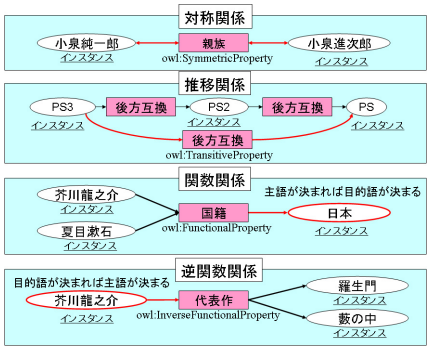

図 3: プロパティタイプの抽出の一例

まず，対称関係プロパティ候補の抽出を行う．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．次に，推移関係プロパティの候抽出を行う．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．同様に関数関係プロパティと逆関数関係プロパティの候補抽出を行う．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候補として抽出する．さらに，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候補として抽出する．最後に，プロパティ $P(n)$ の全トリプル数 A と候補として抽出したトリプル数 α の割合から各関係プロパティの推定を行う．

2.5 プロパティ定義域の洗練

これまでに抽出したプロパティの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った．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

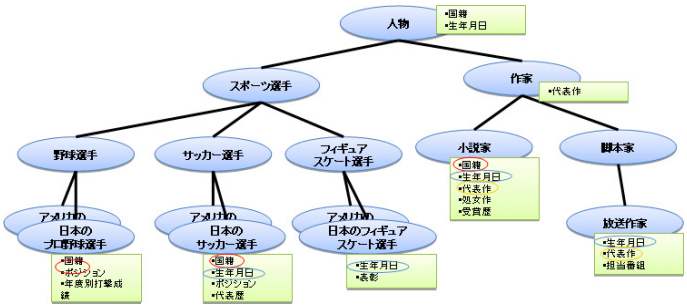

図 4: プロパティ定義域洗練の一例

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，プロパティ定義域を洗練するために，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，定義域のリフトアップが可能になり，先の問題を解消する．図 4 がプロパティ定義域洗練の一例である．

3. 実験結果と考察

実験は 2010 年 11 月時点の Wikipedia ダンプデータ (jawiki-latest-pages-articles.xml)*2 をダウンロードし，データベースは MySQL，実装言語は Java 言語を用いて行った．

3.1 記事のリスト構造からのプロパティ名とトリプルの抽出結果と考察

Wikipedia のダンプデータから 2.2 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．トリプルが多かったプロパティは “スタッフ”，“キャスト”，“テレビドラマ” などテレビ番組に関するものであった．

2,919,470 のトリプルから 1,000 個の標本を抽出し，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，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．例えば，“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．

*2 Wikipedia ダンプデータ: http://download.wikimedia.org/jawiki/

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

Infobox からの抽出法では 1,962,411 のトリプルと 7,137 のプロパティ名が抽出されており，今回の手法と合わせ，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，Infobox からの手法単体での正解率 95.2 ± 1.33%と比較すると，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．

3.2 プロパティ上位下位関係の抽出結果と考察

2.3 で提案した手法により，1,387 の上位下位関係の候補を抽出した．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，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．その結果，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%となっていた．正答例として，“キャスト—出演者” や “スタッフ—監督” 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く，“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．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．

3.3 プロパティタイプの抽出結果と考察

2.4 で提案した手法により，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った．415 の対称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した結果，正答率は 45.1%であった．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” のような対称関係プロパティも抽出されている．

次に，推移関係プロパティの推定を行った．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

プロパティが存在しないという結果となった背景の 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，2.5 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．

次に，関数関係・逆関数関係プロパティの推定を行った．関数関係プロパティ候補として 2,267，逆関数関係プロパティ候補として 3,670 のプロパティを得た．正答率は関数関係プロパティが 54.3%，逆関数関係が 22.4%であった．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，これらは本来データ型となるべきプロパティであり，Infobox からの抽出法でデータ型かオブジェクト型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．逆関数関係が非常に低い正答率となっている理由として，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が言える．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，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”，“収録作品タイトル” などである．

3.4 プロパティ定義域の洗練結果と考察

2.5 で提案した手法により，プロパティ定義域の関係数を 156,674 から 78,616 にまで削減することができた．これにより，各リーフクラス (例えば “埼玉県の動物園” と “東京都の動物園”) に同様のプロパティが定義されていたのが，上位のクラス (“動物園” クラス) にリフトされたことを示している．特に，日本語 Wikipedia オントロジーのハイブランチ構造により，“○○県出身の人物” というクラスがそれぞれ 250 程度のプロパティを所有しており，洗練前は利用しているプロパティの合計は 5,819 もあったが，今回の手法により，1,765 にまで削減することができた．しかし，日本語 Wikipedia オントロジーは中間概念が不足しているため，未だ複数箇所に登場するプロパティが多く存在しており，今後の課題といえる．

表 1: 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
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)	78,616	94.8 ± 1.22%
プロパティ値域 (rdfs:range)	49,262	90.4 ± 1.81%
クラスーインスタンス関係	14,053	88.3 ± 1.92%
Is-a 関係	35,946	92.1 ± 1.65%
プロパティ上位下位	1,387	57.5%

3.5 プロパティ全体の評価と考察

表 1 にトリプル，プロパティ定義域，プロパティ値域，プロパティ上位下位の各関係数および正解率の 95%信頼区間を示す．表 1 より，10,769 のプロパティ名と 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．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オブジェクト型，データ型に加え，新たに，対称関係，推移関係，関数関係，逆関数関係の推定を行った．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．

4. おわりに

本稿では，日本語 Wikipedia をリソースとしてプロパティを備えたオントロジーの構築手法の提案およびその評価を行った．Wikipedia は，Is-a 関係やクラスーインスタンス関係だけでなく，非階層関係も抽出可能であり有用なリソースである．

今後は，オントロジーの規模の拡大，オントロジーの洗練について検討していく一方，構築したオントロジーの利用法や視覚化も検討していく予定である．なお，日本語 Wikipedia オントロジーおよび検索システムを，SourceForge.jp[*3]で一部公開中であり，今後更新する予定である．

参考文献

[Auer 07] Soren Auer,Christian Bizer,Georgi Kobilarov,Jens Lehmann1,Richard Cyganiak,Zachary Ives: DBpedia: A Nucleus for a Web of Open Data ,Lecture Notes in Computer Science ,Springer Berlin / Heidelberg ,pp.722-735(2007)

[Johannes 10] Johannes Hoffart, Fabian Suchanek, Klaus Berberich, Gerhard Weikum: YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia, Research Report MPI-I-2010-5-007, Max-Planck-Institut für Informatik(2010)

[玉川 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 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)

∗3 http://wikipedia-ont.sourceforge.jp/