3F2-3

日本語 Wikipedia からの汎用オントロジーの構築と評価

Building up a General Ontology from Japanese Wikipedia

桜井 慎弥[*1] Shinya Sakurai　手島拓也[*1] Takuya Tejima　石川雅之[*1] Masayuki Ishikawa　森田 武史[*1] Takeshi Morita　和泉 憲明[*2] Noriaki Izumi　山口 高平[*1] Takahira Yamaguchi

[*1] 慶應義塾大学 Keio University

[*2] 産業技術総合研究所 National Institute of Advanced Industrial Science and Technology

In this paper, we propose a method for building a general ontology from Japanese Wikipedia to decrease the cost for building and updating general ontology. We take Wikipedia as a semi-structured resource with wide-range concept coverage and new concept coverage. For building is-a relation, we apply string matching to category tree. To exploit more instances, we apply a scraping algorithm to list articles. Case studies show us that a general ontology can be build well from Wikipedia.

1. はじめに

オントロジーの中でも幅広い分野の一般的知識を記述した汎用(言語)オントロジーは，現在では英語版としては WordNet，日本語版としては EDR 電子化辞書がよく知られており，セマンティック Web 研究における貢献度は非常に高い．しかし，これらのオントロジーは膨大な時間とコストをかけて人手で構築されているため，固有名詞も含め，日々生まれ出る新しい語彙定義への即時対応が難しいのが現状である．

そこで本研究では，即時更新性，語彙網羅性に優れたオンライン百科事典 Wikipedia から汎用オントロジーを構築することを目的とする．Wikipedia の半構造化された情報資源に着目し，これをオントロジーに変換する．本稿では，Wikipedia カテゴリ階層に対する文字列照合を行うことによってオントロジーのクラス階層を構築し，一覧記事に対するスクレイピングを行うことによってインスタンスを収集する手法を提案する．ケーススタディとして実際に Wikipedia のデータを利用してオントロジーを構築し，構築されたオントロジーの品質についての評価をする．

2. 関連研究

Wikipedia からオントロジーを構築する主な研究を紹介する．

DBpedia[Auer 07]は，Wikipedia の半構造化情報を 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た．リソースとしては主に，英語 Wikipedia のインフォボックスや外部リンク，所属カテゴリといった半構造情報を利用している．しかし，抽出した情報は特にフィルタリングされておらず，インフォボックスから抽出した情報に関しては不適切な情報も大量に含まれてしまっている．

YAGO[Fabian 07]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 Wikipedia のカテゴリ階層中に存在するカテゴリであり，英語に特化した構文解析によって特定される．この *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 *Relation* を用いてメタデータを記述し，非階層構造も構築している．大規模なインスタンスの構築を可能にしているが，YAGO で提案されている手法は英語 Wikipedia に特化した手法である．本研究では，日本語 Wikipedia に対応した手法を提案し，インスタンス構築を行っている．

連絡先:桜井慎弥，山口高平，慶應義塾大学理工学部
〒223-8522 神奈川県横浜市港北区日吉 3-14-1
Tel: 045-566-1614，E-mail: {s_saku,yamaguti}@ae.keio.ac.jp

[Ponzetto 07]では，Wikipedia カテゴリ階層からの is-a 関係の抽出が試みられている．手法としては，カテゴリリンクに六つのメソッドを適用することによって関係を抽出している．メソッドの中でも主なものは，簡単な文字列照合によるものである．本研究では，前方文字列照合部除去という手法を取り入れることによって，is-a 関係の規模の拡大を行っている．

3. 日本語 Wikipedia からの汎用オントロジー構築

3.1 汎用オントロジー構築支援手法

Wikipedia から大規模なオントロジーを構築するにあたって，クラス階層については主に[Ponzetto 07]でのカテゴリ階層からの簡単な文字列照合による抽出，インスタンスは[Fabian 07]の英語に特化した手法での収集が行われているという現状である．これを踏まえ本研究では，クラス階層については Wikipedia のカテゴリ階層に新たな文字列照合を適用することによって規模を拡張し語彙の網羅性を高める．さらにインスタンスについては一覧記事にスクレイピングを適用することによって日本語に対応した大規模な収集を行う．図 1 は，Wikipedia のカテゴリ階層と一覧記事を用いた汎用オントロジーの構築の概念図である．

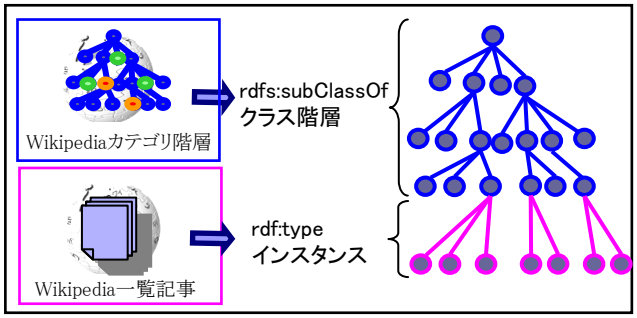

図 1　汎用オントロジーの構築の概念図

3.2 カテゴリ階層

カテゴリ階層とは，記事の分類を目的としたカテゴリが階層化されたものである．サブカテゴリは，あるカテゴリに属する記事が増加してくるとそのカテゴリの下位に作成され，カテゴリの記事の再分類に使用される．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，その名称は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．前者は単に関連の深いキーワードを示す is-a 関係としては不適切な関係であるが，後者はオントロジーの is-a 関係に相当する関係となっている．カテゴリ階層からは，この複合

語に着目した文字列ベースでの抽出手法を用いることによって，is-a 関係を抽出できる可能性が高い．

3.3 一覧記事

一覧記事とは，物事のリストが記述された記事である．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．Wikipedia の記事の中でも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せいもあり一覧記事の執筆者は非常に多く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．そのため，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と考えられる．

3.4 クラス階層構築

本実験では複合語を利用してカテゴリ階層から is-a 関係を抽出するための手法として後方文字列照合と前方文字列照合部除去の2通りの文字列照合を行い，クラス階層を構築する．

(1) 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．この手法は，[Ponzetto 07]で既に実践されている手法である．

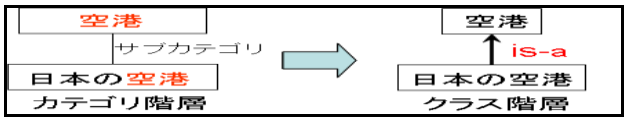

図 2　後方文字列 照合

(2) 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し，照合部を除去する手法である．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．

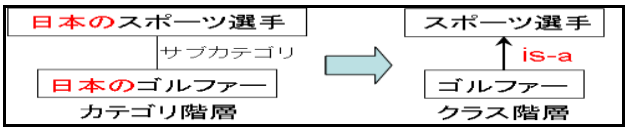

図 3　前方文字列照合部除去

3.5 インスタンス収集

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．しかし一覧記事は記述形式が統一されているため，含まれる不要情報もパターン化することができ，除去することも十分可能である．本実験では一覧記事からノイズを取り除くための手法としてスクレイピングを行い，インスタンスを収集する．

4. 実装

Wikipedia の全記事，内部リンク，カテゴリリンクなどはフリーでダウンロードすることができるため(これらのデータはダンプデータ[*1] と呼ばれる)，これを利用してクラス階層の構築とインスタンスの収集を行う．

(1) クラス階層構築

図 4 に具体的なクラス階層抽出の手法を示す．ダンプデータの categorylinks.sql で表わされるテーブルには，それぞれ *cl_from* と *cl_to* のカラムに全記事とそれが所属するカテゴリの対応が表わされている．しかしこれはカテゴリ同士の対応以外も含んでおり，さらに記事のカラムは実際の記事名を表す文字列ではなく id で表わされている．ここでダンプデータの page.sql も利用する．page.sql は全記事の id と記事名と記事の namespace の対応を表している．namespace とは記事の属性を表すもので，カテゴリ記事の namespace は 14 である．categorylinks.sql と page.sql のテーブルを結合させ，親カテゴリと子カテゴリの対応を表したテーブル *new_categorylinks* を生成する．このテーブルに図 5 に示した後方文字列照合を行うクエリ A，前方文字列照合部除去を行うクエリ B を実行し，is-a 関係を抽出する．

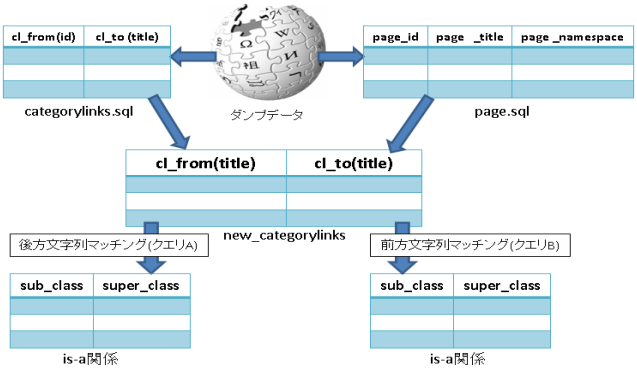

図 4　クラス階層抽出

・クエリA

```
SELECT super_class,sub_class FROM `new_categorylinks`
 WHERE sub_class REGEXP CONCAT('.*',super_class,'$')
```

・クエリB

```
SELECT
 TRIM(leading CONCAT(SUBSTRING_INDEX(super_class,'の',1),'の') FROM super_class),
 TRIM(leading CONCAT(SUBSTRING_INDEX(super_class,'の',1),'の') FROM sub_class)
  FROM `new_categorylinks`
   WHERE SUBSTRING_INDEX(super_class,'の',1)=SUBSTRING_INDEX (sub_class,'の',1)
    AND super_class LIKE '%の%'
```

図 5　文字列照合を行うクエリ

(2) インスタンス収集

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 6 のようになっている．

以下，スクレイピングの具体的な内容を説明する．

①大まかな不要情報の除去

図 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‘＊’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(筆者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．図の中の”[[ ]]”は，Wikipedia の内部リンクを表している．

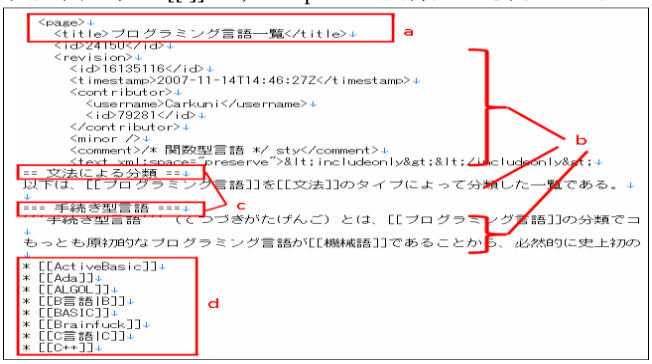

図 6　一覧記事ソーステキストの一部

[*1] http://download.wikimedia.org/jawiki/

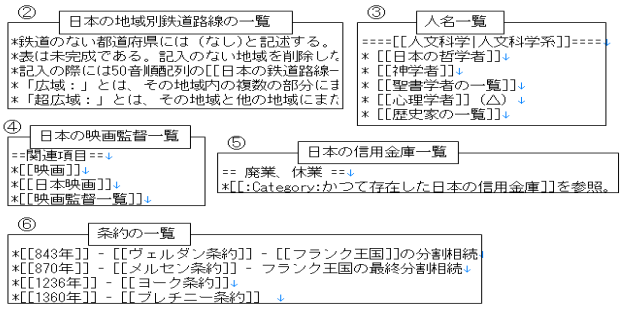

図 7　一覧記事の不要な情報の例

以下からは図 7 に示される例のように，‘＊’行に含まれる不要な情報を除去する．

②一覧記事の説明に使用される‘＊’行を除去
③「＊ ～の一覧」と同じ目次見出しの下位にある‘＊’行を除去
④不要な目次見出し下位の‘＊’行を除去
⑤不要な‘＊’行を除去
⑥不要な年号記述部分を除去
⑦‘＊’行の中でどの部分がインスタンスでどの部分が不要情報であるかを特定する6つパターンを作成し，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．

5. ケーススタディ

5.1 実験方法

2007 年 11 月現在のダンプデータをダウンロードして，オントロジーの構築を行った．データベースには MySQL，スクレイピングを行う実装言語には Java を使用した．

(1) クラス階層構築

カテゴリ階層を構成しているリンクの数は 87,126 個であった．

文字列照合は単純な 1 世代の親子関係だけではなく，2 世代のカテゴリリンクまで検索の対象を広げて適用し，クラス階層の抽出を行った．クラス階層の評価方法は，抽出した全リンクからのランダム標本抽出によるリンクの正解率の区間推定を行う．正解の判断は，下位概念が上位概念の性質を継承しているか否かという点を基準とした．正解率の 95%信頼区間の算出式として，有限修正を加えた以下の式①を利用する．

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

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．

(2) インスタンス収集

Wikipedia の一覧記事数は約 5,500 ページであった．

ダウンロードした xml テキストファイルに対してスクレイピングを行い，インスタンスの抽出を行った．インスタンスの評価方法もクラス階層の評価と同様に抽出したリンクから 1,000 個の標本抽出を行い，リンクの正解率の区間推定を行う．標本数に対して母数が非常に大きい場合は，使用する式は 5.1.(1)の式①から有限修正部を除いたものとする．

5.2 実験結果

(1) クラス階層構築

後方文字列照合によって 4,671 個，前方文字列照合部除去によって 2,521 個で，計 7,192 個の is-a 関係を抽出した．また，この is-a 関係を構成する概念数は 6,672 個であった．抽出した 7,192 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を利用して真の正解率の 95%信頼区間を算出すると，91.2±1.63%という結果が得られた．表 1, 2 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 は誤りの例とその内容を表している．次にクラス階層のルートとなっている各クラスから全てのリーフのクラスへのパスを調べた．抽出したパスの本数は 153,188 本であり，構造全体の階層の深さの平均は約 6.001 本で，分散は約 3.18 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り，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グラフを記述した．図 8 にそれを示す．

表 1　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
高等学校	通信制高等学校
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 2　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
齧歯類	ハムスター

表 3　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
地理	建築物	抽象的な語が親
教育	コミュニティ･カレッジ	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
経済	国立銀行	抽象的な語が親

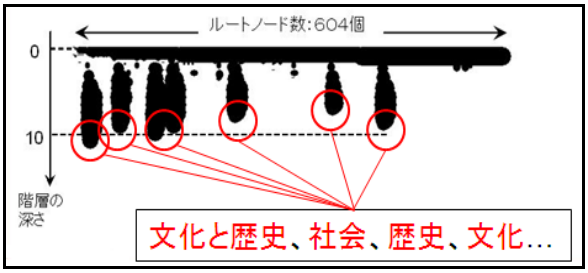

図 8　オントロジーの全体像

(2) インスタンス収集

実験によって得られたインスタンスは 332,299 個，一覧記事の記事名から生成されたクラス数は 2,265 個であった．標本抽出をして正誤判定をした結果，正解率の 95%信頼区間は，96.9±1.1%であった．表 4 に抽出されたリンクの例を示す．表 5 は，インスタンスを多く持つクラスを表したものである．また誤りにはどのようなものがあったかを，表 6 に示す．

表 4　抽出したインスタンスの例

クラス	インスタンス
国会議員	松田竹千代
作曲家	本田雅人
映画スタッフ	高村倉太郎
国鉄･JRの車両形式	ヤ230
陸上競技選手	山田宏臣
神社	仙台東照宮
東北地方の道路	八戸南環状道路

表 5　インスタンスを多く持つクラスの例

クラス	インスタンス数
東京大学の人物	3901
日本の峠	3131
日本の漫画家	2472
アメリカ海軍駆逐艦	2144
日本の声優	2125
愛知県出身の人物	2001
人名 い	1768

表 6　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
國學院大學の人物	伊藤誠（経済学者）
世界の民族衣裳	台湾
国際競技連盟	バイアスロン
スポーツ競技	サッカー

5.3 考察

(1) クラス階層構築

全体的な正解率という点ではかなり良い数値を得られたと思われる. 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 関係を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 の一つ目の誤りにように, 子クラスが親クラスと照合していても「反」や「非」などの否定語が子クラスの先頭にきている関係は is-a 関係としては誤りになってしまう. 次に二つ目の誤りは, インスタンスを表してしまっている誤りである. Wikipedia では, 有名なものであれば明らかにインスタンスであるものでもカテゴリ化され, クラスのように扱われる傾向があり, 後方文字列照合ではこのような誤抽出をしてしまう可能性が高くなっていた. 表 3 は,「抽象的な語が親クラスとなっている」という要因の誤りが大部分を占めていたということも示している. ここでいう抽象的な語とは, Wikipedia カテゴリ階層の上位に存在しているカテゴリ名である. 日本語 Wikipedia のカテゴリ階層の最上層部は, 上位オントロジーと呼ばれるもののように物ごとの厳密な分類がなされているわけではなく,「科学」,「学問」,「技術」,「自然」,「社会」,「地理」,「人間」,「文化」,「歴史」に「総記」を加えた 10 の「主要カテゴリ」がルートとなっている. この粗い分類であるルート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いことは明らかである.「抽象的な語」による誤りを多く抽出してしまった理由は, Wikipedia ではこれが分類の基幹となっているため上位部分以外の階層の中でも複合語の形でこの抽象的な語による分類が何度も登場し, それが前方照合部除去の条件に適合したためだと考えられる.

オントロジーの全体像である図 8 を見ると, あるクラスの部分だけ急激に階層が深くなっていることがわかる. 急激に深くなっているルート概念は, “文化と歴史”, “地理・事物”, “社会”, “歴史”, “文化”といった, 抽象的な概念である. このような単語以外で僅かながら深い階層が見られるのは, “機器”, “法”, “スポーツ組織”, “人物”であった. これら以外は深さ 2～4 程度の平坦な階層しか構築されていない. Wikipedia 主要カテゴリとなっているような抽象的な語が多くの誤りの階層を生み出していることが明らかになっていたが, 加えてこの図から, そのような語からしか深い階層構造を築くことができていないということもわかる. カテゴリ階層から文字列照合を利用してクラス階層を構築する場合, 上位部分の整備が極めて重要であるということである. 上位オントロジーの整備によって今回構築したクラス階層がさらによりよいものになる可能性がある.

(2) インスタンス収集

抽出したリンクの正解率を見ると, 全体の抽出の精度はかなりの高さを持っていたことがわかる. 表 5 を見ると, 人物のインスタンス数が圧倒的に多いことがわかる. 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 4 の例や, 表 5 の “日本の峠”や “アメリカ海軍駆逐艦”など, 分野にとらわれない抽出もできている.

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 6 の一つ目, 二つ目の誤りはそれぞれ, 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ルール, “( )”の注釈を除去するルールが不足していたために起こった. 三つ目, 四つ目の誤りは, ‘＊’や‘＃’の行の中のどの部分がインスタンスそのものを表しているかを特定するためのパターンが不足していたために起こってしまった誤りである. ルールの不足以外にも, 表 6 の五つ目の誤りのようにクラスーサブクラス関係を表してしまっているものもあった.

6. おわりに

本稿では, 汎用オントロジーを構築するために日本語 Wikipedia のカテゴリ階層からクラス階層を構築し, 一覧記事からインスタンスを収集する手法を提案した. 文字列照合では特に前方文字列照合部除去で文字列に依存しない is-a 関係を抽出することができ, 一覧記事からは日本語に対応した手法で大量のインスタンスを収集することに成功した. またその正解率も共に 90%を超える精度の高さであった. しかし, 質の高い汎用オントロジー構築のためには大きな課題が残されている. 複合語に着目した文字列照合は「抽象的な語」による多くの is-a 関係の誤りを生み, 文字列を利用した全自動構築には限界が見えたともいえる結果となった. この問題を解決するため, 今後は既存の上位オントロジーを利用し, さらに人間とのインタラクションを取り入れながら半自動的に Wikipedia から汎用オントロジーを構築していく予定である. インスタンス収集に関しては, スクレイピングのさらなるパターン整理が必要になってくることに加え, 一覧タイトルから生成されたクラスをクラス階層に融合させるための手法も提案していかなければならない. また, 一覧記事以外にもインスタンス収集に利用可能なリソースを探していく必要もある.

参考文献

[Auer 07] Sören 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.

[Fabian 07] Fabian M. Suchanek,Gjergji Kasneci,Gerhard Weikum: Yago: a core of semantic knowledge, Proceedings of the 16th international conference on World Wide Web, ACM , pp. 697-706,2007.

[Ponzetto 07] Simone Paolo Ponzetto,Michael Strube: Deriving a Large Scale Taxonomy fromWikipedia, Proc. AAAI-2007 , AAAI Press,pp.1440-1447,2007.

[中山 07]中山浩太郎, 原隆浩, 西尾章治郎: 人工知能研究の新しいフロンティア:Wikipedia, 人工知能学会誌, Vol.22, No.5, pp.693-701,2007

[武田 04] 武田 英明: 上位オントロジー(<特集>開発されたオントロジー), 人工知能学会誌, Vol.19, No.2, pp. 172-178, 2004