1N2-OS-39a-3

英語版 Wikipedia オントロジー構築と YAGO との比較評価

Building English Wikipedia Ontology and Comparing with YAGO

川上 時生 Tokio Kawakami　森田 武史 Takeshi Morita　山口 高平 Takahira Yamaguchi

慶應義塾大学理工学部
Keio University, Faculty of Science and Technology

We have proposed a construction method of Japanese Wikipedia Ontology which is high precision and large scale ontology. In this paper, we propose a construction method of ontology applying Japanese Wikipedia Ontology method to Wikipedia. We also compare it with overseas Wikipedia ontology such as YAGO and show differences.

1. はじめに

我々はこれまでに日本語 Wikipedia における様々な機能(カテゴリツリー, 一覧記事, Infobox, Infobox テンプレート, 定義文, 目次見出し)から, Is-a 関係や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, 同義語, トリプルといった概念および概念間の関係を抽出することにより, 高精度かつ大規模な汎用オントロジーである日本語 Wikipedia オントロジーの構築手法を提案してきた[玉川 13]. 一方で日本語 Wikipedia オントロジーは言語依存した処理が多く, 他言語のコミュニティでは利用できず, また海外の Wikipedia オントロジー(YAGO など)との比較ができないという問題があった.

本論文では, 日本語 Wikipedia オントロジーの構築手法を英語版 Wikipedia に応用したオントロジーの構築手法について提案するとともに, YAGO といった海外の Wikipedia オントロジーと比較評価する.

2. 関連研究

YAGO2[Johannes 10]は, YAGO の知識ベースの拡張として, これまでの WordNet と Wikipedia のカテゴリとの対応付けを行うだけでなく, Wikipedia と GeoName[*1] から時空間的情報を抽出することで, さらなるオントロジーの拡張を目指している. またその拡張版である YAGO3[Farzaneh 15]では英語版 Wikipedia だけではなく, その他の言語の Wikipedia を利用し, 多言語への拡張を行っている.

3. 英語版 Wikipedia オントロジー構築手法

以下では日本語 Wikipedia オントロジーの構築手法を応用した, 本研究の提案手法の詳細について述べる.

3.1 Is-a 関係抽出手法

(1) カテゴリ階層に対する文字列照合

日本語 Wikipedia オントロジーでは Wikipedia のカテゴリ階層から Is-a 関係を抽出するための文字列照合として「後方文字列照合部除去」と「後方文字列照合部除去」を行っていた. 本研究でもこの二つの文字列照合を応用してカテゴリ階層からの Is-a 関係抽出を行う.

連絡先 : 川上時生, 慶應義塾大学理工学部管理工学科,
〒223-8522 神奈川県横浜市港北区日吉 3-14-1,
TEL : 045-566-1614, Email : kawakami0412@keio.jp

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. ただし英語版では親カテゴリ名「Japan」, 子カテゴリ名「People from Japan」のように子カテゴリ名が「任意の文字列＋前置詞＋親カテゴリ名」となっていた場合, 明らかに間違った Is-a 関係を抽出してしまうことが多いので、ここでは省く. 後方文字列照合は, [Ponzetto 07]で既に提案されている手法である.

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で“任意の文字列＋の”という部分が一致しているものを抽出, 照合部を除去する手法である. 英語版ではこれを応用し, 名詞の後ろの修飾部が一致しているものを抽出、照合部を除去する. ここで名詞の後ろの修飾部に限定したのは事前の実験により, 名詞を後ろから修飾する場合には名詞の意味が限定されることが多く, 正しい is-a 関係が抽出されやすいことがわかっているためである. 例えば、図 1 では「based in」の前の名詞は組織を表す名詞がくる場合が多く, 結果として「Company」Is-a「Organization」という正しい Is-a 関係を得ることができる.

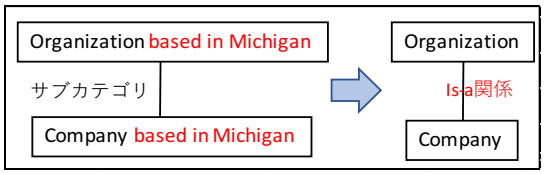

図 1 前方文字列照合部除去の例

(2) Infobox テンプレート名とカテゴリ名の照合

本手法では抽象的な Infobox テンプレートと, 領域によっては多くの具体的な概念を持つカテゴリとの関係に着目し, 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. 抽出手順を以下に示す.

1. カテゴリ名とテンプレート名の単純文字列照合
2. 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すべてのカテゴリ名とのマッチング
3. マッチングによって得られたカテゴリの配下にあるカテゴリ階層を Is-a 関係として抽出
4. 子クラスが「任意の文字列＋前置詞＋親クラス」の形になっていた場合は, これを排除する

*1 http://www.geonames.org/

(3) 目次見出しのスクレイピング

見出しに「分類」,「種類」を意味する単語が含まれる記事は分類階層が正しく記述されていることが多い. これに着目し,「Classification」,「Taxonomy」,「Genre」が含まれる記事内の階層関係をスクレイピングにより, Is-a 関係として抽出する.

3.2 クラスーインスタンス関係抽出手法

一覧記事とは, ある基準に従って, 関連する物事が列挙された記事である. 英語版 Wikipedia では一覧記事は「List of ~」の形で存在しているため, こうした記事に着目し, 記事名をクラス, 記事内に列挙された物事をインスタンスとみなして, スクレイピングによりクラスーインスタンス関係を抽出する.

3.3 トリプル抽出手法

Infobox が有する「記事ー項目ー値」という三つ組を「インスタンスープロパティープロパティの値」として抽出する. ここでは Java Wikipedia API (Bliki engine)[*2]を用いて HTML に変換することで, プロパティに当たる部分を統一して抽出する. またプロパティのタイプについて, そのプロパティの目的語がインスタンスかリテラルかを調べることで, owl:ObjectProperty か owl:DatatypeProperty に分類する.

3.4 定義文からの上位下位関係抽出手法

Wikipedia の定義文を Stanfordparser[*3] を用いて形態素解析を行うことで, 上位下位関係を抽出する. 図 2 の例では「author」,「writer」を上位語,「novelist」を下位語として抽出できる. また抽出した上位下位関係を前処理で抽出したクラスの集合を用いて, Is-a 関係とクラスーインスタンス関係に分類する. 具体的には上位語・下位語ともにクラスの集合に含まれていた場合は Is-a 関係, 上位語のみがクラスに含まれていた場合はクラスーインスタンス関係に分類し, どちらにも当てはまらない場合は Unknown とする.

A **novelist** is an author or writer of novels, though often novelists also write in other genres of both fiction and non-fiction. Some novelists are professional

図 2 記事「Novelist」の定義文

4. 評価

4.1 Is-a 関係抽出結果

(1) カテゴリ階層に対する文字列照合による抽出結果と考察

カテゴリ階層から後方文字列照合によって 281,394 個, 前方文字列照合部除去によって 23,516 個の Is-a 関係を抽出した. 重複を除くと全体として 302,425 個の関係を抽出することが出来た. 抽出した関係から 1,000 個の標本を抽出し, 以下の式により正解率の 95%信頼区間を推定した.

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

その結果, 後方文字列照合について 95%信頼区間は 96.2±1.18%, 前方文字列照合部除去について 95%信頼区間は 80.2±2.42%であった. 全体としては 93.4±1.54%という結果を得た. 表 1 は上の 2 行が後方文字列照合の手法により, 下の 2 行が前方文字列照合部除去の手法により抽出した関係を示している. 表 1 より, 前方文字列照合部除去によって文字列に依存しない関係を抽出できていることがわかる.

表 1 文字列照合で抽出した Is-a 関係の正解例

親クラス	子クラス
magazine	Japanese magazine
American person	American musician
Ship	Ferry

次に, 表 2 に誤って抽出された Is-a 関係を示す. 表 2 の 1, 2 行目の誤りは、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ているため, 結果として文字列照合によりクラスーインスタンス関係が抽出されてしまうことがある. これは 3.2 項で述べたクラスーインスタンス関係抽出手法により抽出された結果を利用することで排除できる. 表 2 の 3 行目の誤りは, Wikipedia カテゴリ階層の上位に存在する抽象的なカテゴリを親に持つ階層の場合に, 誤った Is-a 関係を抽出している例である. 英語版 Wikipedia の上位カテゴリは,「Arts」,「Culture」,「Health」,「Politics」など 21 個の主要カテゴリから構成されており, これが Wikipedia 階層の分類の基幹となっている. この誤りはルートからの階層の深さで限定することで排除できると考えられる.

表 2 文字列照合で抽出した Is-a 関係の誤り例

親クラス	子クラス
Asia	Laos
Landshut	EV Landshut
Politics	Sector

(2) Infobox テンプレート名とカテゴリ名の照合による抽出結果と考察

3.1(2)項で述べた手法により, 6,315 個の Is-a 関係を抽出した. 抽出した関係から 1,000 個の標本を抽出した結果, 95%信頼区間は 80.1±2.27%であった. 表 3 に抽出した正解例を示す.

表 3 Infobox テンプレート名とカテゴリ名の照合で抽出した Is-a 関係の正解例

親クラス	子クラス
Japanese singer	Japanese soprano
Embedded system	Onboard computer
Computer hardware	Computing output device

表 3 より, 文字列に依存しない関係を多く抽出できていることがわかる. 一方で誤ってクラスーインスタンス関係が抽出されることも多かった. これは抽出した関係の多くは「Mountains」や「Islands」,「Seas」といった地形に関するカテゴリがルートカテゴリとなっており, これらの下位カテゴリにはインスタンスが多く含まれるためである. また今回は Infobox テンプレートとカテゴリ名の照合を文字列一致により照合したが, 文字列一致していなくても意味的に一致するものもある. 例えば,「instrument」テンプレートは楽器記事に利用されるテンプレートであるが, カテゴリ階層に「instrument」というカテゴリはなく, 代わりに「Musical instrument」というカテゴリが存在する. こうしたものから抽出するためには, 前処理で形態素解析により名詞句に限定するなどの処理が必要である.

*2 http://code.google.com/p/gwtwiki
*3 http://nlp.stanford.edu/software/lex-parser.shtml

(3) 目次見出しのスクレイピング抽出結果と考察

3.1(3)項で述べた手法により，83,003 個の Is-a 関係を抽出した．抽出した関係から 500 個の標本を抽出した結果，95%信頼区間は 65.8±2.92%であった．表 4 に抽出した正解例を示す．

表 4 目次見出しのスクレイピングで抽出した Is-a 関係の正解例

親クラス	子クラス
Field artillery	Mountain gun
Family Delphinidae	Genus Deophinus
Idiophone	Slit drum

表 4 より，文字列照合では抽出できなかった関係を抽出できていることがわかる．また先ほどの Infobox テンプレートとカテゴリ名の照合では抽出できない，生物系の関係も多く抽出することが出来た．一方で間違って抽出された関係も多くあった．本手法は他の Is-a 関係抽出手法と異なり，本文の一部を利用しているため，意図しない書き方をしている記事も多い．そのため，不適切な分類階層がそのまま Is-a 関係として抽出され，精度が下がってしまっている．

4.2 クラスーインスタンス関係抽出手法

Wikipediaダンプデータから抽出した一覧記事に対して3.2で述べた手法により，クラスー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数は 1,767,124 個，クラス数は 33,806 個，クラスーインスタンス関係数は 2,705,573 個であった．抽出したクラスーインスタンス関係から 1,000 個の標本を抽出し，正解率の区間推定を行った．その結果，89.2±1.92%であった．表 5 に正しく抽出できた関係を，表 6 に誤って抽出された関係を示す．

表 5 一覧記事から抽出した関係の正解例

クラス	インスタンス
Japanese writer	Fukuzawa Yukichi
Anime aired on Nippon Television	Hunter×Hunter
Programming language	Ruby (programming language)

表 6 一覧記事から抽出した関係の誤り例

クラス	インスタンス
Composition for piano and orchestra	Nikolai Kapustin
Bulldog mascot	Carthage Independent School District

表 5 を見ると，作家などの人物やプログラミング言語などの幅広いインスタンスを抽出できていることがわかる．一方で表 6 より，「Composition for piano and orchestra」というクラスのインスタンスとして人物が含まれていることがわかる．図 3 はこの誤ったクラスーインスタンス関係を抽出している一覧記事である．

List of compositions for piano and orchestra

From Wikipedia, the free encyclopedia

- Nikolai Kapustin
  - Concertino for piano and orchestra, Op. 1 (1957)
  - Concerto for piano and orchestra No. 1, Op. 2 (1961)
  - Concerto for piano and orchestra No. 2, Op. 14 (1974)
  - Concerto for piano and orchestra No. 3, Op. 48 (1985)

図 3 誤ったクラスーインスタンス関係が抽出される一覧記事

図 3 からわかるようにこの記事では作品の一覧記事であるにも関わらず，演奏者が項目として含まれてしまっており，その結果誤ったクラスーインスタンスが抽出されている．このように一覧記事において階層関係となっている項目の場合は，最下位にある項目のみを抽出するなどの処理が必要であると考えられる．

4.3 トリプル抽出手法

Wikipediaのダンプデータから8,311,427のInfoboxと，12,039のInfoboxテンプレートを抽出し，22,767,071 個の Infobox トリプルを抽出した．また Infobox トリプルにおけるプロパティの種類は，12,088 個であった．表 7 に抽出したプロパティ名の内，利用頻度が高い上位 5 つのプロパティ名を示す．

表 7 利用頻度が高い上位 5 つのプロパティ

プロパティ名	トリプル数	タイプ
Born	1,236,081	Object
Country	692,929	Object
Website	564,401	Datatype
Location	513,020	Object
Died	477,372	Object

また全トリプルから1,000個の標本を抽出し，正解率の区間推定を行った．結果は 93.1±1.57%であった．誤りとしては図 4 のような Infobox の場合，「Years」がプロパティ，「Team」がプロパティの値として抽出されてしまっている．

Teams managed	
Years	**Team**
2009	Al-Majd[1]
2011–2012	Al-Oruba[2]

図 4 トリプル抽出に失敗する Infobox の例

4.4 定義文からの上位下位関係抽出手法

3.4 項で述べた手法により，定義文から結果として 3,114,222 個の上位下位関係を抽出することが出来た．また Is-a 関係とクラスーインスタンス関係に分類した結果，クラスーインスタンス関係が 2,036,428 個，Is-a 関係が 2,898 個に分類された．結果として約 65%の上位下位関係を分類することが出来た．一方で約 3 割の上位下位関係が unknown に分類された理由として，これまでに抽出したクラスの数が不足している点が挙げられる．この点については Is-a 関係の抽出手法をさらに改善することで向上すると考えられる．

また分類されたクラスーインスタンス関係と Is-a 関係についてそれぞれ精度を求めると，クラスーインスタンス関係の精度は 93.7±1.51%，Is-a 関係の精度は 80.0±2.01%であった．表 8 に誤って抽出されたクラスーインスタンス関係を，表 9 に誤って抽出された Is-a 関係を示す．

表 8 定義文から抽出されたクラスーインスタンス関係の正解例

親クラス	子クラス
bishop	Anglican bishop in Jerusalem
Field	Gender study

表 9 定義文から抽出された Is-a 関係の正解例

親クラス	子クラス
Capital	London
Programming language	C++

クラスーインスタンス関係については本来クラスに含むべきものが含まれていないことが原因で誤って抽出されるものが多かった. 逆に Is-a 関係については本来クラスに含まれるべきでないものが含まれていることが原因となっているものが多かった. これらの精度を上げるためには Is-a 関係の精度を高めていく必要がある.

5. YAGO との比較

YAGO3 は現在他言語対応している. 今回は, 同じリソースを用いて構築されたオントロジーで比較するために, YAGO2 を比較対象として抽出方法及び関係数, 抽出される関係の比較を行う.

表 10 本研究と YAGO との抽出方法の比較

	本研究	YAGO
Is-a 関係, クラスーインスタンス関係の抽出方法	カテゴリの文字列照合	WordNet を利用した Wikipedia カテゴリによる記事分類
	Infobox テンプレート名とカテゴリ名の照合	
	目次見出しからの Is-a 関係抽出	
	一覧記事からのクラスーインスタンス関係抽出	
	定義文からの上位下位関係抽出	

表 11 本研究と YAGO との抽出数・精度の比較

	本研究		YAGO	
	抽出数	精度	抽出数	精度
Is-a 関係	349,982	92.5± 1.53%	367,040	93.4%
クラスーインスタンス関係	5,092,036	92.4± 1.64%	8,414,398	97.7%

抽出方法については, YAGO では WordNet のクラスと Wikipedia のカテゴリを対応付けることで階層を構築し, そのカテゴリに属している記事をインスタンスとして抽出している. この手法はカテゴリを持つ記事全てがインスタンスとなる可能性があるため, 多くのインスタンスを抽出することが可能である. 一方で本研究ではカテゴリ階層だけでなく, 目次見出しや定義文, 一覧記事中の項目といった本文中の情報も利用しており, 精度は下がるものの, 記事が存在しないインスンタンスやクラスの抽出が可能という点で差別化が出来ている.

表 11 では抽出数・精度を比較している. ここでは 3 項で述べた各処理で抽出した関係を統合し, Is-a 関係に誤って含まれているクラスーインスタンス関係などの不適切な関係を排除した結果を示している. また YAGO についても Wikipedia から抽出した関係のみを対象としている. Is-a 関係については抽出数・精度ともに YAGO に近い数字になっているが, クラスーインスタンスの抽出数については YAGO に大きく劣る結果となった. これは本研究手法では一覧記事に記載されているか定義文が存在する記事からしかクラスーインスタンスを抽出することが出来ないためである. 一方で定義文や一覧記事にしか書かれていない関係があった場合には, YAGO で抽出できない関係を抽出できる. 表 12 に抽出される関係の比較結果を示す. ここでは本研究で正しく抽出することができた関係 1000 個について, それと意味的に近い関係が YAGO に存在するかどうかを判定している.

表 12 本研究と YAGO との抽出した関係の比較

	YAGO にない関係数
Is-a 関係	589 / 1000
クラスーインスタンス関係	422 / 1000

表 12 より, 実際に YAGO で抽出できない関係を多く抽出できていることがわかる. YAGO で抽出できない Is-a 関係の特徴としては, 中間概念を含む関係が多いことが挙げられる. YAGO は Is-a 関係の親クラスに WordNet の末端のクラスを利用しているため, WordNet に存在しないクラスは YAGO でも抽出できない. 例として WordNet には「voice actor」というクラスが存在しないため, 「Japanese voice actor」Is-a「voice actor」という関係は抽出できないが, 本研究手法では抽出が可能である. また YAGO では抽出できないクラスーインスタンス関係としては, 一覧記事には書かれているが記事自体は存在しないインスタンスが多かった. またカテゴリに反映されていない情報が定義文に記載されていた場合にも YAGO にはない新たな関係を抽出することが出来た.

6. おわりに

本稿では, 日本語 Wikipedia オントロジーの構築手法を英語版 Wikipedia に応用したオントロジーの構築手法について提案し, その評価を行った. また海外の Wikipedia オントロジーである YAGO と比較を行うことで, YAGO で抽出できない関係を抽出可能であることを示し, その有用性を示した.

今後の課題としてはまだ英語版に対応できていない日本語 Wikipedia オントロジーの処理(定義域抽出, 値域抽出など)について, 適用できるように改良していく必要がある.

参考文献

[玉川 13] 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, 人工知能学会セマンティック Web とオントロジー研究会, SIG-SWO-A1203-01 (2013)

[Johannes 10] Johannes Hoffart, Fabian Suchanek, Klaus Berberich, Gerhard Weikum: YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia, Reserch Report MPI-I-2010-5007, Max-Planck-Institut fur Informatik (2010)

[Farzaneh 15] Farzaneh Mahdisoltani, Joanna Biega, Fabian M. Suchanek: A Knowledge Base from Multilingual Wikipedias, In: CIDR (2015)

[Ponzetto 07] Simone Paolo Ponzetto, Michael Strube: Deriving a Large Scale Taxonomy from Wikipedia, Proceedinggs of national conference on Artificial intelligence, pp.1440-1447 (2007)