2I3-NFC4-3

日本語Wikipediaインフォボックスからのプロパティ自動抽出

Automatic Extraction of Properties from Japanese Wikipedia Infobox

玉川 奨[*1] Susumu Tamagawa　桜井 慎弥[*1] Shinya Sakurai　手島 拓也[*1] Takuya Tejima　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]独立行政法人 産業技術総合研究所 National Institute of Advanced Industrial Science and Technology

For cost reduction of ontology construction, more attention comes to ontology learning research and Wikipedia becomes popular as information source. However, ontology learning from Wikipedia usually takes care of class-instance relationship and is-a relationship, thus we have less research focused on non-hierarchical relationship. So here is discussed how to extract properties automatically from Japanese Wikipedia Infobox and learning non-hierarchical relationship. The learned property includes owl:Object/DatatypeProperty(property types), rdfs:domain (property domains) and rdfs:range (property ranges).

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，構築コストが高い．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，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その情報資源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，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，Wikipediaからのオントロジー学習手法に関する研究はクラスーインスタンス関係やIs-a階層関係に着目されているものが多く，非階層関係に着目されている研究は少ない．

そこで本論文は，日本語Wikipediaから半自動的にプロパティの抽出およびその定義域と値域を定義する手法を提案する．本手法では，Infoboxトリプルからプロパティを自動的に抽出する．さらに，一覧記事のスクレイピングにより抽出したクラスーインスタンス関係およびカテゴリ階層とInfoboxテンプレートの照合により抽出したIs-a関係をそれぞれ用いて，各プロパティの定義域と値域の定義も行う．

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，クラス・プロパティを手動で構築しており，構築したクラス数も170と少ない．

連絡先: 玉川 奨，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
{s_tamagawa,yamaguti}@ae.keio.ac.jp

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 とは英語 Wikipedia のカテゴリであり，“American singers of German origin” カテゴリのように，カテゴリ名の head 部分である “singers” が複数形になっているカテゴリのことである．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メタデータを記述し，非階層関係も構築している．非階層関係に着目している点で，高度なオントロジーであるが，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，プロパティに関する検討が弱い．

3. Infoboxからのプロパティ構築

3.1 Wikipedia オントロジー

我々はこれまで，Wikipedia から大規模なオントロジー（Wikipedia オントロジー）を学習する手法の提案をしてきた [桜井 09]．Wikipedia オントロジーは以下の六つの関係定義を行うことによって構築される．

1. Is-a 関係 (rdfs:subClassOf)
2. クラス-インスタンス関係 (rdf:type)
3. Infobox トリプル (owl:Object/DatatypeProperty)
4. プロパティ定義域（rdfs:domain）
5. プロパティ値域（rdfs:range）
6. 同義語 (skos:altLabel)

本論文では Infobox トリプルからプロパティを抽出するだけでなく，いくつかの Infobox テンプレートにモデリングを行うことでプロパティを owl:ObjectProperty と owl:DatatypeProperty に分類する．さらに，我々がこれまでに提案した手法 [桜井 09] のうち，Infobox テンプレート名とカテゴリ名の照合による手法を用いて各プロパティの定義域を，一覧記事に対するスクレイピング手法で抽出したクラスーインスタンス関係と，Infobox テンプレート名とカテゴリ名の照合による手法で抽出した Is-a 関係を用いることで各プロパティの値域をそれぞれ抽出する．

3.2 Infobox トリプルのモデリングによるプロパティ抽出

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－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，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を抽出できるが，いくつかの問題が存在する．まず，Infobox には記事の種類ごとにテンプレートが存在し，かつ英語版 Wikipedia のテンプレートを利用できるという問題がある．これは MediaWiki を用いて日本語版と英語版で完全互換性をとることで，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が別のものとなってしまう．図 1 の例では，記事ソース内には「Genere」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,「Genere」プロパティとしてそのまま抽出してしまう．次に，全てのプロパティの値をリテラルとして抽出すると，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，プロパティの種別がわからないという問題がある．図 1 の例では，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．

以上 2 つの問題点に対応するため，以下の 1〜4 の手順でプロパティの抽出を行う．

1. 記事ごとに Infobox とそのテンプレートの情報をデータベースに格納
2. Java Wikipedia API (Bliki engine)[*5] を用いて HTML ソースに変換することで，Infobox を日本語記述で統一して抽出し，Infobox から「インスタンス－プロパティ－リテラル」の形でデータベースに格納
3. 40 種類の Infobox テンプレートにおけるプロパティについてモデリングを行う
4. 3. とデータベースに格納したトリプルから新たなトリプルを抽出

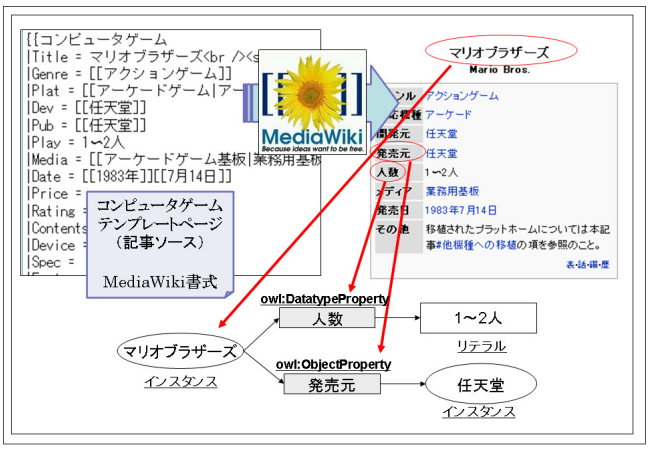

図 1: Infobox トリプルからのプロパティ抽出の一例

*5 http://code.google.com/p/gwtwiki/

手順 3 においてモデリングとは，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 ダンプデータを使用し，事前に抽出した際に Infobox の総数が約 20 万 2000 個だったのに対し，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．以上の手順より，7 割程度の Infobox に対して適切に Infobox トリプルが抽出できると考えられる．

3.3 プロパティ定義域抽出

[桜井 09] のプロパティ定義域の抽出法を用いてプロパティ定義域を抽出する．ここで抽出した各プロパティ定義域は定義域として最上位の概念であるテンプレート名だけでなく，より具体化され，ドメインに特化した定義域である．

3.4 プロパティ値域抽出

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の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．そこで値域の抽出には,「一覧記事のスクレイピングによるインスタンス抽出手法」によって抽出されたクラス－インスタンス関係を用いる手法と，定義域の抽出と同様に「Infobox テンプレート名とカテゴリ名の照合による Is-a 関係の抽出法」により得た Is-a 関係を用いる手法を提案する．

まず，3.2 で抽出した owl:ObjectProperty となる各プロパティの目的語 (インスタンス) に着目する．Wikipedia の性質上，既に記事が書かれている単語はリンクされている場合が多く，とりわけ Infobox をトリプルと据えた場合の値部分の単語の記事が既存の場合にはリンクされている可能性が高い．そこで，プロパティの値となるインスタンス名と一覧記事から抽出したクラス－インスタンス関係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) をプロパティの値域として抽出する．

次に，先の手法では抽出できないプロパティの値域を抽出するために，前述した手法と同様に，各プロパティ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「Infobox テンプレート名とカテゴリ名の照合による Is-a 関係の抽出法」により得た Is-a 関係における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．さらに抽出されたクラスの Is-a 関係における最上位概念 (Infobox テンプレート名) 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各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 2 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－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 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．

4. 実験結果と考察

実験は 2010 年 3 月時点の Wikipedia ダンプデータ (jawiki-latest-pages-articles.xml)[*5]をダウンロードし，データベース

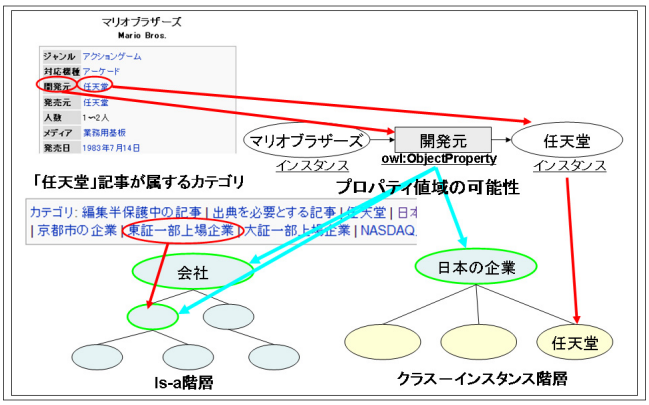

図 2: プロパティ値域の抽出の一例

は MySQL，実装言語は Java 言語を用いて行った．

4.1 Infobox トリプルによるプロパティの抽出結果と考察

Wikipedia のダンプデータから 230,552 の Infobox と，833 の Infobox テンプレートを抽出し，3.2 で述べた手法により，1,287,635 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，7,085 であった．表 1 に Infobox トリプルで利用頻度が高いプロパティを示す．

表 1 より，Infobox トリプルで利用頻度が高いプロパティの多くは owl:ObjectProperty となっている．この原因としては，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，3.2 で述べた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の高い，人物，Album，Single，会社，駅などに記載されたプロパティに多く見られる．

全 7,085 プロパティのうち，モデリングを行った 40 個の Infobox テンプレートから 286 のプロパティについて，owl:ObjectProperty と owl:DatatypeProperty の分類ができた．これら 286 のプロパティを持つ Infobox トリプル数は 964,902 であった．上記モデリングにより，74.9%の Infobox トリプルを分類できたことになる．

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

全 Infobox 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．ここでは，インスタンス－プロパティ－プロパティの値の関係が成立するものを正解とした．その結果，正解率の 95%信頼区間は，94.1 ± 1.46%であった．そのうちモデリングにより分類できた Infobox トリプルの正解率は 98.6%と高精度であった．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．) さらに，スポーツ選手に多く見られる Infobox であるが，年度ごとの成績が記述されている場合に，トリプルの述語として「2004 年/2005 年度」という記述が用いられる場合があり，このような年度によるプロパティを抽出してしまっていた．(例えば，バスケットボール選手であるマイケル・ジョーダンの場合には，経歴プロパティの値として，「1984-1993，1995-1998，2001-2003」を抽出し，さらにこれらの「1984-1993，1995-1998，2001-2003」をプロパティとして「シカゴ・ブルズ，シカゴ・ブルズ，ワシントン・ウィザーズ」という値を抽出してしまう．数値データはリテラルによる記述が望ましいので，モデリング方法を検討する必要がある．)

4.2 プロパティ定義域の抽出結果

4.1 で示した 7,085 のプロパティに対して，3.3 の手法により，全てのプロパティが Infobox テンプレート名を最上位概念として定義域が抽出でき，その関係数は 10,639 であった．10,639 のプロパティ定義域から先と同様に正解率の 95%信頼区間を算出した結果，正解率の 95%信頼区間は，95.5 ± 1.22%だった．プロパティ定義域を誤って抽出したのは，4.1 で誤って抽出したプロパティに対するプロパティ定義域のみであった．例えば，「1990 － 1992 年」というプロパティに対してバスケットボール選手という定義域を誤って抽出していた．また，下位概念も含めた関係数は 192,052 であった．

4.3 プロパティ値域の抽出結果と考察

クラス－インスタンス関係を用いた抽出法により，1,893 のプロパティについて値域を定義でき，プロパティと値域の関係数は 18,719 であった．表 2 に利用頻度が高いプロパティと値域を示す．「ジャンル」プロパティの値域として「ポピュラー音楽のジャンル」や「発売元」プロパティの値域として「ゲーム会社」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としては日本語 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，さらにクラス－インスタンス関係抽出において，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と抽出されているためである．

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

表 1: Infobox トリプルで利用頻度が高いプロパティの例

プロパティ	利用インスタンス数	プロパティのタイプ
所在地	32,038	owl:ObjectProperty
本社所在地	31,959	owl:ObjectProperty
生年月日	28,519	owl:DatatypeProperty
ジャンル	26,267	owl:ObjectProperty
出身地	26,016	owl:ObjectProperty

表 2: クラス－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	利用インスタンス数	値域
ジャンル	10,441	ポピュラー音楽のジャンル
国籍	9,127	島国
対応機種	2,938	ゲーム機
発売元	2,248	ゲーム会社

∗6 Wikipedia ダンプデータ: http://download.wikimedia.org/jawiki/

表 3: Infobox テンプレート名とカテゴリ名の照合による Is-a 関係を用いた値域抽出法により抽出した値域の例

プロパティ	値域 (最下位概念)	値域 (最上位概念)
本社所在地	日本	国
出身地	ヨーロッパの首都	国
放送局	日本ラジオ広告推進機構加盟局	日本のラジオ局
在籍チーム	フランスのサッカークラブ	サッカークラブ

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－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．プロパティの値域定義における誤りの多くは，クラス－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－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．

次に，Is-a 関係を用いた値域の抽出を行った．1,104 のプロパティについて値域を定義でき，最下位概念との関係数として 8,729，最上位概念との関係数として 2,337，重複を除くと 11,065 のプロパティと値域の関係が抽出できた．ここで最下位概念とは記事が属するカテゴリとの照合により得られたクラスであり，最上位概念とはカテゴリ名と Infobox テンプレート名の照合によって得られた Is-a 関係において，最下位概念のルートとなるクラス (Infobox テンプレート名) である．表 3 にプロパティと最下位概念及び最上位概念の値域を示す．利用頻度が高い値域の殆どが「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」クラスとなるものが多い．しかし，クラス－インスタンス関係を用いた抽出法では抽出できない,「出身地」プロパティの値域として「ヨーロッパの首都」や,「在籍チーム」プロパティの値域として「フランスのサッカークラブ」など，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，本来はプロパティのタイプが owl:DatatypeProperty になるため，値域はリテラル (rdfs:Literal) となる．さらに,「代表者」プロパティの値域として「存命人物」など抽象的すぎる概念が抽出された場合もあった．これらは Is-a 関係における抽出の誤りが影響している．これらは Is-a 関係の抽出の誤りが影響しているためである．

4.4 プロパティ全体の評価と考察

表 4 に抽出したプロパティ・定義域・値域の全関係数・正答率・定義率を示す．ここで，正答率とは Wikipedia のコンテキストにおいて妥当のものの割合であり，定義率とは抽出した全 7,085 プロパティ中で定義できたプロパティの割合である．全体として非常に高い正答率で抽出できている．プロパティに関しては約 120 万ものトリプルのうち 7 割程度のものが owl:ObjectProperty と owl:DatatypeProperty に分類できており，精度も高い．定義域に関しては最上位の定義域の関係数は 10,639 であるが，全てのプロパティが定義域を持っている．値域は精度は高いものの関係数としては定義域に比べると少なく，また定義できたプロパティも全体の 3 割程度となってしまっている．これは今回の抽出において，プロパティの目的語となるインスタンスと記事名にしか着目しておらず，Wikipedia において記事が存在しない場合には抽出が出来ないためである．今後，別の方法により抽出を試みる必要がある事を示している．また，抽象的すぎるクラスがプロパティの定義域や値域として定義されている場合もあり，より具体的な定義域および値域を定義するための手法を検討する必要もある．

表 4: プロパティ全体の関係数と正答率と定義率

	全関係数	正答率	定義率
プロパティ	1,203,404	94.1 ± 1.46%	-
プロパティ定義域	192,052	95.5 ± 1.22%	100%
プロパティ値域 (全て)	29,386	94.0 ± 1.45%	30.2%
クラス-インスタンス関係	18,719	93.2 ± 1.52%	26.7%
Is-a 関係	11,065	95.1 ± 1.28%	15.6%

5. おわりに

本論文では，日本語 Wikipedia をリソースとして Infobox を用いてプロパティを抽出し，各プロパティの定義域と値域の抽出をすることで，プロパティ構築手法の提案と評価を行った．Wikipedia は，Is-a 関係やクラス－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，オントロジー学習において，コストレスで大規模なオントロジーを構築するために有用なリソースであることを示すことができた．

今後は，オントロジーの規模の拡大およびプロパティ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も検討していく予定である．現在，Web 上で公開されている RDF データは多く存在し，特にインスタンスを記述した RDF データを公開・共有・連結し合うべきという風潮が高まっている (Linked Data)．そこで，本研究で構築中の Wikipedia オントロジーの利用法として，Linked Data を用いた検索支援 TOOL WiLD(Wikipedia Linked Data Application) を現在開発中である．実装は主に 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いる．ユーザーが検索インタフェースを通してキーワードを入力すると，検索 API が Wikipedia オントロジーの関連する概念を RDF/XML 形式で出力し，その出力を受け取った検索インタフェースモジュールはユーザが閲覧しやすい形式でブラウザに表示する．ユーザはこの結果を見てさらに別の入力を与えることで，コンピュータのインタラクションを通じて，欲している検索結果を得ることができる．なお，日本語 Wikipedia オントロジーおよびその検索システムを，SourceForge.jp*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)

[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)

[桜井 09] 桜井慎弥, 手島拓也, 森田武史, 和泉憲明, 山口高平, ”Wikipedia オントロジーに基づくドメインオントロジー構築支援環境の実現と評価”, 第 23 回人工知能学会全国大会 2G1-NFC5-1(2009)

∗7 http://wikipedia-ont.sourceforge.jp/