Wikipedia からの大規模な上位下位関係の獲得

隅田飛鳥* 吉永直樹† 鳥澤健太郎* 萬成賢太郎*
北陸先端科学技術大学院大学* 日本学術振興会†

1 はじめに

本稿では，Wikipedia から高精度で大量の上位下位関係を自動獲得する手法について述べる．上位下位関係は，情報検索や Web ディレクトリなど，情報爆発時代の膨大な Web 文書へのアクセスを容易にする様々な技術への応用が期待されている．これまで，一般の文書を知識源として，様々な上位下位関係の獲得手法が提案されている [2, 8, 1, 6, 10]．しかしながら，概念具対物関係を含む広範な上位下位関係を獲得しようとすると，これらの手法では大量の文書を大規模な計算機資源を用いて処理する必要があり，例えば，我々が以前提案した上位下位関係の獲得手法 [10] を用いた場合，0.7TB の Web 文書を処理しても僅か約 40 万件の上位下位関係しか取れないなど，100 万件以上の上位下位関係を獲得するのは容易ではない．

これに対し本研究では，様々な事物に関する常識的知識をより密に記述する Wikipedia[*1]を知識源として，超大規模な上位下位関係データベースを「手軽に」構築することを目指す．具体的には，我々がこれまで開発した，Wikipedia の階層構造から上位下位関係を獲得する既存手法 [11, 15] を拡張し，Wikipedia の定義文やカテゴリタグから獲得された上位下位関係候補についても，機械学習を用いて適切な上位下位関係を選別することで，Wikipedia 全体から高精度で大量の上位下位関係を獲得する．

実験では，約 1.8GB の日本語版 Wikipedia から，約 188 万件の上位下位関係を 89.8%以上の適合率で獲得することができた．

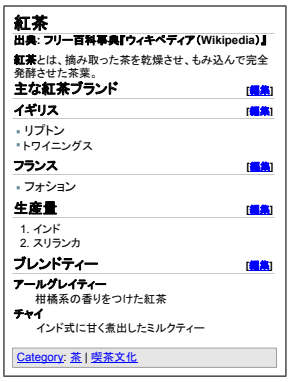
2 背景知識

本節では，まず上位下位関係獲得の知識源として利用する Wikipedia について説明する．その後，これまで提案されている Wikipedia からの上位下位関係獲得手法について説明する．

2.1 Wikipedia の記事の構造

Wikipedia は Web 上に構築された誰でも編集可能な百科事典であり，多種多様な事物を見出しとする記事から構成される．図 1(a) は見出し語「紅茶」に対する記事の例である．以下で，我々が上位下位関係候補獲得の知識源として利用する，Wikipedia の各記事に含まれる定義文，カテゴリ，及び階層構造について説明する．

[*1] http://ja.wikipedia.org/



(a) ブラウザ表示

```
1  紅茶とは、摘み取った茶を乾燥させ、もみ込んで
   完全発酵させた茶葉。
2  = 主な紅茶ブランド =
3  == イギリス ==
4  * リプトン
5  * トワイニングス
6  == フランス ==
7  * フォション
8  = 生産量 =
9  # インド
10 # スリランカ
11 = ブレンドティー =
12 ;アールグレイティー :柑橘系の香りをつけた紅茶
13 ;チャイ :インド式に甘く煮出したミルクティー
14 [[Category:茶]]
15 [[Category:喫茶文化]]
```

(b) MediaWiki コード

図 1: 「紅茶」に関する Wikipedia の記事

表 1: 階層構造に関する MediaWiki 構文の修飾記号

優先度	修飾記号の種類	記述方法	例
1	節見出し	$=+$ *title* $=+$	== イギリス ==
2	定義の箇条書き	;*title*:*def*.	; チャイ:ミルクティー
3	番号付き箇条書き	$\#+$ *title*	# インド
3	番号なし箇条書き	$*+$ *title*	* リプトン

注: *title* は見出しを，+は直前の記号が連続して出現しうることを示す．

定義文　多くの Wikipedia の記事は，その冒頭に見出し語を簡潔に定義する文を含む [4]．また，見出し語の定義にはその上位語が頻繁に用いられる．例えば，図 1 の記事では，冒頭の文「紅茶とは，摘み取った茶を乾燥させ，もみ込んで完全発酵させた茶葉。」により，紅茶の上位語 (の一つ) である茶葉を用いて紅茶が説明されている．我々は，Kazama らの手法 [4] と同様に，Wikipedia の各記事の冒頭の一文を見出し語に対する定義文とみなし，上位下位関係獲得の知識源とする．

カテゴリ　Wikipedia の各記事には，執筆者により見出し語の分類や関連語などのカテゴリタグが付与されており，本稿ではこれらのカテゴリタグを上位下位関係獲得の知識源として利用する．図 1 の記事には,「茶」と「喫茶文化」がカテゴリとして付与されている．このように，カテゴリは上位語そのものであることが頻繁にある [9].

階層構造　Wikipedia の記事は，HTML より明確な構造をもつ MediaWiki 構文により記述されており，多段

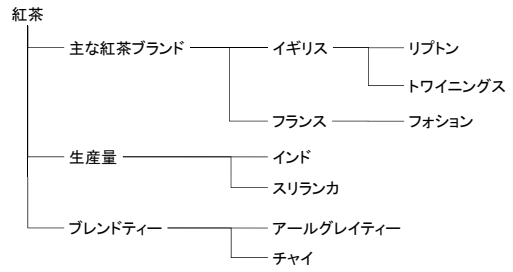


図 2: 図 1 の記事から抽出された階層構造

の箇条書きに相当する階層構造を含む．本稿では MediaWiki 構文のうち，記事の階層構造を扱う表 1 の修飾記号に注目し，記事から *title* をノードとするグラフ構造として階層構造を抽出する．具体的には，*title* に付与されている修飾記号の優先度と長さに従ってノードの親子関係を決定することで階層構造を抽出する．例えば，図 1(a) のページからはその MediaWiki コード (図 1(b)) を元に図 2 のような階層構造が抽出できる．定義文やカテゴリを上位下位関係の知識源とする場合，下位語は記事の見出し語に制限されるため，獲得できる下位語の数は Wikipedia の記事数より少なくなるが，階層構造ではそのような制限が無いため，より多様な上位下位関係が獲得できると期待される．

2.2 Wikipedia からの上位下位関係の獲得手法

本節では，既存の Wikipedia の定義文，カテゴリ，階層構造からの上位下位関係獲得手法を紹介する．なお，以下の研究は全て英語版 Wikipedia を利用して行われたものである．

Kazama らは，英語の固有表現抽出タスクのために，Wikipedia の記事の見出し語に対し，その冒頭の一文 (定義文) 中の特定の語彙統語パタンにマッチする表現を上位語として獲得している [4]．Herbelot らは，Wikipedia の記事の全文を意味解析し，定義文に対応する項構造を認識することで，約 88.5%の適合率で上位下位関係を獲得している [3]．また，Ruiz-Casado らは WordNet [5] を利用して学習された上位下位関係を記述パターンを用いて，69%の適合率で上位下位関係が獲得できたと報告している [7]．我々の提案手法では，Kazama らの手法を日本語版 Wikipedia に適応し，定義文から上位下位関係候補の獲得を行う．

一方，Suchanek [9] らは Wikipedia の各記事の見出し語に対し，記事に付与されたカテゴリのラベルを上位語として上位下位関係を獲得する手法を提案している．彼らは，英語特有の経験則を用いてカテゴリを選別し，外的知識として WordNet を利用することで，約 95%と高精度で大量の上位下位関係を獲得している．

最後に，我々がこれまでに提案した階層構造からの上位下位関係の獲得手法 [11, 15] について述べる．我々はこれまで，Wikipedia の階層構造が定義文やカテゴリよりも多くの適切な上位下位関係候補を含むことを示した [11, 15]．特に [15] では，階層構造中で子孫関係にある全てのノード対を上位下位関係候補とし，サポートベクタマシン (SVM) [13] を用いてフィルタリングすることで，約 90%の適合率で約 134 万件の上位下位関係を獲得できている．本研究では，この階層構造からの上位下位関係獲得手法を定義文やカテゴリから獲得した上位下位関係の選別に応用することで，WordNet などの外的な言語資源を用いることなく，機械学習のみで高精度の上位下位関係を Wikipedia から獲得することを試みる．

```
とは*<上位語>。
は.*<上位語>の一つ。
は.*<上位語>の代表的なものである。
は.*<上位語>のうちの一つ。
```

図 3: 定義文から上位語を抽出する語彙統語パターン

3 提案手法

我々の提案する上位下位関係の獲得手法は，まず Wikipedia の記事の定義文，カテゴリ，階層構造から上位下位関係候補を抽出する (Step1)．その後，得られた上位下位関係候補を SVM によりフィルタリングする (Step2)．以下で，各 Step について詳しく述べる．

3.1 Step1: 上位下位関係候補の抽出

Step1 では，Wikipedia の各記事に含まれる定義文，カテゴリ，階層構造から上位下位関係候補を抽出する．

定義文からの上位下位関係候補の獲得　まず，定義文から上位下位関係候補を抽出するために，Tsurumaru らの既存研究 [12] を参考に，人手で図 3 のような定義文から上位語を抽出する語彙統語パターンを 1,334 パターン用意した．図 3 で，<上位語>は任意の名詞連続にマッチする変数である．これらのパターンを，各記事の定義文 (第一文) に適用することで，見出し語を下位語，パターンで認識された名詞連続を上位語とする上位下位関係候補を獲得する．例えば，図 1(a) の記事の定義文「紅茶とは，摘み取った茶を乾燥させ，もみ込んで完全発酵させた茶葉。」に対して，パターン「とは*<上位語>。」を適用すると上位下位関係候補「茶葉/紅茶[*2]」が得られる．

カテゴリからの上位下位関係候補の獲得　次に，各記事に付与されているカテゴリを上位語，記事の見出し語を下位語として上位下位関係候補を獲得する．例えば，図 1(a) の記事からは,「茶/紅茶」,「喫茶文化/紅茶」という上位下位関係候補が得られる．

階層構造からの上位下位関係候補の獲得　最後に [15] と同様に，各記事の階層構造の各ノードと子孫関係にあるノードとの全ての組み合わせを上位下位関係候補として抽出する．例えば，図 2 の階層構造からは,「ブ

[*2]以降,「**上位語/下位語**」は上位下位関係の候補を指す

代表的な X，代表 X，主要な X，主な X，主要 X，基本的な X，基本 X，著名な X，大きな X，他の X，一部 X，代表的 X，基本的 X，著名 X，一部の X，X の一覧，X 一覧，X 詳細，X リスト，X の詳細

図 4: 冗長な上位語の簡略化のためのパターン

レンドティー/チャイ」や,「紅茶/リプトン」などの上位下位関係候補が抽出できる．さらに階層構造から抽出した上位下位関係候補については，冗長な上位語を簡略化するため，図 4 のパターンをもつ上位語候補からパターン中の X 以外の部分を取り除く．ここで，X は任意の文字列とする．例えば，上位語「主な紅茶ブランド」はパターン「主な X」を適用することで,「紅茶ブランド」と置換される．

以上の手続きで各知識源から獲得された上位下位関係候補には，明らかに誤りと見なせる下位語が含まれていたため，得られた上位下位関係候補の上位語または下位語が ‘※’ や ‘⇔’ などの特殊な記号を含んでいたり，文字化けしている場合には取り除いた．

3.2 Step2: SVM による上位下位関係候補のフィルタリング

Step2 では，Step1 で抽出した上位下位関係候補から SVM [13] を用いて誤りの候補を取り除く．具体的には各上位下位関係候補から生成した素性ベクトルを SVM に入力し，その結果得られた SVM のスコアが閾値以上の上位下位関係候補を正しい上位下位関係として獲得する．定義文とカテゴリから獲得された上位下位関係候補に対しては，同じ素性セットを用いた SVM を構築し，さらに階層構造から獲得された上位下位関係候補は，階層構造特有の素性も利用して SVM を構築した．以下で，各 SVM で共に用いた素性を説明した後，階層構造の SVM にのみに用いた属性を説明する．

表 2 に各 SVM の学習に用いた素性を示す．素性 POS, MORPH は，上位語と下位語がそれぞれ特定の品詞，形態素を含むとき発火する素性であり，素性 EXP は上位語と下位語がそれぞれ特定の文字列に一致するときに発火する素性である．素性 ATTR は上位語と下位語がそれぞれ属性語に一致するときに発火する．属性語とは，例えば,「紅茶」の属性は「生産量」や「産地」のように，事物の様々な観点を表現した語で，上位語または下位語になりにくい．属性判定に用いた属性リストについては [11] を参照されたい．最後に素性 LCHAR は上位語と下位語の末尾の 1 文字が「高**校**/公立**校**」のように同じであるとき発火し，末尾の 1 文字が共通する上位下位関係候補は，適切な関係であることが多いことを反映している．

一方，階層構造から獲得したノイズの多い上位下位関係候補を分類する SVM には，上記の素性に加えて PAT, LAYER, DIST の 3 つの素性を用いた．素性 PAT は，階層構造から取り出した上位下位関係候補の上位語が図 4 のパターンにマッチするときには適切な上位下位関係であることが多いという傾向を反映している．

表 2: Wikipedia から獲得した上位下位関係候補の分類に用いた素性

素性の種類	素性の発火条件
POS	上位語/下位語の末尾以外の形態素の品詞に X を含む 上位語/下位語の末尾の形態素の品詞が X
MORPH	上位語/下位語の末尾以外の形態素に X を含む 上位語/下位語の末尾の形態素が X
EXP	上位語/下位語が X
LCHAR	上位語と下位語の末尾の 1 文字が一致
ATTR	上位語/下位語が属性 X に一致
PAT	Step1 で上位語が図 4 のパターンに一致
LAYER	階層構造で上位語/下位語に付与された修飾記号が X
DIST	階層構造で上位語と下位語の間の距離が 2 以上 階層構造で上位語と下位語の間の距離が 1

注: 素性の発火条件に上位語/下位語とある場合，上位語，下位語それぞれについて別の素性を発火させることを意味する．

表 3: Wikipedia から獲得した上位下位関係候補

知識源	関係候補数	適合率	期待される適切な上位下位関係数
定義文	158,177	89.4%	141,410
カテゴリ	596,463	70.5%	420,506
階層構造	6,564,317	28.4%	1,864,266

次に，素性 LAYER は，階層構造で上位語と下位語に付与されていた修飾記号の種類に応じて発火する．最後に，素性 DIST は，抽出元の階層構造中での上位語候補と下位語候補の距離 (辺の数) に応じて発火し，抽出元の階層構造中で上位語候補と下位語候補が近いほど適切な上位下位関係であるという傾向を反映する．例えば，図 2 の場合,「紅茶/リプトン」の距離は 3 となる．

4 実験

提案手法の有効性を評価するため，2007 年 3 月の日本語版 Wikipedia の全記事から Wikipedia 内部向けの記事を取り除いた 276,323 記事に対して，提案手法を適用した．また，形態素解析には Mecab[*3]を，SVM は TinySVM[*4]を利用した．SVM のカーネルは 2 次の多項式カーネルを利用した．

まず Wikipedia の記事に Step1 を適用し，定義文から 158,177 件，カテゴリから 596,463 件，階層構造から 6,564,317 件の上位下位関係候補を獲得した．得られた上位下位関係候補から，各知識源ごとに 1,000 件ずつランダムに取り出してテストデータとした．テストデータにおける上位下位関係候補の適合率は表 3 のようになった．これより，ノイズの割合が増えるものの，定義文よりカテゴリの方が，また，カテゴリより階層構造の方がより多くの適切な上位下位関係を含むことが確認できる．

続いて，階層構造から獲得した上位下位関係候補のうち予め抽出したテスト用データを除いたものから訓

[*3] http://mecab.sourceforge.net/
[*4] http://chasen.org/~taku/software/TinySVM/

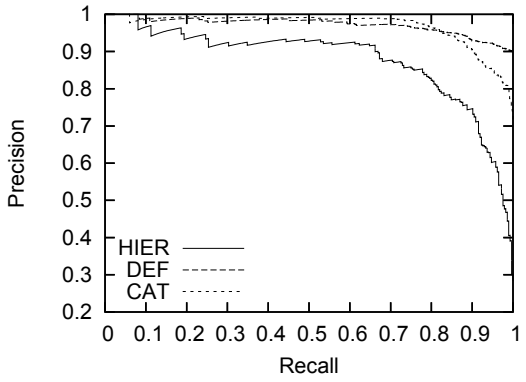


図 5: 上位下位関係候補のフィルタリングの P-R 曲線

表 4: 適合率が 90%以上になるように SVM のスコアの閾値を設定した場合の上位下位関係の獲得数

知識源	閾値	獲得関係数	適合率	期待される適切な上位下位関係数
定義文	-1.00	156,560	90.1%	141,061
カテゴリ	-0.33	420,586	90.1%	378,948
階層構造	0.36	1,349,622	90.0%	1,214,660
合計[*5]		1,885,502	>89.8%	>1,693,403

練データを取り出した．まず全体から 9,000 件，抽出元の階層構造中で上位語と下位語が直接の親子関係にあった候補から 9,000 件，図 4 のパターンにマッチしていた上位下位関係候補から 10,000 件をそれぞれランダムに取り出し，人手で正解をつけた．これらから重複を除いて得られた 29,900 件を訓練データとして用いた．

このようにして得られた全訓練データと 3.2 節で述べた素性を用いて，定義文とカテゴリから獲得した上位下位関係候補の分類のための SVM と，階層構造から獲得した上位下位関係候補の分類のための SVM を学習した．図 5 は，それぞれの SVM のスコアの閾値を変化させた際のテストデータにおける適合率と再現率関係を示したものである．階層構造から獲得した上位下位関係候補のみから訓練データを作成しているのにも関わらず，構築された SVM は定義文 (DEF) やカテゴリ (CAT) から獲得された上位下位関係候補を分類する際にも有効に働いていることが分かる．

最後に，それぞれの知識源から獲得した上位下位関係候補について，SVM のスコアの閾値を変化させ，テストデータでの適合率が 90%を超えるように調整した結果を表 4 に示す．適合率 89.8%[*5]以上で，約 188 万の上位下位関係を獲得できた．なお，上位下位関係の語彙数 (上位語と下位語のを合わせた異なり語数) は，933,832 語であった．また，閾値を上げて適合率を 95%以上にした場合でも，983,126 件の上位下位関係が獲得できることが分かった．

[*5]獲得関係数は重複を除いた関係数を示す．期待される適切な上位関係数を計算するためには，それぞれの知識源から獲得された適切な上位下位関係がどの程度重複しているか見積もる必要があるが，ここでは重複が最も大きかった場合，すなわち，複数の知識源から獲得された上位下位関係が全て正解だったとみなして，期待される適切な上位下位関係数の最小見積もりを計算した．

5 まとめ

本稿では，Wikipedia の定義文，カテゴリ，階層構造をを知識源とし，既存手法 [11] を改良した上位下位関係獲得手法を提案した．実験では，日本語版 Wikipedia 約 28 万記事から，約 188 万件の上位下位関係を適合率 89.8%以上で獲得することに成功した．Wikipedia のサイズが日進月歩で増えていることを考えると非常に有望な結果だと言えよう．

今後の課題としては，実アプリケーション [14] への応用を通して獲得された上位下位関係の評価を行うことが挙げられる．

参考文献

[1] O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*, Vol. 165, No. 1, pp. 91–134, 2005.

[2] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proc. of COLING*, pp. 539–545, 1992.

[3] A. Herbelot and A. Copestake. Acquiring ontological relationships from wikipedia using rmrs. In *Proc. of the ISWC 2006 workshop on Web Content Mining with Human Language Technologies*, 2006.

[4] J. Kazama and K. Torisawa. Exploiting wikipedia as external knowledge for named entity recognition. In *Proc. of EMNLP-CONLL*, pp. 698–707, 2007.

[5] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database. In *Journal of Lexicography*, pp. 235–244, 1990.

[6] P. Pantel and M. Pennacchiotti. Espresso: leveraging generic patterns for automatically harvesting semantic relations. In *Proc. of COLING-ACL*, pp. 113–120, 2006.

[7] M. Ruiz-Casado, E. Alfonseca, and P. Castells. Automatic extraction of semantic relationships for wordnet by means of pattern learning from wikipedia. In *Proc. of NLDB*, pp. 67–79, 2005.

[8] K. Shinzato and K. Torisawa. Acquiring hyponymy relations from web documents. In *Proc. of HLT-NAACL*, pp. 73–80, 2004.

[9] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A core of semantic knowledge unifying wordnet and wikipedia. In *Proc. of WWW*, 2007.

[10] A. Sumida, K.Torisawa, and K.Shinzato. Concept-instance relation extraction from simple noun sequences using a search engine on a web repository. In *Proc. of the ISWC 2006 workshop on Web Content Mining with Human Language Technologies*, 2006.

[11] A. Sumida and K. Torisawa. Hacking wikipedia for hyponymy relation acquisition. In *Proc. of IJCNLP*, pp. 883–888, 2008.

[12] H. Tsurumaru, T. Hitaka, and S. Yoshida. An attempt to automatic thesaurus construction from an ordinary japanese language dictionary. In *Proc. of COLING*, pp. 445–447, 1986.

[13] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[14] 鳥澤健太郎, 隅田飛鳥, 野口大輔, 風間淳一. 自動生成された検索ディレクトリ「鳥式」の現状. 言語処理学会第 14 回年次大会発表論文集, 2008.

[15] 隅田飛鳥, 吉永直樹, 鳥澤健太郎. Wikipedia の階層構造を知識源とする上位下位関係の自動獲得. 第 70 回情報処理学会全国大会講演論文集, 2008.