Wikipedia の階層構造を知識源とする上位下位関係の自動獲得

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

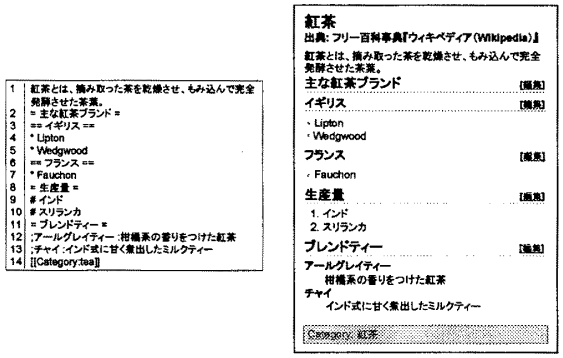
1 はじめに

本稿では，Wikipedia の階層構造を知識源として，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おり，これまでにも様々な上位下位関係の抽出手法が開発されてきた [2, 1, 3]．本研究では，Wikipedia の階層構造から上位下位関係を獲得する手法 [4] をより多くの上位下位関係を獲得できるように改良し，約 134 万個の上位下位関係を適合率 90%で獲得することができた．

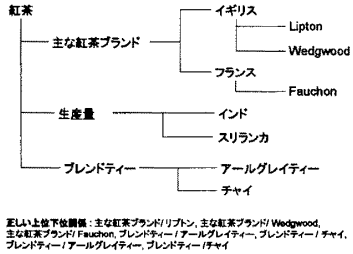
2 Wikipedia の階層構造

提案手法について述べる前に，本研究で上位下位関係の知識源として用いる Wikipedia の記事の階層構造について簡単に説明する．

Wikipedia は誰でも自由に書き込める大規模な百科事典であり，その記事は HTML より明確な構造をもつ MediaWiki 構文により記述される．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順位と長さに従ってノードを配置する．例えば，図 1(b) のページからはそのソース図 1(a) を元に図 1(c) のような階層構造が抽出される．



(a) ソース (b) スクリーンショット



(c) 階層構造

図 1: 「紅茶」に関する Wikipedia の記事の例

表 1: 階層構造に関する修飾記号

優先順位	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*	* Lipton

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．

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 2: 冗長な上位語の簡略化のためのパターン

3 提案手法

我々がこれまでに開発した既存手法 [4] では，Wikipedia の記事の階層構造から直接の親子関係にあるノードを上位下位関係候補として抽出し，SVM [5] を用いて正しい上位下位関係を獲得していた．本研究では，より多くの上位下位関係を獲得するために，Wikipedia の記事の階層構造から全ての祖先・子孫関係にあるノードを上位下位関係候補として抽出する (Step1)．その後，得られた上位下位関係候補を新しい素性を用いた SVM によりフィルタリングをする (Step2)．

以下で，各ステップ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階層構造から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 1(c) の階層構造からは，「ブレンドティー/チャイ」[*1]や，「紅茶/Lipton」などの上位下位関係候補が抽出できる．この際，冗長な上位語の簡略化のため，図 2 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．

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 を用いて誤りの候補を取り除く．具体的には各上

Hyponymy Relation Acquisition from Hierarchical Layouts in Wikipedia
Asuka Sumida*, Naoki Yoshinaga†, Kentaro Torisawa*
*Japan Advanced Institute of Science and Technology
†Japan Society for the Promotion of Science

[*1] 以降，「上位語/下位語」は上位下位関係の候補を指す

表 2: 本研究で新たに追加した素性

素性の種類	素性が発火する条件
DIST	上位語と下位語間の距離が 2 以上である 上位語と下位語間の距離が 1 である
PAT	Step1 で上位語が図 2 のパターンのいずれかに一致
LCHAR	上位語と下位語の末尾の 1 文字が一致

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．

素性として既存手法 [4] では上位語候補と下位語候補の品詞，形態素，表記，抽出元のノードに付与されていた修飾記号に加え，上位語候補あるいは下位語候補が属性かどうかの情報を利用した．本研究では以上の素性に加え表 2 の素性を追加する．素性 DIST は上位語候補と下位語候補の距離が抽出元の階層構造中で短いほど適切な上位下位関係となりやすいという我々の観察結果に基づく．ここで距離は階層構造中で上位語と下位語間に存在する辺の数とする．例えば，図 1(c) の場合,「ブレンドティー/チャイ」は距離 1,「紅茶/Lipton」の距離は 3 となる．次に素性 PAT は，Step1 の時点で図 2 のパターンにマッチしていた上位下位関係候補は適切であることが多いという我々の観察結果を反映するための素性である．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複合語は意味的に似た語が多く，適切な上位下位関係になりやすいことを反映している．

3.3 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 820,074 記事から Wikipedia 内部向けの 543,751 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[*2]を，SVM は TinySVM[*3]を利用した．SVM のカーネルは次数 2 の多項式カーネルを利用し，閾値は 0 とした．

まず Wikipedia の記事に Step1 を適用し，6,564,317 の上位下位関係候補を抽出した．この中からランダムに選んだ 1,000 件の上位下位関係候補を人手で正解をつけ，これをテストデータとした．次に残りの上位下位関係候補から 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 2 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．

表 3 に提案手法と既存手法 (S&T [4]) とを比較した結果を示す．表 3 の各列は左から順に適合率，上位下位関係数，およびこれらより求めた期待される正しい上位下位関係の数を示す．これより提案手法では既存手法に比べ獲得できる上位下位関係数，適合率ともに大幅に向上していることがわかる．

次に，Step2 で利用した各素性ごとの精度の比較を表 4 に示す．表 4 の各列は左から順に精度，適合率，再現率，F 値を表す．本研究で提案した素性セット (ALL)

[*2] http://mecab.sourceforge.net/

[*3] http://chasen.org/~taku/software/TinySVM/

表 3: 獲得した上位下位関係の適合率の比較

獲得手法	適合率	上位下位関係数	期待される正しい上位下位関係数
既存手法 S&T [4]	76.4	633,122	484,117
提案手法 (Step1)	28.4	6,564,317	1,864,266
(Step2)	**85.2**	1,738,500	1,481,400

表 4: 新たに追加した素性の効果

素性の種類	精度	適合率	再現率	F 値
ALL−DIST	89.3	83.9	77.1	80.4
ALL−PAT	89.5	83.8	78.2	80.9
ALL−LCHAR	88.9	85.0	73.9	79.0
ALL−DIST−LCHAR−PAT	88.6	82.0	76.8	79.3
ALL	**89.7**	**85.2**	77.1	**81.0**

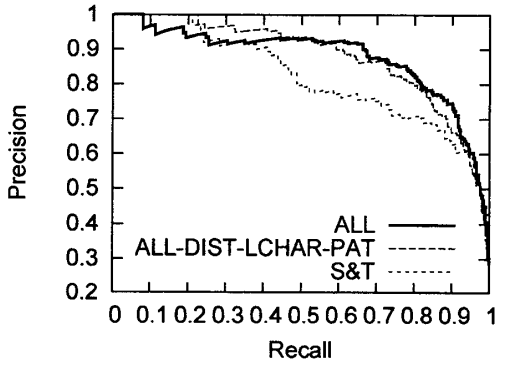


図 3: 適合率と再現率とのトレードオフ

により，既存手法で提案された素性セット (ALL-DIST-LCHAR-PAT) に比べ F 値が 1.7%向上した．

最後に適合率と再現率とのトレードオフの関係を図 3 に示す．横軸は再現率，縦軸は適合率を表す．このグラフより，SVM の閾値を大きくすることで，より信頼性の高い上位下位関係を獲得できることがわかる．例えば，SVM の閾値を 0.36 とすると約 134 万の上位下位関係を適合率 90%で獲得できる．

4 まとめ

本稿では，Wikipedia の階層構造を知識源とし，既存手法 [4] を改良した上位下位関係獲得手法を提案した．実験では，約 134 万の上位下位関係を適合率 90%で獲得することに成功した．

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

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

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