社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2003－ＮＬ－158　(15)
2003／11／7

HTML文書からの単語間の上位下位関係の自動獲得

新里　圭司　　鳥澤健太郎

† 北陸先端科学技術大学院大学 情報科学研究科　〒923-1292 石川県能美郡辰口町旭台 1-1
E-mail: †{skeiji,torisawa}@jaist.ac.jp

あらまし　本稿では, 単語の上位下位関係をWWW上のドキュメントより自動獲得する手法を提案する. 従来より, 単語の上位下位関係は自然言語処理において重要な知識であると見なされており, 多くの自動獲得手法が提案されてきた. それらの多くは, 名詞句の並置などの文の表層のパターンに注目するものがほとんどであった. 本稿で提案する手法は, これらとは異なるアプローチをとる. より具体的には,1) Web上にあるHTMLタグの繰り返しパターン,2) 従来情報検索などで使われてきたDF, IDFなどの統計量,3) 名詞が持つ主として動詞との係り受け関係の三種の情報を組み合わせることで, 単語の上位下位関係を自動的に獲得することを目指す.

キーワード　知識獲得,　上位語,　下位語,　統計的自然言語処理,　WWW

Automatic acquisition of hyponymy-relations from HTML documents

Keiji SHINZATO and Kentaro TORISAWA

† Graduate School of Information Sciences,
Japan Advanced Institute of Science and Technology,
1-1 Asahidai, Tatsunokuchi-machi, Nomi-gun, Ishikawa, 923-1292 JAPAN
E-mail: †{skeiji,torisawa}@jaist.ac.jp

Abstract　This paper describes an automatic acquisition method for hyponymy relations. Hyponymy relations play a crucial role in various natural language processing systems, and there have been many attempts to automatically acquire the relations from large-scale corpora. Most of the existing acquisition methods rely on particular linguistic patterns, such as juxtapositions, which specify hyponymy relations. Our method, however, does not use such linguistic patterns. We try to acquire hyponymy relations from three different types of clues. The first is repetitions of HTML tags found in usual HTML documents on the WWW. The second is statistical measures such as DF and IDF, which are popular in IR literatures. The third is a verb-noun co-occurrences found in normal corpora.

Key words　Knowledge acquisition,　Hypernym,　Hyponym,　Statistical Natural Language Processing,　World Wide Web

1. はじめに

近年, 膨大な量の文書が計算機で扱えるようになり, 多種多様な自然言語処理技術が利用されるようになってきた. しかし, より知的で高度な処理を行うためには, 単語間の上位下位関係 (hyponymy relation), 類似関係 (synonymy relation), 包含関係 (part-whole relation) などの知識がまだまだ不足しており, このような知識の獲得は今後ますます重要なものになるといえる. そこで本稿では, Web上にある大量のHTML文書から単語間の上位下位関係を自動的に獲得する手法について述べる. Web上のHTML文書を対象としたのは, 新聞記事などの他のコーパスとくらべ, (1) 量が豊富にあり新しい情報が掲載されるのが早い, (2)HTML文書製作者の何らかの意図に基づいて文書がタグ付けされており, 単語間の上位下位関係を獲得するという今回の目的にはその情報が使えるのではないか, と考えたためである.

Miller [10] の定義に従えば, 単語Aが単語Bの上位語 (hypernym) である (または, 単語Bが単語Aの下位語 (hyponym) である) とは, "B is a (kind of) A" が言える時であると定義されており, 本研究でもこの定義に従う. また単語Aが単語Bの上位語であるということを次の形式で記述する.

$$hypernym(A, B)$$

例えば, 茄子と野菜, 秋刀魚と魚, 冷蔵庫と機械の間には次のよ

うな関係が成り立つ.

hypernym(“野菜”, “茄子”*)*

hypernym(“魚”, “秋刀魚”*)*

hypernym(“機械”, “冷蔵庫”*)*

このような単語間の上位下位関係は他の自然言語処理アプリケーションに対して有用である. 例えば, 情報検索における検索質問拡張では, 検索語に加え, 検索語の類義語, 上位語, 下位語を付け加えて検索することで, 再現率が向上することが報告されている [7]. これは, 特許検索等の検索に漏れがあっては困るようなシステムに, 単語の上位下位関係が有効であること示している. また, 質疑応答システムにおいても, 「ニューヨーク市の市長は誰か」や, 「ナディア・コマネチは誰か」といった類の質問に答える際に単語間の上位下位関係は利用されている [1].

2. 先行研究

これまでに単語間の上位下位関係の獲得について多くの研究が行われてきた. しかし, そのほぼ全てが, 新聞記事などのコーパスから構文パターンのマッチングにより上位下位関係の獲得を行うものとなっている [1,2,4,5,8,9]. 例えば, あらかじめ

$$such\ NP\ as\ \{NP,\}*\ (or|and)\ NP$$

というパターンを用意しておき,

...works by such authors as Herrick, Goldsmith, and Shakespeare.

という文にパターンを適用すると

hypernym(author, Herrick),

hypernym(author, Goldsmith),

hypernym(author, Shakespeare)

という単語間の上位下位関係を得ることができる. しかし, このようなパターンによる方法は, パターンと適合する文数がコーパス中に少ないため, 大量の上位下位関係を取得するのが難しいという問題がある. この問題を解決するために, 大量の上位下位関係を獲得するためのパターンを生成する手法も提案されてはいる [9] が, 上位下位関係を表すパターンの数には自ずから限界があると考えられる. 我々はこのような点を考慮して, パターンによらない上位下位関係獲得手法を開発することにした.

パターンによる上位下位関係の獲得を日本語の新聞記事に対して行った研究として今角 [4] がある. 今角は新聞記事 4 年分 (約 232 万文) のテキストから, 同格, 並列関係をもつ文を抽出し, パターンマッチを行うことで, およそ 15000 件の上位下位関係が獲得でき, そのうち 600 個について人手で評価したところ精度が 77.2[%] であったと述べている.

3. 提案手法

提案手法は, 以下の仮説に基づいて上位下位関係を獲得する.

[仮説 1] HTML 文書に現れる箇条書きなどの繰り返しパターンに現れる単語, 表現は意味的に類似しており, 共通の上位語を持ちやすい.

[仮説 2] 仮に共通の上位語を持つ, 単語, 表現の集合が与えられたとすると, それらの単語 (の少なくとも一つ) を含む文書群には, 共通して共通の上位語が現れやすい. また, それ以外の文書群には上位語が比較的含まれにくい.

[仮説 3] 上位語と下位語は意味的に類似しており, この類似性は, 上位語と下位語の持つ係り受け関係によって捕らえることができる.

たとえば, パソコン関係の HTML 文書などには, よく対応 OS などと称して以下のような箇条書きが現れる.

- Windows 2000
- Windows XP
- Mac OS X
- Linux redHat 9
- Linux redHat 7.2

仮説 1 によれば, これらは共通の上位語を持ちやすいことになり, 下位語の候補としてあつかっていいことになる. また, 仮説 2 によれば, これらの下位語候補を含む文書を大量に持ってくると, それらには共通して上位語「OS」や「ソフトウエア」が含まれやすいことになる. したがって, この「共通して現れること」をスコアとして定式化できれば, それによって上位語をある程度の精度で求めることができることになる. しかしながら, 共通して現れる単語は, なにも上位語に限らず, 下位語と共通して関連の強い単語も共通して出現しやすい. たとえば, 上記の例であれば,「パソコン」「Pentium」「Internet」なども共通して出現しやすいことになるだろう. しかし, 仮説 3 によって, このような強い関連を持つ語, 表現の中から, 上位語を絞り込むことが可能となる. 例えば,

「先日, Windows をバージョンアップした.」

という文の「Windows」の部分を, その上位語である「OS」や「ソフトウェア」などに置き換えても, それらの語と「バージョンアップする」との係り受け関係は意味的に適切である. しかし, 上位語ではないが「Windows」と関連の強い「パソコン」や「Pentium」,「Internet」などの語で置き換えると, 文の意味は通らなくなる. このような, 上位語候補が持つ係り受け関係での選好の差を類似度として捉えられれば, 下位語候補を含む文書に共通して現れる語から上位語を絞ることができると考えられる.

本手法は, 大きくわけて 3 つの手順からなる. 各々の手順は, 上記の仮説の各々に対応している. まず HTML 文書から下位語候補のリストを抜き出す (ステップ 1) . そして, それら下位語候補に関する HTML 文書を検索エンジンにより収集し, それらに含まれる各単語のスコアを計算し上位語の候補を発見する (ステップ 2). 最後に, 得られた上位語の候補と各下位語の候補の類似度をはかることで各下位語候補に共通の上位語の獲得を行う (ステップ 3). 本節では以降, 各ステップについて詳しく述べる.

3.1 下位語候補リストの獲得 (ステップ 1)

以下では図 1 に示した HTML 文書の一部を例に, 下位語候補リストの獲得アルゴリズムについて述べる. 本手法ではまず HTML 文書中に現れる表現の「パス」を求める. ここでいう

「パス」とは, テキストがどのようにタグ付けされているかを表すものであり, 大ざっぱにいえば, ある表現がどのようなタグで囲まれているのかをそのネストの順序にしたがって, タグのリストで表したものである. 図1の文書中の各葉ノードは以下に示すパスを持っている. たとえば, 表現「CD」はタグ<LI>,</LI>に囲まれており, さらに<UL>,</UL>にも囲まれている. これらのタグを,「CD」を囲む順序にしたがって並べれば,「CD」のパス (UL, LI, CD) が得られることになる.

(UL, LI, CD)
(UL, UL, LI, ロック・ポップス)
(UL, UL, LI, R & B)
(UL, UL, LI, ヒップホップ)
(UL, LI, テレビゲーム)
(UL, UL, LI, プレイステーション)
(UL, UL, LI, ドリームキャスト)

本手法では, このパスを基にして下位語候補のリストを抽出する. より具体的には, 共通のパスを持つ語, 表現を下位語候補のリストとして抽出する。しか

しながら, ただ単に共通のパスを持つ要素だけを集めてくると意味的に類似した下位語候補の獲得は行えない. 図1の場合, 同じパスを持つ要素をそれぞれまとめると,

[ロック・ポップス, R & B, ヒップホップ, プレイステーション, ドリームキャスト]
[CD, テレビゲーム]

という下位語候補のリストが得られるが, ゲーム機と音楽が同じリストに含まれてしまっている. この原因は同一タグの異なる出現を区別できていないからである. そこで, タグにその出現順序を付与しこのタグの出現順序を含めてパスを求める. 図1の場合だと,

(UL#1, LI#1, CD)
(UL#1, UL#2, LI#1, ロック・ポップス)
(UL#1, UL#2, LI#2, R & B)
(UL#1, UL#2, LI#3, ヒップホップ)
(UL#1, LI#3, テレビゲーム)
(UL#1, UL#4, LI#1, プレイステーション)
(UL#1, UL#4, LI#2, ドリームキャスト)

というようなパスが得られる. ここで#数字はタグの出現順序を表している. しかし, 今度はどのパスも一意になってしまい, 同じパスを持つ要素を得ることができなくなる. そこで, 葉ノードからN代前の先祖まではパスの中に順序を含めないというようにする. 図1の場合, $N=1$ とすると

(UL#1, LI, CD)
(UL#1, UL#2, LI, ロック・ポップス)
(UL#1, UL#2, LI, R & B)
(UL#1, UL#2, LI, ヒップホップ)
(UL#1, LI, テレビゲーム)
(UL#1, UL#4, LI, プレイステーション)
(UL#1, UL#4, LI, ドリームキャスト)

のようなパスを得ることができ, 同じパスを持つ葉ノードごとにまとめると,

[ロック・ポップス, R & B, ヒップホップ]
[プレイステーション, ドリームキャスト]
[CD, テレビゲーム]

というように意味的に類似した下位語候補リストを得ることが可能になる.

3.2 上位語候補の獲得 (ステップ2)

ステップ2では, ステップ1により獲得された各下位語候補を, 検索語としてサーチエンジンに問い合わせ, 各下位語候補に関連するHTML文書を収集し, その中から各下位語候補共通の上位語候補を獲得する. ステップ2の動作は前述の仮説2に基づくが, この仮説2は以下の2つの関係に分解できる.

[関係1] 上位語は下位語候補に関連のある文書に現れやすい
[関係2] 上位語は下位語候補に関連のない文書には現れにくい

これらの関係を踏まえ, 各下位語候補に関連のある文書群に含まれる単語 i のスコアを, 情報検索で用いられる df, idf といったスコア [11], [12] を用いて, 式1により求める.

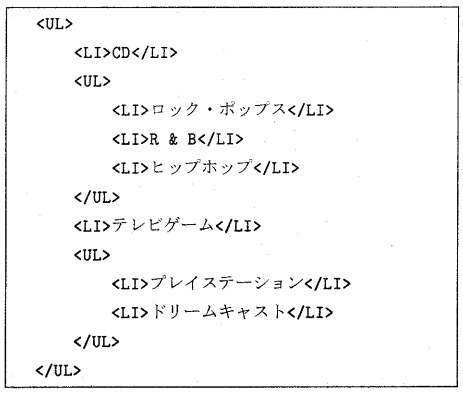

図1 HTML文書の一部
Fig. 1 An example of HTML document

$$score_i = df_{i_{local}} \cdot idf_{i_{global}} \tag{1}$$

$$idf_{i_{global}} = \log \frac{N}{n_i}$$

n_i:HTML文書N件中で単語 i を含む文書数

$df_{i_{local}}$ は各下位語を検索語として得られたHTML文書群中で, 単語 i を含む文書数であり, 上で述べた関係1に対応する. また, $idf_{i_{global}}$ は下位語候補を含んでいる, いないを問わず大量のHTML文書N件より得た単語 i の文書頻度の逆数であり, 関係2に対応している. この値は単語 i が各下位語に関連のある文書にだけ現れるような場合に大きくなり, 反対に各下位語候補に関連のない他の多くの文書に出現する程小さくなる.

3.3 尤もらしい上位語の選択 (ステップ3)

ステップ3ではステップ2より獲得された上位語の候補の中から, 仮説3に基づきステップ1により獲得された下位語候補全体との意味的な類似度を計算し, 尤もらしい上位語の候補を選択する. 下位語候補全体との類似度をはかる理由は, 下位語候補単独ではその出現頻度が相対的に小さいため, 下位語候補に関する種々の係り受け関係を獲得することができず, 下位語候補の係り受け関係を的確に捉えているとはいえないからである.

各上位語候補と下位語候補全体の類似度を計算するために, 各上位語候補と下位語候補全体のそれぞれの係り受け関係をベクトルで表現する. そして, 互いのベクトルの類似度を, 上位語候補と下位語候補全体の意味的な類似度として用いる. 上位語候補$_i$ の係り受け関係を表現したベクトル $\overrightarrow{hyper_i}$ と下位語候補全体の係り受け関係を表現したベクトル $\overrightarrow{hypos}$ は以下のよ

うに表される．ここで $\{t_1, t_2, \cdots, t_m\}$ は，m 個の下位語候補をあらわし，$\{h_1, h_2, \cdots, h_o\}$ は o 個の上位語候補を表すものとする．また，$\{s_1, s_2, \cdots, s_N\}$ は，動詞の格要素を表す．すなわち，s_i は動詞 v と助詞 p の組 $\langle v, p \rangle$ であらわすことができるが，これは動詞 v に対して助詞 p で係る格要素を表す．

$$\overrightarrow{hypos} = (k_1, k_2, \ldots, k_N), \tag{2}$$

$$k_n = \sum_{i=1}^{m} f(t_i, s_n) \tag{3}$$

$$\overrightarrow{hyper_j} = (l_1, l_2, \ldots, l_N), \tag{4}$$

$$l_n = f(h_j, s_n) \tag{5}$$

N:格要素の総数
m:下位語候補の数

関数 $f(t, s)$ は単語 t（下位語候補もしくは上位語候補）が格要素 s に現れる回数を求める関数である．

ベクトルの類似度を計算するには幾つかの方法があるが，本研究では文書検索の分野などでよく用いられているコサイン尺度 [11], [12] を用いる．コサイン尺度による 2 つのベクトル $\vec{x}$ と $\vec{y}$ の類似度は以下の式で計算でき，値が 1 に近い程 2 つのベクトルは類似している．

$$sim(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{T} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{T} {x_i}^2 \times \sum_{i=1}^{T} {y_i}^2}} \tag{6}$$

4. 実　　験

4.1 下位語候補リストの獲得（ステップ 1）

Web より収集した約 80 万ページの HTML 文書に対し，節 3.1 で述べた方法を適用する前に，以下に述べる前処理を施した．

Tidy による HTML 文書の整形　Web 上に公開されている HTML 文書には，HTML タグの入れ子関係が間違っていたり，終了タグが省略されてしまっているものがある．そのような文書からは繰り返し出現するタグのパターンを認識することが難しいため，HTML 文書中の誤りを自動的に修正するフリーのユーティリティである Tidy(注1) を使用し，HTML 文書を well-formed な XML 文書に変換した．

テーブルデータに関する処理　HTML 文書中の表データも下位語候補リストを作成する上で重要なデータとなる．しかし，節 3.1 で提案した手法を HTML 文書中の表データに適用すると，表データの行方向に関して下位語候補を獲得することになる．しかし，ブラウザにより HTML 文書を閲覧する場合，横方向より縦方向に閲覧していく機会の方が多いため表データ中の類似したデータも列方向に並ぶと予想される．そのため HTML 文書中の表データを転置することで節 3.1 で述べた手法により表の列方向のデータの獲得が行えるようにした．

以上の前処理を施した後，節 3.1 で述べた手順により，下位語候補リストの獲得を行った．下位語候補リストを獲得する際，タグの出現順序を考慮しない先祖の数は 2 とした．獲得できた下位語候補リストの要素で，表 5 に示す正規表現にマッチするものは，下位語である可能性が低いため削除した．さらに，節 3.1 で述べた手順では，単語に限らず文も獲得されてしまうので，下位語候補の文字数が 12 文字以上の物は文であると判断し下位語候補リストから削除した．さらに金額や日付などの下位語候補リストができてしまうのは望ましくないため，下位語候補に含まれる数字は削除した．不要語と文字数による削除を行った上で，下位語候補リストの要素数が 4 個未満のものは妥当な下位語候補リストではないとして破棄した．その結果獲得できた下位語候補の数は 78040 個であった．しかし，これには同一の要素からなる下位語候補リストもあるため，それらを排除すると 51651 個であった．実際に獲得できた下位語候補リストの例を表 1 に示す．例の内，いくつかの下位語候補リストは，本来の意味での下位語のリストとは呼べず，このプロセスでのエラーを含んでいることに注意されたい．

表 1　実際に獲得できた下位語候補リスト
Table 1　Examples of hyponym candidates list

CLASS 51606
ソノラマ文庫, 集英社コバルト文庫, 富士見ファンタジア文庫 富士見ミステリー文庫
Class 51608
すべて, ウェブ, サウンド, 画像
Class 51609
つまらない, 座布団枚やってくれ, 別に, 面白い
CLASS 51610
エゾシカ, エゾリス, キタキツネ, シマリス, ナキウサギ, 旅日記
Class 51612
アッサム, アッサム CTC, ダージリン, ニルギリ

4.2 上位語の獲得

次いで上位語獲得の実験について述べる．以下で述べる実験において，各下位語候補リストに対して妥当な上位語が獲得できているかどうかの評価は，下位語候補リストの 7 割以上の要素に対して獲得された上位語候補が下位語候補リストの上位語としてふさわしい場合に正解であるとし，そうでない場合は不正解であるとして判断した．また，単語毎の上位下位関係獲得の精度についても報告する．

4.2.1 $df \cdot idf$ 法による上位語候補の獲得（ステップ 2）

まず，$df \cdot idf$ 法を用いたステップ 2 のみでの性能を評価した．実験データとして，節 4.1 より獲得した下位語候補リスト群の中から，ランダムに抽出した 1000 個の下位語候補リストを用い，それらに対して上位語候補の獲得を行った．要素数が 10 個以上の下位語候補リストに関しては，リストのなかからランダムに 10 個要素を選び出し，それを改めて下位語候補リストとした．実際に下位語候補に関連するドキュメントを収集してくる際，サーチエンジンとして goo(注2) を利用し，各下位語候補ごとに収集してくるドキュメントの数は 100 件とした．しかし，goo によるヒット件数が 100 件に満たない下位語候補に関しては，

(注1)：http://www.w3.org/People/Raggett/tidy/から入手可能

(注2)：http://www.goo.ne.jp/

全て収集した．また形態素解析器として JUMAN[注3] を用いた．

実際に獲得できた上位語候補のうち，名詞，未知語以外の単語は上位語候補から削除した．また名詞であっても副詞的名詞，形式名詞，数詞，時相名詞，固有名詞であるもの，未知語であってもその他に分類されるものは，上位語にはなりにくいと考えたため削除した．さらに，表 6 にあげる不要語リストに含まれるものは削除した．この不要語リストは，予め 100 万件の Web ページより求めた文書頻度の高い普通名詞と未知語，及び一般的すぎて上位語としてはふさわしくないだろうと考えられる単語からなる．

提案手法の有効性を示すために，提案手法の他に従来より単語の重み付けによく用いられる $tf \cdot idf$ 法について実験を行い，結果を比較した．その結果を図 2 に示す．このグラフを作成するにあたって，下位語候補リストをそれから求められた上位語候補 (もっとも高いスコアを持つ上位語候補) のスコアでソートした．ここでスコアとは，提案手法に関しては $df \cdot idf$ の値，$tf \cdot idf$ 法については，$tf \cdot idf$ の値を指すものとする．ついで，スコアのもっとも高い下位語候補リストを N 個取ってきたとき，上位語候補のトップに来ているものの可否を人手で判断し，その精度 (precision) を求めた．この N 個を横軸に，また，精度を縦軸にとってプロットしたのが，図 2 のグラフである．このようなグラフを作成した動機は，高いスコアを持つ上位語候補をもつ下位語候補リストは，意味的な共通性をもっており，また，正しい上位語が求められる可能性が高いという仮説を立てたからである．この仮説は，グラフが右肩下がりであることからある程度確認されたことになる．

また，実験では下位語候補リストを 1000 個抽出したが，評価の手間がかかることと，予備実験の結果，上位語候補のスコアが低いものは，上位語獲得の精度が極端に低下することがわかったため，1000 個のリストの内，スコアの上位 100 程度のみを用いることとした．

結論として，図 2 より $tf \cdot idf$ 法と比べ $df \cdot idf$ 法の方が精度が高いことがわかる．これは，仮説 2 で述べた「上位語は各下位語候補を含む文書に共通して現れている」という上位語の特性を $df \cdot idf$ 法の方がより反映しているためだと考えられる．また $df \cdot idf$ 法により実際に獲得された上位語の例を表 2 に示す．

さらに，$df \cdot idf$ 法については，各下位語候補リストごとに求めた上位語のうち，スコアの高いもの上位 n 個の中に正しい上位語がどの程度含まれているか，n を 1 から 5 まで変化させた時の上位語の含有率を調べた．その結果を図 3 に示す．図 3 より n の数を増やせば含有率は上がるが，精度向上の割合は上位 1 と 2 の時が最も大きく，それ以降の変化は減少している．

4.2.2 係り受け関係を考慮した上位語候補の獲得 (ステップ 3)

前節で述べた $df \cdot idf$ 法に加えて係り受け関係を考慮すること，即ちステップ 3 を導入することで性能が向上するかどうか実験した．実験データとして，節 4.1.1 で獲得した上位語候補のうち，$df \cdot idf$ スコアの上位 M 個について $df \cdot idf$ スコアと類似度の積を計算し，その値を各上位語候補の新たなスコアとした．M を 1,2,5 としたそれぞれの場合について，上位語獲得の精度の評価を行った．

ここで，上位語候補の係り受けベクトル $\overrightarrow{hyper_i}$ は，新聞記事 33 年分 (3.01GB) から，また，下位語候補の係り受けベクトル $\overrightarrow{hypos}$ は，節 4.2.1 でのべたように検索エンジン goo を利用してダウンロードしたものから得ている．係り受け解析に関しては，既存の構文解析器 [6] を利用した．[注4]

実験結果を図 4 に示す．図 2 の場合と同様に，下位語候補リストをそれから求められた上位語候補 (もっとも高いスコアを持つ上位語候補のスコア) でソートし，その上位 N 個を取ってきて精度を測定している．これにより N=2 とした時の精度が最

表 2 $df \cdot idf$ 法により獲得した上位語の例
Table 2 Results obtained by the $df \cdot idf$ method

CLASS 116						
ぶっすま, 堂本兄弟, USO？ジャパン						
巨人中毒, 金曜日のスマたちへ, 特上！天声慎吾						
1	番組	(704.606)	○	6	テレビ	(463.819)
2	中居	(681.666)		7	剛	(460.814)
3	SMAP	(551.070)		8	出演	(454.898)
4	放送	(508.821)		9	慎	(451.685)
5	吾	(472.595)		10	正広	(429.327)
CLASS 6						
キエボ, ペルージャ, ピアチェンツァ , パルマ						
レジーナ, ユベントス, インテル, ラツィオ, ブレシア						
1	セリエ	(641.682)		6	選手	(457.377)
2	試合	(554.879)		7	サッカー	(431.401)
3	イタリア	(527.822)		8	ローマ	(377.709)
4	チーム	(524.874)	○	9	移籍	(367.771)
5	中田	(490.872)		10	10 ミラン	(365.391)
CLASS 45						
携帯電話, 電話帳, インターネット電話, 電報						
国際電話, 電話加入権, PHS 電話						
1	電話	(753.098)		6	利用	(528.679)
2	サービス	(617.945)		7	インターネット	(479.701)
3	NTT	(587.682)		8	回線	(442.853)
4	料金	(581.762)		9	PHS	(386.041)
5	通話	(572.606)		10	接続	(385.680)
CLASS 886						
彼の気持ち, このままは辛い, いきなりフラレタ事						
結婚できない, 結婚生活, 歳の彼女, 苦しくてボロボロです						
告白したいけど, 分かりません, さみしくて逆上した私						
1	結婚	(665.532)		6	恋愛	(480.541)
2	気持ち	(608.339)		7	告白	(459.929)
3	相手	(547.287)		8	意味	(431.618)
4	こと	(482.660)		9	顔	(426.914)
5	言葉	(480.614)		10	俺	(413.293)

○は正解を表す。下位語候補のリストの内 7 割をこえる語に対して適切な上位語であれば正解としている．

(注3)：http://www.kc.t.u-tokyo.ac.jp/nl-resource/juman.html

(注4)：論文 [6] では素性構造の単一化を行っているが，本実験で用いたバージョンでは，単一化の近似だけを行っている．

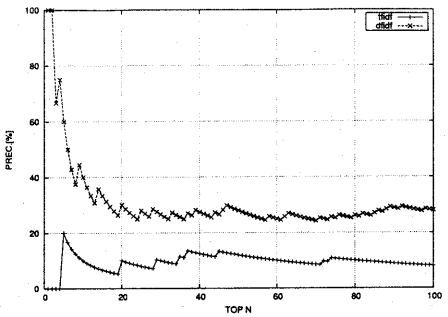

図 2 $af \cdot idf$ 法と $tf \cdot idf$ 法との比較

Fig. 2 The precision of the $tf \cdot idf$ and $af \cdot idf$ methods

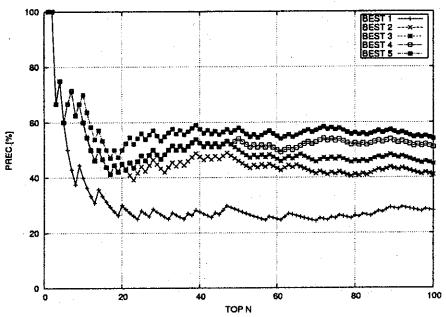

図 3 $af \cdot idf$ 法で上位語獲得を行った時の上位語の含有率

Fig. 3 The precision of the $af \cdot idf$ methods

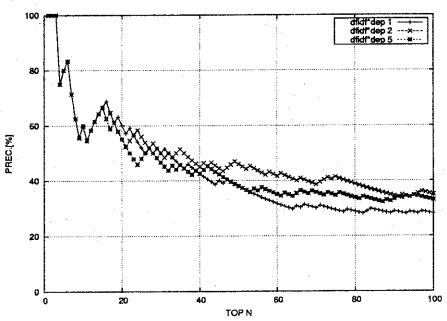

図 4 係り受けによる類似度を考慮した場合の上位語獲得精度

Fig. 4 The precision when the word similarities are considered

も高いのが確認できる. また, 上位 50 個の下位語候補リストでは 50%に少し欠る程度の精度が出ていることがわかる.

4.2.3 ページ数による絞込み

節 4.2.2 において, 一番精度の高かった M=2 の時に対して, 下位語候補リストの各要素の検索ヒット件数に下限を設け, 下限を下回る要素をもつ下位語候補リストを排除しての実験を行った. ヒット件数の下限を 0 から 100 まで変化させた時の精度の様子を図 5 に示す. おおむね下限を 10 件程度にすることによって, さらに性能向上がはかれることがわかった. 特にスコアが高い下位語候補リストで性能が向上している. 表 3, 4 に, 本実験において一番精度の高かった条件 (PAGE=10) の時に

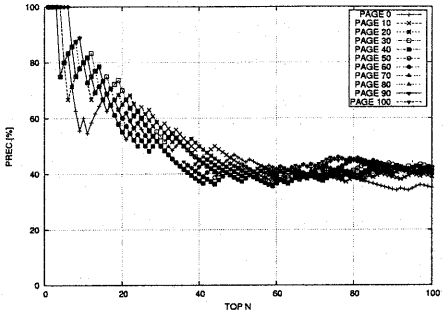

図 5 下位語候補の検索ヒット件数による絞り込みを行った時の上位語獲得精度

Fig. 5 The precision when the hit numbers of the search are considered.

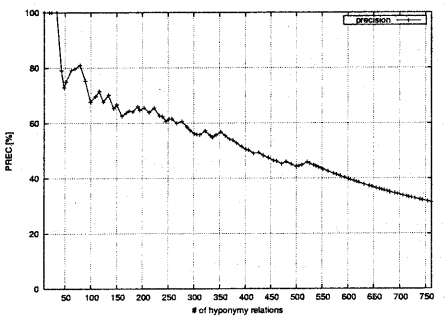

図 6 単語レベルでの上位下位関係獲得の精度

Fig. 6 The precision of hyponymy relations on the word level

獲得された上位語の例を示す.

4.3 単語間レベルで上位下位関係の獲得

今までの実験は, 各下位語候補リストに対して正しい上位語が獲得できているかについて行っていたが, ここでは, 単語間レベルで正しい上位下位関係の獲得がどのくらいできているかについて実験し, 評価した. 節 4.2.2 で下位語候補リストに対する上位語獲得の精度の一番高かった条件 (PAGE=10) を用い, そのときにどのくらいの正しい上位下位関係が獲得できているか調べた. 単語間レベルでの上位下位関係獲得の精度を図 6 に示す。ここで横軸は, これまでのグラフと同様に, 下位語候補リストを上位語のスコアにしたがってソートし, 各々のリストの要素, すなわち, 下位語をソート順に並べたときに, 上位 N 個までを取ってきたときの精度である。これまでのグラフと同様に上位語スコアで上位 100 個の下位語候補リストを対象としたが, その中に含まれる下位語候補は計 7 6 1 個であった。単純な比較は本来困難であるが, 図 5 に比べるとやや高い精度が出ているように見える。これは, 下位語候補のリスト全体としては適切な上位語でなくても, リストに含まれる語のいくつかに対して適切な上位語である場合があり, 図 6 では, それらも評価の対象になっているからである。最後に, 実際に獲得できた単語レベルでの上位下位関係を図 7 に示す.

表 3　ページ数による絞込みを行った時に獲得された上位語の例
Table 3　An example of hypernyms

CLASS 13					
フルート, トランペット, テューバ, ホルン, クラリネット 打楽器, ユーフォニアム, サクソフォーン, ファゴット, コントラバス					
1	楽器	1330.169	0.20484	272.472	○
2	奏者	865.189	0.16013	138.544	
3	演奏	849.680	0.08498	72.204	
4	管楽器	651.028	0.00000	0.000	
5	音楽	616.303	0.10227	63.027	
CLASS 47					
タガジョーウルフ, タマモミニスター, タガジョーシュネル メイショウカンウ, ヒノデコウジ, ダイコシャデ, アドマイヤストーム ナスケンエアスト, アイマストウイン, クロガネジョー					
1	馬	921.866	0.15332	141.342	○
2	ダート	751.394	0.00000	0.000	
3	レース	725.181	0.02537	18.398	
4	着	604.500	0.00000	0.000	
5	出走	597.950	0.00471	2.815	
CLASS 92					
魚, デジタルペット, 昆虫, 小動物, 爬虫類, いぬ, ねこ, 家畜, 鳥					
1	動物	329.135	0.28499	93.801	○
2	ペット	335.076	0.25282	84.714	○
3	更新	325.963	0.15512	50.563	
4	紹介	313.775	0.10976	34.438	
5	飼育	297.588	0.09530	28.360	
CLASS 29					
かりの舞, キヌヒカリ, ゆめあこがれ, 類, ハナエチゼン ヒノヒカリ, アキニシキ, コガネマサリ, あきたこまち, 日本晴,					
1	コシヒカリ	1047.871	0.18572	194.613	
2	米	1092.120	0.04003	43.717	○
3	品種	946.344	0.08842	83.680	○
4	栽培	720.722	0.04064	29.288	
5	玄米	686.991	0.05751	39.508	
CLASS 3					
健康診断, 日帰りドック, 一泊ドック, 脳ドック					
1	検査	776.476	0.69158	536.996	○
2	ドック	723.071	0.00000	0.000	
3	人間ドック	670.087	0.00000	0.000	○
4	受診	568.814	0.06732	38.294	
5	健康	465.159	0.07235	33.652	

○は正解を表す。下位語候補のリストの内 7 割をこえる語に対して適切な上位語であれば正解としている.

```
hypernym(“歌”, “大きな古時計”)
hypernym(“ゲーム”, “リッジレーサー”)
hypernym(“車”, “エスティマ”)
hypernym(“成分”, “黒ゴマ抽出物”)
hypernym(“症状”, “嘔吐”)
hypernym(“山”, “トムラウシ”)
```

図 7　実際に獲得された単語レベルでの上位下位関係
Fig. 7　An example of hyponymy relations

表 4　ページ数による絞込みを行った時に獲得された上位語の例
Table 4　An example of hypernyms

CLASS 12					
金賞, 特別奨励賞, グランプリ, 銀賞					
1	賞	477.538	0.62567	298.783	○
2	受賞	489.856	0.01759	8.618	
3	作品	261.281	0.01335	3.488	
4	入賞	236.207	0.01794	4.237	
5	日本	220.302	0.01882	4.146	
CLASS 95					
アイドルビデオ, 裏ビデオ, アダルトビデオ, 河村恵美子, 流出ビデオ					
1	ビデオ	575.553	0.15977	91.957	○
2	アイドル	410.053	0.01991	8.166	
3	DVD	296.632	0.06143	18.222	
4	写真	278.106	0.09695	26.963	
5	生	262.197	0.01229	3.223	
CLASS 72					
ワイルドアームズ, メタルギア, リッジレーサー, レガイア伝説 やるドラシリーズ, 女神転生, モンスターフゼム, セガ総合					
1	ゲーム	752.176	0.14895	112.039	○
2	プレイ	422.833	0.00000	0.000	
3	PS	410.583	0.00000	0.000	
4	主人公	371.195	0.05902	21.909	
5	発売	370.173	0.04861	17.993	
CLASS 41					
あんなに一緒だったのに, いつの日にか, 晩秋, 強烈ロマン 大切なもの, 大きな古時計, 大切なもの, 大きな古時計 またあえる日まで, 真実の詩, あさっては Sunday, 花鳥風月					
1	歌	520.263	0.29477	153.359	○
2	曲	730.203	0.19657	143.539	○
3	言葉	397.379	0.04583	18.212	
4	て	364.653	0.02098	7.649	
5	声	362.453	0.03222	11.679	
CLASS 28					
結合型エストロゲン, リン酸オセルタミビル, ファモチジン マレイン酸フルボキサミン, ザフィルルカスト, 硫酸プロタミン ウリナスタチン, ガドテル酸メグルミン, ラマトロバン, リルゾール					
1	薬	1614.283	0.12212	197.142	○
2	投与	1600.502	0.07487	119.833	
3	塩酸	1587.177	0.00000	0.000	
4	酸	1527.111	0.00000	0.000	
5	治療	1472.175	0.02922	43.018	
CLASS 35					
イノシトール, ナイアシン, ビオチン パントテン酸, エルゴステロール, コリン					
1	ビタミン	806.974	0.22600	182.379	○
2	ビタミンB	708.156	0.00000	0.000	
3	成分	623.047	0.26749	166.661	○
4	代謝	575.848	0.00000	0.000	
5	酸	556.331	0.00000	0.000	

○は正解を表す。下位語候補のリストの内 7 割をこえる語に対して適切な上位語であれば正解としている.

表 5　不要語リスト 1
Table 5　List of stop words 1

ふりがな	詳細	サーチエンジン	備考
終わりに	終りに	電話番号	コメント
おわりに			
^トップ	^ホーム	^リンク	^ヘルプ
^ニュース	^プレゼント	^カテゴリ	^サポート
^お問い合	^次の	^前の	^新着
^メール			
履歴$	リンク集$	連絡先$	内容$
他$	配布$	サービス$	メニュー$
情報$	目次$	もくじ$	予定$
管理人$	一覧$	方法$	窓口$
案内$	名称$	写真$	種別$
ページ$	チャット$	コーナー$	CHAT$
BBS$	著作権$	インフォメーション$	について$
戻る$	趣旨$	予約$	動画$
名$	から$	掲示板$	。$
、$	?$	!$	
.+と.+	.+・.+	.+, .+	.+／.+
.+ &.+			
.*ダウンロード.*	.*ログイン.*	.*更新.*	
.* (.*			

表 6　不要語リスト 2
Table 6　List of stop words 2

彼ら	物	あなた	ご覧	な	無料	必要
ほか	ぼく	僕	以下	一般	名	品
一部	下	下記	何	画面	夜	南
会	各種	株式	巻	関係	訳	日
基本	期間	気	系	個人	論	杯
向け	国際	最終	最新	妻	版	
作	姿	子	私	誌	母	晩
事	事項	時間	次	自分	北	彼女
室	手	種類	集	所	味	東
書	女	女性	商品	詳細	無断	等
上	情報	状況	心	新	父	堂
人	人間	人気	西	先	武	内容
線	送料	多く	対象	沢	部	話
男	中心	昼	著	丁目	別	彼
目	誰	大人	子供	前半	編	番号
後半	だ	朝	長	登	方法	ヶ
ゴール	ページ	クリック	サイト	ページ	ホームページ	メール
リンク	HP	MAIL	URI	THE		

5. おわりに

5.1 本研究のまとめ

本研究の目的は, 文の表層のパターンに注目せずに大量の単語の上位下位関係を, WWW 上のドキュメントより自動的に獲得することであった. WWW 上のドキュメントを対象とすることで, 新聞記事等のコーパスから獲得するのが難しいと考えられる上位下位関係, 例えば図 7 のような関係を獲得することができた. HTML 文書から上位下位関係を獲得するならば, 文書のタイトルや表やリストのキャプションを獲ってくればよいと思われるかもしれない. しかし, 実際にタイトルに上位語が含まれていないケースや, タイトルだけでは上位語を一意に絞れないケースも多く, また表やリストのキャプションの自動認識も HTML 文書の構造が複雑なため, それほど簡単ではない.

5.2 今後の課題

今後の課題としては, 精度の向上がまず第一に挙げられる. 精度を向上させるために, 下位語に関係の深い非上位語をまだ排除しきれていないので, 何らかの手段により排除したいと考えている. 前節で述べたようなタイトル情報の利用も有用であるかもしれない. また, 複数の単語からなる上位語の生成も行いたいと考えている. 例えば, 現在のシステムでは商店名からなる下位語候補リストに対して, ただ単に店としか上位語を求めることができないが, 下位語候補リストの商店が何の店なのかがわかるとより有益な情報であるといえる.

文　献

[1] Michael Fleischman,Eduard Hovy and Abdessamad Echihabi, Offline Strategies for Online Question Answering: Answering Questions Before They Are Asked, In proceedings of ACL2003, pp1–7, 2003. I

[2] Emmanuel Morin and Christian Jacquemin, Automatic acquisition and expansion of hypernym links, Computer and the Humanities 2003, forthcoming, 2003.

[3] 浅井達哉, 安部賢治, 川副真治, 坂本比呂志, 有村博紀, 有川節夫, 半構造データからの頻出パターン発見アルゴリズム, 第 13 回データ工学ワークショップ, C5-1, 2002.

[4] 今角恭祐, 並列名詞句と同格表現に着目した上位下位関係の自動獲得, 九州工業大学修士論文, 2001.

[5] Sharon A. Caraballo, Automatic construction of a hypernym-labeled noun hierarchy from text, In Proceedings of ACL'99, pp120–126, 1999.

[6] 金山博, 鳥澤健太郎, 光石豊, 辻井潤一, 3 つ組・4 つ組モデルによる日本語係り受け解析, 自然言語処理, Vol. 7(5), 2000.

[7] Rila Mandala, Takenobu Tokunaga and Hozumi Tanaka, The Use of WordNet in Information Retrieval, Proceedings of the COLING-ACL workshop on Usage of Wordnet in Natural Language Processing. pp.31 – 37, 1998.

[8] Ellen Riloff and Jessica Shepherd, A Corpus-Based Approach for Building Semantic Lexicons, In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing (EMNLP-2), pp117–124, 1997.

[9] Marti A. Hearst, Automatic acquistition of hyponyms from large text corpora, In Proceedings of the 14th International Conference on Computational Linguistics, pp539–545, 1992.

[10] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross and Katherine J. Miller, Introduction to WordNet: An on-line lexical database, Journal of Lexicography, 3(4):pp235–244, 1990.

[11] 北研二, 津田和彦, 獅々堀正幹, 情報検索アルゴリズム, 共立出版, 2002.

[12] 徳永健伸, 情報検索と言語処理, 東京大学出版会, 1999.