修 士 論 文

HTMLタグの繰り返しパターンに注目した知識の自動獲得

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

新里 圭司

修 士 論 文

HTMLタグの繰り返しパターンに注目した知識の自動獲得

指導教官　鳥澤健太郎 助教授

審査委員主査　鳥澤健太郎 助教授
審査委員　東条敏 教授
審査委員　島津明 教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

210044 新里 圭司

提出年月: 2004 年 3 月

概 要

本稿では，WWW 上に大量に存在する HTML 文書から広範な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提案する．WordNet に代表されるような大規模なシソーラスを自動生成するという目的のもと，従来より単語間の意味的関係の自動獲得に関する研究は盛んに行われてきた．しかし，そのほとんどは Hearst[4] が用いた“ *such as* パターン ”に代表される，構文パターン（*lexico–syntactic pattern*）のマッチングによりコーパス中から獲得するものであった．しかし，(a）単語間の意味的な関係を表す構文パターンがコーパス中に頻繁に現れることは稀であり，また（b）たとえ大量のテキストを持ってきたとしても，構文パターンに現れない単語や句が大量に存在するため，従来手法では大量かつ幅広い単語間の上位下位関係を獲得することが難しいという問題があった．

以上の理由より，本研究では構文パターン以外の上位下位関係の特性を捕らえる手がかりを用いることで獲得を試みる．具体的には，(1）HTML タグにより与えられる文書の構造，(2）情報検索などの分野で用いられる *df* や *idf* などの統計量，(3）大量の新聞記事から収集した名詞と動詞の係り受け関係，(4）予備実験により得られたヒューリスティクなルール，の 4 つの異なる要素を組み合わせることで上位下位関係の獲得を行う．

実際に WWW 上より収集した HTML 文書集合に対し，本稿で提案する手法を適用することで，テキストの量が少ないという理由により，従来手法では獲得することが難しい上位下位関係を，提案手法では獲得できることが実験により確かめられた．

目 次

第1章　はじめに

1.1　研究の背景と目的

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*[1]）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，WWW 上に大量にある HTML 文書から広範な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．Miller によれば，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[2]されており [8]，本研究でもこの定義に従う．また単語 B が単語 A の下位語であるということを次の形式で記述する．

HYPONYM(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．

HYPONYM(“野菜”，“茄子”)
HYPONYM(“魚”，“秋刀魚”)
HYPONYM(“機械”，“冷蔵庫”)

本研究で，WWW 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比べ，1) 量が豊富にある，2) 新規に「発明された」語や表現が素早く掲載される，3) 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いる，といった特徴を HTML 文書は持っており，広範な単語間の上位下位関係を獲得するためにその特徴が使えるのではないかと考えたためである．

従来より研究されてきた単語間の上位下位関係の獲得手法は，新聞記事などのコーパスから構文パターン（*lexico–syntactic pattern*）のマッチングにより獲得するものがほとんどであった [4, 5, 2, 16, 9, 3, 14]．しかし従来の方法では，コーパス中に上位下位関係を表す構文パターンがそれほど頻繁に現れず，たとえ大量のテキストをもってきたとしても，

[1] “*meronymy relation*” ともいう．

[2] より正確には “*A concept represented by a lexical item L_0 is said to be a hyponym of the consepct represented by a lexical item L_1 if native speakers of English accept sentences consutructed from the frame* **An L_0 is a (kind of) L_1.**” と定義されている．

構文パターンに現れない単語や句が大量にあるといった問題により，大量かつ幅広い単語間の上位下位関係を獲得することが難しいという問題があった．そのため，本研究では従来法で用いられてきたような構文パターンによる獲得は行わず，構文パターン以外の上位下位関係の特性を捕らえる手がかりを用いることで獲得を試みる．具体的には，タグにより与えられる HTML 文書の構造，情報検索などの分野で用いられる df や idf などの統計量，新聞記事から収集した名詞と動詞の係り受け関係，予備実験より得た知見に基づき作成したヒューリスティックなルール，の 4 つの異なる要素を組み合わせることで上位下位関係の獲得を行う．その結果，実際に WWW より収集してきた約 87 万件の HTML 文書から，下位語の集合（の候補）を約 9 万個獲得することができた．そして，その中からランダムに抽出した集合 2,000 個について評価を行ったところ，2,000 個の集合に含まれる約 14,000 個の順序付けられた上位下位関係のうち，全体の約 3.6 %にあたる上位 501 個については 85 %，全体の約 5 %にあたる上位 700 個の上位下位関係については 75 %，約 10 %にあたる 1,400 個については 60 %程度の精度で正しい上位下位関係を獲得することができた．

このような単語間の上位下位関係は種々の自然言語処理アプリケーションにおいて有用であると考えられる．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，QA の分野においても,「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，単語間の上位下位関係を利用して答えるといった研究が行われている [3]．

1.2 本論文の構成

以下，本稿では第 2 章で関連研究について解説する．本稿で提案する手法は大雑把に分けると，HTML 文書中から意味的に類似した表現同士を獲得する Web マイニング的要素を含んだ部分と，コーパス（本研究では，HTML 文書をコーパスとしている）から，複数の下位語に共通の上位語を獲得する知識獲得の部分の 2 つの部分からなる．そのため第 2 章では，まず HTML 文書中から情報抽出を試みた研究について説明し，続いてコーパス中から単語間の意味的関係（上位下位関係や包含関係）の獲得を試みた研究についてそれぞれ説明する．次に第 3 章では，本研究で提案する「**構文パターンによらない上位下位関係の獲得方法**」について説明する．第 3 章では，まず本稿で提案する手法のおおまかな流れについて解説し，その後本研究で提案する手法を 4 つステップに区切り，各ステップについて説明する．続いて第 4 章と第 5 章では，提案手法の評価実験について述べる．本研究では 4 種類の実験を行ったが，それぞれの実験についてグラフ及び実験結果を示し，考察を与える．最後に，第 6 章にて提案手法の今後の課題について触れ，本研究のまとめを行う．

第2章　関連研究

本研究は，大別するとHTML文書中のタグ情報を利用して下位語の集合を獲得するWebマイニング的要素を含んだ部分と，与えたれた下位語の集合に共通する上位語を大量のテキストから獲得する知識獲得の部分に分けることができる．本章では，これら両方についての先行研究について述べる．

2.1　HTML文書中のタグを利用した情報抽出

HTMLタグを利用して，HTML文書中から情報抽出を試みた研究として，藤井ら[19]，Sakamotoら[10]の研究がある．藤井らはWWWを事典として扱うことを目的に，WWW上に大量に存在するHTML文書から任意の用語に関する定義文の抽出を行っている．またSakamotoらは，HTML文書中に現れる表現がどのようなタグのパターンで囲まれるかを学習するTree–Wrapperを提案し，論文検索サイトより得られるHTML文書を基に学習されたTree–Wrapperを用いて，論文のタイトル，著者名，概要といった3種類の情報を抽出している．本節では，これらHTMLタグを利用して情報抽出を行った研究について説明する．

2.1.1　WWWからの事典的知識の抽出に関する研究

本節では藤井らの行った，WWW上に大量に存在するHTML文書を利用して，用語に関する事典的知識を自動的に抽出する研究について述べる．

概要

藤井ら[19]は，WWW上に新規性や専門性の高い情報が多く流通している点に注目し，WWW上に大量に存在するHTML文書から任意の用語に関する事典情報を自動的に生成するシステムの提案・開発を行っている．例えば藤井らが開発しているウェブ事典検索システムCyclone[1]を用い「2の補数」について調べてみると，図2.1に示すような定義文が得られる．藤井らの提案する事典的知識獲得手法は以下の3つのステップからなる．

[1] http://cyclone.slis.tsukuba.ac.jp/

PIC16 シリーズの命令 [コンピュータ] キャッシュ

2 の補数 **2 の補数**とはマイナスの数値を表すものです。例えば 10 進数で–1 は 8 ビットの 2 進数で表すと 11111111 になります。検算すると以下のようになり、–1 を表していることが分かります。オーバーフローが発生しますが、数値はゼロになります。2 進数の足し算の方法は 10 進数と同じように下位の桁から足して、桁上げがあった場合にはそれを含めて次の桁を計算するという方法で行います。マイナスの値を使うのには条件があります。8 ビットで表せる数値は 0〜255 の 256 種類ですが、マイナスの数値のを使うとした場合には–127〜+127 の 255 種類になります。一つ少ないのは 10000000 が使われないからです。このビットの並びは –0 を表しますが、計算上では使えません。最上位のビット 7 はプラスかマイナスを表す符号ビットとしての意味を持ちます。数値がプラスだけなのかプラス／マイナスを表すのかは処理するときに意識する必要があります。例えば –127 を 2 進数で表すと 10000001 です。プラスのみの数値とすると 129 を表しているように見えてしまいます。**2 の補数**への変換の方法は以下のように行います。例として 56 を –56 に変換してみます。(1) プラスの数値から 1 を引く 56 – 1 = 55(2) これを 2 進数に変換する 55 → 00110111(3) 0 と 1 を逆にする 00110111 → 1100100011001000 が –56 を表す 2 進数です。検算してみます。答えはゼロになりました。

図 2.1 Cyclone より得られる「2 の補数」の定義文

1. 事典的知識を生成したい用語を含んでいるHTML文書を既存の検索エンジンを使って収集する
2. 文書表現や，HTMLタグによって与えらえるレイアウト情報に基づいて，HTML文書から用語を説明している個所を抽出する
3. 抽出された複数の用語説明を，分野や語義に基づいて分類することで組織化し，利用者の閲覧効率の向上をはかる

藤井らは，情報処理技術者試験に出題された専門用語96語を用いて評価実験を行った．その結果，提案したシステムが生成した事典情報は，既存の事典より網羅性が高く，実用レベルの質に達していると報告している．

以下では，HTMLタグから与えられるレイアウト情報を利用して知識（ここでは，用語説明）を自動的に抽出するという点で本研究と関連のある，用語説明個所抽出処理について説明する．

HTML文書からの用語説明個所の抽出

藤井らの提案したシステムで用いられている用語説明個所抽出処理は，HTML文書中で用語説明が行われていそうな部分の見当をつける手がかり特定処理と，見当をつけた部分から実際の抽出範囲を特定する範囲特定処理の2つに分けることができる．両方の処理は，ともにHTMLタグから得られるレイアウト情報を利用している．

手がかり特定処理　藤井らはHTML文書中で用語説明を行っている個所の見当をつけるため，文章表現に関する手がかりとHTMLタグに関する手がかりの2つを用いている．HTMLタグに関する手がかりを用いている理由は，用語説明の手がかりは文章表現だけではなくHTML文書中のレイアウトからも得られることがあるためである．

まず，文章表現に関する手がかりであるが，藤井らは「CD–ROM世界第百科事典」から半自動的に抽出した,「XとはYである」や「XをYと定義」といった18種類の手がかり（以下では，文章表現テンプレートと呼ぶ）を用い，HTML文書中で用語説明の行われていそうな個所を特定している．

続いてHTMLタグに関する手がかりとして，藤井らは用語説明を含むHTML文書に典型的に見られるHTMLタグの使用を分析し，以下に示す2つの手がかりを用いている．1つ目の手がかりは，<DT>，<B>，<Hx>等のタグ（<Hx>のxは数字を表す）で説明の対象となる用語が囲まれている場合，後続する段落を用語説明個所として見なすというものである．この時，見出しとして「(用語）とは」,「(用語）とは？」などの表現が使用されることもあるため，タグだけではなくこれらの表現も手がかりとして利用する．次いで2つ目の手がかりは，<A>タグにより説明の対象となる用語にリンクが付与されていた場合，そのリンク先を用語説明個所として見なすというものである．リンク先としては，他のページや同一ページ内の別の個所が考えられる．見出しによる手がかりの場合と同様「(用語）とは」,「(用語）とは？」などの表現に対してリンクが付与されることがあるため，それら

の表現も手がかりとして利用する．HTML タグを利用することで，文書表現テンプレートでは特定できない手がかりを得ることが可能になる．

範囲特定処理　文章表現テンプレートを用いることで，用語説明を文単位で抽出することができる．しかし，用語説明は文章や箇条書きによって行われることがあるため，テンプレートにマッチした文は用語説明抽出のための 1 つの手がかりでしかなく，範囲特定処理によって，文よりもさらに大きな範囲を用語説明として獲得する必要がある．

また，HTML タグを手がかりとして抽出個所の見当をつけた場合は，見出しやリンクが指す位置から一定の領域を抽出する必要がある．ここでいう領域とは，段落や箇条書きのように複数の文で構成された意味的なまとまりのことを指す．しかし，このような段落や箇条書きといった単位を，テキスト情報だけを頼りに正しく特定することは困難である．そこで藤井らは，段落や箇条書き単位の説明個所を抽出するために，HTML タグによって与えられるレイアウト情報を用いて範囲特定を行っている．具体的には，テンプレートにマッチした文を含む領域や，見出しやリンク先に続く一定の領域のうち，以下の条件に当てはまる領域を用語説明個所として抽出している．

1. 対象用語が用語定義を表すタグ<DT>で囲まれている場合は，その用語の説明個所を表すタグ<DD>で囲まれた領域
2. 段落を表すタグ<P>で囲まれている領域（終了タグ</P>が省略されている場合は次の<P>タグが現れるまでの領域）
3. 箇条書きを表すタグ<UL>で囲まれている領域
4. 抽出を行う場所から見て N 文（藤井らは経験的に $N = 3$ としている）

以上より，文章表現パターンと HTML タグによって与えられるレイアウト情報の両方を用いることで，藤井らは HTML 文書中に現れる用語説明の抽出を行っている．

本研究との比較

第 3 章で説明するように，本研究でも HTML 文書中のタグから得られる情報を利用して意味的に類似した要素（下位語の集合）の獲得を行っている．本研究と藤井らの研究の異なる点は，藤井らの研究は知識（用語説明）を獲得する際に，文章表現パターンや特定のタグ（<DT>，<B>，<Hx>）を用いているのに対し，本研究では文章表現パターンや特定のタグを用いず，個々の HTML 文書が持っている構造を利用することで知識（下位語の集合）を獲得を行っている点である．そのため，本研究で提案する手法は，どのようなタグに囲まれている表現であっても，ある一定の構造を HTML 文書が持っていれば知識として獲得することが可能である．

2.1.2 HTML Wrapper を用いた HTML 文書からの情報抽出

本節では，HTML Wrapper を用いて HTML 文書から情報抽出を行う研究について説明する．以前より HTML 文書を対象に情報抽出を行う研究はされてきたが，そのほとんどは個々の HTML 文書の構造に依存したヒューリスティックな手法であった．しかし，Kushmerick[6] は，HTML 文書からの情報抽出を，ラッパー帰納として明確化することで，計算論的手法に基づく情報抽出の枠組みを提案した．これは，帰納学習の 1 つであり，学習アルゴリズムは任意の組 $\langle P_i, L_i \rangle$ に対して $W(P_i) = L_i$ となるような関数（プログラム）W を出力することが目的である．ただし P_i は 1 つの HTML 文書であり，L_i は P_i から切り出すべきテキストの場所を指定したファイルである．また，学習アルゴリズムに与えられる訓練例は，組 $\langle P_i, L_i \rangle$ の列である．この学習アルゴリズムによって出力されたプログラムのことを **HTML Wrapper** と呼ぶ．以下では，Sakamoto らの提案した Tree–Wrapper について説明する．

概要

Kushmerick の提案した LR Wrapper は，HTML タグによって与えられる文書の構造を無視し，HTML 文書を ASCII 文字の並びとして捉え，その中で抽出したいデータがどのような文字列に囲まれやすいかを学習するものであった．そのため，Kushmerick の提案した LR Wrapper では，正しい抽出が行えない場合がある．そこで Sakamoto らは，HTML 文書を DOM（Document Object Model）と呼ばれる木構造としてみなすことで，HTML タグによって与えられる文書の構造を捉えることのできる，Tree–Wrapper を提案した [10]. Tree–Wrapper では，木構造で表現された HTML 文書の中で，抽出したいデータがどのようなパスの葉ノードとして現れるのかを学習する．

村上ら [15] は，Sakamoto らの提案した Tree–Wrapper を実装し，WWW 上に実際に存在する HTML 文書から情報抽出実験を行った．実験に用いた HTML 文書は，論文検索サイトである citeseers[2]からダウンロードした，1,300 件の HTML 文書である．そして，ダウンロードした HTML 文書中に記載されている論文のタイトル，著者名，アブストラクトの 3 種類の情報を抽出するために，ダウンロードされた文書量のおよそ 1 %にあたる，10 件の HTML 文書を無作為に選びだし，Tree-Wrapper の学習を行った．そして，学習された Tree–Wrapper を用いて残りの 1,290 件の HTML 文書から先程の 3 種類の情報の抽出実験を行った．その結果，学習に用いた HTML 文書中のテーブル要素に指定されていない属性を持つ 3 件の HTML 文書を除く，1,287 件の文書に対して，正しく情報を抽出することができたと報告されている．

[2]http://citeseer.nj.nex.co

本研究との比較

Sakamoto らの提案した Tree–Wrapper を用いて，本研究で獲得している下位語の集合を獲得することは難しいと考えられる．それは，Sakamoto らの手法は少量ではあるが，獲得したいデータ及び，そのようなデータが HTML 文書中のどの部分に現れているかを示した学習データを必要とするためである．本研究で獲得している下位語の集合は，多種多様な HTML 文書の様々な部分から獲得されたものである．そのため，Sakamoto らの手法を用いて下位語の獲得を行うことを考えた場合，個々のページごとに Tree–Wrapper を学習する必要がある．しかし，前述したように Tree–Wrapper 学習のためには，正解データを必要とするため，大量の HTML 文書から下位語を獲得しようとすると，大量の学習データが必要となり現実的でないと考えられる．

2.2 構文パターンを用いた知識の自動獲得

ここでは，構文パターンを用いて新聞記事などのタグなしコーパスから，自動的に知識を獲得する方法について説明する．ここで知識とは，単語間の上位下位関係や，包含関係といった主に単語間の意味的な関係のことを指す．本節では，まず単語間の上位下位関係を構文パターンを用いて獲得する Hearst[4]，今角 [16]，安藤ら [14] の研究について説明する．次いで，単語間の包含関係を Hearst 同様，構文パターンを用いて獲得する Berland ら [1] の研究について説明する．

2.2.1 構文パターンによる単語間の上位下位関係の獲得

これまでにも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，そのほとんどが，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って上位下位関係の獲得を行うものとなっている [4, 5, 2, 16, 9, 3, 14]．本節では，構文パターンを用いてコーパスから単語間の上位下位関係の自動獲得を最初に行った，Hearst[4] の手法について述べ，その後 Hearst の手法を日本語の新聞記事に対して適用した，今角 [16]，安藤ら [14] の研究について説明する．

先行研究の概要

Hearst[4] は構文パターンを用いて新聞記事などのコーパスから単語間の上位下位関係を自動的に獲得する手法を提案している．Hearst が提案した手法は “*B such as A*” のような多くの場合単語間の上位下位関係を表している構文をあらかじめパターン化しておき，これらのパターンをコーパス中の文にマッチさせることで単語間の上位下位関係の獲得を行うものである．Hearst はこのような単語間の何らかの意味的関係を表す構文のパターンのことを *Lexico-syntactic pattern* と呼んでいる（本稿では**構文パターン**と呼ぶ）．

Hearst は “*such as* パターン” 以外にも上位下位関係を表す幾つかのパターンを発見している．Hearst が発見した構文パターンを図 2.2 に示す．

Hearst は図 2.2 に示した，“*or other* パターン” を百科事典[3]中のテキストに対し施すことで，提案手法の評価実験を行った．その結果，約 52 %の精度で妥当な上位下位関係を獲得することに成功したと報告している．

Hearst が提案した，構文パターンを用いて単語間の上位下位関係を獲得する手法を，日本語の新聞記事に対して適用し上位下位関係の自動獲得を試みた研究として今角 [16]，安藤ら [14] の研究がある．

今角は「言い換え（*paraphrasing*）」に必要な言語知識を自動獲得することを目的に，文中に現れる同格表現や並列名詞句を手がかりに単語間の上位下位関係の獲得を行っている．実験データとしては毎日新聞 4 年分（およそ 232 万文）を用いており，構文解析の結果より得られる同格・並列表現を含む文に対し，以下にに示すような構文パターンを用いて上位下位関係の獲得を行っている．

名詞句 A「**名詞句 B**」
名詞句 A など (、｜の) **名詞句 B**
名詞句 A のような**名詞句 B**

その結果，約 15,000 件の上位語下位語対が獲得でき，そのうち 600 件について人手で評価を行ったところ，その精度は 77.2 %であったと報告している．

また安藤らは，現在人手で作成されている連想概念辞書 [17] のような大規模なシソーラスを自動的に生成するための準備として,「X などの野菜」といった構文パターンを用いて，連想概念辞書に登録されている日常性の高い基本的な単語約 60 語について，新聞記事からその下位語の獲得を行っている．安藤らが下位語を抽出する対象とした上位語（安藤らはこのような語を**対象語**と呼んでいる）の例を以下に示す．

家具，果物，楽器，乗り物，動物，野菜，食べ物

安藤らは単語間の上位下位関係を表す構文パターンを新聞記事中から獲得するために，連想概念辞書に登録されている上位語とその下位語を利用している．安藤らは，上位語とその下位語を共に含んでいる文章をコーパスより抽出し，抽出された文章群の中から上位語を含む文節が下位語を含む文節に係っているもの及び，下位語を含む文節が上位語を含む文節に係っているものを中心に，上位下位関係を表す構文パターンがあるかどうか調べた．その結果，約 30 種類の構文パターンを見つけることができ，その中から用例の少ないものを除いた，以下に示す 7 種類の構文パターンを下位語獲得のために用いている．

(1) **下位語**など**対象語**	(2) **下位語**などの**対象語** *
(3) **下位語**のような**対象語** *	(4) **下位語**に似た**対象語**
(5) **下位語**以外の**対象語**	(6) **下位語**という**対象語**
(7) **下位語**と呼ばれる**対象語**	

[3] *Grolier's American Academic Encyclopedia*

表 2.1 安藤らの手法により獲得された上位下位関係の例

上位語 (対象語)	獲得された下位語（一部）
家具	ソファ，テーブル，いす
果物	リンゴ，ミカン，メロン
楽器	ピアノ，バイオリン，ギター
乗り物	飛行機，自転車，ジェットコースター
動物	人間，猫，猿
野菜	トマト，ニンジン，キャベツ
食べ物	果物，パン，米

ここで，パターンの後の “∗” のついているものは，今角も上位下位関係獲得に用いているパターンである．またパターン (1)，(2)，(3)，(5) については，

ビオラや**チューバ**などの**楽器**を失った．

のように下位語が並列して列挙されている場合があるため，そのような場合には，列挙されている語も下位語として獲得できるようにパターンを拡張し，柔軟性を持たせている．

構文解析済みの新聞記事 6 年分に対して，上に示した 7 種類の構文パターンを適用することで，連想概念辞書より選びだした対象語に対応する下位語の獲得を行っている．表 2.1 に安藤らの手法によって実際に獲得された下位語の例を示す．その評価は,「A は B である」という文の，A の部分に獲得された下位語を，B の部分に対象語を当てはめた時，文として成立するかどうかによって行っている．その結果，いずれのパターンについても約 60 %から 85 %程度（期待値[4]は 68.2 %）の精度で正しい上位下位関係が獲得できたと報告している．一見すると，今角のものより精度が低いように思えるかもしれない．しかし，今角が用いているパターンに限ればその獲得精度の期待値は 81.3 %であり，安藤らの方が若干高い．しかし，今角の手法と比べ安藤らの手法は，上位語を連想概念辞書から人手で獲得している．そのため，安藤らの手法の方が今角の手法より，はじめからより多くの情報が人によって与えられていると考えることができる．また，抽出に用いているコーパスの量も今角が新聞記事 4 年分に対し，安藤らは 6 年分とより多くのコーパスを用いている．このことから，一概に安藤らの手法の方が今角の手法より優れていると断定することはできない．

本研究との比較

本節で挙げた構文パターンを用いて上位下位関係を獲得する従来の手法と，本稿で提案する手法とではその獲得方法が全く異なる．従来手法が構文パターンを用いて獲得を試み

[4]論文 [14] では各パターンにより獲得できた下位語数とその精度しか報告されていない．そのため，ここで挙げた期待値は筆者が論文から求めた値である．

ているのに対し，本稿で提案する手法は構文パターン以外の，上位語と下位語がもつ特性を利用して上位下位関係の獲得を行っている．

また従来の手法では，コーパス中に現れる上位下位関係を表す構文パターンを，人手もしくは半自動的に生成しているため，どうしても人手の介入を避けられない．しかし本稿で提案する手法は，入力された下位語の集合に共通の上位語を，統計量を用いて自動的に獲得するため，人手の介入はない．もちろん，下位語の集合の獲得に関しても人手を介入することがない．そのため，低コストで上位下位関係を獲得することができる．

2.2.2 構文パターンを用いた包含関係の自動獲得

本節では，構文パターンと確率値を用いてコーパスから単語間の包含関係の獲得を行った Berland ら [1] の研究について説明する．

概要

Hearst の研究を受け Berland らは，構文パターンを用いてコーパス中から単語間の包含関係（*part–whole relation*）の獲得を試みた [1]．図 2.3 に Berland らが作成した単語間の包含関係を表す構文パターンを示す．これらの構文パターンは，論文 [4, 5] で述べられている方法に基づいて作成されている．

Berland らは図 2.3 に示す構文パターンを用いて，コーパス中から「車」に関する包含関係を獲得する予備実験を行い，その結果，精度の高かったパターン (1) 及び (2) を用いて提案手法の評価実験を行っている．

Berland らの手法は Hearst の提案した手法同様，構文パターンのマッチングにより，入力された全体を表す語に対する部分を表す語を獲得しているが，獲得された部分を表す語に対して，確率値を用いてスコアを求めるという操作を新たに導入している．これにより，獲得された部分を表す語の集合の中から，確率値の高い語を取り出すことで，より尤もらしい語だけを獲得することができるようになる．Berland らは獲得された部分を表す語の確率値を以下の式に基づいて求めている．

$$p(w, W(w)|p, P(p))$$

ここで，w，p は全体（*whole*）もしくは部分（*part*）を表す語の確率変数であり，$W(w)$ 確率変数 w がパターン (1)，(2) 中で全体を表す語として現れていることを，$P(p)$ は確率変数 p がパターン (1)，(2) 中で部分を表す語として現れていることをそれぞれ示している．

Berland らは，本，建物，車，病院，工場，学校の全 6 単語に関して，その部分を表す語の獲得を行った．全体を表す各単語ごとに，獲得された上位 50 個の部分を表す語が，妥当であるかどうかの評価を行った．評価には 5 人の被験者を用い，過半数を越える被験者が妥当であると判断した場合，正しい包含関係が獲得できたしている．その結果，上位 50 個の部分を表す語を獲得した場合，およそ 55 %の精度で正しい語を獲得することができ

たと報告されている．また，上位 20 個までに獲得する語の数を限定すれば，その精度はおよそ 70 %であり，比較的高い精度で単語間の包含関係を獲得することに成功している．

本研究との比較

Berland らと本研究とでは，獲得の対象としている知識が異なる．しかし，Berland らは構文パターンにより獲得された部分を表す語を，包含されやすさを表す確率値に従ってソートし，その上位幾つかを獲得することで相対的に高い精度で単語の包含関係の獲得を行っており，本研究でも Berland ら同様，獲得された下位語の集合と上位語の組をその類似度に従ってソートし，その上位幾つかを獲得することで相対的に高い精度で上位下位関係を獲得を行っている．

(1) *NP such as* {*NP,* }* {*(or* | *and)*} *NP*
The bow lute, such as Bambara ndang, ...
$\Longrightarrow$ HYPONYM(*"bow lute"*, *"Bambara ndang"*)

(2) *such NP as* {*NP,* }* {*(or* | *and)*} *NP*
...works by such authors as Herrick, Goldsmith, and Shakespeare.
$\Longrightarrow$ HYPONYM(*"author"*, *"Herrick"*)
HYPONYM(*"author"*, *"Goldsmith"*)
HYPONYM(*"author"*, *"Shakespeare"*)

(3) *NP* {*,NP*}*, *or other NP*
Bruises,...,broken bones, or other injuries ...
$\Longrightarrow$ HYPONYM(*"injury"*, *"bruise"*)
HYPONYM(*"injury"*, *"broken bone"*)

(4) *NP* {*,NP*}*, *and other NP*
...temples, treasuries, and other important civic buildings.
$\Longrightarrow$ HYPONYM(*"civic building"*, *"temple"*)
HYPONYM(*"civic building"*, *"treasury"*)

(5) *NP, including* {*NP,*}* {*or* | *and*} *NP*
All common–law countries, including Canada and England ...
$\Longrightarrow$ HYPONYM(*"common–law country"*, *"Canada"*)
HYPONYM(*"common–law country"*, *"England"*)

(6) *NP, especially* {*NP,*}* {*or* | *and*} *NP*
...most European contries, especially France, England, and Spain.
$\Longrightarrow$ HYPONYM(*"Eurpean country"*, *"France"*)
HYPONYM(*"European country"*, *"England"*)
HYPONYM(*"European country"*, *"Spain"*)

図 2.2 英語において上位下位関係を表す構文パターン

(1) *whole* NN[–PL]'s POS *part* NN[–PL] ∗
. . . building's basement . . .

(2) *part* NN[–PL] of PREP {the | a} DET *mods* [JJ | NN]* *whole* NN ∗
. . . basement of a building . . .

(3) *part* NN in PREP {the | a} DET *mods* [JJ | NN]* *whole* NN
. . . basement of a building . . .

(4) *parts* NN–PL of PREP *wholes* NN–PL
. . . basements of buildings . . .

(5) *parts* NN–PL in PREP *wholes* NN–PL
. . . basements in buildings . . .

Format: type_of_word TAG type_of_word TAG . . .
NN = Noun, NN–PL = Plural Noun,
DET = Determiner, PREP = Preposition,
POS = Possessive, JJ = Adjective

（注）パターンの後に "∗" のついているものは実際に実験で用いられているパターンである.

図 2.3 Berland らが作成した単語間の包含関係を表す構文パターン

第3章　提案手法

3.1　概要

本研究では，以下に示す3つの仮説をたて，単語間の上位下位関係の獲得に用いている．

仮説1 HTML 文書中に現れる箇条書きやリストボックス，テーブルのセルなどの要素は，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 共通の上位語をもつ下位語の集合が与えられた時，各下位語に共通する上位語は各下位語を（少なくとも1つ）含む文書に現れやすく，それ以外の文書には比較的現れにくい

仮説3 上位語と下位語は意味的に類似しており，そ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捉えることができる

そして，上の仮説に基づいた以下に示す4つのステップを経ることで単語間の上位下位関係の自動獲得を行う．ここに挙げたステップ1，2，3は上の仮説1，2，3とそれぞれ対応している．

ステップ1 HTML 文書中のタグ情報に基づいた下位語候補集合の獲得

ステップ2 *df*，*idf* などの統計量に基づく上位語候補の獲得

ステップ3 上位語候補と下位語候補間の意味的類似度に基づく上位語候補と下位語候補集合の並べ替え

ステップ4 ヒューリスティックなルールを用いた上位語候補と下位語候補集合の組の取捨選択

ここでステップ4は，上位下位関係獲得の精度を改善するために，ステップ1，2，3を通して獲得された上位下位関係を，予備実験により得られた知見に基づき作成されたヒューリスティックなルールに従い修正，または削除するステップである．

本手法では，ステップ1において WWW より大量の HTML 文書をダウンロードし，その中から仮説1に従い同じリストの項目になっている表現や，同じテーブルの要素となっている表現を獲得する．例えば，図3.1に示すような HTML 文書を考えた場合，ステップ1では次のようなパソコンの周辺機器とソフトウェアのジャンルからなる2つの集合を獲得する．

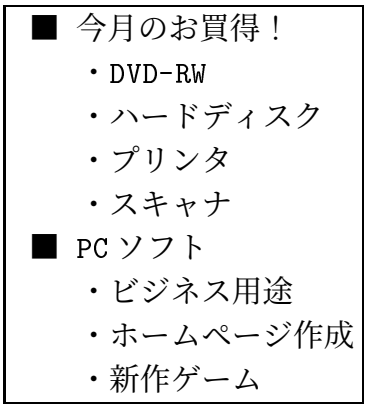

図 3.1 HTML 文書中に現れる箇条書きの例

{DVD–RW，ハードディスク，プリンタ，スキャナ }
{ ビジネス用途，ホームページ作成，新作ゲーム }

本研究では，ステップ 1 で獲得された集合の各要素を**下位語候補**と呼び，同じ集合の下位語候補同士は共通の上位語（この例でいえば「機器」や「ジャンル」）を持つと考える．また，獲得された下位語候補の集合を**下位語候補集合**と呼ぶ．ここで “候補” と付いているのは，ステップ 1 で獲得される HTML 文書中の表現の集合が，必ずしも共通の上位語を持つとは限らないためである．

次いでステップ 2 では，従来より情報検索の分野などでよく用いられている *df* や *idf* といった統計量を利用し，ステップ 1 で獲得された各下位語候補に共通な上位語を獲得する．そのためにステップ 2 では，まず下位語候補を少なくとも 1 つ含むような文書を既存の検索エンジンを用いて WWW からダウンロードする．そして，ダウンロードした文書中に含まれる名詞のスコアを計算し，スコアの最も高かった名詞を上位語の候補として獲得する．本研究では，この獲得された名詞のことを**上位語候補**と呼ぶ．ここでも “候補” と付いているのは，ステップ 2 で獲得された名詞が最終的な上位語となるわけではなく，獲得された名詞のうち幾つかは後述するステップ 4 で修正される可能性があるためである．ステップ 2 で用いる名詞のスコアの計算式は，仮説 2 に基づき，下位語候補を検索語としてダウンロードした文書集合中の多くの文書に現れやすい名詞ほど高いスコアを得るようにする．先程の例でいえば，DVD–RW やハードディスクを検索語として WWW よりダウンロードした文書集合には，実際に多くの文書中に正しい上位語である「機器」が含まれることになり,「機器」は高いスコアを得ることになる．

しかし，上位語ではないが DVD–RW やハードディスクと関連の強い名詞，例えば「データ」などの語も，多くの文書中に現れるため高いスコアを得てしまう．そこでステップ 3 では，このような上位語ではない名詞を誤って上位語候補として獲得している上位語候補と下位語候補集合の組を，最終的な出力結果から削除する．そのためステップ 3 では，仮説 3 に基づき，上位語候補と下位語候補の持つ係り受け関係から，両者間の意味的類似度を計算し，その値に従って上位語候補と下位語候補集合の組をソートする．上位語候補

```
<UL>
    <LI>今月のお買得！</LI>
    <UL>
        <LI>DVD-RW</LI>
        <LI>ハードディスク</LI>
        <LI>プリンタ</LI>
        <LI>スキャナ</LI>
    </UL>
    <LI>PC ソフト</LI>
    <UL>
        <LI>ビジネス用途</LI>
        <LI>ホームページ作成</LI>
        <LI>新作ゲーム</LI>
    </UL>
</UL>
```

図 3.2 HTML 文書のソースの例

と下位語候補集合の組をソートすることにより，ソートされた組の上位幾つかを最終的な出力結果とすることで，上位語候補と下位語候補に類似性の見られない組に関しては最終的な出力結果から削除することができる．例えば先程の例において，上位語候補として「データ」が獲得された場合,「データ」と DVD–RW，ハードディスク，プリンタ，スキャナは似た係り受け関係を持ちにくいため，類似性が弱いと考えられ，最終的な出力結果からは除かれる．

最後にステップ 4 として，予備実験より得た知見を基に作成したヒューリスティックなルールを，ステップ 1 から 3 までで獲得された上位語候補と下位語候補集合の組に対して適用し，上位語候補の修正や，上位語候補と下位語候補集合の組の削除を行う．そして，ステップ 4 を施した後，残った上位語候補と下位語候補の組の中から，上位幾つかを最終的に獲得された上位下位関係として獲得する．

以上が，本研究で提案する構文パターンを用いずに単語間の上位下位関係を獲得する手法の概要である．以降本節では，各ステップについて説明する．

3.2 下位語候補集合の獲得 (ステップ 1)

ステップ 1 は，WWW より大量にダウンロードしてきた各 HTML 文書から，前述した仮説「**HTML 文書中に現れる箇条書きやリストボックス，テーブルのセルなどの要素は，意味的に類似しており共通の上位語を持ちやすい**」に基づき，共通の上位語を持つであろうと考えられる意味的に類似した表現の集合を，それら表現を囲んでいる HTML タグ

に注目して獲得する．ステップ 1 は HTML 文書中のテーブル要素を転置する「前処理」，HTML 文書中のタグ情報に基づいて下位語候補集合を獲得する「下位語候補集合獲得処理」，獲得された下位語候補集合を整理する「後処理」の 3 つの処理からなる．以下本節では，各処理について説明する．

3.2.1 前処理

HTML 文書中の表データも下位語候補集合を獲得するうえで重要なデータである．後述する下位語候補集合獲得処理を 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集合を獲得することになる．しかし，吉田ら [13] によれば表データ中に現れる属性（例えば「血液型」）に対するその値（A 型，B 型，AB 型，O 型）は，行方向ではなく列方向に並びやすいという結果が得られている．これは，表データ中の類似した要素は行方向ではなく列方向に並びやすいということを示している．このことは，ブラウザにより HTML 文書を閲覧する場合，横方向ではなく縦方向に閲覧していく機会の方が圧倒的に多いということからも想像がつく．そこで，後述する下位語候補集合獲得処理により表データから意味的に類似した下位語候補集合を得るために，前処理として HTML 文書中に現れる表データの転置を行う．これにより，HTML 文書中に現れる表データの列方向に関して下位語候補集合を得ることが可能になり，意味的に類似したより多くの下位語候補集合を表データから獲得することが期待できる．

3.2.2 下位語候補集合の獲得処理

以下では図 3.2 に示した HTML 文書の一部を例に，下位語候補集合の獲得方法について述べる．下位語候補集合を獲得するにあたり，まず最初に 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，HTML 文書中の表現がどのようにタグ付けされているかを表すものであり，表現を囲んでいるタグをそのネストの順序にしたがって，リスト形式で表したものである．図 3.2 において，表現「今月のお買得！」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，表現「今月のお買得！」を囲む順序にしたがって並べれば，そのパスとして (UL，LI，今月のお買得！) が得られる．図 3.2 に示した HTML 文書中の各表現は以下のようなパスを持っている．

(UL，LI，今月のお買得！)
(UL，UL，LI，DVD–RW)
(UL，UL，LI，ハードディスク)
(UL，UL，LI，プリンタ)
(UL，UL，LI，スキャナ)
(UL，LI，PC ソフト)
(UL，UL，LI，ビジネス用途)
(UL，UL，LI，ホームページ作成)
(UL，UL，LI，新作ゲーム)

下位語候補集合獲得処理では，HTML 文書中に現れる同じパスを持つ表現同士をまとめ，下位語候補集合として獲得する．しかし，ただ単に同じパスを持つ表現を集めてきただけては意味的に類似した下位語候補集合を獲得することはできない．例えば図 3.2 の場合，同じパスを持つ表現同士をまとめると，

{DVD–RW，ハードディスク，プリンタ，スキャナ，ビジネス用途，ホームページ作成，新作ゲーム }
{ 今月のお買得 !，PC ソフト }

という 2 つの下位語候補集合が得られるが，周辺機器と PC ソフトのジャンルが混ざっていたり，関係のない表現同士であったりと，どちらの集合にも意味的な類似性をみることができない．この原因は同一タグの出現順序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考慮し，改めてパスを求めることにする．図 3.2 の場合だと，

(UL#1，LI#1，今月のお買得！)
(UL#1，UL#2，LI#1，DVD–RW)
(UL#1，UL#2，LI#2，ハードディスク)
(UL#1，UL#2，LI#3，プリンタ)
(UL#1，UL#2，LI#4，スキャナ)
(UL#1，LI#3，PC ソフト)
(UL#1，UL#4，LI#1，ビジネス用途)
(UL#1，UL#4，LI#2，ホームページ作成)
(UL#1，UL#4，LI#3，新作ゲーム)

というパスが得られる．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，同じパスを持つ表現を得ることができなくなる．そこで，表現からみて N 個前のタグまでは，タグの表記に出現順序を含めないようにする．図 3.2 の場合，$N = 1$ とすると

(UL#1，LI，今月のお買得！)
(UL#1，UL#2，LI，DVD–RW)
(UL#1，UL#2，LI，ハードディスク)
(UL#1，UL#2，LI，プリント)
(UL#1，UL#2，LI，スキャナ)
(UL#1，LI，PC ソフト)
(UL#1，UL#4，LI，ビジネス用途)
(UL#1，UL#4，LI，ホームページ作成)
(UL#1，UL#4，LI，新作ゲーム)

のようなパスを得ることができる．これらを同じパスを持つ表現ごとにまとめると，

{DVD–RW，ハードディスク，プリンタ，スキャナ }
{ ビジネス用途，ホームページ作成，新作ゲーム }
{ 今月のお買得 !，PC ソフト }

というように，意味的に類似した共通の上位語を持つであろう表現の集合を得ることが可能になる．本研究では経験的に $N = 1$ をとして下位語候補集合の獲得を行っている．

3.2.3 後処理

下位語候補集合獲得処理により獲得した下位語候補集合の要素間の意味的類似性をあげるために，ステップ 1 では後処理として獲得された下位語候補集合のうち，以下の条件に当てはまる下位語候補，もしくは下位語候補集合を削除する．

条件 1 文字列長が長い，もしくは文字種が頻繁に入れ替わる下位語候補

条件 2 表 3.1 に示した正規表現パターンに適合する下位語候補

条件 3 要素数が 3 個以下，もしくは 20 個以上の下位語候補集合

条件 1 に当てはまる下位語候補を削除する理由は，下位語候補集合獲得処理において，下位語候補として獲得されてしまった文を削除するためである．下位語候補獲得処理は，単に HTML 文書中の表現が持つパスしか考慮していないため，同じパスを持つ「語」の他にも，同じパスを持つ「文」も獲得してしまう．しかし，ステップ 1 では下位語を獲得することを目的としているため，下位語候補獲得処理で誤って下位語として獲得されてしまった文は削除する必要がある．そこで本研究では，文字列長が 12 以上の表現，もしくは文字種が 5 回以上入れ替わる表現を文として判断し，削除する．次に，条件 2 に当てはまる下位語候補を下位語候補集合から削除する理由は，表 3.1 に示した正規表現パターンに適合する下位語候補は，他の下位語候補と共通な特性を持ちにくいためである．表 3.1 に示したパターンに適合する表現を削除することで，獲得された下位語候補間の意味的な

表 3.1 不要語リスト 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.*	.* (.*

類似性の向上が期待できる．最後に要素数が 3 個以下，もしくは 20 個以上の下位語候補集合を削除する理由は，要素数が 3 個以下の下位語候補集合については，各下位語候補間に意味的な類似性が見られにくいためであり，要素数が 20 を越える下位語候補集合に関しては，以降のステップにおいて処理に多大な時間がかかってしまうためである．

3.3 df，idfに基づく上位語候補の獲得 (ステップ 2)

ステップ 1 では，HTML 文書中に現れる個々の表現が持つパスに注目することで，共通の上位語を持つであろうと考えられる下位語候補集合を獲得した．ステップ 2 ではステップ 1 で獲得した各下位語候補を含む文書中から，前述した 2 番目の仮説「**共通の上位語をもつ下位語の集合が与えられた時，各下位語に共通する上位語は各下位語を（少なくとも 1 つ）含む文書に現れやすく，それ以外の文書には比較的現れにくい**」に基づき，情報検索の分野などで従来より用いられている dfや idf といった統計量を利用して各下位語候補に共通する上位語候補を獲得する．

ステップ 2 では上位語候補の獲得を行うにあたり，まず 2 つの文書集合を準備する．1 つ目の文書集合は，大量の HTML 文書集合の中から無作為に選んだ HTML 文書からなるもので，これを**大域的文書集合**と呼ぶ．この文書集合は一般的な文脈においての単語の文書頻度を求める際に使用する．次いで 2 つ目の文書集合は，ステップ 1 で獲得された下位語候補集合の各要素を 1 つでも含む文書を，既存のサーチエンジンより収集し作成するもので，**局所的文書集合**と呼ぶ．この文書集合は与えられた下位語候補集合の各要素と，ステップ 2 で獲得する上位語候補の関連の強さを測る際に用いる．

以下では，ステップ 1 より獲得された下位語候補集合を C，大域的文書集合を G，C の各要素を検索語として WWW より収集した局所的文書集合を $LD(C)$ と記述する．また，$LD(C)$ に含まれる全ての名詞の中から，普通名詞，サ変名詞，地名を表す名詞を抽出し，その中から表 3.2 に挙げた不要語リストに含まれる語を削除して得られる名詞の集合を N とする．表 3.2 に示した不要語リストは，予備実験より得られた明らかに上位語にはなりにくい名詞，もしくは上位語として獲得されても価値の薄いと考えられる名詞からなる．ステップ 2 では，上位語候補 $h(C)$ を以下の式により求める．

$$h(C) = argmax_{n \in N}\{df(n, LD(C)) \cdot idf(n, G)\}$$

$$idf(n, G) = \log \frac{|G|}{df(n, G)}$$

ここで $df(n, D)$ は，文書集合 D 中で名詞 n を含む文書数を返す関数であり，$|G|$ は文書集合 G に含まれる文書数を表す．上式は，局所的文書集合中の多くの文書に現れ，かつ大域的文書集合中の文書には相対的にあまり現れない名詞を上位語候補として獲得する．

表 3.2 不要語リスト 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	長
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	方法	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	登	ヶ	ゴール	クリック
メール	ＭＡＩＬ	ＵＲＩ	ＴＨＥ			

また，下位語候補集合中の特定の要素のみに関連の強い語，例えば第 3 章冒頭で挙げた DVD–RW，ハードディスク，プリンタ，スキャナからなる下位語候補集合の例で言えば，プリンタに対する「インク」という語は，この時点で上位語候補として獲得されることはない．その理由は，全下位語候補を検索語として得られた文書集合中での各名詞の文書頻度を上式では用いているため，特定の下位語候補にのみ関連の強い語の文書頻度は，下位語候補全体に関連の強い語の文書頻度より低くなり，$df(n, LD(C)) \cdot idf(n, G)$ で求められるスコアも相対的に低くなりやすいからである．

それにも関わらず，この名詞のスコア付け方法だけでは十分な精度で妥当な上位語を獲得することができない．先程の式は上位語ではないが各下位語候補に非常に関連の深い語を上位語候補として獲得してしまうことがある．先程の例で言えば，上位語候補として「機器」が獲得されることが望ましいが，各下位語候補と関連の強い語である「データ」を獲得してしまう可能性もある．しかし，上位語ではないが全ての下位語候補と非常に関連の強いこのような語は，次のステップ 3 である程度除去することが可能である．

また本研究では，上位語候補を獲得する際，各名詞のスコア付けに $df(n, G)$ を用いているが，これを文書集合 G 中における名詞 n の出現頻度を求める関数 $tf(n, G)$ に変更することも可能である．$tf(n, G)$ に変更すると，従来より単語の重みを計算する際に利用されている *tf·idf* 法と同じになる．本研究では，各名詞のスコア付けに $tf(n, LD(C)) \cdot idf(n, G)$ を用いた場合についても上位語獲得実験を行った．しかし，$df(n, LD(C)) \cdot idf(n, G)$ を用いた場合と比べ，高い精度で正しい上位下位関係を獲得するには至らなかった．詳細につ

いては第 4 章にて述べる．

3.4 意味的類似度に基づく上位語候補と下位語候補集合の並べ替え (ステップ 3)

ステップ 2 までで下位語候補集合とその上位語候補の組を獲得することができた．これらの組の集合を以下の形式で表すことにする．

$$\{\langle h(C_1), C_1\rangle, \langle h(C_2), C_2\rangle, \cdots, \langle h(C_m), C_m\rangle\}$$

ここで $C_1, \cdots, C_m$ はステップ 1 より獲得された下位語候補集合を表しており，$h(C_i)$ はステップ 2 で獲得された C_i に対する上位語候補を表している．ステップ 3 では，先述した 3 番目の仮説「**上位語と下位語は意味的に類似しており，そ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捉えることができる**」に基づき，$h(C_i)$ と C_i の各要素の持つ係り受け関係から両者の意味的類似度を計算し，その類似度に基づいて $\langle h(C_i), C_i\rangle$ 各組の順位付けを行う．本研究で提案する手法は，ステップ 3 で求める順位に基づき，後述するステップ 4 を適用後，その上位 k 組を本提案手法によって獲得された最終的な上位下位関係として出力する．すなわち，残りの $m-k$ 組は，間違った上位語が獲得されやすいという理由から削除する．

3.3 節でも述べたようにように，ステップ 2 の結果には上位語ではないが各下位語候補と非常に関連の強い語を上位語候補として獲得してしまっている組が存在する．例えば，第 3 章冒頭で挙げた DVD–RW，ハードディスクなどからなる下位語候補集合の例に対し，その上位語候補として「機器」ではなく「データ」を獲得してしまっているケースもある．ステップ 3 では，このような誤った上位語が獲得されている上位語候補と下位語候補集合の組を，最終的な出力結果から削除する．本研究では，下位語候補集合の各要素と関連は強いが上位語ではない語（先程の例でいえば「データ」）は，下位語候補集合の各要素との意味的な類似性が弱く，逆に妥当な上位語（「機器」）は意味的な類似性が強いと考える．そのため，上位語候補と下位語候補間の意味的類似度に従って上位語候補と下位語候補集合の組を順位付けすれば，誤った上位語が獲得されている組に対して低い順位を付けることが期待できる．このように，誤ったものを低く，正しいものを高く順位付けし，その上位幾つかを最終的な獲得結果とすることで，相対的に高い精度で上位下位関係の獲得が可能になる．

ステップ 3 では上位語候補と下位語候補間の意味的類似度を次のようにして計算する．まず，局所的文書集合中から，各下位語候補が含まれる文を抽出し，係り受け解析を行う．そして，係り受け解析の結果から各下位語候補がどのような動詞に係りやすいかを計算する．ここで，下位語候補集合 C の要素のいずれかが助詞 p を介して動詞 v に係る頻度を，$f_{hypo}(C, p, v)$ と記述する．そして，すべての助詞を $\{p_1, \cdots, p_l\}$，すべての動詞を $\{v_1, \cdots, v_m\}$ で表したとき，下位語候補全体の係り受け関係を表したベクトル（以降では

この係り受け関係を表したベクトルのことを “**係り受けベクトル**” と呼ぶ）を以下のように定義する．

$$hypov(C) = \langle f_{hypo}(C, p_1, v_1), f_{hypo}(C, p_2, v_1), \cdots, f_{hypo}(C, p_{l-1}, v_m), f_{hypo}(C, p_l, v_m)\rangle$$

ここで，下位語候補全体から係り受けベクトルを生成している理由は，単独の下位語候補の係り受け関係だけでは，出現頻度が低いために的確な係り受け関係を捕らえているとは考えにくいためである．続いて下位語候補集合と同様，上位語候補 $h(C)$ の係り受けベクトルを次のように定義する．

$$hyperv(h(C)) = \langle f(h(C), p_1, v_1), f(h(C), p_2, v_1), \cdots, f(h(C), p_{l-1}, v_m), f(h(C), p_l, v_m)\rangle$$

ここで $f(h(C), p, v)$ は，新聞記事 33 年分[1]より求めた，上位語候補 $h(C)$ が助詞 p を介して動詞 v に係る頻度を表している．この時，新聞記事中に 500 回以上現れない語に関しては，出現頻度が少ないために正しい係り受け関係を得ることができないという観点から，係り受け関係の学習は行わなかった．そのため，このような語が上位語候補として獲得された組に関しては，上位語候補と下位語候補全体との類似度を 0 とした．また，今回係り受けベクトルを作成するために，新聞記事より学習した係り受けデータを用いた理由は，単に大量の新聞記事が既に構文解析済みであったためである．WWW 上から大量の文書を収集し，それらから的確な語の係り受け関係を求めることは今後の課題である．

下位語候補全体と上位語候補の意味的な類似度は，両者の係り受けベクトルの類似度で求める．2 つのベクトルの類似度を求める方法は幾つかあるが，本研究では文書検索などに用いられているコサイン尺度 [11] を用いてベクトルの類似度を計算した．任意の下位語候補集合 C とその上位語候補 $h(C)$ の意味的類似度は以下の式で計算される．

$$sim(h(C), C) = \frac{hypov(C) \cdot hyperv(h(C))}{|hypov(C)| \times |hyperv(h(C))|}$$

ステップ 3 では，ステップ 2 までで獲得された上位語候補と下位語候補集合の組 m 個からなる集合 $\{\langle h(C_i), C_i\rangle\}_{i=1}^{m}$ を以下のスコアに基づいて並べ替える．

$$sim(h(C_i), C_i) \cdot df(h(C_i), LD(C_i)) \cdot idf(h(C_i), G)$$

ここで並べ替えを行う際に，意味的な類似度だけではなく，ステップ 2 で計算された上位語候補の df·idf 値も考慮していることに注意されたい．

また上記のスコアの異なる利用方法として，1 つの下位語候補集合に対して，ステップ 2 より獲得された上位 j 個の上位語候補を上式を用いて改めて順位付けをし，そのトップを上位語候補として獲得するという方法も考えられる．本研究でも，同様の手法を実装し上位下位関係獲得の性能を評価した．しかし，上記のスコアを用いて上位語候補を改めて順位付けすることによる有意義な精度の向上は見られなかった．その詳細については第 4 章で述べる．

[1]読売新聞 1987–2001，毎日新聞 1991–1999，日経新聞 1990–1998; 計 3.01GB

3.5 ヒューリスティックなルールを用いた上位語候補と下位語候補集合の組の取捨選択 (ステップ 4)

ステップ 4 では，予備実験より得た知見に基づき作成したヒューリスティックなルールを，ステップ 3 までで得られた上位語候補と下位語候補集合の組に適用することで，上位下位関係の獲得精度の改善をはかる．ステップ 4 で使用したルールは以下の 3 つである．

ルール 1 獲得された上位語候補を検索語として検索エンジンに問い合わせた，その結果得られるヒット件数が，各下位語候補を検索語として得られたヒット件数の総和よりも少ない場合，その上位語候補と下位語候補集合の組を削除する

ルール 2 獲得された上位語候補が，下位語候補集合のいずれかの要素の部分文字列として現れていた場合，以下の条件に当てはまるような上位語候補と下位語候補集合の組は削除する

- 上位語候補が下位語候補の末尾以外の場所で部分文字列として現れている
- 下位語候補集合の半分以上の要素について上位語候補が末尾に現れていない

ルール 3 獲得された上位語候補が地名を表す語である場合，上位語候補を「地名」に変更する

ルール 1 では，誤って獲得された上位語候補を持つ組を削除することで精度の改善をはかる．一般に，上位語は下位語と比べより広い文脈で使われており，下位語候補を含む文書より上位語候補を含む文書の方が WWW 上により多く存在しているはずと考えることができる．このような一般的な上位語が持っていると考えられるこの特性を利用したのがルール 1 である．

次いでルール 2 では，誤った上位語候補が獲得されている組を削除するのに加え，意味的類似性の見られない下位語候補集合を持つ組についても削除することで精度の改善をはかる．しかし，このルールは獲得された上位語候補が下位語候補の部分文字列として現れない場合には適用されない．日本語では，複合名詞の主辞は主として末尾に現れる名詞であるため，下位語候補集合の多くの要素で共通の語が末尾に現れている場合，その語は妥当な上位語である可能性が高いと考えられる．それに対し，獲得された上位語候補が下位語候補の末尾以外の場所に現れる場合，その上位語候補は妥当な上位語である可能性は低いと考えられる．また，下位語候補集合の一部の要素の末尾にだけ上位語候補が現れる場合，その下位語候補集合は意味的な共通性が見られ難く，そのような下位語候補集合は共通する上位語も持ち難いと考えられる．そのため，このような条件に該当する上位語候補や下位語候補集合を持つような組は最終的な出力結果から削除する．

最後にルール 3 では，獲得された誤った上位語候補を正しい上位語に置換することで精度の改善をはかる．予備実験において，下位語候補集合の要素が地名の場合，それら地名を含む地域を指す地名が，上位語候補として獲得されているケースが頻繁に見られた．

例えば，下位語候補集合が「京都」,「東京」,「大阪」,「石川」という表現からなっていた場合，妥当な上位語としては「地域」や「都道府県」などが考えられるが，ステップ 3 までで述べた方法で上位語候補を求めると,「日本」という結果が得られる．実際に獲得された上位語候補「日本」は，本研究でたてた仮説を満足するが,「京都」や「東京」,「大阪」,「石川」に対して「日本」という語は包含関係 (part-whole relation) を表す語であり，上位語ではない．そこで，地名を表す表現からなる下位語候補集合に対しても正しい上位語を獲得できるようにするため，獲得された上位語候補が地名を表す語であった場合は，それを「地名」に置き換える．

以上のルールをステップ 3 までで得られた結果に対して適用することで，幾つかの上位語候補と下位語候補集合の組が削除または修正され，性能のさらなる向上が見込めることを実験により検証する．

第4章　提案手法の評価実験

4.1　準備

本研究では，フリーのファイルダウンロードソフトである wget[1] を使用し，約 4.66×10^6 件の HTML 文書（重複なし）を WWW よりダウンロードした．そして，ダウンロードした文書集合の中から 1.00×10^6 件の HTML 文書 (約 1.26GB，タグなし) を無作為に選びだし，新しく文書集合を作成した．作成した文書集合は，一般的な文脈においての単語の文書頻度を求めるために用いる大域的文書集合として上位語候補を獲得する際に用いる．大域的文書集合から単語の一般的な文書頻度を求めるにあたり，大域的文書集合中に含まれる各 HTML 文書からタグを除去し，JUMAN[18] を用いて形態素解析を行った．そして，各単語がどのくらいの数の HTML 文書に含まれているかを計算し，単語の文書頻度を表すリストを生成した．この時，文書頻度が 30 に満たない単語に関しては，的確な文書頻度が得られていないと考え，リストから除外している．

次いで，評価実験に用いる下位語候補集合を獲得するため，先程の 4.66×10^6 件の文書集合より，約 8.71×10^5 件の HTML 文書 (10.4GB，タグあり) を選びだした．そして，HTML 文書中で省略されている終了タグの補完や，誤ったタグの入れ子構造を持つ部分を適切な入れ子構造への変換を行う，フリーのユーティリティである HTML Tidy[2] を用いて，各 HTML 文書を XML 文書へ変換した．その結果，5.90×10^5 件の XML 文書を得た．なぜこのような処理を行ったかというと，終了タグが省略されてしまっていたり，タグの入れ子構造が間違っているような HTML 文書からは，DOM（Document Object Model）を生成することができないためである．DOM とは，W3C[3] により開発されている，XML 形式で記述されたデータへアクセスするための標準的なインターフェース及びオブジェクトモデルのことであり [12]，本研究では DOM を利用して XML 形式に変換された各 HTML 文書から木構造を生成し，下位語候補集合の獲得を行っている．HTML の規格では終了タグの省略を許しているタグが複数存在するため，実際にダウンロードした一部の HTML 文書を眺めると，省略の許されている終了タグを省略している場合が多々ある．同様に，HTML 文書制作者の記述ミスにより，HTML 文書中に現れるタグの入れ子構造が間違っているような文書も存在する．このような文書を，終了タグの省略や，タグの入れ子構造の誤りなどが許されていない[4]，HTML よりも厳格な規格を持つ XML に

[1] http://www.gnu.org/software/wget/wget.html
[2] http://www.w3.org/People/Raggett/tidy/
[3] http://www.w3c.org
[4] HTML の規格でもタグの入れ子構造の誤りは許されていない．しかし，そのような HTML 文書をブラ

変換することで，適切なDOMを生成することができ3.2節で述べた手法により下位語候補集合の獲得が可能になる．

次に3.2節で述べた，手法により，5.90×10^5 件のXML文書から下位語候補集合の獲得を行った結果，9.02×10^4 個の下位語候補集合（重複あり，全部で 6.01×10^5 個の下位語候補を含んでいる）を獲得した．実際に獲得された下位語候補集合の例を表4.1に示す．続いて，9.02×10^4 個の下位語候補集合の中から重複を除き，無作為に選択した2,000個を評価実験に用いるテストセットとした（本手法の開発には，約4,000個の下位語候補集合を用いている）．このテストセットとして選択した2,000個の下位語候補集合には，全部で13,790個の下位語候補が含まれている．

最後に，3.3節で述べた方法で上位語候補を獲得するにあたり，個々の下位語候補を検索語として検索エンジンgoo[5]より検索し，その結果得られた文書集合のうち上位100件をそれぞれダウンロードして局所的文書集合を作成した．この時，検索結果が100件に満たない下位語候補に関しては，検索により得られた全ての文書をダウンロードした．作成した局所的文書集合は，大域的文書集合と同様，その中に含まれる各HTML文書からタグを除去し，JUMANを用いて形態素解析を行った．続いて，下位語候補全体の係り受けベクトルを以下の手順で求めた．まず，局所的文書集合中の文書から下位語候補を含む文を抜き出してJUMANを用いて形態素解析し，その結果に含まれる下位語候補をすべて文字列「KEYWORD」で置換した．これは，下位語候補全体で係り受けベクトルを求めるのを簡単にするためである．この時，HTML文書から文を抽出する際，全ての英数字を全角に変換しているため，はじめから文中に現れている「KEYWORD」という文字列の係り受け関係まで係り受けベクトルに誤って含まれることはない．そして，すべての下位語候補を文字列「KEYWORD」に置換した形態素解析結果に対し，既存の構文解析器[6][20]を用い係り受け解析を行った．

4.2 評価実験

本研究では，提案手法の評価実験として以下に示す3つの実験を行った．まず始めに，提案手法により獲得された上位語候補と下位語集合の組のうち，上位200個について，獲得された上位語候補が下位語候補集合の各要素に対して正しいかどうかを人手（本人）で評価した．ここで，本研究で提案した手法により獲得された上位語と下位語集合の組は，獲得された上位語の $sim(h(C), C) \cdot df(h(C), LD(C)) \cdot idf(h(C), G)$ のスコアでソートされていることに注意されたい．

次いで2つ目の実験として，本研究で上位下位関係を獲得する際に提案したステップ2，3，4，及びステップ4で使用しているヒューリスティックなルール1，2，3が，それぞれ

ウザで閲覧すると，ブラウザが誤ったタグの入れ子構造を都合の良いように解釈し，レンダリングを行うため，HTML文書の閲覧者はそのような誤りに気づきにくい．

[5]http://www.goo.ne.jp/

[6]論文[20]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．

表 4.1 実際に獲得された下位語候補集合の例

ID	下位語候補集合の要素
10397	広井法代, 山川純子, 池田和子, 柏木久美子
13364	家族パス, 少年パス, 成人パス, 青年パス
16653	わからない, 回だけ, 回以上, 回以上回未満
21561	あじさい茶屋, まぐろ市場, クローバー, ドトール, ドムドム, ベックス, ラガール, ランパデール, 小竹林, 東神奈川そば店, 道中そば, 本郷台そば店
28931	DDI ポケット, ドコモ関西, ドコモ四国, ドコモ東海, ドコモ北陸, 九州通信ネットワーク, 四国情報通信ネットワーク
30288	違法でない, 違法又は不当, 勧告, 義務を果たしているか, 通知, 日以内, 理由がある, 理由がない
35645	阿部委員, 紫芝委員, 津金委員, 眞柄委員
42817	たまソフト, テトラテック, テリオス, トラヴュランス, トロピカルソフト, トンキンハウス, 天津堂
51462	旧海運局, 旧陸運局, 工事発注見通しの公表, 所在地, 情報公開
53147	なし, 円, 朝回, 不可
56681	NTT 東日本, アイコム, オムロン, コレガ, プラネックス, マイクロ総研, メルコ, ヤマハ
58174	監督, 原作, 作品A, 時間, 出演
59502	さんた, ふるみそ, ほぴの, やまおり, やまそと
60446	会計掛, 雑誌情報掛, 参考調査掛, 資料サービス掛, 庶務掛, 図書受入掛, 相互利用掛, 電子情報掛, 洋書目録情報掛, 和書目録情報掛
69064	学会, 技術, 研究, 工学, 通信, 電子, 福祉
81250	異常接近図, 接触事故現場図, 損傷した米軍偵察機, 米太平洋軍司令部
81347	エアクリーナ, エンジンオイル, オイル漏れ, ステアリングラックブーツ, タイヤ, ドライブシャフト, ドライブシャフトブーツ, ブレーキパッド, ブレーキフルード, ブレーキライニング, ベアリング類, マフラー, ライト類
84414	リスク人年, 交絡, 信頼区間, 統計的有意性, 盲検化
87815	こころのくすり最新事情, 生命の冒険上, 天才たちの宇宙像, 脳ミソを哲学する
89958	よたれダンス, 心頭滅却, 人形劇, 念動拳, 念動爆砕拳
91921	ｈｉｒｏの伝言板, これだけの情報を, 画像転送伝言板, 恐竜王国勝山, 新型画像ボード, 福井のプロバイダみてね
93842	ハンドル, バウ, バウサイド, パドル, ビッグブレード, ピッチ, ピッチング, フィニッシュ, フォア, フォワード, ブレード, ペア, ポートサイド, ポンド, 腹切り
101220	一般財形, 期日指定定期, 住宅財形, 年金財形
102908	フランクフルト, アルコール, コーンポタージュ, タイヤキクリーム, ビール, ドリンク, みかん飴, ベーコンチーズ, ラテンな飲み物, 甘酒, 焼きおにぎり
105270	学園天国, 学校, 受験, 進路, 制服, 席替え, 部活, 忘れたくない思い出, 北辰, 旅行
107754	SCSI カード, ケーブル, サウンドカード, メディア
109103	いわき, 会津, 喜多方, 局計, 郡山, 須賀川, 相馬, 白河, 富岡, 福島
117830	エリアを選択, 江東区, 港区, 荒川区, 渋谷区, 新宿区, 杉並区, 世田谷区, 千代田区, 台東区, 大田区, 中央区, 中野区, 板橋区, 品川区, 文京区, 豊島区, 目黒区
121554	再び保存, 再確認, 変更を加えてみよう, 保存しましょう
125072	加佐志-, 狭山台-, 掘兼, 上奥富, 上奥富-, 新狭山-, 柏原, 堀兼-

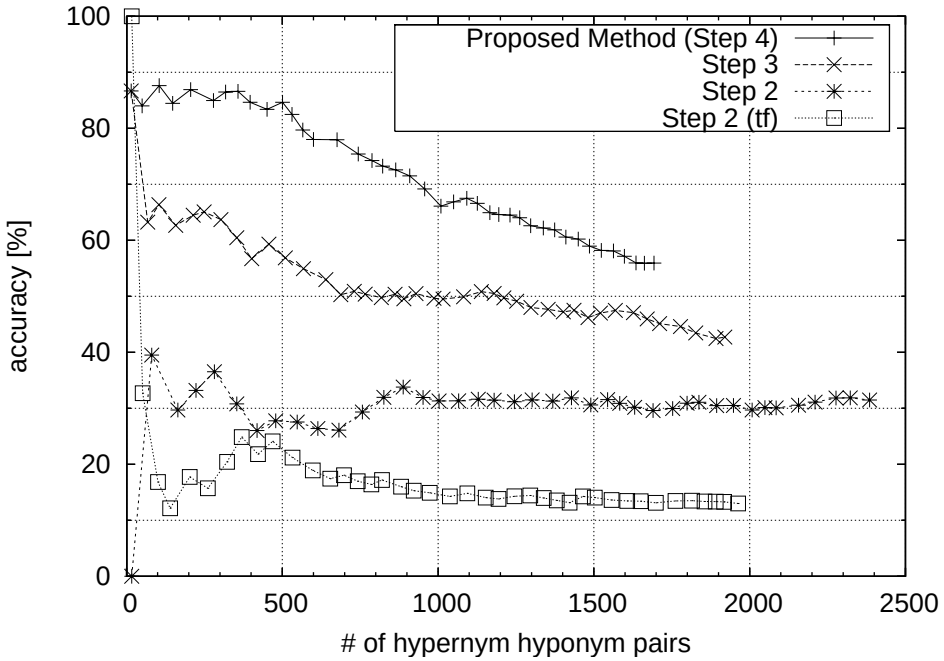

図 4.1 各ステップを経ることでの精度の移り変わり

正しい上位語を獲得するために有効に働いているかを確認するため，それぞれのステップ及びルールを抜いた場合での，上位下位関係の獲得実験を行い，本研究で提案した各ステップ及び各ルールが上位下位関係を獲得する際に有効に働いているかどうかの検証を行った．

そして最後に本研究で用いた上位語候補のスコア $sim(h(C), C) \cdot df(h(C), LD(C)) \cdot idf(h(C), G)$ の利用方法が妥当であることを確認するため，3.4 節で述べたように $df \cdot idf$ スコアにより獲得された上位語候補の上位幾つかを，類似度考慮したこのスコアで改めてランキングし，そのトップを上位語候補として獲得する方法との比較実験を行った．

以下，本節では上で述べた評価実験の結果について述べる．

4.2.1 提案手法の精度の評価実験

最初の評価実験として，ステップ 2，3，4 を経ることにより上位下位関係の獲得精度が変化することを確認するため，各ステップごとに獲得された上位下位関係の評価を行った．表 4.2 及び図 4.1 に，ステップ 2，3，4 を経ることによって変化する精度の様子を示す．各ステップでは，獲得された上位語候補と下位語候補集合の組のソートを行っている．ソートの基準として，ステップ 3，4 では獲得された上位語候補の $sim(h(C), C) \cdot df(h(C), LD(C)) \cdot idf(h(C), G)$ の値を，ステップ 2 では $df(h(C), LD(C)) \cdot idf(h(C), G)$ の値を用いている．つまり，ステップ 2 までで獲得された結果をソートする時は，上位語候補と下位語候補集合の類似度は考慮されていない．また，各ステップでは上位語候補と下位語候補集合の組

のソートを行った後，全体の 1 割にあたる上位 200 組を最終的に獲得された上位下位関係として評価対象にしている．残りの 1,800(=2,000–200) 組は，前述したように間違った上位語候補が獲得されやすいという観点から評価対象から外した．

本研究では，上位下位関係獲得の精度を次のようにして求めている．まず，ソートされた上位語候補と下位語候補集合の組のうち，上位 n 組を取り出す．そして，その中で正しい上位下位関係が獲得されている割合を計算し，それを上位下位関係獲得の精度とした．グラフの横軸は，ソートされた上位語候補と下位語候補集合の組の上位 n 組中に含まれる下位語候補の数を示しており，縦軸は正しい上位語が獲得された下位語候補の割合を示している．つまり，グラフの各線は以下の式に従って描画されている．

$$\langle \sum_{k=1}^{j} |C_k|, \frac{\sum_{k=1}^{j} correct(C_k, h(C_k))}{\sum_{k=1}^{j} |C_k|} \rangle$$

ここで j は $1 \leq j \leq 200$ であり，$|C_k|$ は下位語候補集合 C_k の要素数である．さらに $correct(C_k, h(C_k))$ は実際に獲得された上位語候補 $h(C_k)$ が正しい上位語である下位語候補集合 C_k 中の要素数を表している．

図 4.1 中で “Step 4” と示した曲線が，今回提案した手法により獲得された上位下位関係の獲得精度を表している．この図より獲得された全上位下位関係数の約 3.6 %にあたる上位 501 個（獲得された上位語の異なり数は 36 個）の関係を取り出した場合，その精度はおよそ 84.6 %を示している．さらに，全関係数の約 5 %にあたる上位 701 個（獲得された上位語の異なり数は 44 個）の関係を取り出した場合，その精度はおよそ 75 %，約 10 %にあたる上位 1398 個（獲得された上位語の異なり数は 93 個）の場合で 61 %と，獲得する上位下位関係数を多くするほど徐々にその精度が落ちていくことが図よりわかる．このことから，獲得された上位語候補の $sim(h(C), C) \cdot df(h(C), LD(C)) \cdot idf(h(C), G)$ の値に基づいて，上位語候補と下位語候補集合の組をソートすることで，正しい上位語が獲得されている組に対しては高い順位を，誤った上位語が獲得されている組に対しては低い順位をつけることができていることがわかる．

3.3 節で述べたように，ステップ 2 において $df(n, LD(C)) \cdot idf(n, G)$ ではなく，$tf(n, LD(C)) \cdot idf(n, G)$ により各名詞のスコア付けを行い上位語候補を獲得した結果が図 4.1 中の “Step 2(tf)” で示したグラフである．図より，$tf(n, LD(C)) \cdot idf(n, G)$ に比べて $df(n, LD(C)) \cdot idf(n, G)$ の方が上位語候補獲得の精度が高いことが確認できる．この結果から上位語候補獲得というタスクにおいては，$tf(n, LD(C)) \cdot idf(n, G)$ より $df(n, LD(C)) \cdot idf(n, G)$ の方が適していることがわかる．

表 4.3 に実際に獲得された上位語と下位語候補集合の例を示す．下位語候補集合の幾つかの要素がその末尾に共通の語を持っている場合，その語が上位語になりやすいというのは自明であるため，表 4.3 にそのような例は載せていない．また，この表はステップ 4 で獲得された最終的な結果ではなく，ステップ 3 により獲得された上位語候補と下位語候補集合の組の結果を示している．表 4.3 より，ステップ 4 の各ルールを適用することで，ステップ 3 で獲得された結果のうち幾つかが修正または削除される様子が確認できる．

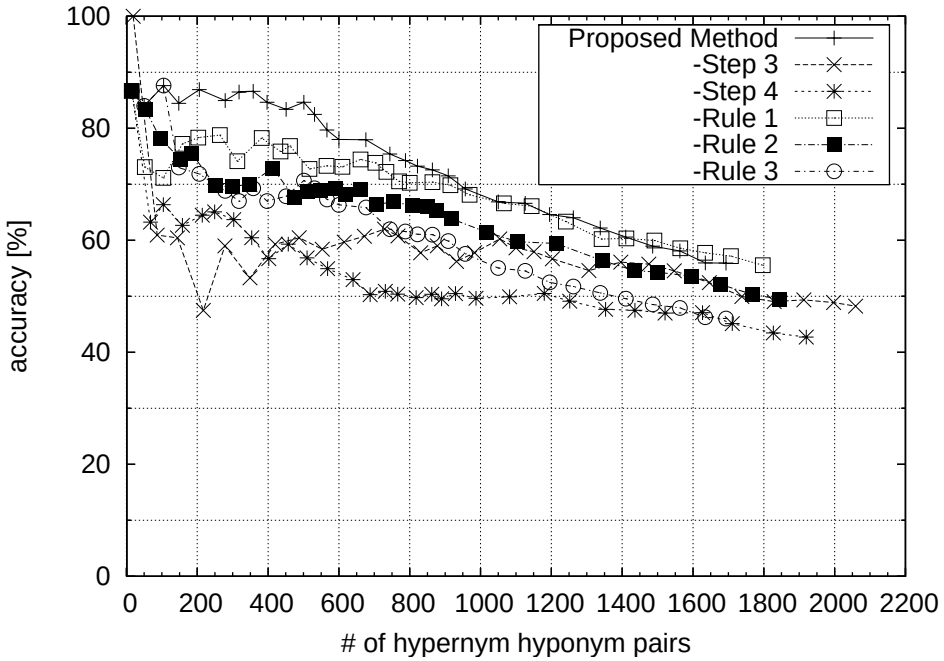

図 4.2 各ステップ及び各ルールの効果

4.2.2 各ステップ及び各ルールの有効性の評価

次にステップ 3，4，およびステップ 4 で用いている各ルールがどのくらい精度の向上に貢献しているのかを確認するため，各ステップ，もしくは各ルールをそれぞれ抜いた時での，上位下位関係獲得の性能を評価した．ステップ 3，4 を抜いた時の結果を表 4.4 に，ステップ 4 の各ルールを抜いたときの結果を表 4.5 に示す．さらに，これらの表から作成したグラフを図 4.2 に示す．図 4.2 において “– Step X” もしくは “– Rule X” となっているものは，今回提案した手法から “ステップ X”，または “ルール X” を抜いた時の精度を表している．また，ステップ 3 を抜いた場合は，各上位語候補と下位語候補集合の組を獲得された上位語候補の $df(n, LD(C)) \cdot idf(n, G)$ の値に基づいてソートしている．

図 4.2 より，どのステップやルールを抜いた場合でも，上位下位関係獲得の精度が低下しているため，どのステップやルールも精度の向上に有効に働いていることが確認できる．上位 200 組の上位語候補と下位語候補集合の組を獲得した時，ステップ 3 を抜いた場合で 7.7 [%]，ステップ 4 を抜いた場合では 13.2 [%] 程度の精度の低下が見られる．このことから，ステップ 3 よりもヒューリスティックなルールを適用することで精度の向上を図るステップ 4 の方が，精度の向上により働いていることがわかる．また，ステップ 4 の中でもとりわけ，地名を表す上位語を「地名」に変換するルール 3 を抜いた場合が最も精度が落ちていることから，提案手法により獲得された上位下位関係の中には地名に関するものが多く含まれていると考えられる．実際，提案手法により獲得された正しい上位下位関係の数は 947 個であり，ルール 3 を抜いた場合では 779 個に減少していることが表

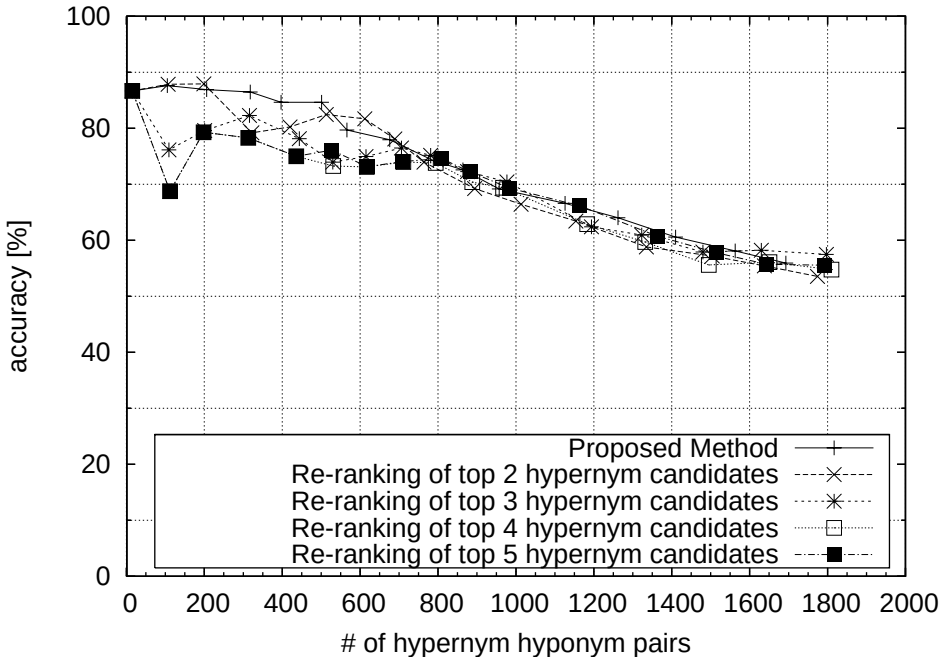

図 4.3 リランキングの効果

4.5 よりわかる．この数字から，正しく獲得された上位下位関係のうち，およそ 17.7[%] は HYPONYM(“地名”, “東京都”) のような上位下位関係で占められていることがわかる．

4.2.3 リランキングによる効果の評価

3.4 節で触れたように，ステップ 2 より獲得された上位 j 個の上位語候補を $sim(h(C), C) \cdot df(h(C), LD(C)) \cdot idf(h(C), G)$ の値に基づき改めて順位付けを行い，そのトップを上位語候補として獲得した場合の，上位下位関係獲得実験の結果を表 4.6 及び図 4.3 に示す．図より，$sim(h(C), C) \cdot df(h(C), LD(C)) \cdot idf(h(C), G)$ の値を用いて上位語候補をリランキングしても，有意義な精度の向上がみられないのがわかる．

表 4.2 各ステップ終了時点での上位下位関係獲得の精度

順位	上位下位関係獲得の精度 (正しく上位語が獲得された下位語候補数/下位語候補の総数)			
	提案手法 (ステップ 4)	ステップ 3	ステップ 2 (df·idf)	ステップ 2 (tf·idf)
1	86.667 (13/15)	86.667 (13/15)	0.000 (0/17)	100.000 (17/17)
5	84.000 (42/50)	63.235 (43/68)	39.506 (32/81)	32.692 (17/52)
10	87.619 (92/105)	66.346 (69/104)	29.697 (49/165)	16.832 (17/101)
15	84.459 (125/148)	62.658 (99/158)	33.184 (74/223)	12.143 (17/140)
20	86.893 (179/206)	64.486 (138/214)	36.525 (103/282)	17.734 (36/203)
25	84.946 (237/279)	65.060 (162/249)	30.791 (109/354)	15.709 (41/261)
30	86.478 (275/318)	63.696 (193/303)	26.014 (109/419)	20.433 (66/323)
35	86.592 (310/358)	60.452 (214/354)	27.766 (133/479)	24.865 (92/370)
40	84.635 (336/397)	56.716 (228/402)	27.555 (151/548)	21.801 (92/422)
45	83.370 (376/451)	59.300 (271/457)	26.384 (162/614)	24.094 (113/469)
50	84.631 (424/501)	56.863 (290/510)	26.100 (178/682)	21.201 (113/533)
55	82.486 (438/531)	54.930 (312/568)	29.326 (222/757)	18.896 (113/598)
60	79.682 (451/566)	52.969 (339/640)	31.961 (264/826)	17.431 (114/654)
65	78.000 (468/600)	50.291 (346/688)	33.784 (300/888)	18.052 (126/698)
70	77.959 (527/676)	50.889 (372/731)	31.933 (304/952)	16.981 (126/742)
75	75.403 (561/744)	50.392 (386/766)	31.306 (314/1003)	16.412 (129/786)
80	74.239 (585/788)	49.756 (407/818)	31.332 (334/1066)	17.195 (141/820)
85	73.236 (602/822)	50.348 (434/862)	31.621 (357/1129)	16.005 (141/881)
90	72.569 (627/864)	49.551 (441/890)	31.441 (371/1180)	15.293 (141/922)
95	71.507 (650/909)	50.484 (469/929)	31.165 (388/1245)	14.902 (145/973)
100	69.175 (662/957)	49.645 (490/987)	31.490 (410/1302)	14.258 (148/1038)
110	66.857 (702/1050)	49.908 (540/1082)	31.863 (455/1428)	14.050 (162/1153)
120	66.607 (750/1126)	50.508 (596/1180)	31.541 (487/1544)	14.297 (178/1245)
130	64.603 (772/1195)	49.121 (615/1252)	30.166 (492/1631)	13.966 (187/1339)
140	64.025 (808/1262)	47.672 (645/1353)	29.949 (525/1753)	13.150 (187/1422)
150	62.182 (832/1338)	47.493 (682/1436)	31.083 (571/1837)	14.039 (211/1503)
160	60.567 (854/1410)	47.009 (715/1521)	30.493 (594/1948)	13.441 (216/1607)
170	58.950 (876/1486)	47.081 (766/1627)	30.112 (617/2049)	13.125 (223/1699)
180	58.093 (908/1563)	45.120 (772/1711)	30.552 (659/2157)	13.499 (245/1815)
190	55.963 (915/1635)	43.459 (794/1827)	31.840 (725/2277)	13.326 (252/1891)
200	55.936 (947/1693)	42.708 (820/1920)	31.488 (751/2385)	12.990 (255/1963)

表 4.3 提案手法により獲得された下位語集合とその上位語の例

Step4 での順位	Step1 で獲得された下位語候補集合	Step2 で獲得された上位語	Step3 での順位	Step4 のルール			Step4 で獲得された上位語
				1	2	3	
10	朗読者*, オブジェクト指向入門*, 月の砂漠をさばさばと*, もこもこもこ*, ソフトウェア職人気質*, 入門 JavaScript*	本	23	−	−	−	本
16	テディベア*, チョウカイリョウガ*, ヨシフサキング*, プラントタイヨオー*, ナスノホシヒメ*, フローレスライン*, ノーザンカピタン*, ミヤビリージェント*, クラレットパンチ*, トーセンダンディ*, アーサーズフェイム*, ケイアイチャンス*, ロイスジュニア*, カナハラドラゴン*, ウインシュナイト*, ダイワサイレンス*, マチカネラッパ* , マイネルグリズリー*, ミスタードン*	馬	42	−	−	−	馬
21	コオリガモ*, ビロードキンクロ*, アカハジロ*, クビワキンクロ*, メジロガモ*, アカハシハジロ*, キンクロハジロ*, コケワタガモ*, スズガモ*, ホオジロガモ*, シノリガモ*, クロガモ*, ホシハジロ*, ケワタガモ*, ヒメハジロ*, アラナミキンクロ*, オオホシハジロ	鳥	53	−	−	−	鳥
29	殺人*, 放火*, 強姦*, 侵入盗*, 侵入強盗*, 非侵入盗*, 非侵入強盗*	犯罪	68	−	−	−	犯罪
47	将軍*, 宮本武蔵*, 羅生門*, 七人の侍*, ミッドウェイ*, 無法松の一生*, 太平洋の地獄*, 武士道ブレード*, 価値ある男*, 用心棒』『赤ひげ, 大統領の堕ちた日*	映画	112	−	−	−	映画
69	モスクワ*, キエフ*, タシケント*, ミンスク*, トビリシ*, ドゥシャンベ*, ビシュケク*, アスタナ*, キシニョフ*, エレバン*, バクー*, アシハバード*	ロシア	169	−	−	+	地名
78	福留宏紀*, セギノール*, 藤井康雄*, シェルドン*, 五島裕二*, 玉木朋孝*, 塩谷和彦*, 平野恵一*,	選手	196	−	−	−	選手
81	ワイヤレスカード, 小電力セキュリティ, ＰＨＳ陸上移動局, 市民ラジオ, 特定小電力機器,	無線	200	−	−	−	無線
82	大切なもの*, もらい泣き*, 大きな古時計*, 星屑の街*, 白い花*, 未完成のメロディ*	曲	201	−	−	−	曲
86	踊る大捜査線*, プロジェクトX, 世紀を越えて, 彼女たちの時代*	ドラマ	207	−	−	−	ドラマ
106	桑田真澄*, 上原浩治*, ワズディン*, 武田一浩*, 木村龍治*, 真田裕貴*, 鄭ミン台*, 趙成ミン*	投手	250	−	−	−	投手
116	イワウメ*, チシマザサ, キバナシャクナゲ*, ミヤマナルコユリ*	花	280	−	−	−	花
127	シイタケ*, サンゴハリタケ*, サンコタケ*, シロオニタケ*, シロイボカサタケ*	キノコ	306	−	−	−	キノコ
139	音楽, 映画, マンガ, 出会い, 芸能人	サイト	324	−	−	−	サイト
150	夏目漱石, 芥川竜之介, 鷹野つぎ, 国木田独歩, 徳冨蘆花, 菊池寛, 若山牧水, 梶井基次郎, 夢野久作, 宮本百合子, 田中貢太郎, 夢野久作海若藍平 ブレイクウィリアム,	作品	343	−	−	−	作品
172	新年, 万聖節, 主顕節, メーデー, クリスマス, イースター, 解放記念日, 聖母受胎祭, 聖ステファノの日, 聖母昇天祭	日本	391	−	−	+	地名
−	銀河群, 構成メンバー, 局部銀河群 アンドロメダ銀河*, 銀河系*,	銀河	10	−	+	−	−
−	ブラジル, フィリピン, 韓国, インド, アメリカ, タイ 中国, ペルー, オーストラリア, アルゼンチン, スペイン	日本	80	+	−	+	−

“*” が後についている下位語候補は，提案手法により妥当な上位語が獲得されたものを示す.

表 4.4 ステップ 2，3 を抜いた時の上位下位関係獲得の精度

順位	上位下位関係獲得の精度 (正しく上位語が獲得された下位語候補数/下位語候補の総数)		
	提案手法	ステップ 3 を抜いた場合	ステップ 4 を抜いた場合
1	86.667 (13/15)	100.000 (19/19)	86.667 (13/15)
5	84.000 (42/50)	60.920 (53/87)	63.235 (43/68)
10	87.619 (92/105)	60.417 (87/144)	66.346 (69/104)
15	84.459 (125/148)	47.465 (103/217)	62.658 (99/158)
20	86.893 (179/206)	58.993 (164/278)	64.486 (138/214)
25	84.946 (237/279)	53.295 (186/349)	65.060 (162/249)
30	86.478 (275/318)	59.286 (249/420)	63.696 (193/303)
35	86.592 (310/358)	60.451 (295/488)	60.452 (214/354)
40	84.635 (336/397)	58.409 (323/553)	56.716 (228/402)
45	83.370 (376/451)	59.675 (367/615)	59.300 (271/457)
50	84.631 (424/501)	60.714 (408/672)	56.863 (290/510)
55	82.486 (438/531)	62.003 (452/729)	54.930 (312/568)
60	79.682 (451/566)	60.756 (466/767)	52.969 (339/640)
65	78.000 (468/600)	57.762 (480/831)	50.291 (346/688)
70	77.959 (527/676)	58.998 (518/878)	50.889 (372/731)
75	75.403 (561/744)	56.223 (524/932)	50.392 (386/766)
80	74.239 (585/788)	57.782 (568/983)	49.756 (407/818)
85	73.236 (602/822)	60.247 (635/1054)	50.348 (434/862)
90	72.569 (627/864)	58.636 (645/1100)	49.551 (441/890)
95	71.507 (650/909)	57.950 (667/1151)	50.484 (469/929)
100	69.175 (662/957)	56.692 (682/1203)	49.645 (490/987)
110	66.857 (702/1050)	54.594 (713/1306)	49.908 (540/1082)
120	66.607 (750/1126)	56.089 (783/1396)	50.508 (596/1180)
130	64.603 (772/1195)	55.729 (822/1475)	49.121 (615/1252)
140	64.025 (808/1262)	54.557 (844/1547)	47.672 (645/1353)
150	62.182 (832/1338)	52.430 (863/1646)	47.493 (682/1436)
160	60.567 (854/1410)	49.942 (868/1738)	47.009 (715/1521)
170	58.950 (876/1486)	49.043 (897/1829)	47.081 (766/1627)
180	58.093 (908/1563)	49.347 (944/1913)	45.120 (772/1711)
190	55.963 (915/1635)	48.873 (976/1997)	43.459 (794/1827)
200	55.936 (947/1693)	48.227 (993/2059)	42.708 (820/1920)

表 4.5 ステップ 4 の各ルールを抜いた時の上位下位関係獲得の精度

順位	上位下位関係獲得の精度 (正しく上位語が獲得された下位語候補数/下位語候補の総数)			
	提案手法	ルール 1 を抜いた場合	ルール 2 を抜いた場合	ルール 3 を抜いた場合
1	86.667 (13/15)	86.667 (13/15)	86.667 (13/15)	86.667 (13/15)
5	84.000 (42/50)	73.077 (38/52)	83.333 (45/54)	84.000 (42/50)
10	87.619 (92/105)	71.154 (74/104)	78.125 (75/96)	87.619 (92/105)
15	84.459 (125/148)	77.215 (122/158)	74.510 (114/153)	72.973 (108/148)
20	86.893 (179/206)	78.325 (159/203)	75.543 (139/184)	71.845 (148/206)
25	84.946 (237/279)	78.788 (208/264)	69.721 (175/251)	68.817 (192/279)
30	86.478 (275/318)	74.121 (232/313)	69.565 (208/299)	66.981 (213/318)
35	86.592 (310/358)	78.272 (299/382)	70.029 (243/347)	69.274 (248/358)
40	84.635 (336/397)	75.862 (330/435)	72.881 (301/413)	67.003 (266/397)
45	83.370 (376/451)	76.840 (355/462)	67.579 (321/475)	67.849 (306/451)
50	84.631 (424/501)	72.727 (376/517)	68.750 (352/512)	70.659 (354/501)
55	82.486 (438/531)	73.274 (414/565)	68.852 (378/549)	69.303 (368/531)
60	79.682 (451/566)	73.115 (446/610)	69.322 (409/590)	67.314 (381/566)
65	78.000 (468/600)	74.433 (492/661)	68.123 (421/618)	66.333 (398/600)
70	77.959 (527/676)	73.826 (519/703)	69.091 (456/660)	65.828 (445/676)
75	75.403 (561/744)	72.207 (530/734)	66.383 (468/705)	61.962 (461/744)
80	74.239 (585/788)	70.519 (543/770)	66.887 (505/755)	61.548 (485/788)
85	73.236 (602/822)	70.250 (562/800)	66.131 (535/809)	61.071 (502/822)
90	72.569 (627/864)	70.370 (608/864)	66.040 (562/851)	60.995 (527/864)
95	71.507 (650/909)	69.836 (639/915)	65.297 (572/876)	59.846 (544/909)
100	69.175 (662/957)	68.111 (660/969)	63.834 (586/918)	57.576 (551/957)
110	66.857 (702/1050)	66.604 (710/1066)	61.455 (625/1017)	55.048 (578/1050)
120	66.607 (750/1126)	66.084 (756/1144)	59.729 (660/1105)	54.529 (614/1126)
130	64.603 (772/1195)	63.336 (786/1241)	59.439 (721/1213)	52.469 (627/1195)
140	64.025 (808/1262)	60.209 (808/1342)	56.357 (758/1345)	51.743 (653/1262)
150	62.182 (832/1338)	60.340 (852/1412)	54.603 (783/1434)	50.598 (677/1338)
160	60.567 (854/1410)	59.960 (894/1491)	54.303 (814/1499)	49.574 (699/1410)
170	58.950 (876/1486)	58.568 (916/1564)	53.571 (855/1596)	48.520 (721/1486)
180	58.093 (908/1563)	57.772 (944/1634)	52.117 (874/1677)	47.921 (749/1563)
190	55.963 (915/1635)	57.176 (976/1707)	50.339 (890/1768)	46.239 (756/1635)
200	55.936 (947/1693)	55.537 (998/1797)	49.485 (913/1845)	46.013 (779/1693)

表 4.6 リランキングを行った場合の上位下位関係の獲得精度

順位	上位下位関係獲得の精度 (正しく上位語が獲得された下位語候補数/下位語候補の総数)				
	提案手法 (上位 1)	上位 2	上位 3	上位 4	上位 5
1	86.667 (13/15)	86.667 (13/15)	86.667 (13/15)	86.667 (13/15)	86.667 (13/15)
5	84.000 (42/50)	84.000 (42/50)	61.111 (33/54)	68.750 (33/48)	68.750 (33/48)
10	87.619 (92/105)	87.850 (94/107)	76.147 (83/109)	68.750 (77/112)	68.750 (77/112)
15	84.459 (125/148)	89.130 (123/138)	81.208 (121/149)	74.839 (116/155)	74.839 (116/155)
20	86.893 (179/206)	87.940 (175/199)	79.602 (160/201)	79.293 (157/198)	79.293 (157/198)
25	84.946 (237/279)	89.394 (236/264)	78.884 (198/251)	74.409 (189/254)	74.409 (189/254)
30	86.478 (275/318)	79.193 (255/322)	82.278 (260/316)	78.275 (245/313)	78.275 (245/313)
35	86.592 (310/358)	79.231 (309/390)	77.926 (293/376)	74.595 (276/370)	74.595 (276/370)
40	84.635 (336/397)	80.238 (337/420)	78.153 (347/444)	75.000 (327/436)	75.000 (327/436)
45	83.370 (376/451)	81.250 (377/464)	77.453 (371/479)	76.987 (368/478)	76.987 (368/478)
50	84.631 (424/501)	82.456 (423/513)	73.962 (392/530)	73.258 (389/531)	76.091 (401/527)
55	82.486 (438/531)	82.601 (470/569)	74.211 (423/570)	72.154 (412/571)	73.739 (424/575)
60	79.682 (451/566)	81.699 (500/612)	74.959 (461/615)	73.096 (451/617)	73.139 (452/618)
65	78.000 (468/600)	80.581 (527/654)	75.758 (500/660)	73.123 (487/666)	74.926 (508/678)
70	77.959 (527/676)	78.084 (538/689)	76.487 (540/706)	73.980 (526/711)	74.048 (525/709)
75	75.403 (561/744)	73.943 (542/733)	75.370 (560/743)	74.769 (566/757)	74.369 (560/753)
80	74.239 (585/788)	73.953 (565/764)	75.160 (587/781)	73.804 (586/794)	74.629 (603/808)
85	73.236 (602/822)	72.149 (601/833)	72.323 (601/831)	72.934 (609/835)	73.571 (618/840)
90	72.569 (627/864)	69.239 (619/894)	72.260 (633/876)	70.383 (625/888)	72.222 (637/882)
95	71.507 (650/909)	68.163 (653/958)	72.707 (666/916)	70.096 (654/933)	69.840 (653/935)
100	69.175 (662/957)	66.436 (673/1013)	70.420 (688/977)	69.390 (671/967)	69.309 (682/984)
110	66.857 (702/1050)	64.760 (702/1084)	66.605 (722/1084)	67.317 (725/1077)	68.263 (727/1065)
120	66.607 (750/1126)	63.432 (732/1154)	62.395 (745/1194)	62.891 (744/1183)	66.151 (770/1164)
130	64.603 (772/1195)	61.722 (774/1254)	61.624 (774/1256)	60.893 (777/1276)	62.354 (800/1283)
140	64.025 (808/1262)	58.801 (785/1335)	60.922 (806/1323)	59.685 (795/1332)	60.748 (828/1363)
150	62.182 (832/1338)	56.742 (808/1424)	58.222 (825/1417)	57.876 (823/1422)	59.791 (858/1435)
160	60.567 (854/1410)	57.181 (860/1504)	57.905 (857/1480)	55.585 (831/1495)	57.794 (875/1514)
170	58.950 (876/1486)	55.880 (879/1573)	58.317 (915/1569)	55.570 (868/1562)	55.745 (883/1584)
180	58.093 (908/1563)	55.433 (908/1638)	58.221 (949/1630)	56.087 (926/1651)	55.718 (916/1644)
190	55.963 (915/1635)	54.370 (927/1705)	56.746 (980/1727)	56.042 (960/1713)	56.005 (970/1732)
200	55.936 (947/1693)	53.551 (950/1774)	57.485 (1033/1797)	54.751 (991/1810)	55.556 (995/1791)

第5章　他の手法との比較実験

評価実験により，本研究でたてた3つの仮説及び，それに基づく各ステップが単語間の上位下位関係の獲得に有効であることがわかった．本章では，提案手法が既存の方法と比べて良いかどうか判断するために，既存の手法との比較実験を行う．以下では，まず比較対象として今回挙げた4つの手法について説明し，その後それら4つの手法と提案手法との比較実験の結果について述べる．

5.1　比較実験に用いる手法

本研究では，下位語候補集合の接尾辞に注目する手法（手法1），下位語候補集合を獲得した箇条書きや表データのキャプションに注目する手法（手法2），従来手法のようにパターンを用いる手法（手法3）及びこれら3つの手法を組み合わせた手法（手法4）の全4種類の手法を比較対象とし，提案手法との比較実験を行った．各手法に関するより詳しい説明は以下のとおりである．

手法 1 複数の下位語候補の末尾で共有される語を上位語として獲得する方法を手法1とする．これは，日本語が主辞後続型言語（Head Final Language）であるため，下位語候補集合中の多くの要素の末尾に共通して現れるような語は，妥当な上位語である可能性が高いという考えに基づいている．手法1ではこの考えに基づき，まず複数の下位語候補の末尾に共通して現れる語を収集する．そして，収集された語の中で最も文字列長の長い語を上位語として獲得し，それが妥当な上位語であるかどうかを評価する．手法1と提案手法の精度にあまり差が見られない場合，提案手法により獲得された上位語のうち大部分が，3.3節で述べた統計的な手法を用いなくとも下位語候補の字面を見ることで簡単に獲得できる語であると考えられる．しかしながら，逆に提案手法と手法1の精度の間に有意義な差が見られる場合は，下位語候補集合中の各要素の末尾で共有されていない語を上位語として獲得できていると考えることができ，3.3節で述べた上位語候補獲得のための統計的な手法の必要性を実証することができると考えられる．

手法 2 下位語候補集合を獲得したHTML文書中の箇条書きや表データのキャプションから，上位語を獲得する方法を手法2とする．一般にHTML文書中に現れる箇条書きや表データのキャプションには，上位語が含まれやすいと考えられる．そこで手法2では，人手により下位語候補集合を獲得した箇条書きや表データの直上，も

表 5.1 比較実験に用いた正規表現パターン

先行研究で用いられている構文パターン	本研究で実装した正規表現パターン
名詞句 A「**名詞句** B」	**上位語**「**下位語**」
名詞句 A に似た**名詞句** B	**下位語** .* に似た .* **上位語**
名詞句 A と呼ばれる**名詞句** B	**下位語** .* と呼ばれる .* **上位語**
名詞句 A 以外の**名詞句** B	**下位語** .* 以外の .* **上位語**
名詞句 A のような**名詞句** B	**下位語** .* のような .* **上位語**
名詞句 A という**名詞句** B	**下位語** .* と (い | 言) う .* **上位語**
名詞句 A など**名詞句** B	**下位語** .* など (、| の)? .* **上位語**
名詞句 A などの**名詞句** B	
名詞句 A など、**名詞句** B	
——	**下位語** .* (ら | たち) .* **上位語**

各上位語と下位語は「」や “” で囲まれていても構わない.

しくはさらにその 1 つ上に存在するキャプションを抜き出し，その中に適切な上位語が含まれているかどうかを評価する．少なくとも現段階では HTML 文書中の箇条書きや表データのキャプションから自動的に上位語を獲得する手法がないため，この手法により得られた上位下位関係獲得の精度は，キャプションから上位語の獲得を試みる方法の上限だということに注意されたい.

手法 3 今角 [16]，安藤ら [14] の研究で用いられている構文パターンを基に作成した正規表現パターンにより上位語を獲得する方法を手法 3 とする．手法 3 で用いている正規表現パターンを表 5.1 に示す．手法 3 では，提案手法によりあらかじめ獲得された正しい上位下位関係を正規表現パターンに与え，そのパターンに適合する文が与えられた文書集合中に現れるかどうかを評価の対象としており，先に挙げた手法 1 及び手法 2 とはその評価基準が異なる．つまり手法 3 では，本研究で提案した手法により獲得された正しい上位語と下位語の組を，正規表現パターンを用いて獲得できるかどうかだけしか確認していない．また，手法 3 で用いているパターンは，構文解析結果ではなく文の表層的な情報だけを利用して上位下位関係の獲得を行っているため，先行研究で用いられている構文パターンとは若干異なる．この若干の差異が，何らかのエラーの原因となるかもしれない．しかし手法 3 で用いている正規表現パターンは，先行研究で用いられている構文パターンで獲得される上位下位関係を漏れなく獲得することができるため，今回の評価基準の場合は問題ないと思われる．そのため，正規表現パターンを用いることにより得られる上位下位関係獲得の精度が，構文パターンを用いて上位下位関係の獲得を試みる手法の最大値であると言える.

手法 3 では，正規表現パターンを適用する文書として，ステップ 2 で上位語候補を

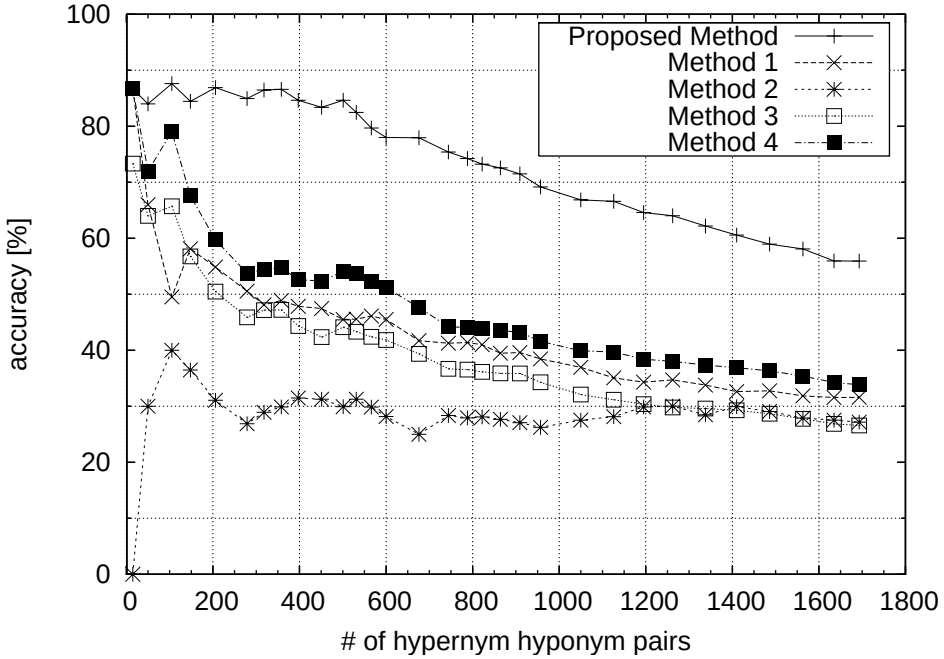

図 5.1 提案手法と他の手法の比較

獲得する際に用いた局所的文書集合中に含まれている HTML 文書からタグを除いたものを利用している．そのため，もし提案手法の方が手法 3 よりも良い結果が得られれば，少なくとも構文パターンにより獲得することができない上位下位関係を，提案手法は少量のテキスト（即ち，下位語候補 1 つあたり最大で 100 文書）からでも獲得することができるということになる．

手法 4 手法 1，2，3 を組み合わせたものを手法 4 とする．その評価は，本研究で提案した手法で獲得できた正しい上位下位関係のうちどのくらいの関係を手法 4 で獲得できるか，という観点で行った．この時，正しい上位下位関係が獲得できているかどうかの判定は，今回提案した手法により獲得された正しい上位語と下位語の組を，手法 1，2，3 のいずれかで獲得できていれば，正しい上位下位関係が獲得できたとした．比較実験により得られる提案手法と手法 4 の精度の差は，提案手法で獲得できて，手法 1，2，3 では獲得できない正しい上位下位関係数の差を表す．

5.2 実験結果

表 5.2 及び図 5.1 に提案手法と手法 1，2，3，4 の上位下位関係獲得精度を示す．図 5.1 では，各グラフとも提案手法と同じ方法で下位語候補集合をソートしており（つまり手法 1，2，3，4 とも，$sim(h(C), C) \cdot df(h(C), LD(C)) \cdot idf(h(C), G)$ の値を用いてソートしている），その上位語を獲得する方法だけが異なっている．この図より，今回比較対象とし

て挙げた 4 つの手法と比べ，本研究で提案した手法はより多くの上位下位関係を獲得できていることがわかる．このことから，1 つの下位語候補あたり最大で 100 件の文書を収集し作成した文書集合から上位下位関係を獲得する場合においては，手法 1，2，3，4 では獲得できないような関係を，提案手法はかなりの数獲得できているということがわかる．

また，下位語候補集合の要素の末尾に共有される語を上位語として獲得する手法 1 の結果を見ると，グラフが右下がりになっているのが確認できる．これより，末尾に共通の語を持つような下位語候補からなる下位語候補集合，つまり下位語候補間の意味的な類似性の高い下位語候補集合が，$sim(h(C), C) \cdot df(h(C), LD(C)) \cdot idf(h(C), G)$ を基準としたソートにより，比較的高い順位に集められていることがわかる．そのため，この結果からも $sim(h(C), C) \cdot df(h(C), LD(C)) \cdot idf(h(C), G)$ の値を用いて上位語候補と下位語候補集合の組のソートを行うステップ 3 が有効に働いていることが確認できる．

手法 1 と提案手法を比べると，両者の獲得精度の間に有意義な差が見られる．このことから，ステップ 2 で述べた統計的な尺度（$df(n, LD(C)) \cdot idf(n, G)$）を用いて，HTML 文書集合の中から上位語候補を発見することの有効性が示されたと考えられる．

次いで，下位語候補集合を獲得した HTML 文書中の箇条書きや表データのキャプションから上位語を獲得する方法である手法 2 のグラフを見ると，その獲得精度はおよそ 30 %程度であることが確認できる．HTML 文書中の箇条書きや表データから獲得した下位語候補集合の上位語を獲得することを考えた場合，安直に下位語候補集合を獲得した箇条書きや表データのキャプション中に含まれる語を対象にして上位語の獲得を行えば良いと考えるかもしれない．しかし，実際にそれらのキャプションから上位語の獲得を試みるとその精度は最高でも 30 %程度でしかなく，高い精度で正しい上位語を獲得するこが難しいということがこのグラフからわかる．さらに，今回の場合は計算機を用いて自動的に箇条書きや表データのキャプションから上位語を獲得しているわけではなく，人手によって獲得しているため，実際に計算機を用いて箇条書きや表データのキャプションから上位語の獲得を行った場合は，さらに精度が下がることが予想される．

また各手法と提案手法の上位下位関係獲得精度の差，特に手法 3 との差というのは，大量の文書を WWW より集め，それをコーパスとして用いることで縮まる可能性があると考えられる．しかし，大量の HTML 文書を用いて比較実験を行うことは，文書をダウンロードするのに多大な時間を要するため難しい問題であり，そのような条件下で比較実験を行うことは今後の課題である．

表 5.2 他の手法との比較実験結果

順位	上位下位関係獲得の精度 (正しく上位語が獲得された下位語候補数/下位語候補の総数)				
	提案手法	接尾辞	キャプション	パターン	組み合わせ
1	86.667 (13/15)	86.667 (13)	0.000 (0)	73.333 (11)	86.667 (13)
5	84.000 (42/50)	66.000 (33)	30.000 (15)	64.000 (32)	72.000 (36)
10	87.619 (92/105)	49.524 (52)	40.000 (42)	65.714 (69)	79.048 (83)
15	84.459 (125/148)	58.108 (86)	36.486 (54)	56.757 (84)	67.568 (100)
20	86.893 (179/206)	54.854 (113)	31.068 (64)	50.485 (104)	59.709 (123)
25	84.946 (237/279)	50.538 (141)	26.882 (75)	45.878 (128)	53.763 (150)
30	86.478 (275/318)	48.113 (153)	28.931 (92)	47.170 (150)	54.403 (173)
35	86.592 (310/358)	48.883 (175)	28.492 (102)	47.207 (169)	54.749 (196)
40	84.635 (336/397)	47.859 (190)	30.227 (120)	44.332 (176)	52.645 (209)
45	83.370 (376/451)	47.450 (214)	30.155 (136)	42.350 (191)	52.328 (236)
50	84.631 (424/501)	45.509 (228)	28.942 (145)	44.112 (221)	54.092 (271)
55	82.486 (438/531)	45.574 (242)	30.320 (161)	43.315 (230)	53.672 (285)
60	79.682 (451/566)	46.113 (261)	28.975 (164)	42.403 (240)	52.297 (296)
65	78.000 (468/600)	45.500 (273)	27.333 (164)	41.833 (251)	51.167 (307)
70	77.959 (527/676)	41.716 (282)	24.260 (164)	39.349 (266)	47.633 (322)
75	75.403 (561/744)	41.263 (307)	27.688 (206)	36.694 (273)	44.220 (329)
80	74.239 (585/788)	41.371 (326)	27.284 (215)	36.548 (288)	44.036 (347)
85	73.236 (602/822)	40.998 (337)	27.494 (226)	36.131 (297)	43.917 (361)
90	72.569 (627/864)	39.468 (341)	27.083 (234)	35.880 (310)	43.519 (376)
95	71.507 (650/909)	39.604 (360)	26.513 (241)	35.864 (326)	43.234 (393)
100	69.175 (662/957)	38.454 (368)	25.705 (246)	34.274 (328)	41.588 (398)
110	66.857 (702/1050)	36.952 (388)	27.048 (284)	32.095 (337)	39.905 (419)
120	66.607 (750/1126)	35.080 (395)	27.709 (312)	31.172 (351)	39.698 (447)
130	64.603 (772/1195)	34.310 (410)	29.456 (352)	30.377 (363)	38.410 (459)
140	64.025 (808/1262)	34.707 (438)	29.635 (374)	29.794 (376)	38.035 (480)
150	62.182 (832/1338)	33.782 (452)	28.102 (376)	29.596 (396)	37.369 (500)
160	60.567 (854/1410)	32.624 (460)	29.645 (418)	29.291 (413)	36.879 (520)
170	58.950 (876/1486)	32.773 (487)	28.802 (428)	28.668 (426)	36.339 (540)
180	58.093 (908/1563)	31.862 (498)	27.639 (432)	27.767 (434)	35.381 (553)
190	55.963 (915/1635)	31.560 (516)	27.339 (447)	26.850 (439)	34.251 (560)
200	55.936 (947/1693)	31.601 (535)	26.994 (457)	26.580 (450)	33.845 (573)

第6章　おわりに

6.1　まとめ

本稿では，構文パターンを用いずに WWW 上の HTML 文書から単語間の上位下位関係を獲得する手法を提案し，実験によりその有効性を示した．具体的には，HTML タグにより与えられる HTML 文書の構造，情報検索の分野などで用いられる *df*, *idf* などの統計量，新聞記事より獲得した名詞と動詞の係り受け関係，予備実験より得た知見に基づき作成したヒューリスティックなルールの 4 種類の情報を用いることで，利用可能な文書の量が少なすぎて既存の方法では獲得できないような上位下位関係であっても，今回提案した手法で同量の文書から獲得できることが実験により確認できた．そのため，少なくとも少量の文書から上位下位関係を獲得する際に，今回提案した手法が有効であることがわかった．

6.2　今後の課題

本稿で提案した手法により大量の上位下位関係の獲得を試みると，その精度は獲得する上位下位関係の数に比例して低下する．例えば，獲得された下位語候補集合の上位 1 割を最終的な出力とすると，その獲得精度は 60% 程度であり，十分高い精度で上位下位関係の獲得が行えているとは言えない．そのため，今後の課題としては，まず上位下位関係の獲得精度の向上が挙げられる．幸い，今回提案した手法は，構文パターンを用いて上位下位関係を獲得する既存の手法と組み合わせることが可能である．そのため，提案手法と既存の手法を組み合わせることで精度の向上が見込めるのではないかと考えている．具体的には，上位下位関係を表す構文パターン（例えば，**下位語**などの**上位語**）に，本稿で提案した手法により獲得された上位語と下位語を当てはめて文を生成し，その文が意味的に正しいかどうか判定する篩にかけることで，誤った上位語と下位語の組を削除することができ，それにより精度の向上が図れるのではないかと考えている．

また，現在の方法には，ユーザの所望する上位下位関係を獲得することができないという問題点がある．提案手法を用いてユーザが所望する上位下位関係を獲得するには，その下位語候補集合を含むような HTML 文書をユーザ自身が WWW 上から探してくる必要があるため，ユーザの望む上位下位関係を自動的に獲得することはできない．そのため，ユーザの望む上位下位関係を自動獲得できるように本提案手法を拡張することが今後の課題の 1 つであると考えている．

また3つ目の課題として，複数の語からなる上位語の獲得を行いたいと考えている．現在の提案手法では，DVD–RW，ハードディスク，プリンタなどのパソコン周辺機器からなる下位語候補集合に対しては「機器」としか上位語を求めることができないが，なんらかの手法により「周辺機器」,「パソコン周辺機器」という上位語が獲得できれば，他の自然言語アプリケーションにとって，より有用な情報となるのではないかと考えている．

謝辞

本研究を進めるにあたり，日頃から研究方針及び，研究内容について非常に熱心なご指導を賜りました鳥澤健太郎助教授にこの場を借りて厚く御礼申し上げたいと思います．

また，本研究にご理解と多大なご協力を賜りました，東条敏教授をはじめとする知識工学講座の皆様方に深く感謝致します．

参考文献

[1] Matthew Berland and Eugene Charniak. Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 57–64, 1999.

[2] Sharon A. Caraballo. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 120–126, 1999.

[3] Michael Fleischman, Eduard Hovy, and Abdessamad Echihabi. Offline strategies for online question answering: Answering questions before they are asked. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics*, pp. 1–7, 2003.

[4] Marti A. Hearst. Automatic acquistition of hyponyms from large text corpora. Technical Report S2K–92–09, 1992.

[5] Marti A. Hearst. Automated discovery of wordnet relations. In Christiane Fellbaum, editor, *WordNet: an electronic lexical database*, chapter 5, pp. 131–151. MIT Press, 1998.

[6] Nicholas Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, Vol. 118, No. 1–2, pp. 15–68, 2000.

[7] Rila Mandala, Takenobu Tokunaga, and Hozumi Tanaka. The use of wordnet in information retrieval. In *Proceedings of the COLING–ACL workshop on Usage of Wordnet in Natural Language Processing*, pp. 31–37, 1998.

[8] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. Introduction to wordnet: An on-line lexical database. In *Journal of Lexicography*, pp. 235–244, 1990.

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

[10] Hiroshi Sakamoto, Hiroki Arimura, and Setsuo Arikawa, editors. *Extracting Partial Structures from HTML Documents*. AAAI Press, 2001.

[11] G. Salton and M. E. Lesk. Computer evaluation of indexing and text processing. *Journal of the ACM*, Vol. 15, No. 1, pp. 8–36, January 1968.

[12] XML/SGML サロン. 標準 XML 完全解説. 技術評論社, 1998.

[13] Minoru Yoshida, Kentaro Torisawa, and Jun'ichi Tsujii. A method to integrate tables of the world wide web. In *Proceedings of the International Workshop on Web Document Analysis*, pp. 31–34, 2001.

[14] 安藤まや, 関根聡, 石崎俊. 定型表現を利用した新聞記事からの下位概念単語の自動抽出. 情報処理学会 研究報告 2003-NL-157, pp. 77–82, 2003.

[15] 村上義継, 坂本比呂志, 有村博紀, 有川節夫. HTML からのテキストの自動切りだしアルゴリズムと実装. 情報処理学会論文誌:数理モデル化と応用, Vol. 42, No. SIG 14(TOM 5), pp. 39–49, 2001.

[16] 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.

[17] 岡本潤, 石崎俊. 概念間距離の定式化と電子化辞書との比較. 自然言語処理, Vol. 8, No. 4, 2001.

[18] 黒橋禎夫, 長尾真. 日本語形態素解析システム JUMAN version 3.61 使用説明書. http://www.kc.t.u-tokyo.ac.jp/nl-resource/juman.html, 1999.

[19] 藤井敦, 石川徹也. World wide web を用いた事典知識情報の抽出と組織化. 電子情報通信学会論文誌, Vol. Vol.J85–D–II, No. 2, pp. 300–307, 2001.

[20] 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, No. 5, 2000.