論 文 Technical Papers

# 階層的知識間の調整規則の学習

## Learning of Alignment Rules between Concept Hierarchies

市瀬 龍太郎 ICHISE, Ryutaro
国立情報学研究所 知能システム研究系 知識処理研究部門
Knowledge Systems Research, Intelligent Systems Research Division, National Institute of Informatics
ichise@nii.ac.jp, http://research.nii.ac.jp/~ichise/

武田 英明 TAKEDA, Hideaki
（同 上）
takeda@nii.ac.jp, http://research.nii.ac.jp/~takeda/

本位田 真一 HONIDEN, Shinichi
（同 上）
honiden@nii.ac.jp, http://research.nii.ac.jp/~honiden/

**keywords:** machine learning, categorization, concept hierarchy, web mining

**Summary**

With the rapid advances of information technology, we are acquiring much information than ever before. As a result, we need tools for organizing this data. Concept hierarchies such as ontologies and information categorizations are powerful and convenient methods for accomplishing this goal, which have gained wide spread acceptance. Although each concept hierarchy is useful, it is difficult to employ multiple concept hierarchies at the same time because it is hard to align their conceptual structures. This paper proposes a rule learning method that inputs information from a source concept hierarchy and finds suitable location for them in a target hierarchy. The key idea is to find the most similar categories in each hierarchy, where similarity is measured by the $\kappa$(kappa) statistic that counts instances belonging to both categories. In order to evaluate our method, we conducted experiments using two internet directories: Yahoo! and LYCOS. We map information instances from the source directory into the target directory, and show that our learned rules agree with a human-generated assignment 76% of the time.

## 1. ま え が き

近年の情報ネットワークの発達により，個人が入手できる情報は，飛躍的に大きくなり，それとともに多くの情報をいかにして管理するかが重要になってきた．人間が情報を入手，または作成する時は，概念階層を使って情報を管理することが多い．例えば，プログラミングを効率化するために，「クラスライブラリー」という情報を作成する時には，継承などを効率的に行うために，概念階層が利用される．また，知識管理に使う「オントロジー」 [溝口 99] という情報も，それぞれの関係を表すために概念階層が利用される．その他にも図書館の「本」という情報の分類など，情報の管理に概念階層を用いる例は，枚挙にいとまがない．最近では，階層的に情報を記述することで，情報自体を概念階層を用いて管理するXML [Wor 01] も多くの場所で利用されはじめている．

情報を管理する時には，それぞれの情報利用者の目的，収集している情報などによって，情報管理方法への要求が異なる．そのため，同じような概念階層を利用して情報を管理しているにもかかわらず，それぞれの管理者，利用者などによって，別々の概念階層を用いて情報が管理されていることが多い．それは，概念階層に一貫性が必要な点や，分散管理の許容性の点などを考えると，現実的な方法ではあるが，情報の再利用という観点からは効率が悪い．一方で，情報を一箇所で集中的に管理するという手法もある．しかし，その場合には，全ての情報利用者の目的，得られる全ての情報などについて考慮しながら，概念階層の設計を行わなければならないため，一貫性の維持が非常に困難になるなどの問題が生ずる．

本論文では，それぞれの概念階層が持つ分類知識を相互に利用できるようにする手法を提案する．分類知識が分散して存在する環境を知識共生 [武田 01] の環境としてとらえ，共生している他の知識源から知識を取り込むことで，自分の持つ知識を拡張していく手法である．そのようなアプローチを取ることで，他の知識源が持つ情報に関する知識を自分の知識として取り込み，有効に利用できるようになる．また，知識共生環境では知識の分散管理が前提となるため，一貫性の管理が容易になる．しかし，このような環境下では，他者の知識は，異なる概念階層，語によって管理されているため，そのまま他者から情報を持って来たとしても，自分の持つ概念階層のどこに位置するべき情報なのかを同定し，利用することは難しい．そこで，本研究では，情報のインスタンスに基づいて他の知識との相違を調整する規則を学習する手

法を提案する．

次の第 2 章では，本研究で仮定する階層的知識源についての定義をおこなう．第 3 章では，2 つの知識源に対して，知識源の相違を調整するような規則を学習する手法について述べる．この規則を利用することで，他の知識源の持つ知識を自分の持つ知識源に取り込むことが可能となる．第 4 章では，提案する手法の有効性を確認するために，その手法に基づいて作られたシステム HICAL について述べ，インターネットディレクトリーを知識源として適用した実験について報告する．第 5 章では，本研究と関連研究の比較を行って，本研究の特徴を明らかにし，第 6 章で本研究をまとめる．

## 2. 階層的知識源

この章では，本研究で仮定する階層的知識源についてモデル化を行う．ここで対象とするのは，概念階層に基づいて分類が行われ，管理されている情報源である．例えば，クラスライブラリーやインターネットディレクトリー，図書の目録分類，オントロジーなどがそのようなものとしてあげられる．これらの情報源における概念階層は，最も一般的な概念を最上位として，順により詳細な分類を示す概念からなる階層構造を成している．個々の情報はこの概念階層の中のいずれかに割り当てられて管理される．なお，分野や目的によって最も下位の概念にしか個々の情報が割り当てられない場合と任意の概念に割り当て可能な場合があるが，本研究ではより一般的に概念階層を扱いたいので，任意の概念に割り当て可能であるとする．

本論文では分類に使われる概念階層は木構造であると仮定する．先に述べたクラスライブラリーやインターネットディレクトリー，目録分類など様々な分野で用いられている概念階層として木構造が多く用いられている．本研究では，このような概念階層を利用することを目的としているため，その基本構造である木構造を対象とすることにしている．すなわち，有向非循環のグラフを対象としている．なお，複数の上位ノードをもつものは単一の上位ノードを持つものを複数作る形で展開を行うことで木構造として扱う *1．木構造で示された概念階層を単純化し，グラフを用いて表すと， 図 1 のように表せる．この図では，概念階層が木構造で表され，インスタンスが木のノードに割り当てられる事になる．本論文では，以降，情報の実体をインスタンスと呼ぶ．黒点がある概念を表し，白点がインスタンスを表す事となる．ここで提供される知識は，概念階層の構造によって異なり，それぞれが分類を行う知識を表していると考えられる．本論

*1 木構造より一般的な構造としては束がある．束による表現については，形式概念分析 (Formal Concept Analysis) [Ganter 99] という方法があるが，木構造で表された概念階層を直接利用することはできない．

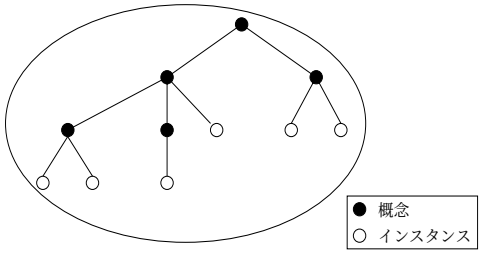


**図 1** 階層的知識源のモデル

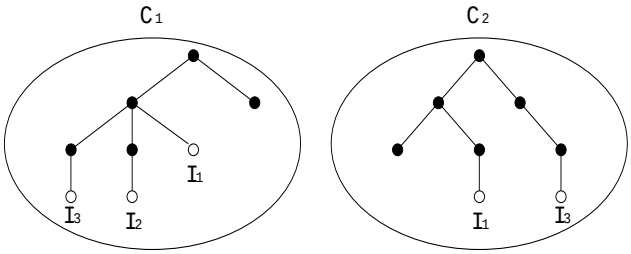


**図 2** 複数の階層的知識源における問題点

文では，情報を階層的に分類するものを知識と呼び，知識を提供できるものを階層的知識源と呼ぶことにする．

階層的知識源は，その中に含まれるインスタンスの種類，概念階層の構築者などによって，異なった階層構造を有する．したがって，このような知識源が複数存在する時に，他者の知識を利用するのは困難が伴う．例えば，図 2 では，2 つの知識源が存在している．概念階層 $C_1$ には，インスタンスとして $I_1, I_2, I_3$ の 3 つが存在しているが，概念階層 $C_2$ には $I_1, I_3$ の 2 つのみが存在しており，$I_2$ は含まれていない．この時，そのまま $I_2$ を $C_2$ に持って来ただけでは，$C_2$ 上のどこに位置するべき情報なのかが $C_2$ には分からない．したがって，そのまま $I_2$ を利用する事はできない．本研究では，これらの知識源における概念階層の対応を学習する事によって，他の知識源との相違を調整し，他の知識源が持つ知識を利用できるようにする手法を提案する．

## 3. 知識源の調整手法

この章では，知識源を調整する手法について述べる．ここで，調整とは，ある知識源が持つ知識を異なる知識源に取り込むために，両者の知識源の相違を吸収することを意味する．本論文で提案する手法では，まず，インスタンスを利用する事で，知識源の持つ分類基準の類似性を発見する．そして，類似したカテゴリー間でインスタンスを移動する規則を学習することで調整を実現する．手法の詳しい説明を行う前に，例を用いて簡単に概要の説明を行う．

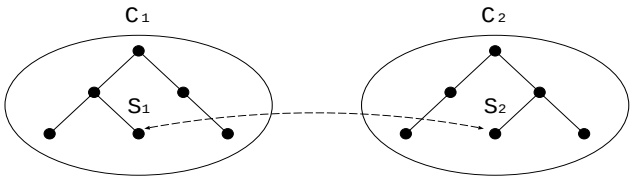


**図 3** 類似する分類の発見の例

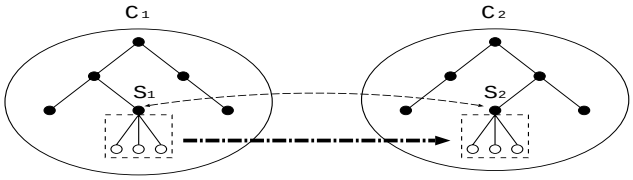


**図 4** 学習した規則を用いたインスタンスの移動の例

### 3·1 変換規則学習の例

本研究で提案する手法では，概念階層 $C_1$ に含まれるインスタンスを概念階層 $C_2$ に移動させるのに 3 つの段階をふむ．まずはじめに，インスタンスの分類に関する統計的な手法に基づき，類似している概念の発見を行う．図 3 は，概念階層 $C_1$ 中の $S_1$ と概念階層 $C_2$ 中の $S_2$ が類似概念として発見された状態を示す．図中の破線が類似概念の対応を表す．次に，インスタンスを移動させるための規則が発見された類似概念に基づいて構成される．この例でできる規則は,「$S_1$ に含まれるインスタンスは，類似概念 $S_2$ にも含まれる」という規則である．最後にここで作られた規則に従って，実際のインスタンスの移動が行われる．この例では， 図 4 の太破線で示されたようにインスタンスが移動し，同じインスタンスが $S_2$ に割り当てられる．これにより，$C_1$ に出現するインスタンスを $C_2$ 上のある概念に割り当てることができるようになる．

### 3·2 類似概念の発見

類似概念の発見は，2 つの知識源の最上位の概念からその下位の概念を順次調べていくことで行われる．最上位の概念同士を調べ，それが類似概念と見なされるならば，その下位の概念間でも，類似概念がある可能性があると考えて調べる．以降，類似概念と見なすことができる概念同士の組を類似概念ペアと呼ぶ．このアルゴリズムは 図 5 で表される．まず 2 つ概念階層 $C_1, C_2$ の最上位の概念 $N_{10}, N_{20}$ に対して，$\kappa$ 統計量を用いて類似概念ペアかどうかの判定を行う．$\kappa$ 統計量に関しては， 3·3 節で詳しく述べる．類似概念ペアと判定された場合には，その類似概念ペアを $R$ に記録するとともに，$make\_comb$ を使って新たな類似概念ペアの候補を作成する．下位の概念間が類似概念ペアになっている可能性を調べるためである．$make\_comb$ が概念 $N_1, N_2$ に対して作成する候補は，以下の 3 つによりできる全てのペアである．

```
Input:   N10,  // C1 の最上位概念
         N20,  // C2 の最上位概念
Output:  R;    // 類似概念ペアの集合
begin
  t := 1;
  R := φ;
  Xt := φ;
  X1 := [[N10, N20]];
  while Xt ≠ φ
    while Xt ≠ φ
      I := Xt の要素;
      N1, N2 := I 中の 2 つの概念;
      if N1, N2 が類似
        Xt+1 := Xt+1 + make_comb(N1, N2);
        R := R + I;
      fi;
      Xt := Xt − I;
    end;
    t := t + 1;
  end;
  return R;
end;
```

**図 5** 類似概念同定アルゴリズム

- $N_1$ と $N_2$ の子のペア
- $N_1$ の子と $N_2$ のペア
- $N_1$ の子と $N_2$ の子のペア

ここで作成した概念ペアは，次に類似性を調べる候補となる．これを，再帰的に計算し，類似性の検査を行っていく．生成される候補全てが検査され，類似概念ペアが生成されなくなった時に，システムは停止する．そして，類似概念ペアが蓄えられた $R$ を出力する．

### 3·3 $\kappa$ 統 計 量

知識源における概念階層は，インスタンスをある概念に従って分類したものである．ここで扱う概念階層は，木構造をしているため，ある概念ノードより下に属するインスタンスはその概念ノードに属していると判定できる．そのため，ある概念ノードを選択した時に，任意のインスタンスがその概念に適合するか否かを容易に判定する事ができる．すると，2 つの知識源における任意の 2 つの概念ノードに対して，共有インスタンスの分類を元に概念基準の類似性の判定を行う事ができるようになる．ここで，概念基準とは，ある概念に属するか否かの判断の基準のことを意味する．本研究では，この概念基準の類似性の判定に，$\kappa$ 統計量 [Fleiss 73] を用いた．$\kappa$ 統計量では，2 つの概念に対して， 表 1 のような分割表

**表 1** 2 つの概念によるインスタンスの分割表

| | | 概念 $N_2$ | |
|---|---|---|---|
| | | 含まれる | 含まれない |
| 概念 $N_1$ | 含まれる | $m_{11}$ | $m_{12}$ |
| | 含まれない | $m_{21}$ | $m_{22}$ |

を作成する．表 1 はある概念に含まれるインスタンスの数と含まれないインスタンスの数を一覧にしたものである．$N_1, N_2$ は，それぞれ概念階層 $C_1, C_2$ 中のある概念を表し，$m_{**}$ はそれぞれの分類に含まれるインスタンスの数を表している．ここで，2 つの概念基準が近ければ，表 1 中の $m_{11}$ と $m_{22}$ の数が多くなり，$m_{12}$ と $m_{21}$ の数が少なくなる．逆に概念基準が遠ければ，$m_{12}$ と $m_{21}$ の数が多くなり，$m_{11}$ と $m_{22}$ の数が少なくなる．$\kappa$ 統計量では，このことを利用して 2 つの概念基準が等しいか否かをある有意水準で判定を行う．

$\kappa$ 統計量では，まず，概念基準の一致率 $P$ と偶然の一致率 $P'$ を次の式により計算する．

$$P = \frac{m_{11} + m_{22}}{m_{11} + m_{12} + m_{21} + m_{22}}$$

$$P' = \frac{(m_{11} + m_{12})(m_{11} + m_{21})}{(m_{11} + m_{12} + m_{21} + m_{22})^2} + \frac{(m_{21} + m_{22})(m_{12} + m_{22})}{(m_{11} + m_{12} + m_{21} + m_{22})^2}$$

その時，$\kappa$ 統計量は，次式で表される．

$$\kappa = \frac{P - P'}{1 - P'}$$

次に，二つの概念基準の一致率が 0 である事を意味する $\kappa = 0$ であるかの検定を行う．そのために，次の値 $Z$ を計算する．

$$Z = \kappa \sqrt{\frac{(m_{11} + m_{12} + m_{21} + m_{22})(1 - P')}{P'}} \quad (1)$$

$Z$ は正規分布に従うため，有意水準を 5%とした時に，次の式を満たせば，概念基準の一致率が 0 であるとの帰無仮説が棄却される．

$$Z \geq 1.64486$$

仮説が棄却される時には，概念基準が一致していると判定できる．

### 3·4 規 則 の 生 成

図 5 のアルゴリズムにより生成された類似概念ペアの集合に対して，インスタンスがある概念に含まれているならば，それと対応する類似概念にも含まれるという形式の規則を生成し，学習結果として利用する．たとえば，$C_1$ 上の $N_1$ と $C_2$ 上の $N_2$ が類似しているということを 図 5 のアルゴリズムが出力した場合には,「$C_1$ の $N_1$ に属するインスタンス $I_i$ は，$C_2$ の $N_2$ に属する」という規則が生成される．

本論文では,「ある概念に属するインスタンスは，それと類似している概念にも含まれる」ということを仮定している．しかし，ここで類似性の判定に使われるインスタンスと実際に移動が行われるインスタンスは異なることに注意されたい．片方に含まれないインスタンスを移動させるのが目的となるため，そのインスタンスを用いても，類似性の判定は行えない．本論文では，類似性の判定に，共有インスタンスを用い，実際に移動するのは片方にしか含まれないインスタンスである．

図 5 のアルゴリズムでは，出力される類似概念ペアが常に 1 対 1 であることは保証していない．つまり，一つの概念に対して，複数の概念が対応する場合が考えられる．その場合には，式 (1) の値が高い類似概念ペアを選択する．検定を行う時に，式 (1) の値が高い程，現象が起こる確率が低いと言えるためである．

### 3·5 計 算 量

この節では， 図 5 のアルゴリズムの計算量について考察する．2 つの概念階層 $C_1, C_2$ があり，簡単のためそれぞれに含まれる概念ノードが $b$ 個の副概念に分かれていると仮定する．また，含まれる概念階層の深さを $d$ とする．このとき，$C_1, C_2$ に含まれる概念ノードの数 $N$ と $b, d$ は，次のような関係になる．

$$N = b^0 + b^1 + \ldots + b^d$$

全ての類似性を比較する場合には，$N \times N$ 個の比較を行わなければならない．したがって，計算量のオーダーは $O(b^{2d})$ となる．

次に，図 5 の計算量について考える．一回の類似性の検査で，$n$ の割合で，類似している概念と判定されることにする．$make\_comb$ で作成される組合せの数は，$b + b + b^2$ である．なぜならば，親のノードと別の概念階層の子のノードの組合せが $b$ 個でき，それが，$C_1, C_2$ の組合せ方によって 2 通りできる．さらに，子のノード同士の組合せが $b^2$ 個できるためである．$n$ の割合で，類似していると判定されるため，検査によって生成される組合せの個数の期待値は，$n \times (b^2 + 2b) + (1 - n) \times 0$ となる．$(1 - n) \times 0$ という式は，類似概念ペアと判定されない時に，類似概念ペアの候補が，生成されないからである．したがって，調べたノードの数に，期待値を掛けたものを階層の最上位から足し合わせていくと，調べなければならないノードの組合せの数 $M$ が計算できる．$M$ は，次の式で表される．

$$M = b^0 + n^1(b^2 + 2b)^1 + n^2(b^2 + 2b)^2 + \ldots + n^d(b^2 + 2b)^d$$

したがって， 図 5 のアルゴリズムの計算量のオーダーは $O(n^d \times b^{2d}) = O((n \times b^2)^d)$ となる．$n$ は，類似していると判定される割合なので，$0 \leq n \leq 1$ を満たす．し

たがって，図 5 のアルゴリズムを使用すると，全ての組合せを計算する場合と比較して，計算量が減少すると言える．すなわち，深さのべき乗のオーダーであることには変わりはないが，幅方向が制限された探索に相当することになる．

## 4. インターネットディレクトリーを用いた実験

この章では，前章で述べた手法の妥当性を評価するために，SICStus Prolog を使って実装されたシステム HICAL を用いた実験について報告を行う．

### 4·1 実 験 設 定

実験を行う対象として，インターネットディレクトリーの Yahoo! Japan [Yah 00] と LYCOS Japan [Lyc 00] の分類体系を知識源の概念階層として用い，そこに含まれる外部リンク (URL) をインスタンスとして用いた．実験に使うデータは 2000 年の 8 月から 9 月にかけて収集を行った．Yahoo!の概念階層には，約 41,000 個の概念があり，約 224,000 個の URL が登録されている．一方，LYCOS の概念階層には，約 5,700 の概念があり，約 48,000 個の URL が登録されている．登録 URL 数を見ると，Yahoo! の方が圧倒的に多く，知識源としての LYCOS は冗長に見えるが，LYCOS に収録されているインスタンスの半数の約 25,000 個しか，Yahoo!と共有されていない．これは，量的に大きい知識源が 1 つあったとしても，中に含まれる知識には偏りがあるため，他者の知識が必要になるということを表している例であると言える．

Yahoo!と LYCOS からは以下の 3 つの概念とその副概念を含む概念階層を選択し，実験を行った．

- Yahoo! : Arts / Humanities / Literature
  LYCOS : 芸術と人文科学 / 文学
- Yahoo! : Business_and_Economy / Companies
  LYCOS : 経済・産業 / 企業
- Yahoo! : Recreation
  LYCOS : 趣味・スポーツ

これらの実験データの概念数，インスタンス数，共有インスタンス数は 表 2 の通りである．

### 4·2 実 験 手 順

実験は，次の手順で行った．まず，Ruby で作成したツールを用いて，Yahoo!，LYCOS からデータを収集した．その後，Web 文書の URL を抽出し，ランダムで等数になるように 10 分割した．10 分割したうちの 9 つは，規則の学習に使う訓練例とし，残りの 1 つは，分類がどの程度正確に行われるかを調べるためのテスト例とした．実験データを 10 分割することで，訓練例とテスト例の組合せは，10 個作成できる．この 10 個のデータを用いて，実験を 10 回行い，その平均値を実験結果とした．このように実験することで，実際に使う際の正答率を調べることができるからである．

実験データの作成後，共有する訓練例，その訓練例と概念の関係，各概念間の関係を SICStus Prolog で作成した HICAL システムに入力し，規則の学習を行った．規則の学習にかかった時間は，後述の実験 1 の設定で，PentiumIII-733MHz のマシンを使うと，文学の領域で約 1 時間 50 分，企業の領域で約 9 時間，趣味・スポーツの領域で約 7 時間であった．学習の際の $\kappa$ 統計量の有意水準は 5%とした．その後，学習された規則の妥当性を調べるために，Ruby で作成した評価器でテスト例の評価を行い，正答率を計算した．評価器で使われる正答の評価方法として，いくつかの手法が考えられる．この実験では，2 つの評価手法を提案して，それぞれの評価器を用いて実験を行った．その手法は以下の 2 つである．

(1) インスタンスがテスト例と同じ概念に分類された時に正答とする手法.
(2) インスタンスがテスト例と同じ概念に分類されるか，その 2 つ上まで[*2]の概念に分類された時に正答とする手法.

テスト例は，両者が共有する URL から作られるため，予めどの概念に所属するべきかを知ることができる．その概念に URL が割り当てられた時に正答とするのが評価法 1 である．評価法 2 は評価法 1 の場合に加えて，所属するべき概念の 2 つ上の概念までに分類された場合にも正答としたものである．評価法 1 は，厳密な評価方法であるが，情報の移動元となる概念階層が十分な量の概念を概念階層の中間に含んでいなければならないという制約がある．なぜならば，詳細な分類を行っていないものから，詳細な分類を行っている階層に情報を移動させることになると，そもそも元の知識源にある分類知識以上の知識が必要になってしまうからである．一方の評価法 2 は，情報源となる概念階層と情報の移動先となる概念階層のどちらが詳細な階層を持っているかということに依存しないので，階層構造の調整の評価方法として現実的な方法であると言える．

### 4·3 実 験 1

実験 1 として，学習された規則の妥当性を調べるために，インスタンスが属している概念の規則のみを使って評価を行った．この場合には類似概念が学習された概念に属するインスタンスだけがテストされるため，本研究で提案する「ある概念に属するものは，それと類似している概念にも属する」という仮説の検証を行うのに適していると考えられる．実験結果は，図 6 となった．左の図が Yahoo!から LYCOS への移動を行う規則の結果であり，右の図が LYCOS から Yahoo!へ移動を行う規則

*2 2 階層としたのは実験的な結果による．移動可能階層を 0,1,2 と変更するとそれぞれ向上がみられたが，3 とすると 2 からほぼ変化がなかったため，上限を 2 とした．

**表 2** 実験に利用したデータの数

| | Yahoo! | | LYCOS | | 共有 |
|---|---|---|---|---|---|
| | 概念数 | インスタンス数 | 概念数 | インスタンス数 | インスタンス数 |
| 文学 | 493 | 3192 | 186 | 1119 | 468 |
| 企業 | 7554 | 58609 | 413 | 5904 | 3992 |
| 趣味・スポーツ | 3164 | 19609 | 709 | 4941 | 1939 |

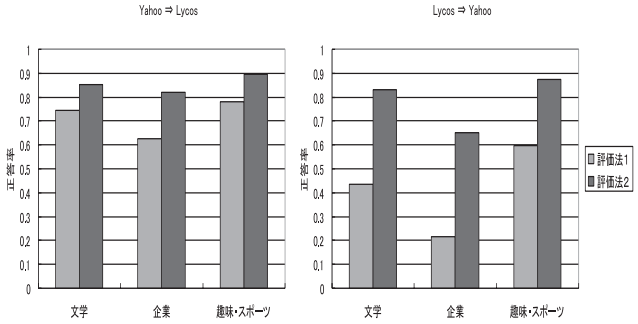


**図 6** 実験 1 の結果

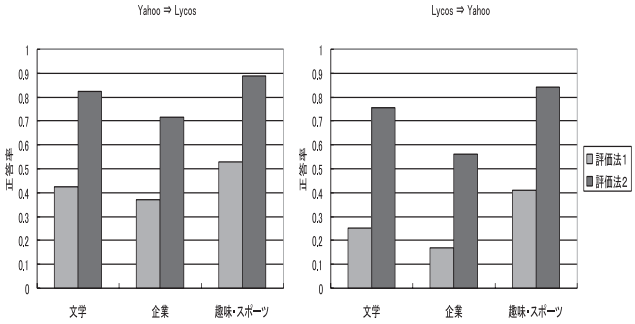


**図 7** 実験 2 の結果

の結果である．

図 6 から分かるように，文学と趣味・スポーツの実験では 8 割を越える正答率を出している．また，企業の実験においても Yahoo!から LYCOS への実験は 8 割以上の正答率を出している．このことから,「ある概念に属するインスタンスは，類似する概念にも属する」という本研究における仮定は高い精度で機能していると言える．

企業の領域において,「LYCOS から Yahoo!」への規則の正答率は 7 割以下と他の場合に比べて低い．これは，概念の数が大幅に違うことが影響していると考えられる．概念の数の違いは，詳細な分類階層と簡単な分類階層という形で，分類階層に影響を与える．結果として,「LYCOS から Yahoo!」への規則は，簡単な分類体系から複雑な分類体系への規則を学習していることになる．そのような場合には，少ない分類知識から，多くの概念への分類規則を生成しなければならない．したがって，問題が難しくなるため，必然的に正答率がさがるのだと考えられる．

### 4·4 実　　験　　2

実験 1 では，インスタンスが属する概念を利用した規則のみを使った．その実験では，類似概念が発見されなかった場合には，規則が学習されないため，インスタンスを分類できないという問題が生じる．そこで，インスタンスが属する概念に対して，規則が学習されなかった場合には，その概念より上で，最も下位の概念に割り当てられた規則を替わりに利用する手法をここでは使う．そうすると，他の知識源が持つ全てのインスタンスに対して，分類を行うことができるようになる．つまり，より実際に利用する場面に即した評価を行うことができるようになると言える．この実験では，実験 1 の時とは異なり，全てのインスタンスを利用するため，分類を行うデータの数は増える．そのため，正答率を計算する際の分母が実験 1 よりも大きくなる点に注意されたい．

この結果を示したものが，図 7 である．実験 1 と比べて分類を行うデータ数が多くなるため，単純な比較はできないが，全体的に多少正答率が低くなっているものの，文学と趣味・スポーツの領域の正答率が高く，それに比べて企業の領域は低くなっている傾向には変わりがない．この実験では，文学と趣味・スポーツの領域で，8 割程度の正答率，企業の領域て 6 割程度の正答率を得ることができた．この結果は，後述する類似研究 [Agrawal 01] の手法と比べて，高い精度でインスタンスの移動を行えることを示している．

### 4·5 実　　験　　3

次に規則の選択手法の違いによる生成規則の性能の違いを調べるために，規則の選択方法を変えて，実験 2 と同様の実験を行った． 3·4 節で述べたように，本研究で提案したアルゴリズムは類似概念が常に 1 対 1 であることを保証していない．その時には，式 (1) の値が高いものを採用した．しかし，選択する際に，より特殊な類似概念ペアを選択する手法もある．ここでの，特殊な類似概念ペアとは，ある概念 $S_1$ に対して，概念 $S_2$ と $S_3$ の 2 つのペアが発見された場合に，$S_2$ と $S_3$ を比較し，概念階層上で下位になる概念とのペアのことを意味する．$\kappa$ 統計量を利用する際には，特殊な概念ペアほどサンプルとなる数が少くなるため，信頼性が低いと判定されやすくなる．ところが階層的知識においては，より特殊な類似概念ペアの方が，一般的な類似概念ペアよりも有効であることも考えられる．それを調べるために，規則の選択方法のみ，実験 2 と条件を変えて，実験を行った．

実験結果は， 図 8 のようになった．実験 2 と同様に，

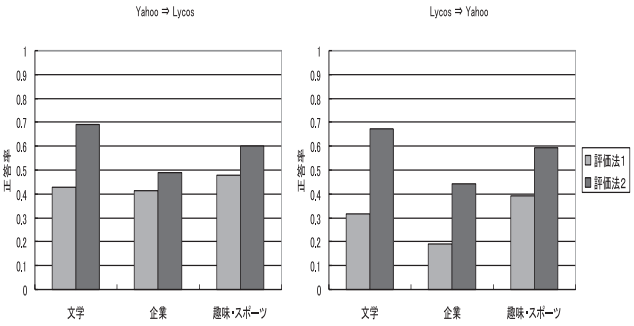


**図 8** 実験 3 の結果

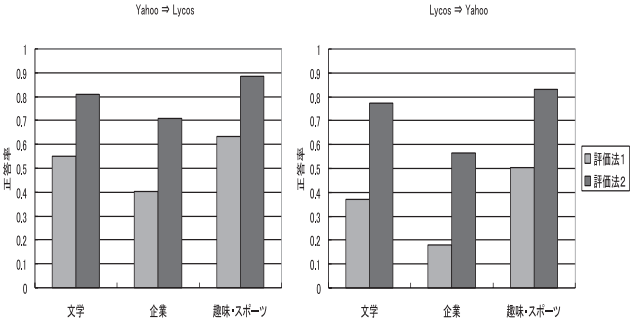


**図 9** 実験 4 の結果

企業の領域での正答率が，他と比べて低い傾向は変わらない．評価法 1 での正答率に多少の向上も見られるが，評価法 2 では逆に正答率が大きく下がる．この手法は，学習された概念に対しての正答率を向上させるが，親のカテゴリーの規則を使った場合には，かえって正答率を低下させてしまう難点があるといえる．

### 4·6 実　　験　　4

次に，実験 4 として，$\kappa$ 統計量の有意水準による違いを確認するため，有意水準を 10%として実験した．その他の設定は，実験 2 と同じである．その結果は，図 9 のようになった．

実験 2 の結果と比較すると，評価法 1 での正答率は向上するが，評価法 2 の正答率は変化がないと言える．有意水準の基準を低く設定したために，学習する規則の数は実験 2 の時よりも増えることになるが，そのために，インスタンスが属する概念の規則の替わりに親の概念の規則を使ってインスタンスを移動させることが少なくなり，結果として評価法 1 の正答率が向上したと考えられる．

### 4·7 学習規則の質について

定性的な評価を行うために，文学の領域で学習された規則について解析を行った．文学の領域においては，10 回の実験の結果，平均で 137 個の規則を学習している．その中には,「Yahoo! : Arts / Humanities / Literature / Genres / Literary_Fiction / Authors / Murakami_Haruki」と「LYCOS : 芸術と人文科学 / 文学 / 小説 / 村上春樹」のように，概念のラベル名の対応を取るだけで，発見できるような規則もあったが,「Yahoo! : Arts / Humanities / Literature / Poetry / Waka / Kajin / Masters / Murasakisikibu」と「LYCOS : 芸術と人文科学 / 文学 / 日本文学 / 上代・中世文学 / 源氏物語」のように，概念のラベル名の対応だけでは関係を学習できないものもあった．

一般的に使われる同義語辞書は同じものに対しての識別効果は高い．しかし，分類ラベルとして使われる場合には，同じ分類基準を持っているにもかかわらず，この例のように必ずしも同義語が使われるとは限らない．本研究で提案している手法は，同義語辞書を利用せずに形式のみを利用しているため，辞書に依存するような問題を起こさずに，このような関係を抽出できる．このような手法は，語によらない概念同士の関係を発見できるため，知識発見の手法にも応用できるのではないかと考えられる．

次に，規則の質的な妥当性を調べるために，テスト例に対して分類の間違いを起こした規則の調査を行った．あるランダムで選んだデータセットに対する結果を取り出し，Yahoo!から LYCOS への規則を対象として調査を行った．その結果，間違えと判定された分類の規則でも内容上の大きな間違えは多くないということが分かった．たとえば,「大阪府立国際児童文学館」の Web ページは，テスト例では「LYCOS : 芸術と人文科学 / 文学 / 児童文学 / 児童文学館」に分類されていたが，HICAL を使うと「LYCOS : 芸術と人文科学 / 文学 / 記念碑・記念館」に分類されてしまう．これは正答率を計算する際に分類間違えとして計算されることになるが，内容的には完全な分類間違えとは言いがたい．Yahoo!では，児童文学館に類似する分類を持っていないため,「大阪府立国際児童文学館」を「Yahoo! : Arts / Humanities / Literature / Literary_Libraries」として「図書館・文学館」とまとめて分類している．そのため，LYCOS のみが持つ「児童文学館」というカテゴリーへの学習が行われず，類似している「LYCOS : 芸術と人文科学 / 文学 / 記念碑・記念館」への規則が学習されることになる．このように見ていくと，間違えた 8 つの規則の内，7 つは妥当な答えを返していた．完全な間違えと言える 1 つは,「LYCOS : 芸術と人文科学 / 文学 / ミステリー」に属する作家のページを「LYCOS : 芸術と人文科学 / 文学 / SF・ホラー・ファンタジー」に誤分類していた.「Yahoo! : Arts / Humanities / Literature / Genres / Young_Adult / Authors」に属するものは「LYCOS : 芸術と人文科学 / 文学 / SF・ホラー・ファンタジー」に属するという一般的な規則が学習されたためである．

### 4·8 今後の拡張点

現状のシステムでは，まったく独立な 2 つの分類基準が概念階層の上下で用いられた場合に，うまく動作しないと考えられる．ここで言う独立とは，2 つの基準のどちらが上位の概念に来ても分類が可能なことである．例えば,「料理」に関連した概念階層を考えてみる．その時に，概念階層 $C_1$ では，まず上位の分類として「食材」で分類を行い，その下層で「料理の国名」によって分類するとする．しかし，もう一つの概念階層 $C_2$ では，上位の分類として「料理の国名」を使い，下位の分類で「食材」を使っているとする．このような分類は,「料理の国名」と「食材」がお互いに影響を与えないため，その概念階層の利用目的，利用者，コンテンツなどによって，どちらの概念体系も利用されることがありうる．このような場合には，本研究で提案する手法は，概念の最上位から調べていく手法を取っているため，対処ができなくなると考えられる．しかし，概念の下位から調べていく手法と組み合わせることによって，このような場合にも対処できるようになると考えられるため，両者の組合せについて今後は考えて行く必要性がある．

現状のシステムにおいては，知識源として 2 つの概念階層を考えている．しかし，実際の状況においては，知識源として利用できるものは多数ある．本手法を利用した場合には，2 つずつ統合していくことで多数の知識源に対応することも可能であるが，同じ情報が多数の知識源から獲得できるような状況においては，どの知識源を利用すれば，精度の高い知識を得られるかが異なるはずである．したがって，知識源が多数になった時に，どのように他の知識源を利用するかを考えていく必要性がある．

## 5. 関 連 研 究

本研究で定義した問題に対して，従来の機械学習 [Mitchell 97] のアプローチと本研究のアプローチは，学習に使う情報の面で大きく異なる．たとえば，[Koller 97, Wang 99] では，それぞれのインスタンスに付けられた属性を用いて，分類規則の学習を行っているのに対し，本研究では，他の知識源が持つ知識を利用することで，学習を行っている．このようなアプローチを取ることで，既にある信頼性が高い知識を利用できる点が異なっていると言える．このような情報を利用するアプローチは，Agrawal らの研究 [Agrawal 01] でも見られ，属性のみを利用した場合よりも効果的であることが確認されている．

Agrawal らの研究 [Agrawal 01] では，他の知識源が持つ分類知識の利用により，従来の機械学習の手法よりも精度の高い分類知識の導出に成功している．この研究では，属性だけを使う Naive Bayes 法 [Mitchell 97] を拡張することで，既にある分類の知識も規則の学習に使えるようにしている．本研究と比較すると，他の知識源を利用する点では類似しているが，手法自体が異なる．この研究では階層的な分類に関しては取り扱っていないが，本研究では階層的分類を取り扱うことができる．また，本研究の手法では，文書の中に含まれる語を利用していないが，この研究では Naive Bayes 法に基づいているため，文書に含まれる語を利用しなければならない．実験条件が全く同じでないため直接的な比較はできないが，この論文に掲載されている Google と Yahoo!を用いた同様の実験では，実験した全ての領域の平均で，62%程度の正答率となっている．しかし，本研究では，この実験の設定に近い実験 2 において，全ての領域の正答率の平均を取ると，76%程度の正答率を出している．

データ統合の手法 [Doan 00] と比較すると，データ統合の手法では，全てのデータを一つの体系に統合して取り扱おうとしているのに対し，本手法が分散する知識を分散したまま取り扱うことに主点をおいている点が異なる．前者の立場では，与えられたデータを統合可能な汎用の概念階層が与えられる．そのため，必然的に統合する階層と統合される階層の 2 者は構成が類似することになる．しかし，本研究の立場では，分散する知識を分散したまま利用することに主眼をおいている．そのため，階層構造が全く異なるような場合にも対処しなければならない．提案手法では，2 つの階層構造が異なる場合には，親の階層に遡る事で対応関係を学習することができる．

オントロジーの併合 (merging)/調整 (alignment) を行うシステムの Chimaera [McGuinness 00] や PROMPT [Noy 00] と比較すると，これらのシステムは，併合/調整の際にユーザの介在が欠かせない点で異なっている．また，これらのシステムは，類似した概念を発見するのに語の類似性を利用しているため，4·7 節で述べたような，語とは離れた類似概念を発見することができない．さらに，発見できるものは使用する辞書などの影響を受けやすい．一方，HICAL では，形式的な情報のみを利用するため，このような影響はないという利点がある．

ブックマーク共有システムの Siteseer [Rucker 97] や Blink [Bli 00] と比較すると，HICAL が異なる知識源の知識を利用している点で，非常に似ている．しかし，これらのシステムとは，HICAL が概念階層を利用しているという点で大きく異なる．Siteseer や Blink は与えられた概念に含まれるインスタンスの数のみしか利用していないが，HICAL は階層構造を利用することで，より有効に他者の知識を利用している．階層構造を用いた場合には，あるインスタンスにちょうど適する概念が存在しない場合に，親の概念を利用することができる．kMedia [Takeda 00] は階層構造を用いたブックマーク共有システムであるが，PROMPT などと同様に語の影響を受けてしまう．

## 6. む す び

本論文では，さまざまな情報を管理する概念階層を一つの知識源としてみなし，その知識源のモデル化を行っ

た．そして，それぞれの知識源が持つ知識を相互に利用できるようにする手法について述べた，その手法は，インスタンスの分類の類似性に基づいて，各概念間の類似性を同定し，他の知識源との相違を調整する規則として学習する機械学習の手法である．その手法に対する有効性を評価するために，システム HICAL を構築し，インターネットディレクトリーの Yahoo!と LYCOS を用いて実験を行った．実験によって，提案手法を用いると他の知識源から知識を適切な場所に取り込めることが分かり，知識源が相互に利用できるようになることが示された．

今後は，4･8 節で述べたような拡張を実施していくと同時に，XML ドキュメントの統合に適用したり，オントロジーの統合に適用をすることで，新たな問題点の発見や応用を考えていきたい．また，4･7 節で示したように，この手法を用いると，異なる知識同士のインターラクションによって，新たな知識がつくり出されることが分かった．このような知識を利用できるような知識発見システムについても考えて行きたい．

## ◇ 参 考 文 献 ◇


[Agrawal 01] Agrawal, R. and Srikant, R.: On Integrating Catalogs, in *Proceedings of the Tenth International World Wide Web Conference (WWW-10)*, pp. 603–612 (2001).

[Bli 00] Blink.com, http://www.blink.com/ (2000).

[Doan 00] Doan, A., Domingos, P., and Levy, A.: Learning Source Descriptions for Data Integration, in *Proceedings of the International Workshop on The Web and Databases (WebDB)*, pp. 81–86 (2000).

[Fleiss 73] Fleiss, J. L.: *Statistical Methods for Rates and Proportions*, John Wiley & Sons (1973), 佐久間 昭訳，邦題：「係数データの統計学」，東京大学出版会，1975.

[Ganter 99] Ganter, B. and Wille, R.: *Formal Concept Analysis - Mathematical Foundations*, Springer (1999).

[Koller 97] Koller, D. and Sahami, M.: Hierarchically classifying documents using very few words, in Fisher, D. H. ed., *Proceedings of the 14th International Conference on Machine Learning*, pp. 170–178, Nashville, US (1997), Morgan Kaufmann Publishers, San Francisco, US.

[Lyc 00] LYCOS Japan, http://www.lycos.co.jp/ (2000).

[McGuinness 00] McGuinness, D. L., Fikes, R., Rice, J., and Wilder, S.: An Environment for Merging and Testing Large Ontologies., in Cohn, A. G., Giunchiglia, F., and Selman, B. eds., *Proceedings of the Conference on Principiles of Knowledge Representation and Reasoning (KR-00)*, pp. 483–493, S.F. (2000), Morgan Kaufman Publishers.

[Mitchell 97] Mitchell, T. M.: *Machine Learning*, McGraw Hill (1997).

[溝口 99] 溝口理一郎：オントロジー研究の基礎と応用，人工知能学会誌，Vol. 14, No. 6, pp. 977–988 (1999).

[Noy 00] Noy, N. F. and Musen, M. A.: PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment, in *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000)*, pp. 450–455, Menlo Park (2000), AAAI Press.

[Rucker 97] Rucker, J. and Polanco, M. J.: Siteseer: Personalized Navigation for the Web, *Communications of the ACM*, Vol. 40, No. 3, pp. 73–75 (1997).

[Takeda 00] Takeda, H., Matsuzuka, T., and Taniguchi, Y.: Discovery of Shared Topics Networks among People – A Simple Approach to Find Community Knowledge from WWW Bookmarks, in Mizoguchi, R. and Slaney, J. eds., *Proceedings of the 7th Pacific Rim International Conference on Topics in Artificial Intelligence (PRICAI-2000)*, Vol. 1886 of *LNAI*, pp. 668–678, Berlin (2000), Springer.

[武田 01] 武田, 市瀬, 村田, 本位田：知識共生プロジェクトーネットワーク情報の自律的生態系を目指してー, 情処研報, Vol. 2001, No. 41, pp. 25–33 (2001).

[Wang 99] Wang, K., Zhou, S., and Liew, S. C.: Building Hierarchical Classifiers Using Class Proximity, in Atkinson, M., Orlowska, M. E., Valduriez, P., Zdonik, S., and Brodie, M. eds., *Proceedings of the 25th international Conference on Very Large Data Bases*, pp. 363–374, Los Altos, CA 94022, USA (1999), Morgan Kaufmann Publishers.

[Wor 01] Extensible Markup Language, http://www.w3c.org/XML/ (2001).

[Yah 00] Yahoo! Japan, http://www.yahoo.co.jp/ (2000).


〔担当委員：河野浩之〕

2001 年 6 月 25 日 受理

## 著 者 紹 介

**市瀬 龍太郎**（正会員）

1995 年東京工業大学工学部情報工学科卒業．1997 年同大学大学院情報理工学研究科計算工学専攻修士課程修了．2000 年同大学院博士課程修了．博士 (工学)．2000 年より国立情報学研究所知能システム研究系助手．2001 年よりスタンフォード大学言語情報研究所客員研究員．人工知能，機械学習などの研究に従事．電子情報通信学会，情報処理学会，日本認知科学会，各会員．

**武田 英明**（正会員）

1991 年 3 月東京大学 大学院工学系研究科博士課程修了．1993 年 4 月奈良先端科学技術大学院大学助手．1995 年 4 月同助教授．2000 年 4 月国立情報学研究所助教授．現在に至る．人工知能特に知識共有，ネットワークコミュニティ，実世界エージェントなどの研究に従事．AAAI，電子情報通信学会，情報処理学会など各会員．

**本位田 真一**（正会員）

1978 年早稲田大学大学院理工学研究科電気工学専攻修士課程修了．(株) 東芝を経て 2000 年文部科学省国立情報学研究所教授，現在に至る．2001 年から東京大学大学院情報理工学系研究科教授を併任．エージェント技術，オブジェクト指向技術，ソフトウェア工学の研究に従事．IEEE，日本ソフトウェア科学会，情報処理学会など各会員．工学博士．