自 然 言 語 処 理 78－6
（1990．7．19）

記述量圧縮の観点から見た概念体系の構築

松川智義、岸本行生、三池誠司、横田英司、高井貞治、天野真家
（株）日本電子化辞書研究所

概念分類項目間の関係記述データ（項目間記述データ）を、その内容を保存したまま階層化して概念体系を構築する方法を提案する。この方法に基づき概念体系を試作した。その結果、記述量が８９%圧縮されること、６７%のノードに意味的共通性があることを確認した。これにより、本概念体系が、ａ．カテゴリ間記述データの検証作業の効率化、ｂ．ＥＤＲ概念体系における上位概念設定のための予備実験、に用いることができることがわかった。

Construction of Hierarchical Concept Classification Based On Compaction of Concept Description

Tomoyoshi MATSUKAWA, Yukio KISHIMOTO, Seiji MIIKE,
Eiji YOKOTA, Sadaharu TAKAI, Shin-ya AMANO

Japan Electronic Dictionary Research Institute, Ltd.

The method for construction of concept classification by hierarchizing the description data of the relations among concept categories is proposed. The hierarchical concept classification was built by using this method. It was confirmed that the amount of description was compacted at 89.2 %, and that the 67 % nodes in the classification had semantic consistency. With the results, we confirmed that the method can be used for the followings: a. improving efficiency of the verification of the description data of the relations among concept categories, b. pre-experiments for induction of super-concepts in the EDR Concept Classification.

1．はじめに

様々な概念間の関係に関する大規模な知識を信頼性の高い状態で計算機上に実現することは、自然言語処理技術の大きな課題となっている。そのような知識の表現形態としては、概念を階層状に分類整理し、概念間の関係をその階層内に表現するという形態がある。これは、シソーラスと総称される。

田中・仁科は、自然言語処理に必要となるシソーラスとして、上位/下位関係、部分/全体関係を表すシソーラスなどと共に、第3層のシソーラスとして「動詞の世界から名詞の世界を分類したもの」を挙げている[1]。また、荻野らは、動詞と共起する格要素との関係に着目した動詞の分類を試みた[2]。このような、他の概念に対する振る舞いに基づく概念の階層的な分類は「概念体系」と呼ばれ、シソーラスの中でも特に重要なものの1つである。

概念体系の構築の際には、次の3点が問題となる。

1. 体系構築のための基礎データの準備，
2. 体系内に記述された知識の効率的な検証，
3. 体系内の節点の導出．

1. としては「実例文の人手による分析データ」が考えられるが、網羅的な概念間の関係を実例文だけから得ることは困難であるので、概念体系構築のための基礎データとしては不十分である。一方、「概念の分類カテゴリ間の関係を内省に基づいて記述したデータ（カテゴリ間記述データ）」の場合、概念間の関係を網羅的に記述することが可能であるので、体系構築のための基礎データとして有効である。ここで、カテゴリ間記述データとは、2つのカテゴリを1つの関係ラベルで結んだもの（カテゴリ間記述タプル）の集合であるとする。

一方、基礎データ内の記述の妥当性を検証する作業においては、個々の記述ごとの検証が不可欠であるので、検証の作業量はデータの記述量に比例して大きくなる。したがって、2. のためには、基礎データの記述内容を保存したまま階層化によって記述量を圧縮し、その上で検証作業を行なうことが有効である。

また、概念体系を節点の階層として構築することの利点の1つとして、節点ごとに概念を分類整理しておくことで、記述の保存を節点単位で行なうことが可能となり、体系の保守管理が容易になるということが挙げられる。その際に、節点に分類された概念の内容を把握するためには、それらの概念に意味的な共通性があることが望ましい。したがって、3. において導出される節点は、「振る舞い」だけでなく、「意味」的にも共通性がある概念を分類できるものである方がよい。

そこで、本稿では、a．記述量の圧縮によって記述の効率的な修正を実現し、b．意味的な把握が可能な概念体系の節点を導出するための、カテゴリ間記述データに基づいた概念体系の構築法を提案する。

以下では、2で、概念体系構築の方法を説明し、3で、その方法を用いてEDRで作成された概念分類項目間の関係記述データ（項目間記述データ）[5]から概念体系を試作した結果について述べ、4で、作成した概念体系を利用してカテゴリ間記述データを洗練させる方法について議論する。

2．概念体系構築の方法

本稿で報告する概念体系の構築法は、以下の3つの工程から成る。

a．カテゴリ間記述のグループ化
b．グループの階層化
c．下位ノードの付加

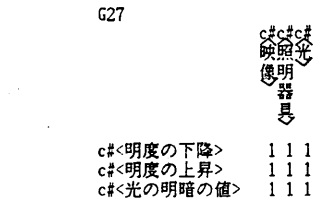
2．1～3で、それぞれについて説明する。

2．1　カテゴリ間記述のグループ化

カテゴリ間記述データのような2項関係の実例データは、1/0行列とみなすことができる。このような1/0行列の部分行列で、すべて1で埋まったもの（完全部分行列）やある程度1で埋まったもののうち、意味的共通性が認められるものをグループと呼ぶことにする。また、1/0行列内からグループを探し出す操作をグループ化と呼ぶことにする。

```
c#<映像>      <-obj— c#<明度の下降>
c#<映像>      <-obj— c#<明度の上昇>
c#<映像>      <-obj— c#<光の明暗の値>
c#<照明器具> <-obj— c#<明度の下降>
c#<照明器具> <-obj— c#<明度の上昇>
c#<照明器具> <-obj— c#<光の明暗の値>
c#<光>        <-obj— c#<明度の下降>
c#<光>        <-obj— c#<明度の上昇>
c#<光>        <-obj— c#<光の明暗の値>
```

(a) カテゴリ間記述データ




```
c#<明度の下降>     1 1 1
c#<明度の上昇>     1 1 1
c#<光の明暗の値>   1 1 1
```

(b) カテゴリ間記述のグループ化

```
G27<-obj—G27

列側
  G27 : (c#<映像> c#<照明器具> c#<光>)
行側
  G27 : (c#<明度の下降> c#<明度の上昇> c#<光の明暗の値>)
```

(ただし、「A:(c#B ..)」は、「A<-instance_of—c#B」の略記)

(c) 所属関係による項目間記述の階層化

図1．カテゴリ間記述のグループ化

カテゴリ間記述データをグループ化した後、グループの列側、行側にそれぞれ記号を与えれば、グ

ループに属するカテゴリ間記述をそれらの記号間の記述（グループ間記述）として抽象化して表現することができる。これは、グループに属するカテゴリの上位概念を設定し、上位概念と各カテゴリの間に所属関係（instance_of で表現）をつけて階層化することに対応している（図1）。

このとき、グループの行側/列側に属するカテゴリの個数をそれぞれｋ，ｌとすると、

$$k l - (k + l + 1)$$

だけ、記述量が減少する。カテゴリ間記述タプルがｋｌだけ減るかわりに、グループの所属関係記述タプルが（ｋ＋１）、グループ間記述タプルが１、それぞれ増えるからである。

そこで、記述量圧縮のために、カテゴリ間記述データをグループ化する。グループ化は、ａ．グループの自動抽出、ｂ．グループの修正、という手順で行なう。以下で、ａ．ｂ．それぞれについて説明する。

２．１．１　グループの自動抽出

単語共起データからボトムアップ的に単語のグループ化を行なう方法として、筆者の一人はＤＭ分解算法を利用した方法を提案した[7]。この方法は、一般の２項関係データのグループ化に応用することができる。そこで、この方法を用いてカテゴリ間記述データのグループ化を行なう。

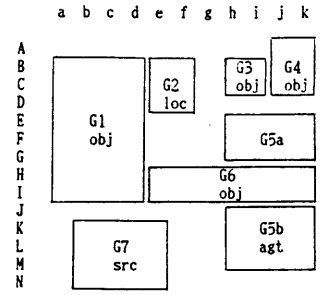
ただし、カテゴリ間記述についてのグループは、完全部分行列であるとする。これは、文献[7]で単語共起についてのグループを、ある程度１で埋まった部分行列としていたことと異なる。単語共起データの場合、可能な単語共起を網羅したデータを得ることは不可能である。これに対してカテゴリ間記述データの場合は、網羅的にカテゴリ間記述を行なうことが前提となっている。そこで、カテゴリ間記述データに対しては、完全部分行列を探すことでグループ化を行ない、元のデータの持つ情報を保存することにした。なお、完全部分行列を求めるためには、文献[7]のアルゴリズムにおいて、パラメータ「１-０反転率ｑ」を０と設定すればよい。

また、文献[7]では、完全部分行列（意味素性と呼んでいた）を大きさ（グループに属する要素の個数）の大きいものから探していた。単語共起データを表現したモデル上での動作を保証するためであった。しかし、カテゴリ間記述データの場合はこの方針にはこだわらず、アルゴリズムの実行の途中でしきい値（例えば６）よりも大きな完全部分行列が見つかるごとに出力することにする。

２項関係データからグループを自動抽出した結果の例を図２(a)に示す。これから、図２(b)(c)の抽象化した記述が得られる。

２．１．２　グループの修正

グループの修正は、ａ．グループにカテゴリを付加したり、ｂ．グループからカテゴリを削除したりすることを繰り返すことで行なう。



（G5a，G5b を合わせて、G5 であることに注意）

(a) グループ化の結果

列側	行側
G1 : (a b c d)	G1 : (B C D E F G H I)
G2 : (e f)	G2 : (B C D)
G3 : (h i)	G3 : (B C)
G4 : (j k)	G4 : (A B C)
G5 : (h i j k)	G5 : (E F G J K L M)
G6 : (e f g h i j k)	G6 : (H I)
G7 : (b c d e)	G7 : (K L M N)

(b) 所属関係の記述

G1 <-obj— G1
G2 <-loc— G2
G3 <-obj— G3
G4 <-obj— G4
G5 <-agt— G5
G6 <-obj— G6
G7 <-src— G7

(c) グループ間の関係記述

図２．グループ化に基づく抽象化した記述

カテゴリの付加

グループに含まれていないカテゴリで、意味的な共通性と振る舞いの共通性から考えて、グループに含まれていてもよいと思われるものがあれば、そのカテゴリをグループに付加する。

カテゴリの削除

グループの中に、他のカテゴリと比較して意味的な共通性の認められないカテゴリが含まれている場合、あるいは、カテゴリ間記述データの不備から振る舞いの共通しないカテゴリが含まれている場合は、そのカテゴリをグループから削除する。

２．２　グループの階層化

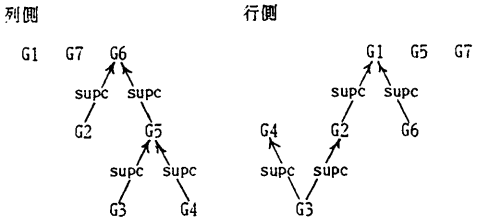
図２(b)にも示したように、カテゴリ間記述をグループ間記述に置き換える場合は、カテゴリのグループへの所属関係を別に記述する。この所属関係の記述量を、グループ間の包含関係（上位/下位関係）の記述を導入することによって減少させることができる。具体的には、以下のようにする。

1. 列側、行側それぞれについて、グループ間の包含関係を調べる。
2. グループを節点とし、グループ間の包含の直前/直後関係を枝とする半順序グラフを作成する。

その枝ごとに上位/下位関係（supc で表現）をグループ間の関係として記述する。以降、グラフの節点としてのグループをノードと呼ぶ。

3. 各ノードについて、ノードに属するカテゴリで下位のノードにも重複して属しているものは、その所属関係の記述を削除する。

各グループに属するカテゴリは、ノードの上位/下位関係を下位（子孫）に辿っていくことによって得ることができる（図3）。



(a) 上位／下位関係による階層化

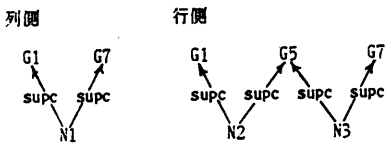
列側
G1 : (a b c d)
G2 : (e f)
G3 : (h i)
G4 : (j k)
G5 : (
G6 : (
G7 : (b c d e)

行側
G1 : (
G2 : (
G3 : (B C)
G4 : (A
G5 : (E F G J K L M)
G6 : (H I)
G7 : (K L M N)

(b) 所属関係の記述の削除

図3．グループの階層化

上位/下位関係にある2つのノードの集合差の大きさをmとすると、3.において、上位/下位関係の記述1つごとに、カテゴリのノードへの所属関係の記述がm個減る。グループ化の方法から、m≧1が言えるから、グループの階層化によって、全体の記述量は減少こそすれ増加することはない。



(a) 交わりノードの導入

列側
G1 : (a
G2 : (e f)
G3 : (h i)
G4 : (j k)
G5 : ()
G6 : (g)
G7 : (
N1 : (b c d)

行側
G1 : (
G2 : (D)
G3 : (B C)
G4 : (A)
G5 : (
G6 : (H I)
G7 : (
N2 : (E F G)
N3 : (K L M)

(b) 所属関係の記述の変更

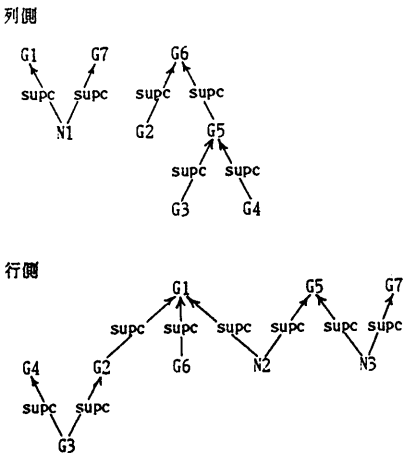
図4．下位ノードの付加

2．3　下位ノードの付加

交わりのある2つのノードの下位に新たにノードを導入することによって、さらに所属関係の記述量を減少させることができる（図4）。具体的には、以下のようにする。

1. 交わりが2以上ある2つのノードを見つけて、それらの下位に新たなノード（交わりノード）を導入する（supcで表現）。
2. 交わりに属するカテゴリを交わりノードに所属させる（instance_ofで表現）と同時に、上位の2つのノードにおけるそれらの所属関係の記述を削除する。
3. 交わりノード間の包含関係を調べて、上位/下位関係で表される部分を修正する。

1.で、交わりを2以上としたのは、2つの上位ノードと交わりノードとの間の上位/下位関係のための記述量が2だけ増加するのを打ち消すためである。交わりに属するカテゴリの個数をnとすると、交わりノードを導入するごとに、（n－2）だけ所属関係の記述量が減少する。



(a) 上位／下位関係

列側
G1 : (a)
G2 : (e f)
G3 : (h i)
G4 : (j k)
G5 : ()
G6 : (g)
G7 : (e)
N1 : (b c d)

行側
G1 : ()
G2 : (D)
G3 : (B C)
G4 : (A)
G5 : (J)
G6 : (H I)
G7 : (N)
N2 : (E F G)
N3 : (K L M)

(b) 所属関係

G1 <-obj— G1
G2 <-loc— G2
G3 <-obj— G3
G4 <-obj— G4
G5 <-agt— G5
G6 <-obj— G6
G7 <-src— G7

(c) ノード間の関係

図5．概念体系

この作業では、1. における交わりのある2つのノードの選び方に任意性がある。したがって、交わりが2以上あるノードの対をすべて計算して求めた後、その中から交わりノードの選択を人手で行なうという方法も考えられる。

また、交わりノードをこの作業の対象に随時加えていくことにより、さらに記述量を減少させていくことも考えられる。しかし、明らかに計算量の爆発を引き起こし現実的ではないので、これは行なわない。

以上の工程により概念体系を得る。図5に、図2(a)のデータから得られる概念体系を示す。なお本体系の作成は、2項関係抽象化支援システム[6]を用いて、計算機支援の下に行なうことが可能である。

3. 概念体系の試作結果

EDRでは、概念の分類カテゴリである概念分類項目（項目）を、モノ概念について約 200個、コト概念について約 250個、それぞれ設定し、それらに対して約 30万の概念を分類する作業を行った[4]。一方、可能な項目間の関係を人手で記述して、網羅的な項目間記述データを作成した[5]。

今回、この項目間記述データを対象に概念体系の試作を行なった。その結果、39,256タプルの項目間記述データのうち、36,562 タプルがグループ化され、2,694 タプルがグループ化されずに残った。グループの個数（ノード間記述の記述数）は 75、交わりノードの個数は 48（列側 37、行側 11）、上位/下位関係の記述数は 178（列側 127、行側 51）、所属関係の記述数は 1,279（列側 978、行側301)であった。したがって、記述量の圧縮率は、

$$(39,256-(75+178+1,279+2,694))/39,256 = 89.2\ \%$$

になる。

試作した概念体系を図6に示す。列側、行側それぞれの上位/下位関係、及び、ノード間の関係記述をネットワーク状に表現し、項目のノードへの所属関係の概略をノードの横に付記した。また、一部のノードには、ノードの子孫を総称する名称を（）で囲んで付記した。

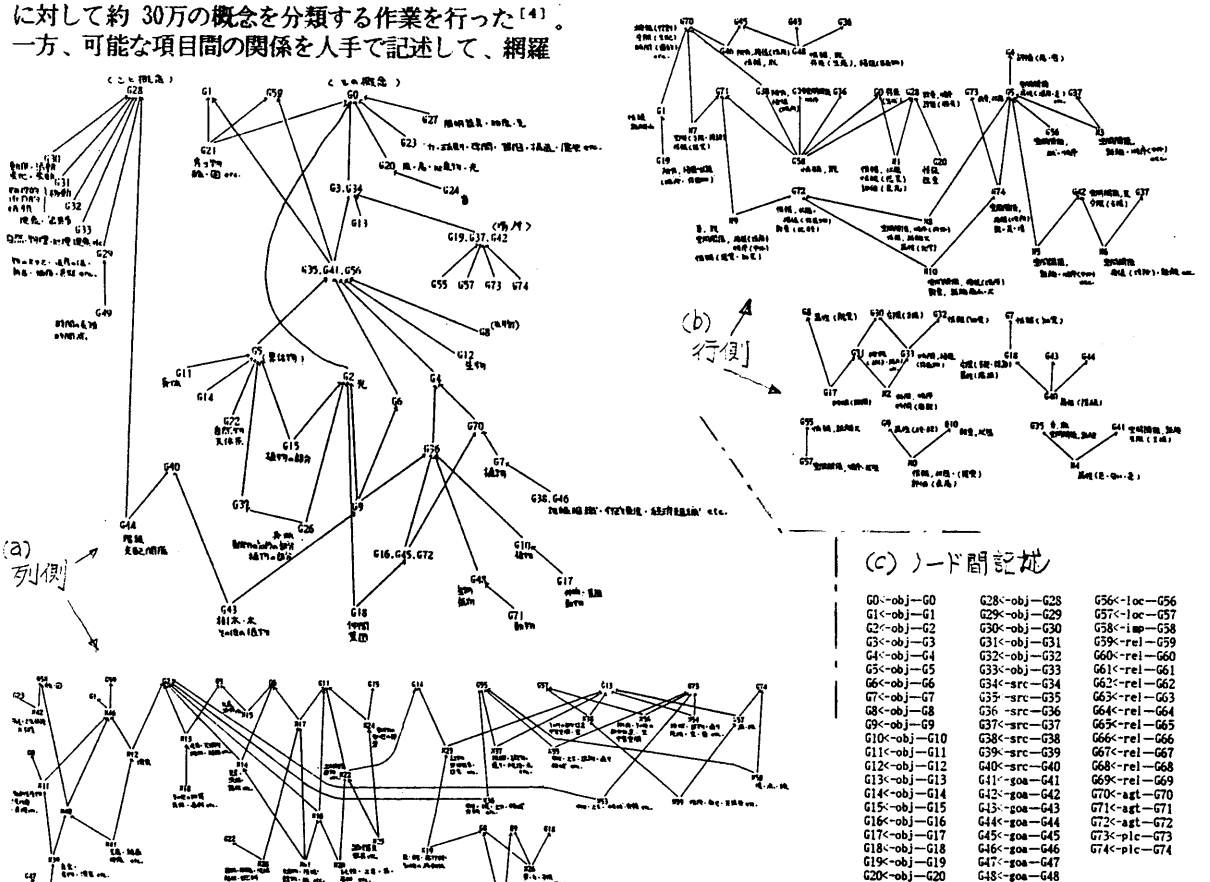


図6．概念体系の試作結果

本概念体系におけるノードは、項目間記述データにおいて振る舞いの共通する項目がまとめられたものである。しかし、1でも述べたように、体系の保守管理及び項目間記述データ検証の観点からは、ノードに属する項目の間に、意味的な共通性が存在することが望ましい。そこで、試作した概念体系内の各ノードに属する項目に意味的共通性があるかどうかを調査した。その際、意味的共通性は、例えば、

N28:(C#<固体>、C#<樹脂>、C#<液体>、
C#<粘体>、C#<燃料>)

に対する「物質」や、

N2 :(C#<時間的順序が変化する>、
C#<回数>、C#<回数が増える>、
C#<ある時間にコトが満ちる>)

に対する「時間」のように、項目を総称する簡潔な表現があるかどうかで判断した。結果を表1に示す。

表1．意味的共通性によるノードの分類

	列側ノード	行側ノード
意味的共通性あり	57	39
意味的共通性なし	13	34
所属項目数 1	10	8
所属項目数 0	32	5

所属する項目数が 0 及び 1 のノードを計算から除いたとしても、半数以上（67%）のノードに意味的共通性が認められたことがわかる。特に、列側のノードは約8割のノードに意味的共通性が認められた。したがって、本体系を1で述べた目的に用いることが可能であると言える。

4．本概念体系の利用法

1でも述べたように、本概念体系構築の主な目的は、a．カテゴリ間記述データの効率的な検証、b．概念体系内の節点導出のための予備実験、である。以下では、特にa．に関して前節の、項目間記述データを基にした概念体系の試作結果（試作体系）を例にして説明する。

本概念体系を利用することにより、項目間記述データの効率的な検証が可能になる主な原因は以下の5点である。

a．ノード単位で検証できること。
b．適切なレベルにおける検証が可能なこと。
c．体系における項目の相対的な配置に基づく検証（大域的な検証）が可能なこと。
d．階層外記述に対する検証が行なえること。
e．体系構築作業を分業化できること。

以下に、それぞれについて詳しく述べる。

ノード単位の検証

本体系においては、項目間の関係はすべてノード間記述として記述されるので、項目間記述データの検証もノードごとに行なうことが可能である。ノード間記述の表現している項目間記述の集合は、ノードの子孫を辿ることにより図7のように得られる。すなわち、図7のようにノード間記述の行側・列側のノードをそれぞれ子孫全体について展開して得られるすべての項目の組み合わせについて、元のノード間記述と同じ関係が成立する。ノード単位で項目間記述を検証する場合、このように子孫の項目を展開して一覧した上で、検証を行なう。

```
G5<-obj-G5

列側
G5 :(c#<具体物>)
  _ G11:(c#<身体>)
      _ N16:()
          _ N20:(c#<ひも類> c#<針類> c#<人形> c#<管> c#<網> c#<棒> c#<台>
                 c#<刃物> c#<工具> c#<接続具> c#<操作器具> c#<医療器具>
                 c#<筆記具> c#<遊具> c#<楽器> c#<数物>
                 c#<貨幣> c#<券> c#<札（フダ）> c#<宝石> ... 計 22)
          _ N61:(c#<化合物> c#<金属> c#<塩・苷・かすり> c#<紙>
                 c#<陸地（地圏構成物）> c#<水地（水圏構成物）>
                 c#<繊維質> c#<紙質> c#<木質> c#<金属質> c#<土石質>
                 c#<嗜好品> c#<飲み物> c#<野菜> c#<調料>
                 c#<着装品の材料> c#<衣服の素材> ... 計 35).
      _ N17:()
          _ N28:(c#<固体> c#<樹脂> c#<液体> c#<粘体> c#<燃料>)
          _ N61.
      _ N22:(c#<土木施設> c#<区域施設> c#<建物>)
          _ N25:(c#<調理器具> c#<食器> c#<容器（食器以外）>
                 c#<洗浄・清掃用具> c#<家具>)
          _ N20.
      _ N24:(c#<動物のその他の部分>)
          _ N25.
  _ G14:()
      _ N23:(c#<人工物> c#<民族・種族> c#<組織性のないその他の集団>
             c#<照明器具> c#<計算用具> c#<建具> ... 計 10)
          _ N19:(c#<車（地図移動体）> c#<船（水図移動体）>
                 c#<飛行機（気図移動体）> c#<その他の移動体>)
          _ N22.
  _ G15:(c#<植物の部分>)
      _ N24.
  _ G22:(c#<自然物> c#<天体系>)
  _ G39:(c#<生物の部分> c#<動物の部位> c#<消化器> c#<泌尿器>
         c#<神経系> c#<感覚器> c#<植物の部分> ... 計 12)
      _ G26:(c#<身体> c#<動物のその他の部分> c#<植物の部分>)
      _ N62:(c#<音楽> c#<伝達手段> c#<説> c#<伝達内容>
             c#<書物> c#<文学のジャンル> c#<演芸>)

行側
G5 :(c#<空間的に変化する> c#<一体のものの分離> c#<身につける物を脱ぐ>
     c#<接触相手から離れる> c#<共点に空間的に近付く> c#<空間的距離を保つ>
     c#<通ずかる> c#<空間的順序が相対的に変わる>
     c#<物の形や大きさの変化> c#<物の形> c#<物の大きさ> c#<物の模様>
     c#<物の温度の減少> c#<物の温度の増大>
     c#<色彩> c#<匂いの変化> c#<味の変化> ... 計 22)
  _ G56:(c#<ものが空間的にすく> c#<空間的順序が変わる>)
  _ G74:(c#<ある場所における空間的順序の形成> c#<ものが一体化する>
         c#<場所やモノへの接触> c#<身につける物を着用する> c#<付着>
         c#<物が空間的に広る> c#<物が空間的に集まる> c#<ものが空間的に満ちる>)
      _ N10:(c#<住む> c#<場所やモノへの接触の離続>
             c#<数量や程度の減少> c#<数量や程度の増大>)
  _ N3 :(c#<空間的にある距離移動する> c#<空間的にある方向からある距離移動する>
         c#<空間的に出る> c#<生理的に出る>)
  _ N5 :(c#<空間的にある距離移動する> c#<空間的にある方向からある距離移動する>
         c#<空間的に入る> c#<摂取>)
  _ N8 :(c#<空間的に入る> c#<摂取> c#<化学的性質の変化> ... 計 4)
```

図7　G5 の子孫

3で示した通り、本試作体系の半数以上のノードには意味的共通性がある。また、意味的共通性がないノードに関しても、ノード内をいくつかに分割すれば、そのそれぞれについて意味的共通性が認められる場合が多い。したがって、各ノードの子孫全体を1つのものとして把握することが比較的容易である。

例えば、図7．において、列側は、G5 以下は「具体物」、N20 以下は「形状、工具・器具、券など」、N61 以下は「材質、陸・水地、飲食物、材料など」、N23 以下は「集団、器具、移動体」、G39

以下は「生物の部分、音楽・書物・演芸など」などと把握できる。一方、行側は、G5 以下は「空間関係」、G74 以下は「順序形成、接触、集散、満ちること、など」、N3 以下は「移動、出ること、など」などと把握できる。（ここでは便宜的に把握の仕方を「」で囲って表現した。しかし、実際にノードの子孫を１つのものとして把握する際には、ノードの子孫に属する項目に関する具体的な印象があれば十分であり、特に言葉にする必要はない。）

ノードをそれぞれ１つのものとして把握してノード間記述タプルを検証する。その場合、項目間記述タプルをランダムに検証する場合と比べて、記述量の圧縮率だけ作業の効率化が望める。ノード間記述タプルごとに、その子孫の項目間記述タプルの正否について一括して検証できるからである。例えば、図７のノード間記述タプル

G5 <-obj－ G5

の検証を、列側、行側のノードG5をそれぞれ、「具体物」「空間関係」として把握し、

「具体物」<-obj－「空間関係」

の妥当性を検証することで、このノード間記述の子孫の 3,990 個の項目間記述タプルを一括して検証できる。

適切なレベルにおける検証

前節のようなノード単位の検証だけでは、ノード間記述の十分な検証が行なえない場合がある。ノードの把握の仕方が粗すぎると感じられる場合である。その場合、子孫ノードのレベルでさらに細かい検証を加えることが必要になる。このときも、子孫ノードをそれぞれ１つのものとして把握することができれば、やはり作業の効率化が望める。例えば、先のノード間記述

G5 <-obj－ G5

の妥当性を検証する場合、

G11		G5(ﾉｰﾄﾞのみ)
G14	<-obj－	G56,74
G15,39		N3,5,8

のように、子ノード間の記述に展開してから、各子ノードを１つのものとして把握した上で、それぞれの記述を一括して検証する。

また、子ノード間記述を検証する代わりに、ノードの子孫の項目を流し読みして、ノードに対して抱いていた印象と相違がないことを確認するという方法も考えられる。このとき、印象と相違する項目対しては、さらに詳しい検証を加える。

以上のように、項目間記述の検証をレベルに合わせて段階的に行なえるので、ランダムに検証する場合に比べて、ａ．無駄な作業を自然に避けられる、と同時に、ｂ．作業に連続性が生じるので作業者の負担感も軽減される。

大域的な検証

項目間記述タプルの作成作業は、項目の組み合わせごとに関係の可能性を内省することによって行なわれる。その際、他の項目間記述タプルとの整合性は基本的には考慮されない。つまり、項目間記述タプルは、そのタプルに対する局所的な内省のみによって作成される。したがって、概念体系内における各項目の相対的な配置に基づいて、大域的に項目間記述を検証することが必要となる。

反対概念を表すコト項目などは、あらかじめ同じ振る舞いをすると予想される。例えば、c#<認知対象との認知的距離減少>、c#<認知対象との認知的距離増大>、という２つの項目は、それぞれ反対概念として設定されたものであるので、振る舞いは同じになると考えられる。ところが、図６の概念体系においては、前者は、G1、G20、G23、後者は、G55、N8、とまったく異なるノードに所属している。この差異が妥当であるかどうかは、各項目に対する記述を調べることにより判断しなければならない。

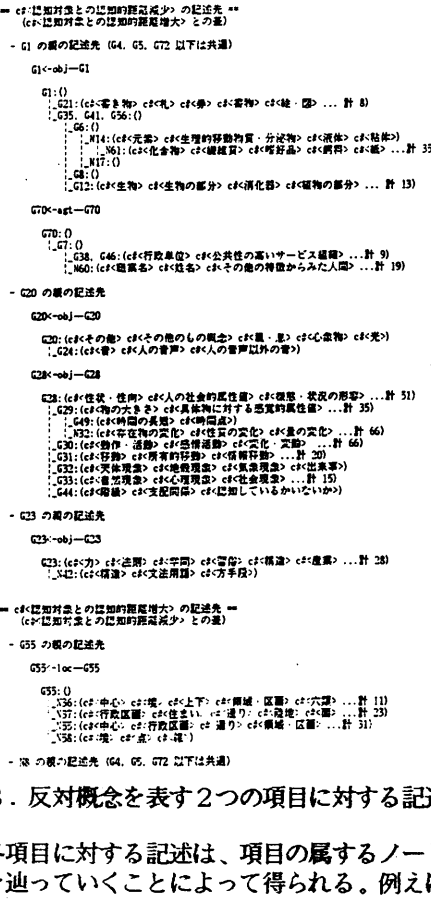


図８．反対概念を表す２つの項目に対する記述

各項目に対する記述は、項目の属するノードの祖先を辿っていくことによって得られる。例えば図８では、G1、G20、G23、及び、G55、N8、の祖先ノードの記述を展開することにより、上記の２つの項目に対するすべての記述を得ている。これを基にノード単位の記述の検証を行ない、両項目の振る舞いの

差異の妥当性を判断する。その際に、不適切な項目間記述タプルの発見が期待できる。

反対に、軸の異なるコト項目[4]や、項目の一覧表における上位項目が異なるモノ項目[3]のような、まったく異質の項目が共通のノードに属している場合にも、不適切な項目間記述タプルが発見できる可能性がある。例えば、図６の概念体系において、G41 ≒ のコト側には、「空間関係」を表す他の３つの項目とは異質な C#<恩や害を与えている> という項目が所属している。また、G44 のモノ側には、C#<階級> C#<支配関係>と共に C#<教育組織> C#<経済組織> が所属している。これらのノードについても、祖先をたどり、これらが振る舞いを同じくする項目間記述を一覧して、その妥当性を検討する（図９）。

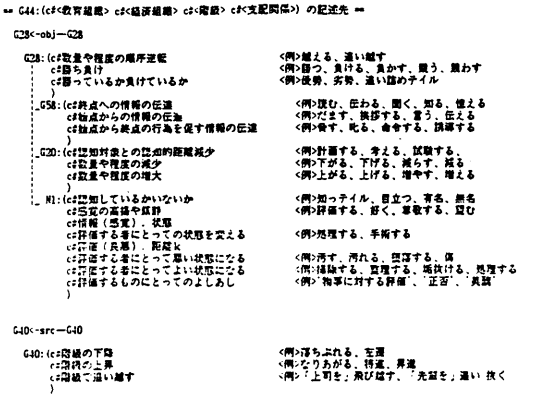


図９．同一ノード内の異質な項目に対する記述

階層外記述に対する検証

項目間記述データをグループ化する際、どのグループにも属さない項目間記述タプル（階層外記述タプル）が最後に残る。これらのタプルは、各項目の「個性的な」特徴を表現していると考えられる。例えば図１０はそのような階層外記述タプルの例である。逆に、階層外記述タプルが各項目の個性的な特徴を表現していない場合は、必要な項目間記述タプルが欠落していたり、不適切な項目間記述タプルが存在していたりすることになる。したがって、階層外記述タプルを調べることによっても、項目間記述データの検証が行なえる。

```
c#<味の値> <-goa- c#<味の変化>
c#<匂いの値> <-goa- c#<匂いの変化>
c#<色彩の値> <-goa- c#<色彩の変化>
c#<教育組織> <-plc- c#<機能の程度が上がる>
c#<経済組織> <-agt- c#<所有物の終点への移動>
c#<経済組織> <-agt- c#<所有物の始点からの移動>
```

図１０．階層外記述タプルの例

体系構築作業の分業化

本概念体系構築法の特徴として、体系構築の工程が、ａ．項目間記述データの作成、ｂ．項目間記述の階層化による概念体系の構築、という２つの独立した工程からなる、ということが挙げられる。このうち、ｂ．は（半）自動的に実現できるので、実際には、ａ．の工程を繰り返していく過程で随時ｂ．を行ないその結果を再びａ．に反映させる、という形になる。これにより、試行錯誤的な体系の構築が可能となる。また、両工程を分業化することにより、各作業をそれぞれの専門家が集中的に遂行することで、作業効率の向上が図れる。

６．おわりに

カテゴリ間の関係記述データを、その内容を保存したまま階層化して概念体系を構築する方法を提案した。この方法に基づき概念体系を試作した。その結果、記述量が 89.2 % 圧縮されること、67 % のノードに意味的共通性があることを確認した。これにより、本概念体系が、ａ．カテゴリ間記述データの検証作業の効率化、ｂ．概念体系における節点の導出のための予備実験、のために利用できることがわかった。

今後の課題としては、以下の２点が挙げられる。

1. 項目間記述データの検証・修正作業に繰り返し用いることで、本構築法の有効性を確認する。

2. 項目間記述データよりさらに細かく個別概念間の振る舞いの差を記述したカテゴリ間記述データに対しても、本構築法により有効な概念体系が得られることを確かめる。

【謝辞】

本研究の機会を与えてくださった、ＥＤＲの横井所長、並びに、終始有益なご意見や励ましを頂いた安達久博研究員（現在、㈱東芝 総合研究所）をはじめとするＥＤＲの研究員の皆様に感謝いたします。また、データの整理に協力してくれた茂木みどり研究補助員に感謝します。

【参考文献】

[1] 田中穂積、仁科喜久子：上位/下位関係シソーラスＩＳＡＭＡＰの作成，情報処理学会自然言語処理研究会，64-4，(1987)．

[2] 荻野孝野、他：共起関係に着目した動詞分類について，情報処理学会自然言語処理研究会，71-2，(1989)．

[3] 日本電子化辞書研究所：概念辞書（第２版），TR-012，(1989)．

[4] 岸本行生、他：概念分類項目の設定，情報処理学会第４１回全国大会，(1990)．

[5] 横田英司、他：概念分類項目間の関係記述，情報処理学会第４１回全国大会，(1990)．

[6] 松川智義、他：２項関係抽象化支援システムの基本機能，情報処理学会第４１回全国大会，(1990)．

[7] 松川智義、他：共起関係に注目したＤＭ分解と確率的推定による単語のクラスタリング，情報処理学会自然言語処理研究会，72-9，(1989)．