

フレーム形式の知識ベースにおける単語概念の学習アルゴリズム

椎谷秀一 瀧口伸雄 小谷善行
東京農工大学工学部電子情報工学科
コンピュータサイエンスコース

入力された例文の深層格構造を用いて知識ベースにおける概念の動的な補足・修正を行う手法を実現した。動的階層化アルゴリズムとして、カテゴリー化アルゴリズム、上方組替えアルゴリズム、下方組替えアルゴリズムの3種類と、上下関係の入力による組替えを提案する。深層格構造における各概念の他の概念との意味的なつながりの規則性をその概念の意味と定義し、この意味を知識ベース上で表現する。本手法が実際に自然言語処理システムに適用できるかを検証するために、100の例文を入力し、どのような知識ベースが構築されるかを実験した。これにより、構築された知識ベースの正当性を確認した。

A Learning Algorithm of Word Concepts in Frame Based Knowledge Base System

Shuuichi SHIITANI Nobuo TAKIGUCHI Yoshiyuki KOTANI

Department of Computer Science

Tokyo University of Agriculture and Technology

2-24-16 Nakacho Koganei Tokyo 184 Japan

A method of concept learning is designed, which constructs and modifies concepts dynamically in a knowledge based system, using deep case structures from example sentences. The method consists of the three algorithms, a categorizer and an upper hierarchy recomposer, the lower hierarchy recomposer and a hierarchy relation replacer. A meaning of a word is defined as a set of regulation that describes semantic connections between each word and other words in deep case structures. We construct the experimental frame based knowledge base system from 100 sentences and investigate the contents from the point of an application of a natural language processing system. As a result, our method is evaluated to be effective in constructing a knowledge base system.

1 はじめに

初期の自然言語処理の研究は、理論を確立することが目的となっていた。それらの研究は少ない単語で成立している世界を対象にしていた。代表的な研究に、積み木の世界を対象とした T. Winograd の SHRDLU システム[1]がある。SHRDLU は積み木の世界に対する指令・質問を処理するために、200語の語彙に関する情報を所持している。

1980年以降になると、理論研究の成果と計算機技術の進歩により、自然言語処理システムの実用化を考えるようになった。それにとまって、対象にする世界も、小さな世界から大きな実世界へと変わった。

多量の語の辞書を作成するには、多大な労力を必要とする。入力する手間や時間を考えると、手入力での辞書作成は困難である。

辞書をあらかじめ用意しておくことが困難なことから、学習によって辞書を構築しようという動きがある。知識工学において知識学習に関する研究が盛んに行われたこともあり、辞書学習の研究は数多く行われた。

ところが、計算機辞書にしても辞書学習の研究にしても、統語情報に関するものはかなり整備されてきているが、意味情報にはあまり目が向けられていない。その理由として、意味情報の表現方法がしっかりと確立されていないことがあげられる。

学習に用いられている意味表現形式は、大きく分けて二つの方法がある。一つは、単語より下位レベルの最小単位（基本概念）を用意し、その組合せで単語の意味を表そうというものである[7]。この意味表現形式を用いた単語の意味学習の問題点として、A)基本概念の選定が困難である、B)適切な例文の入力が必要である、C)名詞の意味が学習できない、などが挙げられる。

もう一つの意味表現形式は、意味素性と述語素の考えを用いたものである[6]。意味素性とは、その単語がどのような意味カテゴリーに含まれるものかを表す情報である。この意味素性と述語素の

学習には、意味素性の学習と述語素の学習が依存しあっているという問題がある。つまり、意味素性を学習するには述語素が、述語素を学習するには意味素性が必要になってくる。両方を同時に学習することはできない。

本研究では、前節で述べた問題点を考慮して、単語の意味学習を行うシステムを作成する。作成する学習システムは、実用的な自然言語処理システムに組み込んで利用することを目標にする。具体的には、A)無作為な例文から学習可能であること、B)一つの例文ごとに帰納学習すること、C)品詞にかかわらず、自立語をすべて学習できること D)細かい意味を学習すること、に設計方針を置く。

2 本研究の意味表現形式

まず、本研究で扱う単語の意味を定義する。

文の意味の表現形式としていくつかの方法があるが、その中で深層格を表す意味ネットワークを取り上げる。

ある単語を含むような例文がいくつか入力されると、その単語のネットワーク上での他の単語とのつながりかたにその単語独自の規則性が見いだされる。つまり「その単語がどのようなラベルのノードを持つことができるかという規則」と「個々のノードがどのような単語とつながるかという規則」である。この規則を単語の意味と定義する。

本システムでは、一つの単語は一つの意味しか持たないと仮定する。複数の意味がある単語が文の中でどの意味で使われているかは、簡単には判断できない。

単語の意味の計算機上での表現には、我々が研究してきた物語知識ベース FAIRY を用いた。知識ベース FAIRY は、Minsky のフレーム理論をもとに設計された。FAIRY におけるフレームの形式を図1に示す。

ファセットは、スロット値の属性を表す。

value ファセットは、スロット値が事実であることを表している。default ファセットは、事実ではないが、一般的にそのスロット値であると考

えられる場合に使う。require ファセットは、スロット値を制限するために使う。

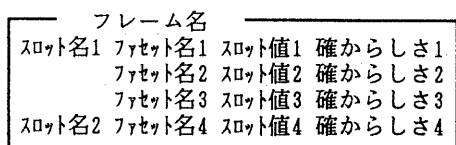


図1 FAIRY におけるフレーム形式

3 システムの概要

本研究では、例文から単語の意味を学習するシステムを作成した。システムの概観図を図2に示す。

本研究で設計・作成したのは、入力文格納・階層化の部分である。入力文格納・階層化では、深層格構造をもとにして実体領域を拡張する。日本語文から深層格構造を得るためには、知識ベース FAIRY の表記領域・言語領域はすでに構築されていないなければならない。

4 知識ベース

知識ベース FAIRY は、知識の内容をはっきりと

区別するために、次の四つの領域に分かれている。

- 表記領域：文字列・形態素に関する知識
- 言語領域：統語に関する知識
- 実体領域：意味に関する知識
- 管理領域：知識ベース管理用のメタ知識

実体領域はものごとの概念を記述するところである。ある一つ概念に一つのフレームが対応し、そのフレームのつながりによって、文の知識が表現される。

個々の特定の概念に対応し、その性質を記述しているフレームをインスタンスフレームと呼ぶ。一方、一般的な概念に対応するフレームをクラスフレームと呼ぶ。

4.1 フレームの分類

本研究では知識ベースの階層化と細かい意味記述の実現のために、クラスフレームを利用方法別に3種類に分類した。

1) 語意フレーム

語意フレームは、各概念の一般的な意味を直接に表記するものである。一つ概念に一つの語意フレームが対応している。

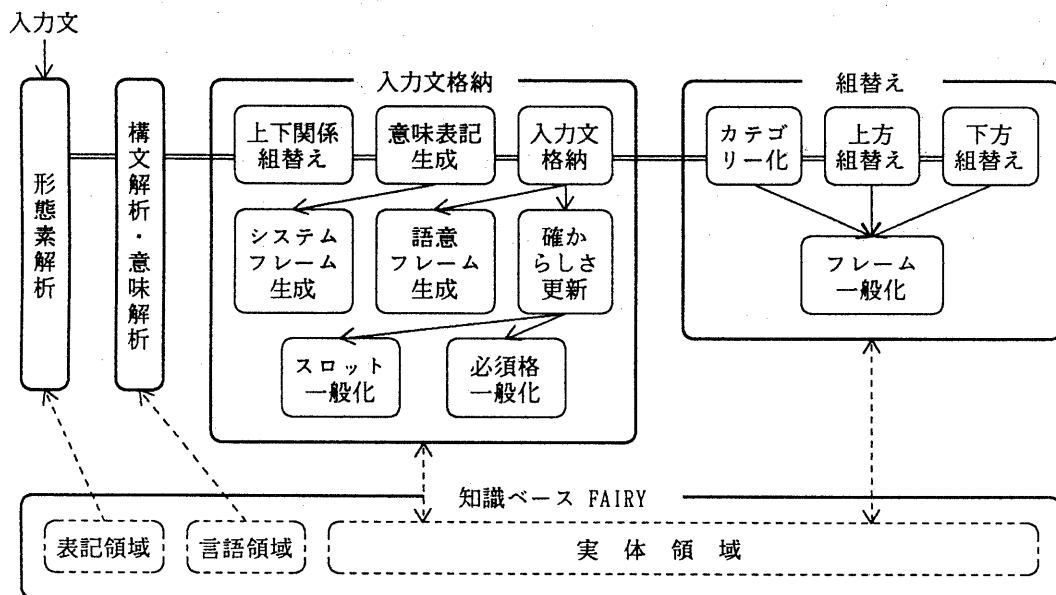


図2 システム構成図

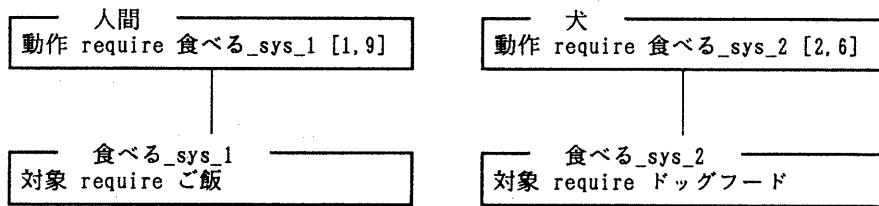


図3 人間と犬の「食べる」の違い

語意フレームには、A)そのフレームがどのようなスロット名を取ることができるか、B)各スロット名にどのようなスロット値を取ることができるか、といった情報が記述される。これはそのフレームが表す概念の意味である。

2) オートフレーム

本研究では、同じ性質を持ったフレームを一つのカテゴリーにまとめることによって、知識ベースを階層化する。カテゴリーを表すためにシステムが自動的に生成したフレームをオートフレームと呼ぶ。

オートフレームはカテゴリーに対応しており、そこに記述されている性質は、そのカテゴリーの性質である。よって、オートフレームの性質は、そのオートフレームの下位にあるすべての語意フレーム・オートフレームが、共通して持っているものであるといえる。

3) システムフレーム

人間と犬とでは、食べる物が違う。これは、人間が行う「食べる」と、犬が行う「食べる」が、それぞれ異なる対象を取ることの意味している。

そこで、人間の「食べる」と犬の「食べる」を図3のように区別する。この制限を行うのがシステムフレームである。

4.2 「スロット」スロット

各フレームがどのようなスロット名を取ることができるかを記述するためのスロット名が「スロット」である。「スロット」は、スロット値として、そのフレームが取り得るスロット名を取る。

4.3 確からしさ

1) インスタンスフレームの確からしさ

インスタンスフレームの確からしさは、事実なのかそれとも推測なのかという内容の信頼性を表している。

確からしさは、0から100の整数で表される。大きいほどそのスロットの信頼性が高く、小さいほどそのスロットの情報は信頼できないものになる。

確からしさが100ならば確実な事実の情報を意味し、確からしさが0ならばまったく信用できないたらしめな情報を意味する。

2) 語意フレームの確からしさ

語意フレームの確からしさは学習結果がどのくらい信頼できるのかを表している。

語意フレームの確からしさは二つの要素からなるリスト形式で記述する。第2要素には入力されたフレーム中に対応するスロット名が出現した回数が入り、第1要素には、そのうち対応するスロット値が出現した回数が入る。

3) オートフレームの確からしさ

オートフレームの確からしさは、その性質がそのカテゴリー全体の性質としてどのくらいの確であるかを表している。

オートフレームの確からしさは二つの要素からなるリスト形式で記述する。第2要素にはそのオートフレームよりも下位にある語意フレームの数が入り、第1要素にはそれらの語意フレームのうち対応するスロットを伴っているものの数が入る。

5 意味情報の獲得

5.1 意味表記

語意フレームに記述された意味を表記するため

のスロットを意味表記と呼ぶ。意味表記はスロット値として語意フレーム・システムフレーム・オートフレームのいずれかを取る。

入力された深層格構造の内容が1レベル先のフレームしか制限していない場合は、意味表記は深層格構造のファセットを require に変えただけのものになる。一方入力された深層格構造の内容が2レベル以上先のフレームを制限している場合は、意味表記のスロット値はシステムフレームになる。

5.2 システムフレームの生成

フレーム F_1 の S_1 スロットのスロット値が F_2 であるとする。もしフレーム F_2 にスロットが一つ(S_1 スロットの逆スロット)しかない場合、 F_2 は語意フレームに変換する。

インスタンスフレーム F_2 に二つ以上のスロットがある場合(図4)は、フレーム F_1 の S_1 スロットのスロット値はシステムフレームになる。

システムフレームを生成するにはまず F_2 から S_1 の逆スロットの S_2 を削除する。新しくシステムフレーム名を作り、そこに残ったスロット($S_3 \sim S_n$)を格納する。

ここで、 $F_3 \sim F_n$ がインスタンスフレームだった場合、それらが二つ以上のスロットを持っていれば、それらもシステムフレームに変換する。

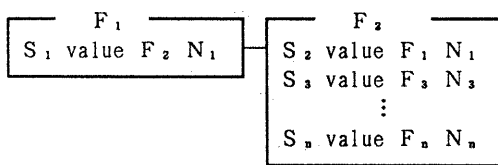


図4 システムフレームに変換される入力フレーム

5.3 語意フレームへの意味表記の格納

入力フレームの内容を意味表記に変換したら、知識ベースの内容と比較し、知識ベースを拡張する。入力されたフレームが語意フレームならそのフレーム自身に、インスタンスフレームならば対応する語意フレームに意味記述を行う。語意フレームがまだ知識ベースにないときは、インスタンスフレームから語意フレームを生成する。

5.4 語意フレームの生成

入力されたインスタンスフレームが知識ベースにないときは、インスタンスフレーム名の添字を削除して語意フレーム名を導き出す。

語意フレーム名が決定したら、そのフレームが知識ベースに存在するかどうかを調べる。もし存在しなければ、新しく作らなければならない。新しい語意フレームは、制限が何もされていないのでトップノード「実体領域」の下に位置させる。

5.5 「スロット」スロットの確からしさの更新

フレームが入力されたら、そのフレームに対応した語意フレームの「スロット」スロットの確からしさの第2要素をインクリメントする。

入力されたフレームに記述されているスロット名がすでに知識ベースの対応した語意フレームに存在するときは、「スロット」スロットの確からしさの第1要素をインクリメントする。

そのスロット名が初出の場合は、対応する語意フレームに新しい「スロット」スロットを記述する。

5.6 その他のスロットの確からしさの更新

「スロット」スロットの確からしさの第1要素をインクリメントすると同時に、入力されたスロット名と同じスロット名を持つスロットの確からしさの第2要素をインクリメントする。

次に入力されたスロットと同じスロットが対応する語意フレームに存在するときは、そのスロットの確からしさの第1要素をインクリメントする。

語意フレームに存在しないときは、新しくその語意フレームに記述する。

6 階層化アルゴリズム

入力された深層格構造から得た意味表記を知識ベースに格納すると、その新しい意味表記にしたがって階層木を組み替える必要がでてくる。しかし階層木全体を組み替えるのでは、木が小さいときはよいが大きくなると計算時間が多くかかる。本研究では、入力によって変更のあったフレーム

に着目し、そのフレームと周囲のフレームとを新たに組み替える必要がある部分だけ組替えを行う。

6.1 親密度

組替えは基本的に同じスロットを持つフレーム同士を一つのカテゴリーにまとめる作業である。どのフレームとどのフレームを一つのカテゴリーにまとめればよいかを決定するのが親密度である。

親密度は、二つのフレームがどのくらい似ているかを表しており、値として0以上の整数をとる。値が大きいかほどその二つのフレームは似通っていることになり、親密度が0ならその二つのフレームを新たに一つのカテゴリーにまとめることができないことを意味している。

二つのフレームがどのくらい似ているかとは、すなわち同じスロットをどれだけ持っているかということである。しかしスロット値が異なってもそれらが互いに上下関係にある場合は、上位のスロット値が下位のスロット値を含むことから、両者は似ているものととらえられる。

フレーム F_x のスロットの集合を、

$$S_x = \{ (N_{xm}, V_{xm}) \mid 1 \leq m \leq n_x \}$$

N:スロット名 V:スロット値 n_x :スロット数

とする。またフレームAとフレームBが上下関係にあるとき、その間にある ako-example ノードの数を上下距離 L とする。A=Bならば、上下距離は0となる。AとBが上下関係にない場合は、 $L = \infty$ である。二つのスロット (N_1, V_1) と (N_2, V_2) がどのくらい似ているかを示す関数 I を、

$$I((N_1, V_1), (N_2, V_2)) = \begin{cases} 100 / (L(V_1, V_2) + 1) & (N_1 = N_2 \text{ のとき}) \\ 0 & (N_1 \neq N_2 \text{ のとき}) \end{cases}$$

とする。以上の定義のもとで親密度 $C(A, B)$ は、

$$C(A, B) = \sum_i \sum_j I((N_{Ai}, V_{Ai}), (N_{Bj}, V_{Bj}))$$

となる。

6.2 カテゴリー化アルゴリズム

階層化のはじめは、入力によって変更のあったフレーム(中心フレーム)と兄弟関係にあるフレームに着目して階層化を行う(図5)。

互いに兄弟関係にある二つのフレームA, Bを一つのカテゴリーにまとめるアルゴリズムを以下に示す。A, Bの親フレームをC, AとBに共通のスロットの集合をSとする。

[カテゴリー化アルゴリズム]

- 1) Cがオートフレームで、Cの子フレームがAとBだけの場合、新たにSを記述するフレームXをCとする。
- 2) (1)以外の場合は、新しくSを記述するためにオートフレームXを生成する。XはCの子フレームに位置させ、AとBをXの子フレームにする。
- 3) XにSを記述する。
- 4) A, Bのうち、オートフレームであるものからSを消去する。
- 5) Sを消去した結果 ako-example 以外のスロットがなくなった場合、
 - 5-1) そのフレームを消去する。
 - 5-2) AまたはBの子フレームは、Xの子フレームにする。
 - 5-3) 新たに兄弟フレームが増えたため、AとBのうち残っているフレームを中心フレームとして階層化を行う。
- 6) Xに変更があったため、Xを中心フレームとして階層化を行う。
- 7) Cを中心フレームとしてフレーム一般化を行う。

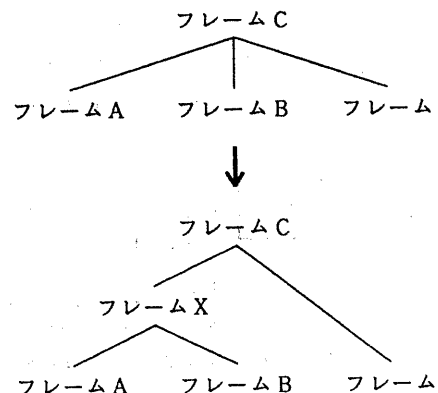


図5 カテゴリー化

6.3 上方組替えアルゴリズム

カテゴリー化だけで階層化を行うと、例文の入力順に極端に依存した知識ベースが生成される。入力順への依存を減らすためには、より広範囲のフレームの階層化が必要である。

そこで、兄弟フレームとのカテゴリー化の次に

親フレームの兄弟フレームと比較して知識ベースを変更する。これを上方組替えと呼ぶ。

上方組替えを行うには、中心フレームとその親フレームとの上下関係を壊さなければならない。そこで中心フレームとその親フレームとの親密度を求め、それよりも選んだフレームの親密度が高い場合に限り上方組替えを実行する(図6)。

中心フレームをA、上方組替えの対象となるフレームをBとしたときの上方組替えアルゴリズムを以下に示す。Aの親フレームをC、Cの親フレームをDとする。

[上方組替えアルゴリズム]

- 1) Aがオートフレームの場合、Cに記述されているスロットをAにコピーする。
- 2) AをDの子フレームの位置に移動する。
- 3) Cのスロットの確からしさを訂正する。
- 4) Cの子フレームがオートフレーム一つの場合、そのオートフレームに記述されているスロットをCにコピーし、そのオートフレームは消去する。
- 5) AとBとでカテゴリー化を行う。
- 6) Cを中心フレームとしてフレーム一般化を行う。

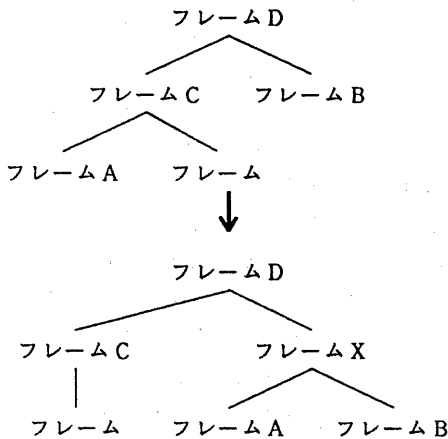


図6 上方組替え

6.4 下方組替えアルゴリズム

上方組替えの逆の処理を行うのが下方組替えである。中心フレームと、中心フレームの兄弟フレームの子フレームとを比較して知識ベースを変更する。

兄弟フレームの子フレームの集合をNとする。

Nの要素に対して、それぞれの親フレームとの親密度と中心フレームとの親密度を計算する。親フレームとの親密度よりも中心フレームとの親密度が大きいものの集合をN' とすると、N' のうち中心フレームとの親密度が最も高いものが下方組替えの対象フレームとなる。

下方組替えアルゴリズムは対象フレームから見れば上方組替えアルゴリズムと同じである。そこで対象フレームが決定したら、対象フレームを中心フレーム、中心フレームを対象フレームとして上方組替えアルゴリズムを行う。

6.5 上下関係の組替え

ako-example は知識ベースの階層構造を構築するための重要なスロットである。入力された深層格構造に ako-example スロットが含まれているときは、それだけは他のスロットとは違った処理を行う。ako-example の処理は、入力されたときの知識ベースの状態に依存する。

A) 両方のフレームが知識ベースにない場合

F₁の ako がF₂であるという入力があったとする。このとき知識ベースにF₁、F₂の語意フレームが存在しない場合はそれぞれの語意フレームを生成する。入力にしたがってF₁をF₂の子フレームとし、F₂は「実体領域」フレームの子フレームに位置させる。「実体領域」とF₂の間の確からしさは100になる。F₂とF₁の間の確からしさは入力のとおりである。

B) 上位のフレームが知識ベースにない場合

知識ベース上のF₁の上位フレームとF₂の関係を決定するには情報が足りない。そこでこのような場合は、F₁の親フレームの位置にF₂を格納する。もとのF₁の親フレームをCとすると、F₂の親フレームはCになる。F₂とCの間の確からしさはまったく不確定なものなので0にする。ただしCが「実体領域」の場合だけは確からしさは100になる。

C) 下位のフレームが知識ベースにない場合

F₁が知識ベースにない場合は、F₁をF₂の子フレームになるように生成すればよい。そのときの

確からしきは入力のとおりである。

D) 上下関係が正しく存在する場合

すでに知識ベースに入力と同じ上下関係があるときは、その入力によって行うことはなにもない。ただし知識ベースの確からしさが入力の確からしさより小さければ、確からしさを入力のものに変更する。

E) 知識ベースの上下関係と逆の入力の場合

知識ベースでは F_2 の上位に F_1 があるような場合は矛盾が生ずる。 F_2 から F_1 までの階層のどこかに間違いがあるか、あるいは入力が間違っていると考えられる。そこでそれらのうち確からしさの一番小さい部分が間違いであると推定し、組替えを行う。このときのアルゴリズムを以下に示す。

[上下関係の矛盾解消アルゴリズム]

- 1) F_2 と F_1 の間にあるフレームのうち語意フレームの集合を S 、オートフレームの集合を A とする。
- 2) S の要素から、 ako スロットの確からしさが最も小さいものを選ぶ。その確からしさを x とし、そのフレームを F_x とする。
- 3) 入力の確からしさを P とする。 $x \geq P$ なら、 S の各要素の ako スロットの確からしさを $-P$ し、終了する。
- 4) S の各要素の ako スロットの確からしさを $-x$ する。
- 5) 確からしさを 0 として F_1 の親フレーム C の子フレームの位置に F_x を移動する。また確からしさを $P-x$ として F_1 を F_2 の子フレームにする。
- 6) A の要素のうち、 F_1 より下位にあるものを消去する。
- 7) F_1 、 F_2 および S の要素のうち F_1 より下位にあるものそれぞれを中心フレームとしてカテゴリ化を行う。

F) 知識ベースでは上下関係のない場合

F_1 、 F_2 とも知識ベース上に存在はしているが、両者が上下関係のない場合も知識ベースと入力とどちらが間違っているかを確からしさの大小で決定する。知識ベースでの F_1 の親フレーム C との間の確からしさを x とすると、 $x \geq P$ のときは知識ベースが正しかったことになる。ただし反例の入力で確からしさが減り $x - P$ となる。

$x < P$ のときは入力が正しかったことになり、 F_2 の子フレームの位置に F_1 を移動させる。その

ときの確からしきは $P - x$ である。その後 F_2 を中心フレームとしてカテゴリ化を行う。

6.6 一般化

A) スロット一般化

同じスロット値が多く出現していれば、次にそのスロット名が現れたときも同じスロット値であるだろうという推論を行うことができる。これをスロット一般化という。

確からしさの高いスロットのファセットを `require` から `default` に変える。スロット一般化を行うための条件は、確からしさを $[m, n]$ とすると、 $m/n > 0.8$ (ただし、 $n > 5$) である。 $n > 5$ という条件がついているのは、かたよった入力 ($m = n = 1$ など) を除去するためである。

B) 必須格一般化

スロット名が「スロット」の場合は、スロット一般化を行わず、必須格一般化を行う。「スロット」スロットの確からしさが高ければ、次にそのフレームが入力されたときもそのスロット名を伴っているであろうという推論ができる。

必須格一般化を行うための確からしさの条件はスロット一般化と同じで、`require` を `default` に変えるかわりに `value` に変える。

C) フレーム一般化

ある意味カテゴリを表すオートフレーム A があり、 A の下位にある語意フレームのうちほとんどのものが同じスロットを持っていれば、それは A 全体に共通するものと推論する。フレーム一般化は、そのスロットを A に記述する処理である。

フレーム一般化を行う条件は、下位の語意フレーム数が5以上で、そのうちの80%以上のフレームが同じスロットを持っていることである。

7 実験と評価

実験データとして、動詞4個(表1)、名詞10個(表2)からなる単語群を用意した。その単語をIPAL[2]の格支配関係と意味素性にしたがって組み合わせた24文を入力例文とする。日本語文の深

表1 実験データ（動詞）の格支配関係

単語	格支配関係
こぼれる	(PRO) が (LOC) に
混む	(PRO/LOC) が
凍る	(CON/LOC) が
憧れる	(HUM) が (ABS) に

表2 実験データ（名詞）の意味素性

意味素性	単語
CON	HUM 彼, 彼女
	ORG 会社, 企業
	PRO 車, 紙
ABS	LOC 公園, 東
	TIM 夕方
	ACT 勉強

層格構造への変換は人間が行った。

まず、入力順序による学習結果の違いを調べるために24文をランダムに入れ換えた実験データを5個作り、それぞれ実行した。次に全集合（24文）のうち30%（7文）50%（12文）80%（19文）をランダムに選んでそれぞれ5個ずつの実験データを作り、どの程度の学習効果が得られるかを調べた。

また、大きな知識ベースを構築する実験として、

2~4個の単語からなる100文を実験データとした。こちらも深層格構造への変換は人間が行った。

7.1 階層化アルゴリズムの検証

24文を入力した実験の結果、構築された知識ベースの階層構造の一つを図7に示す。

全24文を入力して構築された知識ベースの階層は、どれもほぼ同じ形式になった。これは入力の順序にあまり依存していないことを意味する。

入力の順序に依存しないことから、例文をすべて入力してから知識ベースを構築したときと同じ結果が得られることがわかった。

80%、50%の入力による知識ベースの階層構造は、全24文を入力したものとほぼ同じである。30%の場合は未カテゴリー化部分が目立つが、良い例文が得られた場合は完全な階層構造とほとんど変わらないものが生成された。このことから少ない例文から全体を推測することが可能であることがわかった。

7.2 大きな知識ベースの構築

2~4個の単語からなる簡単な文100文からどのような知識ベースが構築されるかを実験した。

例文に含まれる単語の種類は267種で、同じ単語の繰り返しがほとんどない。その結果、階層化によって生成されたオートフレームは、そのほとんどがどのようなスロット名をとるかを制限したものであり、スロット値を制限したものは数例だけであった。また階層の最大の深さ（システムフレ

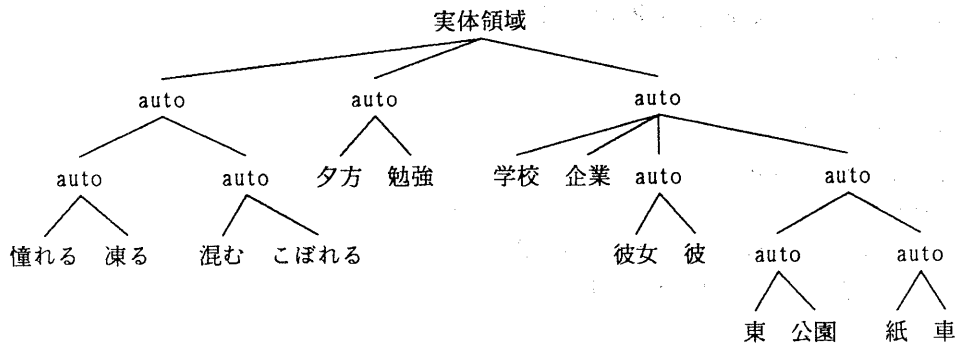


図7 構築された知識ベース

ーム・インスタンスフレームは除く)は4であった。同じ単語がもっと何度も出現すれば、よりしっかりした知識ベースが構築されると思われる。

それでも、似ている単語が同じカテゴリーに属し、品詞の違う単語が一つのカテゴリーにまとめられている例もあり、品詞によらない知識ベースの階層化が可能であることがわかった。

8 今後の課題

本研究では、一つの単語が一つの意味しか持たないと仮定している。しかし現実には、一つの単語が複数の意味を持つことはごく自然なことである。このような場合にも対処しなければならない。

本研究で作成したシステムは、深層格構造を入力としている。日本語文から深層格構造への変換は、FIARYの言語領域にある情報(意味素性・述語素)を用いて行うが、本システムで獲得した知識ベースを用いれば、より確実な解析が可能になるであろう。そのためには解析と学習を同時に行わなければならない、よりヒューリスティックな処理が必要になる。

本研究の問題点として、同じ単語が何度か入力されなければうまく学習ができない点があげられる。学習効果を上げるためにより例文の入力をうながすような工夫も必要である。

9 まとめ

本研究では深層格構造での単語と単語の関係の規則が意味を表しているという立場で、単語の意味を学習し、知識ベースを動的に階層化する方法を設計・実現した。

階層化アルゴリズムとして、

- ・カテゴリー化アルゴリズム
- ・上方組替えアルゴリズム
- ・下方組替えアルゴリズム

を設計した。他に、ako-example スロットが入力されたときの組替えアルゴリズムと、一般化アルゴリズムを設計した。

また実験の結果、本アルゴリズムによって構築

した知識ベースが、入力の順序に依存しないことを確認し、学習アルゴリズムとして妥当であることがわかった。

参考文献

- [1] T. Winograd著, 淵一博, 田村浩一郎, 白井良明訳: 言語理解の構造, 産業図書(1976)
- [2] 情報処理振興事業協会技術センター: 計算機用日本語基本動詞辞書 I P A L (Basic Verbs), 情報処理振興事業協会(1983)
- [3] G. A. Ringland, D. A. Duce edited: Approaches to Knowledge Representation, RESEARCH STUDIES PRESS LTD. (1988)
- [4] 荒木健治, 桃内佳雄: 名詞述語文における意味概念の学習, 情報処理学会NL-79-4(1990)
- [5] 小泉和彦, 大川剛直, 馬場口登, 手塚慶一: クラスフレーム生成によるフレームの階層化, 情報処理学会AI-65-7(1989)
- [6] 白石誠, 納富一宏, 内山明彦: 概念の動的管理システム, 情報処理学会NL-85-9(1991)
- [7] 白石正人, 横田将生: 自然言語概念学習に関する考察, 情報処理学会NL-83-19(1991)
- [8] 仁木和久, 石崎俊: 概念の帰納的学習, 人工知能学会誌 Vol. 3 No. 6(1988)
- [9] 岡田和久, 高田正之, 小谷善行: 知識ベースにおける動詞的概念の学習, 情報処理学会AI-69-7(1990)
- [10] 謝建明, 瀧口伸雄, 小谷善行: 例文からの動詞概念の獲得と評価, 情報処理学会NL-88-9(1992)