

発想支援システム「知恵の泉[®]」

A System for Creativity Support “Chie-no-izumi[®]”

折原 良平*
Ryohei Orihara

* (株)東芝 研究開発センター システム・ソフトウェア生産技術研究所 技術研究部
Research Department, Systems & Software Engineering Laboratory, Research & Development Center, Toshiba Corporation, Kawasaki 210, Japan.

1992年12月7日 受理

Keywords: analogical reasoning, creativity support, cognitive models.

Summary

A system for creativity support “Chie-no-izumi” is presented. Implementing a model of creation based on analogical reasoning, Chie-no-izumi gives users some new concepts which are hints to novel ideas. The framework of analogical reasoning which is employed is “paraphrasing-based analogical reasoning (PA)”. One concept, which is defined in the knowledge base, may be used to define the other concepts. PA generates new concepts by analogically reasoning on such a hierarchic structure of concepts. Domain division, which is postponed in past studies of analogical reasoning, plays an important role in PA. PA can derive several results of analogical reasoning from one knowledge base through different domain divisions.

The prototype system for Chie-no-izumi has an interface to natural language (Japanese) inputs, does domain division for analogical reasoning, and graphically exhibits the process of analogical reasoning. Utilizing the information obtained from kana-kanji conversion, the prototype easily parses several kinds of sentences. Introducing heuristics to select external tokens, domain division is automated. Employing not only visual but also audio aides, a user can understand the process of analogical reasoning without difficulty.

This paper describes what the model of creation is, how the prototype system works, and how it supports human creativity in planning new products.

1. はじめに

著者らは、かねてから発想支援システム「知恵の泉[®]」の開発を通して、計算機による発想支援の方法について検討してきた[楠井 90]。知恵の泉は、類推によって発想をモデル化し、その発想モデルを計算機に実装することで発想機能を実現することをねらったシステムである[折原 90b]。ユーザが知恵の泉と対話することにより、アイデア豊かな人物と対話しているような効果を得られ、その結果ユーザ自身の発想が助長されることを知恵の泉はねらっている。

計算機による発想支援には、

① 非創造的活動を計算機が行うことで、人間を創

造的活動に専念させ、その結果発想支援効果を得る。

② 何らかの発想技法のプロセス(の一部)を機械化する。

③ 計算機自身がアイデア(の原型)を与える、という三つのレベルがあり[Young 88]、知恵の泉は、③をねらった数少ない例の一つ[折原 93]である。

著者らの発想支援の研究は、発想モデルの検討から始まった。その成果として、発想をモデル化するための類推の枠組みである“言換えに基づく類推(Paraphrasing-based Analogical reasoning: PA)”を定義した[Orihara 90a]。次に、実際の発想支援効果を確認するため、プロトタイプを作成し[折原 91]、現実のデータを入力して、どのような支援が行えるかを観察し

た。

本稿では、知恵の泉の発想モデルとプロトタイプシステムについて説明した後、実際に知恵の泉を用いてどのような発想支援が行えたかを紹介する。

2. 「知恵の泉」の発想モデル

2・1 発想のモデル化

知恵の泉では、発想を機械学習の文脈で捉える [Orihara 90a]。すなわち、理想的な知識ベース ι があると仮定し、ある時点で人間や計算機の持っている知識ベース γ がその部分集合である場合に、人間や計算機が ι に関するデータを与えられて、自分の持つ知識ベースを ι へと近づけようとする行動として発想を捉えるのである。

ここで、知識ベースを ι に近づけるためにどんな推論エンジンを用いるかによって、発想を特徴づけることができる。これが、創造工学で発想の四つのパターンとして類推・帰納・普遍化・極限化があげられていることと符合する [伊東 83]。最も重要とされているのが、類推による発想である。類推とは、知識ベース内にいくつかの領域を想定し、基底領域と呼ばれる一般には知識が肥沃な領域から、目標領域と呼ばれる現在注目している部分へと知識を写像し、目標領域における新たな知識を生成することによって行う推論である。

2・2 言換えに基づく類推

[1] ボトムアップな類推

類推は、次の4ステップ [Holland 86] からなる。

1. 基底と目標の内部表現を生成する。
2. 目標の適切な類似物の候補としての基底を選ぶ。
3. 基底と目標の間の対応関係を同定する。
4. 目標における知識を生成するべく写像を拡大する。

類推の機械化に関する研究の歴史は古い [Gentner 83, Haraguchi 85, Haraguchi 86]、それらは上記の1はすでに完了していると仮定し、4ステップのうち2～4のみを扱ったものであった。1は、さらに次の三つのサブステップに分けられると考えられる。

- a. 知識を内部表現に変換する (知識ベースの生成)。
- b. 知識ベース内に領域を設定する。
- c. 目標領域を同定する (他の領域は基底の候補となる)。

る)。

類推を問題解決の文脈で用いるにあたっては、bに關しては自然な領域分割が設定できる*1。また、解くべき問題が与えられるのであるから、cは自明である。加えて、aは類推に限らずAIのアプローチに共通の課題であるから、問題解決における類推研究においてステップ1を扱わないのは妥当な態度といえる。

一方、発想とは、ある驚くべき観察を説明するために仮説を生成することである [Hanson 58] が、発想あるいは企画立案における発想では、“ある驚くべき観察”は実際には存在せず、“こうなってほしい状態(を観察したなら驚くかもしれない)”が漠然と頭の中にあるだけである。問題解決では、与えられた問題を目標として推論が行われるが、“こうなってほしい状態”は、多くの場合抽象的で、推論をガイドする役に立たない。これは、発想支援を必要とする場面では、何が問題かわからないことが多いからである。

したがって、問題解決のようなトップダウンな枠組みにおいて、発想における類推を考えるべきではない。知恵の泉では、これまでの類推研究とは異なり、知識ベース中に領域が設定されていることを仮定していない。その意味で、これまでの研究とは異なり、類推のステップ1を無視していないといえることができる。

[2] 類推の対象

知恵の泉では、概念の定義・被定義関係に基づく階層構造に関して類推を行うことにより、新たな概念を作り出すような発想を扱う。したがって、この発想をモデル化するには概念の定義(言換えと呼ぶ)を対象とした類推機構が必要となる。

[3] 類推による発想をモデル化するためのPA

PAは、以上のような要請を満たす類推機構として定義されたものである。その概略は、以下のとおりである [Orihara 90a]。

- PAは、 $\forall x_1, \dots, x_n (p(x_1, \dots, x_n) \Leftrightarrow L_1 \wedge \dots \wedge L_m)$ という形の知識 (p の言換え) を対象とする。このとき、 \Leftrightarrow の右辺を言換えの本体という。
- 定数、関数、述語記号をまとめてトークンと呼び、それらが外部・内部的トークンに分類される。
- 言換えの集合が知識ベースとして与えられ、複数の領域にまたがって現れるトークンはすべて外部的であるように、領域分割が行われる。
- トークンとトークンの同値類のリストをT-同値と呼ぶ。初期的には、T-同値はすべての外部的トークンをそれ自身に同値であるとする。異なる領域に属する二つの言換えが、あるT-同値によって同値とされる少なくとも一つずつのトークンを

*1 その知識で解ける問題に関する領域分割。

含み、かつ、その T-同値に現れないトークンとトークンの組一つ以上を同値と考えれば二つの言換えが等しくなるとき、類似であるという。このとき、T-同値はそのトークンの組を同値とするように拡大される。領域が、他の領域に属する言換えと類似な言換えを含むなら、それらの領域は類似である。

- 領域 S の言換の本体が、T-同値によってそれと同値な他の領域 O の式とすることができ、その言換の頭部(同値記号の左辺) H はそうできないとき、領域 O において H に当たるアトムをユーザから得て新たな言換えを作り、O と T-同値を拡大する。これを言換えに基づく類推と呼ぶ。

3. 「知恵の泉」プロトタイプ

3.1 プロトタイプ設計にあたって

著者らは、入力として言換の集合と外部的トークンの集合を与えられ、ユーザと対話を行いながら PA を行うシステム(PA システム)をワークステーション上に実現した[Orihara 90a]。しかし、PA システムを実際に発想作業に用いるには、(1)入力が困難、(2)領域分割が自動化されていない、(3)類推過程が理解しにくい、という問題点があった。そこで、次の方針でプロトタイプを設計した。

1. 入力形式は、自然言語(日本語)とする。対話的な処理を用いて、類推システムのサブシステムとし

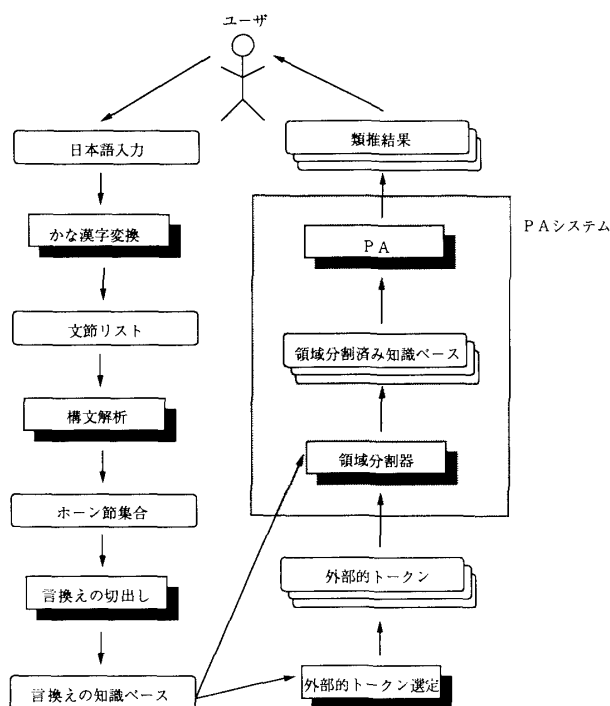


図1 プロトタイプの構成図

てできるだけ手軽な方法で実現することをねらう。

2. 与えられた知識ベースにヒューリスティックを適用して数種類の外部的トークン集合を生成し、領域分割を自動化する。
3. 類推の進行状態を表示するグラフィカルなウィンドウインタフェースを用意する。

図1に、知恵の泉プロトタイプの構成図を示す。以下の節では、各部分の実現について順番に述べる。

3.2 プロトタイプの入力系

PA では、領域間の類似性の検出には意味情報を用いないので、入力インタフェースに意味解析は不要であり、入力された文を構文解析して Prolog の節形式に変換できればよい。このようなニーズに応えるものとしては、井上の方法[有川 90, 井上 88]がある。

(1) 井上の方法

井上の方法は、文節に区切られた日本語のリストを入力とし、対応するホーン節を出力する。日本語は、

- ・主語、目的語、述語という並びをしている。
- ・主語や目的語の語尾には格を表す助詞がつく。

という特徴を持っているので、ホーン節の3形態(事実、規則、質問)にあたる構文規則を用意しておくことにより、辞書なしで変換を行うことができる。

この方法で問題になるのは、引数の並びが異なるために意味の等しいアトムが単一化不可能とされてしまう場合である。井上の方法では、推論中に動的に引数を加えたり、引数位置を変えることによってこの問題に対処している。しかし、これでは推論系と入力系の結合が密になり、推論系のなかで入力系に依存する部分が多くなってしまふ。今回は、既存の PA システムをできるだけそのまま使いたないので、すべてのアトムを

述語(引数リスト、助詞リスト)

という形に正規化し、単一化の機構を拡張した。

このほか、関数構造(「～の…」)や並列構造(「～と…」)を認識できない、連体修飾語を扱えないという井上の方法の欠点を改良した。連体修飾語に関しては、仮想的な助詞“連体修飾”を持つ、と考えている。

井上の方法に以上の拡張を施したものを、jparse システムと呼ぶ。

(2) 字句解析について

分かち書きが行われる欧文と異なり、日本語処理における字句解析は難しい部分である。しかし、日本語では付属語の種類が限られているため、文節を切り出せば、そこから単語を取り出すのは容易である。

観点を試してみよう。計算機に日本語を扱わせるためには、何らかの形で計算機に日本語を入力しなければならない。今日、そのために使われるのは多くの場合“連文節かな漢字変換”である。連文節かな漢字変換では入力時に文節が意識されるので、この文節情報をそのまま保存すれば簡単に字句解析を行える。

そこで、このアイデアを nemacs エディタとかな漢字変換システム Wnn, および両者のインタフェースである「たまご」システムを用いて実現した。

〔3〕 言換えの切出し

jparse システムは、言換えを「～である」などの述語を持った事実として認識する。したがって、jparse システムの出力からしかるべき言換えを切り出す機構が必要である。これが paraph システムである。

次の 6 通りの事実が言換えとして認識される。

1. 動詞 + タイプ + である or だ or です ([上層名詞, …, 下層名詞, …], [は or が or とは, …, を, …])
2. タイプ + である or だ or です ([上層名詞, …, 下層名詞, …, 動詞], [は or が or とは, …, を, …, 連体修飾])
3. 上層名詞 + である or だ or です ([下層名詞, …, 動詞 + タイプ], [を, …, は or が or とは])
4. 上層名詞 + である or だ or です ([下層名詞, …, 動詞, タイプ], [を, …, 連体修飾, は or が or とは])
5. 上層名詞 + という or という or と呼ぶ ([下層名詞, …, 動詞 + タイプ], [を, …, を])
6. 上層名詞 + という or という or と呼ぶ ([下層名詞, …, 動詞, タイプ], [を, …, 連体修飾, を])

ここで、上層名詞とは、言い換えられている概念で、言換えの頭部の述語である。タイプとは、上層名詞の、is-a 概念階層における上位概念を表す。下層名詞とは、タイプのほかに、上層名詞を定義するのに用いられる概念である。これらは、次の形に変換される。

上層名詞 ($[X]$, $[izIsa]$) \Leftrightarrow タイプ ($[X]$, $[izIsa]$),
 下層名詞 ($[izSkolem(X)]$, $[izIsa]$), …,
 動詞 ($[izSkolem(X)]$, …, 他の引数, X),
 $[izBefore, …, \text{他の引数の助詞}, izAfter]$).

…は修飾語に対応する。 $izSkolem$ はスコレム関数を表す。 $izIsa$ は is-a 関係を表す仮想的な助詞である。 $izBefore$ $izAfter$ はそれぞれ下層名詞, 上層名詞を表す仮想的な助詞である。

〔例 3・1〕 言換えと認識される事実

煮た料理である ([シチュー, 肉, 鍋], [は, を,

で])

シチューである ([肉, 鍋, 煮た料理], [を, で, は])

□
 言換えを認識すると, paraph システムはタイプと上層名詞の間の is-a 概念階層における親子関係を記録する。「こと」と「もの」は組込みのタイプとしてあらかじめ登録されている。上位概念がわからない場合、「もの」の子供であると仮定する。

関数形 $b(a)$ が引数に含まれている場合には, paraph システムはこれを「 a の b 」に展開し、同時に

$$a \text{ の } b([X], [izIsa]) \Leftrightarrow a([izSkolem(X)], [izIsa]),$$

$$b([izSkolem(X), X], [izBefore, izAfter]).$$

なる言換えを生成する。

3・3 外部的トークンの自動選定

一つの言換への集合に対して、異なる外部的トークン集合を与えることにより、さまざまな類推結果を得ることができるのが PA の特徴である。しかし、どのような外部的トークンを与えるのが有効かに関する指針はなく、この特徴を生かすことは難しかった。

PA における外部的トークンとは、複数の領域に現れることのできるトークンであるが、初期的には一つの領域にしか現れていなくても、新しい言換えが類推された結果、複数の領域に現れることになる場合もある。したがって、ある領域分割を与える外部的トークンの集合は一意でない。

外部的トークンは、それが存在しない領域に対しても無条件に写像することができるので、より多くの類推を行うためには外部的トークンが多いほうが有効である。そこで、次のような外部的トークン自動選定の機構を考案した。

ステップ 1. ヒューリスティックによって仮の外部的トークン集合(外部的トークンの種と呼ぶ)を選定し、それに基づいて領域分割を行う。

ステップ 2. そのようにして作られた領域分割を与える外部的トークン集合のうち、集合の包含関係において極大のものを外部的トークンとして選ぶ。これは、外部トークンの種に対して内部的トークン一つずつ加えていくことによる。

言換えが他の言換えと、高々一つのトークンしか共有しない場合には次の定理が成り立つので、この方法によりある標準領域分割を与える最大の外部的トークン集合を得ることができる。個々の言換えが比較的簡単で、知識ベースが多くのトークンを用い

て表現されているような場合には、この条件が満たされていることが期待できる。

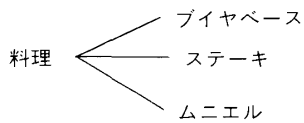


図2 概念階層

[定理 3・2] 同じ標準領域分割(可能な最も細かい領域分割)[Orihara 90a]を与える外部的トークン集合の族は、集合の包含関係に関して束をなす。 ■

ヒュースティックは、次の三つである。

H1. 動詞、連体修飾語、道具格の語は外部的トークン。言換えの本体の最後のアトムの述語を動詞、「で」格*2の引数を道具格の語と考える。連体修飾語と

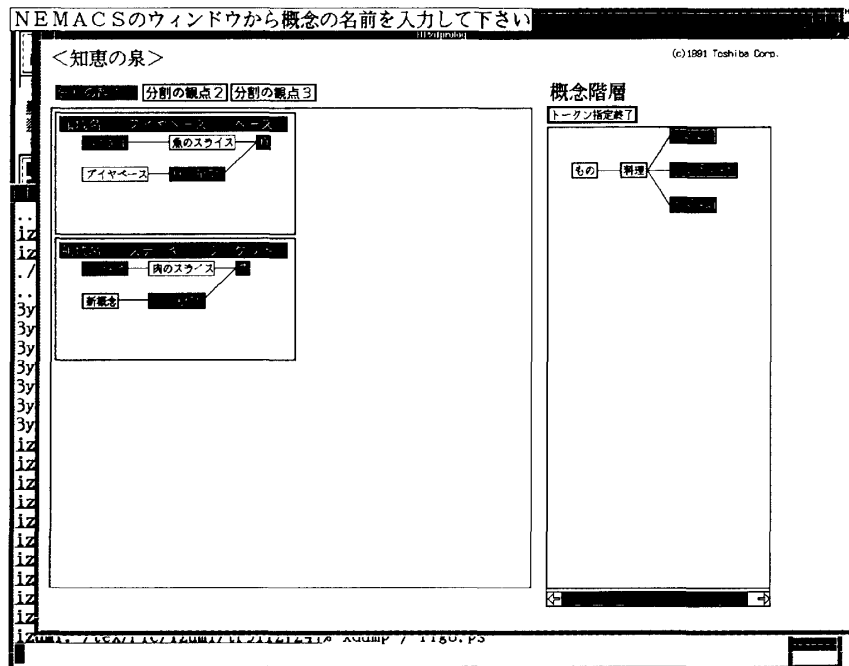


図3 シチュエを類推

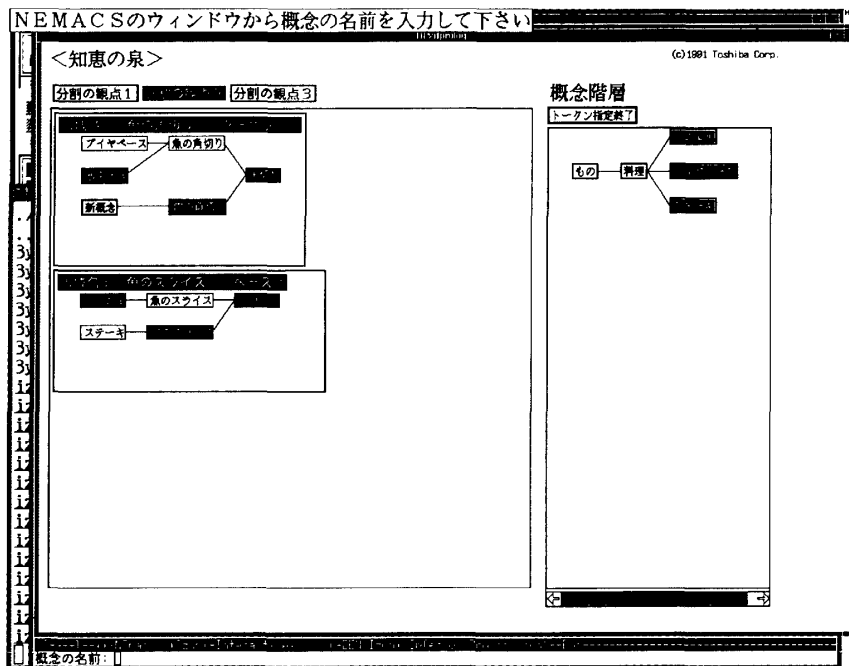


図4 サイコロステーキを類推

* 2 対応する助詞として X を持つ引数のことを“X 格の引数”と呼ぶ。

は、連体修飾格の引数を指す。

H2. izBefore/izAfter 階層の最下位は外部的。

H3. 概念の is-a 階層のユーザの指定した下位部分構造が内部的トークンを形成。

分割のために用いられる外部的トークンの種は

Ext1. H1 によるものと H3 によるものの合成。

Ext2. H2 によるものと H3 によるものの合成。

Ext3. H1, H2, H3 によるものの合成。

の 3 種類である。H3 による外部的トークンの種はユーザの知識を反映しているので、常に用いている。

[例 3・3] 外部的トークン自動選定

次の 3 文が入力されたとする。

(1) “ステーキとは、肉のスライスをフライパンで焼いた料理である”，(2) “ブイヤベースとは、魚の角切りを鍋でゆでた料理である”，(3) “ムニエルとは、魚のスライスをフライパンで焼いた料理である”。

H1, H2, H3 はそれぞれ次のような外部的トークンの種を与える。

(H1) [角切り, スライス, ゆでた, 焼く, フライパン, 銅]^{*3}

(H2) [肉, 魚]

(H3) [料理]

ただし、H3 に関しては、この例で形成される概念階層(図 2)において、ユーザが領域を特徴づけるトークンとして“ブイヤベース”，“ステーキ”，“ムニエル”を指定した場合である。

各ヒューリスティックによる領域分割は次のとおり。

(Ext1) : {(1)}, {(2)(3)}

(Ext2) : {(1)(2)}, {(3)}

(Ext3) : {(1)}, {(2)}, {(3)}

Ext2 による分割では、ステップ 2 によって外部的トークンが拡張され、

[鍋, ゆでた, フライパン, 焼く, 魚, 肉, 料理]を得る。 □

3・4 出力系の設計

[1] 概念の is-a 階層の表示

ヒューリスティック H3 において、領域を特徴づけるトークンをユーザに指定させるには、概念の階層構造をウィンドウに表示し、そのなかでユーザにマウスで指定してもらう。図 3, 図 4 の右側のウィンドウが概念の階層構造を示している。

* 3 「焼いた」は「焼く」に変換される。

* 4 図中では“西京焼き”。

ユーザは、領域を特徴づけられるトークンのラベルの上でマウスをクリックする。すると、表示が反転する。図 3, 図 4 では、“ムニエル”，“ブイヤベース”，“ステーキ”がクリックされている。このように、is-a 階層木の葉および子孫が姉妹を持たない節が選択されている状態がデフォルトである。

[2] 領域構造の表示

各領域は、内部的トークンのグラフで表される。グラフの意味は、線で結ばれた左側のトークンが、右側のトークンを用いて言い換えられるという、言換え・被言換えの階層である。

[3] 類推過程の表示

PA システムでは、知識ベースが大きくなってくると、どのように類推が進んでいるのかを把握することが困難であった。そこで、領域間で対応づけられているトークンを同じ色で示すことにした。新しい概念を類推したときには、“新概念”という仮のトークンを導入し、それを新概念の類推元となったトークンとともに点滅させる。また、オーディオデバイスを用いてドラの音を出すことでユーザの注意を喚起する。

図 3 は、前節で用いた例を入力した後、Ext 1 による分割において、ブイヤベースからの類推で肉の角切りを煮込んだ料理を類推し、概念の名前の入力をユーザに求めているところである。白黒のため図では判別できないが、“肉の角切り”と“魚の角切り”が同じ色で表示され、対応づけられていることを示している。一方、図 4 は、Ext 2 による分割において、ムニエルからの類推で魚の角切りを焼いた料理^{*4}を類推した後、ステーキからの類推で肉の角切りを焼いた料理を類推したところである。

4. 知恵の泉による発想支援

これまでの、簡単な例を用いて知恵の泉の発想モデルとプロトタイプの動きを説明してきたが、本章では、実際の知恵の泉を発想支援に用いた例を示す。

4・1 知恵の泉の運用方法

知恵の泉を発想支援に用いるにあたっては、2 通りの運用方法が考えられる。

1. 広範な知識をあらかじめ与えておき、ユーザは自分に身近な知識を入力して領域分割と類推を行わせる。この方法では、まったくかけ離れた領域からの類推による極めて斬新なアイディアの提案が期待できる反面、アイディアの提案数は多くないと予想される。なぜなら、関連の少ない領域どう

しは外部的トークンを共有する可能性が低いからである。PA の定義により、外部的トークンを共有しない領域間では、言換えに基づく類推は起こらない。外部的トークンの同値性に関する理論を、同義語辞書の形で与えることにより、この問題を回避することは可能である。

2. 分野を限定し、その分野に関する概観的な知識を与えて、領域分割と類推を行わせる。この方法では、当該分野を領域分割によってさらに細かい“副分野”に分割し、副分野間で類推を行うことになる。分野を限れば、同義語辞書を用意することも容易になり、多くのアイデアの提案が期待できる。しかし、同じ分野内での類推に過ぎないため、斬新なアイデアが提案される可能性は低い。なお、この方法では、与えた知識をもとにボトムアップ的にアイデア提案を待ってもよいし、ユーザが知識をさらに与えたうえで類推を行わせてもよい。

今回は、準備の容易さから、上記2の運用方法を試すことにした。分野としては、マルチメディアシステムを選んだ。これは、著者らに入手可能なところに、サーベイ的なデータベースが計算機可読なメディアの形で用意されていた[野村総研 91]ためである。この実験は、知恵の泉の支援のもとに、新しいマルチメディアシステムの企画を行うことにあたる。

なお、今回上記2の方法を試したのは、1のアプローチを不可能と考えたからではない。最近の大規模な同義語辞書がCD-ROMなどにより計算機可読な形で入手可能となっており、1の方法を実現することは今後の課題である*5。

〔1〕 知識の入力

前述したデータベースは871件の事例を収録しているが、このなかには、標準化に関する委員会活動や、企業の設備投資などに関するものも数多く含まれている。そのなかから、システムやソフトウェアに関するものを抽出した。知恵の泉の入力インタフェースは、かな漢字変換時の情報を用いて字句解析を行うシステムであるため、今回の実験のようにすでに電子的な形で存在するテキストに対してはそのまま用いることはできない。そこで、入手により、各文を文節のリストの形に変換し、これを構文解析器に入力する方法をとった。この際、「知恵の泉」の構文解析器が解析できない重文および複文は、複数の単文に読み換えた。

* 5 ICOT ALT-WGでの議論および匿名の査読者の示唆による。

* 6 Ext 3による分割に関しては未試行である。

データベース中では、同じ意味を表すのに多くの異なった語が用いられていたため、同義語辞書を用意した。同義語辞書は、二つの語の同義性を記述した関係の集合である。

〔2〕 実験の規模

- ハードウェア：東芝 AS 4075 (Sun SparcStation 2)
- 入力 DB 項目数：117 件
- 入力文数：439 文
- 言換えの数：565 個 (入力文から 441 個、関数形に対して 124 個自動生成)
- 同義語辞書の大きさ：同義関係 106 組
- トークンの総数：1 211 個
- 外部的トークンの数：Ext 1 で 823 個、Ext 2 で 464 個、Ext 3 で 1 027 個
- 領域分割
 - is-a 階層中のトークンの選択：デフォルト
 - 所要時間：約 3 時間
 - 得られた領域の数：Ext 1 で 128 個、Ext 2 で 74 個、Ext 3 で 162 個
- 類推開始後 20 時間で得られた新概念：Ext 1 で 22 個、Ext 2 で 9 個*6

〔3〕 類推結果の例

Ext 1 による分割において、類推開始後 20 時間で得られた新概念のなかから興味深いものを三つ紹介する。

- **マルチメディア会社情報**：基底領域は、工場の製造工程を収録し、さまざまな観点から製造工程を分析することで、改善提案を行いやすくする“製造工程収録ソフト”に関する領域で、目標領域は、全米の企業の住所を収録したソフトである“The American Business Disk”，およびストラビンスキーの「春の祭典」を解説とともに収録したマルチメディア CD-ROM ソフト“The - Rite - of - Spring”に関する領域である。製造工程収録ソフトは、その映像から映像へと自由に移動できる機能を持っている。知恵の泉は、“製造工程収録ソフトの映像内”からの類推で“The American Business Disk の映像内”なる概念を作り、The American Business Disk に映像を付加することを示唆したうえで、その映像間を自由に移動できる機能を持たせることを提案した。この概念には、例えば**マルチメディア会社情報**という名前がふさわしい。
- **会社分析システム**：さらに同じ基底・目標の組で類推を続けていくと、知恵の泉は、映像間移動機能を用いて製造工程を分析するためのソフトであ

る“製造工程分析用ソフト”からの類推で、“マルチメディア会社情報”を用いて分析を行うシステムを提案した。この概念には、例えば、**会社分析システム**という名前がふさわしい。

- **VR ハイパーテキスト**：基底領域は、飛行機設計システム、戦闘シミュレータなど、いくつかの仮想現実感(VR)システムに関する領域であり、目標領域は、プロトタイピングに適したハイパーテキストシステム“リンクハイパーテキストシステム”に関する領域である。VRによる飛行機設計システムは、仮想空間で飛行機に乗ることができる機能を持っている。知恵の泉は、“仮想空間飛行機のなか”と“リンクハイパーテキストシステムのなか”の類似性を認識することで“仮想空間飛行機”と“リンクハイパーテキストシステム”を対応づけ、その結果リンクハイパーテキストシステムのなかに入ってしまうような機能を提案した。この概念には、例えば **VR ハイパーテキスト**という名前がふさわしい。

4・2 実験の評価と考察

ここでは、知恵の泉の発想支援効果について定性的な分析を行う。定量的な評価では、支援効果をユーザの個性と独立に評価しなければならない。それには複数のユーザを被験者とした実験が必要であり、今後の課題と考えている。

〔1〕 支援モデルの妥当性

知恵の泉では、類似性の発見には意味情報を用いていないので、知恵の泉の提案する概念それ自体は、既

存の言葉の組合せに基づいて新たな組合せを導くという比較的皮相なものである。しかし、上に示した例からもわかるとおり、そうした新規な言葉の組合せがユーザの発想を刺激し、それによりユーザの頭のなかに豊かなイメージが形成され得る。これは、他者との対話と同様な効果により発想支援を行うという知恵の泉のモデルが妥当であったことを示している。

このモデルにおいては、類推結果そのものが有益であるかではなく、それがいかにユーザの発想を刺激し得るかが重要である。類推結果は、基底領域から目標領域に知識を持ち込む方法の例として用いられている場合が多い。例えば、前述の“The American Business Disk”の映像内では、知恵の泉は単に“The American Business Disk”に映像を付加することを示唆しているにすぎない。しかし、その類推の基底である“製造工程収録ソフトの映像内”に関連してユーザが「製造工程のさまざまな側面を収録しているソフトである」という背景知識を持っているため、このイメージも“The American Business Disk”へと投射され、“マルチメディア会社情報”というアイディアにつながっていったのである。

これを、模式的に表したのが図5である。知恵の泉は、与えられた比較的少量の知識の上で類推を行っている(図中s, t)。ユーザは、知恵の泉に与えられた知識を含む大量の背景知識(図中S, T)を持っており、知恵の泉の行った類推をヒントとして、背景知識の上で類推を行い、アイディアを得る。本実験のように、知恵の泉に与えられた知識をユーザが事前には知らない場合でも、ユーザは知恵の泉との対話中にそれを知り、その知識はユーザの持っている背景知識と結合されるので、図5に示す図式はそのまま適用できる。

したがって、基底領域と目標領域が十分容易に理解できるものであれば、類推を行うまでもなく、類似性の検出された領域の組を提示するだけで発想支援効果が得られると考えられる。これは、本実験において、領域群のなかから基底-目標領域として扱っている領域を取り出して表示する機能を付加することにより、直観的には確認された。領域の粒度が大きいなどの理由により、ユーザが領域および領域間を理解することが困難な場合には、知識の持込み方の例としての類推結果以上の情報を与えることが発想支援には必要となるであろう。

他方、柔軟な類推システムの設計という立場からは、図5に示す図式全体を類推システムに取り込むことが考えられる。すなわち、大量の知識上で類推を行う代わりに、その一部のみを取り出して領域間の類似性を

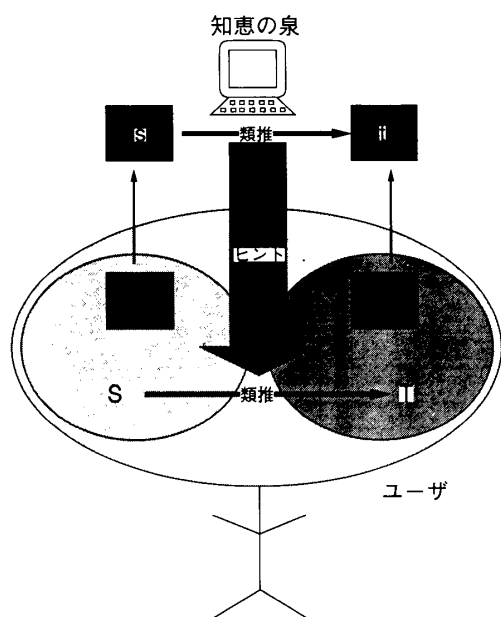


図5 知恵の泉による発想支援の図式

判定し、それをヒントとしてもとの知識上での類推を実現するのである。具体的な方法に関しては未検討だが、例えば theory abstraction の枠組み [Tenenber 89, Okubo 92] が参考になると思われる。

〔2〕 所要時間

本実験では、いくつかの興味深い概念の提案を得ることができたが、処理時間の遅さは実用化への大きな壁といえる。現在、知恵の泉との対話にユーザが拘束されることのないよう、「～の…」という形の概念から類推された概念には自動的に命名し、ユーザによる命名を必要とする新たな概念を発見したときには、電子メールを送る機能を付加している。

処理の遅さは、大量の領域から順に2個ずつ取り出して、それらの間を比較するというボトムアップな処理の流れに起因している。本実験の Ext 1 による分割では、8128通りの比較を行うことになる。ここで、何らかの情報によりユーザの注目している領域と、そうでない領域とを区別することができれば、AM [Lenat 82] の agenda メカニズムに類似した仕組みを用いることにより、しらみつぶしのボトムアップ類似性検出を避けることができ、高速化が期待できる。しかし、たくさん領域に対して注目度を割り当てることは、労力および領域の理解の点でユーザにとって困難な作業であり、この方法を導入するには工夫が必要である。さらに、思ってもみなかった領域間の類似性を知恵の泉が与えるところに発想支援効果がある以上、ユーザの注目していないところを無視してしまつては、支援効果が得られない可能性もある。

他方、ある基底-目標領域の組に関する計算と、他の組に関する計算は独立に行えるので、並列計算機を用いることにより劇的な高速化が期待できる。

〔3〕 類推に貢献した構造

上に示した例も含めて、「AのB」という関数構造に基づく類推が多く観察された。Ext 1 による分割においては22個中7個、Ext 2 による分割においては9個中4個である。これには、AとA'の類似性を与えられて、「AのB」からの類推で「A'のB」、すなわちA'の新たな機能を導く場合と、「AのB」と「A'のB」とを与えられて、同じ機能を持つものとしてAとA'の類

似性を導く場合とがある。“マルチメディア会社情報”は前者の、“VRハイパーテキスト”は後者の例である。

5. おわりに

類推による発想をモデル化するための類推メカニズム PA を定義し、これに日本語入力インタフェース、外部的トークン選定機構、ウィンドウインタフェースを付け加えることで発想支援システム「知恵の泉」のプロトタイプを作成した。

プロトタイプを用いて、マルチメディアシステムの企画を行う実験を行い、興味深いアイデアの提案を得た。これにより、知恵の泉を分野を限って用いる運用方法の有用性が実験的に示された。本稿では述べなかつたが、知恵の泉の特許明細書のブラッシュアップに用いる実験も行い、良好な結果を得ている。

最近、類推に関する研究が盛んであり、類推による問題解決も各種提案されているが、「正しい類推とはどのようなものか」がはっきりしていない現在では、実世界の問題解決には適用しにくい。これに対し、本研究では企画作業の支援を適用対象としており、類推システムの提示したアイデアを人間が取捨選択するという立場を取っているため、現時点での類推技術でも安心して使えるというメリットがある。

今後の課題としては、前章で述べたもののほかに、領域分割法の高度化がある。知恵の泉の発想支援効果は新奇な類比の発見によるが、それは領域分割の方法に大きく依存している。プロトタイプで用いたヒューリスティックによる外部的トークン選定は効率的な方法といえるが、高度な応用のためには、より柔軟な領域分割方法を検討する必要がある。

謝 辞

的確な指摘をして下さった査読者の方、データを提供して下さった(株)東芝 コンセプトエンジニアリング開発部の方々、および日頃より御指導いただく(株)東芝 研究開発センターの渡辺貞一システム・ソフトウェア生技術研究所長、新井政彦部長に感謝する。

◇ 参 考 文 献 ◇

- [有川 90] 有川節夫：類推を用いる知識処理システムの開発、技術報告 課題番号 62880008、平成元年度文部省科学研究費補助金(試験研究(2))研究成果報告書(1990)。
[Gentner 83] Gentner, D.: Structure-Mapping: A Theo-

- retical Framework for Analogy, *COGNITIVE SCIENCE*, Vol. 7, pp. 155-170(1983).
[Hanson 58] Hanson, R. N.: *Patterns of Discovery*, Cambridge (1958).

- [Haraguchi 85] Haraguchi, M.: Towards a Mathematical Theory of Analogy, *Bull. of Inform. Cybernetics*, Vol. 21, pp. 29-56(1985).
- [Haraguchi 86] Haraguchi, M.: Analogical Reasoning using Transformation of Rules, *Proc. LPC'85, Springer-Verlag LNCS 221*, pp. 56-65(1986)
- [Holland 86] Holland, J. H., *et al.*: *INDUCTION*, The MIT Press(1986).
- [井上 88] 井上 仁, 有川節夫: 類推のための日本語インタフェースについて, 第 2 回人工知能学会全大, pp. 509-512(1988).
- [伊東 83] 伊東俊太郎: 科学における創造性, 日本創造性学会編, 創造の理論と方法, pp. 73-84, 共立出版(1983).
- [楠井 90] 楠井洋一, ほか: 発想支援システム「知恵の泉」, 東芝レビュー, Vol. 45, pp. 413-416(1990).
- [Lenat 82] Lenat, D. B.: AM: Discovery in Mathematics as Heuristic Search, In Davis, R. and Lenat, D. B.(eds.): *KNOWLEDGE-BASED SYSTEMS IN ARTIFICIAL INTELLIGENCE*, chapt. 1, pp. 1-225, McGraw-Hill(1982).
- [野村総研 91] 野村総合研究所: 感性メディアテクノロジーの市場展望と技術開発戦略(1991).
- [Okubo 92] Okubo, Y. and Haraguchi, M.: PLANNING WITH ABSTRACTION BASED ON PARTIAL PREDICATE MAPPINGS, *Proc. ALT'92*, pp. 183-194(1992).
- [Orihara 90a] Orihara, R., *et al.*: On Paraphrasing-based Analogical Reasoning—as a theoretical base of the abduction support system, *Proc. ALT'90*, pp. 134-148(1990).
- [折原 90b] 折原良平, ほか: 発想支援システム「知恵の泉TM」の類推機構と類推に基づく発想の意味, 第 4 回人工知能学会全大, pp. 123-126(1990)
- [折原 91] 折原良平, 今野 宏: 発想支援システム「知恵の泉TM」の試作, 日本ソフトウェア科学会第 8 回大会論文集, pp. 5-8(1991).
- [折原 93] 折原良平: 発想支援システムの動向, 情報処理, Vol. 34, No. 1, pp. 81-87(1993).
- [Tenenbergs 89] Tenenbergs, J. D.: Inheritance in Automated Planning, *Proc. 1st Int. Conf. on Knowledge Representation and Reasoning*, pp. 475-485(1989).
- [Young 88] Young, L. F.: Idea Processing Support: Definitions and Concepts, *Decision Support and Idea Processing Systems*, chapt. 8, pp. 243-268, Wm. C. Brown Pub.(1988).

[担当編集委員・査読者: 原口 誠, 篠原靖志]

◇ 付 録 ◇

A. 定理 3・2 の証明

集合の族は集合の包含関係に関して半順序集合であるから, 最大元と最小元が存在することを示せばよい.

最大元の存在を示す. 外部的トークン集合 E が標準領域分割 D を与えるとする. [Orihara 90a] の定理 4・2 よりこれは一意である. $e \notin E$ に対し, $E \cup e$ の与える標準領域分割が D と異なる場合は, e が, D においてある領域に属する複数の言換えが共有する

トークンである場合に限られる. トークン $e_1, e_2 \notin E$ に対し, $E \cup \{e_1\}$ と $E \cup \{e_2\}$ がともに標準領域分割 D を与えるならば, 仮定により, ある言換えが e_1 と e_2 の両方を他の言換えと共有することはないので, $E \cup \{e_1\} \cup \{e_2\}$ も標準領域分割 D を与える. よって, 同じ標準領域分割を与える外部的トークン集合の族には最大元が存在する.

最小元に関しても, 双対な議論により存在を示せる.

著 者 紹 介



折原 良平(正会員)

1986 年筑波大学第三学群情報学類卒業. 1988 年同大学院工学研究科電子・情報工学専攻博士前期課程修了. 同年, (株)東芝入社. 現在, 同社研究開発センター システム・ソフトウェア生産技術研究所に勤務. 1993~95 年にかけて University of Toronto 客員研究員. 発想支援技術, 類推, 帰納推論の研究に従事. 情報処理学会, 日本ソフトウェア科学会各会員.