

## プリミティブマッチング法：隠喩理解のための計算モデル

諏訪 正樹、元田 浩  
(株)日立製作所基礎研究所

隠喩文は、抽象概念の物理的イメージに関して人間が持っている知識を体現するものである。本論文は、隠喩文を解析することによりその知識を抽出し、人間の心的活動を計算機上で視覚化するための基礎的検討を行なうことを目的にしている。筆者らは、隠喩文が内包する類比的対応関係を発見する計算モデルとして、構造的同型性制約と、プリミティブマッチングという意味的制約を併用する手法、プリミティブマッチング法（PM法）を提案した。本論文では、実際の隠喩文をPM法で解析した結果から、その類比決定の性能を従来法と比較し、更に、使用した隠喩文が内包している「抽象概念の物理的イメージ」を例証する。

### **Primitive Matching Approach: A Calculation Model for Finding Analogical Correspondences in Metaphorical Sentences, - Preliminary Experimental Results -**

Masaki Suwa      Hiroshi Motoda  
Advanced Research Laboratory, Hitachi Ltd.  
Hatoyama, Hiki-gun, Saitama Pref. 350-03, Japan.  
PHONE: 81-492-96-6111, FAX: 81-492-96-6006  
suwa@harl.hitachi.co.jp, motoda@harl.hitachi.co.jp

Metaphor stands on metaphorical mappings between abstract and physical concepts, i.e. *knowledge about physical images of concepts in the mental world*. Our goal is to come up with some insights on visualization of human mental behaviors by computationally acquiring those pieces of knowledge from the metaphorical sentences we human beings use in everyday life. We proposed a method, called *primitive-matching approach*, as a model for detecting metaphorical mappings of sentences, which employs both structure-mapping theory and a semantic constraint of primitive-matchings. In this report, we made an experiment of processing 17 sentences by PM approach, from which we compared the performance of PM approach to structure-mapping approach and also we've come up with some knowledge about physical images of abstract concepts.

## 1 はじめに

ある概念（ターゲット概念）を別の概念（ソース概念）で見立てることによって前者のある特徴を際立たせるという隠喩的手法は、我々人間が考え又は表現する上で中心的役割を果たしている [2] [6]。Barnden は、我々人間は心的概念を物理的概念に例えること（つまり、隠喩的表現を使うこと）なしには心の状態を十分に表現できないと主張している [1]。彼の主張によれば、隠喩文は、心的概念を物理的概念に見立てる見立て知識、即ち人間固有の「心的概念の物理的イメージ」の宝庫であるといえる。本論文の目的は、隠喩文を計算機で処理することによって「心的概念の物理的イメージ」を抽出し、その知識を用いて人間の心的活動を視覚化する方法論を確立するための糸口を発見することにある。

隠喩理解のための計算モデルに関する研究は、自然言語処理的アプローチと類推的アプローチに大別できる [7]。自然言語的アプローチでは、ソース概念が選択制限違反を起こした時に、ソース概念と「意味的に最もマッチ」し選択制限を犯さない様なターゲット概念を発見し、そのターゲット概念でソース概念を置換することにより隠喩文を処理する。概念間の意味的類似性を判定する明示的な知識を駆使して「意味的にマッチする」概念を発見する [3] [10] [8] 点に特徴がある。一方、類推的アプローチでは、概念の意味的類似性に関する知識を用意するのは困難であることから、概念の意味を構造として表現した上で、構造間の類比（analogical mapping）をもって両概念間の類似性を判定する。ソース概念にあつてターゲット概念にない構造を後者に転写することにより、わざわざ隠喩的表現を使用した発話者の意図が明らかになる [4] [2] 点が、類推的アプローチの特徴点である。

本論文では、類比が、心的概念と物理的概念の隠喩的結び付きを表現するために適当であると考え、類推的アプローチにより隠喩文処理を行なう。類推的アプローチに属する研究の大半は、2つの意味構造間の類比を決定する際に如何なる制約（又は知識）を使うかに関するものである。Holyoak の分析 [5] によると、それらの制約は次の3種類に大別できる。構造的同型性制約（structural consistency）、意味的類似性に関する知識（semantic similarity）、運用的制約（pragmatic centrality）である。諏訪らは、隠喩文処理におけるこれらの制約の適用性に関して検討し、以下の洞察を得た [9]。概念木（abstraction hierarchy）などの意味的類似性知識だけで隠喩文に埋め込まれている隠喩的結び付きを発見することは不可能であり、基本的には構造的同型性制約を用いるのがよいが、それのみでは類比の発見に多大の計算時間がかかって好ましくない。また、自然言語文の意味決定が話者の意図や（話す）目的などの運用的制約（文脈）に左右されることは間違いない事実で隠喩文もその例外ではないが、隠喩文の場合、発話者の意図とは、通常ではターゲット概念からは連想しない性質をソース概念を使うことによって顕現化することであり、それは隠喩文理解の結果として発見すべきことであつて処理過程の知識として予め与えておくことは好ましくない。これらの検討の結果、諏訪らは、構造的同型性制約及び意味的制約の両者を併用する手法、プリミティブマッチング法（PM法）を提案している [9]。2章でPM法に関して概説する。

本論文では、PM法を隠喩理解の計算モデルとしてインプリメントし、17個の隠喩文を例にして、PM法および構造的同型性制約のみを使った手法（SM法）の性能を比較するための基礎実験を行なった。3章で、計算モデルのアルゴリズムを詳説する。4.1章でPM法の類比決定性能に関する実験結果について議論し、4.2章で17個の隠喩文から抽出した「心的概念の物理的イメージ」に関する知識を例証する。

## 2 プリミティブマッチング法

### 2.1 基本コンセプト

PM法は、2つの概念の意味構造において、まず一次のプリミティブが一致する箇所を探し、それにより生成できる構造間対応関係を部分類比として決定し、その類比を拘束条件にして意味構造全体の類比を構造的同型性制約 [4] によって決定するという手法である。この拘束条件が、2つの構造間で類比付け可能な高次関係を求める際に無駄な探索を未然に枝刈りし、計算コストを軽減するのに有効であると期待できる。

まず、予め与えられた抽象的プリミティブを用いて言葉の意味を表現しなければならない。抽象的プリミティブは、(1) オブジェクトの属性、(2) オブジェクトの状態、(3) オブジェクトに関する動作・作用、(4) オブ

ジェクトどうしの関係、(5) 関係と関係の間の高次関係（因果関係を含む）に分類できる。(1) が0次のプリミティブ、(2)(3)(4) が1次のプリミティブ、(5) が高次のプリミティブである。

## 2.2 意味記述法

まず、本論文で例題として議論に使用する隠喩文を挙げる。”He shot down my opinion in the argument.”である。動詞 shoot down が隠喩的に使われた隠喩文であり、それに最も近い意味の動詞は criticize である。本論文では、動詞が隠喩的に使われている隠喩文だけを対象にする。

抽象的プリミティブを用いて記述した”A shoots down B”の意味構造を Fig.1 に示す。

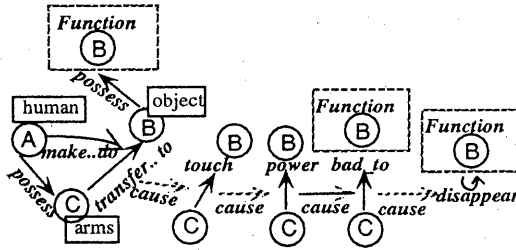


Fig.1 : Semantic representation of 'shoot down'

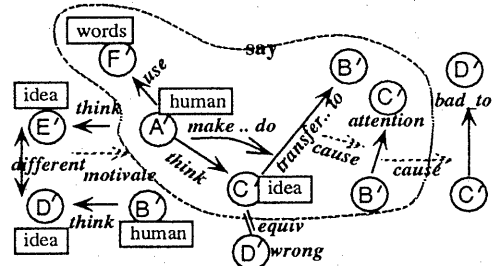


Fig.2 : Semantic representation of 'criticize'

「A が所有 (possess) しているもの C を、B に向けて移動 (transfer-to) させることに加担 (make...do) し、それが原因 (cause) で C が B に接触 (touch) し、それが原因 (cause) で C から B に対して力が働き (power)、それが原因 (cause) で B の機能 (function-of) にとって悪い (bad-to) 影響を及ぼし、それが原因 (cause) で B の機能 (function-of) が消滅 (disappear) する」

と読む。下線で示した言葉が抽象的プリミティブである。機能 (function-of) はオブジェクトの属性、とって悪い (bad-to) はオブジェクトの状態、所有 (possess)、接触 (touch)、移動 (transfer-to)、消滅 (disappear)、力が働き (power) はオブジェクトの動作・作用、加担 (make...do)、原因 (cause) は高次因果関係のプリミティブである。オブジェクトをノード、プリミティブをリンクとみなすと、言葉の意味はグラフ構造として表現されていることになる。更に、その動詞はどのような目的語や主語をとるか（上記記述の中のオブジェクト A,B,C は、オブジェクトの概念を表す taxonomy のどこに位置するものか）を記述しておく。

何を抽象的プリミティブとして選ぶかは難しい問題であるが、本論文では触れない。収集法および実際に使用したプリミティブ集合に関しては、[9] を参照されたい。

## 3 PM 法アルゴリズム

PM 法による計算モデルは、2つの動詞の意味記述を入力とし、両者構造間に成り立つすべての類比を生成する。

D および D' を2つの動詞の意味記述構造とする。hr、h、α、p、t を、それぞれ、高次関係式、高次関係の述語記号、一次関係式、一次関係の述語記号、オブジェクトを表す項とする。この時、類比φは、項、一次関係式、高次関係式の1対1の対応関係の集合で定義される。

PM 法では以下の3つのステップを経て、意味構造間の類比を決定する。

### 1. (一次プリミティブのマッチング)

以下のルールにしたがって、一次述語記号プリミティブの一致する箇所を探す。一次関係式  $\alpha = p(t_1, \dots, t_n) \in D$  および  $\alpha' = p'(t'_1, \dots, t'_n) \in D'$  と現在の類比φに関して、もし  $p = p'$  で、 $\{ \langle \alpha, \alpha' \rangle \} \cup \{ \langle t_i, t'_{j(i)} \rangle \mid i = 1, \dots, n \} \cup \phi$  が1対1対応を満たすなら、類比φを以下のように更新する。 $\phi = \phi \cup \{ \langle \alpha, \alpha' \rangle \} \cup \{ \langle t_i, t'_{j(i)} \rangle \mid i = 1, \dots, n \}$ 。このプロセスを繰り返して最終的に得られる類比をφ<sub>1</sub>とする。

## 2. (高次構造のマッピング)

$\phi_1$  と矛盾しないような高次関係どうしのマッピングを探す。 $\alpha_i = p_i(t_{i,1}, \dots, t_{i,m(i)})$ ,  $\alpha'_i = p'_i(t'_{i,1}, \dots, t'_{i,m(i)})$  とする。高次関係式  $hr = h(\alpha_1, \dots, \alpha_n) \in D$  および  $hr' = h'(\alpha'_1, \dots, \alpha'_n) \in D'$  と、現在の類比  $\phi$  に関して、もし、すべての  $i$  ( $1 \leq i \leq n$ ) に対して、1対1対応類比  $\psi_i = \{ \langle \alpha_i, \alpha'_i \rangle \} \cup \{ \langle t_{i,j}, t'_{i,k(j)} \rangle \mid j = 1, \dots, m(i) \}$  が存在し、しかも、 $(\bigcup_i \psi_i) \cup \phi$  が1対1対応という条件を満たすなら、類比  $\phi$  を以下のように更新する。 $\phi = \phi \cup \{ \langle hr, hr' \rangle \} \cup (\bigcup_i \psi_i)$ . このプロセスを繰り返して最終的に得られる類比を  $\phi_2$  とする。

## 3. (異なる一次関係どうしのマッピング)

述語記号は異なっても引数が既に  $\phi_2$  によって類比付けられている一次関係どうしを以下のルールにしたがって類比つける。 $\langle \alpha, \alpha' \rangle \notin \phi_2$  を満たすような一次関係式  $\alpha = p(t_1, \dots, t_n) \in D$  および  $\alpha' = p'(t'_1, \dots, t'_n) \in D'$  に関して、もし、 $\{ \langle t_i, t'_{j(i)} \rangle \mid i = 1, \dots, n \} \subset \phi_2$  なら、類比  $\phi$  を以下のように更新する。 $\phi = \phi \cup \{ \langle \alpha, \alpha' \rangle \}$ . このプロセスを繰り返して最終的に得られる類比を  $\phi_{total}$  とする。

一般に、プリミティブマッチングプロセスでは複数通りの類比  $\phi_1$  が得られ、その各々に関して  $\phi_{total}$  が生成される。その中で、もっとも対応関係の数が多いものを2つの意味構造の類比として生成する。

例文”He shot down my opinion in the argument”において、上記のアルゴリズムに従いソース概念 shoot down に対応する適切なターゲット概念を発見するという問題を考えよう。主語に’human being’を取り、目的語に’opinion’を取るような動詞がターゲット概念の候補である。例えば、’refuse’、’agree’、’disagree’、’listen to’、’neglect’、’understand’、’insist’などの動詞が挙げられる。それぞれの候補に関して’shoot down’との間に類比を求め、対応関係の数が最大の類比（最大類比）を生成した候補動詞をターゲット概念として選ぶ。上記の例文の場合、’criticize’が最大類比を生成する。ここでは、’shoot down’と’criticize’の間に生成される類比を例として挙げる。“A’ criticizes D”の意味構造を Fig.2 に示す。PM法で’shoot down’と’criticize’の類比を求めると、以下の2つの類比が得られる。

$$\begin{aligned} \phi_{total} = \{ & \langle c, c' \rangle, \langle b, b' \rangle, \langle transfer\_to(c, b), transfer\_to(c', b') \rangle, \langle bad\_to(c, function(b)), bad\_to(c', d') \rangle, \\ & \langle function(b), d' \rangle, \langle a, a' \rangle, \langle make\_do(a, transfer\_to(c, b)), make\_do(a', transfer\_to(c', b')) \rangle, \\ & \langle touch(c, b), attention(b', c') \rangle, \\ & \langle cause(transfer\_to(c, b), touch(c, b)), cause(transfer\_to(c', b'), attention(b', c')) \rangle, \\ & \langle cause(touch(c, b), bad\_to(c, function(b))), cause(attention(b', c'), bad\_to(c', d')) \rangle, \\ & \langle cause(transfer\_to(c, b), bad\_to(c, function(b))), cause(transfer\_to(c', b'), bad\_to(c', d')) \rangle, \\ & \langle possess(a, c), think(a', c') \rangle, \langle possess(b, function(b)), think(b', d') \rangle \} \end{aligned}$$

$$\begin{aligned} \phi_{total} = \{ & \langle c, c' \rangle, \langle b, b' \rangle, \langle transfer\_to(c, b), transfer\_to(c', b') \rangle, \langle bad\_to(c, function(b)), bad\_to(c', d') \rangle, \\ & \langle function(b), d' \rangle, \langle a, a' \rangle, \langle make\_do(a, transfer\_to(c, b)), make\_do(a', transfer\_to(c', b')) \rangle, \\ & \langle power\_op(c, b), attention(b', c') \rangle, \\ & \langle cause(transfer\_to(c, b), power\_op(c, b)), cause(transfer\_to(c', b'), attention(b', c')) \rangle, \\ & \langle cause(power\_op(c, b), bad\_to(c, function(b))), cause(attention(b', c'), bad\_to(c', d')) \rangle, \\ & \langle cause(transfer\_to(c, b), bad\_to(c, function(b))), cause(transfer\_to(c', b'), bad\_to(c', d')) \rangle, \\ & \langle possess(a, c), think(a', c') \rangle, \langle possess(b, function(b)), think(b', d') \rangle \} \end{aligned}$$

$$\text{In both cases, } \phi_1 = \{ \langle c, c' \rangle, \langle b, b' \rangle, \langle transfer\_to(c, b), transfer\_to(c', b') \rangle, \langle function(b), d' \rangle, \langle bad\_to(c, function(b)), bad\_to(c', d') \rangle \}$$

最初のステップであるプリミティブマッチングプロセスで得られる類比  $\phi_1$  はただ1種類であるが、その後のプロセスで2通りに場合分けされる。両者の違いは、attention(b,c) が類比付けられる相手が touch(c,b) か power\_op(c,b) かの違いである。この場合、両者とも対応付けの数が同じなので、PM法ではどちらの類比が適切かを決めることはできない。本論文では、両方とも適切であるとしておく。

## 4 実験結果

### 4.1 PM 法による計算コスト

我々は、Lakoff の著書 [6] および他の文献から、動詞が隠喩的に使われている 17 個の隠喩文を集め、構造的な同型性制約だけによる類比決定法 (structure mapping approach、SM 法と呼ぶ) に比べて、PM 法が計算コストの面でどれだけ有利かを調べた。各々の隠喩文に関して、文中で隠喩的に使われている動詞 (ソース概念) に対応するターゲット概念の動詞は計算モデルに与えて、両者の意味構造間のすべての類比を算出するのに要する CPU 時間を測った。

Figure 3 に、実験に使った 17 個の隠喩文を示す。各々に付記した括弧内の動詞が、それぞれに与えたターゲット概念動詞である。Table 1 に、PM 法および SM 法で要した計算コストと、PM 法によるコストゲイン値 (PM 法により何倍速くなったか) をまとめた。17 のケースは、両動詞の意味構造に共通の一次プリミティブが存在した隠喩文 (Group 1) と、存在しなかった隠喩文 (Group 2) に分類できる。Group 2 では、PM 法は SM 法に比べてわずかではあるが多くの計算コストを要する。それは、共通の一次プリミティブが存在しないので、そのマッチングを探すプロセスが無駄に終り、その後のプロセスでは SM 法と全く同じプロセスをたどることになるからである。一方、Group 1 では、ほとんどの隠喩文で PM 法による効果が見られる。その理由は、高次関係式どうしの類比付けが可能かどうかを調べる時に、一次プリミティブのマッチングにより得られた類比との矛盾チェックを行なうことによって、高次関係式の内部構造にまで深く調べなくても簡単に類比付けの可否を判定できる、即ち、全体の類比を求める際の探索空間が絞り込まれるからである。

- Case 1: We need to explore the issues carefully. (examine)  
 Case 2: He tried to grind out new solutions. (elaborate)  
 Case 3: If you use that strategy, he'll wipe you out. (destroy)  
 Case 4: The great idea was planted in his youth. (developed)  
 Case 5: He shot down my opinion in the argument. (criticize)  
 Case 6: Where'd you dig up that idea? (discover)  
 Case 7: They exploded his latest theory. (disprove)  
 Case 8: I just can't swallow that claim. (accept)  
 Case 9: He pointed out the demerits of the strategy. (show)  
 Case 10: Tigers crushed Giants in the baseball game. (defeat)  
 Case 11: The argument collapsed. (fail)  
 Case 12: She killed a process. (stop)  
 Case 13: I got out of a lisp. (stop doing)  
 Case 14: We need to let that idea percolate. (spread)  
 Case 15: There are too many facts here for me to digest them all. (understand)  
 Case 16: His words carry little meaning. (represent)  
 Case 17: Try to pack more thought into fewer words. (express by)

Fig. 3 The metaphorical sentences employed

Table 1 The calculation-time costs, using PM approach and SM approach

Group	Case No.	PM (msec)	SM (msec)	Cost Gain (PM/SM)
1	1	949	5684	6.0
	2	7130	39350	5.5
	3	1351	5134	3.8
	4	2351	8585	3.7
	5	2615	8451	3.2
	6	299	868	2.9
	7	4984	15452	3.1
	8	734	1350	1.8
	9	417	500	1.2
	10	217	284	1.3
	11	3168	2300	0.73
2	12	67	68	1.0
	13	83	100	1.2
	14	550	450	0.82
	15	367	350	0.95
	16	567	450	0.79
	17	832	837	1.0

一次プリミティブマッチングによる探索空間の減少割合と CPU 時間コストゲインの関係を明確にするために、両動詞の意味構造の類比付け複雑性 (mapping complexity と呼ぶ) という新しい指標を導入する。mapping complexity  $MC(v1, v2, \phi)$  は、動詞  $v1$  と  $v2$  の類比を求めるプロセスの途中において、現在までに得られている類比が  $\phi$  であるとした時、どのくらいの類比付けの複雑性が残っているかを大まかに表す指標であり、次の式に従い計算する。

$$MC(v1, v2, \phi) = N_{f,1}(v1, \phi) * N_{f,1}(v2, \phi) + N_{f,2}(v1, \phi) * N_{f,2}(v2, \phi) + N_{f,n}(v1, \phi) * N_{f,n}(v2, \phi) + N_{h,1}(v1, \phi) * N_{h,1}(v2, \phi) + N_{h,2}(v1, \phi) * N_{h,2}(v2, \phi) + N_{h,n}(v1, \phi) * N_{h,n}(v2, \phi),$$

但し、 $N_{f,1}(v, \phi)$ 、 $N_{f,2}(v, \phi)$ 、 $N_{f,n}(v, \phi)$ 、 $N_{h,1}(v, \phi)$ 、 $N_{h,2}(v, \phi)$ 、 $N_{h,n}(v, \phi)$  は、それぞれ、動詞  $v$  の意味記述中において類比  $\phi$  の要素として現れない一引数の一次関係式の数、動詞  $v$  の意味記述中において類比  $\phi$  の要素として現れない二引数の一次関係式の数、動詞  $v$  の意味記述中において類比  $\phi$  の要素として現れない三引数以上の一次関係式の数、動詞  $v$  の意味記述中において類比  $\phi$  の要素として現れない一引数の高次関係式の数、動詞  $v$  の意味記述中において類比  $\phi$  の要素として現れない二引数の高次関係式の数、動詞  $v$  の意味記述中において類比  $\phi$  の要素として現れない三引数以上の高次関係式の数を意味する。つまり、 $MC(v1, v2, \phi)$  は 2 つの動詞の意味記述において残っている、構造的に類比付け可能かどうかを調べなければならない一次及び高次関係式の組合せの数を表しており、 $\phi$  という部分類比が得られている段階において全体類比決定のために残された探索空間の大きさを意味している。

一次プリミティブのマッチングのあとに残っている mapping complexity の割合 ( $MC(v1, v2, \phi_1)/MC(v1, v2, \{\})$ ) を横軸に、PM 法によるコストゲインを縦軸にとり、Group 1 に属する 11 のケースについて両者の関係を調べた結果が Fig.4 である。

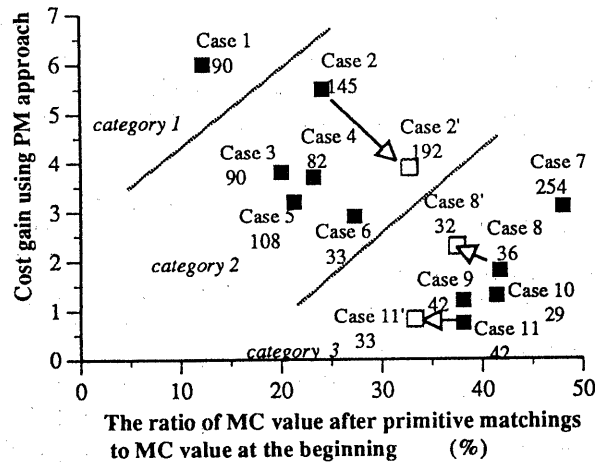


Fig.4 The results of cost gain by PM approach

但し、 $\phi_1$  は一次プリミティブのマッチングにより得られた部分類比である。2 つの動詞の意味記述の中に幾つの一次プリミティブマッチングが存在するかによって、11 のケースを 3 つのクラスに分類して、Fig.4 の結果を考察する。Category 1 は、3 つのマッチングが存在する場合で Case1 がそれに属する。Category 2 は、2 つのマッチングが存在する場合で Case2 から Case6 がそれに属する。Category 3 は、1 つのマッチングが存在する場合で Case7 から Case11 がそれに属する。プリミティブマッチングの個数が多いほど、それによって枝刈りされる探索空間がおおきくなり、その結果 PM 法が SM 法に比べてより有利になるという傾向が見られる。この結果は、プリミティブマッチングにより構造的な同型性の発見プロセスにおける無駄な探索が避けられることを例証するものである。

次に、カテゴリー内でのコストゲインの比較について考察する。Fig.4 の各々のプロットには、それぞれのケースの類比決定の matching complexity 値 ( $MC(v1, v2, \{\})$ ) が付記されている。それは、ほぼ 2 つの動詞の意味の記述量に比例する値である。Category 2 および 3 において、記述量が比較的大きい場合には PM 法が SM 法より優れているという傾向が見られる。

次に、生成可能なすべての類比の中には不必要なプリミティブマッチングによる間違っただけの類比が含まれることもあり、その場合に計算コストがどのくらい大きくなるかについて議論する。Case8 および Case11 では、2 つの動詞の意味構造の中に不必要なプリミティブマッチングを起こす可能性のある記述があった。そこで、意図的にその不必要なマッチングを起こす主要原因である記述を取り除いて (それぞれの動詞の意味の本質的な性

質を変えないという条件で<sup>1)</sup>、すべての類比を生成するに要する計算コストを再度測定した。また、Case2では、元々不必要なマッチングを起こすような記述は存在しなかったが、正解の類比には影響を及ぼさない範囲でわざわざ不必要なマッチングを起こすような記述を付け加えて計算コストを再度測定した。これらの3つのデータは、Fig.4において中抜ききの四角のマークで示されている。不必要なプリミティブマッチングが存在した場合、確かに計算コストが大きくなることが確かめられる。興味深いことに、不必要なマッチングによるコストゲインの低下の度合いが、全データに関するコストゲインのMC値に対する傾向のカーブとほぼ一致している。不必要なマッチングによるコストゲインの低下という本質的に不可避な性質が、PM法によるコストゲインを相殺するほどのものかどうか、それとも依然PM法は計算コスト面で有利なのかは、より多くの証拠を集めなければ結論はでない。

#### 4.2 心的概念の物理的イメージ

本論文中で使ったプリミティブは、'attention', 'think', 'know', 'mean'などの心的概念を表すものと、'power\_op', 'touch', 'located'などの物理的概念を表すものに大別できる。類比を求めた結果として得られた両概念間の対応関係は、「心的概念の物理的イメージ」に関する人間固有の知識であるとみなせる。

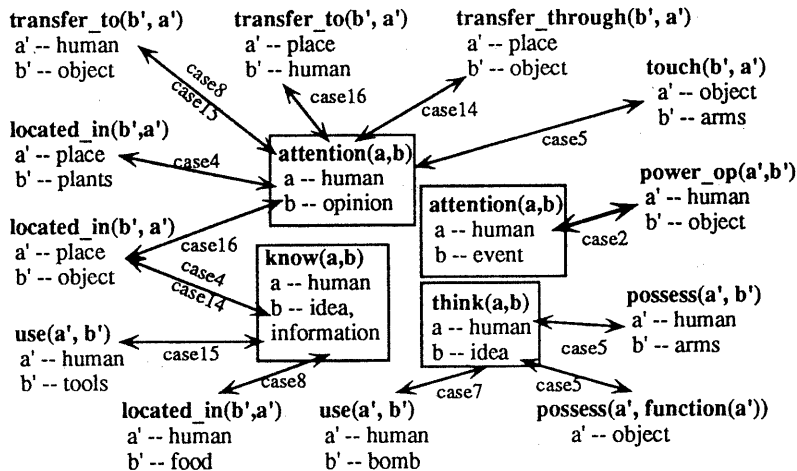


Fig.5 Physical images of concepts in the mental world

Fig.5に、17個の隠喩文を処理した結果として得た、'attention', 'think', 'know'の物理的イメージに関する知識を示す。図中の矢印が、それぞれ、心的概念(長方形で囲まれたもの)の物理的イメージを表している。それぞれの心的概念の物理的イメージは、その概念述語がどのような種類のオブジェクトを引数にとるかによって変わる。例えば、'attention'の場合、第2引数に'event'をとる場合、その'event'は'object'であると見立てられ、関係式'attention(human,event)'は関係式'power\_op(human,object)'に見立てられる。つまり、「人が何かの'event'に気付くとは、すなわち、何かの'object'に外力をおよぼすことに等しい」という隠喩的知識である。また、'attention'が第2引数に'opinion'か'idea'をとる場合、'opinion'(または'idea')は'object'もしくは'human'に見立てられ、'attention'の第1引数である'human'は'places'に見立てられる。これは、「人間がある意見や考え方に気がつくとは、即ち、その意見や考え方があたかも物体のようにその人のいる場所へ移動してくる('transfer.to')もしくはその場所を通過し('transfer.through')、何らかの相互作用を起こす('touch')かその場所にとどまる('located.in')ことに等しい」という隠喩的知識である。

<sup>1)</sup> それを取り除いても生成される正しい類比は変わらないように配慮した

## 5 結論

プリミティブが一致する部分の部分類比を拘束条件にして意味構造全体の類比を決定する手法（プリミティブマッチング法）を用いて、隠喩文の類比発見するアルゴリズムを作成した。17個の隠喩文において類比決定を行い以下の結果を得た。プリミティブマッチング法（PM法）は、2つの意味構造中に対応付け可能な同じプリミティブが存在する場合に、明らかに類比決定における探索空間を狭める効果があり、構造的同型性制約（SM法）のみで類比決定を行なうよりも計算量的に有利である。また、プリミティブマッチングプロセスによる探索空間の低減率が同程度であるような2つの隠喩文に関してPM法によるコストゲインを比較したところ、使われている動詞の意味記述量が大きい程、コストゲインがおおきいという興味深い傾向を得た。更に、類比決定の産物として、3つの心的概念、'attention', 'think', 'know'の物理的イメージに関する知識を得た。

## 参考文献

- [1] J. A. Barnden. Belief, metaphorically speaking. In *Proceedings of 1st international conference on principles of knowledge representation and reasoning*, pp. 21-32, 1989.
- [2] J. G. Carbonell. Metaphor - A key to extensible semantic analysis. In *Proceedings of the Third Annual Conference of the Cognitive Science Society*, pp. 292-295, 1980.
- [3] D. Fass. *Collative Semantics: A Semantics for Natural Language*. PhD thesis, New Mexico State University, New Mexico, 1988.
- [4] D. Gentner. Structure mapping: A theoretical framework for analogy. *Cognitive Science*, Vol. 7, No. 2, pp. 155-170, 1983.
- [5] K. J. Holyoak and P. Thagard. Analogical mapping by constraint satisfaction. *Cognitive Science*, Vol. 13, pp. 295-355, 1989.
- [6] G. Lakoff and M. Johnson. *Metaphors we live by*. University of Chicago Press, 1980.
- [7] J. H. Martin. *A Computational Model of Metaphor Interpretation*. Academic Press (Perspectives in Artificial Intelligence series), 1990.
- [8] S. W. Russell. Computer understanding of metaphorically used verbs. *American Journal of Computational Linguistics*, Vol. 44,, 1976.
- [9] M. Suwa and H. Motoda. Learning metaphorical relationships between concepts based on semantic representation using abstract primitives. In *Proceedings of the IJCAI 1991 Workshop: Computational Approaches to Non-Literal Language: Metaphor, Metonymy, Idiom, Speech Acts, Implicature*, 1991.
- [10] Y. Wilks. Making preferences more active. *Artificial Intelligence*, Vol. 11, pp. 197-223, 1978.