

〔非公開〕

TR-C-0055

ユーザモデルを用いた知的文献検索システムの仕様

加納 康男
YASUO KANOU

1990. 8. 28

ATR通信システム研究所

ユーザモデルを用いた知的文献検索システムの仕様

加納 康男

Yasuo Kanou

ATR 通信システム研究所

1990年8月28日

目次

1 システムの全体構成	5
1.1 概要	5
1.2 本実験システムの各部の機能概要	6
1.2.1 問い合わせ文の構文解析	6
1.2.2 ユーザモデル	7
1.2.3 隠れた検索要求の推論処理	7
1.2.4 ユーザモデル構築部：対話によるユーザモデルの構築	8
1.2.5 検索式合成部	9
1.2.6 検索文献の表示と利用者による適合文献の選択	9
1.2.7 検索文献からの意図理解	9
1.2.8 文献検索効率評価部	11
2 構文解析	13
2.1 単純入力	13
2.1.1 氏名の入力	13
2.1.2 検索要求文の入力	13
2.2 回答	13
2.2.1 肯定回答	13
2.2.2 否定回答	13
2.2.3 関係の重みの修正	14
2.3 未知語の説明	14
2.3.1 階層関係	14
2.3.2 同義関係	15
2.3.3 関連関係	15
2.3.4 返答付き未知語獲得	15
2.4 SUBJECT の格表現について	15
2.4.1 単一概念	15
2.4.2 限定表現	16
2.5 併記	16
2.6 例	16
3 文献検索システム対話の方法	19

4	ユーザモデル構築の対話処理仕様書	23
4.1	名前の入力	23
4.2	検索要求文の入力	23
4.3	照合規則	24
4.3.1	単一概念同士の照合 [概念同士の照合]	24
4.3.2	構造をもつ概念の照合 [構造同士の照合]	27
4.4	ユーザモデル操作のルール	30
4.4.1	意図の重みの修正	30
4.4.2	未知語の獲得	32
4.4.3	関係構造の構築 [関係構造の構築]	33
4.5	システムの理解内容の確認と構造の修正	36
4.6	全体の対話について共通	38
5	真意の推論処理仕様書	39
6	複数の検索意図を反映するユーザモデルを用いた知的文献検索システム	41
6.1	基本概念	41
6.2	処理の流れ	41
6.3	ユーザモデルの構築と対話	42
6.3.1	ユーザモデル修正操作・オペレーションの定義	42
6.3.2	対話を行うタイミングとユーザモデル構築の処理	42
6.4	検索意図の識別子の決定方法	44
6.4.1	基本意味構造が構造を持つとき	44
6.4.2	基本意味構造が単一概念のとき	44
6.5	深層意図の推論	45
7	検索式合成部	47
7.1	論理式合成部の機能	47
7.2	基本アーキテクチャ	47
7.3	入力の形式	47
7.4	出力ファイルの形式	47
7.4.1	出力ファイルの例	47
7.4.2	出力ファイルのフォーマット	48
7.5	最適論理式の合成方法	49
7.5.1	有効グラフの作成	49
7.5.2	合成規則	49
7.6	条件の緩和規則	51
8	フィードバック付き文献検索システム仕様書	53
8.1	フィードバック付き文献検索システムの概要	53
8.2	検索実行部	54
8.2.1	処理内容	54
8.2.2	入力ファイルフォーマット	55
8.2.3	出力ファイルフォーマット	56
8.3	抽出文献選択部	56

8.3.1	処理内容	56
8.3.2	入力ファイルフォーマット	56
8.3.3	出力ファイルフォーマット	57
8.4	ユーザモデル修正部	58
8.4.1	処理内容	58
8.4.2	キーワード解析処理	58
8.4.3	ユーザモデル更新処理部	59
9	フィードバック付き文献検索システム操作方法	61
A	典型的対象世界知識	67

第 1 章

システムの全体構成

1.1 概要

本実験システムは、文献データベースに対する日本語で記述されたあいまいな問合せ文を理解し、それに基づく検索を行なうとともに、その性能を評価するためのものである。このために、本システムでは次のようなアプローチを採用している。

ユーザモデルの導入

ユーザモデルは、利用者単位の個人用シソーラスのことである。これに利用者独自の言葉や検索要求の意味構造を持たせる。

ユーザモデルを使用した問合せ文の理解

問合せ文の理解は、ユーザモデルの中から利用者の検索意図に照応する部分構造（意味構造と呼ぶ）を抽出することに対応する。この照応作業により、あいまいな問合せ文の理解を行なう。

システムは、利用者各々に対して、1つのユーザモデルを持つ。ユーザモデルを用いることによって、利用者独自の言葉を理解するとともに問い合わせ文に現れなかった検索要求を推論して補完することが可能となる。

あいまいな問合せ文の検索意図は、次のようにして抽出できる。

- (1) 問合せ文に直接表明されている検索意図。
- (2) システムと利用者との対話を通して、利用者が表明する検索意図。
- (3) システムの検索結果から利用者が選択した文献を解析することによって発見される検索意図。

本システムでは、文献データベースはキーワードにより索引付けされた、従来型のものとしている。問合せ文の理解は、上記の検索意図を抽出し、それに相当するキーワード集合およびそれを用いた論理式に変換することである。

また、本文献検索システムに対しては、従来より文献検索の指標として用いられている文献検索効率だけでなく、問い合わせ理解に対する指標としてより直接的なキーワード検索効率を用いた評価を行ない、その有効性を示す。

本システムでは問い合わせ文に含まれるあいまい性のうち、次のものを対象とする。

- (1) 利用者独自の言葉、あるいは特有の言葉の使い方
利用者の問い合わせ文には、システムの知識にある言葉（キーワード）以外の独自の言葉を含むことがある。またキーワードであっても、利用者特有の意味で使用することがある。
- (2) 限定の不足
利用者が検索要求を問い合わせ文で十分に限定しない。

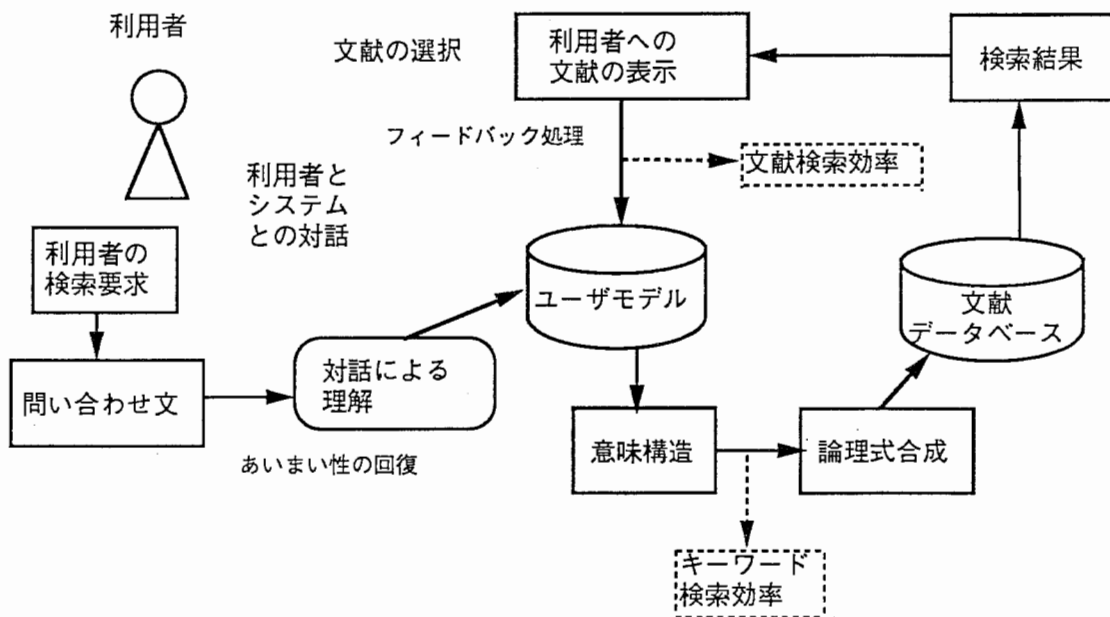


図 1.1: ユーザモデルを用いた知的文献検索実験システムの構成

(3) 過剰の限定

利用者が問い合わせ文で使った言葉の意味を広く解釈する方が望ましい場合がある。例えば、“Dempster Shafer 理論”と言ってもこれ以外の不確実性を表現するものを含むことがある。

(4) 複数の検索要求を同時に行なう

同じ利用者であっても、異なる意味構造を持った複数の検索要求を同時に行なうことがある。

図 1 に、本実験システムの構成を示す。本システムは、SUN-3/260 上の ART、LISP、C を用いてインプリメントされる。

1.2 本実験システムの各部の機能概要

1.2.1 問い合わせ文の構文解析

利用者の問い合わせ文は「(名詞句表現)に関する文献が欲しい」という形式で表現されるものとする。この名詞句表現を、表 1.1 に従って、概念を表現している 2 つの名詞とその間の関係構造を抽出し、3 項形式で表現する。

例: 「意味ネットワークによる対象世界モデルの表現」

(手段限定 (対象限定 表現 対象世界モデル)
意味ネットワーク)

この構文解析処理は、構文解析ツール「PARSER」を用いる。

表 1.1: 問い合わせ文中の名詞間の関係の分類 (例)

関係の種類	関係を表す言葉
階層関係	を含むなどの としての 特に 例えば
対象限定	のでの における への に対する
手段限定	による を用いた を使った を使用した
目的限定	のための に用いる
因果関係	による
関連関係 (併記)	とおよび や または あるいは

1.2.2 ユーザモデル

ユーザモデルは、名詞で表された概念ノードと概念ノード間の関係を表すリンクを持つ意味ネットワークで表現する。概念ノードには、検索キーワードに対応するキーワードノードと、独自語に対応する独自語ノードを設ける。

リンクは、次のように表現される。

((ノード N_p) (ノード N_q) (関係の種類 R)
(意図のレベル L) (意味構造識別子 I_i))

関係の種類 R には、階層関係、関連関係、同義関係、限定関係 (対象限定、手段限定、目的限定、因果関係) の 4 種の関係の種類があり、これらは問い合わせ文中の助詞 (表 1.1) や対話の内容から決定される。

意図のレベル L は、2つの概念間での利用者の連想の強さを示し、‘初期状態’、‘意図する’、‘意図しない’の 3 段階ある。重みのついたリンクは、問い合わせ文で現れなかった検索要求を推論するために用いる。

意味構造識別子 I_i は、同一のユーザモデル内にある複数の検索要求に対応する意味構造を識別するためのものである。

複数の意味構造を識別するために、構文解析された意味構造に対応するユーザモデル上の構造 (基本意味構造上) のノードも意味構造識別子 I_i をもつ。

ユーザモデルは、初期状態として、キーワードノード間を初期状態のリンクで結んだ典型的な対象世界モデル (シソーラス) として定義されている。ユーザモデルは、利用者の問い合わせ文と利用者との対話から初期シソーラスを修正することによって構築される。

現在のシステムでは、人工知能分野を対象分野とする。典型的な対象世界モデルを付録 A に示す。これは、約 800 のキーワードノードからなる。本システムでは、ユーザモデルを ART のスキーマを用いて表現する。即ち、ノードをスキーマ、リンクをリンクスキーマとして表現する。ノードは、キーとして階層番号 (hno) を持っており、さらにノードの日本語名称を表す j -alias を属性として持っている。

1.2.3 隠れた検索要求の推論処理

ユーザモデルを用いて、問い合わせ文に含まれない検索要求を推論するとともに、これに相当するキーワードを抽出する。問い合わせ文の関係構造に対応するユーザモデル上の構造 (基本意味構造) から、対応する意味構造識別子を持ち、‘意図する’レベルにあるリンクを次の推論規則によってトレースしてキーワードを抽出する。

- (1) 上位キーワード ⇒ 下位キーワード
- (2) 関連関係、同義関係、限定関係リンクは両方向に推論する
- (3) 独自語 ⇒ キーワード

ここでトレースされたリンクとノードの構造が、問い合わせ文に対応する意味構造である。

1.2.4 ユーザモデル構築部：対話によるユーザモデルの構築

問い合わせ文に含まれる名詞の意味や関連する概念を利用者に尋ねることにより検索要求を明確化し、これを意味構造としてユーザモデルに反映する。この対話によってユーザモデルが構築される。

意味構造の決定・構築

構文解析された問い合わせ文に対応するユーザモデル上の構造（基本意味構造）を同定もしくは構築し、対応する意味構造を構築する。

構文解析された問い合わせ文の関係構造と、既にユーザモデル内に構築された意味構造の関係によって、意味構造の識別および構築は次の3つの場合に分けられる。

Case 1: 基本意味構造が、既に意味構造識別子を付与された意味構造の部分集合で、識別子が一意の場合基本意味構造内の識別子が、入力された問い合わせ文に対応する意味構造識別子と考える。基本意味構造から、この識別子を持ち、‘意図する’レベルを持つリンクをトレースしてキーワード集合を抽出する。

Case 2: 問い合わせ文において、利用者が新しい概念や概念間の新しい関係を表明した場合新しい検索要求として、新しい識別子をもった意味構造を構築する。このような、ユーザモデル構築のための対話には次のようなものがある。

(1) あいまいな概念の意味の明確化

問い合わせ文中の名詞に対応する概念ノードに対して初期状態レベルのリンクで連結されている下位または関連、限定関係の概念ノードがあれば、これらを利用者に問い合わせ、その回答に応じてリンクの意図のレベルを‘意図する’もしくは‘意図しない’に修正する。これによって、一つのあいまいな概念に対する意味構造が規定される。

(2) 未知語に対する処理

問い合わせ文中の言葉が未知語の時は、利用者にその言葉の説明を求める。システムは、対話に基づいてこれを新たなリンク（意図するレベル、新しい識別子をもつ）と新たな独自語ノードとしてユーザモデルに組み入れる。

(3) 関係構造の付加

利用者がユーザモデルにある概念ノードを用いてそれらの間の新たな関係を表明したときには、その関係に基づいたリンク（意図するレベル、新しい識別子をもつ）を付加する。

Case 3: 基本意味構造内に、既に付与されている意味構造識別子が複数あり一意に決まらない場合すでに構築された意味構造を利用して、次のように新しい意味構造識別子をもつ意味構造を構築する。

基本意味構造内に既に付与されている意味構造識別子が付与され、‘意図する’レベルにあるリンクをトレースして候補となるキーワードを抽出し、これを利用者に表示する。利用者は、検索要求に対応するキーワードを選択する。利用者が選択したキーワードをトレースしたリンクには、新しい意味構造識別子を付加する。

これによって、既に構築された意味構造を利用して、新しい検索要求に対する意味構造が簡単に構築できる。

利用者への確認と補足する要求の入力対話の終りに当たって、抽出された全てのキーワードを利用者に表示し確認を求める。

利用者が、新たなキーワードを要求した場合は、そのキーワードと基本意味構造にある概念ノードの間に、‘意図する’レベルで現在の意味構造識別子を持つリンクを付与する。

利用者が、表示されたキーワード集合の中のキーワードを否定した場合には、そのキーワードをトレースしたリンクの意図のレベルを‘意図しない’に修正する。

1.2.5 検索式合成部

ユーザモデルを用いた論理式の合成方法

対話の終了後、ユーザモデルにおける検索要求の意味構造を利用して文献データベースをアクセスするための論理式を合成する。基本的な考え方は次の2点である。

- (1) 意味構造におけるリンクの構造と、関係の種類および意図のレベルを利用して論理式を合成する。意味構造に含まれる全てのキーワードおよび関係構造を利用して作られた最も制約条件の強い論理式を最適論理式とする。
- (2) 最適論理式で文献データベースをアクセスしたときに、対応する文献が少ない場合には、制約条件を緩和した論理式によって文献データベースを検索する。このようにして順に制約条件を緩和した論理式を合成して文献データベースを検索する。制約条件を緩和したことは論理式に与えられる評点に反映する。

1.2.6 検索文献の表示と利用者による適合文献の選択

評点の高い、即ち制約条件の高い最適な論理式から、順に制約条件の弱い論理式を用いて文献データベースをアクセスする。

利用者には、評点の高い論理式で検索された文献から順に、各評点の論理式ごとに検索された文献集合を表示する。最終的に得られた文献数が少なくとも20件以上かつ多くとも100件になるまで条件の緩和された論理式を用いる。(表示された文献数が20件を越えた時点で終了)。

評点の低い論理式によって検索された文献集合には、評点の高い論理式によって検索された文献集合を含む。そこで、低い評点で示された表示文献では、評点の高い文献は除いてある。利用者はマウスクリックによって適合文献を選択する。ここで選択された文献から、後述するように、各論理式ごとの検索効率を計算するとともに、フィードバック情報が作成される。

1.2.7 検索文献からの意図理解

フィードバック手法の必要性

利用者が表面上あまり意識しておらず、従って対話により抽出できないキーワードであっても、検索された文献集合に高い確率で存在するものがある。

このようなキーワードは、利用者が陽に意識していないが、文献データベースの検索のために有効なものである。このキーワード集合を抽出し、論理式に含めて再び検索を行なうことによって、文献の検索効率を向上させることが期待できる。

本システムでは、検索文献のうち利用者が選択した文献の中から、対話処理だけでは抽出できなかった検索要求に適合する隠れたキーワードを発見し、これをユーザモデルの意味構造に反映して、

再び検索を行なう。これをフィードバック処理と呼ぶ。新たに抽出されたキーワードをユーザモデルに反映しておくことにより、次の問い合わせ時にはこれを自動的に抽出できる。

検索結果からの検索意図の抽出方法

検索された文献のうち利用者が選択した文献に含まれるキーワードを解析することにより、利用者が意識していないが検索に有効なキーワードを抽出する。

即ち、利用者が選択した文献に付与されているキーワードのうち、

- 全文献中の出現頻度に比べて高い頻度で出現するキーワード
- 高い頻度で共起して出現するキーワード

を、利用者の検索要求を代表するキーワード考える。これらの検索要求を代表するキーワードのうち、対話処理で抽出できなかったものを抽出する。

ところで、対象とする分野において、データベース中 50% 以上の文献に付加されるようなその対象分野そのものを表したり、広い意味で与えられているようなキーワード(例えば人工知能分野では、“人工知能”、“エキスパートシステム”、“知識表現”といったキーワード)も上記の条件にあてはまる。しかし、このようなキーワードがフィードバックされて意味構造に含まれると、極めて広い範囲の文献が検索され再現率の低下を招くため、条件から省くことが必要である。

また、フィードバックされるキーワードは、対話によって抽出されたキーワードとの関連性を持つものに限ることが望まれる。

このような見地から、検索文献からの意図理解の結果をユーザモデルの意味構造にフィードバックするために、次の基準を満たすキーワードペアを抽出する。

(1) 次の条件を満たすキーワード集合を抽出する。

$$\frac{S_n}{S} > \frac{N_n}{N} \times 2$$

S_n : 利用者が選択した文献中、
そのキーワードを含む文献数

S : 利用者が選択した文献数

N_n : 全文献中、そのキーワードを含む文献数

N : 全文献数

(2) (1) で得られたキーワード集合において、利用者が対話によって選んだキーワードと共起関係が最大のキーワードのペア。(但し、既に対話によって抽出されて意味構造に含まれていたキーワードとのペアは除く)

検索結果からのユーザモデルの構築方法

フィードバック処理により抽出したキーワードを検索要求を表すものとしてユーザモデルに取り込んでおけば、以降の問い合わせ文の処理において自動的に隠れた検索要求として推論することができる。

これは次のように、抽出されたキーワードペアをユーザモデルのノードとその間のリンクとして付加することによって行なう。

- ユーザモデル上の対応するキーワードノード間で、既に現在の意味構造識別子を持ったリンクが存在すれば、その意図のレベルを‘意図する’に修正する。

- ユーザモデル上の対応するキーワードノード間に、現在の意味構造識別子を持ったリンクが存在しないならば、それらのノード間に現在の意味構造識別子を持った関連関係の‘意図する’レベルのリンクを追加する。

1.2.8 文献検索効率評価部

本実験システムでは、次のように定義するキーワード検索効率および文献検索効率を用いて、評価を行なう。

評価方法の概念については、ATR テクニカルレポート TR-C-0011 および TR-C-0041 を参照のこと。

キーワード検索効率

問合せ文の意味構造中のキーワードから、次のようにキーワード検索効率（キーワード適合率 (Kp_q) およびキーワード再現率 (Kr_q)) を定義する。

$$Kp_q = \frac{|Eq'|}{|Eq|}, \quad Kr_q = \frac{|Eq'|}{|Pq'|}$$

| Pq' |: 全キーワードのうち利用者の検索意図に適合したキーワードの数

| Eq |: システムが抽出したキーワードの数

| Eq' |: システムが抽出したキーワードのうち利用者の検索意図に適合したキーワードの数

文献検索効率

文献検索効率（文献適合率 (Rp_q) および文献再現率 (Rr_q)) は、それぞれ上記のキーワード検索効率の定義の“キーワード”を“文献”と読み換えたものとなる。

文献検索効率は順位付け出力を採用するシステムにおいては一般に再現率-適合率曲線で表現される。

文献検索効率評価部では、検索式合成部を通して作成された検索式の実行の結果得られた文献を、利用者が全文献に対して付与した○×に従って、キーワード検索効率および文献検索効率を計算するものである（問合せ文の検索意図にあった文献は○、そうではないものは×）。これらの検索効率は、対話処理に得られるものだけでなく、対話処理の各段階における適合率/再現率とその推移も求める。

第 2 章

構文解析

システムとの対話の中で利用者が入力する文の「PARSER」による解析結果を示す。
最終システムでは、回答を選択で行なう質問に対しては、マウスクリックによる選択を行なうので、
ここで示す自然言語入力を行なわない。

2.1 単純入力

2.1.1 氏名の入力

入力文 “名前は、SIMEI です”

格表現

((input simei) SIMEI))

2.1.2 検索要求文の入力

入力文 “SUBJECT についての文献が欲しい”

入力文 “SUBJECT に関する文献が欲しい”

格表現

((action retrieval)

(object (subject SUBJECT)))

2.2 回答

2.2.1 肯定回答

入力文 “そうです”

入力文 “はい”

格表現

((action answer)

(response positive))

2.2.2 否定回答

入力文 “いいえ”

入力文 “違います”

入力文 “ありません”

格表現

((action answer)
(response negative))

2.2.3 関係の重みの修正

入力文 “A 1 と A 2 だけを含みます”

格表現 ((action answer)
(select intention)
(selected (A 1 A 2)))

入力文 “全て含みます” “全てを含みます”

格表現 ((action answer)
(select intention)
(selected all))

入力文 “A 1 と A 2 は含みません”

格表現 ((action answer)
(select intention)
(denied (A 1 A 2)))

入力文 “どれも含みません”

“全体の概念が欲しい”
“制限しません”

格表現 ((action answer)
(select intention)
(denied all))

2.3 未知語の説明

(action explain) によって未知語を獲得する

2.3.1 階層関係

入力文：

“MICHIGO とは A の下位概念です”

“MICHIGO は A の一部です”

“MICHIGO は A 分野の言葉です”

“MICHIGO は A の下位の言葉です”

格表現

((action explain)
(object (subject (階層下位 MICHIGO A))))

入力文：

“MICHIGO とは A の上位概念です”
“MICHIGO は A から構成されています”
“MICHIGO は A の上位の言葉です”

格表現:

((action explain)
(object (subject (階層上位 MICHIGO A))))

2.3.2 同義関係

入力文 “MICHIGO とは A のことです”

格表現

((action explain)
(object (subject (同義関係 MICHIGO A))))

入力文 “MICHIGO とは BUBYA 分野の A のことです” (準完全照合の選択の時に限る)
“MICHIGO は BUNYA 分野の A のことです”

格表現

((action explain)
(object (subject (同義関係 MICHIGO A)
(subject (階層下位 MICHIGO BUNYA))))))

2.3.3 関連関係

入力文 “MICHIGO は A に関係した言葉です” “MICHIGO は A に関係する言葉です”

入力文 “MICHIGO は A に関連した言葉です” “MICHIGO は A に関連する言葉です”

格表現:

((action explain)
(object (subject (関連関係 MICHIGO A))))

2.3.4 返答付き未知語獲得

入力文 “いいえ, A は BUNYA 分野の言葉です” など

格表現:

((((action answer)
(response negative))
(action explain)
(object (subject (関係名 A BUNYA))))))

2.4 SUBJECT の格表現について

2.4.1 単一概念

構文 “A” (1つの名詞)

格表現

(subject A)

2.4.2 限定表現

目的の限定

入力構文：Aを目的としたB, AのためのB, A用のB

格表現 (subject (目的本体 B A))

生成文：AのためのB

手段の限定

入力構文：Aを用いたB, Aを使ったB, Aに基づくB, AによるB, AのあるB

格表現 (subject (手段本体 B A))

生成文：Aを用いたB

対象の限定

入力構文：AにおけるB, AのB, AでのB, AとのB

格表現 (subject (対象本体 B A))

生成文：AのB

因果関係

入力構文：Aを原因とするB

格表現 (subject (因果結果 B A))

生成文：Aを原因とするB

階層関係 (階層上位)

構文：AなどのB (例: 定性推論などの推論方式), Aの上位のB

格表現 (subject (階層上位 B A))

生成文：AなどのB

階層関係 (階層下位)

構文：Aの下位のB

格表現 (subject (階層下位 B A))

生成文：Aの下位のB

2.5 併記

1つの文中に複数の概念を記述する場合

構文：AとB, AおよびB, AまたはB, AあるいはB

格表現 (subject (併記 A B))

ここで A, Bは, それぞれ上記の単一概念, 限定表現または併記とする。

2.6 例

(例) フレームシステムによる文献検索のエキスパートシステムに関する文献が欲しい。

((action retrieval))

(object ((subject (手段本体
(対象本体 エキスパートシステム 文献検索) フレームシステム))))))

(例) 定性推論あるいは類推などの推論方式を使ったエキスパートシステムに関する文献が欲しい。

((action retrieval)
(object ((subject (手段本体 エキスパートシステム
(階層上位 推論方式
(併記 定性推論 類推))))))))))

(例) 情報検索システムでの知識表現と情報検索システムでの専門知識の獲得についての文献が欲しい。

((action retrieval)
(object
((subject (併記 (対象本体 知識表現 情報検索システム)
(対象本体 (対象本体 獲得 専門知識) 情報検索システム))))))

第 3 章

文献検索システム対話の方法

システムからの質問を自然言語で回答する時に使える構文を示す。

最終システムでは、回答を選択で行なう質問に対しては、マウスクリックによる選択を行なうので、ここで示す自然言語入力を行なわない。

1. 生成文 “あなたの名前を入力して下さい”

⇒ 「名前は<名前>です」

<名前>は漢字で表現する。処理が続く。

⇒ 「おわり」

プログラムの終了。

2. 生成文 “どのような文献をお探しですか？”

⇒ 「<関心のある項目>についての文献が欲しい」

<関心のある項目>の表現で用いることができる言葉

- (1) 任意の名詞
- (2) 別に定めた助詞

<関心のある項目>の中で使える助詞

対象の限定	における	において	に対する	への	の	での
手段の限定	による	を使った	を用いた	を使用した	に基づく	
目的の限定	のための	用の	を目的とした	に用いる	を目的とする	
因果関係	による	を原因とした	を原因とする			
階層関係	などの	を含む	の上位の			
	としての	特に	例えば	の下位の		
併記	と	や	あるいは	および	または	

(併記を意味する助詞を使用する場合の制約条件)

<1>併記を意味する助詞が結び付ける言葉は2つまでとする。

例 AとB

可

AとBとC

不可

<2>併記を意味する助詞が結び付ける言葉のうち的一方には他の助詞を含まないこと。

例 AにおけるBとC 可
AにおけるBとCにおけるD 不可

(この場合は要求が2つあるものとみなす。

要求が2つある場合、1回目の処理が終わった後、要求を付加することができる)

<3>併記以外の助詞が結び付ける言葉のうち的一方には併記を意味する助詞を含まないこと。

例 AとBにおけるC 可
AとBにおけるCとD 不可

(この場合は'AとBにおけるC'および'AとBにおけるD'という

2つの要求があるものとみなす。

要求が2つある場合、1回目の処理が終わった後、要求を付加することができる)

3. 生成文

「Aとは、

1. A 1

2. A 2

.....

n. A N

のうち、何番の意味ですか？

表示番号：該当語

0 : 該当無 入力 = 」

⇒ 番号の入力

Aと表現した言葉に対して同じ意味のもしくは関連する言葉が表示される。

もしこのなかに同じもしくは関連すると思う言葉があれば最適な物を1つ番号で選択する。

もしこのなかに同じもしくは関連すると思う言葉がなければ'0'を入力する。

4. 生成文

「Aには、

A 1

A 2

... A n

という下位の(関連する)概念がありますがこれを含みますか?」

または

「Aには、

A 1

A 2

... A n

を対象とする / 目的とする / 手段とする概念がありますがこれを対象 / 目的 / 手段として含みますか?」

⇒ このうち幾つかを意図するものとして含むとき

「A 1とA 2だけを含みます」

⇒ このうち幾つかを意図しないとき

「A 1 と A 2 は含みません」

⇒ 表示されたものを全て意図するものとして含むとき

「全て含みます」

⇒ 表示されたものは全て意図しないとき

あるいは全体の概念として検索を意図しており、限定したくない時

「どれも含みません」 「限定しません」

5. 生成文

「Aの意味を説明して下さい。

Aとは」

⇒ Aの意味を次の言い方で説明する、

「(Aとは) Xのことです。

Xに関係した / する言葉です。

Xに関連した / する言葉です。

Xの下位概念です。

Xの一部です。

X分野の言葉です。

Xの下位の言葉です。

Xの上位概念です。

Xから構成されています。

Xの上位の言葉です。」

Xの表現は<関心のある項目>の表現に準ずる。

6. 生成文

「選択するキーワードはこれでよろしいですか？

A 1

A 2

...

A N 」

⇒ 選択すべきキーワードが表示のとおりでよい時。

「はい」

⇒ 不要と思われるキーワードがある時。

「A 1 と A 2 は含みません」

⇒ これ以外にさらに欲しい概念がある時

「<関心のある項目>を含みます」

⇒ 表示されていることばやそれ以外の言葉についてさらに関連する言葉を思い付いた時。

5. での言い方と同じように説明を与える。

「Aとは.....」 「Aは.....」

7. 生成文 “ AはB<の上位概念／の下位概念／のこと／に関係した言葉>ですか?”

Aという言葉について、意味を尋ねている。

⇒ この文が正しい時。

「はい」

⇒ この文が正しくない時。

「いいえ」

「いいえ、Aは<の上位概念／の下位概念／のこと／に関係した言葉>です」

8.生成文 “出力ファイルのDirectoryに 指定がありますか ? (Y or N)”

⇒ N を入力する。

第 4 章

ユーザモデル構築の対話処理仕様書

Sun 上の ART を用いた、対話によるユーザモデルの構築の仕様を示す。文中、[] は状態名を示す。

なお、この章では、ユーザモデルにおける意味構造識別子は考慮していない。

最終システムでは、回答を選択で行なう質問に対しては、マウスクリックによる選択を行なうので、ここで示す自然言語入力を行なわない。

4.1 名前の入力

[名前を入力待ち]

生成文 “あなたの名前を入力して下さい”

入力文 “名前は<名前>です”

格表現 ((input simei) NAME)) (NAME は ローマ字)

処理

if NAME と一致する UM がある

⇒ NAME と一致する UM (シーケンス番号の最新のもの) をロードする。

(name_user_modeln.dat) n はシーケンス番号。

if NAME と一致する UM がない

⇒ 典型的対象世界知識をロードする。

入力文 “おわり”

処理 プログラムの終了。

4.2 検索要求文の入力

[検索要求文入力待ち]

生成文 “どのような文献をお探しですか？”

入力文 “SUBJECT についての文献が欲しい”

格表現

((action retrieval)
(object ((subject SUBJECT))))

処理 SUBJECT と UM のノードとの照合を行う。

[照合開始状態] ⇒ 照合の節を参照

4.3 照合規則

[照合開始状態]

格表現内に含まれる各概念とUMのノードの照合を行う。

以下の処理の結果,[照合確定状態]となる。

格表現

```
((action retrieval)
 (object ((subject SUBJECT))))
```

SUBJECT は単一の場合と (関係名 SUBJECTa SUBJECTb) なる構造を持つ場合がある。

ここで SUBJECTa を 構造内の 本体の概念 (LHS)

SUBJECTb を 構造内の 対象の概念 (RHS) と呼ぶ。

関係名について

対象世界知識において 関係の種類は

(階層上位, 階層下位) (関連関係, 関連関係) (同義関係, 同義関係)

(対象本体, 対象対象) (目的本体, 目的対象) (手段本体, 手段対象) (因果結果, 因果原因)

(併記, 併記)

がある。以下の処理で関係の名前のみが異なる処理を1つで代表させて記述する場合には, inverse-relation に注意すること。

[照合確定状態]での処理

[照合確定状態]となった概念に対しては, 前に特別の指示がない限り

活性スロット値を活性にする。

トレーススロット値を活性にする。

重要度スロットをインクリメントする。

4.3.1 単一概念同士の照合 [概念同士の照合]

SUBJECT が単一の場合 A である場合 (関係構造を持たない場合)

次の場合に, 単一の場合 A がUMのノード UMN に照合されたとする。

概念 A がストリングとしてUMのノード UMN に一致する。 <完全照合>

概念 A がUMのノード UMN のサブストリングに一致する。 <準完全照合>

概念 A のサブストリングがUMのノード UMN に一致する。 <準完全照合>

その他の場合 <失敗> ⇒ [未知語獲得]

1. <完全照合>

概念 A がストリングとしてUMのノード UMN に一致した。

⇒ [照合確定状態]

2. <準完全照合> [準完全の照合]

2.1 [概念の一意照合]

if (<準完全>) で照合された概念がただ1つもしくは (<準完全>) で照合された概念で重要度スロット値が最大 (これを M とする) の概念 UMN がただ1つ存在する。

⇒

利用者へ確認する.

生成文

「a とは,

1. umn

のうち, 何番の意味ですか

表示番号: 該当語

0 : 該当無 入力 = 」

if 入力: '1' のとき

⇒ a スキーマを独自語として定義する.

a スキーマと umn スキーマ間に同義関係を設定する.

([未知語獲得] の処理に準づる.)

⇒ [照合確定状態]

if 入力: '0' のとき

⇒ [未知語獲得] 処理によって a スキーマに新たな説明を求める.

⇒ [照合確定状態]

2.2.[概念の照合結果複数]

if (<準完全>) で照合された概念であって重要度スロット値が最大(これを m とする)の概念が複数存在する.

⇒

利用者に重要度スロット値にかかわらず総てのサブストリングマッチングできた概念 umn を示して問い合わせを行い, 選択させる.

生成文

「a とは,

1. umn1

2. umn2

...

n. umnn

のうち, 何番の意味ですか ?

表示番号: 該当語

0 : 該当無 入力 = 」

if 入力: '0 以外の番号' のとき

⇒

a スキーマを独自語として定義する.

番号で指定された umn スキーマと a スキーマ間に同義関係を設定する.

([未知語獲得] 処理に準づる)

⇒ [照合確定状態]

if 入力: '0' のとき

⇒ [未知語獲得] 処理によって a スキーマに新たな説明を求める.

⇒ [照合確定状態]

3.[照合失敗]

如何なる umn とも照合できなかった場合

⇒ [未知語獲得]

⇒ [照合確定状態]

4.[例外処理 1: 痕跡関係の処理]

if

(照合処理を行う間に UMにおいて jalias=umn なるスキーマに対して上記照合処理が成功 [照合確定状態])

and umn スキーマの階層関係 or 同義関係 or 関連関係の value スロットの重みとして痕跡関係が存在する.)

or

(推論ルールの途中でスキーマの階層関係 or 同義関係 or 関連関係の value スロットの重みとして痕跡関係が存在する.)

```
(defschema umn
  (関係名 co-schema)
  (value (関係名 co-schema 痕跡)))
```

⇒

痕跡関係の存在を利用者に確認する.

生成文 “umn は co-schema <の上位概念/の下位概念/のこと/に関係した言葉>ですか?”

(< > は関係名<階層下位/階層上位/同義関係/関連関係>に対応して生成する)

if 入力文 “はい” のとき

格表現

```
((action answer)
 (response positive))
```

⇒ umn スキーマの (value (関係名 co-schema 痕跡)) を (関係名 co-schema 意図する) に修正する.

if 入力文: “いいえ, a は bunya 分野の言葉です” など未知語獲得表現を付加したとき

格表現

```
((action answer)
 ((response negative))
 ((action explain)
 (object ((subject (関係名 a bunya))))))
```

⇒ 格表現内の概念 a と UMの (j-alias) 名 umn とのズtringマッチングは失敗とする.

⇒ a に対して [未知語獲得]

if 入力文 “いいえ” のとき

格表現

```
((action answer)
```

(response positive))

⇒ a に対して [未知語獲得]

4.3.2 構造をもつ概念の照合 [構造同士の照合]

subject 内に含まれる概念が、2つの概念とその関係 (関係名 B A) で与えられる場合は、ボトムアップ的に内部の構造から初めて、[概念同士の照合]を繰り返して行なう。

A = 単一概念 — (関係名 単一概念 A) | (関係名 A 単一概念)

B = 単一概念 — (関係名 単一概念 A) | (関係名 A 単一概念)

各概念の照合結果が完全/準完全/失敗

関係構造が対応/非対応

重要度による候補が一意/複数

によって場合分けを行い、それぞれの場合について処理を記述する。

番号	概念の照合		関係構造	準完全概念の候補数		処理の概要
a-a	完全	完全	対応	概念完全より一意に決定		[照合確定状態]
a-b	完全	完全	非対応	概念完全より一意に決定		[関係構造の構築]
b-a	完全	準完全	対応	一意		[概念の一意照合]
b-b	完全	準完全	対応	複数		[概念の照合結果複数]
b-c	完全	準完全	非対応	一意		[概念の一意照合] [関係構造の構築]
b-d	完全	準完全	非対応	複数		[概念の照合結果複数] [関係構造の構築]
c-a	準完全	準完全	対応	一意	一意	[概念の一意照合] 2回
c-b	準完全	準完全	対応	一意	複数	[概念の一意照合] [概念の照合結果複数]
c-c	準完全	準完全	対応	複数	複数	[概念の照合結果複数] 2回
c-d	準完全	準完全	非対応	一意	一意	[概念の一意照合] 2回 [関係構造の構築]
c-e	準完全	準完全	非対応	一意	複数	[概念の一意照合] [概念の照合結果複数] [関係構造の構築]
c-f	準完全	準完全	非対応	複数	複数	[概念の照合結果複数] 2回 [関係構造の構築]

d	完全	失敗	対応しない	概念完全より一意に決定	[未知語獲得] [関係構造の構築]
e-a	準完全	失敗	対応しない	一意	[概念の一意照合] [未知語獲得] [関係構造の構築]
e-b	準完全	失敗	対応しない	複数	[概念の照合結果複数] [未知語獲得] [関係構造の構築]
f	失敗	失敗	対応しない	不能	[未知語獲得] 2回 [関係構造の構築]

a. [完全完全照合]

BとAが共にUMのノードと完全にストリングマッチングする時

a-a. 関係構造が対応する時

⇒ [照合確定状態]

(UMにはまったく同一の概念のペアが2箇所にはないことを仮定する)

a-b. 関係構造が対応しない時

⇒ 概念Bと概念Aに対して [関係構造の構築]

⇒ [照合確定状態]

b. [完全準完全照合]

BとAの一方がサブストリングマッチングし、

他方がストリングマッチングする時

(サブストリングマッチングした概念をX (XはBまたはA) とする。

ストリングマッチングした概念をY (YはBまたはA) とする.)

if サブストリングマッチングした概念Xがただ1つ存在もしくは

重要度スロット値が最大(これをmとする)の概念ただ1つ存在する。

⇒ サブストリングマッチングした概念X (XはBまたはA) に対して

[概念の一意照合]

⇒ [照合確定状態]

if サブストリングマッチングした概念Xのうち

重要度スロット値が最大(これをmとする)の概念が複数存在する。

⇒ サブストリングマッチングした概念X (XはBまたはA) に対して

[概念の照合結果複数]

⇒ [照合確定状態]

[照合確定状態] となった概念Xと完全照合した概念Yの間の関係名が
parser 出力の関係名と一致する場合

⇒ [照合確定状態]

[照合確定状態] となった概念Xと完全照合した概念Yの関係名が parser 出力の関係名と一致しない場合

⇒ 概念Xと概念Yに対して [関係構造の構築]

⇒ [照合確定状態]

c.[準完全準完全照合]

BとAの両方ともサブstringマッチングした場合

⇒ b.[完全準完全照合] と同じ

但し b.[完全準完全照合] の処理における準完全に対する処理を2回行う。

d.[完全失敗照合]

BとAの一方のみがstringマッチング、他方が失敗のとき

⇒ stringマッチングが成立しない概念に対して [未知語の獲得]

⇒ stringマッチングが成立した概念と獲得された概念の間に [関係構造の構築]

⇒ [照合確定状態]

e.[準完全失敗照合]

BとAの一方のみがサブstringマッチング、他方が失敗のとき

e-a. サブstringマッチングした概念のうち、重要度スロットの値 m が最大となる概念Xがただ1つある。

⇒ 概念X (XはBまたはA) に対して [概念の一意照合]

⇒ サブstringマッチングが成立しない概念に対して [未知語獲得]

⇒ 概念Xと獲得した概念の間に [関係構造の構築]

⇒ [照合確定状態]

e-b. サブstringマッチングした概念のうち、重要度スロットの値 m が最大となる概念Xが複数ある。

⇒ 概念X (XはBまたはA) に対して [概念の照合結果複数]

⇒ 失敗の概念に対して [未知語獲得]

⇒ 概念Xと獲得した概念の間に [関係構造の構築]

⇒ [照合確定状態]

f.[照合不可状態]

BとAのいずれもがUMのノードと照合できない場合

⇒ 概念Bに対して [未知語獲得]

⇒ 概念Aに対して [未知語獲得]

⇒ 概念Bと概念Aの間に [関係構造の構築]

⇒ [照合確定状態]

4.4 ユーザモデル操作のルール

4.4.1 意図の重みの修正

任意の時点で、意図されるとして活性化されたすべてのノードA (活性スロット=on) に対して、初期状態のレベルの階層下位 or 関連関係 or 対象対象 or 目的対象関係で接続されている関連ノードがある。

⇒ 初期状態のレベルの階層下位 or 関連関係 or 対象対象 or 目的対象 or 手段対象関係で接続されているノードを利用者に順に提示して、その概念を意図しているかどうかを問い合わせ、各関係リンクのレベルを“意図している”もしくは“意図していない”のどちらかに決定する。

ノードAは次の場合に活性化される。

<1>照合確定によって活性フラグがonとなる。

<2>意図の重みの修正のルール(このルールが再帰的に用いられて)活性フラグがonとなる。

<3>別に定める推論ルールによって活性フラグがonとなる。

a. ノードAにA1, A2,,, Anというスキーマが初期状態レベルの階層下位関係で接続されている場合の処理。

生成文 “Aには、

A1

A2

... An

という下位の概念がありますがこれを含みますか?”

a-1. 入力文 “A1とA2だけを含みます”の時

格表現 ((action answer)

(select intention)

(selected (A1 A2)))

処理 AとA1の階層関係のレベルを“意図する”に修正する。

AとA2の階層関係のレベルを“意図する”に修正する。

AとA3の階層関係のレベルを“意図しない”に修正する。

.....

AとAnの階層関係のレベルを“意図しない”に修正する。

a-2. 入力文 “全て含みます”の時

格表現 ((action answer)

(select intention)

(selected all))

処理 AとAiの階層関係のレベルを“意図する”に修正する。(i=1 n)

a-3. 入力文 “A1とA2は含みません”の時

格表現 ((action answer)

(select intention)

(denied (A1 A2)))

処理 A と A 1 の階層関係のレベルを“意図しない”に修正する。
A と A 2 の階層関係のレベルを“意図しない”に修正する。
A と A 3 の階層関係のレベルを“意図する”に修正する。
.....
A と A n の階層関係のレベルを“意図する”に修正する。

a-4. 入力文 “どれも含みません” “限定しません”
“全体の概念が欲しい”の時

格表現 ((action answer)
(select intention)
(denied all))

処理 A と A i の階層関係のレベルを“意図しない”に修正する。 (i=1 n)

b. ノード A に A 1, A 2 ... A n というスキーマが初期状態レベルの関連関係で接続されている場合の処理。

生成文 “A には,
A 1
A 2
... A n
という関連する概念がありますがこれを含みますか?”

b-1. 入力文 “A 1 と A 2 だけを含みます”の時

格表現 ((action answer)
(select intention)
(selected (A 1 A 2)))

処理 A と A 1 の関連関係のレベルを“意図する”に修正する。
A と A 2 の関連関係のレベルを“意図する”に修正する。
A と A 3 の関連関係のレベルを“意図しない”に修正する。
.....
A と A n の関連関係のレベルを“意図しない”に修正する。

b-2. - b-4.

条件, 処理ともに 階層関係の場合と同様とする。
但し 関係名を関連関係とする。

c. ノード A に A 1, A 2 ... A n というスキーマが初期状態レベルの (対象, 目的, 手段) 対象で接続されている場合の処理。

生成文 “A には,
A 1
A 2
... A n
を対象とする概念がありますがこれを対象として含みますか?”
(目的とする概念がありますがこれを目的として含みますか?)

(手段とする概念がありますがこれを手段として含みますか?)

条件, 処理ともに 階層関係の場合と同様とする.
但し 関係名を対象, 目的, 手段) 対象とする.

4.4.2 未知語の獲得

未知語とはUMに照合できるノードがなかった概念をさす.

if (利用者の表現 (則ち格表現) 内に未知語がある.)

or

((action explain)

(object ((subject (関係名 MICHIGO A))))))

がすでに入力されていて関係名=階層関係 or 関連関係 or 同義関係である.)

⇒ <処理の流れ>

(1) 利用者によるその言葉 MICHIGO の説明を求める.

未知語の言葉の説明によってできる関係リンクは階層関係, 関連関係, 同義関係である.

(この処理を以下の [未知語獲得] に示す)

(2) さらに, もし, この独自語が限定関係を含む表現内で使用されていれば,

対応する限定関係リンクも付与する.

([関係構造の構築] の処理)

(例) MICHIGO における A

((action retrieval)

(object ((subject (対象本体 A MICHIGO))))))

の時

[1][未知語獲得] によって MICHIGO スキーマの階層関係 or 関連関係

or 同義関係を設定する.

[2] の関係名に応じて [関係構造の構築] を行う.

[未知語獲得] (この処理 [未知語獲得] は照合過程または最終確認状態のなかで呼び出される.)

生成文 “MICHIGO の意味を説明して下さい.

MICHIGO とは”

但し, (action explain)

(object ((subject (関係名 1 MICHIGO A))))))

がすでに入力されている場合はこの生成文は不要.

(1) 階層関係

入力文:

“(MICHIGO とは)A の下位概念です”

“(MICHIGO とは)A の一部です”

“(MICHIGO とは)A 分野の言葉です”

格表現

((action explain)

(object ((subject (階層下位 MICHIGO A))))

処理:

MICHIGO スキーマを定義する.

活性スロット = ON

重要度スロット = 1

A スキーマに階層下位リンクを付加し MICHIGO スキーマと接続する.

A スキーマの value スロットを (階層下位 MICHIGO “意図する”) にする.

MICHIGO スキーマの value スロットを (階層上位 A “意図する”) にする.

MICHIGO スキーマの DK スロット = A スキーマの DK スロット + 1

もし A スキーマが関係構造を持つ場合は A として A の本体の概念を用いる.

入力文:

“(MICHIGO とは)A の上位概念です. ”

“(MICHIGO は)A から構成されています. ”

格表現:

((action explain)

(object ((subject (階層上位 MICHIGO A))))

処理:

関係名は階層上位として処理は階層下位の場合と同一.

(2) 同義関係

入力文: (MICHIGO とは)A のことです.

格表現:

((action explain)

(object ((subject (同義関係 MICHIGO A))))

処理:

関係名は同義関係として処理は階層下位の場合と同一.

(3) 関連関係

入力文: (MICHIGO とは)A に関係した言葉です.

格表現:

((action explain)

(object ((subject (関連関係 MICHIGO A))))

処理:

関係名は関連関係処理は階層下位の場合と同一.

4.4.3 関係構造の構築 [関係構造の構築]

<処理の概要>

照合された 2 つの概念 B と A に対して関係を明言した.(関係名 B A)

⇒ その 2 つの概念の間に所定の関係リンクを付加する.

1 LHS,RHS が単一概念の時 (関係 1 A B))

2 LHS が単一概念,RHS が構造を持つ時 (関係 1 A (関係 2 B C))

- 3 LHS が構造を持つ,RHS が単一概念の時 (関係1 (関係2 A B) C)
 4 LHS,RHS が構造を持つ時 (関係1 (関係2 A B)(関係3 C D))

1 LHS,RHS が単一概念の時 [単一単一関係構造の構築]

目的の限定

構文: Aを目的としたB, AのためのB, Aに用いるB, A用のB

格表現 (subject (目的本体 B A))

処理: 目的限定関係を構築する.

(A, B共に完全照合の時)

A, B共に [照合確定状態]

(Aが完全照合, Bが完全照合でない時)

Bに対して照合処理

(Aが完全照合でない, Bが完全照合の時)

Aに対して照合処理

(A, B共に完全照合でない時)

Bに対して照合処理

Aに対して照合処理

スキーマAの目的本体スロットにスキーマ名Bを代入する

スキーマAの value スロットを(目的本体 B “意図する”)にする.

スキーマBの value スロットを(目的対象 A “意図する”)にする.

(この処理は以下の場合も同様)

手段の限定

構文: Aを用いたB, に基づく, による, を使った, を使用した

格表現 (subject (手段本体 B A))

処理: 手段限定関係を構築する.

スキーマAの手段本体スロットの値にスキーマ名Bを代入する.

スキーマAの value スロットを(手段本体 B “意図する”)にする.

スキーマBの value スロットを(手段対象 A “意図する”)にする.

対象の限定

構文: AにおけるB, AのB, での, において, に対する, への

格表現 (subject (対象本体 B A))

処理: 対象限定関係を構築する.

スキーマAの対象本体スロットの値にスキーマ名Bを代入する.

スキーマAの value スロットを(対象本体 B “意図する”)にする.

スキーマBの value スロットを(対象対象 A “意図する”)にする.

因果関係

構文: Aを原因とするB, Aを原因としたB, AによるB

格表現 (subject (因果結果 B A))

処理: 因果関係を構築する.

スキーマAの因果結果スロットの値にスキーマ名Bを代入する。
スキーマAの value スロットを(階層上位 B “意図する”)にする。
スキーマBの value スロットを(階層下位 A “意図する”)にする。

階層関係(階層上位)

構文:(Bは, Aよりも上位にあるとする) Aの上位のB, などの, を含む
格表現 (subject (階層上位 B A))

処理: 階層関係を構築する。

スキーマAの階層上位スロットの値にスキーマ名Bを代入する。
スキーマAの value スロットを(階層上位 B “意図する”)にする。
スキーマBの value スロットを(階層下位 A “意図する”)にする。

階層関係(階層下位)

構文:(Bは, Aよりも下位にあるとする) Aの下位のB, AとしてのB, A特にB, A例えばB
格表現 (subject (階層下位 B A))

処理: 階層関係を構築する。

スキーマAの階層下位スロットの値にスキーマ名Bを代入する。
スキーマAの value スロットを(階層下位 B “意図する”)にする。
スキーマBの value スロットを(階層上位 A “意図する”)にする。

2 LHS が単一概念,RHS が構造を持つ時 (関係1 A (関係2 B C))

- a. (関係2 B C) に対して [単一単一関係構造の構築]
- b. (関係2 B C) をBとみなし,(関係1 A B) に対して [単一単一関係構造の構築]

但し 関係2 が併記の場合は,

(関係1 A C) に対して [単一単一関係構造の構築] を行う。

3 LHS が構造を持つ,RHS が単一概念の時 (関係1 (関係2 A B) C)

- a. (関係2 A B) に対して [単一単一関係構造の構築]
- b. (関係2 A B) をAとみなし,(関係1 A B) に対して [単一単一関係構造の構築]

但し 関係2 が併記の場合は,

(関係1 B C) に対して [単一単一関係構造の構築] を行う。

;;;4 LHS,RHS が構造を持つ時 (関係1 (関係2 A B)(関係3 C D))

- ;;; a. (関係2 A B) に対して [単一単一関係構造の構築]
- ;;; b. (関係3 C D) に対して [単一単一関係構造の構築]
- ;;; c. (関係2 A B) をA,(関係3 C D) をCとみなし,
;;; (関係1 A C) に対して [単一単一関係構造の構築]
- ;;; このような関係となるのは関係1が併記の場合に限られる。
- ;;; 最初の Version でこのような処理は扱わない。

4.5 システムの理解内容の確認と構造の修正

格表現に表れた概念とUMとの照合が総て完了し、かつ、別に定めた推論処理が終了して活性化されたスキーマの活性化スロットがすべてONとなった。

⇒

活性化スロットがONであるスキーマをのJ-aliasによって表示する。

生成文 “ 選択するキーワードはこれでよろしいですか？ ”

[最終確認状態] の処理

(1) 入力文 “ はい ” の時

格表現

((action answer)

(response positive))

処理

1. 対話毎の活性キーワードスキーマおよび [最終確認状態] での活性キーワードスキーマを評価ファイル (key_wordn.dat) に書き出す。
(評価ファイルの生成仕様については、別に定める.)
2. 現状の対象世界知識を利用者のユーザモデルとしてファイルに書き出す。
(name_user_modeln.dat)
3. 活性化されているスキーマ名とそれらの間の関係構造を論理式合成部に出力する。

(2) 入力文 “ いいえ ” の時

格表現

((action answer)

(response negative))

処理

生成文 “ 他に何を含まれますかまたは何が不要ですか？ ”

[最終確認状態] の処理に戻る。

(3) 入力文 “ いいえ, A は含みません ” または
“ A は含みません ” の時

格表現

(((action answer)

(response negative))

((action answer)

(select intention)

(denied A)))

または

((action answer)

(select intention)

(denied A))

A は (併記 A B C ...) であってもよい。

処理

< 1 > A スキーマの活性化フラグを非活性とする.

< 2 >

A スキーマに接続されかつトレーススロット = ON であるスキーマ X に対して

< case1 > スキーマ X がキーワードの場合

A スキーマに接続されかつトレーススロット = ON であるスキーマ X に対して

(value スロット = 関係名 X 意図しない)

A スキーマに対して

(value スロット = 関係名 A 意図しない)

< case2 > スキーマ X が独自語の場合

独自語からの推論処理を逆に辿って, その間の推論上の全ての関係リンクの value を痕跡にする.

while(条件を満たすスキーマ X がなくなるまで)

A スキーマに接続されかつトレーススロット = ON であるスキーマ X に対して

(value スロット = 関係名 X 痕跡)

A スキーマに対して

(value スロット = 関係名 A 痕跡)

< 3 > [最終確認状態] の処理に戻る.

(4) 入力文 “いいえ, A を含みます” または

“A を含みます”

A は一般の構造を持つ

格表現

```
((action answer)
 (response negative))
((action answer)
 (select intention)
 (selected A)) )
```

または

```
((action answer)
 (select intention)
 (selected A ))
```

処理

照合規則によって A と UM のノードとの照合を行う.

(現在活性化されている状態に加えて

“A についての文献が欲しい” が入力されたものと同じ)

(5) 入力文が 未知語獲得表現であった場合

格表現 ((action explain)

(object ((subject (関係名 MICHIGO A))))))

処理

現在活性化されている状態に加えて [未知語獲得] 処理および意図推論処理を行う

(6) その他の入力の時

処理

生成文 “もう一度入力して下さい”

[最終確認の後] の処理に戻る.

4.6 全体の対話について共通

生成文に対して有効でない入力があった場合, その状態を変えずに再入力を促す.
生成文 “もう一度入力して下さい.”

第 5 章

真意の推論処理仕様書

ユーザモデルを利用した隠れた検索要求の推論は、構築されたユーザモデルの上で、構文解析された構造に対応する部分を出発点として、意図するリンクをトレースして対応するキーワードを抽出することである。この章では、意味構造識別子は考慮していない。

ユーザモデルをトレースするための規則を以下に記す。

関係リンクの種別ごとの推論における性質

関係リンクの種類		推論の方向	推論後に活性化されるノード
階層関係	独自語含む	(注1)	DK値が小さい方のノードだけ
意図している	キーワード間	上位語→下位語	両方のキーワード
関連関係	独自語含む	(注1)	DK値が小さい方のノードだけ
意図している	キーワード間	両方向 ←→	両方のキーワード
同義関係	独自語含む	(注1)	DK値が小さい方のノードだけ
	キーワード間	両方向 ←→	両方のキーワード
限定関係 意図している (注2)	独自語含む	両方向 ←→	両方のノード
	キーワード間	両方向 ←→	両方のキーワード
階層, 関連関係 限定関係 意図していない	独自語含む	推論しない	表明された方のノードのみ
	キーワード間		

DK = distance from keywords; キーワードからの距離
(キーワードから独自語へのパスを形成するリンクの数)

Wi = 連想の強さを表す係数

(注1) 独自語→キーワード

(DK値が大きなノード→DK値が小さいノード)

(注2) 限定関係(意図している)については、そのリンクについているノードの種類がキーワードであっても独自語であっても順に活性化ノードが広がって行く。

独自語ノードには必ず階層or関連or同義関係によってキーワードにつながってゆくpathをもっているから、限定関係の活性化推論の後、階層or関連or同義関係の活性化推論を行えばよい。

活性化と推論のルール

1. 対象とする概念ノード

利用者が表現した言葉と構造の一部が一致した(照合が確定した)ことにより もしくはこの'活性化と推論のルール'によって活性化されたノード。

2. 活性化されたノードに隣接し、表1に示す『推論後に活性化されるノード』にあたるノードがある。

→『推論後に活性化されるノード』を活性化する。

3. 活性化されたノードが独自語である。(この規則は規則2.に含まれている)

→階層or関連or同義関係によって独自語からDK値の小さい独自語を活性化する。

(このことによって、必ず独自語からキーワードを抽出することが保証される)

4. 推論におけるループ性の排除

既に活性化されたノードに対しては推論を行わない。

キーワード抽出のルール

最終的に活性化された全てのノード集合のうちの、キーワードノードを、意図するキーワードとして抽出する。

第 6 章

複数の検索意図を反映するユーザモデルを用いた知的文献検索システム

6.1 基本概念

複数の検索意図を反映させるユーザモデルの表現方法と対話による構築方法およびその利用方法を提案する。

- 1つの検索意図の意味構造ごとに意図の識別子を付与することによって意味構造を検索意図ごとに区別できるようにする。
- 検索要求が入力されたときに、これがどの検索意図に対応するかを判断し、その意図の識別子が付与された意味構造を積極的に利用することによって、他の検索意図の意味構造による干渉を防止する。
- 必要ならば他の意図の意味構造を利用することによってユーザモデルの構築操作を簡略化する。

6.2 処理の流れ

1. 検索要求文に対する照合と基本意味構造の生成

検索要求文の構造とユーザモデルの構造の照合により、検索要求文に対応するユーザモデル上での意味構造の位置を決定する。

この時検索要求文に独自語があればその意味を対話により獲得する。また準完全な単語はその意味を同定する。

新しく構築されたリンクとノードには新しい意図の番号を付与する。この結果できた意味構造を基本意味構造という。

2. 検索意図の識別

基本意味構造が既存のどの検索意図に対応するか、則ち識別番号(識別子)がどれかを決定する。決定された識別子を現在の識別子と呼ぶ。

複数の候補がある場合や未だ識別子が与えられていない場合は新しい識別子が付与される。また複数の候補がある場合にはこれらを副識別子とする。

3. 副識別子に対する候補キーワードの決定

基本意味構造に連結され、かつ決定された副識別子が付与されたリンクのみを深層意図の推論規則に従ってトレースし、得られたキーワードのうち必要なキーワードを利用者に問い合わせる。必要な構造に対しては現在の識別子として新しい識別子が付与される。

4. 深層意図を表す意味構造の構築

基本意味構造内の概念ノードに対して、初期状態の下位、関連、限定のリンクによって結ばれている概念ノードがあるとき

これらの概念を利用者に表示してこれらの概念を意図しているかどうかを利用者に質問し、回答によりこれらのリンクに対して識別番号のついた重みを付与する。

この処理は、副識別子に対する候補キーワードの決定において示された全てのキーワードが否定された時、もしくは初期状態のリンクのみがついた概念ノードに対してのみ行う。この結果できた意味構造を構造化意味構造という。

5. 当該検索意図に対する深層意図の推論

構造化意味構造に連結され、かつ決定された現在の識別子が付与されたリンクのみを深層意図の推論規則に従ってトレースし、要求文で限定しなかった深層意味構造を得る。

この結果できた意味構造を拡張意味構造という。

6. 他の検索意図の意味構造の利用

拡張意味構造内のノードに対して、深層意図の推論条件を満たす他の識別子の意味構造が接続されているなら、この意味構造内のキーワード集合を意図するかどうかを質問し、必要なら現在の意味構造に取り込む。(深層意図の推論条件のうち限定リンクについてはその'本体ノード'から'限定ノード'の方向に限る)。

この結果できた意味構造を深層意味構造という。

7. 意味構造の精練化

対話によって、キーワードの削除や付加を行い最終的な深層意味構造を構築する。最終的な深層意味構造のリンクに現在の識別子を付与する。

6.3 ユーザモデルの構築と対話

6.3.1 ユーザモデル修正操作・オペレーションの定義

OP1. キーワード間の初期状態である階層、関連、限定リンクに、指定された意図の識別子および意図の重みの情報(意図する or 意図しない)を付与する。

OP2. 既存ノードに階層、関連、同義リンクによる新しい関係リンクと新しい独自語ノードを付加する。関係リンクには指定された意図の識別子および重みの情報(“意図する”)を付与する。新しい独自語ノードには、キーワードからの距離を示す情報を与える。

OP3. 既存ノード間に新しい関係リンクを付加する。関係リンクには指定された意図の識別子および重みの情報(意図する or 意図しない)を付与する。

OP4. 既存ノード間の現在の識別番号に対する重みを“痕跡”に修正する。

6.3.2 対話を行うタイミングとユーザモデル構築の処理

1. 未知語獲得

利用者の言った言葉が未知語であるとき

未知語の意味を質問する。

未知語を新たな独自語として作成し、利用者が答えた概念のノードとの間に新しい意図の識別子(最大の識別番号)をもった“意図する”重みのリンクを付加する。

(OP2 の操作)

2. 準完全語同定

利用者の言った言葉が準完全であるとき

準完全語の意味を同定する。

準完全語を新たな独自語として作成し、利用者が答えた概念のノードとの間に新しい意図の識別子(最大の識別番号)をもった“意図する”重みのリンクを付加する。

(OP2の操作)

3. 候補となるキーワードの選択

副識別子が付与されたリンクをトレースして得られた候補キーワードを示してこの中でどのキーワードを意図するかを質問する。

意図すると回答されたキーワードを活性化したリンクに新しい識別子と意図する重みを付与する。また意図しないと回答されたキーワードを活性化したリンクに新しい識別子と意図しない重みを付与する。

意図するとしたリンク構造が分離しないようにリンク構造を再合成する。

(OP3の操作)

4. 深層意図を表す意味構造の構築(下位・関連・限定概念の明確化)

(1) 基本意味構造内の概念ノードで、現在の識別子で重み付けられたリンクはなく、初期状態のレベルの下位、関連、限定のリンクによって結ばれている概念ノードがあるとき

これらの概念が検索意図に含まれるかどうかを質問し、回答によってリンクに識別子と重みの情報(意図する or 意図しない)を付加する。

(2) 基本意味構造内の概念ノードで、副識別子に対する候補キーワードの決定において示された全てのキーワードが否定され、下位、関連、限定のリンクによって結ばれている概念ノードがあるとき

これらの概念(否定されたキーワードを除く)が検索意図に含まれるかどうかを質問し、回答によってリンクに識別子と重みの情報(意図する or 意図しない)を付加する。

(OP1の操作)

5. 他の検索意図の意味構造の取り込み

拡張意味構造内の概念ノードに、深層意図の推論条件を満たす他の識別番号をもつリンクが連結されているとき

その概念ノードから他の識別番号による深層意図の推論規則によって活性化される概念構造を作り(但し、限定リンクによる活性化は、その‘本体ノード’から‘限定ノード’の方向に限る)、このなかにあるキーワード集合を表示してこれらを意図するかどうかを質問し、必要なら現在の意味構造に取り込む。

利用者が意図しているものとして取り込んだ意味構造上のリンクには新たに現在の識別子も付加しておく。

(OP3の操作)

6. 深層意味構造の精練化

深層意味構造から抽出したキーワードが正当かどうかを利用者に確認する。

回答により以下のように深層意味構造に対する精練化を行う。

- キーワードが適当であるとする時
対話を終了する。
- 不要なキーワードがあるとする時
不要なキーワードを活性化したリンクの現在の識別子に対する重みを“痕跡”に修正す

る。これによってリンク構造が分離しないように再合成する。

(OP4 の操作)

- さらに必要なキーワードがあるとする時
必要なキーワードを、現在の識別子のリンクにて検索要求文の基本概念に連結する。
(OP3 の操作)

6.4 検索意図の識別子の決定方法

基本意味構造に対して検索意図の識別子を決定し現在の識別子とする。

このとき用いる識別子の情報は、基本意味構造内のリンク上の識別子（'意図する'または“痕跡”の重みを持つもの）と、ノードに与えられている識別子の2種類である。ノードに与えられている識別子よりリンク上の識別子を優先する。

識別子が一意に決定できる時はこれを現在の識別子とし、一意に決定できないときは、新しい識別子を現在の識別子とすると共に、候補となる識別子を副識別子としてこれらに係るキーワードから必要なものの選択を行う。

6.4.1 基本意味構造が構造を持つとき

- (1) 基本意味構造に含まれるリンクおよびノードに新しい識別子のみを含む時
現在の識別子: 新しい識別子。
副識別子: なし。
- (2) 基本意味構造に含まれるリンクに既存の識別子のみが与えられている時
 - 基本意味構造に含まれるリンクに与えられた既存の識別子が唯一の時
現在の識別子: 既存の唯一の識別子。
副識別子: なし
 - 基本意味構造に含まれるリンクに与えられた既存の識別子が複数ある時
 - (a) 複数ある既存の識別子の構造が同一の時
現在の識別子: 複数ある既存の識別子のうちの最新のもの。
副識別子: なし
 - (b) 複数ある既存の識別子の構造が異なる場合
現在の識別子: 新しい識別子
副識別子: 既存の複数の識別子
- (3) 基本意味構造に含まれるリンクに既存の識別子と新しい識別子が与えられている時
現在の識別子: 新しい識別子。
副識別子: 既存の複数の識別子
- (4) 基本意味構造に含まれるリンクに新しい識別子のみが与えられ、ノードに既存の識別子が与えられている時
現在の識別子: 新しい識別子
副識別子: 既存の複数の識別子

6.4.2 基本意味構造が単一概念のとき

- (1) 概念ノードに識別子が与えられていない時。現在の識別子: 新しい識別子。
副識別子: なし。

表 6.1: 意図の決定方法; 構造を持つ場合

モード	リンク	ノード	現在の識別子	副識別子
A	新	新	新	なし
B	新	従	新	従
C	従1	-	従1	なし
D	従1 従2(従1 従2の構造は同じ)	-	従1 従2の新しい方	なし
E	従1 従2(従1 従2の構造は違う)	-	新	従1 従2
F	新 従1	-	従1	なし
G	新 従1 従2	-	新	従1 従2

表 6.2: 意図の決定方法; 単一概念の場合

モード	ノード	現在の識別子	副識別子
P	なし	新	なし
Q	従1	従1	なし
R	従1 従2	新	従1 従2

(2) 概念ノードに与えている識別子が既存の識別子である時

- 概念ノードに与えている既存の識別子が唯一のとき
現在の識別子: 既存の唯一の識別子.
副識別子: なし.
- 概念ノードに与えている既存の識別子が複数のとき
現在の識別子: 新しい識別子.
副識別子: 既存の複数の識別子

6.5 深層意図の推論

構造化意味構造を中心に、現在の識別子(若しくは副識別子)が付与された“意図する”重みを持つリンクをトレースして必要な概念を抽出する。

推論方向は以下の通り。

- (1) 上位キーワード ⇒ 下位キーワード (現在の識別子のみトレース)
- (2) 関連, 同義, 限定リンクは両方向に推論する. (現在の識別子のみトレース)
- (3) 独自語 ⇒ キーワード (識別子は任意のものを使用可)

優先順位は(1)(2)(3)の順。

第 7 章

検索式合成部

7.1 論理式合成部の機能

ユーザモデルにおける検索意図の意味構造を利用して、文献データベースをアクセスするための最適な論理式を合成する。

7.2 基本アーキテクチャ

- (1) ユーザの検索意図に対応する意味構造におけるリンクの意味および意図の強さを利用して論理式を合成する。
- (2) 最適な論理式で文献データベースをアクセスしたときに、対応する文献が少ない場合には、制約条件を緩和した論理式によって文献データベースをアクセスする。このようにして初めて1件以上の文献が得られたときの文献集合に対する適合率、再現率を計算する。制約条件を緩和したことは論理式に与えられる評点に反映する。

7.3 入力の形式

入力は、検索要求理解部で抽出したキーワードファイル、およびその時のユーザモデル（スキーマ）である。

7.4 出力ファイルの形式

7.4.1 出力ファイルの例

```
(t001 rel)
(taiwa-001)
(000 f001 (AND 7_10 1_19_3))
(100 f002 (OR 7_10 1_19_3))
(taiwa-002)
(000 f001 (AND 7_10 (OR 1_20 1_19_3)))
(100 f002 (OR 7_10 1_20 1_19_3))
(taiwa-003)
(000 f001 (AND 7_10 (OR 1_20 1_19_3_0)))
(010 f002 (AND 7_10 (OR 1_20 1_19_3 1_19_3_0)))
```

(110 f003 (OR 7_10 1_20 1_19_3 1_19_3_0))
 (taiwa-004)
 (000 f001 (AND 7_10 (OR 1_19_3_0 (AND 1_20 7_10))))
 (010 f002 (AND 7_10 (OR 1_19_3 1_19_3_0 (AND 1_20 7_10))))
 (110 f003 (AND 7_10 (OR 1_19_3 1_19_3_0 1_20 7_10)))
 (210 f004 (OR 1_19_3 1_19_3_0 1_20 7_10))
 (taiwa-005)
 (000 f001 (AND 7_10 (OR 1_20_0 1_19_3_0)))
 (010 f002 (AND 7_10 (OR 1_20_0 1_19_3_0 1_20)))
 (020 f003 (AND 7_10 (OR 1_20 1_20_0 1_19_3 1_19_3_0)))
 (120 f004 (OR 7_10 1_19_3 1_19_3_0 1_20 1_20_0))
 (taiwa-006)
 (000 f001 (AND 7_10 (OR 1_20_0 1_19_3_0)))
 (010 f002 (AND 7_10 (OR 1_20_0 1_19_3_0 1_20)))
 (020 f003 (AND 7_10 (OR 1_20 1_20_0 1_19_3 1_19_3_0)))
 (120 f004 (OR 7_10 1_19_3 1_19_3_0 1_20 1_20_0))

7.4.2 出力ファイルのフォーマット

- 1 レコード目
 (テストケース No. 関係情報使用有無)
 テストケース No. : 4 バイトの文字
 関係情報使用有無 : 'rel'(関係情報使用有)、'non_rel'(関係情報使用無)
- 2 レコード目
 (taiwa-XXX) : 対話番号
- 3 レコード目
 (評点 検索式 No. 論理式) の並び
 評点 : 3 桁の数字 (max 000)
 検索式 No. : 4 バイトの文字
 論理式 : LISP の S 式 (プレフィックス形式) で記述された論理式。演算子 (and/or) のアークギュメントには、キーワードとして対象分野知識の階層 No. がくる。
- 以下、対話回数に応じて 2 レコード目と 3 コード目の繰り返し

7.5 最適論理式の合成方法

7.5.1 有効グラフの作成

論理式のための構造は、独自語とこれに係るリンクを除き、意図の重みが、“意図する”および“意図しない”をもつリンクからなる構造に対して、真意の推論時での推論の方向で決定される方向性を持つリンクを有する有効グラフで表現する。

(“意図しない”リンクについては、「すでに活性化した意図するキーワード → 意図しないキーワード」の方向とする。)

具体的には、次の処理を行う。

- 最適検索式の合成に使用する意味構造の抽出
真意理解部が出力した意味構造から、次のものを抽出して新たな構造を作成する。
 - 推論処理部がトレースした（活性化フラグがオンまたはトレース）キーワードノード
 - 推論処理部がトレースした（活性化フラグがオンまたはトレース）キーワードノード間のリンク

このとき、下に示す規則により各リンクを有向リンクと解釈して、有向グラフとして表現する。

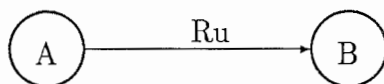
- 1) 階層リンクは、階層上位（始点ノード）から階層下位（終点ノード）への有向リンク
- 2) 限定リンクは、本体（始点ノード）から対象（終点ノード）への有向リンク
- 3) 関連リンクは、真意理解部での推論の方向（最初に活性化したノードから後に活性化したノードへの有向リンク）

7.5.2 合成規則

以下の合成規則を用いて、意味構造の有効グラフをトップダウンで（基本意味構造を出発点として）、論理式に置き換えて行く。

規則1: 各リンク種別に対する基本変換規則

意図する階層関係	下位概念のみを与える
意図する関連関係	OR で結合
意図する限定関係	AND で結合
意図しない階層関係	上位概念と、下位概念の否定を AND で結合
意図しない関連関係	意図しないキーワードは無視する
意図しない限定関係	意図しないキーワードは無視する



関係 Ru が意図する階層関係のとき	B
関係 Ru が意図する関連関係のとき	A+B
関係 Ru が意図する限定関係のとき	A*B
関係 Ru が意図しない階層関係のとき	A*¬B
関係 Ru が意図しない関連関係のとき	A

R1!=R2 のとき (規則 3)

		R2			
		意図する・階層	意図する・関連	意図する・限定	意図しない・階層
R1	意図する・階層	$B+C$	$B+C$	$B*C$	$B*\neg C$
	意図する・関連	$C+B$	$A+B+C$	$(A+B)*C$	$(A+B)*\neg C$
	意図する・限定	$C*B$	$(A+C)*B$	$A*B+A*C$	$A*B*\neg C$
	意図しない・階層	$C*\neg B$	$(A+C)*\neg B$	$A*C*\neg B$	$A*\neg B*\neg C$

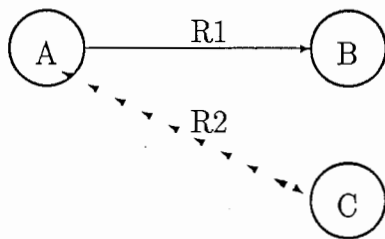
関係 Ru が意図しない限定関係のとき A

規則 2: 同一種類の複数の出力リンクをもつノードに対する変換規則

規則 1 によって得られた論理式の OR 結合を与える. 但し、意図しない階層のときは、意図しないノードの OR 結合の否定との AND 結合とする。

下図 R1=R2 のとき (規則 2)

関係 R1 が意図する階層のとき $B+C$
 関係 R1 が意図する関連のとき $(A+B)+(A+C) = A+B+C$
 関係 R1 が意図する限定のとき $(A*B)+(A*C) = A*(B+C)$
 関係 R1 が意図しない階層のとき $A*\neg(B+C) = A*\neg B*\neg C$

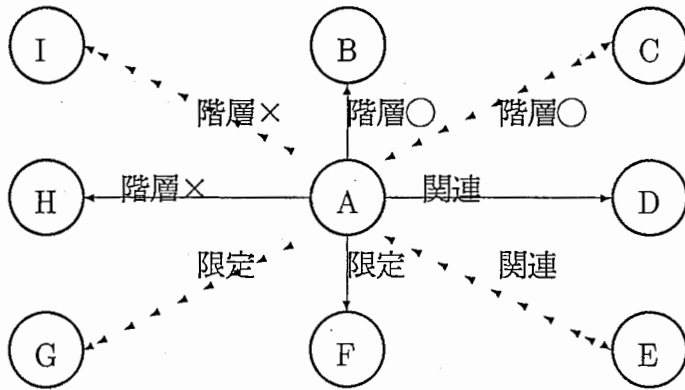


規則 3: 異なる種類の出力リンクをもつノードに対する変換規則

同一種類の出力リンクをグループ化する。各グループの中で、規則 2 を適用して論理式を作る。各グループでできた論理式を、意図する・階層関係、関連関係、限定関係、意図しない・階層関係の順に、規則 1 を適用して論理式を置き換えていく。

規則 4 多段のリンクがあるとき

- (1) まず、同一レベルにあるリンクについて処理を行なう。
- (2) さらにリンクがつながっているノードに対しては、その先のリンクに対して変換処理を行ない、この結果を、その (1) での式のノード名と置き換える。



例 図のような場合、論理式は

$$((B+C)+(D+E))*(F+G)*\neg(H+I) = (B+C+D+E)*(F+G)*\neg H*\neg I$$

となる。

7.6 条件の緩和規則

(1) 規則 1

意味構造上において、最も基本意味構造から遠い構造 (論理式において最も括弧の入れ子が深いレベル) に対して、意図しない階層リンクの変換規則は、上位のキーワードのみを与える。この規則は、これまで得られた論理式のうち対応する部分を、 $A*\neg B \rightarrow A$ に置き換えることである。

低い階層から順に処理を行ない、最終的に基本意味構造における階層リンクに対応したところで処理を終える。

(2) 規則 2

意味構造上において、最も基本意味構造から遠い構造 (論理式において最も括弧の入れ子が深いレベル) に対して、意図する階層リンクの変換規則を OR 結合とする。

この規則は、これまで得られた論理式のうち対応する部分を、 $B \rightarrow (A+B)$ に置き換えることである。

低い階層から順に処理を行ない、最終的に基本意味構造における階層リンクに対応したところで処理を終える。

(3) 規則 3

意味構造上において、最も基本意味構造から遠い構造 (論理式において最も括弧の入れ子が深いレベル) に対して、限定リンクの変換規則を OR 結合とする。

この規則は、これまで得られた論理式のうち対応する部分を、 $A*B \rightarrow A+B$ に置き換えることである。

意味構造の最も外側から順に処理を行ない、最終的に基本意味構造における限定リンクに適応したところで処理を終える。

(全ての論理式が OR 結合となる。)

処理順位は、規則1 → 規則2 → 規則3 とする。

評点の付加

- (1) 規則1 に対して
緩和規則を1つ適用して、ある階層の処理を終えるごとに評点(初期値)000の第1桁目に1を加える。
- (2) 規則2 に対して
緩和規則を1つ適用するごとに評点の第2桁目に1を加える。
- (3) 規則3 に対して
緩和規則を1つ適用するごとに評点の第3桁目に1を加える。

第 8 章

フィードバック付き文献検索システム仕様書

8.1 フィードバック付き文献検索システムの概要

利用者の問い合わせ文を理解し、それに基づいて文献データベースを検索する。システム全体として、以下の4つのフェーズで処理を行なう。

(フェーズ1) 問い合わせ意味構造の作成

問い合わせ文をユーザモデルに照応し、キーワードによる意味構造を作成する。

(フェーズ2) 文献データベースの検索

意味構造を、キーワードによる論理式へ変換し、検索を実行する。
検索式合成部および検索実行部から成る。

(フェーズ3) 検索結果からの意図理解

検索結果を利用者に提示して、検索意図にあった文献を選択してもらう。
利用者が選択した文献から検索意図の抽出を行ない、意味構造を修正する。
抽出文献選択部およびユーザモデル修正部から成る。

(フェーズ4) フィードバック結果を利用した文献データベースの検索

(フェーズ3)で修正された意味構造を論理式へ変換し、検索を実行する。
利用者には、(フェーズ2)で示した文献以外の付加検索文献が表示される。
プログラムとしては、(フェーズ2)(フェーズ3)の各部に含まれる。
このフィードバックは、1回だけとする。

今回は、実時間でこれらの処理が行なえるようにする。

上記処理部のうち、ここでは、(1) 検索実行部 (2) 抽出文献選択部 (3) ユーザモデル修正部 を作成する。

図 8.1 にフィードバック付き文献検索システム入出力関係を示す。

(1) 検索実行部

検索式合成部で作られた論理式をもとに、文献データベースをアクセスして、抽出文献を得る。

これは、従来 ORACLE データベースを用いて行なっている機能である。実時間性を高めるために、C 言語で記述する。

(2) 抽出文献選択部

抽出された文献・キーワードを、評点をグループとして利用者に表示し、検索意図に適合した文献を選択させる。

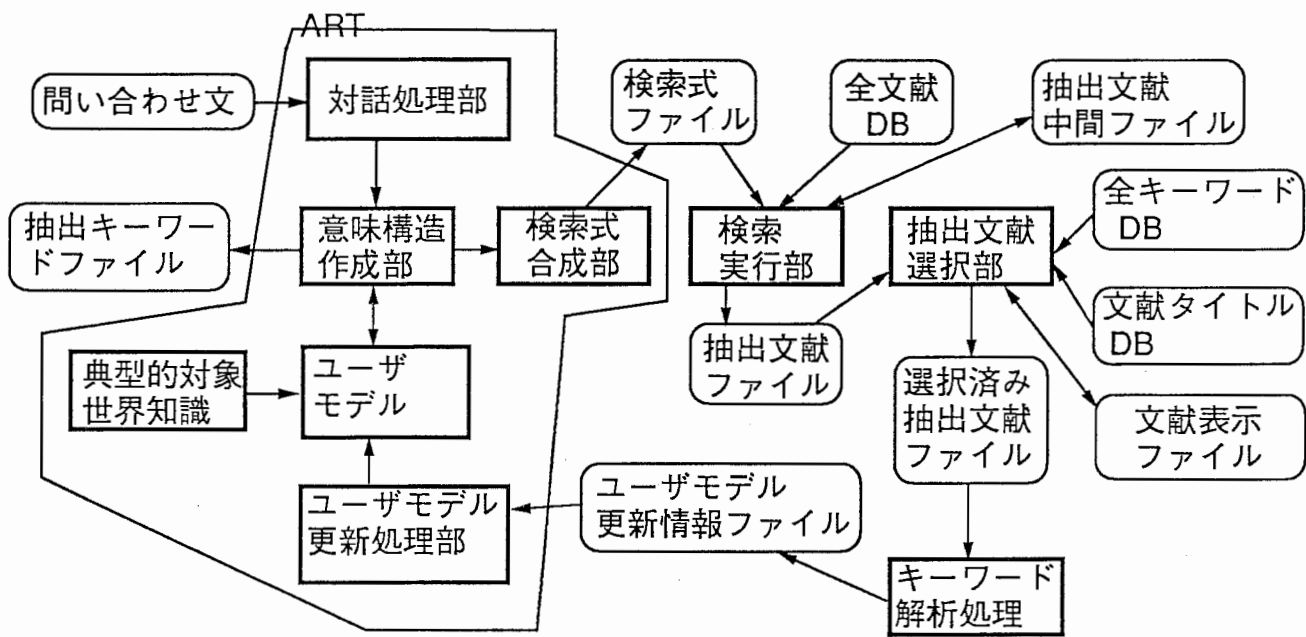


図 8.1: フィードバック付き文献検索システム入出力関係

(3) 文献選択によるユーザモデル修正部

利用者が選択した文献から検索意図の抽出を行ない、意味構造を修正する。

8.2 検索実行部

8.2.1 処理内容

(処理 1)

各検索式を全文庫データベース内の文献に付与されたキーワード No. のリストと照合して、対応する文献 ID を抽出し、抽出文献中間ファイルに格納する。抽出文献中間ファイルは、文献検索効率計算のために、常に残しておく。検索式は、キーワード No. を演算子 NOT, AND, OR で結合したものである。NOT 演算子は、そのキーワードを含んではならないことを陽に表現している。

(処理 2)

抽出文献中間ファイルから、利用者に表示するための、高い評点を持つ抽出文献ファイルを作る。

抽出文献ファイルは、次の基準と処理を満たした文献が格納される。

> フィードバック前

(step 1) 少なくとも 1 件以上得られた検索式で検索された文献グループから、評点が高い (絶対値としては小さい) ものから 1 グループ。

(step 2) 論理式の合成上、評点の低い (絶対値としては大きい) 文献グループには、評点の高い (絶対値としては小さい) 文献グループにある文献を含んでいるから、これらを削除する。

(step 3) この時点で文献数が 20 に満たなければ次に評点が低い (条件が緩い) 論理式を用いて検索した文献グループ。

(step 2,3) はトータルの文献数が 20 以上 100 以下になるまで続く。20 件を越えたら終了。

但し、評点が最高 (000) である時は、100 件を越しても表示する。

> フィードバック後

(step 1) 少なくとも 1 件以上得られた検索式で検索された文献グループから、評点が高い (絶対値としては小さい) ものから 3 グループ。但し 100 件以下とする。

但し、評点が最高 (000) である時は、100 件を越しても表示する。

(step 2) 論理式の合成上、評点の低い (絶対値としては大きい) 文献グループには、評点の高い (絶対値としては小さい) 文献グループにある文献を含んでいるから、これらを削除する。

(step 3) フィードバック時の抽出文献ファイルに既に存在していた文献は、除く。

8.2.2 入力ファイルフォーマット

(1) 検索式ファイル

検索式合成部によって作られた、評点づけられた複数の検索式を S 式で表現したデータ。

<例>

(T001 REL)

(0 F001 (OR (AND 7.10 (OR 1.15 1.20 1.19.3)) A.18.9))

(1 F002 (OR (AND (OR 1.5 7.10) (OR 1.15 1.20 1.19.3)) A.18.9))

(11 F003 (OR 1.5 7.10 1.15 1.20 1.19.3 A.18.9))

<フォーマット>

- 1 レコード目

(テストケース NO. 関係使用有無)

テストケース NO. = 4 バイト文字

関係使用有無 = rel/non_rel

- 2 レコード目以降

(評点 検索式 NO. 論理式)

評点 = 3 桁までの数字

検索式 NO. = 4 バイト文字

論理式 = <キーワード No. > もしくは ¬<キーワード No. > (注 1) を演算子 (AND または OR) で結合したリスプの S 式。

(注 1) ¬ は NOT を表す。

(2) 全文献データベース

全文献の文献 ID と各文献に付与されたキーワード No. のリスト。

ORACLE のテーブル newindex のキーワード No. の列を、複数データ持たせたものに対応する。

<例>

BUNKEN-0001 1.13.7 1.15 3.1 4.4.1

BUNKEN-0002 0 1.13.2 1.2.0 IA10.1

BUNKEN-0003 1.13.2 1.15 1.2.0 IA08.1 A.2.4

<フォーマット>

文献 ID キーワード No. リスト

(注) キーワード No. は、数字および ' 大文字の ' アルファベットとする。

8.2.3 出力ファイルフォーマット

(1) 抽出文献中間ファイル

検索式ファイルの検索式を用いて全文献データベースに対して検索されて抽出された全ての文献リスト。

文献リストは、評点ごとにグループ分けされている。

<例>

0 BUNKEN-0243 0 1.5

0 BUNKEN-1051 1.0 1.19.3 1.20 4.4.1 7.10

0 BUNKEN-1052 1.19.3 1.20 7 7.10 a.18.9

<フォーマット>

評点 (最大3桁の数字) 文献ID キーワードNo. リスト

(2) 抽出文献ファイル

利用者に表示する文献の情報を与えるための、高い評点を持つ、圧縮された文献リスト。

<例>

1 0 BUNKEN-0243 0 1.5

1 0 BUNKEN-1051 1.0 1.19.3 1.20 4.4.1 7.10

1 0 BUNKEN-1052 1.19.3 1.20 7 7.10 a.18.9

<フォーマット>

評価 (1/0) 評点 (最大3桁の数字) 文献ID キーワードNo. リスト

評価値はデフォルトで1として付加される。

8.3 抽出文献選択部

8.3.1 処理内容

(処理1) 抽出文献ファイルの文献ID、キーワードNo.を、文献タイトルデータベースおよび全キーワードデータベースを参照して、対応する文献タイトルおよびJ-ALIAS名に変換し、文献表示ファイルを作成する。

(処理2) 文献表示ファイルを画面上に表示する。

同一評点の文献をグループとして、評価(○/×)、規格化評点(999 - 評点値)、文献ID、文献名とキーワードリストを利用者に表示し、意図するかしないかをマウスクリックによって○×で示してもらう。

評価は、最初はすべて○にみえる。

これは、キーワードの最終確認のあとの文献検索結果に対する処理となる。

利用者が選択した結果は、文献表示ファイルの評価フィールドに直接反映する。

(処理3) 選択終了後、文献表示ファイルの評価フィールドの値を、抽出文献ファイルの評価フィールドに反映させ、これを選択済み抽出文献ファイルとして出力する。

1回の対話セッションの中では、フィードバック前と、フィードバック後の2回の処理が発生する。

8.3.2 入力ファイルフォーマット

(1) 抽出文献ファイル(上記を再記載)

<例>

1 0 BUNKEN-0243 0 1.5
1 0 BUNKEN-1051 1.0 1.19.3 1.20 4.4.1 7.10
1 0 BUNKEN-1052 1.19.3 1.20 7 7.10 a.18.9

<フォーマット>

評価 (1/0) 評点 (最大 3 桁の数字) 文献 ID キーワード No. リスト
評価値はデフォルトで 1 として付加される。

(2) 全キーワードデータベース

全キーワードのキーワード No. と、対応する J.alias 名の表。
ORACLE のテーブル indexterm に対応する。

<例>

0	人工知能
1	応用
1.0	地図
1.1	ゲーム

<フォーマット>

キーワード No.(char(16)) J.alias(代表するもの 1 つ)

(3) 文献タイトルデータベース

全文献の文献 ID と、対応する文献タイトルを格納するデータベース。
ORACLE のテーブル bibltab に対応する。

<例>

0 0 BUNKEN-0002 人工知能と分散処理 次世代 FMS の主要技術
0 0 BUNKEN-0003 FMS 設計用のエキスパートシステム
0 0 BUNKEN-0004 発達したエキスパート CAPP システムのサイクル

<フォーマット>

評価 (1/0) 評点 (最大 3 桁の数字) 文献 ID タイトル

(これは、実験用評価プログラム hyouka_new で用いるデータに合わせている。)

8.3.3 出力ファイルフォーマット

(1) 文献表示ファイル

利用者に表示するための、抽出した文献タイトルおよびキーワード名の表。

<例>

0 999 BUNKEN-0243 人間と計算機の相互作用の意図段階改善への人工知能適用
人工知能 マンマシンインターフェース
1 999 BUNKEN-1051 日本語インタフェースを有する知識処理型マルチメディア地図情報処理シ
テム GENTLE
地図 情報検索 データベース 知識ベース 自然言語インタフェース
1 999 BUNKEN-1052 知的インタフェースと自然言語処理
情報検索 データベース 自然言語処理 自然言語インタフェース 自然言語

<フォーマット>

評価 (1/0) 規格化評点 (3 桁の数字) 文献 ID(char(11)) タイトル 改行コード 空白(char(18)) J.alias
名のリスト

(2) 選択済み抽出文献ファイル

<フォーマット>

抽出文献ファイルと同一。

8.4 ユーザモデル修正部

8.4.1 処理内容

検索文献から利用者の検索意図に相当するキーワードを抽出し、それをもとにユーザモデルを修正する。次の2つの処理からなる。

- (キーワード解析処理)
選択済み抽出文献ファイルで利用者が選択した文献に付与されているキーワードの頻度分布および共起関係を解析し、キーワードの組をユーザモデル更新情報ファイルに書き出す。
- (ユーザモデル更新処理)
ユーザモデル更新情報ファイルの情報をもとに実際にユーザモデルを修正する。

1回の対話セッションの中では、フィードバック前と、フィードバック後の2回の処理が発生する。

8.4.2 キーワード解析処理

処理内容

(処理 1) キーワードの頻度分布表 (選択済み抽出文献ファイルで利用者が選択した文献に高い頻度で出現するキーワードの分布) および
共起関係分析表 (2つのキーワード間で共起関係の強い (2つのキーワードが同時に1つの文献に付与される頻度が高い) キーワードの組) を作成する。

(処理 2) 次の条件を満たすキーワード集合を抽出する。

$$\frac{S_n}{S} > \frac{N_n}{N} \times 2$$

S_n : 利用者が選択した文献中、
そのキーワードを含む文献数

S : 利用者が選択した文献数

N_n : 全文献中、そのキーワードを含む文献数

N : 全文献数

(処理 3) (処理 2) で得られたキーワード集合において、利用者が対話によって最終的に選んだキーワードと共起関係が最大のキーワードのペアを抽出し、ユーザモデル更新情報ファイルに書き出す。(但し、既に対話によって抽出されて意味構造に含まれていたキーワードとのペアは除く。また共起関係が、1のものは除く。)

(注 1) あらかじめ検索式ファイルからキーワード No. でソートした利用キーワードファイルをつくっておく。ファイル形式不問。

入力ファイル

- 選択済み抽出文献ファイル

出力ファイル

- ユーザモデル更新情報ファイル
ユーザモデルにリンクを張るべき2つのキーワード No. の組
<フォーマット> 不問

8.4.3 ユーザモデル更新処理部

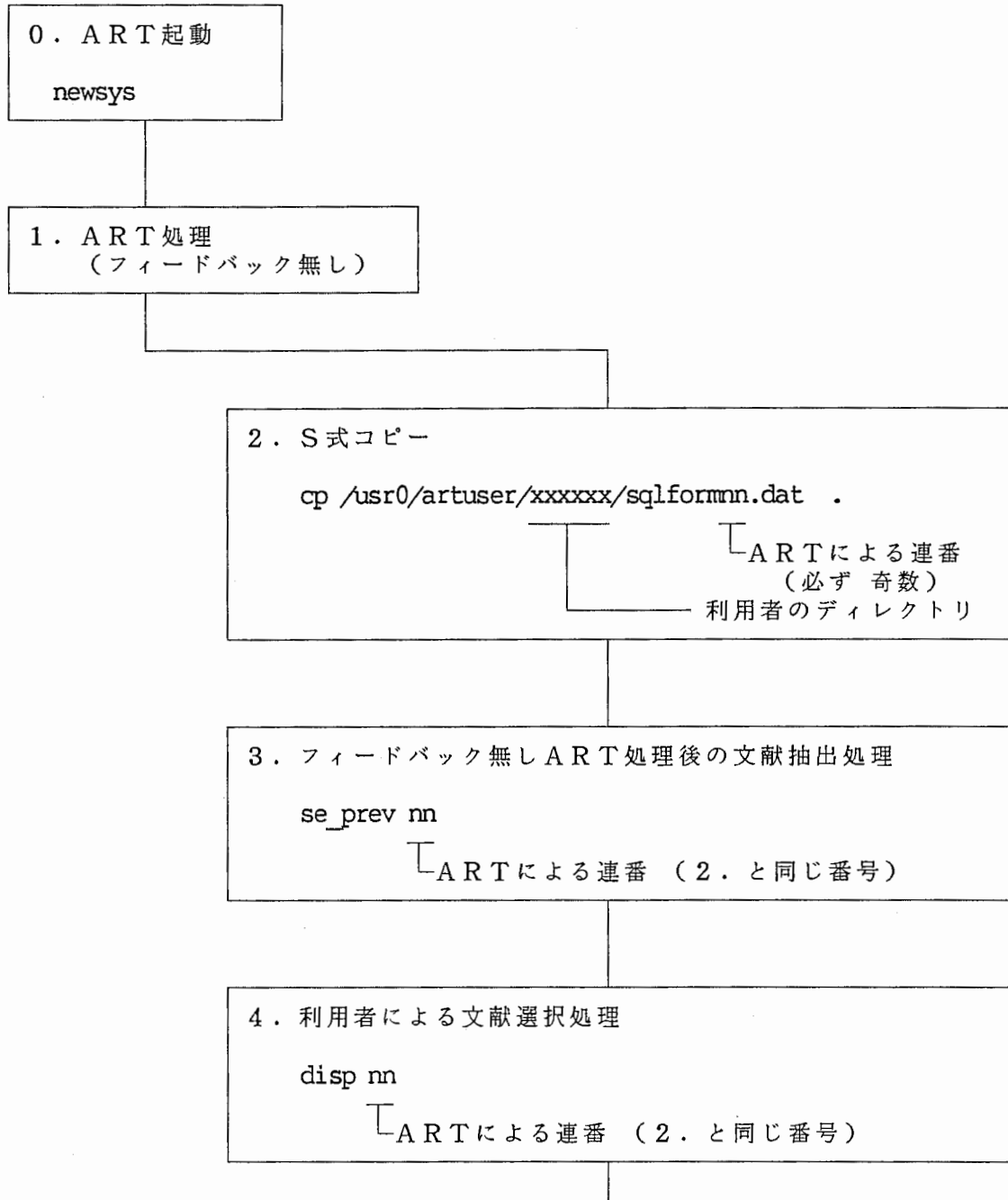
キーワード解析処理で与えられたユーザモデル更新情報ファイルの情報を元に、ユーザモデルを更新する。

- ユーザモデル更新情報ファイルのキーワードの組に対して、それに対応するユーザモデルのノード間で、既に、現在の意味構造識別子を持ったある種類の関連が存在すれば、その関連の強さを”意図する”に変更する。
- ユーザモデル更新情報ファイルのキーワードの組に対して、それに対応するユーザモデルのノード間で、現在の意味構造識別子を持った関連が存在しないならば、そのノード間に、現在の意味構造識別子を持った関連関係(”意図する”)のリンクを追加する。ここで付加されたリンクの推論における方向性は、両方向であるが、(フェーズ4)の論理式合成のための意味構造では、基本意味構造からの推論方向に準ずる。

第 9 章

フィードバック付き文献検索システム操作方法

<p>ART 処理</p> <ul style="list-style-type: none"> • ディレクトリ (/usr1/ARTK) • ユーザー (kita) 	<p>フィードバック処理</p> <ul style="list-style-type: none"> • ディレクトリ (/usr0/kita/gousei_new) • ユーザー (kita)
--	--



5. 利用者が選択した文献から、フィードバック処理
のためのキーワードの組を抽出

feedback_proc mn

└ARTによる連番 (2. と同じ番号)

6. ART処理 (続き)

(フィードバック付き
ART処理)

フィードバックのファイルは
/usr0/kita/gousei_new/koushin.f
(固定)

7. S式コピー

cp /usr0/artuser/xxxxxx/sqlformn.dat .

└ARTによる連番
(必ず 偶数)
利用者のディレクトリ

8. フィードバック付きART処理後の文献抽出処理

se_after mn

└ARTによる連番 (7. と同じ番号)

9. 利用者による文献選択処理

disp mn

└ARTによる連番 (7. と同じ番号)

1. へ 戻る

0. ART 起動

1) ARTシステムの起動

1-1) セーブファイルによる起動

90.08.31 現在のARTシステムが、newsys という名前でセーブしてあるので、このファイルを実行する。

1-2) 基本的な起動

システムの変更を行った場合や newsys が存在しないときには、以下のようにして、ART起動後に、必要なファイルをロードする。

```
art
> (load "/usr0/kita/ARTK/rule/system_load.lisp")
> (load "/usr0/kita/ARTK/gousei/load_gsystem.lisp")
> (load "/usr0/kita/new_net/n2_load_new.lisp.bin")
```

*) セーブファイルを作成するには、ファイルロード後

```
> (disksave "ファイル名" :gc t)
```

とする。

2) graphics-studio の起動

```
> (graphics-studio)
```

3) システムの起動

3-1) ART-Lisp の起動

graphics-studio の COMMAND WINDOW に Control + e を
入力する。
(プロンプトが => から ART3.1 Lisp> に変わる。)

3-2) システムの起動

```
ART3.1 Lisp> (system-start)
```

1. ART処理 (フィードバック無し)

- ・ 名前入力後の

「ロードするフィードバック結果が有りますか？ (Y or N)」

に対して、必ず **n** を入力する。

- ・ 間違えて **y** を入力したときは、一度処理を終了して、もう一度最初からやり直し。

4. 利用者による文献選択処理

- ・ 表示は最初 「×」 (非選択) となっている。
- ・ 該当する行で 左マウスクリック により、「○」 (選択) 「×」 (非選択) が トグル表示される。
- ・ Quit のエリアで 左マウスクリック により、終了する。

6. ART処理 (フィードバック付き)

- ・ 名前入力後の

「ロードするフィードバック結果が有りますか？ (Y or N)」

に対して、必ず **y** を入力する。

- ・ 間違えて **n** を入力したときは、一度処理を終了して、もう一度最初からやり直し。
- ・ 入力するファイル名は、必ず `/usr0/kita/gousei_new/koushin.f` とすること。

9. 利用者による文献選択処理

- ・ 4. と同様

*) 終了方法

(1) プログラムの終了

(1-1) 正常終了

- ・ 最後の (フィードバック付き) ART 処理終了後の

「あなたの名前を入力して下さい」

に対して,

おわり

と, 入力する。

(1-2) 異常終了 (y, n を間違えたような場合)

- ・ キーボードの R 1 5 (右下にある) を押す。

(上記いずれの場合もプロンプト 「=>」 が表示される。)

(2) ART の終了

- ・ プロンプト 「=>」 に対して,

exit

と, 入力する。

- ・ 終了してもよいか, という確認メッセージに対して,

y

と, 入力する。

付録 A

典型的対象世界知識

実験システムにおいてユーザモデルの初期状態として用いた対象世界知識を示す。

```

*****
*** Universal Knowledge's Representation of ***
*** Artificial Intelligence ***
*** ATR COMMUNICATION SYSTEMS RESEARCH LABORATORIES ***
*** 1988. 11. 1 v. 6. 2 ***
*** 1988. 12. 13 v. 7. 0 ***
*** 対象世界知識 ***
*****

```

0 General

```

<j-alias> 人工知能 A I
<jicst-kw> 人工知能

```

1 Applications and Expert Systems

```

<j-alias> 応用 アプリケーション
<jicst-kw> 応用

```

1_0 Cartography

```

<j-alias> 地図, 図形

```

1_1 Games (8_2 Graph and Tree Search Strategies)

```

<j-alias> ゲーム
<jicst-kw> ゲーム

```

1_1_0 Chess

```

<j-alias> チェス
<jicst-kw> 二人ゲーム

```

1_1_0_0 Plausible Move Generators

```

<j-alias> 有望手生成

```

1_1_0_1 Static Evaluation

```

<j-alias> 静的評価

```

1_1_0_2 Horizon Effect

```

<j-alias> 視界効果

```

1_1_0_3 Chunking Methods

```

<j-alias> チャンキング法

```

1_1_1 Checkers

```

<j-alias> チェッカー
<jicst-kw> 二人ゲーム

```

1_1_2 Backgammon

```

<j-alias> バックギャモン
<jicst-kw> 二人ゲーム

```

1_1_3 Bidding Games

```

<j-alias> トランプゲーム

```

1_1_4 Wagering Games

```

<j-alias> 賭博ゲーム

```

1_1_5 War Games

```

<j-alias> 戦争ゲーム
<jicst-kw> ゲーム * 戦略

```

1_2 Industrial Applications

```

<j-alias> 産業
<jicst-kw> 産業

```

1_2_0 Automatic Assembly

```

<j-alias> 組立 自動組立
<jicst-kw> 自動組立

```

1_2_1 Parts Handling

```

<j-alias> 部品ハンドリング

```

1_2_2 Inspection

```

<j-alias> 検査
<jicst-kw> 検査

```

1_2_3 Welding

```

<j-alias> 溶接
<jicst-kw> 溶接

```

1_2_4 Planning for Production

```

<j-alias> 生産計画
<jicst-kw> 生産計画

```

1_2_5 Inventroy

```

<j-alias> 在庫管理
<jicst-kw> 在庫管理

```

1_2_6 Material Handling

```

<j-alias> 物流 マテリアルハンドリング

```


<j-alias> 財務計画, 財務管理
 <jicst-kw>財務管理

1_9 Natural Language Processing==>17 Natural Language Processing1

1_10 Mathematical Aids(3_3 Mathematical Induction ,a_10 確率)
 <j-alias> 数学
 <jicst-kw>数学

1_11 Education
 <j-alias> 教育
 <jicst-kw>教育訓練

1_12 Library Applications
 <j-alias> 図書, 図書館
 <jicst-kw>図書

1_13 設計
 <j-alias> 設計

1_13_0 Computer System Design
 <j-alias> コンピュータシステム設計 計算機システム開発
 <jicst-kw> 計算機システム開発

1_13_0_0 Software Design
 <j-alias> ソフトウェア設計
 <jicst-kw> ソフトウェア設計

1_13_0_0_0 Program Design
 <j-alias> プログラム設計
 <jicst-kw> プログラム設計

1_13_0_0_1 Database Design
 <j-alias> データベース設計
 <jicst-kw> ソフトウェア設計*データベース

1_13_0_1 Hardware Design
 <j-alias> ハードウェア設計
 <jicst-kw> ハードウェア設計

1_13_0_2 Network System Design
 <j-alias> ネットワークシステム設計
 <jicst-kw> ソフトウェア設計*ネットワーク

1_13_1 VLSI Design Aids
 <j-alias> 超 L S I 設計 V L S I 設計
 <jicst-kw> 超LSI*回路設計* C A D

1_13_2 CAD
 <j-alias> C A D
 <jicst-kw> C A D

1_13_5 Layout Design
 <j-alias> レイアウト設計, 配置設計
 <jicst-kw> 配置設計

1_13_6 CAM
 <j-alias> C A M
 <jicst-kw> C A M

1_13_7 Engineering Automations
 <j-alias> 自動設計
 <jicst-kw>自動設計

1_13_8 製品設計==>1_2_9 Products Design 製品設計

1_14 System Troubleshooting
 <j-alias> トラブルシューティング 故障修理

1_15 Expert Systems
 <j-alias> エキスパートシステム
 <jicst-kw> エキスパートシステム

1_15_0 Expert System Languages <==[5_5 Expert System Languages]
 <j-alias> エキスパートシステム言語, エキスパートシステム開発支援環境
 <jicst-kw> エキスパートシステム*(支援プログラム+プログラミング言語)

1_15_1 Expert Knowledge (a_8 専門家)
 <j-alias> 専門知識 専門家知識

1_15_2 Plausible Reasoning
 <j-alias> 妥当な推論

1_15_4 Generation of Explanations(7_10 Natural Language Interfaces)
 <j-alias> 説明生成
 <jicst-kw> エキスパートシステム*システムインタフェイス

1_15_5 Deep Models
 <j-alias> 深いモデル

1_15_7 Validation of Expert Systems <==[IA06_3_2 エキスパートシステム検証]

- <j-alias> エキスパートシステム検証
- 1_15_8 MetaKnowledge
 - <j-alias> メタ知識
- 1_17 Aviation Applications
 - <j-alias> 航空
 - <jicst-kw> 航空交通管制
 - 1_17_0 Pilot Aids
 - <j-alias> パイロット支援
 - 1_17_1 Air Traffic Control
 - <j-alias> 航空交通管制
 - <jicst-kw> 航空交通管制
- 1_18 パターン認識(==>IA06_8_3)
 - <j-alias> パターン認識
- 1_19 Text Analysis
 - <j-alias> テキスト解析 文書処理
 - <jicst-kw> 文書処理
 - 1_19_0 Text Summarizing
 - <j-alias> テキスト要約 自動抄録法
 - <jicst-kw> 自動抄録法
 - 1_19_2 Automatic Indexing(1_20 Data Base)
 - <j-alias> 自動インデクシング 自動索引法
 - <jicst-kw> 自動索引法
 - 1_19_3 Information Retrieval(1_20 Data Base)
 - <j-alias> 情報検索
 - <jicst-kw> 情報検索
 - 1_19_3_0 Document Retrieval
 - <j-alias> 文献検索
 - 1_19_4 Text Generation <==[7_0_2 Text Generation]
 - <j-alias> テキスト生成
 - <jicst-kw> 文書処理
- 1_20 Data Base(1_19_2 Automatic Indexing, 1_19_3 Information Retrieval)
 - <j-alias> データベース
 - 1_20_0 Relational Data Base
 - <j-alias> 関係データベース リレーショナルデータベース
- 1_21 communication <==[上位 1_7 Military Applications]
 - <j-alias> 通信 コミュニケーション
- 1_22 Decision Aids <==[上位 1_7 Military Applications]
 - <j-alias> 意志決定支援
- 2 Automatic Programming
 - <j-alias> 自動プログラミング
 - <jicst-kw> 自動プログラミング
 - 2_0 Automatic Analysis
 - <j-alias> 自動解析
 - 2_1 Program Modification
 - <j-alias> プログラム修正
 - <jicst-kw> デバッグ
 - 2_2 Program Synthesis
 - <j-alias> プログラム合成
 - <jicst-kw> 自動プログラミング
 - 2_3 Program Transformation
 - <j-alias> プログラム変換
 - <jicst-kw> プログラム変換
- 3 Inference
 - <j-alias> 推論 推論方式
 - <jicst-kw> 人工知能推論
 - 3_1 Deduction
 - <j-alias> 演えき推論
 - 3_2 Logic Programming (4_1_0 First Order Predicate Calculus)
 - <j-alias> ロジックプログラミング, 論理プログラミング
 - <jicst-kw> 論理プログラミング
 - 3_2_0 Logic Programming Theory
 - <j-alias> ロジックプログラミング理論, 論理プログラミング理論
 - 3_2_1 Logic Programming Applications
 - <j-alias> ロジックプログラミング応用, 論理プログラミング応用

- 3_2_2 Logic Programming Languages <=[5_1 Logic Programming Languages]
 - <=[5_4_5 Logic-Based Languages]
 - <j-alias> ロジックプログラミング言語, 論理プログラミング言語, 論理型言語
 - <jicst-kw> 論理プログラミング* プログラミング言語
- 3_2_2_0 Prolog
 - <j-alias> プロローグ, P R O L O G
 - <jicst-kw> Prolog
- 3_3 Mathematical Induction(1_10 Mathematical Aids)
 - <j-alias> 数学的推論
 - <jicst-kw> 人工知能推論* 論理
- 3_4 Metatheory
 - <j-alias> メタ理論, メタ推論
 - <jicst-kw> 人工知能推論* 階層構造* 論理
- 3_5 Nonmonotonic Reasoning
 - <j-alias> 非単調論理
 - <jicst-kw> 人工知能推論* 記号論理[非単調論理]
- 3_5_0 Circumscription
 - <j-alias> サーカムスクリプション
 - <jicst-kw> 人工知能推論* 記号論理
- 3_5_1 Data Dependencies
 - <j-alias> データ依存
- 3_6 Resolution
 - <j-alias> 導出原理
 - <jicst-kw> 導出原理
- 3_6_0 Unification Algorithms
 - <j-alias> ユニフィケーション
 - <jicst-kw> 導出原理[ユニフィケーション]
- 3_6_1 Resolvents
 - <j-alias> 導出形
 - <jicst-kw> 導出原理
- 3_6_2 Termination
 - <j-alias> 終了 停止
 - <jicst-kw> 停止問題
- 3_7 Belief Revision <=[11_2 Belief Revision]
 - <j-alias> 信念の翻意
- 3_8 Theorem Proving
 - <j-alias> 定理証明
- 4 Knowledge Representation <=[対象 1_15_1 Expert Knowledge]
 - <j-alias> 知識表現
 - <jicst-kw> 知識表現
- 4_0 Frames and Scripts (7_8 Story Understanding)
 - <j-alias> フレーム スクリプト
 - <jicst-kw> 知識表現
- 4_0_0 Defaults
 - <j-alias> デフォルト
- 4_0_1 Stereotypes and Prototypes [<=[上位 ソフトウェア]
 - <j-alias> ステロタイプ プロトタイプ
 - <jicst-kw> プログラム設計
- 4_0_2 Generation of Expectations
 - <j-alias> 予期の生成
- 4_0_3 Frame Languages ==>[5_4_0 Frame-Based Languages]
- 4_0_4 Frame-Driven Systems
 - <j-alias> フレーム型システム フレームシステム
 - <jicst-kw> エキスパートシステム* 知識表現
- 4_0_5 Inheritance Hierarchy (5_4_2 Structured Inheritance Languages)
 - <j-alias> インヘリタンス
 - <jicst-kw> 知識表現* 階層構造[継承]
- 4_1 Predicate Logic
 - <j-alias> 述語論理
 - <jicst-kw> 述語論理
- 4_1_0 First Order Predicate Calculus (3_2 Logic Programming)
 - <j-alias> 一階述語論理
 - <jicst-kw> 述語論理
- 4_1_1 Skolem Functions
 - <j-alias> スコーレム関数

- <jicst-kw> 述語論理 * 論理関数
- 4_1_2 Second Order Logic
 - <j-alias> 二階述語論理
 - <jicst-kw> 述語論理
- 4_1_3 Modal Logic
 - <j-alias> 様相論理
 - <jicst-kw> 様相論理
- 4_1_4 Fuzzy Logic(IA08_2_5 あいまいさ IA08_2_25 不確実性)
 - <j-alias> ファジイ論理
 - <jicst-kw> ファジイ集合
- 4_1_6 Temporal Logic
 - <j-alias> 時制論理
 - <jicst-kw> 時間論理
- 4_3 Representation Languages ==>[5_4 Knowledge Representation Languages]
- 4_4 Procedural and Rule-Based Representations
 - <j-alias> ルール表現
 - <jicst-kw> 知識表現
- 4_4_0 Production Rule Systems(5_4_4 Rule-Based Languages)
 - <j-alias> プロダクションルール, プロダクションシステム
 - <jicst-kw> プロダクションシステム[プロダクションルール]
- 4_4_0_0 Analysis of Protocols
 - <j-alias> プロトコル解析
- 4_4_0_1 Production Rule Ordering
 - <j-alias> ルールの順序付け
- 4_4_0_2 Agendas
 - <j-alias> アジェンダ
- 4_4_0_3 Blackboards
 - <j-alias> 黒板, 黒板モデル
- 4_4_0_3_0 HEARSAY-2
 - <j-alias> H E A R S A Y - 2
- 4_4_1 Knowledge Bases
 - <j-alias> 知識ベース
 - <jicst-kw> 知識ベース
- 4_5 Semantic Networks (5_4_6 Semantic Network Languages)
 - <j-alias> 意味ネットワーク
 - <jicst-kw> 知識表現 * ネットワーク構造[意味ネットワーク]
- 4_6 Connectionist Systems (7_2_10 Connectionist Parsing Models)
 - <j-alias> コネクショニストシステム
 - <jicst-kw> 知識表現 * ネットワーク構造
- 4_7 Multiple Agent/Actor Systems
 - <j-alias> マルチエージェント, マルチアクター
- 4_8 Constraints (8_2_10 Constraint Propagation)
 - (5_4_3 Constraint Languages)
 - <j-alias> 制約条件
- 4_9 Discrimination Trees and Networks
 - <j-alias> 弁別木, 弁別ネットワーク
- 4_10 Belief Models (7_8 Story Understanding)
 - <j-alias> 信念モデル
 - <jicst-kw> 心理学 * 知識表現
- 4_11 Representation of the Physical World (7_8 Story Understanding)
 - <j-alias> 物理的世界表現
- 4_11_0 Representation of Time
 - <j-alias> 時間表現
 - <jicst-kw> 時間 + 時間論理
- 4_11_0_0 Interval-Based Time Representation
 - <j-alias> 時間間隔表現
- 4_11_0_1 Point-Based Time Representation
 - <j-alias> 時点表現
- 4_11_1 Representation of Space
 - <j-alias> 空間表現
- 4_11_2 Causal Knowledge
 - <j-alias> 因果関係知識表現
- 4_12 Representation of Natural Language Semantics (7_2_1 Natural Language Understanding)
 - <j-alias> 自然言語意味表現
 - <jicst-kw> 自然語 * 自動言語処理 * 知識表現

- 4_12_0 Conceptual Dependency
 - <j-alias> 概念依存
 - <jicst-kw> 自然語*自動言語処理*知識表現
- 4_12_1 Preference Semantics
 - <j-alias> 優先的意味論
- 4_12_2 Procedural Semantics
 - <j-alias> 手続的意味論
- 4_12_3 Logic Based Natural Language Representation
 - <j-alias> 論理に基づく自然言語表現
- 4_12_4 Montague Semantics
 - <j-alias> モンタギュー意味論
- 4_13 Modeling(11 cognitive modeling and psychological studies of intelligence)
 - <j-alias> モデル, モデリング
- 5 Programming Languages <== [上位 a_15 ソフトウェア]
 - <j-alias> プログラミング言語, プログラム言語
 - <jicst-kw> プログラミング言語
 - 5_0 Functional Programming Languages
 - <j-alias> リスト処理言語 関数型言語
 - <jicst-kw> リスト処理言語
 - 5_0_1 Garbage Collection
 - <j-alias> ガーベッジコレクション
 - <jicst-kw> ガーベッジコレクション
 - 5_0_2 Deep vs_ Shallow Binding
 - <j-alias> バインディング
 - <jicst-kw> リスト処理言語
 - 5_0_3 Lazy Evaluation
 - <j-alias> レイジー評価
 - 5_0_4 CDR Coding
 - <j-alias> C D R コーディング
 - 5_1 Logic Programming Languages ==> [3_2_2 Logic Programming Languages]
 - 5_2 Object-Oriented Languages
 - <j-alias> オブジェクト指向言語
 - <jicst-kw> オブジェクト指向プログラミング
 - 5_2_1 Abstract Data Types
 - <j-alias> 抽象データ型
 - <jicst-kw> 抽象データ型
 - 5_4 Knowledge Representation Languages <== [4_3 Representation Languages]
 - <j-alias> 知識表現言語
 - <jicst-kw> 知識表現*プログラミング言語[知識表現言語]
 - 5_4_0 Frame-Based Languages <==[4_0_3 Frame Languages]
 - <j-alias> フレーム型言語, フレーム言語
 - <jicst-kw> 知識表現*プログラミング言語
 - 5_4_2 Structured Inheritance Languages (4_0_5 Inheritance Hierarchy)
 - <j-alias> 階層構造言語
 - <jicst-kw> 知識表現*階層構造*プログラミング言語
 - 5_4_3 Constraint Languages (4_8 Constraints, 8_2_10 Constraint Propagation)
 - <j-alias> 制約言語
 - 5_4_4 Rule-Based Languages (4_4_0 Production Rule Systems)
 - <j-alias> ルール型言語, ルール言語
 - <jicst-kw> 知識表現*プログラミング言語
 - 5_4_4_1 Rete Algorithm
 - <j-alias> リートアルゴリズム
 - 5_4_5 Logic-Based Languages ==>[3_2_2 Logic Programming Languages]
 - 5_4_6 Semantic Network Languages (4_5 Semantic Networks)
 - <j-alias> 意味ネットワーク言語
 - <jicst-kw> 知識表現*ネットワーク構造*プログラミング言語
 - 5_5 Expert System Languages ==>[1_15_0 Expert System Languages]
 - 5_6 Programming Environment and Aids <==[1_13_3 Programming Aids]
 - <j-alias> プログラミング環境 プログラミング支援
 - <jicst-kw> 支援プログラム+プログラム設計*計算機利用
 - 5_6_0 Tracing and Debugging Aids
 - <j-alias> デバッグ支援
 - <jicst-kw> デバッグ*支援プログラム
 - 5_6_1 File Maintenance System

- <j-alias> ファイル管理システム
 - <jicst-kw> データベース管理システム
 - 5_6_2 Editors
 - <j-alias> エディタ
 - <jicst-kw> エディタ
 - 5_6_4 Screen Management Systems
 - <j-alias> ウィンドウ
- 5_8 Programming Theory <==[上位 IA10_2_3 計算理論]
 - <j-alias> プログラミング理論, プログラム理論
 - <jicst-kw> プログラム理論
 - 5_8_1 Recursive
 - <j-alias> 再帰
 - <jicst-kw> 帰納的関数
 - 5_8_2 Iterative
 - <j-alias> 繰り返し
 - 5_8_3 Conditional Branch
 - <j-alias> 条件分岐
 - 5_8_4 Programming Verification <==[IA06_3_1 Programming Verification]
 - <j-alias> プログラム検証
 - <jicst-kw> プログラム検証
- 6 Learning
 - <j-alias> 学習
 - <jicst-kw> 知識獲得
 - 6_0 Analogies
 - <j-alias> 類推
 - <jicst-kw> 人工知能推論*(学習機械+知識獲得)
 - 6_0_0 Geometric Analogies
 - <j-alias> 幾何学的類推
 - 6_0_1 Natural Language Analogies
 - <j-alias> 自然言語類推
 - 6_0_2 Structural Analogies
 - <j-alias> 構造類推
 - 6_0_3 Functional Analogies
 - <j-alias> 機能類推
 - 6_1 Concept Learning
 - <j-alias> 概念学習
 - <jicst-kw> 知識獲得*概念
 - 6_1_0 Near-Miss Analysis
 - <j-alias> ニアミス解析
 - 6_1_1 Version Spaces
 - <j-alias> バージョン空間
 - 6_1_2 Schema Acquisition
 - <j-alias> スキーマ獲得
 - 6_1_3 Learning of Heuristics
 - <j-alias> 経験則学習
 - 6_2 Induction
 - <j-alias> 帰納推論 <==[上位 3 Deduction]
 - <jicst-kw> 学習機械[帰納]
 - 6_3 Knowledge Acquisition <==[対象 1_15_1 Expert Knowledge]
 - <==[手段 a_1 会話]
 - <==[手段 IA10_1 経験]
 - <==[手段 IB01_1 観察]
 - <==[手段 8_3 発見的方法]
 - <j-alias> 知識獲得
 - <jicst-kw> 知識獲得
 - 6_3_1 Dialogues with Experts(a_8 専門家)
 - <j-alias> 専門家との対話
 - 6_5 Language Acquisition
 - <j-alias> 言語獲得
 - <jicst-kw> 知識獲得*自然語
 - 6_5_0 Acquisition of Grammar
 - <j-alias> 文法獲得
 - <jicst-kw> 知識獲得*文法
 - 6_6 Parameter Learning
 - <j-alias> パラメータ学習

- <jicst-kw>知識獲得*パラメータ
- 6_7 Associative Learning
 - <j-alias> 連想的学習 連想学習
 - <jicst-kw>知識獲得*連想
- 6_8 Learning of Skills
 - <j-alias> 技能学習
- 7 Natural Language Processing <==[1_9 Natural Language Processing]
 - <j-alias> 自然言語処理
 - <jicst-kw> 自然語*自動言語処理
 - 7_0 Language Generation
 - <j-alias> 言語生成
 - <jicst-kw> 会話形処理
 - 7_0_0 Speech Synthesis
 - <j-alias> 音声合成
 - <jicst-kw> 音声合成
 - 7_0_1 Discourse Generation
 - <j-alias> 談話生成
 - <jicst-kw> 会話形処理
 - 7_0_2 Text Generation=>[1_19_4 Text Generation]
 - <j-alias> テキスト合成
 - <jicst-kw> 文書処理
 - 7_1 Language Models
 - <j-alias> 言語モデル
 - <jicst-kw> 自然語*知識表現
 - 7_2 Syntactic Parsing
 - <j-alias> 構文解析
 - <jicst-kw> (構文分析+文法)*自動言語処理
 - 7_2_0 ATNs
 - <j-alias> 拡張遷移網
 - <jicst-kw> 自動言語処理*文法
 - 7_2_1 Deterministic Parsers
 - <j-alias> パーサ, 決定論パーサ
 - <jicst-kw> 構文分析
 - 7_2_2 Phrase Structure Grammars
 - <j-alias> 句構造文法
 - <jicst-kw> 句構造文法
 - 7_2_3 Transformational Grammars
 - <j-alias> 変形文法
 - <jicst-kw> 生成文法
 - 7_2_4 Functional Grammars
 - <j-alias> 関数文法
 - <jicst-kw> 文法
 - 7_2_5 Case Grammars
 - <j-alias> 格文法, 文脈自由文法
 - <jicst-kw> 文脈自由文法
 - 7_2_6 Semantic Grammars
 - <j-alias> 意味文法
 - <jicst-kw> 自動言語処理*文法
 - 7_2_7 Logical Grammars
 - <j-alias> 論理型文法
 - <jicst-kw> 自動言語処理*文法*論理
 - 7_2_8 Integrated Parsers
 - <j-alias> 統合化パーサ
 - <jicst-kw> 構文分析
 - 7_2_9 Word Expert Parsers
 - <j-alias> 語エキスパートパーサ
 - 7_2_10 Connectionist Parsing Models (4_6 Connectionist Systems)
 - <j-alias> コネクションパーシングモデル
 - 7_3 Machine Translation
 - <j-alias> 機械翻訳, 自動翻訳
 - <jicst-kw> 機械翻訳
 - 7_3_1 Machine-Aided Translation
 - <j-alias> 翻訳支援
 - 7_4 Speech Recognition and Understanding

- <j-alias> 音声認識
- <jicst-kw> 音声認識
- 7_4_0 Acoustic Analysis
 - <j-alias> 聴覚解析, 聴覚分析
 - <jicst-kw> 聴覚*音声認識
- 7_4_1 Phonological Analysis
 - <j-alias> 音韻解析, 音韻分析
 - <jicst-kw> 音韻論+音声分析
- 7_4_2 Isolated Word Recognition
 - <j-alias> 離散の音声認識
 - <jicst-kw> 音声認識
- 7_4_3 Continuous Speech Recognition
 - <j-alias> 連続音声認識
 - <jicst-kw> 音声認識
- 7_4_4 Speaker Recognition [<== 上位 IA06_8_3 パターン認識]
 - <j-alias> 話者認識
 - <jicst-kw> 話者認識
- 7_6 Lexicology and Lexicography
 - <j-alias> 辞書
 - <jicst-kw> 自動辞書
- 7_6_0 Machine-Readable Dictionaries
 - <j-alias> 機械可読辞書
 - <jicst-kw> 機械可読資料*自動辞書
- 7_6_1 Thesauri
 - <j-alias> シソーラス
 - <jicst-kw> シソーラス
- 7_7 Discourse Understanding
 - <j-alias> 談話理解
 - <jicst-kw> 言語理解[談話理解]
- 7_7_0 Speech Acts<==[7_11_0 Speech Acts]
 - <j-alias> 発話行為
 - <jicst-kw> 言語理解
- 7_7_1 Context for understanding==>[7_12_4 Context for understanding]
- 7_8 Story Understanding (4_0 Frames and Scripts,
 - 4_10 Belief Models,
 - 4_11 Representation of the Physical World
 - 7_12 Natural Language Understanding)
- <j-alias> 物語理解
- <jicst-kw> 言語理解
- 7_8_0 Story Grammars
 - <j-alias> 物語文法
- 7_8_1 Story Points
 - <j-alias> 物語の要点
- 7_8_2 Narrative Units
 - <j-alias> 説話
- 7_8_3 Story Morals
 - <j-alias> 物語の教訓
- 7_10 Natural Language Interfaces <== [1_5_0 Natural Language Interfaces]
 - <== 対象 1_20 Data Base
 - <== 対象 1_15 Expert Systems
 - <== 対象 9 Robotics
 - (1_15_4 Generation of Explanations)
- <j-alias> 自然言語インタフェース
- <jicst-kw> (システムインタフェース+会話形処理+質問応答システム)*自動言語処理
- 7_11 Dialogue Understanding
 - <j-alias> 対話理解
 - 7_11_0 Speech Acts ==>[7_7_0 Speech Acts]
 - 7_11_1 Context for Understanding==>[7_2_1_4 Context for Understanding]
- 7_12 Natural Language Understanding (7_8 Story Understanding)
 - <j-alias> 自然言語理解
 - <jicst-kw> 言語理解
- 7_12_2 Pronoun Resolution
 - <j-alias> 発話
 - <jicst-kw> 音韻論
- 7_12_3 Ellipsis

- <j-alias> 省略
 - <jicst-kw> 言語理解 * 言語学
 - 7_12_4 Context for Understanding [<==7_7_1 Context for Understanding] [<==7_11_1 Context for Understanding]
 - <j-alias> 文脈理解
 - 7_12_5 Anaphora
 - <j-alias> 照応
- 8 Problem Solving, Control Methods and Search <==[IA10_2 理論]
 - <j-alias> 探索, 探索理論
 - 8_0 Backtracking
 - <j-alias> バックトラッキング, 後戻り
 - <jicst-kw> 木探索 + 探索理論 + 探索問題 + データ探索
 - 8_1 Dynamic Programming
 - <j-alias> 動的計画法
 - 8_2 Graph and Tree Search Strategies (1_1 Games)
 - <j-alias> 木探索, グラフ探索
 - <jicst-kw> 木探索 + 探索理論 + 探索問題 + データ探索
 - 8_2_0 Depth First Search
 - <j-alias> 縦型探索
 - <jicst-kw> 木探索 + 探索理論 + 探索問題 + データ探索
 - 8_2_1 Breadth First Search
 - <j-alias> 横型探索
 - <jicst-kw> 木探索 + 探索理論 + 探索問題 + データ探索
 - 8_2_2 Best First Search
 - <j-alias> 最良優先探索
 - <jicst-kw> 木探索 + 探索理論 + 探索問題 + データ探索
 - 8_2_3 Branch and Bound
 - <j-alias> 分枝限定法
 - <jicst-kw> 木探索 + 探索理論 + 探索問題 + データ探索
 - 8_2_4 Hill-Climbing
 - <j-alias> 山登り法
 - <jicst-kw> 木探索 + 探索理論 + 探索問題 + データ探索
 - 8_2_5 Minimax
 - <j-alias> ミニマックス探索
 - <jicst-kw> 木探索 + 探索理論 + 探索問題 + データ探索
 - 8_2_6 Alpha-Beta Algorithm
 - <j-alias> アルファベータアルゴリズム
 - 8_2_7 A* Algorithm
 - <j-alias> A*アルゴリズム
 - <jicst-kw> 木探索 + 探索理論 + 探索問題 + データ探索
 - 8_2_8 Beam Search
 - <j-alias> ビーム探索法
 - <jicst-kw> 木探索 + 探索理論 + 探索問題 + データ探索
 - 8_2_10 Constraint Propagation (4_8 Constraints, 5_4_3 Constraint Languages)
 - <j-alias> 制約伝播
 - <jicst-kw> 木探索 + 探索理論 + 探索問題 + データ探索
 - 8_2_11 Relaxation Methods
 - <j-alias> 弛緩法
 - <jicst-kw> 木探索 + 探索理論 + 探索問題 + データ探索
 - 8_2_12 Marker Passing
 - <j-alias> マーカパッシング
 - 8_2_13 Bidirectional Search
 - <j-alias> 双方向探索
 - <jicst-kw> 木探索 + 探索理論 + 探索問題 + データ探索
 - 8_2_14 Data-Driven or Top Down Search
 - <j-alias> トップダウン探索, データ駆動探索
 - <jicst-kw> 木探索 + 探索理論 + 探索問題 + データ探索
 - 8_3 Heuristic Methods
 - <j-alias> 経験則, ヒューリスティック, 発見的方法
 - 8_3_2 Strategies
 - <j-alias> 方略
 - <jicst-kw> 人工知能推論 + 問題解決
 - 8_3_3 Default Reasoning <==[上位 3 Deduction]
 - <j-alias> デフォルト推論
 - <jicst-kw> 人工知能推論

- 8_3_6 Qualitative Reasoning <==[上位 3 Deduction]
 - <j-alias> 定性推論
 - <jicst-kw>人工知能推論
- 8_4 Plan Execution, Formation, Generation
 - <j-alias> プランニング, 計画
 - 8_4_0 Means-Ends Analysis
 - <j-alias> 手段-目的解析
 - 8_4_1 Forward Chaining
 - <j-alias> 前向き推論 <==[上位 3 Deduction]
 - 8_4_2 Backward Chaining
 - <j-alias> 後ろ向き推論 <==[上位 3 Deduction]
 - 8_4_4 Generate and Test
 - <j-alias> 生成-検査法
 - 8_4_5 Hierarchical Planning
 - <j-alias> 階層的プランニング, 階層的計画
 - 8_4_6 Metaplanning
 - <j-alias> メタプランニング, メタ計画
 - 8_4_7 Plan Verification
 - <j-alias> 計画の検証
 - 8_4_8 Plan Modification
 - <j-alias> 計画の修正
 - 8_4_9 Multiple Goals
 - <j-alias> 多重ゴール, 多重目標
- 8_5 Matching
 - <j-alias> マッチング, パターンマッチング, 照合
 - <jicst-kw> パターンマッチング
- 8_6 Conflict Resolution <== 対象 4_4_0 Production Rule Systems
 - <j-alias> 競合解消
- 9 Robotics
 - <j-alias> ロボティックス, ロボット
 - <jicst-kw> 知能ロボット
 - 9_0 Manipulator
 - <j-alias> マニピュレータ
 - 9_0_0 Arms ==>a_13_1_1_3
 - 9_0_1 Hands ==>a_13_1_1_5
 - 9_1 Propelling Mechanisms
 - <j-alias> 移動ロボット, 移動体
 - 9_1_0 Wheeled Vehicles
 - <j-alias> 車輪付ロボット
 - 9_1_1 Walkers
 - <j-alias> 歩行ロボット
 - 9_1_2 Underwater Vehicles
 - <j-alias> 水中ロボット
 - 9_1_3 Airborne Vehicles
 - <j-alias> 飛行ロボット
 - 9_2 Sensors
 - <j-alias> センサ, 知覚器
 - 9_2_2 Tactile Sensors
 - <j-alias> 触覚器
 - <jicst-kw> 触覚器
 - 9_2_3 Kinesthetic Sensing
 - <j-alias> 運動検出器
 - 9_2_4 Proximity Sensors
 - <j-alias> 近接センサ
 - 9_2_4_0 Acoustic Proximity Sensors
 - <j-alias> 音声近接センサ
 - 9_2_4_1 Optical Proximity Sensors
 - <j-alias> 光近接センサ
 - 9_3 Autonomous Robots <== 1_7_0 Autonomous Vehicles
 - <j-alias> 自律ロボット, 自走車, 自動走行車
 - 9_3_2 Obstacle Avoidance
 - <j-alias> 障害物回避
 - 9_3_3 Terrain Following
 - <j-alias> 地形追跡
 - 9_3_4 Robot Guidance

- <j-alias> ロボット案内
- 9_3_5 Target Tracking[==> 1_7_3 Target Tracking]
- 9_4 Manipulator Planning, Tracking, and Control
 - <j-alias> マニピュレータプランニング
 - 9_4_0 Hand-Eye Coordination
 - <j-alias> 手と目の協調
 - 9_4_1 Multiple Arm Coordination
 - <j-alias> 複数の腕の協調
 - 9_4_2 Collision Avoidance
 - <j-alias> 衝突回避
 - 9_4_3 Error Recovery
 - <j-alias> 誤り回復, 誤り訂正
 - 9_4_5 Control via Servomechanisms <==[上位 IB03_1 制御]
 - <j-alias> サーボコントロール サーボ制御
 - 9_4_6 Control via Processors
 - <j-alias> マイコン制御
- 9_5 robot control languages
 - <j-alias> ロボット制御言語
- 9_6 Cybernetics
 - <j-alias> サイバネティクス
- 11 cognitive modeling (4_13 Modeling)(IC01 心理学)
 - <j-alias> 認知モデル
 - <jicst-kw> 心理学+知覚+認識[認知科学][心理モデル]
 - 11_0 emotion modeling
 - <j-alias> 情緒モデル
 - <jicst-kw> 心理学+知覚+認識
 - 11_2 Belief Revision ==>[3_7 Belief Revision]
 - 11_6 models of memory(IC01_21 記憶)
 - <j-alias> 記憶モデル
 - <jicst-kw> 記憶構造
 - 11_6_0 semantic memory
 - <j-alias> 意味記憶
 - 11_6_1 Episodic memory and reminding
 - <j-alias> エピソード記憶
 - 11_6_2 skill memory
 - <j-alias> 技術記憶
 - 11_7 models of consciousness and phenomenology
 - <j-alias> 意識モデル 心理モデル 心モデル メンタルモデル
 - <jicst-kw> 知識表現* (心理学+知覚+認識)

一般用語, 他分野に位置する言葉

- a_1 会話(1_5_1 Question Answering)
 - <j-alias> 会話 対話
 - <jicst-kw> 会話型処理
- a_2 シミュレーション
 - <j-alias> シミュレーション 模擬実験
 - <jicst-kw> シミュレーション
 - a_2_1 アナログシミュレーション
 - a_2_2 デジタルシミュレーション
 - a_2_3 環境シミュレーション
 - a_2_4 計算機シミュレーション
 - a_2_4_1ゲーミングシミュレーション
 - a_2_4_2 プログラムシミュレーション
 - a_2_4_3 論理シミュレーション
 - a_2_5 システムシミュレーション
 - a_2_6 生体シミュレーション
 - a_2_7 ハイブリッドシミュレーション
 - a_2_8 飛行シミュレーション
- a_3 プロトコル
- a_4
- a_5
- a_6 統計
 - <j-alias> 統計
 - a_6_1 統計データ
 - <j-alias> 統計データ

- <jicst-kw> 統計
- a_6_2 統計的方法
 - <j-alias> 統計的方法
 - <jicst-kw> 統計的方法
 - a_6_2_1 実験計画法
 - a_6_2_2 統計解析
 - a_6_2_3 統計的決定<==[IA06_1_2 統計的決定]
 - a_6_2_3_1 仮説検定
 - a_6_2_3_1_1 カイ二乗検定
 - a_6_2_3_2 統計的推定
 - a_6_2_3_2_1 ベイズ推定
 - <j-alias> ベイズ推定
 - <jicst-kw> Bayes推定
 - a_6_2_3_2_2 極値解析
 - a_6_2_3_2_3 最小二乗推定
 - a_6_2_3_2_4 最優法
 - a_6_2_3_2_5 パラメータ推定
 - a_6_2_3_2_6 不偏推定
 - a_6_2_3_2_7 ロバスト推定
 - a_6_2_4 ノンパラメトリック法
 - a_6_2_5 ポピュレーションバランス
- a_7 記憶方式
 - a_7_1 仮想記憶方式
 - a_7_2 実記憶方式
 - a_7_3 スタック方式
 - a_7_4 バッファ方式
 - a_7_5 番地指定方式
 - a_7_6 連想記憶方式
- a_8 専門家(6_3_1 Dialogues with Experts)(1_15_1 Expert Knowledge)
- a_9 障害物
- a_10 確率(1_10 Mathematical_Aids)
- a_11 関数
 - a_11_1 応答関数
 - a_11_2 重み関数
 - a_11_3 帰納的関数
 - a_11_4 初等関数
 - a_11_4_1 三角関数
 - a_11_4_2 指数関数
 - a_11_4_3 対数関数
 - a_11_4_4 双曲線関数
 - a_11_5 相関関数
 - a_11_6 直交関数
 - a_11_7 メンバシップ関数
 - a_11_8 論理関数
- a_12 ハードウェア
 - a_12_1 演算プロセッサ
 - a_12_1_1 光プロセッサ
 - a_12_1_2 マイクロプロセッサ
 - a_12_2 専用プロセッサ
 - a_12_2_1 アレイプロセッサ
 - a_12_2_1_1 シストリックアレイ
 - a_12_2_2 高級言語マシン
 - a_12_2_3 データベースマシン
 - a_12_2_4 連想プロセッサ
 - a_12_3 卓上計算機
 - a_12_4 中央処理装置
 - a_12_5 電子計算機
 - a_12_5_1 アナログ計算機
 - a_12_5_2 デジタル計算機
 - a_12_5_2_1 オフィースコンピュータ
 - a_12_5_2_2 データフロマシン
 - a_12_5_2_3 パーソナルコンピュータ
 - <j-alias> パーソナルコンピュータ, パソコン
 - <jicst-kw> パーソナルコンピュータ
 - a_12_5_2_4 並列計算機
 - a_12_5_2_5 マイクロコンピュータ

- <j-alias> マイクロコンピュータ, マイコン
- <jicst-kw> マイクロコンピュータ
- a_12_5_2_6 ミニコンピュータ
- a_12_5_3 ハイブリッド計算機
- a_12_5_4 専用計算機
 - a_12_5_4_1 制御用計算機
 - a_12_5_4_2 机上計算機
 - a_12_5_4_3 ワードプロセッサ
 - <j-alias> ワードプロセッサ, ワープロ
 - <jicst-kw> ワードプロセッサ
- a_13 身体部位
 - a_13_1 四肢
 - a_13_1_1 上し
 - a_13_1_1_1 肩
 - a_13_1_1_2 ひじ
 - a_13_1_1_3 腕 <== 上位 9_0 Manipulator
 - a_13_1_1_4 手首
 - a_13_1_1_5 手 <== 上位 9_0 Manipulator
 - a_13_1_2 下し
 - a_13_1_2_1 また
 - a_13_1_2_2 大たい
 - a_13_1_2_3 ひざ
 - a_13_1_2_4 足首
 - a_13_1_2_5 足
 - a_13_2 頭部
 - a_13_2_1 顔
 - a_13_2_2 眼
 - a_13_2_3 鼻
 - a_13_2_4 ほお
 - a_13_2_5 口
 - a_13_2_6 あご
- a_14 人文科学
 - <j-alias> 人文科学, 社会科学, 社会的問題
 - <jicst-kw> 人文科学
- a_15 ソフトウェア
 - a_15_1 プロトタイプ [==> 4_0_1 Stereotypes and Prototypes]
 - a_15_2 プログラム言語, プログラミング言語 ==> [5 Programming Languages]
- a_17 効率化(IA08_1_8 効率)
- a_16
- a_18 言語
 - a_18_1 エキスパートシステム言語 ==> 1_15_0 Expert System Languages
 - a_18_2 論理型言語 ==> 3_2_2 Logic Programming Languages ロジックプログラミング言語 論理プログラミング言語 論理型言語
 - a_18_3 プログラミング言語 ==> 5 Programming Languages プログラミング言語, プログラム言語
 - a_18_4 ロボット制御言語 ==> 9_5 robot control languages ロボット制御言語
 - a_18_5 自然言語意味表現 ==> 4_12 Representation of Natural Language Semantics
- cs
 - a_18_6 自然言語類推 ==> 6_0_1 Natural Language Analogies
 - a_18_7 言語獲得 ==> 6_5 Language Acquisition 言語獲得
 - a_18_8 自然言語処理 ==> 7 Natural Language Processing 自然言語処理
 - a_18_9 自然言語(4_12 Representation of Natural Language Semantics)
- 抽象的概念
 - IA06_1 評価
 - IA06_2 統合 <== 対象 IA10_3 情報
 - IA06_3 検証
 - IA06_3_1 Programming Verification ==> [5_8_4 Programming Verification]
 - IA06_3_2 エキスパートシステム検証 ==> [1_15_7 Validation of Expert Systems]
 - IA06_5 防止
 - IA06_5_1 事故防止
 - IA06_6 決定
 - IA06_6_1 意志決定
 - IA06_6_1_1 多基準意志決定
 - IA06_6_2 統計的決定 ==> [a_6_2_3 統計的決定]

- IA06_7 選択
 - IA06_7_1 経路選択 <== 対象 I 9_3 Autonomous Robots|
 - <j-alias> 経路選択, 経路選定
 - <jicst-kw> 経路選定
 - IA06_7_2 特徴選択
 - IA06_7_3 信号選択
- IA06_8 認識
 - IA06_8_1 同定
 - IA06_8_2 鑑別
 - IA06_8_3 パターン認識
 - IA06_8_3_1 音声認識 ==>[7_4 Speech Recognition and Understanding|
 - IA06_8_3_2 図形認識
 - IA06_8_3_2_1 文字認識
 - IA06_8_3_2_1_1 手書き文字認識
 - IA06_8_3_2 話者認識 ==>[7_4_4 Speaker Recognition|
- IA08_1 性能
 - IA08_1_1 可読性
 - IA08_1_2 信頼性
 - IA08_1_2_1 信頼度
 - IA08_1_3 スループット
 - IA08_1_4 汎用性
 - IA08_1_5 分解能
 - IA08_1_6 保全性
 - IA08_1_6_1 保全度
 - IA08_1_7 安定性 <==[対象 4_4_1 Knowledge Bases|
 - IA08_1_8 効率(1_17 効率化)
- IA08_2 性質
 - IA08_2_1 移植性 <== 対象 7_10 Natural Language Interfaces
 - IA08_2_2 完全性 <== 対象 4_4_1 Knowledge Bases
 - <j-alias> 完全性, 無矛盾性
 - IA08_2_3 不完全性
 - IA08_2_4 互換性
 - IA08_2_5 あいまいさ(4_1_4 Fuzzy Logic)
 - <j-alias> あいまいさ あいまい性
 - IA08_2_6 安全性
 - IA08_2_7 インテグリティ
 - IA08_2_8 柔軟性
 - IA08_2_9 冗長性
 - IA08_2_10 一意性
 - IA08_2_11 非一意性
 - IA08_2_12 対称性
 - IA08_2_13 非対称性
 - IA08_2_14 属性
 - IA08_2_15 多義性
 - IA08_2_16 適応性
 - IA08_2_17 適性
 - IA08_2_18 等価性
 - IA08_2_19 非等価性
 - IA08_2_20 独立性
 - IA08_2_21 人間性
 - IA08_2_22 特殊性
 - IA08_2_23 品質
 - IA08_2_24 確実性
 - IA08_2_25 不確実性(IA08_2_5 あいまいさ, 4_1_4 Fuzzy Logic)
 - IA08_2_26 連続性
 - IA08_2_27 不連続性
 - IA08_2_28 有効性
 - IA08_2_29 類似性
- IA08_4 分布
- IA08_5 組成
- IA10_1 経験
- IA10_2 理論
 - IA10_2_1 制御理論(IB03_1 制御)
 - IA10_2_2 グラフ理論

- IA10_2_3 計算理論
 - IA10_2_3_1 プログラム理論 ==> [5_8 Programming Theory]
- IA10_2_4 情報理論
- IA10_2_5 探索理論 ==> [8 Problem Solving, Control Methods and Search]
- IA10_3 情報
- IA10_4 環境
- IA10_5 境界
- IA10_6 経路
 - IA10_6_1 経路選択 [==>IA06_7_1 経路選択]
- IA10_7 信号
- IA10_8 時間
 - IA10_8_1 時間表現 ==>4_11_0 Representation of Time
- IA10_9 空間
 - IA10_9_1 空間表現 ==>4_11_1 Representation of Space
- IA10_10 期間
 - IA10_10_1 時間表現==>4_11_0 Representation of Time
- IA11_1 抽出
 - IA11_1_1 特徴抽出
 - IA11_1_2 標本抽出
- IA11_2 操作
- IB01_1 観察
- IB01_2 実験
- IB02_1 検出
- IB02_2 計測
- IB03_1 制御 <==[対象 9_3 Autonomous Robots]
 - (IA10_2_1 制御理論)
 - IB03_1_1 サーボ制御 ==> 9_4_5 Control via Servomechanisms
 - IB03_1_2 自動制御
 - IB03_1_3 誤り制御
 - IB03_1_3_1 誤り検出
 - IB03_1_3_2 誤り訂正, 誤り回復 [==>9_4_3 Error Recovery]
 - IB03_1_4 プロセス制御
 - IB03_1_5 最適制御
 - IB03_1_6 マイコン制御 [==>9_4_6 Control via Processors]
- IB10_1 試験
- IC01 心理学(11 cognitive modeling)
 - IC01_1 心理測定
 - IC01_2 心理効果
 - IC01_3 識別過程
 - IC01_4 人格
 - IC01_5 性格
 - IC01_6 気質
 - IC01_7 夢
 - IC01_8 本能
 - IC01_9 欲求
 - IC01_10 欲求不満
 - IC01_11 行動
 - IC01_12 感情
 - IC01_13 感動
 - IC01_14 表情
 - IC01_15 身振り
 - IC01_16 態度
 - IC01_17 観念
 - IC01_18 意識
 - IC01_19 知能
 - IC01_20 思考
 - IC01_21 記憶(11_6 models of memory)
 - IC01_22 知覚