

〔非公開〕

TR-C-0046

知的文献検索実験システムの仕様

木下 茂行
SHIGEYUKI KINOSHITA

1990. 2. 20

A T R 通信システム研究所

知的文献検索実験システムの仕様

木下茂行

Shigeyuki KINOSHITA

ATR 通信システム研究所

2月19日, 1990

目次

1 システムの全体構成	5
1.1 概要	5
1.2 実験システムの各要素の機能概要	5
1.2.1 ユーザモデルの表現について	5
1.2.2 問合せ文理解部	7
1.2.3 ユーザモデル獲得処理部	7
1.2.4 照合処理部	8
1.2.5 真意推論部	8
1.2.6 検索式合成部	8
1.2.7 SQL 文交換部	9
1.2.8 文献検索効率評価部	9
1.2.9 文献検索効率評価部 2 (キーワードマッチング)	10
1.2.10 検索結果からの問合せ理解部	10
2 要求照合部	11
2.1 要求照合アルゴリズムの概要	11
2.2 処理例	11
2.3 検索要求文の解析について	12
2.3.1 検索要求文の解析の概要	12
2.3.2 検索要求の表現方法	12
2.4 検索要求の対象分野知識への照合	13
2.4.1 対象分野知識への照合の概要	13
2.4.2 対象分野知識への照合のためのルール	13
2.5 未知語の意味の推論	15
2.5.1 未知語の意味の推論の概要	15
2.5.2 未知語の意味の推論のためのルール	15
2.6 照合の実現方法	17
2.6.1 照合の実現方法の概要	17
2.6.2 照合のアルゴリズム	17
2.6.3 一般用語の扱い	17
2.6.4 前記のアルゴリズムによる処理例	18
2.7 関係情報を使用しない要求照合部について	19
2.7.1 関係情報を使用しない要求照合部の作成方法	19
2.7.2 対象分野知識への照合のためのルール (2.4.2 節) の変更内容	20
2.7.3 未知語の意味の推論のためのルール (2.5.2 節) の変更内容	21

3	検索式合成部	23
3.1	検索式合成部の概要	23
3.2	入力の形式	23
3.3	出力ファイルの形式	24
3.3.1	出力ファイルの例	24
3.3.2	出力ファイルのフォーマット	24
3.4	意味構造変換部	24
3.5	検索式合成部	25
3.5.1	概念ノード間の関係情報を使用しないもの	25
3.5.2	概念ノード間の関係情報を使用するもの	27
4	SQL文変換部	31
4.1	SQL文変換部の概要	31
4.2	入力ファイルの形式	31
4.2.1	入力ファイルの例	31
4.2.2	入力ファイルのフォーマット	31
4.3	出力ファイルの形式	32
4.3.1	出力ファイルの例	32
4.3.2	出力ファイルのフォーマット	33
4.4	論理式の変換について	34
4.4.1	論理式の標準形について	34
4.4.2	和積標準形への変換規則	34
5	文献検索効率評価部	37
5.1	文献検索効率評価部の概要	37
5.2	処理の流れ	38
5.3	各処理の内容	39
5.3.1	検索実行処理	39
5.3.2	入力ファイルのフォーマット	39
5.3.3	評価入力処理	40
5.3.4	分布グラフ作成処理	40
5.3.5	検索効率計算処理	40
5.3.6	推移グラフ作成処理	41
6	文献検索効率評価部2 (キーワードマッチング)	43
6.1	文献検索効率評価部2の概要	43
6.2	処理の流れ	44
6.3	各処理の内容	45
6.3.1	キーワードリスト作成処理	45
6.3.2	検索実行処理	45
6.3.3	入力ファイルのフォーマット	46
6.3.4	評価入力処理	46
6.3.5	分布グラフ作成処理	47
6.3.6	推移グラフ作成処理	48

6.4	検索式合成プログラムの処理時間について	48
6.4.1	検索式合成プログラムのアルゴリズムの概要	48
6.4.2	意味情報を使用しないアルゴリズムの処理が長時間となる理由	48
6.4.3	対策	49
6.4.4	文献検索効率評価部2（キーワードマッチング）が存在する理由	49
7	検索結果からの問合せ理解部	51
7.1	検索結果からの問合せ理解部の概要	51
7.2	フィードバック処理部	52
7.2.1	処理の概要	52
7.2.2	処理の流れ	53
7.2.3	入力の形式	53
7.2.4	出力ファイルの形式	55
7.2.5	各部の処理内容	55
7.3	ユーザモデル構築部	57
7.3.1	処理の概要	57
7.3.2	処理の流れ	58
7.3.3	入力の形式	58
7.3.4	出力ファイルの形式	59
7.3.5	各部の処理内容	59
A	文献検索効率評価実験の実行環境について	63
B	文献検索効率評価実験のお願い	65
B.1	概要	65
B.1.1	配布物	65
B.1.2	注意事項	65
B.2	締切	65

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

ここでのユーザモデルは、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分類体系を持たせる。

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。

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、上記の検索意図をすべて抽出し、それに相当するキーワードに変換することである。

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をもちいてインプリメントされる。また、文献データベースはSUN-3上のORACLEを用いて構成される。

1.2 実験システムの各要素の機能概要

1.2.1 ユーザモデルの表現について

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。リンクには、4種類の関係、階層関係、関連関係、同義関係、限定関係（対象、手段、目的、因果関係）がある。

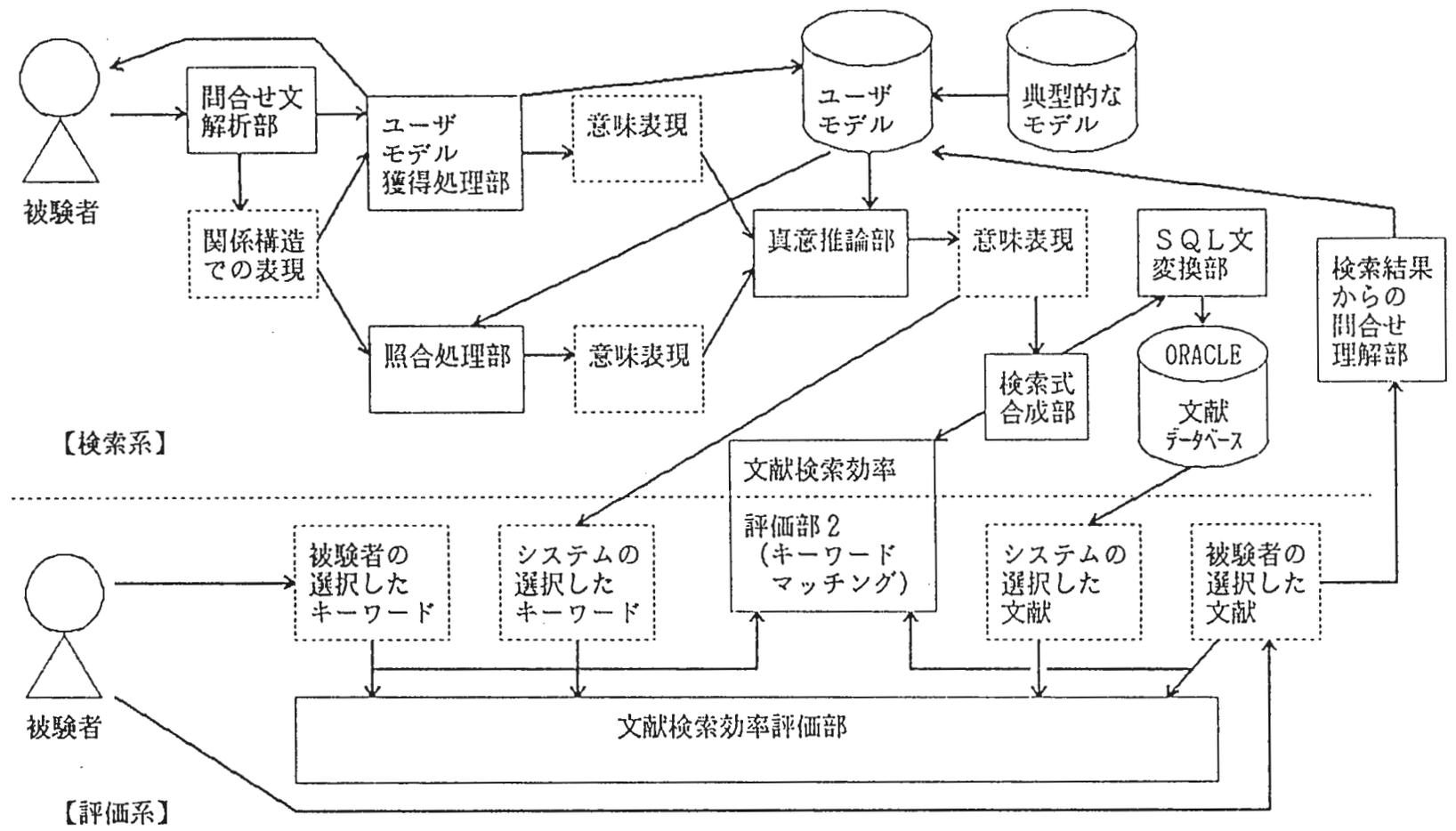


図 1.1: 知的文献検索実験システムの構成

初期状態のユーザモデルは典型的な対象世界モデルとし、利用者との対話によってこれを変形するとともに独自語を取り込む。リンクには、2つの概念間での利用者の連想の強さを示す3段階のレベル（意図のレベル：初期状態、意図する、意図しない）を持つ。これらのレベルは、隠れた検索意図を推論するために用いる。

本システムでは、ユーザモデルをARTのSCHEMAを用いて表現する。即ち、ノードをSCHEMA、リンクをSCHEMA間のリンクとして表現する。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。

1.2.2 問合せ文理解部

利用者の問合せ文は「(名詞句表現)が欲しい」という形式で表現されるものとし、この名詞句表現から概念を表現している名詞とその間の関係を抽出し表現する。問合せ文は、キーボードから日本語で入力する。

例：「意味ネットワークによる対象世界モデルの表現についての文献が欲しい」

問合せ文理解部では、この下線部を下に示すような格フレームライクな表現に変換する。

(手段 (対象 表現 対象世界モデル)
意味ネットワーク)

この処理は、構文解析ツール「PARSER」を用いる。本レポートでは、問合せ文理解部についてこれ以上は述べない。

1.2.3 ユーザモデル獲得処理部

ユーザモデル獲得処理部は、問合せ文に含まれる名詞の意味を利用者に尋ねることにより明確化しながら、問合せ文をユーザモデルに照応するものである。また同時に、ユーザモデルの構築を行なう。本レポートでは、ユーザモデル獲得処理部についてもここで記述されている以上は述べない。これは、次のような処理により行なう。

(a) 名詞の意味の詳細化

問合せ文中の名詞は、利用者の要求を特定するためのより詳細な言葉で規定されるべきであると考えられる。このため、問合せ文中の名詞に対応する概念ノードに対し、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合せ、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。

(b) 未知語に対する処理

ユーザモデルにない言葉（未知語）は、既知の名詞で規定されるべきであると考えられる。このため、問合せ文中の名詞が未知語の時は、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。

(c) 関係構造の付加と削除

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構造を表明したときには、その関係に基づいたリンクを付加する。一方、利用者がシステムから提示された関係構造を否定した時には、ユーザモデルの関係構造を修正する。

1.2.4 照合処理部

照合処理部は、利用者との対話を行なうことなく、問合せ文中の名詞に対しても、名詞間の関係情報を手がかりとして問合せ文をユーザモデルに照応することによって検索意図を抽出する。これは、次のような処理により行なう。

(a) 未知語を含まない場合の処理

問合せ文の名詞を含む関係構造全体をユーザモデルに照合し、その関係構造に最も近いユーザモデルの部分の意味構造とする。この時、名詞あるいは名詞間の関係が完全に一致しなくても、ある程度の許容範囲内で一致するものを見つけ出す。このため、照合の結果は一意ではなく、複数の候補が存在することがある。

(b) 未知語に対する処理

ある関係構造の片方の名詞が未知語で他方が既知（照応に成功したもの）の場合には、既知の名詞と名詞間の関係を使用して、次の2つの規則に従って処理する。

規則1 未知語を、既知の名詞に対応するユーザモデル上の概念ノードに対してその関係の種類に対応したリンクで繋がれた概念ノードと考える。

規則2 未知語を、既知の名詞と同じ概念ノード、あるいはその上位の概念ノードと考える。

1.2.5 真意推論部

ユーザモデル獲得処理部、および照合処理部は、いずれも直接表明された、および対話で利用者が表明した名詞の意味の理解である。推論処理部では、これらの処理に引続きユーザモデルを利用して対話を行わずに、問合せ文で使用した名詞の意味を明確化する。さらに、検索意図に相当するキーワードを選択する処理を行なう。これにより、上記の二つの手法を、対話による理解に対しては対話回数を削減するという意味で、類推による理解に対しては検索効率を高めるという意味で機能を補間する。

この処理は、問合せ文に含まれる名詞に直接に関連するユーザモデルの概念ノードから、“意図する”とラベル付けられたリンクを辿って意図しているキーワードを抽出する。これは、問合せ文中の名詞を自動的により詳細な名詞で置き換えるとともに、それに対応するキーワードを抽出することに相当する。

本レポートでは、推論処理部についてもこれ以上は述べない。

1.2.6 検索式合成部

検索式合成部は、検索要求理解部（要求照合部、個人性獲得部、あるいは真意推論部）が作成した意味表現を、検索のためのキーワードによる論理式に変換するものである。この方法は、論理式型の検索を行なうものである。

検索式合成部では、検索要求理解部で作成した意味構造が持つキーワードとその間の意味的な関連を、AND/ORの関係に変換することにより、検索式（最適検索式）を作成する。意味構造は、ある概念（キーワード）とそれを修飾する概念（キーワード）を表現していると考えられる。この修飾関係をAND（同じ修飾関係のものが複数あればそれをOR）で表現することにより、検索式を作成する。これは、意味構造中のすべての条件を記述した、最も検索条件のきついものである。

この意味構造から、他の概念（キーワード）を修飾するもののうち最も修飾関係が弱いと思われるものを順次削除して行くことにより、複数の検索式を作成する。この複数の検索式を条件のきついものの順に評点付けし、文献を順位付け出力するものである。

この最適検索式の作成と、その後の検索条件の弛緩は、意味表現の中に表現されている概念ノード間の関係情報を使用したものと、使用しないもの二種類の処理を行うことにより、この両者を比較する。概念ノード間の関係情報を使用しないものは、意味構造のノードの結合関係のみから最適な（最も条件のきつい）検索式を合成する。また、最適な検索式から順次機械的に条件を緩めた検索式を合成する。概念ノード間の関係情報を使用したものは、意味構造のノードの結合関係および結合の意味情報から最適な（最も条件のきつい）検索式を合成する。また、結合の意味情報を使用して検索に与える影響の少ないと思える条件から順次緩めた検索式を合成する。

1.2.7 SQL 文変換部

検索式合成部が作成した論理式は、LISP の S 式で記述されている。SQL 文変換部は、これを ORACLE 上で実現されている文献データベースを検索するための SQL 文に変換するものである。検索式合成部では最適検索式から、順次条件を緩めた検索式を複数作成する。SQL 文変換部では、これらの複数の検索式を一連に処理し、結果を一つのテーブルに格納するための SQL 文を作成する。

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、文献検索効率の評価を行なう。

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、問合せ文から検索要求理解部（要求照合部、個人性獲得部、あるいは真意推論部）および検索式合成部を通して作成し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○×（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×）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。これらの検索効率は、対話処理（個人性獲得部+真意推論部）終了後、および類推処理（要求照合部+真意推論部）終了後に得られるものだけでなく、対話処理の各段階、および類推処理の照合処理終了時と推論処理終了時における適合率/再現率の平均値とその推移も求める。

1.2.9 文献検索効率評価部2 (キーワードマッチング)

文献検索効率評価部2 (キーワードマッチング) は、検索式合成部の意味表現の中に表現されている概念ノード間の関係情報を使用しないものを代替するものである。これを作成する理由は、検索式合成部の概念ノード間の関係情報を使用しない手法は、入力のあるケースでは組合せの爆発を起し、処理が適切な時間で終了しないことがあるのを避けるためである。

この方法は、検索要求理解部 (要求照合部、個人性獲得部、あるいは真意推論部) で作成した意味構造に含まれるキーワードを抽出し、これを文献に付与されているキーワードと照合し、一致したキーワード数で文献を評点付けする。この評点にしたがって、文献を順位付け出力するものである。

この順位付け出力された文献を、利用者が付与した○×に従って評価する。即ち、意味構造と文献のキーワードのマッチングの程度に応じた評点を元に文献検索効率を計算するものである。

1.2.10 検索結果からの問合せ理解部

システムが検索した文献のうち、利用者が選択したものに含まれるキーワードを解析することにより、利用者があまり意識していない検索意図を抽出することができる。フィードバック処理部では、このような検索意図を、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次の条件を満たすものであると考える。

- 高い頻度で出現するキーワード
- 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

これらの検索意図を代表するキーワードのうち、以前の間合せ理解処理で抽出でなかったものを抽出する。このようにして新たに抽出したキーワードを以前に抽出していたものに加えることにより、より検索効率の高い文献を検索できると考えられる。

検索結果からのユーザモデルの間合せ理解部では、上記の基準で利用者が選択した文献集合 (即ち、最終的に利用者の検索意図に適合すると判断された文献) から抽出されたキーワードと他のキーワードとの関連を抽出し、それに基づいてユーザモデルのノード間のリンクを修正するものである。これは、利用者が選択した文献集合中で抽出されたキーワードともっとも強い共起関係にあるキーワードと関連があると見なすことである。即ち、抽出されたキーワードを最も共起関係の強い名詞の意味を明確化するものと位置付けることである。関連の種類、および修正の方法は、次の通りである。

- ユーザモデル上に2つのキーワードノード間ですでに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。
- ユーザモデル上二つのキーワードノード間に関連が存在しないならば、それらのノード間に関連関係のリンクを追加する。

第2章

要求照合部

2.1 要求照合アルゴリズムの概要

要求照合アルゴリズムは、次のような特徴を持つ。

- 検索要求文の解析の結果得られた格表現を、ボトムアップに照合する。
- 照合で得られる複数の候補のそれぞれをビューポイントに対応させ、複数の仮説を同時に持ったまま推論を行う。

2.2 処理例

検索要求文：「意味ネットワークによる対象世界モデルの表現について調べたい」

検索要求文を解析

上記の検索要求文は、解析の結果下に示すような格表現となる。

格表現：(手段 (限定 表現 対象世界モデル)
意味ネットワーク)

検索要求文の理解

上記の検索要求文は、下に示すようなプロセスで対象分野知識と照合する。

対象世界モデル：未知語

意味ネットワーク：意味ネットワーク(4.5)

表現 (限定 表現 対象世界モデル) (手段 (限定 表現 対象世界モデル)
意味ネットワーク)

専門知識の表現(1.15.3)	専門知識の表現(1.15.3)	(手段 1.15.3 4.5)
	エキスパートシステム(1.15)	(手段 1.15 4.5)
知識表現(4)	知識表現(4)	(手段 4 4.5)
知識表現言語(5.4)	知識表現言語(5.4)	(手段 5.4 4.5)
	プログラミング言語(5)	(手段 5 4.5)

右端のつの仮説のうち、照合のレベル(4レベルに分類)の高いものを採用。

表 2.1: 概念間の関係の分類

概念間の関係	関係の意味 (関係名)	使用される言葉 (助詞)
非関係的言及	併記	と および や または あるいは
階層的関係の 言及	限定	の での における にたいする
	例示	をふくむ などの としての とくに たとえば
非階層的関係 の言及	手段	による をもちいた をつかった を使用した
	目的	のための にもちいる
	状況	における での
	(対象	への に対する)
	因果関係	による

2.3 検索要求文の解析について

2.3.1 検索要求文の解析の概要

検索要求文を対象分野知識と照合するために、検索要求文の中に含まれる概念構造を表現する必要がある。検索要求は、「(名詞句表現)が欲しい」(直接的表現)という形式で言われると仮定する。この名詞句表現の中に、利用者が表現した検索要求の概念構造がある。検索要求文に現れる名詞=概念と考えると、概念間の関係としては次のものがある。

これらの関係は、一般的には格表現によって表すが、ここでは以下の照合の方法の簡単化のために、等価的に二つの概念とその間の関係の組合せとして表現する。

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 表現 対象世界モデル)
意味ネットワーク)

複雑な検索要求文の表現では、上記の名詞の位置に概念構造(格表現)が現れる。

格表現の概念間の表現を概念間の表現で行うことにより、次のような利点がある。

- 検索要求の理解の単位がつの概念とその間の関係になる。これは対象分野知識上のつの概念ノードとその間のパスを考えることになり、理解(照合)の記述が簡単になる。また、検索要求文全体の理解も、つの概念が一つの名詞であるか、或は再び概念間の関係であるので、前述の理解の単位を再帰的に記述すればよい。
- 検索要求文の記述が木表現になっており、これをボトムアップに照合すればよい。即ち、照合の手順も記述していることになる。

2.3.2 検索要求の表現方法

検索要求文は、パーザからの出力として次のように表現される。

```
((action retrieval)
(object ((subject (前述の格表現))))))
```

ここで、それぞれのものは次のような意味を持つ。

action : 入力文の種類を表す。

retrieval : 入力文が検索要求文であることを表す。

object : 入力文（ここでは検索要求文）の対象を表す。

subject: 検索要求文が、主題 (subject) について述べている事を表す。

格表現 : (関係名 概念 1 概念 2) という形式で表現する。

但し、1) 関係名は、表の関係名を使用する。

2) 概念 1 あるいは概念 2 も上記の字組で表現される事もある。

概念 1 あるいは概念 2 の表現方法（上記の字組で表現されない場合）：

1) 単一の用語の場合は、次のように表現する

('special 用語) : (専門用語の場合)

('general 用語) : (一般用語の場合)

2) 複合の用語の場合は、1) の用語の部分を上記の表現のリストで表す。

(例)

('special (('special 用語 1) ('general 用語 2))

2.4 検索要求の対象分野知識への照合

2.4.1 対象分野知識への照合の概要

パーザから出力された検索要求文は、まず対象分野知識と照合する。検索要求文で使用される言葉（名詞）の制約がないため、この照合は一意には行われぬ。このため、可能性のある候補を残しながら、推論を進める。この複数の候補は、ビューポイントとして表現する。

この複数の候補のうち、有望な物を採用する為に、照合のレベルを4つ（完全、準完全、部分、失敗）に分類し、レベルの高いものから順次仮説として使用する。（但し、失敗は仮説を生成しない）

2.4.2 対象分野知識への照合のためのルール

対象分野知識への照合へは、以下で述べるルールを使用する。

(a) 単一の概念との照合

問い合わせを表現している用語と、対象世界知識の概念を表現する用語の照合を行う。

- 完全 : ストリングとして等しい
- 準完全 : 問い合わせを表現している用語が対象世界知識の概念を表現する用語のサブストリングである
- 部分 : 問い合わせを表現している用語の一部（単語）¹が対象世界知識の概念を表現する用語のサブストリングである
- 失敗 : 上記以外

例 : 「自然言語」

7 Natural Language Processing と準完全照合

(対象分野知識の Natural Language Processing を表すスキーマ中の j-alias に「自然言語処理」を持つ)

(b) 二つの概念とその間の関係との照合

二つの概念が対象分野知識上の同じ概念に照合する場合 ⇒ 全体でその概念に照合

- 完全 : 二つの概念が共に完全、或はどちらかが準完全或は二つの概念を結合したストリングが、対象分野知識上の一つの概念にストリングとして等しい
- 準完全 : 二つの概念が共に準完全、或は二つの概念の片方が完全或は準完全で、他方が部分完全
- 部分 : 二つの概念が共に部分完全
- 失敗 : 上記以外

例：「専門の知識」 (限定 知識 専門)

知識 : 1.15.3 Representation of Expert Knowledge (j-alias 専門知識) に照合

専門 : 1.15.3 Representation of Expert Knowledge (j-alias 専門知識) に照合

↓

専門知識 : (限定 知識 専門) の二つの概念 “知識” と “専門” を結合したストリング
: 1.15.3 Representation of Expert Knowledge (j-alias 専門知識) に完全照合

二つの概念が対象分野知識上の別の概念に照合する場合 ⇒ 次のように、関係の種類に応じて処理する。

関係が併記の場合

二つの概念の対象分野知識での関係にはかかわらずに照合する。

- 完全 : 二つの概念が共に完全
- 準完全 : 二つの概念が共に準完全、或は片方が完全
- 部分 : 二つの概念が共に部分完全、或は片方が完全或は準完全
- 失敗 : 上記以外

例：「推論方式と(その)適用例」 (併記 推論方式 適用例)

推論方式 : 3 Deduction と照合

適用例 : 1 Applications and Expert Systems と照合

関係が限定の場合

二つの概念は対象分野知識での関係は階層関係あるいは対象関係にあるとして照合する。

- 完全 : 二つの概念が共に完全で、両者に対象分野知識上で階層関係あるいは対象関係²が存在する
- 準完全 : 二つの概念が共に完全、或は二つの概念のどちらか (或は両方) が準完全
- 部分 : 二つの概念のどちらか (或は両方) が部分完全
- 失敗 : 上記以外

例：「ロジックプログラミングの理論」 (限定 理論 ロジックプログラミング)

理論 : 3.2.0 Logic Programming Theory と照合

ロジックプログラミング : 3.2 Logic Programming と照合

関係が例示の場合

二つの概念は対象分野知識での関係は階層関係にあるとして照合する。階層関係は単純限定より強い制約と考える。

- 完全 : 二つの概念が共に完全で、両者に対象分野知識上で階層関係が存在する
- 準完全: 二つの概念が共に完全、あるいは二つの概念のどちらか（或は両方）が準完全で、両者に対象分野知識上で階層関係が存在する
- 部分 : 二つの概念が共に準完全、或は二つの概念のどちらか（或は両方）が部分完全で、両者に対象分野知識上で階層関係が存在する
- 失敗 : 上記以外

例: 「機械翻訳などの自然言語処理」 (例示 自然言語処理 機械翻訳)

自然言語処理 : 7 Natural Language Processing と照合

機械翻訳 : 7.3 Machine Translation と照合

関係が手段、目的、状況、(対象)、因果関係の場合

二つの概念は対象分野知識での関係は非階層関係にあるとして照合する。

- 完全 : 二つの概念が共に完全で、両者に対象分野知識上で非階層関係が存在する
- 準完全: 二つの概念が共に完全、或は二つの概念のどちらか（或は両方）が準完全で、両者に対象分野知識上で関係（階層或非階層）が存在する
- 部分 : 二つの概念のどちらか（或は両方）が準完全、あるいは片方が完全で他方が部分完全、あるいは両方が部分完全で、両者に対象分野知識上で関係（階層或非階層）が存在する
- 失敗 : 上記以外

例: 「意味ネットワークを使った知識表現」 (手段 知識表現 意味ネットワーク)

知識表現 : 4 Knowledge Representation と照合

意味ネットワーク : 4.5 Semantic Network と照合

2.5 未知語の意味の推論

2.5.1 未知語の意味の推論の概要

上記の照合で失敗した概念を、未知語であると考え、この未知語に対して、他方の概念が照合に成功した場合には次のようなルールにより既知の概念からその意味を推論する。この推論においても、未知語の意味に対して複数の解釈（仮説）が発生するが、現在の所複数の仮説に対する優先順位付は考えていない。また、下記のルールで推論できない未知語は、その仮説を棄却する。

2.5.2 未知語の意味の推論のためのルール

未知語の意味は下記のルールで推論する。このとき、未知語として意味を推論されたルールのマッチングのレベルは、他方の既知の概念のマッチングのレベルを一ランク下げたもの（完全ならば準完全、準完全ならば部分完全、部分完全ならば失敗）とする。

(a) 概念間の関係を使用しない推論

既知の概念と同義であると解釈する

一つ概念に対して複数の名詞を使った表現（名詞句表現）をした場合に相当する

例：「工程管理と進捗管理」（併記 工程管理 進捗管理）
 工程管理：1.2.4 Planning for Production と照合
 進捗管理：未知語 → 1.2.4 Planning for Production と推論する
 ↓
 (併記 工程管理 進捗管理)：1.2.4 Planning for Production

全体として、既知の概念の上位の概念に対応すると解釈する

例：「工程管理と進捗管理」（併記 工程管理 進捗管理）
 工程管理：1.2.4 Planning for Production と照合
 進捗管理：未知語
 ↓
 (併記 工程管理 進捗管理)：1.2 Industrial Applications

(b) 概念間の関係を使用した推論

関係が併記の場合

二つの概念の対象分野知識での関係は仮定していないので、上記のルールのみを使用する。

関係が限定、例示の場合

二つの概念は対象分野知識での関係は階層関係にあると仮定する。即ち、未知の概念は既知の概念の対象分野知識上の上位（或は下位）の概念であると推論する。

例：「航空機の監視」（限定 監視 航空機）
 監視：未知語
 航空機：1.17 Aviation Application
 ↓
 監視：1.17.1 Air Traffic Control

関係が手段、目的、状況、（対象）、因果関係の場合

- 二つの概念は対象分野知識での関係は非階層関係にあると仮定する。即ち、未知の概念は既知の概念の対象分野知識上で非階層関係で結合されていると推論する。

例：「病院業務における知識ベース」（状況 知識ベース 病院業務）
 知識ベース：4.4.1 Knowledge Bases
 病院業務：未知語
 ↓
 病院業務：1.4.0 Medical Applications
 (4.4.1 Knowledge Bases と 1.4.0 Medical Applications の間に非階層関係がある場合)

- 関係の意味によって、未知の概念の上位の概念を推論することができる。

例：関係が目的で目的を表す概念が未知 ⇒ 「応用」の下位の概念であると推論

2.6 照合の実現方法

2.6.1 照合の実現方法の概要

前述の検索要求文の対象分野知識への照合および未知語の意味の推論は、検索要求文の格 V 表現に対する置き換え規則として記述する。則ち、検索要求文で利用者が使用した言葉は、置き換え規則により順次（ボトムアップに）対象分野知識の概念に置き換えることにより照合を行う。

照合の最終結果は、意味表現として対象分野知識と同様な表現を行う。則ち、対象分野知識のサブセットが意味表現であると考えられる事が出来る。途中経過は、検索要求文の格表現（二進木表現）中のノードが、対応する対象分野知識の概念及びそれ等の関係（意味表現）に関連付けられる。二進木のトップレベルのノードの対応付けが終了した時点で、照合は終了する。

2.6.2 照合のアルゴリズム

- (1) 検索要求文の格表現を二進木表現に変換する。
- (2) 二進木表現の葉ノードの照合を行う。対応する対象分野知識の概念があれば、その対象分野知識の概念をコピーし、葉ノードと関連付けを行う。対応する対象分野知識の概念がなければ、前述の未知語の意味の推論ルールによる推論を行う。照合のレベル（完全、準完全、部分、失敗）を葉ノードに書いておく。
- (3) 二進木表現の葉ノード以外のノードの照合を行う。このとき、下位のノードの両方が既知（完全、準完全、部分）であれば、照合ルール、片方が未知（失敗）であれば未知語の意味の推論ルールによる推論を行う。また、両方が未知であれば、そのノードの照合のレベルも失敗とする。いずれの場合も、照合のレベルをそのノードに書いておく。
照合に成功（完全、準完全、部分）すれば、下記のルールに従って対応する対象分野知識の概念コピーし、意味表現を作成してそのノードと関連付けを行う。

- － 単一の対象分野知識の概念に対応する場合
step(2)と同様。
- － 複数の対象分野知識の概念に対応する場合（関係名が併記）
下位のノードに関連付けられているノードをそれぞれそのままそのノードと関連付ける。
- － 複数の対象分野知識の概念に対応する場合（関係名が併記以外）
そのノードを三字組（関係名, 概念 1, 概念 2）で表現した場合の概念 1 を上位、概念 2 を下位とし、関係名を両者の関係とする構造を作成し、概念 1 をそのノードと関連付ける。下位のノード（概念 1 あるいは概念 2）が複数の関連付けられた対象分野知識の概念を持っていれば、概念 1 および概念 2 に関連付けられた対象分野知識の概念の総ての組み合わせに対して、上記の処理を行う。

- (4) トップレベルのノードに関連付けられる対象分野知識の概念の構造が作成されたならば、それを意味表現とする。トップレベルのノードの照合のレベルが未知（失敗）であれば、その照合場失敗である（ビューポイントを削除する）。

2.6.3 一般用語の扱い

検索要求文の中で名詞（葉ノード）として表れる言葉には、専門用語と一般用語がある。前節で述べたアルゴリズムは専門用語の場合についてである。一般用語については、基本的には照合の対象とし

ないことで処理する。具体的な一般用語の取り扱いのルールを下記で示す。

三字組 (関係名, 概念1, 概念2) に対して、

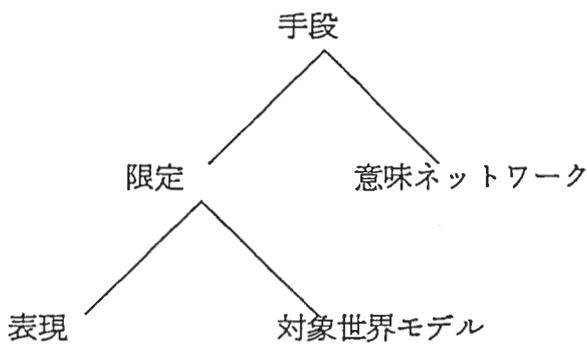
- 概念1あるいは概念2が一般用語の場合には、全体として一般用語でないものと置き換える。
- 概念1および概念2の両方一般用語の場合には、全体として一般用語であるとする。
- 最上位のノードが一般用語の場合には、照合は失敗である。

2.6.4 前記のアルゴリズムによる処理例

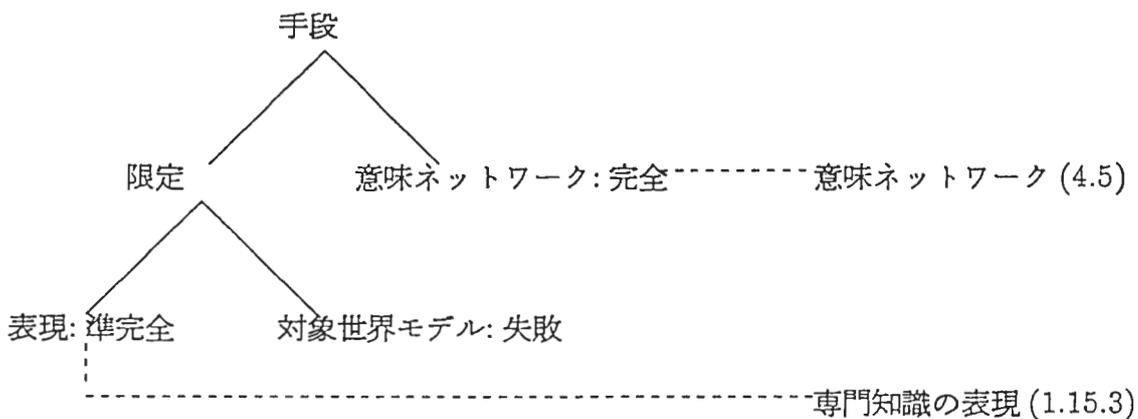
検索要求文：「意味ネットワークによる対象世界モデルの表現について調べたい」

格表現：(手段 (限定 表現 対象世界モデル)
意味ネットワーク)

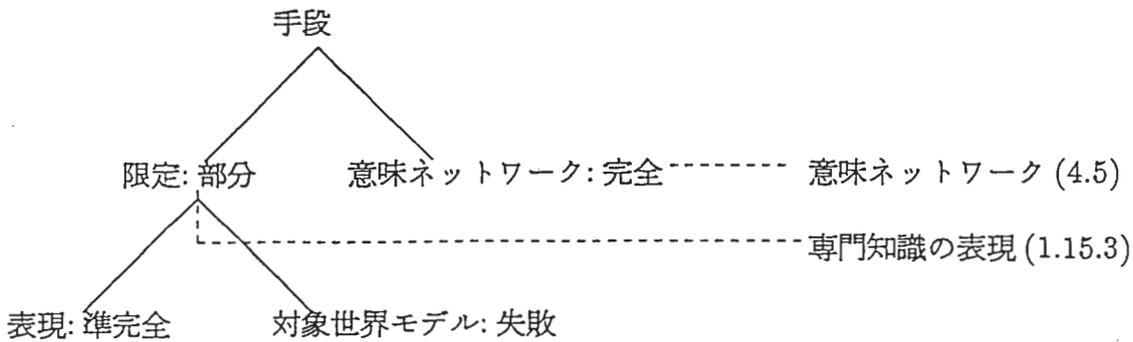
step(1): 検索要求文の格表現を二進木表現に変換



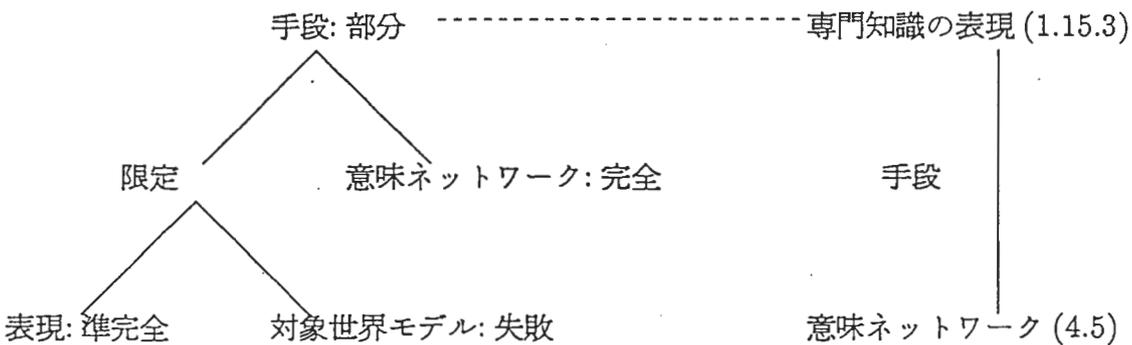
step(2): 葉ノードの照合



step(3-1): 上位のノードの照合



step(3-2): 最上位のノードの照合



2.7 関係情報を使用しない要求照合部について

概念間の関係情報を使用しない要求照合部を以下に示す。これは、既存の要求照合部（関係情報を使用するもの）を、次のように変更して作成する。

2.7.1 関係情報を使用しない要求照合部の作成方法

関係情報を使用しない要求照合部は、当節まで述べてきた要求照合部（以下、要求照合部（関係情報を使用するもの）と呼ぶ）の一部を変更して作成する。要求照合部（関係情報を使用するもの）の2.4.2節、対象分野知識への照合のためのルールの箇所および2.5.2節、未知語の意味の推論のためのルールの概念間の関係を使用した推論の箇所を以下で示すものに置き換える。それ以外の部分については、要求照合部（関係情報を使用するもの）と同じ。

2.7.2 対象分野知識への照合のためのルール (2.4.2 節) の変更内容

(a) 単一の概念との照合

問い合わせを表現している用語と、対象世界知識の概念を表現する用語の照合を行う。

- 完全 : スtringとして等しい
- 準完全: 問い合わせを表現している用語が対象世界知識の概念を表現する用語のサブStringである
- 部分 : 問い合わせを表現している用語の一部 (単語)³が対象世界知識の概念を表現する用語のサブStringである
- 失敗 : 上記以外

例: 「自然言語」

7 Natural Language Processing と準完全照合

(対象分野知識の Natural Language Processing を表すスキーマ中の j-alias に「自然言語処理」を持つ)

(b) 二つの概念とその間の関係との照合

二つの概念が対象分野知識上の同じ概念に照合する場合 ⇒ 全体でその概念に照合

- 完全 : 二つの概念が共に完全、或はどちらかが準完全或は二つの概念を結合したStringが、対象分野知識上の一つの概念にStringとして等しい
- 準完全: 二つの概念が共に準完全、或は二つの概念の片方が完全或は準完全で、他方が部分完全
- 部分 : 二つの概念が共に部分完全
- 失敗 : 上記以外

例: 「専門の知識」 (限定 知識 専門)

知識: 1.15.3 Representation of Expert Knowledge (j-alias 専門知識) に照合

専門: 1.15.3 Representation of Expert Knowledge (j-alias 専門知識) に照合

↓

専門知識: (限定 知識 専門) の二つの概念 “知識” と “専門” を結合したString
: 1.15.3 Representation of Expert Knowledge (j-alias 専門知識) に完全照合

関係が併記の場合

二つの概念の対象分野知識での関係にはかかわらずに照合する。

- 完全 : 二つの概念が共に完全
- 準完全: 二つの概念が共に準完全、或は片方が完全
- 部分 : 二つの概念が共に部分完全、或は片方が完全或は準完全
- 失敗 : 上記以外

例: 「推論方式と (その) 適用例」 (併記 推論方式 適用例)

推論方式: 3 Deduction と照合

適用例 : 1 Applications and Expert Systems と照合

関係が例示、限定、手段、目的、状況、因果関係の場合

二つの概念は対象分野知識での関係は何らかの関係（階層関係あるいは非階層関係）にあるとして照合する。

- 完全 : 二つの概念が共に完全で、両者に対象分野知識上で何らかの関係（階層関係あるいは非階層関係、関係の方向は問わない）が存在する
- 準完全 : 二つの概念のどちらか（或は両方）が準完全で、両者に対象分野知識上で何らかの関係（階層関係あるいは非階層関係、関係の方向は問わない）が存在する
- 部分 : 二つの概念が共に準完全、或は二つの概念のどちらか（或は両方）が部分完全
- 失敗 : 上記以外

例：「機械翻訳などの自然言語処理」（例示 自然言語処理 機械翻訳）

自然言語処理 : 7 Natural Language Processing と照合

機械翻訳 : 7.3 Machine Translation と照合

2.7.3 未知語の意味の推論のためのルール（2.5.2 節）の変更内容

未知語の意味の推論のためのルール（2.5.2 節）のうち、概念間の関係を使用した推論の箇所を以下で示すものに置き換える。

関係が併記の場合

二つの概念の対象分野知識での関係は仮定していないので、上記のルールのみを使用する。

関係が限定、例示、手段、目的、因果関係の場合

二つの概念は対象分野知識での関係は何らかの関係にある仮定する。即ち、未知の概念は既知の概念の対象分野知識上のいずれかのリンクで結合された概念であると推論する。

例：「航空機の監視」（限定 監視 航空機）

監視 : 未知語

航空機 : 1.17 Aviation Application

↓

監視 : 1.17.1 Air Traffic Control

第3章

検索式合成部

3.1 検索式合成部の概要

検索式合成部は、検索要求理解部（要求照合部、個人性獲得部、あるいは真意推論部）が作成した意味表現を、検索のためのキーワードによる論理式に変換するものである。次のような特徴を持つ。

- 意味表現の中に表現されている概念ノード間の関係情報を使用した検索式の合成と、使用しない検索式の合成の二種類の処理を行う。
- 出力は、ファイルに書き出す。ファイルには、LISPのS式で論理式が記述されている。また、論理式は特別な制約はなく記述されるものとする。

処理は、大きく次の2つのフェーズで行われる。

- フェーズ1（検索式合成用意味構造への変換）
真意理解部が出力した意味構造から、検索式合成に必要な情報（階層上位のキーワードなど）を復元する。
- フェーズ2（検索式の合成）
変換した意味構造をもとに、検索式を合成する。概念ノード間の関係情報を使用したものと使用しないものの二種類の処理を行う。
 - 概念ノード間の関係情報を使用しないものは、意味構造のノードの結合関係のみから最適な（最も条件のきつい）検索式を合成する。また、最適な検索式から順次機械的に条件を緩めた検索式を合成する。
 - 概念ノード間の関係情報を使用したものは、意味構造のノードの結合関係および結合の意味情報から最適な（最も条件のきつい）検索式を合成する。また、結合の意味情報を使用して検索に与える影響の少ないと思える条件から順次緩めた検索式を合成する。

3.2 入力形式

入力は、検索要求理解部（要求照合部、個人性獲得部、あるいは真意推論部）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。これらの詳細な説明は省略する。

3.3 出力ファイルの形式

3.3.1 出力ファイルの例

```
(t002 rel)
(100 f001 (and 3.2.1 4.3 (or 3.1 (and 4.3 5.6)) (or 11.0 12.3)))
(90 f002 (or 3.2.1 (and 4.3 (or 3.1 (and 4.3 5.6)) 11.0)))
```

3.3.2 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レコード目以降
(評点 検索式 No. 論理式)
 評点 : 3桁の数字 (max 100)
 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。

3.4 意味構造変換部

意味構造変換部では、次の三つの処理を行う。

- (1) 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。
 - 真意理解部が出力した意味構造のうち、真意理解部がトレースした (活性化フラグがオンまたはトレース) キーワードノード
 - 真意理解部が出力した意味構造のうち、真意理解部がトレースした (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
- (2) 階層上位のキーワードの追加
(1) の処理で作成した構造に含まれるキーワードノードのうち、階層上位で意図の強さが“意図する”であるキーワードノードを構造に追加 (該当のキーワードノードとは階層関係のリンクで結合する) する。この処理は、追加するキーワードノードが無くなるまでリカーシブに実行する。
- (3) 階層上位のキーワードのマーキング
(2) の処理で作成した構造に含まれるキーワードノードのうち、階層リンクで結合されたものを最下位のものを除いてマーキングする。

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）

3.5 検索式合成部

3.5.1 概念ノード間の関係情報を使用しないもの

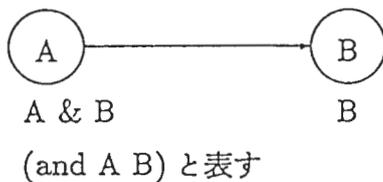
(a) 最適検索式の合成

以下に述べる規則に従い、変換した意味構造からキーワードノードの階層番号スロットの値をキーワードとして使用しそれらを論理演算子 (and/or) で結合することにより、検索式を合成する。最適検索式とは意味構造のすべての情報を用いた最も検索条件のきつい検索式のことである。また、最適検索式の評点は 100 とする。

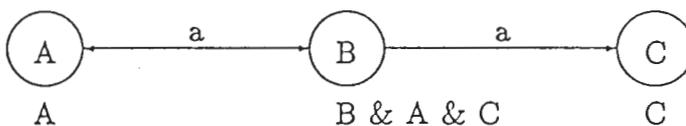
最適検索式は、終端ノード（そのノードが持っている有向リンクがすべてそのノードを終点とする）から出発し、以下の規則に各ノードに論理式を割り当てる（ラベリング）することにより合成する。すべての開始ノード（そのノードが持っている有向リンクがすべてそのノードを始点とする）がラベリングされた時点で処理を終了する。最適検索式は、各開始ノードに割り当てられた論理式を or で結合したものである。

規則-1 終端ノードは、そのノードが持っている階層 No. をラベルとする。

規則-2 ただ一つの出力リンクを持つノードは、そのノードが持っている階層 No. と、出力リンクの終点ノードが持っている階層 No. を and で結合したものをラベルとする。

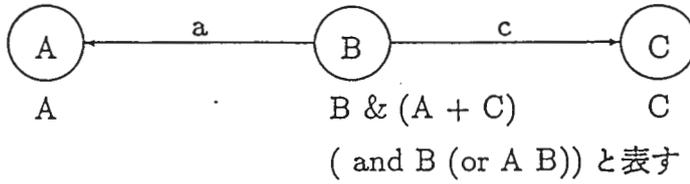


規則-3 複数の出力リンクを持つノードは、リンクの種類が異なるものに対しては、そのノードが持っている階層 No. と、すべての出力リンクの終点ノードが持っている階層 No. を and で結合したものを更に and で結合したものをラベルとする。

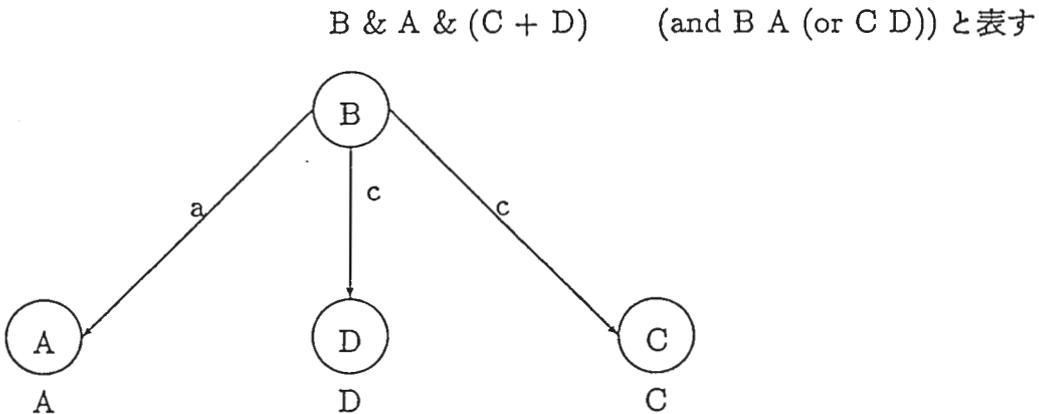


規則-4 複数の出力リンクを持つノードは、リンクの種類が同じものに対しては、そのノードが持っている階層 No. と、すべての出力リンクの終点ノードが持っている階層 No. を or で結合したものを更

に and で結合したものをラベルとする。



規則-5 複数の出力リンクを持つノードのうちリンクの種類が同じものと異なるものが混在する場合には、or の結合を優先する。



(b) 最適検索式から検索条件を緩めた検索式の合成

検索条件を緩めた検索式の合成は、次の2つのステップで行う。

step-1 最適検索式の最簡形の積和標準形への変換 (最適検索式の展開)

特別な制約なしに記述されている最適検索式 (プレフィックス形式) を展開し、積和標準形 (プレフィックス形式) に変換する。

例:

$$(A+B)*(A+C)+D*E*F : (\text{or } (\text{and } (\text{or } A \text{ B}) (\text{or } A \text{ C})) (\text{and } D \text{ E } F))$$

↓

$$A+B*C+D*E*F : (\text{or } A (\text{and } B \text{ C}) (\text{and } D \text{ E } F))$$

step-2 検索条件を緩めた検索式の合成

step-1 で作成した和積標準形の最適検索式から、次の規則に基づいて検索条件を緩めた検索式を合成する。

- 以前に作成した検索式のうち、2つ以上のキーワードの積になっている項のうちの1つ (例えば、 $(\text{and } D \text{ E } F)$) を、そのうちの1つのキーワード除いたものの総ての組み合わせを or で結合したもの (例えば、 $(\text{or } (\text{and } E \text{ F}) (\text{and } D \text{ F}) (\text{and } D \text{ E}))$) で置き換えたものを新たな検索式とする。

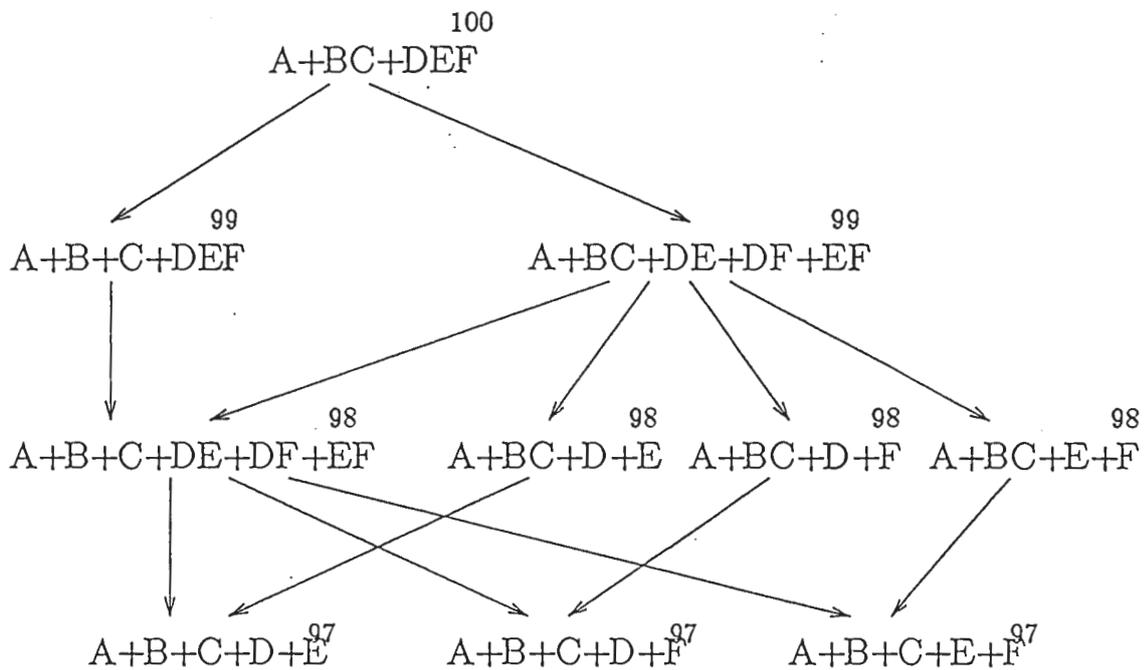
- 合成した検索式は、最簡形の積和標準形で表現する。
- 上記の規則は、すべての項について行なう。
- 以前に作成した検索式の評点から、1を引いたものを新たな検索式の評点検索式とする。

(c) 積和標準形への変換規則

論理式は次の規則に基づいて積和標準形へ変換する。

- 規則-1 : $A * A = A$
- 規則-2 : $A + A = A$
- 規則-3 : $A + A * B = A$
- 規則-4 : $A * (B + C) = A * B + A * C$
- 規則-5 : $(A + B) * (C + D) = A * C + A * D + B * C + B * D$

[処理例]



3.5.2 概念ノード間の関係情報を使用するもの

(a) 最適検索式の合成

概念ノード間の関係情報を使用しない場合と同じ。

(b) 最適検索式から検索条件を緩めた検索式の合成

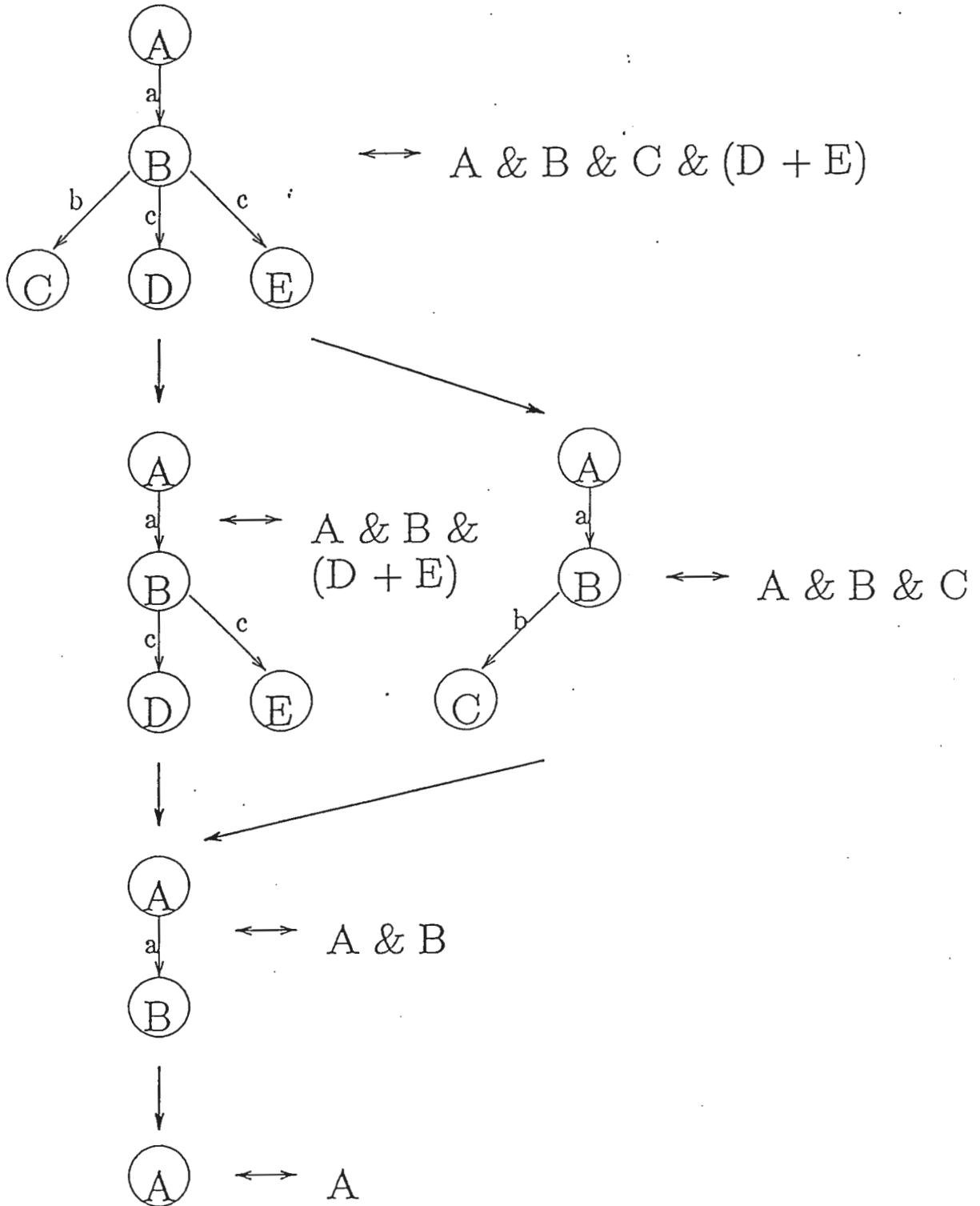
次の規則に基づいて、以前に作成した意味構造を変形（検索条件の緩和）し、変形した意味構造に対して最適検索式の合成の場合と同じ規則を用いて検索式を合成する。

- 開始ノード（そのノードが持っている有向リンクがすべてそのノードを始点とする、複数ありうる）からの距離が一番遠いノードを削除する。このとき、そのノードと同じリンクで結合されている並列ノードがあれば、これも同時に削除する。評点は以前の意味構造のものから、削除されるノードが持っているリンクに応じて次の表の値を引くことにより求める。尚、複数のノードを削除した場合でも1つの種類のリンクであれば1つのリンクと考える。複数のリンクがあれば、それらの和を引く。

リンクの種類	評点
階層関係	1
目的限定	1
手段限定	1
対象限定	1
状況限定	1
因果関係	1
関連関係	1

- 終端ノードがマーキングされたものである場合には、そのノードのラベルをユーザモデル上で“意図しない”とラベル付けられたものを除くすべての階層下位ノードとそのノードを or で結合したものに置き換える。
- 意味構造が開始ノードのみになった時点で、処理を終了する。

[処理例]



第4章

SQL 文変換部

4.1 SQL 文変換部の概要

SQL 文変換部は、検索式合成部が作成した検索のためのキーワードによる論理式を ORACLE の SQL 文に変換するものである。次のような特徴を持つ。

- 検索式合成部が作成した論理式は、ファイルから入力する。ファイルには、LISP の S 式で論理式が記述されている。また、論理式は特別な制約はなく記述されているものとする。
- 出力は、ORACLE の SQL*PLUS の START コマンドにより実行できる形式で作成する。SELEC 文は、キーワードの和 (or) の積 (and) の形式となる。
- 処理内容は、特別な制約なく記述された論理式を和積形式に変換することがメインとなる。これは、論理式の展開規則を用いて行う。

4.2 入力ファイルの形式

4.2.1 入力ファイルの例

```
(t002 rel)
(100 f001 (and 3.2.1 4.3 (or 3.1 (and 4.3 5.6)) (or 11.0 12.3)))
(90 f002 (or 3.2.1 (and 4.3 (or 3.1 (and 4.3 5.6)) 11.0)))
.
.
.
```

4.2.2 入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 レコード目以降
(評点 検索式 No. 論理式)
 評点 : 3 桁の数字 (max 100)

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。

4.3 出力ファイルの形式

4.3.1 出力ファイルの例

```
rem *****
rem          GENERATED SQL STATEMENTS : 8/2/1988
rem          test case No. : t002
rem          using relationships information
rem *****
```

```
create table result_rel
```

```
  (hyouten number(3),
   formula char(4),
   bunkenid char(11),
   title char(160));
```

```
insert into result (hyouten,formula,bunkenid,title)
  select 100 hyouten, 'f001' formula,bunkenid, title
  from vbibl
  where hno = '3.2.1'
intersect
select 100 hyouten, 'f001' formula,bunkenid, title
  from vbibl
  where hno = '4.3'
intersect
select 100 hyouten, 'f001' formula,bunkenid, title
  from vbibl
  where hno = '3.1'
  or hno = '5.6'
intersect
select 100 hyouten, 'f001' formula,bunkenid, title
  from vbibl
  where hno = '11.0'
  or hno = '12.3'
insert into result (hyouten,formula,bunkenid,title)
  select 90 hyouten, 'f002' formula,bunkenid, title
  from vbibl
  where hno = '3.2.1'
intersect
select 90 hyouten, 'f002' formula,bunkenid, title
  from vbibl
```

```

        where hno = '3.1'
        or    hno = '5.6'
intersect
select 90 hyouten, 'f002' formula,bunkenid, title
      from vbibl
      where hno = '11.0'
      or    hno = '12.3'

```

4.3.2 出力ファイルのフォーマット

- コメント部

```

rem *****
rem                               GENERATED SQL STATEMENTS : 生成日付
rem                               test case No. : テストケース No.
rem                               関係情報使用有無
rem *****

```

生成日付：プログラムの走行日付

テストケース No.：入力ファイルのテストケース No.

関係情報使用有無：関係情報使用有 (入力ファイルの関係情報使用有無 = 'rel')

⇒ 'using relationships information'

関係情報使用無 (入力ファイルの関係情報使用有無 = 'non_rel')

⇒ 'without using relationships information'

- CREATE 文

```

create table テーブル名
  (hyouten number(3),
   formula char(4),
   bunkenid char(11),
   title char(160));

```

テーブル名：関係情報使用有 (入力ファイルの関係情報使用有無 = 'rel')

⇒ 'result_rel'

関係情報使用無 (入力ファイルの関係情報使用有無 = 'non_rel')

⇒ 'resultr_nonrel'

- INSERT 文

```

insert into result (hyouten,formula,bunkenid,title)
  select 評点 hyouten, 検索式 No. formula,bunkenid, title

```

```

        from vbibl
        where hno = 'キーワード1-1'
        or    hno = 'キーワード1-2'
        or    .
        or    .
intersect
.
.
intersect
select 評点 hyouten, 検索式 No. formula,bunkenid, title
      from vbibl
      where hno = 'キーワードn-1'
      or    hno = 'キーワードn-2'
      or    .
      or    .

```

1つの INSERT 文は入力ファイル中の1つのレコード(2レコード目以降)に対応する。'intersect'で区切られた1つの select 文は、入力レコード中の論理式を変形して得られた和積形の1つの和の項に対応する。

例:

```

(A+B)*(C+D)
  select ~ from vbibl where hno = 'A' or hno = 'B'
  intersect
  select ~ from vbibl where hno = 'C' or hno = 'D'

```

評点：入力レコード中の評点

検索式 No.：入力レコード中の検索式 No.

4.4 論理式の変換について

4.4.1 論理式の標準形について

論理式の標準形には、積和標準形と和積標準形がある。ここでは、上記に述べたような方法で SQL 文に変換するため、特別な制約なしに記述されている論理式(プレフィックス形式)を和積標準形(プレフィックス形式)に変換する。

例:

```

A+B*C+D*E*F : (or A (and B C) (and D E F))
              ↓
(A+B+D)*(A+B+E)*(A+B+F)*(A+C+D)*(A+C+E)*(A+C+F) :
(and (or A B D) (or A B E) (or A B F) (or A C D) (or A C E) (or A C F))

```

4.4.2 和積標準形への変換規則

論理式は次の規則に基づいて和積標準形へ変換する。

- 規則-1 : $A * A = A$

- 規則-2 : $A+A=A$
- 規則-3 : $A+A*B=A$
- 規則-4 : $A+B*C=(A+B)*(A+C)$

変換の手順は、規則-4を用いて順次和積形に変換して行きながら、その途中で規則-1～3を用いて式の簡素化を行う。論理式が下記の形式になった時点で処理を終了する。

項-1

(and 項-1 項-2 . . . 項-n)

ここで、項-iはキーワード単独か、(or キーワード-1 キーワード-2 . . . キーワード-m)の形式である。

第5章

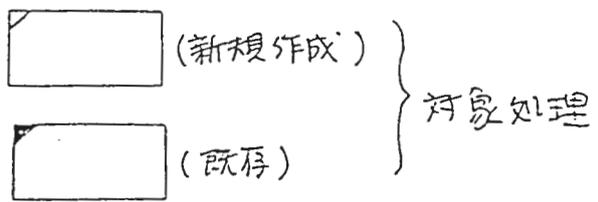
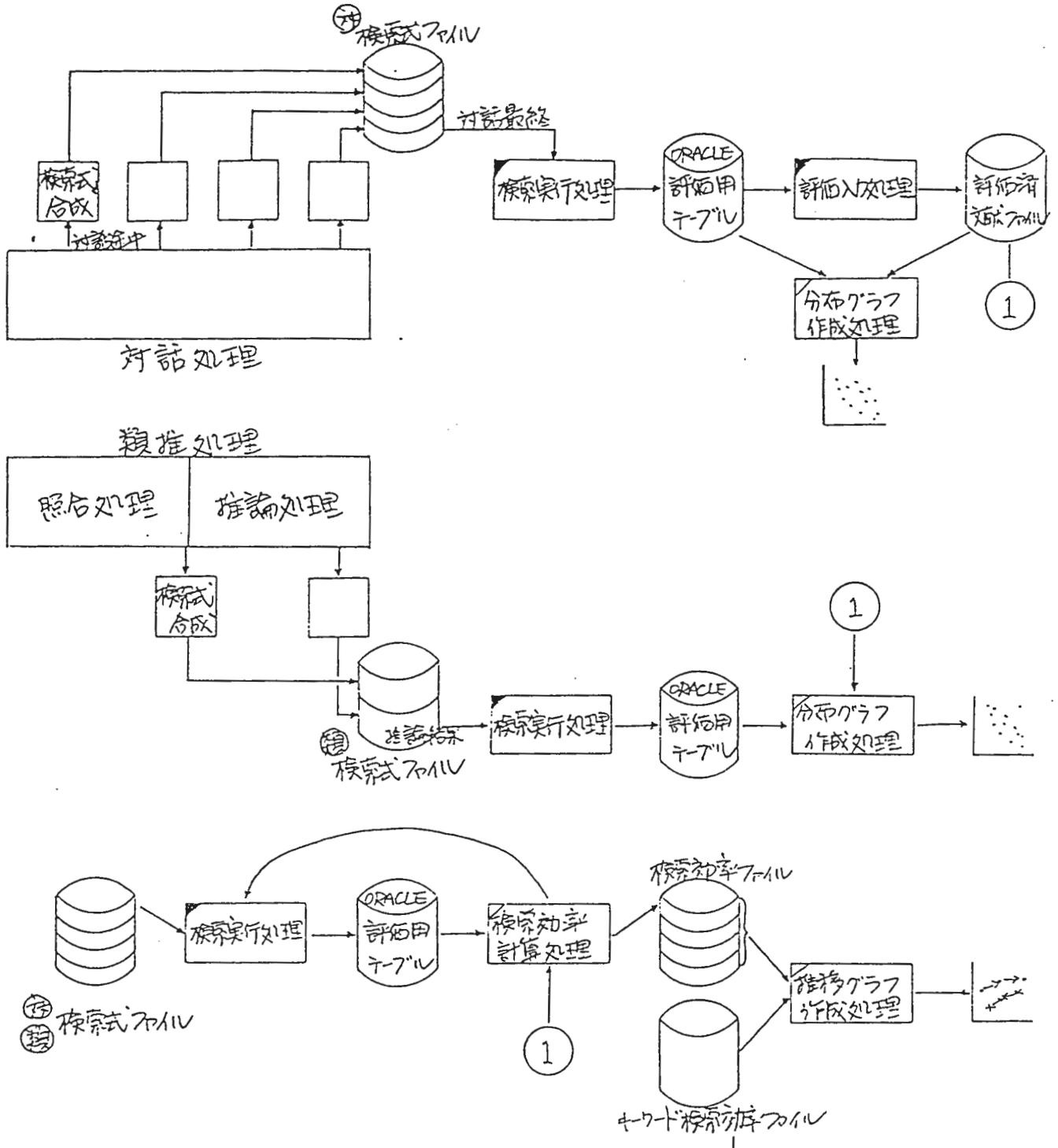
文献検索効率評価部

5.1 文献検索効率評価部の概要

文献検索効率評価部は、問合せ文から検索要求理解部（要求照合部、個人性獲得部、あるいは真意推論部）および検索式合成部を通して作成した検索式の実行の結果得られた文献を、利用者が付与した○×に従って評価するものである。次の2種類の評価を行う。

- 対話処理（個人性獲得部＋真意推論部）終了後、および類推処理（要求照合部＋真意推論部）終了後の意味表現（キーワード）を使用して合成したそれぞれの検索式によって得られた文献集合に対する適合率／再現率の分布。
- 対話処理の各段階、および類推処理の照合処理終了時と推論処理終了時における適合率／再現率の平均値とその推移。

5.2 処理の流れ



5.3 各処理の内容

5.3.1 検索実行処理

(a) 入力ファイルの形式

(t002 rel)

(100 f001 (and 3.2.1 4.3 (or 3.1 (and 4.3 5.6)) (or 11.0 12.3)))

(90 f002 (or 3.2.1 (and 4.3 (or 3.1 (and 4.3 5.6)) 11.0)))

.
.
.

5.3.2 入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 レコード目以降

(評点 検索式 No. 論理式)

評点 : 3 桁の数字 (max 100)

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。

(b) 出力ファイルの形式

出力は ORACLE のテーブル (評価用テーブル) の作成として行う。評価用テーブルのフォーマットを下に示す。評価用テーブルは、検索式 No. の検索式の実行の結果得られる総ての文献を持つ。従って、ある文献が複数の検索式の実行の結果得られることがある。

(評価用テーブルのフォーマット)

評点	検索式 No.	文献 No.	タイトル
----	---------	--------	------

(c) 処理内容

• SQL 文への変換

“SQL 文変換部の仕様” 参照。(向平氏作成図済、但し要変更)

• SQL の実行

SQLPLUS の STASRT コマンドにより、上記で作成した SQL 文を実行し、評価用テーブルを作成する。

5.3.3 評価入力処理

この処理は、キーワードに対する評価入力（○×の付与）とプログラムを共用する。

(a) 入力ファイルの形式

評価用テーブルを入力とする。

(b) 出力ファイルの形式

評価用テーブル中に含まれる文献（ダブリを排除したもの）に対して○×を付与したもの（評価済文献ファイル）。

(c) 処理内容

- 評価用テーブルのソーティングとダブリの排除
評価用テーブル中に含まれる文献を、評点でソーティングすると共に、文献No.の同一の文献については評点の最大のものを残して他は削除する。この処理は、SQL文によりORACLE(SQLPLI)にて行う。
- ○×の付与 上記の処理結果を利用者に提示し、○×を付与してもらう。この処理は、キーワードに対する○×の付与とプログラムを共用する。

5.3.4 分布グラフ作成処理

(a) 入力ファイルの形式

評価用テーブル（あるいは同一の形式のファイル）、および評価済文献ファイルを入力とする。

(b) 出力ファイルの形式

分布グラフを書くための、グラフ表示プログラム入力用データ。

(c) 処理内容

評価用テーブル（あるいは同一の形式のファイル）中の同一検索式No.の文献グループに対して、次の計算式で求めた適合率/再現率をグラフの一点のデータとする。

(適合率) =

$$\frac{(\text{○を付与された文献の数})}{(\text{グループ中の全文献の数})}$$

(再現率) =

$$\frac{(\text{○を付与された文献の数})}{(\text{評価済文献ファイル中の○を付与された全文献の数})}$$

5.3.5 検索効率計算処理

対話処理および類推処理の各段階で作成した検索式の実行結果として得られた評価用テーブルに対して処理を行う。

(a) 入力ファイルの形式

評価用テーブル（あるいは同一の形式のファイル）、および評価済文献ファイルを入力とする。

(b) 出力ファイルの形式

推移グラフの一点に対応する適合率／再現率データ。

(c) 処理内容

分布グラフ作成処理と同じ処理で得られる適合率／再現率の総ての点の平均値を求め、それを出力データとする。

5.3.6 推移グラフ作成処理

(a) 入力ファイルの形式

対話処理および類推処理の各段階での検索効率計算処理の出力データを累積した検索効率ファイル、およびキーワード検索効率の評価で求めたキーワード検索効率ファイル（対話処理および類推処理の各段階でのキーワード適合率／再現率）。

(b) 出力ファイルの形式

推移グラフを書くための、グラフ表示プログラム入力用データ。

(c) 処理内容

2つの入力ファイルから、1枚の推移グラフを書くための、グラフ表示プログラム入力用データを作成する。

第6章

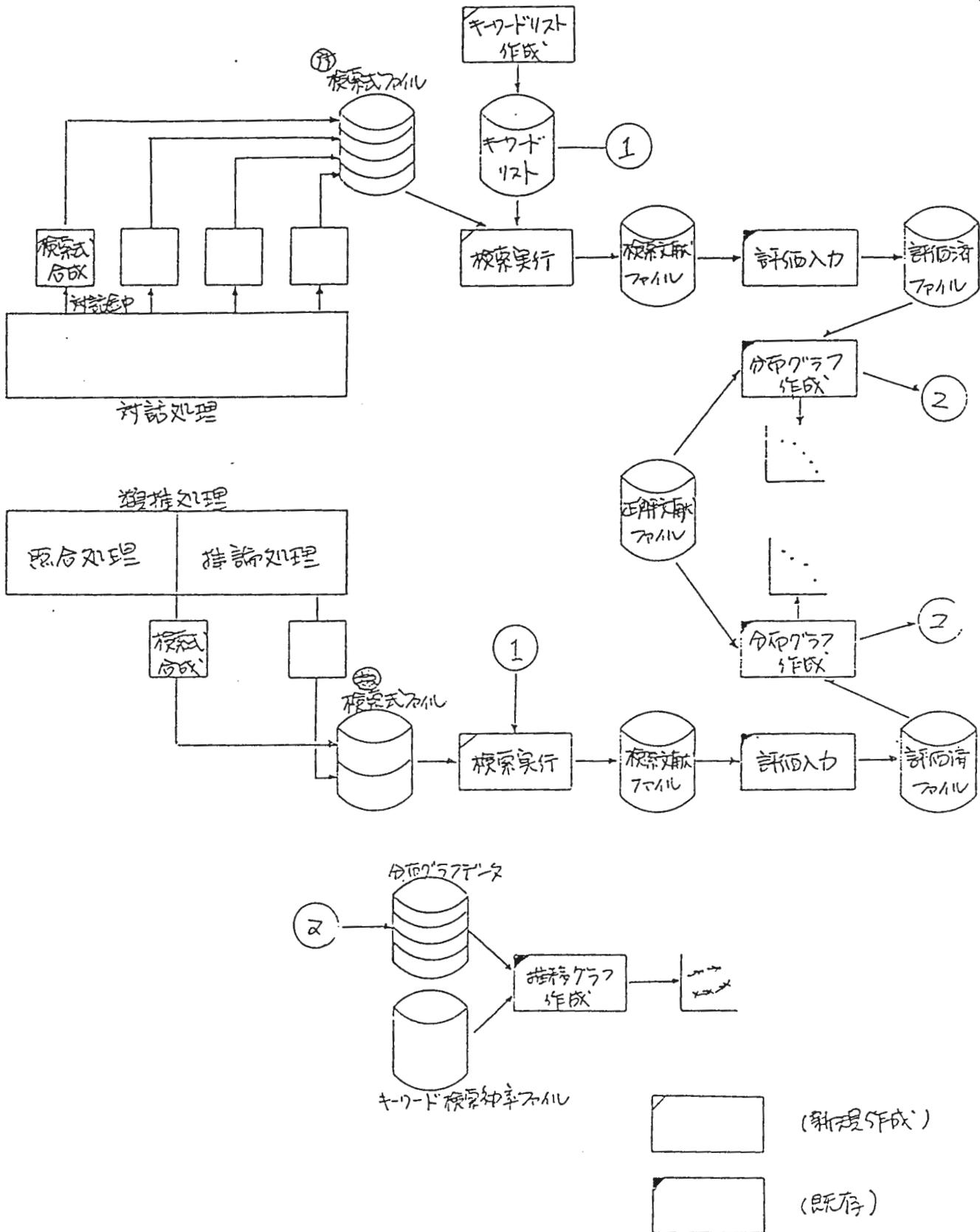
文献検索効率評価部2（キーワードマッチング）

6.1 文献検索効率評価部2の概要

文献検索効率評価部2は、問合せ文から検索要求理解部（要求照合部、個人性獲得部、あるいは真意推論部）および検索式合成部のうちフェーズ1の検索式合成用意味構造に含まれるキーワードを、各文献に付与されているキーワードとマッチングすることにより検索を行ない、利用者が付与した○×に従って評価するものである。この処理では検索式を作成せず、代わりに意味構造と文献のキーワードのマッチングの程度に応じた評点を与え、これを元に検索効率を評価する。次の2種類の評価を行う。

- 対話処理（個人性獲得部+真意推論部）終了後、および類推処理（要求照合部+真意推論部）終了後の意味表現中のキーワードを使用して検索した結果の適合率/再現率の分布。
- 対話処理の各段階、および類推処理の照合処理終了時と推論処理終了時における適合率/再現率の平均値とその推移。

6.2 処理の流れ



6.3 各処理の内容

6.3.1 キーワードリスト作成処理

(a) 処理の概要

ORACLE で持っている文献データベースから、以降の処理で利用しやすい Lisp 形式の文献毎のキーワードのリストに変換する。

(b) 入力データベースの形式

文献テーブル : テーブル名 = bibltab

文献 No. bunkenid char(11)	タイトル title char (160)
-----------------------------	--------------------------

キーワードテーブル : テーブル名 = newindex

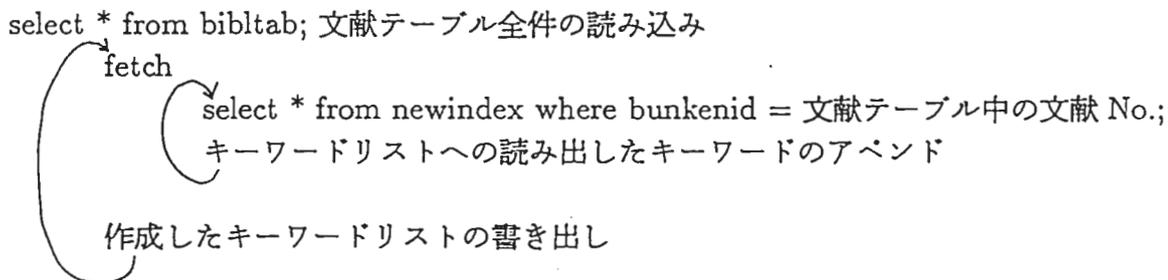
文献 No. bunkenid char(11)	キーワード hno char (16)
-----------------------------	------------------------

(c) 出力ファイルの形式

Lisp の S 式の形式のキーワードのリスト
(文献 No., キーワード 1, キーワード 2, ..., キーワード n)

(d) 処理内容

次のような流れで処理を行なう



6.3.2 検索実行処理

(a) 処理の概要

検索式合成用意味構造に含まれるキーワードをキーワードリスト作成処理で作成した文献のキーワードリストとマッチングすることにより、各文献に評点を与える。

(b) 入力ファイルの形式

(t002 rel)

(100 f001 (and 3.2.1 4.3 (or 3.1 (and 4.3 5.6)) (or 11.0 12.3)))

(90 f002 (or 3.2.1 (and 4.3 (or 3.1 (and 4.3 5.6)) 11.0)))

6.3.3 入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レコード目以降

(評点 検索式 No. 論理式)

評点 : 3桁の数字 (max 100)

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。

(c) 出力ファイルの形式

出力ファイルのフォーマットを下に示す。出力ファイルは、評点が0でない全ての文献を持つ。

評点	検索式 No.	文献 No.	タイトル
----	---------	--------	------

(d) 処理内容

● 検索用キーワードの抽出

入力ファイル中のレコードのうち、評点が100のもの論理式中のキーワードを洩れとダブリがないように抽出する。

● 文献の評点の計算と、出力ファイルへの書き出し

各文献のキーワードリストと検索用キーワードを比較し評点を計算する。評点は、各検索用キーワードが文献のキーワードリストに含まれていれば1、含まれていなければ0とし、それぞれの検索用キーワードの評点の和を文献の評点とする。文献の評点が0でなければ、出力ファイルへ書き出す。

6.3.4 評価入力処理

(a) 処理の概要

作成済みの評価入力プログラムを使用し、検索実行処理で作成した文献ファイルに対して評価入力(○×の付与)する。

(b) 入力ファイルの形式

検索実行処理での出力ファイル。

(c) 出力ファイルの形式

入力ファイルに対して○×を付与したもの（評価済文献ファイル）。

(d) 処理内容

- 入力ファイルのソーティング 入力ファイルを、評点順にソーティングする。
- ○×の付与 上記の処理結果を被験者毎に用意している正解文献ファイルと照合し、○×を付与する。この処理は、作成済みのプログラムを使用する。

6.3.5 分布グラフ作成処理

(a) 処理の概要

同一評点グループの文献を単位として検索効率を計算し、グラフを作成する。

(b) 入力ファイルの形式

被験者毎に用意している正解文献ファイル、および評価済文献ファイルを入力とする。

(c) 出力ファイルの形式

分布グラフを書くための、グラフ表示プログラム入力用データ。

(d) 処理内容

ある評点より大きな評点を持つ文献を一つのグループとみなす。同一グループの文献に対して、次の計算式で求めた適合率／再現率をグラフの一点のデータとする。

(適合率) =

$$\frac{(\text{○を付与された文献の数})}{(\text{グループ中の全文献の数})}$$

(再現率) =

$$\frac{(\text{○を付与された文献の数})}{(\text{評価済文献ファイル中の○を付与された全文献の数})}$$

	評点	文献 no.
グループ 1	10	a
	10	b
グループ 2	9	c
	9	d
	9	e
グループ 3	8	f
	7	g
グループ 4	7	h
	6	i
	:	

6.3.6 推移グラフ作成処理

(a) 処理の概要

対話処理および類推処理の各段階で作成した検索処理の実行結果として得られた適合率/再現率の平均を求め、これをひとつの対話処理あるいは類推処理の単位にグラフ表示する。

(b) 入力ファイルの形式

対話処理および類推処理の各段階での検索効率計算処理の出力データを累積した検索効率ファイル、およびキーワード検索効率の評価で求めたキーワード検索効率ファイル(対話処理および類推処理の各段階でのキーワード適合率/再現率)。

(c) 出力ファイルの形式

推移グラフを書くための、グラフ表示プログラム入力用データ。

(d) 処理内容

入力ファイル中の一つの対話あるいは類推処理の各段階(照合処理の終了、推論処理の終了)毎に適合率/再現率の平均値を求め、1枚の推移グラフを書くための、グラフ表示プログラム入力用データを作成する。

6.4 検索式合成プログラムの処理時間について

ここでは、文献検索効率評価部2(キーワードマッチング)が存在する理由について記述する。

6.4.1 検索式合成プログラムのアルゴリズムの概要

検索式合成プログラムは、意味構造から検索式を作成するものである。対話による理解、あるいは類推による理解で得られた意味構造から直接作成した検索式(これを、最適検索式と呼ぶ)から、条件を緩めていくことによって、複数の検索式を合成する。この条件の緩め方に、次の二つの手法を採用し、それぞれの検索結果を比較する。

- 意味情報を使用するもの
最適検索式を作成した後、意味構造のネットワークで最も話題の中心から遠いものから順にキーワードを削除した意味構造を作成して行き、それぞれの意味構造に対して、最適検索式の作成と同じ規則で検索式を作成する。
- 意味情報を使用しないもの
最適検索式を作成した後、検索式の中の and で結合された項から、順次キーワードを外していくことにより、検索式を作成する。(別紙2)

6.4.2 意味情報を使用しないアルゴリズムの処理が長時間となる理由

(a) 時間計算量

簡単のために、最適検索式を n 個のキーワードを and で結合したものとする。この場合、作成するすべての検索式の数(キーワードを and で結合したもの)は、 $\sum nC_i$ となる。この検索式をすべて1回

の処理で作成できるとすれば、時間計算量のオーダーもこれと同じになる。従って、時間計算量 $> O(n^2)$ となる。

(a) 空間計算量

現在のアルゴリズムでは、処理途中の検索式をすべてリスト形式でプログラム中に持っている。このため、空間計算量も時間計算量と同じ ($\sum nCi > O(n^2)$) となる。現実的には、大きなリスト (10,000 エントリ程度) をプログラムで参照する場合には、リストをスキャンする時のディスク I/O のオーバーヘッドが大きくなり、CPU がほとんど使われないような状態になる。これが処理が長時間かかる最大の原因と思われる。

6.4.3 対策

(a) その1

検索式を作成せずに、約 800 の文献のすべてを作成した意味構造と比較し、それぞれに評点を与える。これを評点順にソーティングし、検索効率を求める。評点順にソーティングされた文献リストから検索効率を求める方法は、すでに存在する。

この方法は、評価実験用にのみ適用可能なものであり、実際の文献データベースのフロントエンドには適用はできない。実際のフロントエンドを作成する際の、適切な検索式を合成するルールを抽出するための方法として位置付けることができる。

この方法では、抽出したキーワード数に関係なく、文献数 n のオーダー $O(n)$ の時間計算量となり、また空間計算量は文献数 n に関係なくコンスタントとなる。

(b) その2

上記の対策その1より得られたルールを元に、検索式合成のプログラム (今回のような無条件に検索式を作成しないもの) を作成する。これは、今回の意味情報を使用しない検索式合成の手法が Breadth First 型の全解探索的な動きをするのに対して、ヒューリスティックスを導入して探索空間を有限時間ないで処理が可能な範囲に限定するものと考えられる。このため、キーワード数 n に対して今回のようにオーダー $O(n^2)$ 以上ではないにしろ、オーダー $O(n)$ 以上の計算量となることが予想される。この検索式合成のプログラムを実行するには、高速の lisp 処理系が望ましい。

6.4.4 文献検索効率評価部2 (キーワードマッチング) が存在する理由

前述の対策その1を実現するものとして、文献検索効率評価部2 (キーワードマッチング) を作成している。従って、文献検索効率評価部2 (キーワードマッチング) はより良い実現法が存在すればそれと置き換えるのが望ましい。

第7章

検索結果からの問合せ理解部

7.1 検索結果からの問合せ理解部の概要

利用者の問合せ文を理解し、それに基づいて文献データベースを検索するために、システム全体として次のような3つのフェーズで問合せ文を処理するものとする（図1）。

(フェーズ1) 問合せ文のユーザモデルへの照応

問合せ文をユーザモデルに照応し、意味構造を作成する。この過程で、上記の検索意図のうち、主に(a)と(b)を抽出する。

(フェーズ2) 文献データベースの検索

意味構造を、キーワードによる文献データベースの検索式へ変換し、検索を実行する。

(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い、利用者が選択した文献から検索意図の抽出を行ない、意味構造を変更する。

これらのうち、フェーズ1とフェーズ2については作成済みであり、今回はフェーズ3の部分を作成する。

検索結果からの問合せ理解部では、次の二つの機能を実現する。

- 検索結果からの検索意図の抽出

システムが検索した文献のうち、利用者が選択したものに含まれるキーワードを解析し、利用者の検索意図のうち今まで抽出できていないものを抽出する。これは、次のようなキーワードを抽出することにより行なう。

- － 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
- － 利用者が選択した文献に、全文献中の出現頻度比べて高い頻度で出現するキーワード

この処理は、抽出可能なキーワードがなくなるまで繰り返して行なう（フィードバック処理）。

- 抽出した検索意図からのユーザモデルの構築

フィードバック処理終了後の利用者が選択した文献集合（即ち、最終的に利用者の検索意図に適合すると判断された文献）から、上述の処理で抽出されたキーワードと他のキーワードとの関連を抽出し、それに基づいてユーザモデルのノード間のリンクを修正することによってユーザモデルの構築を行なう。キーワード間の関連の有無は、二つのキーワードの共起関係の強さによって判断する。関連の種類、および修正の方法は、次の通りである。

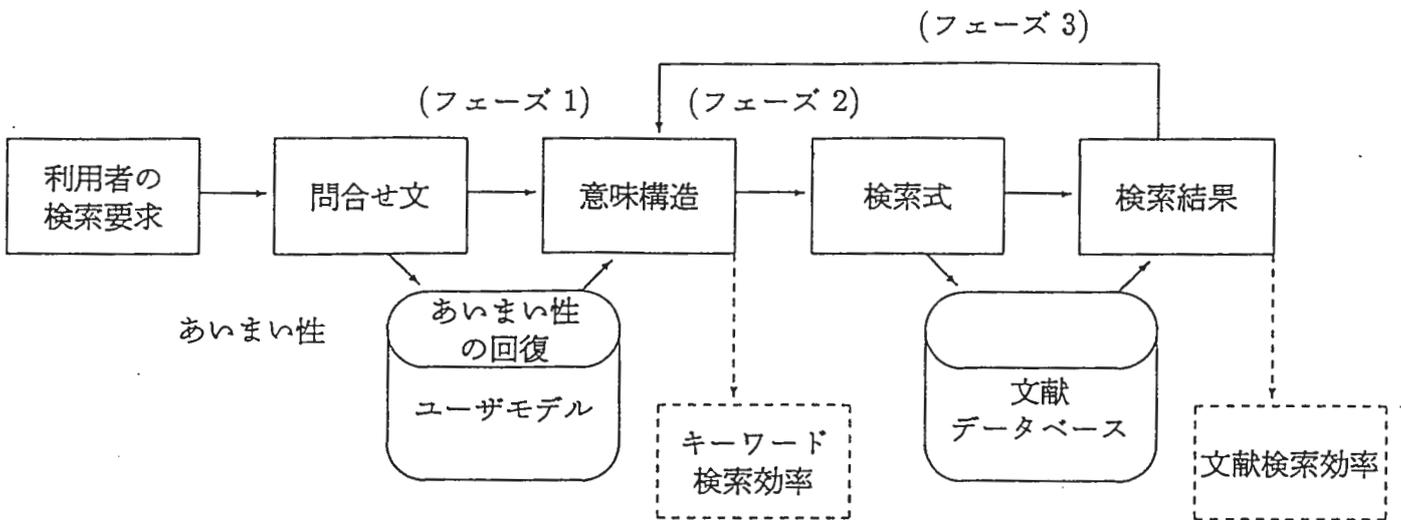


図 7.1: 文献検索システムのモデル

- ユーザモデル上二つのキーワードノード間ですでに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。
- ユーザモデル上二つのキーワードノード間に関連が存在しないならば、それらのノード間に関連関係のリンクを追加する。

7.2 フィードバック処理部

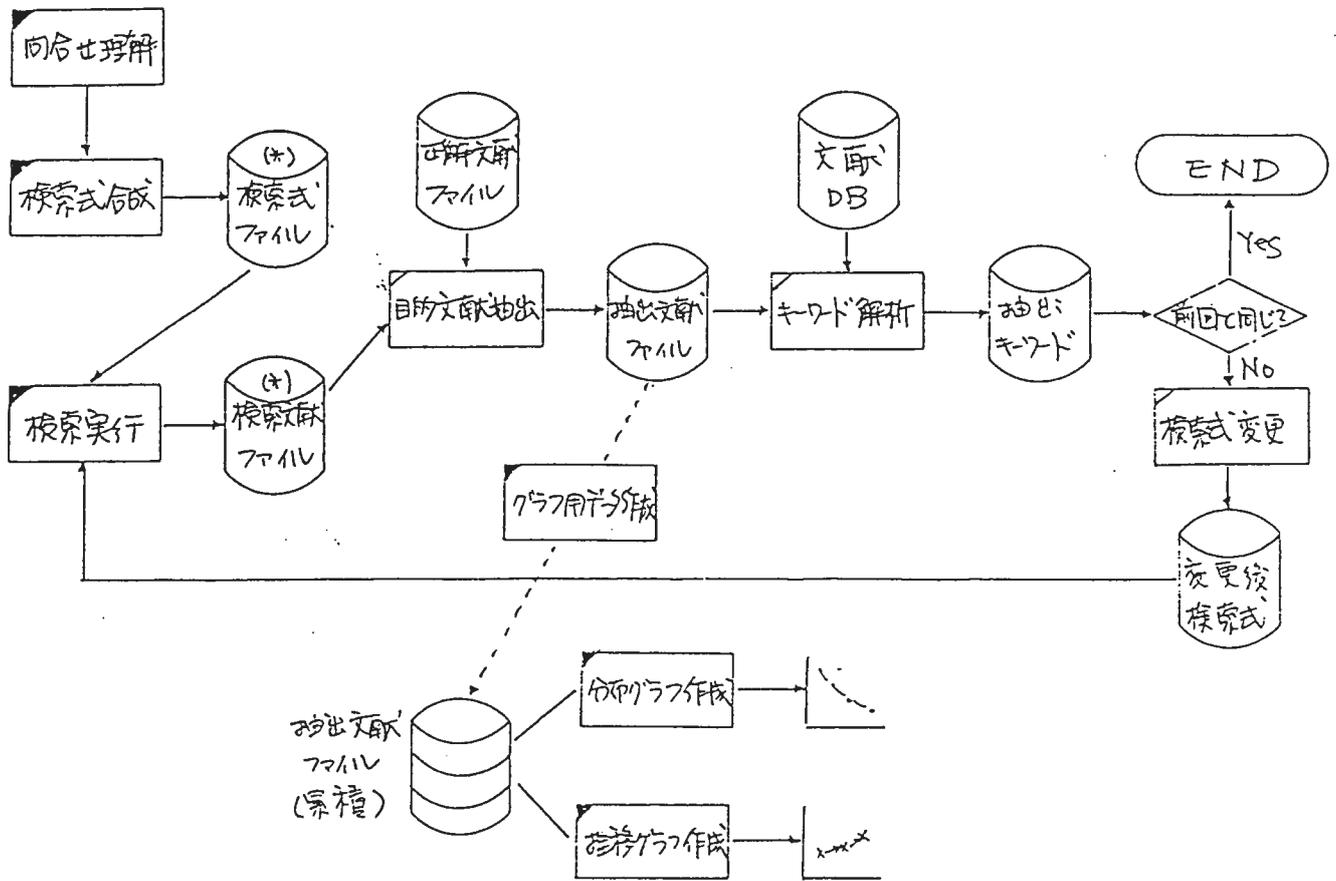
7.2.1 処理の概要

次のような処理により、利用者の検索意図に相当するキーワードを抽出し、それを今までのキーワードに付加して更に検索を行なう。

- 利用者の検索意図に合う文献の抽出（目的文献抽出処理）
あらかじめ用意してある利用者の目的の文献と検索結果の文献を照合し、両方に含まれている利用者の検索意図に合う文献を抽出する。
- 利用者の検索意図に合う文献のキーワードの解析（キーワード解析処理）
利用者の検索意図に合う文献に付与されているキーワードを文献データベースから検索する。このようにして得られたキーワードのうち、次のものを抽出する。
 - 利用者の検索意図に合う文献に、高い頻度で出現するキーワード
 - 利用者の検索意図に合う文献に、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
- 文献の検索（検索式変更処理）
以前に抽出していたキーワードと、上記の処理により抽出したキーワードを合わせて検索実行部（キーワードマッチング）に入力するためのデータを作成し、検索実行する。

この処理は、抽出可能なキーワードがなくなるまで繰り返して行なう（フィードバック処理）。

7.2.2 処理の流れ



7.2.3 入力の形式

(a) 検索式ファイル

検索式合成処理により作成された、複数の検索式をS式で表現したデータ

<例>

(t002 rel)

(100 f001 (and 3.2.1 4.3 (or 3.1 (and 4.3 5.6)) (or 11.0 12.3))))

(90 f002 (or 3.2.1 (and 4.3 (or 3.1 (and 4.3 5.6)) 11.0))))

⋮

<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レコード目以降

(評点 検索式 No. 論理式)

評点 : 3桁の数字 (max 100)

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。

(b) 検索文献ファイル

検索実行処理により作成された、検索結果の文献を表すデータ。データは、評点の昇順にならんでいる。

<例>

5 F001 BUNKEN-1052
4 F001 BUNKEN-2241
4 F001 BUNKEN-1003
3 F001 BUNKEN-2158

⋮
⋮
⋮

<フォーマット>

評点	検索式 No.	文献 No.
----	---------	--------

(c) 正解文献ファイル

被験者が、全文献から自分の検索意図に合うものを抽出するために○を付与したデータ。データは、文献No.の昇順にならんでいる。

<例>

0 0 BUNKEN-0241 故障診断エキスパートシステム SHOOTX における知識表現
0 0 BUNKEN-0242 知識ベースシステムにおける学習 可能性のあるアプローチ
1 0 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
0 0 BUNKEN-0244 組立てセルを対象とした知識の利用と表現
0 0 BUNKEN-0245 AI はどのようにしてオフィスオートメーションに頭脳を付加するか

⋮
⋮
⋮

<フォーマット>

評価	dummy	文献 No.	タイトル
----	-------	--------	------

(d) 文献データベース

文献を表現している ORACLE のデータベースで、次の二つのテーブルからなる。

文献テーブル : テーブル名 = bibltab

文献 No. bunkenid char(11)	タイトル title char (160)
-----------------------------	--------------------------

キーワードテーブル : テーブル名 = newindex

文献 No. bunkenid char(11)	キーワード hno char (16)
-----------------------------	------------------------

7.2.4 出力ファイルの形式

(a) 抽出文献ファイル

データの形式は、検索文献ファイルと同じ。検索文献ファイルの中から、目的文献抽出処理により抽出したものを格納する。

(b) 抽出キーワードファイル

抽出文献ファイルに含まれる文献に付与されているキーワードのうち、前回の処理までで抽出できなかったものを格納する。

(c) 変更後検索式ファイル

検索式ファイル（あるいは、前回の処理で作成した変更後検索式ファイル）を元に、検索式変更処理により作成した検索式の情報を格納する。データの形式は、検索式ファイルと同じ。

7.2.5 各部の処理内容

(a) 目的文献抽出処理部

step 1 検索文献ファイルの中から、下記の基準で文献を選択する。

- 評点の高いものから 20 文献を選択する。
- 上記の 20 番目と同じ評点を持つものは、すべて選択する。
- 以上の基準で選ばれたものの中から、評点が 1 のものを除く。

step 2 step 1 で選択した文献を正解文献ファイルと照合し、正解文献ファイル中で評価が 1 のものを抽出文献ファイルに出力する。

(b) キーワード解析処理部

キーワード解析処理は、抽出文献ファイル中の文献に付与されているキーワードの頻度解析を行い、頻度の高いキーワードを抽出キーワードファイルに書き出すものである。次の手順で処理する。

step 1 次のような一連の文献データベース検索用 SQL ステートメントを合成する。

```
drop table temp1;
drop table temp2;
```

```
create table temp1
  (bunkenid char(11),
   hno char(16));
```

```
create table temp2
  (count number(3),
   hno char(16));
```

```
insert into temp1 (bunkenid,hno)
select bunkenid,hno
  from vbibl
  where bunkenid in ('BUNKEN-1052', 'BUNKEN-1105', 'BUNKEN-1095', 'BUNKEN-1051');
```

```
insert into temp2 (count,hno)
select count(*),hno from temp1
  group by hno;
```

```
select X.count,Y.count,X.hno,j_alias
  from temp2 X,keycount Y,indexterm Z
  where X.hno = Y.hno
  and Y.hno = Z.hno
  order by X.count desc,Y.count;
```

'BUNKEN-xxxx' は、抽出文献ファイル中の文献の文献 No. である。

step 2 step 1 で合成したステートメントを実行する。この結果、下のような表が得られる。

COUNT	T-COUNT	HNO	J_ALIAS
4	15	7.10	自然言語インターフェース
3	12	7	自然言語処理
3	129	4.4.1	知識ベース
2	18	1.19.3	情報検索
2	44	1.20	データベース
2	48	1.5	マンマシンインターフェース
1	6	1.0	地図

step 3 step 2 で得られた表から、次の基準を満たすキーワード (HNO) を抽出し、抽出キーワードファイルに書き出す。

- 出現頻度 (COUNT) が、抽出文献ファイル中の文献数の 1/3 を越えるもの。
- $COUNT/T_COUNT$ が、(抽出文献ファイル中の文献数) /80 を越えるもの。
- 前回までの処理で、抽出できていなかったもの。

(c) 検索式変更処理部

キーワード解析処理部で作成した抽出キーワードファイルから、次のような検索式を合成し、変更後検索式ファイルに出力する。

(100 F001 (OR 1.5 A.18.9 A.18 7.10 1.20 1.19.3 6 7 1.15))

評点 : 100 (コンスタント)

検索式 No. : F001 (コンスタント)

論理式 : 抽出キーワードファイルに含まれるすべてのキーワード (HNO) を OR で結合したもの。

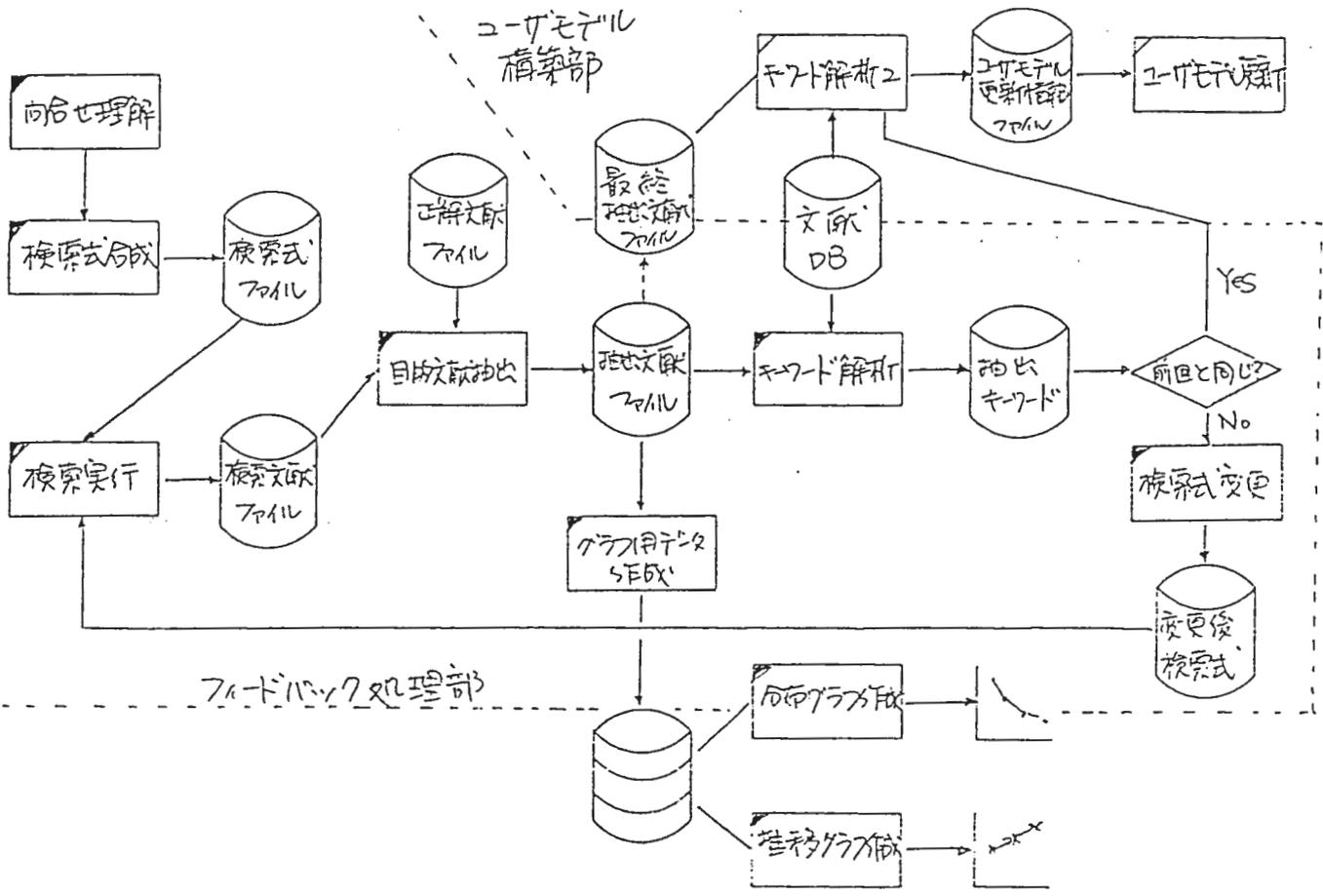
7.3 ユーザモデル構築部

7.3.1 処理の概要

次のような処理により、検索結果から利用者の検索意図に相当するキーワードを抽出し、それを元にユーザモデルの構築を行なう。

- 最終の利用者の検索意図に合う文献のキーワードの解析 (キーワード解析処理 2)
フィードバック処理で最終的に選択された文献に付与されているキーワードを解析する。これは、文献データベースからキーワードに関する次のような情報を検索することである。
 - 最終の利用者の検索意図に合う文献に高い頻度で出現するキーワード、および、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 (フィードバック処理部と同じ)
 - 二つのキーワード間で共起関係の強い (二つのキーワードが同時に一つの文献に付与される頻度が高い) キーワードの組
- ユーザモデルの更新 (ユーザモデル更新処理)
キーワード解析処理 2 で求めたキーワードを、キーワード解析処理 2 で求めた共起関係に基づきユーザモデルを更新する。即ち、
 - ユーザモデル上で二つのキーワードノード間にすでに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。
 - ユーザモデル上で二つのキーワードノード間に関連が存在しないならば、それらのノード間に関連関係のリンクを追加する。

7.3.2 処理の流れ



7.3.3 入力の形式

(a0 最終抽出文献ファイル)

フィードバック処理部の検索実行処理の最終の処理で作成された、検索結果の文献を表すデータ。即ち、抽出文献ファイルの最終回のものである。データは、評点の昇順にならんでいる。

- <例>
- 5 F001 BUNKEN-1052
 - 4 F001 BUNKEN-2241
 - 4 F001 BUNKEN-1003
 - 3 F001 BUNKEN-2158
 - ...
 - ...
 - ...

<フォーマット>

評点	検索式 No.	文献 No.
----	---------	--------

(b) 初期検索式ファイル

フィードバック処理開始前に検索式合成処理により作成された、複数の検索式を S 式で表現したデータ

<例>

(t002 rel)

(100 f001 (and 3.2.1 4.3 (or 3.1 (and 4.3 5.6)) (or 11.0 12.3)))

(90 f002 (or 3.2.1 (and 4.3 (or 3.1 (and 4.3 5.6)) 11.0)))

.

.

.

<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 レコード目以降

(評点 検索式 No. 論理式)

評点 : 3 桁の数字 (max 100)

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。

7.3.4 出力ファイルの形式

(a) ユーザモデル更新情報ファイル

最終抽出文献ファイルを解析して得られたユーザモデル更新情報、即ち、初期検索式ファイルには含まれていないキーワード (キーワードの頻度分析の結果)、及びそのキーワードをユーザモデルに登録するための情報 (キーワードの共起関係の分析の結果) を格納する。

ファイル形式等の詳細は特に規定しない。

7.3.5 各部の処理内容

(a) キーワード解析処理部

キーワード解析処理は、最終抽出文献ファイル中の文献に付与されているキーワードの頻度分析、及び共起関係の分析を行ない、その結果をユーザモデル更新情報ファイルに書き出すものである。次の手順で処理する。

step 1 次のような一連の文献データベース検索用 SQL ステートメントを合成する。

```
drop table temp1;
```

```
drop table temp2;
```

```
create table temp1
  (bunkenid char(11),
   hno char(16));

create table temp2
  (count number(3),
   hno char(16));

insert into temp1 (bunkenid,hno)
select bunkenid,hno
  from vbibl
  where bunkenid in ('BUNKEN-1052', 'BUNKEN-1105', 'BUNKEN-1095', 'BUNKEN-
1051');

insert into temp2 (count,hno)
select count(*),hno from temp1
  group by hno;

select X.count,Y.count,X.hno,j_alias
  from temp2 X,keycount Y,indexterm Z
  where X.hno = Y.hno
  and Y.hno = Z.hno
  order by X.count desc,Y.count;
(頻度分析用)

select X.hno,Y.hno,count(*)
  from temp1 X,temp1 Y
  where X.bunkenid = Y.bunkenid
  and X.hno != Y.hno
  group by X.hno,Y.hno
  order by count(*) desc;
(共起関係分析用)
```

'BUNKEN-xxxx' は、抽出文献ファイル中の文献の文献 No. である。

step 2 step 1 で合成したステートメントを実行する。この結果、下のような表が得られる。
(頻度分析用)

COUNT	T-COUNT	HNO	J_ALIAS
4	15	7.10	自然言語インターフェース
3	12	7	自然言語処理
3	129	4.4.1	知識ベース
2	18	1.19.3	情報検索
2	44	1.20	データベース
2	48	1.5	マンマシンインターフェース
1	6	1.0	地図

(共起関係分析用)

HNO	HNO	COUNT(*)
4.4.1	7.10	3
7	7.10	3
1.19.3	1.20	2
1.19.3	7.10	2
1.5	7	2
4.4.1	1.5	2
1.20	7.10	2
1.5	7.10	2

この表では、(キーワード1、キーワード2)と(キーワード2、キーワード1)という組が必ず存在する。

step 3 step 2 で得られた表(頻度分析用)から、次の基準を満たすキーワード(HNO)を抽出し、ユーザモデル更新情報ファイルに書き出す。

- 出現頻度(COUNT)が、抽出文献ファイル中に文献数の1/3を越えるもの。
- COUNT/T-COUNTが、(抽出文献ファイル中の文献数)/80を越えるもの。
- 前回までの処理で、抽出できていなかったもの

step 4 step 2 で得られた表(共起関係分析用)から、次の基準を満たすキーワード(HNO)の組を抽出し、ユーザモデル更新情報ファイルに書き出す。

- step 3で頻度分析用の表から抽出されたキーワード(HNO)と、共起頻度(COUNT(*))が最大のキーワード(HNO)との組(この基準を満たすものが複数あれば、それらすべて)
- 上記の基準で抽出したキーワードの組がすべて頻度分析用の表から抽出されたキーワード同士の組であれば、それ以外の組で共起頻度(COUNT(*))が最大のもの(頻度分析用の表から抽出されたキーワード同士の組は除く)すべて
- step 3で抽出されたキーワード(HNO)すべてに対して、必ず共起関係にある他のキーワードが存在する

(b) ユーザモデル更新処理部

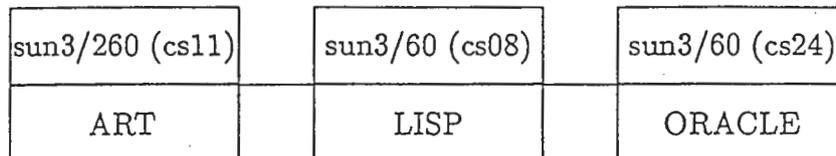
キーワード解析処理2でユーザモデル更新情報ファイルに書き出された情報を元にユーザモデルを更新する。この処理は、次のように行なう。

- ユーザモデル更新情報ファイルに書き出された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にすでに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。すでに関連の強さが“意図する”であれば、何もしない。
- ユーザモデル更新情報ファイルに書き出された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に関連が存在しないならば、それらのノード間に関連関係のリンク（関連の強さは“意図する”）を追加する。関係は、双方向とする。

付録 A

文献検索効率評価実験の実行環境について

文献検索効率評価実験は、下図のようなマシン構成で行なっている。



処理の手順は、次の通りである。

cs11 対話による理解処理：問合せ文 ⇒ 意味表現（キーワード抽出）

類推による理解処理：問合せ文 ⇒ 意味表現（キーワード抽出）

cs08 意味表現（キーワード）から、検索式への変換（SQL文）

cs24 検索式の実行

cs11 文献の適合率／再現率の計算、およびグラフ表示

このうち、cs08で実行している意味表現（キーワード）から検索式への変換処理が、組合せの爆発が生じるものがあり、時間が掛かっている。この処理は、開始から終了までの間にとくにオペレータとの対話が必要ないため、これを複数のマシンで実行すれば、処理の効率化が計れる。そこで、現在使用されていないマシン（cs32、cs35）を意味表現（キーワード）から検索式への変換処理のために使用したい。

1件当たりの処理時間。（1人の被験者当たり3つの問合せ、被験者は4人）

- 類推による理解処理の1人の被験者の1つの問合せ当たりの処理時間：
1日強
- 対話による理解処理の1人の被験者の1つの問合せ当たりの処理時間：
類推による理解処理の3倍程度の処理時間が掛かると思われる

これらの処理は、同時に並行して複数のものを走行させることができる。

付録 B

文献検索効率評価実験のお願い

B.1 概要

2月16日のフリーディスカッションでお願いしたように、下記の要領で文献検索効率評価実験を実施しますので、協力方宜しくお願いいたします。

2 実施要領

前回(88年12月)の問合せ理解実験の時に作成して頂いた問合せ文(3種類)の目的とする文献を、約800の文献リストの中から選択して下さい。

B.1.1 配布物

以下のリスト(3組)を配布しますので、問合せ文の目的とする文献に相当する文献ナンバーの前に○をつけてください。

- 文献リスト(3組)
- 前回(88年12月)の問合せ文と、対話処理のログリスト(3種類)

B.1.2 注意事項

文献リストには日本語の文献タイトルのみしか表示していません。これだけの情報では不十分な場合には、JICST文献抄録(木下保管)を参照して下さい。特に、次のようなものも有りますので、日本語の文献タイトルだけを頼りにせずにJICST文献抄録を参照するようにして下さい。

- 日本語の文献タイトル
BUNKEN-2141”さえずりー静止飛行 異常な鳥, 適用可能規則とデフォルトブルーバについてのいくつかの注意”
- 英文タイトル
Tweety - still flying. Some remarks on abnormal birds, applicable rules and a default prover.

B.2 締切

3月17日