

自然言語からの SQL 生成における非明示条件の補完

7F-2

芝 温子 宮部 隆夫 市山 俊治

NEC 関西 C&C 研究所

1 はじめに

自然言語を解釈し、データベース検索用の SQL を生成する自然言語インタフェースでは、SQL 生成において入力文で明示されない情報の補完が必要な場合がある。それらは、1) 入力文の不完全性に起因する指示語や省略表現の補完 [1] と、2) データベース構造に起因する検索の対象や条件の補完に大別できる。

筆者らが開発を進めている自然言語インタフェース構築キット 'IF-Kit' [2] においては、検索式生成の高速化のため副問い合わせ形式をできるだけ避けて SQL を生成するので、特に 2) について厳密な処理が必要となる。

本稿では自然言語からの SQL 生成における、データベース構造に起因した補完問題として、a) テーブル結合条件の補完、b) 検索の対象・グループ化項目の補完について述べ、その解決手法を提案する。

2 IF-Kit における SQL 生成の概略

IF-Kit では、自然言語入力文から対象システムによらない概念依存構造を生成し、拡張意味ネットワークの連想探索を利用したデータベースシステム上の概念空間への変換を行う [3]。この出力を意味表現と呼び、検索の対象、条件、表示方法からなる。図 1 のデータベースに対する入力とその意味表現の一例を図 2 に示す。

テーブル名	フィールド名	キー	キー依存	結合キー
会社	会社番号	○		営業.会社番号
	名前		○	
	業界カテゴリー			
営業	会社番号	○		会社.会社番号
	年度	○		
	営業利益			

キー依存: 単一キーのテーブルにおいて、キーと同様にその値から一意にレコードが決まるフィールド

図 1: データベーススキーマの一例

意味表現の条件 (A) 内のフィールドはさらに条件をとることができ、それを条件 (A) に対してサブ条件と呼ぶ。サブ条件があると、データベース上は同一でも検索式中では別テーブルとして扱うべきものが存在する。そのため SQL 生成部で、テーブルに (テーブル名 + 番号) を別名として付け、データベースや検索言語に関する知識を用いて、この意味表現から検索言語の文法に即した SQL

A Method of Complementing Implicit Conditions in SQL Generation from Natural Language Query.
Haruko SHIBA, Takao MIYABE, Syunji ICHIYAMA
Kansai C&C Research Lab., NEC Corp.

入力文: NEC よりも多くの営業利益をあげた会社は?

意味表現: { 対象 [field(会社1.名前)],

```

条件 [gt(arg1 field(営業2.営業利益),
      arg2 field(営業3.営業利益);
      サブ条件 [eq(arg1 field(会社4.名前),
                  arg2 value(NEC))]]]] }
    
```

図 2: 意味表現の一例

形式に変換する。

3 テーブル結合条件の補足

自然言語で明示されないテーブル間の結合条件は、意味表現で与えないため補足する必要がある。以下、図 1 のデータベースにおいて説明する。

3.1 条件の修飾関係からの結合条件補足

意味表現では概念上の条件の係受け情報を保持しているので、条件内のテーブル (条件テーブル) と、その修飾先のテーブル (係先テーブル) との間に、データベースに登録してある結合キーをすべて用いて結合条件を設定する。フィールドと値の比較の場合、フィールドのテーブルを条件テーブルとする。フィールド同士の比較の場合、条件内の 2 つのテーブルの一方は比較基準を示すもの (基準テーブル) なので、基準テーブルと係先テーブルとは結合しない。図 3(1) は、対象のフィールドを頂点とし、その修飾条件を下に付けたツリー構造で意味表現を表したものである。図 3(1) において条件テーブルとして、

1. サブ条件 1 (またはサブ条件 2) しか存在しない場合 → テーブル 2 (またはテーブル 1)
2. 比較条件が範囲比較 (>, ≥, <, ≤) の場合 → テーブル 1

を選び、係先のテーブル 0 と結合する。これは、サブ条件がないものは比較基準の役割を果たさず、範囲比較では第二引数を比較基準とするように意味表現を決めているためである。上記に当てはまらない場合は以下を行う。

- テーブル 0 とテーブル 1, 2 とを結合する。ただしテーブル 1, 2 でそれぞれサブ条件 1, 2 により値の限定がされたキーは結合キーから除き、矛盾が生じるのを防ぐ。

以上の処理は、図 3(1) のツリー構造での最下条件からボトムアップに行う。

図 2 の例では、補足する条件として図 3(2) の結合条件 a: '会社 4. 会社番号 = 営業 3. 会社番号'、結合条件 b: '会社 1. 会社番号 = 営業 2. 会社番号' の順に決まる。

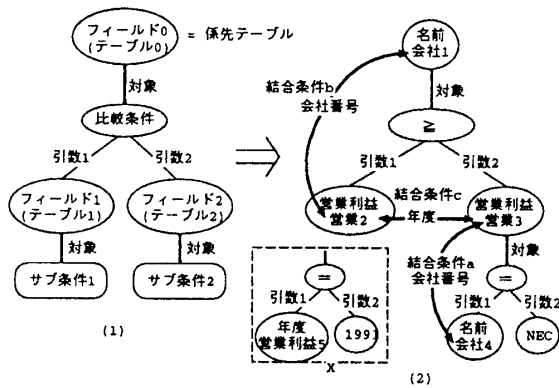


図 3: 条件テーブルの選択について

3.2 比較の引数テーブル間での結合条件補足

IF-Kit では、比較の引数における対象や条件の省略を入力文解釈において補完し、意味表現を作る。よって省略表現により引数のテーブルの一方だけでキーの値が決まり、他方のテーブルとの結合が必要となることは起きない。例えば「1991年のNECより多くの営業利益を挙げた会社は。」の場合、「1991年のNECの営業利益より、1991年の営業利益が多い会社」と解釈して、図3(2)のXのサブ条件が営業2,3の両方に付く意味表現を作り、営業2,3テーブル間の結合条件補足は必要ない。

しかし多対多のフィールド同士の比較条件で、引数テーブルの両方で制約されていない結合キー(KEY1)がある場合には、引数テーブル間に結合条件の補足が必要となる。このとき両引数のテーブルをKEY1により結合すれば良い。図2の例では、前節で述べた結合条件だけでは年度に関係なく営業利益を比較してしまうが、結合条件c: '営業2.年度 = 営業3.年度' を補足すると各年度毎に比較を行う。

4 対象およびグループ化項目の補足

IF-Kit では、データベース構造を知らない検索者を想定する。そのため検索者がデータベースを十分知らずに入力で複数の検索結果を区別するに足る対象を指定しない、または省略する場合が考えられる。このとき次のような検索の対象の補足が必要となる。

4.1 検索の対象がフィールドの場合

検索式内のテーブルで条件から値が一意に決まらないキーが検索の対象に入っていない場合、複数の検索結果の区別がつかず補足が必要である。例えば図2の例で対象を補足しなければ会社名だけ表示され、何年度においてNECより多くの営業利益をあげたのかわからない。

出力情報を区別するために、値が一意に決まらないキーを対象に補足する。ただし、値が一意に決まらないキー同士が結合されている場合、両方を対象に補足しては冗長となるため、一方のみを補足する。

図2の場合、'営業2.年度' を対象に補足する。

4.2 検索の対象が集合関数の場合

総計や平均など集合関数が対象の場合、集合関数を実行するグループを指定した、グループ化項目を対象に補足する。例えば、図1のデータベースで「業界毎の営業利益は?」では「業界」を補足する。しかし図4の売上テーブルで「NECの各月毎の売上は?」に対して上記の補足方法では図5(a)のような結果となり、意図した(b)のような検索をしない。これは「月」の値(1(月))の示す実体が、「年度」の値(1990(年度)や1991(年度))により異なるからである。2つのキーの一方の値の実体が他方のキーの値に依存する場合に、2つのキーを同じ属性と呼び、後者は前者より上位であると呼ぶ。グループ化項目内に同属性のより上位のキーが存在せず下位のキーだけが存在する場合、実体が異なるものを等しく扱うことになるので、補足が必要である。

テーブル名	フィールド名	キー	属性	上位からの順位	結合キー
売上	会社番号	○	A	1	会社.会社番号
	年度	○	B	1	
	月	○	B	2	
	日付	○	B	3	
	売上額				

図 4: 同属性のキーを持つデータベーススキーマの例

月	売上額	年度	月	売上額
1	(1990年1月と1991年1月の売上総計)	1990	1	(1990年1月の売上総計)
2	(1990年2月と1991年2月の売上総計)	1990	2	(1990年2月の売上総計)
3	(1990年3月と1991年3月の売上総計)	1991	1	(1991年1月の売上総計)
...

図 5: 「NECの各月毎の売上は?」の検索結果

解決方法としては、各キーに関し図4のように属性の同異や上下位関係の情報を保持する。対象が集合関数の場合、入力グループ化項目のキーと同属性で上位のキーを、検索の対象とグループ化項目に補足する。

5 おわりに

自然言語からのSQL生成における、テーブル結合条件補足と対象・グループ化項目補足の問題点とその解決法について述べた。現在、本稿で述べた方式をIF-Kitにおいて実現し、評価を行っている。また本方式のテーブル結合条件は、検索の可能性にのみ着目し設定したが、今後は検索効率も考慮し、直接結合できないテーブル間の最適パスの選択手法なども考えていく予定である。

【参考文献】

[1] 宮部, 「日本語インタフェース文脈文法」情報処理学会第45回全国大会 2F-1 論文集 (1990)
 [2] 市山, 「自然言語インタフェース構築キットの提案」情報処理学会第43回全国大会論文集 1H-2 (1991)
 [3] 山口、市山, 「拡張意味ネットワークの連想検索に基づく言語理解」情報処理学会第43回全国大会論文集 5H-10 (1991)