データベース用自然言語インタフェースにおける データベースの自動連結

5Q-2

伊藤 篤　高橋 清一

㈱ＣＳＫ総合研究所

１．はじめに

我々は、パソコンベースで普及しているリレーショナル・ＤＢＭＳを対象とし、自然言語インタフェース「ｄＢｍａｔｅ」を開発した。このシステムは、自然言語からＤＢＭＳのコマンドを生成・実行し、自然言語による情報検索，情報操作を可能にしている。

ここでは、自動的にデータベースを選択し、さらに必要に応じて２つのデータベースを連結する手法について述べる。

２．概要

ユーザの入力した自然言語文には、操作対象となるデータベースの指定が欠けている場合がある。このようなときは、自然言語文に含まれるフィールド指定から、対象となるデータベースを選択しなければならない。また、場合によっては一つのデータベースから情報を得ることができないので、このときはデータベースを連結することが必要となる。データベースを連結する場合には、キーフィールドを定めなければならない。

さらに、自然言語文にはデータベース指定が含まれていることもあるので、このような場合は、フィールド指定から導き出された対象データベースとの整合性をとることが必要になる。

手順としては、まず、一つのデータベースから情報を得られる場合の、そのときのデータベースの集合（nrel）を作成する。次に、データベースを連結しなければならない場合の、そのときの２つのデータベースとキーフィールドの集合（rel）を作成する。最後に、この２つの集合とユーザのデータベース指定との整合性をとり、さらに、必要ならメニュー形式でユーザに選択させる。

３．データベースの選択

ユーザが作業対象としている全てのデータベースの集合を Dbs とし、Dbs に属する全てのフィールドの集合を Fields とする。ユーザの入力した自然言語文に含まれるフィールドの集合を $I \subseteq Fields$ とする。

$$Dbs = \{d_1, d_2, \ldots, d_n\}$$
$$Fields = \{f \mid \exists d \in Dbs, d\text{はフィールド}f\text{を持つ}\}$$

$f \in Fields$ は、各データベースごとに違うのではなく、フィールド名が同じならば同じフィールドと考える。従って、f を持つデータベースの集合が考えられ、これを D(f) とする。

$$D(f) = \{d \in Dbs \mid d\text{はフィールド}f\text{を持つ}\}$$

さらにフィールドには属性として、頻度 $freq(d,f) \in \{many, few, normal\}$ を持たせる。頻度はそのフィールドに含まれるデータの異なり語数から算出されるもので many, normal, few の順に異なり語数が多いことを意味する。異なり語数が多いほどキーフィールドに向くと言える。フィールド名が同じでもデータベースによって頻度は異なる場合があるので、freq() の引数にはデータベースの指定が必要である。ｄＢｍａｔｅは、あらかじめ（システム立ち上げ時に）データベースを調べて、この情報を得る。

まず、最初のステップとして、I から２つの集合、nrel と rel を作成する。

１）nrel の作成

nrel はデータベースを連結しないで情報が得られる場合の候補となるデータベースの集合である。nrel は次のようにD(f)の積集合として定義される。

$$nrel = \{d \in Dbs \mid \forall f \in I\text{に対して}d \in D(f)\}$$

２）rel の作成

rel は２つのデータベースを連結して情報が得られる場合の候補となるデータベースとそのときのキーフィールドの集合である。２つのデータベースは順序が違っていても同じものとする。

I の全ての２つの要素の組合せ (f_1, f_2) に対してデータベースの組合せ (d_1, d_2) の集合 r を求める。f_1, f_2 と d_1, d_2 の関係は「(d_1 は f_1 を持ち f_2 を持たない) かつ(d_2 は f_2 を持ち f_1 を持たない) かつ、d_1, d_2は共通のフィールドを持つ」となることである。

$$\begin{aligned} r = \{(d_1, d_2) \mid\ & d_1 \in Dbs, d_2 \in Dbs, d_1 \neq d_2, \\ & \exists f_1, f_2 \in I, f_1 \neq f_2, \\ & \quad d_1 \in D(f_1), d_1 \notin D(f_2), \\ & \quad d_2 \notin D(f_1), d_2 \in D(f_2), \\ & \exists k \in Fields, \\ & \quad d_1 \in D(k), d_2 \in D(k)\} \end{aligned}$$

この (d_1, d_2) の組合せに対してキーフィールドを次の戦略によって定める。

①共通のフィールドが一つならば、それをキーフィールドにする。

②共通のフィールドが複数あるならば、$freq(d_1, k) = many$ かつ $freq(d_2, k) = many$ となるフィールドを選ぶ。

③②を満たすものがなければ、$freq(d_1, k) = many$ または $freq(d_2, k) = many$ となるフィールドを選ぶ。

④②～③によってもキーフィールドが一意に決まらない場合には、データベース d_1 でできるだけ左側にあるフィールドを選ぶ。

キーフィールド k が決まったら、この (d_1, d_2, k) の集合を rel' とする。すると rel は次のように定義される。

$$rel = \{(d_1, d_2, k) \in rel' \mid \forall f \in I\text{に対して}d_1 \in D(f)\text{または}d_2 \in D(f)\}$$

Automatic connection of two databases,
in Natural Language Interface for DBMS
Atsushi ITOH, Seiichi TAKAHASHI
CSK Research Institute

３）ユーザ指定との整合性

集合 nrel と rel が作成されると、ユーザのデータベース指定と見比べて場合分けし、操作対象データベースを定める。ユーザの指定したデータベースの集合を J⊆Dbs とする。

①J＝∅（空集合）のとき

ⓐnrel≠∅ のとき

nrel の中からいくつかをユーザにメニューで選択させ、これを新たに nrel とする。

ⓑnrel＝∅ かつ rel≠∅ のとき

rel の中からいくつかをユーザにメニューで選択させ、これを新たに rel とする。

ⓒそれ以外の場合

エラーとする。

②J≠∅ のとき

ⓓnrel≠∅ のとき

J∩nrel≠∅ ならば、J∩nrel を新たに nrel とする。このとき、J-nrel≠∅ ならばユーザに余分なデータベース指定があることを意味する警告を出す。

J∩nrel＝∅ ならば、J の持つ全てのフィールドを I に付け足して、もう一度だけ ２）３）をする。それでもダメならエラー。

ⓔnrel＝∅ かつ rel≠∅ のとき

まず、rel を次の３つに分ける。

$r_1=\{(d_1,d_2,k)\in rel \mid d_1\in J かつ d_2\in J\}$

$r_2=\{(d_1,d_2,k)\in rel \mid d_1\in J または d_2\in J\}-r_1$

$r_3=rel-r_1-r_2$

ここで、r1≠∅ ならばこれを新たに rel とする。

$r_1=\varnothing$ かつ $r_2\neq\varnothing$ ならば、r_2 の中からいくつかをユーザに選択させて新たに rel とする。

それ以外ならばエラーとする。

ⓕそれ以外の場合

エラーとする。

４）データベース選択の例

次のような２つのデータベース「在庫表」と「台帳」がある場合の例をやってみる。

データベース「在庫表」

製品番号	在庫 …
K0010	1100
T1121	253
K0030	741
⋮	

データベース「台帳」

製品名	製品番号	原価
CRIPRO	A1134	68000
dBmate	K0010	28000
APLISP	K0030	111000
	⋮	

入力文は「在庫×原価と製品名を求める」とする。データベースの集合 Dbs とフィールドの集合 Fields は次のようになる。

Dbs＝{在庫表,台帳}
Fields＝{製品番号,在庫,製品名,原価,...}

また、入力文に含まれるフィールドの集合 I とデータベース指定の集合 J は次のようになる。

I＝{在庫,原価,製品名}
J＝∅

さて、nrel を求める。D(在庫)＝{在庫表},D(原価)＝{台帳},D(製品名)＝{台帳} なので、nrel は次のように∅になる。

nrel＝D(在庫)∩D(原価)∩D(製品名)
　＝{在庫表}∩{台帳}∩{台帳}
　＝∅

よって、一つのデータベースから情報を得ることができないことがわかる。

次に、rel を求める。組合せ（在庫,原価）に対しては、(在庫表,台帳）が、(在庫,製品名）に対しては（在庫表,台帳）というデータベースの組合せがみつかる。(原価,製品名）に対しては対応する組合せはない。従って、r は次のようになる。

r＝{(在庫表,台帳)}

この組合せに対してキーフィールドとなるのは製品番号だけなので rel' は次のようになる。

rel'＝{(在庫表,台帳,製品番号)}

在庫表，台帳とも I の要素を全て持つので、rel＝rel' である。さらに、J＝∅,nrel＝∅ から rel は次のようになる。

rel＝rel'
　＝{(在庫表,台帳,製品番号)}

rel の中からユーザにメニューで選択させ新たに rel としなければならないが、要素が一つしかないので、このままでよい。

以上から、「在庫表」と「台帳」をフィールド「製品番号」をキーにして連結すればよいことがわかった。

４．コマンド生成

nrel または rel が定まると、これを元にコマンドを生成する。すなわち、nrel≠∅ ならば、nrel に含まれるデータベースの数だけコマンドを生成する。nrel＝∅ ならば rel の要素の数だけ、データベースの連結をするコマンドを生成する。

エラーのときはデータベース指定に誤りがあることを示すメッセージを出力する。

５．おわりに

ｄＢｍａｔｅでは、この方法を用いてデータベースの自動連結を行なうことができた。

今のところ、データベースの連結は２つまでなので、今後の課題として、３つ以上の場合の連結に対応していきたい。

謝辞

本研究開発の機会を与えて下さり、討論に加わっていただいた当社関係各位に深く感謝の意を表します。

参考文献

[１]伊藤，高橋：データベース用自然言語インタフェース「ｄＢｍａｔｅ」，第38回全国大会講演論文集