動的質問合成に基づくオブジェクトベース利用者インターフェース

1E－5

吉川 正俊　柴田 典男　植村 俊亮
奈良先端科学技術大学院大学 情報科学研究科

1 まえがき

オブジェクトベースにおいてあらかじめ登録しておいた辞書中の文字列と質問部品の対をもとに、利用者が与えた疑似自然言語の名詞句/節から対応する質問言語文を動的に合成する利用者インタフェースの設計・開発を行ない、自然言語インタフェース [1] に対する現実的なアプローチを目指している。

このような利用者インタフェースでは、質問間の包摂関連や質問結果の包含関係をもとに対応する**合成文字列間の汎化、包含**などの関連を定義することができる。この文字列間の関連は文書自身を利用者インタフェースとするデータベースにおいて以下のような目的に使用できる：(1) **質問改良**: ある質問の答が空集合であった場合、単に空集合を返すのではなく、その質問の関連質問が存在するならそれを利用者に教えることができる。(2) **ハイパーリンク**: 2つの文字列間にデータベースを介した関連がある場合、どのような関連を持つかを明示したリンクを2つの文字列間に張ることができる (3) **キャッシュとしての利用**: 既に行なわれた代表的な質問とその結果をキャッシュしておく [3] ことにより、関連する質問を迅速に処理できる場合がある。たとえば質問 A の答が質問 B の答の部分集合である場合、質問 B を処理するためにまずキャッシュにある質問 A の答を返しておき、バックグラウンドで質問 B と質問 A の差集合を検索することにより、利用者は答の部分集合を高速に入手できる。

本論文では、質問間の関連を抽出するための基本的な技法について述べる。また、質問結果の出力にも注目し、利用者自身が必要とするデータを選択できるようなシステムを考え、その質問結果に対する質問や更新などの問題についても検討する。

2 質問の動的合成

質問の動的合成を行なうための部品辞書には、オブジェクトベーススキーマのクラスやメンバ変数などの構成要素を表す単語とそれに対応する質問部品を格納している。ここでは利用者が与えた名詞句/節の疑似自然言語文に対応する質問文を自動的に合成する機構を与える。本論文で対象とする質問は基本的に ONTOS の SQL[2] で表現される。ただし、(1) from 句には一つのクラスは高々一回のみ現れる、(2) where 句には条件式の連言のみを許し選言や否定は許さない、(3) select 句には from 句で宣言した変数のうちの一つが現れる、(4) where 句に現れる経路式の長さは高々2とする。という四つの条件を満足する質問クラスのみを考える。

User Interfaces of Obejectbase based on Dynamic Query Synthesis
Masatoshi YOSHIKAWA, Norio SHIBATA and Shunsuke UEMURA
Nara Institute of Science and Technology

辞書は名詞節のためのもの (DictE) とそれらを連結する語句のためのもの (DictR) の二種類から成る。部品辞書 DictE, DictR の例を図 1に示す。各辞書には文字列と対応する SQL 文が格納されている。ただし、SQL 文は合成が容易なように select, from, where のそれぞれの句に分けて部品化して格納されている。入力文字列中の名詞句に相当する部分文字列間の包含関係をグラフで表現すると枝に順序のついた二分木となる。文字列から SQL への変換はこの二分木を後順 (postorder) で走査することにより順に行なう。

3 文字列間の関連性

自然言語での入力質問文をフルテキストデータベースの中の一部の文字列としたとき、データベースから返ってきた結果に相当する箇所が文章中にあればその部分を強調表示することは意味のあることである。さらに一歩進めて、ある質問に関連のある文字列にも何らかのリンク付けを行なうことで、その文字列どうしの関連を表すことができる。

3.1 関連性の定義

3.1.1 質問（結果）の包含性に基づく関連性

質問どうしの関連を考えた場合に、その関連がスキーマを決定した時点ですでに定まったものか、あるいは与えるデータベースインスタンスによって動的に決定されるものかで分類することができる。

スキーマに基づく定義 スキーマを与えた時点で決定される関連は以下の3種類である。ただし、関連を求めたい質問の組を q_1, q_2、データベースを Dで表す。

定義 1.1 $for \forall D, q_1(D) \subseteq q_2(D)$

定義 1.2 $for \forall D, q_1(D) \cap q_2(D) = \phi$

定義 1.3 定義 1.1、定義 1.2 以外（すなわち、スキーマだけでは q_1, q_2の関連が一意に決定できない）

与えられた二つの質問 q_1, q_2が上記のいずれかの関連性を持つかを判定することを考える。まず明らかなこととして、質問 q_1のキークラス（`select` 句の要素を含むクラス）の先祖、子孫にあたるクラスを質問 q_2が含んでいなければ関連を持つことはなく、**定義 1.3** に分類される。そこで、

> 質問 q_1, q_2のキークラスを考える。これらが異なる場合や、キークラスが同じ場合でも修飾している語句のクラスがデー タベースインスタンスに無関係に共通部分を持たない場合は**定義 1.3** の関連があるといえる

に従ったアルゴリズムで**定義 1.3** の判定を行う。また、**定義 1.3** に該当しない質問については、

DictE

文字列	select 句部品	from 句部品	where 句部品	クラス名
寺院	t	Temple t		Temple
京都	c	City c	c.name = "Kyoto"	City
大都市	c	City c	c.population >= 1000000	City

DictR

文字列	直前の名詞節のクラス	直後の名詞節のクラス	where 句部品
の	City	Sights	c = s.location

図 1: 部品辞書

> q_1のキークラスがq_2のキークラスの先祖（子孫）クラスにあたり、かつq_1の修飾部がq_2の修飾部の先祖（子孫）クラス的役割 を果たすなら**定義 1.1** の関連を持つ。さもなければ**定義 1.2** の関連を持つ

に従ったアルゴリズムで判定される。

例としてq_1:「京都の寺院」とq_2:「大都市の寺院」を考える。キークラス「寺院」は共通であるが、修飾部「京都」と「大都市」が共通部分を持つかどうかは、スキーマだけからは判断できない。そこで、この組合せは**定義 1.2** に分類される。

インスタンスに基づく定義 ある質問に具体的なデータベースインスタンスを与えたときに得られる関連が３種類ある。これらは解の共通部分で分類される。

定義 2.1 $q_1(D) \subseteq q_2(D)$ ただし $q_1(D) \neq \phi$

定義 2.2 定義 2.1 でなくかつ $q_1(D) \cap q_2(D) \neq \phi$

定義 2.3 $q_1(D) \cap q_2(D) = \phi$

２つの質問がこれらの内のどの関連を持つかは、

> ２つの質問q_1, q_2のキークラス間に先祖/子孫関連のないものは**定義 2.3** の関連を持つ。それ以外の場合は両方の質問に具体的なデータベースインスタンスを与え、一方の結果が他方の結果の真部分集合であるなら**定義 2.1**、さもなければ**定義 2.2** である

に従ったアルゴリズムで判定される。

3.1.2 文字列の包含性に基づく関連性

例えば、「京都の寺院が所有する仏像」と「京都の寺院」は一般的に見て関連があるが、前節のアルゴリズムを用いると関連はない。両者の関連について考えてみると、質問q_1を表す文字列s_1が質問q_2を表す文字列s_2の一部になっているため質問q_1とq_2の間に関連があるのである。そこで、前節のアルゴリズムでは選ばれなかった候補文字列s_2が質問文字列s_1の一部、またはs_1がs_2の一部になっていないかを比較すればよい。

3.2 質問間の距離

is_a 関係を表現したグラフを考えることで、前節で述べた関連の中に近いものと遠いものが存在することがわかる。具体例を以下に示す。

		遠い	近い
定義 1.1	q_1	国宝仏像	仏像
	q_2	美術品	国宝仏像
定義 1.2	q_1	生駒	京都
	q_2	大都市	大都市
定義 1.3	q_1	大阪	京都
	q_2	仏像	奈良

4 質問結果に対する質問

利用者が試行錯誤的に検索を行ない、質問結果をもとに検索対象を明確化し再質問を行なう場合に新たに利用者によってなされた質問に関する SQL 部品をもとの SQL 質問文に追加することで質問合成を行う。すなわち、現在 **from** 句にかかれているクラスの属性を知りたい場合には **select** 句をその属性に書きかえればよい。また、**from** 句に書かれていないクラスについての情報を得たい場合には、そのクラスと現在の質問の関連を **where** 句に追加した上で **from, select** 句を書きかえればよい。

この方法の利点は、一度合成した SQL 質問文に対して再合成するので比較的はやくできるということである。またシステムを実装するのも既存のものを利用するため簡単である。またこの方法の欠点としては、質問結果に対する質問にもかかわらず最初からすべてのデータを検索してしまうことである。

5 あとがき

今後は本論文で述べた関連性をもとに質問間の関連の概念の厳密化を行なう。また、これらの機能を持つオブジェクトベースの利用者インターフェースの設計を進める。

参考文献

[1] Matthias Jarke, Jon A. Turner, Edward A. Stohr, Yannis Vassiliou, Norman H. White, Ken Michielsen: A Field Evaluation of Natural Language for Data Retrieval, *IEEE Transactions on Software Engineering*, Vol.SE-11, No.1, January 1985.

[2] ONTOS, Inc.:ONTOS Object SQL Guide for ONTOS DB Release 2.2, February, 1992.

[3] Wiktor RZECZKOWSKI and Kazimierz SUBIETA: Stored queries– a data organization for query optimization, *Data & Knowledge Engineering*, 3, 1988, pp.29-48.