基于查询语义的数据库中文界面研究

张 凯 吴丽辉 李盛韬 程学旗

中国科学院计算技术研究所 软件研究室 北京 100080
Email: zk@ict.ac.cn

摘 要：文章提出了一种基于数据库查询语义的数据库中文界面处理方法。这种方法主要关注那些对SQL语句生成有重要影响的词汇，并计算可能出现的语义，同时对语义进行可能性排序。与以往的基于语法的方法相比较，这种方法在用户友好度和响应速度上有显著的提高。

关键词：数据库，中文界面

Chinese NLIDB Based on Query Intention

Zhang Kai Wu Lihui Li Shengtao Cheng Xueqi

Software Division, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080
Email: zk@ict.ac.cn

Abstract: This article brings forward a Chinese NLIDB based on query intention. This method regards important words for generating SQL query, computes all possible query meanings, and then ranks these meanings. Compared with syntax-based methods, it greatly improves query effect both in friendly response and in response speed.

Keywords: Database, NLIDB

1 引言

数据库中文界面是指使用中文自然语言对数据库进行查询的系统。例如：用户输入“列出北京的供应商提供的零件”，系统给出查询结果。它的关键步骤是要将中文查询句转换为数据库的SQL语句。由于一般用户不会使用SQL语言，而且他们对库结构缺乏了解。这种界面在应用中还是有实际意义的。随着语音识别等输入技术的发展，它一定会得到更广泛的应用。

近年来，国内研制出很多相关系统，如RCHIQL、NCHIQL、NLCQI等([许龙飞 2002]、[崔宗军 2000]、[曹礼德 1986] 、[吴照临 1992]、[顾国梁 1990]、[张亚南 1993])。他们所用的是类似于语法和模板的技术。由于查询的对象是数据库，大部分系统都充分利用了ER模型。但是基于语法的系统必然带来对查询句的很多限制和语法分析的时间损耗。人们一般称其为“受限汉语”。实际上汉语的语法特征并不明显，陈力为就曾经在《中文信息处理丛书》序言中指出：“汉语的语法尚未形成规范化，而且人们习惯于非规范化的语法。”那么这种受限的汉语将会阻碍系统较大规模的应用。

想做到完全不受限在目前是不可能的。我们这里做的工作主要目的是将这种受限尽可能

的降到用户可以接受的程度。我们将注意力集中到那些“重要词”（能在 SQL 语句中显式或隐式出现的词）上，根据这些重要词产生若干个可能语义（这里称之为查询语义），按可能性排列，反馈给用户选择。这样可以充分利用数据库的语义和应用领域的“相对狭窄”的特点，放宽对用户语言的限制，极大的提高系统的可用性。

在本文第二节中，我们将介绍总体结构。第三节给出详细算法。第四节通过一个实例说明算法。最后是结束语。

2 总体结构

系统的结构如图 1 所示。我们将在下一节详细介绍各个部分。

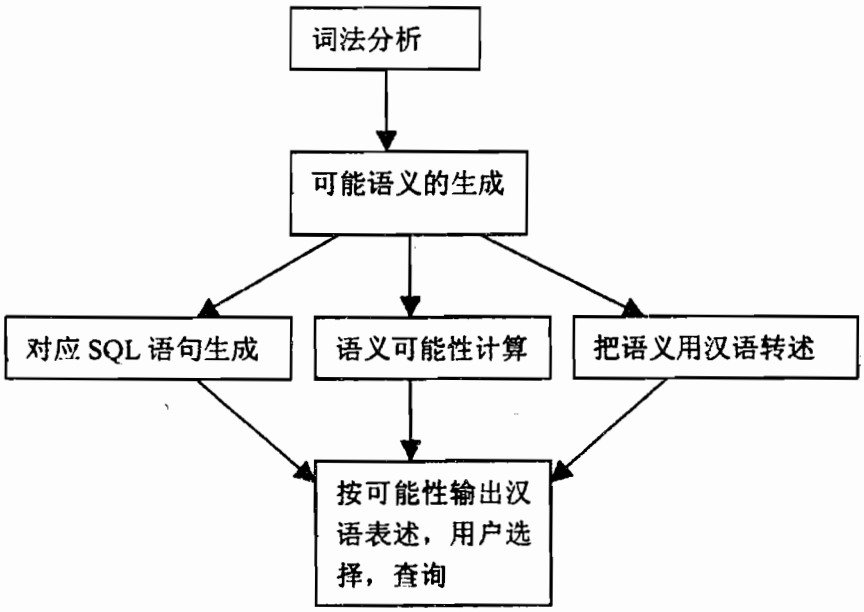

图 1 系统流程图

3 详细算法

为了更好说明算法，我们将应用领域定为供应商为工程提供零件的领域。数据库中有四个表：供应商 S(供应商号、供应商名、城市、状态)，工程 J（工程号、工程名、城市、预算），零件 P（零件号、零件名、颜色、重量），供应关系 SPJ（供应商号、工程号、零件号、数量）。

3.1 词法分析

在自然语言处理系统中一般采用分词技术。我们在实际应用中发现，很多词对 SQL 语句几乎没有什么直接的影响。这样我们就可以不进行完整的分词，而把主要的精力放在“重要词”上，即那些在 SQL 语句中显式或隐式出现的词。我们得到的切分结果是一个个构成链状的词，我们称之为词链。

例如：

请列出广州的供应商为上海的工程提供的零件

一般的分词方法的正确结果应为：

请/列出/广州/的/供应商/为/上海/的/工程/提供/的/零件

在我们的系统中得到的一条词链为

列出—广州—供应商—上海—工程—提供—零件

与词性标注相类似，对词链中的词，我们需要标记类型和辅助信息。因为我们处理的是数据库查询句，所以主要的类型有实体名 E，属性名 A，属性值 Va，查询动词 Vq，联系动词 V，比较词 Com，某些介词 Prep。在标记类型后，还要有辅助信息。如果有两条词链，它们的各个词是相同的，但是有的词辅助信息（相当于标注）不同，那么这两条词链应认为是不同的词链。经过词法分析后，我们最终得到的是所有可能的词链。

标记类型	辅助信息
实体名	对应的数据库表名
属性名	表名，字段名
联系动词	表名(可无)
属性值	数据库表名，字段名，属性值的类型(字符串，数值等)

表 1 辅助信息表

分词词典的构造时，应用领域的实体名一般是比较少的，我们可以穷举出来。其他如属性名，联系动词，比较词等也基本上可以穷举。对于字符串的属性值，如果某个字段可以出现的属性值是在一个不太大的集合里（如广州的几个区名），我们可以把它们全部录入即可。数量词可以很容易区分出来，不需要放入词典中。

3.2 可能查询语义的生成

汉语查询句一般分为祁使句和疑问句。本文只讨论祁使句的处理。根据处理的复杂度，我们将语义生成分为嵌套语义处理、或者语义处理、无或者无嵌套语义处理几部分。其中无或者无嵌套语义处理是基础，嵌套语义处理和或者语义处理只是一些特例的处理。在实际中，无或者无嵌套语义的查询占到绝大多数。因此在这里我们只讨论无或者无嵌套语义的处理。这里形成语义结构主要就是两个任务：一个是生成查询目标，一个是生成查询条件。

1．查询目标的生成

我们一般可以认为它符合三段式：查询动词+查询条件+目标短语。我们的生成算法就可以是从右向左扫描词链即可。

根据对大量查询句的分析，目标短语一般包括以下模式:

- 查询实体(如零件)
- 查询实体+属性组(如零件的零件号和零件名)
- 代表实体的属性(如："列出白色的键盘",键盘是一个零件名)
- 同一个表的属性组(如"列出零件号和零件名")

2．查询条件的生成

查询条件的确定主要是以属性值为核心的，一个属性值一般可以生成一个条件。根据属性值的类型，采取不同的策略。具体算法如下：

Step 1：对所有的属性值进行如下分析：

（1）属性值的类型是字符串，

它一般是相等的条件。根据它的辅助信息生成一个条件。

（2）属性值的类型是数值，对其左端进行分析，

A.左端是一个比较符，而且比较符左端有一个属性名

如果属性名和数值的辅助信息相符可以生成条件

否则取消词链

B.左端是一个属性名，

如果属性名和数值的辅助信息相符可以生成等于条件

否则取消词链

C.其他情况下，根据数值的本身的辅助信息生成等于条件。

Step 2：查找未利用的比较符，观察其两端是不是属性名。（例如：列出数学好于英语的学生）

Step 3：语义结构生成后，可以根据常识过滤一些语义结构。

3.3 语义结构及其与 SQL 的转换

语义结构是一种中间表示。语义结构的表示如下：

1. 查询目标表名，列名
2. 条件的数组,条件信息包括表名、属性名、比较符、属性值（或者另外一个表名和属性名）

我们可以看出：语义结构和 SQL 语句有很好的对应关系，从语义结构生成 SQL 是一件较为简单的事情。这里需要注意的是：

1. 统计涉及的表名，并保证在 ER 图上的连通性。
2. 填写连接属性条件，p.pno=spj.pno 等。

3.4 语义可能性计算

可能的语义生成之后，语义结构的数量较多，我们应该计算出哪些是最有可能的语义（SQL 语句）。这就是语义可能性的计算。

经过对各种考虑因素的探讨和试验，我们认为如下的一组因素可以较好得计算语义可能性：(按重要程度排序)

1．词链在查询句的覆盖长度，越长越好.

这里主要是考虑到自然语言中的词的最大匹配原理，

例如：*列出广州所有的供应商号*

供应商号，供应商，供应都是系统中的词

（列出—广州—供应商号）的概率就大于

（列出—广州—供应商）

（列出—广州—供应）

2．查询所涉及的表的个数,越少越好

在计算表的数目的时候，首先要解决连通问题。因为一个合理的查询,在 ER 图中一般表现为一个连通的图。我们看以下例子：

查出广州的供应商给北京的工程的零件。

假设这里"给"不是一个词表中的词。因为供应商，工程，零件三者必须通过 SPJ 表才能连接起来。我们在计算表的数目的时候，应当把作为连接的表考虑进去。在这里我们也可以看到如果某些词的语义缺失，系统可以根据 ER 图对这些语义自动补足。

我们看这样一个例子：

列出在上海的工程.

该例子中，上海既可以看作是工程的城市,也可以看作是供应商的城市。如果把它看作是工程的城市，语义与查询句语义一致。如果把它看作是供应商的城市，其语义是：“列出由上海的供应商供应零件的工程”，语义与查询句语义不符。

这里把上海作为供应商城市时，要涉及到三个表,把上海作为工程的城市只涉及一个表，按照上述原则，把上海看作是修饰工程的,权重较大.

3．修饰距离和，越短越好.

如:*列出广州的供应商为上海的工程提供的零件.*

可能有两种语义:

a. 认为广州修饰供应商,上海修饰工程

b. 认为广州修饰工程,上海修饰供应商.

修饰距离定义为修饰词到被修饰词的距离。因为自然语言的特点是修饰词和被修饰词之间一般较近修饰,这样修饰距离较近的语义就是较为正确的答案.

此外我们还可以利用数据库和应用领域中的其他信息

3.5 反馈算法

这里反馈算法较为简单，可以用模板的方法生成。例如：在查询目标为零件，供应商和工程都出现的情况下，模板为：

列出<供应商短语>为<工程短语>提供的<零件短语>[<属性列表>]

4. 运行实例

例：*帮我找一下广州的供应商提供给上海的工程的零件*

1．找词链，这里我们找到的词链总共有八条。其中一条是：

找(Vq)—广州(Va,S,city)—供应商(E,S)—提供(V ,SPJ) —上海(Va,S,city)—工程(E,J)—零件(E,P)。

其它词链有三条与该词链词相同，但辅助信息不同，另外有四条是由（找—广州—供应—提供—上海—工程—提供—零件）生成的。

2．生成语义结构

首先找查询目标，从后向前找，找到查询目标为实体“零件”。

然后找条件，这里作为属性值的是“广州”和“上海”，找到两个条件。

这几个词链都可以生成语义结构。这里需要注意的是，有四个词链的语义有问题，正常用户不可能有这样的查询要求（即供应商的城市不可能既为广州也为上海），所以这些语义被过滤。

其它 4 个语义结构进入下一步处理。

3．语义可能性计算

首先根据覆盖长度，出现“供应”的词链权值小于“供应商”的。在出现“供应商”的两条词链，根据修饰距离长短确定哪个词链对应语义的可能性更大，所以“广州”修饰“供应商”的语义较好。

由这个语义结构转换成的 SQL 语句将是：

Select p.*

```
From s,p,j,spj
Where s.city="广州" and j.city="上海"
  And s.sno=spj.sno and p.pno=spj.pno and j.jno=spj.jno
```

5 结束语

文章中提出的模型摆脱了受限汉语文法的束缚，在测试运行中，用户的查询句基本上都有正确的反馈，而且对于绝大多数查询句，系统的第一候选即是用户的需求。在运行速度上由于没有语法分析过程，也比既有系统快了很多。当然这种方法还有很多值得探索的地方，如怎样才能较为完美的处理与或非问题，如何充分利用周围的虚词等。

致谢：感谢北京大学唐世渭教授、俞士汶教授、杨冬青教授的指导。

参 考 文 献

[张凯 2001] 张凯 RCHIQL 中非规范查询句的处理 北京大学硕士论文，2001

[许龙飞 2002] 许龙飞，杨晓昀，唐世渭 基于受限汉语的数据库自然语言接口技术研究 软件学报 第 13 卷 第 4 期， 2002

[崔宗军 2000] 崔宗军,唐世渭,杨冬青，“基于 ER 模型的数据库受限汉语查询界面 RChiQL 的文法分析系统研究”，中文信息学报，第 14 卷,第 4 期，2000.7

[曹礼德 1986] 曹礼德，姚天顺，关系数据库上泛关系查询与中文查询语言的接口，中文信Q 息学报，1986 年，第 1 期

[吴照临 1992] 吴照临，高广峰，CDSA 模型及其在关系数据库自然语言接口中的实现，中文信息学报，卷 5（4）， 1992

[顾国梁 1990] 顾国梁，王能斌。数据库汉语查询接口的设计与实现，计算机学报，第 13 卷第 12 期，1990

[张亚南 1993] 张亚南，徐洁磐。数据库 NL 界面上汉语查询的 EAAD，计算机学报，第 16 卷第 12 期，1993