

基于自然语言处理技术的政务智能搜索引擎应用探索

姚俊华¹, 汤代佳²

(1.深圳市南山区政务服务数据管理局, 广东 深圳 518000;

2.深圳市联合信息技术有限公司, 广东 深圳 518000)

✉yaojunhuasz@163.com; tangdaijiasz@163.com



摘要: 为适应问答系统智能化程度越来越高的特点, 提出基于自然语言处理转化为SQL(Structured Query Language, 结构化查询语言)语言技术的政务智能搜索引擎系统。用户通过输入自然语言问题直接获得相关数据, 数据可以表格、图形等方式直观地显示。建立了融合SQL语法和增强列信息的算法模型SQL Model, 利用NL2SQL(Natural Language Processing to Structured Query Language, 自然语言转化为结构化查询语言)技术设计政务智能搜索引擎系统, 并以某市的人口数据进行实验。实验结果表明, 该技术可有效降低数据应用的复杂度, 实现多维度复杂查询, 降低业务部门数据搜索应用难度, 提高政务数据搜索效率。

关键词: 问答系统; NL2SQL; 政务智能搜索引擎系统; SQL Model

中图分类号: TP31 **文献标识码:** A

Research on the Application of Government Affairs Intelligent Search Engine based on NL2SQL Technology

YAO Junhua¹, TANG Daijia²

(1.Shenzhen Nanshan District Government Affairs Service Data Management Bureau, Shenzhen 518000, China;

2.Shenzhen United Information Technology Company, Shenzhen 518000, China)

✉yaojunhuasz@163.com; tangdaijiasz@163.com

Abstract: In order to accommodate the increasingly intelligent characteristics of the question answering system, this paper proposes an intelligent search engine system for government affairs based on the technology of transforming natural language processing into SQL (Structured Query Language) language. Users can directly obtain relevant data by inputting natural language questions, and the data can be displayed intuitively in tables, graphs, and so on. An algorithm model of SQL Model is established that incorporates SQL syntax and enhances column information. NL2SQL (Natural Language Processing to Structured Query Language) technology is used to design the intelligent search engine system for government affairs, and an experiment is carried out with the population data of a city. The experimental results show that this technology can effectively reduce the complexity of data application, realize multi-dimensional complex query, reduce the application difficulty of data search in departments, and improve the data search efficiency of government affairs.

Keywords: question answering system; NL2SQL; government affairs intelligent search engine system; SQL model

1 引言(Introduction)

随着“互联网+”政务公开工作的不断推进, 各级政府网站的信息数量快速增加, 对搜索结果的准确度要求和搜索引擎的便利度要求进一步提高。近年来, 人工智能在机器视觉领域取得了突破进展, 人工智能与NLP(Natural Language Processing, 自然语言处理)结合的NL2SQL将非专业用户查询

数据库的需求直接转换为SQL查询语句, 无须人为构造SQL语句再进行查询。

NL2SQL技术能有效降低数据检索难度, 在国内外得到广泛研究。对于英文NL2SQL, 研究人员先后提出了将输入语句转为SQL的算法^[1-2], 增加查询系统的维度和数量^[3-4], 基于语义规则和动名词列表实现语句和数据库的映射^[5-6]; 并开发

Analyza系统^[7]为缺少SQL基础的用户提供自行访问数据的途径和NL2SQL的相关系统和算法^[8]，以此提高自然语言处理和SQL查询的转换便利度。对于中文NL2SQL，则有构建语义依存树作为自然语言处理和SQL查询的中介^[9]；基于状态转换图的词法分析和语法分析方法^[10]，将问题转为SQL语句；构建基于Nivre算法的语义依存树生成模型^[11]，提出了一种歧义语言的解决方法；提出改进单词提取技术^[12]简化自然语言与SQL代码的转换方法。

根据政务数据的应用特点，基于NL2SQL技术构建了SQL Model(Structured Query Language Model, 结构化查询语言模型)，用于自动生成SQL语句，并通过设计政务智能搜索引擎系统，实现数据搜索并得到结果。为验证模型有效性，本文采用深圳市人口数据的多个维度信息进行实验。

2 SQL Model构建(SQL model construction)

此部分主要介绍在单表查询的场景下完成自然语言处理数据库任务的SQL Model，它是一种以BERT(Bidirectional Encoder Representations from Transformer, 双向转换的编码器)模型和GRU(Gate Recurrent Unit, 门控循环单元)模型为基础，融合SQL语法和增强列关系的神经网络模型。

2.1 SQL Model思路

2.1.1 融合SQL语法

目前，电子政务系统主要使用关系数据库存储数据，需要采用SQL语句对关系数据库进行访问和处理操作。根据SQL语法规则将SQL查询语句分为查询列、条件列、聚合符号、操作符、条件值，分别建立一个子模型完成预测，在基于SQL语法规则建立模块之间的关系依赖图，如图1所示。

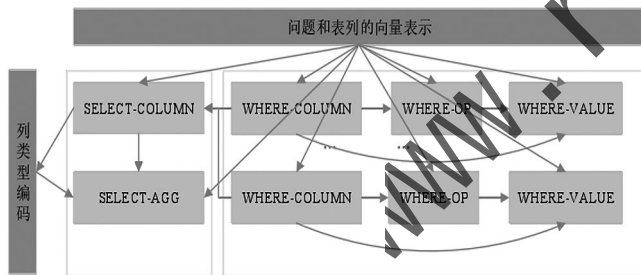


图1 SQL语法规则关系依赖图

Fig.1 SQL syntax rule dependency graph

图1中，信息沿着箭头的方向在子模型之间传递，子模型按照传递的顺序执行。这样的设计既保证模型满足语法约束，也符合SQL查询语句编写过程中的逻辑，模型在更新过程中会逐渐改进，使学习到的SQL查询语句符合SQL语法规则。

2.1.2 列单元关系的增强

在实际场景中，用户的表达同数据库的列名及单元格相比并不总是一致的，这会导致列名和自然语言表述不一致的问题。列名能够反映的信息是非常有限的，并且一个列名也会有不同的表述方式，这将会引起列信息不足的问题，即自然语言处理难以预测SQL语句需要操作的表列。

单元格信息是表列的具体表示，表列名是对该列所有单元格信息的概括。由于表列信息不足，因此经常出现仅通过列名不能获悉正确且具体的列信息，导致提取的数据错误，

无法满足用户需求。针对以上问题，采用的解决方案是将单元格信息加入表列信息中。已有一些学者将单元格信息加入表列中，比如将单元格向量和表列向量拼接为新的表列表示，其中单元格向量由以单元格的共现次数作为权重得到^[6]。这种加入单元格信息的方式对模型的性能有一定的提升，但存在引入噪声单元格的可能性。

通过注意力机制在表列信息中加入单元格信息，通过模型自动学习得出信息增强的列，而不是简单的信息计算与拼接。由于BERT模型具有自我注意力机制的核心计算方式，故不需要额外搭建自我注意力机制模块，从而降低了SQL Model的复杂度，提高了模型的优化效率，加快了模型的优化速度。

2.2 SQL Model结构

在SQL Model中实现了融合SQL语法和增强列单元关系，整个模型分为共享层和子任务层，模型的结构如图2所示。自然语言问题、表列信息和共现单元格信息先传入共享层，由BERT和BI-GRU(Bi-directional Gated Recurrent Unit, 双向门控循环单元)学习到语义信息，并增强列单元格信息。共享层学习到的语义信息再传输到子任务层，融合SQL语法，本模型将按照关系依赖图依顺序执行子模型，组合所有子模型的预测结果即为最终的SQL查询语句，SQL Model框架如图2所示。

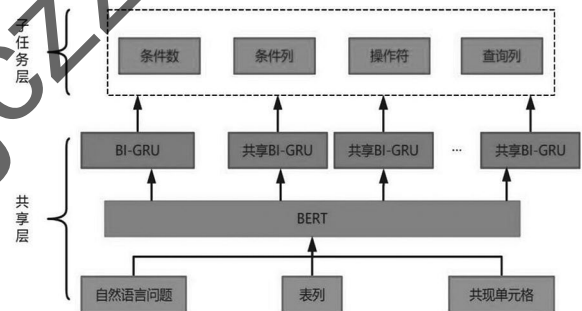


图2 SQL Model框架

Fig.2 SQL model framework

2.2.1 共享层

共享层主要是进行信息融合和特征提取，由BERT和BI-GRU构成。首先将自然语言问题、表列和共现单元格输入BERT完成信息融合和初步的特征提取，然后通过BI-GRU完成更深层次的特征提取工作。

对于给定的一个表格和自然语言问题，希望生成问题对应的SQL查询语句。使用BERT编码器对输入进行编码，对于输入的每个单词的字典索引，编码器都能产生相对应的输出向量，BERT的自注意力机制可以将单元格信息融合到表列信息中，实现对表列的增强。

得到BERT模型的输出之后，使用两个BI-GRU分别完成自然语言问题和表列信息的提取，这两个BI-GRU的工作方式不完全相同。如图3所示，在自然语言问题上运行BI-GRU，能得到每个时间节点的输出；对于表列，每一个列都运行BI-GRU模型，并且只保留最后一个时刻的输出。

BERT和BI-GRU组合成为共享层，能够进行信息融合和特征提取，输出自然语言问题及表列问题的特征向量，将这些特征向量输入子任务层以完成SQL查询语句的生成任务。

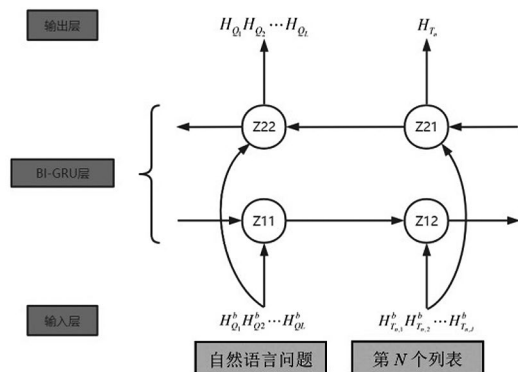


图3 BI-GRU完成深层次的特征提取

Fig.3 Completion of deep feature extraction by BI-GRU

2.2.2 子任务层

SQL查询语句的生成任务可以分成四个子模块，分别为条件数子模型、条件列子模型、操作符子模型和查询列子模型。每个模块完成对应的功能，将每个模块的预测结果进行合并得到最终的预测结果。

整体的预测流程如下：首先，对条件数进行预测，即预测WHERE(SQL中用于规定选择标准的语法)子句的条件数量。其次，模型开始预测WHERE子句中每个条件的条件列、操作符及列值。最后，执行SELECT(SQL中用于从数据库中选择数据的子句)子句部分，并引入条件列的信息。执行完所有的模块之后，将预测结果进行组合得到最终的SQL查询语句。

(1)条件数子模型

确定查询语句的第一步是要预测WHERE语句的条件数,本文采用的条件数个数0—4,将本问题看作一个五分类的问题。本任务需要预测与自然语言问题相关联的表列,所以需要将自然语言问题与表列信息进行深度融合。

对条件数进行预测具体包括以下步骤。

步骤1: 计算每个表列的注意力得分 $a_i^{w-num} = W_1 \times H_{T_i}$, 得分向量为 $a^{w-num} = [a_1^{w-num}, a_2^{w-num}, \dots, a_N^{w-num}]$ 。

步骤2：对得分向量进行归一化得到权值， $\beta^{w-num} = \text{softmax}(a^{w-num})$ 。

步骤3: 对每个列编码 H_{t_i} 进行加权求和得到整个表列的特征, 如式(1)所示。

$$T^{w-num} = \sum_i \beta_i^{w-num} \times H_{T_i} \quad (1)$$

步骤4: 用步骤1到步骤3的方法构造整个问题的向量表示 q^{w-num} 。

步骤5: 将表列向量 T^{w-num} 作为GRU的初始化向量, 并将自然语言问题 H_Q 输入GRU中, 得到融合率表列信息的自然语言问题向量序列 $[h_{q_1}^{w-num}, h_{q_2}^{w-num}, \dots, h_{q_L}^{w-num}]$ 。

步骤6: 在问题表示上应用一个多层的感知器, 得到SQL查询语句中概率最大的条件数 K^{w-num} , 如式(2)所示。

$$K^{w-num} = \max(\text{softmax}(W_2 \times \text{relu}(W_3 \times q^{w-num}))) \quad (2)$$

其中, W_1 、 W_2 、 W_3 都是模型的训练参数。

(2)条件列子模型

孩子模块的任务是对条件列进行预测，是整个WHERE子句中最重要的一部分。关键在于预测表列是如何学习出自自然语言问题与该列的相关信息，具体步骤如下。

步骤1: 计算自然语言问题中每个词关于每个表列的注意力得分矩阵, 如式(3)所示。

$$V^{w-col} = H_T(H_O \times W_4)^T \quad (3)$$

步骤2: 在矩阵 V^{w-col} 上, 应用 softmax 函数, 如式(4)所示。

$$A^{w-col} = \text{softmax}(V^{w-col}) \quad (4)$$

步骤3: 通过哈达马计算进行加权, 如式(5)所示。

$$M^{w-col} = A_{reshape}^{w-col} \times H_{O-reshape} \quad (5)$$

步骤4: 对 M^{w-col} 在第二维度进行求和得到自然语言问题关于表列信息, 如式(6)所示。

$$H_{O|T}^{w-col} = \text{sum}(M^{w-col}) \quad (6)$$

步骤5: 将注意力信息表示和表列表示在最后一个维度进行拼接, 如式(7)所示。

$$E^{w-col} = [H_T \times W_5 : H_{OT}^{w-col} \times W_6] \quad (7)$$

步骤6: 通过relu函数和softmax函数获取条件列的最终预测 p^{w-col} , 如式(8)所示。

$$p^{w-col} = \text{softmax}(\text{relu}(E^{w-col})W_7^T) \quad (8)$$

公式中的 W_4 、 W_5 、 W_6 、 W_7 都是模型的训练参数。根据条件数的预测结果, 选择前 K 个概率最高的列作为条件列的最终预测结果。

(3)操作符子模型

该模块是对条件列的操作符进行预测，本文的数据集的操作符只有(<, >, =)，所以可以将该任务视为一个三分类问题。在自然语言问题和条件列已知的情况下，只需要计算它们之间的注意力信息。

对于注意力信息的计算,首先从共享层学习到的表列矩阵 H_{T-1}^{uscp} 中提取得到条件列的向量表示,组成一个新的表列矩阵,其中每一行表示为一个条件列的特征,当预测的条件列个数少于4时,把不足4的行用数字零进行填充;并通过式(6)对这个新的表列矩阵进行列注意力机制计算,然后参照式(7)进行向量拼接,最终计算得到操作符,如式(9)所示。

$$p^{w-op} = \text{softmax}(\text{relu}(E^{w-op} \times W_7^T)W_8^T) \quad (9)$$

(4) 查询列子模型

该查询列子模型的任务是对查询列进行预测，查询列与条件列的任务相似，两者的区别在于条件列查询的结果只有一列，而查询列子模型查询的结果可以是多列，即将本任务看作一个 N 分类问题， N 代表当前表格的列数，该子模型的关键是通过学习得出自然语言问题，并按照每个表列信息进行表示。

首先通过执行式(3)一式(6)的列注意力机制得到注意力信息 H_{QT}^{s-col} ；其次将条件列的信息 E^{w-col} 引入这个查询列子模型中，采用拼接的方式将条件列信息、表列信息和注意力信息在第二个维度进行拼接，得到 $E^{s-col} = [H_T \times W_{14} : H_{QT}^{s-col} \times W_{15} : E^{w-col} \times W_{16}^T]$ ；最后通过 relu 函数和 softmax 函数得到查询列的最终预测，如式(10)所示。

$$p^{s-col} = \text{softmax}(\text{relu}(E^{w-col})W_{17}^T) \quad (10)$$

其中, $p^{s \cdot col}$ 表示所引列的概率分布, 概率分布中概率最大的就是查询列的预测结果, 其中 W_{14} 、 W_{15} 、 W_{16} 、 W_{17} 是模型的训练参数。当执行完所有的模块之后将预测结果进行组合就可以得到最终的SQL查询语句。

3 政务智能搜索引擎系统(Intelligent search engine system design for government affairs)

3.1 NL2SQL技术

NL2SQL属于语义分析的子领域,其本质是将自然输入语言问句转换成计算机可读懂、可执行、符合计算机规则的语义表示,使计算机理解自然语言,并能够准确生成表达式

句语义的可执行程序式语言。NL2SQL具有广阔的应用场景，一方面可用于基于结构化知识的智能交互(问答)，通过提取用户提问中的时点、地点、事件等多个维度的信息，完成检索需求，解决复杂的多维度问题；另一方面可用于搜索引擎优化，在搜索时区别对待常规文本和表格文本，提高搜索引擎的智能化程度。NL2SQL的应用改善了用户体验，让用户使用感更好，能够更方便快捷地获得信息服务。此外，可以有效激活企业数据库的知识价值，让数据库直接通过人工智能为用户服务，减少专业人员和中间流程引入的信息壁垒。

3.2 政务智能搜索引擎系统设计

政务智能搜索引擎系统，以SQL Model算法为基础，整体可以分为前端服务、后端平台、解析引擎及执行引擎四个模块。其中，解析引擎是NL2SQL系统的核心功能部件，包括信息抽取、表格筛选、模型编辑及指令编辑四个子模块，通过将用户搜索需求输入解析引擎，解析引擎根据用户搜索需求生成对应的SQL指令，如图4所示。

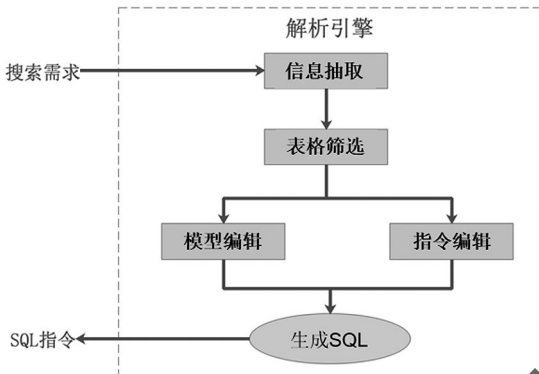


图4 智能搜索引擎系统流程图

Fig.4 Flow chart of intelligent search engine system

在搜索需求输入模型之前，系统先对其中的关键信息进行抽取，该信息可以用于表格的筛选。虽然理论上SQL Model可以编码多张表格，但是为了提高系统的性能，通常会对所有的表格先进行一次预筛选。预筛选的方式是用问句中抽取出的信息对所有表的表名及表头进行检索，召回头部的表格再送入模型中。系统完成对数据检索后，将数据应用形态进行呈现。例如，当有关部门无法通过自行操作相关页面获得数据时，就可以通过智能搜索引擎直接输入“南山区人口数据”，此时后台能自动得出数据，输出数据结果如图5所示。



图5 智能搜索引擎界面

Fig.5 Interface of intelligent search engine

例如，有关部门如需在征兵季了解南山辖区内适龄男性的数量，就可以在系统中输入“南山区人口年龄分布情况”和“南山区人口街道分布情况”进行搜索。对于搜索方式，通常做法是开发公司在后台数据库中用SQL语句进行处理和统计得到相关的数据。而通过政务智能搜索引擎系统，用户不

仅可以基于搜索数据结果获取相关信息，还可以向下钻取详细数据，通过表格、图形等方式直观地展示数据结果，如图6所示。

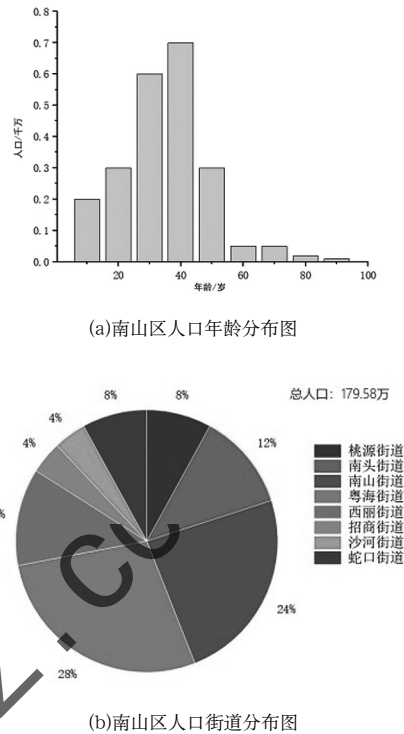


图6 政务智能搜索引擎系统搜索结果

Fig.6 Search results of intelligent search engine system

基于NL2SQL技术搭建了政务智能搜索引擎系统，业务部门可以直接在搜索界面上输入自然语言查询语句，NL2SQL会自动在后台将输入的自然语言查询语句转化为SQL语句后直接执行SQL语句，最后获得数据并显示数据，如图7所示。



图7 NL2SQL总体流程图

Fig.7 NL2SQL overall flow chart

利用NL2SQL技术搭建政务智能搜索引擎，可以提高政务数据搜索效率，降低数据应用的复杂度。本文结合数据治理技术理论和政务大数据搜索实际场景，提出政务智能搜索引擎技术框架，探索智能搜索应用新模式，可为开发和构建政务智能搜索引擎提供新的思路和具体的实现途径。

4 结论(Conclusion)

在人工智能和大数据技术高速发展的背景下，深度学习技术与各个领域的融合逐渐加深，自然语言处理领域中，传统的方法需要在后台数据库中用SQL语句进行处理统计，效率低，不能满足用户需求，而使用深度学习技术与自然语言处理领域相结合的方法可以快速且直观地将数据呈现给用户

(下转第58页)