计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200200086

结合关系分类与修正的SQL语法结构构建

万文军[1]　窦全胜[1,2]　崔盼盼[1]　张　斌[1]　唐焕玲[1,2]
1 山东工商学院计算机科学与技术学院　山东 烟台 264000
2 山东省高等学校未来智能计算协同创新中心　山东 烟台 264000
(wanwenjun131@163.com)

摘　要　针对嵌套查询中SQL语法结构难以构建的问题,提出结合关系分类与修正的GSC-RCC方法,以3类实体间关系表示SQL语法。首先设计关系分类深度模型,并引入列名常用词提升模型性能,用以确定语句中每个实体对所属不同关系的概率,以此生成无修正无向图;然后设计基于SQL语法的关系修正算法,对无向图进行修正,以此构建SQL语法结构。在房产数据查询任务中,GSC-RCC对多条件含嵌套复杂查询的语法结构生成准确率为92.25%,且可减轻模型对语句样本数的依赖。

关键词:NL2SQL;SQL语法结构;关系分类;关系修正;深度学习
中图法分类号　TP312

SQL Grammar Structure Construction Based on Relationship Classification and Correction

WAN Wen-jun[1],DOU Quan-sheng[1,2],CUI Pan-pan[1],ZHANG Bin[1] and TANG Huan-ling[1,2]
1 School of Computer Science and Technology,Shandong Technology and Business University,Yantai,Shandong 264000,China
2 Co-innovation Center of Shandong Colleges and Universities:Future Intelligent Computing,Yantai,Shandong 264000,China

Abstract　Aiming at the problem that the SQL grammar structure in nested query is difficult to construct,the GSC-RCC method combining relation classification and modification is proposed,and the SQL grammar is represented by three types of entity relationships. Firstly,the relational classification depth model is designed,and the column name common words are introduced to improve the performance of the model,so as to determine the probability of different relations of each entity pair in the statement,and then generate unmodified undirected graph. Then the relationship correction algorithm based on SQL grammar is designed to modify the undirected graph and finally construct the SQL grammar structure. In the real estate data query task,for multi-conditional query statements with nested conditions,the grammar structure generation accuracy of GSC-RCC method is 92.25%,and the method can reduce the dependence of the model on the number of statement sample.

Keywords　NL2SQL,SQL grammar structure,Relationship classification,Relationship correction,Deep learning

1　引言

自然语言转结构化查询语言(Nature Language To Structured Query Language,NL2SQL)任务旨在将自然语言解析为数据库系统可理解的规范语义表示,可以帮助用户在不掌握SQL语法的情况下,以自然语言对数据库进行查询和分析,为用户降低数据分析成本、提升决策效率和改善体验都具有重要意义。

在NL2SQL任务中,通过自然语言构建SQL的语法结构是关键的一步。目前这方面的研究多针对不含嵌套的简单语句,对于含有嵌套结构的复杂语句的处理效果并不理想,然而在自然语言中,含嵌套语句结构的复杂语句是普遍存在的。正确解析此类语句较为困难,这也是相关研究在实际应用中效果不佳的重要原因。本文中选取的研究对象是受限领域下的嵌套查询,受限汉语指对汉语自然语言强加一定规则而得到该自然语言的子集,但即使如此,由于存在嵌套等复杂语法情况,即使在确定词汇所属语义类别的基础上,对其进行解析仍较为困难。针对该问题,本文提出一种基于关系分类与修正的语法结构构建(Grammar Structure Construction based on Relationship Classification and Correction,GSC-RCC)方法。该方法主要由关系分类和关系修正两个步骤组成,关系分类步骤的特征输入是已标记实体语义类别的受限汉语查询语句,该步骤采用结合注意力机制的双向门控循环网络(Bi-GRU)[1],以此模型确定语句中所有实体对之间的关系概率分布;关系修正步骤的目的是根据约束规则,对预测的实体关系进行修正,以确定最终的SQL语法结构。本方法以实体间关系表示语法结构,从而适应在限定领域内含嵌套的复杂查询。

本文第2节介绍相关工作及本文方法与其他研究的不同之处;第3节详细介绍GSC-RCC方法的SQL语法结构生成步骤及具体细节;第4节介绍了文中的数据集,并从多个角度以实验形式验证本文方法的有效性;最后对本文方法进行了归纳总结并提出下一步研究任务。

基金项目:国家自然科学基金(61976125,61976124,61772319,61773244);高校科技计划项目(J18KA340,J18KA385)
This work was supported by the National Natural Science Foundation of China (61976125,61976124,61772319,61773244) and High Education Science and Technology Planning Program of Shandong Provincial Education Department (J18KA340,J18KA385).
通信作者:窦全胜(douquansheng1@126.com)

2 相关工作

近年来，随着自然语言处理技术的不断发展和人们对终端软件体验要求的不断提高，NL2SQL 受到相关领域研究者的广泛关注，早期方法多以统计学理论为基础，代表性的研究工作包括：Popescu 等[2]使用基于统计技术的解析器作为“插件”，能够精确地克服解析器的错误，并将解析的问题正确映射到相应的 SQL 查询；密西根大学研究者 Li 等[3]提出 XML 数据库的通用自然语言查询接口构建方法，该方法可以接受任意英语查询语句，并将其转化成 XQuery 表达式，Li 的方法通过在语法解析树中引入标识附件（Token Attachment）和标识关系（Token Telationship）的概念来解决属性名称混淆的问题，这些思想对于 NL2SQL 同样有着较强的借鉴意义；特伦托大学研究者 Alessandra 等[4]提出一个 NL2SQL 模型，该模型通过依赖关系分析和预先构建一组合理的 SELECT，WHERE 和 FROM 子句来生成一组候选 SQL 查询，并使用基于树核的 SVM-ranker 对其进行重排序，从中选择出最有可能的目标 SQL 语句；Poon 等[5]提出一种基于语义分析的无监督方法，以依赖树表示查询语句，并通过潜在的语义状态注释依赖树的节点和边缘，同时使用数据库的信息对注释进行间接监督，以此确定 SQL 语法结构。

近年来，深度学习技术[6-9]被引入 NL2SQL 领域，通常的做法是：将 SQL 语句视为 SQL 关键字和语义成分的序列，使用机器翻译领域常用的“序列-序列”模型[10-12]预测 SQL 结构，在此基础上生成目标 SQL 语句。一些代表性的工作包括：Cai 等[13]提出了一个新颖的与 SQL 语法结构紧密结合的增强型编-解码框架，将标注了语义特征的分词序列作为输入提供给编码器，解码器输出 SQL 关键字和语义成分构成的序列。为解决直接将自然语言描述映射到目标代码的序列不能保证语法正确性的问题，基于“序列-集合”的模型[14-16]被提出。其中，Xu 等[14]将 SQL 结构分解，分别预测如聚合函数、运算符等 SQL 成分，其对条件子句参照“条件列-运算符-值”的顺序依次预测，条件之间均为并列关系，顺序靠前的任务预测结果作为特征辅助预测顺序靠后的任务，以此确定 SQL 结构；在此基础上，Yu 等[15]利用了词汇的类型信息，以此更好地理解自然语言问题中的稀有实体和数字。

上述工作多针对自然语言中不含嵌套的简单语句，对于含嵌套的查询语句，有以下工作。文献[17]构建了常见语义模板，当自然语言表述符合语义模板时，将自然语言查询进行语义分析，最终解析为 SQL 语句。文献[18]使用基于 LSTM 网络的“序列-序列”模型，使用增量编号表示同类型的多个实体，将 SQL 语句结构视为 SQL 关键字与这些编号的序列。文献[19]提出一种基于模板检索与“序列-树”模型的方法，基于编辑距离检索与用户查询语句匹配的抽象语法树模板类别，以此类别模板的 N-Gram 动作序列指导生成 SQL 语句中的不同部分，其本质是先通过模板约束结构的搜索范围，再通过模型预测结构中的不同成分，每个模板需要大量示例，且每次添加新的模板都需要进行训练。在文献[19]的基础上，文献[20]改进了模板检索的方法，构建架构候选搜索网络应用于模板的初步搜索，以基于组合特征的匹配网络[26]确定查询语句所属模板，通过少量示例即可支持新的 SQL 模板，且无需再次训练，最后以指针网络（Pointing Network）[21]填充预测模板的可变槽。文献[3]提出一种基于 SQL 语法和 SQL 生成历史的解码器，设计多种预测模块以适应不同的 SQL 组件，以堆栈组织解码过程，在每个解码步骤中，从堆栈中弹出一个 SQL 标记实例（SQL 关键字或 SQL 语义成分），根据生成历史结合相应预测模块预测下一个标记实例，并将预测的标记实例压入栈中，直至堆栈为空，以解码过程中不同的关键字出现的顺序表示 SQL 成分之间的语法结构。文献[22]提出一种应用于复杂和跨域 NL2SQL 任务的分段递归神经网络模型，引入了自注意力（Self-attention）机制[23]，按“列名-运算符”的顺序预测列名与运算符，采用基于 LSTM 网络的分类器判断该条件语句中是否包含嵌套查询。如否，则预测条件值；如是，则预测一条查询语句，通过不断循环即可确定 SQL 语法结构。

综上，目前处理嵌套查询的方法多采用“序列-序列”或“序列-树”深度模型。针对含嵌套的复杂查询，“序列-序列”与“序列-树”模型均需构建大量查询语句训练样本，且“序列-序列”模型不保障语法的正确性。本文方法通过修正算法可保证语法的正确性，且本文模型的处理对象为语句中的实体对关系，每条查询语句可构建多个关系分类训练样本，可以将较少语句样本扩充至较大规模关系分类训练样本，降低了模型对于查询语句训练样本集规模的依赖。

3 基于关系分类与修正的 SQL 语法结构构建

如何根据自然语言查询构建与之对应的 SQL 语法结构是 NL2SQL 的关键，本文提出一种基于关系分类与修正的语法结构构建方法 GSC-RCC，并取得较好效果。

GSC-RCC 方法可分为两个步骤：1）关系分类。设计含注意力机制和组合特征的关系分类模型，模型的输入为查询语句及表格特征，输出该实体对所属不同连接关系的预测概率，直至覆盖语句中所有实体对，并生成由实体和连接关系构成的未修正无向图。2）关系修正。对步骤 1）所获取的无向图加以修正，去除个别实体对关系连接预测错误的情况，并根据修订后的无向图确定 SQL 语法结构。

3.1 相关符号

本文中 SQL 语句的语法结构如图 1 所示。

```
SELECT $AGG $COL
WHERE $COL $OP $VALUE|$WHERE
AND ($COL $OP $VALUE|$WHERE)*
```

图 1 SQL 语法结构

Fig. 1 SQL Syntax Structure

其中，SELECT，WHERE，AND 为 SQL 关键字；符号“$”连接的标记符表示需填充的 SQL 语义成分；AGG，COL，OP 与 VALUE 分别表示聚合函数、列名、运算符和条件值，是查询语句中最基本的语义成分；“VALUE|WHERE”表示条件值 VALUE 或嵌套子句 WHERE，其中嵌套子句 WHERE 符合“$COL $OP $VALUE”形式，且此处的列名 $COL 默认为上层条件语句中的条件列；(…)* 表示 0 或无穷多个条件语句。

给定查询语句 $q=\{e_1,e_2,\cdots,e_n\}$，e_i 为一具体词汇，称作一个实体，n 为词汇个数，$t_i\in\{\text{AGG},\text{COL},\text{OP},\text{VALUE},\text{O}\}$ 表示词汇 e_i 的 SQL 语义类型。这里 AGG，COL，OP，VALUE 的含义如上文所述，O 表示词汇不含任何 SQL 语义。我们的任务目标是：根据自然语言查询，填充图 1 所示的 SQL 语

法结构，其中填充成分以二元组$\langle e_i, t_i \rangle$表示，对查询语句中缺省的语义词汇暂时不填充，以此表示 q 对应的 SQL 语法结构。

以查询语句“莱山区比万象城小区便宜的小区”为例，通过分词及槽填充步骤可以生成如下形式的序列：“〈莱山$_1$，VALUE〉〈区$_2$，COL〉〈比$_3$，O〉〈万象城$_4$，VALUE〉〈小区$_5$，COL〉〈房价$_6$，COL〉〈便宜$_7$，OP〉〈的$_8$，O〉〈小区$_9$，COL〉”，并在此基础上进一步构建如图 2 所示的 SQL 语法结构：

SELECT $AGG <小区$_9$,COL>
WHERE <区$_2$,COL> $OP <莱山$_1$,VALUE>
AND <房价$_6$,COL> <便宜$_7$,OP> (**SELECT** <房价$_6$,COL>
WHERE <小区$_5$,COL> $OP <万象城$_4$,VALUE>)

图 2 SQL 语法结构样例

Fig. 2 SQL Syntax Structure

其中，$ 连接的标识符表示相应实体未在实体集合中出现，“小区$_9$”为目标列，“区$_3$”与“莱山$_1$”表示同一普通条件语句中的列与值，该 SQL 查询有含嵌套子句的条件语句“**SELECT 房价**$_6$ 便宜$_7$(**SELECT** 房价$_6$ **WHERE** 小区$_5$ $OP 万象城$_4$)”。

不妨令 e_i，e_j 为表示 SQL 成分的实体词汇，实体对$\langle e_i, e_j \rangle$的连接关系为 $rela_{i,j} \in \{\text{dc}, \text{nc}, \text{none}\}$，其中 dc，nc 和 none 分别表示直接连接(direct connection)、嵌套连接(nested connection)与无连接(none)，不同的关系连接满足以下规则：

1)令 $c_q \leftarrow e_c e_o e_v$ 表示一条普通条件语句或嵌套子句，语句 c_q 中 3 个实体的语义类型形如〈COL，OP，VALUE〉，则对于 $\forall e_i, e_j \in \{e_c, e_o, e_v\}$ 且 $t_i \neq t_j$，有 $rela_{i,j} = \text{dc}$。

2)令 $s_q \leftarrow e_{sa} e_{sc}$ 表示一条 SELECT 子句，其中实体词汇 e_{sa} 与 e_{sc} 的语义类型 t_{sa} 与 t_{sc} 分别为 AGG 与 COL，则对于 $\forall e_i$，$e_j \in \{e_{sa}, e_{sc}\}$ 且 $t_i \neq t_j$，有 $rela_{i,j} = \text{dc}$。

3)令 $n_q \leftarrow e_c e_o where$ 表示一条含嵌套条件语句，其中实体词汇 e_c 与 e_o 的语义类型与 t_o 分别为 COL 与 OP，则对于 $\forall e_i$，$e_j \in \{e_c, e_o\}$ 且 $t_i \neq t_j$，有 $rela_{i,j} = \text{dc}$；嵌套子句满足 $where \leftarrow e_{wc} e_{wo} e_{wv}$ 形式，其中 $where$ 中的 3 个实体的语义类型形如〈COL，OP，VALUE〉，则对于 $\forall e_i, e_j \in \{e_{wc}, e_{wo}, e_{wv}\}$ 且 $t_i \neq t_j$，有 $rela_{i,j} = \text{dc}$；对于 $\forall e_i, e_j \in \{e_o, e_{wv}\}$ 且 $t_i \neq t_j$，有 $rela_{i,j} = \text{nc}$。

4)除上述 3 种关系类型之外，关系连接 $rela_{i,j} = \text{none}$。

3.2 关系分类模型

GSC-RCC 方法中的关系分类模型是一个结合两种注意力机制及组合特征的深度模型。模型结构如图 3 所示，模型输入包括两部分：1)已标记待预测实体对$\langle e_i, e_j \rangle$位置的查询语句；2)向量化的表格特征，可细分为列名与列值特征。模型输出：实体对$\langle e_i, e_j \rangle$所属不同连接关系的预测概率。在中间层部分模型采用结合词注意力机制的 Bi-GRU 网络提取查询语句的表示向量 $\boldsymbol{v}_q$，并结合列注意力机制与 $\boldsymbol{v}_q$ 分别生成列名向量 $\boldsymbol{v}_{\text{columns}}$ 与列值向量 $\boldsymbol{v}_{\text{cols}}$，将不同特征向量直接拼接即得总的表示向量 $\boldsymbol{v}_{\text{context}}$；输出层的目的是对 $\boldsymbol{v}_{\text{context}}$ 进行解码，通过输出层获得实体对所属不同连接关系的预测概率，损失函数采用负交叉熵损失函数。

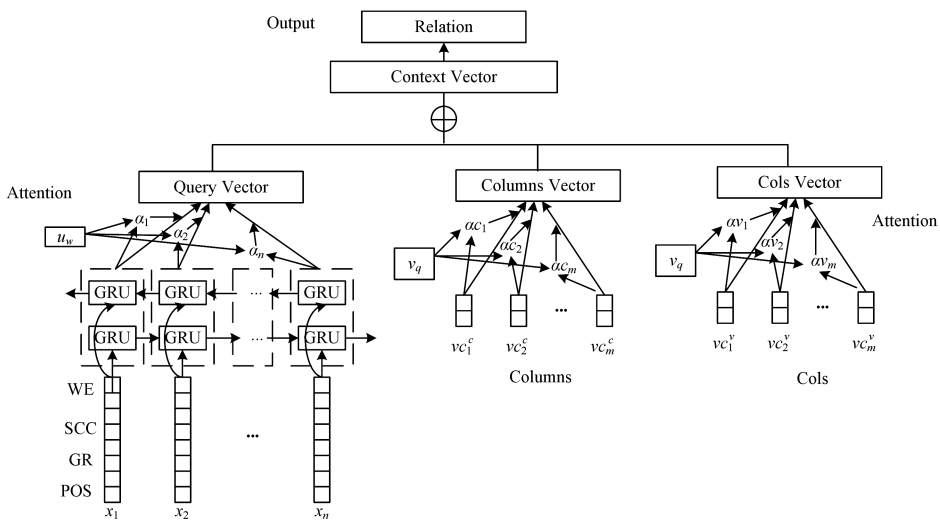


图 3 关系分类模型

Fig. 3 Relational classification model

3.2.1 查询语句向量化

给定一条含 n 个词汇的查询语句 $q = e_1 e_2 \cdots e_n$，选取 4 类特征对每个词汇进行特征表示：自然语言词嵌入(Word Embedding，WE)特征、SQL 成分类别(SQL Component Category，SCC)、语法(Grammar，GR)特征和位置(Position，POS)特征。不同特征的描述如下：

1)对于词嵌入向量特征(WE)，采用预训练的词向量模型[24]，每个词汇的词表征维度为 200 维。

2)对词汇的 SQL 成分类别(SCC)进行标注，SQL 成分类别包括聚合函数(AGG)、列名(COL)、运算符(OP)和条件值(VALUE)，标注例句如“烟台/VALUE 市/COL 绿色家园/VALUE 小区/ COL 的/O 最低/ AGG 房价/ COL”。

3)采用自然语言分析工具 pyltp 对查询语句进行依存句法分析 (Dependency Grammar)获取语法特征(GR)，依存句法分析通过将查询语句转换为解析树，从而确定词汇间的依存关系，如主谓关系 SBV、核心关系 HED、动宾关系 VOB、右附加关系 RAD 等。对于含 n 个词汇的句子 $s = e_1 \cdots e_n$，其依存句法树以 $T = \{(h, m, l): 0 \leqslant h \leqslant n, 1 \leqslant m \leqslant n, l \in L\}$ 表示，其中(f, m, l)表示核心词 e_h 至修饰词 e_f 的关系为 l，L 为依存关系集合。由于依存关系以二元组$\langle e_h, e_f \rangle$形式存在，难以直接作为单个词汇特征，故将关系标记至修饰词，$root$ 代替句子的根实体，可得到语法(GR)特征。如图 4 所示，查询语句“房价/高于/60 万/的/小区”中“房价-高于”为主谓关系(SBV)，于是将词汇 “房价”标注为 SBV。

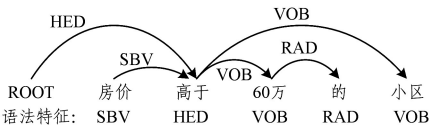


图 4 依存句法树

Fig. 4 Dependency syntactic tree

4)引入特征 $POS=\{POS_1,POS_2\}$ 标记实体对$\langle e_i,e_j\rangle$的位置，如图 5 所示，以 POS_1 为例，POS_1 表示以实体 e_i 为参照的位置特征，将 e_i 的位置标记为 0，e_i 左侧的词汇依次递减标注为$-1,-2,-3$ 等，e_i 右侧的词汇依次递增标注为 1,2,3 等，POS_2 同理。

			e_1		e_2		
	5月	之前	[开盘]	的	[小区]	有	哪些
POS_1	-2	-1	0	1	2	3	4
POS_2	-4	-3	-2	-1	0	1	2

图 5 位置特征

Fig. 5 Location feature

按以上步骤处理，词嵌入特征 WE 采用预先训练的词向量模型，其余特征均采用查找表嵌入方法进行初始向量化表示，不同特征及维度描述如表 1 所列，模型输入的查询语句可以向量矩阵 $\boldsymbol{M}_{\text{query}}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n]\in\mathbb{R}^{n\times m}$ 表示，此处 n 为查询语句中词汇个数，$\boldsymbol{x}_i$ 为词汇 e_i 的表示向量，$m=200+40+40+20+20=320$ 表示词汇不同特征的维度之和。

表 1 特征描述及维度

Table 1 Feature description and dimensions

特征	描述	维度
WE	词嵌入向量	200
SCC	SQL 成分类别	40
GR	语法特征	40
POS	位置特征	20+20=40

3.2.2 表格初始向量化

引入表格中的列名与列值，作为模型的输入特征。由于自然语言的开放性，同一列名可能有多个常用词，如列名“小区均价”的常用词汇包括“房价”和“售价”，为避免这些常用词与其他列名的常用词相混淆，引入列名常用词，以常用词的加权平均向量替代列名向量。针对列名与列值，向量表示方法如下。

1)列名：令 col_i 表示表中第 i 个列名，列名 col_i 的常用词集合为$\{col_{i,1},col_{i,2},\cdots,col_{i,n}\}$其中 $col_{i,j}$ 表示 col_i 的第 j 个常用词，n 为常用词个数。令 $w_{i,j}=c_{i,j}/\sum_j^n c_{i,j}$ 表示常用词 $col_{i,j}$ 的权重，其中 $c_{i,j}$ 为 $col_{i,j}$ 在训练集中的出现次数。令 $\boldsymbol{v}c_{i,j}^c$ 为 $col_{i,j}$ 的表示向量，则列名 col_i 的表示向量 $\boldsymbol{v}\,c_i^c=\sum_j^n wc_{i,j}\cdot\boldsymbol{v}c_{i,j}^c$。

2)列值：考虑到在相同列中，不同值的语义相似性较强，为简化计算，从同一列中选取前 a 个值的表征向量求平均，得到该列的向量表示，令 $\boldsymbol{v}c_i^v=\sum_j^a\frac{1}{a}\cdot\boldsymbol{v}c_{i,j}^v$ 表示第 i 列的列值向量，其中 $\boldsymbol{v}c_{i,j}^v$ 为第 i 列第 j 个值的表示向量，每列选取 $a=5$ 个值。

进行上述处理后，以 $\boldsymbol{M}_{\text{columns}}=[\boldsymbol{v}\,c_1^c,\boldsymbol{v}\,c_2^c,\cdots,\boldsymbol{v}\,c_m^c]\in\mathbb{R}^{m\times d}$ 表示表格中列名的输入表示向量，以 $\boldsymbol{M}_{\text{cols}}=[\boldsymbol{v}\,c_1^v,\boldsymbol{v}c_2^v,\cdots,\boldsymbol{v}c_m^v]\in\mathbb{R}^{m\times d}$ 表示列值的输入表示向量，其中 m 与 d ($d=200$)分别为列数与词向量维度，列名采用的词向量表示模型与上文查询语句中的词汇向量表示模型相同，均为腾讯 *AI LAB* 中文预训练词向量模型[24]。

3.2.3 中间层

接收查询语句、列名与列值的初始向量矩阵 $\boldsymbol{M}_{\text{query}}$，$\boldsymbol{M}_{\text{columns}}$ 和 $\boldsymbol{M}_{\text{cols}}$ 之后，为抽取查询语句中的语义信息，文中采用 Bi-GRU 网络抽取语句的语义特征，GRU 网络增加了循环神经网络(Recurrent Neural Network，RNN)不具备的信息记忆单元，能更好地抽取词汇依赖关系。对于句子中第 t 个字，$\overrightarrow{h}_t$ 和$\overleftarrow{h}_t$ 分别表示该字在最后时刻的正向和反向输出，拼接后得到输出 $h_t=[\overrightarrow{h}_t\oplus\overleftarrow{h}_t]$。

在实际场景中，由于查询语句中不同词汇对于语义的影响不同，引入基于词的注意力机制提取词汇关键信息。无注意力机制的模型对语句“找一下/莱山/区/平均/房价”中不同词汇是平等对待的，引入注意力机制后“房价”比“找一下”在分类过程中影响更大。注意力层的计算过程如式(1)－式(3)所示，最终得到查询语句的表示向量 $\boldsymbol{v}_q$：

$$\boldsymbol{u}_t=\tanh(\boldsymbol{W}_w h_t+\boldsymbol{b}_w) \tag{1}$$

$$a_t=\frac{\exp(\boldsymbol{u}_t^{\mathrm{T}}\boldsymbol{u}_w)}{\sum_t\exp(\boldsymbol{u}_t^{\mathrm{T}}\boldsymbol{u}_w)} \tag{2}$$

$$\boldsymbol{v}_q=\sum_t a_t h_t \tag{3}$$

其中，$\boldsymbol{W}_w$ 与 $\boldsymbol{b}_w$ 分别为可训练的权重矩阵与偏置向量；$\boldsymbol{u}_t$ 表示利用全连接层获取的第 t 个词汇的隐层表达；$\boldsymbol{u}_w$ 为可训练的单个词汇上下文向量，采用随机方式进行初始化；tanh 为激活函数；α_t 为词汇权重；$\boldsymbol{v}_q$ 为生成的查询语句表示向量。

对于列名和列值，考虑到在使用不同的查询语句对表格进行查询时，不同列对预测结果存在不同的影响，故引入注意力机制为不同的列分配不同的权重，帮助抽取特征中的关键信息，经过注意力层处理后得到列名向量 $\boldsymbol{v}_{\text{columns}}$ 和列值向量 $\boldsymbol{v}_{\text{cols}}$，其中 $\boldsymbol{v}_{\text{columns}}$ 的计算过程如式(4)、式(5)所示，$\boldsymbol{v}_{\text{cols}}$ 同理：

$$ac_i=\frac{\exp(\tanh(W[\boldsymbol{w}c_i^c;\boldsymbol{v}_q])+\boldsymbol{b})}{\sum_{i=1}^m\exp(\tanh(W[\boldsymbol{w}c_i^c;\boldsymbol{v}_q])+\boldsymbol{b})} \tag{4}$$

$$\boldsymbol{v}_{\text{columns}}=\sum_{i=1}^m ac_i\boldsymbol{v}c_i^c \tag{5}$$

其中，$\boldsymbol{w}c_i^c$ 是第 i 个列名的输入表示向量，$\boldsymbol{M}$ 与 $\boldsymbol{b}$ 表示可训练的权重矩阵和偏置向量，tanh 为激活函数，ac_i 表示 $\boldsymbol{v}c_i^c$ 与 $\boldsymbol{v}_q$ 之间的相似度权重。同理，列值的表示向量为 $\boldsymbol{v}_{\text{cols}}$，经过向量拼接得到模型总的表示向量 $\boldsymbol{v}_{\text{context}}=[\boldsymbol{v}_q;\boldsymbol{v}_{\text{columns}};\boldsymbol{v}_{\text{cols}}]$。

3.2.4 输出层

经过特征提取，得到模型总的表示向量 $\boldsymbol{v}_{\text{context}}$。再经过 Dropout 层以避免过拟合，输出层采用 Softmax 分类器输出概率向量 $p\in\mathbb{R}^3$，用以表示实体对在 3 类关系连接上的预测概率。模型的损失函数 loss 采用式(6)所示的负交叉熵损失函数，其中 $y\in\mathbb{R}^3$ 是真实关系类别的独热编码表示。

$$loss=-\sum_i^3 y_i\log p_i \tag{6}$$

对语句中不同实体对分别进行预测，根据预测结果可生成未修正无向图 $G=\langle N,S\rangle$，其中 N 为节点集合，每个节点对应一个二元组$\langle e,t\rangle$，e 与 t 为上文中出现的实体词汇和 SQL 语义类别；边集合 S 中每条边对应一个二元组$\langle r,p\rangle$，其中 r 与 p 分别表示连接关系及概率，除必定为无连接关系的实体对之外，每两个节点之间含 $k=3$ 条边，分别对应不同的关系连接类型。

3.3 关系修正

在上一步骤中，关系分类模型对实体对连接类型进行了预测，并且可以根据预测结果生成无向图 $G=\langle N,S\rangle$。因查询语句通常包含多对实体对，需多次关系预测方可覆盖所有实体对，以查询语句“烟台/B-VALUE 市/B-COL 的/O 小区/B-COL”为例，实体对$\langle$市$_2$，小区$_4\rangle$根据本文关系连接描述，语义类别相同的实体之间必为无连接，不必进行预测，故分类模型需对$\langle$烟台$_1$，市$_2\rangle$和$\langle$烟台$_1$，小区$_4\rangle$两个实体对进行关系预测。为避免语句中个别实体对关系被错误预测，我们设计关系修正算法解决该问题，修正算法与具体的查询语句无关。采用上一步骤生成的无向图 $G=\langle N,S\rangle$ 作为修正算法的输入。

首先定义如下约束 C_1 和 C_2。

约束 C_1：给定边 $s=\langle r,p\rangle$，边 s 上的节点为 $n_1=\langle e_1,t_1\rangle$ 与 $n_2=\langle e_2,t_2\rangle$，且 $t_1\neq t_2$，满足下列 4 条规则中任意其中一条规则即视为满足约束 C_1：

1）$r=dc$ 且 $t_1,t_2\in\{\text{COL,OP,VALUE}\}$。

2）$r=dc$ 且 $t_1,t_2\in\{\text{AGG,COL}\}$（边 s 连接聚合函数和相应的目标列）。

3）$r=nc$ 且 $t_1,t_2\in\{\text{OP,VALUE}\}$（边 s 在 1 个含嵌套子句的条件语句中，连接子句中的条件值和父句中的运算符）。

4）$r=$none。

约束 C_2：给定待确认边集合 C，令 N_1 为 C 中所有边上的节点构成的集合，$\forall n=\langle s,t\rangle\in N_1$，同时满足下列所有规则即视为满足约束 C_2：

1）令 $s=\langle r,p\rangle\in C$ 为连接 n 与 $n_1=\langle e_1,t_1\rangle$ 且满足 $r\neq$ none 与 $t_1,t_2\in\{\text{COL,OP,VALUE}\}$ 的边，当 r,t_1,t_2 确定时，s 至多存在一条（在同一条件语句中，确定位置的某一实体词汇仅表示 SQL 语法结构中的唯一语法成分）。

2）令 $s=\langle r,p\rangle\in C$ 为连接 n 与 $n_1=\langle e_1,t_1\rangle$ 且满足 $r=dc$ 与 $t_1,t_2\in\{\text{AGG,COL}\}$ 的边，当 t_1,t_2 确定时，s 至多存在一条（同一目标列至多和 1 个聚合函数相搭配，反之同理）。

其中，约束 C_1 基于本文关系连接的表述，并结合实体的 SQL 语义类型，以判断实体对之间是否可能存在待定的关系连接类型；约束 C_2 基于本文关系连接和 SQL 语法结构的表述，将已确定的边和边上节点视为整体，判断其是否能正确生成 SQL 语法结构。在约束 C_1 和 C_2 的基础上，构建如算法 1 所示的关系修正算法，修正算法可以生成表示 SQL 语法结构的无向图，无向图中每两个节点之间只有唯一确定的边。

算法 1 结合 SQL 语法的关系修正算法

输入：表示关系预测结果的无向图 G=〈N,S〉，其中 N 为节点集合，S 为边集合

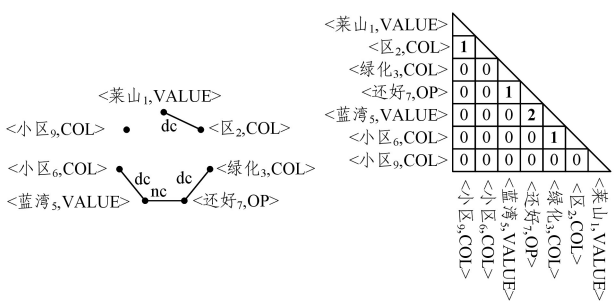
输出：经过修正之后表示 SQL 语法结构的无向图 G_m

初始化：待确认边集合 C=∅

算法过程：

1. while ∃ s=〈r,p〉∈S do
2. 选择 S 中关系概率 p 最大的边 s=〈r,p〉，如 s 存在多条，则任意选取一条；/* 根据关系概率排序选择边 */
3. 将边 s 加入集合 C；/* 暂定 s 为正确连接边 */
4. if（边 s 与连接 s 的节点 n_1 与 n_2 满足约束 C_1）and（C 满足约束 C_2）then /* 约束 C_1 判断实体对之间是否可能存在待定关系连接类型，约束 C_2 判断已确定的边和节点是否能正确生成 SQL 语法结构 */
5. 从 S 中删除所有连接节点 n_1 与 n_2 的边；/* 确定连接节点 n_1 与 n_2 之间的边为 s */
6. else 从 S 与 C 中删除边 s；/* 确定节点 n_1 与 n_2 之间的边不为 s */
7. end while
8. 返回由 C 构成的无向图 G_m.

以“莱山/B-VALUE 区/B-COL 绿化/B-COL 比/O 蓝湾/B-VALUE 小区/B-COL 还好/B-OP 的/O 小区/B-COL”为例，经过关系分类及关系修正步骤，可得到表示 SQL 语法结构的无向图和最终的 SQL 语法结构，如图 6 所示。图 6(a)给出经过修正返回的无向图，为便于观察图中隐藏无连接关系，图 6(b)给出无向图的邻接矩阵，其中数字 0，1 和 2 分别表示无连接、直接连接与嵌套连接；图 6(c)为最终得到的 SQL 语法结构。由已修正的无向图可知，条件值“莱山$_1$”与列名“区$_2$”的连接关系为直接连接；列名“绿化$_3$”和运算符“还好$_7$”的连接关系为直接连接；“还好$_7$”与“蓝湾$_5$”为嵌套连接关系；“小区$_6$”与“蓝湾$_5$”为直接连接关系；其余实体对之间均为无连接关系。



(a)SQL 语法结构的无向图　　(b)SQL 语法结构的邻接矩阵

SELECT $AGG <小区$_9$,COL>
WHERE <区$_2$,COL> $OP <莱山$_1$,VALUE>
AND <绿化$_3$,COL> <还好$_7$,OP> (**SELECT** <绿化$_3$,COL>
WHERE <小区6,COL> $OP <蓝湾5,VALUE>)

(c)生成的 SQL 语法结构

图 6 SQL 语法结构的不同形式表示

Fig. 6 Different forms of SQL syntax structure

确定经修正的无向图之后，根据上文中关系连接的定义可确定最终的 SQL 语法结构，在句中未体现聚合函数且列名“小区$_9$”与其他实体均为无连接关系的情况下，可知“小区$_9$”表示目标列；列名“区$_2$”和条件值“莱山$_1$”为直接连接关系，可知“区$_2$”与“莱山$_1$”同属一个条件子句，且句中无运算符实体与“莱山$_1$”为嵌套连接关系，从而可知这两个实体并非属于条件语句中的嵌套子句，且句中未出现与“莱山$_1$”或“区$_2$”为直接连接关系的运算符实体，可判断这两个实体所属的条件语句不含嵌套结构。综上可知，“莱山$_1$”和“区$_2$”同属一个无嵌套条件语句；列名“绿化$_3$”和运算符“还好$_7$”的连接关系为直接连接；“绿化$_3$”与“还好$_7$”与同属一个条件子句，列名“小区$_6$”与条件值“蓝湾$_5$”为直接连接关系；“小区$_6$”与“蓝湾$_5$”同属一个条件子句，且运算符“还好$_7$”与条件值“蓝湾$_5$”为嵌套连接关系；条件值“蓝湾$_5$”所在的条件子句为运算符“还 好$_7$”修饰的嵌套子句，该条件语句中的嵌套子句为“**SELECT** 小区$_6$ **WHERE** 小区$_6$ ≠OP 蓝湾$_5$”。根据上述推断，即可确定最终图 6(c)所示的 SQL 语法结构。

4 实验验证

4.1 数据描述

本文数据集来源于二手房购买过程中的常见问题，采用爬虫技术，抓取房屋交易网站安居客上关于烟台、青岛等地的二手房数据信息，查询包括房屋单价、总价、小区均价和户型等 31 种提问方式。手动构建并标记 1 891 条查询语句，其中单条件无嵌套查询语句 326 条，单条件嵌套查询语句 365 条，多条件无嵌套查询语句 455 条，多条件嵌套查询语句 745 条。根据上文中关系连接的描述，部分实体对之间由于实体语义类别的原因，连接关系必定为无连接，如表示聚合函数与运算符的实体词汇之间必定为无连接关系，此类样本不参与训练，剔除必为无连接关系的实体对后，获得 30 152 条关系分类训练样本，其中同一查询语句构建的关系分类样本同时归入训练集或同时归入测试集。由于无连接及直接连接实体对样本数量较多，在对这两类样本下采样后，得到由 11 109 条关系分类样本构成的数据集，其中数据集的样本分布情况如表 2 所列。

表 2 关系分类数据集分布情况

Table 2 Distribution of relational classification data sets

类别	直接连接	嵌套连接	无连接
训练样本	3996	903	4008
测试样本	1001	203	998

4.2 超参数设置及影响

文中关系分类联合深度模型的优化策略为 Adam 自适应矩估计优化策略，模型的部分超参数设置如表 3 所列。

表 3 超参数设置

Table 3 Hyperparameter setting

参数	描述	值
nbatch	每批次样本数	128
Epoch	迭代次数	200
units	GRU 隐藏层单元个数	250
d	Dropout 层丢失率	0.5
lr	学习率	0.0001

文中分别研究 GRU 隐藏层单元个数 $units \in \{50,100,\cdots,500\}$，Dropout 层丢失率 $d \in \{0.1,0.15,\cdots,0.8\}$ 对于评价指标 F1 的影响，其中实验过程重复 10 次，实验结果分别如图 7 与图 8 所示。由图 7 可知，Dropout 丢失率 d 在 0.5 时模型达到最优，这主要是因为当 d 在 0.5 左右时，Dropout 随机生成的网络结构最多。GRU 隐藏层单元数在 250 时达到最优。同时从表中可以看出 Dropout 丢失率和 GRU 隐藏层单元数对结果的影响是有限的，这主要是由于样本数据集有限，神经网络更加容易拟合。

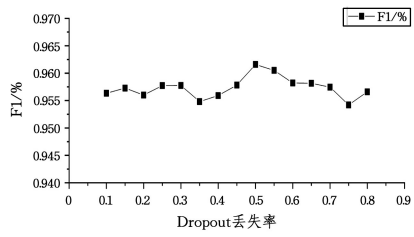


图 7 Dropout 丢失率的影响

Fig. 7 Influence of Dropout loss rate

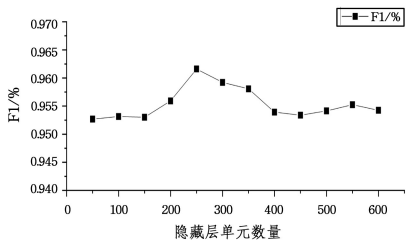


图 8 隐藏层单元数量的影响

Fig. 8 Influence of number of hidden layer units

4.3 不同特征选择标准特性分析

为验证不同特征选择方案对于关系分类结果的影响，选取不同的特征组合方式进行对比实验。词汇的备选特征包括词汇的嵌入向量(WE)、SQL 成分类别(SCC)、语法特征(GR)与位置特征(POS)。表格的备选特征包括以列名常用词表示的列名特征，以 C 表示；直接以列名表示的列名特征，以 CN 表示；列值特征，以 V 表示。评价指标选取查准率(Precision，P)、召回率(Recall，R)和 F1 值，实验重复 10 次。实验结果如表 4 所列。

表 4 关系分类实验中不同特征选择的对比实验结果

Table 4 Comparison of experimental results of different feature selection in relation classification experiment

(单位：%)

方法	直接连接			嵌套连接			无连接		
	P	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1
WE+POS	94.28	94.35	94.33	92.89	92.42	92.65	94.42	94.34	94.38
WE+POS+SCC	94.90	94.99	94.97	93.44	93.75	93.59	94.69	94.71	94.70
WE+POS +SCC+GR	95.10	94.89	94.94	93.50	93.87	93.68	94.86	94.70	94.78
WE+POS+SCC+GR+CN	96.13	96.01	96.07	95.43	94.71	95.07	94.38	95.50	94.93
WE+POS+SCC+GR+C	96.59	96.12	96.24	95.63	94.99	95.31	94.42	95.54	94.98
WE+POS+SCC+GR+CN+V	96.30	**96.37**	96.33	96.18	95.03	95.60	95.35	95.15	95.25
WE+POS+SCC+GR+C+V	**96.72**	96.35	**96.44**	**96.22**	**95.17**	**95.69**	**96.41**	**96.28**	**96.34**

从表 4 可以发现，基于列名常用词构建的列名特征与直接以列名表示的列名特征相比，在多个实验指标中有着更好的表现，这是由于基于列名常用词构建列名特征考虑了语义表达的开放性。从表 4 还可以看出，随着选取特征的增多，模型的分类性能有更好的表现，结合文中所有特征的模型实验效果最佳，这是由于不同的特征从不同角度反映了查询语句的隐藏信息，而深度模型将这些隐藏信息抽取后对于语义信息有更好的理解。

4.4 不同关系分类方法的对比实验结果

为验证关系分类模型的性能，本文采用支持向量机(Support Vector Machine，SVM)[25]、Multi-Level Attention CNN[26]、Attention CNN[27]等 3 种方法与 GSC-RCC 模型进

行对比实验，评价指标选取查准率 P、召回率 R 和 F1 值，实验重复 10 次，对比实验结果如表 5 所列。可以看出，基于神经网络模型的研究方法较 SVM 有着更好的分类性能。在所有的深度模型中，本文方法在 3 项评价指标中大部分处于最佳水平，其原因主要在于本文方法考虑了更多的词汇及表格特征。

表 5 不同关系分类方法的对比实验结果

Table 5 Comparative experimental results of different methods of relationship classification

（单位：%）

方法	直接连接			嵌套连接			无连接		
	P	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*
SVM	75.56	72.99	74.27	70.25	75.48	77.01	76.24	78.48	79.99
Multi-Level Attention CNN	95.75	95.90	95.82	95.34	94.37	94.85	94.59	95.99	95.28
Attention CNNs	95.49	95.51	95.50	95.17	**95.20**	95.19	94.89	95.20	95.04
本文关系分类模型	**96.72**	**96.35**	**96.44**	**96.22**	95.17	**95.69**	**96.41**	**96.28**	**96.34**

4.5 实验结果对比

为验证本文方法与其他方法对于不同类型查询语句 SQL 语法结构的预测效果，采用 Attention CNNs 与文中修正步骤相结合的方法（简称“Attention CNNs＋修正”）、Pointing Network[27] 与本文方法进行比较。“Attention CNNs＋修正”方法与 GSC-RCC 均为基于关系分类预测 SQL 语法结构的方法；Pointing Network 为“序列-序列”模型，此处用于预测 SQL 语法结构，输入序列为查询语句与关键字的联合序列，此处关键字集合为{**SELECT**, **WHERE**, **AND**, [,]}，输出为表示 SQL 语法结构的序列，形式上以“**SELECT** $AGG $COL **WHERE** $COL $OP [$COL $OP $VALUE] | $VALUE (**AND** $COL $OP [$COL $OP $VALUE] | $VALUE) *”表示。其中，$ 连接的标识符表示查询语句中的实体词汇，[$COL $OP $VALUE] | $VALUE 表示嵌套子句或者条件值，(…) * 表示 0 至无穷项条件语句，“(”“)”“|”“ * ”仅为更好地描述输出形式，在实际输出序列中不出现，当查询语句中语义成分出现缺省时，缺省成分不进行预测。

为验证本文方法是否可以减少模型对查询语句样本数的依赖，在测试集查询语句样本数相同的情况下，每种方法均采用 50%，100% 两种不同规模的查询语句训练样本数，数字表示查询语句训练样本数占正常训练过程中训练样本数的百分比。结果采准确率（Accuracy）作为评价指标，实验重复 10 次，实验结果如表 6 所列。

表 6 SQL 语法结构预测准确率的对比结果

Table 6 Comparison of results of SQL syntax structure prediction accuracy

（单位：%）

参数	Attention CNNs＋修正		Pointing Network		GSC-RCC	
	50%	100%	50%	100%	50%	100%
单条件无嵌套查询语句	93.20	94.54	92.46	93.94	**94.35**	95.85
单条件含嵌套查询语句	90.71	92.20	87.85	90.45	**91.74**	**93.25**
多条件无嵌套查询语句	90.75	92.99	88.58	91.47	**91.97**	**93.58**
多条件含嵌套查询语句	87.97	90.87	85.01	89.28	**89.61**	**92.25**

可以看出，在测试集中本文关系分类模型对于 4 类查询语句都有着较好的预测效果，但复杂查询语句的预测准确率略低于简单查询语句。在两种训练集规模下，本文方法总体均为最优，当训练集规模从 100% 降低至 50% 时，“Attention CNNs＋修正”方法、Pointing Network 和 GSC-RCC 的平均准确率分别下降了 2.16%，3.10% 和 1.94%，其中基于关系分类的“Attention CNNs＋修正”方法与 GSC-RCC 的准确率下降程度更低，说明本文方法可以有效减少训练深度模型所需查询语句的样本数。

4.6 错误结果分析

由测试集可以得知，预测错误的数据主要集中于含嵌套与多条件语句中，为进一步分析错误的可能原因，我们对部分关系分类错误数据进行分析，部分错误数据的分析结果如表 7 所列。在第一条样例“系统/O 找找/O 比/O 万象城/B-VALUE 这个/O 小区/B-COL 要/O 早/B-OP 建造/B-COL 的/O 小区/B-COL”中，“小区”前后出现 2 次，且与“早”分别为直接连接与无连接，但预测的关系类别均为直接连接，预测概率分别为 0.985 与 0.763。其原因是 2 个“小区”语义相似，且与“早”的位置距离也相近，可能对模型造成错误干扰，由于同一条件值不能与多个列名同时为直接连接，故以文中的修正算法进行修正，修正后连接关系分别为直接连接与无连接，与真值相符。在第二条样例“哪些/O 小区/B-COL 不低于/B-OP 8 千/B-VALUE 不高于/B-OP 1.2 万/B-VALUE”中，“小区”与“不低于”之间为无连接关系，但被错误预测为直接连接，预测概率为 0.653。其原因是条件语句中列名“房价”缺失，且缺失列名的语句在样本集中较少，模型没有完全拟合这类样本，应对数据集进行进一步扩充，并将值的数据类型考虑至约束策略中。

表 7 关系分类实验中的错误样例

Table 7 Examples of errors in relation classification experiments

查询语句	真值	预测值	预测概率	是否被修正
比/万象城/要/小区/要/[早]/建造/的/[小区]	无连接	直接连接	0.763	是
哪些/[小区]/[不低于]/8 千/不高于/1.2 万	无连接	普通连接	0.653	否

结束语 文中针对限定领域内（文中为房产查询领域）含嵌套查询中 SQL 语法结构难以确定的问题，提出基于关系分类思想的 GSC-RCC 方法。该方法核心是以 3 类实体间关系表示 SQL 语法结构中不同成分的语义关联，通过选取合适特征，如引入列名常用词以改善语义多样性，并加入注意力机制改善网络模型的效果，并以结合 SQL 语法的修正算法对预测结果进行修正，最终生成与自然语言查询相匹配的 SQL 语法结构。实验结果表明，GSC-RCC 可有效预测含嵌套复杂查询的语法结构，对多条件含嵌套复杂自然语言查询的预测准确率为 92.25%。下一步的工作是根据关系分类的错误样例对

系统进行改善，将值的数据类型考虑至约束策略中，并对 SQL 成分缺失的情况进行研究，以改善系统在复杂查询中的适应性。除此之外，还将针对不同领域及开放查询领域开展算法研究工作。

参 考 文 献

[1] CHUNG J，GULCEHRE C，CHO K H，et al. Empirical evaluation of gated recurrent neural networks on sequence modeling [J]. arXiv：1412.3555，2014.

[2] POPESCU A M，ARMANASU A，ETZIONI O，et al. Modern natural language interfaces to databases：Composing statistical parsing with semantic tractability [C]// Proc of the 20th Int Conf on Computational Linguistics. Stroudsburg，PA：Association for Computational Linguistics，2004：141-147.

[3] LI Y，YANG H，JAGADISH H V. Constructing a Generic Natural Language Interface for an XML Database [M]. Berlin，Heidelberg：Springer Berlin Heidelberg，2006：737-754.

[4] ALESSANDRA G，MOSCHITTI A. Translating Questions to SQL Queries with Generative Parsers Discriminatively Reranked [C]// Proc. of COLING. New York：ACM，2012：401-410.

[5] POON H. Grounded Unsupervised Semantic Parsing [C]// Proc of the 51st Annual Meeting of the Association for Computational Linguistics. Stroudsburg，PA：ACL，2013.

[6] LI H，XU J. Semantic Matching in Search [J]. Foundations and Trends in Information Retrieval，2014，7(5)：343-469.

[7] SERBAN I，SORDONI A，BENGIO Y，et al. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models [C]// Proc of the 30th AAAI Conf on Artificial Intelligence. Menlo Park，CA：AAAI Press，2016.

[8] DOLAN W B，BROCKETT C. Automatically Constructing a Corpus of Sentential Paraphrases [C]// Proc of the Third Int Workshop on Paraphrasing(IWP). 2005.

[9] BROWN P F，DELLA PIETRA S A，DELLA PIETRA V J，et al. The Mathematics of Statistical Machine Translation：Parameter Estimation [J]. Computational Linguistics，1993，19(2)：263.

[10] VINYALS O，KAISER L，KOO T K，et al. Grammar as a Foreign Language [C]// Proc of Advances in Neural Information Processing Systems. New York，NY：Curran Associates，2015：2773-2781.

[11] ZHONG V，XIONG C，SOCHER R. Seq2SQL：Generating Structured Queries from Natural Language using Reinforcement Learning [J]. arXiv：1709.00103，2017.

[12] DONG L，LAPATA M. Coarse-to-Fine Decoding for Neural Semantic Parsing [J]. arXiv：1805.04793，2018.

[13] CAI R C，XU B Y，ZHANG Z J，et al. An Encoder-Decoder Framework Translating Natural Language to Database Queries [J]. arXiv：1711.06061，2017.

[14] XU X J，LIU C，DAWN S. SQLNet：Generating Structured Queries From Natural Language Without Reinforcement Learning [J]. arXiv：1711.04436，2017.

[15] YU T，LI Z F，ZHANG Z L，et al. TypeSQL：Knowledge-based Type-Aware Neural Text-to-SQL Generation [J]. arXiv：1804.09769，2018.

[16] HWANG W，YIM J，PARK S，et al. A Comprehensive Exploration on WikiSQL with Table-Aware Word Contextualization [J]. arXiv：1902.01069，2019.

[17] EPSTEIN，SAMUEL S. Transportable Natural Language Processing through Simplicity-The PRE System [J]. Acm Transactions on Information Systems，1985，3(2)：107-120.

[18] IYER S，KONSTAS I，CHEUNG A，et al. Learning a Neural Semantic Parser from User Feedback [J]. arXiv：1704.08760，2017.

[19] FINEGAN-DOLLAK C，KUMMERFELD J K，ZHANG L，et al. Improving Text-to-SQL Evaluation Methodology [J]. arXiv：1806.09029，2018.

[20] LEE D，YOON J，SONG J，et al. One-Shot Learning for Text-to-SQL Generation [J]. arXiv：1905.11499，2019.

[21] VINYALS O，FORTUNATO M，JAITLY N. Pointer Networks [C]// Proc of Advances in Neural Information Processing Systems. New York，NY：Curran Associates，2015：2692-2700.

[22] LEE D. Recursive and Clause-Wise Decoding for Complex and Cross-Domain Text-to-SQL Generation [J]. arXiv：1904.08835，2019.

[23] SONG Y，SHI S，LI J，et al. Directional Skip-Gram：Explicitly Distinguishing Left and Right Context for Word Embeddings [C]// Proc of the 2018 Conf of the North American Chapter of the Association for Computational Linguistics：Human Language Technologies，Stroudsburg，PA：ACL，2018：175-180.

[24] LIN Z H，FENG M W，SANTOS C N D，et al. A Structured Self-attentive Sentence Embedding [J]. arXiv：1703.03130，2017.

[25] CORTES C，VAPNIK V. Support-Vector Networks [J]. Machine Learning，1995，20(3)：273-297.

[26] ZENG D J，LIU K，LAI S W，et al. Relation Classification via Convolutional Deep Neural Network [C]// Proc. of COLING. New York：ACM，2014：2335-2344.

[27] ZHU J，QIAO J，DAI X，et al. Relation Classification via Target-Concentrated Attention CNNs [C]// Int. Conf. on Neural Information Processing. Berlin：Springer，2017：137-146.

WAN Wen-jun，born in 1996，postgraduate. His main research interests include natural language processing and deep learning.

DOU Quan-sheng，born in 1971，Ph.D，professor，is a member of China Computer Federation. His main research interests include natural language processing and deep learning.