文章编号：2095-2163(2022)07-0001-07 中图分类号：TP391.1 文献标志码：A

基于知识增强的 NL2SQL 方法

王秋月，程路易，徐 波，王志军
（东华大学 计算机科学与技术学院，上海 201620）

摘 要：面向关系型数据库的自然语言查询技术的核心是将自然语言解析成 SQL 查询语句（NL2SQL）。目前，大多数 NL2SQL 方法仅对自然语言问句和表模式进行编码，难以充分理解问句的语义信息，产生的歧义可能导致预测出错。针对此问题，本文提出了基于知识增强的 NL2SQL 模型 KESQL，首先使用实体链接技术将问句中的实体链接到外部知识图谱，通过引入问句中命名实体在外部知识图谱的知识来增强 NL2SQL 模型对于问句的理解能力，进而提高解析效果；选取 DBpedia 作为外部知识图谱，针对图谱中的各类知识，提出了基于符号化和向量化的知识增强方案，系统地论证了引入不同知识的效果及不同融合方式的优劣，实验结果充分验证了知识增强对 NL2SQL 任务的有效性。

关键词：NL2SQL；实体链接；知识增强

NL2SQL method based on knowledge enhancement

WANG Qiuyue, CHENG Luyi, XU Bo, WANG Zhijun
(School of Computer Science and Technology, Donghua University, Shanghai 201620, China)

【Abstract】The core of natural language query technology for relational database is to parse natural language into SQL query statements (NL2SQL). Currently, most NL2SQL methods only encode natural language utterance with table schema, which makes it difficult to fully understand the semantic information of the utterance and may lead to prediction errors due to ambiguity. This paper proposes an NL2SQL model KESQL (NL2SQL with Knowledge Enhancement) based on knowledge enhancement. Firstly, the entity link technology is used to link the entities in questions to external knowledge graph. By introducing the knowledge of named entities in questions in external knowledge graph, NL2SQL model can enhance the understanding ability of the utterance and then improve the parsing effect. In this paper, DBpedia is selected as the external knowledge graph. We propose two knowledge enhancement schemes based on symbolization and vectorization for all kinds of knowledge in the graph. We systematically demonstrate the effects of introducing different knowledge and the advantages and disadvantages of different fusion methods. Experiments fully verify the effectiveness of knowledge enhancement for NL2SQL tasks.

【Key words】NL2SQL; entity linking; knowledge enhancement

0 引 言

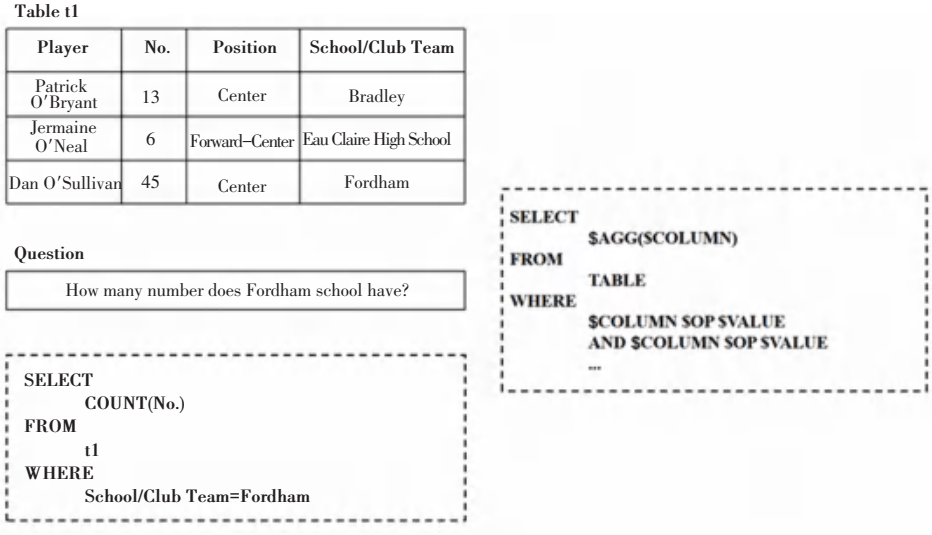
关系型数据库是信息系统的基础和核心。用户可以用 SQL 查询来检索数据库中的数据，但这通常对用户的 SQL 掌握水平有一定要求。而通过自然语言直接与数据库交互可以帮助非技术用户获取到关系型数据库中的信息，提高用户的使用效率和体验。因此，本文研究的任务是将自然语言问句转化为 SQL 查询（NL2SQL）。目前解决此任务的主流方法是基于草图的模型，其考虑了 SQL 的句法模式，通过 SQL 语句中的关键词如“SELECT”、“FROM”、“WHERE”等将原任务拆解为多个子任务。草图框架具体实例如图 1 所示，以问句“How many number does Fordham school have?”为例，其对应的表包含列名 Player（选手）、No.（编号）、Position（位置）、School/Club Team（学校/俱乐部球队）等。按照草图框架，需要完成 6 个子任务的预测，根据模板进行槽填充构建完整的 SQL 语句。第一，需要预测 SELECT 从句中出现的列（SELECT-COLUMN 子任务），此例中预测结果“No.”；第二，需要预测 SELECT 从句中出现的列对应的聚合操作（SELECT-AGGREGATION 子任务），此例中预测结果为“COUNT”；第三，需要预测 WHERE 从句中条件的数量（WHERE-NUMBER 子任务），此例中预测结果为 1 个；第四，需要预测 WHERE 从句中出现的列（WHERE-COLUMN 子任务），此例中预测结果为

作者简介：王秋月（1998-），女，硕士研究生，主要研究方向：自然语言处理——NL2SQL；程路易（1993-），男，硕士研究生，主要研究方向：自然语言处理——对话系统；徐 波（1988-），男，博士，讲师，硕士生导师，主要研究方向：知识图谱、自然语言处理、人工智能等；王志军（1973-），男，博士，副教授，硕士生导师，主要研究方向：物联网信息服务、数据库、信息检索。
通讯作者：王志军 Email：Zjwang@dhu.edu.cn

收稿日期：2022-01-05

"School/Club Team";第五,需要预测 WHERE 从句中条件列对应的操作(WHERE-OPERATOR 子任务),此例中预测结果为"=";第六,需要预测 WHERE 从句中条件列对应的条件值(WHERE-VALUE 子任务),此例中预测结果为"Fordham"。

虽然目前 NL2SQL 任务上提出的方法达到了比较好的效果,但仍然不足,这些方法都是对问句进行直接编码,缺乏语义信息,不能充分地理解问句。



(a) 草图样例 (b) SQL 草图框架

图 1 基于草图的方法

Fig. 1 Sketch-based approach

针对自然语言问句存在语义缺失的问题,本文考虑利用外部知识图谱来对自然语言问句进行语义增强,使其包括充分的语义信息。基于知识增强的 NL2SQL 方法主要面临 3 个挑战:

(1)对问句的哪些部分进行增强;

(2)用外部知识图谱中的哪类知识进行增强;

(3)如何进行增强。

针对第一个挑战,本文提出对问句中出现的命名实体进行增强,并使用现有的实体链接技术,将问句中的命名实体链接到外部知识图谱中;针对第二个挑战,本文将知识图谱中的知识类别分为摘要(Abstract)、类型(Type)、标签(Category)、语义关系(Infobox)4 种,并系统调研了各种类型知识的增强效果;针对第三个挑战,本文分别提出了一种基于符号化知识的增强方法和两种基于向量化知识的增强方法。在公开的大规模的 NL2SQL 数据集 WiKiSQL 上进行实验,实验结果证明了本文提出的增强方法的有效性。

1 相关工作

NL2SQL 的方法主要分为两大类。

第一类是基于 Seq2seq 模型,采用"编码器-解码器"将此任务转化为从文本到 SQL 的翻译任务,代表方法是 Seq2SQL,将 SQL 语句划分为 SELECT 从句和 WHERE 从句两部分,并分开独立生成,这类方法存在两种缺陷:第一种是 WHERE 从句中可能包含多个条件三元组,多个条件三元组之间的顺序并不影响最终的执行结果,但会极大地影响以之前的标记预测下一个标记的方式进行预测的 Seq2seq 模型的性能,Seq2SQL 使用强化学习来消除顺序问题,但准确率依然不高;第二种缺陷是 Seq2seq 模型没有充分利用 SQL 句法结构来限制输出空间,模型复杂且准确率不高。

第二类方法是基于草图的方法。文献[1]基于此想法提出了 SQLNet 模型,根据 SQL 语句的句法结构将 SQL 查询分解为 6 个子任务。预定义草图包含各个子任务的依赖关系,每个子任务的预测只基于其所依赖的部分。与 Seq2SQL 不同,SQLNet 采用了顺序到集合的方法和基于列字段的注意力机制,消除了顺序问题,提高了准确率。在此基础上,文献[2]提出执行指导编码(Execution-Guided Decoding),可以理解为一种后验操作,假设生成的 SQL 查询可以执行,通过执行结果来排除错误的候选 SQL 查询。随着动态表示学习的发展,更多的方法选择预训练语言模型作为编码器。文献[3]提出的 SQLova 使用表感知的 BERT 作为编码器,在编码后提出了 3 种不同的解码器变体,其中一种变体类似 SQLNet,3 种解码器之间的精度差也论证了预训

练语言模型的有效性；文献[4]提出 X-SQL，使用 MT-DNN 作为编码器，将全局上下文信息融合到表模式中，为下游任务提供更好的表达，显著地提高了性能。不同于连接问句和表中所有列名作为输入的 SQLova 和 X-SQL，文献[5]提出的 HydraNet 将问句和表中的各个列名单独拼接送入编码器，不需要额外的池化操作或长短期记忆网络来获得一个列的向量表示，可以更好地获得列的表示。

2　方法

2.1　任务定义

基于知识增强的 NL2SQL 任务定义：给定一个数据库，还有一个外部知识图谱，包括实体的各类知识，如：摘要、类型、标签和语义关系等，如图 2 所示。其中，摘要是对实体的描述，类型说明了实体的所属类型，标签简要介绍了实体的特征，而语义关系包含了实体的属性及属性值。在不使用表中字段值的前提下，输入一个自然语言问句，输出对应的 SQL 语句。

形式化表示为：给定一个包含多张来自不同领域的表的数据库 $\Phi = \{t_i\}_{i=1}^{m}$，和一个知识图谱 $\Psi = \{(e_j, K_{a_j}, K_{t_j}, K_{c_j}, K_{i_j})\}_{j=1}^{n}$，其中，$e_j$ 表示知识图谱中的第 j 个实体，$K_{a_j}, K_{t_j}, K_{c_j}, K_{i_j}$ 分别表示第 j 个实体的摘要、类型、标签、语义关系。基于知识增强的 NL2SQL 任务的目标是：给定一个自然语言问句 $q = \{x_1, x_2, \cdots, x_n\}$ 和数据库中查询的表 t_q，其包含候选列 $C_{t_q} = \{c_1, c_2, \cdots, c_n\}$。通过实体链接技术从知识图谱 Ψ 中找到问句 q 中出现的实体集合 E_q，结合实体集合 E_q 中每个实体的知识，生成该问句 q 对应的 SQL 查询语句 s。

2.2　系统框架

本文工作的核心思想是利用知识来增强问句的语义信息，实现问句到 SQL 语句的转化。本文提出基于知识增强的 NL2SQL 模型 KESQL，模型结构如图 3 所示。

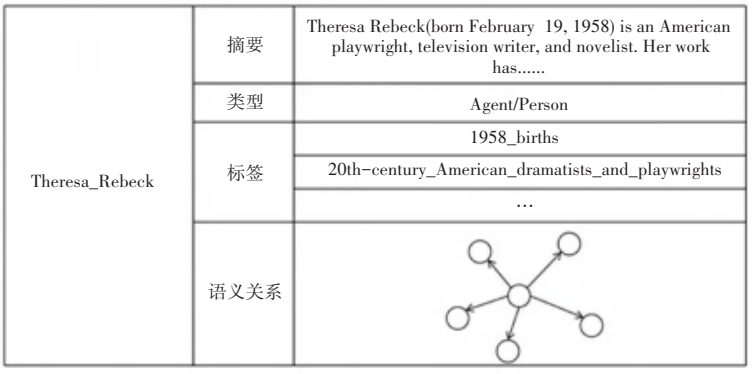


图 2　实体的相关知识

Fig. 2　Related knowledge of entity

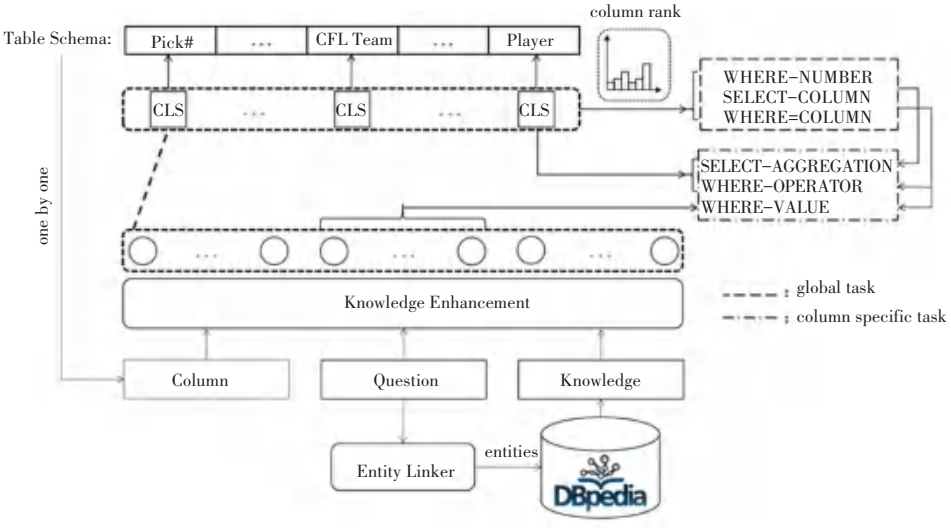


图 3　KESQL 模型结构图

Fig. 3　Structure diagram of KESQL model

本文的系统框架主要包含 3 个部分:首先对问句进行实体链接,找到问句中出现的命名实体,将其链接到知识图谱中,以获得这些命名实体的更多语义信息;其次,在知识增强模块从符号化和向量化两个角度实现增强,将问句与知识融合对齐,使模型更充分的理解问句;最后,利用知识增强后的编码层输出来解码草图结构中的各个子任务。

目前已经有很多成熟的实体链接工具,本文选择了主流的实体链接工具 DBpedia Spotlight,将自然语言文本中的命名实体链接到知识图谱 DBpedia 中。

3 模型

3.1 输入模块

给定一个问句 q,问句对应的实体的某类知识 K 以及相应的表 t_q,对表中的每个候选列 c_i,考虑其字段类型信息 $type_{ci}$,组成输入序列($Concat(type_{c_i}, c_i), q, K$)。其中,$Concat(\cdot)$ 表示一个将多个句子连接成一个字符串的函数。

3.2 知识增强模块

本文从符号化和向量化两个角度进行知识增强,包含一种符号化增强方法和两种向量化增强方法。

3.2.1 符号化知识的增强方法

符号化知识的增强是指将实体知识表示为一个字符串,直接与自然语言问句拼接,再进行后续的编码等操作。本文将结构化和非结构化的实体知识统一转化为自然语言作为实体的符号化知识描述,见表 1。

表 1 实体符号化知识描述的生成

Tab. 1 The generation of entity symbolic knowledge description

知识类别	原始形式	符号化知识
摘要	sentence1.sentence2…	entity:sentence1.
类型	$\{t_1, t_2, \cdots, t_n\}$	entity:t_1 *and* t_2…
标签	$\{c_1, c_2, \cdots, c_n\}$	entity:c_1 *and* c_2…
语义关系	$\{[k_1, v_1], [k_2, v_2], \cdots, [k_n, v_n]\}$	entity:k_1 *is* v_1 *and* k_2 *is* v_2…

直接拼接问句和实体符号化知识描述,作为编码器的输入,输入序列为 $[CLS]column[SEP]question$, $symbolic\ knowledge[SEP]$, 其中 $column$,$question$, $symbolic\ knowledge$ 是列信息 $Concat(type_{c_i}, c_i)$、问句、实体符号化知识分词后的形式,本文将此方法称为符号化知识增强(Knowledge Enhancement with Symbolic Knowledge),如图 4 所示。

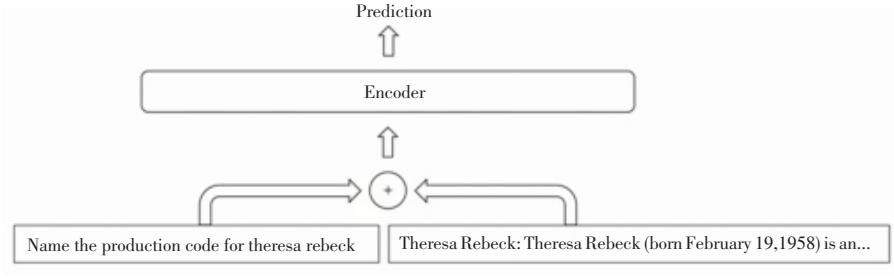


图 4 符号化知识增强

Fig. 4 Knowledge enhancement with symbolic knowledge

3.2.2 向量化知识的增强方法

向量化知识的增强方法是指将实体知识转化为向量化表示,再与问句中所对应的实体指称项(mention)的向量化表示进行融合。具体来说,对于问句 q 中出现的每个实体 e_i,假设其对应的问句中的实体指称项为 $q[p_i:q_i]$,p_i、q_i 分别表示实体指称项的起始索引和结束索引。首先,通过不同的方法来获得实体 e_i 的向量化表示 h_{e_i},然后将其输入到一个线性层中,再与向量化后的实体指称项对齐;其次,通过线性组合实体指称项和实体的向量化表示来获得知识增强的实体指称项的向量化表示,计算方式如公式(1)所示。

$$Enhance(q[p_i]) = Embedding(q[p_i]) + \alpha \times Linear(h_{e_i}) \quad (1)$$

其中,$Enhance$ 为增强后的实体指称项的向量化表示,$Embedding()$ 为语言模型的嵌入层,在训练过程中 α 从 0 退火到 λ,$\lambda \in [0,1]$。

本文提出了两种不同的实体向量化方法。第一种方法,即图向量化知识增强(Knowledge Enhancement with Graph Embedding),在给定图结构

知识的情况下，包含实体与实体之间的语义关系，来自于全部实体的语义关系，对图结构的实体知识进行编码，获得实体的向量化知识表示。使用知识图谱向量化（Knowledge Graph Embedding，KGE）的方法来获得每个实体的向量化表示。本文使用提出的 TransE 模型来进行知识图谱向量化，称为图向量化知识增强（Knowledge Enhancement with Graph Embedding），如图 5 所示。

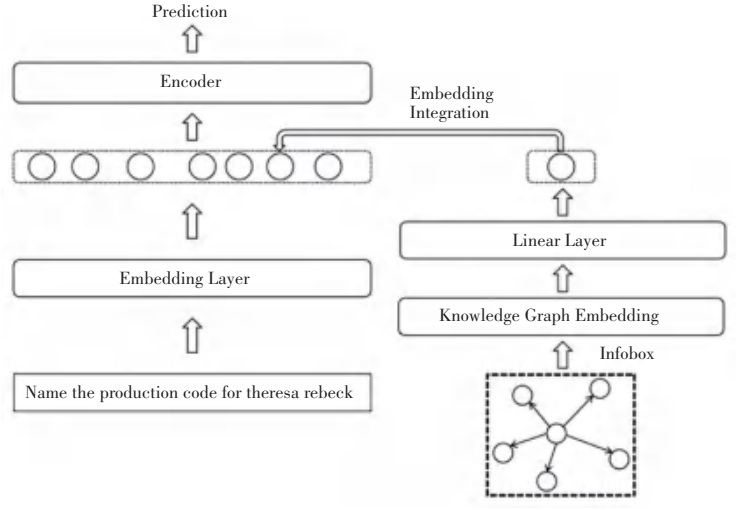


图 5　图向量化知识增强

Fig. 5　Knowledge enhancement with graph embedding

第二种方法指在缺乏图结构知识的情况下，通过语言模型对文本形式的实体知识进行编码，获得实体的向量化知识表示。因为实体知识总是以被描述的实体开头，如“Theresa Rebeck：Theresa Rebeck（born February 19，1958）is an American playwright，television writer，and novelist.”，所以采用实体知识描述的第一个标记的词向量作为实体知识的向量化表示，此方法称为文本向量化知识增强（Knowledge Enhancement with Textual Embedding），如图 6 所示。

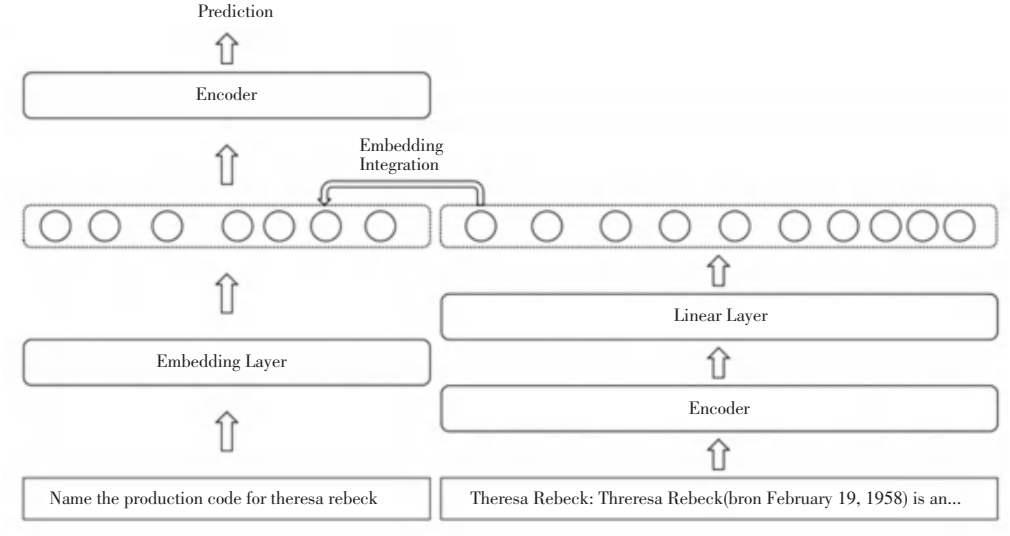


图 6　文本向量化知识增强

Fig. 6　Knowledge enhancement with textual embedding

在问句和知识融合后，再对增强后的问句进行编码，为下游子任务提供更好的表示。

3.3　解码模块

此模型采用 HydraNet 模型的解码方式，采用草图框架，将 SQL 查询划分为 6 个子任务，根据各个子任务是否依赖具体的列，将其划分为全局任务和局部任务两类。

全局任务主要包含 3 个子任务 SELECT－COLUMN、WHERE－NUMBER、WHERE－COLUMN。SELECT-COLUMN 子任务的目标是预测 SELECT 从

句中出现的列,本研究 SELECT 从句中出现的列固定为 1 个。所以,对所有候选列计算其出现在 SELECT 从句的分数,选择分数最高的列,式(2):

$$P_{sc}(c_i) = sigmoid(W_1\, h^{c_i}_{[cls]}) \quad (2)$$

其中,$h^{c_i}_{[cls]}$ 表示表 t_q 的第 i 列与问句交互后得到的全局向量表示,$P_{sc}(c_i)$ 为第 i 列出现在 SELECT 从句中的分数。

WHERE - NUMBER 子任务的目标是预测 WHERE 从句中条件列的数量。本文任务中 WHERE 从句中条件列可以为空,至多出现 4 个条件列,将问题转化为五分类任务。问句对应的 SQL 中的 WHERE 从句包含多少条件列,主要取决于问句,但是本模型中的问句和表中每个列单独交互,得到的多个全局表示都可以预测出 WHERE - NUMBER,需要根据每个列的相关度来对预测结果加权,式(3)~式(5):

$$P_{re}(c_i) = sigmoid(W_2\, h^{c_i}_{[cls]}) \quad (3)$$

$$P_{wn}(num_j \mid c_i) = \mathrm{softmax}(W_3[j,:].h^{c_i}_{[cls]}) \quad (4)$$

$$\hat{num} = \mathrm{argmax} \sum_{c_i \in C_t} P_{re}(c_i)\, P_{wn}(num_j \mid c_i) \quad (5)$$

其中,$P_{re}(c_i)$ 为第 i 列出现在 SQL 中的分数,$P_{wn}(num_j \mid c_i)$ 为第 i 列与问句交互后得到的全局表示预测 WHERE 从句中列数量为 j 的概率,将 $P_{re}(c_i)$ 和 $P_{wm}(num_j \mid c_i)$ 加权求和,取概率最高的数为 WHERE-NUMBER。

WHERE - COLUMN 子任务的目标是预测 WHERE 从句中出现的列,对所有候选列计算其出现在 WHERE 从句的分数,选择分数最高的前 WHERE-NUMBER 个列,式(6)。

$$P_{wc}(c_i) = sigmoid(W_4\, h^{c_i}_{[cls]}) \quad (6)$$

其中,$h^{c_i}_{[cls]}$ 表示表 t_q 的第 i 列与问句交互后得到的全局向量表示,$P_{wc}(c_i)$ 为第 i 列出现在 WHERE 从句中的分数。

局部任务依赖从句中的列的预测结果,包括 SELECT - AGGREGATION、WHERE - OPERATOR、WHERE - VALUE。SELECT - AGGREGATION 子任务的目标是预测 SELECT 从句中的列对应的聚合操作,从 A =['', 'MAX', 'MIN', 'COUNT', 'SUM', 'AVG']中选择概率最大的聚合操作,式(7)。

$$P_{sa}(A_j \mid c_i) = \mathrm{softmax}(W_5[j,:].h^{c_i}_{[cls]}) \quad (7)$$

其中,$h^{c_i}_{[cls]}$ 表示表 t_q 的第 i 列与问句交互后得到的全局向量表示,$P_{sa}(A_j \mid c_i)$ 为第 i 列对应第 j 个聚合操作符的概率。

WHERE - OPERATOR 子任务的目标是预测 WHERE 从句中的条件列对应的操作符,从 O = ['=', '>', '<']中选择概率最大的操作符,式(8)。

$$P_{wo}(O_j \mid c_i) = \mathrm{softmax}(W_6[j,:].h^{c_i}_{[cls]}) \quad (8)$$

其中,$h^{c_i}_{[cls]}$ 表示表 t_q 的第 i 列与问句交互后得到的全局向量表示,$P_{wo}(O_j \mid c_i)$ 为第 i 列对应第 j 个操作符的概率。

WHERE - VALUE 子任务的目标是预测 WHERE 从句中条件列对应的条件值,可以被理解为从问句中抽取一段文本,预测条件值在自然语言问句中的起始位置,式(9)和式(10):

$$P^{start}_{wv}(x_j = start \mid c_i) = \mathrm{softmax}(W_7\, h^q_j) \quad (9)$$

$$P^{end}_{wv}(x_j = end \mid c_i) = \mathrm{softmax}(W_8\, h^q_j) \quad (10)$$

其中,$P^{start}_{wv}(x_j = start \mid c_i)$,$P^{end}_{wv}(x_j = end \mid c_i)$ 分别为第 i 列对应的条件值以问句的第 j 个标记为起始位置、结束位置的概率。

4 实验

4.1 实验准备

实验使用的 WikiSQL 数据集是大型 NL2SQL 数据集之一,基于维基百科文章构造自然语言问句和对应的 SQL 查询。训练集、验证集、测试集基于不同的表,分别包含56 355,8 421,15 878个问句-SQL 查询对。本文选用 RoBERTa-base 作为基础编码器,AdamW 为优化器。

执行指导编码(Execution-Guided Decoding),简称 EG,利用 SQL 查询的执行结果来指导编码过程。如果预测的 SQL 执行结果出错或返回空结果,EG 将认为此条 SQL 预测错误,会将其排除选择概率次高的 SQL。在模型预测结束后,运用 EG,进一步提升模型的效果。

4.2 评估指标

使用目前主流的两种评估指标,即逻辑形式准确率(LF) 和执行结果准确率(EX),来评估模型的效果。逻辑形式准确率是指预测生成的 SQL 与真实标注的 SQL 匹配的比例(这里匹配指 SQL 语句完全一致);执行结果准确率是指执行预测生成的 SQL 的结果与执行真实标注的 SQL 的结果匹配的比例。

4.3 基线

本文以当前最优模型 HydraNet 作为基础模型,将问句与相应表中的所有列名分别交互,得到编码层输出,再根据草图框架,利用列排序等方法对各个子任务进行解码预测。

本文以 RoBERTa-base 作为基础编码器复现了 HydraNet 模型,将其作为实验的基线。统一运用

EG，对得到的实验结果做进一步的比较。

4.4 结果

本文模型的逻辑形式准确率和执行结果准确率均优于基线，在摘要、类型、标签、语义关系上分别运用符号化知识增强（SK）、文本向量化知识增强（TE）、图向量化知识增强（GE）得到逻辑形式准确率和执行结果准确率见表 2、表 3。

表 2　测试集上逻辑形式准确率

Tab. 2　Logical form accuracy (%) on WikiSQL test set of various methods

模型	摘要	类型	类型	语义关系
HydraNet		80.8		
KESQL+SK	80.5	81.7	80.2	81.0
KESQL+TE	81.8	**82.1**	81.6	**82.1**
KESQL+GE	–	–	–	81.6
HydraNet+EG		84.9		
KESQL+SK+EG	84.6	85.0	84.9	85.0
KESQL+TE+EG	**85.6**	85.5	85.3	85.4
KESQL+GE+EG	–	–	–	85.3

表 3　测试集上执行结果准确率

Tab. 3　Execution accuracy (%) on WikiSQL test set of various methods

模型	摘要	类型	类型	语义关系
HydraNet		86.4		
KESQL+SK	86.0	87.1	85.6	86.4
KESQL+TE	86.9	**87.3**	86.9	**87.3**
KESQL+GE	–	–	–	86.8
HydraNet+EG		91.0		
KESQL+SK+EG	90.8	90.8	90.9	91.0
KESQL+TE+EG	**91.2**	**91.2**	**91.1**	**91.2**
KESQL+GE+EG	–	–	–	**91.2**

通过表 2 和表 3，可以观察到：

（1）类型和语义关系可以更充分地补全 NL2SQL 任务中缺失的语义，对效果的提升最明显。因为类型信息仅是几个具体类型的拼接，语义关系包含的是关键的键值对，相对来说带来的干扰更小；而摘要的第一句话虽然非常重要，但太过精练反而可能会漏掉一些信息；

（2）向量化知识增强的方法都比符号化知识增强的方法好，因为符号化知识增强的方法过于简单直接，融入外部知识反而带来一定的噪声干扰。

5　结束语

本文针对自然语言问句存在语义缺失的问题，使用了外部知识图谱来对自然语言问句进行语义增强，使其包括充分的语义信息。本文提出对问句中出现的命名实体进行增强，使用现有的实体链接技术来将问句中的命名实体链接到外部知识图谱中，并提出了一种基于符号化知识的增强方法和两种向量化知识的增强方法。同时，系统调研了摘要（Abstract）、类型（Type）、标签（Category）、语义关系（Infobox）等 4 种类型知识的增强效果，最终发现使用类型和语义关系两种类型的知识来进行文本向量化知识增强的效果最好。

参考文献

[1] XU X, LIU C, SONG D. Sqlnet: Generating structured queries from natural language without reinforcement learning[J]. arXiv preprint arXiv:1711.04436, 2017.

[2] WANG C, TATWAWADI K, BROCKSCHMIDT M, et al. Robust text-to-sql generation with execution-guided decoding[J]. arXiv preprint arXiv:1807.03100, 2018.

[3] HWANG W, YIM J, PARK S, et al. A comprehensive exploration on wikisql with table-aware word contextualization[J]. arXiv preprint arXiv:1902.01069, 2019.

[4] HE P, MAO Y, CHAKRABARTI K, et al. X-SQL: reinforce schema representation with context[J].arXiv preprint arXiv:1908.08113, 2019.

[5] LYU Q, CHAKRABARTI K, HATHI S, et al. Hybrid ranking network for text-to-sql[J]. arXiv preprint arXiv:2008.04759, 2020.