一种依需聚合的语义解析图查询模型

李 青[1],钟 将[1],李立力[2],李 琪[3],张淑芳[4],张 剑[5]
(1. 重庆大学计算机学院,重庆 400044;2. 重庆大学土木工程学院,重庆 400044;
3. 绍兴文理学院计算机科学与工程系,浙江绍兴 312000;4. 重庆电子工程职业学院,重庆 401331;
5. 重庆西信天元数据资讯有限公司,重庆 401121)

摘 要: 本文设计并实现了依需聚合的语义深层网查询模型——SemtoSql +. 提出以长短期记忆网络为基础,采用词嵌入技术将语料库训练为模型输入的词向量;并结合依赖关系图,将 SQL 语句四个层级的生成问题转换为依赖关系图中槽的填充问题,同时引入注意力机制有效避免了传统模型中的顺序问题;采用随机蒙蔽机制,构建依需聚合的增强型 SemtoSql + 模型.

关键词: 自然语义处理; 复杂事件; 语义网; 深度学习
中图分类号: TP302.1 **文献标识码:** A **文章编号:** 0372-2112 (2020)04-0763-09
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2020.04.018

Semantic Parsing Graph Query Model for On-Demand Aggregation

LI Qing[1], ZHONG Jiang[1], LI Li-li[2], LI Qi[3], ZHANG Shu-fang[4], ZHANG Jian[5]
(1. *College of Computer Science, Chongqing University, Chongqing* 400044, *China*;
2. *School of Civil Engineering, Chongqing University, Chongqing* 400044, *China*;
3. *Department of Computer Science and Engineering, Shaoxing University, Shaoxing, Zhejiang* 312000, *China*;
4. *Chongqing College of Electronic Engineering, Chongqing* 401331, *China*;
5. *Chongqing Xixintianyuan Data Information Co., Ltd., Chongqing* 401121, *China*)

Abstract: In this paper, we design and propose SemtoSql +, a semantic deep network query model based on demand aggregation. At the same time, it is a network to address the complex and cross-domain Text-to-SQL generation task. Based on LSTM and Word2Vec embedding technology, the corpus is trained as the input word vector of the model. Combined with the dependency graph method, the problem of SQL statement generation transforms into slot filling. SemtoSql + divides complex tasks into four levels and constructs by the need of aggregation, using the attention mechanism to effectively avoid the order problem in the traditional model and using a random masked mechanism to enhance the model.

Key words: natural semantic processing; complex events; semantic web; deep learning

1 引言

近年来,随着人工智能技术的发展,自然语言查询的要求已不限于简单的数据检索,更需要系统能有效快速地响应包含复杂业务逻辑和语义的查询检索需求. 采用语义分析技术,将自然语言的表达映射到机器可以理解的结构化查询语言已成为研究的热点. 其中,由文本到 SQL 任务是自然语言处理(NLP)中语义分析最重要的子任务之一[1]. 针对各种传统 NLP 查询任务,可以采用最先进的递归神经网络的方法在简单的 WIKISQL 数据集上获得了精确匹配精度[2,3]. 然而,正如 Catherine[4] 所说评估模型应衡量对真实不可见数据的推广程度. 传统模型只是学习匹配简单语义解析结果,难于真正理解输入的复杂语义含义,更难推广到新的程序和数据库. 最重要的是,这些现存的复杂数据集并不包括跨领域的复杂语义查询[5].

收稿日期:2018-12-17;修回日期:2019-11-18;责任编辑:孙瑶
基金项目:国家重点研发计划(No. 2017YFB1402401);中央高校研究生科研创新项目(No. 2018CDYJSY0055); 重庆市研究生科研创新项目(No. CYB18058);陕西省教育厅科学技术研究计划(No. 18JK1130);重庆市技术创新与应用示范项目(No. cstc2018jszx-cyzdX0086);重庆市技术创新与应用发展重点项目(No. cstc2019jscx-fxyd0142);重庆市社会事业与民生保障科技创新专项(No. cstc2017shmsA0641);重庆市教育委员会科学技术研究计划青年项目(No. KJQN201903112)

由此可见,更要求系统能够理解复杂自然语言问题并生成对应的 SQL 查询. 本文以 Yu 等人提出的 Spider 任务为例[5]进行研究,其是一个大规模的人类标记文本到 SQL 的基准数据集. 其重新将 SQL 查询进行困难级别划分,定义了一些新的复杂的跨域文本到 SQL 的任务. 如果不能真正理解输入问题的语义,任务就无法轻松解决. 例如,sqlnet 模型[6]的准确率只有 14.3%,typesql 模型的准确率只有 10.6%.

本文为了更好的解决这些问题,设计并实现了依需聚合的语义深层网查询模型——SemtoSql + 模型. 主要贡献如下:

(1)基于草图的方法构建 SemtoSql + 模型,并将任务视为一个插槽填充问题. 通过构建依赖关系图中分组不同的插槽来捕获属性之间的关联关系;依据 SQL 子句构建特定语法树的解码器,保存具有 SQL 图路径历史记录,构建支持表的列注意编码器.

(2)采用细粒度刻画语义分析的子任务,并在解码过程的不同语义层级引入注意力机制. 高层级结构被编码成矢量,约束了较低级的生成过程,来引导解码.

(3)为了处理具有单向约束问题的复杂跨域文本到 SQL 的任务,本文提出"随机蒙蔽"机制训练任务目标. 该思想来源于自然语言翻译问题中完形填空任务的启发[7].

2 相关工作

语义解析是将自然语言映射到准确意义的逻辑形式和可执行程序,然后在知识图上执行这些程序的过程[8~15]. 最近,将自然语言文本转化到结构化查询语言的 Text-to-SQL 问题得到了广泛地关注[16~19]. 与此同时,Sequence-to-Sequence(Seq2Seq)模型随着技术的发展在语义解析方向上的应用也取得了可喜的进展[20~22].

大多数传统 RNN 方法都集中在构造特定的表模型[6,23~27]. 其中,Dong 等人开发了树形结构模型,并在多个数据集中验证了模型的良好性能[20,28]. Xiao 等人首先利用语法来限制解码过程[29]和语义表示方法[30]. Yin 等人提出深度优先的语法模型,该模型从左到右顺序生成语法树[31]. Rabinovich 等人提出抽象语法网络和 NoSQL[32]思想,子模型根据生成的树结构进行动态组合. SEQ2TREE 模型[21]是 2016 年由 Li 等人提出的 Seq2Seq 神经网络语义解析器,并在大量非人工标注的语义解析数据集上实现了最优性能. 其将输入的自然语言编码成矢量表示形式,并利用长短时记忆神经网络将其相应的逻辑形式生成序列或树. SQLizer[33]是一种依赖现有语义的解析器[34,35]. 其将一个自然语言的问题映射为依赖关系图的查询方法. 由于其不需要进行数据预训练,因此它不受限于任何特定类型的数据库. Seq2SQL 模型[25]是 2017 年由 Salesforce 公司提出的,主要针对自然语言数据库操作方法进行深入研究. 模型具有对未知模式的泛化能力,可有效解决 Seq2Seq 模型的低效性问题,并构造 WiKiSQL 数据集. SQLNET[7]是 2017 年由 Xu 等人提出的解决 NL2SQL 任务的模型,它在 Seq2SQL 模型的基础上提出了基于 SQL 语法结构的依赖关系图来生成 SQL 查询语句. 虽然从一定程度上解决了 Where 子句中条件的顺序问题,但其 Where 子句的语法结构单一,只存在 AND 语法,并不符合人们自然语言描述问题的方式,以至于其不适用于包含 OR 语法的描述方式. SyntaxSQLNet[1]是第一个有效解决复杂领域文本到 SQL 的生成任务的模型. 该模型使用了一个基于树的 SQL 生成器,它能够解决嵌套查询隐含的数据库.

本文结合依赖图方法,将 SQL 语句生成问题转化为插槽填充问题. 同时,结合了 2018 年最新提出的 SyntaxSQLNet 模型[1]和 NoSQL[32]思想构建 SemtoSql + 模型. 由于本文将复杂的任务分成四个层次,使得方法灵活性更强,更容易将草图映射到不同类型的事件层面,成功地解决了传统语义解析中事件时序约束和谓词约束描述能力不足的问题.

3 语义解析数据集

本文在图 1 中对比了各种语义分析从文本到 SQL 的数据集. 由图 1 可见在 9 种不同组件(如#Q 等)中的数据分布情况.

可以根据图 1 中十种数据集的对比得出如下结论. ATIS[33]和 Geo-Query[34]是固定模式数据集,仅解决封闭域上的复杂或组合问题. Academic[35]是特定领域数据集,仅包含小于 200 维的 SQL 查询. WikiSQL 数据集[24]的 SQL 查询和 Table 数量较大,但仅包含简单 SELECT 和 WHERE 子句的 SQL 查询. 此外,每个数据库只是一个简单的表,没有任何外键. 且难于处理复杂的 SQL 查询(如:通过组、顺序或嵌套的查询)和具有多个表和外键的数据库. Yu 等人提出的 Text-to-SQL 数据集 Spider[5],很好地解决了复杂跨领域数据查询的数据集的缺失问题[36~38]. 使其成为第一个复杂的跨域文本到 SQL 数据集. 其主要有以下三个方面的特征:海量性、复杂性、跨域性. 其包含 1 万余个问题,由 200 个复杂的跨域数据库组成,且查询几乎涵盖所有的 SQL 组件. 此外,各数据库都含有用外键链接的多个表.

4 研究方法

本文将任务视为一个插槽填充问题,采用基于草图的方法构建 SemtoSql + 模型,其模型结构如图 2 所示.

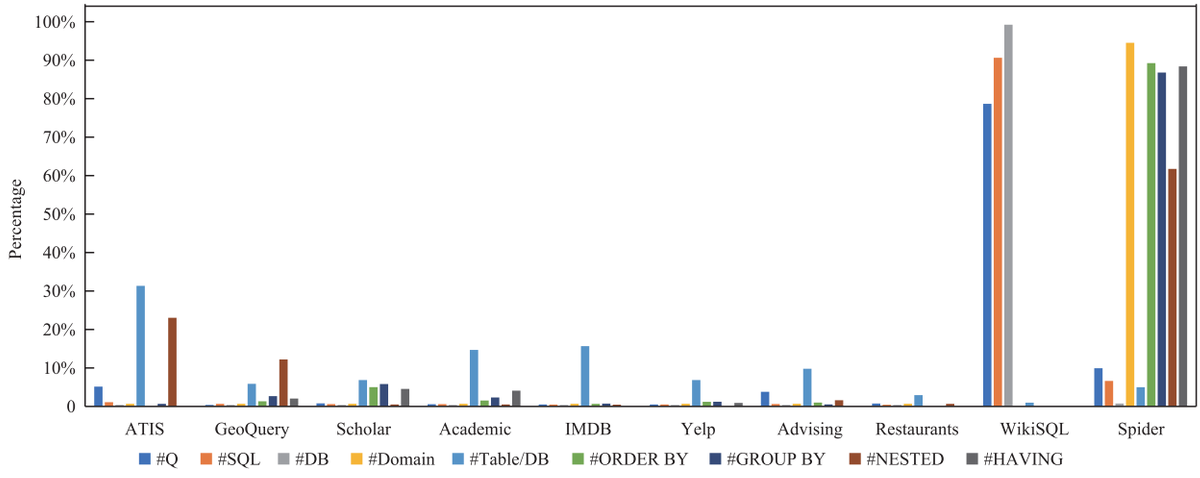


图1　十种语义解析数据集的比较图

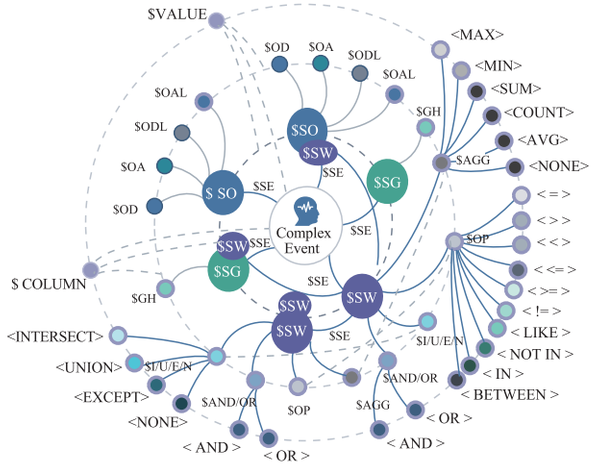


图2　SemtoSql+模型结构图

4.1　复杂任务描述模型

在进行复杂任务描述时,事件最基本的元素. 本文对原有事件定义方式进行改进,按粒度等级将事件分为原子事件、组合事件、聚集事件和复杂事件. 并给出自然语言实例和 SQL 查询描述实例,如图 3 所示. 数据查询所使用的 SQL 语句主要为 SELECT 语句. 其中 Token(即 SELECT, WHERE, GROUP BY 和 ORDER BY 等)表示 SQL 关键字,以 $ 开头的 Token 表示要填充的插槽, $ 之后的名称表示需要预测的类型. 各粒度等级定义如下:

原子事件(Atomic Events, T_{ATE})　是指事件产生的最小原子语义描述单元.

组合事件(Composite Events, T_{CSE})　是指由简单原子事件按照一定归类组合而成.

聚合事件(Aggregate Events, T_{AGE})　是指与单个对象实例关联的某类组合事件数据按照某种时段特征函数聚集来生成事件.

复杂事件(Complex Events, T_{CPE})　是指由一系列原子事件、聚合事件或复杂事件通过再次复合得到的新事件.

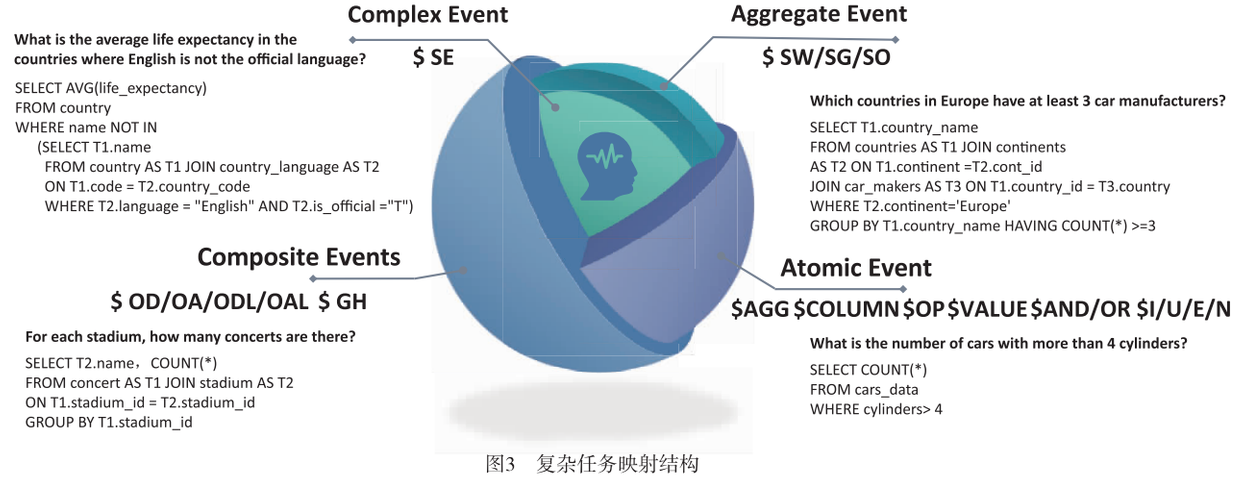


图3　复杂任务映射结构

4.2 语义依赖关系图

因结构化查询语言(SQL)与自然语言表述结构不同,导致传统语义场模型的映射网络难于进行. 同时,由于查询并不需要利用自然语言中全部词语信息,导致在自然语言中的无用字词和未登录词成为查询干扰,诱使模型难于得到令人满意的结果. 本文将 SQL 语句进行拆分,分解出不同层级与自然语言的依赖关系,并依据依赖关系图,将查询问题转化为槽填充问题,以增强 SQL 语句生成的准确率.

本文结合 SQL 语法特点构造解码器以生成复杂任务查询,将每个解码步骤设计为可递归调用的模块. 通过 SQL 语句结构语法进行划分后,需要通过自然语言进行预测关键词之外的 Slot 值. 在 SQL 查询基础形态 SELECT 子句中,COLUMN 对应数据表中的列名插槽则需要根据自然语言问题进行预测;而针对聚合运算符的分类问题 AGG 子句,则需要同时依赖于自然语言问题和 COLUMN. 在 WHERE 子句中,经过 OR、AND 关键字的分割后其每一部分均可表示为 {$COLUMN, $OP, $VALUE} 的形式. 其中,COLUMN 依赖于自然语言问题,而 OP 和 VALUE 则类似于 AGG 子句,同时依赖于自然语言问题和 COLUMN. 而在 GROUP BY 和 ORDER BY 子句中,嵌套着其他原子事件模块.

由此可以看出,AGG、OP、VALUE 的预测均依赖于其前置的 COLUMN,能够有效避免传统 Seq2Seq 模型中乱序的问题. 同时,使用槽填充的方法可以有效预测基本 SQL 结构. 这里预测得到的 {$COLUMN, $OP, $VALUE} 保证了查询条件的一致,且无需再通过其他方式进行 OP、VALUE 顺序的预测.

4.3 输入编码器

由依赖关系图可知,语义解析模块可以分为三种类型的信息:查询问题、表模式和 SQL 解码历史依赖槽路径. 首先,自然语言问题的词向量输入后该模型通过 Bi-LSTM 对问题句进行编码. 同时对每个模块 SemtoSql + 模型采用注意力机制来编码表示问题,如下式所示:

$$E_{A|B}=\mathrm{softmax}(E_A(WE_B)^T) \tag{1}$$

经过训练得到的 $E_{A|B}$ 能够通过注意力权值对 A 与 B 的关系进行表示. 经过 Softmax 激活函数能够将结果映射至[0,1]区间. 为了增强模型泛化能力,SemtoSql + 模型同时支持自顶向下(T_{CPE}-T_{AGE}-T_{CSE}-T_{ATE})、自底向上的两种构建方法(T_{ATE}-T_{CSE}-T_{AGE}-T_{CPE}). SemtoSql + 模型通过 Bi-LSTM 将列上的隐藏状态表示为问题嵌入(E_Q)、图路径(E_{GP})和列嵌入(E_{COL}),同时定义 $\boldsymbol{U}$ 为各全链接层的权重矩阵,$\boldsymbol{W}$ 为各级 Embedding 的权重矩阵,且 $\boldsymbol{U}$ 和 $\boldsymbol{W}$ 表示模块之间不共享的可训练参数. 下面将针对模型的四个模块进行分述.

4.3.1 复杂事件模块 T_{CPE}

$SE 选择列取决于表列以及问题. 由于复杂事件可能包含多个聚合事件、组合事件、原子事件. $SE 是从 < SELECT1,SELECT2,… > 中选择的. T_{CPE} 定义为

$$T^n_{CPE}=\mathrm{softmax}\left(U_{CPE}\tanh\left(\begin{array}{l}(W^n_Q E^n_{Q|AGE})^T\\+(W^n_{GP}E^n_{GP|CPE})^T\\+(W^n_{CPE}E^n_{CPE})^T\end{array}\right)\right) \tag{2}$$

4.3.2 聚合事件模块 T_{AGE}

$SW/SG/SO 给定 SQL 查询中的关键字权重、图路径和序号,并根据 $SW/SG/SO:<WHERE,GROUP BY,ORDER BY > 中的关键字输入隐藏状态.

$$T^n_{AGE}=\mathrm{softmax}\left(U_{AGE}\tanh\left(\begin{array}{l}(W^n_Q E^n_{Q|AGE})^T\\+(W^n_{GP}E^n_{GP|AGE})^T\end{array}\right)\right) \tag{3}$$

$$T^{val}_{AGE}=\mathrm{softmax}\left(U_{AGE}\tanh\left(\begin{array}{l}(W^{val}_Q E^{val}_{Q|AGE})^T\\+(W^{val}_{GP}E^{val}_{GP|AGE})^T\\+(W^{val}_{AGE}E^{val}_{AGE})^T\end{array}\right)\right) \tag{4}$$

4.3.3 组合事件模块 T_{CSE}

$OD/OA/ODL/OAL 在这个模块中,自底向上的建立模块应关注 $COLUMN 模块中的每个预测列,自顶向下的建立模块更应关注 ORDER BY 子句的预测. 预测列来自 $OD/OA/ODL/OAL:ORDER BY-<DESC,ASC,DESC LIMIT,ASC LIMIT >.

$$T_{ODAL}=\mathrm{softmax}\left(U_{ODAL}\tanh\left(\begin{array}{l}(W_Q E_{Q|CL})^T\\+(W_{GP}E_{GP|CL})^T\\+(W_{CL}E_{CL})^T\end{array}\right)\right) \tag{5}$$

$GH 类似于上一模块构建方法,自底向上和自顶向下的建立模块. 预测是否在 HAVING 子句中.

$$T_{GH}=\mathrm{softmax}\left(U_{GH}\tanh\left(\begin{array}{l}(W_Q E_{Q|CL})^T\\+(W_{GP}E_{GP|CL})^T\\+(W_{CL}E_{CL})^T\end{array}\right)\right) \tag{6}$$

4.3.4 原子事件模块 T_{ATE}

$AGG 在这个模型中,首先预测 $COLUMN 模块的每个列. 给定 SQL 查询中关键字的数量和权重,并根据 < MAX、MIN、SUM、COUNT、AVG、NONE > 中的关键字输入隐藏状态.

$$T^n_{AGG}=\mathrm{softmax}\left(U_{AGG}\tanh\left(\begin{array}{l}(W^n_Q E^n_{Q|CL})^T\\+(W^n_{GP}E^n_{GP|CL})^T\\+(W^n_{CL}E^n_{CL})^T\end{array}\right)\right) \tag{7}$$

$$T^{val}_{AGG}=\mathrm{softmax}\left(U_{AGG}\tanh\left(\begin{array}{l}(W^{val}_Q E^{val}_{Q|CL})^T\\+(W^{val}_{GP|CL}E^{val}_{GP|CL})^T\\+(W^{val}_{CL}E^{val}_{CL})^T\end{array}\right)\right) \tag{8}$$

$COLUMN 为了让模块准确的选择预测表列.

首先针对每个列名求其在自然语言问题中出现的概率. $ COLUMN 模块将条件于列的问题、图路径和类型隐藏状态的加权和传递给预测.

$$T_{\mathrm{COL}}^{n}=\mathrm{softmax}\left(\boldsymbol{U}_{\mathrm{COL}}\tanh\left(\begin{array}{l}(\boldsymbol{W}_{\mathrm{Q}}^{n}\boldsymbol{E}_{\mathrm{GP|COL}}^{n})^{\mathrm{T}}\\+(\boldsymbol{W}_{\mathrm{GP}}^{\mathrm{val}}\boldsymbol{E}_{\mathrm{GP|COL}}^{n})^{\mathrm{T}}\\+(\boldsymbol{W}_{\mathrm{COL}}^{n}\boldsymbol{E}_{\mathrm{COL}}^{n})^{\mathrm{T}}\end{array}\right)\right)\tag{9}$$

$$T_{\mathrm{COL}}^{\mathrm{val}}=\mathrm{softmax}\left(\boldsymbol{U}_{\mathrm{COL}}\tanh\left(\begin{array}{l}(\boldsymbol{W}_{\mathrm{Q}}^{\mathrm{val}}\boldsymbol{E}_{\mathrm{Q|COL}}^{\mathrm{val}})^{\mathrm{T}}\\+(\boldsymbol{W}_{\mathrm{GP}}^{\mathrm{val}}\boldsymbol{E}_{\mathrm{GP|COL}}^{\mathrm{val}})^{\mathrm{T}}\\+(\boldsymbol{W}_{\mathrm{COL}}^{\mathrm{val}}\boldsymbol{E}_{\mathrm{COL}}^{\mathrm{val}})^{\mathrm{T}}\end{array}\right)\right)\tag{10}$$

$ OP　对于每个预测条件列, $ OP 模块将预测为三分类问题,并在 < = , > , < , > = , < = ,! = , LIKE,NOT IN,IN,BETWEEN > 中进行选择.

$$T_{\mathrm{OP}}^{n}=\mathrm{softmax}\left(\boldsymbol{U}_{\mathrm{OP}}\tanh\left(\begin{array}{l}(\boldsymbol{W}_{\mathrm{Q}}^{n}\boldsymbol{E}_{\mathrm{Q|CL}}^{n})^{\mathrm{T}}\\+(\boldsymbol{W}_{\mathrm{GP}}^{n}\boldsymbol{E}_{\mathrm{GP|CL}}^{n})^{\mathrm{T}}\\+(\boldsymbol{W}_{\mathrm{CL}}^{n}\boldsymbol{E}_{\mathrm{CL}}^{n})^{\mathrm{T}}\end{array}\right)\right)\tag{11}$$

$$T_{\mathrm{OP}}^{\mathrm{val}}=\mathrm{softmax}\left(\boldsymbol{U}_{\mathrm{OP}}\tanh\left(\begin{array}{l}(\boldsymbol{W}_{\mathrm{Q}}^{\mathrm{val}}\boldsymbol{E}_{\mathrm{Q|CL}}^{\mathrm{val}})^{\mathrm{T}}\\+(\boldsymbol{W}_{\mathrm{GP}}^{\mathrm{val}}\boldsymbol{E}_{\mathrm{GP|CL}}^{\mathrm{val}})^{\mathrm{T}}\\+(\boldsymbol{W}_{\mathrm{CL}}^{\mathrm{val}}\boldsymbol{E}_{\mathrm{CL}}^{\mathrm{val}})^{\mathrm{T}}\end{array}\right)\right)\tag{12}$$

$ VALUE　为了校验在操作符后是否存在其他子查询,这个模块允许递归地解码查询. 复杂事件可以包含多个 T_{AGE}、T_{CSE}、T_{ATE},其意义是预测新的子查询或终端值.

$$T_{\mathrm{V}}=\mathrm{softmax}\left(\boldsymbol{U}_{\mathrm{V}}\tanh\left(\begin{array}{l}(\boldsymbol{W}_{\mathrm{Q}}\boldsymbol{E}_{\mathrm{Q|CL}})^{\mathrm{T}}\\+(\boldsymbol{W}_{\mathrm{GP}}\boldsymbol{E}_{\mathrm{GP|CL}})^{\mathrm{T}}\\+(\boldsymbol{W}_{\mathrm{CL}}\boldsymbol{E}_{\mathrm{CL}})^{\mathrm{T}}\end{array}\right)\right)\tag{13}$$

$ AND/OR　当存在多个 $ COLUMN 模块时,模块需要从 < AND,OR > 中预测模块间关系.

$$T_{\mathrm{AO}}=\mathrm{softmax}\left(\boldsymbol{U}_{\mathrm{OA}}\tanh\left(\begin{array}{l}(\boldsymbol{W}_{\mathrm{Q}}\boldsymbol{E}_{\mathrm{Q}})^{\mathrm{T}}\\+(\boldsymbol{W}_{\mathrm{GP}}\boldsymbol{E}_{\mathrm{GP}})^{\mathrm{T}}\end{array}\right)\right)\tag{14}$$

$ I/U/E/N　同样,在这个模块中,给定 SQL 查询中关键字的数量和权重,图路径和隐藏状态类型取决于 < INTERSECT,UNION,EXCEPT,NONE > 中的关键字.

$$T_{\mathrm{IUEN}}=\mathrm{softmax}\left(\boldsymbol{U}_{\mathrm{IUEN}}\tanh\left(\begin{array}{l}(\boldsymbol{W}_{\mathrm{Q}}\boldsymbol{E}_{\mathrm{Q|IUEN}})^{\mathrm{T}}\\+(\boldsymbol{W}_{\mathrm{GP}}\boldsymbol{E}_{\mathrm{GP|IUEN}})^{\mathrm{T}}\\+(\boldsymbol{W}_{\mathrm{CL}}\boldsymbol{E}_{\mathrm{IUEN}})^{\mathrm{T}}\end{array}\right)\right)\tag{15}$$

4.4　随机蒙蔽机制

在本文中,该模块通过基于微调的方法来改进 SemtoSql + 模型. 采用随机蒙蔽机制的灵感来源于 BERT(Bidirectional Encoder Representations from Transformers)模型中 MLM(Masked Language Model)的思想[7].

随机蒙蔽机制是随机地掩盖了输入自然语言查询语句中的一些令牌,其目标是仅根据上下文预测蒙面词的原始层级事件. 与传统语言模型从左到右的预训练方式不同,随机蒙蔽机制的目标允许表示融合上下文中的每级内容,且损失函数由每级内容的 loss 值组成. 本文 SemtoSql + 模型将各个层级的事件采用随机掩盖的方法,随机屏蔽一定比例的输入令牌,然后再训练模型预测那些被的屏蔽令牌,使其正例与负例间 L2 范数值最小以达到模型调优的目的. 引入随机蒙蔽机制后,目标函数优化为下式:

$$L_{\mathrm{loss}}=[\rho_{\theta}(S_{n},\varphi)+(1-\rho_{\theta}(S_{n},\varphi^{-}))]+\xi\parallel\theta\parallel^{2}\tag{16}$$

其中,$\rho_{\theta}(S_{n},\varphi)$表示预测的输出与正例事件之间的距离,$\rho_{\theta}(S_{n},\varphi^{-})$为预测的输出与所有错误事件中得分最高的那个类别之间的距离,$\xi\parallel\theta\parallel^{2}$为正则项.

通过上述目标函数的优化,可将模型内随机蒙蔽后产生的预测事件与正例事件间的距离减小,从而达到提高模型效率的目的. 首先,训练数据生成器随机在各级中选择 15% 的事件作为[MASK]令牌. 其次,在所选事件部分的 80% 情况下,用[MASK]随机屏蔽所选事件,其余 20% 保持事件不变. 如此一来,随机替换只发生在所有令牌的 12%(即 15% 的 80%). 由此可见,随机蒙蔽不会损害模型的语言理解能力反而可以深化训练模型,让模型经过更多的预训练步骤后收敛. 样例可见表 1 所示.

表 1　[MASK]生成语句的样例

自然语言描述的问句	What is the average life expectancy in the countries where English is not the official language?
SQL 语句	SELECT AVG(life_expectancy) FROM country WHERE name NOT IN (SELECT T1. name FROM country AS T1 JOINcountry_language AS T2 ON T1. code = T2. country_code WHERE T2. language = "English" AND T2. is_official ="T")
[MASK]生成语句	SELECT AVG(life_expectancy) FROM [MASK] WHERE name NOT IN (SELECT T1. name FROM country AS T1 JOINcountry_language AS T2 ON T1. code = T2. country_code WHERE T2. language = "English" AND T2. is_official ="T")

5　实验结果

5.1　实验环境及数据集

本文模型实现基于 Python3.6 采用 PyTorch,采用 Bengio 提出的 Xavier 初始化方法进行[39]. 其中范围选取为[a,

b],随机初始化值为 $\boldsymbol{W}\sim\boldsymbol{U}\left[-\frac{\sqrt{6}}{\sqrt{d_{\text{in}}+d_{\text{out}}}}+\frac{\sqrt{6}}{\sqrt{d_{\text{in}}+d_{\text{out}}}}\right]$,矩阵大小为 $\boldsymbol{W}\in\mathbb{R}^{d_{\text{in}}\times d_{\text{out}}}$;所有网络最多运行 300 个 Epochs 并设置 Dropout 为 0.2;并将所有隐藏层的尺寸和召回率分别设置为 120 和 0.3;编码器 Encoder 为两层双向 LSTM,每层设置 100 个 LSTM 单元;使用 Adam 优化器[40],学习速率初始设置为 0.001,训练中采用动态调节的方式. 以方便对梯度的一阶矩估计(即,梯度的均值)和二阶矩估计(即,梯度的未中心化的方差)进行综合考虑,计算出更新步长.

在实验中,主要使用了 Spider 数据集,并在 WiKiSQL、ATIS 和 GEO 数据集上进行验证实验. Spider 数据集是一种新的大型人工注释文本到 SQL 数据集,具有复杂的 SQL 查询和跨域数据库. 本文将其分为 Train、Dev 和 Test 三部分. 并以 Salesforce 公司在 Amazon 上以众包形式得到 WiKiSQL 数据集上进行主要验证实验. 该数据集是提取于维基百科的 24241 个 HTML 表的数据,并人工注释了 80654 个自然语言问题实例与其对应的 SQL 查询语句.

5.2 评估指标

SemtoSql + 模型的评估以生成 SQL 语句的准确度为核心. 通过各级精度 ACC 指标评价生成查询语句的准确率.

$$\text{ACC}_n=\frac{N_{\text{ACC}}^n}{N_n} \tag{17}$$

设 N_n 为测试集中不同层级数据总数,N_{ACC}^n 为模型生成 SQL 语句与实际字符串匹配的数量. 其准确精度定义为 ACC_n. 同时,所有层级的准确精度定义为ACC_{all}. ACC_n 是测试集中不同层级的准确精度,N 为层级数目($N=4$).

5.3 实验结果与讨论

表 2 显示了 Spider 数据集上任务的主要结果. SemtoSql + 模型使用四个评估指标与之前的模型结果进行比较,其在验证集与测试集的评估结果如表 2 所示. 表 3 提供了 SELECT, WHERE, GROUP BY, ORDER BY 和 KEYWORD 子句的细分结果. 表 4 对比 SemtoSql + 模型在不同的数据集(Spider、WiKiSQL、ATIS 和 GEO)中的实验效果.

表 2 不同级别上对 SQL 查询进行精确匹配的准确性

Method	Test ACC						Dev ACC
	TATE	TCSE	TAGE	TCPE	All	Training time	All
seq2seq	21.10%	1.22%	1.17%	0.32%	5.95%	42h23min	3.48%
SQLNet	37.90%	11.90%	4.90%	1.10%	13.95%	23h2min	12.13%
TypeSQL	32.20%	6.20%	3.67%	0.68%	10.69%	21h45min	10.34%
IncSQL	58.20%	28.54%	21.79%	2.46%	27.75%	21h9min	25.49%
SyntaxSQLNet	71.11%	26.12%	23.03%	4.12%	31.10%	19h39min	28.56%
SemtoSql +	79.85%	34.69%	29.78%	9.26%	38.40%	11h7min	35.21%

表 3 在测试集中所有 SQL 查询的事件匹配结果对比

Method	SELECT	WHERE	GROUP BY	ORDER BY	KEYWORKD
seq2seq	13.00%	1.50%	3.30%	5.30%	8.70%
SQLNet	44.50%	19.80%	29.50%	48.80%	64.00%
TypeSQL	36.40%	16.00%	17.20%	47.70%	66.20%
IncSQL	59.21%	41.92%	23.40%	52.77%	67.33%
SyntaxSQLNet	62.50%	34.80%	55.60%	60.90%	69.60%
SemtoSql +	66.30%	42.45%	58.20%	62.11%	71.54%

表 4 在 SPIDER,WIKISQL,ATIS 和 GEO 数据集中 SemtoSql + 模型匹配准确率对比

Method	Spider		WiKiSQL		ATIS		GEO	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
SemtoSql +	35.21%	38.40%	79.20%	71.70%	89.30%	87.20%	89.60%	88.50%

如果不考虑数据库的内容,SemtoSql + 模型在精确度上比以前最好的工作高出 7.3% 到 38.4%. 根据表 2,SemtoSql + 将复杂事件的准确率提高了 5.14%,聚合事件提高了 6.75%. SemtoSql + 通过以一种简单但合理的方式编码不同级别的事件模型组件,在最具挑战性的子任务原子事件和复合事件子句上获得了更高的结果. 由于模型使用了 SQL 历史纪录和数据增强的模型,所以在所有 SQL 上实现了 38.4% 的精确匹配. 由于采

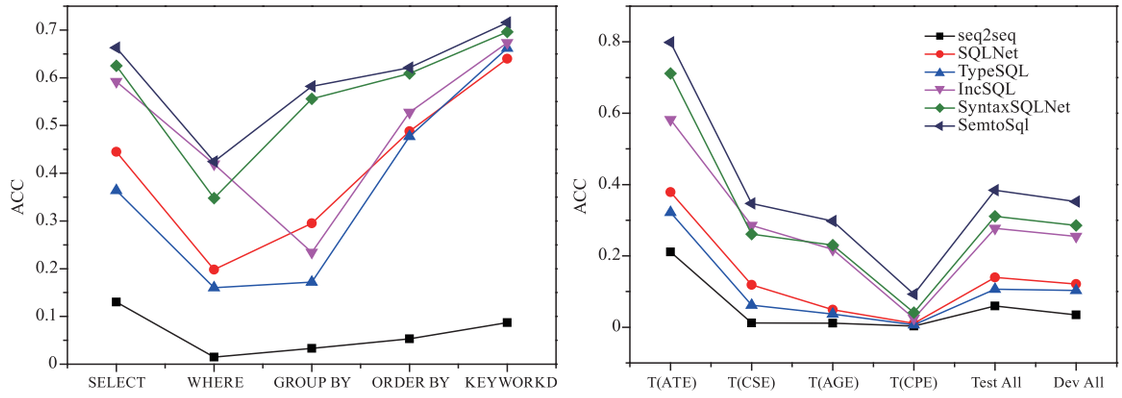


图4　不同模型准确率对比图

用了随机蒙蔽机制,使得语义理解有了更多的可能性. 因此不同事件层面的语义理解也更加深入. 同时,这个结果表明句子语义和图形路径历史信息对这个复杂文本到 SQL 任务是有益的. 实验结果如图 4 所示.

由于复杂事件的问题查询包含不同层面的嵌套查询. 正如表 3 所示,SemtoSql + 模型在不同的主要事件 SQL 查询上都表现出更优越的性能. 这个结果表明,在不可见的复杂数据库中,支持表的编码对于预测正确的列非常重要. 为了理解对模型的可扩展性,将其分别在不同的数据集上进行对比实验,实验结果表明该模型性能最好,如表 4 和图 5 所示.

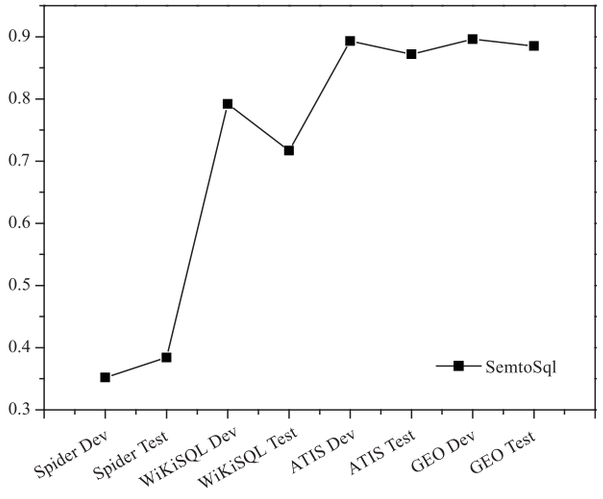


图5　SemtoSql模型在四个不同的数据集上是准确率对比图

相比 SyntaxSQLNet,SemtoSql + 模型的最大优势在于复杂事件的预测. 可以看到将 Seq2Seq 的解决方案改为依赖关系图中槽的填充问题. 预测所得的复杂事件约束条件之间为并列关系,不会因顺序问题而影响准确率,因而能够得到更好的效果. 同时,使用 Attention 机制能够使模型对在不同子网络预测时对自然语言问题序列中不同的单词与各级子事件进行关注.

6　总结

本文提出的 SemtoSql + 模型在解决复杂的跨域文本到 SQL 任务上表现出较好的结果. 成功引入当下最先进的随机屏蔽机制,深化模型预训练结果使其输出性能变优. 同时本文提出的模型能够较准确地预测含有嵌套的复杂 SQL 查询,并具有一定泛化性. 未来,还希望探索更多的 SemtoSql + 模型变体,例如引入不同的混沌理论,以更加自然的方式解决自然语言查询任务上 Text-to-SQL 的问题.

参考文献

[1] Yu T,Yasunaga M,Yang K,et al. Syntaxsqlnet:Syntax tree networks for complex and cross-domain text-to-SQL task [J]. arXiv Preprint,2018,arXiv:1810.05237.

[2] Shi T,Tatwawadi K,Chakrabarti K,et al. IncSQL:Training incremental text-to-SQL parsers with non-deterministic oracles[J]. arXiv Preprint,2018,arXiv:1809.05054.

[3] Yu T, Li Z, Zhang Z, et al. Typesql: Knowledge-based type-aware neural text-to-SQL generation[J]. arXiv Preprint,2018,arXiv:1804.09769.

[4] Finegan-Dollak C,Kummerfeld J K,Zhang L,et al. Improving text-to-SQL evaluation methodology[J]. arXiv Preprint,2018,arXiv:1806.09029.

[5] Yu T,Zhang R,Yang K,et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task[J]. arXiv Preprint,2018,arXiv:1809.08887.

[6] Xu X,Liu C,Song D. Sqlnet:Generating structured queries from natural language without reinforcement learning[J]. arXiv Preprint,2017,arXiv:1711.04436.

[7] Devlin J,Chang M W,Lee K,et al. Bert:Pre-training of deep bidirectional transformers for language understanding [J]. arXiv Preprint,2018,arXiv:1810.04805.

[8] Zelle J M,Mooney R J. Learning to parse database queries

using inductive logic programming[A]. Proceedings of the National Conference on Artificial Intelligence[C]. AAAI Press,1996. 1050 – 1055.

[9] Zettlemoyer L,Collins M. Online learning of relaxed CCG grammars for parsing to logical form[A]. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)[C]. Association for Computational Linguistics,2007. 678 – 687.

[10] Zettlemoyer L S,Collins M. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars[J]. arXiv Preprint, 2012, arXiv: 1207. 1420.

[11] Walker R D. Standards for antimicrobial susceptibility testing[J]. American journal of veterinary research,1999,60(9):1034.

[12] Khoussainova N, Balazinska M, Suciu D. Probabilistic event extraction from RFID data[A]. Proceedings of the 24th International Conference on Data Engineering[C]. ICDE Presss,2008. 1480 – 1482.

[13] Banarescu L,Bonial C,Cai S,et al. Abstract meaning representation for embanking[A]. Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse[C]. LAW@ ACL Press,2013. 178 – 186.

[14] Artzi Y,Zettlemoyer L. Weakly supervised learning of semantic parsers for mapping instructions to actions[J]. Transactions of the Association for Computational Linguistics,2013,1:49 – 62.

[15] Reddy S, Lapata M, Steedman M. Large-scale semantic parsing without question-answer pairs[J]. Transactions of the Association for Computational Linguistics, 2014, 2: 377 – 392.

[16] Warren D H D,Pereira F C N. An efficient easily adaptable system for interpreting natural language queries[J]. Computational Linguistics,1982,8(3 – 4):110 – 122.

[17] Popescu A M, Etzioni O, Kautz H. Towards a theory of natural language interfaces to databases[A]. Proceedings of the 8th International Conference on Intelligent User Interfaces[C]. US:ACM Press,2003. 149 – 157.

[18] Li F, Jagadish H V. Constructing an interactive natural language interface for relational databases[J]. Proceedings of the VLDB Endowment,2014,8(1):73 – 84.

[19] Wang C,Cheung A,Bodik R. Synthesizing highly expressive SQL queries from input-output examples[J]. ACM SIGPLAN Notices,2017,52(6):452 – 466.

[20] Dong L,Lapata M. Language to logical form with neural attention[J]. arXiv Preprint,2016,arXiv:1601. 01280.

[21] Jia R, Liang P. Data recombination for neural semantic parsing[J]. arXiv Preprint,2016,arXiv:1606. 03622.

[22] Che Z,Purushotham S,Cho K,et al. Recurrent neural networks for multivariate time series with missing values[J]. Scientific Reports,2018,8(1):6085.

[23] Yin P,Lu Z,Li H,et al. Neural enquirer:Learning to query tables with natural language[J]. arXiv Preprint,2015, arXiv:1512. 00965.

[24] Zhong V,Xiong C,Socher R. Seq2sql:Generating structured queries from natural language using reinforcement learning[J]. arXiv Preprint,2017,arXiv:1709. 00103.

[25] Wang C, Brockschmidt M, Singh R. Pointing out SQL queries from text[OL]. Microsoft Research, https://www. microsoft. com/en-us/research/publication/pointing-sql-queries-text/,2018.

[26] Iyer S,Konstas I,Cheung A,et al. Learning a neural semantic parser from user feedback[J]. arXiv Preprint, 2017,arXiv:1704. 08760.

[27] Suhr A,Iyer S,Artzi Y. Learning to map context-dependent sentences to executable formal queries[J]. arXiv Preprint,2018,arXiv:1804. 06868.

[28] Alvarez-Melis D,Jaakkola T S. Tree-structured decoding with doubly-recurrent neural networks[A]. ICLR 2017 Conference Submission[C]. ICLR Press,2016. 1 – 17.

[29] Xiao C,Dymetman M,Gardent C. Sequence-based structured prediction for semantic parsing[A]. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1:Long Papers)[C]. Association for Computational Linguistics Press, 2016. 1341 – 1350.

[30] 彭鑫,赵文耘,钱乐秋. 基于领域特征本体的构件语义描述和组装[J]. 电子学报,2006,34(S1):2473 – 2477.
PENG Xin,ZHAO Wen-yun,QIAN Le-qiu. Semantic representation and composition of business components based on domain feature ontology[J]. Acta Electronica Sinica, 2006,34(S1):2473 – 2477. (in Chinese)

[31] Yin P,Neubig G. A syntactic neural model for general-purpose code generation[J]. arXiv Preprint,2017,arXiv: 1704. 01696.

[32] 田野,袁博,李廷力. 物联网海量异构数据存储与共享策略研究[J]. 电子学报,2016,44(2):247 – 257.
TIAN Ye,YUAN Bo,LI Ting-li. Amassive and heterogeneous data storage and sharing strategy for internet of things[J]. Acta Electronica Sinica,2016,44(2):247 – 257. (in Chinese)

[33] Price P J. Evaluation of spoken language systems: The ATIS domain[A]. Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley,Pennsylvania[C]. USA:ACL Press,1990. 1 – 5.

[34] Tang L R,Mooney R J. Using multiple clause constructors in inductive logic programming for semantic parsing[A].

European Conference on Machine Learning[C]. Berlin, Heidelberg:Springer,2001. 466 – 477.

[35] Roy S B, De Cock M, Mandava V, et al. The Microsoft academic search dataset and KDD cup 2013[A]. Proceedings of the 2013 KDD Cup 2013 Workshop[C]. USA: ACM Press,2013. 1.

[36] 黄名选,蒋曹清. 基于完全加权正负关联模式挖掘的越-英跨语言查询译后扩展[J]. 电子学报,2018,46(12):3029 – 3036.
HUANG Ming-xuan, JIANG Cao-qing. Vietnamese-English cross language query post-translation expansion based on all-weighted positive and negative association patterns mining[J]. Acta Electronica Sinica,2018,46(12):3029 – 3036. (in Chinese)

[37] 张晓刚,杨路明,潘久辉. 面向数据集成的一种高效一致性查询方法[J]. 电子学报,2014,42(8):1474 – 1479.
ZHANG Xiao-gang, YANG Lu-ming, PAN Jiu-hui. An efficient consistent query answering method for data integration[J]. Acta Electronica Sinica,2014,42(8):1474 – 1479. (in Chinese)

[38] 李勇钢,崔超远,乌云,孙丙宇. 基于快速语义修复的操作系统隐藏对象检测技术[J]. 电子学报,2018,46(5):1025 – 1031.
LI Yong-gang, CUI Chao-yuan, WU Yun, SUN Bing-yu. The OS hidden object detection technology based on fast semantic repair[J]. Acta Electronica Sinica, 2018, 46(5):1025 – 1031. (in Chinese)

[39] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks[J]. Journal of Machine Learning Research,2010,9:249 – 256.

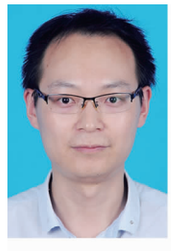
[40] Kingma D P, Ba J. Adam:A method for stochastic optimization[J]. arXiv Preprint,2014,arXiv:1412. 6980.

作者简介

李　青　女. 1989 年 5 月出生,陕西西安人. 现为重庆大学博士生,研究主要研究方向为自然语言处理、复杂事件检测、医学信息学.
E-mail:liqing@ cqu. edu. cn

钟　将(通信作者)　男. 1974 年 4 月出生,重庆江津人. 博士,现为重庆大学教授,研究主要研究方向为自然语言处理、数据挖掘研究.
E-mail:zhongjiang@ cqu. edu. cn

李立力　男. 1989 年出生,陕西铜川人. 现为重庆大学博士生,研究主要研究方向为桥梁健康监测、数据挖掘研究.
E-mail:lilili@ cqu. edu. cn



李　琪　男. 1987 年出生,江苏盱眙人. 博士,现为绍兴文理学院讲师,主要研究方向为图计算、数据挖掘.
E-mail:liqi0713@ foxmail. com

张淑芳　女. 1972 年出生,陕西人澄城人. 博士,现为重庆电子工程职业学院副教授,主要研究方向为图数据挖掘、高性能计算.
E-mail:roseymen2000@ foxmail. com

张　剑　男. 1972 年出生. 高级工程师,硕士,现为重庆西信天元数据资讯有限公司总经理. 主要研究方向为自然语言处理、数据挖掘.
E-mail:13608341660@ 139. com