软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2021,32(3):805−817 [doi: 10.13328/j.cnki.jos.006171]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于 Seq2Seq 模型的 SparQL 查询预测*

杨东华[1,2], 邹开发[2], 王宏志[2], 王金宝[2]

[1](哈尔滨工业大学 分析测试与计算中心,黑龙江 哈尔滨 150001)

[2](哈尔滨工业大学 计算学部,黑龙江 哈尔滨 150001)

通讯作者: 王金宝, E-mail: wangjinbao@hit.edu.cn

摘 要: 近年来,随着以数据为中心的应用大量增加,图数据模型逐渐被人们所关注,图数据库的发展也非常迅速,对于用户而言,往往更关心其在使用数据库过程中的效率问题.主要研究如何利用已有的信息进行图数据库的查询预测,从而进行数据的预加载与缓存,提高系统的响应效率.为了使得方法具有跨数据移植性,并深入挖掘数据间的联系,将 SparQL 查询提取为序列的形式,使用 Seq2Seq 模型对其进行数据分析和预测,并使用真实的数据集对方法进行测试,实验结果表明,本方案具有良好的效果.

关键词: 图数据库;SparQL;查询预测;Seq2Seq 模型

中图法分类号: TP311

中文引用格式: 杨东华,邹开发,王宏志,王金宝.基于 Seq2Seq 模型的 SparQL 查询预测.软件学报,2021,32(3):805−817. http://www.jos.org.cn/1000-9825/6171.htm

英文引用格式: Yang DH, Zou KF, Wang HZ, Wang JB. SparQL query prediction based on Seq2Seq model. Ruan Jian Xue Bao/ Journal of Software, 2021,32(3):805−817 (in Chinese). http://www.jos.org.cn/1000-9825/6171.htm

SparQL Query Prediction Based on Seq2Seq Model

YANG Dong-Hua[1,2], ZOU Kai-Fa[2], WANG Hong-Zhi[2], WANG Jin-Bao[2]

[1](Center of Analysis, Measurement and Computing, Harbin Institute of Technology, Harbin 150001, China)

[2](Faculty of Computing, Harbin Institute of Technology, Harbin 150001, China)

Abstract: In recent years, with the large increase in data-centric applications, graph data models have gradually attracted people's attention, and the development of graph databases is also very rapid. Users are often more concerned about their efficiency in using databases. This work mainly studies how to use the existing information to query and predict the graph database, so as to preload and cache the data, and improve the response efficiency of the system. In order to make the method cross-data portable and dig deep into the connections between the data, this study extracted SparQL queries into the form of sequences, used the Seq2Seq model to analyze and predict its data, and tested the method using real data sets. Experiments show that the proposed scheme in this study has a sound effect.

Key words: graph database; SparQL; query prediction; Seq2Seq model

近年来,支撑人工智能的数据管理和分析技术已成为当前大数据与人工智能领域的研究热点,在图数据库领域也是如此.利用图数据库中的数据管理与分析技术,可以有效促进人工智能领域的发展.例如:在类似于微信或者 Facebook 这样的社交网络中,存在着大量用户与用户之间的关系,适合用图数据模型进行存储.在商业领域,公司控股关系、法人关系等也构成了一个庞大的网络,这些数据更适合使用图数据库进行存储,并借助人工智能,利用数据管理与分析技术为用户、公司等提供更多的帮助.

* 基金项目: 国家自然科学基金(61772157, 61832003, U1866602, 61602129)
Foundation item: National Natural Science Foundation of China (61772157, 61832003, U1866602, 61602129)
本文由"支撑人工智能的数据管理与分析技术"专刊特约编辑陈雷教授、王宏志教授、童咏昕教授、高宏教授推荐.
收稿时间: 2020-06-10; 修改时间: 2020-09-03; 采用时间: 2020-11-06; jos 在线出版时间: 2021-01-21

查询预测是图数据库领域支撑人工智能的技术之一,结合各种人工智能方法,利用已有的查询对用户未来的查询进行预测,然后进行数据预加载,从而提高图数据库的响应效率.这些方法对人工智能本身的发展起到了推动作用.

以 SparQL 查询为例,目前已有一些研究人员提出了 SparQL 查询预测的算法.这些算法往往分为两类.

- 一类方法如文献[1−3],从图数据库读取数据源的信息,然后根据这些信息来对用户可能的查询进行预测.这样的方法能够结合数据信息,但也使得算法不具有跨数据移植性.当更换了其他数据源之后,可能会因为某些信息的缺失而使算法无法正确工作;
- 另一类方法则通常从图数据库的工作日志中获取用户的历史查询的信息,并计算这些历史查询间的相似度来预测下一个可能的相似的 SparQL 查询.虽然具有了跨数据移植性,但仍然具有局限性.文献[4,5]中使用图模式匹配的算法要求历史查询必须存在重复性,才可以匹配到连续相似的查询以创建模板;文献[6,7]中使用相似度的算法,由于独立计算三元组的相似度导致三元组之间的联系没有被考虑到,影响了预测的准确率.

使用图模式匹配的方法仅仅考虑相似的查询,而不考虑查询间的联系;而利用三元组的相似度进行计算,则将查询本身进行细化了,没有考虑到同一个查询内三元组间的影响.因此,本文将查询转化为序列,不同的查询序列进行连接,将查询内部三元组的联系以及查询间的联系信息存储于序列之中,进而本文使用 Seq2Seq 模型处理这些序列数据,提出了一种基于 Seq2Seq 模型的 SparQL 查询预测算法.本文的主要研究重点在于如何将用户的查询转化为可计算的特征向量,合适的特征向量能够提高 Seq2Seq 模型的学习效果,以及如何对模型的结构进行优化也是需要解决的问题.

本文第 1 节主要回顾目前图数据库以及 SparQL 查询预测的研究成果.第 2 节给出本文所需的背景知识.第 3 节介绍预测算法的具体流程.第 4 节则是利用 USEWOD 数据集对算法进行测试,并进行结果分析和讨论.第 5 节总结本文的主要工作和对未来工作的展望.

1 相关工作

1.1 图数据库

近些年,图数据库的发展迅速,一方面,Neo4j,OriendDB,ArangoDB 等数据库仍然广泛使用;另一方面,研究者们也提出了一些新的图数据库,如 TigerGraph[8],SeQuery[9],GraphSE2[10],Graphflow[11]等,它们往往采用了不同的解决方案,从而解决不同的实际应用问题.文献[12]分析了 Neo4j,AllegroGraph,ArangoDB,InfiniteGraph,OrientDB 等流行图数据库的特点,从存储结构、易用性、性能等方面对各个数据库做了介绍,并指出了对于图数据库最重要的几个特性(灵活的结构、支持的查询语言、分片、备份、多模型、多架构、可扩展性、云读取),并从这几个维度为上述图数据库进行了评分.文献[13]则是比较了同一数据模型在关系数据库和图数据库上的存储,证明了图数据库在图查询和可视化上的优势.

图数据库的应用也变得越来越广泛.文献[14]基于图数据库实现了一种位置图模型,用于根据场所描述信息对其进行建模,在这个基础上,实现了地理位置匹配、推理与查询功能.文献[15]则基于 Neo4j 图数据库构建了煤矿领域的知识图谱,同时,设计并开发了煤矿监测监控原型系统,体现了图数据库在知识图谱领域的应用.文献[16]则是以 Neo4j 作为后端实现了一个图查询系统,并比较了 Cypher,Gremlin,Java 作为查询语言的方案,比较了几种方案的优缺点.

1.2 SparQL查询预测

SparQL 查询预测指利用已有的信息对用户接下来可能发出的 SparQL 查询进行预测,目前主要有两种方法. (1) 根据 RDF 数据中的信息对 SparQL 查询进行预测;(2) 根据用户的历史 SparQL 查询进行预测.

第 1 类方法相对传统,文献[1]使用本体元数据的信息进行下一步的查询预测.相比之下,文献[2]则提出了一种新的方法,该方法使用一张预先计算好的属性值相似表进行查询预测.文献[3]则利用从知识图谱构建出的语

言模型来实现查询预测.这些方法通常预先进行信息计算,然后进行查询预测.但是其共同点是需要从数据中获取信息,这意味着它们构建出的方法/模型不是跨数据源可用的.因为不同的数据源信息不同,也可能存在缺失,因此这些方法存在很大的局限性.

另一类方法则是利用图数据库的历史查询日志来做查询预测,这样的方法具有较强的数据移植性,因为它们通常是通过分析查询而不是数据本身.文献[4]检测历史查询中所存在的重复模式,然后使用一种自下而上的图模式匹配算法进行查询模板的创建,从而实现查询预测.文献[5]则是在此基础上将这些查询模板与不同的策略相结合,来进一步扩展可能的查询,但是并没有得到结论性的结果.一些方法使用相似性来进行查询预测,如文献[6]使用图编辑距离来计算不同的 SparQL 查询之间的相似性,从而使用相似的查询构建新的查询;文献[7]则利用 SparQL 的三元组的主语、谓语和宾语间的相似性得到三元组的相似性,进而计算查询的相似性,从而进行查询预测.

第 2 种方法虽然具有较好的数据移植性,但是无论是基于模板的方法还是基于相似性的方法,都只能使用历史查询中的相似查询,如果查询呈现周期性的波动,则很难进行预测.另一方面,SparQL 查询的长度是不确定的,而这样的方法导致新的查询的长度必然等于这些相似的历史查询,因此这一问题也需要解决.

2 背景知识

本文旨在根据图数据库所存储的数据以及其历史工作负载等信息,对工作负载、数据特征等进行有效的统计,并设计合理的方案提取出相关特征;根据这些特征,进行对用户可能的查询的预测,从而可以对用户需要的数据进行预加载以及进行合适的查询优化.

为了较为方便地提取工作负载以及进行预测,本文主要研究的数据对象为关联数据.关联数据是语义网的主题之一,描述了通过可链接的 URI 方式来发布、分析、连接 Web 中各类资源的方法.关联数据通常以资源描述框架(resource description framework,简称 RDF)的形式进行描述,而查询 RDF 使用的语言是 SparQL(SPARQL protocol and RDF query language).

定义 1(SparQL 图模式). 一个 SparQL 查询通常可以被表示为一个图结构,定义符号 B 为空白节点,I 表示国际化资源标识符(internationalized resource identifier,简称 IRI),L 表示字面量,V 则表示变量,那么一个 SparQL 图的图模式通常可以如下递归定义[17].

(1) 一个有效的三元组 $T\in(IVB)\times(IV)\times(IVLB)$是一个基本的图模式(basic graph pattern,简称 BGP),三个元素分别被称之为主语、谓语和宾语;

(2) 对于基本的图模式 BGP_i 和 BGP_j,其连接(BGP_i and BGP_j)也是一个 BGP;

(3) 如果 P_i 和 P_j 是图模式,那么(P_i and P_j),(P_i *union* P_j)以及(P_i *optional* P_j)也是图模式;

(4) 如果 P_i 是一个图模式,而 R_i 是一个 SparQL 的内建表达式(如 *lang*(?*name*)=“*en*”表示限定 *name* 变量的语言是英语),那么表达式(P_i *filter* R_i)是一个图模式.

可以从数据库中获取用户的若干个历史 SparQL 查询,进而对用户接下来的查询进行预测.据此,给出 SparQL 查询预测问题的定义.

定义 2(SparQL 查询预测问题). 符号 N 表示用户查询的个数,$Q_1,Q_2,\ldots,Q_n$ 表示用户最新发出的 N 个连续的 SparQL 查询,Q 表示用户接下来的 SparQL 查询,则 SparQL 查询预测问题即在给定 $Q_1,Q_2,\ldots,Q_n$ 的情况下,对 Q 进行预测.

此外,本文在研究中使用到了 Seq2Seq 模型以及注意力机制和集束搜索,在此给出介绍.

2.1 Seq2Seq模型

本研究中将 SparQL 查询转化为序列,因此将问题变成了一个序列到序列的问题.而文献[18]提出的 Seq2Seq 模型通常被用以处理序列到序列的任务,Seq2Seq 模型使用两个循环神经网络(recurrent neural networks,简称 RNN)[19]对序列进行处理和输出,RNN 是一种隐藏层互相连接的神经网络,被广泛应用于处理序列数据,其计算流程见公式(1)、公式(2).

$$h_t=\sigma_h(W_h x_t+U_h h_{t-1}+b_h) \tag{1}$$

$$y_t=\sigma_y(W_y h_t+b_y) \tag{2}$$

σ_h与σ_y分别表示隐藏层与输出层的激活函数,W,U和b均为对应的参数,h_t表示t时刻的隐藏层状态,y_t表示t时刻的输出.

Seq2Seq 模型则使用了两个 RNN 进行处理,分别称之为编码器 Encoder 与解码器 Decoder.Encoder 读取输入信息,将学习到的信息存储于公式(1)中计算出的隐藏层状态h_t中;Decoder 则读取最终时刻的 Encoder 的隐藏层状态作为自己的初始隐藏层状态,并不断进行迭代产生输出序列,从而完成从序列到序列的任务.

2.2 注意力机制

注意力机制希望模型对输入具有“注意力”[20].具体地,解码器应当在输出的不同时间步都对编码器的每个状态(对应输入序列的每个时间步)给予不同的权重,于是在每个输出时刻,都可以通过这些不同的权重对编码器的隐藏层信息进行加权,从而得到一个随着时间步变化的上下文信息,该上下文信息相当于模型在当前时间步进行输出时,关注不同的输入而得到了不同的信息.将这些上下文信息作为对应时间步的输入之一,增强解码器输出时的判断能力.这些权重采取随机初始化的方式,在 Seq2Seq 模型训练时不停调整,以获取更丰富的信息.

2.3 集束搜索

集束搜索(beam search)是一种动态规划算法[21].集束搜索只应用于 Seq2Seq 模型的预测时:在当前时间步进行输出时,不再选择概率最高的结果进行输出,而是保留概率最高的K个;而在下一个时间步进行输出时,在计算概率时不再单纯计算输出某个元素的概率,而是计算从第 1 个时间步到当前时间步的所有输出形成的输出序列的联合概率,从而可以根据联合概率对当前时间步的输出进行排序,同样地,再次保留概率最高的 K 个作为当前时间步的输出给予下一个时间步使用.

使用集束搜索机制,一方面可以降低单个时间步预测出错时所带来的误差,因此可能在下一个时间步正确的输出会获得更高的概率;另一方面,在每个时间步都计算整体的输出序列的概率而不是单个时间步的输出元素的概率,更好地考虑了序列元素之间的相关性.

3 SparQL 查询预测算法

本研究从图数据库的工作日志中提取出用户的 SparQL 查询信息,然后进行查询预测.这样做的优点是具有跨数据的移植性,即使更换了数据源仍然适用.已有的这类研究通常使用图匹配或者相似度的方法,这意味着用户过去的历史查询需要保持相似,当其发生周期性的变化时,无论是图匹配还是相似度,都无法找到这种周期性的规律;另一方面,基于相似度的方法通常利用历史查询中相似的三元组构建预测查询中的三元组,而没有利用到三元组之间的信息.

本文提出的基于 Seq2Seq 模型的 SparQL 查询预测算法将定义 1 中的图模式转化为序列,利用 Seq2Seq 模型挖掘序列内部元素的关联性,保证利用了三元组之间的信息;同时,将若干个历史查询连接为同一个序列,从而使得模型可以学习到查询间的规律性,降低了对于查询连续相似的依赖.

想要从序列中学习到丰富的信息,那么就需要将文本形式的 SparQL 查询转化为可计算的特征向量,这部分将在第 3.1 节中介绍;在得到特征向量之后,如何利用 Seq2Seq 模型进行查询预测以及如何进行模型优化将在第 3.2 节中介绍.

3.1 特征转化

3.1.1 特征转化的要求

从查询转化到的特征向量将用于查询预测任务,因此对特征向量的要求较高,也是本文着重解决的问题,具体地,特征转化的要求主要有如下几点.

(1) 信息完备性

所得到的特征应当尽可能地囊括原始数据,也就是 SparQL 图模式中所蕴含的信息,要能够恰好地表达出三

元组所包含的信息的同时,也要能够表达三元组间的联系,不可以将三元组独立拆分,这也是现有的大多数方法所不具备的.

(2) 还原性

一些提取 SparQL 特征向量的工作,如文献[22],将 SparQL 查询转化为语法树的形式,然后通过将其中的一些关键信息(如 UNION 操作的个数等)进行记录,然后转化为特征向量.但这样所得到的特征向量并不具备还原性,即无法从特征向量还原到 SparQL 查询或者图模式,那么就无法进行查询的相关预测.因此,对特征转化的要求包括需要能够从特征向量转化为 SparQL 图模式,从而判断查询预测的准确性等.

(3) 泛化能力

查询预测是根据用户的历史查询去预测用户下一步的查询,而事实上,用户下一步的查询中可能会出现历史查询中所没有出现过的变量或其他信息,因此在提取特征时,便不能过度依赖于三元组字段的内容本身,否则当用户发出新的陌生的 SparQL 查询时将无法转化,故泛化能力是指需要能够预测训练时历史查询中所不包含的三元组或变量等信息.

3.1.2 特征转化的设计思想

基于第 3.1.1 节中所列出的 3 点要求,本文将通过如下的设计思想进行特征转化方案的设计.

(1) 将三元组视为整体,查询视为一个序列

三元组包括由主语、谓语和宾语构成,而这三者之间往往存在一定联系,如果将三元组的信息拆分进行表示,那么将会破坏其整体性,无法较好地表达其三元组本身的信息,因此需要将三元组视为一个整体进行对待.其次则是将查询整体视为序列,从而包含三元组之间的关联.如果以符号 T 表示三元组,那么一个 SaprQL 查询的图模式部分的例子通常类似于:“{{$T_1T_2T_3$} UNION {{T_4T_5} AND {T_6}} OPTIONAL {T_7}}”.从这个例子中可以看到:除了三元组之外,图模式中还包括了 UNION,AND 和 OPTIONAL 这类 SparQL 操作符以及括号等符号.事实上,无论是三元组还是这些操作符,都在图模式中表达着 SparQL 的语义信息,甚至于左右大括号这类符号也传递着嵌套信息等语义信息.因此,若想要完整的表示 SparQL 图模式,就必须将所有的这些语义信息进行尽可能地充分的表达.因此,本文借鉴了自然语言处理领域对于类似信息的处理方式,将 SparQL 图模式视为一个符号序列,无论是三元组还是操作符,抑或是括号等其他符号,均视为该符号序列中的一部分进行单独表示,从而将整个序列的语义信息进行充分的表达,也能使三元组之间的联系信息存储在序列之中.

(2) 以位置信息表达三元组

本研究使用一组查询中“第 N 个三元组”的形式定义三元组的编号,即:其在序列中的位置决定了其编号,而不是使用内容信息.如果以三元组本身的内容进行信息表达,那么对于未知内容的三元组,则无法提前了解其信息,也就无法进行表达了,故而这种方法将不再具有泛化能力.本文的这种形式判断三元组在这一组查询中所出现的次序,以此进行三元组的编号,在已经使用序列表达整个 SparQL 图模式的情况下,对于相似的序列模式,其所包含的三元组的位置信息也是相似的,即使在用户的新查询中包含了训练时未曾出现过的三元组,但是其位置信息往往已经出现过,从而可以以位置信息匹配曾经出现过的模式.

另一方面,这样做带来的更大好处是可以避免产生内容偏好,即不同用户所发出的 SparQL 查询在内容上是不同的,而内容上的不同并不能称之为一个有用的信息,因为往往不同用户所发出的查询可能形式是相同的,但是内容完全不同,使用内容表达会减少所能学习到的信息,而使用位置信息表达则可以很好地解决这一问题.

(3) 对三元组使用等价类划分

以三元组“?sub rdf:person_name “kaily”@en”和“?sub rdf:person_name “jack”@en”为例,这两个三元组均是为了查询特定姓名的实体,其差异仅仅在于谓语不同.事实上,在大量观察数据记录后,可以发现这样的情况占据大多数,有时不同的 SparQL 图模式的差异仅仅在于这类三元组的差异,而字面量的变化对于模型或者算法的意义是不大的,这并不能成为有效的信息.因此,本文对三元组进行等价类的划分以解决该问题,这里给出等价三元组的定义.

定义 3(三元组等价类). 对于三元组 A 与三元组 B,若 A 与 B 的主语和谓语字段均相同,而 A 与 B 的宾语字

段不同,且 *A* 与 *B* 的宾语字段均为字面量(以字符串表达的值),则称 *A* 与 *B* 是等价的三元组.

一方面,等价的三元组可以查询到所有这一类三元组的信息,从而减少模型或者算法所需要学习的冗余信息;另一方面,这样的处理方式同样可以降低训练时未出现的三元组所带来的影响,因为它们通常也可以被归纳为某个等价类三元组.

3.1.3 特征转化的具体流程

基于第 3.1.1 节中的要求以及第 3.1.2 节中特征转化的设计思想,现给出特征转化的具体流程.

对于一个 SparQL 查询的图模式,将其视为一个序列,序列中的每个元素均使用 One-Hot(即对于一个 *D* 维向量,使用第 *K* 维为 1 且其他维均为 0 的方式表示数字 *K*)的形式进行表示,因此需要定义该向量的长度.

使用符号 *HIS_LEN* 表示预测时使用的历史查询的个数,则 *HIS_LEN*+1 个查询为一组数据(最后一个为要预测的查询),并定义在一组 SparQL 查询的图模式中最多可能出现的三元组的数量为 MAX_*TRIPLE*,而除了三元组之外,还需要被转化的符号和操作符总共有 7 个,分别为 AND、UNION、OPTIONAL、{、}、起始符号 Start 和结束符号 End,其中,Start 和 End 分别用于表示一个序列的起始和结束,从而方便算法或者模型进行预测.

由此,序列中的每个元素所对应的是一个(MAX_*TRIPLE*+7)维的 One-Hot 形式的向量,以上 7 个符号分别定义为 0~6,三元组则按照这组查询中的“第 *N* 个三元组”的形式以数字 *N*+7 所对应的 One-Hot 向量进行表示.算法 1 为单个元素转化的具体流程伪代码.

算法 1. 特征转化的流程.

输入:*queries* 查询的数组;

输出:*vectors* 查询数组对应的向量数组.

```
1:  初始化空数组 vectors;
2:  初始化空数组 querySet;
3:  for query in queries do
4:      for symbol in query do
5:          初始化数组 vector,长度为(MAX_TRIPLES+7),内容均为 0;
6:          if symbol in [“AND”,“UNION”,“OPTIONAL”,“{”,“}”] then
7:              查找 symbol 对应的符号索引 index;
8:              vector[index]=1;
9:          else
10:             if query in querySet then
11:                 vector[querySet.indexOf(query)+7]=1;
12:             else if querySet 中某个 q 与 query 等价  then
13:                 vector[querySet.indexOf(q)+7]=1;
14:             else
15:                 querySet.add(query);
16:                 vector[querySet.indexOf(query)+7]=1;
17:             end if
18:         end if
19:     end for
20:     vectors.add(vector)
21: end for
22: return vectors
```

算法使用 *querySet* 数组控制等价的三元组,第 10 行~第 16 行中,对于每个三元组,算法首先寻找是否已经存在相同的或者等价的三元组,如果有则使用其索引,否则才创建新的索引;第 6 行~第 8 行中,对于一般符号的处

理,则是直接使用查表法寻找对应的索引,将 One-Hot 向量的对应位置置 1.假设查询序列的平均长度为 m,总共有 n 个查询,算法对于每个查询仅遍历其序列一次,因此算法的时间复杂度为 $O(nm)$.

对于一个 SparQL 图模式对应的序列中的所有元素,均采取上面的方式进行转化,最终可以得到一个长度不定的 One-Hot 向量序列,作为特征提取阶段的最终产出.

3.2 查询预测

查询预测阶段使用 Seq2Seq 模型对第 3.1 节中所提取到的特征向量进行学习,但直接将 One-Hot 向量序列输入进模型的方法并不合理,因为 One-Hot 向量只能简单地表达类别信息,其维度较低,而低维语义空间所蕴含的语义信息显然不如高维空间丰富.如果能够将这一低维度的 One-Hot 向量映射为一个高维向量,那么特征向量可以借助高维的语义空间向 Seq2Seq 模型提供更加丰富的语义信息.

具体地,定义新的维度为 D,D 通常要比 MAX_*TRIPLE* 大很多,如 1 024,太大的数字可能会导致内存等资源不足,因此需要权衡.在初始化 Seq2Seq 模型时,设置一个映射函数,将所有 MAX_*TRIPLE*+7 种向量映射为一个 D 维的高维向量.对于这些高维向量的设定,采取类似于编码器隐藏层初始状态的处理方式,即对其进行随机初始化,因为人为地指定这样的维度的内容是近似于不可能的,因此这些向量设定为可训练的.在 Seq2Seq 模型的训练过程中,不断地根据反向传播调整这些向量,这样也可以使得那些表示三元组的向量更具位置上的语义性.

通过特征向量升维,可以大幅提升 Seq2Seq 模型能够从输入中学习到的信息,从而提高模型的预测能力.

另一方面,在预测用户的下一步可能发出的查询时,通常要使用到不止一个历史查询,而单个 SparQL 图模式转化得到的序列长度通常在 30~50 左右,这意味着输入的序列的长度是较高的.另一方面,输出时模型也不应当考虑到输入序列的所有内容,譬如在输出某个三元组时,模型在这一时刻应当关注于输入序列中同样是三元组的那些信息而不是操作符等符号,因为输入序列中的三元组更能在输出三元组时表达更多的信息.因此,本文为 Seq2Seq 模型引入了第 2.2 节中的注意力机制,强调不同的输出时刻对输入给予不同的权重.

此外,为了降低单个时间步预测出错的影响,利用整体序列的概率替代单个预测的概率.本文在此基础之上再引入了第 2.3 节中的集束搜索机制,对 Seq2Seq 模型进行优化.

以简单 SparQL 查询的图模式“{T_1 UNION T_2}”为样例输入(即只使用一个历史查询),假定输出为“{T_3 UNION T_4}”,图 1 表示了该模型的工作流程,其中,⟨start⟩和⟨eos⟩分别表示起始符号 Start 与结束符号 End 对应的 One-Hot 向量在使用特征升维后对应的向量.

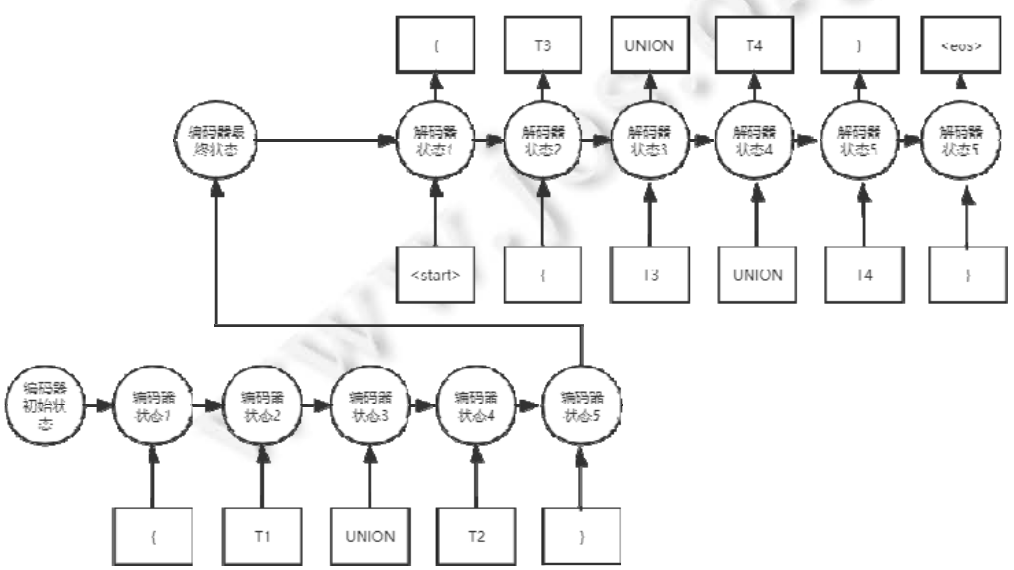


Fig.1 Process of query prediction using Seq2Seq model

图 1 使用 Seq2Seq 模型进行查询预测的流程

编码器接受不定长的输入,并维护编码器状态.编码器状态即保存了输入序列中的信息,当输入处理完毕之后,此时编码器的状态作为解码器的初始状态,解码器读取状态中的信息,并开始产生预测序列.

现有的基于历史查询进行预测的算法通常对历史查询的要求较高,如果历史查询并不连续相似,则无法匹配到相应的图模式,也无法利用查询间的相似度进行判断.而实际上,真实的数据中有些用户的查询是呈现周期性波动的.本文使用 Seq2Seq 模型不仅利用到了查询内三元组间的信息,还利用到了不同查询间的信息,即使历史查询并不相似,也可以捕获到查询间的联系,从而提升了预测任务的准确率;另一方面,模型可以判断输出的停止时间,因此可以输出不定长度的序列.

4 实验结果与分析

4.1 USEWOD数据集

本文使用了 USEWOD2016 数据集,该数据集存储了 DBPedia 等知识库的 SparQL 查询日志,其中记录了用户在 DBPedia 的 SparQL 查询接口中所发出的 SparQL 查询以及用户 ID 和发出查询的时间.

图 2 为 USEWOD2016 数据集中的内容示例,图中共 3 项数据记录,每一行的第 1 个字段为用户的 ID,可以看到,这 3 条查询请求均来自于同一用户;第 2 个字段则是时间字段,表明用户发出该查询的具体时间;时间字段之后则是用户通过 Web 客户端所发出的 SparQL 查询的 HTTP 请求格式,记录了用户所发出的 SparQL 查询的相关信息.

```
e075d098bc0412dbc3b86e2474a76441 - - [17/Aug/2015 03:00:00 +0200] "GET /sparql?query=ASK%0AWHERE%0A++%7
B+++%7B+%3Chttp%3A%2F%2Fdbpedia.org%2Fresource%2FHimara_revolt_of_1912%3E+%3Chttp%3A%2F%2Fwww.w3.org%2F
1999%2F02%2F22-rdf-syntax-ns%23type%3E+%3Chttp%3A%2F%2Fxmlns.com%2Ffoaf%2F0.1%2FPerson%3E%7D%0A++++UNIO
N%0A++++++%7B+%3Chttp%3A%2F%2Fdbpedia.org%2Fresource%2FHimara_revolt_of_1912%3E+%3Chttp%3A%2F%2Fwww.w3.
org%2F1999%2F02%2F22-rdf-syntax-ns%23type%3E+%3Chttp%3A%2F%2Fdbpedia.org%2Fontology%2FPerson%3E%7D%0A++
%7D%0A HTTP/1.1" 200 44 "-" "R" "-"

e075d098bc0412dbc3b86e2474a76441 - - [17/Aug/2015 03:00:00 +0200] "GET /sparql?query=ASK%0AWHERE%0A++%7
B+++%7B+%3Chttp%3A%2F%2Fdbpedia.org%2Fresource%2FQarah_Zia_od_Din%3E+%3Chttp%3A%2F%2Fwww.w3.org%2F1999%
2F02%2F22-rdf-syntax-ns%23type%3E+%3Chttp%3A%2F%2Fxmlns.com%2Ffoaf%2F0.1%2FPerson%3E%7D%0A++++UNION%0A+
++++++%7B+%3Chttp%3A%2F%2Fdbpedia.org%2Fresource%2FQarah_Zia_od_Din%3E+%3Chttp%3A%2F%2Fwww.w3.org%2F1999
%2F02%2F22-rdf-syntax-ns%23type%3E+%3Chttp%3A%2F%2Fdbpedia.org%2Fontology%2FPerson%3E%7D%0A++%7D%0A HTT
P/1.1" 200 44 "-" "R" "-"

e075d098bc0412dbc3b86e2474a76441 - - [17/Aug/2015 03:00:00 +0200] "GET /sparql?query=ASK%0AWHERE%0A++%7
B+++%7B+%3Chttp%3A%2F%2Fdbpedia.org%2Fresource%2FPiero_Ceccarini%3E+%3Chttp%3A%2F%2Fwww.w3.org%2F1999%2
F02%2F22-rdf-syntax-ns%23type%3E+%3Chttp%3A%2F%2Fxmlns.com%2Ffoaf%2F0.1%2FPerson%3E%7D%0A++++UNION%0A++
++++%7B+%3Chttp%3A%2F%2Fdbpedia.org%2Fresource%2FPiero_Ceccarini%3E+%3Chttp%3A%2F%2Fwww.w3.org%2F1999%2
F02%2F22-rdf-syntax-ns%23type%3E+%3Chttp%3A%2F%2Fdbpedia.org%2Fontology%2FPerson%3E%7D%0A++%7D%0A HTTP/
1.1" 200 43 "-" "R" "-"
```

Fig.2 Example of USEWOD2016 dataset

图 2 USEWOD2016 数据集示例

4.2 评价指标

为了更加全面地评价算法的预测效果,本文定义 3 个指标对预测结果进行评价.

(1) 完全预测准确率

该指标作为最严格的评价指标,即当且仅当算法所做出的预测与用户所发出的真实预测完全相同(序列长度相同且内容相同)时,视为模型的预测是完全正确的,完全预测准确率则等于完全预测正确的数据量与所有数据量的比例.

(2) 完全缓存命中率

本文进行查询预测的目的是为了提前对数据进行预加载,即做预缓存,那么数据的缓存命中率也是重要的评价指标.因此,本文定义:若用户的查询中的三元组均被算法预测的查询所覆盖,则视为完全命中缓存,完全缓存命中率则等于完全命中缓存的数据量与所有数据量的比例.

(3) 平均缓存百分比

即使没有完全命中缓存,对其中一部分数据进行预加载也能够提高数据库对于用户所发出查询请求的响应速度,因此,该评价指标计算用户的查询中的三元组被算法预测的查询所覆盖的比例,并在所有数据中计算该比例的平均值.

使用以上 3 个指标,可以更全面地评价算法的预测效果.

4.3 实验设计

首先给定本文在实验中设定的一些参数,MAX_*TRIPLE* 值设定为 32,因此 One-Hot 向量的维度为 39,特征向量升维后的维度 D 为 1 024,循环神经网络的隐藏状态层的维度为 1 024,集束搜索的搜索个数 K 为 5.除此之外,在一些方面设计了对比实验,具体如下:

(1) 历史查询个数,观察使用的历史查询个数对算法结果的影响;

(2) 注意力机制,观察注意力机制的是否使用对算法结果的影响;

(3) 集束搜索,观察集束搜索的是否使用对算法结果的影响.

4.4 实验结果与分析

(1) 关于历史查询个数的实验

在关于历史查询个数的对比实验中,*HIS_LEN* 参数分别被设定为 1~5,训练数据设定为 60 822 组,注意力机制被使用,不使用集束搜索方法,测试数据设定为 3 000 组,图 3 表明了历史查询个数对实验结果的影响,其中,横轴为 *HIS_LEN* 参数,纵轴为 3 个评价指标的数值,实线、虚线、点状线分别对应第 5.2 节中的评价指标(1)~(3).

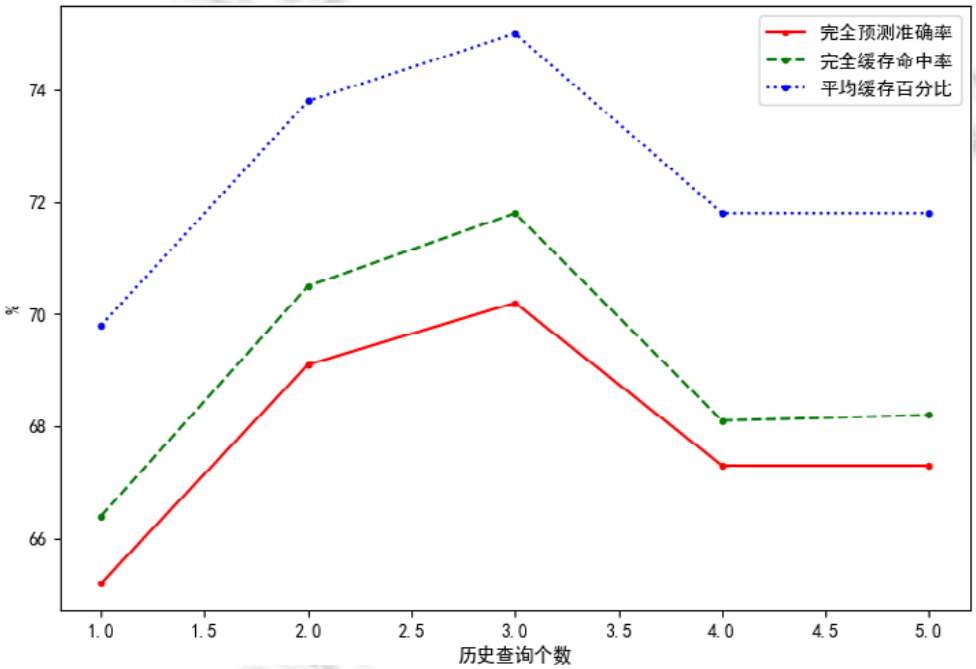


Fig.3 Impact of the number of historical queries on results

图 3 历史查询个数对实验结果的影响

可以看到:3 个评价指标的变化趋势是基本相同的;并且指标的评价标准越严格的情况下,其数值则越低.值得注意的是:当历史查询个数的值为 3 个的时候,准确率(这里准确率指代所有指标)达到极值;并且当历史查询个数由 2 变为 3 时,准确率的提升小于由 1 变为 2 时的提升;而当历史查询个数变得更多时,准确率开始下降.这是因为虽然循环神经网络主要是用以处理不定长的序列数据,但是其处理长期依赖的能力也是有限的,较长的

输入序列仍然会使得循环神经网络的记忆能力降低.当历史查询个数为 3 时,Seq2Seq 模型所处理的输入序列的总长度可以达到 150,这几乎已经快超出了循环神经网络的处理能力,因此当历史查询个数再次增加,输入序列的总长度再次增加时,Seq2Seq 模型陷入了欠拟合的状态,因此准确率开始下降.而当历史查询个数较低时,特别是当历史查询个数只有一个时,对于模型来说,此时其是无法从历史查询中判断该段序列中所存在的模式与规律的,输入序列所带来的信息太少,因此准确率要比历史查询个数较多时更低一些.因此在本文的实验中,历史查询个数为 3 是最为合适的值.

(2) 关于注意力机制的实验

在关于注意力机制的对比实验中,*HIS_LEN* 参数分别被设定为 1~5,训练数据设定为 60 822 组,不使用集束搜索方法,测试数据设定为 3 000 组,图 4 表明了注意力机制对实验结果的影响,其中,横轴为 *HIS_LEN* 参数,纵轴为 3 个评价指标的数值,图 4(a)~图 4(c)分别对应第 5.2 节中的评价指标(1)~(3),实线为使用注意力机制,虚线为未使用注意力机制.

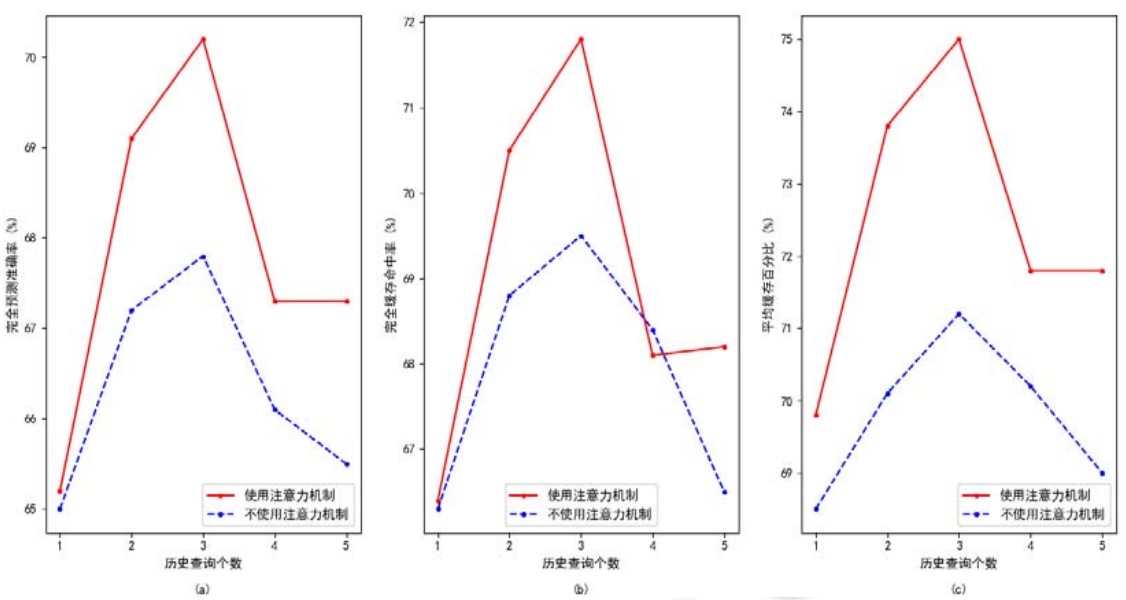


Fig.4 Impact of attention mechanism on results

图 4 注意力机制对结果的影响

可以看到,3 组曲线的变化趋势与对比基本是相同的.在不使用注意力机制的情况下,3 种评价指标均有不同幅度的下降,表明注意力机制能够提升模型的预测效果.此外,当历史查询个数较低,尤其是只有一个的时候,注意力机制对于准确率的提升并不大.这是因为此时输入中只有一个历史查询,输入信息较少,注意力机制并不能发挥其效果,在不同时刻关注不同的输入,因此对于模型没有太多的提升;而在历史查询个数较多的时候,模型准确率的下降也得到了一定的改善,因为注意力机制可以使得模型从较多的查询序列中关注其中更有价值的信息,从而提升预测的效果.

(3) 关于集束搜索的实验

在关于集束搜索的对比实验中,*HIS_LEN* 参数分别被设定为 1~5,训练数据设定为 60 822 组,注意力机制被使用,测试数据设定为 3 000 组.图 5 表明了历史查询个数对实验结果的影响,其中,横轴为 *HIS_LEN* 参数,纵轴为 3 个评价指标的数值,图 5(a)~图 5(c)分别对应第 5.2 节中的评价指标(1)~(3),实线为使用集束搜索,虚线为未使用集束搜索.

相比注意力机制,集束搜索对于模型准确率的提升更为明显一些.当历史查询个数增大到 3 以上时,准确率

的下降明显变缓.因为即使某个时刻的预测是出错的,正确的输出也可能属于概率较高的输出之一而被集束搜索所保留,并在下一时刻通过整体的序列的概率再次利用到正确的输出进行判断.这使得即使模型因为序列长度的提升降低了记忆能力,但仍然通过这种机制罗列出模型学习到的各种序列模式,并挑选出概率最高的序列进行输出.因此,集束搜索能够较好地提升历史查询个数较多时候的效果.但历史查询个数较少时,整体序列的特点并未被模型所过多地学习,因此集束搜索机制也无法使得这种情况下的正确率变得较高.

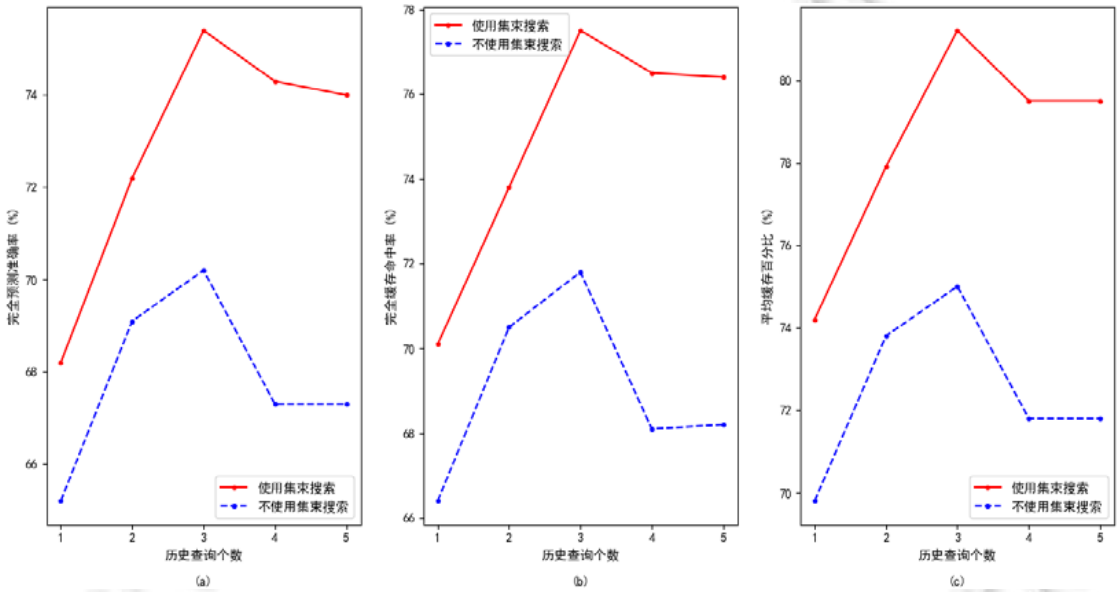


Fig.5 Impact of beam search on results

图 5 集束搜索对结果的影响

综合以上 3 组实验可以发现:在已有的实验参数下,最合适的历史查询个数为 3.注意力机制与集束搜索均被使用,本文测定了此设定下的指标,完全预测准确率达到 77.3%,完全缓存命中率达到 80.1%,平均缓存百分比达到 82.5%,总体达到了较为不错的水平.

5 结 论

本文从 RDF 图数据库的查询日志中提取出 SparQL 查询,并进一步将 SparQL 查询提取为 SparQL 图模式,进行可还原的特征转化,得到了蕴含丰富信息的特征向量.为了能够从特征向量里充分挖掘序列之间的关联性,本文使用了 Seq2Seq 模型进行 SparQL 图模式的预测,Seq2Seq 模型可以处理不定长的输入与输出.在此基础上,利用注意力机制与集束搜索对模型进行优化,使得模型的完全预测准确率达到 77.3%,完全缓存命中率达到 80.1%,平均缓存百分比达到 82.5%.

本文所使用的方法一方面避免了收集数据源的信息,从而使得方法具有跨数据移植性,并且利用位置信息代替内容信息之后,也使得该方法不会因为查询的数据不同而失效,因此具有良好的跨数据移植性;另一方面,使用 Seq2Seq 模型充分挖掘了数据间的关联性与规律,因此即使用户的 SparQL 查询呈现周期性变化,也可以进行预测,此外也利用了循环神经网络的特性,较好地解决了 SparQL 查询长度不确定的问题.

本文现有的工作中,所进行特征转化的主要是 SparQL 图模式中的三元组以及 UNION,AND 和 OPTIONAL 操作符.实际上,SparQL 查询中还存在 FILTER,LIMIT 等操作符,这些操作符是对数据的进一步过滤,与 UNION,AND 等操作符的语义信息不同,且这些操作符以及其携带的表达式信息在转化时很难保持还原性.因此,未来的工作将重点研究如何对这些操作符进行特征转化;另一方面,由于将 SparQL 图模式视为了一个序列,那么序

列中括号等操作符所表达的嵌套信息就必须被正确呈现,这也是未来研究工作中的重点.

此外,除了用于查询图数据库的 SparQL,查询传统关系数据库的 SQL 语言也具有与 SparQL 类似的特征,如果同样以位置信息记录 SQL 中的字段信息,同时将 SQL 中的 AND,HAVING 等操作符进行特征转化,那么本文提到的方法也可以适用于 SQL 语言,这也是未来会进行的工作.

References:

[1] Pan LH, Zhang JY, Zhang YJ, *et al*. Construction of knowledge graph in coal mine field. Computer Applications and Software, 2019,36(8):47−54,59 (in Chinese with English abstract). [doi: 10.3969/j.issn.1000-386x.2019.08.009]

[2] Holzschuher F, Peinl R. Querying a graph database—Language selection and performance considerations. Journal of Comput Syst Sci., 2016;82(1):45−68. [doi: 10.1016/j.jcss.2015.06.006]

[3] Hurtado C, Poulovassilis A, Wood P. Query relaxation in RDF. Journal on Data Semantics X.:31−61. [doi: 10.1007/978-3-540-77688-8_2]

[4] Hogan A, Mellotte M, Powell G, *et al*. Towards fuzzy query-relaxation for RDF. Lecture Notes in Computer Science, 2012:687−702. [doi: 10.1007/978-3-642-30284-8_53]

[5] Elbassuoni S, Ramanath M, Weikum G. Query relaxation for entity-relationship search. The Semanic Web: Research and Applications, 2011:62−76. [doi: 10.1007/978-3-642-21064-8_5]

[6] Lorey J, Naumann F. Detecting SPARQL query templates for data prefetching. The Semantic Web: Semantics and Big Data, 2013:124−139. [doi: 10.1007/978-3-642-38288-8_9]

[7] Lorey J, Naumann F. Caching and prefetching strategies for SPARQL queries. In: Proc. of the Advanced Information Systems Engineering. 2013. 46−65. [doi: 10.1007/978-3-642-41242-4_5]

[8] Deutsch A, Xu Y, Wu M, *et al*. TigerGraph: A native MPP graph database. arXiv preprint arXiv:1901.08248, 2019.

[9] Hu G, Secario M, Chen C. SeQuery: An interactive graph database for visualizing the GPCR superfamily. Database, 2019. 2019. [doi: 10.1093/database/baz073]

[10] Lai S, Yuan X, Sun S F, *et al*. GraphSE2: An encrypted graph database for privacy-preserving social search. arXiv preprint arXiv: 1905.04501, 2019.

[11] Kankanamge C, Sahu S, Mhedbhi A, *et al*. In: Proc. of the 2017 ACM Int'l Conf. on Management of Data (SIGMOD 2017). 2017. [doi: 10.1145/3035918.3056445]

[12] Fernandes D, Bernardino J. Graph databases comparison: AllegroGraph, ArangoDB, InfiniteGraph, Neo4J, and OrientDB. In: Proc. of the 7th Int'l Conf. on Data Science, Technology and Applications. 2018. [doi: 10.5220/0006910203730380]

[13] Fosic I, Solic K. Graph database approach for data storing, presentation and manipulation. In: Proc. of the 2019 42nd Int'l Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). 2019. [doi: 10.23919/mipro.2019.8756793]

[14] Chen H, Vasardani M, Winter S, *et al*. A graph database model for knowledge extracted from place descriptions. ISPRS Int'l Journal of Geoinf, 2018,7(6):221. [doi: 10.3390/ijgi7060221]

[15] Zhang W, Sheng Q, Qin Y, *et al*. SECF: Improving SPARQL querying performance with proactive fetching and caching. In: Proc. of the 31st Annual ACM Symp. on Applied Computing (SAC 2016). 2016. [doi: 10.1145/2851613. 2851846]

[16] Rico M, Touma R, Queralt A, *et al*. Machine learning-based query augmentation for SPARQL endpoints. In: Proc. of the 14th Int'l Conf. on Web Information Systems and Technologies. 2018. [doi: 10.5220/0006925300570067]

[17] Pérez J, Arenas M, Gutierrez C. Semantics and complexity of SPARQL. Lecture Notes in Computer Science, 2006,30−43. [doi: 10.1007/11926078_3]

[18] Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. In: Proc. of the Advances in Neural Information Processing Systems. 2014. 3104−3112.

[19] Elman J. Finding structure in time. Cognitive Science, 1990,14(2):179−211. [doi: 10.1207/s15516709cog1402_1]

[20] Xu K, Ba J, Kiros R, *et al*. Show, attend and tell: Neural image caption generation with visual attention. In: Proc. of the Int'l Conf. on Machine Learning. 2015. 2048−2057.

[21] Freitag M, Al-Onaizan Y. Beam search strategies for neural machine translation. arXiv preprint arXiv:1702.01806, 2017.

[22] Zhang W, Sheng Q, Qin Y, *et al*. Learning-based SPARQL query performance modeling and prediction. World Wide Web, 2017,21(4):1015−1035. [doi: 10.1007/s11280-017-0498-1]

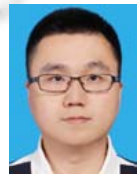
附中文参考文献:

[1] 潘理虎,张佳宇,张英俊,等.煤矿领域知识图谱构建.计算机应用与软件,2019,36(8):47−54,59. [doi: 10.3969/j.issn.1000-386x.2019.08.009]

杨东华(1976－),男,博士,副教授,博士生导师,主要研究领域为数据库系统,大数据管理与分析.

邹开发(1996－),男,硕士,主要研究领域为大数据分析与管理.

王宏志(1978－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为数据库系统,大数据管理与分析.



王金宝(1983－),男,博士,副教授,CCF 专业会员,主要研究领域为大数据分析.