G2S:基于语义块的知识图谱问答语义解析

高留杰[1],赵 文[2],张君福[3],姜 波[4]
(1. 北京大学软件与微电子学院,北京 100871; 2. 北京大学软件工程国家工程研究中心,北京 100871;
3. 北京北大软件工程股份有限公司,北京 100080; 4. 96901 部队 31 分队,北京 100085)

摘 要: 问题意图理解是知识图谱问答的主要任务之一,语义解析是当前理解问题意图的主流方法. 其主要挑战是如何充分利用知识图谱上下文理解问句中的隐含实体或关系,以及时间、排序和聚合等复杂约束条件等意图. 为了应对这些挑战,本文提出了一种基于语义块的知识图谱问答语义解析框架——Graph-to-Segment,框架中的语义解析模型结合了基于规则的准确度和基于深度学习的覆盖度,实现了问题到语义块序列的解析和语义查询图的构造. 框架将问题意图使用基于语义块的语义查询图表示,将问题的语义解析建模为语义块序列生成任务,采用编码器-解码器神经网络模型实现问题到语义块序列的解析,然后通过语义块组装形成语义查询图. 同时,结合知识图谱中的上下文信息,模型使用图神经网络学习问题的表示,改进隐含实体或关系的语义解析效果. 在两个知识图谱问答数据集上的实验表明,模型性能达到了良好的效果.

关键词: 知识图谱; 问答; 语义解析
中图分类号: TP183 **文献标识码:** A **文章编号:** 0372-2112 (2021)06-1132-10
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.12263/DZXB.20200157

G2S:Semantic Segment Based Semantic Parsing for Question Answering over Knowledge Graph

GAO Liu-jie[1],ZHAO Wen[2],ZHANG Jun-fu[3],JIANG Bo[4]
(1. *School of Software and Microelectronics*, *Peking University*, *Beijing* 100871, *China*;
2. *National Engineering Research Center for Software Engineering*, *Peking University*, *Beijing* 100871, *China*;
3. *Beijing Beida Software Engineering Co. Ltd*; 4. *Unit* 31 *of* 96901 *PLA Troops*, *Beijing* 100085)

Abstract: Question understanding is one of the important tasks of question answering over knowledge graph, where semantic parsing is the mainstream approach for understanding question utterance. The most significant challenge in this task is to understand the implicit entities, relations and the utterances of complex constraints such as time, ordinal, and aggregation in the question with the context of knowledge graph. In this paper, we propose graph-to-segment, a semantic segments based semantic parsing framework for question answering over knowledge graph. Our semantic parsing model integrates both rule-based and neural-based approaches to parse the semantic segment sequences and constructs the semantic query graphs with high accuracy and coverage. These semantic segment-based semantic query graphs, which consist of the semantic segments, are used to represent the utterance of questions. Question semantic parsing is modeled as a sequence generation task, where an encoder-decoder neural network is used to generate the semantic segments from natural language questions. Additionally, with the context information of knowledge graph, a graph neural network is used to learn the representation of questions to improve the effect of semantic parsing on implicit entities or relations. Experimental results show that our model achieves good performance on the two datasets.

Key words: knowledge graph; question answering; semantic parser

1 引言

知识图谱问答(Question Answering over Knowledge Graphs,KGQA)的目标是使用知识图谱中的事实来回答自然语言问题,实现用户无需了解图谱的数据结构便能高效、轻松地访问图谱中的重要知识的目标. 知识图

收稿日期:2020-02-11;修回日期:2020-04-21;责任编辑:梅志强

谱问答需要捕获自然语言的语义,核心在于问句语义和知识语义的理解和相似度计算,近年来,随着深度学习在自然语言处理领域的快速发展及其在问答任务上的良好表现,利用深度神经网络来改进面向知识图谱的问答成为研究的主要方向,研究方法主要分为基于排序的方法和基于翻译的方法[1].

基于排序的方法通过学习问句与候选答案的向量表示,采用端到端的方法根据上下文信息对候选答案进行评分排序得到最终答案. 文献[2]第一个提出了采用神经网络的 KGQA 方法,将问句与对应的子图嵌入进行匹配评分,文献[3~5]等在表示学习时考虑更多的问句上下文信息,缩小搜索空间. 文献[6~8]改进网络模型,加入注意力机制、多列卷积或双向 RNN 等方式来改进排序效果. 文献[9~12]尝试对问句对应的候选逻辑形式进行评分排序. 此类方法不需要手工规则和词汇表,但是不能建模更多语义信息,在生成候选集合时依赖于问句中的中心实体,且解释性较差,中间过程无法与用户进行交互.

基于翻译的方法将问句语义解析为机器可理解的逻辑形式,进而在知识图谱中进行查询或推理得到最终答案,是当前语义解析方法的主要技术途径. SPARQL 是最常见的逻辑形式,大部分知识图谱问答使用其作为问题意图的逻辑形式表示[13~15]. 也有研究提出了 Lambda-DCS[16,17]、CCG[18,19]和 FunQL[20]等逻辑形式用以改进问题语义的表示. 传统生成算法[1,19]依赖于词典和定义的文法,复杂度较高,领域迁移难度大. 基于深度学习的语义解析大部分借鉴翻译模型,将解析过程建模为 Seq2Seq 的过程[21~23],使用编码器-解码器网络架构完成序列的生成,发挥了循环神经网络强序列预测能力的优势,但把问句作为简单序列处理时难以建模结构信息,同时忽略了与知识库联系的上下文信息.

一个知识图谱如中包含了实体类型、关系类别和相关实例信息. 以 SPO 三元组存储的关系分为两类:一类用于连接实体与实体,另一类则连接了实体和对应的文字属性. 一个领域知识图谱的形式化表示为:$G=(T,R,E,I)$,其中 $T=\{t_1,t_2,\cdots,t_{|T|}\}$ 是一个实体类型集合,$E=\{e_1,e_2,\cdots,e_{|E|}\}$ 是一个实体集合,其中的实体所属类型是 T 的一个元素,$R=\{r_1,r_2,\cdots,r_{|R|}\}$,$r_i\in R^e\cup R^l$ 是一个二元关系类型集合,其中 R^e 表示实体与实体间的关系,R^l 表示实体与其文字属性的关系. $I=R^e\cup R^l$ 是关系实例的集合,其元素表示为 $r(e_1,e_2)$,其中 $r\in R$,$e_1,e_2\in E$. 如文献[1,5]中的知识图谱示例所示. 领域知识图谱问答一般围绕领域相关的问题进行,问题的理解可以分为三个子任务[1,10]:实体链接、关系识别以及逻辑和数字操作相关的约束条件标识. 知识图谱问答除了多实体、多关系的挑战外[24],还存在以下挑战:

隐式实体和关系:领域问答主要围绕具体领域进行提问,会省略意图明确的实体或关系,默认为领域上下文中的信息. 比如问句"how many states are there ?"中默认为在知识图谱中的上下文信息,即位于 us 的 state. 比如"how many rivers does alaska have ?"中 river 与 alaska 之间的关系单纯通过对句子的语义解析很难正确理解问句意图,需要借助知识图谱上下文补充更多的输入信息,现有的语义解析未充分利用特定知识图谱问答中的领域知识[25,26].

约束条件:知识图谱问答中存在大量时间、排序、聚合等约束条件,对问句的语义解析造成了较大的困难[11],设计的中间表示或逻辑形式主要面向通用领域[20,26],未充分考虑领域中基于片段的大粒度复用的特点.

多意图组合:问题一般由多个意图组合而成,每个意图表示了对预期结果的一种限定,简单问句中意图较单一,知识图谱问答中的问题一般都涉及若干意图,特定领域的问答意图较集中,通过不同的组合方式来实现复杂的问题意图,如何通过拆分意图来实现语义解析也是一个较大的挑战[27,28].

考虑到以上挑战,我们提出了基于语义块的语义解析框架——Graph-to-Segment(简称 G2S),用来完成知识图谱问答中的问题意图理解. 针对领域知识图谱问答中多意图组合的挑战,结合领域问题中意图集中的特点,本文总结提出了六种语义块模式用来表示问题的意图,通过语义块的生成和组装来完成问题的语义解析. 首先,将问题中的复杂语义拆分为由多个表示最小语义的语义块组成的序列,每个语义块对应到知识图谱上的单步查询或推理,例如在问题"how many capitals does rhode island have ?"中包含三个语义块:①capital的 count、②位于某个 state 的 capital 和③id 为 rhode island 的 state,三个语义块分别描述了问题意图的一部分,复杂问题一般通过多个语义块组成的序列来描述. 然后,基于生成的语义块序列信息进行组装可形成问题的完整意图,从而得到语义解析的结果,比如上述问题,可以将语义块②中的 state 用③表示的实体集合替换,然后替换到①的 capital 上,即得到问题的语义表示. 本文主要贡献如下:

(1)提出了一个知识图谱问答语义解析框架——G2S,定义了若干语义块模式用以表示问题的语义.

(2)将问题的语义解析建模为语义块序列的生成问题,利用图神经网络学习了 G2S 的模型,用以将问句解析为语义块序列,在两个数据集上验证,达到了良好的效果.

(3)针对隐式实体和关系的挑战,结合知识图谱上

下文信息,构造问句的上下文词典附加问句上作为图神经网络的输入,改进了语义解析的效果.

2 问题语义解析

我们的方法架构如图 1 所示,给定输入的自然语言问句,借助知识图谱生成问句的上下文词典作为附加信息输入到图神经网络中,得到网络的输出用以生成表示问题意图的语义块,从而可构造形成语义查询图表示解析后的问句意图. 下面章节我们详细描述框架的各部分内容.

2.1 语义查询图

参考文献[5,10,11,25],我们将问题用语义查询图表示. 如图 1(d)是一个问句的语义查询图示例,其中圆圈代表答案,圆角矩形代表知识图谱中的实体类型,实线箭头代表实体间的关系,菱形代表对实体应用的约束或聚合运算等操作. 给定一个问句,其语义查询图的生成可以看作是逐个识别其中的语义块并拼接到现有语义图的过程. 如图 2 中的虚线矩形演示了通过语义块形成问题语义表示的过程.

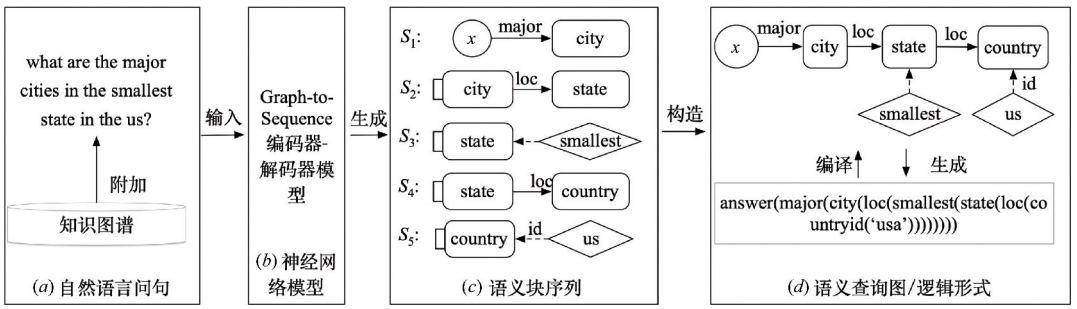


图1 问题“what are the major cities in the smallest state in the us?”的基于语义块的语义解析过程示意图,也是本文的方法架构

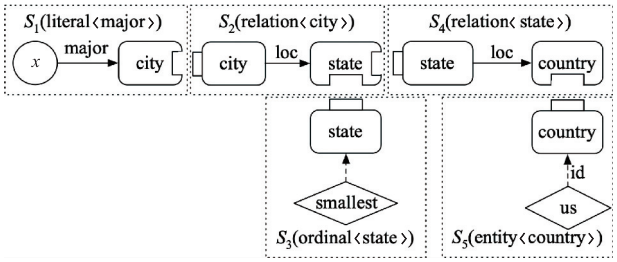


图2 从问句“what are the major cities in the smallest state in the us?”中生成语义块并构造形成语义查询图的过程示意,依次生成 $S_1,S_2,\dots,S_5$ 并拼接到前面结果上

2.2 语义块模式

我们定义了如下语义块模式用以表示问题意图(其中 p_t 表示类型为 t 的模式 p,t 是此模式的语义块表示的实体集合中的元素类型):

实体模式:表示一个指定类型的实体集合,例如 entity(city)表示所有的 city、entity(country,id,'us')表示 id 为 us 的 country, entity(flight,month,'july')表示 month 为 july 的 flight:

$$\text{entity}_t=\{(t,\text{atrr},\text{value}):t\in T,\text{attr}\in\mathbf{R}^t\} \quad (1)$$

关系模式:表示从类型为 t_2 的右实体集合经过关系 r 推理得到的类型为 t 的左实体集合,例如 relation(state,loc,:country)表示与指定的 country 存在 loc 关系的 state 等:

$$\text{relation}_t=\{(t,p,\{e:e\in t_2\}):p\in\mathbf{R}^e,(t,t_2)\in T\} \quad (2)$$

属性模式:表示在 SPO 三元组中对 object 为 literal 类型的实体属性进行计算的结果,比如 literal(len,:river):

$$\text{literal}_t=\{(p,\{e:e\in t\}):p\in\mathbf{R}^l,t\in T\} \quad (3)$$

排序模式:表示经过排序后得到指定位置的实体,比如 ordinal(max,:state)、ordinal(min,:city) 等:

$$\text{ordinal}_t=\{(\text{ord},\{e:e\in t\}):\text{ord}\in\{\max,\min\},t\in T\} \quad (4)$$

聚合模式:表示对输入指定的实体集合计算数量后的单一值,如 aggr(count,:city):

$$\text{aggr}_{n,n\in R}=\{(\text{aggr},\{e:e\in t\}):\ \text{aggr}\in\{\text{count},\text{average}\},t\in T\} \quad (5)$$

集合模式:表示对输入的两个或多个实体集合,计算集合的指定操作如交集、并集后的实体集合:

$$\text{join}_t=\{(\text{join},\{e:e\in t\},\{e:e\in t\}):\ \text{join}\in\{\text{intersetion},\text{union},\text{exclude}\},t\in T\} \quad (6)$$

表 1 列出了各种模式的示例和图示.

2.3 问题语义解析

我们的语义解析框架在领域知识图谱上,根据输入的自然语言问句及其对应的语义查询图学习语义解析模型,来完成问句意图解析. 如图 1 所示,给定问句 X,借助知识图谱 G 生成问句的上下文词典 X^C 输入到预训练的图神经网络模型中得到对应的语义块序列 Y 即为问句对应的语义查询图. 使用 Y 可直接在知识图谱上匹配得到答案[5],也可以转化为对应的逻辑形式推理得到最终答案[16]. 因此,问题的语义解析主要分为两步:问题上下文词典构造和语义块序列生成.

问题上下文词典构造:使用 X 和 G 可以得到 X 关联的候选实体的类别 T^{in},基于 G 可以得到 T^{in} 中元素之间的关系 R^{in},从而得到带有上下文信息的问题表示 X^C

表1　语义块模式的作用及示例说明

模型	作用	示例及说明
实体模式	通过id或属性约束查询实体集合	country — id — us entity(country, 'us'):id为us的country
关系模式	通过t_2类型的集合与r关系查询t_1类型的实体集合	city — loc — state relation(city, :state):满足与指定的state集合存在loc关系的city实体集合
属性模式	查询实体集合的literal属性	len — river literal(len, :river):得到指定的river集合的len属性
排序模式	对给定的实体集合通过指定方式排序后得到指定位置的实体	state — smallest ordinal(min, :state):对给定的state按从小到大排序,得到top1的实体集合
聚合模式	对给定的实体集合通过指定方式求聚合值	count — city aggr(count, :city):求给定的city类型的实体集合的数量
集合模式	对给定的两个实体按指定方式进行合并操作	intersection — city, city join(intersetion, :city, :city):求两个给定的city类型集合的交集

$=\{X, T^{\text{in}}, R^{\text{in}}\}$作为G2S模型的输入,我们将在章节3.1详述处理过程.

语义块序列生成:使用一个编码器-解码器网络,可以将一个输入的问题X^c解析为一个语义块的序列Y,我们需要:①一个图编码器,将输入X^c编码为向量表示;②一个解码器,用来在编码向量的条件下生成Y,我们在章节3详述网络模型.

3　语义块生成神经网络

我们基于文献[29]中的网络架构,采用一个基于编码器-解码器的图神经网络来完成语义块序列的生成. 如图3,模型主要包括一个由节点嵌入层和图嵌入层组成的图编码器、一个节点注意力机制和一个基于RNN的序列解码器,以下详细描述网络模型.

3.1　问题预处理

问题词语义化:首先使用WordNet[30]将问句X中的所有词转成为对应的语义词,这样对于多种形态(过去式、复数等)的词可以对应到同一个语义,比如city、cities使用WordNet处理后均转化为citi.

候选类别:我们使用实体链接工具[31],给定输入X,可以得到其在特定领域的相关类别的集合T^{in},识别出其中的实体进行如下处理:①知识图谱实体:对于可以链接到知识图谱上的实体,将实体替换为其对应的知识图谱中的类型,同时在R^{in}中添加一个元素,用来标识之间的isA关系;②知识图谱类型:将识别出的知识图谱类型直接添加到T^{in}中,并建立is关系.

候选关系:基于上一步中得到的候选实体类别T^{in},在知识图谱中找出类别之间可能的关系,建立关系集合R^{in},同时保留上一步建立的关系.

带有上下文的问题转换为图:将X中的词按顺序与其后的词建立不带标签的边,加入到R^{in}中,形成预处理后的图形式,作为图神经网络的输入.

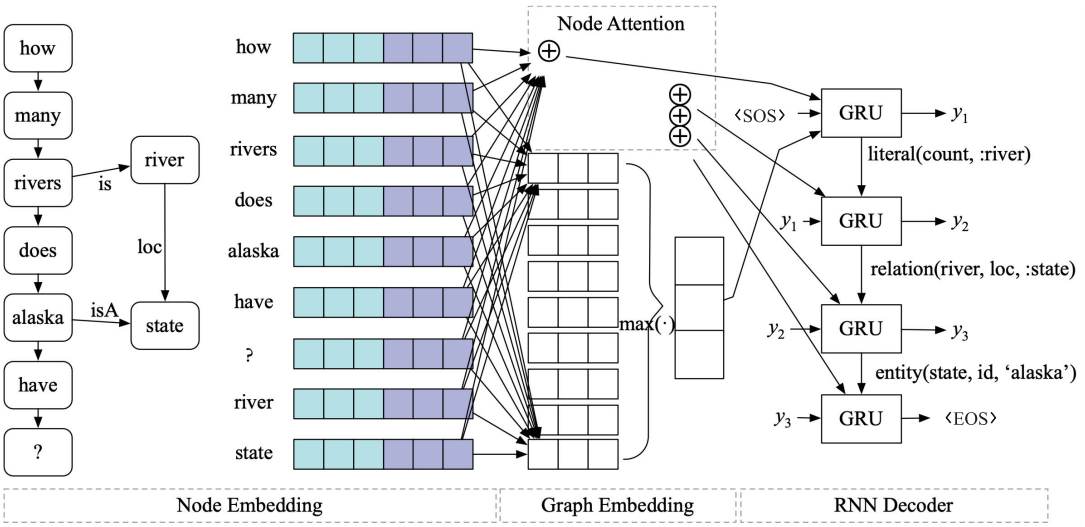


图3　G2S框架神经网络架构

3.2　图编码器

节点向量:节点包含两类,一类是问句中出现的单词组成的词典V^q,我们使用随机向量初始化. 另一类是问句连接的候选实体类别词典T^{in},此词典由知识库中

的类别 T 中的元素组成,使用随机向量初始化.

边向量:问句嵌入后的边有两类,一类边带有标签,这些边的标签与知识图谱中的 R 对应,另一类是无标签的边,我们使用与 R 共享的随机向量初始化每类边.

节点嵌入层:我们采用文献[29]中的节点嵌入生成过程生成节点的向量表示,节点的向量表示 h_v^j 包括 K 阶邻节点的信息组成,一类是指向此节点邻节点 $\boldsymbol{h}_{v\leftarrow}^j$,另一类是此节点指向的邻节点 $h_{v\rightarrow}^j$.

$$\boldsymbol{h}_v^j=[h_{v\rightarrow}^j,h_{v\leftarrow}^j] \tag{7}$$

图嵌入层:采用基于 Pooling 的图嵌入方式形成整个问题图的向量表示,此处采用 max-pooling 方式.

3.3 序列解码器

本文采用标准的序列解码器[32]解码语义块序列. 解码器为一个双向 GRU 模型,每次迭代生成一个语义块,最终形成语义块的序列 Y. 在第 i 步,根据此步骤的隐状态 s_i 得到此步骤的 y_i,同时把当前步骤的 s_i 和预测输出的 y_i 为输入,更新下一步的隐状态 s_{i+1}.

$$s_1=\tanh(\boldsymbol{W}^{(s)}[h_m^F,h_1^B]) \tag{7}$$

$$s_{i+1}=\mathrm{GRU}([\phi^{(y)}(y_i),c_i],s_i) \tag{8}$$

其中 c_i 为注意力上下文,参见 3.4 节,$\phi^{(y)}(y_i)$ 为语义块的嵌入.

语义块的嵌入:解码过程需要章节 2.2 中描述的每种语义块的嵌入. 语义块由名称和参数两部分组成:名称表示和类型为语义块的结构,其他参数表示语义块的语义,例如语义块 relation(city,loc,:state)的结构信息为 relation(city),其语义信息为(loc,:state). 分别嵌入语义块的结构和语义信息,以简化参数信息,使得不同的语义块模式间可以共享参数的结构或语义部分信息[25]. 除实体模式与实例 E 和 I 相关外,其他语义块均只与实体的类型 T 和关系类型 R 相关,我们基于知识图谱得到语义块空间,使用随机向量表示每种语义块的结构和语义部分.

语义块解码控制器:预测出语义块序列 Y 中每个语义块包含名称 p、实体类型 t 和输入参数 s 三个部分的内容. 使用 Beam 搜索[33]时,结合已经生成的语义块中可以接受的参数类型 s,与此时得到的语义块的输出类型 t 时行匹配,得到可以匹配的语义块来确定序列在当前时刻的输出.

3.4 节点注意力机制

序列解码器是一个 RNN,用来基于已预测的结果 $y_{<i}$、RNN 在此时的隐状态 s_i 和一个上下文向量 c_i 来预测第 i 步的输出 y_i. c_i 为编码器端的注意力表示,本文的网络模型中,其依赖于图神经网络的各节点表示. 每个节点表示包含关于输入的问题图的信息,同时将关注点放在当前的节点表示为中心的部分,c_i 即为这些节点表示的加权计算,计算公式如下:

$$c_i=\sum_{j=1}^{|X^c|}a_{ij}h_j,\ \text{e.t.}\ a_{ij}=\frac{\exp(e_{ij})}{\sum_{k=1}^{|X^c|}\exp(e_{ik})},\ e_{ij}=\mathrm{a}(s_{i-1},h_j) \tag{10}$$

其中 a_{ij} 为每个节点表示的权重,通过权重来控制上下文向量在当前步骤的关注点. a 是一个对齐模型,它基于 RNN 的隐状态 s_{j-1} 和输入的问题图节点 j 的向量表示,对位置 j 周围的输入节点和位置 i 处的输出匹配程度进行评分. 我们将 a 参数化为一个前馈神经网络,该神经网络与 G2S 神经网络模型中的其他组件联合训练得到.

4 实验验证

本节讨论实验,并与现有研究对比,以验证本文中方法的性能.

4.1 数据集

我们分别在两个数据集验证 G2S 模型的性能.

GEO880 是关于美国地理信息的问答数据集,包含 880 个问答对,问题采用自然语言描述的形式,答案为 Prolog 格式的逻辑形式,借鉴文献[18],我们使用标准的 600/280 建立训练集和测试集.

ATIS 是关于民航航班的问答数据集,包含近 5000 个关于航班的自然语言问答对,逻辑形式为 Lambda-DCS,借鉴文献[19],我们使用标准的 4473/448 建立训练集和测试集. 为了能够基于 3 中的网络架构训练语义块序列神经网络,我们对两个数据集中训练和测试数据的答案逻辑形式进行预处理,将其转化为语义块序列的形式,如表 2 所示.

4.2 实验建立

我们的模型包含图编码器的参数、GRU 的参数 $W^{(s)}$、问句中的词嵌 $\phi(x)$、模型序列中的语义块嵌入 $\phi(y)$. 使用有监督的方法通过训练语料训练得到这些参数. 给定训练语料中的问句 X 和其对应的语义块序列 Y,我们使用极大使然函数生成 X 对应的 Y,目标函数为:$Y^*=\sum_{i=1}^{n}\log P(Y_i|X_{<i})$.

参考文献[29],我们使用 Adam 优化器训练和更新参数,批大小设置为 30,学习率设置为 0.01. 为了避免过拟合,我们在解码层应用了 dropout 策略[34],将比率设置为 0.2. 在图编码器中将默认跳跃点 K 大小设置为 3,节点初始特征向量的大小设置为 100,使用 ReLU 作为非线性函数. 聚合器的参数是随机初始化的. 解码器具有 1 层,隐藏状态大小为 256. 由于具有均值聚合器和基于池的图嵌入的G2S通常比其他配置执行得更

表 2　GEO 和 ATIS 数据集中问答对及其对应的语义块示例,第二列给出了数据集中序列的平均长度,可以看出语义块可有效降低序列长度

数据集	平均长度	示例
GEO	7.6	问句:what are the major cities in the smallest state in the us?
	28.2	逻辑形式:answer(A,(major(A),city(A),loc(A,B),smallest(B,(state(B),loc(B,C),const(C,countryid(usa))))))
	2.9	语义块序列:literal(major,:city) relation(city,loc,:state) ordinal(smallest,:state) relation(state,loc,:country) entity(country,id,'usa')
ATIS	11.1	问句:what are the flights between dallas and pittsburgh on july eight ?
	28.4	逻辑形式:(_lambda $0e (_and(_flight $0)(_from $0 dallas:_ci)(_to $0 pittsburgh:_ci)(_day_number $0 8:_dn)(_month $0 july:_mn)))
	6.1	语义块序列:entity(flight) relation(flight,from,:city) entity(city,id,'dallas') relation(flight,to,:city) entity(city,id,'pittsburgh') entity(flight,day_number,'08') entity(flight,month,'july')

好,因此我们将此设置用作默认模型. 解码器输出时 Beam size 设置为 5. 我们通过迭代 80 次来训练模型. 模型采用 Pytorch 实现,每次迭代通过使用测试集预测得到 Y^*,累计 Y^* 与 Y 的误差到损失值.

4.3　实验结果

此部分我们讨论 G2S 模型的在 GEO 和 ATIS 数据集上的效果. 基于本文所述方法,我们采用三种配置来训练解析模型:第一种是基本的 G2S 模型;第二种是将语义块分解后嵌入到网络的解码层来实现语义块序列的预测(G2S(+MP));第三种在第二种的基础上增加了序列解析的控制器(G2S(+MP+Controller)),用来控制语义块序列组装为查询图的合法性. 我们采用在测试集上的准确率来评估各系统的性能,表 3 列出了我们的模型与现有研究的实验结果对比.

模型在 GEO 上的准确率为 86.4%,添加了语义块分解嵌入后达到了 88.3%,解码控制器提升了预测的准确率,达到了 90.5%,同时在 ATIS 数据集上三种配置的结果分别是 83.9%、85.1%、85.7%. 综合语义查询图的强表示能力和编码器-解码器模型的强预测能力,特别是基于图神经网络可以将带有上下文问题图无损映射到向量空间进行表示学习,超过了大多数以前的方法,达到了良好的效果.

表 3　在 GEO 和 ATIS 数据集上的实验准确率

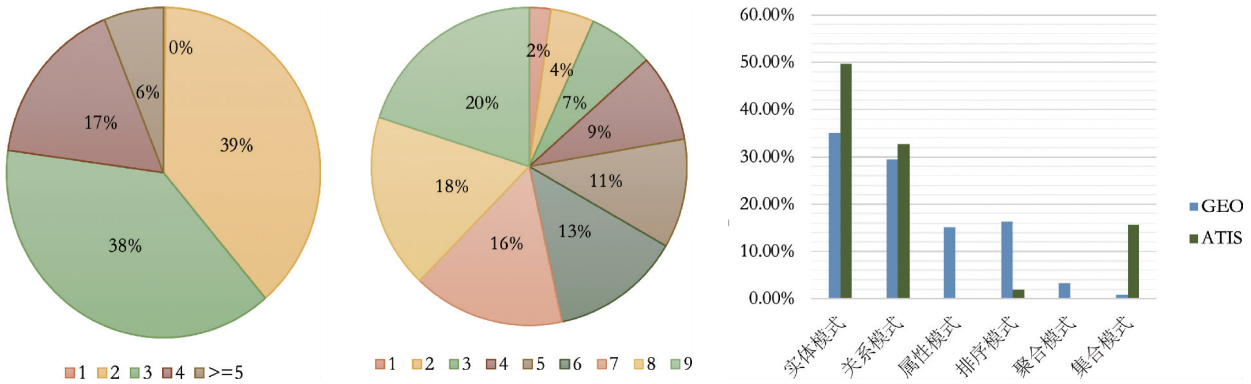
系统	GEO	ATIS
Zettlemoyer and Collins (2007)[19]	86.1	84.6
Kwiatkowski et al. (2011)[35]	88.6	82.6
Liang et al. (2011)[17] (+lexicon)	**91.1**	–
Zhao et al. (2015)[36]	88.9	84.2
Rabinovich et al. (2017)[37]	87.1	**85.9**
Jia and Liang (2016)[21] (+data)	89.3	83.3
Dong andLapata (2016)[23]:2Seq	84.6	84.2
Dong andLapata (2016)[23]:2Tree	87.1	84.6
Seq2Act (+C1+C2)[25]	88.9	85.5
G2S	86.4	83.9
G2S(+MP)	88.3	85.1
G2S(+MP+Controller)	90.5	85.7

语义块嵌入:我们的模型默认将每个语义块独立嵌入,同时我们也使用语义块的结构和语义信息分解嵌入的方式进行了实验. 通过实验显示对准确度有较大影响,在 GEO 数据集上提升了 1.9,在 ATIS 数据集上提升了 1.2. 可以看出,分解大粒度的语义块信息,有助于共享各部分的上下文信息,在 GEO 和 ATIS 这样的小规模数据集上,能够有效提升模型的预测水平.

解码控制器:加入解码控制器,对模型两个数据集上的预测准确度有不同的影响,在 GEO 数据集上提升了 2.5,在 ATIS 数据集上提升了 0.7. 从实验可以看出,解码控制器基于实体类型进行优化控制,而 ATIS 数据集上的实体类型个数较少,所以对结果的影响相比 GEO 数据集较小. 同时,控制器可以确保生成的语义块能够正确组装为语义查询图.

4.4　实验分析

语义块长度:表 2 中展示了逻辑形式与语义块在每个数据集中上的平均长度,可以看出,对比 Seq2Act[25] 中的动作序列长度(GEO 为 18.2,ATIS 为 25.8)与数据集原始逻辑形式的序列长度(GEO 为 28.2,ATIS 为 28.4),语义块可有效降低预测的目标序列的长度. G2S 语义块序列长度是 GEO 数据集逻辑形式的 10.3%,是 Seq2Act 在 GEO 上长度的 15.9%,是 ATIS 数据集逻辑形式的 21.5%,是 Seq2Act 在 ATIS 上长度的 23.6%. 序列长度的下降对提升预测的准确度有较大影响,同时也有效降低了语义图构造的复杂性.



(*a*) GEO (*b*) ATIS (*c*) 语义块模式分布

图4 GEO和ATIS数据集中的答案逻辑形式转化为语义块后的序列长度分布情况，图(*c*)为语义块模式在两个数据集上的分布情况

语义块分布：转化后的语义块模式和实体分布情况如图4(*a*)和4(*b*)所示. 从图中的情况可以看出，GEO数据集中的语义块长度主要分布在2和3，这两个长度的数据占全部数据的76.9%，而ATIS数据集中的语义块长度主要分布在4～6，这三个长度的数据占全部数据的78.6%. 从图4(*c*)语义块模式的分布来看，实体模式和关系模式占两个数据的大部分语义块模式，GEO为64.5%，ATIS则达到了82.3%，除此之外，GEO数据集中更多的是属性模式和排序模式(31.4%)，ATIS数据集中则多以集合模式为主(15.7%)，模式的数据分布反映出了不同领域问题的特点.

图神经网络：通过将问题以图的方式嵌入到神经网络中，可以加入更多的上下文信息，对模型语义解析的效果有较大的提升. 我们使用了两种图构造的方式：链式连接和全连接. 链式连接方式只将问题中的词与其后的一个词建立连接关系，而全连接方式将问题每个词与其后的所有词建立连接关系，通过实验验证两种方式对实验结果的影响不大.

4.5 错误分析

我们对输出的错误信息进行了分析，语义块序列解析错误主要分为以下两类.

丢失问题语义：注意力模型没有考虑对齐历史，使某些单词在解析过程中被忽略. 例如在表4中的第一、二种情况下，在解码过程中会忽略“which border texas”和“big”，使得解析后的结果中缺失了问题意图的进一步限定，使用神经机器翻译中使用的显式单词覆盖模型可以进一步解决此问题[38].

语义理解错误：由于GEO和ATIS的数据量相对较小，因此在测试集中会存在训练集中未出过的单词或提问方法，使得模型预测的结果出现偏差，在表4中的第三种情况，使用“how many”来提问即可得到正确结果. 一种解决方案是在未注释的文本数据上学习单词嵌入，然后将其用作问题单词的预训练向量，或换为模型可识别的形式[39,40].

表4 一些错误分析示例，每个示例包括问句、最佳答案和模型预测的结果

错误类型	示例
丢失问题语义	问句：what are the populations of states which border texas? 答案语义块： literal(population,:state) relation(state,next_to,:state) entity(state,id,‘texas’) 预测语义块： literal(population,:state) entity(state,id,‘texas’)
丢失问题语义	问句：how many big cities are in pennsylvania ? 答案语义块： aggr(count,:city) join(intersection,:city,:city) entity(city,major,1) relation(city,loc,:state) entity(state,id,‘pennsylvania’) 预测语义块： aggr(count,:city) relation(city,loc,:state) entity(state,id,‘pennsylvania’)
语义理解错误	问句：give me the number of rivers in california. 答案语义块： aggr(count,:river) relation(river,loc,:state) entity(state,id,‘california’) 预测语义块： literal(len,:river) relation(river,loc,:state) entity(state,id,‘california’)

5 结论

本文提出了一种知识图谱问答的语义解析框架——G2S，结合了语义解析中基于规则的准确度和基于深度学习的覆盖度，同时考虑知识图谱的上下文信息，将问题的语义解析建模为图到序列的编码器-解析器任务，语义块的解析不依赖于问答输出的逻辑形式，具有较强的适应性. 我们在两个数据集上验证了框架

模型的可行性,实验结果表明模型达到了良好的效果.

下一步,为了进一步提升语义解析框架的有效性,我们准备设计语义块的组合模式算法,改进大粒度语义解析的复用性问题,为了解决训练语料不足的问题,我们准备参考相关研究[41]设计一个可交互的领域知识图谱语料自动或半自动生成算法和工具,通过交互式UI可以在不需要具备领域专业知识的前提下快速设计和形成特定领域知识图谱的问答语料.

参考文献

[1] Nilesh Chakraborty, Denis Lukovnikov, Gaurav Maheshwari, et al. Introduction to Neural Network based Approaches for Question Answering over Knowledge Graphs[EB/OL]. http://export.arxiv.org/abs/1907.09361, 2019.

[2] Antoine Bordes, Sumit Chopra, Jason Weston. Question answering with subgraph embeddings[A]. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing[C]. Doha, Qatar: EMNLP, 2014. 615 – 620.

[3] Li Dong, Furu Wei, Ming Zhou, et al. Question answering over freebase with multi-column convolutional neural networks[A]. 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing[C]. Beijing, China: ACL, 2015. 260 – 269.

[4] Denis Lukovnikov, Asja Fischer, Jens Lehmann, et al. Neural network-based question answering over knowledge graphs on word and character level[A]. WWW '17 Proceedings of the 26th International Conference on World Wide Web[C]. Perth, Australia: IW3C2, 2017. 1211 – 1220.

[5] Sen Hu, Lei Zou, Jeffrey Xu Yu, et al. Answering natural language questions by subgraph matching over knowledge graphs[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(5), 824 – 837.

[6] Yuanzhe Zhang, Kang Liu, Shizhu He, et al. Question Answering over Knowledge Base with Neural Attention Combining Global Knowledge Information[EB/OL]. https://arxiv.org/abs/1606.00979, 2016.

[7] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, et al. Translating embeddings for modeling multi-relational data[A]. Advances in Neural Information Processing Systems 26[C]. Lake Tahoe, Nevada: NIPS, 2013. 2787 – 2795.

[8] Zihang Dai, Lei Li, Wei Xu. CFO-conditional focused neural question answering with large-scale knowledge bases[A]. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics[C]. Berlin, Germany: ACL, 2016. 800 – 810.

[9] Gaurav Maheshwari, Priyansh Trivedi, Denis Lukovnikov, et al. Learning to rank query graphs for complex question answering over knowledge graphs[A]. The Semantic Web-ISWC 2019[C]. Springer, Cham: ISWC, 2019. 487 – 504.

[10] Wen-tauYih, Ming-Wei Chang, Xiaodong He, et al. Semantic parsing via staged query graph generation-question answering with knowledge base[A]. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing[C]. Beijing, China: ACL, 2015. 1321 – 1331.

[11] Jun-Wei Bao, Nan Duan, Zhao Yan, et al. Constraint-based question answering with knowledge graph[A]. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers[C]. Osaka, Japan: COLING, 2016. 2503 – 2514.

[12] Hongye Tan, Honghong Zhao, Ru Li, Bei Liu. A pipeline approach to free-description question answering in chinese gaokao reading comprehension[J]. Chinese Journal of Electronics, 2019, 28(1), 113 – 119.

[13] Shizhu He, Yuanzhe Zhang, Kang Liu, et al. CASIA@V2: A mln-based question answering system over linked data[A]. Question Answering over Linked Data (QALD-4)[C]. Sheffield, United Kingdom: CLEF, 2014. 1249 – 1259.

[14] Saeedeh Shekarpour, Edgard Marx, Axel-Cyrille Ngonga Ngomo, et al. SINA: Semantic interpretation of user queries for question answering on interlinked data[J]. Journal of Web Semantics, 2015, 30: 39 – 51.

[15] Jie Liu, Wei Li, Liming Luo, et al. Linked open data query based on natural language[J]. Chinese Journal of Electronics, 2017, 26(2), 230 – 235.

[16] Jonathan Berant, Andrew Chou, Roy Frostig, et al. Semantic parsing on freebase from question-answer pairs[A]. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing[C]. Seattle, Washington, USA: EMNLP, 2013. 1533 – 1544.

[17] Percy Liang, Michael I Jordan, Dan Klein. Learning dependency-based compositional semantics[J]. Computational Linguistics, 2013, 39(2), 389 – 446.

[18] Luke S Zettlemoyer, Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars[A]. UAI'05 Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence[C]. Arlington, Virginia, USA: AUAI, 2005. 658 – 666.

[19] Luke Zettlemoyer, Michael Collins. 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[C]. Prague, Czech Republic: EMNLP-CoNLL, 2007. 678 – 687.

[20] Jianpeng Cheng, Siva Reddy, Vijay A Saraswat, et al. Learning an executable neural semantic parser [J]. Computational Linguistics, 2019, 45(1): 59 – 94.

[21] Robin Jia, Percy Liang. Data recombination for neural semantic parsing[A]. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics[C]. Berlin, Germany: ACL, 2016. 12 – 22.

[22] Chunyang Xiao, Marc Dymetman, Claire Gardent. Sequence-based structured prediction for semantic parsing [A]. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics[C]. Berlin, Germany: ACL, 2016. 1341 – 1350.

[23] Li Dong, Mirella Lapata. Language to logical form with neural attention [A]. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics [C]. Berlin, Germany: ACL, 2016. 33 – 43.

[24] Sen Hu, Lei Zou, Xinbo Zhang. A state-transition framework to answer complex questions over knowledge base [A]. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels[C]. Brussels, Belgium: ACL, 2018. 2098 – 2108

[25] Bo Chen, Le Sun, Xianpei Han. Sequence-to-action-end-to-end semantic graph generation for semantic parsing [A]. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics[C]. Melbourne, Australia: ACL, 2018. 766 – 777.

[26] Pengcheng Yin, Graham Neubig. TRANX: A transition-based neural abstract syntax parser for semantic parsing and code generation[A]. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations[C]. Brussels, Belgium: 2018, EMNLP. 7 – 12.

[27] D Punjani, K Singh, A Both, et al. Template-based question answering over linked geospatial data[A]. Proceedings of the 12th workshop on geographic information retrieval[C]. New York, USA: ACM, 2018. 7 – 16.

[28] Zheng Weiguo, Yu Jeffrey Xu, Lei Zou, et al. Question answering over knowledge graphs-question understanding via template decomposition[J]. PVLDB, 2018, 11(11), 1373 – 1386.

[29] Kun Xu, Lingfei Wu, Zhiguo Wang, et al. Graph2Seq: Graph to Sequence Learning with Attention-based Neural Networks [EB/OL]. https://arxiv. org/abs/1804. 00823, 2018.

[30] George A. Miller. Wordnet: A lexical database for english [J]. Communications of the ACM, 1995, 38 (11), 39 – 41.

[31] Yi Yang, Ming-Wei Chang. S-MART: Novel tree-based structured learning algorithms applied to tweet entity linking[A]. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing[C]. Beijing, China: ACL, 2015. 504 – 513.

[32] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural machine translation by jointly learning to align and translate[A]. 3rd International Conference on Learning Representations [C]. San Diego, United States: ICLR, 2015.

[33] Ilya Sutskever, Oriol Vinyals, Quoc V Le. Sequence to sequence learning with neural networks[A]. 27th International Conference on Neural Information Processing Systems[C]. Montreal, Quebec, Canada: NIPS, 2014. 3104 – 3112.

[34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, et al. Dropout: A simple way to prevent neural networks from overfitting [J]. Journal of Machine Learning Research, 2014, 15 (1): 1929 – 1958.

[35] Tom Kwiatkowski, Luke S Zettlemoyer, Sharon Goldwater, et al. Lexical generalization in CCG grammar induction for semantic parsing [A]. Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing[C]. Edinburgh, Scotland, UK: EMNLP, 2011. 1512 – 1523.

[36] Kai Zhao, Hany Hassan, Michael Auli. Learning translation models from monolingual continuous representations [A]. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies[C]. Denver, Colorado: NAACL, 2015. 1527 – 1536.

[37] Maxim Rabinovich, Mitchell Stern, Dan Klein. Abstractsyntax networks for code generation and semantic parsing [A]. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics[C]. Vancouver, Canada: ACL, 2017. 1139 – 49.

[38] Zhaopeng Tu, Zhengdong Lu, Yang Liu, et al. Modeling-coverage for neural machine translation[A]. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics[C]. Berlin, Germany: ACL, 2016. 76 – 85.

[39] Bo Chen, Le Sun, Xianpei Han, et al. Sentence rewriting for semantic parsing[A]. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics [C]. Berlin, Germany: ACL, 2019. 766 – 777.

[40] Li Dong, Jonathan Mallinson, Siva Reddy, et al. Learning to paraphrase for question answering[A]. Proceedings of

the 2017 Conference on Empirical Methods in Natural Language Processing[C]. Copenhagen, Denmark: EMNLP, 2017. 875 – 886.

[41] Yushi Wang, Jonathan Berant, Percy Liang. Building asemantic parser overnight[A]. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing [C]. Beijing, China: ACL, 2015. 1332 – 1342.

作者简介

高留杰　男,1979 年出生,河南洛阳人,北京大学软件与微电子学院 2016 级博士研究生,主要研究方向为大数据、知识图谱问答和自然语言处理.
E-mail:gaoliujie@ pku. edu. cn

赵　文　男,1967 年出生,辽宁大连人. 现为北京大学软件工程国家工程研究中心研究员、博士生导师,主要研究领域为知识图谱、软件工程、软件安全.
E-mail:zhaowen@ pku. edu. cn

张君福　男,2018 年获得北京大学软件工程博士,主要研究方向为多源数据融合、知识图谱构建、自然语言处理、智能检索等.
E-mail:zhangjf@ beidasoft. com

姜　波　男,1973 年 12 月出生,江西上饶人,高级工程师,主要研究方向软件工程、大数据应用.