一种基于路径挖掘、匹配和排序的知识库问答方法

Wen Dai, Huiwen Liu, Yan Liu, Ninglin Du, Shuai Chen

Xiaomi Corporation, AI
{daiwen, liuhuiwen, liuyan15, duninglin, chenshuai3}@xiaomi.com

Abstract. 基于知识库的问答是问答系统的一个热门研究方向。本文基于路径挖掘、路径和query之间的相似匹配和排序，构建了一种能够通用地处理简单问题和带约束问题的知识库问答方法。本方法主要包含实体/属性值/数值的抽取、路径挖掘、路径匹配和排序三个部分，创新性地将约束转换为一种特殊的三元组路径，采用常规的路径匹配方式来解决带约束的复杂问题。本方法在CCKS 2021生活服务知识图谱问答评测的B榜测试集上取得了0.7885的F1值。

Keywords: 知识库问答，带约束问题，路径挖掘，路径匹配和排序

1 介绍

知识库问答是问答的一个重要分支，它以知识库中结构化的知识为依据来回答用户的问题，在客服、搜索等等场景下具有较高的应用价值。知识库问答可以大致分为简单问题和带约束问题。简单问题，指的是用知识库中的一个三元组就可以获取答案的问题；带约束问题具体指的是带有一定限制条件的问题，比如“长沙有哪些4A级别的景点?”，对应地需要从知识库中找到三元组“?x <城市> <长沙>”，同时答案?x需要满足三元组约束“?x <景点的等级A> '4A'”。复杂类别的约束问题可能还会涉及到数值的大小约束和排序：数值的大小约束，比如“青岛海尔洲际酒店附近2公里的酒店都有哪些”，对应地需要从知识库中找到满足条件的复合值类型(CVT)“<青岛海尔洲际酒店> <附近> ?cvt. ?cvt <实体名称> ?y. ?y <类型> <酒店>.”，同时CVT需要满足距离约束“?cvt <距离值> ?distance. filter(?distance <= 2).”；排序类问题，比如“刚游玩结束厦门星空艺术馆，最少得走多少公里才能到附近酒店”，对应地需要从知识库中找到满足条件的复合值类型(CVT)“<厦门星空艺术馆> <附近> ?cvt. ?cvt <实体名称> ?y. ?cvt <距离值> ?distance. ?y <类型> <酒店>.”，同时需要对距离值进行排序，找出距离最小的那一个结果。带约束问题的问答一直以来都是知识库问答中一个难点问题，相比简单问题，具有更高的应用价值。

本文提出了一种基于路径的知识库问答解决方案，可以同时解决简单问题和带约束的复杂问题，主要由三个模块构成：实体/属性值/数值的抽取、路径挖掘、路径匹配和排序。第一步是实体/属性值/数值的抽取，不同于常规的基于实体链接来抽取实体的方式，为了保证实体模块的召回，这里结合了多种方法来抽取实体/属性值/数值。采用字典树、模糊匹配、NER、duckling和Lucene检索

多种手段，从query中抽取所有可能出现过的实体、属性值和数值类型。第二步是路径挖掘，根据上一模块抽取到的实体、属性值和数值类型，这一步通过从知识库数据中找到相关的三元组，对三元组进行组合来得到普通路径。额外地，数值类型的约束(比如大小比较和排序)，会通过人为构建的三元组的形式来挂载到普通的路径上，形成带有约束的路径（以下称之为推理路径）。第三步是路径的匹配和排序，这一部分采用了常见的预训练模型来对候选路径和query的语义匹配程度来进行打分，打分最高的路径就是整个方法输出的路径结果，将路径转化为对应的知识库查询语句后，就可以获取最后的答案。

2 相关工作

传统的知识库方法主要有两种，基于语义解析的方法(Semantic Paring)和基于检索的方法(Information Retrieval)。语义解析中，需要把自然语言问题转换成为一种中间形式的逻辑表达式(Logical Forms)，然后将它转换为一种可执行的知识库查询语句(比如SPARQL)来查询知识库获取答案[1, 2]。传统的语义解析方法是基于预先定义的模板，来完成自然语义到中间态的转换。这种方法需要依赖大量的人力标注数据，缺乏一定的泛化性。其主要步骤大致分为:实体链接、属性识别、约束挂载和候选图排序[3]。基于信息检索的方法，首先需要识别问题中的实体指称(mention)，将他们链接到知识库中的实体上。再挖掘与这些实体相关联的子图作为候选的子图，抽取与子图相关的特征来作为候选集的排序依据。最后选择排序最高的子图来作为最好的结果[5, 6]。

3 方法

3.1 实体/属性值/数值抽取

从query中抽取实体/属性值/数值是整个pipeline的第一步。这里所谓的属性值指的是，诸如三元组“<刘德华_（中国香港男演员、歌手、词作人）> <出生日期> ‘1961-09-27’”出现过的“1961-09-27”，它是一种非实体类别的字符串或者日期、数值等。数值指的是数值/时间类型的属性值，诸如三元组“<成都泰合索菲特大饭店> <平均价格> ‘597’”中出现过的“597”，或者是关键的时间，比如“18:00”等。上述关键字符的抽取，通常而言是由实体链接来完成的。实体链接如果做得不好的话，会导致“错误的传播”。具体而言，一旦实体链接没有找到正确的实体、属性值或数值的话，后续的模块就不能构建出正确的路径，排序也不会得到正确的结果。然而，若是这一步多抽取了一些错误的实体，对于后续模块的影响也只是多出来一些候选路径。因此，对于整个流程而言，第一步召回率的重要性是要高于准确率的。

考虑到模块对于召回的要求，这里没有采用传统的实体链接方案来做实体/属性值/数值抽取。而是结合了字典树，模糊匹配，duckling，NER和Lucene检索等多种方式来尽可能多地从query中抽取实体/属性值/数值，以下分别描述。

3.1.1 基于字典树的实体/属性值抽取

这一步针对和query在字面上能够匹配的实体和属性值。将知识库三元组中的实体和属性值构建mention后载入到字典树中，与query进行匹配。实体/属性值在构建mention时，可能是一对多的。实体/属性值在大小写归一、时间/日期归一、以及去掉标点符号等前后的字符串都会作为mention载入到字典树中，以提升实体/属性值抽取的召回。

3.1.2 模糊匹配

Query中的mention可能在字面上不完全和实体匹配，比如“中国最早的诗歌总集在内容上可以划分为哪几类”，对应的真实实体可能是“<中国最早的一部诗歌总集>”；“之前的绿城雷迪森酒店那个房间有24小时热水呢?”，对应“<新昌尊蓝山居（原绿城雷迪森大酒店）>”。这类问题无法再通过字典树的方式来进行提取，需要使用模糊匹配的手段来做。

模糊匹配可以分为三个部分：1. 建立字到实体/属性值的倒排索引，按照query中出现的字来获取候选的实体/属性值，缩小实体范围；2. 求解候选实体/属性值和query的最长公共子序列；3. 由公共子序列来定位该候选实体在query中可能的mention，并计算编辑距离，对候选实体排序，选择最优实体。

3.1.3 基于duckling的数值属性发现

在生活类数据中，有一些数值属性需要额外解析。比如在问题“雍和宫附近[3公里]内，能在[晚上六点]入住的酒店有哪些”中，“3公里”和“晚上六点”需要解析成SPARQL中需要的数值3.0和时间18:00。

这一类解析可以称作结构化解析，不仅需要进行解析，还要完成数值和单位等的拆解。这部分使用了Xiaomi/MiNLP下的duckling-fork-chinese[1]，该项目基于规则生成再加上PCFG做组合排序。这里针对问题中的数字、时间、货币、距离、面积等进行了解析。在解析结果的基础上为下游提供建议的数值候选。

表1. 基于duckling的数值解析结果示例

维度	示例查询	结果示例
时间	二零零三年六月十二日	2003-06-12
	晚上六点	18:00
数字	一百二十三	123

[1] https://github.com/XiaoMi/MiNLP/tree/main/duckling-fork-chinese

3.1.4 基于NER的指称识别

有一些目标实体类似于“<成都魔幻星空酒店>”，而在问题中的表达可能是“成都的魔幻星座酒店”，这种问题会对使用字典树的方法造成一些麻烦。利用SPARQL的结果对问题进行指称反标来构建训练数据，在这个基础上进行NER训练可以得到指称的序列标注。这里使用了BERT-NER-Pytorch[2]项目中的BERT+CRF来进行序列标注。对标注的结果再利用Lucene进行模糊检索，取Top 5的实体作为实体候选。

3.1.5 基于Lucene的长实体/属性值发现

长实体/属性值是指描述信息比较长的实体和属性值，比如“"2009年第一轮第3位被雷霆队选中"的球员球衣号码是？”中的“2009年第一轮第3位被雷霆队选中”这一类在问题中与目标有可能差别比较大，但是长实体在句子中可能占据了大半部分的字符。以问题作为输入，基于Lucene检索实体会有比较好的效果。对知识库中的三元组中包含的实体和属性值建立Lucene检索。问题检索得到的Top结果会加入到实体/属性值的候选中。

3.2 路径挖掘

在用知识库来回答简单问题和带约束的问题中，答案为知识库中的某些节点，或这些节点的聚合、排序、过滤等逻辑推理的结果。按照答案的不同，将路径挖掘的过程分解为两个步骤：潜在答案节点的路径挖掘（以下简称节点路径）、逻辑推理路径挖掘（以下简称推理路径）。推理路径以节点路径作为基础，通常用于带聚合、排序等约束的问答。

以“离天坛最近的酒店人均800以下酒店是哪一家？”为例来说明路径挖掘过程，假设上游模块所给的初始化节点为“<天坛公园>”、“<酒店>”。

3.2.1 节点路径挖掘

1、 路径的深度拓展

深度拓展是指在当前节点（初始化节点或者当前路径的答案节点），沿知识库中的三元组方向进行拓展。这里定义一种路径类型描述规则：I指的是初始化节点，S指的是以当前节点作为三元组Subject进行拓展，O指的是当前节点作为Object进行拓展。拓展类型是在现有路径的基础上叠加，用“-”连接。同一种拓展类型代表一类路径集合。示例query经过深度拓展后，得到如下结果：

表2. 路径扩展示例

Base类型	新路径	答案变量	新类型
I	<天坛公园> <附近> ?cvt	?cvt	I-S
I-S	<天坛公园> <附近> ?cvt. ?cvt <实体名称> ?a	?a	I-S-S
I	?a <类型> <酒店>	?a	I-O

[2] https://github.com/lonePatient/BERT-NER-Pytorch

2、 路径组合

当多个不同路径的答案存在交集时，这些路径可以在该答案节点进行合并，得到新的路径。同样地，这里定义一种组合描述规则：在一个组合约束中可能存在着多个不同的路径类型，针对每个类型设置了2个参数，第一个为是否允许该类型路径多次出现在组合路径中（0为不允许，1为允许），第二个为是否该类型路径必须出现在组合路径中（0为非必须，1为必须）。合理地配置这两个参数可以为路径挖掘的过程剪枝，大幅提升路径挖掘的效率。整合两个参数后，可将参与到约束组合的路径类型描述为TYPE_X_Y，TYPE为原有路径类型，X的位置为路径类型是否可重复，第二个Y的位置为路径类型是否是必须的。多个路径类型的组合可描述为{TYPE_X_Y, TYPE1_X1_Y1}(n)，n代表组合约束的数量。组合过程如表3所示：

表3. 组合路径示例

Base类型	新路径	答案变量	新类型
I-O I-S-S	?a <类型> <酒店>. <天坛公园> <附近> ?cvt1. ?cvt1 <实体名称> ?a.	?a	{I-O_0_1, I-S-S_0_1}(2)

路径深度拓展中，可能会因为节点对应的三元组数量太多导致路径数量爆炸。必要时可对拓展所用的三元组做一些筛选和过滤，也可在路径组合时设计剪枝策略。例如：同一个mention或者不同mention在query中的位置有重叠，这些mention所对应的实体不可能同时在路径中出现。

另外，节点路径的结果可能存在重复。原因是路径中的变量节点可存在多个满足条件的实体，不管是路径的深度拓展还是组合都是以知识库中的节点为连接点，所以在不同的连接点上，可能得到相同的路径，需要对路径进行去重。

3.2.2 推理路径挖掘

表4. 推理属性及其类型示例

推理属性名	推理变量类型
<平均价格>	价格
<酒店入住开始时间>	时间
<酒店退房时间>	时间
<押金>	价格
<酒店入住结束时间>	时间
<距离值>	距离
<房屋面积>	面积
<容纳人数>	人数

1、 推理变量补充

推理路径是以节点路径作为基础，对路径中推理变量进行聚合、排序、过滤等操作来扩展得到的。推理变量在节点路径中可能已经存在，也有可能是缺失

的。在推理路径生成前，需要在节点路径上补充缺失的推理变量，补充的方法和路径的深度拓展相似，但有以下区别：1、补充推理变量后的路径的答案和答案变量不会改变；2、补充推理变量的方式只有S类型拓展；3、可用于路径拓展的三元组范围的属性名有限，表4中列举了一些可用于推理的三元组的属性名和变量类型。补充推理变量后的路径仅为中间状态路径，不作为最终的候选路径，该操作的路径类型用V来表示。

表5. 推理变量补充

Base类型	新路径	新类型
{I-O_0_1, I-S-S_0_1}(2)	?a <类型> <酒店> . <天坛公园> <附近> ?cvt1 . ?cvt1 <实体名称> ?a . ?a <平均价格> ?price . ?cvtf1 <距离值> ?distance . ?a <押金> ?price1.	{I-O_0_1, I-S-S_0_1}(2)-V

2、 过滤

过滤是根据价格、距离值等约束条件来对推理变量进行一定的限制。这里将SPARQL中的“filter (?price < 800)”改写为“?price <小于> 800”的三元组描述形式。此外，定义约束类型“<大于>、<小于>、<等于>、<大于等于>、<小于等于>”。在之前的工作中，已经从query中抽取了“价格、时间、距离、面积、人数”等类型的数值，如果与推理变量类型相同，则将其组合成为一个过滤约束。如果路径中的多个变量具有相同的变量类型，则每一个变量均需与抽取的值组合形成过滤约束，如表6中的“<平均价格>”和“<押金>”。对路径添加过滤操作用F来描述，过滤约束的过程如表6所示。

表6. 过滤约束示例

Base类型	新路径	新类型
{I-O_0_1, I-S-S_0_1}(2)-V	?a <类型> <酒店> . <天坛公园> <附近> ?cvt1 . ?cvt1 <实体名称> ?a . ?a <平均价格> ?price . ?cvtf1 <距离值> ?distance . ?a <押金> ?price1 . ?price <小于> 800 .	{I-O_0_1, I-S-S_0_1}(2)-V-F
{I-O_0_1, I-S-S_0_1}(2)-V	?a <类型> <酒店> . <天坛公园> <附近> ?cvt1 . ?cvt1 <实体名称> ?a . ?a <平均价格> ?price . ?cvtf1 <距离值> ?distance . ?a <押金> ?price1 . ?price1 <大于> 800 .	{I-O_0_1, I-S-S_0_1}(2)-V-F

3、 排序

排序是指对推理变量进行排序（升序或降序），在SPARQL中可描述为“ORDER BY ASC/DESC(?v) SKIP k LIMIT n”。下面将排序推理问题总结为4个槽位，如表7所示。

表7. 排序推理槽位及说明

名称	说明
排序变量	“?v”为排序依据的变量名
排序规则	“ASC/DESC”为升序和降序
范围	位置取序列中k+1到k+n的实体
约束	Base的路径均为约束

结合任务的特点，默认$k=0, n=1$。排序推理的关键是确定排序变量和排序规则，针对Base路径中的每个推理变量，都可设置两种排序规则。同上，形如“ORDER BY ASC(?distance) LIMIT 1”的SPARQL句法会被改写为“?distance <排序> <升序>.”的三元组描述形式。对路径添加排序操作用OB来表示，其过程如表8所示。

表8. 排序示例

Base类型	新路径	新类型
{I-O_0_1, I-S-S_0_1}(2)-V	?a <类型> <酒店> . <天坛公园> <附近> ?cvt1 . ?cvt1 <实体名称> ?a . ?a <平均价格> ?price . ?cvtf1 <距离值> ?distance . ?a <押金> ?price1 . ?price <排序> <降序> .	{I-O_0_1, I-S-S_0_1}(2)-V-OB
{I-O_0_1, I-S-S_0_1}(2)-V-F	?a <类型> <酒店> . <天坛公园> <附近> ?cvt1 . ?cvt1 <实体名称> ?a . ?a <平均价格> ?price . ?cvtf1 <距离值> ?distance . ?a <押金> ?price1 . ?price <小于> 800 . ?distance <排序> <升序> .	{I-O_0_1, I-S-S_0_1}(2)-V-F-OB

因推理变量的补充可能引入一些不必要的约束，例如表8中正确路径中的“?a <押金> ?price1.”，该约束并没有参与答案路径的构建，也没有参与过滤和排序，虽然不影响路径查询的结果，但需要将这类多余的约束从路径中删除，减少对匹配排序的影响。

SPARQL语句可以通过路径和答案变量来生成，对路径中的过滤、排序约束的特殊描述转换成SPARQL的“FILTER”和“ORDER BY”句法即可。

3.3 路径匹配和排序

方案使用预训练模型的句对分类任务来做路径和query之间的匹配和相似度打分。使用了ERNIE[6], BERT[7], BERT-wwm[8], Roberta-wwm[9]四类不同的预训练模型，以及他们的融合结果。其中的关键点如下：

1. 负采样训练

对于每一个问题而言，它在知识库中所对应的路径几乎是唯一的。但是上游模块抽取的候选路径规模可能会十分庞大。如果全量抽取的候选路径全部拿来作为负样本来训练模型，会有严重的数据不平衡问题，这在句对分类任务中会表现得更加地显著。最简单的一种负采样方式是，直接从每个query对应的候选路径(非正确路径)中，随机抽取n个样本，作为负样本来训练模型。这种可能存在的问题是，某一些典型的负例，由于采样数量的限制没有覆盖到，导致模型对这一类的预测会比较差。改进的方案是，在不同的epoch中，不断地更新样本。一个比较直观地做法是，在每一个epoch训练完后，对所有的负样本进行一次全量的预测，找出预测结果较差的负样本，来加入到下一轮的训练样本中。不过，若负样本数量爆炸，全量预测在时间上并非完全可行。另一方面，从阻止训练轮次过多导致模型对正样本过拟合的角度来考虑，即便预测结果较差的负样本数量比较多，增加训练轮次和增加负样本也是有限制的。在上述考虑的基础上，改进方法是在每个epoch后，不会对所有负例进行一次全量的预测，而是从全量的负例中采样N条数据来进行结果预测，再从中选择预测较差的结果加入到下一轮的训练中。

即便在负采样训练时，为了缓解正负样本的不平衡，减少了负样本的数量。但就整个数据的分布而言，负样本远远多于正样本仍是客观存在的。如果过分地追求正负样本的平衡，可能导致样本中负样本数量太少，模型对于负数据的训练不够充分。因此，需要兼顾负样本的数量，同时需要模型对于正样本给予足够的重视。这里采用的做法是，对于正样本，增加它在损失函数中的权重。以交叉熵损失函数为例，常规的形式是$\frac{1}{N}\sum_i -[y_i log p_i + (1-y_i)\log(1-p_i)]$，在对正例加权后，损失函数变为$\frac{1}{N}\sum_i -[w \cdot y_i log p_i + (1-y_i)\log(1-p_i)]$。

2. 路径特征抽取

通常而言，在BERT这类的预训练模型中，句对任务的输入是两个自然语言句子。上游模块构建的路径，是包含有变量的三元组的组合，比如“<刘德华>_<妻子>_?b, ?b_<出生日期>_?answer”，它并非是一种自然语言句子。这里采用了三种不同的方式来对路径进行转换，然后输入到模型中。第一种方式是，根据路径的不同类型来设计模板，构建自然语言句子，比如针对上述路径，模板可以是“<subject>的<predicate_1>的<predicate_2>是什么？”，

对应的路径特征为“刘德华的妻子的出生日期是什么?”。第二种方式是，将所有的中间变量(比如上面例子中的“?b”)去掉，把答案变量“?answer”用特殊符号(比如“^”)代替后直接拼接，上述路径经过转换后对应的特征是“刘德华妻子出生日期^”。第三种方式从预训练模型训练时的特性来考虑，可以把对应的答案节点当做是被mask掉的内容，用预训练的特殊字符[MASK]来表示，而中间变量是一种未知的字符，用[UNK]来表示。三元组内部用逗号拼接，不同的三元组之间采用点号拼接。上述例子对应的特征是“刘德华,妻子,[UNK]. [UNK],出生日期,[MASK].”。三种不同的路径构建方案用于训练多个不同模型，最后做模型融合。

3. 评估指标

尽管路径匹配是一个二分类的任务。但最终需要按照分数对路径进行排序，选择打分最高的路径。测试集上，我们选择了三种与排序相关的指标来衡量模型训练的好坏，分别是AUC, HIT@1和 MRR。第一个是从二分类任务的角度来考虑模型效果，AUC指标在一定程度上能够衡量排序的好坏；剩下的HIT@1和MRR都是rank任务中常见的指标。

4 实验和结果

本文方法在CCKS 2021 生活服务知识图谱问答评测任务上进行了评估。主办方提供PKU-Base作为知识库，并提供了6525条标注了SPARQL语句和答案的数据作为训练集，B榜测试集上提供了1191条问题。不同于往年的任务，今年的数据中包含了带有约束条件的生活服务类的问题。

4.1 实体/属性值抽取召回率评估

以官方提供的训练集作为实体/属性值/数值抽取的评估数据集，测评整个模块针对实体和属性值(数值类型放在属性值里面评估)召回率，如下表9所示。

表9. 实体/属性值的召回评估结果

	实体召回	属性值召回	整体召回
Baseline	93.5%	81.1%	92.1%
Baseline + 模糊匹配	94.9%	81.1%	93.5%

4.2 路径挖掘的召回率评估

以训练集的query和SPARQL中抽取的实体/属性值/数值作为输入，在不受上游实体/属性值抽取模块召回的影响下，单独地评估该模块对正确路径的覆盖和召回情况，6525条路径中，未能抽取出正确路径的query有189条，路径整体的召回率为97.1%。

4.3 路径匹配和路径排序评估

由于全量的候选路径过于庞大，另外，上游模块的迭代更新会影响路径匹配的评估。这里固定了一个候选路径集合，在上游模块提供的候选路径集合不变的情况下，对不同参数和不同预训练模型进行了比较。比较采用的是A榜上提交的F1值，打分并不高的原因是比较只使用了A榜中的部分问题，以及上游模块提供的部分候选路径。由于上游模块提供的候选路径不同，不同表格之间的F1比较是没有意义的。AUC, HIT@1和MRR指标的比较如表10所示。BERT, Roberta-wwm和ERNIE在同等情况下的比较如表11所示。随机负采样与改进后的负采样的比较如表12所示。

额外地，比较了ERNIE模型和多模型融合前后的结果，基础的ERNIE模型的打分为0.5214,多模型融合后打分为0.5409。在A榜上，同时在最终的结果上，以BERT作为基础模型，和多模型融合后的结果进行了比较，结果如表13所示。

选定模型的参数后，区分生活服务的问题和一般问题，分别训练了两类模型，分别预测结果。在A榜数据集上，生活服务部分的融合模型打分为0.1952 (query占比500/2100)，一般问题部分的打分为0.5591 (query占比1600/2100)， 总的F1值为0.7543。在B榜数据集上，合并后整体打分为0.7885。

表10. 不同评估指标的F1值

	F1
AUC + BERT	0.5281
HIT@1 + BERT	0.4975
MRR + BERT	0.4975

表11. 不同预训练模型的F1值

PTM	**F1**
BERT	0.5369
Roberta-wwm	0.5509
ERNIE	0.5592

表12. 不同负采样对应的F1值

	F1
随机负采样	0.5369
改进后的负采样	0.5712

表13. 不同模型的F1值

	F1
BERT (Baseline)	0.7249
模型融合	0.7543

5 结论

本文针对CCKS 2021生活服务知识图谱问答评测任务，提出了一种基于路径挖掘、匹配和排序的能够处理简单问题和带约束问题的知识库问答的方法。该方法将约束转换为一种特殊的三元组构建路径，通过实体/属性值/数值抽取、路径挖掘、路径匹配和排序三个模块后，获取到打分最高的路径，将对应的答案作为最后的返回结果。本方法在CCKS 2021生活服务知识图谱问答评测的B榜测试集上取得了0.7885的F1值。

参考文献

1. Steedman, M., *A very short introduction to CCG.* Unpublished paper. http://www. coqsci. ed. ac. uk/steedman/paper. html, 1996.

2. Berant, J., et al. *Semantic parsing on freebase from question-answer pairs*. in *Proceedings of the 2013 conference on empirical methods in natural language processing*. 2013.

3. Yih, S.W.-t., et al., *Semantic parsing via staged query graph generation: Question answering with knowledge base.* 2015.

4. Yao, X. and B. Van Durme. *Information extraction over structured data: Question answering with freebase*. in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2014.

5. Bordes, A., S. Chopra, and J. Weston, *Question answering with subgraph embeddings.* arXiv preprint arXiv:1406.3676, 2014.

6. Sun, Y. , Wang, S. , Li, Y. , Feng, S. , & Wu, H. . (2019). Ernie: enhanced representation through knowledge integration.

7. Devlin, J. , Chang, M. W. , Lee, K. , & Toutanova, K. . (2018). Bert: pre-training of deep bidirectional transformers for language understanding.

8. Cui, Y. , Che, W. , Liu, T. , Qin, B. , Yang, Z. , & Wang, S. , et al. (2019). Pre-training with whole word masking for chinese bert.

9. Cui, Y. , Che, W. , Liu, T. , Qin, B. , Yang, Z. , & Wang, S. , et al. (2020). Revisiting Pre-Trained Models for Chinese Natural Language Processing.

附录：路径挖掘的主要类型

路径类型	示例 SPARQL
I-S	select ?x where { <新加坡> <水域率> ?x. }
I-O	select ?x where { ?x <代表作品> <悼李夫人赋>. }
I-S-S	select ?y where { <显微镜> <发明人> ?x. ?x <职业> ?y. }
I-S-O	select ?x where { <那些年_（《那些年，我们一起追的女孩》主题曲）> <歌曲原唱> ?y. ?x <歌曲原唱> ?y. }
I-O-S	select ?y where { ?x <主要奖项> "2012 汤姆斯杯冠军". ?x <入选国家队> ?y. }
I-O-O	select ?y where { ?x <出生地> <湖北> . ?y <创始人> ?x . }
{I-O_1_0,I-S_1_0}(2)	select ?x where { ?x <职业> <企业家> . ?x <毕业院校> <南京大学> . } select ?x where { <无间道 I> <主演> ?x . <一代宗师_（2013 年王家卫执导电影）> <主演> ?x. } select ?x where { <腾讯> <创始人> ?x. ?x <星座> <天蝎_（黄道十二宫之一）>. }
{I-O_1_0,I-S_1_0}(2)-S	select ?x where { <神玉艺术馆> <位于> ?cvt. ?cvt <实体名称> <北海公园>. ?cvt <方向> ?x }
{I-O_1_0,I-S_1_0}(3)	select ?x where { ?x <作词> <海源>. ?x <作曲> <海源>. ?x <演唱> <海源>. }
{I-O_1_0,I-S_1_0}(3)-S	select ?y where { ?x <组成> <瞿塘峡>. ?x <组成> <巫峡_（长江三峡第二峡）>. ?x <组成> <西陵峡>. ?x <门票价格> ?y. }
{I-S-S_0_0,I-O_1_1}(3)	select ?y where { <大唐不夜城> <附近> ?cvt. ?cvt <实体名称> ?y. ?y <是否有接站服务> <有接站服务（收费）>. ?y <类型> <酒店>. }
{I-S-S_0_0,I-O_1_1}(3)-V-F	select ?y where { <孤山公园> <附近> ?cvt. ?cvt <实体名称> ?y. ?cvt <距离值> ?distance. filter(?distance <= 5). ?y <城市> <杭州>. ?y <类型> <酒店>. }
{I-S-S_0_0,I-O_1_1}(2)	select ?x where { <首都宾馆（原首都大酒店）> <附近> ?cvt. ?cvt <实体名称> ?x. ?x <类型> <景点>. }
{I-S-S_0_0,I-O_1_1}(2)-V-F	select ?x where { <首都宾馆（原首都大酒店）> <附近> ?cvt. ?cvt <实体名称> ?x. ?cvt <距离值> ?d. ?x <类型> <景点>. filter(?d<=3) }
{I-O_1_0,I-S_1_0}(2)-V-F	select ?x where {?x <城市> <济南>.?x <景点的等级 A> "5A".?x <平均价格> ?price filter(?price < 100) }
{I-S-S_0_0,I-O_1_1}(2)-V-F-OB	select ?y where { <奥林匹克森林公园> <附近> ?cvt. ?cvt <实体名称> ?y. ?cvt <距离值> ?distance. ?y <类型> <酒店>. ?y <平均价格> ?price. filter(?distance <= 5). } ORDER BY desc(?price) LIMIT 1
{I-S-S_0_0,I-O_1_1}(2)-V-OB	select ?distance where { <西安梦幻星空抖乐园（钟楼旗舰店）> <附近> ?cvt. ?cvt <实体名称> ?y. ?cvt <距离值> ?distance. ?y <类型> <酒店>. } ORDER BY ASC(?distance) LIMIT 1
{I-S-S_0_0,I-O_1_1}(2)-S-V-OB	select ?distance where { <神玉艺术馆> <附近> ?cvt. ?cvt <实体名称> ?y. ?cvt <距离值> ?distance. ?y <类型> <酒店>. } ORDER BY ASC(?distance) LIMIT 1
{I-S-S_0_0,I-O_1_1}(2)-S-V-F-OB	select ?price where { <故宫博物院（故宫）> <附近> ?cvt. ?cvt <实体名称> ?y. ?cvt <距离值> ?distance. ?y <类型> <酒店>. ?y <平均价格> ?price. filter(?distance <= 2). } ORDER BY asc(?price) LIMIT 1