文章编号：2095-2163(2020)05-0001-05 中图分类号：TP391.1 文献标志码：A

基于问题生成的知识图谱问答方法

乔振浩，车万翔，刘 挺
（哈尔滨工业大学 计算机科学与技术学院，哈尔滨 150001）

摘 要：基于知识图谱的问答系统是一种新兴的问答模式，能够用准确的答案回答用户提出的事实类问题，但目前广泛使用的基于深度学习技术的问答系统无法在缺少标注数据的情景下工作。针对这一问题，本文提出了基于问题生成的知识图谱问答方法。该方法无需成本高昂的标注数据，编写简单的模版文件即可构建问答系统。在自建哈尔滨工业大学学校信息数据集上测试，相比于深度学习的基准方法，本方法在无标注数据情景下具有可用性。

关键词：知识图谱；问答系统；问题生成

Question answering method based on question generation

QIAO Zhenhao, CHE Wanxiang, LIU Ting
(School of computer science and technology, Harbin Institute of Technology, Harbin 150001, China)

【Abstract】 The knowledge graph based question answering system is a new question answering mode, which can answer the fact questions raised by users with accurate answers. The widely used question answering system based on deep learning technology cannot work in the absence of labeled data. Aiming at this problem, this paper proposes a question answering method based on question generation. This method does not require costly annotation data, and a simple template file can be written to construct a question answering system. On the self-built Harbin Institute of Technology school information data set, compared with the deep learning benchmark method, the test results show the usability of this method without labeling data.

【Key words】 knowledge graph; question answering; question generation

0 引 言

问答系统是自然语言处理的明日之星，已获得了人工智能及其相关产业的广泛关注，并已经在互联网、医疗、金融等领域进行了应用尝试。与目前主流资讯检索技术相比，问答系统有两点不同：首先，查询方式为口语化的问句，使用者不需要思考该使用什么样的问法才能得到理想的答案；其次，查询结果为高精度的网页或明确的答案字符串。

问答系统依赖于优质的知识。依据知识的形式，问答系统可以分为基于文本的问答和基于知识图谱的问答。基于文本的问答，知识通常来源于纯文本，如百科文本、社区问答以及网络文档等；基于知识图谱的问答，知识来自以 RDF 三元组格式存储的知识。相比于文本知识，对于事实型问题（when、who、where、which、what），知识图谱问答较文本问答更为精确和高效，同时也要求知识图谱规模足够大。语义网络技术和自动信息处理系统的发展进步催生了大规模知识图谱。近些年涌现出了大批十亿甚至更大规模的知识图谱，包括 Freebase、DBpedia 等，这些知识图谱的出现使基于知识图谱的问答系统变得可行。

由于深度学习技术在自然语言处理领域的大放异彩，目前很多知识图谱问答系统都是基于各种深度学习模型进行构建，训练深度学习模型需要大量的标注数据[1]。然而在实际应用中，往往只能获取到知识图谱而缺乏标注数据，造成这种困境的原因是相比于构建知识图谱，构造标注数据的成本更高。知识图谱以三元组形式组织，其构建过程并不需要数据库专业知识，当应用场景明确且规模适当时，针对特定应用场景抽取三元组即可完成知识图谱的构建。标注数据由自然语言问句与对应逻辑表达式构成。自然语言问句需要覆盖大部分的知识库三元组，并且要兼顾表达多样性的需求。问句对应的逻辑表达式则需要具有专业知识才能写出，需要专业人士的参与，获取成本较高。

针对上述问题，本文提出一种基于问句生成的知识图谱问答方法。该方法通过模版将知识图谱三元组转换为问句形式，并为生成的问句建立全文索引。当用户查询时，系统首先通过全文检索引擎检索出候选集合，然后通过语义匹配模型为候选集合打分，挑选出得分最高的候选项，最后把候选项对应

作者简介：乔振浩（1994-），男，硕士研究生，主要研究方向：自然语言处理；车万翔（1980-），男，博士，教授，博士生导师，主要研究方向：自然语言处理；刘 挺（1972-），男，博士，教授，博士生导师，主要研究方向：自然语言处理、信息检索、社会计算等。

收稿日期：2020-03-05

的三元组作为答案返回给用户,完成问答操作。通过这种方式,可以在缺少标注数据集的情况下,快速开发出可用的知识库问答系统。经过实验分析,在无标注数据的情况下,本文的提出的模型相比于深度模型有更好的可用性,有助于知识图谱问答系统的推广应用。

1 模版定义

模版是将知识库三元组转换为自然语言表达的工具。给定由三元组(实体 1,关系,实体 2)组成的知识图谱,为一个关系或语义相近的一组关系编写一套模版,模版中包含预留实体槽位的自然语言问句。问句生成时,处理程序依次读入三元组,将三元组解析为(实体 1、关系、实体 2)的形式,从模版库中选取该关系下的全部模版,使用相应实体替换问句中的实体槽位,生成出自然语言形式的句子,生成出的自然语言语句与三元组保持一一对应的关系。以哈尔滨工业大学信息问答为例,以下为 JSON 格式模版的实例:

```
"创办时间":{
  "alias":["创办时间"],
  "templates":[
    "$1 是什么时候创办的?",
    "$1 建校时间",
    "$1 成立于哪一年?",
    "$1 创办时间?",
    "$1 建校于哪一年?",
  ]
}
```

该 JSON 格式的文本中,JSON 键值“创办时间”对应了知识图谱三元组中的关系。JSON 值域中,“alias”代表共享该模版的三元组关系名,凡是包含此类关系的三元组都会通过这条模版进行扩展。“templates”为具体的模版。模版中 $1、$2 分别代表了实体 1 和实体 2。扩展时,程序依次读入的三元组,将三元组解析为实体 1、关系、实体 2 的形式。程序从模版库中选取该关系下的全部模版,使用相应实体替换 $1、$2 符号,生成问句。例如:对于三元组“<哈尔滨工业大学> <创办时间> <1920 年>”将被扩展为“哈尔滨工业大学是什么时候创办的?”、“哈尔滨工业大学建校时间”、“ 哈尔滨工业大学成立于哪一年?”、“哈尔滨工业大学创办时间?”、“哈尔滨工业大学建校于哪一年?”五句语义信息完整的句子。

下面介绍两种模版构造的方法:

(1)从数据集中抽取模版。公开的开放域知识库数据集中包含了丰富的自然语言表达和对应的逻辑表达式。通过替换自然语言问句中的实体为特殊标签 $1,并根据关系名称进行聚类,就可以获得丰富的模版表达。

(2)手工构造模版。开放域知识库问答数据集覆盖的范围较广,但对具体的场景无法全面覆盖。因此,除了从数据集中抽取模版外,还需通过手工构造模版的方式,来提高模版精度,保证大部分的三元组数据都有较好的覆盖,并满足问句多样性的要求。若知识库问答的场景比较新颖,例如:高校信息问答、新冠肺炎知识问答,则主要依赖这种方法。

2 候选集检索与语义匹配

将三元组转换为自然语言问句形式后,当用户输入查询时,只需从生成的句子中选择出最相关的句子并返回对应的三元组即可完成问答操作。通过这种方式,实际上是将知识库问答问题转换为了 FAQ 问题。由于使用模版生成问句时存在排列组合问题,生成的自然语言问句规模比较庞大。为了有效地检索出与用户问句相关的句子,首先采用基于全文检索的检索方法进行粗匹配,缩小候选答案的范围,然后使用基于语义匹配的方式进行细匹配,在语义层面挑选出最佳答案。

2.1 基于全文检索的检索算法

本方法首先使用基于全文检索的检索方式来检索得到候选关系集。在信息检索系统中,文本数据按照存储方式可以分为结构化数据和非结构数据。结构化数据是指具有明确格式,定义清晰的有限长度数据,如数据库、XML 数据。对结构化数据检索时,由于其数据格式定义明确,可以使用规范的查询语句,如 SQL 语句进行查询。非结构数据又称为全文数据,通常具有长度不固定,格式不统一的特点。对此类数据检索最朴素的方式是顺序扫描法,查找时将关键词与全部文本逐个进行对照,直至遍历完全部文档。当处理小数据量文本时,这种方法简单而有效。对于大量的文本顺序扫描的时间复杂度非常高,需要采用更加高效的方法。全文检索的基本原理是从非结构的文本中提取出关键信息,并按照一定规则重新排列为结构化的形式存储,这些提取出的结构化信息称之为索引。检索时根据这些结构化的索引进行检索,从而达到加速的目的。这种先建立结构化信息索引,再通过索引加速搜索的过程就称为全文检索。全文检索应用时分为建立索引和查找索引两个部分:建立索引负责预处理待检索的

文本，根据词法知识对待检索文本进行解析后创建索引；查找索引将用户查询文本进行同样的词法预处理后，检索已创建的索引并返回给用户查询结果。

建立索引时第一步是通过分词组件将文本处理为有意义的独立词元(Token)。这个过程需要将文档分为一个个单词的形式。由于中文文本中字与字之间没有分割符，需要借助外部分词工具进行处理，常用的中文分词工具有 LTP 平台，THULAC 分词等。而对于英文文本，直接使用空格符号进行分割即可，然后去除分词结果中的标点符号，通常使用正则匹配的方法即可完成，最后去除停用词，停用词在文本中会大量出现并且没有实际的含义，会干扰正常的搜索。

第二步处理中，通过语言处理组件将第一步得到的词元进行自然语言相关处理。以英文为例，处理包括：1)大小写转换；2)单词缩减为词根形式，例如：将“games”改写为“game”；3)单词替换为词根形式，如：将“played”替换为“play”。处理后得到的结果称为词(Term)。经过分词组件处理后，使用得到的 Term 创建倒排索引。倒排索引由全部不重复的词构成，每一个词都对应了一个包含该词的全部文档列表。

使用倒排索引技术检索出的结果从众多文档中筛选出了与用户查询有相同词汇的候选文档，然而这些候选项只能保证与查询句是相关的，而无法定量地确定与问句的相关程度。为了准确地反应出候选文档与问句的相关程度，使用 TF-IDF(Term Frequency-Inverse Document Frequency，词频-逆文档频率)对检索出的结果项进行相关度排序。TF-IDF 是用来估计单词在文档库中的重要程度的指标，单词的重要程度与它在文档中出现的次数成正比，但同时又会与它在整个文档库中出现的频率成反比[2]。

词频代表了一个词在某篇文档中出现的频率。在一篇文档中，一个词出现的次数越多，说明这个词对于这篇文档的重要程度越高，越能代表这篇文档。然而仅凭词数来衡量重要程度会出现不相关的长文档得分远高于精炼的短文档得分的问题。词频是词数进行归一化后得到指标，能够无偏地反映词的重要程度。对于某一文档 m 中的词语 i，其词频计算方式如式(1)：

$$\mathrm{TF}_{i,m}=\frac{n_{i,m}}{\sum_j n_{j,m}}. \tag{1}$$

其中，$n_{i,m}$ 是词 i 在文档 m 中出现的次数，分母是文档 m 中全部单词计数之和。逆文档频率代表了词的普遍性程度。越多的文档包含该词，说明这个词越普遍，不足以区分这些文档。对某一特定词语 i，其 *IDF* 值由文档总数除以包含该词的文档数目，再将结果取对数得到式(2)：

$$\mathrm{IDF}_i=\lg\frac{|D|}{|m;i\in d_m|+1}. \tag{2}$$

其中，$|D|$ 代表文档总数，$|m;t_i\in d_m|$ 代表包含词 i 的全部文档数目。分母处+1 是为了防止分母为 0。得到 TF 与 IDF 值后，文档 m 中单词 i 的 TF-IDF 指标的计算方法如式(3)：

$$\mathrm{TF-IDF}_{i,m}=\mathrm{TF}_{i,m}\times\mathrm{IDF}_i. \tag{3}$$

由该公式可以看到，TF-IDF 值平衡了一个单词在某一特定文档和全部文档出现频率的关系，使重要的词语得分高于常见的词语。计算出句子中每个词的 TF-IDF 值后，按顺序排列，得到了该句子的向量。通过计算查询向量 V_q 与候选项向量 V_d 的余弦相似度，就可以判断查询与候选句之间的相似性，公式(4)：

$$\mathrm{score}(V_q,V_d)=\frac{V_q\times V_d}{|V_q|\times|V_d|}=\frac{\sum_{i=1}^{n}V_{qi}\times V_{di}}{\sqrt{\sum_{i=1}^{n}V_{qi}^2}\times\sqrt{\sum_{i=1}^{n}V_{di}^2}}. \tag{4}$$

本方法在开源的全文检索软件库 Lucene 上进行二次开发。Lucene 是由 Apache 软件基金会支持的一套用于全文检索的开源程序库，提供了强大且易用的应用程序接口。Lucene 是目前最广泛使用的免费全文检索程序库。

2.2　语义匹配方法

通过检索模块召回的候选问句只保证了字符级别的重合，为了保证问答效果，需要进一步进行语义级别的匹配，使用预训练模型 BERT 进行语义匹配。预训练语言模型是近年来 NLP 领域最激动人心的进展，包括 ELMo、ULMFiT 及 BERT 等。这些预训练模型可以显著改善下游任务的表现，让 NLP 领域进入了预训练模型的“新时代”。可以从海量无标注数据中学习到潜在的深层语义信息，并且可以通过微调的方式将这些知识迁移到下游任务中，不再需要大量单独标注额外的训练数据。下面介绍 BERT 的原理及其在语义匹配任务上的微调方法。

BERT(Biderectional Encoder Representations)是基于多层双向 Transformer 的深度双向语言模型[3]。相较于传统的 RNN 类循环神经网络，Transformer 有更好的并行运算性能。BERT 使用掩码语言模型进行训练，这种双向预训练方式相比于 ELMo 等使用

单向的模型,可以更好地学习上下文信息。同时,BERT 还使用了下一句预测任务,模型需要根据上下文信息预测两句是否为连续的,该任务使模型具有处理句子的能力。

具体来说,在 MLM 掩码语言模型中,BERT 会随机将 15%Token 替换为[MASK]标签。模型需要根据被替换位置的隐层向量来预测该词。尽管这种方法可以完成双向预训练模型的训练,然而这种方法有两个缺点。第一是预训练过程和微调过程不匹配。在微调时,[MASK]标签并不会出现。为了缓解这种情况,在挑选出 15%Token 后,80%的情况下用[MASK]进行替换,10%的情况随机替换为其他词,10%的情况下保持不变。这样做的目的是使模型适应微调阶段的情况;第二是在预训练过程中,输入模型的一个 Batch 中只预测 15%的 Token,会导致训练效率降低,耗费更多时间。不过 BERT 在众多任务上出色的指标提升说明了虽然模型的收敛速度不如传统的从左到右的语言模型,但模型带来的性能提升远远大于增加的时间成本。

此外 BERT 的训练还包括下句预测的二分类任务,从而赋予模型处理句子级别输入的能力。下句分类的目标是判断输入模型的两个句子是否连续,即判断两句是否具有相同或相近的语义表达。该任务的输入可以从任意的单语语料库中生成,具体来说,挑选出句子 A 后,预处理程序以 50%的概率选择句子 A 的下一句话作为句子 B,50%的情况下从预料库中随机选取任意一句话作为句子 B。本文的语义匹配步骤正是依赖于 BERT 模型的句对分类的能力。

由上述两个任务可以很自然地推测出 BERT 的输入为单个句子或是句子对。对于文本中的每一个词,其输入表示为词嵌入、段嵌入和位置向量的和。词嵌入使用了 WordPiece 技术,将一个单词拆分为更小的单位,这样同一词根不同形式(如时态变化)的单词将共享部分相同的表示,缩小了词表大小,有益于模型收敛。段嵌入向量中只包含 0、1 两个值,该向量是为了区分句对而设置的。由于 Transformer 编码器在输入文本上并行计算,遗失了单词的位置信息,故使用位置向量来补充位置信息。此外,BERT 模型的输入中还定义了两个特殊符号[CLS]与[SEP]。[CLS]符号位于输入的起始位置,在进行句子级别的分类任务时,可以使用该符号对应的向量进行预测。[SEP]符号用于分隔输入的两个句子,与位置向量配合使用。

Google 提供的预训练模型与本文要完成的语义匹配任务仍有一些差距,需要经过微调操作把 BERT 从海量数据中学习到知识迁移到目标领域。对于句子级别的匹配任务,微调时将输入“[CLS]”所对应的最后一层隐层向量 $V \in \mathbb{R}^H$,经过一层线性变换后送入 softmax 函数,计算类别得分即:$P = \mathrm{softmax}(V * W^{\mathrm{T}})$,式中 $W \in \mathbb{R}^{K \times H}$,$P \in \mathbb{R}^K$,$K$ 为分类器标签数目。在语义匹配任务中,K 取 2。使用 LCQMC 数据集与生成的问句进行微调。LCQMC 是中文口语话描述的语义匹配数据集,该数据集定位与本文的任务比较相符。但是该数据集面向领域为开放域问答,问句类型与限定领域问答仍有一定差距。因此本文使用模版生成的问句做了一组微调数据:将生成的问句两两拼接为句对,由同一关系生成的句子标注为正例,不同关系生成的句子标注为负例。语义匹配模型在这两个数据集上进行微调,微调网络结构如图 1 所示。

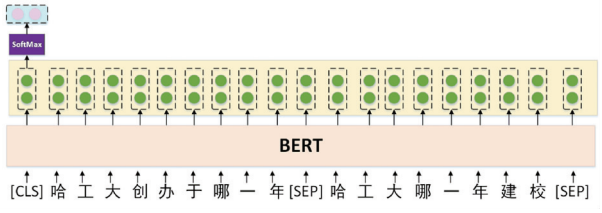

图 1 BERT 微调

Fig. 1 BERT fine-tuning

本文提出了一种无标注数据下快速构建知识图谱问答系统的方法。该方法首先针对知识图谱的应用场景编写模版文件。模版文件结构简单,不需要编写者掌握专业的知识。模版以自然语言形式组织,只需标注者具有应用场景的基本常识即可,标注成本极低。经程序使用知识图谱三元组填充模版后,可以得到具有完整语义信息的问句表达,使用检索工具包 Lucene 为生成的问句建立全文索引。当用户输入查询时,系统首先使用全文检索引擎检索出相关的问句,然后使用预训练模型为候选项打分。这样既保证了检索的速度,又保证了检索结果与查询句语义层面相匹配。最后,系统返回语义匹配模块得分最高项对应的三元组就完成了整个问答操作。本文所提出的方法最大的优点就是不需要标注数据,编写模版的人工成本很低,并且充分发挥了预训练模型的语义匹配能力,有效地提升问答效果。系统流程示意图如图 2 所示。

3 评价指标与实验结果

3.1 实验数据集的构造

本文提出了一种限定领域下无标注数据的知识图谱问答方法。对于无标注数据场景下的系统评价

采用自行构建测试数据的方法进行。首先，人工构建了面向哈尔滨工业大学百年校庆知识问答的知识图谱数据，该知识图谱包含哈尔滨工业大学基本信息、院系专业、长江学者等 4 个领域，共 2922 组三元组数据。然后通过在线文档的方式收集并标注了 250 组标注数据，标注数据包含问句与对应的答案。

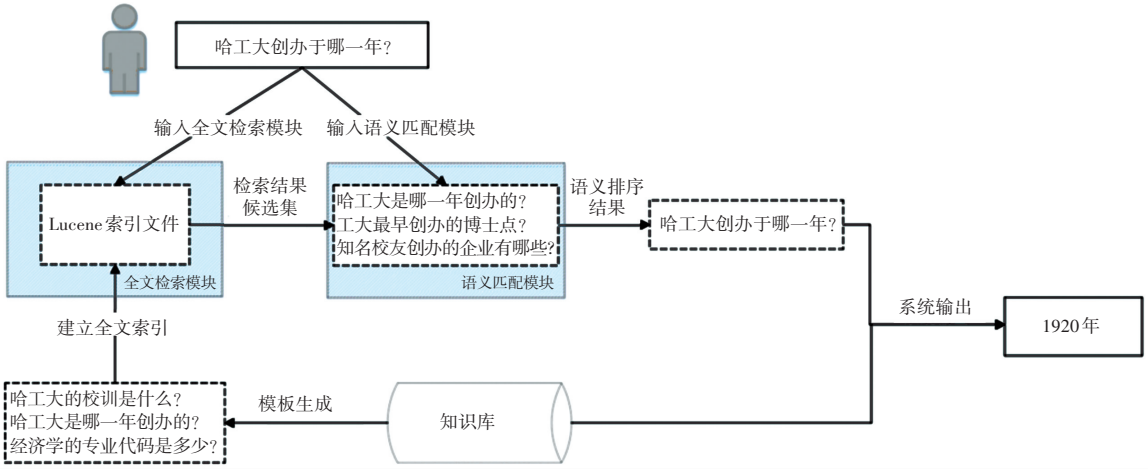

图 2　系统流程示意图

Fig. 2　System flow diagram

3.2　评价指标

基于知识图谱的问答系统使用宏观准确率 P（Macro Precision）与宏观召回率 R（Macro Recall）以及 Averaged F1 值对模型性能进行评价。设 Q 为问题集合，A_i 为问答系统对第 i 个问题给出的答案集合，G_i 为第 i 个问题的标注答案集合，这 3 个指标的定义如公式(5)~公式(7)：

$$P = \frac{1}{|Q|}\sum_{i=1}^{|Q|} P_i, \quad P_i = \frac{|A_i \cap G_i|}{|A_i|}. \tag{5}$$

$$R = \frac{1}{|Q|}\sum_{i=1}^{|Q|} R_i, \quad R_i = \frac{|A_i \cap G_i|}{|A_i|}. \tag{6}$$

$$F1 = \frac{1}{|Q|}\sum_{i=1}^{|Q|} \frac{2P_iR_i}{P_i + R_i}. \tag{7}$$

3.3　实验结果与分析

使用基于语义图的深度学习模型作为实验的 baseline。该模型通过建模语义图的方式将自然语言问句映射为 SPARQL 语句。实验结果如表 1 所示。表 1 显示了本文提出的方法与深度模型在此数据集上的效果的对比情况。从 $F1$ 值上可以看到基于问句生成的知识图谱问答方法相比于深度学习模型有显著的提高。这是由于深度模型需要大量的训练样本，在无标注、少量标注数据情况下很难奏效，而本文提出的方法克服了这一缺点，无需标注数据也可以取得较好的效果。进一步补充了消融实验，消融实验的结果显示了两种微调数据给模型带来的效果提升。在 LCQMC 数据集上进行微调给模型带来了小幅度的提升，而使用生成的问句进行微调对模型影响较大。这说明了两个问题：预训练模型确实有助于下游任务的表现；微调时，需要选择与下游任务紧密结合的数据。

表 1　实验结果

Tab. 1　Experimental results　%

模型	Precision	recall	F1-score
◆基于问句生成的知识图谱问答	83.66	84.40	84.03
-不使用 LCQMC 进行微调	81.14	81.99	81.56 ▼2.47
-使用生成问句进行微调	76.48	77.41	76.94 ▼7.09
◆基于语义图的深度学习模型	69.59	71.34	70.45▼13.58

4　结束语

基于知识图谱的问答系统由于其准确、便捷的知识获取能力受到了学术界和工业界的关注和重视。然而目前主流的深度学习技术需要成本昂贵的人工标注数据驱动，妨碍了知识图谱问答系统的应用。针对这一问题，本文提出了基于问句生成的知识图谱问答方法，该方法充分利用了文本检索技术与预训练语义匹配模型，只需针对知识图谱编写简单的模板文件即可构造问答系统。实验结果表明，该方法在无标注数据的限定领域内具有良好的问答效果。

参考文献

[1] CHAKRABORTY N, LUKOVNIKOV D, MAHESHWARI G, et al. Introduction to neural network based approaches for question answering over knowledge graphs[J]. arXiv preprint arXiv:1907.09361, 2019.

[2] JOACHIMS T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization[R]. Carnegie-mellon univ pittsburgh pa dept of computer science, 1996.

[3] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.