KBQA 技术总结

作者：纽约的自行车

日期：2021 年 12 月 10 号

1 前言

KBQA（Knowledge Base Question Answer）是指将自然语言转换成知识库查询语句。KBQA 方法主要有三类：模板法 [1-2]、语义解析法 [3-5] 和信息检索法 [1, 6-7]。本文简要介绍每种方法的常见方案。

2 模板法

基于模板的 KBQA 旨在利用预定义的模板匹配问题进而得到形式化查询。通常由离线和在线两个过程组成。离线时，主要根据问答历史建立模板库。具体地，归纳总结以往回答过的问题，构造出问题模板与对应的查询模板。在线时，对于一个新输入的问题，首先将其匹配到模板库中的问题模板，进而得到问题模板对应的查询模板。随后，实例化查询模板，即从问题出提取出相应的语义内容，填充模板得到真正的查询。

系统包括实体识别、模板匹配、关系匹配、答案类型匹配、排序。实体识别和关系匹配组合使用字符匹配、词典、词性方法，答案类型匹配和其他的问答系统方法类似，排序使用机器学习方法，其中的特征非常复杂。

模板匹配的核心就是图1中的 Template [2]，显然模板的形式需依据问题去设计，这里只是举了三个例子。

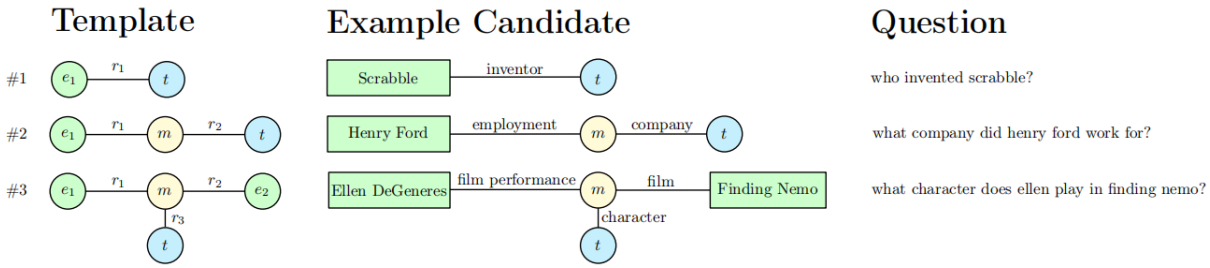

图 1: 模板法

e 表示实体占位符，r 表示关系占位符，m 表示中间变量，t 表示答案。每个问题经过实体识别后得到一组实体提及，将每个实体提及应用到上述模板 #1 和 #2 中，每个模板从知识库中返回一组 query。从实体提及中选择两个用于匹配第三个模板 #3，同样返回一组 query。接下来就是关系匹配，从问题中识别关系，将识别的关系匹配到 query 中的关系，那些没有匹配到关系的 query 被删除。经过关系匹配后剩下的 query 称为 example candidate。图中第一个 example

candidate 对应的问题是 “who invented scrabble?’’，第二个对应的问题是 “what company did Henry Ford work for?’’。对 example candidate 排序得到最佳查询，由此完成从问题到查询的整个过程。

3 语义解析

语义解析方法将问题解析成逻辑形式，然后转化成查询语句。系统可分为四个模块：问题解析、逻辑形式构建、实体链接、查询语句构建。问题解析是从语义和语法层面解释问题，提取关键信息，例如实体提及、关系和限制条件。逻辑形式构建是根据问题解析构建逻辑模板。实体链接是用问题中的实体和关系填充逻辑模板中的符号。查询构建是将逻辑模板转换成查询语句。

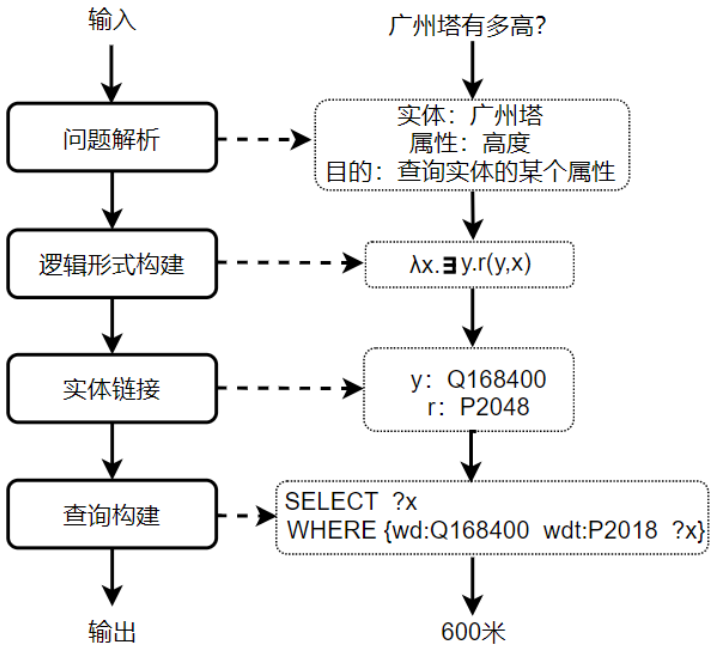

图 2: 语义解析方法

图2是一个完整的语义解析过程。问题 “广州塔有多高” 输入系统中，问题解析过程识别到实体 “广州塔” 和属性 “高度”，这是一个简单问题，只包含一个三元组，因此可以判断问题的目的是想查询实体的属性值。根据解析结果可构建逻辑形式，这种形式没有统一标准，逻辑形式实际上形式化了问题的目的。图中逻辑形式中 x 是查询结果，y 是实体，r 是关系，含义是在知识库中存在 y 并且 y 和 x 存在关系 r。实体链接是将问题中的实体和属性映射到知识库中的实体和属性，在这里广州塔映射到 wikidata 中的 Q168400，高度映射到 wikidata 中的 P2048。根据逻辑形式和实体链接结果就可以构建查询语句。

逻辑形式构建中定义了转换操作，在生成逻辑形式过程中每次选择一个转换操作，这种方法属于早期的语义解析。当前主流的语义解析是查询图法，在查询图法中将逻辑形式构建和实体链接合并为一个过程，利用知识图谱指导生成查询图过程。查询图实际就是图形式的逻辑形式。构建逻辑形式的方法处理简单问题效果不错，但是面对复杂问题会造成逻辑形式数量呈指数型增长，因为多个实体和多个关系相互之间有很多种连接方法。查询图法在解析问题过程中，借助知识库的结构缩小每步的搜索范围，控制查询图生成数量，并且这样生成的查询图属于知识库的子图，不存在匹配错误问题。

无论是构建逻辑形式还是查询图，在这个过程中都会产生多个逻辑形式或查询图，在构建查询语句前还需要对这些逻辑形式或查询图排序，选择最好的用于构建查询语句。排序通常使用深度学习方法，计算查询图与问题的相似度得分。这一步是监督学习，准确的说是弱监督学习，因为数据集没有提供标准的查询图，只有最终的答案。

语义解析的重点在于精确解析问题和排序查询图或逻辑形式。最终构建的查询语句只有一条，所以得到的结果只有一个。

4 信息检索

信息检索方法侧重在答案排序，如何从一组候选答案中选择正确答案是重点也是难点。信息检索一般框架如图3所示。问题 Embedding 模块将问题编码成向量形式；问题 topic entity 识别模块从问题中识别关键实体，为了提高召回率可以将问题中的实体都看作 topic entity；根据 topic entity 从知识图谱中检索候选答案；答案 Embedding 将候选答案分别编码成向量形式；候选答案排序模块计算问题向量和答案向量的相似度得分，最后输出答案。

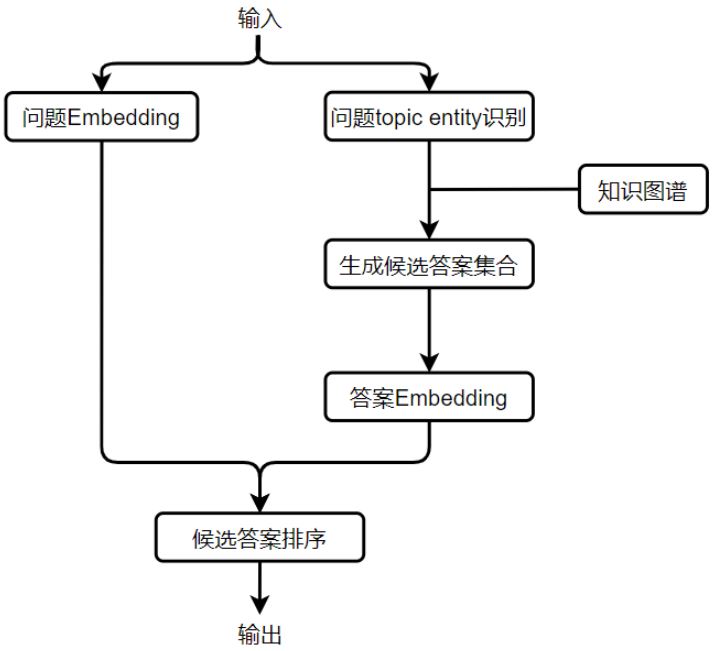

图 3: 信息检索方法

问题 Embedding 模块可采用的方法很多，早期使用 LSTM、CNN 模型，现在流行使用 BERT 等语言模型。

问题 topic entity 识别其实就是实体链接过程。这个模块对应着语义解析方法的问题解析模块，相对于语义解析方法从问题中挖掘很多信息，信息检索只需识别实体即可。

候选答案生成模块使用 topic entity 从知识图谱中检索候选答案。普遍采用的方法是将与 topic entity 有 1-hop 或者 2-hop 的节点视为候选答案。对于简单问题使用 1-hop 即可，对于复杂问题可使用 2-hop，hop 越多生成的候选答案越多，召回率高但是精确度低。在此有个前提是一般认为答案存在于节点而不是关系。

答案 Embedding 生成每个候选答案的向量表示。候选答案只是一个节点，如果仅仅将节点

向量视为答案向量，可能涵盖的信息量不足。普遍的做法是结合了答案的四个方面综合生成答案向量，这四个方面分别是：答案节点、路径上的关系、答案类型和答案节点的上下文。路径是指 topic entity 到答案节点的路径。答案类型是指答案节点的类别，在知识图谱中每个实体节点都有一个类别，属性值也有 string、int、datetime 等这样类别，类别信息是很重要的，如果问题的答案是一个人物，那么类别是人物的答案节点显然得分更高。上下文是指与答案节点有 1-hop 关系的节点和与答案节点相连的关系。如何将节点和关系转换成向量也是个问题，一种做法是使用 TransE 方法训练知识图谱，得到每个节点和关系的向量，另一种做法是使用大规模语料训练好的词向量转化。

得到问题向量和答案向量后，计算两个向量的相似度。相似度计算方法也有很多种，使用 cos 函数是普遍做法。部分问题答案不止一个，如果只返回得分最高的答案显然无法应对这种情况，为此，可以将得分与最高分差距小于 m 的候选答案一起返回。

信息检索方法可以处理多跳多节点的问题，但难以解决含有限制条件的问题。例如问题“最早举办夏季奥运会的亚洲城市”返回的是所有举办过夏季奥运会的城市，因为知识图谱中也不会标注这个信息，并且模型无法理解‘‘最早’’这个条件。

5 总结

基于模板的 KBQA 往往具有较高的精确率，对于能匹配到模板的问题，通常都能正确回答。这是由于高质量的模板可以完整表达出问题的语义。然而，这类方法也始终具有模板生成成本高、模板数量有限、覆盖面不足的问题，成为阻碍性能提升的瓶颈。

目前主流的 KBQA 方法是语义解析和信息检索。语义解析对问题解析成一组精确的查询语句并排序，选择最佳查询语句然后返回最佳答案。语义解析通常需要制定一些规则，用在查询图或逻辑形式构建过程中。信息检索方法使用问题中的 topic entity 从知识图谱中检索大量候选答案，然后结合问题对这些答案排序，返回前 K 个答案。信息检索方法侧重在答案排序，语义解析侧重在问题解析。信息检索方法很难学习到问题中的限制条件，而在语义解析中可以制定规则来解决问题中的限制条件。

参考文献

[1] FADER A, ZETTLEMOYER L, ETZIONI O. Paraphrase-driven learning for open question answering[C]. Sofia, Bulgaria: ACL, 2013, August: pages:1608-1618.

[2] BAST H, HAUSSMANN E. More accurate question answering on freebase[C]. Melbourne, Australiaa: CIKM, 2015, October 19: pages:1431-1440.

[3] SEN H, LEI Z, XINBO Z. A state-transition framework to answer complex questions over knowledge base[C]. Brussels, Belgium: EMNLP, 2018, October 31, November 4: pages:2098-2108.

[4] WENTAU Y, MINGWEI C, XIAODONG H, et al. Semantic parsing via staged query graph generation: Question answering with knowledge base[C]. Beijing, China: ACL-IJCNLP, 2015: pages:1321-1331.

[5] MO Y, WENPENG Y, KAZI H, Saidul, et al. Improved neural relation detection for knowledge base question answering [C]. [S.l.]: ACL, 2017.

[6] YANCHAO H, YUANZHE Z, KANG L, et al. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge[C]. Vancouver, Canada: ACL, 2017, July 30, August 4: pages:221-231.

[7] LI D, FURU W, MING Z, et al. Question answering over freebase with multi-column convolutional neural networks[C]. Beijing, China: ACL-IJCNLP, 2015, July 26: pages:260-269.

[8] ANTOINE B, SUMIT C, JASON W. Question answering with subgraph embeddings[C]. [S.l.]: EMNLP, 2014.