

KBQA: TransferNet 模型的复现和改进

王逸驰

摘要

多跳问答 (QA) 是一项具有挑战性的任务,因为它需要精确的推理问题中每一步的答案。关系可以用知识图谱中的标签 (如配偶) 或文本语料库中的文本 (如结婚 26 年) 来表示。现有的模型通常通过预测顺序关系路径或聚合隐藏的图特征来推断出答案。前者难以优化,后者缺乏可解释性。本复现论文的模型 `transfernet` 同时具有两种特性,且在数据集上表现出了优良的性质,所以选择该论文进行复现。

关键词: 多跳问答; 知识图谱

1 引言

问答 (QA) 在人工智能中扮演着核心角色。它需要机器理解自由形式的问题,并通过分析来自大型语料库的信息来推断答案或结构化知识库。随着深度学习,特别是预训练技术的快速发展。在简单的单跳问题上,模型的能力已经和人类的能力相媲美,然而,需要在多个步骤中对实体关系进行推理的多跳 QA 远远没有解决。因为搜索引擎与问答非常相关,而现代人的生活和工作已经离不开搜索引擎,所以搜索引擎的质量非常重要。所以我认为此选题对现实生活的帮助非常有实际意义。

2 相关工作

2.1 语义解析

基于语义解析方法。这类方法旨在将自然语言解析成逻辑形式 [Berant 和 Liang^[1];Reddy 等人^[2]。它们通过以下步骤预测答案:(1) 通过问题理解模块理解问题,即进行语义和句法分析,并获得编码后的问题,用于后续的解析步骤。(2) 利用逻辑解析模块将编码后的问题转换为未实例化的逻辑形式。未实例化的逻辑形式是没有实体和关系基础的问题的句法表示。逻辑形式的语法和组成部分可以根据系统的具体设计而有所不同。(3) 为了针对 KBs 执行,通过对结构化 KBs 进行一些语义对齐来进一步实例化和验证逻辑形式。请注意,在一些工作中 [Yih et al^[3];Liang 等人^[4]],同时执行逻辑解析和 KB 对齐,其中逻辑形式在部分解析时以 KB 为单位进行验证。(4) 最后,通过 KB 执行模块对 KB 执行解析后的逻辑形式生成预测答案。

2.2 信息召回

另一种主流方法是基于信息召回的方法,考虑到问题中传达的信息,直接从 KBs 检索答案并对其进行排序 [Bordes 等人^[5];Dong 等^[6]。它们包括以下步骤:(1) 从主题实体开始,系统首先从 KBs 中提取出一个特定问题的图表。理想情况下,这个图包括所有与问题相关的实体和关系,分别作为节点和边。基于 ir 的方法无需显式地生成可执行的逻辑形式,而是对图执行推理,然后对图中的实体排序。(2) 接下来,系统通过问题表示模块对输入问题进行编码。该模块分析问题的语义并输出推理指令,这些指令通常用向量表示。(3) 基于图的推理模块通过基于向量的计算进行语义匹配,沿图内相邻实体传播并聚合信息。推理状态在不同的方法中有不同的定义 (例如,预测实体的分布,关系的表示),它

基于推理指令进行更新。最近，几项研究 [Jain^[7];Chen 等^[8]] 多次重复步骤 (2) 和 (3) 进行推理。(4) 利用答案排序模块，根据推理结束时的推理状态对图中的实体进行排序。排名最高的实体被预测为问题的答案。

3 本文方法

3.1 本文方法概述

我们在关系图上进行多跳推理，该关系图以实体为节点，实体之间的关系为边。关系可以有不同的形式，具体来说，约束标签或自由文本。前者也被称为结构化知识图，它预先定义了一组谓词来表示实体关系。后者可以根据实体对的共现情况，从大型文档语料库中轻松提取。图 1 显示了这两种表单的示例。本文分别称它们为标签形式和文本形式，并用混合形式表示由标签和文本组成的关系图。

知识图谱用 G 表示，实体和关系分别用 E 和 R 来表示。 n 表示实体的数量， $n \times n$ 的矩阵 R 中的元素 $v_{i,j}$ 表示 e_i 和 e_j 之间的关系。多跳问题 q 通常从主题实体 e_x 开始，需要跨越关系边到达答案实体 $Y = \{e^{y_1}, \dots, e^{y_{|Y|}}\}$ 。为了推断一个多跳问题的答案，TransferNet 从主题实体开始，跳 T 步。在每一步中，它都关注问题的不同部分，以确定最恰当的关系。TransferNet 为每个实体维护一个分数，以表示它们的激活概率，主题实体初始化为 1，其他实体初始化为 0。在每个步骤中，TransferNet 为每个关系计算一个分数，以表示它们在当前查询中的激活概率，然后跨这些激活的关系传递实体分数。下图显示了该框架。

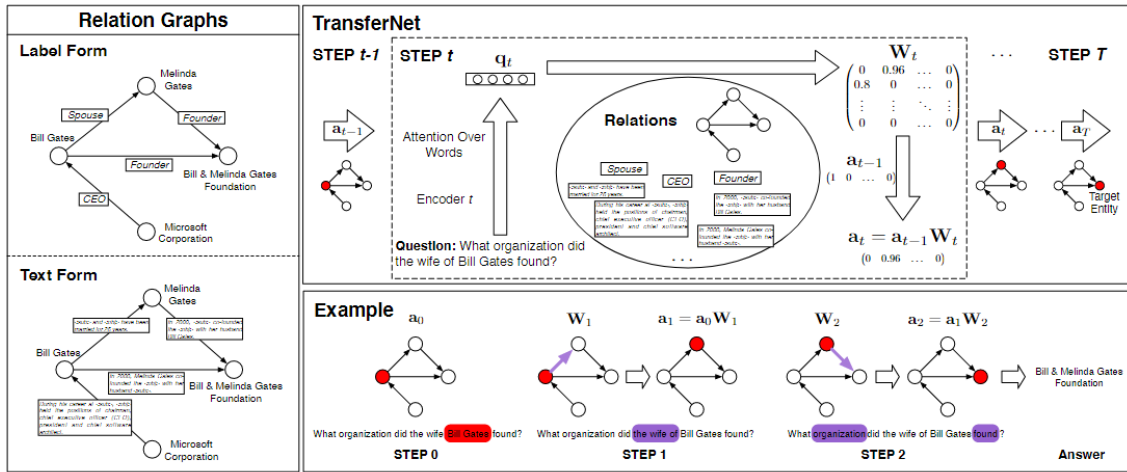


图 1: 方法示意图

3.2 问题编码模块

形式上，我们将步骤 t 的实体分数表示为在 $a^t \in [0, 1]^n$ 处的行向量，其中 $[0, 1]$ 表示 0 和 1 之间的实数。 a^0 为初始分数，即，只有主题实体 e_x 得分到 1。在第 t 步，我们关注问题的一部分，得到查询向量 $q^t \in R^d$ ，其中 d 是隐藏维数。

$$q, (h_1, \dots, h_{|q|}) = \text{Encoder}(q\theta_e) \quad (1)$$

$$qk^t = f^t(q; \theta_{ft}) \quad (2)$$

$$b^t = Softmax(qk^t * [h_1, \dots, h_{|q|}]^T) \quad (3)$$

$$q^t = \sum_{i=1}^{|q|} b_i^t * h_i \quad (4)$$

q 表示问题嵌入。 f^t 是步骤 t 的投影函数，它将 q 映射到特定的查询键 q^t 。 q^t 是根据每个单词的隐藏向量计算得分的 attention。 q^t 是 h_i 的加权和。

3.3 实体转移模块

对于 q^t , TransferNet 计算关系分数 $W^t \in [0, 1]^{n \times n}$: $W^t = g(q^t : \theta_g)$

所以实体在 t 时刻的得分为: $a_j^t = a_j^{t-1} * W_{i,j}^t$ 损失函数为 MSE。

3.4 跳数估计模块

在进行 T 次跳跃后，得到每一步 a^1, a^2, \dots, a^T 的实体得分。通过下式计算加权和作为最终输出:

$$c = Softmax(MLP(q)), \quad (5)$$

$$a^* = \sum_{t=1}^T c_t a^t \quad (6)$$

其中 $c \in [0, 1]^T$ 表示问题的跳数的概率分布， c^T 为跳数 T 的概率值。我们可以通过对问题进行过一个 MLP 和 Softmax 层进行估计跳数 a^* 中得分最高的实体将作为答案输出。

4 复现细节

4.1 与已有开源代码对比

1. 在本文的方法中，最后预测的是实体的得分，即为词的分布，MSE 损失函数更多是假定了为词的分布为高斯的先验，所以此时损失函数个人认为应该用 KL 散度进行拟合更为合理，对损失函数做了一个 MSE 和交叉熵损失函数的插值，但实验效果并不好。

2. 原文在关系得分使用的是对所有谓词概率做累加，复现时改成了求 max。参考代码:<https://github.com/sl>

4.2 复现概述

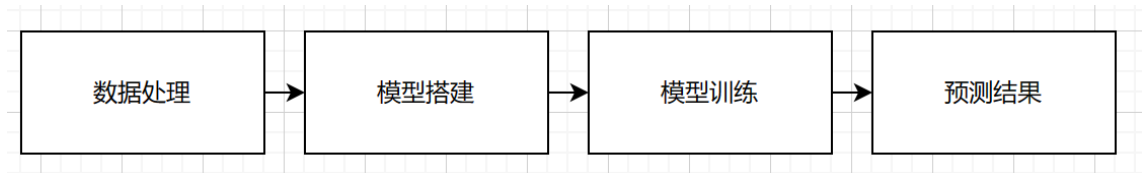


图 2: 总流程

4.2.1 数据处理

首先从知识图谱里将实体 e 和关系 r 做映射，生成 $en2id$ 和 $re2id$ ，为了后续做 embedding 的计算。将训练集的三元组抽出来做成列表，最后将格式 $(ent2id, rel2id, triples, train_data, test_data)$ 存入文件。

4.2.2 模型搭建

与论文中的方法一致，但在实体分数转移时跳数会累积造成梯度爆炸，所以做了截断。

$$Trunc(a) = a/z(a), \quad (7)$$

$$z(a) = a.detach(), if a > 1, a \leq 1 \quad (8)$$

4.2.3 关系得分计算

在知识图谱中，关系用固定的谓词集 P 表示。我们首先用 q^t 计算这些谓词的概率，然后 $r_{i,j}$ 的相应概率为 $W_{i,j}^t$ 。首先预测谓词 p 的分布

$$p^t = Softmax(MLP(q^t)). \quad (9)$$

设 b 表示一对实体之间关系的最大数目，则可表示为 $r_{i,j} = \{r_{i,j,1}, \dots, r_{i,j,b}\}$ ，其中 $r_{i,j,k} \in \{1, 2, \dots, |P|\}$ 。谓词概率是根据关系标签收集的：

$$W_{i,j}^t = \sum_{k=1}^b p_{r_{i,j,k}}^t \quad (10)$$

4.2.4 训练和预测

超参数设置为：学习率 $1e-3$ ，轮数 30 轮，`batch_size`，优化器为 `adam`，编码问题的模型为 `bert`。

5 实验结果分析

5.1 实验环境搭建

平台为 `pytorch`，显卡为一张 2090。

5.2 实验设置

实验数据集：WebQSP[Yih^[9]] 的问题规模较小，但知识图谱规模较大。它包含数千个基于 Freebase 的自然语言问题，Freebase 有数百万个实体和三元组。它的问题不是 1 跳就是 2 跳。接下来 (Saxena 等人，2020 年)，我们对知识库进行了修剪，使其只包含提到的谓词和提到的实体的 2 跳三元组内的谓词。结果，处理后的知识图包括 180 万个实体、572 个谓词和 570 万个三元组。由于 WebQSP 的规模太大，我们只考虑它的标签形式。

CompWebQ[Talmor and Berant^[10]] 是 WebQSP 的扩展版本，问题具有更多跳数的和约束。在 (Sun 等人，2019 之后)，我们使用 PageRank 算法为每个问题检索了一个子图。平均而言，每个子图中有 1948 个实体，召回率为 64%。

| Model | WebQSP | CompWebQ |
|---------------------------------|-------------|-------------|
| KVMemNN (Miller et al., 2016) | 46.7 | 21.1 |
| VRN (Zhang et al., 2017) | - | - |
| GraftNet (Sun et al., 2018) | 66.4 | 32.8 |
| PullNet (Sun et al., 2019) | 68.1 | 47.2 |
| SRN (Qiu et al., 2020) | - | - |
| ReifKB (Cohen et al., 2020) | 52.7 | - |
| EmbedKGQA (Saxena et al., 2020) | 66.6 | - |
| TransferNet (Ours) | 71.4 | 48.6 |

图 3: 实验结果示意

复现结果: WebQSP: 2022-11-29 03:34:55,187 INFO acc: 0.6249188838416613 CompWebQ: 2022-12-02 23:55:32,954 INFO acc: 0.4541, 结果跟原文跟原文相比, 在 CompWebQ 数据集上差别没有特别大, 但在 WebQSP 上差别较大。初始化估计影响较大。

5.3 分析

通过修改损失函数后, 准确率反而掉点, 个人感觉是经验分布仍然不会很接近实际分布, 数据集增大后估计效果会好。

6 总结与展望

改进点: 文章在可解释性上做的非常好, 但在推理过程上仍有进步空间, 例如子图召回本文只是简单的把主题实体三跳范围的子图召回, 如果能利用一些更加精准的召回结果会更好, 且在推理上相对简单, 只是简单的马尔可夫链上的跳跃, 如果加上用 GNN 做信息汇聚或许效果会更好, 在跳数估计上也只是使用了简单的 MLP 进行估计, 如果加入一些先验或许效果会更好, 以后可在这几点上做拓展。

展望: 在 KBQA 领域又出了很多新的高质量的数据集, 如 Kropo, grail 等, 当这些模型在新数据集上是否还能表现出高准确率还是比较期待的。且目前在 chatGPT 出现后也会对该领域有着很大的促进作用。

参考文献

- [1] BERANT J, LIANG P. Semantic parsing via paraphrasing[J]. ACL, 2014.
- [2] REDDY S, LAPATA M, STEEDMAN M. Large-scale semantic parsing without question answer pairs [J]. TACL, 2014.
- [3] YIH W T, CHANG M W, dong HE X, et al. Semantic parsing via staged query graph generation: Question answering with knowledge base[J]. ACL, 2015.
- [4] LIANG C, BERANT J, LE Q, et al. Neural symbolic machines: Learning semantic parsers on Freebase with weak supervision[J]. ACL, 2017.
- [5] BORDES A, USUNIER N, CHOPRA S, et al. Large-scale simple question answering with memory networks[J]. arXiv, 2015.
- [6] DONG L, WEI F, ZHOU M, et al. Question answering over Freebase with multicolumn convolutional neural networks[J]. ACL, 2015.
- [7] JAIN S. Question answering over knowledge base using factual memory networks[J]. NAACL, 2016.
- [8] CHEN Z Y, CHANG C H, CHEN Y, et al. UHop: An unrestricted-hop relation extraction framework for knowledge-based question answering[J]. NNACL, 2019.
- [9] YIH W T, RICHARDSON M, MEEK C, et al. The value of semantic parse labeling for knowledge base question answering[J]. ACL, 2016.
- [10] TALMOR A, BERANT J. The web as a knowledge-base for answering complex questions[J]. NAACL-HLT, 2018.