

把数据集刷穿是什么体验？MetaQA已100%准确率

原创 炼丹学徒 夕小瑶的卖萌屋 2021-05-18 22:20

用力过猛，一不小心 刷穿数据集了呢



文 | 炼丹学徒

编 | 小轶

开始炼丹以来，估计很多小伙伴都和我一样幻想过直接把数据集做到 100% 准确率，然后大吼一声：
这数据集，我做到头了！

然而愿望终究是愿望。大多时候，看着自己手头上用了浑身解数才提了零点几个点的数据集，只想把它换掉。体谅到我们没吃过猪肉——最近清华大学李涓子组发表了一篇多跳问答工作TransferNet，怕是可以带大家看看猪跑了。

把数据集刷穿究竟是一种什么体验？出于好奇心，炼丹学徒带大家一起看一看。

论文题目：

TransferNet: An Effective and Transparent Framework for Multi-hop Question Answering over Relation Graph

论文链接：

<https://arxiv.org/pdf/2104.07302.pdf>

Arxiv访问慢的小伙伴也可以在 【夕小瑶的卖萌屋】 订阅号后台回复关键词 【0518】 下载论文PDF~

多跳问答

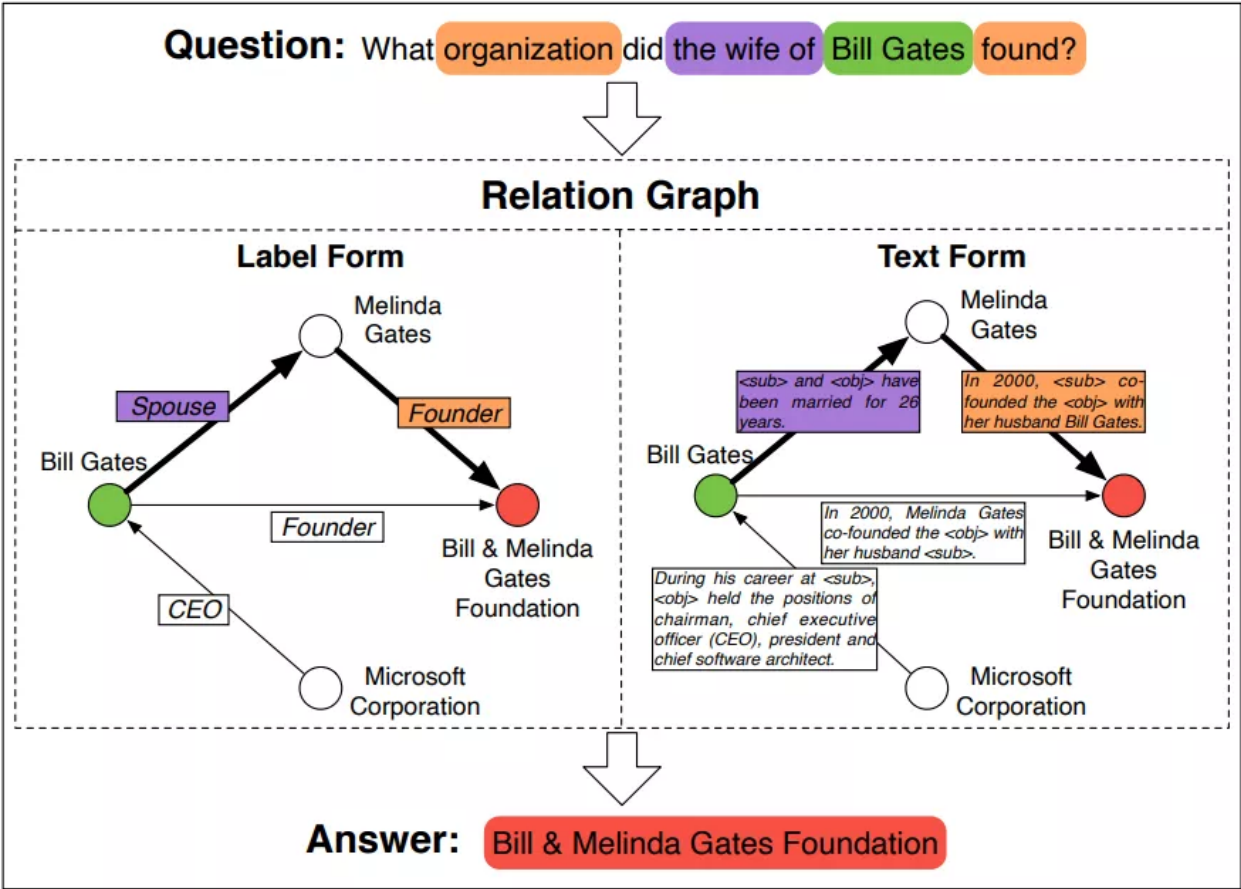
任务定义

简单来说多跳问答是指：针对一个文本问题，在给定关系图上通过多个关系边找到答案的过程。

详细定义如下：

记关系图为 \mathcal{G} ，图中实体为 ξ ，关系边为 \mathcal{R} ；一共有 n 个实体。 \mathcal{R} 可以表示为 $n \times n$ 的矩阵。这个矩阵中的 $r_{i,j}$ 表示头实体 e_i 到尾实体 e_j 之间的关系，是标签格式关系图里定义的关系标签，或者文本格式关系图里检索出来的句子。多跳问答则是：给定问句 q ，从 q 中的话题实体作为起点，经过某些边，达到目标答案实体 $Y = \{e_{y^1}, \dots, e_{y^{|Y|}}\}$ 的过程。

举例说明



这个例子里，提问“ 比尔盖茨的夫人创建了哪个组织 ”（醒醒，比尔盖茨和夫人已经离婚了）。我们首先定位到话题实体“ 比尔盖茨 ”，然后通过“ 的夫人 ”找到实体“ 梅琳达 ”，再通过“ 创建 ”关系找到答案“ 比尔及梅琳达·盖茨基金会 ”。

我们看到：关系图里左边的例子是**标签格式**的关系图（也称作知识图谱，是人工定义的实体和关系图），右边的例子是**文本格式**的关系图，可以从文本库中通过实体名字的共现检索出来。标签格式的更准确，标注代价更大，文本格式的则相反。

主流做法

常见的多跳问答有两个主流做法。

第一种主流做法，是从起始的实体开始预测 **关系-实体序列**，最终得出答案。好处是：从起点实体到目标回答实体的关系路径是显式预测出来的，**推理过程可解释**。坏处是：我们的监督语料只有答案的目标

实体，而从起点实体到目标实体的路径是没有的，因此是弱监督问题，搜索空间太大，存在收敛问题，因此虽然预测结果可解释，但是可能效果不如第二种主流做法。其次，第一种主流做法往往针对标签格式的关系图提出，如果给定的知识是检索出来的文本，很多技术无法使用。

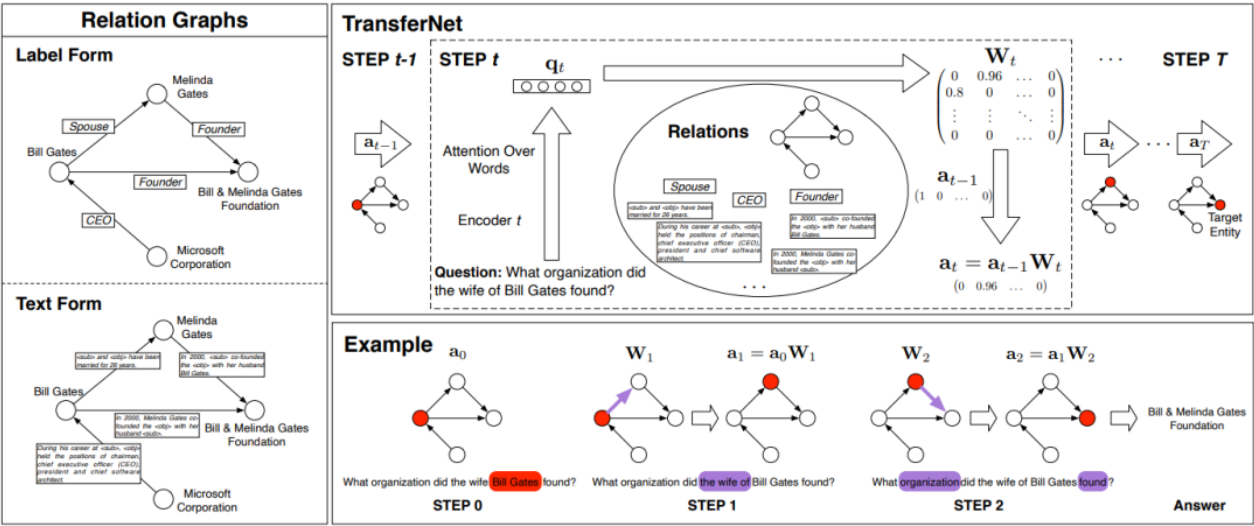
第二种主流做法是把起始实体和附近的关系图喂给模型，直接使用图神经网络（GNN, Graph Neural Networks）预测目标实体作为回答。好处是：监督学习的场景，直接让模型去从附近的关系网络预测目标答案，效果比较好。目前SOTA的方法就是这么做的。缺点是：如何从起始实体预测到目标实体的推理过程是黑箱，模型可能在小区域的图关系里进行了复杂的运算。

TransferNet

看完主流做法，我们直觉地想到：

- 能否提出一个多跳问答的解法，既可以像第一种做法中预测出来关系路径一样可解释，又像图神经网络一样得到不错的结果。
- 如果可以同时解决标签格式，又可以解决文本格式的关系图问答，就更好了！

TransferNet就针对这两个痛点设计了模型：



思路

对于 n 个实体 ξ ， $n \times n$ 的关系 \mathcal{R} 的图 \mathcal{G} ，问题 q ，首先把起始话题实体的分数置为1，其他实体分数置为0，将此分数记为 a_0 。在每一跳后都预测当前关系得分，根据关系的预测概率，更新当前第 t 跳结束后的实体得分 a_t ，直到最大跳数。预测该问题的跳数，按跳数的概率加权每一跳 a_t 得分作为实体的最终得分。

- 首先把起始话题实体的分数置为1，其他实体分数置为0，将此分数记为 a_0
- 每一跳后都给当前关系打分。根据关系得分，更新当前第 t 跳结束后的实体得分 a_t ，直到最大跳数。
- 预测该问题的跳数，并按跳数的概率加权每一跳 a_t 得分作为目标实体的最终得分。

流程

下述所有 θ 都是可训的参数

0) **初始化**: 每个实体的分数 $a^t \in [0, 1]^n$, 初始 $t=0$, 只有问题 q 中的主体实体分数为1, 其他实体都为0。并将问题先过一遍编码器获得一个**初始问题编码** q 和单词隐状态序列 $h_1, \dots, h_{|q|}$

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

1) **问题编码**: 用 q 和 $h_1, \dots, h_{|q|}$ 获得每一跳时的 q^t , 真正后面用的时候用的每一跳时候的问题编码是这个 q^t 。

$$\begin{aligned} qk^t &= f^t(q; \theta_{f^t}) \\ b^t &= \text{Softmax}(qk^t \cdot [h_1; \dots; h_{|q|}]^T), \\ q^t &= \sum_{i=1}^{|q|} b_i^t h_i \end{aligned}$$

可以简单的理解为, 第 t 跳时, 用对应的特定函数 f^t 去计算对 h 的注意力, 然后把 h 加权过来代表此跳的问题编码。

3) **关系得分**: 使用当前跳问题的编码 q^t 计算关系得分 $W^t \in [0, 1]^{n \times n}$ 。对于标签关系图和文本关系图, 计算的方法是不一样的。这一步中, 其实就是用 q^t 去计算每个关系 \mathcal{R} 的分数, 然后汇总到 $n \times n$ 所有实体的关系矩阵中:

$$W^t = g(q^t; \theta_g)$$

4) **实体得分**: 通过3) 中计算得到的实体-实体的关系得分 W^t , 上一条的每个实体分数 a^{t-1} , 计算此跳结束后每个实体的分数 $a^t \in [0, 1]^n$ 。重复上述步骤, 直到超参最大跳数 T 。

$$a^t = a^{t-1} W^t$$

5) **总路径得分**: 在4) 中, 我们得到了每一跳后的实体分数 a^1, a^2, \dots, a^T 。然后我们预测一下这个问题 q 是一个几跳的问题。此处跳数的预测也是概率加权使用的。

$$\begin{aligned} c &= \text{Softmax}(\text{MLP}(q)), \\ a^* &= \sum_{t=1}^T c_t a^t \end{aligned}$$

此处, $c \in [0, 1]^T$ 即为问题跳数的概率分布。最终, a^* 里分数最高的实体作为答案进行输出。

细节

- 在流程4) 中, 由 a^{t-1} 和 W^t 计算 a^t 时, 分数可能会超过 1, 这会导致**梯度爆炸**和 **错误的优化方向**。为了保证 a^t 中所有实体的分数都可以保持在 $[0, 1]$, TransferNet 采用**截断函数 Trunc** 施加在每次 a^t 的计算之后。

$$\begin{aligned} \text{Trunc}(a) &= a / z(a), \\ z(a) &= a. \text{detach}() \quad \text{if } a > 1 \quad \text{else } 1 \end{aligned}$$

- 在流程3) 中，需要由问题编码 q^t 计算关系分数矩阵 W^t 。对于标签格式的关系图，因为关系 \mathcal{P} 是固定的，所以可以直接计算每种关系的分数 $p^t = \text{Softmax}(\text{MLP}(q^t))$ ，如果关系之间不是互斥的，则可以把 Softmax 替换为 Sigmoid 。则实体关系分数矩阵中实体 e_i 和 e_j 之间的关系分数由连接他俩之间的所有关系分数求和或者求极大值得到。对于文本格式的关系图，假设实体 e_i 和 e_j 有 b 个句子 r ，则关系分数矩阵 W^t 的计算与标签格式的类似，为：

$$\begin{aligned} r_{i,j,k} &= \text{Encoder}(r_{i,j,k}; \theta_r), \\ p_{r_{i,j,k}}^t &= \text{Sigmoid}(\text{MLP}(r_{i,j,k}) \odot q^t), \\ W_{i,j}^t &= \sum_{k=1}^b p_{r_{i,j,k}}^t \end{aligned}$$

此处， \odot 为元素对应位置的点乘。

- 文本类型的关系图中，可能文本的描述在处理成关系后存在偏差。比如，Harry Potter, <sub> was published in <obj>, XXX文本关系，如果XXX是United Kingdom，那么<sub> was published in <obj>表达的意思是在何处发表；如果XXX是1997，那么<sub> was published in <obj>描述的是在何时发表的关系。为了区分这种相同文本，不同含义的情况，TransferNet提出额外的语言遮盖预测。所提出的遮盖预测分数如下：

$$m = \text{Sigmoid}(\text{MLP}(q)) \quad m \in [0, 1]^n$$

是所提出的遮盖预测分数。在流程5) 中，把文本类型的预测结果修改为 $\hat{a}^* = m \odot a^*$ 。 \odot 为元素对应位置的点乘。

实验

数据集

- MetaQA**:将wiki-movies的单跳问答拓展到多跳问答的数据集，包含400k问题，最多三跳，包含43k实体，9种关系，135k三元组数据图。TransferNet也使用了基于WikiMoviews的文本关系图问答数据集。
- WebQSP**:基于Freebase，包含一跳和两跳问答，使用标签格式的关系图，1.8m实体，572关系类型，5.7万条关系三元组。
- CompWebQ**: WebQSP的延伸版本。

标签类型关系图结果

Model	MetaQA			WebQSP	CompWebQ
	1-hop	2-hop	3-hop		
KVMemNN (Miller et al., 2016)	95.8	25.1	10.1	46.7	21.1
VRN (Zhang et al., 2017)	97.5	89.9	62.5	-	-
GraftNet (Sun et al., 2018)	97.0	94.8	77.7	66.4	32.8
PullNet (Sun et al., 2019)	97.0	99.9	91.4	68.1	47.2
SRN (Qiu et al., 2020)	97.0	95.1	75.2	-	-
ReifKB (Cohen et al., 2020)	96.2	81.1	72.3	52.7	-
EmbedKGQA (Saxena et al., 2020)	97.5	98.8	94.8	66.6	-

TransferNet (Ours)	97.5	100	100	71.4	48.6
--------------------	------	-----	-----	------	------

Table 2: Hits@1 results of the label-formed datasets. TransferNet achieves 100% accuracy in the 2-hop and 3-hop questions of MetaQA. On WebQSP and CompWebQ it also outperforms baseline models by a large margin.

我们期待的刷穿数据集就出现在上表的TransferNet结果MetaQA多跳问答。

文本类型关系图结果

上表中，左侧是纯文本格式，右侧是在纯文本数据基础上，随机选取50%的标签类型数据添加进关系图。所提升的效果，是因为文本数据集中，存在关系的部分缺失。

可解释性

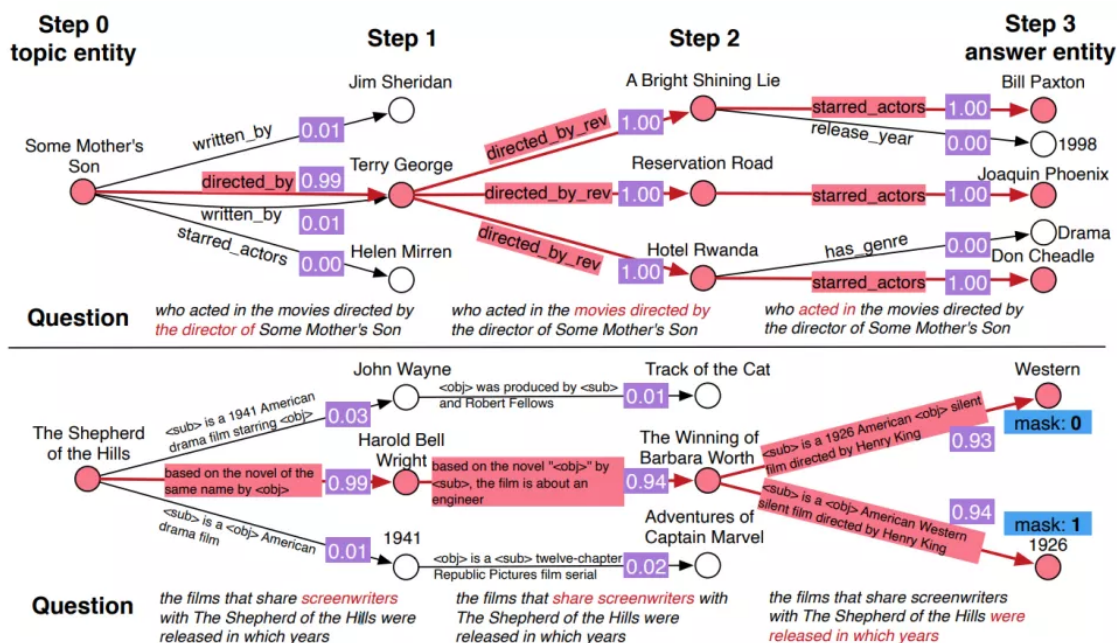


Figure 3: Reasoning process of 3-hop questions. The top is in label form, where the suffix “_rev” means reverse relation. The bottom is in text form, where “mask” in blue means the language mask. We show the relation scores in purple and highlight the activated entities and relations (score > 0.8) and words (score > 0.05) in red.

因为在每一跳的更新里，对目标实体进行0~1的评分，因此分数越高，我们越确定这一跳的实体确信心度。其次，每一跳的更新也是通过关系矩阵进行的，其关系路径是可解释的。

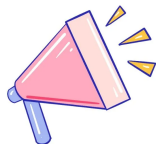
总结

本篇论文我们一起看了多跳问答的新工作TransferNet。论文思路清楚，效果提升明显。此外，我们跟TransferNet一起体验了一把数据集刷穿的感觉~

希望小伙伴们可以向TransferNet学习，一起刷穿数据集。

寻求报道、约稿、文案投放：

添加微信xixiaoyao-1，备注“商务合作”



后台回复关键词【**入群**】

加入卖萌屋NLP/IR/Rec与求职讨论群

后台回复关键词【**顶会**】

获取ACL、CIKM等各大顶会论文集！



喜欢此内容的人还喜欢

我不看好data2vec这类多模态融合的研究

夕小瑶的卖萌屋