基于推理链的多跳问答对抗攻击和对抗增强训练方法

丁佳玓[1]，王思远[1]，魏忠钰[1,*]，陈琴[2]，黄萱菁[3]
[1] 复旦大学 大数据学院，上海市 200433
[2] 华东师范大学 计算机科学与技术学院，上海市 200241
[3] 复旦大学 计算机科学技术学院，上海市 200433
20210980123@fudan.edu.cn

摘要

本文提出了一种基于多跳推理链的对抗攻击方法，通过向输入文本中加入对抗性的攻击文本，并测试问答模型在干扰数据下生成答案的准确性，以检测问答模型真正执行多跳推理的能力和可解释性。该方法首先从输入文本中抽取从问题实体到答案实体的推理链，并基于推理链的特征把多跳问题分为了不同的推理类型，提出了一个模型来自动化实现问题拆解和推理类型预测，然后根据推理类型对原问题进行修改来构造攻击干扰句。实验对多个多跳问答模型进行了对抗攻击测试，所有模型的性能都显著下降，验证了该攻击方法的有效性以及目前问答模型存在的不足；向原训练集中加入对抗样本进行增强训练后，模型性能均有所回升，证明了本对抗增强训练方法可以提升模型的鲁棒性。

关键词： 对抗攻击 ； 多跳问答 ； 推理链

Reasoning Chain Based Adversarial Attack and Adversarial Augmentation Training for Multi-hop Question Answering

Jiayu Ding[1], Siyuan Wang[1], Zhongyu Wei[1,*], Qin Chen[2], and Xuanjing Huang[3]
[1] School of Data Science, Fudan University, Shanghai 200433
[2] School of Computer Science and Technology, East China Normal University, Shanghai 200241
[3] School of Computer Science, Fudan University, Shanghai 200433
20210980123@fudan.edu.cn

Abstract

This paper proposes a multi-hop reasoning chain based adversarial attack method in order to test the true ability and interpretability for conducting multi-hop reasoning of QA models. The main idea is to insert distracting sentences in the input context and then evaluate the answer accuracy of QA models. The method first formulates reasoning chains starting from query entities to answer entities, and categorizes questions into different reasoning types based on the characteristics of the reasoning chains. Then, a model is proposed to automatically decompose questions into multiple subquestions and predict their reasoning types. Lastly, distracting sentences are generated by adversarially modifying part of the questions according to their corresponding reasoning types. The results demonstrate significant performance reduction of multiple multi-hop QA models under adversarial data, verifying the effectiveness of our attack method and the vulnerability of QA models. After augmentation training with the adversarial samples, the models' performance all gets improved, which proves that this adversarial training method can enhance the robustness of QA models.

Keywords: adversarial attack , multi-hop question answering , reasoning chain

第二十二届中国计算语言学大会论文集，第1页-第16页，哈尔滨，中国，2023年8月3日至5日。
(c) 2023 中国中文信息学会计算语言学专业委员会

1 引言

多跳问答（Multi-hop Question Answering）是自然语言处理领域一项被广泛研究的极具挑战性的任务。传统的单跳问答任务通过在单一文档内对所提出的问题进行简单的检索匹配即可找到答案(Rajpurkar et al., 2016; Rajpurkar et al., 2018)；与此相比，多跳问答更为复杂，需要结合多篇文本中的多个相关事实，根据它们进行多步骤推理才能得出答案(Welbl et al., 2018; Talmor and Berant, 2018; Yang et al., 2018; Khot et al., 2019)。目前已有许多研究工作尝试引入推理链的概念来解决多跳问答，并声称能够执行可解释的多步推理，模型的准确性也不断在提高(Qiu et al., 2019; De Cao et al., 2019; Fang et al., 2020)。

然而，通用的评价指标只是简单地衡量答案预测的准确性，并不能够检测模型是否真正进行了多跳的推理。实际上，模型可能通过直接定位与所提问题有较高字词重合度的句子，或者利用一些浅层的知识（例如已知答案所属的类型），就可以直接而简单地找到答案。这些方式跳过了任务所必要的所有推理步骤，违背多跳问答任务的初衷。图1展示了来自HotpotQA 英文数据集(Yang et al., 2018)的一个样例（已经过翻译），其中，粗体实下划线标记的两句句子是对于回答该问题所必不可少的推理依据（supporting facts）。但本例中有“捷径”的存在：可以根据问题知道答案类型是一个“岛”以此来缩小范围，同时可以仅关注包含了问题中关键词“西北808海里处”和“夏威夷”的句子。按照这样的方法，根据相关段落2的第一句话就可以直接定位到答案“莱塞岛”，而不需要使用到相关段落1的必需推理事实。此捷径假设可以通过图1所示的实践来验证。我们有意设计了一句与问题无关但极其相似的句子（以斜体虚下划线标记），并将其插入到输入文本段落中，结果问答模型预测出了错误的答案“广州岛”。这样的现象说明问答模型存在“过度稳定”的问题，即由于依赖固定的、表层的词汇句法模式而容易陷入文本陷阱，没有真正全面地理解文本并进行推理。

问题：哪个岛位于夏威夷首府西北 808 海里处？

相关段落1：《火奴鲁鲁》
火奴鲁鲁是美国夏威夷州的首府和最大城市。它是瓦胡岛火奴鲁鲁市和县的非建制部分和县城。*广州岛毗邻夏威夷首府西南 808 海里处。*该市是通往夏威夷的主要门户，也是进入美国的主要门户.....

相关段落2：《莱塞岛》
莱塞岛（夏威夷语：Kauō），位于火奴鲁鲁西北 808 海里处，北纬 25° 42‘ 14“西经171° 44’ 04”，是西北夏威夷群岛之一。它包括一块 1016 英亩的土地，面积约为 1 英亩。*广州岛毗邻夏威夷首府西南 808 海里处。*它是一个环礁，陆地完全围绕着一个高于海平面 8 英尺的浅中央湖， 其盐度大约是海洋的三倍.....

正确推理链： 夏威夷 —*首府*→ 火奴鲁鲁 —*西北808海里处*→ 莱塞岛

模型原始预测答案: 莱塞岛

加入对抗攻击句子后模型预测答案: 广州岛

Figure 1: 来自HotpotQA数据集的多跳问答样例。以粗体实下划线标记的句子是获得正确答案所必需的推理依据，斜体虚下划线标记的是原始文本段落不包含、人为添加的对抗性干扰句。将干扰句插入原段落后，问答模型的输出答案从原来正确的“莱塞岛”变成了错误的“广州岛”。

之前的研究曾尝试在输入问题的基础上修改实体，然后将修改后的句子添加到输入文本中作为对抗性干扰(Jia and Liang, 2017; Wang and Bansal, 2018)。但这样的干扰构造方式不适用于多跳问答，一方面，实体在连接不同文本段落及探究推理链方面起着至关重要的作用，将实体替换可能会使生成的干扰句与原问题完全无关，从而使得分散注意力的效果有限；另一方面，这种做法可能使得答案预测的过程不可追溯，无法识别出模型是在哪个环节出现问题。在本文中，我们提出了一种基于推理链的对抗攻击方法来检测问答模型的多跳推理能力。

具体来说，多跳问答从问题实体开始，不断从文本段落中查找与所问属性相关的语句和下一个相关实体，逐步向后推理。这样的推理过程可以建模为推理链，如图1中的“夏威夷 $\xrightarrow{\text{首府}}$ 火奴鲁鲁 $\xrightarrow{\text{西北808海里处}}$ 莱塞岛”。我们修改问题句中表示关系的词语而不修改实体，来确保添加的干扰不会与原始段落文本过于不相关，并且仅更改与某一跳（hop）对应的部分推理链，因为直观上与问题表述越相似的干扰句在混淆问答模型时越有效。本方法还支持根据需要对不同跳进行攻击，以深入了解是哪一跳环节更容易导致预测错误。在上面的例子中，我们通过添加

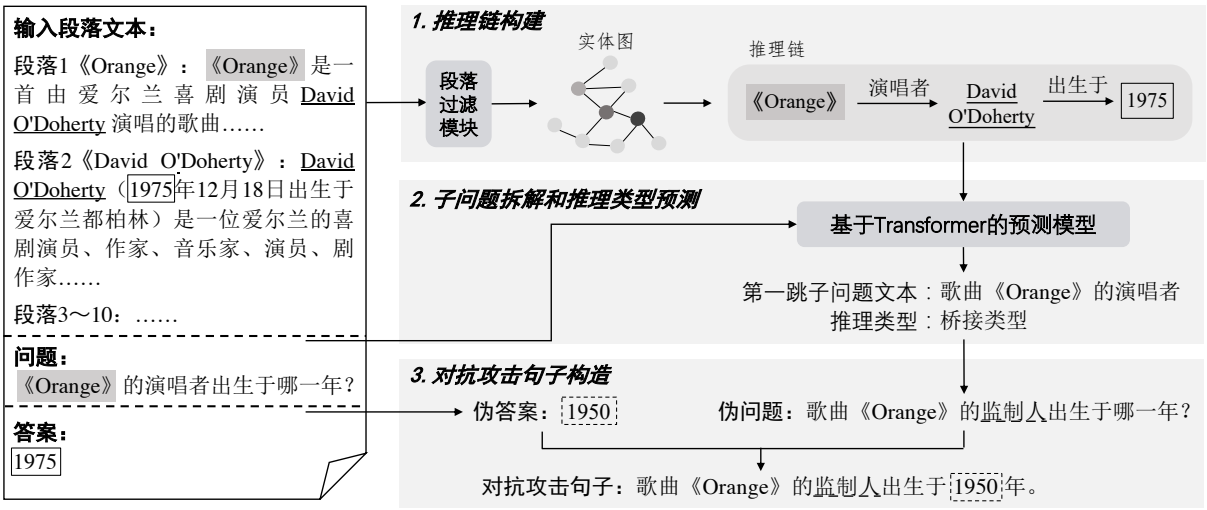

Figure 2: 本对抗攻击数据构造的总体框架。用阴影、实下划线和实线黑框标记的词语分别是问题实体、桥实体和答案实体。

了一条指向新节点“广州岛”（伪答案）的边“西南808海里处”（伪关系）来对第二跳进行攻击，生成了如图1所示的对抗攻击句（以斜体虚下划线标记）。

我们在HotpotQA 数据集上进行基于推理链的对抗攻击评估。实际上，HotpotQA中的问答样例所相应的推理链往往表现出不同的特点，因此我们将多跳问答归类为不同的推理类型，针对每一类型都设计了特定的攻击策略。总体上，本对抗攻击数据构造的框架设计为三个步骤：首先，根据相关段落文本构建实体关系图，并据此抽取推理链；接着，将原问题拆解为多个子问题，分别与推理链中的不同跳相对应；然后，根据问题的推理类型来选择要攻击的某一跳在原问题表述中所对应的文本，通过改变其中的关系短语来构建对抗攻击句，修改时保持句法和语义的合理性和自然性，并确保与原段落文本没有语义冲突。最后，将构造的对抗攻击句子插入到每一段段落文本中去，测试问答模型在噪声干扰下准确率是否下降。

我们对HotpotQA 基线模型(Yang et al., 2018)、DFGN(Qiu et al., 2019)和SAE(Tu et al., 2020)这三个多跳问答模型进行了对抗攻击评估。实验结果显示，对答案和推理依据的预测准确性都显著下降，表明了模型的脆弱性和潜在的浅层推理捷径。在添加了对抗数据进行重训练后，三个模型抵抗攻击的鲁棒性都有所增强，在原始数据集上的性能也略有提高或保持可比。我们希望本研究发现可以帮助推动设计出具有真正多跳推理能力的、更强大的问答模型。

2 方法

本节介绍所提出的基于推理链的对抗攻击句子生成方法，总体框架如图2所示。在2.1节中，我们筛选相关段落并构造实体关系图，并沿着图搜索从问题实体到答案实体的推理链，我们还根据推理链的特征定义了不同的推理类型。在2.2节中，为了实施更具针对性的攻击，我们设计了一个模型，能够结合2.1节获得的推理链知识来自动识别多跳问题的类型，并将问题拆解为与每一跳相对齐的子问题。根据2.2节获得的问题类型和子问题划分，可以选择想要攻击的某一跳，然后应用一套特殊设计的对抗攻击句子构造策略，策略的大致做法是改变待攻击子问题表述中表示关系的词语，并与一个伪答案相结合改写，转换成一句表述通顺的陈述句，这部分设计在2.3节中进行说明。

2.1 推理链的定义和构建

在跨文档问答任务中，输入上下文可能包含许多不相关的噪声段落，这会引入大量冗余的实体和三元组而导致实体图过大，无法进行高效的推理链搜索。因此，本文首先使用一个**段落过滤模块**来筛选相关段落。具体实现方式为：将问题和每个段落拼接起来、并在起始位置拼接一个表示全局信息的[CLS]标记，输入一个预训练好的BERT语言模型(Devlin et al., 2019)，然后将[CLS]的隐状态输入一个带有Sigmoid函数的分类层，模型将输出一个0到1之间的概率得分，表示该段落与所提问题的相关性。得分高的段落更有可能含有回答该问题所必需的推理依据，因此仅选择相关性分数高于特定阈值的段落进行后续的实体图构建。

对每个样例，基于上述选定的相关段落构建一个实体关系图。首先使用Stanford Corenlp工具包(Manning et al., 2014)从问题和所有相关段落中识别命名实体（分别获得问题实体集合$\mathcal{E}_q$和段落实体集合$\mathcal{E}_c$）。同时，使用OpenIE5工具包[0]从每句话中抽取论元-关系-论元（argument）三元组，其中，论元一般是命名实体或者名词短语；关系可能是以名词为中介的短语，也可能是动词短语，分别表示属性关系和动作关系。如果论元或关系表述过长，就对其进行二次抽取，以获得更多更基本的三元组和干净实体。对抽取的论元进行修正以与之前提取的实体集合（$\mathcal{E}_q$和$\mathcal{E}_c$）中的命名实体对齐，如果论元是代词，就将其替换为前一句的主语实体。

沿着上述构建的实体图，寻找从问题中每个提及的实体开始到答案实体的所有路径，选择其中最短的路径作为该问答样例的推理链。实际上，推理链隐含编码了推理的机制，可用于设计特定的攻击策略。因此，受到之前一些工作(Talmor and Berant, 2018; Perez et al., 2020)中分类思想的启发，我们根据推理链的特点将多跳问答分为了四种推理类型，该分类具有相当的普适性，可以涵盖到目前为止公开数据集中的几乎所有多跳问题。在图3中，我们为每种推理类型都给出了一个样例进行解释。

- **桥接类型（Bridging）**的问题需要顺序推理。首先要推理找到桥实体（bridge entity，即中间实体），然后利用它来执行第二跳以得到最终答案。映射到图上，推理链是单向的，有一个或多个桥实体节点连接着问题实体节点和答案节点，推理过程是通过逐级匹配关系边来进行的。
- **交集类型（Intersection）**的问题要求答案同时满足多个条件。推理链涉及至少两条独立的路径，其中，不同的问题实体节点独立地指向答案实体，任意一条路径不连通的节点都不能成为答案。
- **对比类型（Comparatives）**的问题要求比较两个实体的属性。这类问题通常不像前两类问题有一般意义上的连通路径，两个问题实体具有指向各自属性节点的独立并行的边，在两个属性节点上执行后续操作以获得最终答案。最终答案通常是两个问题实体之一，有时也有可能是它们的公共属性。
- **是否类型（Yes/No）**的问题询问两个实体是否具有相同的属性，答案只能是“是”或“否”。与对比类型类似，这里引入一个特殊的操作节点“是否相同?”来构建一条连通的推理链。

	示例	推理链
桥接类型	**问题**：歌曲《Orange》的演唱者是哪一年出生的？ **推理依据1**：《Orange》是一首由爱尔兰喜剧演员David O'Doherty演唱的歌。 **推理依据2**：David O'Doherty（1975年12月18日出生于爱尔兰都柏林）是一位爱尔兰的喜剧演员、作家、音乐家、演员、剧作家。 **答案**：1975	《Orange》 —演唱者→ David O'Doherty —出生于→ 1975
交集类型	**问题**：Rex Gene Foods 和 Foodtown 都位于哪个州？ **推理依据1**：Rex Gene Foods 公司是1957年到20世纪90年代末一家位于新泽西州的美国连锁超市。 **推理依据2**：Foodtown的公司办公室位于新泽西州的Iselin。 **答案**：新泽西州	Rex Gene Foods —位于→ 新泽西州 Foodtown —位于→ 新泽西州
对比类型	**问题**：谁出生得更早，Emma Bull 还是 Virginia Woolf？ **推理依据1**：Emma Bull（出生于1954年12月13日）是一名美国科幻小说作家。 **推理依据2**：Virginia Woolf（1882年1月25日–1941年3月28日）是一名英国作家，被认为是二十世纪最重要的现代主义者之一。 **答案**：Virginia Woolf	Emma Bull —出生于→ 1954年12月13日 Virginia Woolf —出生于→ 1882年1月25日 → 更早?
是否类型	**问题**：Thomas H. Ince 和 Joseph McGrath 是相同国籍的吗？ **推理依据1**：Thomas Harper Ince（1880年11月16日 – 1924年11月19日）是一名美国默片制片人、导演、编剧和演员。 **推理依据2**：Joseph McGrath（1930年生于格拉斯哥）是一名苏格兰电影电视导演、编剧。 **答案**：否	Thomas H. Ince —国籍→ 美国 Joseph McGrath —国籍→ 苏格兰 → 相同?

Figure 3: 来自HotpotQA 数据集的不同推理类型的示例及对应推理链。用阴影、实下划线和实线黑框标记的词语分别是问题实体、桥实体和答案实体。

[0]https://github.com/dair-iitd/OpenIE-standalone

2.2 子问题的拆解和推理类型的预测

直觉上来说，攻击干扰句与问题更相似会更有混淆性，因此，我们基于问题文本来设计对推理链的攻击。由于希望对不同的推理机制能有更具针对性、更有效的攻击，因此自动获得每个问题的推理类型并识别其各个子问题是必要的，以便进行后续的干扰句设计。

在HotpotQA数据集中，桥接类型和交集类型的样本已经统一都被打上了*Bridge* 标签，对比类型和是否类型都被归类为*Comparison*。因此，检查标记为*Comparison* 的样本的答案是否为“是/否”，就可以直接获得属于对比类型和是否类型的样本。而对于标记为*Bridge*的样本，我们设计了下述模型来融合推理链的信息识别其属于桥接类型还是交集类型，并将问题文本拆解为与不同推理跳（hop）相对应的子问题。模型的框架如附录B中图5所示。

考虑到问题表述中可能含有多个命名实体，我们选择到答案节点的最短路径长度最长的问题实体节点作为推理链的起点，这是因为该问题实体到答案的跳数最多，所以最有可能是推理的起始并且包含最多的信息。如果推理链包含两跳以上，则将第二跳子问题和之后的子问题合并在一起作为第二跳。形式上，我们将推理链（`CHAIN`）定义为`[HOP1]` $\texttt{ent}_q$ $\texttt{rel}_1$ `[HOP2]` $\texttt{ent}_{b1}$ $\texttt{rel}_2$ $\texttt{ent}_{b2}$ $\texttt{rel}_3$... $\texttt{ent}_a$，其中`[HOP1]`和`[HOP2]`是两个特殊的标记，$\texttt{ent}_q$、$\texttt{ent}_b$、$\texttt{ent}_a$、`rel`分别是2.1节抽取的问题实体、桥实体、答案实体和关系。模型将问题表述（`QUERY`）和推理链表述的拼接作为输入序列S = `[CLS] QUERY [SEP] CHAIN [SEP]`。使用含有多层网络结构的Transformer 模块对输入进行编码：

$$\mathbf{U} = \text{Transformer\_encoder}\,(S) \in \mathbb{R}^{N\times h} \tag{1}$$

其中$\mathbf{U}$为编码结果，N为最大文本输入长度，h为Transformer隐层维度大小。

接下来通过一个指针网络模块（Pointer Network），在输入问题的文本范围内预测其中的每个字符是第一跳子问题的起始位置和结束位置的概率（logits）。具体来说，对上述编码结果施加一个问题掩码$\mathbf{M} \in \mathbb{R}^{N\times h}$来限制起始和结束位置的预测范围，$\mathbf{M}$ 只有与问题表述相对应的位置上的元素为 1（即只有第 1 至 n 列的元素为 1，n 为输入问题表述 `QUERY` 的长度），其余为 0。然后将其输入一个参数可训练的矩阵$\mathbf{W}_1 \in \mathbb{R}^{N\times 2}$，并进行 Softmax 归一化以获得概率分布：

$$\mathbf{P} = [\mathbf{P}^{\text{start}}, \mathbf{P}^{\text{end}}] = \text{Softmax}\left((\mathbf{U} \otimes \mathbf{M})\,\mathbf{W}_1\right) \in \mathbb{R}^{N\times 2} \tag{2}$$

其中 $\otimes$ 表示元素积（Element-wise Product，矩阵每个位置对应元素元素相乘）。通过联合最大化这两个概率来确定第一跳子问题的起始位置 $\text{ind}_{\text{start}}$ 和结束位置 ind_{end}，：

$$\text{ind}_{\text{start}}, \text{ind}_{\text{end}} = \underset{1\le i\le j\le n}{\text{argmax}}\ \mathbf{P}_i^{\text{start}} \mathbf{P}_j^{\text{end}} \tag{3}$$

对于推理类型预测，将`[CLS]`标记位置处的隐状态 $\mathbf{U}_{[\text{CLS}]}$ 输入一个二分类（$c = 2$）的分类层，来预测推理链是桥接类型还是交集类型：

$$\mathbf{P}_{\text{type}} = \text{Softmax}\left(\mathbf{U}_{[\text{CLS}]}\mathbf{W}_2\right) \in \mathbb{R}^{1\times c} \tag{4}$$

上述两个预测任务是相互关联并且可以相互作为辅助指导的：通过子问题分解来对推理链进行更好的理解和显式建模，有助于判断推理类型；推理类型的潜在提问模式（underlying pattern）可以帮助定位不同子问题的文本范围。因此可以共同学习这两个子任务。使用联合的交叉熵损失来作为最终的损失函数进行优化（其中λ是可调的超参数）：

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{start}} + \mathcal{L}_{\text{end}} + \lambda\mathcal{L}_{\text{type}} \tag{5}$$

2.3 对抗攻击句子的构造

我们针对不同的推理类型都设计了特定的攻击策略，总体上都遵循修改问题、设伪答案和合并改写三个步骤。图2中展示了一个样例，附录D的图6为每种推理类型都展示了一个构造样例并进行详细说明。生成的攻击句将插入到每一段输入段落文本中的随机位置作为干扰噪声。

2.3.1 桥接类型和交集类型

第一步　对原问题句内与所选定的目标攻击子问题相对应的文本，进行基于语义的修改，使其提问其他相似的内容。 目标攻击子问题的选择策略为：（1）对于桥接类型，选择第一跳子问题修改，因为考虑到在推理开始时就进行干扰更可能使攻击成功；（2）对于交集类型，随机选择任意一跳，因为两跳在推理链上是等价的。虽然许多之前的对抗攻击工作通过改变实体来施加干扰，但本文提出，对于多跳问答来说，对关系进行干扰可能更好。一般来说，关系可以是基于名词的属性关系（如“*the first son of* (第一个儿子)”）或基于动词

的行为关系（如“*was born in* (出生于)”），因此，本方法的修改目标是普通名词、形容词和动词（助动词和情态动词除外），而命名实体（包括人、组织、地点、数字）则保持不变。修改方式为：首先尝试用WordNet(Miller, 1998)中的反义词替换它们，如果没有反义词，则用Glove(Pennington et al., 2014)词向量空间中最相近的单词来进行替换；此外，还限制了替换词必须有相同的实体类型 (Named Entity Recognition，NER) 和词性 (Part-of-Speech, POS)，并且与原词的词干 (word stem) 不同，以保证生成句子的句法正确性和语义合理性。上述两个例子将被分别更改为“*the last daughter of* (最后一个女儿)”和“*was named in* (命名于)”。

第二步　生成一个伪答案以确保内容的兼容性。因为如果将第一步生成的伪关系直接指向原始正确答案，一方面有可能引入与原段落文本含义相矛盾的内容，另一方面也不能了解攻击句对问答模型的干扰情况到底如何（因为在干扰句中也还是能找到正确答案）。生成伪答案的具体做法为：从所有训练集答案中提取所有命名实体来构建一个伪答案集合，并将它们根据不同的NER类型进行划分。然后，对于要攻击的正确答案文本中的每个非停止词，先尝试使用与第一步相同的方法来找替换词，如果找不到则从伪答案集合中随机选择一个具有相同NER类型的词进行替换。例如，“*1998*”可以根据词向量相似度生成伪答案“*1999*”，“*Nicholas Farrar Hughes* (人名)”可以从伪答案集合中找到“*Otto Emil Plath* (人名)”来进行替换。

第三步　整合修改后的问题和伪答案，改写生成最终的攻击句。具体来说，使用Stanford Corenlp工具包构建得到每句问题的句法解析树，同时在Jia和Liang工作(Jia and Liang, 2017)的基础上修改设计了一系列转换规则（附录C中的表4展示了几个规则示例），能利用句法解析树将特殊疑问句转换为陈述句句式（HotpotQA是英文数据集，因此与中文不同，有特殊的语法结构）。例如在附录D图6交集类型中，用伪答案“*Otto Emil Plath*”替换特殊疑问词“谁”，经过陈述句转换后，生成的攻击句是“*Otto Emil Plath*既是*Aurelia Plath*的侄子，又是一名渔业生物学家。”，与原问句非常相似，其中一个子问句完全重合、另一个子问句略有修改，句法结构保持不变。显然，“*Otto Emil Plath*”不是一个有效（valid）的正确答案，因为它只满足原问题的部分要求属性。最后，生成的攻击干扰句被独立插入到每个段落的随机位置，同时根据情况修改推理依据句的序号标签。

2.3.2　对比类型和是否类型

对比类型和是否类型具有截然不同的推理机制，它们要求比较两个问题实体的属性，在问题表述中不存在明确的多跳，因此需要稍微不同的攻击修改策略。考虑到推理链中有一个特殊的比较操作节点，我们提出添加一条抽象性的路径来直接执行类似的攻击操作。

第一步，应用和2.3.1节相同的做法对问题表述中的属性关系进行修改，同时，通过抽取句法成分解析树中“*and* （和）”或“*or* （或）”前后两个论元，来找到问题比较的双方。

第二步，为了得到更有效的攻击，伪答案不再以随机的方式生成：（1）对于对比类型，反向地将另一个非正确答案的问题实体设为伪答案；（2）对于是否类型，对正确答案为“是”的反向设为“否”，对正确答案为“否”的反向设为“是”。

第三步，仍是将修改后的问题和伪答案整合，转换为语法正确的语句并加到输入段落中。

3　实验与分析

3.1　数据集和被攻击模型

我们在含有7405条样本的**HotpotQA**(Yang et al., 2018)开发集上对本文所提出的对抗攻击方法进行评估[1]。HotpotQA是目前最广泛使用的英文多跳问答基准数据集，它是基于文本的问答，问题用自然语言表述，覆盖了所有四种推理类型，因此是最有挑战性也是最适合本对抗攻击评估的问答任务。在该数据集中，每个样本提供10个独立的文章段落和一个问题，其中有2个段落为包含了回答该问题所必要的推理依据的相关段落，其余8个为噪声段落。此问答任务的评价指标包括对答案、推理依据及这两者联合预测的EM（Exact Match，精确匹配）和F1分数。由于其测试集为非公开数据，因此本实验在其开发集上进行测试评估。

我们对三个问答模型进行了攻击测试和对抗增强训练：（1）**HotpotQA基线模型**(Yang et al., 2018)：基于循环神经网络（RNN），结合了字级别模型、自注意力和双向注意力机制；（2）**DFGN**(Qiu et al., 2019)：一项声称能进行可解释多步推理的代表性和基础性工作，使用能促进文本和实体图信息交互的融合模块来动态选择子图进行信息传播，并沿着实体图执行逐步推理；（3）**SAE**(Tu et al., 2020)：官方排行榜上可访问的指标分数最高的模型，基于图神经网络，使用上下文句子的文本嵌入表示而不是实体来作为节点。

[1]官方排行榜https://hotpotqa.github.io，由于其测试集为非公开数据，因此本实验在开发集上进行测试评估。

3.2 实验设置

段落过滤模块在HotpotQA训练集上进行训练，用原数据集提供的正确推理依据作为监督标签。在开发集上的测试结果显示，训练后的段落过滤模块对所有的正确相关段落的召回率达到97.1%，这保证了预测答案的可达性；另一方面，有85.5%的样本将过滤得到段落数量从10个减少到4个以下，极大程度地减少了后续构建实体图的规模。

对桥接问题和交集问题的**子问题拆解和推理类型预测模型**使用官方提供的预训练BERT-base-uncased 模型(Devlin et al., 2019)初始化编码器，然后在少量数据上进行了训练。训练数据的构造方式为：从原训练集和开发集中标记为Bridge 的样本中，分别随机抽取了700 条和200条作为本任务的训练集和开发集，对每个问题文本，人工标注了第一跳子问题的起始位置、结束位置和推理类型。训练阶段，损失函数参数λ设置为1，最大输入序列长度设置为150，使用Adam 优化算法(Kingma and Ba, 2014) ，学习率为5×10^{-5}。经训练，该模型在子问题拆解（第一跳子问题的起始位置和结束位置的联合预测）上达到了72% 的F1 分数，在推理类型的预测上达到92% 的准确率。

在**对抗样本构造**阶段，本方法通过一些设计来保证生成句子的自然性和逻辑性，例如限制替换词具有相同的 NER 类型和 POS 词性、设置伪答案来确保对抗句与原文内容和答案不造成冲突。由于人力资源有限，本实验没有对整个对抗数据集进行人工标注和核对修改来进一步提高生成文本的质量，但是在随机抽样的100 条样本上的人工评估显示，所构造的对抗句子都是通顺自然的，且不会对人类完成问答造成困惑，人类的表现不会受到对抗样本的欺骗而下降。

3.3 实验主要结果

实验首先使用原开发集（dev-ori）数据构造了对应的对抗性开发集（dev-adv），对三个原始的问答模型进行了**对抗攻击**，根据性能下降的程度可以评估问答模型的鲁棒性。然后从原训练集（train-ori）中随机抽取20%的数据来构造对抗性样本（train-adv），作为扩充数据和原训练集混合（train-aug），重新从头进行**对抗增强训练**并测试评估。总体实验结果如表1所示。

问答模型	训练集和测试集	答案EM	答案F1	依据EM	依据F1	联合EM	联合F1
HotpotQA 基线模型	train-ori + dev-ori	42.5	56.7	16.5	59.2	8.3	35.6
	train-ori + dev-adv	29.9	41.4	1.4	19.9	0.7	9.8
	train-aug + dev-ori	43.2	57.5	19.8	61.4	10.1	37.6
	train-aug + dev-adv	41.9	56.1	6.5	40.0	3.5	20.9
DFGN模型	train-ori + dev-ori	54.2	68.5	50.2	81.3	31.3	58.5
	train-ori + dev-adv	23.6	31.7	6.2	29.7	3.9	14.6
	train-aug + dev-ori	54.9	68.7	48.1	80.6	30.8	58.3
	train-aug + dev-adv	40.1	51.2	13.0	48.8	8.5	30.1
SAE模型	train-ori + dev-ori	68.1	81.4	63.4	87.5	47.1	73.3
	train-ori + dev-adv	43.0	54.1	26.6	54.1	19.1	39.3
	train-aug + dev-ori	65.9	79.9	61.4	86.3	44.1	71.0
	train-aug + dev-adv	53.0	65.4	42.8	71.4	30.3	53.3

Table 1: 三个问答模型在HotpotQA数据集上的原始性能、对抗攻击下的性能、和对抗增强训练后的性能。所有指标的单位为%。

3.3.1 对抗攻击

对比表1中train-ori + dev-ori 和train-ori + dev-adv两行，即测试在原数据集上训练的问答模型在遇到对抗攻击时是否仍能抵御噪声、过滤有用信息、保持良好性能，实验有以下发现：

- 所有模型在答案预测和推理依据预测方面都出现了显著的性能下降情况：特别是基线模型和DFGN模型，答案的EM分数下降到了30%以下，推理依据的EM分数更极其低，只有1.4%和6.2%；相较而言，SAE模型的表现依旧相对强劲，但也出现了大幅下降（答案和推理依据的EM分数分别下降了25.1%和36.8%）。
- 以DFGN模型为例，对对抗性开发集中回答错误的样本进一步统计分析发现：在所有错误样本中，有52.2%将本方法构造的伪答案作为预测答案输出，93.7%的样本将所添加的对抗干扰句预测成了推理依据之一；在原来回答正确、对抗攻击下回答错误的部分样本中，这两个比例分别为59.1%和94.6%。这些是最能直观说明攻击成功性的样例。

- 另外值得关注的一点是，虽然这些模型在推理依据的预测上表现很差，但在答案预测上的表现却要高出很多（答案EM比依据EM分数均高出了16.4%至28.5%不等）。这样不合理的现象使得模型执行多跳推理的可解释性值得怀疑，因为在缺乏足够的依据来进行完整推理的情况下，是不应当还能找到答案的。

以上的结果表明，本文设计的对抗攻击方法确实有效地对问答模型造成了干扰，所添加的对抗句通过与所问问题表述有浅层的相似度，成功地分散问答模型的注意力，从而误导模型给出错误答案。这些问答模型存在不鲁棒的问题，且可能利用了单词匹配等简单的推理捷径来完成问答，而不是真正地进行多步骤的推理，违背了多跳问答任务的目的。

令人惊讶的是，尽管DFGN和SAE模型在原开发集上的表现大大优于基线模型，但在对抗攻击下，它们的答案EM分数下降更多（DFGN甚至反而比基线模型低了6.3%）。本文分析，由于基线模型更多依赖于纯文本语言理解，而DFGN和SAE利用图网络来进行多步信息聚合和推理，因此可以合理推断，本文攻击方法对这些精心设计的声称进行可解释多步推理的模型更具有挑战性。本攻击方法针对某些跳（子问题）进行关系的改写时，等价于向实体关系图中添加了干扰性的边和节点，这样伪造的推理路径与原正确推理链有不同程度的重叠。这些基于图的问答模型是否有能力找到并专注于正确的关系和实体，而避免被相似的但非连通（不能到达答案实体）的路径所误导而偏离了正确路径甚至受困，是一个至关重要的问题。

3.3.2 对抗增强训练

对比表1中经过正常训练（train-ori）和对抗增强训练（train-aug）后的结果有以下发现：

- 对比train-ori+dev-adv 和 train-aug+dev-adv 两行，在对抗性开发集上，基线模型、DFGN模型、SAE模型的答案EM分数分别提高了13.3%、31.3%和10.0%，推理依据EM分数分别提高了18.4%，41.9%和16.2%，这说明模型学习到了抵御对抗攻击的能力，鲁棒性大大提高。
- 对比 train-ori + dev-ori 和 train-aug + dev-ori 两行，在原干净开发集上，三个模型的性能均呈基本持平的表现，答案EM分数分别提高0.7%、提高0.7%、降低2.2%，推理依据EM分别提高3.3%、降低2.1%、降低2.1%。总的来说，对抗增强训练并不会大幅度削弱模型解决原始问答任务的能力。
- 对比 train-ori + dev-ori 和 train-aug + dev-adv 两行，经对抗增强训练之后的模型在对抗攻击下的性能，相比原模型在原干净开发集上的性能，下降程度已大幅度减少，基线模型的答案EM分数41.9%甚至已与原来的42.5%达到相当的水平，再次有力验证了模型已经具备较强的抵御攻击的能力。

上述实验结果都证明了本文所提出的对抗数据增强训练有助于提升问答模型的性能。**本方法的有效性主要在于增强模型的鲁棒性**，这也是本研究的出发点，通过在训练阶段向模型输入特殊设计的极具干扰性的噪声，使模型适应并学习分辨的能力，在提高寻找答案的能力的同时也提高过滤噪声的能力，从而变得更加稳健。在原始问答任务上，由于所构造的对抗样本本质上并没有引入全新的数据（问题答案对、用于推理的段落上下文都没有增加），因此，模型的问答能力难以得到进一步显著提升。另一方面，由于所添加的对抗句在一定程度上改变了数据的分布特征，使得训练和测试期间的数据分布存在一定差异，因此可能会导致在原始测试集上准确性的些许下降，先前几项关于对抗攻击的研究工作普遍发现了这样的问题(Jia and Liang, 2017; Wang and Bansal, 2018; Jiang and Bansal, 2019)。而经过本文方法增强训练的问答模型均没有表现出原始问答性能的降低，甚至还有略微提升，这说明了本方法能够促进模型在保持原始能力的基础上额外提高抵御干扰的能力，同时做到又好又稳。

3.4 和其他攻击方法的比较

本节将本对抗攻击策略与之前的两种攻击方法进行比较，并分析本方法的优越性。

- **AddSent**(Jia and Liang, 2017)：将整个问题表述进行修改，替换其中所有的名词、形容词、命名实体和数字，并从事先定义的集合中选择固定的伪答案，修改后的句子被添加到输入段落的末尾。该方法是针对单跳问答提出的，由于是在整个问题句子范围内修改，因此对抗句与原问题的关联度和相似度都会降低，也无法定位是在哪个推理环节出了错。并且，对抗句总是被加到段落最后，容易被问答模型捕获这样的特征而造成攻击评估无效。
- **AddDoc**(Jiang and Bansal, 2019)：将相关段落中的桥实体进行替换，并注入伪答案，以此构造整个对抗干扰段落作为额外的输入上下文，其中桥实体是从段落标题中抽取得到的。该方法的问题在于，如果数据集没有事先告知哪些段落是相关段落，或者没有提供文章标题，或者桥实体并不存在于文章标题中，那么就无法进行攻击数据的构造。

此外，这两种方法还有共同的一点不足是，都无法对对比类型和是否类型的问题实现对抗数据的构造。因此，在表2中以DFGN模型为例，仅比较在桥接类型和交集类型样本上的性能。

训练集	开发集	答案EM	答案F1	依据EM	依据F1	联合EM	联合F1
train-ori	AddSent	20.1	29.3	3.1	28.4	1.2	12.1
	AddDoc	38.9	54.8	27.9	63.7	16.4	40.5
	本文方法	**17.7**	**28.1**	**1.4**	**26.5**	**0.5**	**9.3**
train-aug	AddSent	35.3	46.7	7.1	30.6	3.1	18.8
	AddDoc	43.0	59.5	30.0	64.9	18.0	41.3
	本文方法	35.1	46.8	6.3	42.7	4.0	24.7

Table 2: 在桥接类型和交集类型的对抗开发集样本上，DFGN模型在本攻击方法下与其他两种攻击方法下的性能比较。加粗的为最低性能，即代表攻击效果最好。所有指标单位为%。

对抗攻击　使用在原训练数据集（train-ori）上训练的问答模型对三种对抗性开发集进行预测评估的实验结果显示，DFGN的问答准确性在本文方法的攻击下下降最多，验证了本文提出的对抗方法针对多跳问答是更具有混淆干扰性的。总的来说，本攻击方法有三个优点：(1) 可以对所有的推理类型进行更全面攻击，且不受推理依据可获得性的限制；(2) 攻击的成功率更高，本文认为这是由于所构造的攻击句子与问题具有更高的相似程度，以及保留了对多跳推理过程来说更为重要的实体；(3) 能够进行有针对性的攻击（如可以针对某一跳，只对相应的子问题作修改），从而对模型真正的多跳推理能力进行检验和分析。

对抗增强训练　实验还使用本文方法构造的对抗性增强训练数据（train-aug）对DFGN模型进行从头训练，然后在三个对抗性开发集上进行评估，以测试本增强训练的模型是否具备抵御各种多样化攻击的能力。可以看到，在AddSent、AddDoc、本文方法的攻击下，答案EM分数比原模型（train-ori）分别提高了15.2%、4.1%、17.4%。其中，AddSent和本文方法一样施加的是句子级别的扰动，而AddDoc是文档级别的扰动，因此将本增强训练直接迁移至后者所带来的提升效果相对偏少。虽然这两种方法所构造数据的特点和分布均和本文方法有很大不同，但是增强训练后的模型在所有攻击下的表现都获得了一致的提高，这样的结果证明，本文提出的对抗性增强训练所带来的模型鲁棒性加强是广泛的、普适的，能够帮助模型普遍地更好地抵抗其他各种攻击，而非只针对性地局限于本攻击方法。

3.5　对比实验

对比内容	推理类型	攻击目标	HotpotQA基线模型		DFGN模型		SAE模型	
			答案EM	答案F1	答案EM	答案F1	答案EM	答案F1
攻击不同跳	桥接类型	第一跳	**26.7**	**37.6**	**16.4**	**24.9**	**47.7**	**60.9**
		第二跳	27.1	38.0	17.1	25.6	48.0	61.1
		两跳	35.4	48.5	28.5	39.0	52.9	66.8
	交集类型	随机一跳	**28.8**	**43.6**	**26.5**	**37.4**	**48.1**	**63.2**
		两跳	31.5	46.8	32.2	44.6	49.7	65.2
攻击不同类型的词语	全部	关系词	**29.9**	**41.4**	**23.6**	**34.6**	**43.0**	**54.1**
		实体词	30.9	42.2	26.2	36.9	45.1	56.0

Table 3: 对比实验：针对不同跳和不同类型词语的攻击策略效果比较。加粗为攻击效果最好。

3.5.1　攻击不同跳的影响

将多跳问题拆解成多个子问题的操作使得本对抗攻击方法可以根据需要对推理链的任一部分进行攻击，以检测问答模型在不同推理阶段的能力。主实验通过修改桥接类型的第一跳和交集类型的随机一跳来构造对抗句，本部分尝试不同的修改策略，包括 (a) 同时修改第一和第二跳，即对整个问题句子进行修改；(b) 对桥接类型仅修改第二跳。

原模型在对抗性开发集上的结果如表3所示。当同时对两跳进行修改时，问答模型的答案预测能力相对较好，即说明注意力较少受到干扰，这是符合预期的，因为将两跳都改掉之后，生成的攻击句与正确的推理链完全没有重叠之处，无关性高，因此干扰混淆性较低。另外，对于

桥接类型，对第二跳进行攻击的攻击成功率比攻击第一跳的低，这也是容易理解的，攻击第一跳时，问答模型可能在一开始就被误导至错误的推理路径，离正确答案更远，因此最终能找回正确答案的概率也就越低；同时，由于第二跳问题表述与原问题中的文本保持了相同，这样的结果也意味着模型可能在第二跳推理过程中更倾向于使用简单的单词匹配等非多跳推理策略。

3.5.2 攻击实体的影响

本节通过修改问题文本中的命名实体而不修改表示关系的词语，来验证实体在多跳问答任务中起到很重要作用的假设。表3所示的结果验证了这样的假设。修改实体后的答案预测分数都高于修改关系后的，即攻击成功率低，这表明攻击句如果不含有问题所关心的实体，则可能导致相关程度较低，从而缺少干扰性。这也印证了第3.4节中所分析的本文攻击方法优于AddSent和AddDoc方法的原因之一。

3.5.3 不同训练增强数据比例的影响

(a) HotpotQA基线模型的答案EM分数。

(b) DFGN模型的答案EM分数。

Figure 4: 问答模型使用不同比例的增强数据训练后，在原开发集和对抗性开发集上的性能。横轴表示在全部原数据train-ori基础上扩充的对抗样本train-adv的比例，其中，第一列*0%*表示全部使用train-ori，最后一列*0+100%*表示使用全部train-adv且完全不使用train-ori。

主实验的对抗增强训练使用20%的对抗数据与所有原始训练数据相混合，本节尝试不同的数据增强比例。如图4中的结果所示，随着对抗训练样本数量的增加，模型可以更多地学习关于噪声的知识，从而提高抵御攻击的能力，因此在对抗性开发集上的性能呈现上升的趋势（橙色虚线）。在原开发集上，两个问答模型的性能总体上都变化不大，由于对抗样本引入的数据分布特征差异，出现了极微小的降低趋势（蓝色实线）。特别地，实验还尝试了仅使用所有对抗样本进行训练（0% train-ori + 100% train-adv），即最后一列*0+100%*所示，虽然问答模型能更有效地应对攻击（两个模型在对抗性开发集上答案EM分别达到49.0%和44.0%，均高出使用混合训练数据的情况），但是对干净数据的泛化能力会被大大削弱（两个模型在原开发集上的性能均低于使用混合训练数据的情况）。因此，需要谨慎选择合适的对抗数据增强比例，并且注意防止问答模型对对抗攻击数据过拟合，以平衡模型同时具备良好的问答准确性和鲁棒性。

3.6 案例研究

由于多跳问题本身难度不同以及问答模型设计的特点，模型对不同推理类型问题的解决能力存在很大差异，附录E对不同类型的样本进行了分类统计和分析。附录F还通过对一条真实样本的研究，来对本对抗攻击和对抗增强训练进行可解释性分析。

4 结论

本文提出了一项基于推理链的针对多跳问答的对抗攻击方法。通过将多跳推理过程形式化建模成推理链，可以识别不同的多跳推理类型，并对每种推理类型设计更有针对性的对抗攻击策略。本攻击方法通过识别不同跳对应的子问题文本，对关系词进行修改，来支持对其中任意一跳进行攻击，这可以帮助检测模型在推理过程中容易出错的部分。本文攻击评估的三个问答模型在面对攻击干扰时都出现了性能的下降，表明它们不够鲁棒，执行多步推理的可解释性有限。此外，利用所构造的攻击数据作为增强数据进行训练，可以普遍增强问答模型抵御攻击的鲁棒性，同时本工作也希望根据检测到的薄弱之处来促进开发出更好的问答模型。

第二十二届中国计算语言学大会论文集，第1页–第16页，哈尔滨，中国，2023年8月3日至5日。

参考文献

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question answering by reasoning across documents with graph convolutional networks. In *Proc. of NAACL*, pages 2306–2317, June.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NACCL*, pages 4171–4186, June.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Proc. of ACL*, pages 2694–2703, July.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2020. Hierarchical graph network for multi-hop question answering. In *Proc. of EMNLP*, pages 8823–8838, November.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proc. of EMNLP*, pages 2021–2031, September.

Y. Jiang and M. Bansal. 2019. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop qa. *ACL*.

T. Khot, P. Clark, M. Guerquin, P. Jansen, and A. Sabharwal. 2019. Qasc: A dataset for question answering via sentence composition.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, and D. Mcclosky. 2014. The stanford corenlp natural language processing toolkit. In *Proc. of ACL*.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

T. Onishi, W. Hai, M. Bansal, K. Gimpel, and D. Mcallester. 2016. Who did what: A large-scale person-centered cloze dataset. In *Proc. of EMNLP*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Ethan Perez, Patrick S. H. Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. *CoRR*, abs/2002.09758.

Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Proc. of ACL*, pages 6140–6150, July.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proc. of EMNLP*, pages 2383–2392, November.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proc. of ACL*, pages 784–789, July.

N. Shao, Y. Cui, T. Liu, S. Wang, and G. Hu. 2020. Is graph structure necessary for multi-hop reasoning?

L. Song, Z. Wang, M. Yu, Y. Zhang, R. Florian, and D. Gildea. 2018. Exploring graph-structured passage representation for multi-hop reading comprehension with graph neural networks.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proc. of NAACL*, pages 641–651, June.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2020. Is multihop QA in DiRe condition? measuring and reducing disconnected reasoning. In *Proc. of EMNLP*, pages 8846–8863, November.

Ming Tu, Guangtao Wang, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou. 2019. Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. In *Proc. of ACL*, pages 2704–2713, July.

Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. 2020. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9073–9080.

Yicheng Wang and Mohit Bansal. 2018. Robust machine comprehension models via adversarial training. In *Proc. of NAACL*, pages 575–581, June.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *b*, 6:287–302.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proc. of EMNLP*, pages 2369–2380, October-November.

Deming Ye, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, and Maosong Sun. 2019. Multi-paragraph reasoning with knowledge-enhanced graph neural network. *CoRR*, abs/1911.02170.

M. Zhou, M. Huang, and X. Zhu. 2018. An interpretable reasoning network for multi-relation question answering.

附录

A 相关研究

A.1 多跳问答

对于SQuAD(Rajpurkar et al., 2016; Rajpurkar et al., 2018)等单跳问答任务，问答模型可以通过简单地将问题与输入的单个段落中的句子进行匹配来检索答案。而对于多跳问答，只凭任一单句都不足以回答问题，模型需要结合至少两段文本的信息并基于它们进行推理。目前已公开了多个多跳阅读理解数据集，包括Wiki-Hop(Welbl et al., 2018)、ComplexWebQuestion(Talmor and Berant, 2018)、HotpotQA(Yang et al., 2018)等等。

很多多跳问答相关研究采用了图神经网络。DFGN(Qiu et al., 2019)根据输入上下文文本构建实体图，并在图上重复地执行单跳推理，在每一步动态地选择相关子图来传播信息。KGNN(Ye et al., 2019)从知识图谱中获取关系信息，基于共指关系来添加边以增强实体图。CogQA(Ding et al., 2019)通过添加每个现有节点的下一跳实体和可能的答案文本来逐步扩展认知图（cognitive graph）。与实体图不同，SAE(Tu et al., 2020)使用段落文本句子的嵌入表示作为节点，并将推理依据预测任务视为节点分类。HGN(Fang et al., 2020)构建了一个层次图来综合不同粒度级别的信息并促进它们之间的交互作用。类似的研究工作还包括文献(Song et al., 2018; Tu et al., 2019; De Cao et al., 2019; Shao et al., 2020)。除了图神经网络以外，其他的一些方法使用记忆网络来解决多跳问答(Zhou et al., 2018; Onishi et al., 2016)。

A.2 对抗攻击

尽管最近的研究工作展现了在多跳问答任务上的巨大进展，但机器阅读理解和多事实推理的真实能力以及模型的鲁棒性仍然值得怀疑。借鉴计算机视觉领域，可以通过对输入上下文添加轻微的扰动来对自然语言处理任务进行对抗性攻击，被攻击的模型预期会给出错误的输出。其中，添加的扰动不能与原始上下文段落内容有语义冲突，也不能改变原始正确答案。

针对问答，AddSent(Jia and Liang, 2017)是第一项研究对抗攻击的工作，通过替换给定问题中的命名实体、并与从事先定义好的集合中选择的伪答案相结合的方式来生成攻击句，将其添加到输入上下文的最后位置，实验显示16个模型在单跳问答任务SQuAD上都出现了性能的下降。AddSentDiverse(Wang and Bansal, 2018)在AddSent的基础上进行了改进，通过扩大伪答案候选集和改变攻击句的插入位置来使干扰更加多样化，同时发现对抗性再训练可以提高问答模型抵御攻击的鲁棒性。T3(Zhou et al., 2018)设计了树形自动编码器来对文本进行编码，使其保留句法结构和语义信息，然后在词级别和句子级别上施加基于最优算法的扰动，可以实现针对位置的攻击和针对答案的攻击。与单跳问答相比，多跳问答还存在另一种可认为是失败的推理情况，通常称为推理捷径。DiRe(Trivedi et al., 2020)通过在输入段落文本中删除部分推理依

据文本来探究非连续推理情况的存在，多跳问答模型不应当在推理依据缺失的情况下依旧给出正确答案，否则可认为存在推理捷径。为了探究是否存在简单的单词匹配策略，AddDoc(Jiang and Bansal, 2019)用不相关的表述替换相关段落中的真实正确答案和桥实体，并将这个修改后的段落添加到输入文本中作为干扰。

B 子问题拆解和推理类型预测的模型框架图

第2.2节中提出的子问题拆解和推理类型预测的模型框架图5所示。

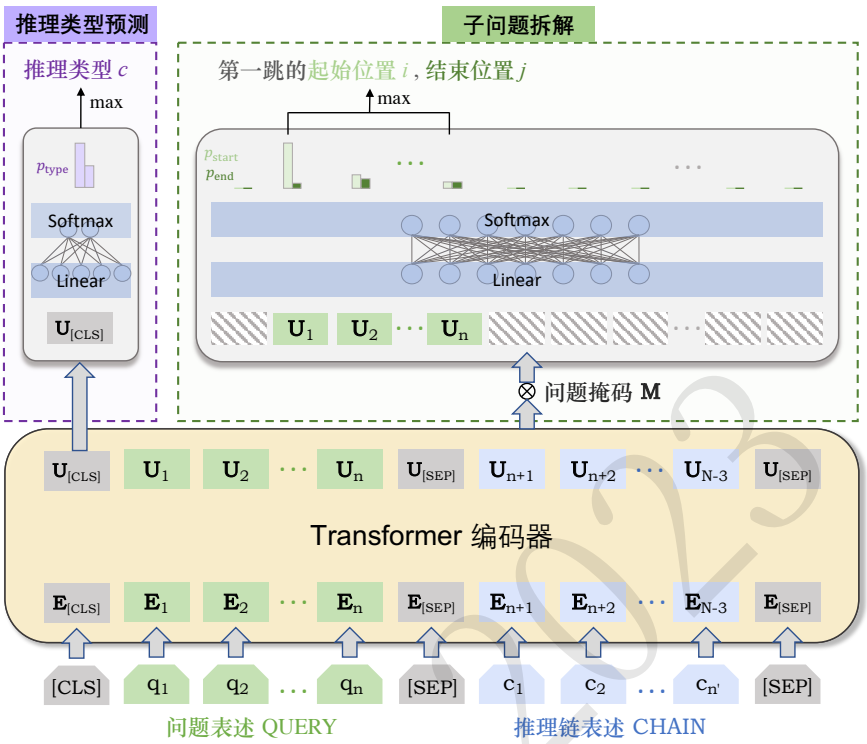

Figure 5: 针对桥接类型和交集类型的子问题拆解和推理类型预测的模型框架。

C 对抗攻击句子构造阶段的第三步中句式转换规则示例

表4展示了第2.3节中提到的所设计的句式转换规则中的部分示例。

疑问句模版：*What/Which $NP $VP ?* 陈述句模版：*The $NP of [Answer] $VP .* 疑问句示例：Which football team won the first prize in the 2022 World Cup? 陈述句示例：The football team of [Argentina] won the first prize in the 2022 World Cup.
疑问句模版：*When/Where $Do $NP $Verb $PP ?* 陈述句模版：*$NP $Verb-tense{2} $NP in [Answer] .* 疑问句示例：Where did Lisa and Jack meet for the first time? 陈述句示例：Lisa and Jack met for the first time in [Shanghai].
疑问句模版：*How $JJ $Be $NP ?* 陈述句模版：*$NP $Be [Answer] .* 疑问句示例：How tall is the highest mountain in the world? 陈述句示例：The highest mountain in the world is [8848.86 meters].
注：*$NP*、*$VP*、*$PP*、*$Verb*、*$JJ*、*$Be*、*$Do*分别为语法解析树中的名词短语、动词短语、介词短语、动词单词、形容词、be动词、助动词，*-tense{2}*表示使用疑问句模版中第2个元素的时态。

Table 4: 基于句法解析树的将特殊疑问句转换为一般陈述句的规则示例。

D 本方法所构造的对抗攻击样本示例

图6为每一种推理类型都展示了一条问答样本，并使用所提出的方法构造了不同的对抗攻击句子。

Figure 6: 根据不同的推理类型施加相应对抗攻击策略的示例及其对推理链的干扰解释。用阴影、实下划线、实线黑框标记的词语分别是问题实体、桥实体、答案实体，用虚下划线、虚线黑框标记的是本攻击方法构造并引入的干扰边（关系）和干扰节点（伪答案实体）。

E 不同推理类型的实验结果

在HotpotQA数据集中，四种推理类型所占的比例不同，问答模型对这四类的解决能力也有很大差异，实验对每一类分别统计了分数。由于三个问答模型的表现有相似的趋势，因此在表5中仅以DFGN模型的表现为例进行说明。

对比train-ori + dev-ori 和train-ori + dev-adv 两列可以发现，在对抗攻击下，桥接类型是最不鲁棒的类型，EM分数下降最多也降至最低，仅有14.6%，交集类型是第二易被攻击的类型，这两类也是数量占比最多的多跳问答类型。这证明了本文基于关系的攻击方法是对多跳问答有针对性的，在中间推理过程中，连续依次的推理跳可能会陷入错误的推理路径，以致偏离真正的答案。另一方面，DFGN擅长回答比较型问题，特别是是否类型，在原开发集上已经达到74.4%，远远高于其他三类，并在攻击下仍能保持在69.1%的高分，下降仅5.3%。对这类问题

推理类型	占比	train-ori +dev-ori	train-ori +dev-adv	train-aug +dev-ori	train-aug +dev-adv
所有数据	100%	54.2	23.6	54.9	40.1
桥接类型	49.6%	54.1	14.6	54.3	33.8
交集类型	32.0%	50.2	22.6	51.3	37.0
对比类型	12.9%	56.3	41.1	56.7	56.4
是否类型	5.5%	74.4	69.1	76.3	76.8

Table 5: DFGN模型在不同推理类型数据上的答案EM分数。所有指标的单位为%。

的攻击成功率较低的原因可能为，本攻击方法没有干扰或破坏问题所关心的属性或正确推理链，而只是添加了一个独立的“相同”或“不相同”的抽象性关系节点，因此对原问题的影响相对有限。

经过对抗增强训练之后，DFGN模型在四种类型上的表现都获得了提升。对比train-ori + dev-adv 和train-aug + dev-adv 两列，对抗增强训练后的模型抵御对抗攻击的能力得到了增强，在四种类型上分别提高了19.2%、14.4%、15.3%和7.7%。其中桥接类型提升最多，说明本增强训练方法可以弥补模型的薄弱之处，对最不鲁棒的部分进行着重加强；而对比类型和是否类型这两类原本就比较稳健的问题，已经达到了和原模型在原干净数据上相当的性能，说明这两类表现优异的推理类型已经几乎能够不再受对抗攻击的干扰。对比train-ori + dev-ori 和train-aug + dev-ori 两列，对抗增强训练后的模型在原干净开发集上的准确性都有了略微的提高，进一步证明了本方法的有效性。

F 案例研究

本部分以图7所展示的一条开发集案例进行研究，以对对抗攻击和对抗增强训练进行可解释性分析。实验计算了问题中每个单词对输入上下文中每个实体对应文本的注意力分数，并进行softmax归一化，然后对所有问题单词进行求和，从而得到每个实体所分配到的总的注意力权重。这些注意力权重一定程度上揭示了模型的推理过程。

在第一跳时，由于是基于上下文和问题之间的双向注意力机制进行推理初始化的，因此权重最高的大多是问题实体；另外，模型也成功定位到了桥实体*Shirley Temple*，分配了较高的注意力；其次是它们的相邻节点（比如在出现同一句句子中的实体），并依此传播信息；在第二跳时，正确答案*Chief of Protocol*已经成为注意力权重最高的实体。

但是在对抗测试样本（train-ori + dev-adv）上，模型在第一跳时就被对抗句分散了较多的注意力，因为句中包含了问题实体；在第二跳时，伪答案集中聚合了来自问题实体的信息，注意力分数超过了正确答案成为最高；在最终预测时，几乎所有概率都落在了重复多次出现的伪答案上。

经过对抗增强训练之后的模型（train-aug + dev-adv），在第一跳时对对抗句的注意力就受到了抑制（仅有0.28和0.34），对桥实体注意力大大加强（达到4.4和3.2），同时也提高了对正确答案的提前关注（达到4）；在第二跳时，对推理依据的关注程度也高于对抗句；最后的答案预测结果显示，对正确答案的预测概率从原来的0.57增加到0.93，不仅预测正确，还大大提高了置信度。

问题	What government position was held by the woman who portrayed Corliss Archer in the film Kiss and Tell? (《吻和倾诉》中扮演柯丽丝·阿彻的女士担任什么政府职位？)
段落1	Kiss and Tell is a 1945 American comedy film starring then 17-year-old Shirley Temple as Corliss Archer. The government position of Director of Diplomacy was held by the man who voiced Corliss Archer in the film Kiss and Tell. (《吻和倾诉》是1945年上映的美国喜剧电影，由当时17岁的秀兰·邓波儿饰演柯丽丝·阿彻。外交部主任的政府职位是由电影《吻和倾诉》中配音柯丽丝·阿彻的男士担任的。)
段落2	Shirley Temple Black was named United States ambassador to Ghana and to Czechoslovakia and also served as Chief of Protocol of the United States. The government position of Director of Diplomacy was held by the man who voiced Corliss Archer in the film Kiss and Tell. (秀兰·邓波儿·布莱克被任命为美国驻加纳和捷克斯洛伐克大使，并担任美国礼宾司司长。外交部主任的政府职位是由电影《吻和倾诉》中配音柯丽丝·阿彻的男士担任的。)
正确答案	Chief of Protocol (礼宾司司长)
预测答案	Director of Diplomacy (外交部主任)

注：用绿色、橙色、蓝色、红色标注的分别是问题实体、桥实体、答案实体、伪答案和伪关系；问题中划横线的为第一跳子问题；段落文本中划虚线的为构造的对抗句，对抗句被插入到每个输入段落的随机位置。

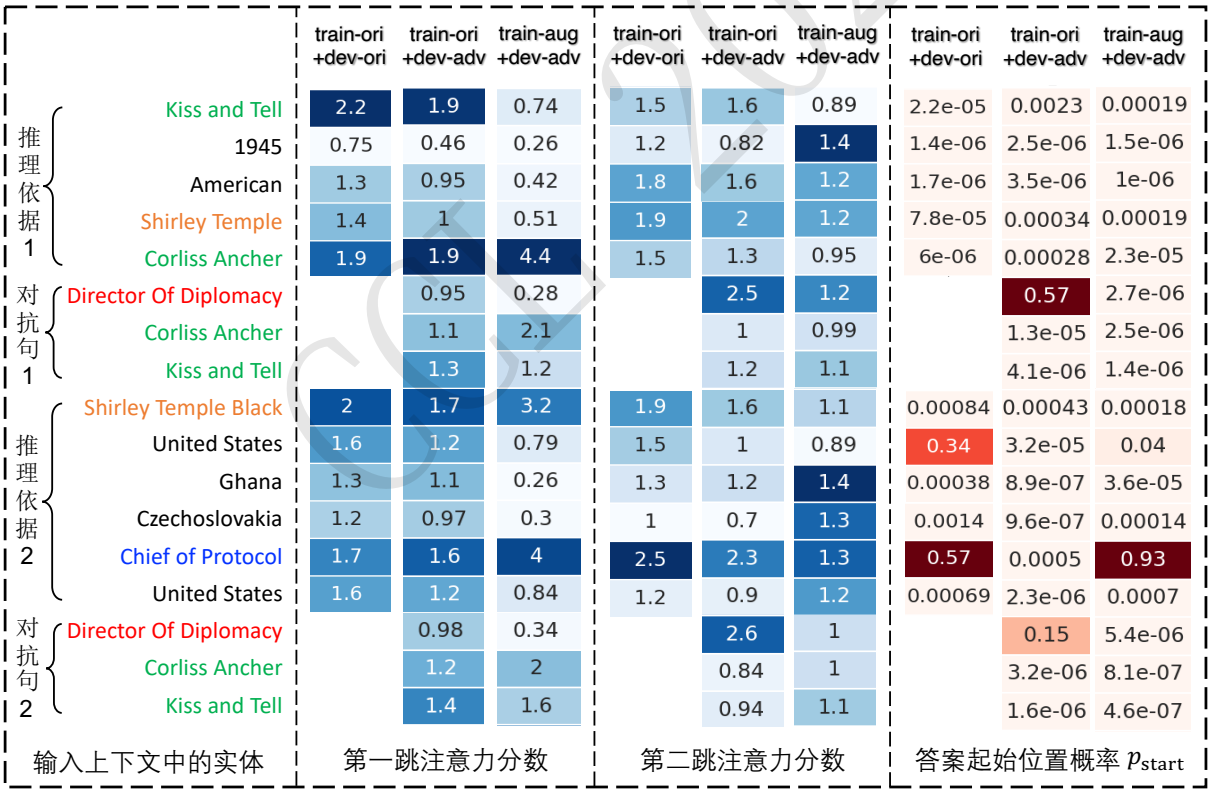

Figure 7: 案例研究：使用不同训练集训练的DFGN模型在原开发集和对抗性开发集上，输入上下文中实体在每一跳受到的注意力权重分数。