FaiRR 模型实现自然语言的演绎推理

王嘉宁

1/11/2024

摘要

自人工智能这一概念兴起以来，构建能够根据给定知识自动推理的系统，尤其是针对自然语言实现自动推理的系统，就一直是人们的重点目标之一。近些年来，先后有许多工作在自然语言的演绎推理领域取得了新的进展，包括基于 transformer 模型 [18] 构建的 RuleTaker 模型 [1] 和 ProofWriter [16] 模型。但这些模型无一例外都是黑盒系统，在生成蕴含推理步骤的证明图时经常会出现一些莫名其妙、不知所谓的错误，很难不让人怀疑这些模型内部推理过程的忠实性。本文通过改用 FaiRR 模型，将推理过程分为规则选择、事实选择和知识组合三个步骤，提高了推理过程的可解释性，并通过数据隔离，即让与推理无关的事实对模型的事实选择和知识组合部分不可见，提高了推理过程的忠实性。通过与 ProofWriter 模型在同数据集上的鲁棒性对比实验，我们发现，FaiRR 模型在同深度的逻辑推理任务中表现与 ProofWriter 模型基本一致，但在替换主语和属性等鲁棒性实验中表现明显更佳，并且基本杜绝了不知所谓的错误，FaiRR 的推理过程明显更容易被人类所理解。

关键词： 自然语言处理；演绎推理；大模型；transformer

1 引言

1.1 人工智能与逻辑推理

什么是人工智能? 要回答这个问题，我想第一要务就是搞懂什么是智能。以我的理解，能够完成人类通过逻辑思考可以完成的任务就足以称为智能。那么基于自然语言的逻辑推理就应当是实现人工智能的重要一环。自从 1956 年 McCarthy 等人提出了构建自动进行逻辑推理系统的理论基础 [10] 之后，如何更高效、更准确的实现自然语言的演绎推理一直是人工智能领域的重点。2020 年，Clark 等人对 McCarthy 的理论进行了现代化的更新，并提出了基于 transformer 的模型 RuleTaker。这个模型可以通过对演绎推理的模拟来预测要证明的结论是否正确。更具体而言，模型根据已知的信息生成新的陈述，然后判断这些陈述中是否蕴含要证明的结论。但 RuleTaker 模型只能预测结论的正确性，无法生成推理步骤，即证明图。为了弥补这个缺陷，Tafjord 等人在 2021 年进一步发展了可以生成推理步骤的模型，ProofWriter 模型，但这些模型都还有一个问题没有解决，那就是这些系统并未明确确保从规则/事实选择到生成中间推理的因果关系。由于这些系统本质上是黑盒模型，不清楚模型是否在没有外部强制的情况下隐式学习到这些约束。已经有人对模型内部推理过程的忠实性提出了质疑 [7]。

由于模型在输入时可以使用完整的信息，它可能使用了理论的其他部分，而不仅仅是预测的证明，来生成推理。

1.2 FaiRR 模型

在 2022 年，Soumya 等人通过开发一个模块化框架来试图解决演绎推理任务中的这些缺点 [14]。在此之前的方法是在单一步骤中生成证明和结论，而新的框架将这一过程分为三个步骤：规则选择、事实选择和知识组合。规则选择步骤决定要在迭代推理步骤中使用的相关规则，而事实选择使用这个规则选择相关的事实。然后，知识组合步骤仅使用所选规则和事实进行推理，生成下一个中间推理。不断循环这个过程，直到生成了要证明的结论或其反例，我们就可以证明或证伪了。值得注意的是，我们严格限制了我们框架每个步骤可访问的信息，以使推理过程更加忠实。例如，事实选择步骤仅依赖于所选规则，而不是规则库中的所有规则。此外，生成的推理明确依赖于所选规则和事实，而不是先前工作中的所有规则和事实。这使得证明图成为选择步骤的副产品，因为我们不需要生成任何单独的证明。由于我们对每个步骤的输入进行了约束，这也使得每个子问题更容易学习，从而生成一个更加健壮的推理模型。

2 相关工作

2.1 文本推理

文本推理是自然语言处理领域中一个已经被深入研究的问题。自然语言推理（NLI） [2] 是进行文本推理以回答给定假设的陈述是否蕴含、矛盾或中性的最突出的任务之一。最近的数据集如 HotpotQA [20]、bAbI [19]、QuaRTz [17]、ROPES [6]、CLUTRR [15] 等研究了对文本输入进行推理的不同方面。这些任务通常需要隐式推理，即模型需要在内部推断解决任务所需的规则，这些规则不会显式的给出。相反，RuleTaker [1] 模型是基于显式推理的（也称为演绎推理）。

2.2 形式推理

有一些先前的工作试图通过直接从文本中解析形式语言来解决推理预测的问题。例如 Rocktaschel 等人 [12] 就是使用神经网络从自然语言中解析形式逻辑，然后进行推理。虽然这种方法更具象征性，但在解析文本时可能会遇到许多困难 [5]。因此，本文所介绍的模型没有采用这种方法，而是直接对给定的自然语言文本进行推理，使其在下游应用中更有用。

2.3 模型可解释性

随着预训练语言模型 [3]，RoBERTa [8] 的出现，越来越多的研究趋向于使用高精度解决各种推理任务。判断这些模型的可信度 [4] 主要是通过了解模型是否实际上学会了解决任务，还是依赖于某些隐性的捷径模式。基于显著性的解释 [9] 主要侧重于识别那些有助于模型解决任务的输入文本中的重要短语。与此相反，证明生成的任务侧重于从给定理论到结论生成推理链。因此，对于最终用户来说，证明链更容易理解，也更有助于调试任何系统性的模型错误。

3 本文方法

3.1 符号说明

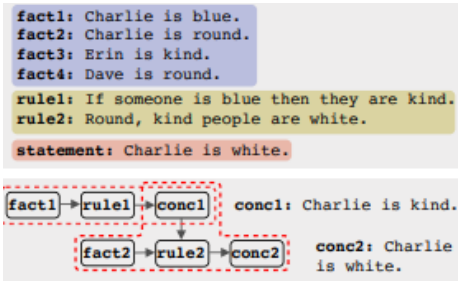

图 1. 理论、陈述以及证明图

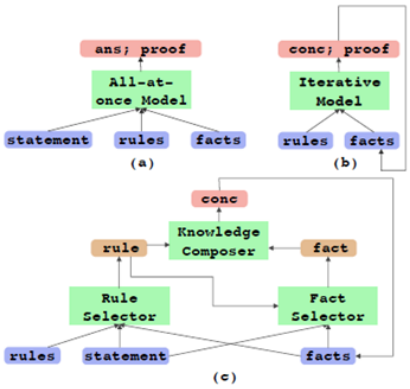

图 2. 不同模型中的推理过程

一个理论 T 包括一组用自然语言表达的事实 $F = f_1, f_2, ..., f_n$ 和规则 $R = r_1, r_2, ..., r_m$。图 1 中展示了一个理论的示例。在这里，蓝色和黄色框中的句子分别表示事实和规则。此外，证明图是一个有向图，连接着描述如何从理论中获得特定推理的事实和规则。在图 1 中，证明图展示了生成推理“Charlie 是白色”的步骤。为了生成证明图，我们可能需要生成一些中间结论 c_i。这些推断被视为理论中扩展事实的一部分。例如，“Charlie 是善良的”是生成正确证明图所需的中间推理。

3.2 任务设置

演绎推理：演绎推理任务描述如下：给定一个理论 T 和一个陈述 s，预测理论是否支持该陈述（蕴含预测），如果是，则生成支持该陈述的证明图（证明生成）。对于图 1 中的示例理论和陈述，我们可以看到该陈述确实被理论蕴含，并生成了相应的有效证明图。该任务的主要目标是评估模型是否能够生成有效的推理链，以证明其蕴含预测的正确性。

推理鲁棒性：我们考虑一个辅助任务，用于评估模型使用的推理能力的鲁棒性。设 P 为一个扰动函数，将给定理论 T（陈述 s）修改为理论 T'（陈述 s'），使得（T' s'）在自然语言形式上只有一些表面变化，但仍需要与（T s）相同的推理过程。例如，改变理论中的主语就是这样的扰动函数的一个示例。我们对每个理论陈述对 T s 进行扰动，将创建的等价集定义为 $E(T, s) = (T'_1, s'_1)...(T'_N, s'_N)$，其中每个 T'_k, s'_k 是通过扰动原始理论得到的，N 是每个理论的总扰动次数。请注意，通过控制 P 的随机性，可以生成不同的 T'_k, s'_k 对。该任务的主要目标是评估模型的预测是否在输入理论变化的情况下一致。

评估指标：我们在研究中主要考虑了两个方面来评估模型性能：（1）蕴含准确度，用于衡量模型准确预测真实陈述蕴含的能力。（2）证明准确度，用于衡量模型是否能够准确预测陈述的证明图。遵循 Saha 等人 [13] 和 Tafjord 等人 [16] 的方法，我们使用了严格的证明评估指标，即预测的证明必须与标准证明完全匹配，并且蕴含被正确预测，才能算生成了正确的证明图。

3.3 FaiRR 模型

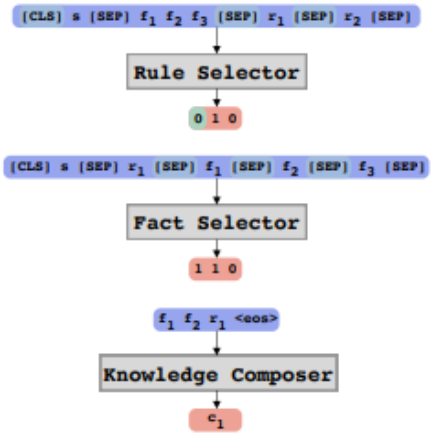

图 3. FaiRR 模型的不同组件

正如图 1 中的示例所示，要通过演绎推理可靠地生成证明图，模型需要生成多个一跳的中间结论，而如何生成中间结论就是 ProofWriter 模型与 FaiRR 模型的最大区别。如图 2 所示，ProofWriter("All") 模型直接生成预测结果与证明图，ProofWriter("Iter") 模型使用所有已知的事实与规则生成中间结论，并将其加入已知事实。而 FaiRR 模型首先使用理论中的规则和事实选择一个规则 r, 随后基于所选规则 r 从事实列表中选择相关的事实。这一步中不可以使用理论中的其他规则 R
r。最后，所选的规则和事实一起用于生成新的结论 c_i。在这个框架中，FaiRR 模型首先通过选择规则步骤明确了一步的证明方法，然后根据事实推理生成中间结论，这样证明图就成为了整个过程的副产品，在得到证明结果时证明图也就同时得到了。

FaiRR 模型结构：首先 FaiRR 模型是一个迭代模型，它逐步生成一跳的中间结论，如图 2 所示，FaiRR 模型有四个组件，分别如下：

规则选择器（RS）：规则选择器是一个基于 RoBERTa 的分类模型 [8]，它以所有的陈述、事实和规则作为输入，并选择一个规则，用于在当前迭代步骤中生成一个中间结论。它的输入形式为 $[CLS]s[SEP]F[[SEP]r_i]_m[SEP]$，并通过线性分类器层对从 [CLS] 令牌和规则前的 [SEP] 令牌中分类的标记嵌入进行分类，生成一个独热输出向量。每个分类是二元分类，但总体上只有一个标记具有正类别。这里 s 表示陈述，F 是事实，并与在先前迭代中生成的任何中间结论连接，r_i 表示包含总共 m 个规则的理论中的第 i 个规则。$[]_m$ 表示持续的连接。规则选择器的示例输入和输出如图 3 所示。如果选择了 [SEP] 标记，我们就选择相应 [SEP] 标记后面的规则；如果选择了 [CLS] 标记，我们就停止迭代。也就是说，[CLS] 选择作为我们迭代模型的停止信号。注意到可能有多个可能的候选规则，因为对于给定的理论，可能存在多个一跳推理，所以我们在每次迭代中随机选择可能的候选规则之一。

事实选择器（FS）：事实选择器是基于 RoBERTa 的标记分类模型 [8]，它接受陈述、由规则选择器选择的规则以及理论中的事实，然后预测一组与规则一起用于生成中间结论的候选事实。它的输入形式为 $[CLS]s[SEP]r[[SEP]f_i]_n[SEP]$，其中 s 是陈述，r 是所选规则，f_i 是包含总共 n 个事实的理论中的第 i 个事实。注意，这些事实还包括先前生成的任何中间结论。$[]_n$ 表示持续的连接。FS 的输出是通过使用线性层对位于事实前的每个 [SEP] 标记嵌入进行分类生成的，通过它确定是否选择相应的事实。事实选择器的示例输入和输出如图 3 所

示。注意到可能有一些规则可以联合推理多个事实以生成结论，图 1 中的“rule2”就是这样一条规则的例子，因此该组件具有选择多个事实的能力。

知识组合器（KC）：知识组合器是一个生成式文本到文本的 transformerT5 [11] 模型，可以将一组事实和一个规则组合起来输出一个新的结论。模型的输入是所选的事实和规则串联在一起，输出是中间结论。知识组合器的示例输入和输出如图 3 所示。

求解器：最后一个组件是求解器，它在所有迭代终止后生效（即一旦规则选择器选择了 [CLS] 令牌表示停止迭代推理生成过程，求解器就开始起效）。与 ProofWriter 类似，我们的求解器在目前已经生成的中间推理中搜索陈述（字符串匹配）。如果找到，则预测理论蕴含该陈述。如果找到陈述的否定，则预测不蕴含。如果这些都不存在，则预测“未知”，即它无法证明或证伪该陈述。而证明图是通过在每一步使用所选规则和事实生成的一跳证明来构建的。例如，在图 1 中，红色虚线框（一跳证明）被拼接在一起以组装完整的证明。对于蕴含为“未知”的情况，返回的证明为“无”，因为在理论中不存在该陈述的证明。求解器不是一个可学习的模块。

3.4 训练方法

由于 FaiRR 模型分为四个组件，其中最后一个组件求解器不需要训练，其他三个组件是分开并按顺序训练的。为了与 ProofWriter 模型进行对比，我们使用与 ProofWriter 相同的数据集。更具体而言，假设对于给定的理论 $T = R + F$，使用规则 r 和事实 f 可以得到可能的中间推理 c。对于 ProofWriter 的一个训练实例，它使用输入 R, F, 输出 c, r, f。我们处理相同的实例以生成三个训练实例，分别用于规则选择器、事实选择器和知识组合器，具体如下：

RS 输入 $= R, F$； RS 输出 $= r$，

FS 输入 $= r, F$； FS 输出 $= f$，

KC 输入 $= r, f$； KC 输出 $= c$。

两种选择器模型将陈述 s 作为模型的输入。此外，规则选择器和事实选择器的输出将被转换为类标签而不是文本，因为我们的选择器是分类模型。我们使用交叉熵损失来训练规则选择器，使用二元交叉熵损失来训练事实选择器。知识组合器则基于语言建模损失进行训练。

3.5 推理过程

完成训练后，在进行推理时，首先规则选择器选择一个规则，用于生成一步的结论。然后，事实选择器根据所选规则选择一些事实，接着将它们集体传递给知识组合器生成结论。这个三步的流程被迭代运行，直到规则选择器通过选择 [CLS] 令牌来输出停止信号，退出迭代。一旦迭代完成，求解器利用已生成的中间推理来确定陈述是否被蕴含，并生成相应证明。

4 复现细节

4.1 与已有开源代码对比

本次复现主要参考了 FaiRR: Faithful and Robust Deductive Reasoning over Natural Language [14] 作者 Soumya Sanyal 公布在 github 上的源码，具体地址为 https://github.com/ink-usc/fairr。其与 ProofWriter 模型最大的区别就是将其拆分成了四个不同的组件，体现在代码

里则为事实选择器、规则选择器、知识生成器和求解器分别封装在不同的 py 文件中。具体而言，例如事实选择器的代码主要逻辑为接受包括超参数设置、模型选择设置、数据集设置和训练设置等数据，然后从 huaggingface.co 网站下载对应的预训练模型，并在设定的数据集上训练。由于 FaiRR 模型的组件都是基于预训练好的模型，因此这里的训练只是针对 ProofWriter 论文所用数据集的微调。

4.2 实验环境搭建

本次论文复现过程主要是在服务器上通过搭建 conda 虚拟 python 环境实现的。虽然原论文作者所提供开源代码中虽然提供了 requirements 文档，但在实际搭建环境了还是出现了一些问题。搭建环境时发现无论选择哪个 py 版本，requirements 文档中的一部分 py 包版本都会发生冲突，甚至有一部分包的版本要求之间本身就有冲突，无法同时启用。经过多次尝试后，最终选定了 py3.9 版本，并对开源代码中给出的 py 包版本作出了修改，并删除了一些过于老旧已被弃用的包，并在代码中修改了对应的部分 (注：修改部分很少，且都是不影响代码逻辑的修改，基本只是把一些老旧弃用的函数修改为新的替代版本)，修改后的 requirements 文档已经放在了以 FaiRR 为名的压缩文件夹中。本次实验所用 GPU 为 RTX4090，DL 开发平台为 pytorch_lightning。

4.3 界面分析与使用说明

本次复现中实验通过执行 py 命令行的方式实现，主要所用的命令行格式如下 (实际使用时按需调整)：

创建数据集:

python process_proofwriter.py --dataset pwq_leq_0to3 --fairr_model fairr_rule --arch roberta_large

训练模型:

python main.py --override fairr_ruleselector,pwq_leq_0to3_OWA_rule

评估模型蕴含准确度与证明准确度:

python main.py --override fairr_inference,evaluate --dataset pwu_leq_3_OWA --ruleselector_ckpt <path_to_trained_checkpoint> --factselector_ckpt <path_to_trained_checkpoint> --reasoner_ckpt <path_to_trained_checkpoint>

创建鲁棒性数据集:

python utils/fact_augmentation.py --split test --dataset depth-3 --names

评估模型鲁棒性

python main.py --override fairr_inference,evaluate --dataset pwur_leq_3_eq_2_name_OWA --ruleselector_ckpt <path_to_trained_checkpoint> --factselector_ckpt <path_to_trained_checkpoint> --reasoner_ckpt <path_to_trained_checkpoint>

4.4 创新点

本文与之前的模型如 ProofWriter 等的最大区别、也是最大创新在于模仿人类逻辑推理过程，将自然语言的演绎推理模型拆分成规则选择器、事实选择器、知识组合器和求解器四

FaiRR与ProofWriter模型测试结果比较						
	蕴含准确度			证明准确度		
推理深度d	PW(“iter”)	FaiRR(author)	FaiRR(me)	PW(“iter”)	FaiRR(author)	FaiRR(me)
3	99.7	96.6	96.7	99.1	95.3	94.2
0-3	99.7	99.6	99.2	99.7	99.6	98.9

图 4. 原数据集实验结果

个组件，同时把与某个组件无关的事实或规则与这个组件隔离开来，以防止模型隐性学习到一些问题与答案的错误关联。也正因此，尽管 FaiRR 模型与 ProofWriter 模型都是基于预训练的 transformer 大模型，但 FaiRR 模型在鲁棒性测试上表现明显优于 ProofWriter 模型的原因。

5 实验结果分析

5.1 数据集

为了与 ProofWriter 模型对比，我们使用了与其一致的 D* 数据集进行实验。这是一组数据集的集合，即 D0、D1、D2、D3、D0-D3 和 D5 数据集，Dn 即表示推理深度最多为 n。这些数据集中的理论都是程序生成的，具有递增的推理深度。例如，D3 数据集包含需要最多 3 跳推理步骤的陈述。D0-D3 包含 D3 中所有理论以及 D0-D2 训练集理论的约 20 此外，我们生成了三个用于评估模型鲁棒性的数据集：

主体鲁棒性：我们通过使用一些原数据集分布外的适当且常见的名称对理论中的主体进行替换。例如，在图 1 中，“Charlie” 可以被替换为 “Paul”，而这在 D* 数据集中没有使用。通过反复扰动理论中的所有适当和常见名称，我们为 D3 数据集的每个理论生成五个新理论。

属性鲁棒性：这里我们替换原数据集分布外的属性。例如，图 1 中的 “blue” 可以被替换为 “soft”。与上述相似，我们为 D3 数据集的每个理论生成五个新理论。

主体 + 属性鲁棒性：这是主体和属性鲁棒性的组合，用于研究当大多数训练词汇被分布外单词替换时模型的性能。每个理论都有新的主体和属性。

5.2 实验结果

如图 4，在原数据集上的实验结果表明，在不同深度的数据集中，无论是在蕴含准确度上，还是证明准确度上，FaiRR 模型都与 ProofWriter 模型差别不大，说明在 FaiRR 模型中所加的数据隔离限制并没有影响模型的逻辑推理能力。

如图 5，在鲁棒性数据集上的实验结果表明，在替换原理论的主体后，ProofWriter 模型的蕴含准确度和证明准确度都出现了不小幅度的下降，而 FaiRR 模型的两种准确度都几乎不变。这暗示了我们 ProofWriter 模型可能确实部分的隐性学习到了理论与结论的关系，而不是全都通过逻辑推理得到的答案。而 FaiRR 模型不受影响的事实表明它并没有这个缺陷。另

FaiRR与ProofWriter模型测试结果比较						
鲁棒性特征	蕴含准确度			证明准确度		
	PW(“iter”)	FaiRR(author)	FaiRR(me)	PW(“iter”)	FaiRR(author)	FaiRR(me)
主体	89.6	96.8	95.9	88.4	95.9	95.2
属性	97.8	96.7	89.3	97.4	95.6	88.4
主体+属性	94.8	95.4	89.0	93.4	94.3	88.1

图 5. 鲁棒性数据集实验结果

参数名	模型名		
	FactSelector	RuleSelector	KnowledgeComposer
train_batch_size	16	16	16
eval_batch_size	16	16	16
lr	1e-5	1e-6	1e-3
max_epochs	10	15	10
weight_decay	0.01	0.01	0.0
dropout	0.1	0.1	0.1
optimizer	adamw	adamw	adamw
lr_scheduler	linear warmup	linear warmup	linear warmup

图 6. 训练参数设置

外，可以注意到实验结果中本人的结果与作者在论文中给出的数据有差距，经过与作者的邮件讨论，基本可以确认是由于 batch_size 的减小导致的。但修改 batch_size 的原因是实验条件所限，由于没有作者所使用的 48GB 显存 GPU，如果使用梯度累加方法，实验周期又会拉的过长，因此只能降低 batch_size。

5.3 训练参数设置

模型训练过程中所用到的主要参数设置如图 6 所示。

6 总结与展望

本次复现成功使用了 FaiRR 模型进行了自然语言的演绎推理任务，FaiRR 模型是一个基于三个模块化组件的忠实而强大的演绎推理模型：分别是规则选择器、事实选择器和知识合成器。FAIRR 模型通过设计确保了从生成证明到蕴含预测的因果关系真实可靠。通过对语言变体鲁棒性的实验，我验证了该模型的有效性。美中不足的是，实验只在推理深度小于等于 3

的数据集上进行了训练和测试，并没有在推理深度更深的数据上进行实验，还无法得知模型是否能在更有挑战性的任务中依旧保持高准确率。希望未来能够基于 FaiRR 模型，作出进一步的优化，使其推理能力更强。

参考文献

[1] Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, 2021.

[2] Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, 2005.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

[4] Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? In *Annual Meeting of the Association for Computational Linguistics*, 2020.

[5] Aishwarya Kamath and Rajarshi Das. A survey on semantic parsing. *ArXiv*, abs/1812.00978, 2018.

[6] Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. Reasoning over paragraph effects in situations. *ArXiv*, abs/1908.05852, 2019.

[7] Zachary C. Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, jun 2018.

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019.

[9] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Neural Information Processing Systems*, 2017.

[10] Allen Newell and Herbert A. Simon. The logic theory machine-a complex information processing system. *IRE Trans. Inf. Theory*, 2:61–79, 1956.

[11] Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2019.

[12] Tim Rocktäschel and Sebastian Riedel. End-to-end differentiable proving. *ArXiv*, abs/1705.11040, 2017.

[13] Swarnadeep Saha, Sayan Ghosh, Shashank Srivastava, and Mohit Bansal. Prover: Proof generation for interpretable reasoning over rules. In *Conference on Empirical Methods in Natural Language Processing*, 2020.

[14] Soumya Sanyal, Harman Singh, and Xiang Ren. FaiRR: Faithful and robust deductive reasoning over natural language. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1075–1093, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[15] Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. Clutrr: A diagnostic benchmark for inductive reasoning from text. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

[16] Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. Proofwriter: Generating implications, proofs, and abductive statements over natural language. In *Findings*, 2020.

[17] Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. Quartz: An open-domain dataset of qualitative relationship questions. *ArXiv*, abs/1909.03553, 2019.

[18] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017.

[19] Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv: Artificial Intelligence*, 2015.

[20] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing*, 2018.