哈爾濱工業大學

毕业设计（论文）开题报告

题　目：基于奖励模型的网络安全问答生成策略研究

专　　业　信息安全

学　　生　王宇

学　　号　2021110556

指导教师　叶麟副教授

日　　期　2024 年 11 月 18 日

哈尔滨工业大学教务处制

目　录

1 课题来源及研究的目的和意义

1.1 课题来源

本研究的课题来源于对大语言模型在网络安全问答生成中应用的探索需求。随着网络安全威胁的不断演变，及时有效地获取网络安全信息变得尤为重要。现有大语言模型在通用知识方面表现出色，但在专业领域，如网络安全中的病毒检测任然存在不足。本研究旨在通过引入奖励模型来提升问答生成的质量和领域相关性，以解决这一问题。

1.2 研究目的

本研究的目的在于实现大语言模型在网络安全问答生成中的优化。具体而言，通过引入奖励模型，我们旨在提升模型的专业性、回答的准确性以及对用户需求的响应能力。此外，本研究还旨在构建一个自动化系统，该系统能够在网络安全领域提供高质量、可靠的问答服务，以满足日益增长的网络安全信息需求。

1.3 研究意义

1. 推动网络安全领域智能化发展：随着网络安全威胁日益复杂和多变，传统的防护手段已经无法满足实时应对和有效预警的需求。通过本研究，大语言模型能够在网络安全问答领域提供更加精准、高效的信息获取机制，帮助安全从业者和普通用户更迅速地识别和应对潜在威胁，从而推动网络安全智能化技术的发展，提升安全防护能力。

2. 提升大语言模型在专业领域的适应性与实用性：虽然现有的大语言模型在广泛的通用任务中表现出色，但在诸如网络安全等专业领域，其能力尚有不足。本研究通过奖励模型与微调技术的结合，探索了一种新的方式使大语言模型能更加深入理解网络安全的复杂场景和问题。这一研究将推动人工智能在专业领域的深入应用，特别是在安全领域的知识迁移与精度提升。

3. 创新性地结合强化学习与问答生成技术：奖励模型作为强化学习中的重要机制，已经在多个领域取得了成功。本研究将这一机制创新性地引入网络安全问答生成任务中，结合奖励反馈机制与问答生成模型，通过优化生成过程的质量和准确性，为网络安全领域提供了更加智能和精细化的技术支持。这一创新不仅丰富了大语言模型在行业应用中的方法论，也为未来人工智能在安全防护中的作用提供了新的发展方向。

2 国内外在该方向的研究现状及分析

2.1 国际研究现状

2.1.1 奖励模型在语言生成中的应用

在 Daniel M. Ziegle 团队的研究方案中[1] 选取了 774M 参数版本的 GPT-2 语言模型，将监督微调的 BookCorpus 数据集作为数据集输入，其奖励模型的训练流程如图：

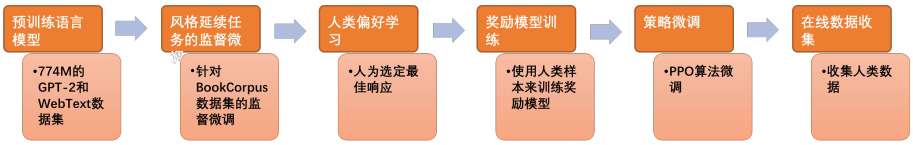

图 1: 奖励模型在语言生成中的应用流程

并且在奖励模型的训练中规定损失函数为：

$$\text{loss}(r) = \mathbb{E}_{(x,\{y_i\}_i,b)\sim S}\left[\log \frac{e^{r(x,y_b)}}{\sum_i e^{r(x,y_i)}}\right]$$

其中，r 是奖励模型，x 是输入，y_i 是可能的输出，b 是人类选择的最佳输出。这个损失函数是基于 softmax 函数的交叉熵损失，用于训练奖励模型以预测人类偏好的最佳输出。

在策略微调中使用 PPO 算法对策略 π 进行微调，优化以下修改后的奖励：

$$R(x,y) = r(x,y) - \beta \log \frac{\pi(y|x)}{\rho(y|x)}$$

其中：$r(x,y)$ 是奖励模型给出的奖励值，$\pi(y|x)$ 是策略 π 在给定输入 x 时生成输出 y 的概率，$\rho(y|x)$ 是初始的语言模型在给定输入 x 时生成输出 y 的概率。，β 是 KL 散度的惩罚系数，用于控制策略 π 与初始语言模型 ρ 之间的差异。

2.1.2 强化学习在语言生成中的应用

Nisan Stiennon 等研究者在使用 GPT-3 参数分别为 13 亿（1.3B）和 67 亿（6.7B）的模型的强化学习（RL）在语言生成任务采用了类似的方法[2]，将人类反馈与强化学习相结合，以提升语言模型在文本摘要任务中的表现。

其研究表明：

- 英语摘要方面，利用人类反馈进行的训练明显优于非常强大的基线。
- 与监督模型相比，人类反馈模型对新领域的泛化效果要好得多。

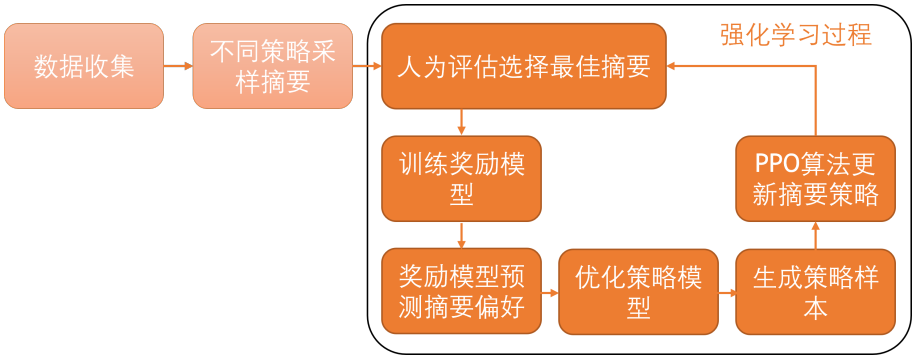

图 2: 强化学习流程

并在其中注重策略的完整奖励：

$$R(x,y) = r_\theta(x,y) - \beta \log\left(\frac{\pi_\phi(y|x)}{\pi_{\text{SFT}}(y|x)}\right)$$

其中，$R(x,y)$ 是给定输入 x 和输出 y 的完整奖励。$r_\theta(x,y)$ 是奖励模型对输入 x 和摘要 y 的评分。β 是 KL 散度惩罚系数。$\pi_\phi(y|x)$ 是 RL 策略生成摘要 y 给定 x 的概率。$\pi_{\text{SFT}}(y|x)$ 是监督学习策略生成摘要 y 给定 x 的概率。

与 PPO 算法的更新：

$$L^{CLIP}(\theta) = \min\left(r_t(\theta)A_t,\ clip\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)A_t\right)$$

2.1.3 大语言模型在网络安全领域的应用

网络威胁情报（CTI）可帮助组织了解潜在网络犯罪分子用来抵御网络威胁的策略、技术和程序。为了保护组织的核心系统和服务，安全分析师必须分析有关威胁和漏洞的信息。但是，分析大量数据需要花费大量时间和精力。为了简化这一过程，通过从传统的 BERT-BiLSTM-CRF 模型中删除 BiLSTM 层。该模型利用基于深度学习的语言模型的优势，有效地从威胁中提取关键威胁情报和新信息。在 BERT-CRF 模型中，BERT 生成的令牌嵌入被直接馈送到条件随机场（CRF）层中，以实现高效的命名实体识别（NER），从而避免了对中间 BiLSTM 层的需求。该模型在实际场景中可以达到 82.64% 的准确率，而对于特定于恶意软件的威胁情报数据，它实现了令人印象深刻的 93.95% 准确率[3]。

2.2 国内研究现状

2.2.1 通过深度学习检测网络异常流量

国内研究中，有基于深度学习的网络异常流量检测方法优劣分析[4]：

1. BiLSTM+ 特征降维：该方法通过主成分分析 (PCA) 和 Hotelling's T^2 的特征降维方法减少了冗余特征，提高了模型的准确性和效率。在 CIC-IDS2017 数据集上的准确率、精确率和 F1 分数表现优异。

2. BiLSTM+CNN：该方法结合了 BiLSTM 和 CNN 的优势，通过多层次特征融合提升了检测模型的稳定性和效率，同时采用自适应平衡训练方法和注意力损失函数降低了误报率。

3. CNN+RNN：该方法通过双分支特征提取网络，分别使用 CNN 和 RNN 提取数据的空间特征和时间特征，无需人工提取流量特征，有效提高了流量异常检测效率。

4. GAN+RF：该方法通过生成对抗网络 (GAN) 增加数据集中的异常流量样本数量，并采用随机森林 (RF) 进行分类，有效解决了数据不平衡问题，提高了检测性能。

5. HRNN：超图循环神经网络 (HRNN) 将网络流量数据表示为超图结构，通过 K 近邻算法 (KNN) 生成超边，学习流量实体间的高阶关联关系，并结合超图卷积和 RNN 提取空间和时间特征，提高了异常流量检测的准确性。

表 1: 异常流量检测方法总结

Table 1: Summary of abnormal traffic detection methods

检测方法	优点	缺点
BiLSTM+ 特征降维	能够有效去除冗余特征	模型泛化性较差
BiLSTM+CNN	训练速度快、误报率低	对小样本类别的适应性较差
CNN+RNN	检测模型的误报率低	模型泛化性较差
GAN+RF	生成流量数据的效果较好	模型计算和存储开销大
HRNN	检测模型的误报率低	超图构建需要较高计算资源

3 主要研究内容

该研究的主要内容包括：数据收集、Bonito 模型生成问答对与指令微调、奖励模型的设计与微调、强化学习与下游神经网络结合调整目标大模型，通过引入奖励模型和指令微调技术，提升目标大语言模型在网络安全问答生成中的专业性和准确性。

3.1 数据收集

1. 网安文本: 通过安全公司、研究机构和开源社区的恶意代码分析报告。以及其他平台 AlienVault OTX、ThreatCrowd、Malware Information Sharing Platform (MISP) 等。从公开的研究报告、白皮书、博客文章等获取情报数据。

2. 网络爬虫：编写 python 脚本获取 VirusTotal、MalwareBazaar、VirusShare、KDD Cup 99、NSL-KDD、CIC-IDS2017 等数据集。

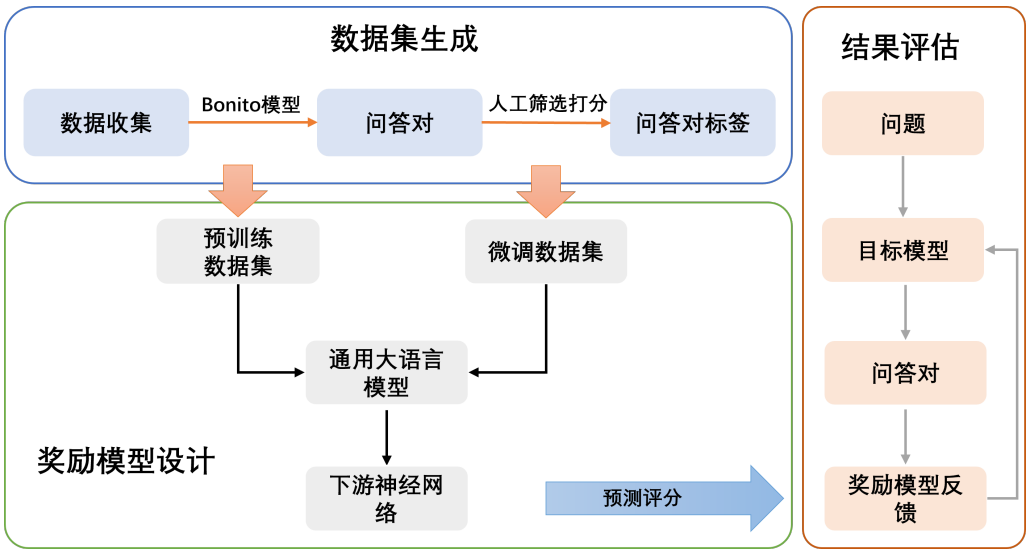

图 3: 研究方案

3.2 Bonito 模型

由于问题是是-否类型，所以使用基于二分类的生成 Bonito 模型来生成一个问答对数据集，包含 SHA256 hash 的样本，并且为每个样本生成与其相关的问答对（例如：提供一个病毒样本的 SHA256 hash，生成相应的“是病毒”或“不是病毒”的问答对）。

3.3 指令微调

将生成的问答对经过人工筛选和打分，即对于每一个样本，人为对 Bonito 模型生成的问答对添加评分标签，确保模型能够准确区分病毒与非病毒。这些比较结果（即评分标签）同时会被用作训练数据，让奖励模型学习预测我们对模型的判断的偏好。而且与微调相比，指令微调需要的数据更少，适应速度更快[5]，指令直接解释需要改进的地方，如果需求发生变化，指令可以迭代，能够通过对话式指导。

3.4 奖励模型

3.4.1 为什么用奖励模型

模型输出可能不符合人类偏好：FT 只是将预训练模型中的知识给引导出来的一种手段，而在 SFT 数据有限的情况下，我们对模型的引导能力就是有限的。这将导致预训练模型中原先错误或有害的知识没能在 SFT 数据中被纠正，从而出现有害性或幻觉[6] 的问题。

3.4.2 奖励模型设计

奖励模型的设计很是关键，因为没有简单的数学或逻辑公式可以切实地定义人类的主观价值。在进行 RLHF 时，需要奖励模型来评估语言大模型（actor model）

回答的是好是坏，这个奖励模型通常比被评估的语言模型小一些（由于模型太大不够稳定，损失值很难收敛且小模型成本较低，因此，RM 模型采用参数量较小的模型）。奖励模型的输入是 prompt+answer 的形式，让模型学会对 prompt+answer 进行打分。

4 研究方案

研究的具体流程共分成三个大部分：即指令集的构造、奖励模型的设计、对目标模型的调整。其中指令集的构造会通过任务指令 +Bonito 模型生成的问答对 + 参数高效微调等步骤完成，奖励模型的设计会通过奖励模型 + 人工反馈 + 参数高效微调等步骤完成，对目标模型的调整会通过奖励模型 + 强化学习与下游神经网络的结合等步骤完成。

4.1 Bonito 模型

由于我们要将问题类型标签与网络安全的数据收集部分（以下简称网安文本）一同送入 Bonito 模型的输入端，所以要对 Bonito 的模型架构稍作调整。

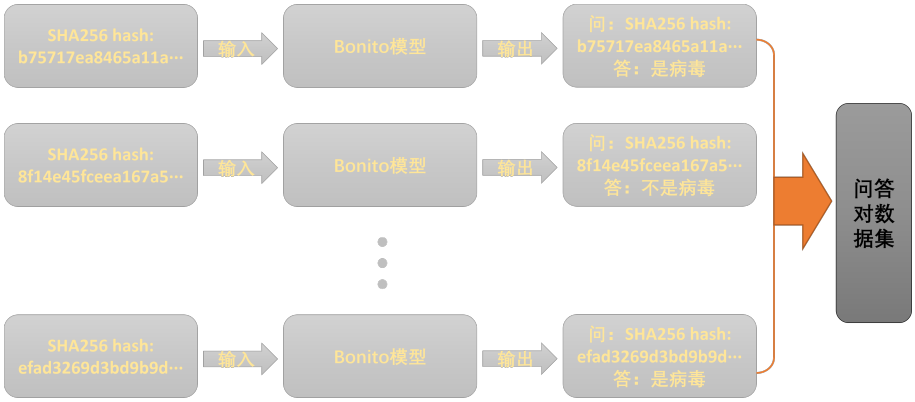

图 4: Bonito 模型生成问答对流程

1. 数据预处理（输入）：由于 Bonito 通常处理数值序列，将网络安全文本和问题类型标签转化为 Bonito 模型可以接受的输入序列，即将文本转换为数值表示：
   - 字符级别的编码（每个字符映射为数字）。
   - 使用词嵌入或预训练的文本编码器（如 Word2Vec、BERT 的嵌入）。
2. 输出格式处理：输出也需要编码为 Bonito 可处理的问答对格式。利用 Bonito 模型从未标注的网络安全文本中生成合成任务数据集。可以通过创建条件任务生成模板（如 “yes-no 问题回答”），使模型从输入的文本中生成相应的任务指令和回答[7]。

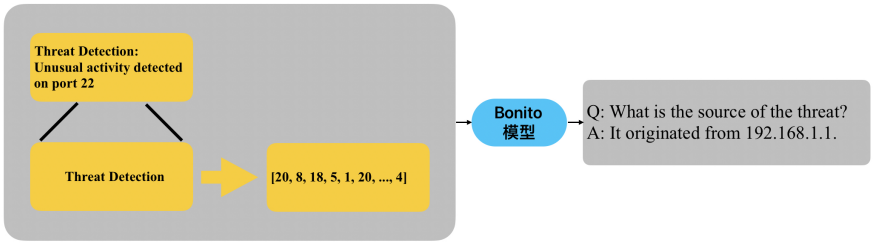

图 5: 数据处理与问答对生成流程

2. 修改 Bonito 模型：由于 Bonito 使用卷积神经网络和 CTC 解码器。为适配文本生成任务，需要以下调整：

    1. 替换输出层：将输出层改为 Seq2Seq 生成任务中常用的 Transformer 解码器或 LSTM 解码器。通过通过自注意力机制能够更好地捕捉序列中的长距离依赖关系，这对于文本生成任务尤为重要，因为文本中往往存在跨越长距离的语义联系。
    2. 增加注意力机制：Transformer 解码器自带的注意力机制可以提升模型的生成能力，通过关注输入序列中与当前输出最相关的部分来生成更准确的文本。
    3. 支持双通道输入：添加一个独立通道处理问题类型标签，并将问题类型标签嵌入与文本嵌入拼接做为输入。

4.2 指令微调奖励模型

通过这些合成的问答对数据集与带评分标签的数据集，对预训练好的大规模语言模型进行有监督的微调，帮助 LLM 拥有更好的推理能力，从而展现出泛化到未见过任务的卓越能力。也就是说，就算微调的指令中没有设计相关的任务，大模型在新任务上的表现也会优于微调之前[7]。指令微调的目标是教模型根据网络安全文本和问题类型标签生成相关的问答对。指令集需要覆盖多种问题类型和场景，保证模型的泛化能力。

1. 参数高效微调（Parameter-Efficient Fine-Tuning）：由于 LoRA[8] 只对自注意力层的权重进行微调，而没有对线性层的权重进行微调，而奖励模型（RM 模型）将 SFT 模型最后一层的 softmax 去掉，即最后一层不用 softmax，改成一个线性层，所以 LoRA 并不适用于该任务。但是譬如 FLAN2022、T5 等大模型通常已经经过大量预训练，因此在指令微调时，通常只需微调部分参数[9]，

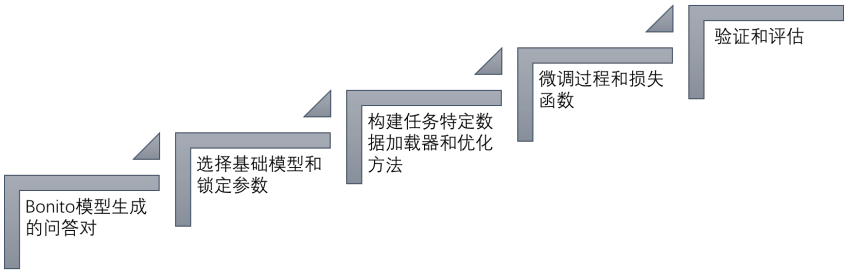

图 6: 指令微调过程

以确保模型保留通用语言能力并提高特定任务表现。也可以使用适配器进行指令微调[10]，从而减少模型修改的复杂性和成本。

2. 构建任务特定数据加载器和优化方法：使用 PyTorch 或 Hugging Face Transformers，可以将是否式问答对任务的数据转换为适合通用大语言模型的输入格式。在优化过程中，使用交叉熵损失函数来计算模型生成的答案和实际答案之间的差异。

   i. AdamW 优化器[10]：AdamW 是适合微调大型语言模型的优化器，可以在训练中减少权重衰减（Weight Decay），有助于模型更好地泛化。

   ii. 学习率调度：使用学习率调度器，如线性下降调度器（Linear Decay Scheduler），在训练过程中动态调整学习率，从而避免在微调时出现过度更新。

3. 微调过程和损失函数设计：在微调过程中，输入模型的每一条数据应包括指令、上下文、问题和答案。微调使用损失函数计算生成的答案与真实答案的差距，并进行反向传播和参数更新。根据任务的流程，可以初步设计一个多层次的损失函数，例如：

   – 正确性损失（Correctness Loss）：衡量模型预测的评价与真实标注的正确性是否一致。

   $$\mathcal{L}_{\text{correctness}} = -\sum_{i} y_{\text{true},i} \log(y_{\text{pred},i}) \tag{1}$$

   其中 i 表示正确性评分的不同类别。

   – 准确性损失（Accuracy Loss）：使用均方误差（Mean Squared Error, MSE）来衡量模型预测的评分与真实标签评分之间的距离。准确性损失可以帮

助模型更精确地对回答质量进行评分。

$$\mathcal{L}_{\text{accuracy}} = \frac{1}{N}\sum_{i=1}^{N}(y_{\text{true, accuracy},i} - y_{\text{pred, accuracy},i})^2 \tag{2}$$

其中 $y_{\text{true, accuracy},i}$ 表示回答的真实准确性评分，$y_{\text{pred, accuracy},i}$ 表示模型预测的准确性评分，N 为样本数量。

- 一致性损失：衡量模型对回答风格或格式的一致性评分与真实标注的匹配程度。如果有关于回答一致性的评分（例如格式和风格是否符合预期），可以将其作为一个二分类任务来处理，使用二元交叉熵损失。

$$\mathcal{L}_{\text{consistency}} = -\left(y_{\text{true}}\log(y_{\text{pred}}) + (1 - y_{\text{true}})\log(1 - y_{\text{pred}})\right) \tag{3}$$

其中 y_{true} 表示真实的一致性标签，y_{pred} 表示模型预测的一致性评分。

最终损失函数:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{\text{correctness}} + \beta \cdot \mathcal{L}_{\text{accuracy}} + \gamma \cdot \mathcal{L}_{\text{consistency}} \tag{4}$$

α、β、γ 可以根据任务的具体需求进行调整。

4.3 奖励模型的设计

此处的通用大模型是用作奖励模型（Reward Model）来调整最终模型，大致的流程涉及上述的指令微调、奖励建模和强化学习（RLHF，Reinforcement Learning with Human Feedback）。奖励模型的反馈与下游神经网络相结合，通过强化学习的方法（如 PPO[11]，Proximal Policy Optimization）来优化目标模型，以提升回答准确性。

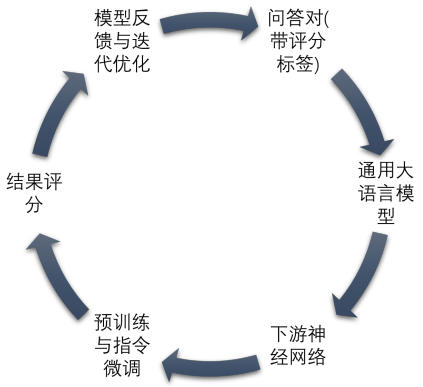

图 7: 奖励模型设计

1. 奖励模型的输入设计[5]：奖励模型的输入有三个：指令集、网安文本、带有评分标签的问答对，指令即为任务目标，为了提高模型的泛化能力，指令集要尽可能的丰富，输入中每一个网安文本对应的带有评分标签的问答对都会做为训练集。其中的文本嵌入过程可以使用 SentencePiece 分词器实现由文本到单词到编码序列到嵌入向量的过程。

2. 奖励模型的输出设计：由于奖励模型的目的是对目标模型的调整，调整方式为接受目标模型的输出的问答对与网安文本，所以奖励模型的输出为对该网安文本提出的问题的回答带上一个评分标签。由于奖励模型的输出要交给下游神经网络继续任务，所以采用最终隐藏状态输出较为合理，其形式为：

   [batch_size, sequence_length, hidden_size]

   ，其中 batch_size 是批次大小，sequence_length 是状态序列长度，hidden_size 是维度。

3. 下游神经网络的设计[12]：

   1. 池化层设计：下游神经网络通过接受奖励模型的最终隐藏状态输出后要将其池化处理将序列数据转为定长做为全连接层的输入。池化层可选择平均池化捕捉序列中的全局信息。

      $$P_{\text{avg}}(i,j) = \frac{1}{K \times K} \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} P_{\text{input}}(i+m, j+n)$$

   2. 全连接层设计：通过多个线性层，将输出层的输出更改为一个做为评分的标量。

4.4 结果评估与闭环反馈

奖励模型的设计通过反复的生成和反馈，逐步提升了目标模型的性能[5]。在结果评估阶段，目标模型会利用奖励模型和人工标注的数据进行测试，以确保其能够有效处理各种类型的问题。通过奖励模型的持续反馈和人类评分之间的对比，进一步调整模型的生成逻辑与评估标准，形成一个完善的反馈回路闭环。

5 进度安排，预期达到的目标

整个项目的进度安排与预期目标如下：

6 课题已具备和所需的条件、经费

1. **已具备的条件：**

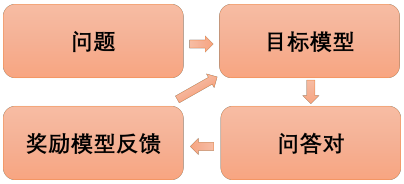

图 8: 奖励模型与目标模型的反馈循环

表 2: 项目任务时间表

Table 2: Timeline of project tasks

任务	时间	预期目标
爬虫开发与文本收集	24 年 11 月-24 年 12 月中旬	编写基本的爬虫代码
数据的预处理	24 年 11 月-24 年 12 月中旬	完成数据清洗和预处理
Bonito 模型的设计	25 年 1 月-25 年 2 月中旬	完成初步问答对生成。
奖励模型设计与实现	25 年 3 月-25 年 4 月	对通用大模型指令微调
结果评估与反馈改进	25 年 4 月-25 年 4 月中旬	根据反馈对目标模型调整

  - **数据收集与预处理能力**：我在暑假期间完成了基本的网络爬虫工具以爬取一定数量的病毒样本，并继续对数据进行了清洗与预处理。
  - **模型开发环境**：目前已经搭建个人模型开发环境：PyTorch 深度学习框架。

2. **所需的条件：更多高质量的数据集**：为了提升问答对的质量，我需要收集和标注更多网络安全病毒样本的高质量数据集，尤其是最新的威胁情报、恶意代码分析报告等。

3. **课题经费**：该课题预计没有经费支出。

7 研究过程中可能遇到的困难和问题，解决的措施

7.1 Bonito 模型数据不平衡

网络安全数据可能存在类别不平衡，即正常样本和恶意样本的数量差异很大，这可能导致模型偏向于预测多数类。

解决方法：

1. SMOTE[13] (合成少数类过采样技术)：可以通过 SMOTE 方法对少数类进行过采样。SMOTE 通过在少数类样本之间生成新的合成样本，而不是简单地重复现有样本，避免了过拟合问题。例如，选择少数类样本的最近邻[14]，然后在样本之间随机生成新的数据点，这样可以扩大少数类数据的表示范围，

从而更好地平衡数据集。由于 SMOTE 对所有少数类样本平等对待，可能会在某些区域生成过多的合成样本，这增加了过拟合的风险。

2. 数据清洗与采样：可以使用欠采样方法[15]，从多数类中随机删除部分样本，或者结合过采样与欠采样的方法，通过清洗掉噪声样本和难以区分的边界样本来平衡数据集。例如，边界样本可以通过最近邻算法来识别，这样有助于去除对模型性能有负面影响的样本 。

7.2 模型过拟合问题

网络安全数据可能包含大量噪声和冗余信息，这可能导致模型过拟合训练数据。并且模型可能在训练集上表现良好，但在未见过的数据上表现不佳。

解决方法：

1. 正则化：在模型训练过程中，可以通过添加正则化项（如 L1 或 L2 正则化）来防止过拟合。正则化项会惩罚模型参数的大小，从而限制模型的复杂度[16]。例如，在神经网络中，可以通过在损失函数中添加 L2 正则化[17] 项来防止过拟合。

2. 加入 Dropout 层[18]。Dropout 通过随机去除部分神经元来减少模型的复杂性，从而降低过拟合的风险。L2 正则化通过在损失函数中添加参数权重的平方惩罚项，限制权重的过大值，这有助于提高模型的泛化能力。

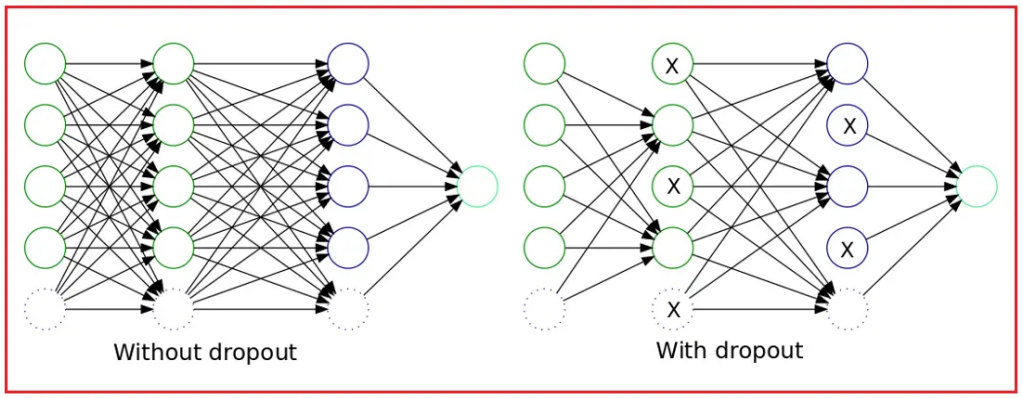

图 9: 应用 Dropout 后的神经网络

3. 交叉验证：利用交叉验证来评估模型的泛化误差。例如，使用 k 折交叉验证[16] 来评估模型在不同数据划分上的表现，以检测过拟合现象。这种方法有助于确保模型不会仅仅学习训练数据，而且加强对未见数据的泛化能力 。

8 主要参考文献

[1] Ziegler D M, Stiennon N, Wu J, et al. Fine-Tuning Language Models from Human Preferences[J/OL]. ArXiv, 2019, abs/1909.08593. https://api.semanticscholar.org/CorpusID:202660943.

[2] Stiennon N, Ouyang L, Wu J, et al. Learning to summarize from human feedback[C] // NIPS '20: Proceedings of the 34th International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc., 2020.

[3] Chen S-S, Hwang R-H, Sun C-Y, et al. Enhancing Cyber Threat Intelligence with Named Entity Recognition Using BERT-CRF[C/OL] // GLOBECOM 2023 - 2023 IEEE Global Communications Conference. 2023: 7532-7537. http://dx.doi.org/10.1109/GLOBECOM54140.2023.10436853.

[4] Yang H, Zhang H, Hu Z, et al. A Review of Network Anomaly Traffic Detection Based on Deep Learning[J/OL]. Journal of Wuhan University (Natural Science Edition), 2024: 1-14. http://dx.doi.org/10.14188/j.1671-8836.2024.0043.

[5] Longpre S, Hou L, Vu T, et al. The Flan Collection: Designing Data and Methods for Effective Instruction Tuning[C/OL] // Krause A, Brunskill E, Cho K, et al. Proceedings of Machine Learning Research, Vol 202: Proceedings of the 40th International Conference on Machine Learning. [S.l.]: PMLR, 2023: 22631-22648. https://proceedings.mlr.press/v202/longpre23a.html.

[6] Ye H, Liu T, Zhang A, et al. Cognitive Mirage: A Review of Hallucinations in Large Language Models[J/OL]. ArXiv, 2023, abs/2309.06794. https://api.semanticscholar.org/CorpusID:261705916.

[7] Nayak N, Nan Y, Trost A, et al. Learning to Generate Instruction Tuning Datasets for Zero-Shot Task Adaptation[C/OL] // Ku L-W, Martins A, Srikumar V. Findings of the Association for Computational Linguistics: ACL 2024. Bangkok, Thailand: Association for Computational Linguistics, 2024: 12585-12611. https://aclanthology.org/2024.findings-acl.748.

[8] Hu E J, Shen Y, Wallis P, et al. LoRA: Low-Rank Adaptation of Large Language Models[J], 2021.

[9] Li X L, Liang P. Prefix-Tuning: Optimizing Continuous Prompts for Generation[J/OL], 2021. https://arxiv.org/abs/2101.00190.

[10] Hu Z, Wang L, Lan Y, et al. LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models[C/OL] // Bouamor H, Pino J, Bali

K. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Singapore: Association for Computational Linguistics, 2023: 5254-5276. https://aclanthology.org/2023.emnlp-main.319.

[11] Wang X, Ma Z, Cao L, et al. A planar tracking strategy based on multiple-interpretable improved PPO algorithm with few-shot technique[J/OL]. Scientific Reports, 2024, 14. https://api.semanticscholar.org/CorpusID:267722253.

[12] Chung H W, Hou L, Longpre S, et al. Scaling Instruction-Finetuned Language Models[J/OL]. ArXiv, 2022, abs/2210.11416. https://api.semanticscholar.org/CorpusID:253018554.

[13] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique[J]. J. Artif. Int. Res., 2002, 16(1): 321–357.

[14] Han H, Wang W Y, Mao B H. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning[J]. Lecture Notes in Computer Science, 2005.

[15] Krawczyk B. Learning from imbalanced data: open challenges and future directions[J/OL]. Progress in Artificial Intelligence, 2016, 5: 221 - 232. https://api.semanticscholar.org/CorpusID:207475120.

[16] Li H, Rajbahadur G K, Lin D, et al. Keeping Deep Learning Models in Check: A History-Based Approach to Mitigate Overfitting[J/OL]. IEEE Access, 2024, 12: 70676-70689. http://dx.doi.org/10.1109/ACCESS.2024.3402543.

[17] Domingos P. A few useful things to know about machine learning[J/OL]. Commun. ACM, 2012, 55(10): 78–87. https://doi.org/10.1145/2347736.2347755.

[18] Cai S, Gao J, Zhang M, et al. Effective and Efficient Dropout for Deep Convolutional Neural Networks[J/OL]. ArXiv, 2019, abs/1904.03392. https://api.semanticscholar.org/CorpusID:102351044.