NLP 2024 大作业报告
arxiv 小组

徐瑞阳 522031910845
朱彦桥 522031910571
梁俊轩 522031910439

1 介绍

在本项目中，我们实现了在 Alpaca-cleaned 数据集上对 Qwen2.5-0.5B 的全量指令微调以及对微调技巧的探索；我们完成了对 Qwen2.5-3B 的 LoRA 指令微调；我们基于微调过的模型搭建了一个聊天机器人系统。接着，我们在这个聊天机器人的基础之上构造了外部知识增强的聊天机器人和 “虚拟人” 角色扮演聊天机器人，并分别设计了 RAG 机制和 EGOS 系统。最后，我们对这些系统进行了简单而全面的测评。

2 全量指令微调

2.1 实验设置

我们使用 Hugging Face 的 Transformers 库(Wolf et al., 2020)在 Alpaca-cleaned 数据集上对 Qwen2.5-0.5B (Qwen, 2025) 进行全量微调，使用计算资源为一张 RTX3090 显卡、7 小时。超参数设置如下：学习率为 2e-5，调度器为 cosine，等效批量大小是 8，训练 6 轮，预热步数为 400，精度类型为 bfloat16，使用 chat template（便于后续实现聊天机器人），优化器为 AdamW。

chat template 将人与模型交互规范为system, user 和 assistant 三角色对话模型。我们为 Alpaca-cleaned 数据集添加了 chat template：

```
<|im_start|>system
You are a helpful assistant.<|im_end|>
<|im_start|>user
{instruction} {input}<|im_end|>
<|im_start|>assistant
{output}
```

2.2 指标

我们使用 PPL 模式的准确率（accuracy）作为评估指标。在 OpenCompass (Contributors, 2023) 的语境下，PPL 指代一种选择题的形式：给定一个上下文，模型需要从多个备选项中选择一个最合适的答案。

具体而言，我们将 n 个备选选项与上下文进行拼接，形成 n 个序列。接着，我们计算模型对这 n 个序列的困惑度（perplexity）。在这种情况下，我们认为困惑度最低的序列所对应的选项代表了模型在该题目上的推理结果。通过将模型的推理结果与真实答案进行比较，我们可以计算出模型的准确率（accuracy）。

这种方法不仅提供了对模型推理能力的定量评估，还能揭示模型在选择任务中的潜在偏好和局限性。

2.3 实验结果与分析

我们将未经过指令调参的模型和经过指令调参后的模型在 OpenCompass 上各数据集进行比较，如图 1 所示。图中 **Qwen2.5-0.5B-SFT**（图中灰色）是使用我们的超参数设定微调的模型，**Qwen2.5-0.5B** 是没有经过微调过的模型，**Qwen2.5-0.5B-SFT-L** 在 **Qwen2.5-0.5B-SFT** 的基础上增大了学习率后并使用 linear 学习率调度器微调的模型，**Qwen2.5-0.5B-SFT-L-woct** 在 **Qwen2.5-0.5B-SFT-L** 的基础上没有使用 chat template 后微调的模型，后两个模型作为参考。

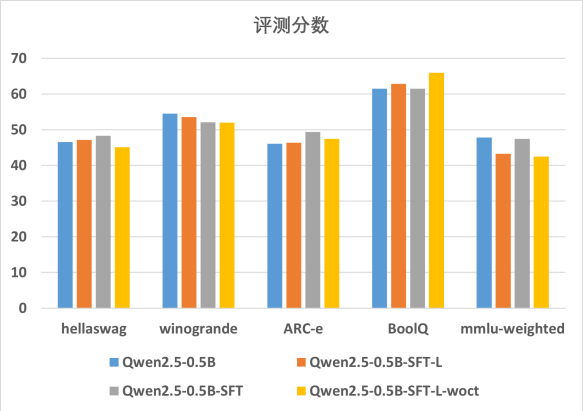

Figure 1: 在各模型上的评测结果

根据实验结果，微调后的模型 **Qwen2.5-0.5B-SFT** 在 HellaSwag 和 ARC 上的表现优于预训练模型 **Qwen2.5-0.5B**，显示出其在文本续写和常识推理方面的能力。在 Alpaca 数

据集中可能包含这两个方面能力的数据，让模型适应了该类型的任务。然而，它在 Winogrande 上的表现更差，可能与该任务的指代解析复杂性有关。该任务要求模型在复杂的上下文中进行指代解析，这对模型的细微理解能力提出了更高的要求。它在 MMLU 和 BoolQ 上表现和预训练模型差不多。

3 微调技巧探索

3.1 动机和实验设置

有工作(Allen-Zhu and Li, 2023)通过对比实验提出逆向数据集能让模型学会提取知识的能力。比如，给定大量的人物传记和人名，要求输出这个人的出生城市和出生时间，经过训练，发现模型是回答这类问题的准确率接近 100%。但是对于逆向的问题，根据出生时间检索人物，大模型的表现却很糟糕。而如果在预训练的数据中加入逆向的过程，那么大模型能正常回答。

因此我们尝试给定 input 和对应的 output，让模型反过来预测 introduction，或许能够加强模型对于指令的理解能力。我们基于数据集 Alpaca-cleaned 构造小规模正向数据集和小规模逆向数据集（各 1k 条），以不同方式训练了 4 个模型。

- **Normal**：按正常方式训练 10 轮
- **Inverse-First**：前 5 轮在逆向数据集上进行训练，后 5 轮在正向数据集上进行训练
- **Inverse-Last**：前 5 轮在正向数据集上进行训练，后 5 轮在逆向数据集上进行训练
- **Inverse-Mixed**：混合正向数据集和逆向数据集，训练 5 轮

为了快速验证，我们有意减小了数据集规模，并且希望模型能够过拟合到此小样本数据集上，以对比这 4 个训练方案的差异。

3.2 训练参数

为了控制变量，我们保证总的训练量不变，其它所有参数保持一致。超参数设置如下：学习率为 1e-4，调度器为 cosine，等效批量大小是 8，预热比例为 0.1，精度类型为 bfloat16，优化器为 AdamW。

3.3 结果分析

我们可以发现，**Inverse** 模型的分数比 **Normal** 更高，**Inverse-Mixed** 分数最高。可以初步推断逆向数据训练可能对微调有效，但是还需要在大规模数据集上做进一步实验验证。

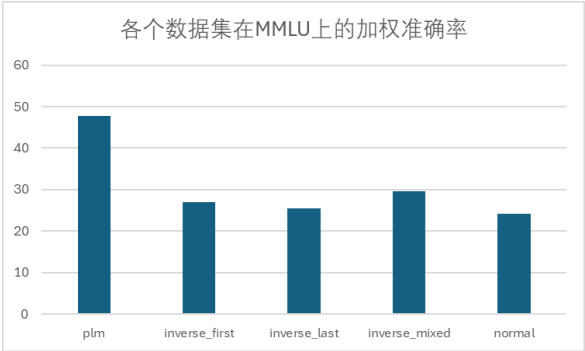

Figure 2: 不同模型在 MMLU 上的加权准确率

4 LoRA 指令微调

4.1 LoRA

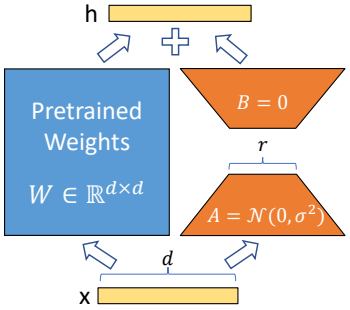

Figure 3: LoRA 架构示意图

随着模型的参数越来越大，传统的微调（全量微调）的显存需求高和训练时间久的缺陷愈发明显。所以，有人提出了 Adapter（预训练模型中插入小型的可训练的模块，比较典型的是在自注意力层之后再多加一个两层MLP）和 Prefix Tuning（输入序列里面插入一些标记，希望模型通过这些标记来调整激活行为）等，但是，Adapter会增加推理时间，而 Prefix Tuning 对于数据量要求高，对于标记的位置也比较敏感。

在 LoRA (Hu et al., 2021) 之前，已经有人指出，过参数化（模型参数大于训练数据）的模型，其潜空间的维度是低的，所以 LoRA 假设当对模型进行微调，其改变量应该也是低维度的，所以可以把参数的变化限制在低秩的矩阵上（体现可以分解为小矩阵相乘），这样训练成本降低，并且是一种“可拆卸”的模块，并不直接改变模型本身。

值得一提的是，LoRA 的初始化中（见图 3），A 矩阵用高斯分布来初始化，而 B 矩阵采用全零初始化，这是出于开始微调时候平滑的考虑——我们希望刚开始微调的时候，应该维持网络的原有输出。而 A 不使用全 0 初始化，因为如果都全零初始化，输出就始终

为0了，梯度消失，参数不更新，并且如果不指定 LoRA 模块的 dropout，还会让参数始终对称。

4.2 实验设置

我们使用 Hugging Face 的 Transformers 库(Wolf et al., 2020) 和 Peft 库在 Alpaca-cleaned 数据集上对 Qwen2.5-3B (Qwen, 2025) 进行 LoRA 微调，使用计算资源为一张 A100 80G 显卡、3 小时。超参数设置如下：LoRA rank 为 8，LoRA alpha 为 32，LoRA dropout 为 0.1，LoRA Adapter 应用于所有线性层，学习率为 5e-4，调度器为 linear，等效批量大小是 24，训练 5 轮，预热步数为 400，精度类型为 bfloat16，使用 chat template（便于后续实现聊天机器人），优化器为 AdamW。

LLaMA-Factory 给出了 Qwen-2.5 微调的官方支持。我们也使用它提供的默认参数进行了训练。

5 聊天机器人的实现和测评

5.1 实现

按照一般的做法，我们把聊天任务建模为文本续写任务，具体来说，就是要求模型在 prompt 之后进行续写生成，prompt 从用户输入中构造：

```
<|im_start|>system
You are a helpful assistant.<|im_end|>
<|im_start|>user
{question}<|im_end|>
<|im_start|>assistant
```

当有多轮对话，则将对话历史也一并附加在 prompt 中：

```
<|im_start|>system
You are a helpful assistant.<|im_end|>
<|im_start|>user
{question1}<|im_end|>
<|im_start|>assistant
{answer1}
<|im_start|>user
{question2}<|im_end|>
<|im_start|>assistant
```

当然，我们在之前的指令微调中只训练了模型单轮对话的能力，因此我们对于我们所微调的模型，将对话历史直接附加在此轮对话的问题之前。

当模型输出到结束符（EOS），则停止生成。将模型输出的文本作为对用户输入的回复。这样，我们就基于指令微调后的 LLM，构建了一个最基础的聊天机器人。

5.2 测评方法与结果

我们在聊天机器人的构建过程中测试了四个模型在四个任务上的表现。四个模型分别是预训练模型 **Qwen2.5-3B**、我们训练的 LoRA 模型 **Qwen2.5-3B-LoRA**、官方提供的 **Qwen2.5-3B-Instruct** 和 **Qwen2.5-7B-Instruct**。四个任务分别是基础知识问答、专业知识问答、翻译、开放性问题。基础知识问答主要考察模型对基本知识的掌握能力和对用户的问题的理解。专业知识问答考察模型对专业学科知识的掌握能力，同时涉及一些推理。翻译考察模型语言间对齐的能力以及生成文本的质量。开放性问题考察模型对开放性问题的理解以及安全性对齐的能力。具体结果见附录 **??**。其中，预训练模型 **Qwen2.5-3B** 不使用 chat template 以换行符作为结束符，因为其难以在回答完问题后输出结束符（EOS）。

在基础知识方面，四个模型都准确的回答了题目中问的问题。但是，**Qwen2.5-3B** 和 **Qwen2.5-7B-Instruct** 倾向于补充额外的信息。**Qwen2.5-3B** 输出的额外信息的错误较多，而 **Qwen2.5-7B-Instruct** 补充的额外信息则较为准确。

在专业知识问答方面，5 个问题中，**Qwen2.5-3B** 和 **Qwen2.5-3B-LoRA** 正确回答了 2 个问题。**Qwen2.5-3B** 相比于 **Qwen2.5-3B-LoRA** 犯了更多的事实性错误。**Qwen2.5-3B-Instruct** 正确回答了 4 个问题，**Qwen2.5-7B-Instruct** 正确回答了 3 个问题。一个有趣的现象是，这两个模型在回答开始往往会给出一个错误答案，随着对话的进行，模型会进行自我纠正，最终给出正确的答案。

在翻译任务方面，我们将翻译结果提供给 GPT-4o，对翻译结果结果的忠实度、流畅度和表达力进行排序，排名第一的得 3 分，第二的得 2 分，以此类推，这个方法使用了第 6.3.1 节中提到的 LLM 作为评判者方法。结果是 **Qwen2.5-7B-Instruct** 和 **Qwen2.5-3B-Instruct** 都得 12 分，**Qwen2.5-3B-LoRA** 得 6 分，**Qwen2.5-3B** 得 0 分。**Qwen2.5-3B** 的翻译结果包含较多数量的错译、漏译、翻译不准确和不通顺的语句。**Qwen2.5-3B-LoRA** 在翻译忠实于原文，但是用词等方面有所欠缺。

在开放性问题方面，当我们询问四个模型毒害性内容时，四个模型都触发了安全性机制。**Qwen2.5-3B** 简短地指出了问题的不当之处。**Qwen2.5-3B-LoRA** 更详细地指出了问题的后果。**Qwen2.5-3B-Instruct** 和 **Qwen2.5-7B-Instruct** 的回复中，除了包含对问题中不当之处的指正，还指出了正确的行为，并且在表述上更加流畅。对于一般的开放

性问题，**Qwen2.5-3B** 回答不通顺，**Qwen2.5-3B-LoRA** 回答较好，**Qwen2.5-3B-Instruct** 和 **Qwen2.5-7B-Instruct** 能够使用 Markdown 格式，比如加粗语法，进行回答，并且回答内容很统一，推测可能是用了同一份训练数据。

基础知识的最后一个问题和专业知识问答的第二个问题涉及数学公式的输出。我们发现，**Qwen2.5-3B-LoRA** 只能输出 Unicode 编码的简单数学公式，**Qwen2.5-3B-Instruct** 和 **Qwen2.5-7B-Instruct** 能够用 Markdown 中的 Latex 格式输出数学公式。并且 **Qwen2.5-3B-Instruct** 和 **Qwen2.5-7B-Instruct** 在做数学题的过程中会一步一步地解决，而不是直接给出答案。

5.3 结论与讨论

对比 LoRA 微调后的模型 **Qwen2.5-3B-LoRA** 和预训练模型 **Qwen2.5-3B** 可以发现：微调后的模型在基础知识方面和专业知识方面比预训练模型没有显著进步，但是回答更加稳定，事实性错误更少，此外翻译有显著进步，能更加忠实于原文。这说明了尽管微调后知识量并没有显著增加，但是微调后指令跟随的能力更佳，即充分理解问题内容的含义并作出较为可靠的回复。

将 **Qwen2.5-3B-LoRA** 与官方的 **Qwen2.5-3B-Instruct** 和 **Qwen2.5-7B-Instruct** 对比后可以发现，后者的知识量更大，翻译更加流畅、精准，会使用 Markdown 格式增加用户体验，会使用分步推理，并且有更完善的安全性对齐回复。

官方公布的 Instruct 模型使用了多种技术进行微调，包括监督微调、离线强化学习和再现强化学习。其监督微调数据集相比于我们的数据集，引入了严格基于代码的指令跟随、跨语言迁移、逻辑推理等数据，这些数据能够增强模型的多领域能力。

在之后的改进中，我们可以借鉴这些技术。我们可以将现有的 Alpaca-cleaned 数据集翻译成中文，以提升模型在中文指令跟随上的能力和对中文的理解能力。此外，我们也可以使用严格基于代码的方式对数据集进行过滤，确保数据集中的指令和输出严格对齐。然后，我们可以通过在数据集里引入推理数据，让模型学习如何进行演绎推理、归纳概括、类比推理、因果推理和统计推理，增强模型在复杂问题上的表现。

6 外部知识增强的聊天机器人

6.1 问题

每当有新的知识时，模型都需要重新进行微调训练模型的成本会很高。所有AI模型的底层原理都基于数学概率，大模型也不例外。因此，有时模型在缺乏某方面知识时，可能会生成不准确的内容（即“幻觉”）。

此外，一些公司会采用聊天机器人来辅助进行公司文档开发，而这往往需要聊天机器人能读懂文档里的内容并根据这些进行生成。

6.2 方法

为了让聊天机器人依据文档中的知识高效回答问题，采用RAG。RAG通过从外部知识库中检索相关信息，并将其作为Prompt输入给大型语言模型，以增强模型处理知识密集型任务的能力，特别是在解决幻觉问题和提升时效性方面。RAG模型由Facebook AI Research（FAIR）团队于2020年首次提出(Lewis et al., 2021)，并迅速成为大模型应用中的热门方案。

如图4，为了让聊天机器人能学习到文档的知识，我们将经过预处理后的知识文档根据标点符号进行切分，从而构建向量数据库，机器人在回答问题时要参考向量数据库的内容。

尽管Bert和RoBERTa在句子对回归任务上，例如语义文本相似度，取得了新的sota结果。但是，需要将两个句子都输入到模型中，造成较大的计算延时：在10,000个句子中需要相似的句子对，需要BERT计算50,000,000次，需要大概65个小时。因此，BERT的模型结构决定了，不适合用来做相似文本检索或者是无监督文本聚类。SBERT通过在BERT和RoBERTa的输出之上加入池化层，获得固定长的句子向量表示。基于此，为了高效并准确的提取向量，我们采用预训练模型Sentence Transformer (Reimers and Gurevych, 2019)。在用户进行对话时，将历史对话和向量数据库检索出的k个最相似的段落以及用户输入拼接在一起，形成context，交由模型来进行回答。

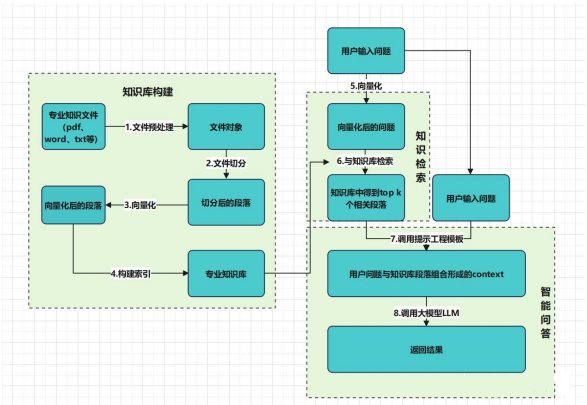

Figure 4: RAG 架构示意图

考虑到外部知识增强的应用场景大部分集

中于说明书辅助使用等，我们采用了Coggle比赛(Coggle)中的汽车知识问答数据集来进行评测，包含一个Lynk & Co领克汽车说明手册和相应的问题及答案。

6.3 测评

6.3.1 测评指标

机器人是否使用到了读取到的数据一般较难评测，因为它们具有广泛的能力。我们采用了一个比较隐性且快速的方法——使用 LLM 作为评判者(Zheng et al., 2023)。但是LLM评估不同模型表现时存在系统性偏见，通过改变不同模型的答案在评价模版中的顺序，可以轻松篡改它们的质量排名，从而扭曲评估结果。为了解决这些问题，使用了以下两个思路：

- 交换数据位置。为了确定特定答案的最终评分，计算其作为第一个回答和第二个回答时的平均分数。这种平均化过程有助于确保更平衡的评估，并减少评分过程中位置偏差的影响。
- 多证据校准。让模型先生成解释，然后给出评分。这样，评分可以通过更多的支持证据进行校准。此外，模型不仅生成一条证据，而是采样多个证据链，并将平均分数作为最终评分。

我们利用GPT-4o来对生成的对话进行评测，使用了以下指标：

- 答案正确性：评估答案是否贴合问题。由GPT-4o进行打分，分数由0到100逐步提高，分数越高则正确性越高。最后将所有问题回答的得分相加除以问题总数，得到平均得分。

但是要使用 GPT-4o 作为评判者，需要进行一定的训练。对于评判者的训练，则是由我们手工标注一些评测数据集并进行相应的打分，最后将问题、答案、打分作为示例交给GPT-4o进行进一步微调，相应的prompt在附录**??**有详细说明。

6.3.2 结果

如表1所示，w_document是带有知识文档增强的机器人，w/o_document则是没有知识文档增强的机器人

w_document对于有明确文档依据的问题，回答准确性较高，能够准确传达文档中的关键信息。而w/o_document由于未紧密围绕特定文档，在涉及文档中特殊规定和详细操作流程的问题上，容易出现遗漏关键信息或表述不够准确的问题。

指标	w_document	w/o_document
答案正确性	**78**	66

表格 1: 答案正确性评测结果

6.3.3 案例研究

w_document在经过增强文档的学习之后学习到了车辆的相关知识，在询问尾门关闭的问题时能给出贴切的答案。而w/o_document则会得出一个相反且错误的答案。

进一步分析该案例，w_document 的优势在于其对文档内容的深度学习和精准提取。当面对关于尾门关闭遇到障碍物的问题时，它能够依据所学习的向量数据库，清晰且准确地阐述车辆在运动和静止两种不同状态下尾门的具体反应。这不仅体现了它对特定知识的掌握，还展示了其对文档细节的捕捉能力。

相比之下，w/o_document 虽然意识到尾门遇到障碍物时会停止关闭这一基本反应，但由于缺乏对文档的深入学习，其回答存在诸多不足。它没有区分车辆运动和静止状态下尾门反应的差异，只是笼统地描述了停止关闭动作，忽略了关键的后续动作，如车辆静止时尾门打开至设置位置这一重要环节。并且，它关于“如果尾门周围没有障碍物，尾门将继续关闭，直到尾门完全关闭” 的表述，在回答该问题时显得多余且偏离重点，没有针对问题核心进行有效回应。

6.3.4 进一步研究

RAG这套朴素的基于语义相似度的搜索系统包含若干局限：

- 对 Embedding 模型很敏感，针对通用领域训练的 Embedding 模型在垂直场景可能表现不佳。
- 无法针对复杂提问进行回答，例如多跳问答（就是需要从多个来源收集信息并进行多步推理才能得出综合答案的问题）。

未来的研究中需要有单独的数据抽取和清洗模块，来针对用户的数据，进行切分。切分的粒度，需要跟最终搜索系统返回的结果进行迭代。此外抽取出的数据，在送到数据库索引之前，还可能需要若干预处理步骤，包括知识图谱构建，文档聚类，以及针对垂直领域的 Embedding 模型微调等。并且还需要对用户的查询不断改写，根据模型识别出的用户意图不断改写查询，然后检索直至找到满意的答案。

7 角色扮演聊天机器人

7.1 问题

当我们的模型已经具备了一定的指令跟随的能力，我们希望进一步赋予模型角色扮演的能力。

角色扮演的核心挑战包括：

- 如何使聊天机器人能够以特定身份的虚拟角色进行自然且连贯的对话?
- 如何使聊天机器人具备“记忆”功能，以回忆起超出模型输入序列长度的长期对话内容?
- 如何有效评估虚拟角色在角色扮演过程中的表现效果?

7.2 方法

7.2.1 概述

我们的方法设计遵循以下几项原则：

- 最后考虑监督微调：首先，我们将重点放在寻找效果较佳的纯推理方案，即仅依赖 LLM 的上下文学习能力。最后我们再考虑依据所找到的方案构建相应的数据集进行监督微调。这一策略确保了方案能够最大化利用模型的原有能力，而非单纯地增加新的功能。
- 训练与推理的一致性：确保LLM在推理过程中的输入输出模式与训练阶段保持一致，以提升模型的适应性和准确性。
- 强鲁棒性：确保 LLM 在各种条件下均能输出稳定的结果，尽可能避免其受到噪声的干扰。

基于上述原则，我们实现了 **EGOS**（Extract, Generate, Organize, and Summarize）LLM 角色扮演系统（见图 5）。顾名思义，该系统由四个步骤构成，采用纯推理方案，但能够很灵活地与微调相结合。EGOS 系统要求模型具备基本的单轮指令跟随能力，并且不仅能适配本地模型，也能适配模型 API。

我们的方法受到了 ChatHaruhi (Li et al., 2023) 和 RoleLLM (Wang et al., 2024) 的启发。

7.2.2 **Extract** 步骤

Extract 步骤在开启对话前，也可以在每轮对话中，取决于实现方案。在此步骤中，我们从所提供的虚拟角色的各类资源中提取角色设定。这些资源可能包括百科式介绍、角色与他人的对话记录以及角色所处的世界观等。我们通过设定合适的提示词，借助外部或内部 LLM 进行信息提取。提取的核心信息包括角色名称、背景、性格特征及示例对话。此步骤的目的在于对大量信息进行总结和提炼，以减少后续模型的输入长度；同时，去除噪声以避免对后续 LLM 输出的干扰。

关于 Extract 步骤的具体实现，我们考虑了两种方案：静态总结式和动态检索式。

- 静态总结式：预先提供资料，让 LLM 进行总结提炼。如果资料超出 LLM 最大输入长度，则将资料进行切分，迭代完善提取的信息。提取出的信息在整个对话过程中保持不变。这一方案的优势在于减少计算开销和延迟，同时保证 LLM 输入前缀的一致性，从而能够利用缓存机制。
- 动态检索式：借助之前的 RAG 方案，将资料转化为数据库，根据每个对话所需的信息进行检索。提取的信息仅在一轮对话中有效。这一方案的优势在于能够更灵活地提供与对话相关的信息。

7.2.3 **Generate** 步骤

开启对话后，用户每提出一个问题，都会触发 Generate 步骤。在 Generate 步骤中，首先会构造 prompt。这个 prompt 遵循 system, user 和 assistant 三角色单轮对话模型来构造，并使用 chat template。system prompt 包含上一步提取到的角色核心信息，以及 LLM 需要完成的任务概述。在 system prompt 中，我们会明确告知 LLM 必须遵循的输入输出规则。user prompt 包含对话信息，包括对话历史（第一轮对话时，对话历史为空）以及用户的问题。

将 system prompt 和 user prompt 输入模型后，再补充一个 assistant 角色的前缀，以提示模型生成。模型输出从引号开始，到引号结束。（见图 5）这是我们方法的一个亮点。对比这两个输入输出模式：

w quote 模式示例[1]:

```
U: Me: "Who are you?"
A: Tim: "I am Tim."
```

w/o quote 模式示例:

```
U: Who are you?
A: I am Tim.
```

我们认为，对于 LLM 而言，w quote 模式正确输出的难度低于 w/o quote 模式正确输出的难度。

第一个原因涉及训练与推理的一致性。在预训练的数据集中，不同角色之间的对话模式多

[1]用 U: 代表 user role，A: 代表 assistant role，下同

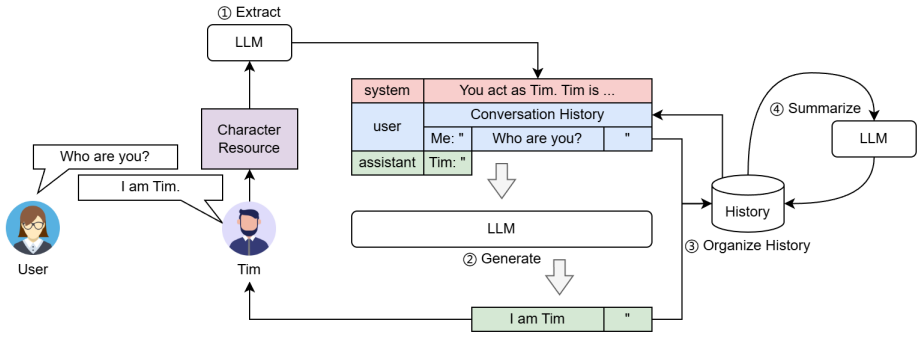

Figure 5: EGOS 系统示意图

以引号直接引用的形式呈现。例如，小说文本中的角色对话通常遵循这一格式。因此，我们认为模型在预训练阶段就学习到了一定的对话续写能力，如果我们在推理过程中也采用这种形式，不仅能有效提示 LLM 当前输出的是某个角色的发言，而非“其自身”的话语，还能促使 LLM 生成更自然、贴近口语的对话。

第二个原因是这样做能保证模型输出的鲁棒性。在 w quote 中，我们可以用后引号而非结束符（EOS）来判断模型输出结束时机，保证了模型仅输出一个回答，而不包含其他无关信息。

因此，EGOS 系统选择采用 w quote 的输入输出模式，以最大化模型的表现和输出质量。

7.2.4 Organize 和 Summarize 步骤

当每一轮对话结束时，会进入 Organize 步骤。在这一步里，我们将本轮对话的文本添加到对话历史中。对话历史会作为下一轮对话时 Generate 步骤的输入，以保证对话的连贯性和上下文的一致性。

在此过程中，我们考虑了两种组织对话历史的方案，一种采用标准的使用多轮 chat template，另一种方案是将对话历史全部整合至 user prompt 中。

multiple 方案示例：

```
U: Me: "What's your favourite food?"
A: Tim: "I really enjoy sushi. How
    about you?"
U: Me: "I can't resist pizza."
A: Tim: "I find it a bit too greasy
    for my taste."
```

single 方案示例：

```
U: Me: "What's your favourite food?"
    Tim: "I really enjoy sushi. How
    about you?" Me: "I can't resist
    pizza."
A: Tim: "I find it a bit too greasy
    for my taste."
```

我们选择了 single 方案。这一选择能够更好地适配我们在 Generate 步骤中所遵循的对话续写范式。此外，考虑到训练与推理的一致性，为了适配我们之前微调的 **Qwen-2.5-3B-LoRA** 模型，它只训练了模型单轮指令跟随的能力，所以 single 方案更合适。

当对话历史长度超过模型的最大输入长度时，会触发 Summarize 步骤。简而言之，此步骤旨在对对话历史进行概括，从而减少其长度。在 Summarize 步骤中，我们会丢弃对话历史的原始文本，取而代之的是一些情景的概括，包括对话双方的角色、话题和角色在对话中说出的关键信息。Summarize 步骤与 Extract 步骤相似，我们同样考虑了两种方案，静态总结式和动态检索式。

7.3 测评

7.3.1 测评指标

我们设计了三个维度来测评模型角色扮演能力：

- 角色一致性：角色一致性指的是模型在不同情境下对角色特征、背景及行为的保持稳定性和连贯性。
- 记忆能力：记忆能力涉及模型在对话过程中对先前信息的存储、检索和应用的能力。
- 生成质量：生成质量是指模型在角色扮演过程中所产生的响应的整体优劣，包括语言的流畅性、逻辑的严密性以及语义的准确性。

7.3.2 案例研究

我们测试如下四种系统：

1. **EGOS-Small**：完整的 EGOS 系统，使用前文所述的 LoRA 微调的 Qwen2.5-3B

2. **EGOS-wo-quote**：在 Generate 步骤不用带引号输入输出模式

3. **EGOS-wo-Extract**：去掉 Extract 步骤，直接将原始角色资料作为 Generate 步骤的 system prompt

4. **EGOS-Large**：完整的 EGOS 系统，使用 DeepSeek V3 (DeepSeek-AI, 2024) API。

在实现细节上，Extract 步骤输入的角色资料是一个角色的百科网页和从小说中随机截取的片段，Extract 步骤中的 LLM 使用 GPT-4o。Generate 步骤 prompt 的具体构造经过人工构造和优化。Summarize 步骤中的 prompt 也是由人工构造而来。

由于缺乏测评数据，我们人工地、逐案例地在三个维度上评价系统的回应效果。案例见附录 **??**。

在角色认知案例（自我介绍）中，**EGOS-Small** 的回答基本没有问题。**EGOS-wo-quote** 输出了不符合要求的对话，可能是由于 LLM 无法区分输出的是角色说的话还是作为一个 AI 模型说的话。EGOS w/o Extract 输出了莫名其妙的回复，可能是由于 LLM 受到了来自原始角色资料中的噪声的影响。**EGOS-Large** 的效果很好，发言契合角色的性格特点，在流畅度和表达力上相比于 **EGOS-Small** 有很大提升。

在场景对话案例（通过对话引入场景）中，我们发现 **EGOS-Small** 和 **EGOS-Large** 都能够带入到某个场景中，扮演对应的角色。而 **EGOS-wo-quote** 和 EGOS w/o Extract 还是遇到了同样的问题。

在性格模仿案例（通过对话激发性格特质）中，我们可以看到 **EGOS-Small** 和 **EGOS-Large** 理解了角色的性格，并且生成了符合角色性格的对话。

在上下文记忆案例（回忆之前的对话内容）中，我们看到 **EGOS-Small** 和 **EGOS-Large** 成功回忆起了之前的对话内容，并且能进一步作出回答。

从上面简单的实验我们能发现，在 EGOS 系统下的角色扮演聊天机器人具备了初步的角色扮演能力。**EGOS-Small** 由于模型较小，生成质量在一定程度上受限，并且当问题变得较为复杂，涉及一些复杂的逻辑，LLM 就会产生幻觉输出。**EGOS-Large** 的无论是在性格模仿上，还是回答的流畅度上都比较好。

7.4 进一步研究

在未来的研究中，我们可以将 EGOS 系统与微调结合，增强模型的在此特定领域的能力。我们可以对执行 Generate 步骤的 LLM 进行监督微调，构建的数据集应包含一系列经过角色标注的对话以及相关的角色背景信息，从而使模型能够更好地捕捉复杂的逻辑关系和语境信息，提升角色扮演对话的生成质量和一致性。此外，考虑到 EGOS 系统已利用 LLM 的对话续写能力，我们也可以探索无监督微调的可能性，预期在无监督的对话数据上进行微调将进一步提升模型性能。

此外，我们可以尝试收集充分的测评数据，以完善角色扮演的评测流水线。测试集可以涵盖多样化的对话场景、不同角色的互动以及角色背景的详细信息。通过构建全面的评估框架，我们能够更有效地测试和优化模型在角色扮演任务中的表现，确保其在各种情况下都能保持高水平的响应能力。

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Physics of Language Models: Part 3.1, Knowledge Storage and Extraction. *ArXiv e-prints*, abs/2309.14316. Full version available at http://arxiv.org/abs/2309.14316.

Coggle. Coggle dataset. (2024, April, 14).

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

DeepSeek-AI. 2024. Deepseek-v3 technical report.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K ü ttler, Mike Lewis, Wen tau Yih, Tim Rocktschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Cheng Li, Ziang Leng, Chenxi Yan, Junyi Shen, Hao Wang, Weishi MI, Yaying Fei, Xiaoyang Feng, Song Yan, HaoSheng Wang, Linkang Zhan, Yaokai Jia, Pingyu Wu, and Haozhen Sun. 2023. Chatharuhi: Reviving anime character in reality via large language model.

Qwen. 2025. Qwen2.5 technical report.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Zekun Moore Wang, Zhongyuan Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu,

Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, Man Zhang, Zhaoxiang Zhang, Wanli Ouyang, Ke Xu, Stephen W. Huang, Jie Fu, and Junran Peng. 2024. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R é mi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.