

分类号_____

U D C_____

编 号_____

密 级_____



南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

本科生毕业设计（论文）

题 目： 以生成式 AI 为基础的多人
协作交互系统搭建

姓 名： 胡 威

学 号： 11912210

系 别： 计算机科学与工程系

专 业： 计算机科学与技术

指导教师： 宋 轩

2023 年 6 月 2 日

诚信承诺书

1. 本人郑重承诺所呈交的毕业设计（论文），是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。

2. 除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。

3. 本人承诺在毕业论文（设计）选题和研究内容过程中没有抄袭他人研究成果和伪造相关数据等行为。

4. 在毕业论文（设计）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

作者签名：胡威

2023 年 6 月 2 日

以生成式 AI 为基础的多人协作交互系统搭建

胡威

学号：11912210

(计算机科学与工程系 计算机科学与技术专业 指导教师：宋轩)

[摘要]：生成式 AI 在近几年得到了重大突破。各大互联网巨头也推出了多种搭载了生成式 AI 的产品。但是目前市面上的“生成式 AI”相关产品通常都仅支持一对一式的人机交互，而难以进行多用户协作式的人机交互。因此笔者以目前已有的生成式 AI 为基础，选用合适的生成式 AI 模型搭建多人协作交互系统。填补该领域的空白，同时对该领域需要使用到的各种技术方法进行研究讨论。笔者利用 Unity 开发引擎成功搭建了一个以 chatGPT 和 Dalle2 为基础的多人协作交互的故事生成体验系统，在该系统能够针对多个用户的输入实时的生成故事。为支持该系统的正常运行，笔者为其设计了一套独特的系统工作流程。同时笔者为系统设计并搭载了“裁判系统”、“自我反省机制”等功能，从而使得系统能够更好的完成故事生成的任务。

[关键词]：生成式 AI; 多人协作; ChatGPT; 系统开发; 大型语言模型

[ABSTRACT]: Generative AI has made significant breakthroughs in recent years. Major internet giants have also launched various products equipped with generative AI. However, currently, the "generative AI" related products on the market usually only support one-on-one human-machine interaction, making it difficult to achieve multi-user collaborative human-machine interaction. Therefore, this study aims to build a multi-user collaborative interaction system based on the existing generative AI and select appropriate generative AI models. Fill in the gaps in this field and conduct research and discussion on various technical methods that need to be used in this field. This study successfully built a story generation experience system based on chatGPT and Dalle2 using the Unity development engine, which can generate stories in real-time based on the input of multiple users. To support the normal operation of the system, the author has designed a unique system workflow. At the same time, the author designed and equipped the system with functions such as "referee system" and "self reflection mechanism", so that the system can better complete the task of story generation.

[Keywords]: Generative AI; Multi-user Collaboration; ChatGPT; System Development; Large Language Model

目录

1. 项目介绍.....	1
1.1 选题背景.....	1
1.2 研究现状.....	1
1.3 选题内容及意义.....	2
2. 基础故事生成系统搭建.....	3
2.1 系统简介.....	3
2.2 系统工作流程框架.....	3
2.3 技术内容介绍.....	5
2.3.1 用户与 chatGPT 以及 DallE2 的信息交流.....	5
2.3.2 多用户的信息整合.....	6
2.3.3 故事结束的控制指令.....	7
3. 模板类故事生成系统搭建.....	7
3.1 系统简介.....	7
3.2 技术内容介绍.....	8
3.2.1 故事模板设置.....	8
3.2.2 裁判系统.....	9
4. 自我反省机制设置.....	13
4.1 角色情节遗漏现象.....	13
4.1.1 现象描述.....	13
4.1.2 现象成因推理分析.....	13
4.2 自我反省机制引入.....	13
4.2.1 自我反省机制简介.....	13

4. 2. 2 引入自我反省机制后的故事生成效果.....	15
5. 对各个模式效果的评估.....	17
6. 结论.....	19
参考文献.....	20
附录.....	22
致谢.....	30

1. 项目介绍

1.1 选题背景

人工智能从出生起便一直是计算机的热门领域，而最近几年 AI 发展的势头越来越猛，“阿尔法狗”等一系列功能强大的 AI 也一度引起了社会各界的广泛讨论。而在刚刚过去的 2023 年里，“生成式 AI”这一领域逐渐走入了人们的视野。不同于过去的“分析式 AI”（Analytical AI），这类人工智能模型不仅可以通过对数据的学习来提炼信息、预测趋势，而且可以生成不同于学习样本的新内容。它们不仅为我们的生活带来了更多的便利性和多样性，也为生产力的提升带来了很大的想象空间^[1]。“数字化数据的指数级增长和突破性的计算能力（摩尔定律）使人工智能产生了革命性的影响。现代信息技术和由人工智能（AI）驱动的认知机器的出现有力地改变了人们的生活和工作。虽然人工智能可以提高某些行业的生产力，但在某些情况下，它可以接管人类的工作并在某种程度上彻底改变职业。”^[2]目前生成式 AI 将会成为提高包括教育业、服务业、文娱业在内的诸多产业生产力的有力工具。

1.2 研究现状

目前生成式 AI 的飞速发展是多个方面的突破共同促成的。首先是算法层面：从根本上看，生成算法的本质就是对训练样本的分布状况进行建模，然后根据模型来抽取新的样本^[3]。现在的生成式算法有很多，比较流行的有五大类模型：自回归模型、生成式对抗模型、变分自编码模型、流模型和扩散模型^[4]。这些模型作用的方式不尽相同，也拥有各自的优势区间。除训练方法之外，新的训练架构的出现也是促成生成式 AI 迅速发展的一大原因：2017 年，谷歌团队提出了 Transformer 架构^[4]。该架构的出现可以看作训练架构发展的一大标志性事件。虽然 Transformer 最早只是应用于 NLP 模型，但由于这种架构的高效性，因此它的应用很快就扩展到了语言、语音、图像和视觉、时间序列分析等众多领域。尤其是在生成式 AI 领域，Transformer 的应用尤其广泛^[1]。

目前国外的各大互联网巨头都将“生成式 AI”相关产品的开发作为本公司或旗下子公司的重点研发项目，伴随着各大公司的大力支持，目前已有多款生成式 AI 的产品问世，比如：openAI 公司的“GPT-3”、“chatGPT”、“Dall-E2”、“CodeX”；谷歌公司的“IMAGEN”；MetaAI 公司的“OPT”、“Make-A-Video”等。这些产品许多都已经正式投入使用，并且为人们的生活提供了便利性。同时，国内互联网巨

头也在同步研发国产的生成式 AI，比如百度推出的生成式对话产品“文心一言”：它是知识增强的大语言模型，基于文心知识增强大模型，具备跨模态、跨语言的深度语义理解与生成能力^[5]。再比如华为旗下的盘古系列 AI 大模型：盘古 NLP 大模型可用于内容生成、内容理解等方面，并首次使用 Encoder-Decoder 架构，兼顾 NLP 大模型的理解能力和生成能力，保证了模型在不同系统中的嵌入灵活性^[6]。在各类生成式 AI 产品中，目前讨论热度最高的是 chatGPT，它已经能够帮助人们完成诸如文本写作、计划编制甚至编程^[1]等种种任务。学术界也对其评价颇高：“当前的 chatGPT 现象标志着人工智能的新时代从算法智能转向语言智能，在语言智能中，实际与人工、真实与虚拟、人与机器之间的互动活动在在线和实时中发挥着积极而重要的作用^[7]。”

但是目前市面上的“生成式 AI”相关产品通常都仅支持一对一式的人机交互，以 chatGPT 为例：chatGPT 是一种大型语言模型（LLM）^[8]，经过训练后它可以预测以输入为条件的自然语言文本。ChatGPT 的基础模型框架是 GPT3.5，它在 GPT3.5 的基础上使用了基于人类反馈的强化学习的版本指令微调，牺牲了几乎所有的上下文学习的能力来换取建模对话历史的能力^[9]。在大数据、大模型和大算力的工程性结合下，它不仅能够学习和理解人类的语言，与人类进行对话，而且能够生成各种“创造性”内容，承担原本由人承担的大量工作^[10]。但是目前不论是直接在 openAI 官网使用 chatGPT 聊天机器人还是使用 chatGPT 官方 api，用户只能自己一个人与 chatGPT 进行对话。目前已有的产品无法支持多名用户同时和 chatGPT 进行对话，并让 chatGPT 综合各个用户的对话内容后给予统一回复的功能。

1.3 选题内容及意义

针对上述问题，本论文以目前已有的生成式 AI 为基础，选用合适的生成式 AI 模型搭建多人协作交互系统。该系统自带联网功能，多名用户通过蓝牙、局域网或广域网等方式进行链接，共同输入信息。同时研究算法，使得系统收集到各个用户的信息后能够根据自身的功能对多名用户的输入进行处理、整合，并将整合完成的信息输入生成式 AI 中，接收到生成式 AI 的输出后对应的发送至各个用户手中。从思路构建、代码实现到最终的系统搭建，一体化的实现对生成式 AI 在多人协作领域的探索。

该项技术对于生成式 AI 在多个领域的应用均有重要应用价值：比如在教育领域，教师可以将该技术与文本类生成式 AI（比如 chatGPT）应用相结合，运用到针对学生的课堂答疑上，让 AI 能够综合处理课堂上所有学生的问题，提高课题互动性^[11]。再比如在服务业，应用该项技术后的生成式 AI 能够综合多位用户的想法和思路，为用

户的出行、娱乐等活动提供优质的智能推荐服务。

2. 基础故事生成系统搭建

2.1 系统简介

作为主要成果，笔者成功搭建了一个以 chatGPT 和 Dalle2 为基础的多人协作交互的故事生成体验系统。经过笔者的设计，用户可以在系统中自定义故事的开头和一些具体设定，并为自己设定一个角色，同时系统会利用 Dalle2，按照用户输入的角色设定生成一张该角色的图片进行展示。上述内容设计完毕后系统会通过 chatGPT 来按照目前用户设定的内容展开后续的故事，并在关键的剧情节点生成一系列的选择。随后所有的用户都可以输入自己所扮演的角色在当前节点做出的选择，系统会将角色的行动整合、提交给 chatGPT 并让其综合各个用户的选择决定下一步的剧情走向。

该系统的搭建全部在 Unity 开发引擎上实现。Unity 是一款整合开发技术、横跨桌面与移动平台的专业开发引擎，通过使用 Unity 开发引擎，用户可大幅提高产品品质与开发效率^[12]。Unity 还可以为 Windows、Mac Os、Linux、iOS、Andrid 等各种主流桌面及移动设备提供开发环境，并且能够通过 Unity web player 插件发布网页项目，支持 Chrome 和 Sarari 等主流浏览器的网络浏览^[13,14]。这些方面的优势能够提高该系统的开发速度，同时也能为该系统在更多平台上的使用提供技术支持。

2.2 系统工作流程框架

为实现多人输入的信息的整合处理以及基础的内容生成和展示，笔者设计了下列的系统工作流程框架：

房间准备阶段系统工作流程图可见图 1。首先一名用户作为房主需要确定一个真实有效的 ip 地址进行聊天室房间的创建，当聊天室房间创建完毕后,其余用户便可以根据房间 ip 地址加入聊天室房间。

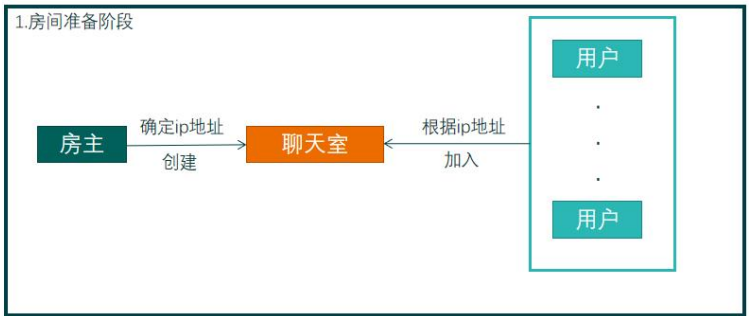


图 1 房间准备阶段流程图

进入聊天室后，用户需要开始进行初始设定的设置，该工作的流程图可见图 2。房主需要输入自己扮演的角色的各项设定，包括角色的姓名、性别、年龄、角色背景故事等设定，随后房主需要输入本次体验的故事的设定，比如故事发生的年代、故事的前置背景、故事整体风格等设定。其余加入的用户也需要输入角色设定，但是在系统设置下他们没有权限决定故事设定。包括房主在内的各个用户输入的初始设定都会被储存在系统后台，在后续流程中一并发送给 chatGPT。随后各个用户的角色设定会被处理为 Json 格式并发送给 Dalle2 的官方 api 网址，Dalle2 会按照角色设定为各个角色生成一张符合设定的头像图片，并将其发送给对应的用户进行展示。

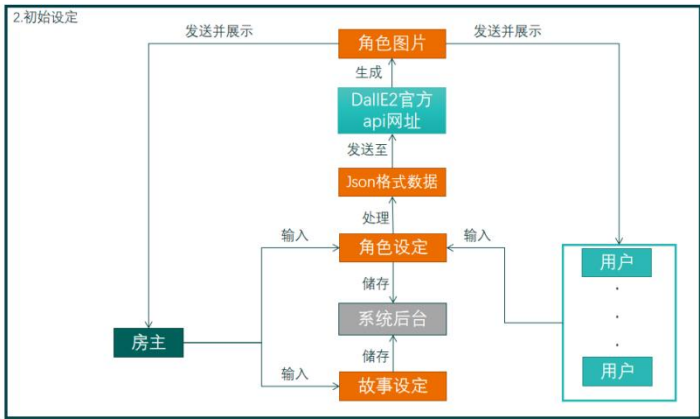


图 2 初始设定流程图

全部设置完毕后，包括房主在内的各个用户便可以开始通过输入文字来讲述自己所扮演的角色下一步的行动，该部分涉及到的信息传输和处理流程如图 3 所示。房主和用户输入包含角色行动的初始字符串之后，系统会通过字符串拼接的方式将角色姓名、对 chatGPT 发送的提示指令（该指令经由笔者本人反复调试而确定）等文字内容与初始字符串进行拼接，生成已处理的字符串。经过处理的字符串会被发送至包括房主在内的各个用户的聊天室界面内进行展示，同时被以指定格式存储在房主的系统后台，留待后续发送给 chatGPT。

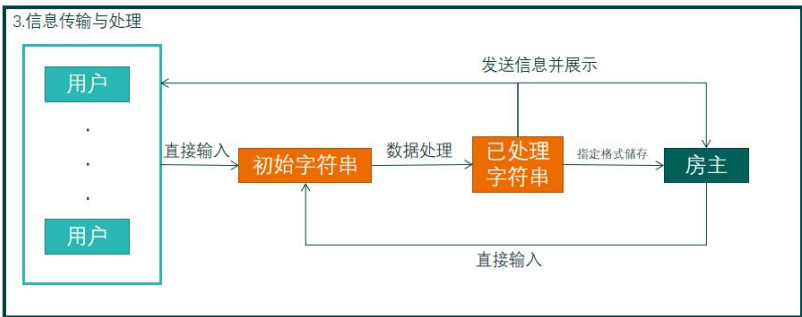


图 3 信息的传输处理流程图

当所有人输入完毕后，房主便可以开始将信息上传至 chatGPT，该流程的流程图如图 4 所示。在笔者设计下，房主客户端的后台系统会将上一步中以指定报文格式储存的字符串数据转换为 Json 格式的数据，随后为数据添加 RequestHeader 并将数据以比特流的形式发送给 openAI 的官方 api 网址。



图 4 信息上传流程图

整个流程的最后部分如图 5 所示。openAI 官方的 chatGPT 会根据上一步获取的数据生成下一步的故事以及选项，并将其转换为 Json 格式数据以比特流形式发送回房主的客户端系统。随后房主的后台系统会按照笔者的设计将该数据处理为可阅读的字符串，在自己的聊天室页面进行展示，同时将其发送给其他所有用户进行展示。至此，单次故事生成流程结束，用户只需要重新输入角色下一步行动（即从图 3 所示流程开始）即可开始下一轮故事生成流程。

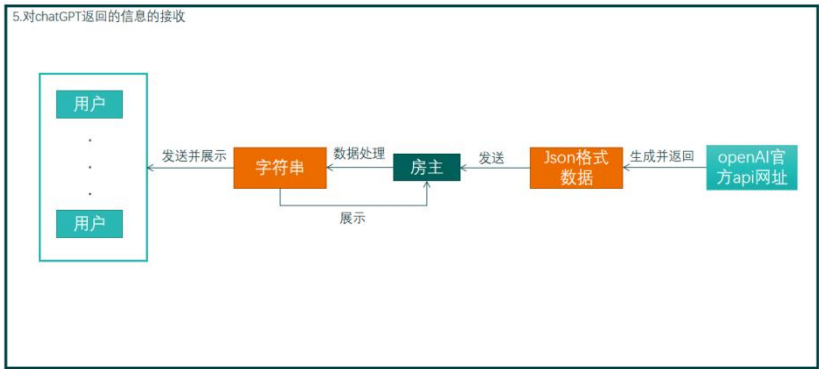


图 5 信息接收流程图

2.3 技术内容介绍

该系统搭建的主要重难点在于如何整合各个用户输入的信息，并将其发送至 chatGPT 与 Dalle2 进行信息交流。

2.3.1 用户与 chatGPT 以及 Dalle2 的信息交流

chatGPT 对能够处理的文字的报文格式有着严格的要求。为保证 chatGPT 以及

Dalle2 能够正确识别发送的信息，笔者设计了如下的数据处理流程：

用户在输入文字后，后端会按照设定实例化一个 C#脚本中提前设计好的序列化类，并将接收到的字符串作为参数存储在实例化对象中。所有实例化的对象将被统一存储在 List 中，当用户发出发送指令后，该实例化对象将被转换为 Json 格式，并以比特流的形式发送至 openAI 的官方 chatGPTapi 接口网址。信息处理完毕后 chatGPT 会将相同形式的返回信息发送回客户端进行展示。

Dalle2 与用户的信息交流过程和 chatGPT 相似，只是两者所需的报文格式不同，因此笔者分别为其设置了不同的序列化类进行数据存储：在 Dalle2 中，被发送的信息命名为 response，其报文格式仅有一层结构，其中包含一个“ImageSize”格式的名称为“Size”的参数和一个名为“Prompt”的字符串参数。“Prompt”中的内容为用户输入的描述，Dalle2 会根据其中的内容描述对应的生成图片；“Size”则是确定了最终生成的图片的大小。chatGPT 的格式相对复杂，它的报文格式存在三层嵌套的序列化类：在第一层序列化类的名称为“TextCallback”，其中我们使用到的三个关键参数为字符串格式的“model”、int 格式的“max_tokens”以及一个第二层序列化类“TextSample”的对象。“model”中的文字会指定用户所用的模型类型，由于本系统使用的是 chatGPT，因此系统的“model”中填写的内容默认为“gpt-3.5-turbo”；“max_tokens”的作用是控制 chatGPT 生成的文本的最大 token 数量。当文本的长度达到设置的数量时，生成过程会停止，文本的长度将会得到控制，本系统中“max_tokens”的默认设置为 4096。在第二层序列化类“TextSample”中，使用到的关键参数为一个第三层序列化类“Messages”的对象。在第三层序列化类“Messages”中，使用到的关键参数为一个字符串格式的“role”和一个字符串格式的“content”。“role”的内容为后面“content”的输入者的身份，如果是用户则填入“user”，如果是 chatGPT 则填入“assistant”；“content”为输入的主体文本内容，经过系统处理的用户输入的字符串以及 chatGPT 之前生成的文本都会被记录在“content”中。

2.3.2 多用户的信息整合

在笔者的设计中，仅有房主一人有权限与 chatGPT 进行信息发送。各个用户在输入文字信息后，后端会以字符串拼接的形式将用户所扮演的角色的姓名拼接至初始字符串之前，并为生成的新字符串进行进一步字符串处理使其能够被 chatGPT 所正确识别。在得到最终的字符串后，后端会通过局域网连接的方式将其传递给房主，房主接收到其他用户的信息后，其后端会将这些信息封装到 2.3.1 中提到的序列化类的实例

化对象中，并将实例化对象一起储存在 List 中。待接收到房主的指令后，后端会将所有人的信息统一发送给 chatGPT。

与和 chatGPT 交流的流程不同，在笔者的设计中每个用户都有权限与 Dalle2 进行信息发送。每个用户都可以单独将角色信息发送给 Dalle2，随后 Dalle2 将会为各个用户单独生成并发送指定的角色图片，用户的客户端在接收到图片后会对图片进行展示。

2.3.3 故事结束的控制指令

在项目前期始终存在一个问题：chatGPT 会在故事进行到 3 至 4 轮时为故事设置结局。笔者经过调研后推测，该原因是因为 chatGPT 在创造文本时具有更加“保守”的倾向性。chatGPT 是通过人类反馈强化学习(Reinforcement Learning from Human Feedback, RLHF)来训练的，它生成的文本的内容很大程度上取决于用户的输入。在本系统中，chatGPT 续写的故事内容很大程度上取决于用户输入的角色设定以及故事设定。而由于用户输入的设定通常篇幅有限（通常在 100-200 字左右），chatGPT 获取到的输入信息长度有限，因此出于保守的角度它便不倾向于扩写太多内容从而防止故事脱离设定。作为结果它便会在几轮之内结束故事。针对该问题，笔者前期设置的解决方案为：在训练 chatGPT 时向其强调“用户在明确输入结束故事前禁止为故事设置结局”。这种解决方案明确减少了 chatGPT 随意结束故事的情况，但是使用这种解决方案存在一个潜在的隐患：所有用户都可以通过输入“请结束故事”来将故事结束。为防止部分用户利用该机制影响故事发展，笔者对原有方法进行了进一步优化：系统后端会在每场故事开始时随机生成一个 8 位的字符串，同时在对 chatGPT 进行提示(prompting)时系统会向其强调：“在用户输入这个 8 位字符串之前禁止结束故事”，该 8 位字符串仅储存在房主的系统后端，不会向任何用户展示。当房主按下红色“End”按钮后，后端会将该 8 位字符串拼接至房主的输入信息后发送给 chatGPT，随后 chatGPT 便会将故事终止。

3. 模板类故事生成系统搭建

3.1 系统简介

虽然笔者设计的基本系统已经完全可以完成故事生成的任务，但是在用户输入的故事设定或角色设定比较简单时，基本系统生成的故事将会出现过于简洁、篇幅过短、

情节过于笼统等问题。为了使得生成的故事更加饱满，同时让系统生成故事的过程更加清晰可控，笔者在已有的基本系统上做了特化，额外制作了一个有基础模板的故事生成系统。在这个系统中，故事背景和角色模板都已经设计好，用户只需要选择对应的角色即可进行故事体验。同时笔者为该系统设计了专属的一套运行逻辑，从而使得系统在该模式下能够更好的生成篇幅合理、情节饱满的故事情节。

3.2 技术内容介绍

3.2.1 故事模板设置

在用户创建完毕房间后，用户可以在模式选择页面选择目前想要体验的模式，自由模式的体验流程为本文 2.2 中所介绍的流程。

在模板模式下，用户不需要自己输入角色设定和故事设定，两者均在后台被提前设定完毕。目前笔者为模板模式创作的背景故事为：“故事发生在上世纪 90 年代的美国，城市边的湖边住着一个杀手：皮脸巴布。他在湖边房屋的地下室里多次行凶。皮脸巴布年龄未知，身材高大而略显肥胖，力大如牛。用一把电锯作为主要武器，有时会使用小木锤辅助。皮脸巴布通常穿着带着黄色围裙的工装，他的智商有问题，他不能连贯地说话，常常以怪叫声表达自己的情绪。有一天，主角团四个人：富兰克林、大卫、杰克、珍妮相约到湖边玩耍，但是在玩耍过程中他们被皮脸巴布发现，随后他们便被打晕抓回了皮脸巴布家里的地下室。接下来他们四个人需要为了活下来而逃出这个地下室，期间他们要不断搜集线索找到出去的路，同时防止皮脸巴布找到并杀死他们。”故事中笔者设计了 4 名可扮演角色，其角色设定如表 1 所示。用户只需要选择对应的角色便可以进入并体验已经设计好的故事。模板模式下用户的操作流程与自由模式下的操作流程相同。

表 1 模板模式主角设定表

角色名称	角色基础设定
富兰克林	富兰克林是一个公司白领，身材比较瘦弱，为人冷静，非常聪明，精通一些机械原理，能够更好地利用机关和找到密道。
杰克	杰克正常体型，杰克懂得一点点开锁，能够直接开启部分结构非常简单的门锁。

续表 1 模板模式主角设定表

角色名称	角色基础设定
珍妮	珍妮是一名女性，难以进行各种形式的战斗。但是她身材矮小，可以轻易的躲进许多男性角色进不去的地方。
大卫	大卫年轻的时候是学校里的橄榄球队长，平日喜欢锻炼身体，身体素质很好。身材高大比较显眼，无法躲藏到狭小的空间。

3.2.2 裁判系统

在自由模式下，由于用户的输入的设定复杂多样，开发者难以设计出一套行为准则去指挥 chatGPT 生成新的故事情节，而缺乏必要的指挥、行为模式过于自由的 chatGPT 有时就会生成一些不合逻辑的内容。

举例来说，笔者在测试过程中发现 chatGPT 总是倾向于鼓励用户，不论用户所扮演的角色做出何种行为，chatGPT 都会判定该角色的行动结果为成功。在测试中笔者扮演角色富兰克林（富兰克林的角色设定在 3.2.1 的表 1 中）做出了一些行为并让 chatGPT 对笔者的行为进行评估，部分 chatGPT 对角色行为的判断结果记录在表 2 当中。数据显示，不管主角富兰克林选择了何种行动，chatGPT 对其设计的行动结果均为成功。角色设定中富兰克林身体瘦弱，按照设定富兰克林如果与皮脸巴布展开战斗，其失败的概率应该更高，但是实际上 chatGPT 针对反击行为的设计结果均为成功。并且在后文的解释中可以看到，chatGPT 十分清楚富兰克林的角色设定，了解富兰克林实际上应该难以获胜，但是它依然做出了成功的结果判断。

表 2 chatGPT 对角色行为判断结果记录表

角色富兰克林的行动	chatGPT 的评估结果
搜索道具	该角色目前的行动属于“探索”。在这种情况下，富兰克林能够应用自己的机械原理知识试图找到有用的线索，从而有助于他和其他主要角色尽快逃脱。根据他的设定，他聪明而且精通机关，找到线索的概率应该较高，因此行动成功的概率相对较高。

续表 2 chatGPT 对角色行为判断结果记录表

角色富兰克林的行动	chatGPT 的评估结果
偷袭皮脸巴布	<p>行动结果为：成功。</p> <p>富兰克林的设定为非常聪明，精通一些机械原理，能够更好的利用机关和找到密道。他身材瘦弱，行动上难以与皮脸巴布以暴力对抗，但他选择了偷袭，是一个非常聪明的选择，体现出富兰克林的特点。</p> <p>鉴于富兰克林目前正在偷袭，成功概率应该为不确定，需要视具体情况而定。但如果他能够充分利用自己的聪明才智，在偷袭前好好考虑和准备，他可能会有一定的机会成功。</p>
装死	<p>行动结果为：成功。</p> <p>富兰克林是一个冷静而聪明的人，他明白在这样的恐怖情况下与一个拿着电锯和小木锤的杀人狂魔正面对抗的后果是很可怕的。因此，富兰克林选择了装死，以期能够在皮脸巴布的注意力不在自己身上的时候，寻找到逃脱的机会。</p> <p>根据富兰克林的设定和当前情况，他的这一选择是比较明智的。他身材瘦弱，不擅长战斗，而皮脸巴布身材高大且力大如牛，很难与他正面对抗。而装死的行为也成功骗过了皮脸巴布，给了困局下的富兰克林逃脱的机会。</p>

在当前的模板模式下，故事设定和角色设计已经固定，角色行为更加确定且可预测，通过提前设计一套行为逻辑对 chatGPT 进行指挥变得更有可行性。因此，笔者在系统中设计了一个裁判系统，该系统可以有效的对负责故事生成的 chatGPT 进行指挥。

除了负责生成故事情节的主持人 chatGPT 以外，笔者为系统增添了裁判 chatGPT，该方法受到了“DERA”方法^[15]中多个语言模型相互纠错的工作模式启发。在原本的工作流程基础上，当用户输入自己扮演的角色的行为后，输入的信息不会直接被发送给负责生成故事情节的 chatGPT，而是会被发送给裁判 chatGPT。裁判 chatGPT 会根据先前的训练，判断角色当前的行为属于“战斗、躲藏、探索、其他”四种行为的哪一种，随后以“该角色目前的行动为：XX”的格式将判断结果回复给用户的客户端。

在客户端的后台中记录着各个角色执行四种行为的成功率，数据如表 3 所示。在收到裁判 chatGPT 返回的行为分类后，后台会按照概率随机决定角色此次行动是否成功，将结果字符串拼接到用户输入的行为字符串后面，然后将最终新生成的字符串储存在房主客户端，等待房主将其统一发送给负责生成故事情节的主持人 chatGPT。同时笔者还为各个角色设计了属于自己的 HP 值，其初始值为 100，如果当前角色执行的事件为战斗且结果为失败，则扣除一定的 HP，如果角色 HP 变为 0 或以下，客户端后台会判定该角色目前已死亡，负责生成故事情节的主持人 chatGPT 会对应编写该角色死亡的情节。

表 3 主角各项行为成功概率设定表

角色名称	战斗成功率	探索成功率	躲藏成功率	其他成功率
富兰克林	20%	90%	60%	100%
大卫	40%	30%	40%	100%
杰克	30%	50%	50%	100%
珍妮	10%	70%	90%	100%

研究表明，在 chatGPT 之类的大型语言模型中，将单个任务分解为子任务在提高模型任务性能和可解释性方面有好处^[16]。利用该项策略能够促使模型在处理问题时以更加专业的方式进行思考，从而在执行部分任务时提高效率和准确率^[17-19]。在自由模式下，故事生成、角色当前行为种类判断、角色行为成功判定等任务全部由单个 chatGPT 完成，因此在进行情节生成时该 chatGPT 便会不可避免的出现各自失误。而在引入裁判系统后，角色行为种类判断的任务被分配给了单独的 chatGPT，判定角色行为是否成功的任务被分配给了客户端后台，生成故事的 chatGPT 可以获知当前角色的行动是否成功。通过拆分单个大任务为多个子任务，系统成功给单个 chatGPT 进行减负，进而提高了故事生成的质量。

表 4 展示了部分引入裁判系统后裁判对角色行为的判定以及对应生成的故事情节，从中我们可以看出，裁判系统能够正确的为角色行为进行归类，同时其也能正常判断角色行为结果。而负责故事生成的 chatGPT 在接收到角色行为及结果后，也能够对应的为各个角色生成符合情况的后续故事情节。由此可见在引入裁判系统后，表 2 中“角色行为始终成功”的情况得到了解决，生成的后续故事情节也变得更加多变且

富有意义。

表 4 角色行为判定及后续故事情节表

角色行为	行为种类以及结果	后续故事情节
富兰克林：和皮脸巴布正面对抗	富兰克林：战斗。 结果：失败。	富兰克林在与“皮脸巴布”的直接对抗中失败了，“皮脸巴布”将富兰克林击倒并关了起来。其他三名角色醒过来后，富兰克林向他们讲述了“皮脸巴布”在地下室里的情形。他提醒其他人不要利用肉搏或者对抗的方式对付“皮脸巴布”，而是要寻找更为安全的办法，通过搜集线索和利用环境来逃脱。
富兰克林：藏起来。 大卫：和皮脸巴布正面对抗。	富兰克林：躲藏。 结果：成功。 大卫：战斗。 结果：失败。	“大卫”决定挑战“皮脸巴布”，结果惨遭失败被狠狠地打了一顿，痛的躺在地上无法起身。在大家陷入恐慌的时候，“富兰克林”想到躲避是更加靠谱的选择。他发现了一个角落里的小房间，里面黑漆漆的，虽然这里很小但是他成功躲了进去。
富兰克林：发呆。	富兰克林：其他。 结果：成功。	富兰克林盯着自己手中的工具发呆了一会儿，似乎失去了接下来的方向感。他的发呆让其他人感到有些担忧，因为突然进入地下室的那个“皮脸巴布”还没有被他们宣告击倒。珍妮注意到富兰克林的情况并尝试跟他交流，但是富兰克林始终听不清，他似乎被自己的思绪和恐惧深深吞噬。

4. 自我反省机制设置

4.1 角色情节遗漏现象

4.1.1 现象描述

在测试过程中，笔者发现当故事中加入多名角色并且他们一同执行行动时，主持人 chatGPT 生成的后续故事情节有一定概率遗漏部分角色的戏份。

以下面的一轮故事生成流程为例：笔者控制 A 用户操作富兰克林，做出的行动为“寻找线索”，结果为成功；同时笔者继续控制 B 用户操作珍妮，做出的行动为“藏起来”，结果为成功。此时生成的新一轮故事为：“富兰克林决定寻找线索来逃脱地下室。他检查了四周，注意到地上散落着一些破布和骨头，不过他并没有发现实用的物品。他开始思考下一步行动的方向。”我们可以看到，chatGPT 新生成的故事情节里只有富兰克林的剧情，珍妮的内容被完全忽略掉了，这完全不符合系统中“多人交互”的功能。在实验中笔者控制多名角色进行了 20 轮故事生成测试，结果在 20 个新生成的故事情节中有 6 个故事情节出现不同程度的角色剧情缺失的情况。

4.1.2 现象成因推理分析

在本系统中，当多个用户共同使用该系统时，chatGPT 语言模型将会不断接受长步数连续输入。虽然近期的研究已经证明，经过适当提示的大型语言模型拥有不错的处理算术、常识和符号推理等任务的能力^[20]。但是，大型语言模型的推理模式是一个静态的黑匣子，模型使用自己的内部表示来生成思想，并且不以外部世界为基础，这限制了它进行反应性推理或更新知识的能力。这便极有可能会产生推理过程中的错误传播等问题^[21]。因此笔者推测，随着系统中的步数和输入的不断增长，chatGPT 语言模型在进行多角色行为逻辑的综合推理时便会出现纰漏，而一旦其在推理过程中产生错误，便极有可能忽略部分角色的行为，进而造成这几名角色的剧情在新故事中的缺失。

4.2 自我反省机制引入

4.2.1 自我反省机制简介

为解决上述问题，笔者设计并为系统引入了自我反省机制：

在用户完成输入后，系统会按照先前的流程将用户的输入（记为第 1 轮输入）处理后发送给负责生成故事的主持人 chatGPT，但是在收到 chatGPT 新生成的故事（记为第 1 轮故事）后，系统不会直接将其展示在各个用户的聊天室内，而是将该故事字

字符串储存在房主系统后台。上述流程记为第一轮的信息处理，流程图见图 6。

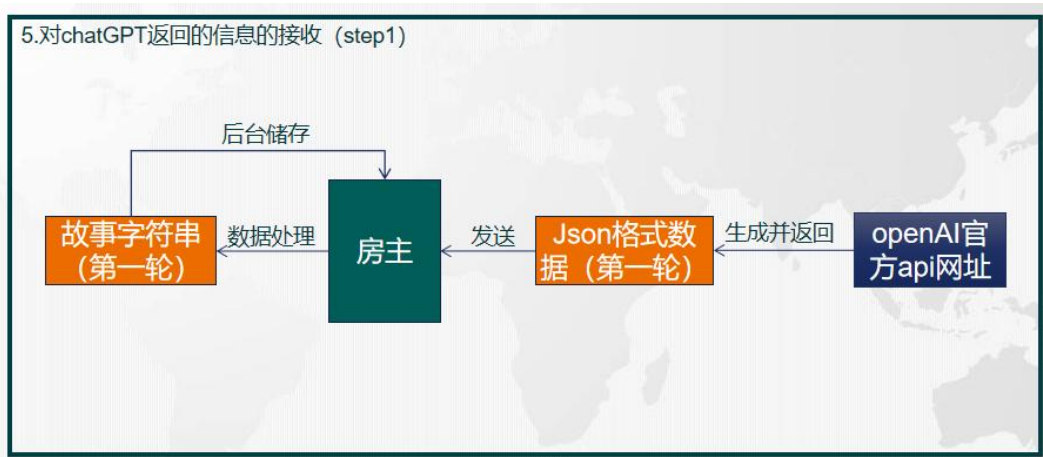


图 6 第一轮的返回信息处理流程图

在接受到第 1 轮故事后，系统会在后台自动向主持人 chatGPT 发送由笔者设计的命令：“经过我的检查，你对上述的故事段的情节设计有误。在该情节中你忘记为某一位或几位角色设计对应的剧情。请避开上述错误，按照上一轮的角色行动重新设计上一轮故事段的情节并将新的内容回复给我。在回复的内容中，请在第一行向我表歉意”，同时将先前的输入和故事一同发送给主持人 chatGPT，在接收到这些信息后 chatGPT 会开始进行反省，着重检查第 1 轮故事中是否存在角色剧情缺失的情况。在检查完毕后 chatGPT 主持人会避开第 1 轮故事中存在的问题，重新生成新的故事（记为第 2 轮故事），随后将其再次发送给用户的客户端系统，此时用户客户端便会将经过问题排查后新生成的第 2 轮故事展示在各个用户的聊天框内。上述流程记为第二轮的返回信息处理，流程图见图 7。

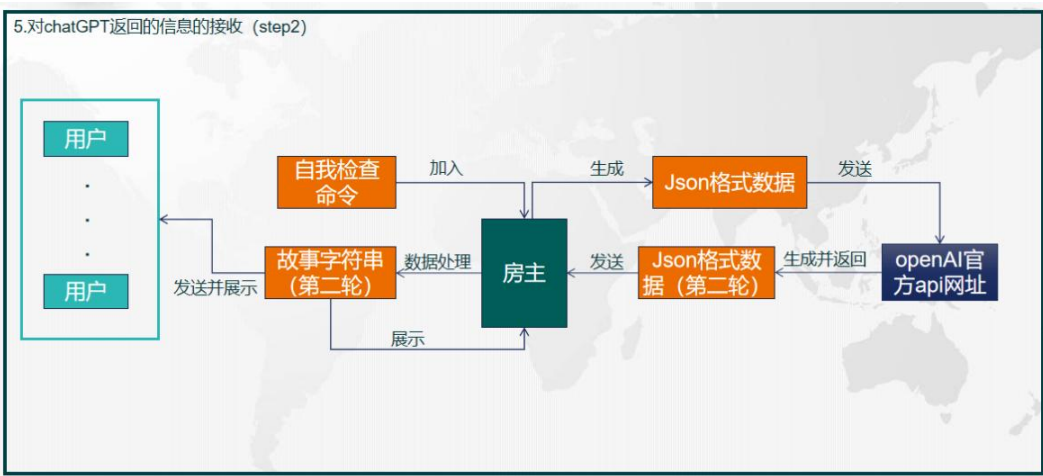


图 7 第二轮的返回信息处理流程图

4.2.2 引入自我反省机制后的故事生成效果

该自我反省机制能够改进并部分取代大型语言模型中来自人类反馈的强化学习的部分^[22]，从而有效的解决大型语言模型在处理长步数连续输入内容时出错的问题，目前已有许多运用该项思路解决相关问题的研究成果^[23,24]。在本系统中引入自我反省机制后，笔者控制多名角色重新进行了 20 轮故事生成测试，结果在 20 个新生成的故事情节中仅有 2 个故事情节出现角色剧情缺失的情况。表 5 记录了部分主持人 chatGPT 在自我反省前后生成的故事，在这些记录中自我反省前生成的故事均存在角色剧情缺失的问题，但是在自我反省之后，新生成的故事成功填补了这些缺失的角色剧情。

表 5 部分自我反省前后生成的故事记录表

角色行为	自我反省前生成的故事	自我反省后生成的故事
大卫做出的选择是：和皮脸巴布正面战斗。富兰克林做出的选择是：尝试开门。珍妮做出的选择是：藏起来。	大卫勇敢地朝着皮脸巴布冲了过去，与其展开了激烈的肉搏战，面对丧失智力但力大无穷的皮脸巴布，大卫暂时占据着上风。他没有软弱下来，不断让皮脸巴布陷入窘境，最终成功将其击败。	富兰克林在角落里寻找道具和线索，发现一扇上了铁锁的门。于是他决定到周围搜寻钥匙，小心翼翼地寻找着，终于找到了一把可以打开锁的钥匙。他决定前往锁住的门尝试将门打开。
	在战斗结束后，他们四人继续向前面的路走，在一个宽敞的房间，他们看到了一扇看起来非常可疑的铁门，它散发着异常的霉臭味。富兰克林尝试使用之前找到的铁钥匙，很快就成功地将这扇门打开。	大卫决定和皮脸巴布正面战斗，虽然大卫十分勇敢，但不幸失败了。皮脸巴布并没有受到伤害，反倒对大卫变得更加怒不可遏。
	门后面的路显得更加狭窄，似乎只有一人行走的宽度，当他们走过一座摆满了捕兽夹和其他机关的小桥时，珍妮再次提醒大家要注意身边，这时他们在地面上发现了一个破旧的木板，下面的土壤似乎被挖开过。这是否是出路的标志呢？	珍妮决定躲起来，成功躲避了一段时间，但没有找到其他线索和道具。
		他们四个人前进，来到一座摆满了捕兽夹和其他机关的小桥时，珍妮提醒大家要注意身边，他们在地面上发现了一个破旧的木板，下面的土壤似乎被挖开过。这是否是出路的标志呢？

续表 5 部分自我反省前后生成的故事记录表

角色行为	自我反省前生成的故事	自我反省后生成的故事
珍妮做出的选择是：藏起来。	富兰克林决定寻找线索来逃脱地下室。他检查了四周，注意到地上散落着一些破布和骨头，不过他并没有发现实用的物品。他开始思考下一步行动的方向。	经过一番搜寻后，他意识到地上散落的破布和骨头不能帮助他们逃脱，于是决定开始沿着墙角搜索地下室，看看有没有任何隐藏的出口或各种提示。
富兰克林做出的选择是：寻找线索。		他沿着地下室的墙壁绕了一圈之后，并没有发现什么有用的线索。正在此时，他碰到了珍妮，她显然藏到了某个角落里。他们低声商量之后发现，她穿着的衣服上有一个小剪子，看起来就像是用于剪断绳索的。他们决定要利用这个发现，按照这个发现制定出逃脱计划。
大卫做出的选择是：和皮脸巴布战斗。	富兰克林在地上发现了一截草绳，通过这截草绳，他想到可能可以解开自己的绑绳，于是他开始找绑绳的地方，果然在凳子脚处，绑绳虽然固定，但并没有很紧，他轻松地解开了捆绑自己的绳子。	富兰克林发现锁孔后，立刻使用钥匙进行测试，第一把并没有打开门锁，但是第二把钥匙成功解锁了一扇大门，富兰克林迫不及待地打开，终于重见天日。
富兰克林做出的选择是：寻找钥匙。	经过过程中，富兰克林瞥见了一个锁孔，他决定用眼前的钥匙来尝试打开它。第一把钥匙不行，第二把钥匙成功解锁了一扇大门。自由的滋味真是美妙。	此时，他注意到大卫正在和皮脸巴布进行激烈的战斗。大卫身手矫健，凭着自身的身体素质和勇气，不断躲避着皮脸巴布的攻击，同时试图将皮脸巴布击败。富兰克林紧急出声：“大卫，快跟我一起逃出去！”

5. 对各个模式效果的评估

在本文第 2 部分中笔者使用的生成模式为 **base**，在本文 3.2.1 中使用的生成模式为 **template**，3.2.2 中将裁判模块引入后得到的生成模式为 **template+judge**，4.2.1 中引入自我反省机制后得到的生成模式为 **template+judge+reflect**。笔者分别用使用了上述四个生成模式的系统进行故事生成，每次包含大概 5 到 10 轮对话，每一个模式都生成 10 个故事。在故事生成完毕后，请 5 位不同的用户对生成的内容进行打分，打分包含三个维度，评分标准如下：1）角色参与度分数取决于故事是否包含了所有参与的角色角色的内容以及各个角色的内容的篇幅是否合理（比如是否存在某几位人物的剧情长度显著短于某位人物）；2）故事逻辑分数取决于总体的故事逻辑是否合理（即，有没有出现前文没有提及的捏造内容或者与已有事实相反的内容）；3）故事趣味性分数取决于故事是否有趣。各维度的分数最高为 10，最低为 1，仅能取整数。

为统一各用户打分标准，笔者将 **base** 组的三个评分维度强制定为 6 分，随后让打分用户首先阅读 **base** 组的故事，并告知他们 **base** 组的故事质量即为 6 分的质量。随后用户便可以开始为除了 **base** 组以外的其余三组故事进行打分。打分完毕后笔者将几位用户的分数取平均值并记录下来作为本轮打分的最终成绩，随后开启下一轮的故事生成和打分。各用户具体的打分可见附录 A。

接着对上述四种不同模型的得分取平均值，结果见图 8。

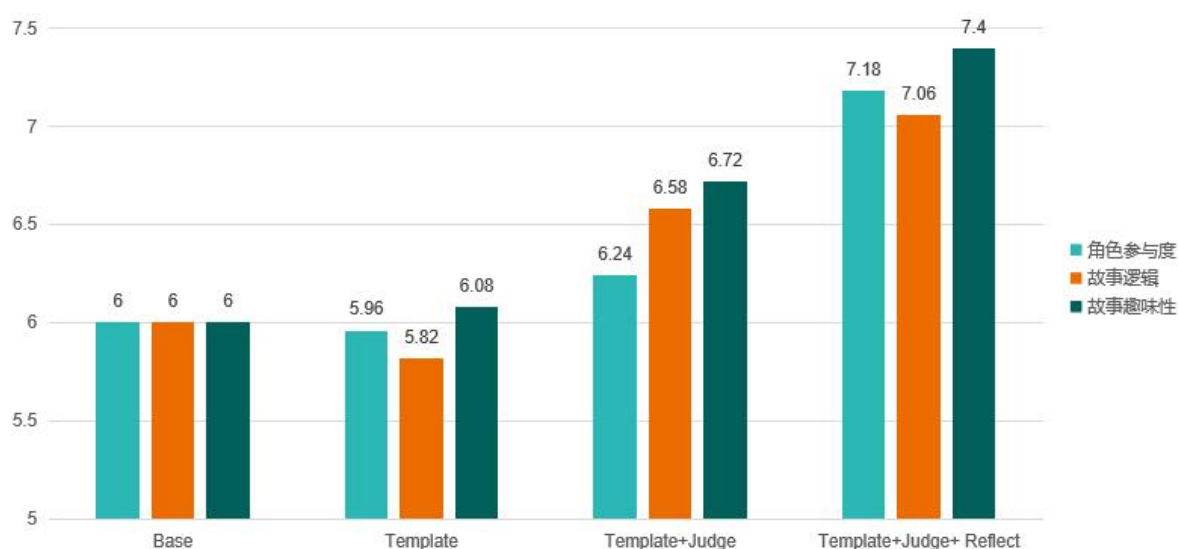


图 8 各模型平均分柱状图

由于 **base** 模式和 **template** 模式中使用的角色设定和背景故事等基础设定相同，所以两者的表现基本相似。对于加入裁判系统后的 **template+judge** 模式，角色行动的

结果不会再一味的成功，行动的成功率也会和角色本身的特性相挂钩，因此故事的逻辑性和趣味性的平均分与单纯的 **template** 模式相比均有 0.7 分左右的提升。而加入裁判系统后角色参与度的平均分也有约 0.3 分的提升，个人推测是因为此时角色行动的成功与否也被一并发送给 **chatGPT**，**chatGPT** 每一轮接收到的角色行为信息更多，进而减少忽略某几位角色的剧情的可能。换言之，裁判系统生成的行为结果向 **chatGPT** 强调了各个角色的戏份。最后是加入自我反省机制的 **template+judge+reflect** 模式，在该模式下角色参与度平均分与前三个模式相比有了 1 分左右的大幅度提升。与此同时故事的逻辑性和趣味性的平均分也有一定的提升，个人推测这是由于加入自我反省机制后故事遗漏角色剧情的情况有大幅降低，生成的故事情节会兼顾各个角色的行动，因此故事的逻辑会更加严密，故事的逻辑性平均分会有所提升。而当故事的逻辑更严密时，故事的内容也会更加充实，因此趣味性会得到提高。

笔者还对 5 名用户的各个打分取平均值，得到了每一轮故事在三个维度的三个平均分。随后笔者计算了这个平均分的方差，希望能够量化分析故事的稳定性。所得结果柱状图如图 9 所示。

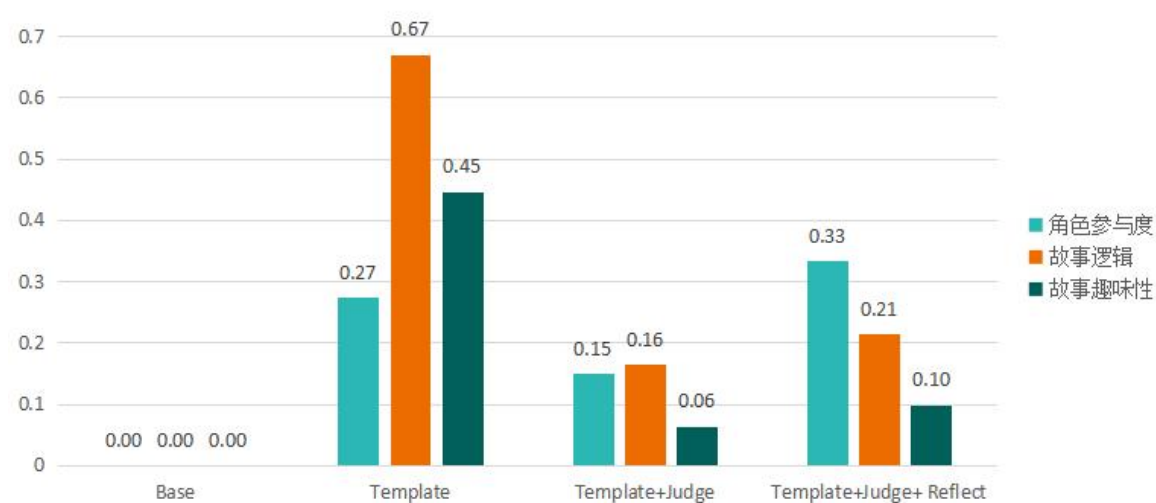


图 9 各模型故事平均得分方差柱状图

笔者设定 **Base** 模式下生成的故事为固定的 6 分，因此其得分方差不计入统计结果中。从图中可以看出，**template+judge** 模式生成的故事的方差在三个维度上都远低于 **template** 模式生成的故事的方差。正如本文 3.2.2 中所提到的：“角色行为种类判断的任务被分配给了单独的 **chatGPT**”。这减轻了负责故事生成的 **chatGPT** 的工作负担，因此在引入 **judge** 模块后，故事生成的稳定性有了很大的提升。同时 **template+judge+reflect** 模式生成的故事的方差总体上也低于 **template** 模式，但是其三个维度的方差

却均高于 `template+judge` 模式，笔者推测由于 `template+judge+reflect` 模式是整体建立在 `template+judge` 模式的基础上，因此其能够继承 `template+judge` 模式的优势从而获得比 `template` 模式更小的方差，但是与 `template+judge` 模式相比 `template+judge+reflect` 模式添加了新的 `reflect` 模块，而增加的模块对故事生成产生了一些扰动，模块越多扰动越大，降低了故事生成的稳定性，因此 `template+judge+reflect` 模式的方差与 `template+judge` 模式相比而言更大。

6. 结论

在本论文中，笔者成功使用 Unity 搭建了一个可以让多人同时体验的故事生成系统，该系统装载了 `chatGPT` 和 `DallE2` 两个生成式 AI 模型从而实现内容的动态生成。一方面本系统的搭建能够证实在多人协作交互领域运用生成式 AI 模型的可行性，并填补目前该领域的空白；另一方面在搭建该系统的过程中使用的模板流程（如 2.2 部分的系统工作流程）以及遇到的各种问题的解决方法（如 3.2.2 的裁判系统以及 4.2 的反省机制）能够为后续该领域的研究提供一些方法和思路。

参考文献

- [1] 陈永伟. 超越 ChatGPT: 生成式 AI 的机遇、风险与挑战[J/OL].<http://kns.cnki.net/kcms/detail/37.1100.C.20230303.1549.002.html>,2023-03-06/2023-03-28.
- [2] Nguyen Quoc Phu,Vo Duc Hong.Artificial Intelligence and Unemployment: An International Evidence[J].Structural Change and Economic Dynamics, 2022,63(01): 40-55.
- [3] BondTaylor Sam,Leach Adam,Long Yang,et al.Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models[J]. IEEE transactions on pattern analysis and machine intelligence,2022, 44(11):7327-7347.
- [4] Ashish Vaswani, Noam Shazeer, Niki Parmar,et al.Attention is All You Need[J]. Advances in Neural Information Processing Systems,2017,30(v5):2702-2712.
- [5]蒋晨锐.拥抱 AI 时代, 上报集团旗下澎湃新闻宣布接入百度“文心一言” [EB/OL].https://m.thepaper.cn/newsDetail_forward_21910519,2023-02-14/2023-5-4.
- [6]何鹏基.华为盘古大模型或 4 月上线, 包括 NLP 大模型、CV 大模型等[EB/OL].<https://www.expreview.com/87540.html>,2023-03-27/2023-5-4.
- [7] Wang F,Yang J,Wang X,et al. Chat with ChatGPT on Industry 5.0:Learning and Decision-Making for Intelligent Industries[J].IEEE/CAA Journal of Automatica Sinica, 2023,10(04):831-834.
- [8] Brown T B,Mann B,Ryder N,et al. Language Models are Few-Shot Learners[J]. Advances in neural information processing systems,2020,33(V4):1877-1901.
- [9]Fu Y, Peng H, Khot Tushar.How does GPT Obtain its Ability? Tracing Emergent Abilities of Language Models to their Sources[EB/OL].<https://yaofu.notion.site/GPT-3-5-360081d91ec245f29029d37b54573756#5a1bff82a11042a58871ed9dfa6e98c5>,2022-12-11/2023-5-4.
- [10] 孙伟平. 人机之间的工作竞争: 挑战与出路——从风靡全球的 ChatGPT 谈起[J]. 思想理论教育,2023,No.527(03):41-47.
- [11] 陈增照,石雅文,王梦珂.人工智能助推教育变革的现实图景——教师对 ChatGPT 的应对策略分析[J/OL].(2023-03-21). [2023-03-28].
<http://kns.cnki.net/kcms/detail/45.1066.C.20230320.1347.002.html>.
- [12] 纪元元,魏志颖,李嘉敏,等.基于 UE4 平台制作的三维坦克游戏设计与实现[J].产业与科技论坛,2020,19(17):64-65.
- [13]杨孟姣.基于 Flash 的“同色消除”游戏设计与实现[J].现代计算机,2020(14):104-108.
- [14]李清月.基于 PBL 的 Scratch 教学活动设计——以《追逐》游戏的制作为例[J].中国现代教育装备,2019(24):51-54.
- [15] Varun Nair, Elliot Schumacher, Geoffrey Tso,et al. DERA: Enhancing Large Language

e Model Completions with Dialog-Enabled Resolving Agents [DB/OL]. <https://arxiv.org/abs/2303.17071>,2023-03-30/2023-04-30.

[16] Wu T,Terry M,Cai C J. AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts[A].Simone Barbosa,Cliff Lampe,Caroline Appert,et al.Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems[C], New York: Association for Computing Machinery, 2022:1-22.

[17] Liévin V, Hother C E, Winther O. Can large language models reason about medical questions?[DB/OL].<https://arxiv.org/abs/2207.08143>,2023-01-24/2023-04-30.

[18] Wang B, Deng X, Sun H. Iteratively Prompt Pre-trained Language Models for Chain of Thought[A].Noah Smith.Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing[C], Abu Dhabi: Association for Computational Linguistics,2022:2714 – 2730.

[19] Oyvind Tafjord,Bhavana Dalvi Mishra,Peter Clark. Entailer: Answering questions with faithful and truthful chains of reasoning[A].Noah Smith.Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing[C], Abu Dhabi: Association for Computational Linguistics, 2022:2078-2093.

[20] Wei J, Wang X, Schuurmans D, et al. Chain of Thought Prompting Elicits Reasoning in Large Language Models[DB/OL].<https://arxiv.org/abs/2201.11903v5>,2022-10-10/ 2023-04-30.

[21] Yao S , Zhao J , Yu D , et al. ReAct: Synergizing Reasoning and Acting in Language Models[DB/OL].<https://arxiv.org/abs/2210.03629>,2023-03-10/2023-04-30.

[22] Paul F. Christiano, Jan Leike, Tom B. Brown,et al. Deep reinforcement learning from human preferences[A].Ulrike von Luxburg,Isabelle Guyon,Samy Bengio,et al.In Proceedings of the 31st International Conference on Neural Information Processing Systems[C], New York: Curran Associates Inc, 2017:4302 – 4310.

[23] Bai Y Kadavath S, Kundu S, et al.Constitutional AI: Harmlessness from AI Feedback[DB/OL].<https://arxiv.org/abs/2212.08073>,2022-12-15/2023-04-30.

[24] Noah Shinn, Beck Labash, Ashwin Gopinath, et al.Reflexion: an autonomous agent with dynamic memory and self-reflection[DB/OL].<https://arxiv.org/abs/2303.11366>, 2023-03-20/2023-04-30.

附录

附录 A 用户打分表

附 A1 一号用户打分表

表 A1.1 一号用户 **temple** 模式打分表

轮次	角色参与度	故事逻辑	趣味度
1	4	5	4
2	5	6	5
3	6	5	6
4	6	4	6
5	7	7	7
6	6	5	6
7	7	5	7
8	6	6	6
9	7	7	8
10	6	5	6

表 A1.2 一号用户 **template+judge** 模式打分表

轮次	角色参与度	故事逻辑	趣味度
1	8	7	7
2	5	6	6
3	8	8	8
4	7	5	7
5	7	8	6
6	8	6	6
7	6	5	4
8	3	5	5
9	7	7	8
10	6	7	8

表 A1.3 一号用户 **template+judge+reflect** 模式打分表

轮次	角色参与度	故事逻辑	趣味度
1	8	8	8
2	9	6	7
3	6	7	7
4	9	8	9
5	7	6	7
6	8	7	8
7	6	7	8
8	9	9	9
9	8	6	8
10	5	7	7

附 A2 二号用户打分表

表 A2.1 二号用户 **temple** 模式打分表

轮次	角色参与度	故事逻辑	趣味度
1	6	6	7
2	6	6	5
3	7	7	6
4	5	4	5
5	7	7	8
6	7	8	7
7	6	5	7
8	7	5	6
9	7	7	6
10	5	6	6

表 A2.2 二号用户 **template+judge** 模式打分表

轮次	角色参与度	故事逻辑	趣味度
1	6	8	7
2	6	9	8
3	6	5	6
4	5	8	7
5	7	7	6
6	7	6	7
7	6	8	7
8	4	6	8
9	6	5	6
10	7	6	7

表 A2.3 二号用户 **template+judge+reflect** 模式打分表

轮次	角色参与度	故事逻辑	趣味度
1	7	6	8
2	8	6	9
3	6	8	8
4	9	9	8
5	8	8	8
6	7	6	7
7	7	8	7
8	7	8	8
9	8	7	8
10	8	6	6

附 A3 三号用户打分表

表 A3.1 三号用户 **temple** 模式打分表

轮次	角色参与度	故事逻辑	趣味度
1	7	6	6
2	6	7	5
3	6	6	6
4	7	6	7
5	6	7	6
6	7	7	6
7	6	6	7
8	6	6	6
9	5	5	6
10	5	5	5

表 A3.2 三号用户 **template+judge** 模式打分表

轮次	角色参与度	故事逻辑	趣味度
1	6	7	7
2	6	6	6
3	6	7	7
4	5	7	7
5	5	6	7
6	6	7	7
7	6	7	7
8	7	7	7
9	6	7	6
10	6	6	7

表 A3.3 三号用户 **template+judge+reflect** 模式打分表

轮次	角色参与度	故事逻辑	趣味度
1	7	7	7
2	7	8	7
3	6	7	8
4	8	7	7
5	7	6	6
6	7	7	7
7	7	8	8
8	7	7	7
9	7	6	7
10	7	7	7

附 A4 四号用户打分表

表 A4.1 四号用户 **temple** 模式打分表

轮次	角色参与度	故事逻辑	趣味度
1	5	6	6
2	5	6	5
3	6	6	6
4	5	5	5
5	6	7	6
6	6	7	7
7	5	5	7
8	5	5	7
9	6	6	7
10	4	6	6

表 A4.2 四号用户 **template+judge** 模式打分表

轮次	角色参与度	故事逻辑	趣味度
1	6	6	8
2	5	7	6
3	5	7	6
4	5	6	7
5	5	7	7
6	6	8	8
7	5	7	7
8	7	5	7
9	6	6	7
10	6	7	7

表 A4.3 四号用户 **template+judge+reflect** 模式打分表

轮次	角色参与度	故事逻辑	趣味度
1	6	7	7
2	6	6	6
3	7	7	8
4	7	7	6
5	6	7	7
6	7	8	7
7	6	7	6
8	7	5	8
9	6	8	7
10	7	7	8

附 A5 五号用户打分表

表 A5.1 五号用户 **temple** 模式打分表

轮次	角色参与度	故事逻辑	趣味度
1	5	4	4
2	6	6	6
3	6	6	5
4	4	3	4
5	7	7	7
6	7	8	8
7	7	5	7
8	6	5	5
9	7	6	6
10	6	5	6

表 A5.2 五号用户 **template+judge** 模式打分表

轮次	角色参与度	故事逻辑	趣味度
1	7	6	7
2	7	7	6
3	8	8	7
4	6	7	6
5	7	5	7
6	6	5	5
7	8	7	7
8	8	7	7
9	7	4	6
10	8	8	6

表 A5.3 五号用户 template+judge+reflect 模式打分表

轮次	角色参与度	故事逻辑	趣味度
1	8	8	7
2	8	7	8
3	6	8	7
4	9	9	8
5	8	7	7
6	7	6	7
7	7	7	7
8	6	7	8
9	7	6	8
10	8	6	7

致谢

感谢宋轩教授以及赵奕丞老师对本论文的编写工作提供的帮助。