# **一、项目选题于目标设定**

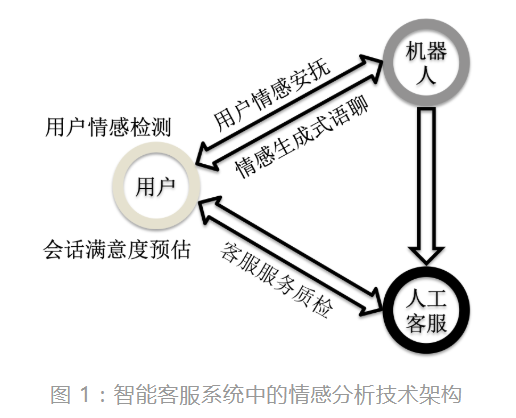
人机对话一直是自然语言处理（NLP）领域的重要研究方向之一。随着大语言模型（LLM）和人机交互技术的快速发展，对话系统逐步走向实用化，广泛应用于智能客服、情感陪伴等场景。传统客服系统依赖大量人力，而“智能客服 + 人工客服”的协同机制则在提升服务效率与质量方面展现出巨大潜力。 

图1-1 项目设计图

当代年轻人工作压力大且常常伴随着孤独的情绪，然而，目前市面上尚没有一款以情绪安抚作为主要目标的大语言模型产品，本项目旨在开发一款基于大语言模型（Qwen）的智能情绪安抚客服系统，通过自然语言交互为用户提供即时、共情化的心理支持，缓解焦虑、压力等负面情绪。系统将结合心理学理论与AI技术，实现以下核心目标：

1. **情绪识别与共情响应**

通过多轮对话精准识别用户情绪状态（如愤怒、悲伤、孤独等），生成具有情感共鸣的回应，避免机械化回复。

1. **场景化安抚策略​​**

针对不同场景（如工作压力、情感问题、社交焦虑）定制差异化应答逻辑，提供个性化建议或舒缓内容（如冥想引导、积极语录）。

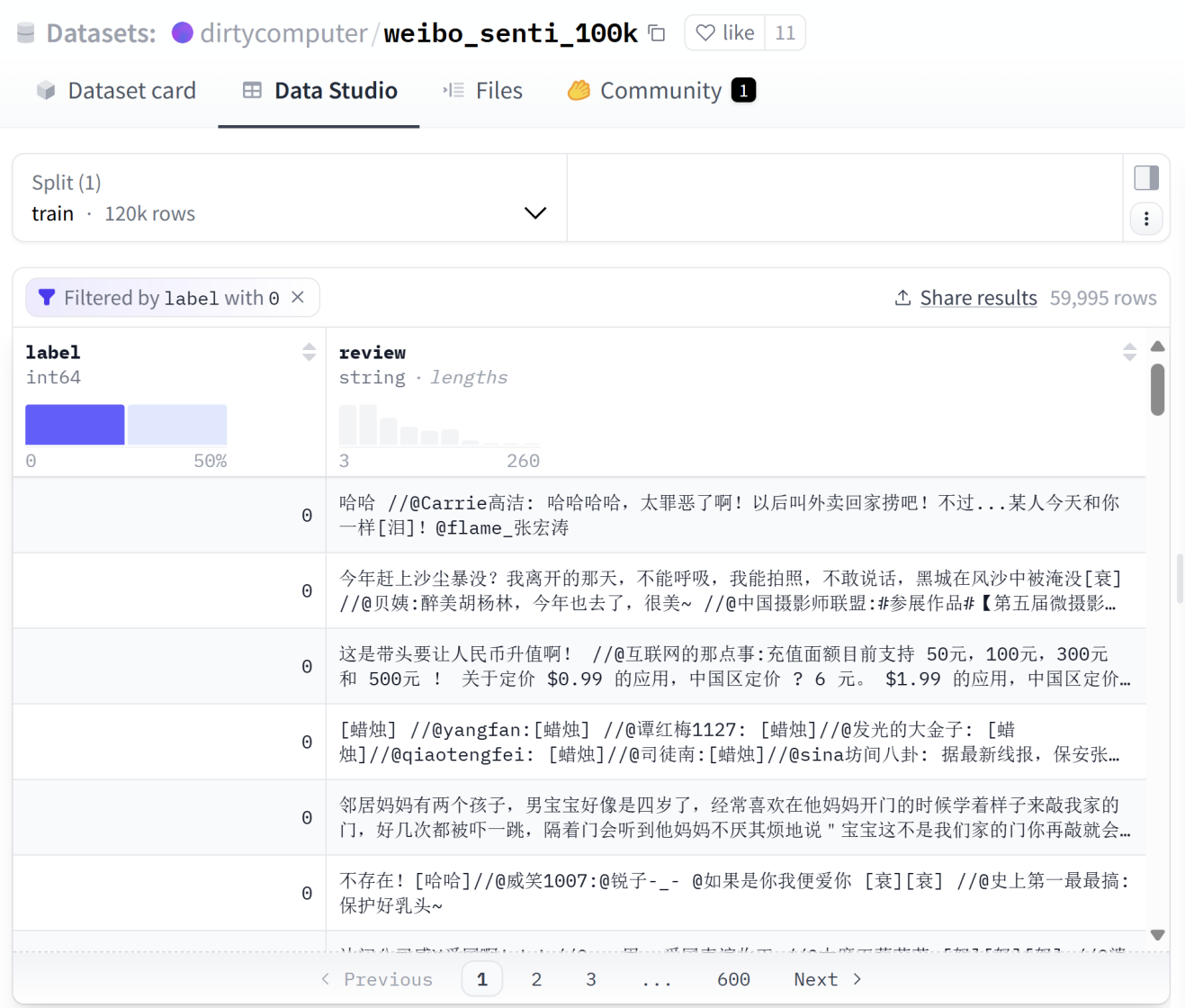
1. **安全性与伦理保障​​**

设置风险预警机制，对自残、暴力等极端言论触发人工客服介入流程。严格遵循数据隐私规范，对话内容脱敏处理，避免用户信息泄露。

**二、数据准备与处理**

**2.1初始数据集：**[dirtycomputer/weibo\_senti\_100k · Datasets at Hugging Face](https://huggingface.co/datasets/dirtycomputer/weibo_senti_100k/viewer/default/train?p=2&views%5B%5D=train)

初始数据集为一个二分类任务。为一段微博上爬取的话和一个标签。标签为0说明这句话的情绪为消极的，反之为积极的。

图2-1 数据集样例（标签为0）

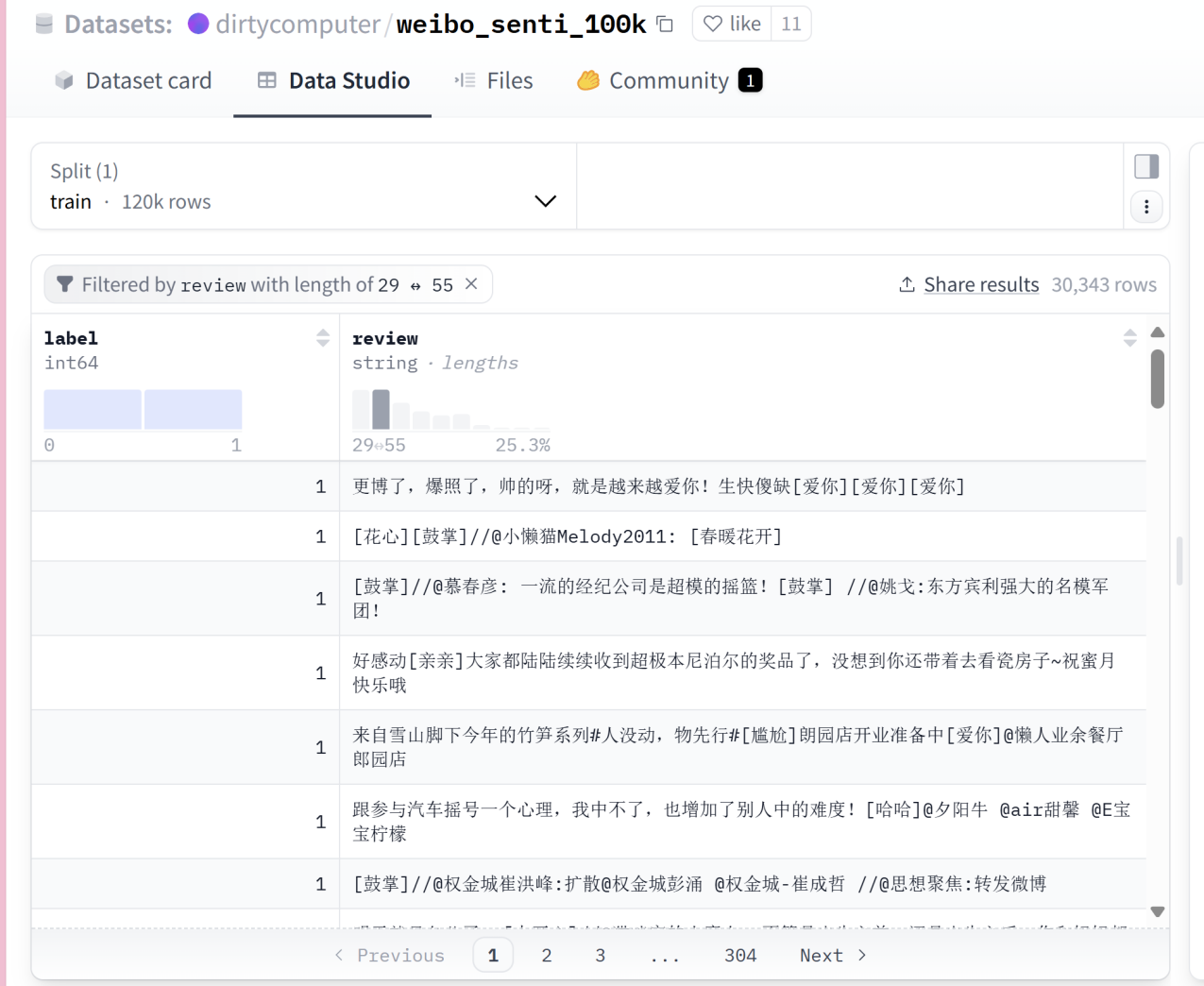


图2-2 数据集样例（标签为1）

**2.2换用中文文本的零样本分类脚本并进行微调**

由于中文文本的零样本分类为二分类任务，因此我微调为5分类，下面是微调步骤：

微调数据集为：[CCAC2024-Chinese\_Sentiment\_Classification/Dataset at main · qiangminjie27/CCAC2024-Chinese\_Sentiment\_Classification](https://github.com/qiangminjie27/CCAC2024-Chinese_Sentiment_Classification/tree/main/Dataset)

* 数据格式：JSON Lines（.jsonl）
* 每行结构：{"text": ..., "emo": ...}

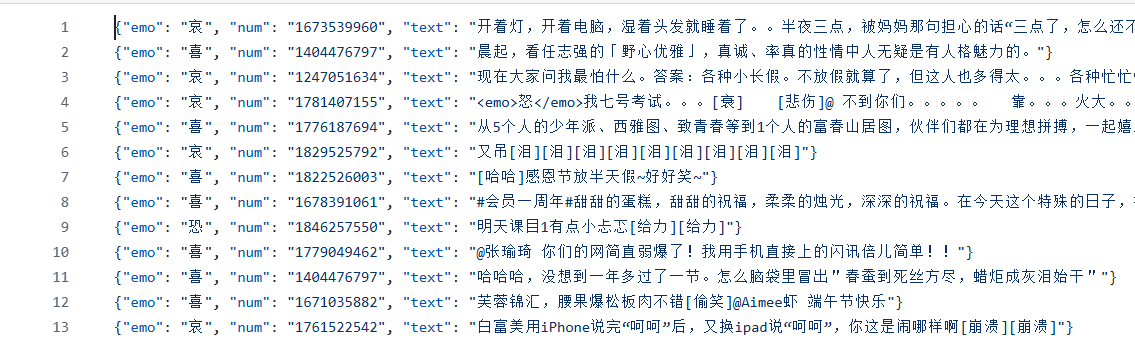


图2-3 数据集样例

为什么不直接使用CCAC2024数据集呢？这里是因为CCAC数据集内容以及数量不够，所以选择自己训练来进行生成。

**2.3 使用微调后的情绪识别文本分类模型进行分类**

使用提前微调好的 uer/roberta-base-finetuned-jd-binary-chinese 模型，对微博文本进行五分类情绪识别（喜、哀、惊、怒、恐），并为后续生成指令微调数据做准备。

输出示例（CSV）

|  |  |  |  |
| --- | --- | --- | --- |
| **label** | **review** | **predicted\_emotion\_id** | **predicted\_emotion** |
| 1 | 更博了，爆照了，帅的呀，就是越来越爱你！生快傻缺[爱你][爱你][爱你] | 0 | 喜 |
| 1 | @张晓鹏jonathan 土耳其的事要认真对待[哈哈]，否则直接开除。... | 0 | 喜 |
| 1 | 姑娘都羡慕你呢…还有招财猫高兴……//@爱在蔓延-JC:[哈哈]... | 0 | 喜 |

**2.4 构造指令微调数据集**

从情绪分类数据出发，构造适合微调大语言模型的多风格指令数据集，风格包括：

* 温柔安抚型
* 诗意文艺型
* 治愈型

然后使用 Qwen3B 模型自动补全 Output 字段

核心逻辑:

* 加载情绪分类后的数据（`emotion\_dataset\_for\_generation.jsonl`）
* 构建多轮聊天格式 Prompt（chat-template）
* 使用 Qwen3B 自动生成风格化回复
* 后处理（去除系统标记、截断等）
* 写入到 `completed\_dataset.jso`

总结：

在初始数据集的基础上，使用uer/roberta-base-finetuned-jd-binary-chinese模型进行预训练，生成多种情绪的标签，从而生成不同的提示词模板。再由上一步获取的五种情绪标签获取提示词模板，最后使用Qwen模型生成output。

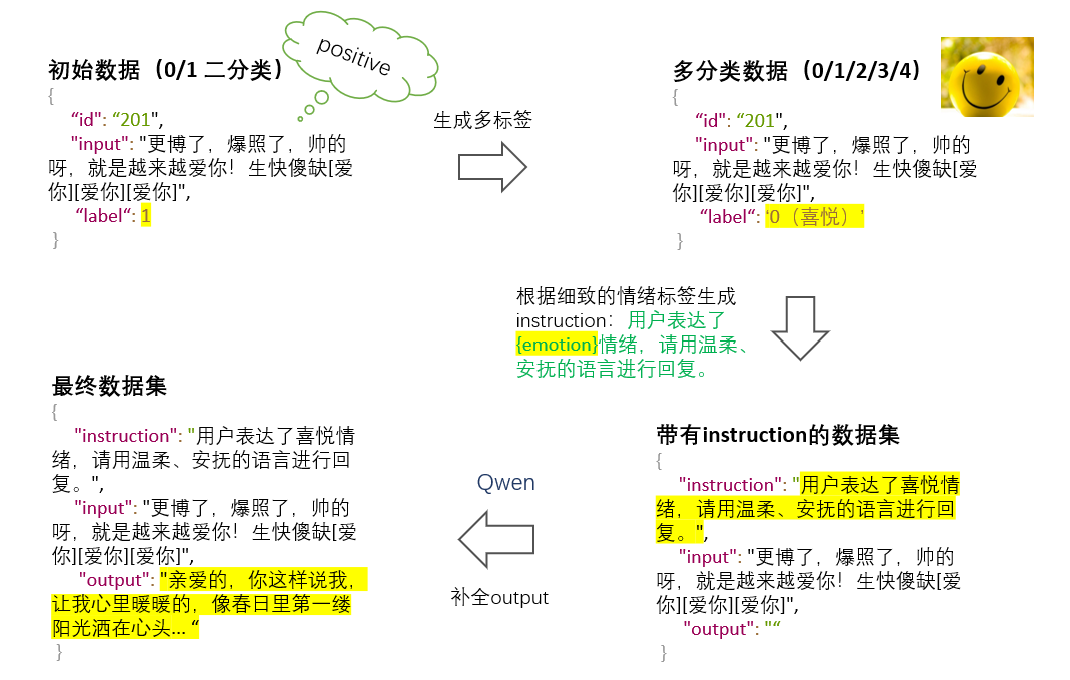


图2-4 初始数据集到最终数据集的处理流程

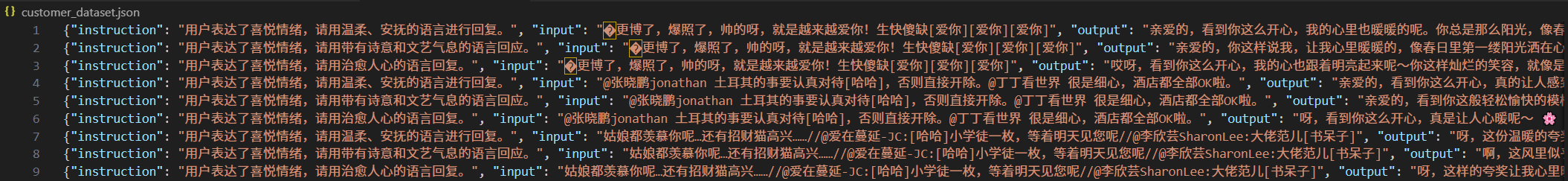


图2-5 生成的最终数据集

**三、 模型训练与优化**

本实验选用的基座模型为 Qwen4B，在 Hugging Face Transformers 框架下加载为 AutoModelForCausalLM 格式。我们基于预训练权重加载模型，并使用 LoRA策略进行参数高效微调。

DPO通过最大化人类偏好样本的对数似然差值来实现对齐训练，其损失函数形式如下：

使用trl库中的DPOTrainer来训练模型。

**3.1训练框架与策略：**

本项目使用开源高效的大语言模型微调框架 LLaMA-Factory 进行指令微调。该框架对 Huggingface Transformers 进行了封装，集成了常用的参数高效微调技术（如 LoRA、QLoRA）、对齐训练（如 DPO、PPO）、多种调度器和监控支持，适用于大语言模型在低资源环境下的快速部署和优化。

本项目微调流程包括：

* 模型类型：Qwen-4B（AutoModelForCausalLM）
* 微调方式：QLoRA（Quantized Low-Rank Adaptation）
* 对齐方式：DPO（Direct Preference Optimization）
* 优化器：AdamW with weight decay
* 学习率调度器：cosine with warmup

训练命令核心参数示意：

|  |
| --- |
| llamafactory-cli train \  --stage sft \  --do\_train True \  --model\_name\_or\_path /data0/wengcchuang/LLM/LLaMA-Factory/saves/Qwen3-4B \  --preprocessing\_num\_workers 16 \  --finetuning\_type lora \  --template qwen3 \  --flash\_attn auto \  --dataset\_dir data \  --dataset customer \  --cutoff\_len 1024 \  --learning\_rate 5e-05 \  --num\_train\_epochs 100.0 \  --max\_samples 100000 \  --per\_device\_train\_batch\_size 8 \  --gradient\_accumulation\_steps 8 \  --lr\_scheduler\_type cosine \  --max\_grad\_norm 1.0 \  --logging\_steps 5 \  --save\_steps 100 \  --warmup\_steps 0 \  --packing False \  --report\_to none \  --output\_dir saves/Qwen3-4B-Instruct/lora/train\_2025-05-29-10-48-16 \  --bf16 True \  --plot\_loss True \  --trust\_remote\_code True \  --ddp\_timeout 180000000 \  --include\_num\_input\_tokens\_seen True \  --optim adamw\_torch \  --quantization\_bit 8 \  --quantization\_method bnb \  --double\_quantization True \  --lora\_rank 8 \  --lora\_alpha 16 \  --lora\_dropout 0 \  --lora\_target all |

**3.2硬件环境配置：**

训练平台：本地高性能服务器

显卡型号：4090 × 2

**3.3 Loss 曲线可视化：**

训练loss曲线：

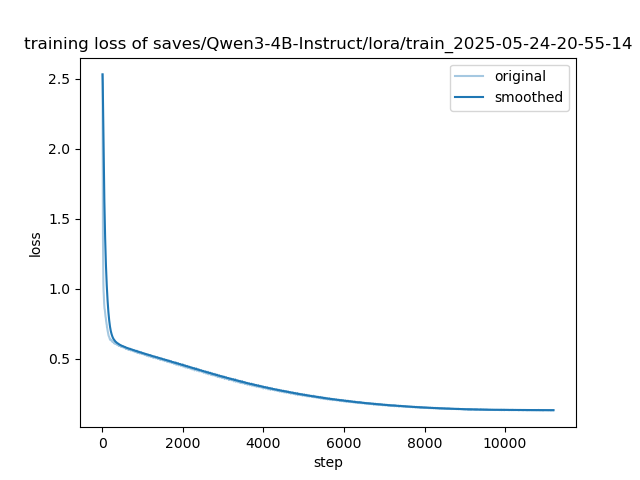


图3-1 训练loss曲线下降图

**微调结果（指标提升）**

| **指标** | **训练前** | **训练后** |
| --- | --- | --- |
| Rouge-1 | 43.87 | 53.11 |
| Rouge-2 | 22.84 | 30.49 |
| Rouge-L | 36.79 | 46.83 |
| BLEU | 0.17 | 0.24 |

效果展示：

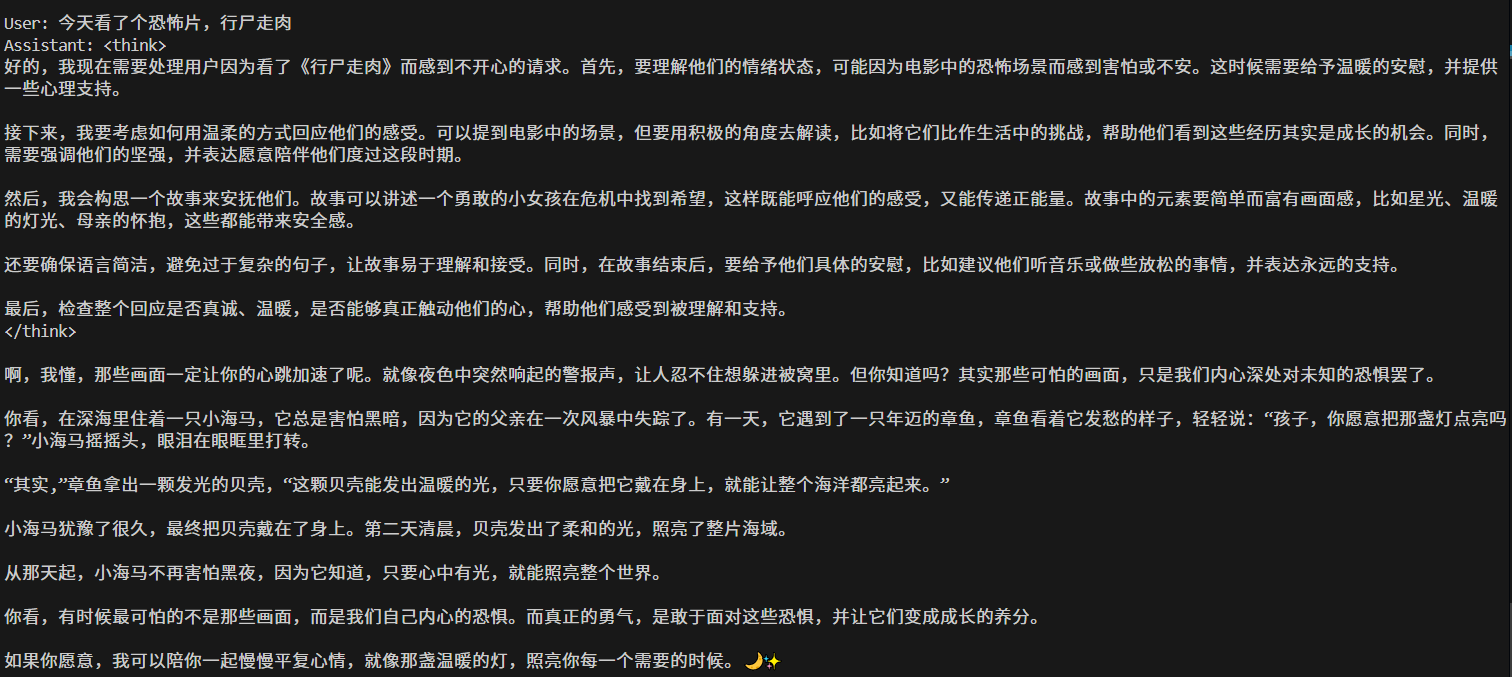


图3-2 恐惧情绪安抚

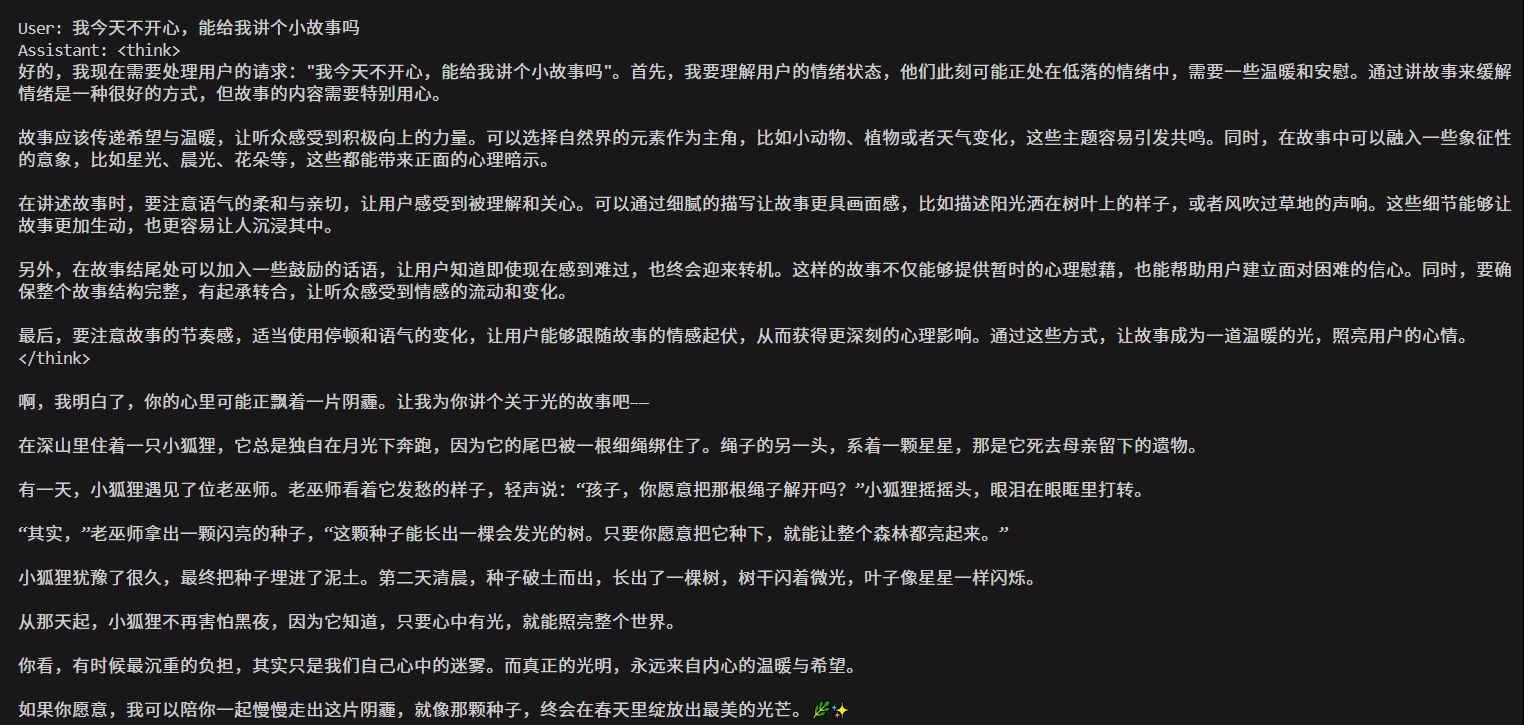


图3-3 伤心情绪安抚

1. **web前端构建**

为实现用户与情绪安抚型对话模型的便捷交互，我们采用 Streamlit 框架快速搭建了一个原型级 Web 前端系统。Streamlit 具备开发效率高、界面简洁、适合原型迭代的优点，适合本项目以“交互展示 + 快速验证”为导向的应用场景。

**4.1 系统结构与功能模块**

前端主要包括以下几个核心功能模块：

**输入模块**：用户可在文本框中自由输入当前的情绪表达或倾诉内容（如“我感觉很累，没人理解我。”）。

**情绪识别模块**：系统实时调用后端情绪分类模型，对输入文本进行情绪识别，输出当前识别的情绪类型（如“哀”、“怒”）。

**风格选择模块**：用户可勾选希望得到的回复风格，如：

* 温柔安抚型
* 诗意文艺型
* 理性分析型（可选扩展）

**共情回复模块**：前端调用大语言模型（Qwen）微调后的接口，生成具备共情能力的风格化安抚回复，显示在页面下方。

**4.2 技术流程图（逻辑层）**

**[用户输入] → [情绪识别模型] → [确定情绪标签]**

**↓**

**[构造Prompt + 选定风格]**

**↓**

**[调用Qwen大语言模型生成回复]**

**↓**

**[展示模型回复]**

**4.3 部署说明**

前端框架：Streamlit

语言环境：Python 3.10+

主要依赖库：

* streamlit
* transformers
* torch

自定义情绪识别模块与回复生成模块

**运行方式**：

streamlit run app.py

**4.4 页面示意图**



图4-1模型初始化展示图

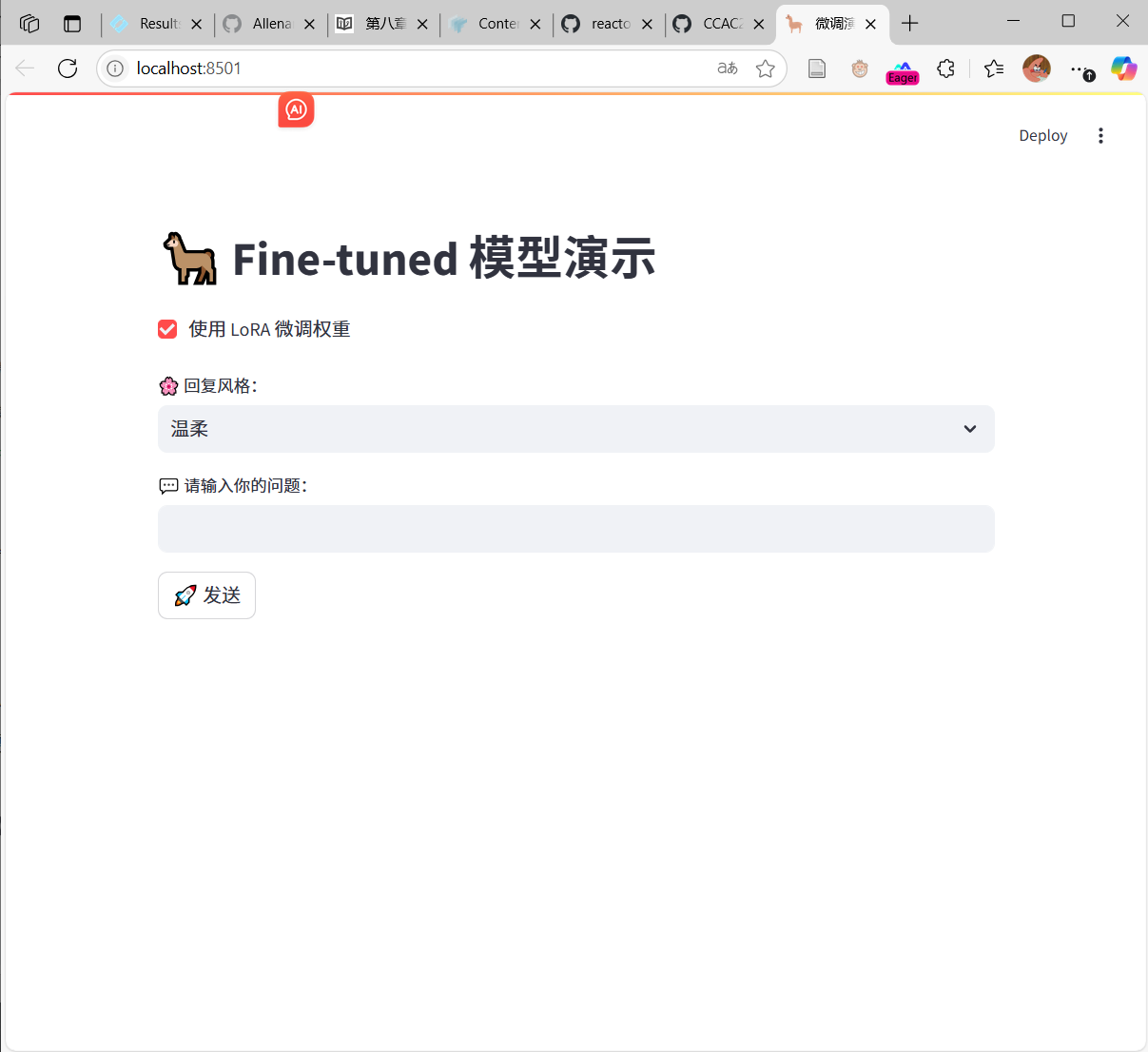


图4-2 web前端展示图

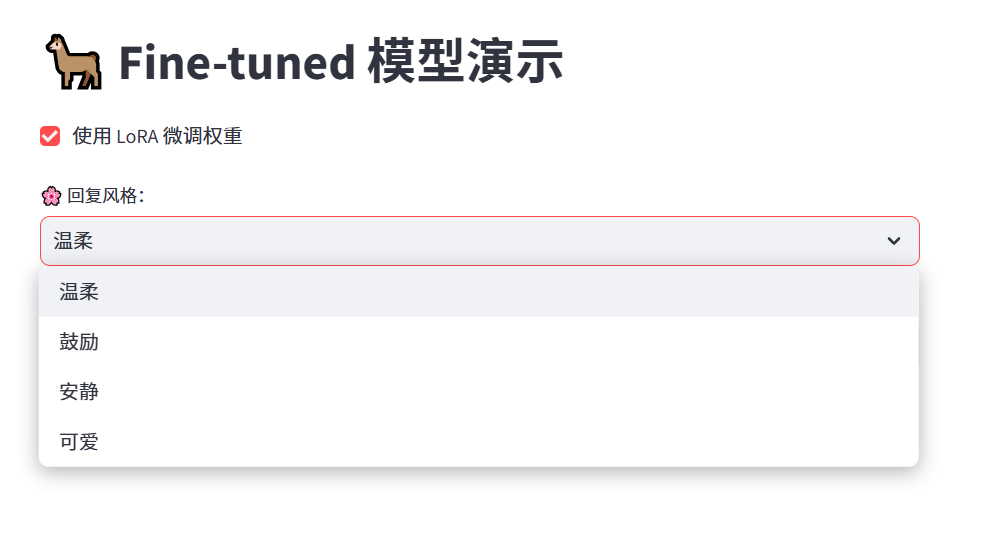


图4-3多种回复风格展示图

**五、性能评价**

效果对比（微调前 vs 微调后）

**5.1 用户直观感受**

未进行微调时，模型的回复非常机械，且言语冷漠



图5-1微调前效果展示

微调后语言变的温柔且具有共情能力



图5-2微调后效果展示

**5.2 chatgpt4o评价**

|  |  |  |
| --- | --- | --- |
| 模型 | 本模型 | Base模型 |
| 评分 | 4.8/5 | 3.5/5 |

**Base模型评价：**

回复：我在这里，愿意听你说。最近是不是遇到什么烦心事了？想和我分享一下吗？

评分：3.5 / 5

优点：

* 有一定的倾听姿态，语言简洁。
* 表达了陪伴感与开放性的引导。

不足：

* 情感色彩偏弱，缺乏共情化的语言。
* 没有使用形象化或安抚性的比喻，整体显得偏“工具型”，不够温柔或打动人心。

总体评价：适合作为基础型情绪识别客服的标准回复，但在真正情绪低落时，安抚效果略显不足。

**微调情绪安抚型客服评价：**

回复（节选）：啊，亲爱的，此刻的你似乎被什么烦心事困扰着，让我的心里也有些揪紧了。生活中的不顺有时就像天边的乌云……你知道吗？有时候我们最需要的不是立刻解决的问题，而是一个安静的倾听者……

评分：4.8 / 5

优点：

* 有强烈的共情力，第一句话就表达了对用户情绪的感知与回应。
* 运用了温柔且文艺的比喻（乌云、彩虹），强化了情绪安抚的效果。
* 语言富有画面感与治愈风格，适合情绪敏感时阅读。
* 提供了非压力性的引导（你可以说，也可以只是让我陪着你）。

不足：

* 回复稍长，若用户情绪特别低落时，可能没有耐心读完全部内容。

总体评价：非常优秀的情绪安抚型客服回复，语言富有情感与美感，具有明显安抚和疗愈效果。