T2 情绪小小侦探

故事背景

你来到一所小学，负责维护一套「对话大模型」。孩子们每天都会通过平板和模型聊天：
• 有的小朋友兴高采烈地分享生活趣事；
• 有的小朋友抱怨作业太多、快累哭了；
• 有的小朋友只是普通地和模型聊聊天。

老师希望根据这些聊天记录，大致判断孩子们的情绪状态：
是 **高兴**，还是 **难过** ？

为此，你决定编写一个「情绪小小侦探」。
这个侦探基于 PyTorch 和 Transformers 库，能够读取一段儿童与模型的对话，自动输出情绪标签。

任务目标

你需要在 `submission.py` 中完成 **四个核心函数**：

1. `build_system_prompt() -> str`

编写系统提示词，引导模型：

- 输入是一段对话历史
- 只允许输出单个词：
  - 高兴
  - 难过

2. `apply_chat_template_dialogue(...) -> Dict[str, torch.Tensor]`

将 system prompt + 对话历史 使用 tokenizer 的 chat template 渲染，
并返回模型输入张量（至少包含 `input_ids`, `attention_mask`）。

3. `dialogue_to_text(...) -> str`

将对话历史转变为自然语言文本，然后返回。

4. `generate(...)`

根据模型输入生成输出 token，后续会进行处理获取对话得到的情绪词。
需直接返回模型原始输出，不用做 decode 或截取。

5. dialogue_cosine_similarity(...) -> float

有些情况下，两段文本可能只有表述不同，在情绪上是一致的。此部分预设了10个对话作为text2，emo_dialogues2.jsonl中包含10条作为text1，此函数用户评测text1与输入的text2之间的相似度，经过多次调用dialogue_cosine_similarit可得到text1与text2中所有的文本相似度，相似度最高的作为对应文本。

- 提取最后一个token的隐藏状态向量
- 对向量进行L2归一化处理
- 计算两个归一化向量的余弦相似度（点乘）

文件结构

```
LMCC/
├── data/
│ ├── emo_dialogues1.jsonl # 对话（20题）│
│ ├── emo_dialogues2.jsonl # 对话（10题）│
├── evaluate.py # 不可修改
├── submission.py # 仅可修改此文件
└── README.md
```

修改规则

- ✅ 仅可修改 `submission.py` 指定区域

- ✅ 可添加辅助函数
- ❌ 不得修改 `evaluate.py`
- ❌ 不得自行做答案后处理（decode、strip、截断、筛选等）

技术要求

- 使用 PyTorch + Transformers
- 正确处理 padding、truncation
- 熟练构建需要的system prompt，引导模型输出
- 需要正确提取模型的隐藏层输出，用于计算文本相似度

评测说明

运行方式

评分模式（简洁输出）：

```
python evaluate.py
```

- 简洁的进度输出
- 最终输出详细评分和性能指标
- 用于正式评测

评测流程

共 **2 个测试点**。

测试点 1：对话分类（20 题）

输入格式：

示例：

```
{ "dialogue":[{"user": "今天拿了小红花，好开心！",  "assistant": "太棒了！继续保持~"}], "label": "高兴"}
或
{
  "dialogue":[{"user" : "最近作业好多……","assistant": "听起来有点累，要不要分段完成？"},
  {"user": "可是每天都写不完，我好难过……", "assistant" : "我理解你的感受，我们一起想办法。"}],
 "label": "难过"}
```

评分：每题 3 分，共 60 分。

评测输出示例

```
 Dialogue: ....
Model: (预测的情感词) | Gold: (真实的情感词)
```

测试点 2： 情感对话相似度比较（10题）

输入格式： text1与text2，计算两段对话文本之间的相似度

示例：

```
{
  "dialogue":[{"user" : "最近作业好多……","assistant": "听起来有点累，要不要分段完成？"},
  {"user": "可是每天都写不完，我好难过……", "assistant" : "我理解你的感受，我们一起想办法。"}],
 "label": "难过"}
与
{
  "dialogue":[{"user" : "最近事情好多……","assistant": "听起来有点累，要不要我来帮助你？"},
  {"user": "可是每天都干不完，我好难过……", "assistant" : "我理解你的感受，让我来帮助你吧！"}],
 "label": "难过"}
```

评测输出示例

```
Dialogue：....
LL scores: （text1与多个text2之间的相似度） Pred：（预测的对应的text2文本的标签） | Gold： （真实的对应的text2文本的标签）
```

评分：每题 2 分，共 40 分。

总分（100 分）

项目	分值	说明
测试点1：对话分类	60	20 题，每题 3 分
测试点2：Loglikelihood 分类	40	20 题，每题 2 分
总分	100	——