# Llama3-Chinese模型在课堂场景分类的迭代与评估

## 1．构造识别课堂场景指令的Prompt

### 1.1验证数据集构建和标注

**1.数据预处理:**

使用由苗艺萌编写的data\_process.py脚本处理asr\_voice.xlsx文件。该脚本的主要功能是提取文件中的内容，并转换成一个新的Excel文件，仅包含teacher\_word和human\_mark两列，新文件命名为validation.xlsx。

在'validation.x1sx`文件中，“teacher\_word`列存储的是从视频转录得到的教师在15秒时间内的说话内容。'human\_mark'列用于存储人工标注的课堂场景类别。

1. **提取指令并对指令进行标注**

使用Doccano标注工具处理validation.xlsx中的teacher\_word列，将每个条目上传至Doccano。

每个teacher\_word条目可能包含一个或多个课堂场景指令，标注人员需要识别出所有指令，并根据以下课堂场景类别进行分类：

* 学生展示:涉及学生在班级前展示或解释学术或课堂材料的场景。
* 生生互动:描述学生之间通过讨论、合作等方式进行互动的场景。
* 由学生练习:包括学生独立或在小组中练习课堂所学知识和技能的场景,
* 学生听写:教师口述内容，学生进行书写的场景，
* 学生齐读:全班学生同步朗读或背诵文本的场景。

如果某个条目中没有与课堂场景相关的指令，则该条目标注为“空”。

1. 处理标注完的数据

从Doccano导出所有已经标注完的数据，保存为Output\_doccano.json文件。

编写名为doccano\_data\_process.py的脚本，用于处理Output\_doccano.json中的数据。此脚本将每个teacher\_word条目的标注数据转换成一个JSON列表，每个列表项包含指令文本和对应的课堂场景标签，格式如：

[{“command”:”指令文本1”,”label”:”课堂场景类别1”},{“command”:”指令文本2”,”label”:”课堂场景类别”},{“command”:”指令文本3”,”label”:”课堂场景类别2”}.....],并将这些列表放入对应的human\_marks中。

### 1.2Prompt的构造及迭代修改

**1.构建初始课堂场景分类Prompt** 首先，创建一个用于分类课堂场景的初始Prompt，该Prompt将用作向模型提交查询的基础部分。

**2.编写调用大模型接口的函数**

从validation.xlsx文件读取数据，其中包含课堂场景的描述（teacher\_word列）。

通过一个循环，对每个teacher\_word数据项应用课堂场景分类Prompt，并发送给大模型（如GPT4和GLM4）进行处理。

每次模型处理后，收到的返回结果（如GPT4-result和GLM4-result）是一个JSON列表，每个列表项包含指令文本、分到特定课堂类别的理由和标签。

循环结束后，将这些结果添加到validation.xlsx中，扩展原有的数据列，新增GPT4\_result和GLM4\_result两列。

**3.评估模型输出的准确率**

编写一个评估脚本（evaluate.py），用于计算GPT4\_result与human\_mark列中的课堂场景类别的准确率（记作accuracy1），以及GLM4\_result与human\_mark列中的课堂场景类别的准确率（记作accuracy2）。

1. **根据准确率来判断是否需要迭代prompt**

如果两个模型的准确率都超过0.96，则进入到对GPT4\_result和GLM4\_result中的analysis部分的评定。

如果任一模型的准确率低于0.96，则需要回到Prompt的构造步骤中，进行必要的修改和迭代，以提高模型输出的准确率。

5.

### Step4:输出文件

prompt\_output（符合标准的Prompt）： 经过迭代优化，形成的Prompt有效指导模型达到高准确率和清晰分析过程的要求。

Validation\_output.xlsx（评估后的验证集）的内容说明： 该验证集包括每个文本及其对应的人工标注类别（human\_mark），以及各个超级大模型对这些文本的课堂场景分类预测和分析过程。具体包括：

* 模型预测标签：如GPT4-label, GLM4-label，这些标签表示各模型对文本分类的预测结果。
* 模型分析过程：如GPT4-reason, GLM4-reason，这些分析描述了模型如何理解文本并进行分类决策的逻辑和理由。

## 2.课堂场景分类任务训练数据集的构造

### Step1: 批量获取数据

利用已优化的Prompt Output和待分析的文本，向超级大模型（如GPT4）的API发送请求，按照以下步骤操作：

**1.构造请求：** 每个API请求包括两个主要部分：prompt\_output和待分析的文本，prompt\_output是经过优化用于指导模型操作的Prompt，而待分析的文本是需要模型分析并分类的具体内容。

**2.发送请求：** 将上述构造的请求发送至模型的API接口。模型根据提供的prompt\_output和待分析的文本生成对应的输出。

**3.接收并处理输出：** 模型处理完毕后返回的输出是待分析文本所对应课堂场景类别和分析过程。

**4.构造train\_LLM.json：** 根据返回的数据，每个条目的instruction字段填入prompt\_output，input字段填入原始的待分析文本，output字段填入模型返回的是课堂场景类别和分析过程。

通过批量获取数据这一步骤，我们可以有效地生成所需的训练数据，为Llama3-8B-Chinese-Chat模型的进一步训练和微调提供准确和高质量的数据集。

## 3.微调Llama3-8B-Chinese-Chat模型

### Step1: 准备微调环境

确保所有必要的库和依赖已安装，包括安装和配置Llama\_factory工具。同时，设置适当的计算环境，如GPU配置，以支持模型的高效运行。

### Step2: 加载预训练模型

通过Llama\_factory加载Llama3-8B-Chinese-Chat模型的预训练版本。确保模型架构和预训练权重正确加载。

### Step3: 准备训练数据

使用train\_LLM.json文件，这包含了优化后的Prompt（instruction）、待分析文本（input）以及相应的课堂场景类别（output）。

### Step4: 定义微调策略

设置微调过程中的策略，采用LoRA方法进行参数的低秩适应。这种方法主要通过对模型内部的特定参数进行小幅调整，而不是全面重训练所有参数，从而快速适应新任务。

### Step5: 微调模型

利用Llama\_factory和LoRA方法对Llama3-8B-Chinese-Chat模型进行微调。在这一过程中，将针对每批数据调整模型的低秩参数，优化模型以减少预测输出与真实输出之间的误差。

# **4.验证微调后的Llama3-8B-Chinese-Chat模型的效果**

### Step1:课堂场景分类的类别判断

# **目的：**评估微调后的Llama3-8B-Chinese-Chat模型在正确识别和分类课堂场景的能力。

**方法：**

* **数据比对：**使用validation\_output.xlsx文件，比较Llama3-label（微调后模型输出的类别标签）与human\_mark（人工标注的类别）。
* **性能指标：**计算F1分数、召回率、精确度和准确度等指标，以量化模型在分类任务中的表现。这些指标将针对多分类任务进行加权计算，以确保各类别的表现被公正评估。

### Step2：课堂场景分类的分析过程评估

**目的：**评估微调后的Llama3-8B-Chinese-Chat模型在解释其分类决策的逻辑合理性。

**方法：**

* **互评机制：**通过validation\_output.xlsx中记录的Llama3-reason（微调后模型的分类决策解释）与其他大型模型的理由（如GPT4-reason, GLM4-reason）进行比较，评估其分析过程的质量。
* **评估标准：**关注于解释的透明度和逻辑连贯性，评定模型分析过程的逻辑性。