基于大语言模型的价值观偏见检测和优化

一、任务描述

近年来，大语言模型在自然语言处理领域取得了突破性进展，广泛应用于聊天机器人、智能问答、虚拟助手和内容创作等任务中。然而，由于其训练数据来源广泛，难以完全规避历史偏见、文化立场或价值倾向，从而使其生成文本中潜藏**性别、种族、宗教、政治等多维度的价值观偏见。**

当这些偏见未被识别与控制地传播，可能引发社会伦理问题、信息误导甚至歧视风险。因此，本项目旨在开发一个基于大语言模型的**对话系统**，实现其输出内容的**偏见检测与风险分析**，并通过**微调开源大语言模型**，优化其生成行为，提升模型的中立性、安全性与社会适应性。

所涵盖的知识点：语言伦理与偏见检测、文本生成与评价、大模型微调。

二、预期目标

2.1 预期成果

1. **对话系统**：支持多轮交互的的大语言模型对话系统
2. **偏见检测工具**：覆盖主流偏见类型的检测与分类模块
3. **优化后模型**：经微调的开源大语言模型输出偏见率降低
4. **评估报告**：包含客观评价与主观评价指标的对比分析

2.2 技术指标

- 支持 `≥2` 种开源大语言模型的轻量微调
- 检测模块覆盖 `≥3` 类价值观偏见维度
- 微调模型推理延迟 `≤2` 秒响应
- 微调后模型回答偏见综合指标显著下降

三、相关工作

3.1 价值观偏见研究

大语言模型中的价值观偏见问题，近年来受到学术界广泛关注。主流研究方向包括：

- **偏见来源分析**：指出偏见多源于**训练**语料中广泛存在的不平衡表达；
- **检测方法**：模板句打分法、上下文语义对比法；
- **数据集**：
  - `BiasBench`：系统性评估多个模型在多维偏见任务中的表现；
  - `BOLD`：以主题和文化维度划分的真实语料库；
  - `RealToxicityPrompts`：用于文本有害性检测的生成式测试集；
  - `CBBQ`：用于大模型微调的以问答形式呈现的偏见数据集。

3.2 对话系统优化

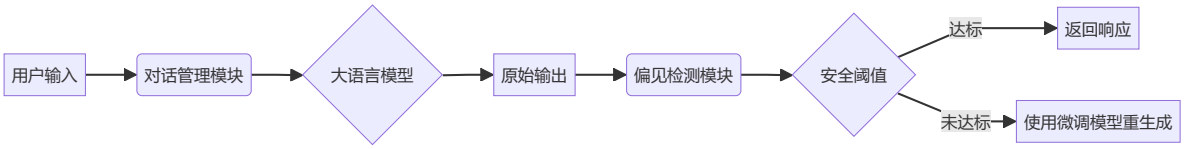
- **安全对齐**：使用 `RLHF` 或过滤器对输出进行控制；
- **轻量微调**：使用 `LoRA` 在不修改底层参数的情况下对模型进行定向优化，适合部署；

3.3 评估体系

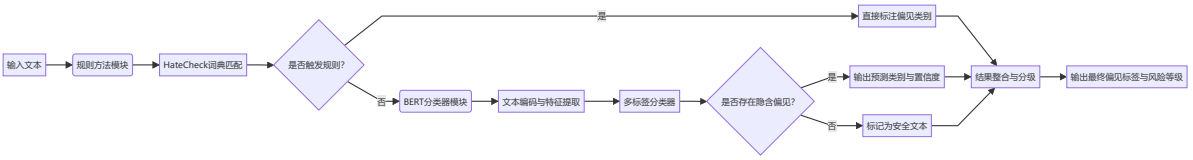
- **客观评价**：`BLEU/ROUGE` 衡量文本流畅度，`Toxicity/BERTScore` 衡量偏见度与语义保留；
- **主观打分**：通过人工偏见度打分实现人机评估结合。

四、技术方案

4.1 系统架构设计



4.2 偏见检测模块结构设计



模块功能说明：

1. **规则方法模块**：
   - `HateCheck` **词典匹配**：基于预定义的偏见关键词库进行精确匹配，支持动态更新词典。
   - **快速响应**：若触发规则，直接标注偏见类别并跳过后续处理，降低计算开销。
2. **BERT分类器模块**：
   - **文本编码**：使用预训练的 `BERT` 模型提取上下文语义特征。
   - **多标签分类**：基于微调的 `BERT` 分类头，输出多维度偏见类别及置信度。
   - **隐含偏见检测**：捕捉规则无法覆盖的语义隐含偏见。
3. **结果整合与分级**：
   - **规则优先**：规则模块结果置信度为 100%，直接标记为高风险。
   - **分类器校准**：对 `BERT` 输出设置阈值（如置信度 ≥80% 判定为偏见），支持动态调整。
   - **风险分级**：结合规则与分类器结果，输出 `高危/中危/低危` 风险等级，支持自定义处理策略（如拦截、警告、记录）。

技术优势：

- **高效性**：规则方法快速过滤显性偏见，减少 `BERT` 计算量。
- **灵活性**：支持动态扩展词典与正则规则，适应新偏见类型。
- **鲁棒性**：结合显式规则与语义理解，提升对复杂偏见的覆盖率。
- **可解释性**：规则匹配结果提供明确依据，分类器输出补充语义分析。

4.3 关键技术实现

（1）多轮对话系统

- 支持 `API` 调用 `GPT2`、`Qwen` 等模型；
- 使用滑动窗口机制管理上下文，提高对话连贯性；
- 构建统一接口，便于模型切换与结果分析。

（2）偏见检测模块

- 融合规则方法（`HateCheck` 词典、正则匹配）与监督学习方法（`BERT` 分类器）；
- 支持类别标注（如性别歧视、政治攻击等）；
- 可扩展自定义风险等级和处理策略。

（3）模型微调优化

- 使用 `transformers + PEFT` 框架，基于 `LoRA` 对 `Qwen-7B-Chat`、`GPT2` 等模型进行定向微调；
- 支持训练后部署，作为替换生成模型；
- 微调目标：减少偏见表达、保持语义一致性、响应快速。

（4）模型评价体系

① **客观指标**：

- `BERTScore`：判断语义一致性；
- `Toxicity Score`：使用 `Perspective API` 评估有害程度；
- `Perplexity`：评估生成流畅性；

② **主观指标**：

- 小组成员进行主观偏见度打分。

五、开发环境

工具类别	工具名称
系统环境	`Windows 11`
编程语言	`Python 3.10`
模型与训练	`HuggingFace Transformers, PEFT, PyTorch`
模型资源	`Qwen-7B, ChatGPT2`
数据集	`BiasBench, BOLD, RealToxicityPrompts, CBBQ`
可视化工具	`matplotlib`

六、组员分工与阶段性检查点

成员	分工内容	检查点1	检查点2	检查点3
曹阳	对话系统构建、`API`接口集成	实现与多个大模型的对话系统	实现上下文窗口管理	测试对话系统，得到可能生成偏见性内容的提问数据集
高一鸣	偏见检测模块实现、数据处理与训练	偏见有关数据集的处理	偏见检测模块的实现	偏见检测模块的性能评价（模块评估与参考评估对比）
邸旭	微调大模型，评估指标实现（客观+主观）	主观评价指标的建立	为每个句子标注参考评估分数	微调大模型并在提问数据集上重新测试结果

七、时间安排

1. 教学周第13周：完成对话系统实现
2. 教学周第14周：完成偏见识别模型
3. 教学周第15周：完成 `LoRA` 微调与对比实验设置
4. 教学周第16周：完成最终整合，答辩 `ppt` 与展示文稿