指令遵循BadCase自动生成方案

如何设计指令遵循的评测题目,寻找AI模型的盲区

核心挑战:为什么AI模型会失败?

问题本质

- **单一约束易满足:** AI模型训练充分
- **多重约束易遗漏:** 注意力分散
- **隐式依赖难发现:** 逻辑推理不足
- **边界条件难把握:** 泛化能力局限

设计目标

让参赛模型(深度思考模式)做不对,且错误明确、公认

核心原则:约束不能冲突

❌ 反例子: 逻辑矛盾

约束1 输出必须在100字以内

约束2 详细展开每个要点,每个不少于80字

✅ 好例子: 兼容且复杂

约束1 输出必须在500-800字之间

约束2 包含5-7个段落

约束3 每段100-150字

多种约束组合策略:攻破不同盲区

并列约束 (Parallel)

同时施加多个独立约束,测试模型能否全面关注。

📝 案例: 撰写产品文案

✓ 格式: Markdown + 表格
✓ 长度: 500-800字
✓ 风格: 商务正式
✓ 内容: 包含5个关键词
✓ 结构: 引言+主体+结论

→ 攻破"注意力盲区"

易遗漏1-2个约束

链式约束 (Chain)

设计有依赖关系的约束序列,测试模型能否按步骤执行。

📝 案例: 分析报告

① 分析问题 → 输出A
② 基于A研究方案 → 输出B
③ 基于B设计方案 → 输出C
④ 基于C评估 → 输出D
⑤ 基于D提建议

→ 攻破"流程盲区"

易跳步骤或混淆依赖

条件约束 (Conditional)

设计条件分支,测试模型能否正确判断并执行。

📝 案例: 内容创作

IF 技术类:
→ 代码+性能分析+架构图
ELSE IF 人文类:
→ 通俗语言+举例+避术语
IF 超1000字:
→ 添加目录+摘要

→ 攻破"分支推理盲区"

易误判条件或错误分支

整体架构: 从Seed初始化到BadCase生成

阶段1: Seed初始化

- **主题选择:** 法律/技术/教育...
- **约束类型:** 并列/链式/条件/嵌套
- **初始约束:** 生成0-3个基础约束

↓

AskAgent
设计初始问题

→

阶段2: 多轮对话

- **Target LLM** 回答问题
- **JudgeAgent** 评估结果
- **追问机制** 增加难度

↓

循环最多3轮
直到出错或达上限

→

阶段3: 保存结果

- **完整对话:** llm_messages
- **错误分析:** violations详情
- **能力边界:** 失败轮次记录

↓

BadCase数据集
高质量评测题目

Seed模版示例

```
主题: "法律咨询"
约束类型: 链式约束
初始约束:
  - 格式: 分段回答
  - 长度: 300-500字
  - 包含: 法律依据引用
```

设计原则

✓约束之间不能冲突
✓从简单开始,逐步加码
✓覆盖多种约束类型组合

核心创新: 自适应追问机制 - 渐进式逼近能力边界

❌ 传统方法: 一次性超难

设计超难题目
10个约束 + 复杂依赖

▼

Target LLM 回答

▼

结果不理想
要么太难 要么太简单

核心问题: 难度不可控,无法精确定位模型能力边界

✅ 我们的方法: 渐进式刁难

轮次1: 基础约束
2-3个简单约束

✓通过 → ▼

轮次2: +格式约束
增加2-3个格式要求

✓通过 → ▼

轮次3: +复杂依赖
增加链式/条件约束

✗失败 → ▼

💾 保存BadCase
精确定位能力边界!

核心优势: 逐轮增加难度,找到模型"恰好失败"的临界点

核心洞察:为什么这套方案有效?

认知负载理论

通过并列、链式、条件等约束组合,逐步突破模型的注意力容量和工作记忆极限。

对抗性样本思维

从设计单个"极难"问题,转变为通过"自适应刁难"来逐步逼近并精确定位模型的能力边界。

可解释性优势

明确的约束 + 结构化评估 = 可解释的错误。不再是"感觉不好",而是"违反了哪条具体规则"。

总结

系统化设计: 通过多种约束组合,全面探测模型在注意力、流程和推理上的盲区。

自动化生成: 运用Agent架构,实现"设计-测试-评估-追问"的闭环,高效产出BadCase。

自适应难度: 核心的"多轮刁难"机制能精准定位模型的能力边界,生成有区分度的评测题目。

Q&A

感谢聆听