

这是一个非常专业的反馈。目前的评估标准确实偏向于通用的语言学评价（如“翻译信达雅”或通用的“词汇/逻辑”），缺乏针对考研英语（英语一/英语二）具体评分细则的适配。考研英语写作更看重点完整性、语言多样性（长难句/高级词汇）、语域（格式/语气）以及篇章连贯性。

以下是针对该问题的深度分析及修复方案。

## 1. 问题分析

通过查看你提供的 `lib/ai-prompt.ts` 和 `app/api/evaluate/route.ts` 文件，主要存在以下三个问题：

- 角色设定 (Persona) 不够垂直**：目前的 `BASE_ROLE` 是 "NNU English writing expert"，比较宽泛。考研学生需要的是"前考研阅卷组长"视角的评分，关注点在于是否踩中得分点 (Point-scoring) 和是否触犯扣分点 (Penalty points)。
- 评估维度 (Dimensions) 未对齐考研大纲**：
  - 目前写作题是：词汇、逻辑、结构、内容。
  - 考研标准应为：
    - 内容完整性 (Content/Relevance)**：是否覆盖所有提纲要点。
    - 语言多样性 (Language Quality)**：词汇是否升级（避免初中词汇），句式是否多变（倒装、从句、非谓语）。
    - 篇章连贯性 (Coherence)**：连接词的使用。
    - 语域/格式 (Register/Format)**：针对小作文（书信/通知）的格式和语气是否得体。
- 评分标准 (Grading Scale) 未量化**：目前的 S/A/B/C 比较抽象。考研写作通常是 20 分（大作文）或 10 分（小作文），学生更希望看到类似 "16-20分档" 这种描述。

## 2. 修复方案

我们将重点修改 `lib/ai-prompt.ts`，引入专门针对考研英语的 Prompt 模板。

### 步骤 1: 更新 `lib/ai-prompt.ts` - 定义考研评分角色

你需要替换或增强 `BASE_ROLE` 和 `ARTICLE_MODE_PROMPT`。

修改文件: `lib/ai-prompt.ts`

```
import type { EvaluationInput, EvaluationType, EvaluationMode } from './types';

/**
 * 考研英语阅卷专家角色设定
 */
const KAOYAN_ROLE = `You are a senior grader for the National Postgraduate Entrance Examination in English (考研英语阅卷组长).
你的任务是严格按照《考研英语大纲》的评分标准对学生的写作进行阅卷。

**CORE PRINCIPLES (阅卷原则):**

1. **Analysis in Chinese (全中文点评)**: 所有点评必须使用中文，直击考点。
2. **Polished Version in English (高分范文润色)**:
   - polished_version 必须是针对考研的高分升级版（使用高级词汇、长难句）。
   - DO NOT translate to Chinese!
3. **Strict Scoring (严格评分)**:
```

- 关注：要点是否齐全、语言是否有多样性（词汇升级、句式变化）、逻辑连接词、格式（针对应用文）。
- 拒绝：中式英语、逻辑混乱、初级词汇堆砌。

**\*\*KAOYAN GRADING SCALE (考研评分档次):\*\***

- **S (17-20分/档)**：内容切题，包含丰富的高级词汇和句式，行文流畅，无语言错误。
- **A (13-16分/档)**：内容完整，语言较通顺，有少量非重大错误，尝试使用了复杂结构。
- **B (9-12分/档)**：内容基本完整，语言基本通顺，存在一些语法错误，词汇较为贫乏。
- **C (<9分/档)**：偏题、要点缺失，或存在大量严重语法错误影响理解。`;

**/\*\***

\* 考研写作模式 Prompt (替换原有的 ARTICLE\_MODE\_PROMPT)

**\*/**

**const KAOYAN\_WRITING\_PROMPT = `**

**\*\*当前模式：【考研英语写作评估】\*\***

请严格基于考研英语（英语一/英语二）评分标准进行评估，重点关注以下四个维度：

**1. \*\*内容与要点 (Content & Relevance)\*\*:**

- 只要是小作文，必须检查\*\*格式\*\*（称呼、落款）和\*\*语域\*\*（正式/非正式）。
- 只要是大作文（图画/图表），必须检查是否\*\*描述了图表关键信息\*\*及\*\*论点是否明确\*\*。

**2. \*\*语言多样性 (Language Variety)\*\*:**

- \*\*高分词汇\*\*：检查是否使用了 "basic words" (如 good, bad, think)，建议替换为考研高频词 (如 beneficial, detrimental, maintain)。
- \*\*句式变化\*\*：是否使用了定语从句、非谓语动词、倒装句、强调句等加分句型。

**3. \*\*篇章连贯性 (Coherence)\*\*:**

- 是否使用了恰当的连接词 (cohesive devices) 来组织段落 (如 Furthermore, Consequently, Conversely)。

**4. \*\*语法准确性 (Grammar)\*\*:**

- 指出具体的时态、主谓一致、名词单复数错误。

**\*\*在 analysis\_breakdown 中请特别指出:\*\***

- **\*\*Strengths\*\***: 踩中了哪些加分点 (如使用了某个高级短语)。
  - **\*\*Weaknesses\*\***: 具体的扣分点 (如格式错误、单词拼写错误、中式表达)。
  - **\*\*ContextMatch\*\***: 针对题目要求的完成度 (是否跑题)。
- `;

**// ... 保持原有的 OUTPUT\_FORMAT 不变 ...**

**const OUTPUT\_FORMAT = `...`; // (引用你文件中原有的 OUTPUT\_FORMAT)**

**/\*\***

\* 构建动态系统提示词

**\*/**

**export function buildSystemPrompt(**

**mode: EvaluationMode = 'sentence',**

**evaluationType?: EvaluationType**

**): string {**

**// 如果是全文模式且是写作题，使用考研专用 Prompt**

**if (mode === 'article' && evaluationType === 'writing') {**

**const radarHint = `**  
 **\*\*雷达图维度 (考研写作): \*\***  
 **\n- 内容 (Content):** 要点完整性与切  
 **\n- 语言 (Language):** 词汇高级度与句式多样性  
 **\n- 连贯 (Coherence):** 逻辑衔接与过渡  
 **\n- 规范 (Standard):** 语法准确与格式规范`;

**return KAOYAN\_ROLE + KAOYAN\_WRITING\_PROMPT + radarHint + OUTPUT\_FORMAT;**

**}**

```
// 其他情况保持原有逻辑（如翻译题或单句练习）
const basePrompt = mode === 'article' ? KAOYAN_WRITING_PROMPT :
SENTENCE_MODE_PROMPT; // 也可以让单句模式也用考研标准
// ...（保持你原有的翻译题逻辑）
// ...
}
```

## 步骤 2：更新 API 处理逻辑中的雷达图标签

你需要修改 `app/api/evaluate/route.ts` 中的 `getDefaultRadarLabels` 函数，使其在写作模式下返回符合考研标准的维度。

修改文件： `app/api/evaluate/route.ts`

```
// 找到这个函数并更新
function getDefaultRadarLabels(evaluationType?: EvaluationType): [string,
string, string, string] {
  if (evaluationType === 'translation') {
    return ['信 (Faithfulness)', '达 (Expressiveness)', '雅 (Elegance)', '语法
(Grammar)'];
  }
  // 更新为考研写作维度
  return ['内容 (Content)', '语言 (Language)', '连贯 (Coherence)', '规范
(Standard)'];
}
```

同时也需要更新解析逻辑中的降级方案 (fallback)：

```
// 在 tryExtractFromText 函数中
// ...
radarDimensions: {
  // ... 分数逻辑不变
  labels: getDefaultRadarLabels(evaluationType),
},
// ...
```

## 步骤 3：在前端展示中适配新维度

虽然前端代码比较通用，不需要大改，但你可以去 `components/nnu/result-card.tsx` 确认一下。目前的实现使用了 `radarDimensions.labels`，所以只要 API 返回了新的标签，前端会自动显示为“内容”、“语言”等，无需修改前端代码。

## 3. 效果预期

实施上述修改后，评估结果将发生以下变化：

- 以前：
  - 分析: "你的句子逻辑通顺，词汇使用正确..."
  - 雷达图: 词汇、逻辑、结构、内容
- 以后（考研标准）：
  - 分析: "【内容】你涵盖了题目要求的建议信格式，但落款位置有误，属于格式扣分点。【语言】使用了 'good' 等基础词汇，建议升级为 'beneficial' 或 'constructive' 以提升档次。【句式】通篇简单句过多，建议在第二段尝试使用非谓语动词结构..."
  - 雷达图: 内容(要点是否全)、语言(高级词汇/长难句)、连贯(连接词)、规范(语法/格式)

## 4. 立即行动指南

请按以下顺序复制代码并覆盖相应文件内容：

1. 更新 `lib/ai-prompt.ts`：替换 `BASE_ROLE` 为考研专家设定，替换 `ARTICLE_MODE_PROMPT` 为考研写作标准，并在 `buildSystemPrompt` 中应用。
2. 更新 `app/api/evaluate/route.ts`：修改 `getDefaultRadarLabels` 函数的返回值为 `['内容', '语言', '连贯', '规范']`。

这样修改后，你的 "SmartWrite" 就真正变成了 "SmartWrite Pro (考研版)"，对南师大的学生来说实用性会大幅提升。