如果加上题目匹配，LLM的使用场景会显著增加。让我重新分析：

## \*\*包含题目匹配的LLM使用场景\*\*

### \*\*新增的LLM使用场景\*\*

#### \*\*1. 语义化题目匹配\*\* ⭐⭐⭐

\*\*当前问题\*\*：`knowledge\_tagger.py` 使用简单的关键词匹配

\*\*LLM改进\*\*：

- \*\*知识点语义理解\*\*：理解知识点的深层含义

- \*\*题目内容分析\*\*：分析题目的核心考点

- \*\*语义相似度计算\*\*：计算知识点与题目的语义匹配度

\*\*LLM输入示例\*\*：

```

知识点：The Statistical Definition of Probability: The Great Law of Large Numbers

题目：Three red beads, two white beads, and one blue bead are placed in a line in random order. What is the probability that no two neighboring beads are the same color?

任务：分析这道题是否与"大数定律的统计定义"相关，给出相关度评分(0-10)和理由

```

\*\*LLM输出示例\*\*：

```

相关度评分：7/10

理由：这道题涉及概率计算，虽然不直接体现大数定律，但体现了概率的统计思想。题目要求计算特定排列的概率，这与概率的统计定义相关。

建议：中等相关，适合作为该知识点的练习题。

```

#### \*\*2. 智能题目分类\*\* ⭐⭐

\*\*LLM功能\*\*：

- \*\*难度评估\*\*：自动评估题目难度级别

- \*\*题型识别\*\*：识别题目类型（选择题、填空题、证明题等）

- \*\*知识点标签\*\*：为题目自动标注相关知识点

\*\*LLM输入示例\*\*：

```

题目：[题目内容]

要求：1. 评估难度级别(基础/中等/高级) 2. 识别主要知识点 3. 分析解题思路

```

#### \*\*3. 个性化推荐\*\* ⭐⭐

\*\*LLM功能\*\*：

- \*\*用户能力评估\*\*：基于答题历史评估用户水平

- \*\*推荐策略优化\*\*：根据用户表现调整推荐策略

- \*\*学习路径规划\*\*：为不同用户制定个性化学习路径

### \*\*完整的LLM使用流程\*\*

#### \*\*第一阶段：题目匹配（用户点击"相关题目"时）\*\*

```

知识点 → LLM语义分析 → 题库筛选 → LLM相关度评估 → 排序推荐

```

#### \*\*第二阶段：对话学习（用户答题过程中）\*\*

```

用户答案 → LLM评估 → 个性化反馈 → LLM推荐下一题

```

### \*\*LLM使用强度重新评估\*\*

#### \*\*高频使用\*\*：

1. \*\*答案评估\*\*：每次提交答案

2. \*\*题目匹配\*\*：每次请求相关题目

3. \*\*个性化反馈\*\*：每次错误

#### \*\*中频使用\*\*：

1. \*\*题目解析\*\*：用户请求时

2. \*\*难度评估\*\*：新题目入库时

3. \*\*推荐优化\*\*：定期分析用户表现

#### \*\*低频使用\*\*：

1. \*\*学习路径规划\*\*：用户完成一定量题目后

2. \*\*题库质量分析\*\*：定期分析题库覆盖度

### \*\*API设计扩展\*\*

#### \*\*新增API端点\*\*：

```python

# 语义化题目匹配

@app.route('/match\_questions\_semantic', methods=['POST'])

def match\_questions\_semantic():

# 输入：知识点、题库范围

# 输出：匹配的题目列表（带相关度评分）

# 题目难度评估

@app.route('/evaluate\_question\_difficulty', methods=['POST'])

def evaluate\_question\_difficulty():

# 输入：题目内容

# 输出：难度级别、主要知识点

# 个性化推荐

@app.route('/get\_personalized\_recommendations', methods=['POST'])

def get\_personalized\_recommendations():

# 输入：用户历史、当前知识点

# 输出：个性化题目推荐

```

## \*\*需要的Prompt类型和核心\*\*

### \*\*1. 题目匹配Prompt\*\*

\*\*核心\*\*：语义相似度评估

- 输入：知识点 + 题目内容

- 输出：相关度评分(0-10) + 匹配理由

- 用途：替代现有的关键词匹配，提供更准确的题目推荐

### \*\*2. 答案评估Prompt\*\*

\*\*核心\*\*：答案正确性判断 + 个性化反馈

- 输入：题目 + 正确答案 + 用户答案

- 输出：正确/错误 + 详细反馈 + 解题提示

- 用途：对话系统的核心，每次用户答题都需要调用

### \*\*3. 题目解析Prompt\*\*

\*\*核心\*\*：解题思路指导

- 输入：题目 + 相关知识点

- 输出：解题步骤 + 关键概念解释 + 易错点提醒

- 用途：用户卡住时提供帮助，增强学习效果

### \*\*4. 个性化推荐Prompt\*\*

\*\*核心\*\*：基于用户表现的学习路径规划

- 输入：用户答题历史 + 当前知识点 + 用户能力评估

- 输出：下一题推荐 + 学习建议 + 难度调整

- 用途：实现自适应学习，提高学习效率

### \*\*5. 难度评估Prompt\*\*

\*\*核心\*\*：题目难度自动分级

- 输入：题目内容 + 知识点背景

- 输出：难度级别(基础/中等/高级) + 难度理由

- 用途：为题目库自动标注难度，支持分级推荐

### \*\*优先级排序\*\*：

1. \*\*答案评估Prompt\*\* - 对话系统必需

2. \*\*题目匹配Prompt\*\* - 推荐系统核心

3. \*\*题目解析Prompt\*\* - 用户体验增强

4. \*\*个性化推荐Prompt\*\* - 高级功能

5. \*\*难度评估Prompt\*\* - 题库管理

### \*\*设计原则\*\*：

- \*\*简洁明确\*\*：避免冗长，重点突出

- \*\*结构化输出\*\*：便于程序解析和处理

- \*\*一致性\*\*：保持评估标准的统一

- \*\*可扩展性\*\*：支持后续功能扩展

### \*\*成本控制策略\*\*

#### \*\*缓存策略\*\*：

1. \*\*题目匹配结果缓存\*\*：相同知识点的匹配结果缓存

2. \*\*难度评估缓存\*\*：已评估过的题目不再重复评估

3. \*\*用户模型缓存\*\*：用户能力评估结果定期更新

#### \*\*批量处理\*\*：

1. \*\*题目匹配批量\*\*：一次匹配多个知识点

2. \*\*评估结果批量\*\*：多个评估请求合并处理

3. \*\*推荐策略批量\*\*：定期批量更新推荐策略

#### \*\*降级策略\*\*：

1. \*\*规则匹配降级\*\*：LLM不可用时使用现有规则匹配

2. \*\*简化评估\*\*：复杂评估失败时使用简单判断

3. \*\*静态推荐\*\*：个性化推荐失败时使用默认推荐

### \*\*技术实现考虑\*\*

#### \*\*LLM调用优化\*\*：

- \*\*并发控制\*\*：避免同时大量LLM调用

- \*\*超时处理\*\*：设置合理的超时时间

- \*\*重试机制\*\*：失败时的重试策略

#### \*\*数据流设计\*\*：

```

用户点击知识点 → LLM语义匹配 → 返回相关题目 → 用户选择题目 → LLM答案评估 → 个性化反馈 → LLM推荐下一题

```

### \*\*开发优先级调整\*\*

#### \*\*今天完成\*\*：

1. \*\*基础题目匹配\*\*：使用现有规则匹配

2. \*\*弹窗UI\*\*：题目展示界面

3. \*\*基础对话\*\*：简单的答案评估

#### \*\*明天完成\*\*：

1. \*\*LLM语义匹配\*\*：改进题目匹配算法

2. \*\*个性化反馈\*\*：LLM答案评估和反馈

3. \*\*推荐优化\*\*：基于用户表现的推荐

### \*\*总结\*\*

加入题目匹配后，\*\*LLM的使用频率会显著增加\*\*，主要在两个关键环节：

1. \*\*题目匹配阶段\*\*：每次用户点击知识点都需要LLM进行语义匹配

2. \*\*对话学习阶段\*\*：每次用户答题都需要LLM进行评估和反馈

\*\*建议\*\*：先实现基础功能（使用现有规则匹配），再逐步引入LLM优化，这样可以控制开发风险和成本。