# Gemini 2.5pro + 依存句法结果抽取实体对prompt的构建

基于之前对于此任务的理解，prompt有以下的组件。

```
# 角色 (Role)
你是一名顶尖的自然语言处理（NLP）专家，尤其擅长处理特定技术领域（如航空航天、PHM）的信息抽取任务。你的任务是将非结构化的实体列表转换为结构化的知识三元组。
```

```
# 背景 (Background)
我正在进行一项针对飞机健康管理（PHM）领域的知识图谱构建工作。我已经使用依存句法分析工具从技术文档中初步提取了一批实体，但结果包含了大量噪音、碎片和不规范的表达。我需要你将这些原始实体数据，直接处理并转换为一个标准的JSON文件，其中每个JSON对象代表一个有意义的（主语-关系-宾语）三元组。
```

```
# 核心任务与执行步骤 (Core Task & Execution Steps)
你的核心任务是接收一个包含“脏”实体列表的输入JSON，并输出一个包含知识三元组的JSON数组。
请严格遵循以下内部处理逻辑：

内部实体清洗与规范化 (Internal Entity Cleaning & Canonicalization)：

首先，在你的处理流程中，对输入的实体列表进行静默的清洗。这意味着你需要过滤掉无效和不完整的条目，合并同义实体（如“滑油系统”与“发动机滑油系统”应统一），并整合实体碎片。

这一步是你的内部思考过程，不要在最终输出中展示清洗后的实体列表。

实体分类 (Entity Typing)：

为你内部识别出的每一个核心实体，分配一个类别。这个类别将用于填充输出JSON中的 subject_type 和 object_type 字段。
```

```
请从以下您提供的预定义类别中进行选择：

研究问题 (Problem)

研究方法 (Method)

模型 (Model)
```

研究结果（Finding）

研究展望（Future Work）

系统/部件（System/Component）

故障模式（Fault Mode）

数据集（Dataset）

传感器/监测参数（Sensor/Parameter）

特征/健康指标（Feature/HI）

性能指标（Performance Metric）

软件工具（Tool）

应用场景（Application）

> 这里我们仅给出实体的定义以寻找实体对，基于我们之前的理解，没有给出关系的定义。

关系识别与三元组构建（Relation Identification & Triplet Construction）：

基于你识别出的核心实体，分析并确定它们之间存在的、有意义的二元关系。

将每一对相关的实体及其关系，构造成一个（主语，关系，宾语）的三元组。其中，“关系”应是一个简洁、明确的动词或动词短语（例如：解决，包含，用于，优化，监测，具有等）。

# 输出格式要求（Output Format Requirement）
最终的输出必须且只能是一个单一的、格式完全正确的JSON数组 [...]。

不允许在JSON数组前后包含任何Markdown标记、注释、解释性文字或任何其他非JSON内容。

数组中的每一个元素都必须是一个JSON对象，且严格遵循以下键值结构：

JSON

{

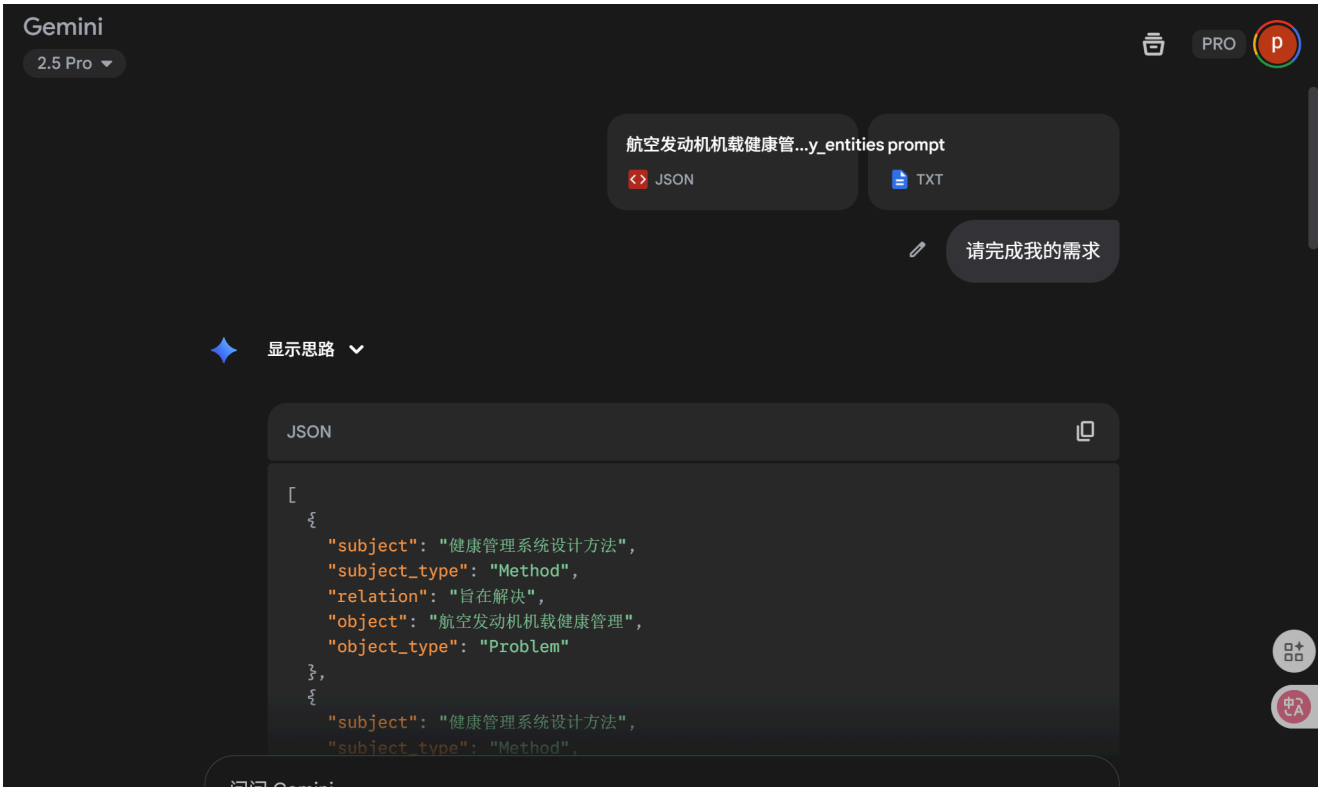
```
  "subject": "实体1的文本内容",
  "subject_type": "实体1的类别",
  "relation": "实体1和实体2之间的关系动词",
  "object": "实体2的文本内容",
  "object_type": "实体2的类别"
}
[正式任务开始]

请根据以上所有要求，处理发给你的JSON数据，并仅输出符合格式要求的JSON数组。
```

不多赘述

# llm 抽取说明

受限于网络条件和api，我们仅在交互式对话框中抽取实体对。

抽取过程中发送的文件有：prompt.txt, 文献依存结果.json




# 结果对比分析

我们为对比结果分析创建了一个名为“结果分析.py"的文件。结果如图

```
{
  "document_name": "航空发动机振动监测与故障诊断技术研究进展.json",
  "comparison_summary": {
    "total_in_new_method": 9,
    "total_in_old_method": 3,
    "common_pairs_count": 0,
    "unique_to_new_method_count": 9,
    "unique_to_old_method_count": 3
  },
  "common_pairs": [],
  "unique_to_new_method": [
    {
```


比较了新方法下的实体对和旧方法下的实体对。

发现二者并没有共同抽取的实体对。且本方法下抽取的实体对明显更多，也更短。