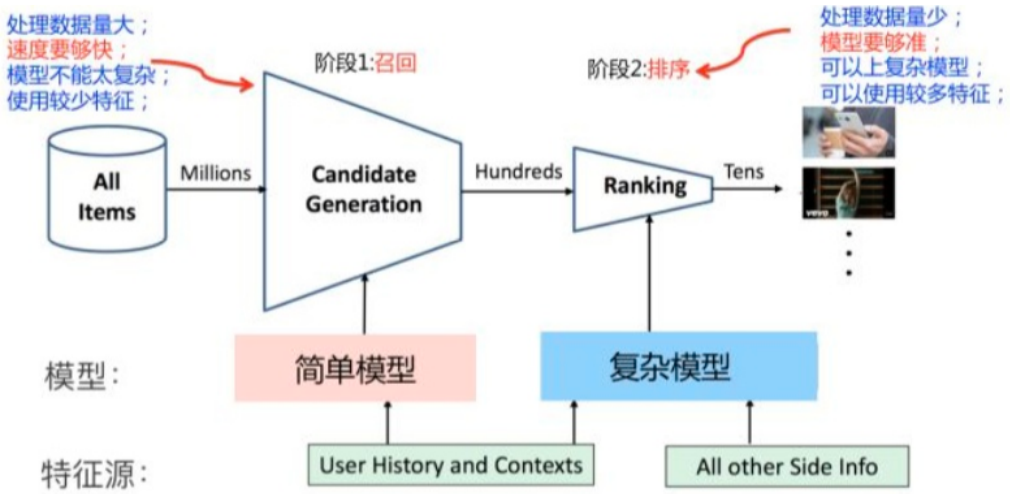
提示词工程考试说明

背景



一个简化的推荐系统流程示意图

在大数据时代，推荐系统已成为各大互联网平台不可或缺的核心技术。推荐系统通过分析用户行为数据和内容特征，为用户提供个性化的内容推荐，有效解决了信息过载问题，提升了用户体验和平台价值。随着大语言模型(LLM)技术的发展，其在推荐系统领域展现出独特优势，包括强大的文本理解和语义分析能力、跨模态理解能力、零样本/少样本学习能力、较强的可解释性以及良好的通用性，这些特点使大语言模型能够为推荐系统带来新的解决方案和性能提升。传统推荐系统的基本流程主要包括数据收集、召回、排序、重排和展示等环节。其中，召回阶段负责从海量候选集中快速筛选出潜在相关物品；排序阶段对召回结果进行精细排序。在多种推荐场景中，序列推荐是一种重要形式，它考虑用户行为的时序信息，基于用户的历史行为序列预测用户的下一个行为，能够捕捉用户兴趣的动态变化，提高推荐的时效性和精准度。

提示词工程（Prompt engineering）作为一门新兴的技术领域，源于大语言模型的快速发展。它是指通过精心设计和优化输入提示（prompts），引导大语言模型生成符合预期的输出内容的方法与技术。随着模型如GPT、Claude等的普及，prompt engineering已从简单的问答交互发展为一种专业技能，涉及提示结构设计、上下文信息组织、任务分解和指令精确表达等多个方面。在推荐系统中，有效的prompt设计能够帮助大语言模型更准确地理解用户偏好，提取关键特征，并生成更符合用户需求的推荐结果，从而弥补传统推荐方法在语义理解和个性化表达上的不足。

考试内容

本次考核聚焦于利用提示词工程(Prompt Engineering)使大语言模型解决推荐系统中的重排任务。考生将获得一个验证集文件(附件：val.jsonl)，该文件中每行代表一条数据，是一个用户的历史电影观看记录。这些数据模拟了真实世界中推荐系统面临的场景，即根据用户历史行为预测用户下一步可能的兴趣点。在推荐系统中，重排(Re-ranking)是推荐流程的关键环节，它直接影响用户体验和推荐效果。

本次考核旨在考察考生运用提示词工程技术引导大语言模型完成这一重排任务的能力，考生需要分析用户的历史观影行为，从中挖掘用户偏好特征，并将这些信息合理地融入提示词中，使大语言模型能够对候选电影进行准确排序。

代码块

```
val.jsonl中，每一条数据的字段说明：
{
  "user_id": 5737,  // 用户编号
  "item_list": [  // 用户历史观看电影列表，按时间顺序排列，越靠后表示越近期观看
    [1836, "Last Days of Disco, The"],  // [电影ID，电影名称]
    [3565, "Where the Heart Is"],
    // ... 更多历史观看记录，长度不定
  ],
  "target_item": [1893, "Beyond Silence"],  // 用户实际观看的下一部电影 [电影ID，电影名称]
  "candidates": [  // 推荐系统召回阶段得到的候选电影列表
    [2492, "20 Dates"],
    [684, "Windows"],
    [1893, "Beyond Silence"],  // 包含用户实际观看的下一部电影
    // ... 更多候选电影，长度不定，一般为20个左右
  ]
}
```

考生需要通过提示词工程，设计合适的提示词(prompt)，使大语言模型完成对candidates的重排任务，输出一个按推荐强度排序的电影ID列表，推荐强度越高的电影排在越前面，表示更可能被用户喜欢。而想要评价模型的推荐重排性能，一般会采用NDCG@K(Normalized Discounted Cumulative Gain)指标评价。NDCG值范围为[0,1]，越接近1表示排序质量越高。

具体来说，给定一个预测的排序列表 $p = [p_1, p_2, ..., p_k]$ 和用户实际观看的下一部电影 g ，我们计算NDCG如下：

1. **相关性评分**：对预测列表中的每个电影 p_i ，如果 $p_i = g$ ，则相关性 $rel_i = 1$ ；否则 $rel_i = 0$
2. **折损累积收益(DCG)**： $DCG@k = \sum_{i=1}^{k} \frac{rel_i}{\log_2(i+1)}$
3. **理想折损累积收益(IDCG)**：在本任务中，由于只有一个相关项，理想情况是将其放在第一位 - 因此 $IDCG@k = \frac{1}{\log_2(1+1)} = 1$
4. **归一化折损累积收益(NDCG)**： $NDCG@k = \frac{DCG@k}{IDCG@k} = DCG@k$

代码块

```
# 计算单个样本的NDCG@k
def calculate_ndcg_for_sample(predicted_list, ground_truth_item, k=10):
```

```
    """
    计算单个样本的NDCG@k

    参数:
        predicted_list: 模型预测的电影ID排序列表 [id1, id2, id3, ...]
        ground_truth_item: 用户实际观看的下一部电影ID
        k: NDCG@k中的k取值

    返回:
        ndcg: NDCG@k分数
    """
    # 截取前k个预测结果
    predicted_list = predicted_list[:k]

    # 计算相关性分数列表
    relevance = []
    for item_id in predicted_list:
        if item_id == ground_truth_item:
            relevance.append(1)  # 相关项
        else:
            relevance.append(0)  # 不相关项

    # 计算DCG@k
    dcg = 0
    for i, rel in enumerate(relevance):
        # 位置i的折损因子为log2(i+2)
        discount = math.log2(i + 2)
        dcg += rel / discount

    # 计算IDCG@k
    # 在本任务中，理想情况是将唯一相关项放在第一位
    idcg = 1 / math.log2(1 + 1)  # = 1

    # 计算NDCG@k
    if idcg > 0:
        ndcg = dcg / idcg
    else:
        ndcg = 0

    return ndcg

# 使用示例
# 假设模型预测的电影ID排序为[111, 1893, 684, 2492, ...]
# 用户实际观看的下一部电影ID为1893
predicted_list = [111, 1893, 684, 2492, 3654, 2422, 176, 1629, 229, 3155]
ground_truth_item = 1893
ndcg = calculate_ndcg_for_sample(predicted_list, ground_truth_item, k=10)
```

```
print(f"NDCG@10 = {ndcg}")
# 输出: NDCG@10 = 0.63093
```

考核方式

考生需要使用OpenAI API接口形式（https://platform.openai.com/docs/guides/text?api-mode=chat）构造prompt，并完成以下两个函数：

函数一：构造提示词

代码块

```
def construct_prompt(d):
    """
    构造用于大语言模型的提示词

    参数:
    d (dict): jsonl数据文件的一行，解析成字典后的变量

    返回:
    list: OpenAI API的message格式列表，允许设计多轮对话式的prompt
    示例: [{"role": "system", "content": "系统提示内容"},
           {"role": "user", "content": "用户提示内容"}]
    """
    # 实现提示词构造逻辑
```

函数二：解析输出

代码块

```
def parse_output(text):
    """
    解析大语言模型的输出文本，提取推荐重排列表

    参数:
    text (str): 大语言模型在设计prompt下的输出文本

    返回:
    list: 从输出文本解析出的电影ID列表（python列表格式，列表的每个元素是整数，表示编号），表示重排后的推荐顺序
    示例: [1893, 3148, 111, ...]
    """
    # 实现输出解析逻辑
```

友情提示：一些平台如DeepSeek，火山引擎，硅基流动等注册会有部分免费的api额度，且此考核项目总体对api消耗需求较少。

提交要求

考生需要提交以下两个文件：

1. **Python文件**：（模版文件：template.py）
   - 文件中有且仅能包含上述两个函数
   - 不允许import第三方库，仅可import Python标准库（如random, re, json等）
2. **探索报告**：pdf文件，记录整个prompt优化探索过程，包含不同提示词策略的尝试、效果和分析。

提交方式：（截止时间 北京时间 5月10号 23:59 ，期间可以多次提交，会自动覆盖先前的提交文件）

1. 进入链接：https://send2me.cn/aNBQT7tn/Q--w3qgZaOspzg
2. 精确填写个人信息，包括报名号、姓名（由填写错误导致的得分缺失，后果自负！）
3. 上传两个需要提交的文件（一个.py文件，一个.pdf文件，命名不做要求，提交系统会自动转换）

评分标准

本项考核的分数由两部分组成：

1. **推荐性能客观得分**（占总分70%）
   - 所有考生在私有测试集(test.jsonl，格式与val.jsonl完全相同，但是考生不可见)上的NDCG@10性能进行排名并赋分
   - 测试统一使用DeepSeek-V3模型（使用官方api测试），temperature设为0，其他参数保持默认
   - 每个样本会进行多次采样取平均指标以保证结果稳定性
2. **提示词主观评价得分**（占总分30%）
   - 由专家老师基于指定评价准则进行评分
   - 评价内容包括提示词的创新性、合理性、可解释性等