为了方便起见，我们的团队将直接读取项目根目录文件下名如document\_test的文件中的所有文章（其所给数据与document.csv文件格式一样），并生成对应每篇文章的问答对。这些问答对将按照填空题、单选题、判断题和复杂问答题的不同类型进行生成。最终，所有的问答对将被保存在一个名为final\_qa.csv（与所给数据qa.csv格式内容一致）的文件中。在此过程中，我们将调用DeepSeek的deepseek-chat大模型来生成填空题、单选题、判断题，调用阿里的qwen2-72b-instruct大模型来生成复杂问答题。

1. **代码运行**
2. 安装代码所需的依赖

pandas 1.4.2

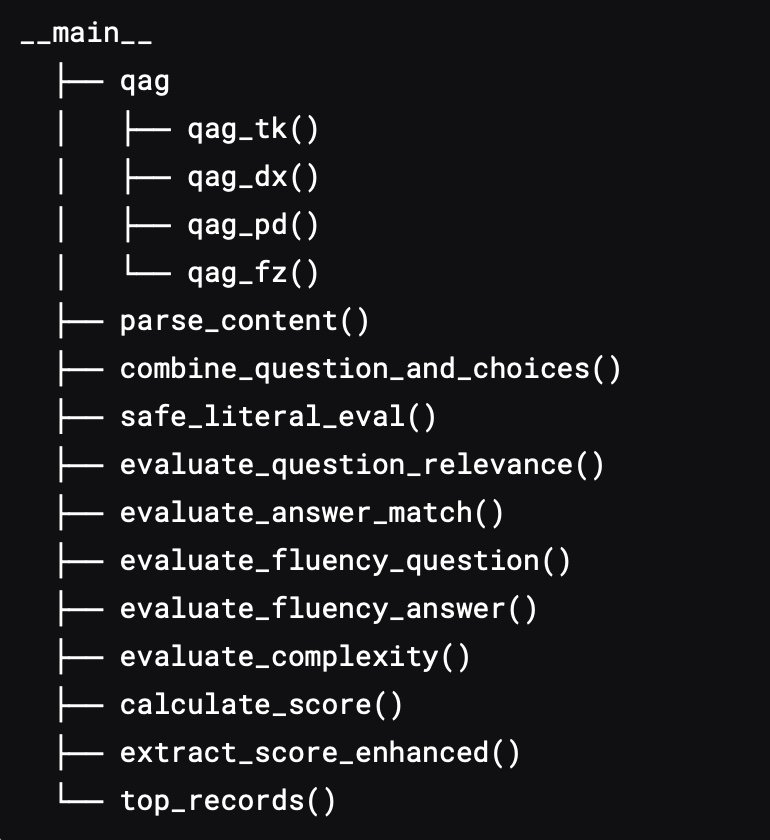
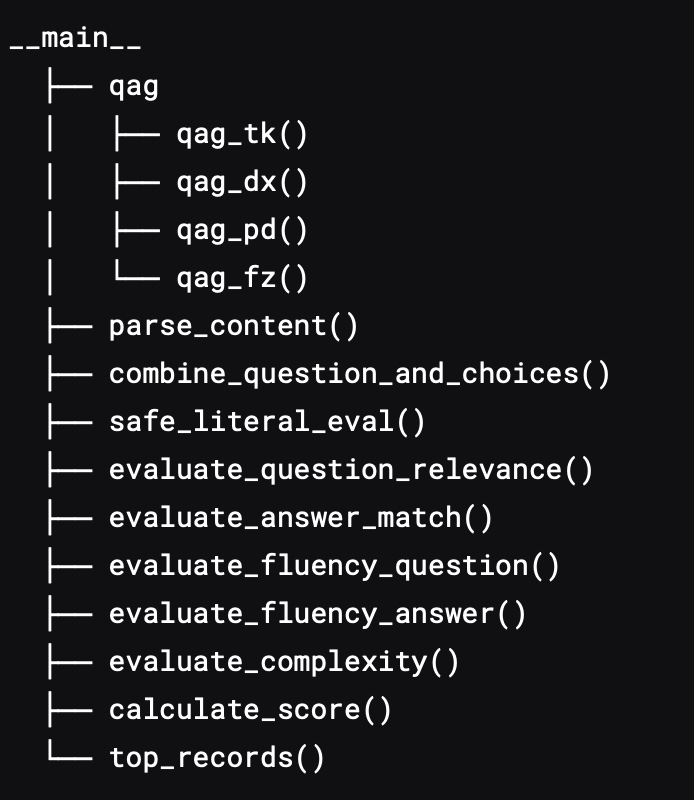
dashscope 1.19.2

1. 代码

代码文件：genqa.py

1. 运行代码
2. 将文章存入到输入文件中，文件名如 document\_test.csv。
3. 修改代码 genqa.py代码中443行 file\_path\_doc = os.path.join(current\_dir, 'document\_test.csv')的文件名跟所给文件名一样。
4. 运行代码genqa.py
5. 输出final\_qa.csv
6. **代码解释**

代码运行的主函数是\_\_main\_\_，调用的其它函数是qag\_tk(), qag\_dx(), qag\_pd(), qag\_fz(), parse\_content(), combine\_question\_and\_choices(), safe\_literal\_eval(), evaluate\_question\_relevance(), evaluate\_answer\_match(), evaluate\_fluency\_question(), evaluate\_fluency\_answer(), evaluate\_complexity(), calculate\_score(),extract\_score\_enhanced(), top\_records()



|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 作用 | 输入 | 输出 |
| qag\_tk() | 生成填空题问答对 | 文章内容（字符串） | 问答对列表，每个元素是一个字典，包含question\_type\_id, question, answer |
| qag\_dx() | 生成单选题问答对 | 文章内容（字符串） | 问答对列表，每个元素是一个字典，包含question\_type\_id, question, choices, answer |
| qag\_pd() | 生成判断题问答对 | 文章内容（字符串） | 问答对列表，每个元素是一个字典，包含question\_type\_id, question, answer |
| qag\_fz() | 生成复杂问答题问答对 | 文章内容（字符串） | 问答对列表，每个元素是一个字典，包含question\_type\_id, question, answer |
| parse\_content() | 将字符串形式的字典转换为实际字典 | 字符串形式的字典 | 实际字典 |
| combine\_question\_and\_choices() | 将问题和选项结合 | 包含问题和选项的字典 | 结合后的字符串 |
| safe\_literal\_eval() | 安全地将字符串形式的字典转换为实际字典 | 字符串形式的字典 | 实际字典或None |
| evaluate\_question\_relevance() | 评估问题与文章内容的相关性 | 文章内容，问题 | 相关性评分（0-100） |
| evaluate\_answer\_match() | 评估问题与答案的匹配度 | 文章内容，问题类型，问题，答案 | 匹配度评分（0-100） |
| evaluate\_fluency\_question() | 评估问题流畅度 | 问题类型，问题内容 | 流畅度评分（0-100） |
| evaluate\_fluency\_answer() | 评估答案流畅度 | 问题类型，答案内容 | 流畅度评分（0-100） |
| evaluate\_complexity() | 评估问题的复杂度 | 问题类型，文章内容，问题内容，答案 | 复杂度评分（0-100） |
| calculate\_score() | 计算每个问答对的评分 | 问题类型ID，S1评分，S2评分，S3评分，S4评分，S5评分 | 综合评分（Sq） |
| extract\_score\_enhanced() | 从文本中提取评分 | 包含评分的文本 | 评分（整数） |
| top\_records() | 获取Sq值最高的前两个或三个记录 | 包含Sq评分的数据分组 | 前两个或三个记录的DataFrame |