**1.3 软件功能与类间的对应关系**

根据对ragas开源软件代码的理解、标注和分析，软件功能与类之间的关

系如表2所示。

这张表格是ragas的软件功能与类间的关系对应关系表，它详细记录了其各项功能与其在代码实现中的具体类和方法之间的映射。通过这张表格，我们可以清晰地看到，每个功能点是如何通过特定的类和方法协同工作来实现的。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 功能名称 | 实现模块 | 实现方法 |
|  | 统一嵌入模型接口 | BaseRagasEmbeddings | embed\_query(), embed\_documents(), set\_run\_config() |
|  | 集成多种嵌入模型（Hugging Face、LangChain、LlamaIndex、Haystack） | HuggingfaceEmbeddings, LangchainEmbeddingsWrapper, LlamaIndexEmbeddingsWrapper, HaystackEmbeddingsWrapper | embed\_query(), embed\_documents(), aembed\_query(), aembed\_documents(), predict() |
|  | 支持异步嵌入计算，提高大规模文本处理效率 | LlamaIndexEmbeddingsWrapper, HaystackEmbeddingsWrapper | aembed\_query(), aembed\_documents() |
|  | 评估集成（支持 LangChain、LangSmith、LlamaIndex） | EvaluatorChain | evaluate\_run(),  evaluate() |
|  | Helicone 监控配置 | HeliconeSingleton | default\_headers(), |
|  | Opik 评估结果追踪 | OpikTracer | \_process\_start\_trace(), \_process\_end\_trace() |
|  | 数据格式转换（支持 LangChain、R2R、Swarm） | langgraph, r2r, swarm | convert\_to\_ragas\_messages(), transform\_to\_ragas\_dataset() |
|  | 统一 LLM 接口 | BaseRagasLLM | generate\_text(), agenerate\_text(), set\_run\_config() |
|  | 适配主流 LLM 框架（LangChain、LlamaIndex、Haystack） | LangchainLLMWrapper, LlamaIndexLLMWrapper, HaystackLLMWrapper | generate\_text(), agenerate\_text() |
|  | 模型集成（支持 OpenAI、Hugging Face、Azure 等） | LangchainLLMWrapper, LlamaIndexLLMWrapper, HaystackLLMWrapper | generate\_text(), agenerate\_text() |
|  | 统一模型调用 | llm\_factory | llm\_factory() |
|  | 数据集样本建模与验证 | BaseSample, SingleTurnSample, MultiTurnSample | validate\_user\_input(), to\_dict(), get\_features(), to\_string() |
|  | 数据集格式转换 | RagasDataset, EvaluationDataset | to\_hf\_dataset(), from\_hf\_dataset(), to\_pandas(), from\_pandas() |
|  | 评估结果计算与存储 | EvaluationResult | total\_tokens(), total\_cost(), upload(), to\_pandas() |
|  | 多轮对话交互验证 | MultiTurnSample | validate\_user\_input(), to\_messages(), pretty\_repr() |
|  | 指标注释管理 | MetricAnnotation, SampleAnnotation | from\_json(), from\_app(), filter(), train\_test\_split() |
|  | 分层抽样与批处理 | SingleMetricAnnotation | sample(), stratified\_batches(), batch() |
|  | 缓存系统管理 | CacheInterface, DiskCacheBackend | get(), set(), has\_key(), cacher() |
|  | 运行参数配置与重试机制 | RunConfig | add\_retry(), add\_async\_retry() |
|  | 异步任务执行与并发控制 | Executor | submit(), results(), run\_async\_batch(), as\_completed() |
|  | 成本与性能监控 | EvaluationResult, CostCallbackHandler | total\_cost(), total\_tokens() |
|  | 数据分块与进度跟踪 | Executor | \_process\_jobs(), \_process\_coroutines() |
|  | 模型运行追踪与可视化 | EvaluationResult | upload(), parse\_run\_traces() |
|  | 输入输出序列化 | BaseSample, EvaluationDataset | to\_dict(), from\_list(), to\_jsonl(), from\_jsonl() |
|  | 异常处理与日志记录 | Executor | wrap\_callable\_with\_index(), is\_event\_loop\_running() |
|  | 随机化与可复现性控制 | RunConfig | \_\_post\_init\_\_()（通过numpy.default\_rng初始化随机数生成器） |
|  | 基于遗传算法优化  prompt | ReverseEngineerPrompt  CrossOverPrompt  FeedbackMutationPrompt  FeedbackMutationPro  mptGeneration  GeneticOptimizer | optimize()  initialize\_population()  feedback\_mutation ()  evaluate\_candidate()  evaluate\_fitness()  cross\_over\_mutation() |
|  | prompt  管理 | PydanticPrompt  ImageTextPrompt  TranslateStatements  PromptMixin  FewShotPydanticPrompt  InMemoryExampleStore  StringPromp | generate()  add\_example()  get\_examples()  set\_prompts()  adapt\_prompts()  save\_prompts()  load\_prompts()  get\_image() |
|  | 知识图谱构建 | Node  Relationship  KnowledgeGraph | add\_property()  get\_property()  add()  save()  load()  remove\_node()  find\_two\_nodes\_sin  gle\_rel() |
|  | 生成mult\_hop测试样本 | TestsetGenerator  Testset  MultiHopQuerySynthesizer  ConceptCombinationPrompt  QueryAnswerGenerationPrompt | generate()  upload()  from\_annotated()  prepare\_combinations()  sample\_diverse\_combinations() |
|  | ⽣成single\_hop测试样本 | TestsetGenerator  Testset  SingleHopQuerySynthesizer  QueryAnswerGenerationPrompt | generate()  upload()  from\_annotated()  prepare\_combinations()  sample\_diverse\_combinations() |
|  | 构建知识图谱中节点间的关系 | CosineSimilarityBuilder  SummaryCosine  SimilarityBuilder  JaccardSimilarityBuilder  OverlapScoreBuilder | transform(）  filter() |
|  | 从⽂本中提取信息 | EmbeddingExtractor  RegexBasedExtractor  SummaryExtractorPrompt  KeyphrasesExtractorPrompt  TitleExtractorPrompt  NERPrompt  TopicDescriptionPrompt  ThemesAndConceptsEx  tractorPrompt | extract()  generate() |

**1.4收获**

在深入研究 Ragas 代码后，本组获得了以下关键经验和知识：

1. 深入理解架构设计：通过对 Ragas 代码的系统性分析，我们加深了对高效机器学习评估框架的理解。我们学习了其模块化架构如何支持不同评估指标的计算，以及如何优化大规模数据处理的性能。这使我们对复杂评估系统的设计有了更清晰的认识。
2. 掌握高质量代码规范：阅读 Ragas 代码的过程中，我们学习到了开源社区的代码风格、文档注释标准以及函数命名原则。这些实践不仅提升了代码的可读性，也为团队协作和长期维护提供了借鉴。
3. 强化代码分析与优化能力：通过研究 Ragas 的实现，我们提升了对机器学习评估方法的理解，掌握了如何解析不同模块之间的交互，评估代码的可扩展性和性能优化策略。这些技能对于未来优化模型评估流程、构建高效评测工具具有重要价值。

本次代码阅读让我们收获颇丰，不仅提升了技术能力，也拓宽了我们对评估框架设计的视野，为未来的开发实践奠定了坚实基础。