

原项目的 sentistrength 包中，所有类对应的 UML 类图如下图所示：



1. 类的职责比较杂乱；
2. 类的分层设计不合理。所有类放在一个包中，但这些类功能各异；
3. 分析代码得知，部分类中存在函数过于复杂、表达式过于复杂等情况。

## 2. 重构软件设计

### 2.1 设计逻辑

1. 将Sentistrength类读取待分析文本的五种算法拆出，设计为单独的策略类，方便算法替换和调用。
2. 将sentistrength类中打印HELP手册的方法重构到一个新的类中，用这个类统一管理所打印的HELP手册的内容以及打印的HELP手册类型。
3. 将sentistrength类中的机器学习算法拆解为单独的策略类，以便于后续对该算法进行调整，或是增设不同的学习算法，以供选择。
4. 基于单一职责原则，将分离出5个策略类的sentistrength类拆为两个类：
  - SentiArgs类：即项目的参数封装类；
  - ArgsAnalyzer类：即参数处理类。
5. 将ClassificationResources类initialise方法中的两个过长的if条件判断表达式提取为方法initLemmatiserOrNot和initPartRules，以便于代码阅读、理解和维护。
6. 将corpus类中的字典权重优化算法拆解为单独的策略类，便于后续对该算法进行扩展及调整，避免使用多重条件判断。
7. 将corpus类中有关输入文本处理的方法合并到NotWait策略类中，令corpus职责更加清晰，策略代码及方法调用更加紧密，便于代码理解与维护。

### 2.2 设计分层

#### 1. args包：

包含ArgsAnalyzer与SentiArgs类，由原先的SentiStrength类拆分而来，负责封装传入的参数列表及对参数进行初始化，分析及分类。

#### 2. classification包：

包含原有的ClassificationOptions, ClassificationResources, ClassificationStatistics, TextParsingOptions, UnusedTermsClassificationIndex类，负责控制十条规则的初始化，并作为工具类，weka类及规则类与corpus交互的中介。

#### 3. corpus包：

包含Corpus, Paragraph, Sentence, Term类，负责语料库的初始化，维护，运行及验证。提供情绪得分，分类结果及各项分类指标的具体实现。

#### 4. rules包：

包含BoosterWordsList, CorrectSpellingsList, EmoticonsList, EvaluativeTerms, IdiomList, IronyList, Lemmatiser, NegatingWordList, QuestionWords, SentimentWords类, 即十个规则的列表。

## 5. strategy包:

- a. optimiseDictionaryWeight包: 包含权重优化的策略及控制上下文
- b. readText包: 包含读取待分析文本的策略
- c. MachineLearning类: 包含机器学习有关方法

## 6. test包: 包含测试类。

## 7. SentiStrength类: 程序主类, 包含启动的方法。

## 8. ShowHelp类: 包含打印帮助文档和命令行中输入的命令参数选项说明的方法。

## 2.3 设计包图

