**Easynlp & Easytext 类间关系分析**

1. **模块分析**

**Easynlp**:（以官方给出的训练BERT模型为例）

**ClassficationDataset 数据加载与预处理模块：**用于对数据集进行处理，其属性包含了数据集的信息，方法用于处理和转换文本数据。

**Application训练框架模块：**这是训练框架的基类模块。

**Optimizer优化器模块：**用于更新和管理模型参数。

**Trainer训练控制模块：**这个类用于进行训练，其属性包含了训练模型时用到的参数，方法包括设置模型和优化器、从检查点恢复模型训练、设置训练数据加载器等模型训练常用操作。

**Evaluator性能评估模块：**用于评估模型的性能。

**Exporter参数转换模块：**用于在不同的深度学习框架之间迁移模型。

**Losses损失函数模块：**用于定义不同的损失函数。

**Easytext:**

**ConfigFactory** 是一个工厂类，用于创建和管理模型的配置。

**MetriccTracker** 模块主要用于跟踪每一个epoch的metric记录，保存best等指标。

**Model** 是一个包含预定义结构和参数的实体，用于从输入的数据中进行学习和预测。

**Loss** 用于评估预测结果和真是值之间的差异。

**ModelMetricAdapter** 适配器，使model和metric能一起工作。

**Optimizerfactory** 是一个工厂类，用于创建和配置优化器。

**LRSchedulerfactory** 是一个工厂类，用于创建和配置学习率调度器。

**Trainer** 用于执行训练，评估等任务。

1. **类间关系分析**

**Easynlp工作流程：（以训练BERT模型为例）**

初始化Easynlp库——解析用户定义的参数——获取预训练模型的路径——创建训练和验证数据集——获取应用模型——创建训练器实例（包含了模型、训练集以及用户定义的参数和评估器）——调用train方法进行训练。

****

图1 官方给出训练BERT模型代码

**Easynlp类间关系：**

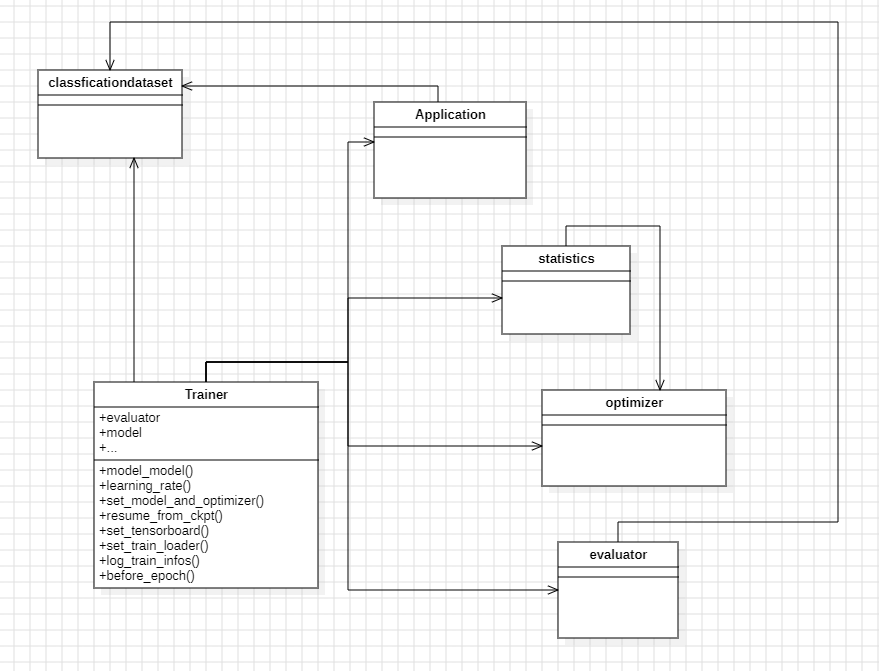


图2 easynlpuml图

模型的建立需要验证集中的数据；评估器也需要验证集中的数据进行评估；模型的训练需要训练集的数据。因此这三类都对数据处理类有依赖关系。优化器需要使用到训练过程中statistics获取到的数据，因此对其有依赖关系。训练类在训练过程中需要通过调用各种模块完成训练过程，因此它对其它的模块都有依赖关系其中的Application，classificationdataset和evaluator类都是来源于不同模型的类对基类的继承，这样就实现了在不同的模型上进行训练。

**Easytext工作流程：**

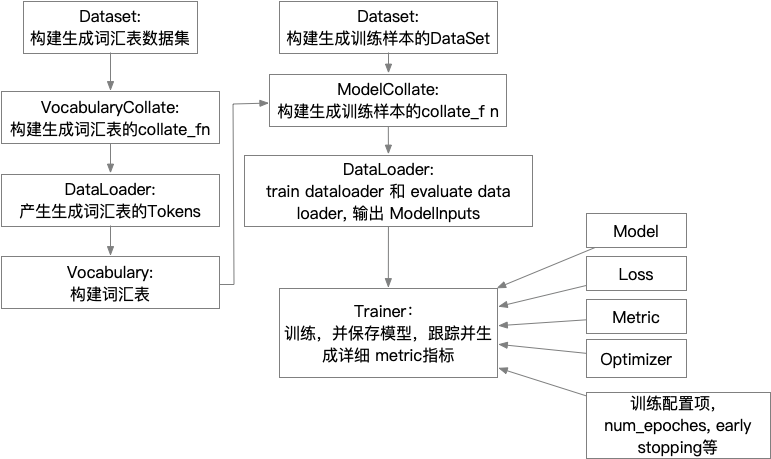
****

图3 easytext工作流程

与easynlp类似，也需要先解析用户自定义参数，构建数据集，然后实例化模型和训练器，再进行训练。

**Easytext类间关系：（以训练ACSA模型为例）**

在easytext里，每个模型有一个单独的训练模块，即Train类，它接受其它模块处理生成的字典、数据集以及各种参数，并实例化Loss、metric等模块最后调用共用的训练器对这些数据进行训练。各个模型也有其对应的数据集与字典等数据，这些数据分别被字典生成模块和数据集生成模块处理后发送给dataloader，在train中进行加载。其它的功能模块如metric，optimizer会在train类中获取数据及信息并被实例化。

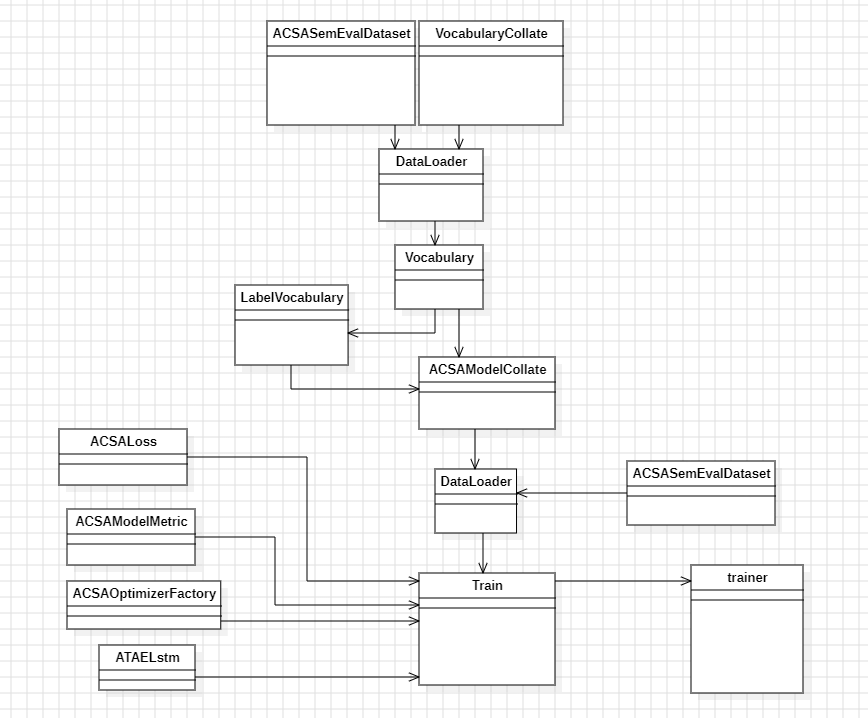
****

图4 easytext类间关系