密级文本分类设计文档

V0.0.1

# 1、项目目的

1. 文本自动分类功能（下面简称“分类功能”）可以对已经分好类的文档进行训练
2. 分类功能支持自动识别训练类别
3. 分类功能支持接口配置相关训练参数
4. 分类功能支持接口获取所有训练结果
5. 分类功能支持获取文本所属分类，平均准确率>90%
6. 分类功能提供Python接口供DGS调用
7. 文本编码格式为UTF-8

# 2、总体设计

系统核心算法框架包含四层：第一层文本表示层，导入文本信息并将文本表示成算法能友好识别的张量。第二层训练和验证调参层，对预处理数据进行训练并通过验证集调整参数。第三层，模型融合层，通过模型组合减少单模型偏差。第四层，数据测试层，测试模型效果。系统框架如下图所示。每层具体设计概要见第三节。



图1 系统框架

# 3 详细设计

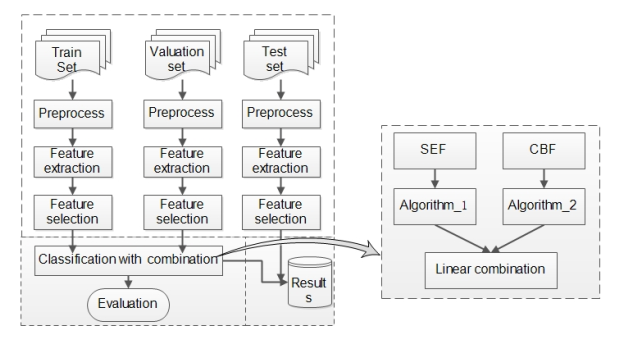


图2 算法流程图

3.1 文本表示层

文本表示层包含：文本预处理，词向量预处理，特征选择，特征提取，验证集划分。文本预处理需要对重复文本，空文本，文本长度过小做提取和删除，以保证算法的准确性。词向量预处理，将文本转换成计算机和算法能识别的向量，组成模型原始特征。特征选择，就是从原始特征中选择与文本密级特性最相关的特征。特征提取，从特征中提取新的特征。验证集划分，为下一层验证调参提供验证集。

3.2 训练和调参层

对单个算法，调用算法对文本进行训练，通过验证集测试模型，并调整算法参数，使模型训练效果达到最优。使用算法包括：Naive bytes, SVM, 神经网络等。

3.3 模型融合

由于单个模型的分类准确性与算法机制有一定关系，比如Naïve bytes 算法要求特征之间相互独立，这就失去了特征之间的关联，SVM通过划分空间来分类，其线性支持向量机在文本分类领域有很好的适应性。模型的组合可以减少单模型的偏差，提升模型在测试集上的准确率。我们把使用线性组合应用在定密系统中，实验结果表明模型准确率有一定的提升。

3.4 模型测试

对测试集进行测试并与真实标签对比，来评估模型的真实好坏。

# 4、程序接口描述

1. 设置训练源，path为具体的待分类文件夹

bool SetTrainingData(string Path)

输入为训练源的路径，输出为bool

1. 设置训练参数

bool SetTrainingConfig(TrainingConfig TrainingPath)

输入为训练参数，输出为bool

1. 开始训练

int StartTraining()

输入无，输出为int类型，0代表成功，其他值代表具体的错误码

1. 清除训练结果

bool ClearTraining()

输入无，输出为bool

1. 获取训练结果，包括分类情况，使用的算法，每种算法的准确度

def GetTrainingData()

1. 获取文本所属分类, 返回

Def GetDocType(String docContent)

输入为文本内容，文本内容采用UTF-8编码，输出为所属类别

# 5、使用说明

详见README.txt