角色分类引擎数据结构设计

本文档主要介绍各模块调用流程及主要代码，其调用流程如图所示：

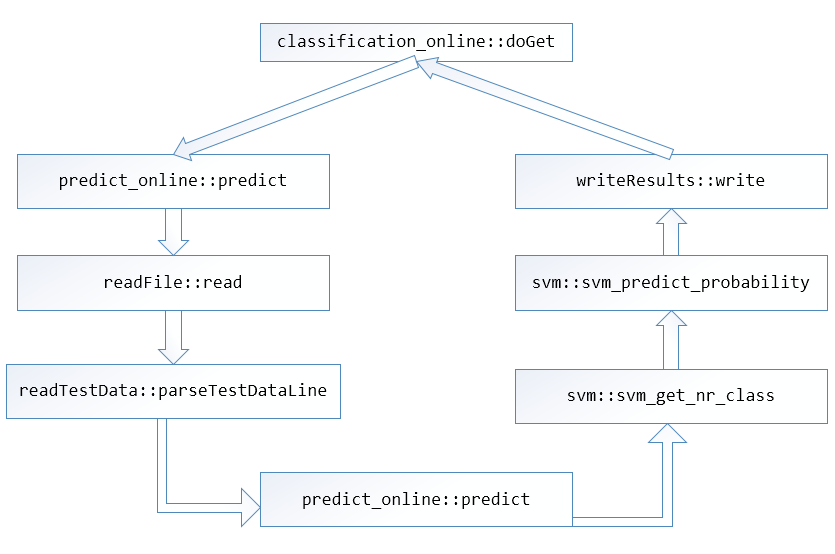


图1 函数调用流程

主要接口和库函数为：Punctuation.h、libcrflearn.so、libPunctuation.so

# 预测代码

predict\_online.java

接口函数：

public void predict(svm\_model model, String testDataFile, Map<String, Integer> wordNumMap, Map<String, Double> wordFeatureValueMap, int valueType, String \_1as, String \_2as, String resultFile)：对testDataFile文件进行分类，并将结果写入resultFile中

参数说明：

svm\_model model：分类调用的SVM模型

String testDataFile：待分类的AB文档

Map<String, Integer> wordNumMap：SVM模型字典词索引Map

Map<String, Double> wordFeatureValueMap：词所对应特征值Map

int valueType：SVM模型表示词的方式，比如1代表布尔，2代表tf等等（1:布尔;2:tf;3:idf;4:idf\_common;5:tf\*idf; 6:tf\*idf\_common;7:tfc;8:itc;9:tfidf\_1;10:tfidf\_2;）

String \_1as： 1代表啥？客户：坐席

String \_2as： 2代表啥？客户：坐席

String resultFile：分类后text\_role结果文件

private predictValueClass predict(List<Map.Entry<Integer, Double>> tempMapList, svm\_model model)：私有函数，由公有函数predict调用，该函数调用SVM模型，对公有函数predict处理的数据tempMapList进行分类，tempMapList抽取测试数据中标注都为A或B的同一类别的数据，结果返回一个predictValueClass对象，该对象保存了SVM模型分类得到的值

参数说明：

List<Map.Entry<Integer, Double>> tempMapList：公有predict函数内部处理得到的标注为同一类别的数据

svm\_model model：分类使用的SVM模型

# 读取待分类文件

readFile.java

接口函数：

public dataClass read(String fileStr)：该函数对分类文件进行处理得到一个dataClass对象，该对象存放了待分类数据各方面信息

参数说明：

String fileStr：待分类数据的路径

# 存储待分类数据各方面信息

dataClass.java

dataClass类中数据成员如下：

String fileName; //待分类文件路径及名称

List<String> contentList = new ArrayList<>();//待分类文件内容，以行为单位（除了包含文件路径及名称的第一行）

String AStr; //待分类文件中标注为A的语音识别内容

String BStr; //待分类文件中标注为B的语音识别内容

String A; //标注标签A代表啥？客户：坐席

String B; //标注标签B代表啥？客户：坐席

dataClass类中函数成员如下：

public void setFileName(String \_fileName)

public String getFileName()

public void setContentList(List<String> \_contentList)

public List<String> getContentList()

public void setAStr(String \_AStr)

public String getAStr()

public void setBStr(String \_BStr)

public String getBStr()

public void setA(String \_A)

public String getA()

public void setB(String \_B)

public String getB()

# 读字典文件

readDic.java

readDic类的数据成员如下：

String \_1as： 1代表啥？客户：坐席

String \_2as： 2代表啥？客户：坐席

String featureNum：SVM模型表示词的方式，比如1代表布尔，2代表tf等等（1:布尔;2:tf;3:idf;4:idf\_common;5:tf\*idf; 6:tf\*idf\_common;7:tfc;8:itc;9:tfidf\_1;10:tfidf\_2;）

readDic类的函数成员如下：

public String get\_1as()

public String get\_2as()

public String getfeatureNum()

public void readDic(String dicPath)：读取字典，然后给该类的三个数据成员赋值

参数说明：

String dicPath：字典路径

# 对待分类的数据每行进行解析

readTestData.java: 对待分类数据每行进行解析，得到一个map（存储该行数据中的词和特征值）

接口函数：

public List<Map.Entry<Integer, Double>> parseTestDataLine(String testDataLine, String wordToken,Map<String, Integer> wordNumMap, Map<String, Double> wordFeatureValueMap, int valueType)：解析待处理文件中每行数据，每个词对应结果map中键，键对应的值为该词的特征值

参数说明：

String testDataLine：待分类文件的一条语音识别结果

String wordToken：取值有三种，分别是 “1”代表字特征，“2”代表字+词特征，“3”代表词特征

Map<String, Integer> wordNumMap：SVM模型字典词索引Map

Map<String, Double> wordFeatureValueMap：词所对应特征值Map

int valueType：SVM模型表示词的方式，比如1代表布尔，2代表tf等等（1:布尔;2:tf;3:idf;4:idf\_common;5:tf\*idf; 6:tf\*idf\_common;7:tfc;8:itc;9:tfidf\_1;10:tfidf\_2;）

角色分类服务用的是8所对应的特征值，该特征计算方法说明如下：

首先了解两个概念：

tf：term-frequency，词频，指某一个给定的词语在该文件中出现的频率

idf：inverse document frequency，逆文档频率，指该文档包含某个词条，但在其他文档中该词条却不存在，即包含该词条的文档数很少

tf-idf(term-frequency, inverse document frequency)是一个用来表示文档中词的重要性的权值，主要思想是：如果某个词或短语在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。

8：itc所对应的特征值计算方法，就是先算每个词的tf-idf，然后计算全部的tf-idf的平方和，开方后再对每个tf-idf做归一化，是一种归一化的计算方法，目的是降低数据尺度，提高分类准确率。

# 将分类结果写入结果文件

writeResults.java

接口函数：

public void write(dataClass dc, String resultPath)：将分类结果写入结果文件

参数说明：

dataClass dc：存储待分类文件的dataClass对象dc

String resultPath；结果文件的路径

# 分词工具

SegMent.java

接口函数：

public List<String> SegMentSentenceWithWordMapReturnList(String strIn, Map<String, Integer> wordNumMap)：该函数对输入字符串strIn进行分词，如果分词结果包含在SVM的字典中就将该分词结果添加到结果集合中

参数说明：

String strIn：待解析的一条语音识别文本

Map<String, Integer> wordNumMap：SVM模型字典词索引Map