# 

**Spark Streaming的分析与应用**

**概要设计说明书**

Version 1.0.0

小组成员：

陈少杰

姜鑫

蒲彦均

邹嘉欣

**版本变更记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 变更时间 | 修改人 | 审核人 | 备注 |
| 1.0 | 2017/05/10 | 姜鑫 | 邹嘉欣、陈少杰、蒲彦均 | 初稿 |

目录

[图片目录 ii](#_Toc482220326)

[表格目录 iii](#_Toc482220327)

[1.概要设计 1](#_Toc482220328)

[1.1工作目标 1](#_Toc482220329)

[1.2 软件基本功能 1](#_Toc482220330)

[1.3软件处理流程 1](#_Toc482220331)

[1.4技术路线 2](#_Toc482220332)

[1.5详细实现方案 2](#_Toc482220333)

[模型训练 2](#_Toc482220334)

[数据的产生 3](#_Toc482220335)

[文本预处理 3](#_Toc482220336)

[文本的分类 3](#_Toc482220337)

[1.6技术难点 3](#_Toc482220338)

[2.运行环境 4](#_Toc482220339)

[2.1硬件环境 4](#_Toc482220340)

[2.2软件环境 4](#_Toc482220341)

[3.参考资料 4](#_Toc482220342)

图片目录

图 1系统架构示意图 2

表格目录

[表 1各模块工作量与来源 3](#_Toc482220318)

[表 2软件环境表 4](#_Toc482220319)

# 1.概要设计

## 1.1工作目标

项目计划基于Spark Streaming框架，以及机器学习、自然语言处理等技术，实现一个针对指定主题短文本的自动识别功能。本软件产品是一项独立的软件，用户根据软件提供的调用命令实现各种指定功能。

期望达到的性能指标是：在十台机器的spark集群上达到每分钟5万条的处理速度。

## 1.2 软件基本功能

本软件以格式化的微博数据为输入，经过软件的一系列处理流程，输出格式化的微博以及其分类标签。输入数据包含微博的所有者，发送时间，微博正文三个字段，输出数据包含以上三个字段以及微博分类标签。

## 1.3软件处理流程

本节主要描述软件对数据的基本处理流程

数据生产模块读取一批固定的、未标注的短文本集合，将其不断发送至kafka的指定topic。

基于Spark Streaming的短文本分类主系统从kafka的上述topic中不断获取实时流数据，根据用户信息等字段可以对短文本进行初步的过滤，而后调用分类模型对短文本流进行分类，打分类标签后存入HDFS和kafka的另一个topic。

分类系统启动时，Driver端进行Spark的任务提交，而实际分类任务运行于各个Worker，各Worker从Kafka中读取待分类短信数据，构建DStream；加载本地分类模型，对DStream进行过滤，转化等操作后完成分类，将结果DStream写回Kafka。

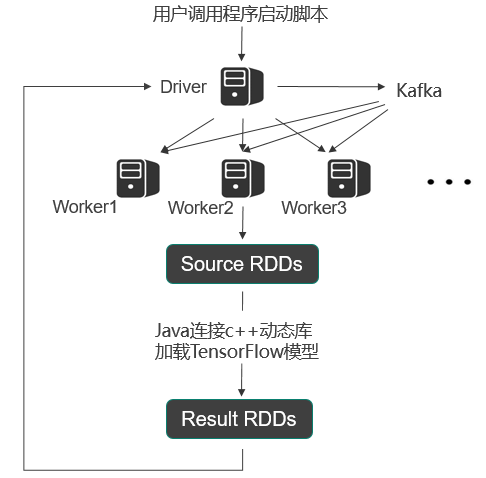


图 1系统架构示意图

## 1.4技术路线

利用已标注的训练样本，进行分词、关键词抽取、词语泛化等预处理后，基于tensorFlow/libsvm训练短文本分类模型。

编写Kafka发送数据程序，读取现有的有限的短文本，产生重复的数据流输入。

编写Spark应用程序，利用Spark API中的filter、map等RDD操作。调用分类模型进行实时流数据短文本分类。

编写kafka消费程序，读取结果topic写入数据库完成短文本的分类任务。

## 1.5详细实现方案

### 模型训练

由于本项目的重点在于Spark Streaming的使用，模型训练并不是我们要着重关心的问题。这里，我们将使用java的libsvm工具作为文本分类的模型训练工具。

### 数据的产生

由于项目能获取的样本数有限，我们采用对同一批数据进行无限复制从而产生流式数据的解决办法。数据的产生由单独的程序控制，程度每隔一段时间对源数据进行读取，并按照数据格式标准将源数据发送到kafka指定主题下，供给分类程序使用。

### 文本预处理

对于输入的文本，首先要进行分词、去停用词这样的基本操作，而为了忽略掉一些对文本分类无意义的词，从而达到更好的分类效果，还需将文本中的量词、人名等进行泛化处理，如将“张小明”泛化为“<人名>”，将“一百只”泛化为“<量词>”等。

### 文本的分类

在spark的各个计算节点上，加载训练好的模型。在文本预处理的基础上，对处理后的文本进行向量化，输入训练好的模型，得出分类结果。将分类结果与源数据进行拼接，返回。

## 1.6技术难点

本课题使用Spark作为分布式计算框架，使用java作为开发语言。没有大的困难点，但是会存在一些诸如分布式模型调用等方面的工程问题。因此项目风险也属于可控范围内，项目是可行的。

为了进一步地明确小组的工作量，以表格的形式给出各部分工作的工作量和来源情况。

表 1各模块工作量与来源

|  |  |  |
| --- | --- | --- |
| 功能模块 | 来源 | 备注 |
| 分词 | 工具 | Jieba分词开源工具 |
| 关键词抽取 | 自主编码 | 实现TF-IDF关键词抽取算法 |
| 词性标注、命名实体识别等 | 工具 | CoreNLP |
| 词语泛化 | 自主编码 | 根据词性标注、命名实体识别等结果进行泛化 |
| 文本向量化 | 自主编码 | 使用bow模型进行文本向量化 |
| 分类 | 工具 | LibSVM |
| Kafka数据产生模块 | 自主编码 | 读取微博文本文件，调用KafkaAPI，发送微博数据到Kafka |
| SparkStreaming分类系统 | 自主编码 | 完成构建DStream，filter，map等数据处理流程，并实现写入HDFS和Kafka |

# 2.运行环境

## 2.1硬件环境

当前实验室拥有12台服务器可供使用，硬件为core i7处理器，16G内存，1TB硬盘。

## 2.2软件环境

软件环境如表2所示

表 2软件环境表

|  |  |
| --- | --- |
| 操作系统 | centos 6.5或Ubuntu16.04及以上 |
| 开发环境 | eclipse |
| 项目管理 | Github |
| Hadoop | 2.2及以上 |
| Spark | 1.6及以上 |

# 3.参考资料

[1]<https://www.ibm.com/developerworks/cn/analytics/blog/analyze-social-media-data-real-time/index.html>

[2] <https://academy.datastax.com/resources/apache-spark-streaming>

[3] <http://www.infoq.com/cn/articles/apache-spark-streaming>