2018 网络零售平台商品分类项目

详细方案

团队：TempName 小组

A 类-1801047-TempName-2018 网络零售平台商品分类项目详细方案

目录

1. 前言

1.1 编写目的

本说明书给出 2018 网络零售平台商品分类项目的设计说明，包括最终实现的项目必须满足的功能、效率、采用实现技术的详细说明。

目的在于：

- 为编码人员提供依据；
- 为修改、维护提供条件；
- 项目负责人将按此计划书的要求布置和控制开发工作全过程。

本说明书的预期读者包括：

- 项目开发人员，特别是编码人员；
- 软件维护人员；
- 技术管理人员；
- 项目负责人和全体干系人。

1.2 背景

分类一直是数据科学界研究的重点问题，它被广泛地应用到生活的各个方面。伴随着电商行业的快速发展。商品的数量越来越多，需要对商品制定分类，便于找寻自己所需的商品。针对现在每天都会产生的大量商品名称，如果人工去为商品分类，不仅工作量巨大，速度慢，而且也会出现分类错误的情况。因此本项目旨在寻找一种分类方法，能够实现商品的快速准确的分类，降低人工成本以及出错率。

待开发项目的名称：2018 网络零售平台商品分类

此项目任务提出者：浪潮卓数大数据产业发展有限公司

此项目任务开发者：TempName 小组

此项目任务对象：450 万个无分类的商品

1.3 术语

- **商品标签：**为了区别商品的出处，在这里特指商品的大类（体现商品生产和流通领域的行业分工，如珠宝饰品）、中类（体现具有若干共同性质或特征商品的总称，如翡翠玉石）和小类（对中类商品的进一步划分，体现具体的商品名称，如项链）。
- **分类：**在本项目中指利用 50 万个商品包含的标签信息，对剩余的 450 万个商品进行合理的标签判定。

1.4 参考资料

- A 类-1801047-TempName-2018 网络零售平台商品分类项目概要介绍
- 结巴中文分词介绍
  (https://blog.csdn.net/haishu_zheng/article/details/80430106)
- 用深度学习解决大规模文本分类问题
  (https://blog.csdn.net/u010417185/article/details/80652233)
- 基于 Text-CNN 模型的中文文本分类实战(https://www.imooc.com/article/40868)
- 卷积神经网络三个概念（epoch，迭代次数，batchsize）
  https://blog.csdn.net/qq_37274615/article/details/81147013
- 准确率、精确率、召回率，F1 值、ROC/AUC 整理笔记
  （https://blog.csdn.net/u013063099/article/details/80964865）
- 需求规格说明标准规范
  （http://blog.sina.com.cn/s/blog_4902a6390102w1k9.html）
- 一文盘点 2012 年以来国内大数据相关政策
  (https://blog.csdn.net/enohtzvqijxo00atz3y8/article/details/80730754）
- 各个算法优缺点
  （https://blog.csdn.net/u013909139/article/details/69740089）

2.项目概述

2.1 适用范围及系统特性简要说明

该项目的应用对象主要是网上零售平台的商品，在我们的生活当中可能会有类似的经历，我们想要购买某件商品，一般是搜索该商品的类别去查询我们想要的产品，但是会存在着搜出来的商品和我们想要商品的类别不符的情况，这就是由于分类不精确导致的问题。因此我们的项目适用于来自不同零售平台的商品，能够通过其商品描述信息自动高效地判定其类别。

2.2 项目创意及特色

2.2.1 项目创意

本项目采用基于 CNN 的文本分类模型实现自动分类。文本分类模型大体上分为基于传统机器学习和基于深度学习的文本分类模型，后者与前者最主要的区别是随着数据规模的增加其性能也不断增长。本项目的数据集在万级以上，因此基于深度学习的文本分类模型能够更加完美地解释它。随着现在大数据时代的到来，基于深度学习模型的文本

分类模型已经成为了主流，其中 CNN 模型在文本分类任务中是兼具效率与质量的理想模型。因此基于 CNN 的文本分类模型具有良好的商业价值和社会应用价值。

2.2.2 项目特色

1. 采用针对大量数据集的深度学习框架从而可以自动地从已构建的数据集上归纳出一套分类规则；

2. 采用结巴中文分词技术能够将句子最精确地切开，适合文本分析；

3. 采用 One-Hot 技术使文本数值化能够有效降低异常值对模型的影响，增强模型稳定性；

4. 采用目前业界普遍认为准确度最高的模型 TextCNN 进行文本分类，兼具效率与质量；

5. 采用 MVC 架构实现用户与系统之间的交互，支持多种查询数据的方式，可视化效果好。

2.3 功能简介

本项目的功能模块如图 1 所示：

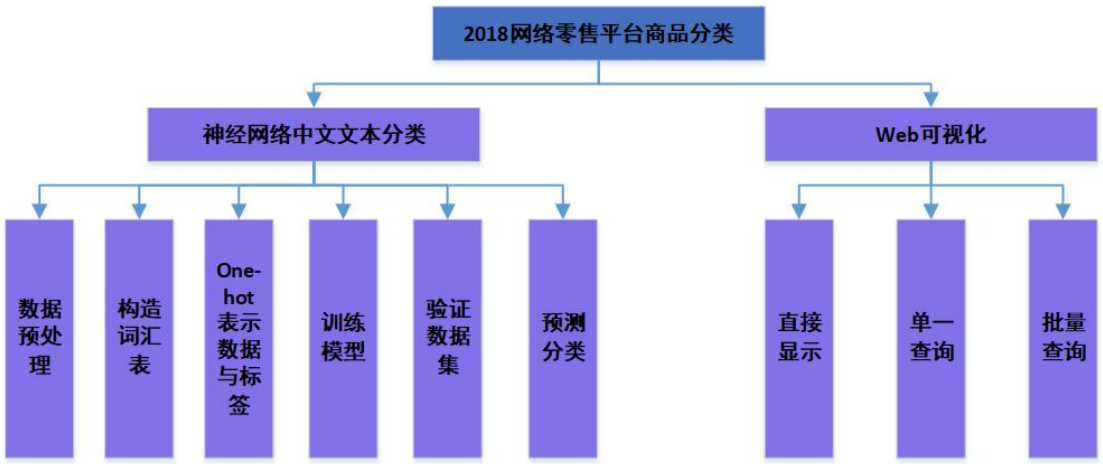

图 1. 2018 网络零售平台商品分类项目功能模块图

本项目的业务需求是通过对 50 万带有商品分类标签的商品进行训练，建立分类模型，对 450 万不带分类标签的商品进行分类，将分类结果存入数据库并用 web 端可视化显示出来。

首先对已知的五十万数据进行预处理，（除去非文本内容，处理编码、某些行存在几百条数据堆积等问题）对合法的中文和英文数据进行结巴分词处理并构造词汇表，利用 one-hot 技术来表示数据与标签，训练集和验证集的比例为 19:1，其中 47.5 万数据作为训练集，2.5 万数据作为验证集。采用模型 TextCNN 对我们的数据集进行反复训练和验证，通过反复调试参数选取一个最佳的模型，用已分类的 2.5 万数据与其已知的分类结果进行比对得到准确率，接着读取待分类的 450 万数据同样进行预处理，预测其分

类结果写入数据库。具体流程如图 2 所示：

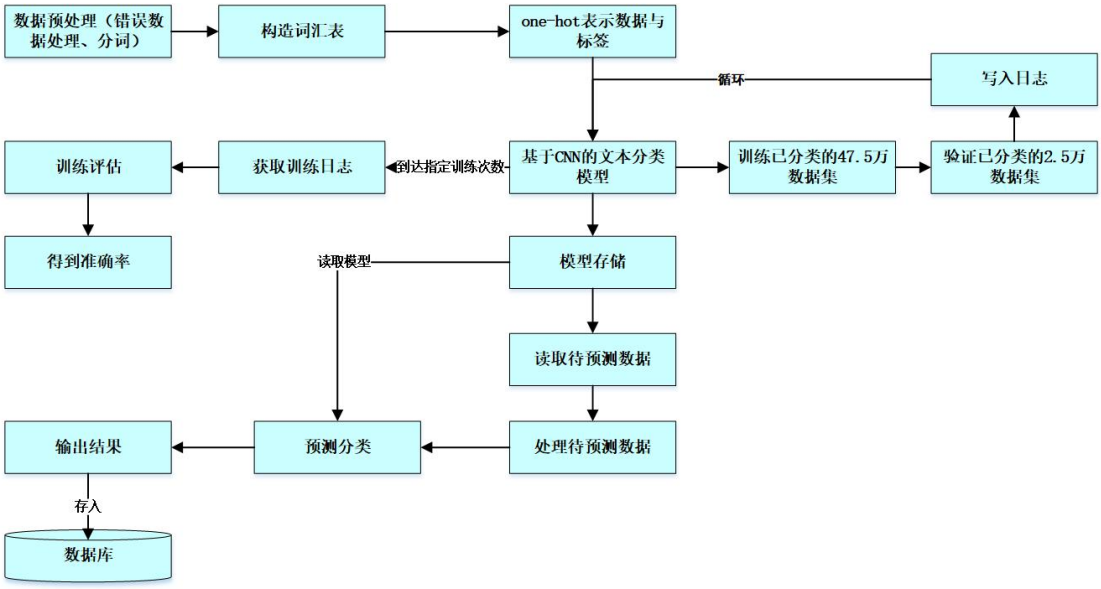

图 2. 基于 CNN 模型的文本分类技术实现流程图

Web 端采用 MVC 架构，实现整体数据显示、单一查询和批量查询这三个功能。其中整体数据显示部分以分页表格的形式呈现数据库中所有的数据（450 万条打好标签的数据）；单一查询通过搜索商品信息，显示出对应的分类信息；批量查询通过上传文件的方式显示出该文件所有的商品信息以及对应的分类信息。其实现框架如图 3 所示：

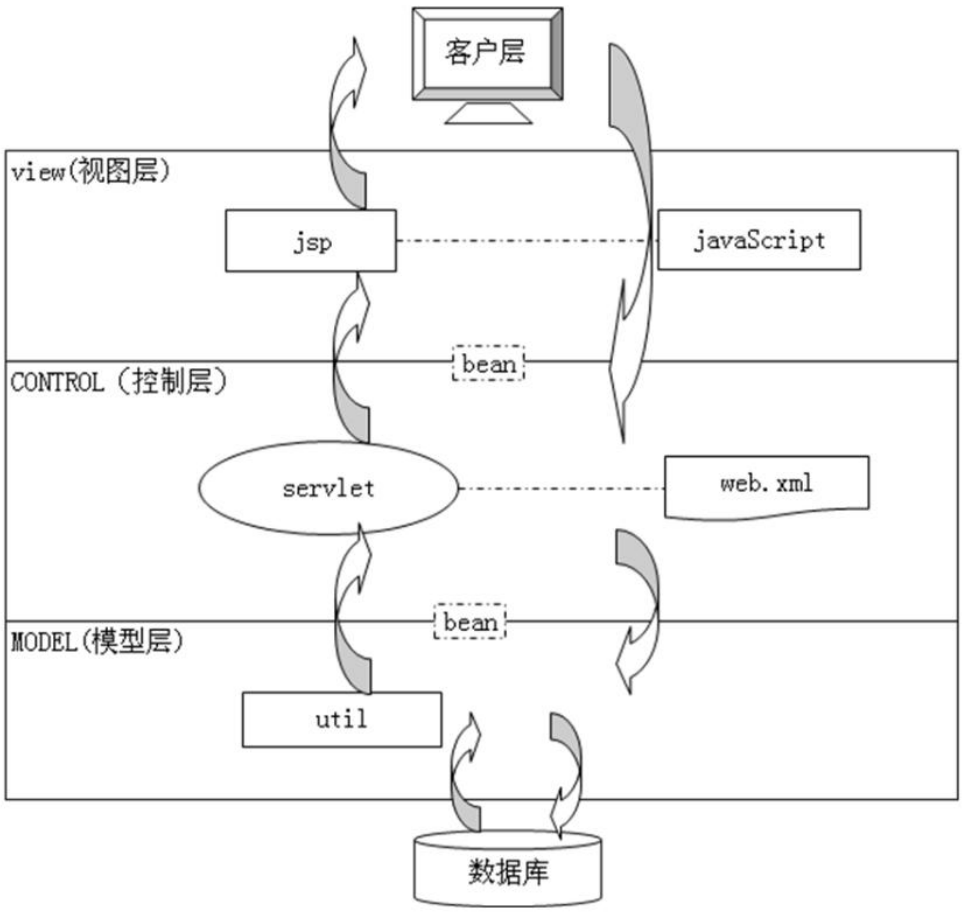

图 3.MVC 实现框架

2.4 开发工具与技术

2.4.1 神经网络文本分类：

分类	名称	版本
开发工具	pycharm	5
重要库	jieba（中文分词）	0.39
	pandas	0.24.1
	numpy	1.16.2
	tensorflow	1.13.1
数据库平台	Mysql	5.7.21
代码托管平台	GitHub	1.3.3

2.4.2 可视化 web 端开发：

分类	名称	版本
开发工具	MyEclipse	4.6.1
应用平台	Tomcat	9.0
开发平台	JDK	1.8.0
数据库平台	Mysql	5.7.21

3.项目主要功能/流程的详细介绍

3.1 数据预处理

我们希望能够得知商品信息,自动对其进行分类,比如针对“腾讯 QQ 币 148 元 148QQ 币 148 个直充 148Q 币 148 个 Q 币 148 个 QQB★自动充值”这样一个商品信息，预期得到“本地生活--游戏充值--QQ 充值”，那么首先第一步就是要对原始数据进行预处理，在本项目中主要处理以下问题：

3.1.1 除去非文本部分

由于原始数据当中可能存在着一些字符、数字等非文本内容，这些对于我们的分类没有任何用处，因此可以直接用 Python 的正则表达式（re）删除，复杂的则可以用 beautifulsoup 来去除。

3.1.2 处理中文编码

我们的项目采用的是 python3.5，由于其不支持 unicode 的处理，因此我们做中文文本预处理时需要遵循的原则是，存储数据都用 utf8，读出来进行中文相关处理时，使用 GBK 之类的中文编码。

3.1.3 处理某些行堆积几百条数据

在我们获取分类的时候我们发现了原始数据当中存在着某些行有几百条数据堆积的情况，因此我们把出现这种情况的行取出来然后一条一条写回去，把它变成符合规范的数据。

3.1.4 结巴中文分词处理

当前市面上存在着很多的分词工具，如结巴中文分词、中科院分词系统、smallseg、snallseg 和 ansj 分词器等，由于我们项目所需要处理的商品信息大多都是中文，而这些分词工具当中最适合处理中文文本的便是结巴中文分词技术，并且该分词技术网上有很好的参考框架和开源代码，因此我们采用该技术对数据进行分词处理。

原理：

基于词典，对句子进行词图扫描，生成所有成词情况所构成的有向无环图；根据有向无环图，反向计算最大概率路径；根据路径获取最大概率的分词序列。

它支持三种分词模式：

精确模式：试图将句子最精确切开，适合文本分析；

全模式：把句子中所有的可以成词的词语都扫描出来，速度非常快，但是不能解决歧义；

搜索引擎模式，在精确模式的基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词。

由于我们的项目主要是对大量的文本进行分析，为了保证较高的准确率，因此我们采用的是精确模式。如：我来到北京清华大学。在经过精确模式的处理下，会被切分成：我/来到/北京/清华大学。

结巴分词技术的接口有很多种类，为了便于对数据进行分析处理，我们采用 jieba.lcut(sentence)接口，其返回的是一个列表，便于我们后期的操作。

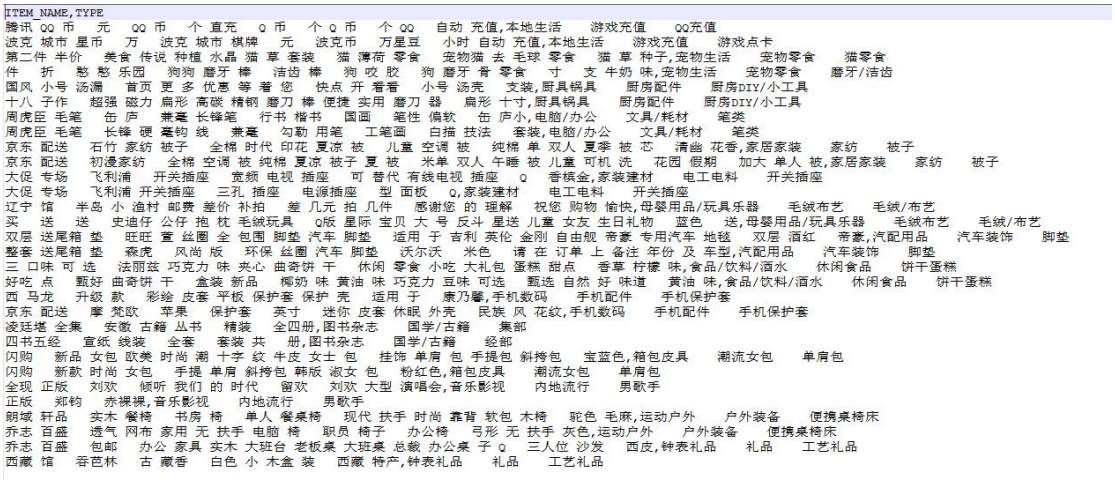

图 4.结巴分词示例图

3.2 one-hot 表示数据与标签

原理：

One-Hot 编码，又称为一位有效编码，主要是采用 N 位状态寄存器来对 N 个状态进行编码，每个状态都由他独立的寄存器位，并且在任意时候只有一位有效。One-Hot 编码是分类变量作为二进制向量的表示。这首先要求将分类值映射到整数值。然后，每个整数值被表示为二进制向量，除了整数的索引之外，它都是零值，并且它被标记为 1。

假设我们对分词后的数据需要进行 one-hot 表示，举个例子：

我们现在有三个特征属性：

性别：["male", "female"]

地区：["Europe", "US", "Asia"]

浏览器：["Firefox", "Chrome", "Safari", "Internet Explorer"]

用 one-hot 表示样本“["male", "US", "Internet Explorer"]编码则为：[1,0,0,1,0,0,0,0,1]，能够快速把数据机器化，便于我们训练数据。

```
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
```

图 5.One-hot 标签示例图

```
[80  7  5  1  1 21  1  1  7  5  1  1  8 81  1  1  4  5  1  1  8  4  5  1
  1  8  7  1  1 22 23  0  0  0  0  0  0  0  0  0]

[24 25 82  1  1 83  1  1 24 25 84  1  1 21  1  1 85  1  1 86  1  1 87 22
 23  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]

[88 89  1  1 90 91 92 93  9 26 10  1  1  9 94  6  1  1 95 96 97  6  1  1
  9 26 98  0  0  0  0  0  0  0  0  0  0  0  0  0]

[ 99   1   1 100   1   1  27  27 101   1   1 102  28  11   1   1 103  11
   1   1  29 104 105   1   1  29  28 106   6   1   1 107   1   1 108 109
   3   0   0   0]

[110  30 111   1   1 112 113 114 115 116 117 118   1   1 119 120 121   1
   1  30 122   1   1 123   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0]

[124 125   1   1 126 127  31 128 129  32  11 130 131  32 132   1   1  31
 133   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0]

[ 33  34   1   1  35 134   1   1  36 135   1   1 136 137   1   1 138   1
   1 139 140   1   1  35 141   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0]

[ 33  34   1   1 142 143 144 145   1   1  36   1   1 146 147   1   1 148
   1   1 149 150   1   1  10   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0]
```

图 6.One-hot 表示数据示例图

3.3 CNN 文本分类模型

3.3.1 为何选择 CNN 文本分类模型

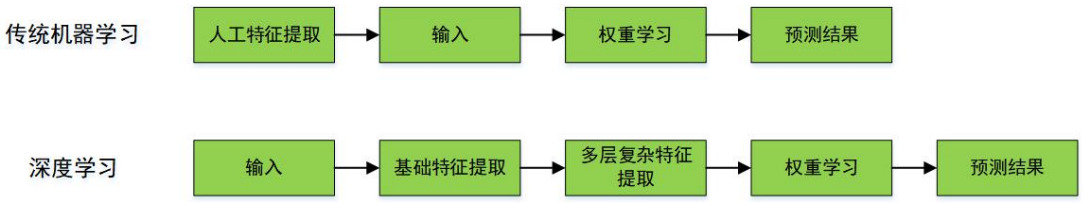

图 7. 传统机器学习和深度学习方式实现步骤

文本分类模型，可以大体上分为基于传统机器学习的文本分类模型和基于深度学习的分类模型。从图 7 中我们也可以看到，深度学习不需要人工提取特征，自动提取初级特征并组合为高级特征，更加适合大数据集的训练，目前基于深度学习模型的文本分类模型已经成为了主流。

Model	fastText	TextCNN	TextRNN	RCNN	DynamicMemory	Transformer
Score	0.362	0.405	0.358	0.395	0.392	0.322
Training	10m	2h	10h	2h	5h	7h

本项目使用 2013 年 Kim 提出的 Text-CNN 模型作为文本分类模型，通过验证试验以及业界的共识，从上表当中我们也可看出，在文本分类任务中，CNN 模型已经能够取到比较好的结果，虽然在某些数据集上效果可能会比 RNN 差一点，但是 CNN 模型训练的效率更高。

所以，一般认为 CNN 模型在文本分类任务中是兼具效率与质量的理想模型。

3.3.2 CNN 模型结构图

CNN 的大致结构如图 8 所示：

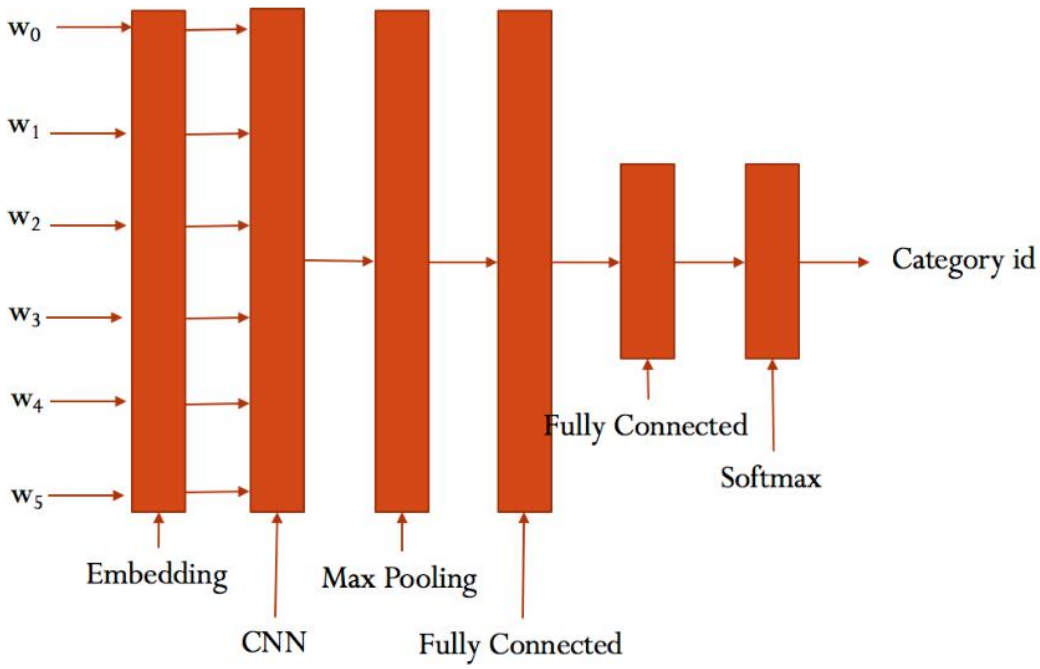

图 8. CNN 的结构图

3.3.3 CNN 配置参数

参数	说明
embedding_dim =128	词向量维度
seq_length =40	序列长度
num_classes=1199	类别数
num_filters=128	卷积核数目
filter_sizes=3,4,5	卷积核尺寸
vocab_size=160000	词汇大小
fc_hidden_size=1024	全连接层神经元
dropout_keep_prob=0.5	防止过拟合
dropout	保留比例
learning_rate=0.001	学习率

batch_size=256	每批训练大小
num_epochs=40	总迭代轮次
evaluate_every=800	每多少步进行一次验证

3.3.4 训练集、验证集的选取

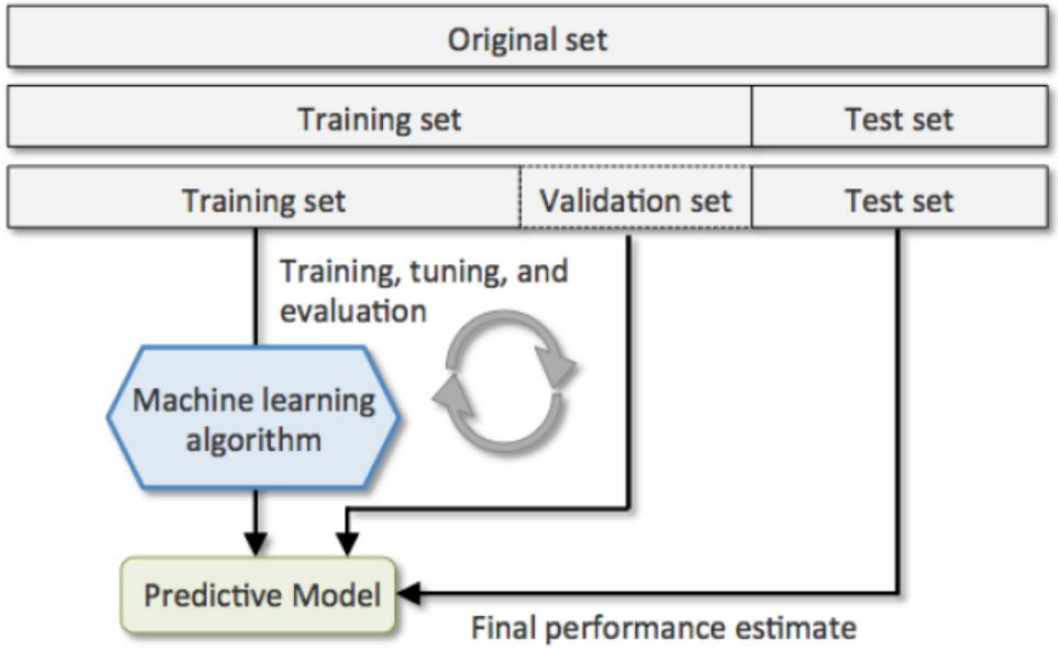

图 9. 训练集、验证集、测试集的作用图

在有监督的机器学习中，数据集常被分成 2~3 分，即：训练集（train set），验证集（validation set）、测试集（test set）。如果我们自己已经有了一个大的标注数据集，想要完成一个有监督模型的测试，那么通常使用均匀随机抽样的方式，将数据集划分为训练集、验证集、测试集，这三个集合不能有交集，常见的比例是 8:1:1，但是针对本项目，官方只提供了一个标注的数据集（作为训练集）以及一个没有标注的测试集，因此我们做模型的时候，就得人工从训练集中划分一个验证集出来，即在所给的 50 万数据当中划分出验证集。

合理的划分训练集和验证集对于训练一个成功的模型显得尤为重要，有一个重要的作用就是防止过度拟合。

当数据集不大的时候，一般将训练集和验证集划分为 7:3，但是到了大数据时代，数据量陡增为百万级别，原则上我们只需取少量部分为验证集即可。在该项目已知分类的数据集为 50 万，为了我们的模型能够达到最佳的效果，我们人工尝试了多种划分方式，但都遵循了训练集远大于验证集的原则。通过比对模型的训练的结果，最终采取训练集：验证集为 19:1.

3.3.5 参数变化过程

为了能够得到一个最佳的训练模型，我们人工调试了训练过程中的一些参数情况，如下表所示：

参数变化过程				
参数	第一次	第二次	第三次	第四次
num_epochs	35	35	100	40
batch_size	64	512	256	256
Learn_rate	0.001	0.01	0.001	0.001
训练集验证集比例	19:1	19:1	4:1	19:1
最终结果	Recall 0.628637, accuracy 0.624419 F 0.625825	Recall 0.824505, accuracy 0.823446, F 0.823799	Recall 0.843764, accuracy 0.83916, F 0.840695	Recall 0.857021, accuracy 0.857925 F 0.853659

根据上表，在这里我们展示第四次效果最佳情况下的一些结果变化过程.

再此之前我们先解释一下一些结果值的概念。

首先有关 TP、TN、FP、FN 的概念。大体来看，TP 与 TN 都是分对了情况，TP 是正类，TN 是负类。则推断出，FP 是把错的分成了对的，而 FN 则是把对的分成了错的。

1. 准确率（Accuracy）。顾名思义，就是所有预测正确（正类负类）的占总的比重。

$$Accuracy=（TP+TN）/（TP+TN+FP+FN）$$

2. 精确率（Precison），查准率。即正确预测为正的占全部预测为正的比例。

$$Precision=TP/(TP+FP)$$

3. 召回率（Recall），查全率。即正确预测为正的占全部实际为正的比例。

$$Recall=TP/(TP+FN)$$

4. F1 值，算数平均数除以几何平均数，且越大越好。

$$2/F1=1/Precision+1/Recall$$

5. Epoch,使用训练集的全部数据对模型进行一次完成训练,被称之为“一代训练”。

6. Batch，使用训练集中的一小部分样本对模型权重进行一次反向传播的参数更新，这一部分样本被称为“一批数据”。

7. Iteration，使用一个 Batch 数据对模型进行一次参数更新的过程，被称之为“一次训练”。

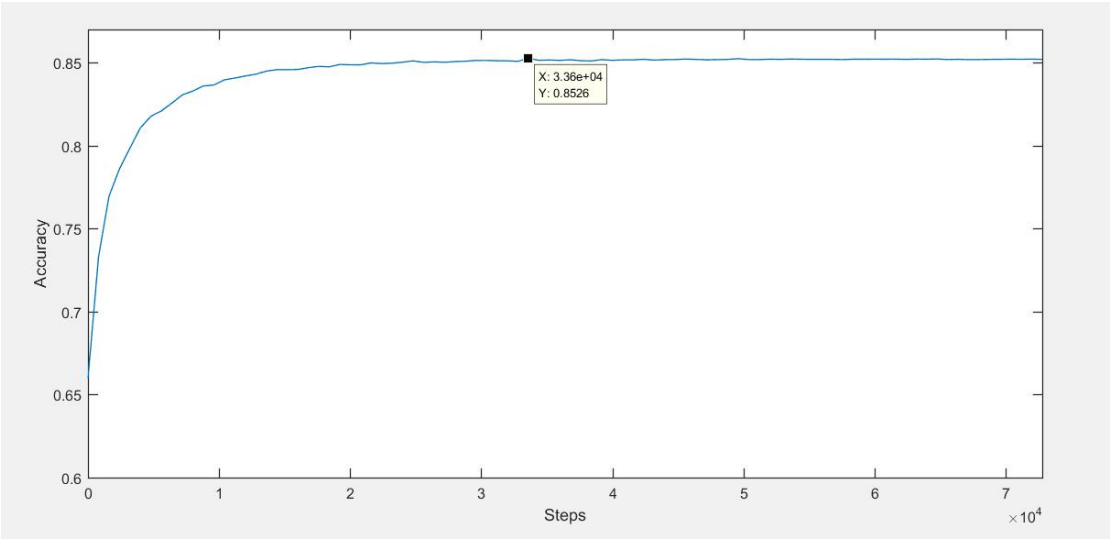

图 10. Accuracy 随 Epoch 的变化情况

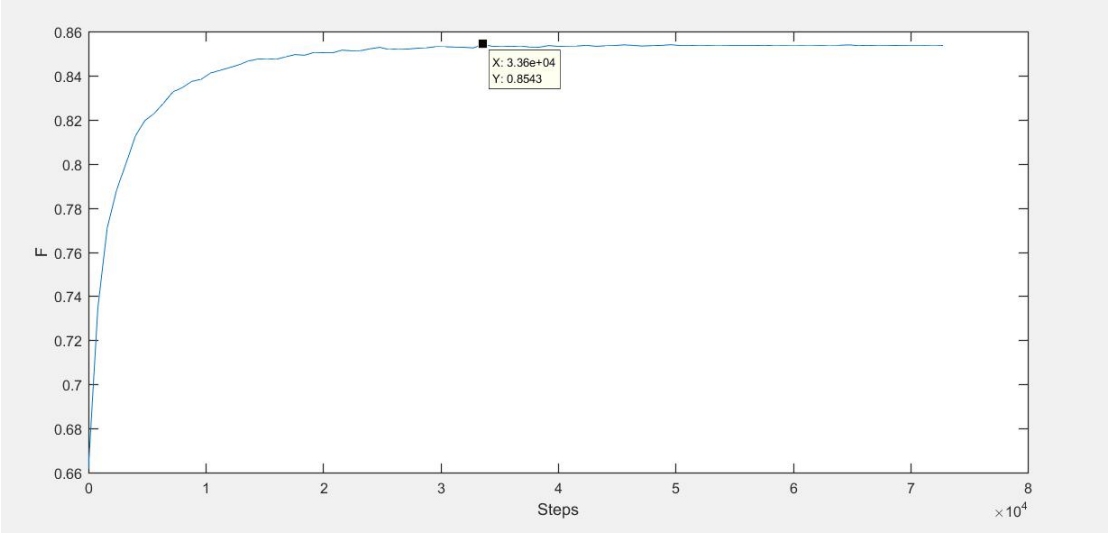

图 11. F 值随 Epoch 的变化情况

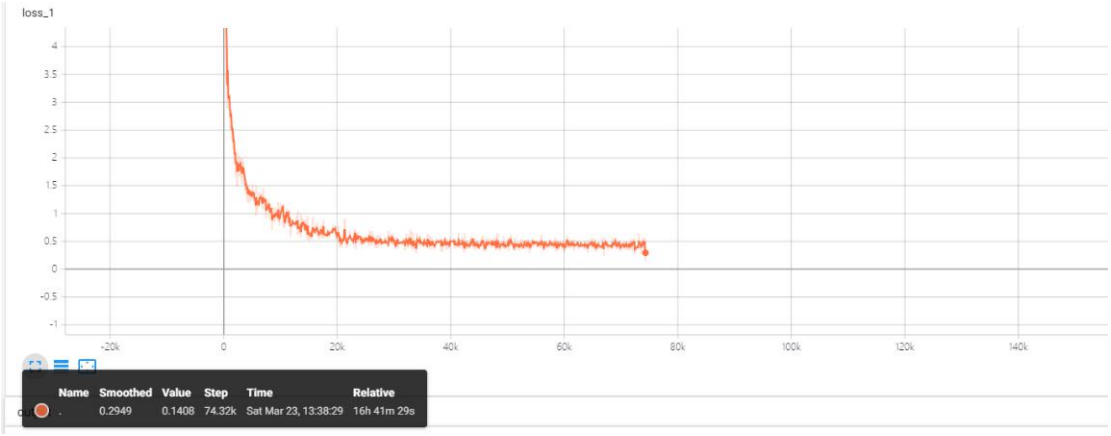

图 12. Loss 值的变化情况

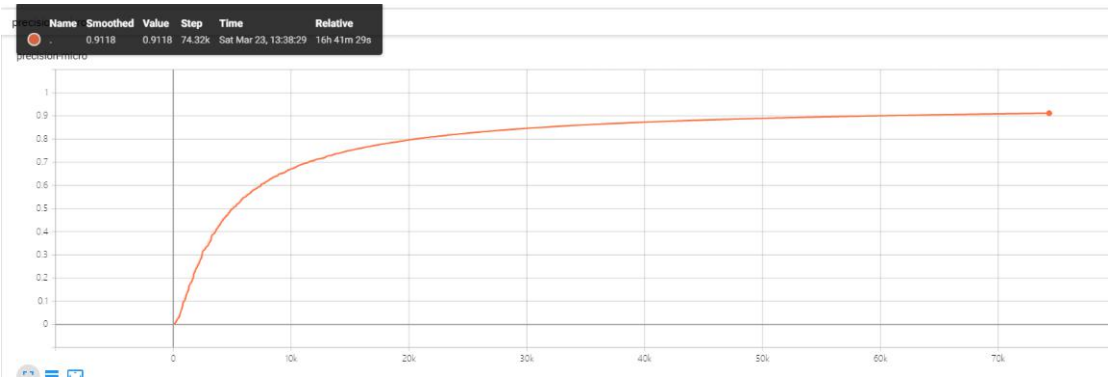

图 13. Precision 值的变化情况

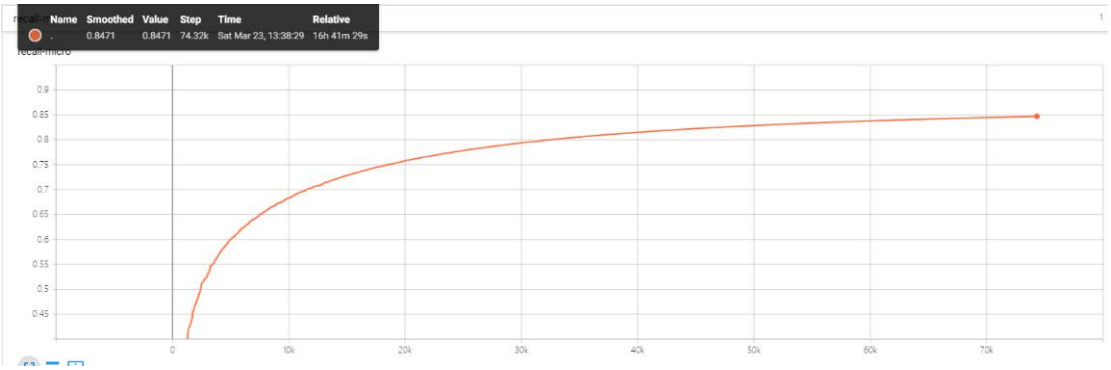

图 14. Recall 值的变化情况

3.4 web 可视化端的主要功能

3.4.1 MVC 架构介绍

MVC 是一种使用 MVC（Model View Controller 模型-视图-控制器）设计创建 Web 应用程序的模式：

Model（模型）表示应用程序核心，是应用程序中用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据。在我们的项目中，模型主要是用来连接、读取数据库以及读取用户上传的文件。

View（视图）是应用程序中处理数据显示的部分。通常视图是依据模型数据创建的。在我们的项目中，视图主要是 jsp 页面显示分类数据，并提供了对 HTML、CSS 和 JavaScript 的完全控制。

Controller（控制器）是应用程序中处理用户交互的部分。处理输入（写入数据库记录）。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。在我们的项目中，控制器接收用户从视图层传来的信息并加以处理并获取模型层读取的数据进行逻辑处理，并将处理结果反馈给视图层，显示到 jsp 页面上。

MVC 分层有助于管理复杂的应用程序，因为您可以在一个时间内专门关注一个方面。例如，您可以在不依赖业务逻辑的情况下专注于视图设计。同时也让应用程序的测试更加容易。

3.4.2 整体数据显示

该部分主要是以分页表格的形式呈现出我们数据库中的所有数据（即 450 万打好标签的商品信息及其分类信息），在该部分我们可以进行翻页和跳页操作，跳页的响应时间在 2s 以内。

MVC 实现流程： 用户进入网页界面，起始页为查询数据库中所有的商品分类，控制器根据请求类型和请求的指令发送到相应的模型，模型根据页面的页码读取数据库中指定的信息（每次读 100 条），完成之后，控制器选择相应的视图以表格形式（可翻页）显示文件中商品以及商品的分类信息。

图 15. 整体数据显示功能界面

3.4.3 单一查询

该部分主要是通过输入商品的具体描述信息，会自动在数据库中检索该条数据的分类信息并显示出来。

MVC 实现流程： 用户进入单一查询界面，在搜索框中输入商品信息，点击查询将请求发送给控制器，控制器根据请求类型和请求的指令发送到相应的模型，模型与数据库交互，根据用户输入的内容查找商品的分类信息，完成之后，控制器选择相应的视图以表格形式显示用户输入的商品以及商品的分类信息，用户可以进行下一次查询，如此循环。

图 16. 单一查询功能界面

3.4.4 批量查询

该部分主要是以上传文件的形式来查询商品的分类信息，上传的文件中包含少量或者大量的商品信息，通过检索数据库呈现出上传文件中所对应的所有商品信息及其分类信息。

MVC 实现流程：用户进入批量查询界面，选择文件，然后将上传请求发送到控制器，控制器将文件上传到服务器，用户点击查询按钮根据请求类型和请求的指令发送到相应的模型，模型根据页面的页码读取文件中指定的商品信息（每次读 100 条,），然后模型与数据库交互，根据读出的批量商品信息查找商品的分类，完成之后，控制器选择相应的视图以表格形式（可翻页）显示文件中商品以及商品的分类信息，用户可以进行下一次查询，如此循环。

图 17. 批量查询功能界面

3.4.5 主要业务流程

Web 主要业务功能的流程图如图 18 所示：

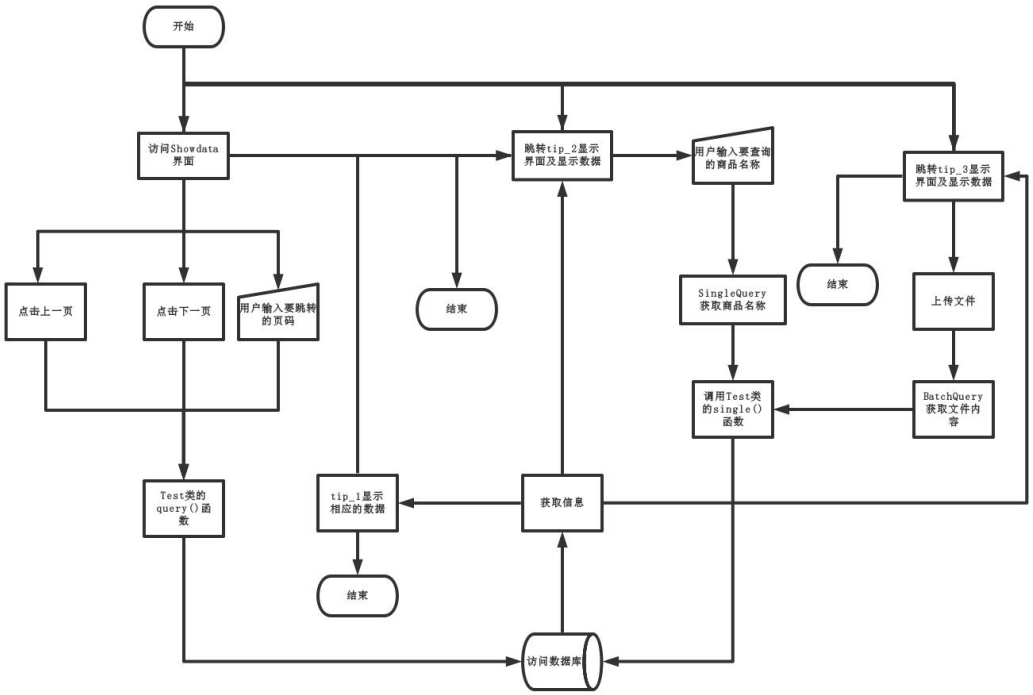

图 18. web 端主要业务功能流程图

4.数据库结构设计

4.1 数据字典

字段	类型	长度	是否为空	说明	约束
content	nvarchar	255	否	商品信息	索引、不为空
type	nvarchar	255	否	类型	不为空

4.2 概念结构设计

实体联系模型（E-R 模型）是被广泛采用的概念模型设计方法。

下面采用 E-R 模型的方法对数据库进行概念设计，如图 19 所示：

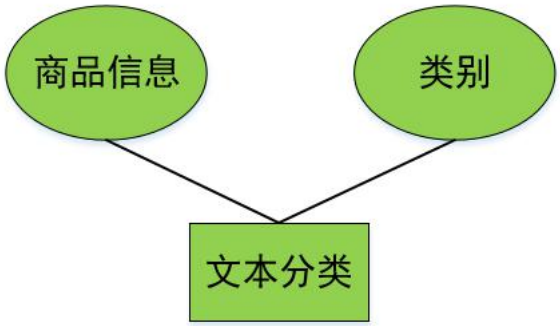

图 19. 数据库概念设计图

4.3 逻辑结构设计

我们将上面的 E-R 图转换成关系模型，在本项目中，无主键，context 为索引。

classification(context,type)

5.重点功能函数说明

5.1 神经网络中文文本分类

5.1.1 HandleOriginData

HandleOriginData			
函数说明：处理原始数据（分词，错误处理）			
参数	参数说明	返回值	返回值说明
inputFileName	数据输入文件名	NoneType	无
outputFileName	数据输出文件名	NoneType	无

5.1.2 build_vocab

build_vocab

函数说明：构造词汇表			
参数	参数说明	返回值	返回值说明
filenames	文件名集合	NoneType	无
vocab_dir	词汇表存储文件	NoneType	无
vocab_size	词汇数量	NoneType	无

5.1.3 Get_Val_Data_From_TrainData

Get_Val_Data_From_TrainData			
函数说明：从给定训练数据分出训练集和验证集			
参数	参数说明	返回值	返回值说明
无	无	NoneType	无
无	无	NoneType	无

5.1.4 Get_Content_Label

Get_Content_Label			
函数说明：获取文件的两列数据(内容以及标签)			
参数	参数说明	返回值	返回值说明
filename	文件名	Content	内容
无	无	items_Label	标签

5.1.5 get_label_using_scores_by_topk

get_label_using_scores_by_topk

函数说明：根据模型预测数据获得预测标签			
参数	参数说明	返回值	返回值说明
scores	模型预测数据	predicted_labels	预测标签
top_num	最有可能的前几位	predicted_values	预测标签的数据

5.1.6 cal_metric

cal_metric			
函数说明： 用于训练时计算回归率，准确率，F1-Measure 值			
参数	参数说明	返回值	返回值说明
predicted_labels	预测标签	rec	回归率
labels	实际标签	acc	准确率
无	无	F	F1-Measure 值

5.1.7 process_file

process_file			
函数说明： 构造词汇表			
参数	参数说明	返回值	返回值说明
filename	文件名	x_pad	处理后的内容
word_to_id	单词转 id 表	label_id	处理后的标签
cat_to_id	分类转 id 表	NoneType	无
numClass	标签种类数量	NoneType	无
max_length	句子分词后最大长	NoneType	无

	度		

5.1.8 predict

predict			
函数说明：预测数据标签并将数据与预测标签结果写入文件			
参数	参数说明	返回值	返回值说明
originContent	待预测原始数据	NoneType	无
predicts	原始数据处理后的数据	NoneType	无

5.1.9 train

train			
函数说明：训练模型			
参数	参数说明	返回值	返回值说明
无	无	NoneType	无
无	无	NoneType	无

5.2 web 可视化

5.2.1 实体类

Classification 类

成员属性		
属性名称	数据类型	描述
context	String	商品的描述

type	String	商品的分类

5.2.2 Model

1.Conn 类 ： 连接数据库

成员属性		
属性名称	数据类型	描述
conn	Connection	连接字符串

成员函数		
getConnection() 该函数的功能：如果成功，返回连接字符串；如果不成功，返回空		
返回值	数据类型	描述
conn	Connection	连接字符串

2.Test 类 ： 操作数据库

成员属性		
属性名称	数据类型	描述
conn	Conn	连接数据库的对象
ps	PreparedStatement	执行查询语句
rs	ResultSet	存储结果集
list	List<Classification>	以 Classification 对象形式存储查到的数据库内容

成员函数
query()

该函数的功能：查询任意位置起的 100 行数据		
参数	数据类型	描述
start	int	起始位置行
返回值	数据类型	描述
list	List<Classification>	以 Classification 对象形式存储查到的数据库内容
number ()		
该函数的功能：计算数据库表的行数		
返回值	数据类型	描述
rowCount	int	数据库表的行数
single()		
该函数的功能：以 Classification 对象形式返回查到的数据库内容		
参数	数据类型	描述
context	String	传入要查询的 context 值
返回值	数据类型	描述
new Classification(rs.getString ("context"), rs.getString("type"))	Classification	Classification 对象

3.ReaderTest 类：读取 csv 文件

成员属性		
属性名称	数据类型	描述
csvReader	CsvReader	读取器

list_qList	List<Classification>	以 Classification 对象形式存储读到的文件内容

成员函数		
getReader()		
该函数的功能：构造读取器，设置读取编码		
参数	数据类型	描述
filepath	String	文件路径
read()		
该函数的功能：存储从文件任意行数起的 100 行		
参数	数据类型	描述
num	int	存储数据的起始行
返回值	数据类型	描述
list_qList	List<Classification>	以 Classification 对象形式存储读到的文件内容

5.2.3 Controller

1. Showdata servlet 类：接收 tip1.jsp 页面查询的请求，存储数据库中前 100 行条数的列表，查询前一页的内容并存储，显示页码数

成员属性		
属性名称	数据类型	描述
test	Test	查询数据库的对象
list	List<Classification>	存储 Classification 对象的列表

sumpage	int	存储一共有多少页

成员函数		
Showdata()		
该函数的功能：构造函数，初始化 test，list，sumpage		
doPost()		
该函数的功能：修改页码减 1，调用 Test 类的 query()函数，将读取到的内容以 Classification 对象形式存储到 session 中，调用 Test 类的 number()函数获取页码数并存储		
参数	数据类型	描述
request	HttpServletRequest	请求
response	HttpServletResponse	响应

2. Next ：接收 tip1.jsp 查询下一页的请求，查询前一页的内容并存储成员属性(Servlet 类)

成员函数		
doPost() 修改页码加 1，调用 Test 类的 query()函数，将读取到的内容以 Classification 对象形式存储到 session 中，调用 Test 类的 number()函数获取页码数并存储		
参数	数据类型	描述
request	HttpServletRequest	请求
response	HttpServletResponse	响应

3. Jump_1： 接收 tip1.jsp 跳页请求并查询存储（Servlet 类）

成员函数
doPost()
该函数的功能：获取文本框输入的页码数，修改页码数，调用 Test 类的 query()

函数,将读取到的内容以 Classification 对象形式存储到 session 中,调用 Test 类的 number()函数获取页码数并存储		
参数	数据类型	描述
request	HttpServletRequest	请求
response	HttpServletResponse	响应

4. SingleQuery: 接收 tip2.jsp 中文本框输入的内容并查询其对应的分类(Servlet 类)

成员函数		
doPost()		
该函数的功能：获取文本框输入的内容，Test 类的 single()函数，将读取到的内容以 Classification 对象形式存储		
参数	数据类型	描述
request	HttpServletRequest	请求
response	HttpServletResponse	响应

5. Upload ： 接收 tip3.jsp 中上传文件的请求并上传文件（Servlet 类）

成员函数		
doPost()		
该函数的功能：上传文件，存储文件路径		
参数	数据类型	描述
request	HttpServletRequest	请求
response	HttpServletResponse	响应

6. BatchQuery: 接收 tip3.jsp 中查询文件的请求，获取文件读取的内容，并将读取的内容查询数据库找到相应的分类信息，最后存储起来显示在页面上。（Servlet 类）

成员属性

属性名称	数据类型	描述
list_1	List<Classification>	以 Classification 对象形式存储文件读取的内容
list_2	List<Classification>	以 Classification 对象形式存储匹配后的内容

成员函数		
doPost()		
该函数的功能：调用 ReaderTest 类的 getReader()函数获取文件信息，调用 ReaderTest 类的 read()函数获取文件的指定 100 行数据的信息，获取读取到的文件信息列表存到 list_1 中并遍历，循环调用 Test 类的 single()函数，将读取到的内容以 Classification 对象形式存到 list_2 列表。		
参数	数据类型	描述
request	HttpServletRequest	请求
response	HttpServletResponse	响应

7. Add：接收 tip3.jsp 查询下一页的请求，使当前的页码数+1，跳转到 BatchQuery（Servlet 类）

成员函数		
doPost()		
该函数的功能：修改页码加 1，跳转到 BatchQuery		
参数	数据类型	描述
request	HttpServletRequest	请求
response	HttpServletResponse	响应

8. Reduce：接收 tip3.jsp 查询上一页的请求，使当前的页码数-1，跳转到 BatchQuery（Servlet 类）

成员函数

doPost()		
该函数的功能：修改页码减 1，跳转到 BatchQuery		
参数	数据类型	描述
request	HttpServletRequest	请求
response	HttpServletResponse	响应

9. Jump_2 ：接收 tip3.jsp 跳页的请求,改变当前的页码数,跳转到 BatchQuery(Servlet 类)

成员函数		
doPost()		
该函数的功能：获取文本框输入的页码数，改变当前的页码数，跳转到 BatchQuery		
参数	数据类型	描述
request	HttpServletRequest	请求
response	HttpServletResponse	响应

6.市场分析及行业分析

6.1 市场分析

6.1.1 政治因素

最初，大数据作为一个新兴概念在国内互联网行业中飞速传播，很多企业选择借助大数据的风口实现再次转型升级。2015 年 9 月，经李克强总理签批，国务院印发了《促进大数据发展行动纲要》，系统部署了我国大数据发展工作。至此，大数据成为国家级的发展战略。2016 年，政策细化落地，国家发改委、环保部、工信部、国家林业局、农业部等均推出了关于大数据的发展意见和方案；2017 年，大数据产业的发展正从理论研究加速进入应用时代；2018 年，大数据产业相关的政策内容已经从全面、总体的指导规划逐渐向各大行业、细分领域延伸，物联网、云计算、人工智能、5G 技术与大数据的关系越走越近。

而文本分类技术作为一种大数据挖掘技术可以把数量巨大但缺乏结构的文本数据组织成规范的文本数据，帮助人们提高信息检索的效率。通过对文本信息进行基于内容的分类，自动生成便于用户使用的文本分类系统，从而可以大大降低组织整理文档耗费的人力资源，帮助用户快速找到所需信息。

因此文本分类技术得到日益广泛的关注，成为信息处理领域最重要的研究方向之一，这些良好的外部政策环境将极大地促进文本分类技术的发展。

6.1.2 经济因素

来自在线调查公司 Statista 的数据显示，在经历了快速增长期后，全球范围内的大数据服务进入了平稳增长的阶段。从图 20 我们可以看到，2015 年全球大数据市场规模将近 1500 亿人民币，同比增长 24.2%；我国大数据市场规模为 160 亿元，仅占全球总市场规模的 10.7%，但同比增长率为 65.3%，是全球增长率的 2.7 倍。

数据来源：Statista　　数据来源：36氪研究院

图 20. 全球大数据市场规模和我国大数据市场规模对比图

文本分类技术目前主要应用在新闻出版按照栏目分类、网页分类、个性化新闻一级垃圾邮件过滤等方面，这些应用渗透了人们生活的方方面面，带来了极大的经济效益，其应用前景会更加广泛。

6.1.3 技术因素

90 年代以来，众多的统计方法和机器学习方法应用于自动文本分类。文本分类技术的研究引起了科研人员的极大兴趣。目前英文自动分类已经取得了丰硕的成果，很多前辈提出了多种成熟的分类方法，如最近邻分类、贝叶斯分类、决策树方法以及基于支持向量机（SVM）、向量空间模型（VSM）、回归模型和神经网络等方法；目前国内中文文本分类研究主要集中在朴素贝叶斯、向量空间模型和支持向量机等技术上。并且我国在中文文本方面也取得了不错的成绩，例如百度搜索引擎、新浪网的中文垃圾邮件分类、北京大学的人民日报语料库、清华大学的现代汉语语料库和中科院的分词系统等等。

目前实现文本分类技术的模型有很多，比如说 fastText、TextCNN、TextRNN 等，这些技术在我们的社会生活当中已经有了广泛的应用，并且很多实现文本分类的方法在网上均有开源的框架和代码供大家参考学习，因此本项目有很好的技术支持。

6.2 行业分析

下表例举了其中文本分类技术的优缺点以及适用场景，由于现在是大数据时代，需要我们处理的数据量越来越大，综合多种文本分类技术，我们可以知道 CNN 已经是一种兼具质量与效率的分类方法，目前广泛地用于文本分类当中，有较好的应用前景。

文本分类技术	优势	劣势	适用场景
TextCNN	共享卷积核，对高维数据处理无压力；特征分类效果好	需要调参，需要大量样本，训练最好要 GPU；物理含义不明确	适合于大数据集下的文本分类
朴素贝叶斯	朴素贝叶斯模型发源于古典数学理论，有稳定的分类效率；对缺失数据不太敏感，算法也比较简单；对小规模的数据表现很好，能处理多分类任务	需要知道先验概率；分类决策存在错误率；对输入数据的表达形式很敏感	适合于增量式训练，常用于文本分类，最常见的适用于垃圾邮件分类
决策树	能够同时处理数据型和常规型属性；在相对短的时间内能够对大型数据源做出可行且效果良好的结果；可以对有许多属性的数据集构造决策树	决策树处理缺失数据时的困难；忽略数据集中属性之间的相关性；过度拟合问题的出现	因为它能够生成清晰的基于特征(feature)选择不同预测结果的树状结构，数据分析师希望更好的理解手上的数据的时候往往可以使用决策树。
遗传算法	与问题领域无关切快速随机的搜索能力；搜索使用评价函数启发，过程简单；使用概率机制进行迭代，具有随机性；具有可扩展性，容易与其他算法结合	算法对初始种群的选择有一定的依赖性；遗传算法的编程实现比较复杂，首先需要对问题进行编码，找到最优解之后还需要对问题进行解码	适用于机器人领域
KNN 算法	理论成熟，思想简单，既可以用来做分类也可以用来做回归；可用于非线性分类；训练时间复杂度为 O(n)；对数据没有假设，准确度高，对 outlier 不敏感；	计算量大；样本不平衡问题（即有些类别的样本数量很多，而其它样本的数量很少）；需要大量的内存；	适用于需要一个特别容易解释的模型的时候。比如需要向用户解释原因的推荐算法。
逻辑回归	分类时计算量非常小，速度很快，存储资源低；便利的观测样本概率分数；对逻辑回归而言，多重共线性并不是问题，它可以	当特征空间很大时，逻辑回归的性能不是很好；容易欠拟合；不能很好地处理大量多类特征或变	适用于工业问题

	结合 L2 正则化来解决该问题；	量；对于非线性特征，需要进行转换；	

7.风险和控制

项目风险是指在项目开发生命周期中所遇到的所有的预算、进度和控制等各方面的问题，以及由这些问题而产生的对项目的影响。项目风险经常会涉及许多方面， 如：缺乏用户的参与，缺少高级管理层的支持，含糊的要求，没有计划和管理等，但是具体项目具体分析，针对我们的项目，总体概括下来应该有以下七大方面。

7.1 时间性风险

风险描述：

该项目开发工作量大，且时间有限（截止时间为 2019 年 3 月 27 日），跑模型花费时间久，如果前期选定错误的模型将会浪费大量的时间，这给项目实施带来较大的时间风险。

风险响应计划：

为保证本项目能在较短的时间内提交，在生存期上应采用敏捷式快速成型技术，尽量利用已有的产品和成熟的技术进行集成，逐步实现该项目的功能和服务，同时在项目的进行中各成员各司其职，尽量把工作并行进行，降低时间风险。

7.2 技术风险

风险描述；

技术的飞速发展和经验丰富队员的缺乏，意味着项目团队可能会因为技巧的原因影响项目的成功。如果项目所要求的技术项目成员不具备或掌握不够，则需要重点关注该风险因素。主要有下面这些风险因素：

①团队成员学习能力不够；

②团队成员对方法、工具和技术理解的不够；

③团队成员应用领域的经验不足；

④团队成员不熟悉新的技术和开发方法应用等。

风险响应计划：

项目组选用项目所必须的技术、在技术应用之前，针对相关人员开展好技术培训工作。项目组一定要本着项目的实际要求，选用合适、成熟的技术，千万不要无视项目的实际情况而选用一些虽然先进但并非项目所必须且自己又不熟悉的技术。

7.3 资源风险

风险描述：

由于本项目小组一共只有 5 名人员，且每名人员负责的是项目不同的模块，真正实现核心技术的开发人员有限，因此本项目实施中可能存在一定的资源风险。

风险响应计划；

合理分配开发人员的工作量，对可以投入的开发人员做到高效利用，同时必要的时候可以进行项目赶工。

7.4 管理风险

风险描述：

在大部分项目里，做技术的占绝对多数，他们主要擅长的是技术研发，在管理方面先天不足，这不利于项目风险管理和控制。并且他们写项目风险管理计划自己检查自己的错误，这是最困难的。然而，像这些问题可能会使项目的成功变得更加困难。如果不正视这些棘手的问题，它们就很有可能在项目进行的某个阶段影响项目本身。主要有以下这些风险因素：

①对任务的定义不够充分；

②项目负责人不能很好的权衡管理与技术；

③实际项目状态；

④团队成员之间的沟通以及团队成员能力和素质风险。

风险响应计划：

①定义项目追踪过程并且明晰项目角色和责任；

②根据项目的实际情况，进行科学的项目风险和控制。在项目开发的过程中，进行必要的项目风险分析，制定符合项目特点的风险评估和监督机制，特别是要定期对项目的风险状况进行评估和监管，发现意外风险或者是风险超出预期的一定要重点关照。

③项目建设之初就和项目组人员约定好沟通的渠道和方式并将合适的人安排到合适的岗位上，项目建设过程中多和项目小组人员交流和沟通、注意培养和锻炼自身的沟通技巧。

7.5 安全风险

风险描述：

项目产品本身是属于创造性的产品，产品本身的核心技术保密非常重要。但一直以来，我们在产品开发这方面的安全意识比较淡薄，对项目产品的开发主要注重技术本身，而忽略了智力成果的保护。在开发项目的过程中，免不了要与其他团队或者相关技术人

员交流学习，并且要用到的一些代码托管平台，很可能会导致产品和新技术的泄密，致使我们的产品被别人窃取，导致项目失败。这也是我们软件项目潜在的风险。

风险响应计划：

产品创造性显著，管理主体需要针对其核心技术内容加以保密处理，加强在此类安全细节上的关注度。比如代码托管平台仓库设为私有等。

7.6 工具风险

风险描述：

项目开发和实施过程，所必须用到的管理工具、 开发工具、 测试工具等是否能及时到位，到位的工具版本是否符合项目要求等，是项目组需要考虑的风险因素。

风险响应计划：

在项目的启动阶段就落实好各项工具的来源或可能的替代工具，在这些工具需要使用之前（一般需要提前一个月左右）跟踪并落实工具的到位事宜。

7.7 系统运行环境风险

风险描述：

目前，大部分项目系统集成和开发是分开进行的因此，系统赖以运行的硬件环境和网络环境的建设进度对软件系统是否能顺利实施具有相当大的影响。

风险响应计划：

给用户写明注意事项，提醒用户正确的操作方法并和用户签定相关的协议、跟进系统集成部分的实施进度、及时提醒用户等。

8.结语

整个模型读取 450 万待预测数据并处理的效率为 12，8571 个/分钟，打标签的效率为 97，825 个/分钟、准确率为 85.792%，为训练集打标签的准确率为 91.593%。实际上如果原始数据集更加规范准确、每类商品信息的数据集更大，我们模型的准确率会更高。

可视化 Web 端的可访问链接:http://www.bestdoublelin.com:8080/fuwu/showdata