解决方案设计报告

题 目：商品类目预测

**组 长 姓 名**   **张胜东**

**组 员 姓 名 王嫄嫄**  **杨启东**

**魏献巍**

**参 赛 团 队 Ace**

**日 期 2016 年 11 月 28 日**

**商品类目预测**

**1. 需求分析**

1.1 功能需求分析

在电商运营中，商品编辑、上架是一个很繁琐的工作。千米通过自动化的方式提升用户的感受其中一个很重要的步骤是将商品加到对应的类目中。具体功能为实现用户在添加、修改商品的时候，利用机器学习算法将用户添加商品自动划分到千米标准类目中，进而实现商品类目预测。

1.2 性能需求分析

1. 模型准确率不低于70%
2. 每个商品条目的预测速度不超过200ms

1.3 可行性分析

1.3.1 技术可行性

本团队在本应用中，用到的核心技术和算法思想分别是：

1. 汉语分词器的设计与使用；
2. 使用多线程提高运算效率；
3. 基于 TF-IDF 算法的关键词抽取。

1.3.2 操作可行性

本团队使用python语言，利用关键字提取出商品特征，采用聚类和分类算法实现商品自动分类。程序具有清晰、易懂、易操作的特点。用户不需要任何相关训练，便可直接将待预测商品类目输入并得到其预测结果，故操作可行性能够得到有效的满足。

**2. 算法和设计**

2.1 系统使用的算法

在提取商品特征关键词时，采用了基于前缀词典的高效词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图 (DAG)；采用了动态规划查找最大概率路径, 找出基于词频的最大切分组合；对于未登录词，采用了基于汉字成词能力的 HMM 模型，使用了 Viterbi 算法。在对数据进行训练时，采用了实时聚类算法。

2.2 系统架构和设计

我们使用了汉语分词器来处理训练集并且提取每个样本描述的关键词。用字典记录下一些商品的品牌等内容从而确保分词的有效性和可靠性。之后我们建立了两个表来存储数据。表一在这里我们定义为类表，表二我们定义为词表。其作用如下所述：

类表内容分别为：1）训练集中出现过的所有类目；2）每个类目中存在的关键词；3）每个关键词在该类目的出现次数（WCN）；4）各类目在测试集中总的样本个数（CSN）。用表格表示见表1。

表1 类表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 关键词1 | 关键词2 | …… | 关键词m | 各类目样本数（CSN） |
| 类目1 | （WCN） |  |  |  |  |
| 类目 2 |  |  |  |  |  |
| …… |  |  |  |  |  |
| 类目n |  |  |  |  |  |

词表内容分别为：1）由分词器处理后的所有关键词；2）训练集中出现过的所有类目；3）每个词在所属类目中出现的次数（WCN）（与类表中WCN同义）；4）每个词在各类目中出现的总次数（KWN）。用表格表示见表2。

表2 词表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 类目1 | 类目2 | …… | 类目m | 词在各类目出现总词数（KWN） |
| 关键词1 | （WCN） |  |  |  |  |
| 关键词2 |  |  |  |  |  |
| …… |  |  |  |  |  |
| 关键词n |  |  |  |  |  |

设某一确定关键词在某一类中的权值W=WCN/KWN，关键词阈值T=WCN/CSN；出于噪音考虑，我们设定T为0.05（即当某个关键词在确定的类目中的T<0.05,我们就可以认为这个关键词在此类中的分类贡献较低或直接认为是一个噪音，从而设定此关键词在本类中的W=0，只有当阈值T>0.05时，关键词才被认定有效，并计算其权值。）。

训练数据集时，我们采用了实时聚类的方式，将每一个样本由分词器提取的提取的关键字直接添加到子类的集合中。预测时，每当输入一个新的样本时，我们先用分词器提取其关键词。然后让其与所有类目进行对比，当关键词阈值满足要求时，计算样本中每个关键词在各类目中的权重。若不满足，则将其在该类目中的权值设为0。所有关键词对比并得到在各类目中的权值之后，计算每个类目中关键词权重总和，并选取得分较高的类目将待测样本分到此类目当中。

补充：鉴于有不少的类目中所有的样品关键词全为英文，我们通过数据库技术对一些权值较高的英文关键词进行了汉语翻译，以应对中文描述输入的场合从而提高了系统整体的精确度；由于本系统的计算量较大，我们也采用了多线程技术来降低计算的时间消耗，来达到更好的应用效果。

2.3系统流程图

系统流程图如图1所示。

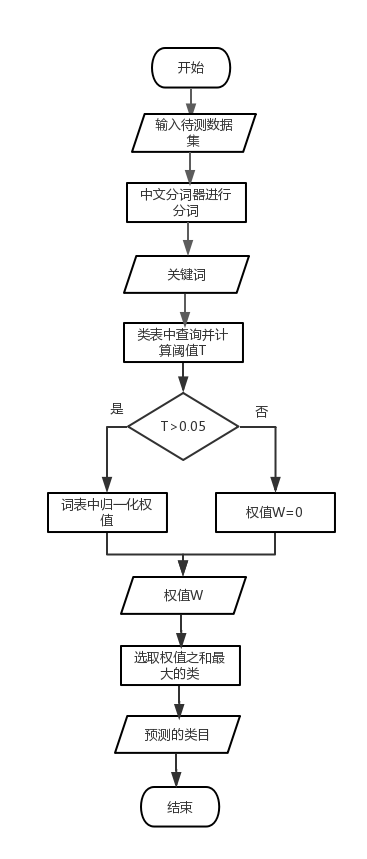


图1 系统流程图

1. **测试结果**

3.1 分词

在正式训练之前我们首先使用了分词工具对训练集进行处理，处理的结果如图2所示：

（注：不带‘\’的语句为原始的商品描述，与之相对的带‘\’的语句是分词器的运行结果）

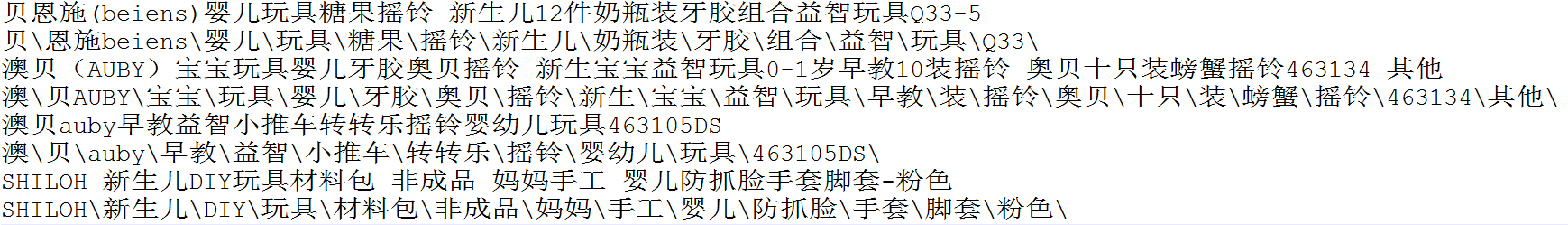


图2 原始分词结果

由上文的分词结果来看我们可以明显的观察到商品的品牌名称被分词器所分开例如：贝恩施为分为“贝\恩施”。我们认为，商品的品牌名称对类目的划分有很好的作用，就像当我们提及“iPhone”的时候，我们就会很快地反应此商品为智能手机属于数码产品类，因此部分的商品的品牌名有很大区分度。不过也存在例外，有些品牌可能会有跨界生产的情况，对此状况我们还有其余的关键词进行把控并不需要过多的关心。

为了确保商品名的正确划分我们添加了用户字典，分词结果如下：

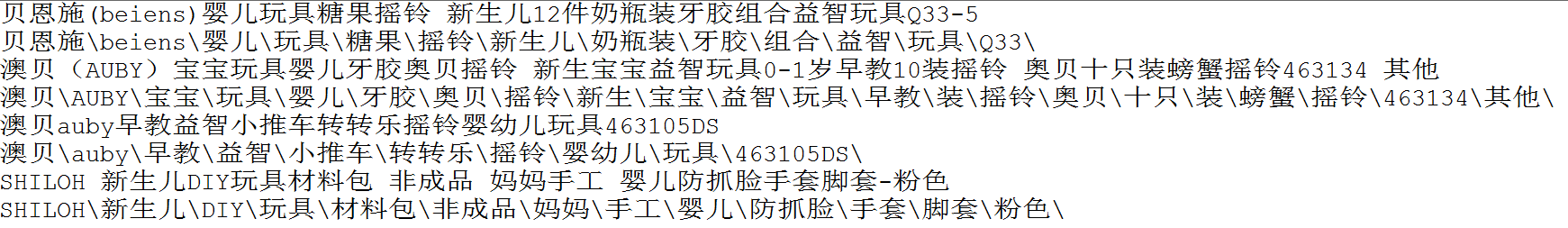


图3 改进的分词结果

很明显地此次的划分效果显著好于前者。由此也确保了我们的分类可靠性。

3.2 流程及正确率：

在先前处理完成之后，我们先将官方给的数据集重命名为train.data\_backup，然后按照1:1000的比例从中提取测试集test.data和训练集train.data，并将测试集的答案存做answer.data，留作对比、计算正确率之用；

接下来我们运行train.py，载入提取出的train.data训练集进行训练，输出参数配置文件parameter\_categorys.dat和parameter\_words.dat；

然后运行test.py，读入参数配置文件parameter\_categorys.dat和parameter\_words.dat，重建模型，并读入test.data测试集进行测试，输出测试结果testanswer.data；

最后运行successrate.py，对比testanswer.data和answer.data，输出正确率。

我们方案的正确率在73%以上，并随着训练集覆盖面的扩大和类目样本数的平均，正确率继续稳步上升。初始的正确率如图4所示：

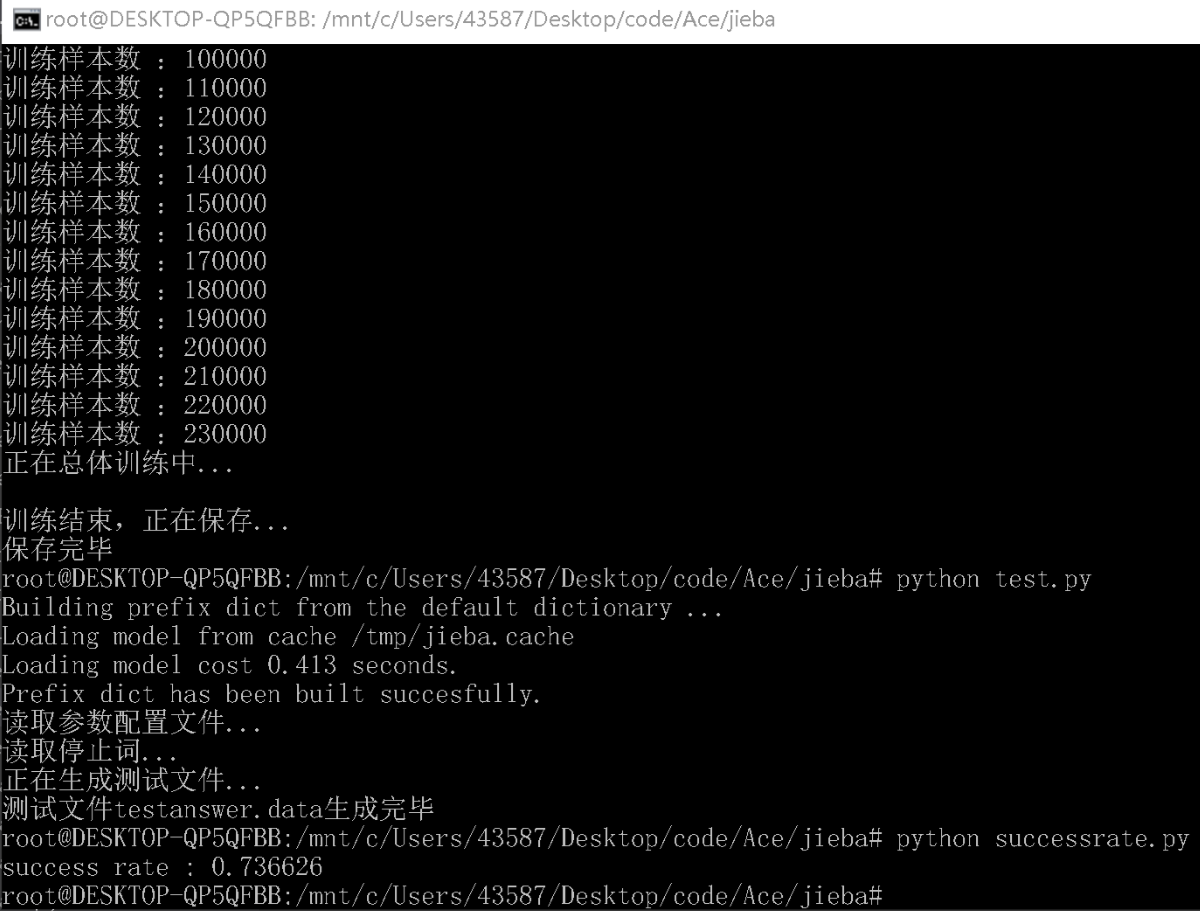


图4 本解决方案的正确率截图

为了能有一个直观的对比，我们采用了Google公司发布的词语分类聚类项目word2vec来做对比，word2vec的正确率如图5所示：

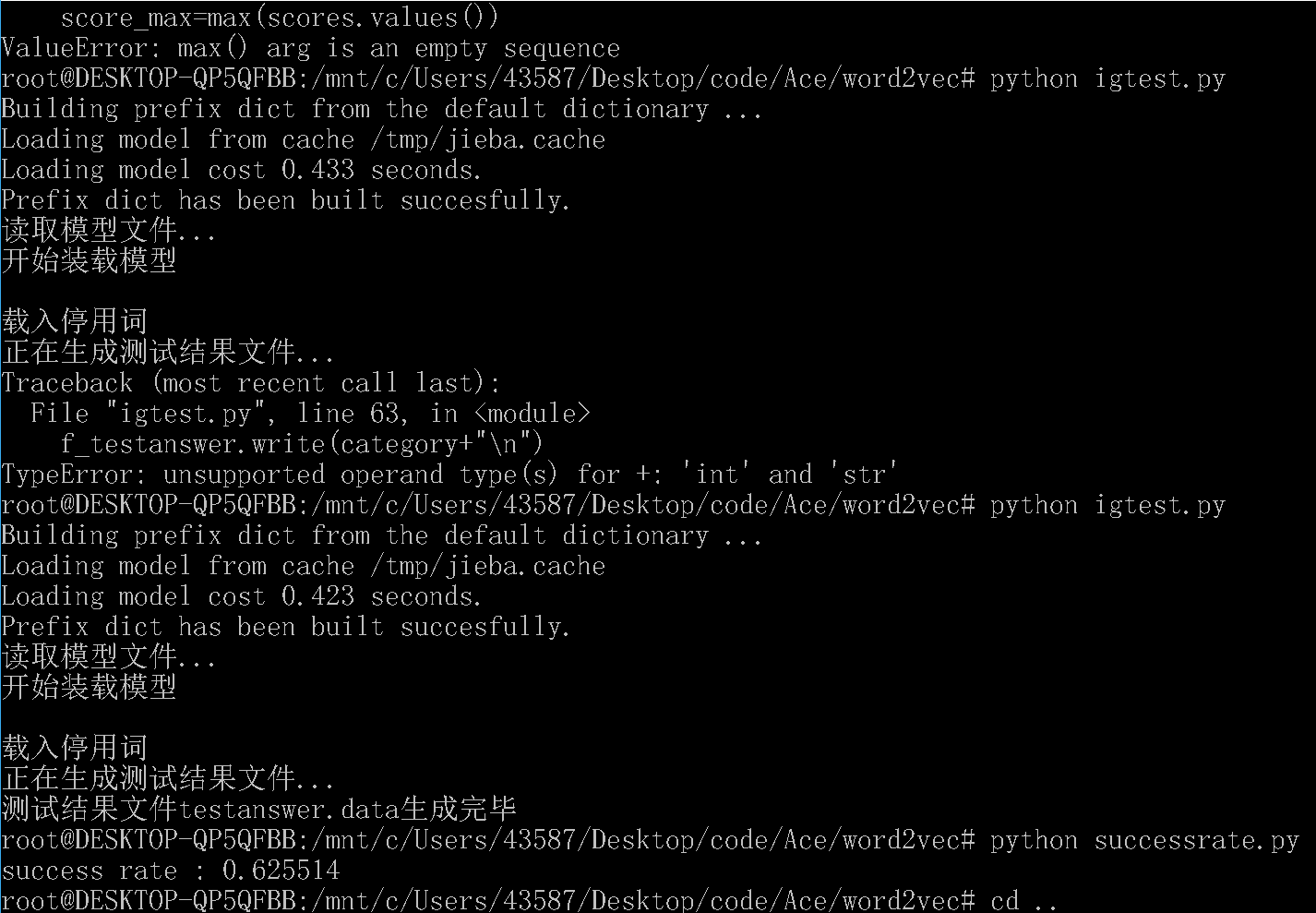


图5 word2vec的正确率截图

事实证明，本解决方案在针对于商品描述的短文本分类及聚类情景下，拥有比Google的word2vec更高的正确率，并且差距极大。

1. **设计总结**

随着物质生活的日益丰富，电商平台不断发展壮大。商品的数目也变得越发庞大。以往的人工分类已经不能满足当今的需求，于是基于机器学习在这次的工程里我们设计了一个聚类的算法。首先，对商品的描述进行自然语言的处理，抽取出用于分类的关键词。然后，将从商品描述中抽取出来的关键词作为一个向量并把向量的分词添加到此商品所属的类目的集合中，根据分词的出现频率更改其权重。最后，输入新的商品描述一并获取其分词向量，对比类目集合的分词权重将其归于计算权重最大的那各类目。自然语言的处理我们使用的是JAVA语言。具体的聚类实现则使用的是PYTHON语言。

为了确保工程的准确和高速，我们对“千米网”进行了系统地调查和分析，同时我们参与了官方的答疑会议，对目标问题获得更加深刻的理解。在github 平台上作了细致的分工以确保编程效率和兼容性。将给出的训练集导入算法中进行训练，获得最终的成果。在测试集上也取得了不错的效果。

现在我们主要面对的问题是：如果训练集的单项类目所给的样本数较少，则此算法不能获得良好运算效果，泛化能力也会因此较弱。若某一大类包含的样本数过大可能会产生压制样本数较少的小类的问题，从而使得在最终运行时很难有测试样本被归入其中。应对这个问题我们采用了归一化的处理办法来降低此影响（具体见算法和设计）。此外，本算法运算效率较低，在测试集较大的时候训练所花时间会大大延长。

将来的工作是使用更大的训练集提高本算法的泛化能力，提高其运行的效率并降低它的消耗。还希望可以采用迁移学习的方法将本算法应用到更广泛的领域，增强算法的重用性。