**Matrix题目分类与推荐系统**

**需求分析报告书**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **日期** | **内容说明** | **作者/修改人** |
| V1.0 | 2018.06.19 | 完整的需求分析报告 | 项目成员 |

**目录**

[**一、 项目简述** 8.3](#_Toc503936567)

[**二、 整体结构** 3](#_Toc503936568)

[**三、 具体实现** 5](#_Toc503936569)

[**3.1 确定标签** 5](#_Toc503936570)

[**3.2 题目分类** 6](#_Toc503936573)

[**3.3 题目划分** 14](#_Toc503936576)

[**3.4 题目推荐** 14](#_Toc503936576)

[**四、 成果展示** 17](#_Toc503936579)

[**五、 分工信息（排名不分先后）** 17](#_Toc503936582)

1. **项目简述**

本项目根据C和C++的课程中涉及的知识点，对matrix网站部分题库的题目进行打标签、分类，并按照不同题目的标签重合度进行推荐。

1. **整体结构**
2. **技术步骤**

实现首先根据C和C++课程中的一些基本的知识点进行标签的制定；将题库中的题目分类到标签上，每个题目包含不同的1—5个标签。然后将每个标签包含的题目输出到json文件里面（每一个标签对应一个json文件，每个文件中题目会有重复）。最后根据题目之间标签的重合度进行题目推荐。

1. **框架层级**

下图说明了整个题目分类和推荐的实现流程。

1. **具体实现**

基于前面所述的流程下，实现各个模块，组合后构成整个matrix题目分类和推荐系统。现根据上图所示流程涉及的技术功能点展开具体实现介绍，并记录开发中仍然存在的问题与完善计划。

**3.1 确定标签**

根据C和C++课程知识点制定标签。

|  |  |
| --- | --- |
| **人工实现标签制定** | |
| 实现描述 | **标签制定如下（根据字母排序）：**  **'all', 'algorithm', 'array', 'basic', 'binary tree', 'bit operation', 'class and objects', 'class template', 'condition', 'constructor', 'deconstructor', 'dynamic programming', 'exception', 'function overload', 'function template', 'function', 'inheritance', 'IO', 'linked list', 'loop', 'macro', 'memory', 'operator overload', 'pointer', 'polymorphism', 'queue', 'recursion', 'sort', 'stack', 'STL', 'string', 'struct'** |
| 存在的问题 | 标签涉及的知识点可能不全 |
| 完善计划 | 可以对标签继续进行细分，比如构造函数可以继续细分为默认构造函数、拷贝构造函数等等。 |

**3.2 题目分类**

|  |  |
| --- | --- |
|  | |
| 实现描述 | （需要描述problem的格式）利用机器学习的朴素贝叶斯算法进行文本分类，把json格式的题目描述和标签的作为训练集，预测未知的题目描述的标签。  加入训练集：      算法的实现用到了sklearn的nb分类器： |
| 存在的问题 | 分类准确率不高。 |
| 完善计划 | 需要配合人工辅助打标签并校正 |

**3.3 将题目划分到标签中**

|  |  |
| --- | --- |
| **每一个标签独立成一个文件，将题目划分到相应文件中** | |
| 实现描述 | 使用js代码，在node环境下使用fs库实现，。  （1）定义一个保存标签对象的数组。  标签对象示例如下：  微信截图_20180621130424  name字段作为标签对应的文件名，并用于匹配problem的tag字段。  problems数组存放属于此标签的problem。  保存标签对象的数组示例如下：  1529544381(1)   1. 对上面标签对象的name字段和problem的tags数组中的tag名字进行比较匹配，相同的就先保存在标签对象的problems数组中。最后再统一写入文件。代码截图如下：   微信截图_20180621092926 |
| 存在的问题 | 匹配采用三层循环，时间复杂度高，速度较慢。 |
| 完善计划 | 可以对匹配算法进行优化，使题目更快地分配入相应的标签文件中。 |

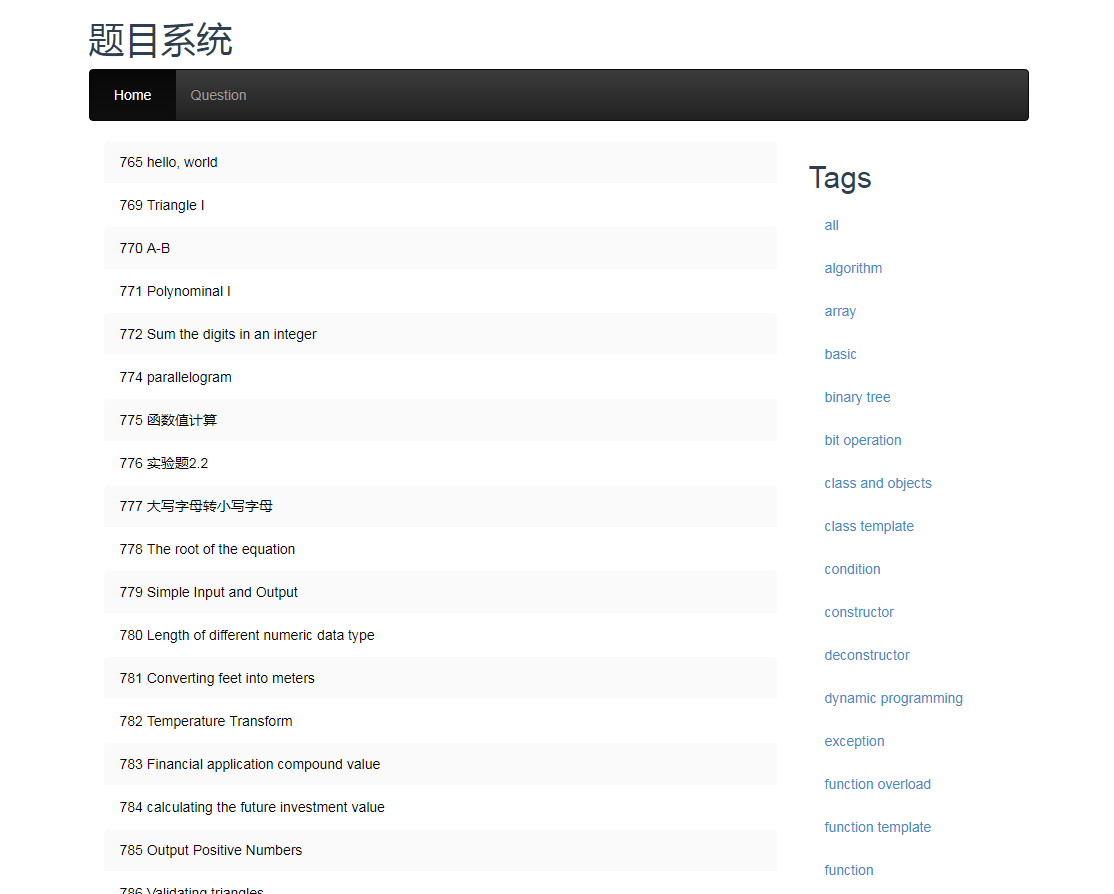
**3.4 题目推荐**

|  |  |
| --- | --- |
| **以一道题目为输入，输出与这道题目标签重合度最高的五道题目。** | |
| 实现描述 | 首先说明用到的变量：  微信截图_20180621124849  matched\_problems用来存放匹配到的题目，current\_problems\_tags为当前题目的标签数组。 从题目输入到输出五道题目，我将推荐算法分为以下4步：   1. 根据当前题目的标签，搜索从3.3得到的相应的标签文件，得到所有与当前题目至少有一个标签相同的题目，放入数组中（由于标签文件中题目有重复，因此数组中会有重复的题目，同时也有当前这道题目，因此需要去掉这些无关项）   实现如下：  微信截图_20180621124730   1. 去掉干扰项（重复的题目、与当前题目id相同的题目，即与自身相同的题目）   去掉与自身相同的题目代码如下：  微信截图_20180621125142  去掉重复题目的代码如下：  微信截图_20180621125149   1. 给每个匹配到的题目增加一个记录匹配度的变量：   微信截图_20180621130609  modified\_matched\_problems是增加匹配度字段后的数组   1. 根据匹配度从大到小排序（简单的选择排序，代码忽略），并将得到的5道匹配度最高的题目返回（假如有5道的话）：   微信截图_20180621130951 |
| 存在的问题 | 按照当前的算法，匹配度相同的题目中，id小的会优先推荐出来 |
| 完善计划 | 可以根据实际需求调整算法，调整匹配度相同题目推荐的优先程度。 |

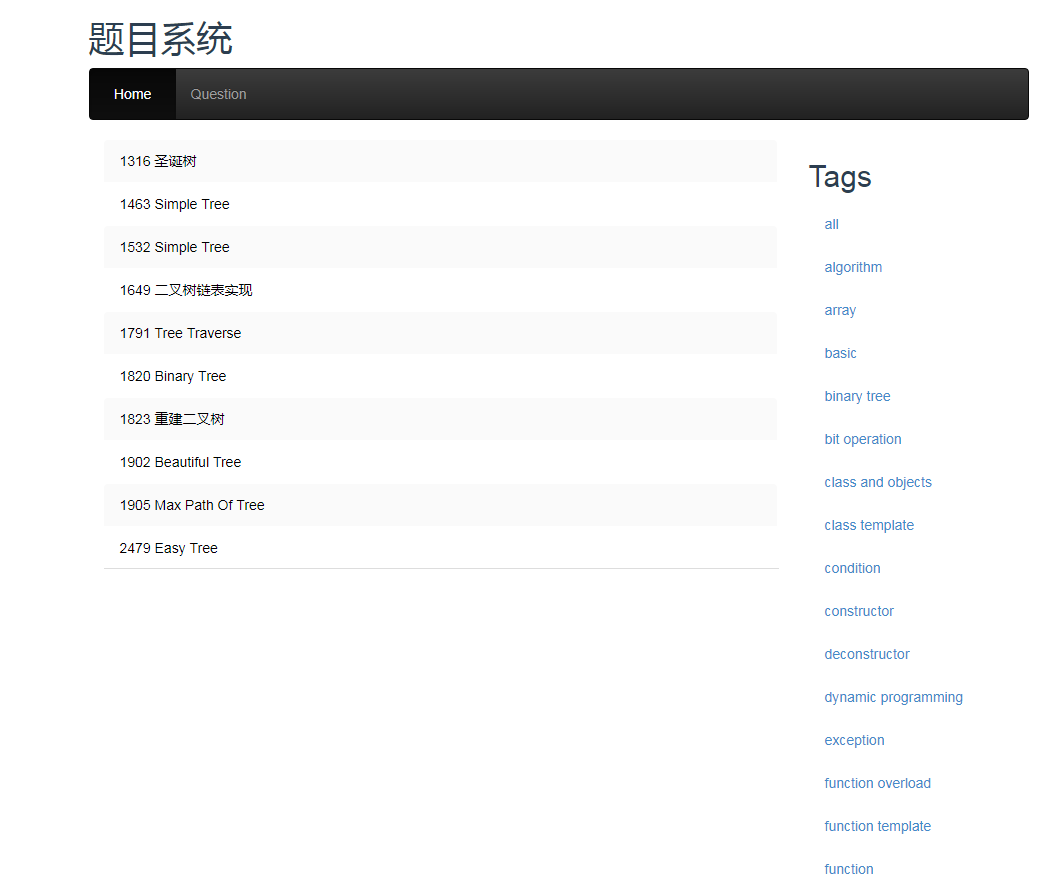
1. **成果展示**

以一个简单的页面来展示成果（用Vue开发）：

主页显示所有的题目：

****

点击右边的Tags显示属于该Tags的题目：（比如下图点击了binary tree）



点击具体的题目：



蓝色边框框住的是当前题目所属的标签，红色边框框住的是系统推荐相似的题目。

1. **分工信息（排名不分先后）**

|  |  |
| --- | --- |
| **姓名** | **负责部分** |
| 冯梓维 | 1.确定标签 2.题目推荐算法的设计实现 3.展示部分的前端和后端实现 |
| 陈亿枚 | 1.确定标签 2.需求文档分析 |
| 蔡冠文 | 1.确定标签 2.题目分类算法的设计实现 |