# 第五章 NLP详细设计

## 5.1.文档介绍

### 5.1.1.文档目的

*作为企业即时通软件设计文档的重要组成部分，本文档主要对该软件各个模块以及模块与模块之间的关系进行了详细描述，并对相关内容作出统一规定，清楚明确软件系统的设计方向,做好准备工作，从需求文档中和现实中抽离出实体，分好模块，明确分工，设计软件系统的整体框架以及各个模块、接口和实体类。*

### 5.1.2.文档范围

*1.文档介绍*

*2.模块命名规则*

*3.模块汇总*

### 5.1.3.读者对象

*1.本系统设计人员：包括模块设计人员*

*2.本系统的系统开发人员：编码人员*

*3.本系统的测试人员*

### 5.1.4.参考文献

崔梦婷，《nlp-需求分析说明书》

### 5.1.5.术语与缩写解释

|  |  |
| --- | --- |
| **缩写、术语** | **解 释** |
| SPP | 精简并行过程，Simplified Parallel Process |
| FD | 框架设计，Framework Design |

## 5.2.模块命名规则

实体类命名规则是使用驼峰命名法，如NlpClassify

包名全部小写，如nlp

数据库中的列名，如T\_NLP\_CLASSIFY

## 5.3.模块汇总

### 5.3.1.模块汇总表

|  |  |
| --- | --- |
| **NLP系统** | |
| NlpClassify | 文本分类 |
| **NLP系统** | |
| Nlpkeyword | 关键词提取 |
| **NLP系统** | |
| NlpWordTagging | 中文分词 |
| **NLP系统** | |
| NlpDependency | 依存关系 |
| **NLP系统** | |
| NlpAbstract | 摘要提取 |
| **NLP系统** | |
| NlpEntityRecognise | 实体识别 |
| **NLP系统** | |
| NlpEntityRelation | 实体关系 |

## 5.4. [A]NLP系统的模块设计

### 5.4.1. 模块A-1

|  |  |  |
| --- | --- | --- |
| 模块名称 | | NlpTextClassify |
| 功能描述 | | 文本分类 |
| 接口与属性 | | **Dao层实体继承**GenericTreeDao、BaseEntity  接口，实体test中继承GenericTreeDaoTestCase、还有testGetRoot()方法  实体属性：text,classify1,classify2,classify3,classify4  Service层建立NlpTextClassifyManager接口，和实现类NlpTextClassifyManagerImpl， |
| 数据结构  与算法 | | *无* |
| 补充说明 | |  |
| 补充说明 |  | |

## 5.5. [B]NLP系统的模块设计

### 5.5.1模块B-1

|  |  |
| --- | --- |
| 模块名称 | NlpKeyword |
| 功能描述 | 关键词提取 |
| 接口与属性 | **Dao层实体继承**GenericTreeDao、GenericDao  、BaseTreeEntity接口，实体test中继承GenericTreeDaoTestCase，有testRoot()  方法  实体属性：text  Service层建立NlpKeyword Manager接口，和实现类NlpKeyword rManagerImpl， |
| 数据结构  与算法 | *无* |
| 补充说明 |  |

## 5.6.[C]NLP系统的模块设计

### 5.6.1.模块C-1

|  |  |
| --- | --- |
| 模块名称 | NlpWordTagging |
| 功能描述 | 中文分词 |
| 接口与属性 | **Dao层实体继承**GenericTreeDao、BaseTreeEntity接口，实体test中继承GenericTreeDaoTestCase  实体属性：text  Service层建立NlpWordTagging Manager接口，和实现类NlpWordTagging ManagerImpl， |
| 数据结构  与算法 | *无* |
| 补充说明 |  |

## 5.7. [D]NLP系统的模块设计

### 5.7.1.模块D-1

|  |  |
| --- | --- |
| 模块名称 | NlpDependency |
| 功能描述 | 依存关系 |
| 接口与属性 | **Dao层实体继承**GenericTreeDao、BaseTreeEntity接口，实体test中继承GenericTreeDaoTestCase  实体属性：text  Service层建立NlpDependency Manager接口，和实现类NlpDependency ManagerImpl， |
| 数据结构  与算法 | *无* |
| 补充说明 |  |

## 5.8.[E]NLP系统的模块设计

### 5.8.1.模块E-1

|  |  |
| --- | --- |
| 模块名称 | NlpAbstract |
| 功能描述 | 摘要提取 |
| 接口与属性 | **Dao层实体继承**GenericTreeDao、BaseTreeEntity接口，实体test中继承GenericTreeDaoTestCase  实体属性：text  Service层建立NlpAbstract Manager接口，和实现类NlpAbstract ManagerImpl， |
| 数据结构  与算法 | *无* |
| 补充说明 |  |

## 5.9. [F]NLP系统的模块设计

### 5.9.1.模块F-1

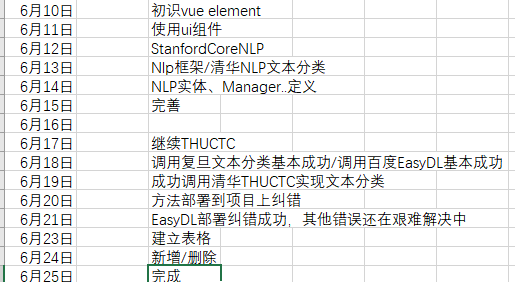
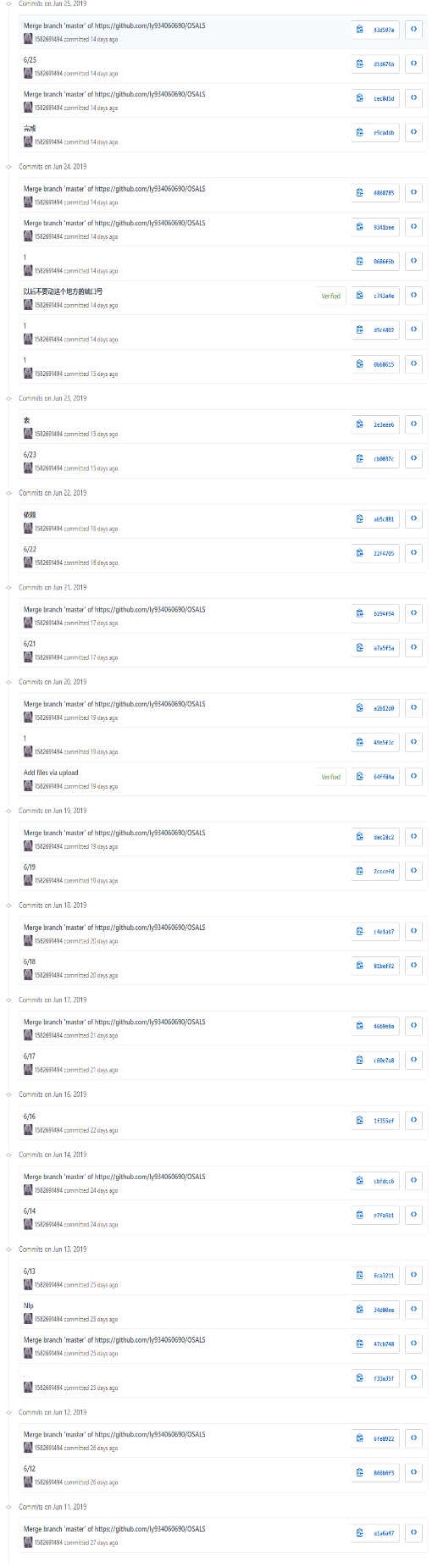
|  |  |
| --- | --- |
| 模块名称 | **NlpEntityRecognition** |
| 功能描述 | 实体识别 |
| 接口与属性 | **Dao层实体继承**GenericTreeDao、BaseTreeEntity接口，实体test中继承GenericTreeDaoTestCase  实体属性：text  Service层建立**NlpEntityRecognition** Manager接口，和实现类**NlpEntityRecognition** ManagerImpl |
| 数据结构  与算法 | *无* |
| 补充说明 |  |

## 5.10.[G]订单管理系统的模块设计

### 5.10.1.模块G-1

|  |  |
| --- | --- |
| 模块名称 | **NlpEntityRelation** |
| 功能描述 | 实体关系 |
| 接口与属性 | **Dao层实体继承**GenericTreeDao、BaseTreeEntity接口，实体test中继承GenericTreeDaoTestCase  实体属性：text  Service层建立**NlpEntityRelation** Manager接口，和实现类**NlpEntityRelation** ManagerImpl |
| 数据结构  与算法 | *无* |
| 补充说明 |  |

## 5.11.Git与Excel的记录



## 5.12.文本分类

### 5.11.1.功能描述

实体关系为自然语言分析的一个分析方面，可以将一段语言进行分类。

### 5.11.2.业务流程图

结束

输入一段文本

开始

EasyDL处**理**

输出结果,判断文本种类

## 5.13.总结

分类(Classification)是指自动对数据进行标注。人们在日常生活中通过经验划分类别。但是要依据一些规则手工地对互联网上的每一个页面进行分类，是不可能的。因此，基于计算机的高效自动分类技术成为人们解决互联网应用难题的迫切需求。与分类技术类似的是聚类，聚类不是将数据匹配到预先定义的标签集合，而是通过与其他数据相关的隐含结构自动的聚集为一个或多个类别。文本分类是数据挖掘和机器学习领域的一个重要研究方向。

  分类是信息检索领域多年来一直研究的课题，一方面以搜索的应用为目的来提高有效性和某些情况下的效率；另一方面，分类也是经典的机器学习技术。在机器学习领域，分类是在有标注的预定义类别体系下进行，因此属于有监督的学习问题；相反聚类则是一种无监督的学习问题。

  文本分类(Text Classification或Text Categorization，TC)，或者称为自动文本分类(Automatic Text Categorization)，是指计算机将载有信息的一篇文本映射到预先给定的某一类别或某几类别主题的过程。文本分类另外也属于自然语言处理领域。本文中文本(Text)和文档(Document)不加区分，具有相同的意义。

  F. Sebastiani以如下数学模型描述文本分类任务：文本分类的任务可以理解为获得这样的一个函数Φ：D×C→{T，F}，其中，D={d1,d2,…,d|D|} 表示需要进行分类的文档，C={c1,c2,…,c|C|} 表示预定义的分类体系下的类别集合，T值表示对于(dj,ci)来说 ，文档dj属于类ci，而F值表示对于(dj,ci)而言文档dj不属于类ci 。也就是说，文本分类的目标就是要寻找一个有价值的函数映射，准确的完成D×C到T/F值的函数映射，这个映射过程本质上讲就是所谓的分类器。

  文本分类的形式化定义如下：

  设i = 1，…，M为文档集合里面的M篇文档， j = 1，…，N为预先定义的N个类别主题，可以给出这样一个分类结果矩阵C=(cij)；其中，矩阵中某一元素cij表示第i篇文档与第j个类别的关系。也就是说，文本自动分类可以归结为确定上面矩阵C的每一个元素的值的过程；使用一个布尔量1或0，如果cij 的值为1，则表示文档i属于第j类，如果值为0，则文档i不能被分入类别j，即：

https://img-blog.csdnimg.cn/20181105191237448.png

对于单类别的分类，即某篇文档只允许被分入一个类别中，我们可以增加限定条件，对于第j行( j = 1，…，N)的所有元素，必须满足：

https://img-blog.csdnimg.cn/20181105191326524.png

  在实际应用中，根据预定义的类别不同，分类系统可以分两种：两类分类器和多类分类器。如果从文本的标注类别上来讲，文本分类又可以分为单标签和多标签两类。文本分类系统的任务简单的说：在预定义分类体系下，根据文本的内容相关性自动地判定文本与类别之间的关联。从数学角度来看，文本分类是一个函数映射过程，它将未标明类别的文本映射到预定义的类别，该映射可以是一一映射，也可以是一对多的映射，因为通常一篇文本可以同时关联到多个类别。

掌握了项目开发的流程。首先，进行需求分析，系统设计，全面的分析项目所需要实现的功能。最后编程逐步实现所需功能，在开发过程中不断完善项目功能，力求能够满足用户需求。

经过本学期的努力，系统终于按预期的结果顺利完成了。从系统最初的需求分析、设计，到最后实现的过程中，经过反复的修改，功能逐渐完善，已经基本达到了实用，方便，人性化等预期目标。通过本次课程，不仅学到了很多专业知识，而且提高了动手能力和解决实际问题的能力，真的受益匪浅。课题完成到现在，从开始连业务流程都不怎么熟悉，到现在完成基本需求功能，遇到了不少问题。本次课程是对JAVAEE所学知识的总结。在开发过程中，常常由于一点失误，导致整个程序的无法运行，这时，只有细心的跟踪调试、阅读相关书籍、查阅资料、向老师咨询，细心检查才能通过，整个过程中，小组团队各成员逐渐学会了当遇到问题时应该如何面对，如何找到最有效的解决方案。学习了系统化的开发思想。相信我们小组各成员以后的学习工作有很大的帮助。