技术设计文档

术语抽取系统

Terminologies Extraction System

文档创建者: Soul Mr. Zhao / 赵翔

创建日期: 2014/9/21

最近修改日期: 2014/9/21

项目地址: <https://github.com/soulzhao/termsDiscoverer>

当前版本: 2.0

状态: 编写中

模板版本: 1.51

文档版本控制

修改记录

| 修改日期 | 修改人 | 版本 | 修改内容 |
| --- | --- | --- | --- |
|  |  |  |  |
| 2014/9/21 | 赵翔 | 1.0 | 编写系统架构，目标平台，里程碑信息以及QA |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

审稿人

| 姓名 | 单位 | 职位 |
| --- | --- | --- |
|  |  |  |
| Soul, Mr. Zhao 赵翔 |  |  |
| Yingjie Zhang, 张英杰 |  |  |
|  |  |  |
|  |  |  |

参考文档

| 文档名 | 文档类型 | 文档地址 |
| --- | --- | --- |
|  |  |  |
| 术语抽取需求分析 | Local MS Word File | <requirements.docx> |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 目录

[一. 项目简介 4](#_Toc399183637)

[1.1 项目总览 4](#_Toc399183638)

[1.2 进度安排 4](#_Toc399183639)

[二. 技术方案 5](#_Toc399183640)

[2.1 目标平台技术 5](#_Toc399183641)

[2.2 基本架构 5](#_Toc399183642)

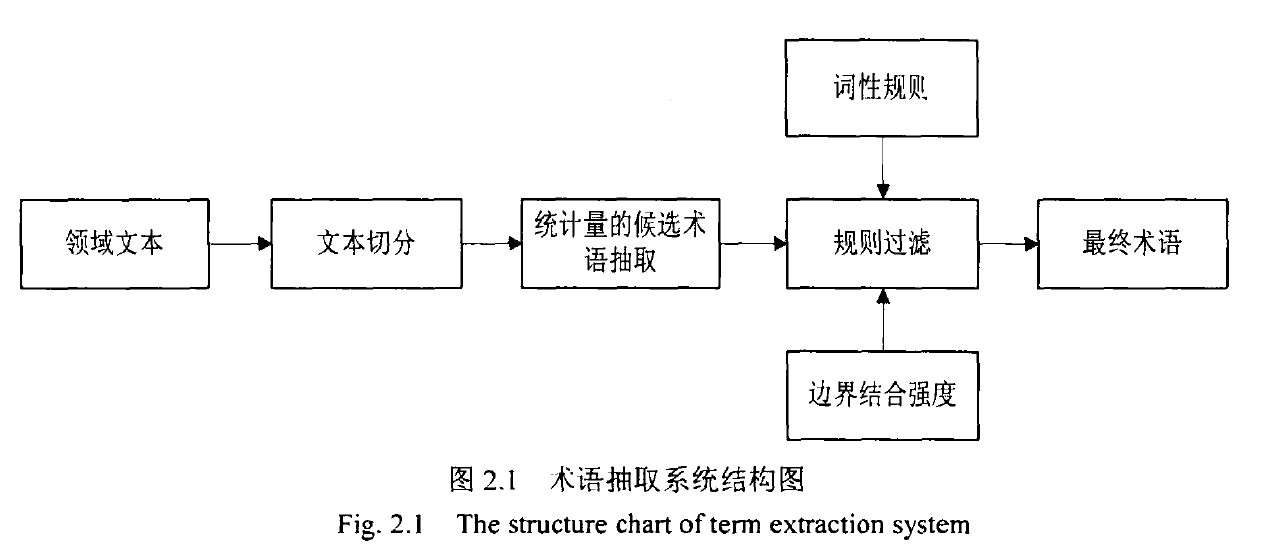
[2.3 核心细节 6](#_Toc399183643)

[三. Q & A 疑难点 7](#_Toc399183644)

# 项目简介

### 1.1 项目总览

术语抽取系统旨在对不同类型文章做处理，利用计算机自动抽识别和抽取出文章术语。术语抽取系统主要由三部分组成：文本切分，基于词频分布和信息熵的候选术语抽取，规则过滤。



### 1.2 进度安排

本项目采用里程碑管理，每个里程碑交付不同的成果。

项目分为四个阶段： 一. 分词阶段， 二. 候选词抽取阶段，三. 规则校验及过滤阶段，四，调试以及调整

| 阶段名 | 交付成果 | | | 预计日期 | |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | |  |
| **第一阶段：**  **分词阶段** | 输入： 文章文档  输出： Patricia Tree索引文件  词分布信息文件 | | | 2014/10/7 | |
| **第二阶段:**  **候选词抽取** | 输入：Patricia Tree索引文件（或直接内存索引）  词分布信息文件(或直接内存结构)  输出：候选词列表 | | | 2014/10/20 | |
| **第三阶段：**  **规则校验及过滤** | 输入：候选词列表  词分布信息文件(或直接内存结构)  输出：术语列表 | | | 2014/11/2 | |
| **第四阶段：**  **调试调整** | 为达到目标识别率，对规则等做调整 | | | 2014/11/08 | |

# 技术方案

### 2.1 目标平台技术

平台的选取要考虑到目标环境的真实情况，程序的性能要求，拓展性要求等情况。综合考虑，程序的运行目标平台和技术使用如下：

| 名称 | 使用 | | 备注 | | |
| --- | --- | --- | --- | --- | --- |
|  |  |  | |  |  |
| **运行平台** | Windows 7+ | |  | | |
| **编程语言** | C/C++语言 – 核心模块  Java语言做拓展编程 | | 利用C/C++语言实现PAT Tree索引的构建是出于效率考虑。  Java语言的选用是出于编程效率和学习成本考虑 | | |
| **开发工具** | MS Visual Studio 2010  Eclipse | |  | | |
| **环境要求** | JDK6+  .NET Framework3.0+ | |  | | |
| **数据库** | Oracle Database 11g Express Edition | | 免费版本 | | |

### 2.2 基本架构

系统架构如下：

**2.2.1 分词索引引擎**

**分词器选择：**

方案1： 分词部分采用张华平博士的NLPIR分词器实现分词效果，针对分词器的结果建立[patricia tree](http://www.baidu.com/link?url=iZbQ7vNOyuB1Pzw4Rl0FUMzvw5J6NL7pMNH-Ta53NsMs1vOPE8dgZ4qsFR2H4HjKhHFFzORD1Zqf4Vr4kJoTqK" \t "_blank)索引。

方案2： 采用Ansj\_seg分词器（开源java分词器，系张华平博士学生所作），好处是全Java平台编程，该分词器速度和效率上并不差。

**索引创建：**

方案1： 利用PAT tree自行实现索引树结构。好处是C或者C++语言实现，能与NLPIR无缝连接，可以再创建索引树的同时去记录词语在文章中的分布情况。坏处是自行实现略有难度。

方案2： 利用现有开源框架去完成词频统计工作，如Cedar、Darts等Double Array Trie框架。好处：简便易行，词语分布信息后期也很好处理。坏处：有一定风险性。

**2.2.2 候选词抽取器**

根据分词索引引擎的方案选择，候选词抽取器有不同的实现方式。

方案1： 若分词索引引擎实现为DLL形式，则可以多种方式直接调用，甚至可以直接访问索引内存地址。

方案2： 若分词索引引擎是一个执行程序，两者间通过索引文件内容交互。这样程序各部分功能分工更加明确，耦合性低。

**候选词评分规则：**

根据不同的候选词评分规则，同样的词语可能有不同的得分，如何协调各规则之间的情况，或者各种情况评分后的得分如何再组织，已经得分阈值的选定都是以后要细细斟酌的地方。

**2.2.3 规则过滤器**

待定。

### 2.3 核心细节

# Q & A 疑难点

1. 分词器User Define Dictionary 定制问题
2. 分词器的选择
3. 索引建设方案
4. 术语识别率调节问题