需求分析报告

# 1. 软件开发目的

实现对海量文本信息的文本关键词的自动提取，减少人工标注所需的大量人力物力，为自然语言处理的相关研究提供基础。

## 1.1 软件定位

### 1.1.1 待解决问题

1. 待处理文本数据库的构建；
2. 文本数据的预处理（去停用词等）；
3. 加权文本网络的构建；
4. 同义或近义词语节点的合并；
5. 加权复杂网络统计参数的计算；
6. 复杂网络经典算法的嵌入；
7. 复杂网络关键节点发现算法的实现；
8. 词语关键度阈值的合理选取；
9. 输出关键词个数的设定；
10. 文本网络的显示，包括初始文本网络、对关键词所对应关键节点突出强调的文本网络（可按需要手动点选是否显示）；
11. 不同类别文本的关键词提取

### 1.1.2 核心用户

从事自然语言处理研究（自动问答系统、自动摘要等）的相关人员或搜索引擎网站建设人员

### 1.1.3 规模与预算

初步规模：支持处理短篇文档，即节点数量较少的小规模网络，且处理过程实时、高效；

终极规模：支持处理长篇文档，即万级节点数量的大规模网络，且处理过程延时较低、响应速度较快；

预算： 2015.11-2015.12 完成平台需求分析；

2015.12概要设计，详细设计；

2016.1 编码与测试

2016.2 后期优化

## 1.2 软件长期扩展计划

### 1.2.1 推广计划

在完成上述待处理问题的基础上，可按开发情况加入下述功能：

1）文本网络的立体结构图显示。

2）文本情感倾向分析功能

### 1.2.2 目标发展方向

平台关键词提取功能可作为接口直接被其他软件或搜索引擎调用

### 1.2.3 版本演进周期

拟循序渐进开发以下版本：

1）纯净版：复杂网络平台，只包含文件读取、经典网络生成、网络基本参数计算、K-Shell分解、网络加权、显示功能各功能模块。显示系统支持用户定制节点颜色或形状、连边颜色及粗细程度等，支持手动拖拽节点。

2）重要节点发现版：在纯净版基础上，加入社团划分、SIR传播等功能模块，应用于网络拓扑结构、重要节点发现及谣言传播等研究领域。

3）关键词提取分析版：在重要节点发现版的基础上，加入文本预处理、文本网络构建、词语相似度计算、关键词提取功能模块，应用于语义分析领域的文本关键词提取研究。

4）用户演示版：只包含文件读取、经典网络生成及显示功能模块，专注于视觉效果，拟采用direct X 3D绘图，极大化用户使用体验。

# 2. 软件需求

## 2.1 软件功能列表

|  |  |
| --- | --- |
| 纯净版 | |
| 功能模块 | 子功能 |
| 文件读写 | 读文件  写文件 |
| 网络生成 | ER随机网络  BA无标度网络  WS小世界网络 |
| 参数计算 | 网络参数  节点参数 |
| 网络加权 | 节点权重  连边权重 |
| 显示功能 | 网络结构图  关系曲线图 |
| 重要节点发现版 | |
| K-Shell分解 | / |
| 社团划分 | / |
| SIR传播 | SIR属于复杂网络传播领域，夏师姐论文已经讨论了复杂网络谣言传播上，重要节点防御和随机节点防御并无结果差异。所以重要节点发现算法对SIR传播并无明显帮助。（建议提取出来一个复杂网络信息传播版本族，即孟师姐主要方向） |
| 关键词提取版本 | |
| 文本预处理 | 去停用词  去无用词性词语 |
| 文本网络构建 | 基础网络构建 |
| 文本网络优化 | 词语相似度计算 |
| 文本关键词提取 | 方法一：统计参数加权  方法二：提出的新算法 |

（王兵使用SIR来验证节点重要性的依据是什么？是认为去除重要的节点后，传播就能被更快抑制吗？）

（你的模型是文档网络，和SIR的人际关系网络存在本质差异，选取的验证模型有问题。可以采用标准结果比对法来验证，即使用已经有关键词的文本，看处理后输出的关键词是否能命中作者的关键词。（这也是验证缓存调度准确率的衡量方法，命中率））



## 2.2 功能分类与优先级

### 2.2.1核心功能

1. 参数计算：网络参数计算、节点参数计算
2. 文本网络构建：基础网络构建

预处理后剩余词语作为候选关键词集，按照连边规则（同一个句子中距离小于等于2的词语之间直接产生连边）构建成复杂网络，根据词语位置及词性为词语节点赋以权重，根据词语在同一个句子中的共现频率给连边加权。

1. 文本网络优化：词语相似度计算

通过计算词语的相似度，找出同义词或近义词（即相似节点），合并这些相似的词语节点及其连边，则网络节点数量相对减少，且不存在相同或相似内容的节点（即降低了网络冗余度），使得找出的关键词更准确，实现网络优化。

1. 文本关键词提取：方法二

结合复杂网络的统计参数及经典算法，提出一种网络重要节点发现新算法，找出的重要节点对应的节点内容就是文本关键词。

### 2.2.2 必要功能

1. 文件读写
2. 文本预处理

目标文本为使用分词软件NLPIR得到的分词结果，保存至txt文档中，再由平台以字符串形式读入，首先系统记录词语位于第几个句子以及在句子中的位置，而后通过比对事先读入的一个停用词表和词性表，如果文档中的词语包含停用词表中的词语或者词性表中的词性，则直接删除，剩下的词语作为候选关键词集。

### 2.2.3 外围功能

1. 网络生成

2）显示功能

### 2.2.4 辅助功能

1. K-Shell分解
2. 社团划分
3. SIR传播

### 2.2.5 非需求功能

1）处理千级节点网络时响应时间低于3s；

2）处理万级节点网络时响应速度低于10s；

总评： 本文对软件的大致框架已经叙述出来，但是对于以下问题没有具体表述。

1、文本关键词从哪来？ 如何存在于系统中？是个列表还是网络？

文本构建成复杂网络后，文本中的每个词语有其对应的节点编号，节点内容即词语则存储在节点类中的内容列表。（此时不是已经生成节点了吗？都放在节点里，为啥还需要这两个列表？。最终在找出网络重要节点之后，输出对应编号节点的内容（即词语）即可。

2、目标文本（操作对象）如何存在于系统中？字符串还是网络？如果是网络，如何将读入文本变成网络？（目标文本指待提取关键字的文本）

目标文本为使用分词软件NLPIR得到的分词结果（目标文本之前已经读入系统了，为何还要再使用第三方软件提取结果？），保存至txt文档中，再由平台以字符串形式读入，首先系统记录词语位于第几个句子以及在句子中的位置，而后经过预处理之后，将剩余词语按照连边规则（同一个句子中距离小于等于2的词语之间直接产生连边）构建成复杂网络，根据词语位置及词性对词语节点进行状态编号及类型编号（应该是类型编号吧，名词 =1 ，动词 =2 之类的），而后根据词语的状态编号及类型编号为词语节点加权，根据词语在同一个句子中的共现频率给连边加权。

(ROM和RAM的区别？ 区分系统数据与用户数据的意义)

系统数据库就像是ROM一样，存放系统执行需要使用的数据，而且通常是不易变数据。但用户数据是日常操作流入，流出的数据，每次使用都不同，就像RAM中的数据。使用数据库的意义就是将系统数据和用户数据的操作进行分割。保证系统的数据无关性（即和用户的解耦合）。下图就是系统数据和用户数据的关系。

系统数据作为辅助，单方面向应用软件提供必要参数，只在极少数情况下修改。用户数据作为软件的操作对象，经过处理后输出结果，随后就关闭用户输入数据。且操作软件不能对用户输入数据有修改操作，更要避免存储用户数据，在大规模作业吞吐量的情况下，会对系统数据库造成巨大压力。





下图1为平台技术流程图，图2为数据流图，图3为例句分词结果及说明。



图1 技术流程示意图



图2 数据流图

其中，图1中的文本预处理即包括图2中的：平台分割字符串、记录词语状态编号及类型编号、去停用词及相关词性词语。图2最终得到的文本网络即无权网络，接着按照图1所示的技术流程进行下一步网络加权及优化操作等。

连边规则：1.同一个句子中距离小于等于2的词语之间产生连边。

2.根据两个词在多个句子中以间距为2共现的频率增加其连边权重。



图3 例句分词结果

分词软件标注的标点包括：左右括号、左右引号、句号、问号、叹号、逗号、分号、顿号、冒号、省略号、破折号、百分号、千分号、单位符号（如￥等）。

构建标点符号识别策略如下：

1.连边规则中的句子定义为：以句号、叹号、逗号、分号、省略号分隔。

2.标点符号括号、引号、顿号、冒号、破折号本身作零间距计算（零间距不就是同一个词吗？间距为1比较妥当吧）。

3.百分号、千分号、单位符号在分词软件中会直接与其修饰的数字分为一个词，如5.7%作为数词词性存在，故此三种标点无需考虑。

3. 目标词提取的核心算法是什么？ 节点重要程度算法吗？说明本软件从属哪个的复杂网络研究方向？

将文本构建成为网络后，目标词即关键词就是网络的重要节点，属于复杂网络重要节点发现研究领域。

4. 关键词库和读入生成的目标文本网络如何交互？

没有关键词库，每一篇读入的已分词文本对应3~6个关键词，也就是说在一个文本网络中找出排名前3或者前6的重要节点，这些节点所对应的内容就是关键词。

系统在对分词文档进行预处理的时候，事先会读入一个停用词表和词性表，如果文档中的词语包含停用词表中的词语或者词性表中的词性，则直接删除，剩下的词语作为候选关键词集。

重点是文本关键词网络的来源（数据库？网络文件？），在系统中如何存在（所有目标文本都使用相同的关键词库？）

没有文本关键词网络，只有文本数据。拟使用的文本数据包括新闻、核心期刊论文摘要，也就是短篇文本的关键词提取。首先人工使用分词软件进行分词，得到所有文本数据的已分词文本文档，存入数据库。（已分词文档是用户数据，与系统无关，为何要存入系统数据库？）

5. 数据库的用途说明（目前只发现需要使用三张表，且之间没有E-R关系）

1）、存放停用词（tabIgnore）和词性表(tabWordType)，这两张表如果需要维护（增删），就需要增加有关的操作窗口。

停用词表是从网上下载的，如“哈工大停用词表”等，目前我下载的有4张停用词表，后期这些表如果被不断更新，我们的数据库中存放的“停用词表”也应该支持更新。而词性表是由用户手动编写，去除个人觉得不可能作为关键词的相应词性词语，如果存放在数据库，也必须支持手动增删。

2）、存放用户样式，包括节点前景色，背景色，边框色，字体，图像缩放尺寸等信息。 （tabStyle）

6. 思考题：架构的纠结。

问题描述：从设计之初，我们预定的基本设计原则就是通用平台+可裁剪功能的1+N结构。举个例子，输入的网络文件可以是多种（XML,MAT,SST），但他们都被统一地读入系统平台，平台完成一系列公用操作（显示，参数计算，图标绘制），随后再通过拷贝构造函数转化为具体的功能网络，进行不同的网络研究领域的计算。如下图（左）所示。



漏斗形结构作为OSI网络框架的模型，重要的作用就是将底层千变万化的物理承载格式和上层千变万化的软件应用模式联系起来。上图右侧就是TCP/IP模型中的结构。左右二者都是使用一个平台层作为承上启下的角色。

我们在本软件中的一个原则就是所有输入的网络文件都要读入到cNet类中，然后经由它再拷贝构造生成具体的功能网络。

但是，在关键词提取算法中，必须要在输入文件中包含（词语内容，词语位置，词性）等数据。如果照常读入Cnet类中，必然会造成数据损失。但如果在平台的节点中加入以上属性，又会破坏平台的一般性。第三种方法就是在特殊功能窗口中单独读入文件，构建关键词网络kNet。但这样就会破坏单一输入接口原则，同时kNet也无法使用cNet的绘图和节点参数计算功能。

现在有两种处理方向：

1，重新设计系统架构，设计一个具有更高扩展性的架构。

2，依照原样继续开发平台，作为主线版本。然后单独拉出一个版本分支，作为专门给关键词发现使用的特型平台网络，cNode节点中加入词性等属性，以后在此本领域继续使用下去。

方案一的好处在于通用型，以后有类似的扩展，不用大规模修改代码。但是缺点是需要花时间。

方案二好处是简单，但版本之间过早出现差异，后期维护起来会比较麻烦。

核心争议点就是时间花在哪？花在现在，则以后简单。现在简单，则以后维护起来费劲。