**系统概要设计**

**一、系统流程框图：**

蛋白质识别

物种识别

原始文档数据

实体识别后的文档

最终结果文档

Normalization

蛋白质字典

物种字典

文档预处理

文档数据文件读取

蛋白质文件读取

物种文件读取

名称标准化

文档数据文件

蛋白质字典文件

物种字典文件

实体细化、过滤

蛋白ID---物种对应关系

输出

**二、数据结构：**

主要需要两种数据结构，一种用来储存表示一篇文档，另一种是匹配时用的字典。

**（1）文档数据结构：**

一个这样的结构表示一篇文档，一篇文档中按摘要和全文分别储存分好的句子。每一个句子上再存这自己被分成的token以及从本句上发现的实体。整个程序围绕这个数据结构展开，各个步骤做要做的工作就是填充、修改这个数据结构中的内容。

如下图所示：

Document

Abstract

Full text

Sentence

Sentence

Sentence

Sentence

Sentence

Sentence

token

token

token

entity

entity

**（2）字典数据结构：**

一个字典首先是一个词条（term）的集合。在集合的基础上，还要提供一定的索引以便查询。我们目前需要两种索引，一种是根据term的ID进行索引（蛋白质的prot AC，物种的NCBI id等），另一种是根据词语组织的类似于倒排文档的索引。一般的生物学字典，每一个term都会有一些描述信息，一般是这个term在文章中可能出现的名字等，将这些描述信息的每个词挑出来，索引到所有包含该词的term的集合。蛋白ID---物种对应关系也使用这样的字典表示，只是使用蛋白的id对物种的term做索引。

这个字典可以每次从文本中取词做索引，也可以将索引结构存储在一个文件中，每次读取这个文件直接建立索引。经过试验，这样能快一些。

ID

ID

ID

ID

Term

Term

Term

Term

Token

Token

Token

Token

Term

Term

Term

Term

Term

Term

Term

Term

Term

**三、处理过程：**

**（1）文档预处理：**

目前需要做的主要工作是分句。前面的步骤中从原始文件读到的文档数据是摘要和全文各一个大段，这里需要将这两个大段分句，得到每一个句子的位置信息，将“句子”这样一个数据结构建立好并与文档一起组织好。使用的分句工具可以是openNLP。

**（2）名称标准化：**

对文档中的文本做一些字符串上的处理，使一些名字的不同写法尽量统一。比如，将罗马字“III IV XI”等换成阿拉伯数字，将希腊字母换成英文写法，将横杠斜杠换成空格等。

这里应该提供一个接口，输入为字符串，输出为标准化后的字符串，方便其他地方调用。

**（3）蛋白质识别、物种识别：**

这两个步骤的逻辑应该是完全一样的，只是使用的字典不同。

处理步骤：

1. 对于每一个句子的每一个token，利用字典的索引找出包含该token的所有term。
2. 对邻近的token找出的term集合进行分析，根据相同的term将token尽量大的合并成组。
3. 对每个组对应的词在字典中再确认一下，如果通过就作为一个识别出的entity。

**（4）实体细化、过滤：**

之所以需要这样一个步骤，是因为经过实践发现，仅凭前面所说的字典匹配方法所得到的结果十分粗糙，字典中有些噪音也会造成很恶劣的影响，比如有些纯数字、单个字母被识别成了实体，而去查一下字典文件发现还真有。所以需要一个步骤将识别出来的实体过滤一下，去掉明显不对的。目前只是用了最简单的方法，用几个简单的正则表达式和常用词表进行过滤。

这个部分还有很多工作可以做，比如使用学习的方法（CRF等）在字典识别的基础上用BIO标签对实体的边界进行界定，字典识别出的结果可以作为一维或几维的特征，应该对机器学习的结果有很大帮助。这样也可以看做是背景知识库（字典）和学习模型的结合。

**（5）Normalization：**

前面几步得出来的entity结果中，肯定有很多对应着多个term，也就是有多个id，这个步骤就是要尽量将多个id进行消减。目前采用的是根据物种进行消歧，也就是根据之前对物种的字典匹配结果。尽量找到一个entity应该属于的物种，然后根据id和物种之间的对应关系，去掉不对的id。

首先使用规则，如果一个蛋白质entity在同一句的前面有物种entity，那么就认为这个蛋白质是这个物种的。

然后，统计整个文档的物种出现的次数，可以设定一些阈值，阈值以上的就认为是这篇文档主要讨论的物种，剩下的蛋白质entity就都认为是这些物种的。

**（6）输出：**

输出的格式目前是一篇文档一个文件，pmid.txt。文件里一句一句地打印，每个句子后面跟着这个句子中的entity。每个句子和entity都给出位置信息。

**四、外部资源：**

**（1）蛋白质字典文件：**

使用丁石林留下的Onbires系统代码中rlt\_sprot\_20061003这个文件。

**（2）物种字典文件：**

使用丁石林留下的Onbires系统代码中FullNameNew.txt

**（3）停用词表：**

在字典进行索引和对token进行字典查询的时候要把停用词过滤掉，所以需要停用词表。但是这个表只是要and the这样的停用词，也就是十几个的样子，而不是常用词表。因为一些常用词还是会出现在实体名字中的，不能过滤掉。

**（4）名称标准化时的替代规则：**

需要一个文件记录在对名称标准化时将哪些词替换成哪些词，比如“iv|5”这样的。

**（5）实体过滤的规则：**

在实体粗略识别之后进行过滤是使用的规则。应该是一些正则表达式，如果一个实体符合，就把它去掉。这里面应该有一个常用词表，因为如果一个实体只有一个词，还是常用词的话，那么应该被去掉。