1. **查询聚类，为的是方便寻找公共结构**

**输入：**多个自然语句的查询

**内部动作：**

1. 使用NLP中计算句子相似度的方法（大量的方法），将这些查询进行分组

2. 将查询语句映射成为SPARQL查询

**输出：**分组后的查询语句

**但是，**这么分组以后很可能面临负载不均衡的问题。很可能出现这种情况，100句话中，99句是类似的，只有一句是不类似的。那就一组有99个查询，一组只有一个查询。出现了很严重的负载均衡问题。

**Example:**

99个相似查询的一类，1个不相同的查询一类

但是，服务器的并行度是4.也就是说当前我们的服务器允许同时做四个查询。那问题就出现了，如何将这两组查询，尽可能平均分配成四组呢，这就是下面负载均衡的问题。

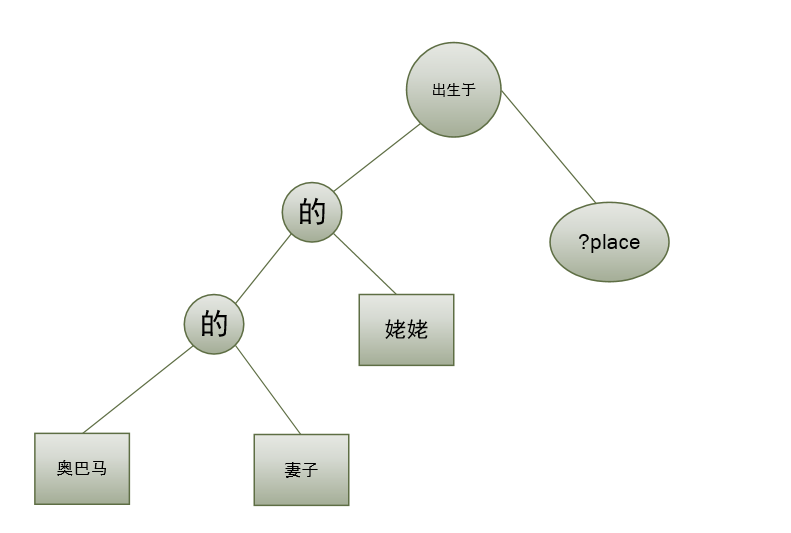
奥巴马的妻子的姥姥出生于?place?

名词：奥巴马，妻子，姥姥

谓词：的，的，出生于

变量：?place

基于谓词将语句分割成两部分。最后的结果如下所示：



1. **实现负载均衡：（基于服务器并行度以及由机器学习确定的查询代价）**

**输入：**（1）中分组了的查询语句

**输出：**基于电脑并行度的比较负载均衡的分组方式

**内部动作：**

1. 假设有一百个查询，如果平均分配那么一组应该有25个查询，同时为了保证（1）中的结果是有效的，我们要尽可能的将（1）中的同类查询分配到同一组中去。
2. 将1中分组的结果按照查询代价排序，经过1之后有了初步的分组结果。但是这还不够，原因在于，尽管查询语句的数量是平均的，但是每个查询耗费的代价是不相同的。因此不同组之间总查询代价极有可能差距也比较大。
3. 第一次平均：找两组差距最大的。例如上面分组后，四组查询代价分别是，100,17,50,30。就把100这组随机挑一些查询放到17那一组。每次找代价最大的一组和最小的一组，分配查询。
4. 启发式算法交换查询语句：经过3之后，四组查询代价基本均衡了，但是还可能存在某种排列方式，能整体降低每一组查询的代价。这时候就随机交换不同组内的同一类查询。
5. **组内执行方式**

尽量使得一个复杂的查询中产生的某一个中间结果是这一组中的一个较为简单的查询。

1. 在adpart上做修改，在原有基础上添加文件处理部分。不完全依赖于内存

意见：

希望能真正从语义角度上实现删除部分查询。语义部分上而不是结构上。

我目前想到的是基本的结构上的优化，但是更好的应该是从语义上提出优化