

多场景分割

刘荣

2014-11-15

目录

- 三元组处理
- 多场景识别应用

三元组处理数据准备

1. 实体列表

如以book qa为例，需要书名，作者名..

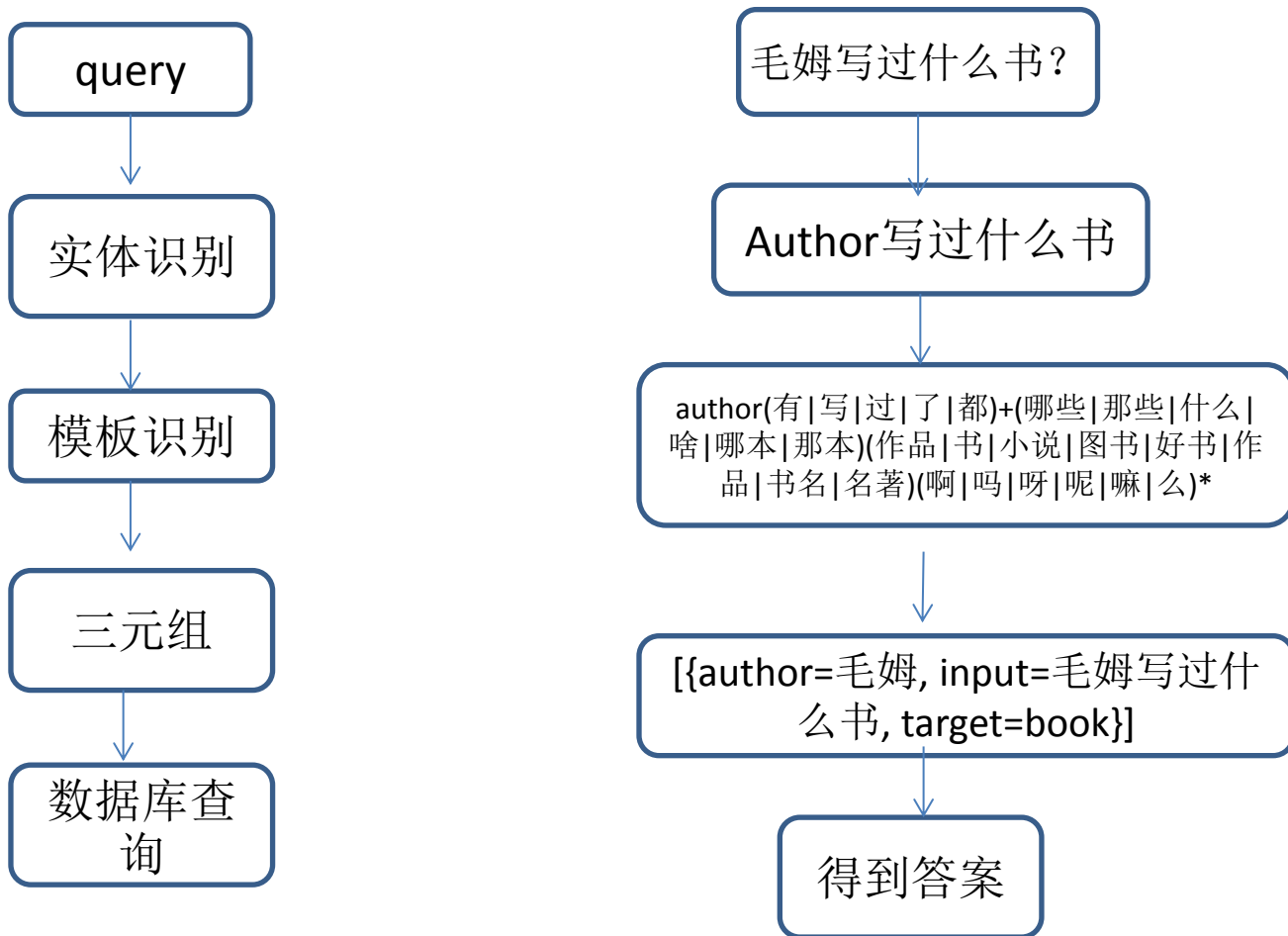
2. 数据库关系

如<人名，作者，书>，<书名，出版，出版单位>..

3. 模板

author(有|写|过)+(哪些|那些|什么)(作品|书|小说|图书|好书)(啊|吗|呀)* book
模板 目标

三元组处理流程

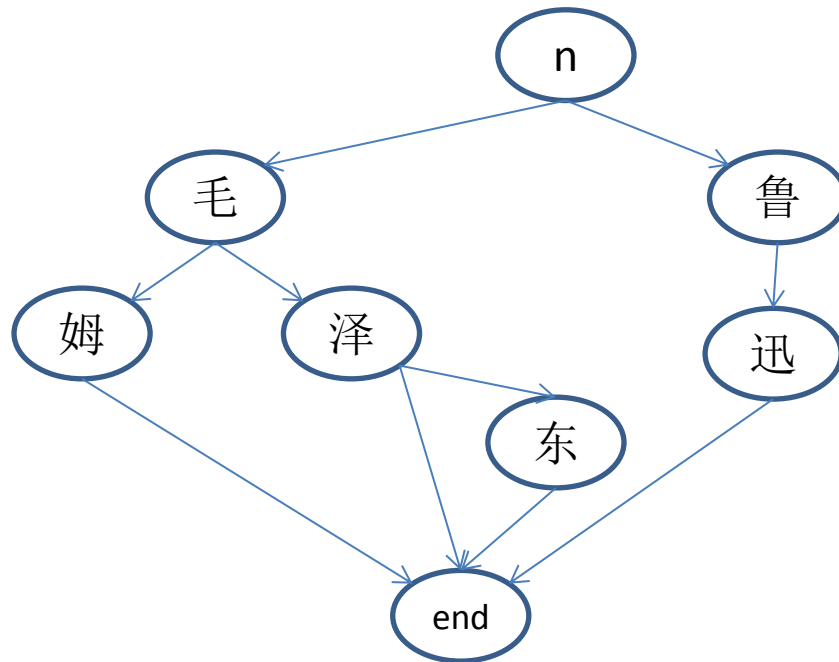


三元组处理细节—实体识别

1. 现有的实体

如“毛姆”，“毛泽东”，“鲁迅”，“毛泽”

2. 实体树



3. 搜索

采用最长匹配，如“毛泽东写过什么书”=>“author 写过什么书”

三元组处理细节—模板匹配

1. 模板

如” book的(谁|作家|作者|写的|文学家)(是|谁)* author

2. 匹配方法

2.1 正则匹配(目前采用的)

没有模糊性，在写模板时需要尽可能的覆盖更多的情况

2.2 模糊匹配

支持模糊匹配，但是对于目前的三元组而言，一般是精确查找，答案唯一对应，用户体验上有可能比较差。

三元组处理细节—数据库查找

1. 三元组

如” [{author=毛姆, input=毛姆写过什么书, target=book}]”

2. 生成查询语句

2.1 程序规则生成(目前采用)

“select author from book”(只是演示)

2.2 自动生成

SPARQL (需要采用RDF实体关系组织形式)

多场景识别

1. 三元组与多场景识别

建立在三元组形式上的多场景识别可以转化为实体属性的分类，既不同的数据库里实体关系对于不同的场景，对于实体的识别和模板的识别中就自动进行了多场景识别。

2. 具体实例

如“鲁迅写过什么书”“那英唱过什么歌”“南山区的热线电话是多少” →
“author 写过什么书”“singer 唱过什么歌”“government 的热线电话是多少”

3. 问题

3.1 歧义问题

3.2 复杂关系问题

总结

- 首先只是目前我们用的三元组的基于知识库的问答系统的处理流程，有可能不是很详细或者还有许多问题，但是这个可以作为第一个版本的Demo.
- 有什么问题可以随时讨论。