浙江大学毕业设计

本科学生中期报告

学生姓名　　周自强
实习单位/实验室　　浙江大学
项目名称　基于大规模知识图谱的规则挖掘系统的实现
导师姓名　　陈华钧
合作导师姓名
项目开始日期　　**2016-3-30**
项目结束日期　　**2016-5-30**

填表日期：　　2016 年　4 月　27 日

一、 项目概况

最近几年，许多知识库例如 Cyc、YAGO、DBpedia、Freebase 兴起。这些知识库由数百万的世界实体以及它们之间的关系所组成，以有向图的形式存储，结点表示实体，连接表示实体间的关系。尽管这样的知识库包含成千上万的实体，它仍然是稀疏的，即它在实体之间缺失了大量的关系。

知识库并不是完整的。这时候，我们需要从已有的知识库中进行规则挖掘，进而找寻两个实体之间的关系，使得知识图谱密集化。

目标和任务：实现一个规则挖掘的系统，分析大规模知识图谱，进行规则挖掘，通过挖掘出的规则进行知识库的补全。

具体分为以下几个部分：

1. 以 AMIE 为基础，分析大规模知识图谱，在知识图谱中实现封闭规则挖掘。

2. 在 AMIE 的基础上编写代码，进一步完善规则挖掘算法的细节，并使用提供的知识库数据进行挖掘测试。

3. 提供对中文知识图谱规则挖掘的支持。

4. 由于 AMIE 对于封闭霍恩规则的挖掘已经有了很好的支持，我们的工作就是在 AMIE 的基础上进行拓展。在封闭规则挖掘的基础上，进一步进行非封闭的规则挖掘。

二、 工作成果及水平

完成工作内容：

1. 了解 AMIE 的整体架构与算法实现。

2. 获取一些知识库，使用 AMIE 进行知识库中规则的挖掘。

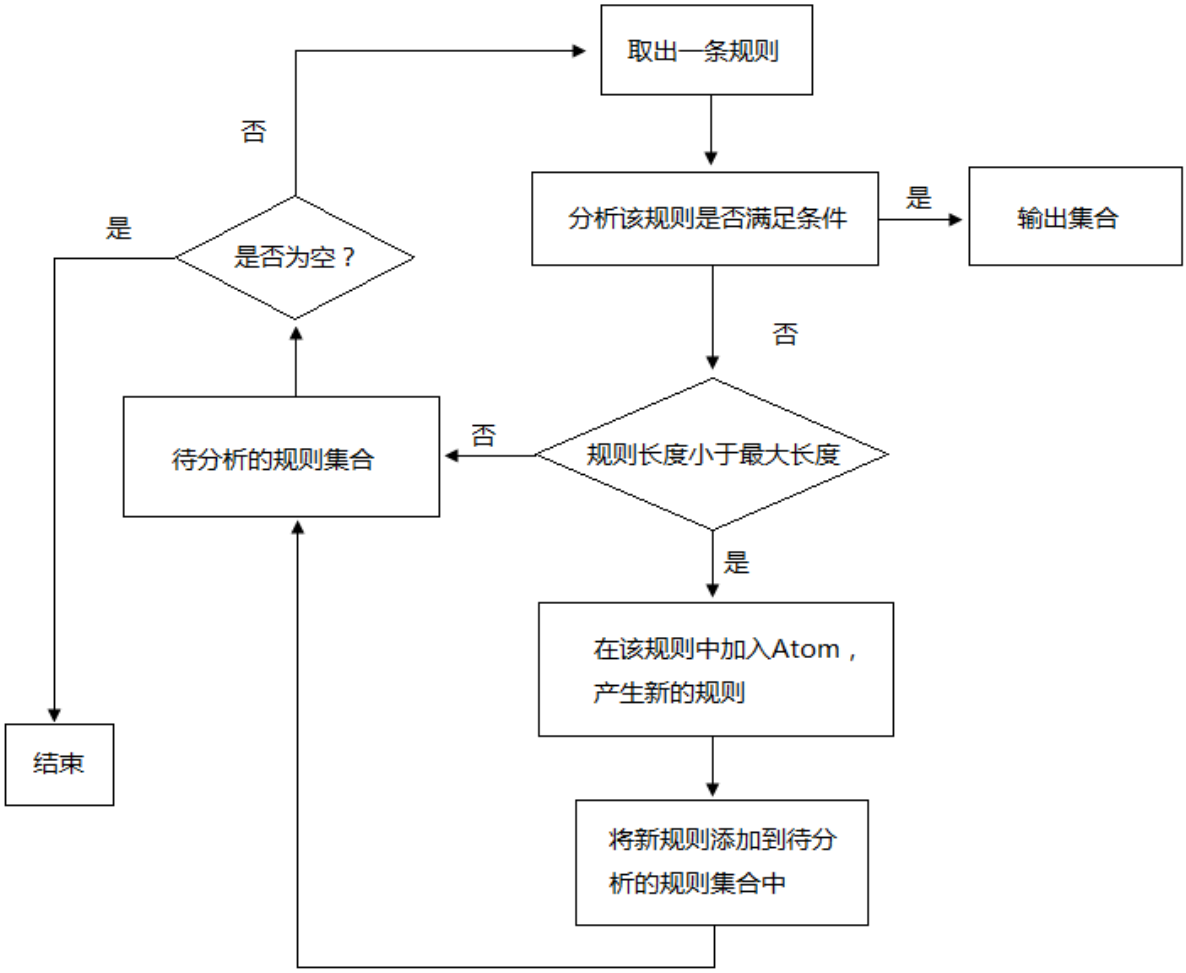
3. 在 AMIE 的基础上进行系统开发，实现 closed Horn 规则挖掘。

4. 编写代码，进一步完善规则挖掘算法的细节，并使用提供的知识库数据进行挖掘测试。

5. 在实现基本的规则挖掘功能的基础上，提供系统对中文知识图谱规则挖掘的支持。

成果（内容及形式）：

1. AMIE 的整体架构



2. 在 AMIE 的基础上进行系统开发并进行知识库的分析

成果形式：Java 工程 RuleMiner

工程结构：
Package:
  com.zzq.ruleminer;
Java source files:
  RuleMiner.java
  KB.java
  MiningAssistant.java
  Rule.java
  RuleGraph.java
  Int.java
类：

RuleMiner，主类，实现 AMIE 整体架构图中的逻辑

KB，知识库类，里面是一些 HashMap，存储了所有的知识三元组。

MiningAssistant，实现规则的推衍，即在一条规则中添加新的 Atom，从而生成许多新的规则。

Rule，存储一条规则，里面是一个三元组列表。

RuleGraph，为了分析两条规则的等价性而使用的辅助类，用来判断两条规则是否等价。

Int，由于 Map 中的元素必须是非基本数据类型，则使用 Int 来表示长整型，内部是一个 long 的数据。

使用 RuleMiner 进行数据挖掘测试：

测试数据来自：
https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/amie/

测试环境：

系统：windows 7 sp1 64 位

CPU：Intel Core i3-4005U CPU 1.70GHz 4-CORE

内存：8G

程序参数设置：
minPcaConfdence=0.1
minStdConfdence=0.1
minHeadCoverage=0.01

测试结果：

Dataset	Fact num	Max depth	Rules mined	Time(s)
Yago2	46654	2	7	0.902
		3	28	6.786
		4	311	52.004
Yago2	948358	2	8	35.073
		3	31	3334.907
		4	\	\

部分结果截图：

```
Mining......Start
Using StdConfidence
Minimum StdConfidence Threshold: 0.1
Minimum PcaConfidence Threshold: 0.1
Minimum HeadCoverage Threshold: 0.01
Max Depth: 3
Rule    StdConfidence   PcaConfidence   Support
?a <imports> ?b =>      ?a <exports> ?b 0.16033755274261605     0.1688888888888889      38
?a <isPoliticianOf> ?b  =>      ?a <livesIn> ?b 0.15789473684210525     0.5     6
?a <produced> ?b        =>      ?a <directed> ?b        0.10714285714285714     0.30414746543778803     66
?b <dealsWith> ?a       =>      ?a <dealsWith> ?b       0.15    0.2102425876010781?      78
?a <exports> ?b =>      ?a <imports> ?b 0.10242587601078167     0.12624584717607973     38
?a <directed> ?b        =>      ?a <created> ?b 0.23692636072572038     0.3769100169779287      222
?b <isMarriedTo> ?a     =>      ?a <isMarriedTo> ?b     0.4475104979004199      0.9130966952264382      746
?a <created> ?b ?a <actedIn> ?b =>      ?a <produced> ?b        0.2777777777777778      0.4878048780487805      20
?a <directed> ?b        ?a <actedIn> ?b =>      ?a <produced> ?b        0.2     0.34285714285714286     12
?a <directed> ?b        ?a <created> ?b =>      ?a <produced> ?b        0.16666666666666666     0.3490566037735849      37
?c <diedIn> ?b  ?c <isCitizenOf> ?a     =>      ?a <hasCapital> ?b      0.19230769230769232     0.20408163265306123     10
?c <isLocatedIn> ?b     ?c <isCitizenOf> ?a     =>      ?a <hasCapital> ?b      0.14285714285714285     0.14285714285714285     4
?c <wasBornIn> ?b       ?c <isCitizenOf> ?a     =>      ?a <hasCapital> ?b      0.20588235294117646     0.23076923076923078     21
?d <diedIn> ?b  ?a <isMarriedTo> ?d     =>      ?a <diedIn> ?b  0.13333333333333333     0.4827586206896552      14
?d <livesIn> ?b ?a <hasChild> ?d        =>      ?a <livesIn> ?b 0.13043478260869565     0.8571428571428571      6
?b <hasCapital> ?d      ?a <livesIn> ?d =>      ?a <livesIn> ?b 0.39215686274509803     0.39215686274509803     20
?d <hasCapital> ?b      ?a <livesIn> ?d =>      ?a <livesIn> ?b 0.18811881188118812     0.18811881188118812     19
?d <livesIn> ?b ?a <isMarriedTo> ?d     =>      ?a <livesIn> ?b 0.25806451612903225     0.6153846153846154      8
?d <graduatedFrom> ?b   ?a <hasAcademicAdvisor> ?d      =>      ?a <graduatedFrom> ?b   0.15384615384615385     0.3076923076923077      4
?a <created> ?b ?a <actedIn> ?b =>      ?a <directed> ?b        0.2777777777777778      0.45454545454545453     20
?a <produced> ?b        ?a <actedIn> ?b =>      ?a <directed> ?b        0.2     0.36363636363636365     12
?a <produced> ?b        ?a <created> ?b =>      ?a <directed> ?b        0.44047619047619047     0.6981132075471698      37
?d <dealsWith> ?b       ?a <dealsWith> ?d       =>      ?a <dealsWith> ?b       0.30728709394205445     0.30728709394205445     350
?c <dealsWith> ?b       ?c <dealsWith> ?a       =>      ?a <dealsWith> ?b       0.16422287390029325     0.23829787234042554     224
?d <hasChild> ?b        ?a <isMarriedTo> ?d     =>      ?a <hasChild> ?b        0.39069767441860465     0.5581395348837209      672
?c <hasChild> ?b        ?c <isMarriedTo> ?a     =>      ?a <hasChild> ?b        0.31700423928403204     0.6185661764705882      673
?b <hasChild> ?d        ?a <hasChild> ?d        =>      ?a <isMarriedTo> ?b     0.1469408224674022      0.2900990099009901      293
?d <isCitizenOf> ?b     ?a <influences> ?d      =>      ?a <isCitizenOf> ?b     0.1794871794871795      0.6363636363636364      7
Mining......OK
Mined 28 rules
Total Time: 6.754 s
```

3. 中文规则的支持

对之前的 Yago2（fact num 46654）的知识库进行一些修改（部分翻译成中文）。测试结果如下：

```
Minimum HeadCoverage Threshold: 0.01
Max Depth: 3
Rule    StdConfidence   PcaConfidence   Support
?a <生产> ?b      =>      ?a <导演> ?b      0.10714285714285714     0.30414746543778803     66
?a <导入> ?b      =>      ?a <导出> ?b      0.16033755274261605     0.1688888888888889      38
?a <导出> ?b      =>      ?a <导入> ?b      0.10242587601078167     0.12624584717607973     38
?b <涉及> ?a      =>      ?a <涉及> ?b      0.15    0.2102425876010781?      78
?b <婚姻> ?a      =>      ?a <婚姻> ?b      0.4475104979004199      0.9130966952264382      746
?a <导演> ?b      =>      ?a <创建> ?b      0.23692636072572038     0.3769100169779287      222
?a <是当地的政治家> ?b      =>      ?a <居住在> ?b      0.15789473684210525     0.5     6
?a <演出> ?b      ?a <创建> ?b      =>      ?a <导演> ?b      0.2777777777777778      0.45454545454545453     20
?a <生产> ?b      ?a <创建> ?b      =>      ?a <导演> ?b      0.44047619047619047     0.6981132075471698      37
?a <生产> ?b      ?a <演出> ?b      =>      ?a <导演> ?b      0.2     0.36363636363636365     12
?d <有孩子> ?b    ?a <婚姻> ?d      =>      ?a <有孩子> ?b    0.39069767441860465     0.5581395348837209      672
?c <有孩子> ?b    ?c <婚姻> ?a      =>      ?a <有孩子> ?b    0.31700423928403204     0.6185661764705882      673
?d <死于> ?b      ?a <婚姻> ?d      =>      ?a <死于> ?b      0.13333333333333333     0.4827586206896552      14
?d <是公民> ?b    ?a <影响> ?d      =>      ?a <是公民> ?b    0.1794871794871795      0.6363636363636364      7
?c <出生> ?b      ?c <是公民> ?a      =>      ?a <首都是> ?b    0.20588235294117646     0.23076923076923078     21
?c <地点位于> ?b  ?c <是公民> ?a      =>      ?a <首都是> ?b    0.14285714285714285     0.14285714285714285     4
?c <死于> ?b      ?c <是公民> ?a      =>      ?a <首都是> ?b    0.19230769230769232     0.20408163265306123     10
?a <创建> ?b      ?a <导演> ?b      =>      ?a <生产> ?b      0.16666666666666666     0.3490566037735849      37
?a <演出> ?b      ?a <导演> ?b      =>      ?a <生产> ?b      0.2     0.34285714285714286     12
?a <演出> ?b      ?a <创建> ?b      =>      ?a <生产> ?b      0.2777777777777778      0.4878048780487805      20
?d <毕业于> ?b    ?a <有学术顾问> ?d      =>      ?a <毕业于> ?b    0.15384615384615385     0.3076923076923077      4
?d <涉及> ?b      ?a <涉及> ?d      =>      ?a <涉及> ?b      0.30728709394205445     0.30728709394205445     350
?c <涉及> ?b      ?c <涉及> ?a      =>      ?a <涉及> ?b      0.16422287390029325     0.23829787234042554     224
?b <有孩子> ?d    ?a <有孩子> ?d    =>      ?a <婚姻> ?b      0.1469408224674022      0.2900990099009901      293
?d <居住在> ?b    ?a <婚姻> ?d      =>      ?a <居住在> ?b    0.25806451612903225     0.6153846153846154      8
?d <居住在> ?b    ?a <有孩子> ?d    =>      ?a <居住在> ?b    0.13043478260869565     0.8571428571428571      6
?b <首都是> ?d    ?a <居住在> ?d    =>      ?a <居住在> ?b    0.39215686274509803     0.39215686274509803     20
?d <首都是> ?b    ?a <居住在> ?d    =>      ?a <居住在> ?b    0.18811881188118812     0.18811881188118812     19
Mining......OK
Mined 28 rules
Total Time: 6.196 s
```

存在问题：

1. 规则之间的比较（图的同构）：

要评价两条规则的等价性是比较麻烦的，特别是在规则长度较长的时候。规则的等价性可以看下面的例子。

?a <isMarriedTo> ?d ?d <hasChild> ?b => ?a <hasChild> ?b

?a <isMarriedTo> ?e ?e <hasChild> ?b => ?a <hasChild> ?b

?f <hasChild> ?b ?a <isMarriedTo> ?f => ?a <hasChild> ?b

这三条规则是等价的，但是要快速判断出来还是比较麻烦的，然而规则之间的等价性判断会在添加规则到集合的过程中大量使用到。因此，如何精确甚至准确地判断是一个关键的问题。

一条规则可以表示成有向图图，可以通过图的同构判断两个规则是否相同。而有向图的同构判断算法又比较复杂，有一种出入度序列法，可以较快地判断（接近多项式时间复杂度），但是在最坏的情况下仍有指数的时间复杂度。

2. 上面 RuleMiner 的规则挖掘都是使用 Dangling Atom 和 ClosingAtom，并没有用到 Instantiated Atom，对它的支持还不够。

3. 当规则长度设置较长，且 fact 数量较多时，挖掘时间会迅速增长，例如测试中的第二条结果，长度增加 1，时间变为原来的近 100 倍。这点需要进一步优化。

预期进展：

1. 优化算法，使用多种方法提高算法效率，进行规则挖掘，并进行结果分析与对比。

2. 学习非封闭规则挖掘的相关知识，获取理论基础。

3. 在实现 closedHorn 规则挖掘的基础上，进一步进行非封闭的规则挖掘，并进行数据测试与比较。

4. 完善一些细节，修复各类 bug，完成毕业设计。

三、 项目收获

1. 项目中所用的技术：

如何在内存中对知识库进行存储是一件很重要的事，不同的数据结构可能会导致知识读取的速度有较大的差异。这里，由于知识是以三元组的形式存储的，我们在 KB 中建立 3 层 HashMap，即 Map(key : Map(key : Map(key : value)))，对应进行 subject-relation-object 的查询，在建立另外 5 个同样的 HashMap，进行 subject-object-

relation、relation-subject-object、relation-object-subject、object-subject-relation、object-relation-subject 的查询，虽然占用空间大了，但是查询时间大大减少了。

2. 解决的关键问题：

之前在规则的等价性判断中遇到一些难题。要准确判断两个规则是否相同比较困难，但是较精确地判断还是比较容易实现的。在比较过程中，先对规则进行简单的判断（例如，判断两个规则的长度、head atom 是否相同等），若相同，接着将规则妆化成有向图，使用图中每个结点的出度、入度以及各条边的值（这里是指 Relation），计算出该规则的一个哈希值，然后对规则的哈希值进行判断。虽然这样不能达到非常准确的要求，但是计算速度快，效率高。

3. 学习和工作方法收获：

在项目的过程中，遇到了各种各样的问题。有些问题比较简单，几分钟就能解决，有些则比较麻烦，几天也不一定搞得定。另外，关于规则挖掘的文献资料大多数是外文的，这也加大了我的学习难度。正是这些查询资料、调试程序的过程，提高了我分析问题、解决问题的能力，增加了我的学习和工作的经验，这是十分难得的。

四、 对工作建议

从设计项目上面来看，这次毕业设计的选题（关联规则挖掘）与实际生产生活相关，应用性较强，而且于个人的专业方向也较为匹配。

虽然这样，它仍旧有太大的局限性。从总体上面来看，首先是毕业设计的题目类型太少，太集中，大多数是数据挖掘、数据处理相关，其他方面的题目较少，不够开放。

单位、实验室方面，由于本次毕业设计是纯数据分析，因此对实验室并没有要求。

五、 其它

本次毕业设计到了中期，完成了部分功能，但仍有一些问题遗留，在接下来的时间会努力解决这些问题，并完成剩余的功能，争取做到最好。