

中山大学硕士学位论文

面向知识图谱溯因诊断的研究

Research on the Abduction of Knowledge Graph

学位申请人：_____

指导教师：_____

专业名称：软件工程_____

答辩委员会主席（签名）：_____

答辩委员会委员（签名）：_____

二零一五年五月

论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

学位论文使用授权声明

本人完全了解中山大学有关保留、使用学位论文的规定，即：学校有权保留学位论文并向国家主管部门或其指定机构送交论文的电子版和纸质版，有权将学位论文用于非赢利目的的少量复制并允许论文进入学校图书馆、院系资料室被查阅，有权将学位论文的内容编入有关数据库进行检索，可以采用复印、缩印或其他方法保存学位论文；可以为建立了馆际合作关系的兄弟高校用户提供文献传递服务和交换服务。

保密论文保密期满后，适用本声明。

学位论文作者签名：

日期： 年 月 日

导师签名：

日期： 年 月 日

论文题目： 面向知识图谱溯因诊断的研究

专 业： 软件工程

硕 士 生：

指导教师：

摘 要

词嵌入是近年来比较热门的一个研究方向，词嵌入的目的是通过低纬度的稠密向量表征出知识库中的实体和关系的语义信息，在有效存储信息的基础上，能够实现实体预测、链接预测等推理任务。目前，大多数的工作集中在找到一个更有效的知识库模型，使得训练出的模型可以在现有的评分标准下取得更好的成绩。然而在现实中经常会出现由于知识库的不完善，导致某些符合客观事实的链接无法被正确的预测。本文的主要工作，是利用本体中溯因推理的技术找出这些链接无法被正确预测的原因，并计算出合乎语义的解释对知识库进行修复。

溯因推理是描述逻辑中一项重要的推理机制，它的目的是找到观察值无法被本体蕴含的原因，并提供合乎逻辑的解释。在本体的溯因诊断中，限制解释集的大小是必须考虑的一个关键因素。为了平衡解释集的表达能力和解空间的矛盾，本文根据决断集的特点提出了概念决断集模版。决断集合模版可以决断集中获取，利用决断集模版我们限制了解释的数量。同时，我们还提出了 $\subseteq_{ds}-minimal$ 概念进一步地在保证语义不缺失的情况下减小解释集数量。

我们的目标是计算所有合理解释，合理的解释需要满足的条件包括可接受，非平凡，一致并且是 $\subseteq_{ds}-minimal$ 的。在本文中，我们提供了完整的算法来计算所有满足条件的解释。我们还利用生成的解释来实现对知识库的修复。我们基于本文提出的算法实现了PBA系统，在PBA系统我们测试了解释集对知识库的修复能力。实验结果证明了我们提出的方法具有有效性以及合理性。

关键词：本体；决断集；溯因诊断；词嵌入

Title: Research on the Abduction of Knowledge Graph
Major: Software Engineering
Name:
Supervisor:

Abstract

Word embedding is a popular research direction in recent years. The purpose of word embedding is to express the semantic information of entities and relationships in the knowledge base through the dense vector of low latitude. Based on the effective storage of information, it can realize entity prediction, link Prediction and other reasoning tasks. At present, most of the work focuses on finding a more efficient model , so that the trained model can achieve better results under the score of benchmark. However, in reality there are often due to the imperfect data set, resulting in some of the objective facts of the link can not be correctly predicted. The main work of this paper is to use the technique of ontology abduction to find out why these links can not be correctly predicted and to calculate the semantic explanation to repair the data set.

Abductive reasoning is an important reasoning mechanism in describing logic, and its purpose is to find the reasons why observations can not be implied by the ontology and provide logical explanations. In the ontology diagnosis of the ontology, limiting the size of the explanatory set is a key factor that must be considered. In order to balance the contradiction between the expression ability of the interpretation set and the solution space, this paper puts forward the concept justification pattern according to the characteristics of the justification. Determine the set of templates that can be determined collectively, using the decision set template we limit the number of explanations. At the same time, we also suggest that the \subseteq_{ds} -*minimal* concept further reduces the number of explanatory sets without guaranteeing that semantics are not missing.

Our goal is to calculate all reasonable explanations, reasonable explanations that need to be met, including acceptable, non-trivial, consistent and is \subseteq_{ds} -*minimal*. In this paper, we provide an algorithm to retrieve all the solutions that

satisfy the condition. Finally, we use the generated triad to repair the knowledge base. We have implemented the PBA system based on the algorithm proposed in this paper. In the PBA system, we have tested the ability to solve the knowledge base. The experimental results show that the proposed method is effective and reasonable.

Key Words: Ontology; Word Embedding; Justification; Abduction Diagnosis

目 录

摘 要	I
Abstract	II
第一章 引言	1
1.1 本文的意义	1
1.2 研究现状	2
1.3 本文的工作	5
1.4 论文结构简介	5
第二章 预备知识.....	7
2.1 知识库模型研究	7
2.2 本体的研究	13
2.3 推理机与决断集的计算	17
2.4 本体溯因诊断的研究	20
第三章 基于决断集模板进行溯因诊断.....	25
3.1 知识库诊断问题	25
3.2 本体的诊断与解释	25
3.3 决断集模版的生成与验证	27
3.4 基于决断集模版的解释	30
3.5 生成基于决断集模版解释的算法	34
3.6 本章小结	38
第四章 解释集生成与嵌入更新	39
4.1 决断集计算	39
4.2 多观察值的最小解释集生成	41
4.3 基于解释集的向量模型更新研究	42
第五章 实验与分析.....	45
5.1 实验工具与环境	45

5.2	程序框架	46
5.3	解释基数对生成解释的效率以及数量对比实验	47
5.4	决断集模板解释修复知识库模型对比实验	49
第六章	总结与展望.....	53
6.1	工作总结	53
6.2	后续工作以及展望	54
参考文献		55
在学期间论文发表情况		61

第 1 章 引言

近年来，人工智能获得了极大的发展，在如今繁多的互联网应用中，人工智能开始支撑着越来越多的应用服务，例如智能搜索、语音识别、翻译以及智能问答等。而作为人工智能的基础，知识的表示与学习的研究获得了越来越多人的重视。知识表示与学习是对知识库加以利用的过程，利用自动或是半自动的方法，将各种无结构的信息转化成机器可读并且可服务于查询、推理等服务需求的知识库模型是知识表示与学习的研究目标。

1.1 本文的意义

进入信息时代，人们花费了大量的精力构建结构化的知识库。利用已有的知识库，进行表示学习的目标，是通过转化规则将现有的知识库的语义信息转化到低维的向量中进行表示，并利用转化后得到的低维向量学习出新的知识[1]。但是，构建出的向量模型与现实世界观察值的不一致是一种出现频率较高的问题，这种不一致会降低向量模型推理结果的可信度，同时也意味着已构建的向量模型对语义信息表达的准确度有待提高[2, 3]。因此，如何找到产生这种不一致的原因，以及如何对已构建的向量模型进行修复，是目前研究的重点。

然而，目前的研究工作，大多数集中在改进向量模型生成的算法上，但是由知识库产生的问题，却还没有做较为深入的研究。知识库是向量模型的生成来源，一个不完整的或者是有缺陷的知识库，会对生成的模型产生重大的影响。与此同时，对本体的研究获得了较大的进展。本体作为语义网的核心，是一种清晰表达语义和知识共享的方式，也是在特定领域之中那些存在着的对象类型或概念及其属性和相互关系[2]，在本体中进行推理可以获得知识库中没有进行表达的知识。而其中，溯因诊断又是本体中一种重要的推理方式。溯因诊断是推理到最佳解释的过程，它是开始于事实的集合并推导出它们的最合适的解释的推理过程，溯因意味着生成假设来解释观察的结论[4-6]。

本文的主要目标，是利用本体语言对知识库与观察值进行表达，再在生成的

本体中进行溯因诊断，利用针对词向量知识库改进的诊断方法，找出知识库模型与观察值产生不一致的原因，并在此基础上，实现对知识库的修复，从而达到修正向量模型的目的。

1.2 研究现状

目前的研究主要集中于知识库模型的构建，利用深度学习构建基于词向量表达的知识库模型和使用基于语义表达的本体进行知识库模型构建是目前主要的两种构建知识库模型的方法。

1.2.1 构建知识库模型的研究

Sutskever, Salakhutdinov, and Tenenbaum(2009)是较早的提出使用贝叶斯聚类网络对关系进行因式分解表达[7]，他们的方法对于每一个实体都使用了超过一个向量进行表达，不过Usunier等人认为对于每个实体都使用多个向量表达难以做到一般化表达[8]。实际上在Bordes随后进行的实验也发现，这种方法构建出来的模型无法轻松地捕获处于稀疏位置的实体[9]。之后又有多项工作针对性地提高这类知识库模型的性能表现，但是大多数情况下提升有限，或者是应用场景狭窄，无法做到一般化。在自然语言处理领域，嵌入的概念最早由Bengio等人提出，在Bengio等人构建的语言框架模型中所有的单词都会被学习出一个嵌入。随后，Bordes提出了一种新的神经网络架构，这种架构被设计成可以将所有实体和关系嵌入到一个灵活且连续的向量空间。但由于嵌入方式的选择问题，导致构建出的模型性能较为低下。因此在2013年，Bordes根据词向量的特性，将知识库中实体的关系翻译成向量中的平移操作，从而提出了用于将实体嵌入到向量空间的翻译模型TransE[9]。TransE相比之前的方法，在不仅模型简单，性能提升幅度大，更为关键的是它提出了一种新的思路，给向量模型的构建提出了新的方向。之后的知识表示学习的模型，大多数都是从TransE扩展而来。

1.2.2 本体与溯因诊断研究现状

利用本体对知识库建模是语义网领域一直备受关注的研究方向。本体的构建方法多样而且复杂，图1.1[2]展示了经典的本体构建流程，在构建本体的过程中，本体与观察值不一致或本体无法蕴含观察值的现象是很常见的[10, 11]。因此在近年来本体的研究热点开始转移到对本体的修复上。溯因推理由于推理方式与人的思考方式具有较高的一致性，因此溯因推理技术被大量应用在本体的修复任务中。

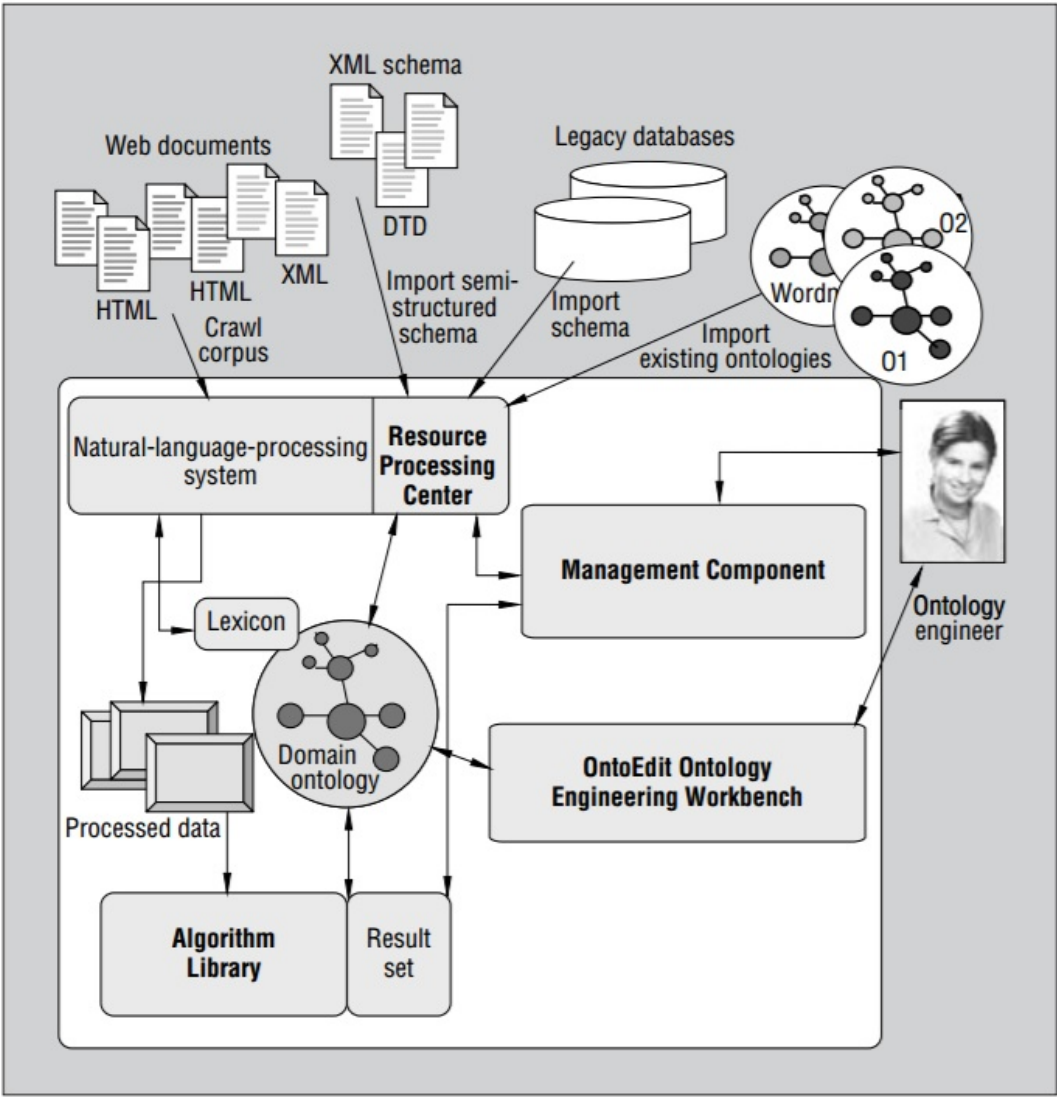


图 1.1: 本体构建

本体的溯因推理研究目前主要集中在三个子问题领域[12–14]，包括概念的溯因推理，ABox的溯因推理问题以及TBox的溯因推理问题。其中由于在生产环境

具有较高的实用性，TBox与ABox的溯因推理问题是目前的研究重点。

在ABox的溯因诊断问题中，S Klarman等人对ABox的溯因推理问题作出了明确定义，他们还对ABox的解进行了限制[15]。溯因推理是根据观察值提出合理假设进行解释的过程。由于需要提出假设，因此溯因推理出的结果会出现解空间无限膨胀的问题。S Klarman提出最小解的概念，对解的公理数进行了限制，从而达到限制解空间的目的。ABox的溯因推理，需要计算出所有满足限制条件的最小解，在把解并入到待诊断的一致本体后，需要满足新构建的本体具有一致性，并使得新构建的本体能够蕴涵出观察值。在Du(2014)的研究中，提出了一种简单可行的方法进行解空间的限制。它提出了一个新的TBox概念*first-order rewritable TBoxes*。这种*first-order rewritable TBoxes*可以保证ABox溯因诊断问题解释的存在以及最小性。同时，为了减少需要计算的解释的数量，文中还提出了一种新的解释概念，称作表达性解释。表达性解释减少了需要计算的解释的数量，而且利用表达性解释还可以计算出所有满足限制的解。之后，为了进一步限制解释的数量，Du和Kewen等人又提出了新的解释概念*cps-minimal IAR*，*cps-minimal IAR*解释大大减少了需要计算的解释的数量，可以进行非常高效地计算并对本体进行诊断。

一般的，溯因推理需要找到需要添加的公理使得公理被加入原知识库后可以蕴涵观察值，而TBox的溯因推理则是需要找到合适的公理集使得这些公理集被加入到原本体后可以蕴涵一条或多条的TBox公理。最早的诊断研究出现在Colucci等人的工作中[14]，他们提出了基于事实前缀的表算法，这种方法可以解决基于 \mathcal{ALN} 描述逻辑的概念诊断问题。这是表算法在诊断问题领域的第一次提出，虽然应用领域不够广泛，但是他提出了一个新的方向。之后Ken Halland等人在2013改进了Colucci等人的算法模型，把应用范围扩展到 \mathcal{ALC} DL[16]。Halland方法虽然扩大了解释的表达能力，但是没有办法为解释提供一个清晰语义解释，使得得到的结果缺少实用性。目前对于TBox诊断的研究较少，且大多数没有提供实现方法以及实验结果又或者是对解释的表达能力进行了限制[17]。

1.3 本文的工作

本文的目标是利用本体中的溯因推理方法达到优化知识库模型的目的。在翻译模型中，基于训练集训练出来的翻译模型具有链接预测的能力。但是并不是所有的正关系都能够被得到的翻译模型正确的预测，这些正确的链接为什么没有被正确预测，以及如何修正模型，使得修正后的模型可以被正确地预测，是本文关心的重点。本文的主要工作包括：

第一，我们针对溯因推理提出解释的定义与限制。

第二，我们基于决断集提出了一个新的概念，决断集模板。我们对决断集模板的相关语义特性进行了证明。第三，我们提出本体中决断集的计算方法。

第四，我们基于决断集模版提出满足 $\subseteq_{ds-minimal}$ 等条件的合理解释生成算法。

第五，我们利用生成的合理解释对知识库进行了诊断。

最后，我们设置了一组实验检验我们提出的决断集模版的计算算法以及解释生成算法的实际效果，另外我们还设置了一组实验用于对比在数据集获得修复之后计算出的翻译模型对观察值的预测能力。

本文的研究成果已经整理成一篇学术论文，被AAAI-17收录。其中，AAAI (American Association for Artificial Intelligence)为中国计算机学会(CCF: China Computer Federation)推荐的A类会议。

1.4 论文结构简介

本文的第一章主要介绍本文产生的背景，回顾目前在知识库模型以及本体领域中的研究成果，并简单介绍了本文的主要工作。

本文第二章内容主要是本文在研究中所涉及到的基本知识，其中包括知识库模型的基本方法和理论，本体研究中的主要理论以及溯因诊断问题的形成和目前相关的研究成果。

本文第三章介绍了决断集，提出了决断集模板的概念，并证明了与之相关的语义特性。利用决断集模板我们提出了一套用于对数据集进行诊断并计算出解释

的理论。

本文第四章介绍了利用决断集模版计算决断集的方法，并探究了利用解释集更新知识库模型的方法。

本文第五章介绍了我们实验的情况，主要介绍了实验环境的设置以及实验的进行方法。最后根据实验的结果得出我们的结论。

本文第六章主要是对本文进行了总结，介绍了本文中的主要研究成果以及不足，并对未来继续进行研究提出了主要方向。

第2章 预备知识

2.1 知识库模型研究

知识库存储的是客观事实的元数据，构建知识库模型是知识表示学习的重要内容。构建知识库模型的主要任务，是寻找构建模型的方法完成对知识库中实体关系进行表达[18]。在近年来，研究者对知识库模型进行了深入的研究，提出了多种知识库模型的构建方法，并对他们的研究成果进行了验证。

2.1.1 知识库与建模

首先我们先给出知识库的定义[18]:

$$G = (E, R, S) \quad (2.1)$$

在这个式子中 G 表示知识库，其中

$$E = \{e_1, e_2, \dots, e_{|E|}\} \quad (2.2)$$

E 表示的由知识库中实体组成的集合， $|R|$ 表示在知识库中二元关系组成的集合。一般的，我们会用 (h, r, t) 表示一个表达实体间二元关系的三元组，其中 $h \in E, r \in R, t \in E$ ，对于知识库中的所有 h 和 t ，我们把它称之为头实体和尾实体， r 则称之为头实体和尾实体的关系。例如(特朗普, 总统, 美国)就是一个二元关系三元组的实例，它表达的意思是特朗普是美国的总统，这里的特朗普和美国就是头实体与尾实体，总统表示的是特朗普和美国之间的关系。在式子2.1中的 S 表示所有二元关系的三元组集合，记作 $S \subseteq E \times R \times E$ 。对知识库 G 的建模可以看做是对 (E, R, S) 建立模型的研究[18]。

目前构建知识库模型主要方法分为两大类，一类是语义网领域的本体构建方法，本体是一种概念模型的建模工具，构建本体虽然比较复杂，目前构造本体的方法大多基于手动或者是手动加半自动的方法[19, 20]，但是这种方法构建出来的本体可以对知识和语义进行表达，而且本体对于在人可读性与机器可读性中间找

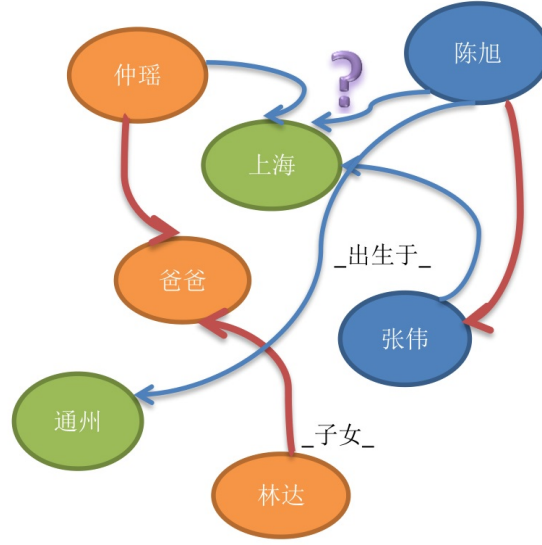


图 2.1: 知识库

到了一个平衡点，在两者的可读性上都要比以往的模型要友好。另一种构建知识库模型的方法是基于机器学习的知识库建模，目前基于机器学习方法的知识库模型有几个具有代表性的模型：距离模型、单层神经网络模型、能量模型、双线性模型、张量神经网络模型、矩阵分解模型和翻译模型[18]。

距离模型是由(Bordes)早期提出的使用结构表示(structured embedding, SE)进行知识表示的方法，这种方法的主要思想是[1, 21]：

- 所有的实体都可以通过使用一个 d -维的向量空间进行表达，把实体转化作多维向量空间中向量的操作记作“嵌入空间”。第 i 个实体会被转化成一个向量 E_i ，其中 $E_i \in \mathcal{R}^d$ 。
- 在被嵌入的向量空间中，对于任意给定的关系类型，都存在一个特定的相似度衡量值，这个衡量值就是关系的向量空间。在判断头实体和尾实体是否满足该关系时，头实体和尾实体会被投射到关系所在的向量空间，在关系的向量空间中，头实体和尾实体的向量距离越近，就表示关系成立的可能性越大。作者对 k^{th} 个给定的关系上的实体对记作 $R_k = (R_k^{lhs}, R_k^{rhs})$ ，其中 $R_k = (R_k^{lhs} \text{ 和 } R_k^{rhs})$ 都是一个 $d \times d$ 的矩阵。因此，

在该关系下 $R_k = (R_k^{lhs} \text{ 和 } R_k^{rhs})$ 的相似度计算如下:

$$S_k(E_i, E_j) = || R_k^{lhs} E_i - R_k^{rhs} E_j || \quad (2.3)$$

距离模型的建模方式利用实体在关系的向量空间上的投影进行相似度计算是合理的, 但是在投影的时候, 距离模型对关系中的头实体和尾实体采用了两个不同的投影矩阵, 这种方法难以保证两个投影矩阵的一致, 因此距离模型在链接预测等任务上没有表现出足够优秀的能力。

单层神经网络模型(single layer model, SLM)是Socher等人提出的新的神经网络模型, 为了解决距离模型头尾实体投影矩阵不同带来的协同带来的关系刻画误差问题, 单层神经网络模型对二元关系三元组使用了如下评分函数[22]:

$$g(e_1, R, e_2) = u_R^T f(M_{r,1} l_h + M_{r,2} l_t) \quad (2.4)$$

在评分函数 $g(e_1, R, e_2)$ 中, $M_{r,1} l_h$ 和 $M_{r,2} l_t$ 是投影矩阵, u_R^T 是关系 R 的表示向量, $f = \tanh$ 。单层神经网络模型一定程度上优化了距离模型刻画关系不精准的问题, 但是却带来了较大的计算量, 降低了计算效率。

DISTMUL是Yang等人提出的一种简化的神经嵌入法, 对于一个给定的由关系三元组 (e_1, r, e_2) (定义 e_1 是在关系 r 下的主体, e_2 是客体) 知识库 KB , DISTMUL的主要想法是对于正向的三元组, 根据评分函数(能量函数)可以获得较高的评分(或者是较低的能量), 而不成立的关系三元组在相同的评分函数则只能得到相反的评分(能量)结果[23]。在DISTMUL模型中, 关系矩阵被限制为对角矩阵, 这种限制虽然增加了矩阵生成的难度, 但是这种对角关系矩阵不仅简化了模型的运算复杂度, 也提高了模型的语义表达效果。利用DISTMUL模型还可以完成知识表示学习的子任务关系规则的挖掘。关系规则是知识库的一个重要部分, 例如, 给定一个事实: 一个人是出生在纽约, 并且纽约是美国的一个城市, 那么这个人的国籍就是美国:

$$BornInCity(a, b) \wedge CityOfCountry(b, c) \implies Nationality(a, c) \quad (2.5)$$

类似的逻辑规则可以帮助从知识库中获取新知识, 而且它也可以优化知识的存储方式, 利用规则的存储减少事实的存储, 从而减小知识库的体

积。更重要的，这些规则可以满足复杂的推理需求。DISTMUL模型挖掘关系规则的主要思想是：如果二元关系三元组 (e_1, r, e_2) 是一个正向例的且二元关系 r 对应的翻译向量 V 满足 $y_a + V - y_b \approx 0$ ，那么应该会有以下规则属性成立：从 $y_a + V_1 \approx y_b$ 和 $y_b + V_2 \approx y_c$ 得出 $y_a + (V_1 + V_2) \approx 0$ 。DISTMUL在挖掘基数为2和基数为3的规则时取得了不错的效果，但是由于算法复杂度较高，在面对更高基数的规则挖掘时耗时增长严重。

2.1.2 翻译模型的研究

2013年由Mikolov领导的一支谷歌研究团队提供了一种对词的向量表示进行运算的方法，这种方法是将深度学习技术引入自然语言处理领域的一项核心技术，Mikolov还提供了一个开源的Word2vec版本，这项技术使得自然语言处理多了一个新的研究方向。受到词向量特性的启发，Bordes等人提出了TransE模型。

TransE是一个为了学习出实体在低维嵌入空间的基于能量的模型。在TransE中，关系会被认为是到嵌入空间的一个翻译[9]。例如假设 (h, l, t) 是一个正向的二元关系三元组，那么在这个模型下尾部实体 t 在空间的嵌入应该约等于头部实体 h 和关系 l 的空间嵌入向量的和：

$$h + l \approx t \quad (2.6)$$

对于一个给定的由三元组 (h, l, t) 构成的集合 S ，其中 $h, t \in E$ (实体的集合)以及关系 $l \in L$ (关系集合)，以下算法的目标是学习关系和实体的嵌入向量。TransE根据能量框架定义三元组的能量等价于相似性测量 $d(h + l, t)$ ，在TransE里 d 可以是曼哈顿距离或者是欧几里得距离。TransE对训练集数据的训练目标是 최소화以下函数：

$$\mathcal{L} = \sum_{(h, l, t) \in S} \sum_{(h', l, t') \in S'_{(h, l, t)}} [\gamma + d(h + l, t) - d(h' + l, t')]_+ \quad (2.7)$$

在以上函数中， $[x]_+$ 表示 $\max(0, x)$ 。由于训练的时候需要生成反例(不成立的二元关系三元组)，这里的反例可以通过正向三元组提取。具体做法是取一正向实例 (h, l, t) ，从三元组中移除头部(或尾部)实体，使用实体集合中选择一个实

体 $h'(h' \in E)$ 组成损坏三元组 (h', l, t) ，其中损坏三元组满足 $(h', l, t) \notin S_{(h,l,t)}$ 。所以对所有损坏三元组，有：

$$S'_{(h,l,t)} = \{(h', l, t) \mid h' \in E\} \cup \{(h, l, t') \mid t' \in E\} \quad (2.8)$$

TransE选择了随机梯度下降法(Stochastic Gradient Descent, SGD)作为优化方法，随机梯度下降法在进行训练的时候，并不需要对所有的和求梯度，因此随机梯度下降法也不需要每次循环的时候更新所有的向量，而只需要对一个批次的向量进行求梯度计算就可以更新 θ 值。

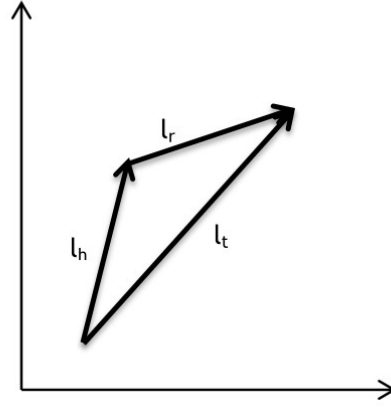


图 2.2: TransE模型

TransE在实验中设置两组实验，分别验证模型在实体预测以及链接预测上的能力，与之对比的还有RESCAL, SE, SME(linear)/SME(bilinear)以及LFM等知识库模型，从实验数据可以看出，TransE取得的效果非常优秀，在三组数据集的两种评价方法下，TransE都取得最好的成绩，值得注意的是在一个体量较大的数据集FB1M的测试中，TransE不仅完成了测试，对比以往的模型还获得了较大的提高。但是，TransE方法也有自身的缺点。假设有两组三元组分别为(上海，位于，中国)记为 (h, l, t) 以及(通州，位于，中国)记为 (h, l, t') 。根据TransE的损失函数，训练出来的模型会有 $h + l \approx t$ 以及 $h + l \approx t'$ ，所以我们有 $t \approx t'$ ，也就是说在TransE的表达模型中，“上海”和“通州”的嵌入向量会近似相等，即使这两个实体具有较大的差异。也就意味着，TransE虽然在处理一对一关系的时候有着

不错得性能，但是在面对一对多的关系的时候，TransE方法具有缺陷。

受TransE的启发，Wang等人在2014年时基于TransE提出了一种新的翻译模型，称之为TransH。TransH的提出就是为了解决在TransE中无法很好处理地一对多，多对多的关系。TransE在建立关系实体的向量空间的时候关系和实体都被嵌入到了平面空间，这个因素限制了TransE在处理多对多、多对一以及一对多的关系的表达能力。因此，为了解决这个问题，在TransH中关系被嵌入到了超平面空间。

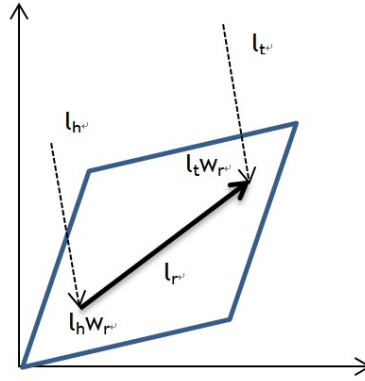


图 2.3: TransH模型

对于一个关系 r ，TransH用超平面 w_r 和在超平面上的向量 d_r 进行表示。特别地，对于一个二元关系三元组 (h, r, t) ， h 和 t 将会被首先投影到平面 w_r 上，投影后的向量分别被记作 h_{\perp} 和 t_{\perp} 。在TransH中，如果 (h, r, t) 是一个正向三元组， h_{\perp} 和 t_{\perp} 会被期望能够被超平面上的向量 d_r 联系起来[24]。因此在TransH中的评分函数为：

$$\| h_{\perp} + d_r - t_{\perp} \|_2^2 \quad (2.9)$$

通过限制 $\| w_r \|_2 = 1$ ，我们可以得到：

$$h_{\perp} = h - w_r^{\top} h w_r, \quad t_{\perp} = t - w_r^{\top} t w_r \quad (2.10)$$

可以得到评分函数:

$$f_r(h, t) = ||(h - w_r^\top h w_r) + d_r - (t - w_r^\top t w_r)||_2^2 \quad (2.11)$$

损失函数为:

$$\mathcal{L} = \sum_{(h, l, t) \in S} \sum_{(h', l, t') \in S'_{(h, l, t)}} [\gamma + f_r(h, t) - f_{r'}(h', t')]_+ \quad (2.12)$$

2.2 本体的研究

语义网的发展依赖于本体的构建, 构建本体的目的是对数据进行结构化处理, 使得本体易于传播的同时有着较高的机器可读性[20]。描述逻辑是当前语义网发展中本体的理论基础, 本体的表达能力由它的描述语言描述逻辑的表达能力决定。描述逻辑根据表达能力的不同, 形成了不同子语言。

2.2.1 描述逻辑与因特网本体语言

描述逻辑(description logic)是一种用于知识表示的逻辑语言和以其为对象的推理方法, 主要用于描述概念分类及其概念之间的关系[25]。描述逻辑是一阶逻辑的可决定性片段, 但是描述逻辑又具有强的表达能力, 描述逻辑在表达能力和推理能力之间取得了平衡。

描述逻辑语言 \mathcal{L} 包含以下三个集合, 分别是一个由个体名组成的集合 N_I , 一个由概念名组成集合 N_C 以及一个由二元关系名组成的集合 N_R 。

定义 2.1 (描述逻辑词汇表) 描述逻辑词汇表 \mathcal{V} 是一个三元组 (N_C, N_R, N_I) , 其中 N_C 是概念名的集合, N_R 是关系名的集合, N_I 是个体名的集合。

我们把 \mathcal{L} 的语义演绎表示为 $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, 其中 $\Delta^{\mathcal{I}}$ 表示 \mathcal{I} 的域, 是一个非空的个体集合, $\cdot^{\mathcal{I}}$ 是一个映射函数, 这个映射函数可以完成概念名到 $\Delta^{\mathcal{I}}$ 子集的映射, 二元关系到 $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ 子集的映射以及个体到 $\Delta^{\mathcal{I}}$ 元素的映射。定义如下[26]:

定义 2.2 一个演绎 \mathcal{I} 是二元组 $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, 其中 $\Delta^{\mathcal{I}}$ 被称为域, $\cdot^{\mathcal{I}}$ 是一个从 N_I 到 $\Delta^{\mathcal{I}}$ 的函数。

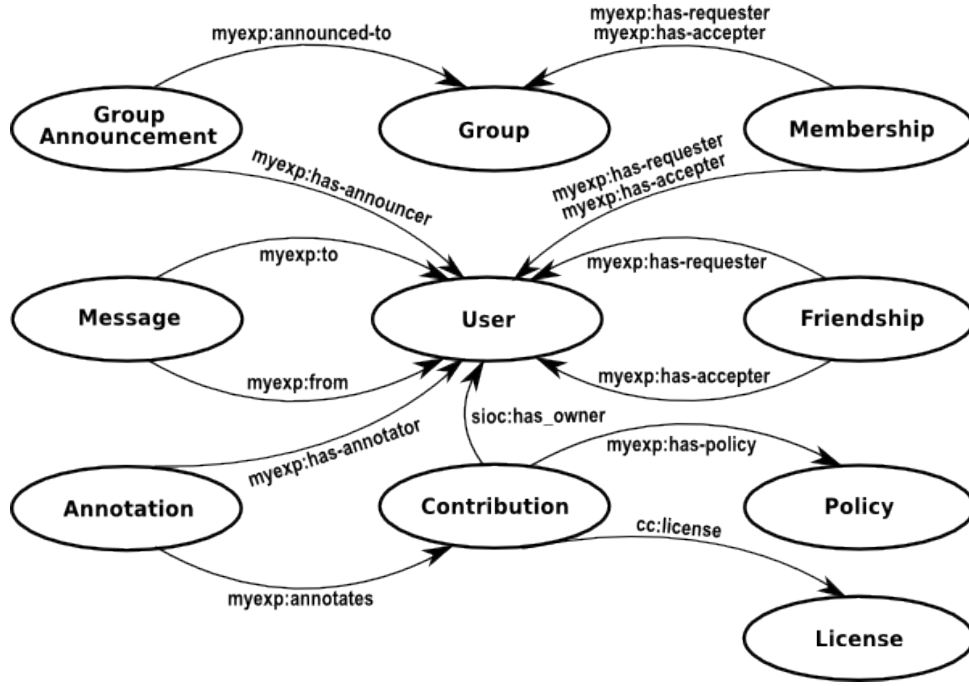


图 2.4: 本体

- $a^I \in \Delta^I$
- $A^I \subseteq \Delta^I$
- $R^I \subseteq \Delta^I \times \Delta^I$
- $(C_1 \sqcap C_2)^I \subseteq C_1^I \cap C_2^I$

在描述逻辑的基本语言片段 \mathcal{ALC} (Attributive Language with Complement)中, 概念表达方式如下例[26, 27]:

例 2.1 令 $N_C = \{Tall, Female, Person, Student, Employee\}$, $N_R = \{hasParent\}$, 可以构建以下概念:

 表 2.1: \mathcal{ALC} 概念表达示例

Concept	Meaning
$Tall \sqcap Person$	Tall person
$Student \sqcup Employee$	student or Employee
$Student \sqcap (\neg Tall)$	non-tall student
$\exists hasChild.Person \sqsubseteq Person$	something having a child who is a person is also a person
$Student \sqcap \forall hasParent.Tall$	Student whose children are all tall person

\mathcal{ALC} 可以表达概念以及个体断言[26]:

1. $C(a)$: a 是 C 的实例。
2. $R(a, b)$: a 和 b 存在关系 R 。
3. $C \sqsubseteq D$: C 是 D 的子概念。
4. $C \equiv D$: C 等价于 D 。

式(1)和式(2)被称作断言公理, 式(3)和式(4)被称作术语公理。在语义网中, 由断言公理组成的集合称为 $assertion\ box(ABox)$, 由术语公理组成的集合称为 $terminological\ box(TBox)$ 。

表 2.2: DL示例公理的语义

Syntax	Semantics
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
$C \equiv D$	$C^{\mathcal{I}} = D^{\mathcal{I}}$
$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
$r(a, b)$	$\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in r^{\mathcal{I}}$

在本文中, 考虑对三元组的表达, 我们使用 \mathcal{ALC} 的扩展DL \mathcal{SHOI} 。 \mathcal{SHOI} 是OWL一个具有较强表达能力的可判定性片段。一个 \mathcal{SHOI} 本体 $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ 包括两个部分, 一个是TBox \mathcal{T} , \mathcal{T} 是一个由有限个概念包含公理 $C \sqsubseteq D$ 组成的集合, 其中 C 和 D 是 \mathcal{SHOI} 的概念, 由以下元素递归性构造:

$$\perp, \top, A, \{a\}, \neg C', C' \sqcap D', C' \sqcup D', \exists r.C', \exists r.\text{Self}, \forall r.C' \quad (2.13)$$

其中 A 是 N_C 中的概念名实例, a 是 N_I 中的个体实例, r 是二元关系, C' 和 D' 是 \mathcal{SHOI} 中的概念。一般的, 为了便于表述, \mathcal{T} 中还包含了RBoxes。在本文中RBox包含以下形式公理的集合, $r_1 \circ \dots \circ r_k \sqsubseteq r$ 以及关系属性断言, 其中所有的属性断言我们都以公理进行表达: 对称性($r \sqsubseteq r^-$), 传递性($r \circ r \sqsubseteq r$)以及自反性($\top \sqsubseteq \exists r.\text{Self}$)。另一个是断言知识的集合ABox \mathcal{A} , 它描述的是TBox词汇表中的个体断言, 包括概念类的成员元素, 二元关系的成员元素二元组以及二元关系的等价关系, 在本文

中我们只需要对知识库中三元组进行表达，所以ABox中我们只考虑形如 $r(a, b)$ 的关系断言，其中 a 和 b 是 N_I 中的个体实例， r 是 \mathcal{SHOI} 中的关系。

在本文中，断言知识部分我们仅考虑二元关系的断言，形如 $r(a, b)$ ，其中 r 是 $\Delta^I \times \Delta^I$ 中的一个二元关系实例， a 和 b 是 Δ^I 中的一个个体。表2.3展示了TBox与ABox的语法以及语义。

语法和语义具有对应关系，如下表

表 2.3: 语法以及语义表

Constructor	Syntax	Semantics
<i>top concept</i>	\top	Δ^I
<i>bottom concept</i>	\perp	\emptyset
<i>conjunction</i>	$C \sqcap D$	$C^I \cap D^I$
<i>existential restriction</i>	$\exists r.C$	$\{X \in \Delta^I \mid \exists y \in \Delta^I: (x, y) \in r^I \wedge y \in C^I\}$
<i>general concept inclusion</i>	$C \sqsubseteq D$	$C^I \subseteq D^I$
<i>role inclusion</i>	$r_1 \circ \dots \circ r_k \sqsubseteq r$	$r_1^I \circ \dots \circ r_k^I \subseteq r^I$

2.2.2 本体推理问题的研究

本体作为一种知识库模型，除了可以存储客观事实，另一个特点是可以存储规则。规则的允许使得本体具有从知识库中获取新知识的能力，在所有的描述逻辑本体中，推理是其中重要的研究任务，本体的推理任务主要有：一致性检测[28]，实例查询，实例计算，子集计算，分类。定义引用自如下[26]：

定义 2.3 一致性 给定一个本体 \mathcal{O} ，判断是否存在一个演绎 \mathcal{I} 满足 \mathcal{O} 。

一致性检测是本体最重要的推理任务之一，不一致的本体会导致查询结果的可信度降低，修复不一致本体是一项重要的任务。

定义 2.4 实例查询 给定一个本体 \mathcal{O} 以及一个概念断言 $C(a)$ ，判断是否满足 $\mathcal{O} \models C(a)$ 。

实例查询是本体在应用中使用最多的场景，在向本体了解个体信息的时候，实例查询可以对给定额查询个体通过搜索知识库与推理相结合的方式给出结果。

定义 2.5 实例计算 给定一个本体 \mathcal{O} 以及概念 C ，计算所有的个体 x 使得 $\mathcal{O} \models C(x)$ 成立。

定义 2.6 子集验证 给定一个本体 \mathcal{O} 以及概念断言 $C \sqsubseteq D$ ，判断本体 \mathcal{O} 是否满足 $\mathcal{O} \models C \sqsubseteq D$ 。

子集又称上下位关系，是本体分类任务中的一个子任务，在区分概念上下位关系以及计算本体概念层次结构中具有重要作用。

定义 2.7 分类 给定一个本体 \mathcal{O} ，计算出所有由原子概念组成的二元组 (A, B) 使得 $\mathcal{O} \models A \sqsubseteq B$ 。

例 2.2 考虑以下本体 $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ ，其中 \mathcal{T} 为：

$$Father \sqsubseteq Male$$

$$Male \sqsubseteq Human$$

\mathcal{A} 为：

$$Father(\text{Tom})$$

$$Male(\text{Tony})$$

显然，在本例中本体 \mathcal{O} 是一致的，又我们可以得出 $\mathcal{O} \models Father \sqsubseteq Human$ ， $\mathcal{O} \models Male(\text{Tom})$ ， $\mathcal{O} \models Human(\text{Tony})$ 。

2.3 推理机与决断集的计算

本体包含TBox与ABox，因此本体具有从已有的知识库中获取新知识的能力，这种本体中没有，但是可以通过推理出来得到的公理就叫做蕴含。一个本体可能会有一个或多个蕴含，在本体的开发中，能够从蕴含逆推出在推理过程相关的公理具有重大的现实意义，这些公理的集合我们称之为决断集[29–31]。

定义 2.8 (决断集) 令 \mathcal{O} 为一个一致的DL本体，且 $\mathcal{O} \models \alpha$ ，其中 α 是蕴含。对于 \mathcal{O} 的一个子集 \mathcal{O}' ，如果对于 \mathcal{O}' 的所有子集 \mathcal{O}'' 满足 $\mathcal{O}'' \models \alpha$ 且 $\mathcal{O}' \models \alpha$ ，那么 \mathcal{O}' 就是本体 \mathcal{O} 中对于 α 的一个决断集。

例 2.3 考虑一个一致的DL本体 \mathcal{O} ，其中TBox包含以下三条公理：

$$\mathcal{T} = \left\{ \begin{array}{l} \textit{Girl} \sqsubseteq \textit{Female} \\ \textit{Female} \sqsubseteq \textit{Person} \\ \exists \textit{giveBirth}.\textit{Person} \sqsubseteq \textit{Female} \end{array} \right\}.$$

ABox包含以下两条公理：

$$\mathcal{A} = \left\{ \begin{array}{l} \textit{Female}(\textit{Mary}) \\ \textit{giveBirth}(\textit{Lily}, \textit{Mary}) \end{array} \right\}.$$

蕴含为：

$$\textit{Female}(\textit{Lily})$$

对于以上例子，我们根据 $\textit{Female}(\textit{Mary})$ 以及 $\textit{Female} \sqsubseteq \textit{Person}$ 可以得出

$$\textit{Person}(\textit{Mary})$$

又根据 $\exists \textit{giveBirth}.\textit{Person} \sqsubseteq \textit{Female}$ ， $\textit{giveBirth}(\textit{Lily}, \textit{Mary})$ 以及个体断言 $\textit{Person}(\textit{Mary})$ ，我们可以得到：

$$\textit{Female}(\textit{Lily})$$

令：

$$\mathcal{S} = \left\{ \begin{array}{l} \textit{giveBirth}(\textit{Lily}, \textit{Mary}) \\ \textit{Female}(\textit{Mary}) \\ \exists \textit{giveBirth}.\textit{Person} \sqsubseteq \textit{Female} \\ \textit{Female} \sqsubseteq \textit{Person} \end{array} \right\}.$$

对于所有的 $\mathcal{S}' \subseteq \mathcal{S}$ 都有 $\mathcal{S}' \not\models \text{Female}(\text{Lily})$, 且 $\mathcal{S} \models \text{Female}(\text{Lily})$, 所以 \mathcal{S} 是 $\text{Female}(\text{Lily})$ 在本体 \mathcal{O} 中的决断集。

寻找决断集模板的方法是本体推理任务中重要的一环, 寻找决断集的方法分为黑盒法和白盒法。

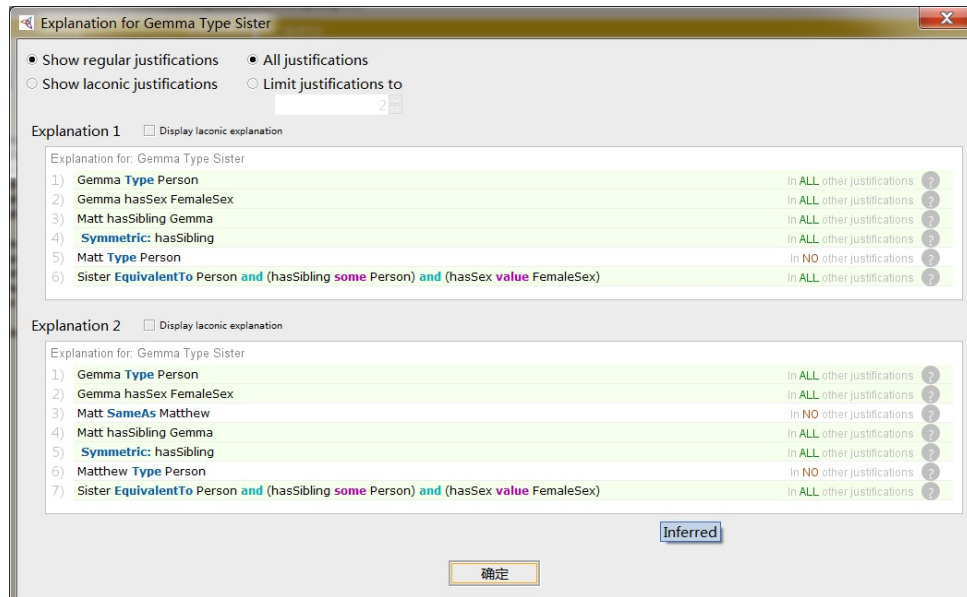


图 2.5: 决断集示例

黑盒法需要依赖于本体的推理机, 并且在方法的实现中不涉及推理机的改动, 推理机在这类算法中扮演黑盒的角色, 黑盒由此得名。推理机在黑盒算法中往往被应用在验证输入本体与蕴含值是否有蕴含关系, 对于这类问题推理机只需要返回“是”或“否”。为了找到蕴含值的决断集, 算法可以在本体中找到合理的子集作为推理机的输入, 然后根据推理机的输出调整下一次输入, 直到找到决断集为止。黑盒法需要有推理机运行蕴含值检测的次数较多, 虽然易于实现, 但是整体算法耗时一般偏大, 以下是Kalyanpur等人提出的黑盒算法示例[29]:

在Kalyanpur的黑盒算法中, 主要分两步: 1) 快速地找到一个决断集的超集 \mathcal{O}' 使得 $\mathcal{O}' \models \alpha$ 2) 移除 \mathcal{O}' 中的一条或多条公理, 使用推理机测试移除后得到的本体 \mathcal{O}' 是否可以蕴含 α , 重复步骤1和2, 就可以得到在 \mathcal{O} 中 α 的一个决断集。可以看出, 可以优化的关键步骤就在于每次添加公理和减少公理的个数, 因为在整个算

```

Input: Ontology  $\mathcal{O}$ , entailment  $\alpha$ 
Output: Ontology  $\mathcal{O}'$ 
 $\mathcal{O}' \leftarrow \emptyset$ 
while  $\mathcal{O}' \not\models \alpha$  do
    | select a set of axioms  $s \subseteq \mathcal{O}/\mathcal{O}'$ ;
    |  $\mathcal{O}' \leftarrow \mathcal{O}' \cup s$ ;
end
for each axiom  $k' \in \mathcal{O}'$  do
    |  $\mathcal{O}' \leftarrow \mathcal{O}' - \{k'\}$ ;
    | if  $\mathcal{O}' \not\models \alpha$  then
    | |  $\mathcal{O}' \leftarrow \mathcal{O}' \cup \{k'\}$ ;
    | end
end

```

Algorithm 1: 寻找单个决断集算法

法中，推理机的蕴含检测是最为耗时的，减少推理机的运行次数可以有效地优化算法的耗时。

白盒法计算决断集并不完全依赖于推理机的计算，其中一种白盒的算法是由Zhou等人提出的基于解释依赖图的决断集算法[30]，这种算法可以在OWL 2 EL中计算本体 \mathcal{O} 蕴含值的决断集。解释依赖图算法需要先使用补全规则对本体 \mathcal{O} 进行蕴含值的计算，然后根据依赖图对本体中的蕴含值进行模板回溯。

2.4 本体溯因诊断的研究

逻辑研究的是基于规则的推理方式，目前的研究中把推理的方式分为三类，分别是演绎、归纳和溯因推理。演绎推理是最常用的推理方式，演绎推理根据已有的前提事实以及规则，得出结论。对于相同的输入，如果严格按照规则进行运算，演绎推理具有相同的输出，具有恒真性(truth-preserving)。归纳推理则是在已知事实的集合中寻找共同特性，推导出更多事实或同类事实的性质[32]。它的推理格式形如以下形式：

- a. 所有已知的A为B。
- b. 因此，A为B。

溯因推理是推理方式中的第三种方式，溯因推理的方式与前两种推理方式有着本质的区别，溯因推理又称作反绎推理，是推理到最佳解释的过程。一般的，它是开始于事实的集合并推导出它们的最合适的解释的推理过程[33]。术语溯因(abduction)意味生成假设来解释观察或结论。因为需要生成假设来解释观察或结论，因此溯因推理会在进行解释的过程中为前提事实增加新的知识使得前提事实与解释的并集可以通过演绎推理的方式演绎出观察值或结论[34]。在描述逻辑本体中，溯因推理是一种重要的推理方式。在描述逻辑本体的构建过程中，本体由于构建不够完善，会经常性地出现本体无法蕴含观察值的现象。因此，这个时候就需要进行利用溯因推理的方法，已构建的本体进行诊断，找出本体不完善的原因，并在找到的原因的基础上，提出解释对本体进行修复。这种找出原因并提出解释的推理方法就叫做溯因诊断。

2.4.1 断言公理的溯因诊断

溯因诊断对修复本体有着重要的意义。对术语断言的溯因诊断问题有如下两个定义[15]:

定义 2.9 (术语断言溯因诊断问题) 令 \mathcal{L}_K 和 \mathcal{L}_Q 为两个DL本体， $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ 是一个 \mathcal{L}_K 中的一个知识库， Φ 是一个在 \mathcal{L}_Q 中的术语断言集合。当且仅当 $\mathcal{K} \not\models \Phi$ 且 $\mathcal{K} \cup \Phi \not\models \perp$ 的时候，二元组 $\langle \mathcal{K}, \Phi \rangle$ 被称作术语断言的溯因诊断问题。

定义 2.10 (术语断言溯因诊断解释) 令 \mathcal{L}_S 为一个DL本体，且 A 为一个多个在 \mathcal{L}_S 中的术语断言集合。对于一个溯因诊断问题 $\langle \mathcal{K}, \Phi \rangle$ ，当且仅当 $\mathcal{K} \cup A \models \Phi$ 我们把 A 称为可接受解释。更多的，我们把 A 称为：

1. (一致) 当且仅当 $\mathcal{K} \cup A \not\models \perp$ 。
2. (非平凡) 当且仅当 $\mathcal{K} \not\models \Phi$ 。
3. (最小) 当且仅当不存在对于 $\langle \mathcal{K}, \Phi \rangle$ 问题的解释 B ，其中 B 是 A 的实例化子集。我们说 B 是 A 的实例化子集当且仅当存在一个重命名映射 $\rho: N_I^*(B) \mapsto N_I^*(A)$ ，其中 $N_I^*(B)$ 和 $N_I^*(A)$ 是来自于 A 和 B 的个体名且不出现于 \mathcal{K} ，使

得 $A \models \rho B$ 。但是对于所有的 $\varrho: N_I^*(A) \mapsto N_I^*(B)$ 满足 $B \not\models \varrho A$ 。满足以上条件的，我们称 A 是问题 $\langle \mathcal{K}, \Phi \rangle$ 的最小解释。

术语断言的溯因诊断需要计算出由一条或多条术语断言的集合，这些集合需要满足最小集的条件。当这些集合被加入到本体中的时候，它需要保持本体的一致性，并且可以使得更新后的本体能够蕴含(\models)含观察值。 $\mathcal{O} \models \alpha$ 表示对于所有满足本体 \mathcal{O} 的模，都可以使得 α 成立。

考虑以下例子：

例 2.4 令 $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$ 是一个 DL 本体，其中 TBox \mathcal{T} 和 ABox \mathcal{A} 包含以下公理：

$$\mathcal{T} = \left\{ \begin{array}{l} \text{Enthusiastic} \sqcap (\exists \text{owns.Cat} \sqcap \text{WorkHard}) \sqsubseteq \text{NicePerson} \\ \text{Person} \sqcap \exists \text{help.Person} \sqsubseteq \text{Enthusiastic} \\ \text{WorkHard} \sqsubseteq \text{Person} \end{array} \right\}.$$

$$\mathcal{A} = \left\{ \begin{array}{l} \text{Cat}(\text{Lily}) \\ \text{Person}(\text{Peter}) \\ \text{WorkHard}(\text{James}) \end{array} \right\}.$$

其中我们有观察值 α ：

$$\alpha = \left\{ \text{NicePerson}(\text{James}) \right\}.$$

根据给出的背景知识可以知道，一个热心的或者是一个工作努力又养猫的人是一个好人，并且乐于助人的人是热心的。又由 ABox \mathcal{A} 可以知道 Lily 是一只猫，Peter 是一个人以及 James 工作努力。由 TBox \mathcal{T} 和 ABox \mathcal{A} ，我们知道 $\mathcal{O} \not\models \alpha$ 。从语义上说，溯因诊断的任务是提出一个合理的假设去解释“James 是好人”。从技术上说，就是要找到合理的断言公理集合 \mathcal{E} 使得 $\mathcal{O} \cup \mathcal{E} \models \text{NicePerson}(\text{James})$ 。例如，在本例中，我们根据观察值和背景知识可以提出三个合理的解释：1) 根据背景知识“热心人是好人”，我们

可以假设“James是热心的”，即 $\mathcal{E} = \{Enthusiastic(James)\}$ ，2)根据一个努力工作又养猫的人是一个好人，而James努力工作且Lily是一只猫，我们可以假设“James拥有Lily”，即 $\mathcal{E} = \{owns(James, Lily)\}$ ，3)根据背景知识乐于助人的人是热心人，热心人是好人，我们可以假设“James帮助了Peter”，即 $\mathcal{E} = \{help(James, Peter)\}$ 。这三种假设都可以使观察值得出合理解释，从而使 $\mathcal{O} \cup \mathcal{E} \models \alpha$ 。

第3章 基于决断集模板进行溯因诊断

本体中的知识库以及规则是本体可以进行推理的前提。目前的本体构造主要由手工或者是半手工构建。由于本体的结构复杂，信息量大，因此本体的构建是一个长期的过程。在本体的构建过程中，构建的本体与观察值的不一致是一个常会发生的问题。因此，找到问题的原因，对观察值提出合理解释成为本体推理中的一个重要任务，这类任务也被称作是溯因推理问题。

3.1 知识库诊断问题

对于一个知识库 $G = (E, R, S)$ ，我们定义在 S 经过深度学习构建出的知识库模型为 Δ_G ，令 Δ_G 无法表达的正向三元组记作 g 。我们的目标是使用本体语言对知识库进行表达，并在本体上对观察值做溯因诊断，因此，我们定义一个由知识库 S 构成的本体为 $\mathcal{O}_G = \mathcal{T}_G \cup \mathcal{A}_G$ 。其中， \mathcal{T}_G 是知识库 G 中的规则集合， $\mathcal{A}_G = \{r_l(h, t) \mid h \in E, t \in E, l \in R\}$ 。另外，我们使用 α_g 表示 \mathcal{O}_G 无法表达的观察值。

在本文中我们考虑 \mathcal{O}_G 的诊断问题，我们对该问题做出以下定义：

定义 3.1 (诊断问题) 我们把 $\Delta_{\mathcal{P}} = (\mathcal{T}_G, \mathcal{A}_G, \alpha_g)$ 称作一个诊断的问题实例，其中本体 $\mathcal{O}_G = \mathcal{T}_G \cup \mathcal{A}_G$ 是一个一致的描述逻辑本体。对于问题 $\Delta_{\mathcal{P}}$ 的一个解释 \mathcal{E} 满足： $\mathcal{T}_G \cup \mathcal{A}_G \cup \mathcal{E} \models \alpha_g$ 且 $\mathcal{T}_G \cup \mathcal{A}_G \cup \mathcal{E} \not\models \perp$ 。

3.2 本体的诊断与解释

对问题 $\Delta_{\mathcal{P}}$ 溯因推理的一个重要目的就是需要找出合理的解释对观察值的进行解释。一般的，这个解释不应该导致诊断本体的不一致，同时，结合本体中的背景知识，新构建出的本体能够蕴含观察值，且新构建的本体能够保持一致的特性。因此，我们对解释有以下定义：

定义 3.2 (解释) 给定一个一致的本体 \mathcal{O} 以及观察值 α ， $\mathcal{O} \not\models \alpha$ ，并且 $\mathcal{O} \cup \{\alpha\}$ 是

一致的, 那么假如存在一个公理的集合 \mathcal{E} 使得 $\mathcal{O} \cup \mathcal{E} \models \alpha$, $\mathcal{E} \not\models \alpha$ 并且 $\mathcal{O} \cup \mathcal{E}$ 是一致的, 我们称这个集合 \mathcal{E} 为在本体 \mathcal{O} 中对 α 的解释。

为了能够满足本体对蕴涵值的推理需求, 解释在本体的溯因诊断中通常会以一种表达能力较高的DL语言比如 $SR\mathcal{OIQ}$ 。表达能力高的语言虽然可以满足本体对观察值的推理需求, 但是较高的表达能力会带来另外一个问题, 解释的空间会无限增大。为了能够尽可能提高解释的表达能力的同时能够限制解释的空间大小, 本文利用模板来实现对解释的空间进行限制。在本体的推理中, 决断集是一个重要的概念, 对于一个一致的本体 \mathcal{O} 以及观察值 α , 我们定义 $Jst(\alpha, \mathcal{O})$ 为本体 \mathcal{O} 对观察值 α 的决断集的集合。因为决断集对蕴涵值具有推理的合理性, 同时决断集满足本体对观察值的解释需要是最小集合的约束, 因此直观地本体对观察值的解释也会遵循相应的模板。

在定义决断集模板之前, 我们首先需要可扩展公理, 对于本体的一条公理, 如果公理中的一个或多个的二元关系或者个体被二元关系变量或者个体变量替换, 则这条公理会被称作可扩展公理。更多地, 一条可扩展公理会被称为可全扩展公理如果这条公理的所有二元关系的个体都被二元关系中的个体变量替换。同时, 可全扩展公理会保留本体中的 \top , \perp 以及个体的不变。在本体中, 公理的类型种类多, 由多条公理组成的决断集会产生数量难以接受的模板, 因此我们需要限制模板的数量, 也就意味着我们需要使用尽可能少的模板来表达尽可能多的决断集, 同时, 对于每个生成的模板, 应该要有一种映射的方式使得决断集与之相对应。这种映射的方式我们称之为可扩展公理的替换, 可扩展公理的替换能够把扩展公理的变量个体映射到个体名或者是个体变量。一个替换会被称作实例化替换如果这个替换能够把所有的个体变脸映射到个体名。

一般地, 我们都需要这样的限制, 对于每个由决断集产生的模板, 都存在一个实例化的替换, 使得这个由决断集产生的模板被映射到一个决断集中。但是这样的限制依旧会产生公理数不受限制的模板, 考虑以下例子:

例 3.1 (解释) 令决断集 \mathcal{J} 是一直本体 \mathcal{O} 中对于原子公理 $\alpha_{(e_1, e_2)}$ 的决断集, J_p 是一个由决断集生成的决断集模板:

$$\mathcal{J} = \{r_M \circ r_N \sqsubseteq r_K, r_M(e_1, e_\mu), r_N(e_\mu, e_2)\}$$

$$\mathcal{J}_p = \{r_M \circ r_N \sqsubseteq r_K, r_M(e_{x_1}, e_{y_1}), \dots, r_M(e_{x_n}, e_{y_n}), r_N(e_m, e_n)\}$$

从 \mathcal{J} 和 \mathcal{J}_p 可以看出, \mathcal{J}_p 是一个公理基数没有上限的决断集模板, 因为存在这样一个映射 θ :

$$\theta = \{e_{x_i} \mapsto e_1, e_{y_i} \mapsto e_\mu, e_m \mapsto e_\mu, e_n \mapsto e_2 \mid 1 \leq i \leq n\}$$

由于 θ 满足条件所有的个体变量被映射到个体实例, 因此映射 θ 是一个实例化替换使得 $\mathcal{J}_p\theta = \mathcal{J}$ 。

3.3 决断集模版的生成与验证

为了避免出现决断集模板基数无限增大的情况, 我们需要的决断集模板的替换做出进一步限制。在实例化替换的基础上, 我们提出差异化实例替换。差异化实例替换不仅需要满足条件所有的个体变量都被映射到个体实例, 还需要满足条件对于所有的不相同变量, 被映射后的个体也不相同。直观地, 一个差异化实例替换会是变量到个体间的一一映射。同时, 一般地一个决断集模板需要能够生成至少一个决断集, 然而为了保证最后生成的解释的合理性, 我们需要决断集模板的所有差异化实例替换都只生成决断集, 因此我们需要限制所有的对于 $\alpha_{(x,y)}$ 决断集模板, 在差异化实例替换的映射下能够维持蕴含 $\alpha_{(x,y)}\theta$ 这一特性。因此我们对决断集模板做出如下定义:

定义 3.3 (决断集模版) 一个由扩展公理组成的集合 \mathcal{J}_p 会被称为在本体 \mathcal{O} 中对于 $\alpha_{(x,y)}$ 的决断集模版如果 \mathcal{J}_p 满足以下两个条件:(1) 存在一个替换 θ 使得 $\mathcal{J}_p\theta \in Jst(\alpha_{(x,y)}\theta, \mathcal{O})$ (2)对于所有的差异化实例替换 σ 存在 $\mathcal{J}_p\sigma \models r(\alpha_{(x,y)}\sigma)$ 。

继续考虑例3.1中的决断集模板:

$$\mathcal{J}_p = \{r_M \circ r_N \sqsubseteq r_K, r_M(e_{x_1}, e_{y_1}), \dots, r_M(e_{x_n}, e_{y_n}), r_N(e_m, e_n)\}$$

根据以上定义, 在任一个一致的本体 \mathcal{O} 中, \mathcal{J}_p 不是一个符合定义的对于二元关系断言实例 $r_K(e_1, e_2)$ 的决断集模板。因为对于决断集模板 \mathcal{J}_p 的差异化实例替换 θ , 存在:

$$\{r_M \circ r_N \sqsubseteq r_K, r_M(e_{x_1}\theta, e_{y_1}\theta), r_N(e_m\theta, e_n\theta)\} \in Jst(r_M(e_{x_1}, e_{y_1}), \mathcal{O})$$

同时:

$$\{r_M \circ r_N \sqsubseteq r_K, r_M(e_{x_1}\theta, e_{y_1}\theta), r_N(e_m\theta, e_n\theta)\} \subset \mathcal{J}_p\theta$$

因此可以得出:

$$\mathcal{J}_p\theta \notin Jst(r_M(e_{x_1}\theta, e_{y_1}\theta), \mathcal{O})$$

同时, \mathcal{J}_p 也不是 $r_M(e_{x_1}, e_{y_k})$ 的决断集模板当 $k > 1$ 。这是因为 $\mathcal{J}_p\sigma \not\models r_M(e_{x_1}\sigma, e_{y_k}\sigma)$, 其中:

$$\sigma = \{e_{x_i} \mapsto e_1, e_{y_i} \mapsto e_2 \mid 1 \leq i \leq n, i \neq k\} \cup \{e_{x_k} \mapsto e_2, e_{y_k} \mapsto e_1\}$$

为了保证决断集能够被映射到决断集空间上, 我们使用从决断集上生成决断集模板的方法来获取决断集模板。我们在生成的决断集的基础上, 对二元关系中的个体进行变量替换, 且对于不相同的个体名, 我们使用不同的个体变量替换。我们对用不相同变量替换不相同个体的过程记作 $lift(S, A, B)$ 。 $lift(S, A, B)$ 表示对公理集合中的所有个体名或是角色与概念, 我们把A映射为变量X, B映射为变量Y, 其它不相同的个体或是角色与概念分别映射到不同的个体变量。

命题 3.1 令 \mathcal{O} 为一致本体, $\alpha_{(e_1, e_2)}$ 是 \mathcal{O} 的一个蕴涵值, 且 \mathcal{J} 是在 \mathcal{O} 中对于蕴涵值 $\alpha_{(e_1, e_2)}$ 的一个决断集。那么 $lift(\mathcal{J}, e_1, e_2)$ 就是一个在 \mathcal{O} 中对于 $\alpha_{(e_1, e_2)}$ 的一个决断集模板。

证明: (1)因为在决断集 \mathcal{J} 与 $lift(\mathcal{J}, e_1, e_2)$ 之间具个体变量到个体实体的一对一映射, 因此这里存在一个 $lift(\mathcal{J}, e_1, e_2)$ 的差异化实例替换 θ 使得 $X\theta = e_1, Y\theta = e_2$,

且 $lift(\mathcal{J}, e_1, e_2) \cdot \theta = \mathcal{J}$ 。(2) 令 σ 为 $lift(\mathcal{J}, e_1, e_2)$ 的一个差异化实例替换, 因此对于 $lift(\mathcal{J}, e_1, e_2) \cdot \sigma$ 必然存在一个到 $lift(\mathcal{J}, e_1, e_2) \cdot \theta$ 个体变量之间的映射 ρ 从而使得 $X\theta = X\sigma$, $Y\theta = Y\sigma$ 且 $lift(\mathcal{J}, e_1, e_2) \cdot \theta \cdot \rho = lift(\mathcal{J}, e_1, e_2) \cdot \sigma$ 。又因为 $lift(\mathcal{J}, e_1, e_2) \cdot \theta \models \alpha_{(X,Y)}\theta$, 因此必然有 $lift(\mathcal{J}, e_1, e_2) \cdot \theta \cdot \rho \models \alpha(X,Y)\sigma$, 考虑到 $lift(\mathcal{J}, e_1, e_2) \cdot \theta \cdot \rho = lift(\mathcal{J}, e_1, e_2) \cdot \sigma$, 因此可知对于所有的差异化实例替换 σ 都有 $lift(\mathcal{J}, e_1, e_2) \cdot \sigma \models \alpha_{(X,Y)}\sigma$, 命题成立。

要使得对于观察值 α 以及一致本体 \mathcal{O} 中的溯因解释 \mathcal{E} 遵循决断集模板, 我们做出如下限制: \mathcal{E} 是观察值 α 在 $\mathcal{O} \cup \mathcal{E}$ 中的决断集的子集, 其中这个决断集是由决断集模板通过差异化实例替换计算得来。我们把这个解释 \mathcal{E} 称作可接受解释:

定义 3.4 给定一个一致本体 \mathcal{O} , 一个可蕴含 $\alpha_{(X,Y)}$ 决断集模板的集合 \mathcal{P} 以及观察值 α , 其中 $\mathcal{O} \not\models \alpha$, 解释 \mathcal{E} 如果满足存在一个决断集模板 $\mathcal{J}_p \in \mathcal{P}$ 以及一个在决断集模板 \mathcal{J}_p 上的差异化实例替换 θ 使得 $\alpha = \alpha_{(X,Y)}\theta$, $\mathcal{E} \subseteq \mathcal{J}_p\theta$ 以及 $\mathcal{J}_p\theta \in Jst(\alpha, \mathcal{O} \cup \mathcal{E})$, 我们就把这样的解释称作可接受解释。

下面给出一个可接受解释例子:

例 3.2 考虑以下本体 \mathcal{O} :

$$\{r_M \circ r_N \sqsubseteq r_K, r_M(e_1, e_3)\}$$

决断集模板 \mathcal{J}_p 为:

$$\{r_M \circ r_N \sqsubseteq r_K, r_M(A, e_w), r_N(e_w, B)\}$$

观察值 α 为:

$$r_K(e_1, e_2)$$

根据决断集模板 \mathcal{J}_p , 我们可以得到 $\mathcal{E} = r_M(e_1, e_3), r_N(e_3, e_2)$ 是一个对于观察值 $r_K(e_1, e_2)$ 在 \mathcal{O} 中的可接受解释, 因为存在一个差异化实例替换:

$$\theta = \{A \mapsto e_1, e_w \mapsto e_3, B \mapsto e_2\}$$

满足 $\alpha_{(e_1, e_2)} = \alpha_{(X, Y)}\theta$, $\mathcal{E} \subseteq \mathcal{J}_p\theta$ 且 $J_p\theta \in Jst(\alpha_{(e_1, e_2)}, \mathcal{O} \cup \mathcal{E})$ 。

给定一个一致本体 \mathcal{O} 以及一个观察 α , 其中 $\mathcal{O} \not\models \alpha$, 且 $\mathcal{O} \cup \{\alpha\}$ 是一致的, 那么本体的使用者更通常会考虑以下三个特性: 非平凡(i.e., $\mathcal{E} \not\models \alpha$), 一致(i.e. $\mathcal{O} \cup \mathcal{E}$ 是一致的)以及子集最小的(i.e., $\mathcal{O} \cup \mathcal{E} \not\models \alpha$ 对于所有的 $\mathcal{E}' \subset \mathcal{E}$)。我们称满足以上三个条件的解释为合理解释。可以看出, 在例子3.2中得到的解释 $\mathcal{E} = \{r_M(e_1, e_3), r_N(e_3, e_2)\}$ 不是一个合理的解释。因为存在一个 \mathcal{E} 的子集 $\mathcal{E}' = \{r_N(e_3, e_2)\}$ 满足 $\alpha_{(e_1, e_2)} = \alpha_{(X, Y)}\theta$, $\mathcal{E}' \subseteq \mathcal{J}_p\theta$ 且 $J_p\theta \in Jst(\alpha_{(e_1, e_2)}, \mathcal{O} \cup \mathcal{E}')$ 。所以, 我们结合以上条件定义如果一个可接受的解释满足非平凡, 一致以及子集最小这三个条件, 那么我们就称它为合理解释。

例 3.3 继续考虑例3.2, 本体 \mathcal{O} 以及 $\mathcal{O} \cup r_K(e_1, e_2)$ 也是一致的。正如前文提到的, \mathcal{E} 并不是一个合理的解释, 因为对于本体 \mathcal{O} 以及观察值 $r_K(e_1, e_2)$, \mathcal{E} 并不是一个最小集使得 $\alpha_{(e_1, e_2)} = \alpha_{(X, Y)}\theta$, $\mathcal{E} \subseteq \mathcal{J}_p\theta$ 且 $J_p\theta \in Jst(\alpha_{(e_1, e_2)}, \mathcal{O} \cup \mathcal{E})$ 。相对的, $\mathcal{E}' = \{r_N(e_3, e_2)\}$ 是根据模板集合 \mathcal{P} 在本体 \mathcal{O} 中对观察值 α 的一个合理解释, 使得 $\mathcal{E}' \not\models r_K(e_1, e_2)$, $\mathcal{O} \cup \mathcal{E}'$ 是一致的且对于所有的 $\mathcal{E}'' \subset \mathcal{E}'$ 不存在 $\mathcal{O} \cup \mathcal{E}'' \models r_K(e_1, e_2)$ 。

3.4 基于决断集模版的解释

在本体的诊断中, 由于解释需要遵循决断集模板, 而且不存在 $\mathcal{J}_p\theta \subset \mathcal{O}$ 的情况, 因此并非所有的变量个体都会被映射到本体 $\{\mathcal{O} \cup \alpha\}$ 的个体集合中。在本文中, 我们称这些变量为解释变量。由于解释变量的存在, 如果想保证诊断的方法能够更加高效, 一种考虑是只计算无法被其它解释通过实例化替换后得到的解释。然而解释变量的允许却会导致解释公理集合基数的变大甚至无法限制, 考虑以下解释 $\alpha_{(A, A)}$, 这个解释可以由解释 $\mathcal{E} = \{\alpha_{(A, e_{x_1})}, \alpha_{(e_{x_2}, e_{x_3})}, \dots, \alpha_{(e_{x_{n-1}}, e_n)}\}$ 通过映射 $\theta = \{e_{x_i} \mapsto A \mid 1 \leq i \leq n\}$ 获得。因此对于解释的替换, 我们提出一个新的概念, 叫做解释差异化替换。解释 \mathcal{E} 的差异化替换会可以把 \mathcal{E} 的解释变量映射到不同的解释变量或者是一个没有在 \mathcal{E} 中已经出现的变量。在前文提到的例子中, θ 并不

是一个解释的差异化替换，因为 e_{x_i} 被映射到的变量 A 是解释 \mathcal{E} 中已经存在的个体实例，所以 θ 不是解释的差异化替换。因此我们提出一个新的子集概念：

定义 3.5 (\subseteq_{ds} -minimal 解释) 一个解释 \mathcal{E} 会被称为 \subseteq_{ds} -minimal 如果这个 \mathcal{E} 满足以下条件：不存在这样一个解释 \mathcal{E}' 使得 $\mathcal{E}' \subseteq_{ds} \mathcal{E}$ 且 $\mathcal{E} \not\subseteq_{ds} \mathcal{E}'$ ，其中 $\mathcal{E}' \subseteq_{ds} \mathcal{E}$ 表示存在一个 \mathcal{E}' 的差异化替换 θ ，使得 $\mathcal{E}'\theta \subseteq \mathcal{E}$ 。

\subseteq_{ds} -minimal解释是一个比子集最小更强的概念，以下定理说明 \subseteq_{ds} -minimal是子集最小的充分条件：

命题 3.2 一个对于观察值 α 在本体 \mathcal{O} 中的 \subseteq_{ds} -minimal 解释 \mathcal{E} 同时也是一个subset-minimal的解释。

证明：对于所有的 $\mathcal{E}' \subset \mathcal{E}$ ，我们有 $\mathcal{E}' \subseteq_{ds} \mathcal{E}$ 且 $\mathcal{E}' \not\subseteq_{ds} \mathcal{E}$ 。所以，如果不存在对于观察值 α 在本体 \mathcal{O} 中的解释 \mathcal{E}' 使得 $\mathcal{E}' \subseteq_{ds} \mathcal{E}$ 且 $\mathcal{E}' \not\subseteq_{ds} \mathcal{E}$ ，那么也不会存在 $\mathcal{E}' \subset \mathcal{E}$ 使得 $\mathcal{O} \cup \mathcal{E}' \models \alpha$ ，所以命题成立。

验证解释 \mathcal{E} 是否是 \subseteq_{ds} -minimal不需要考虑解释 \mathcal{E} 的所有子集组合排列的情况。实际上我们只需要考虑 n 个更小的解释即可。这里的 n 等于 \mathcal{E} 中公理数的基数以及个体实例的总和。在介绍这两种方法之前，我们需要先介绍两种集合概念。一种是公理集 S 的最近变量集 $lifts_1(S)$ ，最近变量集 $lifts_1(S)$ 使用变量替换一个在 S 中已经存在的个体实例。例如：对于 $S = \{r_o(A, B)\}$ 存在两个最近变量集分别是 $\{\alpha_{(e_x, B)}\}$ 和 $\{\alpha_{(A, e_x)}\}$ 。然后，我们定义一个最大真子集，最大真子集有且仅有 S 中的某一公理外的所有公理。我们把最大真子集记作 $subs_1(S)$ 。以下命题展示了验证解释 \mathcal{E} 是否是一个 \subseteq_{ds} -minimal解释的方法。

命题 3.3 对于观察值 α 在本体 \mathcal{O} 中的解释 \mathcal{E} 是一个 \subseteq_{ds} -minimal解释当且仅当 \mathcal{E} 满足以下条件： $\mathcal{O} \cup \mathcal{E}' \not\models \alpha$ 对于所有的 $\mathcal{E}' \in subs_1(\mathcal{E}) \cup lifts_1(\mathcal{E})$ 。

证明：令 \mathcal{E} 是一个在本体 \mathcal{O} 中对于观察值 α 的解释，且 \mathcal{E} 满足 $\mathcal{O} \cup \mathcal{E}' \not\models \alpha$ 对于所有的 $\mathcal{E}' \in subs_1(\mathcal{E}) \cup lifts_1(\mathcal{E})$ 。令 \mathcal{E}' 为公理组成的集合， θ 是 \mathcal{E}' 的一个差异化替换使得 $\mathcal{E}'\theta \subset \mathcal{E}$ 。因此必然存在 $\mathcal{E}'' \in subs_1(\mathcal{E})$ 使得 $\mathcal{E}'\theta \subseteq \mathcal{E}''$ 。因为 $\mathcal{E}'' \cup \mathcal{O} \not\models \alpha$ ，所以

我们有 $\mathcal{E}'\theta \cup \mathcal{O} \not\models \alpha$, 从而 $\mathcal{E}' \cup \mathcal{O} \not\models \alpha$ 。令 \mathcal{E}' 是一组公理集合以及 θ 是 \mathcal{E} 的一个差异化替换使得 $\mathcal{E}'\theta = \mathcal{E}$ 并且 θ 满足存在至少一个实例使得 \mathcal{E}' 中的解释变量被映射到 \mathcal{E} 个体实例中。因此, 必然存在一个 $\mathcal{E}'' \in \text{lift}_{S_1}(\mathcal{E})$ 以及一个 σ 使得 $\mathcal{E}'\sigma = \mathcal{E}''$ 。因为 $\mathcal{E}'' \cup \mathcal{O} \not\models \alpha$, 所以我们又有 $\mathcal{E}' \cup \mathcal{O} \not\models \alpha$ 。因此对于所有满足 $\mathcal{E}' \subseteq_{\text{ds}} \mathcal{E}$ 且 $\mathcal{E} \not\subseteq_{\text{ds}} \mathcal{E}'$ 的 \mathcal{E}' 我们有 $\mathcal{E}' \cup \mathcal{O} \not\models \alpha$ 。所以 \mathcal{E} 是一个对于观察值 α 在 \mathcal{O} 中的 $\subseteq_{\text{ds}}\text{-minimal}$ 解释。

通过使用 $\subseteq_{\text{ds}}\text{-minimal}$, 我们获得了比子集最小更强的对解释的限制能力。因此, 结合上文提到的限制条件, 我们得到一类新的解释集合, 称为基于决断集模板的解释, 定义如下:

定义 3.6 (基于决断集模板的解释) 对于一个给定的本体 \mathcal{O} , 一个基于 $\alpha_{(X,Y)}$ 的模板 \mathcal{P} 以及一个观察值 α 其中 $\mathcal{T} \not\models \alpha$ 且 $\mathcal{O} \cup \{\alpha\}$ 是一致的, 对于 α 在 \mathcal{O} 的解释 \mathcal{E} 被称为基于决断集模板的解释, 如果这个解释满足以下四个条件:

- (非平凡) $\mathcal{E} \not\models \alpha$
- (一致) $\mathcal{O} \cup \mathcal{E}$ 是一致的
- ($\subseteq_{\text{ds}}\text{-minimal}$) 不存在一个对于 α 在 \mathcal{O} 中的解释 \mathcal{E}' 使得 $\mathcal{E}' \subseteq_{\text{ds}} \mathcal{E}$ 且 $\mathcal{E} \not\subseteq_{\text{ds}} \mathcal{E}'$ 。
- (可容许) 存在一个决断集模板 $\mathcal{J}_p \in \mathcal{P}$ 以及 \mathcal{J}_p 的一个差异化替换 θ 使得 $\alpha = \alpha_{(X,Y)}\theta$, $\mathcal{E} \subseteq \mathcal{J}_p\theta$ 且 $\mathcal{J}_p\theta \in \text{Jst}(\alpha, \mathcal{O} \cup \mathcal{E})$

计算基于决断集模板的解释是一个多项式时间内的计算过程。根据观察值 $\alpha_{(A,B)}$ 以及决断集模板集合 \mathcal{P} 计算在本体 \mathcal{O} 中的基于决断集模板解释步骤如下: 对于所有在 \mathcal{P} 中的决断集模板 \mathcal{J}_p , 我们使用二分法对决断集模板 \mathcal{J}_p 进行切分, 其中一部分公理的集合我们把它看做是本体 \mathcal{O} 的对应公理集合, 另一部分我们把它看做是我们要求得的基于决断集模板的解释。给定一个决断集模板 \mathcal{J}_p , 我们定义 \mathcal{J}_p 的二分结果为二元组 $(\mathcal{J}_1, \mathcal{J}_2)$, 其中 $\mathcal{J}_1 \cap \mathcal{J}_2 = \emptyset$ 且 $\mathcal{J}_1 \cup \mathcal{J}_2 = \mathcal{J}_p$ 。我们把 \mathcal{J}_p 二分后的二元组集合记作 $\text{B}^*(\mathcal{J}_p)$ 。同时, 对于一个由多条扩展公理的集合 S 我们把从个体变量映射到个体实例的一一映射操作记作 $\text{inst}(S)$ 。因此, 我们有如下命题:

命题 3.4 给定一个一致的本体 \mathcal{O} , 一个观察值 $\alpha_{(A,B)}$ 且 $\mathcal{O} \not\models \alpha_{(A,B)}$, 以及蕴含 $\alpha_{(X,Y)}$ 的决断集模板的集合 \mathcal{P} , 那么在本体 \mathcal{O} 中基于决断集模板集合 \mathcal{P} 对于观

察值 $\alpha_{(A,B)}$ 的基于决断集模板的解释的集合 $\mathcal{S} = \{inst(\mathcal{J}_1\theta) \mid \mathcal{J}_p \in \mathcal{P}, (\mathcal{J}_1, \mathcal{J}_2) \in B^*(\mathcal{J}_p) \text{ 且 } \theta \text{ 是 } \mathcal{J}_2 \cup \alpha_{(X,Y)} \text{ 的一个差异化实例替换使得 } X\theta = A, Y\theta = B, \mathcal{J}_2\theta \subseteq \mathcal{O}, \text{ 同时 } inst(\mathcal{J}_1\theta) \not\models \alpha_{(X,Y)}, \mathcal{O} \cup inst(\mathcal{J}_1\theta) \text{ 是一致的, 且 } \mathcal{J}_2\theta \cup inst(\mathcal{J}_1\theta) \in Jst(\alpha_{(A,B)}, \mathcal{O} \cup inst(\mathcal{J}_1\theta)), \text{ 以及不存在一个在本体 } \mathcal{O} \text{ 对于观察值 } \alpha_{(A,B)} \text{ 的解释 } \mathcal{E}' \text{ 满足 } \mathcal{E}' \subseteq_{ds} inst(\mathcal{J}_1\theta) \text{ 以及 } inst(\mathcal{J}_1\theta) \not\subseteq_{ds} \mathcal{E}'\}$ 。

证明：(1)令 \mathcal{E} 是集合 \mathcal{S} 中的一个元素，存在一个或多个的 $\mathcal{J}_p \in \mathcal{P}$ ， $(\mathcal{J}_1, \mathcal{J}_2) \in B^*(\mathcal{J}_p)$ 以及对于 $\mathcal{J}_2 \cup \{\alpha_{(X,Y)}\}$ 的差异化实例替换 θ 使得 $\mathcal{E} = inst(\mathcal{J}_1\theta)$ ， $X\theta = A, Y\theta = B, \mathcal{J}_2\theta \subseteq \mathcal{O}, \mathcal{E} \not\models \alpha_{(A,B)}$ ， $\mathcal{O} \cup \mathcal{E}$ 是一致的。 $\mathcal{J}_2\theta \in Jst(\alpha_{(A,B)}, \mathcal{O} \cup \mathcal{E})$ ，并且不存在一个在 \mathcal{O} 中对于 $\alpha_{(A,B)}$ 的解释 \mathcal{E}' 满足 $\mathcal{E}' \subseteq_{ds} \mathcal{E}$ 且 $\mathcal{E} \not\subseteq_{ds} \mathcal{E}'$ 。由于 \mathcal{E} 是由 $\mathcal{J}_1\theta$ 通过实例化替换而来，因此一定存在一个 \mathcal{J}_p 的实例化替换 θ' 使得 $\mathcal{J}_1\theta' = \mathcal{E}$ ， $\mathcal{J}_2\theta' = \mathcal{J}_2\theta, X\theta' = X\theta = A$ 以及 $Y\theta' = Y\theta = B$ 。通过定义可知，我们有 $\mathcal{J}_p\theta' \models \alpha_{(X,Y)}$ 。由于 $\mathcal{J}_p\theta' = \mathcal{J}_2\theta \cup \mathcal{E}$ 并且 $\mathcal{J}_2\theta \subseteq \mathcal{O}$ ，我们有 $\mathcal{J}_p\theta' \subseteq \mathcal{O} \cup \mathcal{E}$ 所以 $\mathcal{O} \cup \mathcal{E} \models \alpha_{(A,B)}$ ， \mathcal{E} 是在本体 \mathcal{O} 中对于观察值 $\alpha_{(A,B)}$ 的一个解释。另外， \mathcal{E} 满足基于决断集模板解释的四个特性：非平凡(i.e., $\mathcal{E} \not\models \alpha$)，一致(i.e. $\mathcal{O} \cup \mathcal{E}$ 是一致的)以及 $\subseteq_{ds} -minimal$ (不存在这样一个解释 \mathcal{E}' 使得 $\mathcal{E}' \subseteq_{ds} \mathcal{E}$ 且 $\mathcal{E} \not\subseteq_{ds} \mathcal{E}'$ ，其中 $\mathcal{E}' \subseteq_{ds} \mathcal{E}$ 表示存在一个 \mathcal{E}' 的差异化替换 θ ，使得 $\mathcal{E}'\theta \subseteq \mathcal{E}$)，以及可容许(存在一个决断集模版 $\mathcal{J}_p \in \mathcal{P}$ 以及 \mathcal{J}_p 的一个差异化替换 θ 使得 $\alpha = \alpha_{(X,Y)}\theta$ ， $\mathcal{E} \subseteq \mathcal{J}_p\theta$ 且 $\mathcal{J}_p\theta \in Jst(\alpha, \mathcal{O} \cup \mathcal{E})$)，所以 \mathcal{E} 是一个在 \mathcal{O} 中对于 $\alpha_{(A,B)}$ 的一个基于决断集模板的解释。(2)令 \mathcal{E} 为在本体 \mathcal{O} 中基于决断集模板 \mathcal{P} 对观察值 $\alpha_{(A,B)}$ 的解释。由于 \mathcal{E} 是一个可接受的解释，因此必定存在一个决断集模板 $\mathcal{J}_p \in \mathcal{P}$ 以及一个 \mathcal{J}_p 的差异化实例替换 θ 使得 $X\theta = A, Y\theta = B, \mathcal{E} \subseteq \mathcal{J}_p\theta$ 以及 $\mathcal{J}_p\theta \in Jst(\alpha_{(A,B)}, \mathcal{O} \cup \mathcal{E})$ 。由于 $\mathcal{E} \subseteq \mathcal{J}_p\theta$ 以及 $\mathcal{J}_p\theta \subseteq$ ，所以必定存在一个 \mathcal{J}_p 的二分 $(\mathcal{J}_1, \mathcal{J}_2)$ 使得 $\mathcal{E} = \mathcal{J}_1\theta$ 以及 $\mathcal{J}_2\theta \subseteq \mathcal{O}$ 。令 θ' 是一个 $\mathcal{J}_2 \cup \{\alpha_{(A,B)}\}$ 使得 $X\theta' = X\theta, Y\theta' = Y\theta$ 以及 $\mathcal{J}_2\theta' = \mathcal{J}_2\theta$ ，所以 $X\theta' = A, Y\theta' = B, \mathcal{J}_2\theta' \subseteq \mathcal{O}$ 以及 \mathcal{E} 是 $inst(\mathcal{J}_1\theta')$ 的重名等价解。又因为 \mathcal{E} 是非平凡的，我们有 $inst(\mathcal{J}_1\theta') \not\models \alpha_{(A,B)}$ 。 \mathcal{E} 是一致的，因此 $\mathcal{O} \cup inst(\mathcal{J}_1\theta')$ 是一致的。 $\mathcal{J}_2\theta \cup \mathcal{E} \in Jst(\alpha_{(A,B)}, \mathcal{O} \cup \mathcal{E})$ ，我们有 $\mathcal{J}_2\theta' \cup inst(\mathcal{J}_1\theta') \in Jst(\alpha_{(A,B)}, \mathcal{O} \cup inst(\mathcal{J}_1\theta'))$ 。又 \mathcal{E} 是 $\subseteq_{ds} -minimal$ 的，因此不存在一个在本体 \mathcal{O} 对于观察值 $\alpha_{(A,B)}$ 的解释 \mathcal{E}' 使得 $\mathcal{E}' \subseteq_{ds} inst(\mathcal{J}_1\theta')$ 并 $inst(\mathcal{J}_1\theta') \not\subseteq_{ds} \mathcal{E}'$ 。

\mathcal{E}' 。

在上文中，我们已经证明了基于决断集模板的解释可以在一个有限的空间中形成。

3.5 生成基于决断集模版解释的算法

在本节中，我们主要介绍生成基于决断集模版解释的算法。给定训练集，观察值以 $r_o(A, B)$ 为例，以及蕴含 $r_o(X, Y)$ 的决断集模版的集合 \mathcal{P} ，算法2 计算出所有相应的基于决断集模版的解释集。对于给定的输入，算法2首先检测表达本体 \mathcal{O} 的一致性，之后对于所有的决断集模版 \mathcal{J}_p ，我们使用二分法生成 $(\mathcal{J}_1, \mathcal{J}_2)$ 。其中 \mathcal{J}_1 用于在 $\mathcal{O} \cup r_o(A, B)$ 中寻找实例并计算出替换 θ ， \mathcal{J}_2 用于生成解释并验证是否合理解释。遍历所有的 $(\mathcal{J}_1, \mathcal{J}_2)$ (除了 $\mathcal{J}_1 = \emptyset$ 或 $\mathcal{J}_2 = \emptyset$)得出解释集。例3.4是根据算法给出的例子。

例 3.4 (诊断) 考虑以下诊断问题，其中观察值为：

$$(Mike, Nationality, China)$$

蕴含值 $r_{Nationality}(Mike, China)$ 的决断集：

$$\mathcal{J} = \left\{ \begin{array}{l} r_{BornInCity} \circ r_{CityInProvince} \sqsubseteq r_{BornInProvince} \\ r_{BornInProvince} \circ r_{ProvinceInCountry} \sqsubseteq r_{Nationality} \\ r_{BornInCity}(Mike, Guangzhou) \\ r_{CityInProvince}(Guangzhou, Guangdong) \\ r_{ProvinceInCountry}(Guangdong, China) \end{array} \right\}.$$

训练集：

$$t = \left\{ \begin{array}{l} (Mike, BornInCity, Hangzhou) \\ (Beihai, CityInProvince, Guangxi) \\ (Zhejiang, ProvinceInCountry, China) \\ (Guangxi, ProvinceInCountry, China) \end{array} \right\}.$$

```

Input: Triples  $\mathcal{A}$  in training data set, observation  $r_o(A, B)$ , Justification
        Pattern  $\mathcal{J}_p$  for  $r_o(X, Y)$ 
Output: Justification Pattern based Explanations  $\mathcal{S}$  for observation  $r_o(A, B)$ 
        in  $\mathcal{A}$  w.r.t  $\mathcal{P}$ 

 $\mathcal{S} \leftarrow \emptyset$ 
 $\mathcal{T}_p \leftarrow TBox \mathcal{T}$  in  $\mathcal{J}_p$ ;
 $\mathcal{O} \leftarrow \mathcal{T}_p \cup \mathcal{A}$ ;
if  $\mathcal{O}$  is not consistent then
    | return  $\emptyset$ ;
end
if  $\mathcal{O} \cup r_o(A, B)$  is not consistent then
    | return  $\emptyset$ ;
end
 $B^* \leftarrow bipart(\mathcal{J}_p)$ ;
for each  $(\mathcal{J}_1, \mathcal{J}_2) \in B^*$  do
     $D \leftarrow \{\theta \mid \theta \text{ is a differentiated substitution for } \mathcal{J}_p \text{ such that}$ 
         $X\theta = A, Y\theta = B, \mathcal{J}_1\theta \subseteq \mathcal{O}\}$ ;
    for each  $\theta \in D$  do
        if  $\mathcal{J}_2\theta \models r_o(A, B)$  then
            | continue;
        end
        if  $\mathcal{J}_2\theta$  or  $\mathcal{J}_2\theta \cup \mathcal{O}$  is not consistent then
            | continue;
        end
         $\mathcal{S}_e \leftarrow \{\mathcal{E}' \mid \mathcal{E}' \in subs_1(\mathcal{J}_2\theta) \cup lifts_1(\mathcal{J}_2\theta)\}$ ;
         $isMinimal \leftarrow true$ ;
        for each  $\mathcal{E}' \in \mathcal{S}_e$  do
            if  $\mathcal{E}' \cup \mathcal{O} \models r_o(X, Y)$  then
                |  $isMinimal \leftarrow false$ ;
                | continue;
            end
        end
        if  $isMinimal$  then
            |  $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{E}'$ 
        end
    end
end
end

```

Algorithm 2: 基于决断集模板诊断算法

根据例子，首先使用本体语言对训练集进行表达，得到ABox \mathcal{A} :

$$\mathcal{A} = \left\{ \begin{array}{l} r_{BornInCity}(\text{Mike}, \text{Hangzhou}) \\ r_{CityInProvince}(\text{Beihai}, \text{Guangxi}) \\ r_{ProvinceInCountry}(\text{Zhejiang}, \text{China}) \\ r_{ProvinceInCountry}(\text{Guangxi}, \text{China}) \end{array} \right\}.$$

然后可得本体 \mathcal{O} :

$$\mathcal{O} = \left\{ \begin{array}{l} r_{BornInCity} \circ r_{CityInProvince} \sqsubseteq r_{BornInProvince} \\ r_{BornInProvince} \circ r_{ProvinceInCountry} \sqsubseteq r_{Nationality} \\ r_{BornInCity}(\text{Mike}, \text{Hangzhou}) \\ r_{CityInProvince}(\text{Beihai}, \text{Guangxi}) \\ r_{ProvinceInCountry}(\text{Zhejiang}, \text{China}) \\ r_{ProvinceInCountry}(\text{Guangxi}, \text{China}) \end{array} \right\}.$$

易得， \mathcal{O} 与 $\mathcal{O} \cup r_{Nationality}(\text{Mike}, \text{China})$ 都是是一致的。又，通过决断集可以得到蕴含 $r_{Nationality}(A, B)$ 的基于决断集模板 \mathcal{J}_p :

$$\mathcal{J}_p = \left\{ \begin{array}{l} r_{BornInCity}(A, C) \\ r_{CityInProvince}(C, D) \\ r_{ProvinceInCountry}(D, B) \end{array} \right\}.$$

观察值 α 为:

$$r_{Nationality}(\text{Mike}, \text{China})$$

对决断集模板进行二分划分 $(\mathcal{J}_1, \mathcal{J}_2)$ ，使得 $\mathcal{J}_1 \cap \mathcal{J}_2 = \emptyset$ 且 $\mathcal{J}_1 \cup \mathcal{J}_2 = \mathcal{J}_p$ 。

(1) $\mathcal{J}_1 = \{r_{BornInCity}(A, C), r_{CityInProvince}(C, D)\}$, $\mathcal{J}_2 = \{r_{ProvinceInCountry}(D, B)\}$ 。由

于不存在一个差异化实例替换 θ 使得 $\mathcal{J}_1\theta \subseteq \mathcal{O}$ ，因此划分1)无法生成合理解释。

(2) $J_1 = \{r_{BornInCity}(A, C)\}$, $J_2 = \{r_{CityInProvince}(C, D), r_{ProvinceInCountry}(D, B)\}$ 。根据划分2), 可得 θ 使得 $\mathcal{J}_1\theta \subseteq \mathcal{O}$, 其中 $\theta = \{A \mapsto Mike, C \mapsto Hangzhou, B \mapsto China\}$, 设 $\mathcal{E}_2 = \mathcal{J}_2\theta = \{r_{CityInProvince}(Hangzhou, e_1), r_{ProvinceInCountry}(e_1, China)\}$, 又因为 $\mathcal{O} \cup \mathcal{J}_2\theta$ 是一致的, $\mathcal{O} \cup \mathcal{J}_2\theta \models \alpha$, 且对于所有的 $\mathcal{E}' \in subs_1(\mathcal{E}_2) \cup lifts_1(\mathcal{E}_2)$ 都有 $\mathcal{O} \cup \mathcal{E}' \not\models \alpha$, 所以 \mathcal{E}_2 是一个合理的解释。

(3) $J_1 = \{r_{BornInCity}(A, C), r_{ProvinceInCountry}(D, B)\}$, $J_2 = \{r_{CityInProvince}(C, D)\}$ 。根据划分3), 可以得到一个 θ 使得 $\mathcal{J}_1\theta \subseteq \mathcal{O}$, 其中 $\theta = \{A \mapsto Mike, C \mapsto Hangzhou, B \mapsto China, D \mapsto Zhejiang\}$, 所以我们设 $\mathcal{E}_3 = \mathcal{J}_2\theta = \{r_{CityInProvince}(Hangzhou, Zhejiang)\}$, 又因为 $\mathcal{O} \cup \mathcal{J}_2\theta$ 是一致的, $\mathcal{O} \cup \mathcal{J}_2\theta \models \alpha$, 且对于所有的 $\mathcal{E}' \in subs_1(\mathcal{E}_2) \cup lifts_1(\mathcal{E}_3)$ 都有 $\mathcal{O} \cup \mathcal{E}' \not\models \alpha$, 所以 \mathcal{E}_3 是一个合理的解释。

(4) $J_1 = \{r_{ProvinceInCountry}(D, B)\}$, $J_2 = \{r_{BornInCity}(A, C), r_{CityInProvince}(C, D)\}$ 。根据划分4), 可以得到一个 θ 使得 $\mathcal{J}_1\theta \subseteq \mathcal{O}$, 其中 $\theta = \{A \mapsto Mike, B \mapsto China, D \mapsto Zhejiang\}$, 所以我们设 $\mathcal{E}_4 = \mathcal{J}_2\theta = \{r_{BornInCity}(Mike, e_1), r_{CityInProvince}(e_1, Zhejiang)\}$, 又因为 $\mathcal{O} \cup \mathcal{J}_2\theta$ 是一致的, $\mathcal{O} \cup \mathcal{J}_2\theta \models \alpha$, 且对于所有的 $\mathcal{E}' \in subs_1(\mathcal{E}_4) \cup lifts_1(\mathcal{E}_4)$ 都有 $\mathcal{O} \cup \mathcal{E}' \not\models \alpha$, 所以 \mathcal{E}_4 是一个合理的解释。

(5) $J_1 = \{r_{CityInProvince}(C, D), r_{ProvinceInCountry}(D, B)\}$, $J_2 = \{r_{BornInCity}(A, C)\}$ 。根据划分5), 可以得到一个 θ 使得 $\mathcal{J}_1\theta \subseteq \mathcal{O}$, 其中 $\theta = \{A \mapsto Mike, C \mapsto Beihai, B \mapsto China, D \mapsto Guangxi\}$, 所以我们设 $\mathcal{E}_5 = \mathcal{J}_2\theta = \{r_{BornInCity}(Mike, Beihai)\}$, 又因为 $\mathcal{O} \cup \mathcal{J}_2\theta$ 是一致的, $\mathcal{O} \cup \mathcal{J}_2\theta \models \alpha$, 且对于所有的 $\mathcal{E}' \in subs_1(\mathcal{E}_5) \cup lifts_1(\mathcal{E}_5)$ 都有 $\mathcal{O} \cup \mathcal{E}' \not\models \alpha$, 所以 \mathcal{E}_5 是一个合理的解释。

(6) $J_2 = \{r_{CityInProvince}(C, D)\}$, $J_1 = \{r_{BornInCity}(A, C), r_{ProvinceInCountry}(D, B)\}$ 。根据划分6), 可以得到一个 θ 使得 $\mathcal{J}_1\theta \subseteq \mathcal{O}$, 其中 $\theta = \{A \mapsto Mike, C \mapsto Beihai, B \mapsto China, D \mapsto Guangxi\}$, 所以设 $\mathcal{E}_6 = \mathcal{J}_2\theta = \{r_{BornInCity}(Mike, Beihai), r_{ProvinceInCountry}(Guangxi, China)\}$ 又 $\mathcal{O} \cup \mathcal{J}_2\theta$ 是一致的, $\mathcal{O} \cup \mathcal{J}_2\theta \models \alpha$ 。但是因为存在一个 $\mathcal{E}' \in subs_1(\mathcal{E}_6) \cup lifts_1(\mathcal{E}_6)$ 使得 $\mathcal{O} \cup \mathcal{E}' \models \alpha$, 所以 \mathcal{E}_6 不是一个合理的解释。

综上所述, 可以得到可接受的解释为 $\{\mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4, \mathcal{E}_5, \mathcal{E}_6\}$, 经验证, 其中合理的解释为 $\{\mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4, \mathcal{E}_5\}$ 。

3.6 本章小结

在本章中，我们首先对求解的问题进行了定义，针对待求解的问题我们首先提出了变量公理的概念。在介绍了决断集后，我们结合变量公理对决断集模版进行了定义。为了限制对解释空间进行限制，我们提出了 \subseteq_{ds} -*minimal*概念，并证明了 \subseteq_{ds} -*minimal*的性质比*subset-minimal*更强。 \subseteq_{ds} -*minimal*的解释不仅满足了最小解的限制，还可以对解释集中的语义等价解进行表达，从而减少了需要计算的解释数量。

我们提出基于决断集模版的目的，就是利用决断集模版生成符合语义逻辑的最小解释。通过对决断集模版进行 $(\mathcal{J}_1, \mathcal{J}_2)$ 划分，我们在知识库中寻找匹配的三元组(如果存在)计算替换 θ ，然后利用替换 θ 计算解释。最后，我们展示了针对问题 Δ_p 和观察值 α 进行解释集求解的完整算法。

第4章 解释集生成与嵌入更新

在上一章中，我们给出了在知识库中进行诊断的基本框架，并且详细地对相关的定理进行了证明。但是我们还没有完成从解释集到嵌入向量的翻译，以及实际的求解算法，因此，在本章当中，我们先给出生成解释训练集的方法，接着给出在解释训练集到嵌入向量的一般性算法。

4.1 决断集计算

基于决断集模版的解释需要特定的决断集模版的集合。在上一章的命题中我们展示了生成决断集模版的方法，我们首先计算本体中所有的原子蕴含值，接着根据蕴含值我们计算出所有蕴含值的决断集，最后通过 $lift$ 操作生成决断集模版变量。然而，这种方法并不完全可行，因为计算所有蕴含值的决断集是非常耗时的[29]。所以我们提出了新的计算决断集的方法。

这种方法是从Du, Junker等人的方法改进而来[28, 35]。对于本体 \mathcal{O} 的蕴含值 $\alpha_{A,B}$ ，我们在每次计算新的决断集之前都会通过决断集模版获取具有相同决断集模版的决断集。然后通过逐条递增的方式增量计算不能从已有的决断集模版获取的决断集，这种方式使用了更有效地剪枝技术和分治策略，以最大程度避免对相同类型的决断集进行重复计算，算法的流程会重复以上步骤，直至所有的决断集被计算出来为止。

在以下算法中，我们对于所有的原子蕴含值进行顺序求解。例如，对于所有由本体 \mathcal{O} 蕴含的形如 $\alpha_{A,B}$ ，我们首先计算所有对于 $\alpha_{A,B}$ 的决断集模版。利用[28]中的搜索方法，它可以递增地计算新的决断集。

命题 4.1 算法3返回在本体 \mathcal{O} 中对于蕴含值 $\alpha_{A,B}$ 的决断集模版集合 \mathcal{P} 以及在 \mathcal{O} 中对于蕴含值 $\alpha_{A,B}$ 的决断集 \mathcal{J} ，且存在差异化实例替换 θ ，决断集模版 $\mathcal{J}_p \in \mathcal{P}$ 以及决断集 \mathcal{J} 使得 $\mathcal{J}_p\theta = \mathcal{J}$ 。

证明：对于每个 $\mathcal{J}_p \in \mathcal{P}$ ，存在一个 \mathcal{O} 的蕴含 $\alpha_{A,B}$ 满足 $\mathcal{J}_p = lift(\mathcal{J}, A, B)$ 。其中 \mathcal{J}_p 是 \mathcal{O} 中的一个决断集模版，另一方面，令 θ 是 \mathcal{J}_p 的一个实例化差异替换

满足 $X\theta = A, Y\theta = B, \mathcal{J}_p\theta \subseteq \mathcal{O}$ 以及对于所有的 $\mathcal{J} \in \text{subs}_1(\mathcal{J}_p\theta)$ 有 $\mathcal{J} \not\models \alpha_{A,B}$ 。根据定义，我们有 $\mathcal{J}_p\theta \models \alpha_{A,B}$ 。由于 $\mathcal{J}_p\theta \subseteq \mathcal{O}$ 且对于所有的 $\mathcal{J} \in \text{subs}_1(\mathcal{J}_p\theta)$ 有 $\mathcal{J} \not\models \alpha_{A,B}$ ，所以 $\mathcal{J}_p\theta$ 是 \mathcal{O} 中对于蕴含 $\alpha_{A,B}$ 的一个决断集。

Input: Ontology \mathcal{O}

$\mathcal{P} \leftarrow \emptyset$ // \mathcal{P} 存储所有的决断集模版

for each $\alpha_{A,B}$ *entailed by* \mathcal{O} **do**

$\mathcal{S} \leftarrow \{\mathcal{J}_p\theta \mid \mathcal{J}_p \in \mathcal{P}, \theta \text{ is a differentiated substitution for } \mathcal{J}_p \text{ such that } X\theta = A, Y\theta = B, \mathcal{J}_p\theta \subseteq \mathcal{O} \text{ and } \mathcal{J} \not\models \alpha_{A,B} \text{ for all } \mathcal{J} \in \text{subs}_1(\mathcal{J}_p\theta)\};$

while $\mathcal{S} = \emptyset$ (then H is set as \emptyset) or there is a minimal hitting set H for \mathcal{S} such that $\mathcal{T} \setminus H \models \alpha_{A,B}$ **do**

$\mathcal{J} \leftarrow \text{FindJustification}(\emptyset, \mathcal{O} \setminus H, \alpha_{A,B});$

$\mathcal{J}_p \leftarrow \text{lift}(\mathcal{J}, A, B)$

$\mathcal{P} \leftarrow \mathcal{P} \cup \{\mathcal{J}_p\};$

$\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathcal{J}_p\theta \mid \theta \text{ is a differentiated substitution for } \mathcal{J}_p \text{ such that } X\theta = A, Y\theta = B, \mathcal{J}_p\theta \subseteq \mathcal{O} \text{ and } \mathcal{J} \not\models \alpha_{A,B} \text{ for all } \mathcal{J} \in \text{subs}_1(\mathcal{J}_p\theta)\};$

end

end

return $\mathcal{P};$

FindJustification($S_u, S_c, \alpha_{A,B}$) // 返回一个 S_c 的最小子集 S 使得 $S_u \cup S \models \alpha_{A,B}$;

得 $S_u \cup S \models \alpha_{A,B}$;

if $S_c = \emptyset$ or $S_u \models \alpha_{A,B}$ **then**

return \emptyset

end

if $|S_c| = 1$ **then**

return S_c

end

Divide S_c into two disjoint subsets S_1 and S_2 such that $S_1 \cup S_2 = S_c$,

$S_1 \cap S_2 = \emptyset$ and $0 \leq |S_1| - |S_2| \leq 1$;

$\Delta_2 \leftarrow \text{FindJustification}(S_u \cup S_1, S_2, \alpha_{A,B});$

$\Delta_1 \leftarrow \text{FindJustification}(S_u \cup \Delta_2, S_1, \alpha_{A,B});$

return $\Delta_1 \cup \Delta_2$

Algorithm 3: 决断集模版的计算方法

定义 4.1 (最小碰集) 一个集合 H 如果满足条件：在一个由公理集组成的集合 S 里，如果对于所有的 $S \in \mathcal{S}$ 有 $H \cap S \neq \emptyset$ ，且 H 是最小集，那么这个 H 就被称为 S 的最小碰集(MHS)。

由于最小碰集与求得的决断集的集合中所有的元素交集都不为空，且是最小集，我们可以判断如果存在一个决断集集合的最小碰集 H 使得 $\mathcal{O} \setminus H \models \alpha_{A,B}$ ，一个新的决断集可以从集合 $\mathcal{O} \setminus H$ 中通过分治法得到[28]。

根据以上算法，我们给出以下例子：

例 4.1 这个例子主要叙述如何使用算法3在本体 \mathcal{O} 进行运算，其中 \mathcal{O} 包含以下六条公理：

$$\mathcal{O} = \left\{ \begin{array}{l} \alpha_1 : A \sqsubseteq \exists r.C \quad \alpha_2 : A \sqsubseteq B \sqcup C \\ \alpha_3 : A \sqcap C \sqsubseteq \perp \quad \alpha_4 : C \sqsubseteq D \\ \alpha_5 : \exists r.D \sqsubseteq B \quad \alpha_6 : D \sqsubseteq E \end{array} \right\}.$$

可以得到其中 $A \sqsubseteq B$ 可由 \mathcal{O} 蕴含。我们以 $A \sqsubseteq B$ 为例进行运算，第一轮运算中，我们使用**FindJustification** $(\emptyset, \mathcal{O}, A \sqsubseteq B)$ 计算第一个决断集。我们把 \mathcal{O} 划分为 $S_1 = \{\alpha_1, \alpha_2, \alpha_3\}$ 以及 $S_2 = \{\alpha_4, \alpha_5, \alpha_6\}$ 。通过递归调用**FindJustification** $(S_1, S_2, A \sqsubseteq B)$ ，我们得到 $\Delta_2 = \emptyset$ 。又通过递归调用**FindJustification** $(\Delta_2, S_1, A \sqsubseteq B)$ ，我们可以得到 $\Delta_1 = \{\alpha_2, \alpha_3\}$ 。所以 $A \sqsubseteq B$ 的第一个决断集是 $\mathcal{J}_1 = \{\alpha_2, \alpha_3\}$ 。通过 $lift\{\mathcal{J}_1\}$ ，我们得到第一个 $X \sqsubseteq Y$ 的决断集模版 $\mathcal{J}_{p1} = \{X \sqsubseteq Y \sqcup Z_C, X \sqcap Z_C \sqsubseteq \perp\}$ 。由于无法在本体 \mathcal{O} 中找到可实例化的决断集，我们计算 \mathcal{J}_1 的MHS H ，令 $H = \{\alpha_2\}$ 使得 $\mathcal{O} \setminus H \models A \sqsubseteq B$ 。我们继续调用**FindJustification** $(\emptyset, \mathcal{O} \setminus H, A \sqsubseteq B)$ 得到第二个决断集 $\mathcal{J}_2 = \{\alpha_1, \alpha_4, \alpha_5\}$ 。由于不存在 $\{\mathcal{J}_1, \mathcal{J}_2\}$ 的MHS H 使得 $\mathcal{O} \setminus H \models A \sqsubseteq B$ ，我们已经得到全部的决断集 $\{\mathcal{J}_1, \mathcal{J}_2\}$ 。

4.2 多观察值的最小解释集生成

针对知识库多观察值的特点，我们在对于单个观察值的解释的基础上还对多观察值的情况进行求解。实际上，对多观察值的解释进行求解可以在分别求解单个观察值解释的基础上继续进行求解。我们把由多观察值构成的本体记作 \mathcal{O}_ϕ 并有如下定义：

定义 4.2 (状态蕴含) 如果对于所有的在 \mathcal{O}_ϕ 中的公理 α 满足 $\mathcal{O} \models \alpha$, 我们就称 \mathcal{O} 状态蕴含 \mathcal{O}_ϕ , 记作 $\mathcal{O} \models_S \mathcal{O}_\phi$ 。

我们令 \mathcal{S}_ε 为 \mathcal{O}_ϕ 中所有观察值 α 的解释集集合。由算法3我们知道在解释集的生成过程中, 我们需要多次地执行返回一个 \mathcal{S}_ε 的最小子集 S 使得 $S_u \cup S \models A \sqsubseteq B$ 的运算, 因此可以知道计算 \mathcal{O}_ϕ 的最小解释集其实是决断集生成算法逻辑上的一个子任务, 因此我们根据决断集生成算法提出了多观察值的最小解释集计算方法。

```

Input: Ontology  $\mathcal{O}$ , observation ontology  $\mathcal{O}_\phi$ , explanation set  $\mathcal{S}_\varepsilon$ 
return FindMinimalExp( $\mathcal{O}$ ,  $\mathcal{S}_\varepsilon$ ,  $\mathcal{O}_\phi$ );
FindMinimalExp( $S_u$ ,  $S_c$ ,  $\mathcal{O}$ ) // 返回一个 $\mathcal{S}_\varepsilon$ 的最小子集 $S$ 使得 $S_u \cup S \models_S \mathcal{O}$ ;
if  $S_c = \emptyset$  or  $S_u \models_S \mathcal{O}$  then
  | return  $\emptyset$ 
end
if  $|S_c| = 1$  then
  | return  $S_c$ 
end
Divide  $S_c$  into two disjoint subsets  $S_1$  and  $S_2$  such that  $S_1 \cup S_2 = S_c$ ,
 $S_1 \cap S_2 = \emptyset$  and  $0 \leq |S_1| - |S_2| \leq 1$ ;
 $\Delta_2 \leftarrow$  FindMinimalExp( $S_u \cup S_1$ ,  $S_2$ ,  $\mathcal{O}$ );
 $\Delta_1 \leftarrow$  FindMinimalExp( $S_u \cup \Delta_2$ ,  $S_1$ ,  $\mathcal{O}$ );
return  $\Delta_1 \cup \Delta_2$ 

```

Algorithm 4: 多观察值的最小解释集生成

4.3 基于解释集的向量模型更新研究

在本节中我们主要考虑向量模型的更新问题。在目前的不少应用中, 知识库模型往往在一个在线的系统中提供服务, 每一时刻都在服务数量较大的用户。因此, 如果问题产生时不能够得到及时地解决, 就会带来较大的影响。所以一般的, 在目前的诊断应用中, 我们对诊断的要求是时效性敏感的, 我们希望在得知问题根源后, 能够尽快地更新知识库模型。

在目前的知识库嵌入研究中, 以Bordes等人提出的TransE影响最大, TransE模型具有语义直观, 模型清晰明了的特点, 因此下文以TransE的模型为基础, 探索在经过决断集模版求解出解释集后, 如何使用新增三元组进行模型更新措施。

在TransE中, 嵌入是一个关键的操作, 它的作用是在实体、关系的集合与一个低维的向量空间中形成一个映射关系。当新的实体、关系被加进知识库后, 意味着我们需要对这个映射关系进行修改。

首先, 我们把知识库记为 G , 对知识库 G 的嵌入记为 $embed(G)$, 在知识库的基础上由知识库嵌入生成的向量模型为 Δ_G , 记作 $embed(G) = \Delta_G$, 并且我们有如下定义:

定义 4.3 (向量蕴含) 若向量模型 Δ_G 中, 测试三元组 (h, r, t) 可以由 Δ_G 成功预测且 $(h, r, t) \notin G$, 我们就称向量模型 Δ_G 向量蕴含 (h, r, t) , 记作 $\Delta_G \models_{\Delta} (h, r, t)$ 。

根据基于决断集模版解释的定义以及对解释集的求解过程可知, 对于所有的 $\alpha \in S_{\mathcal{E}}$, 满足 $\alpha \notin \mathcal{O}_G$, 所以对于由解释集生成的知识库 $G_{\mathcal{E}}$ 有对于所有的 $(h, r, t) \in G_{\mathcal{E}}$ 满足 $(h, r, t) \notin \Delta_G$ 。因此, 在继续训练更新向量模型 Δ_G 之前, 还难以使得尽量多的 $(h, r, t) \in G_{\mathcal{E}}$ 满足 $\Delta_G \models_{\Delta} (h, r, t)$ 。

我们的目标是在 $G_{\mathcal{E}}$ 的加入后训练的结果能够使得 $h + r \approx t$, 其中 $(h, r, t) \in G_{\mathcal{E}}$ 。因此, 我们不对TransE的能量框架做出改变, 对于三元组 (h, r, t) 我们依旧把它的能量定义为 $d(h + r, t)$, 并且 d 为欧几里得距离。由于随机梯度下降具有增量更新的性质, 因此一个可行的方式是使用原来的损失函数继续进行训练直至 \mathcal{L} 的值能够再次收敛到一个局部最小值。

要注意的是, 根据我们基于决断集模版生成的解释集虽然不会对知识库 G 中的关系 r 带来影响, 但是基于决断集模版生成的解释中, 我们是允许实体变量的存在的(特别是当 $\text{Exp.C} \geq 2$ 时), 所以当我们使用解释集生成的三元组 $S_{\mathcal{E}}$ 对知识库进行修复时, 会有新的变量加入, 也就意味着在解释集生成三元组的加入后, $G_{\mathcal{E}}$ 的实体集合 $E' \not\subseteq E$ 。

$$\mathcal{L} = \sum_{(h,r,t) \in S \cup S_{\mathcal{E}}} \sum_{(h',r,t') \in (S \cup S_{\mathcal{E}})'_{(h,r,t)}} [\gamma + d(h + r, t) - d(h' + r, t')]_+ \quad (4.1)$$

由于我们对知识库进行了更新, 我们在生成反例(也就是损坏三元组)的时候需要改变生成的逻辑, 对于所有的损坏三元组 $(h', r, t') \in (S \cup S_{\mathcal{E}})'$ 有 $(h', r, t') \notin$

$(S \cup S_{\mathcal{E}})$ 。

$$(S \cup S_{\mathcal{E}})'_{(h,r,t)} = \{(h', r, t \mid h' \in E')\} \cup \{(h, r, t') \mid t' \in E'\} \quad (4.2)$$

以下方法基于TransE提供的训练方法修改而来。

```

Input: Training set  $S = \{(h, l, t)\}$ , explanation set  $S_{\mathcal{E}}$  and rel. sets  $E \cup E'$ 
        and  $L$ , margin  $\gamma$ , embedding.  $k$ 
initialize  $e \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$  for each entity  $e \notin E$  and  $e \in E'$ ;
while loop times  $t \leq \text{epoch}$  do
     $e \leftarrow e / \|e\|$  for each entity  $e \in E \cup E'$ ;
     $(S \cup S_{\mathcal{E}})_{\text{batch}} \leftarrow \text{sample}(S, b) \cup \text{sample}(S_{\mathcal{E}}, c)$  //sample a minibatch of
    size  $b + c$ , where  $c \geq 1$ ;
     $T_{\text{batch}} \leftarrow \emptyset$ ;
    for  $(h, l, t) \in (S \cup S_{\mathcal{E}})_{\text{batch}}$  do
         $(h', l, t') \leftarrow \text{sample}((S \cup S_{\mathcal{E}})'_{(h,r,t)})$ ;
         $T_{\text{batch}} \leftarrow T_{\text{batch}} \cup \{((h, l, t), (h', l, t'))\}$ ;
    end
    Update embeddings with
        
$$\sum_{((h,l,t),(h',l,t')) \in T_{\text{batch}}} \nabla[\gamma + d(h + r, t) - d(h' + r, t')]_+$$

end
    
```

Algorithm 5: 嵌入更新

第5章 实验与分析

在上两章的理论中，我们提出了基于决断集模板进行溯因诊断的方法，因此在本章中我们进行了两组实验来对上述理论进行验证。在第一组实验中，我们主要观察基于决断集模板的解释的基数生成解释的数量的影响以及对系统运算时间的影响。在第二组实验中我们重点验证生成的解释对知识库模型的诊断修复效果。

5.1 实验工具与环境

在本文中，我们实现了算法2和3中的算法逻辑，并把程序命名成Pattern Based Abduction(PBA)。PBA是一个用Java写成的程序，其中我们使用了本体语言编辑库OWL API来进行本体的生成、编辑与持久化存储，本体语言的推理机我们使用的是Pellet API[36]，我们利用Pellet API完成的主要任务包括蕴含检测，蕴含寻找以及一致性检测。我们使用的数据库系统是Mysql，利用Mysql我们主要完成以下几个任务：第一是完成数据的存储。这里的数据主要包括本体中的ABox，训练集的数据和测试数据，实验完成后生成的解释集合也会存储在相同数据库的不同表中。使用Mysql完成的第二个主要任务是寻找差异化实例替换 θ 。通过把决断集模板编译成sql语句后，我们可以根据数据库中的存储的公理计算出相对应的差异化实例替换 θ 。本文实验中用到的实验数据如下表：

表 5.1: 实验工具

工具	描述
开发语言	Java
JDK	7u121-2.6.8-1ubuntu0.14.04.3
数据库	Mysql(version 5.7)
本体编辑库	OWL API(version 3.5.1)
本体推理机	Pellet API(version 2.3.1)

本次实验的数据，我们使用的两组知识库数据都来自FreeBase，分别是FB15K和FB40K。因为 Δ_p 的问题输入需要设置观察值，因此我们对这组数据进行了预处理。我们从FB15K的有效三元组中抽取了12330组作为 Δ_p 问题的观察值，同时也会从训练集中把这组数据移除，以避免影响实验结果。它们的具体数据如下表：

表 5.2: FreeBase数据集相关数据

Dataset	#Rel	#Ent	#Train	#Valid	#Observation
FB15K	1,345	14,951	483,142	50,000	59,071
FB40K	1,336	39,528	370,648	67,946	96,678

我们所有的实验运行在同一台机器上，机器的配置如下：

表 5.3: 实验环境

环境	描述
CPU	Dual-Core 2.60GHz
RAM	8GB
OS	Windows 7
Java Heap Space	8GB

5.2 程序框架

系统PBA采取了分模块设计，所有的模块如图4.1所示，其中数据解析模块负责对输入的数据集进行解析，并结合本体编辑推理模块完成对数据集的语义表达。本体编辑推理模块除了完成对数据集语义表达的任务以外，还需要完成本体编辑存储以及推理任务，其中推理模块需要在解释生成的过程中对生成的本体和解释进行一致性检测，并验证临时本体是否能够蕴含蕴含值(或观察值)。决断集模板编译模块主要完成寻找差异化实例替换 θ 的任务。寻找差异化实例替换 θ 的任务可以在内存中通过搜索的方式找到，并且计算速度较快，但是因为计算机资源有限，而数据的增大有可能导致计算任务的无法完成，所以我们会把在内存中完

成差异化实例替换 θ 的这种方式称为不可扩展的。在PBA系统中的决断集模板编译模块，我们采用把决断集模板编译成结构化查询语言来查找差异化实例替换 θ 的方法。利用数据库的持久化存储特性，使我们的系统达到可扩展的要求，虽然计算速度相比在内存中搜索的方法要慢，但是这种方法降低了数据集大小的要求，使我们可以完成使用所有收集到的数据进行测试。最后一个模块是解释生成模块，解释生成模块实现了上文中列出的解释生成算法，根据决断集模板，观察值和训练集生成合理解释集。

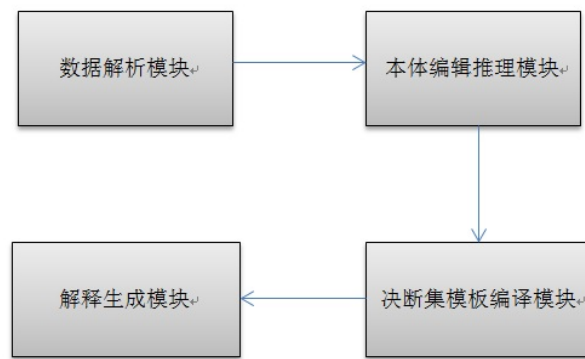


图 5.1: PBA 系统模块

5.3 解释基数对生成解释的效率以及数量对比实验

解释的生成是本文中的重要工作，因此生成的解释基数与生成解释的数量以及生成解释所需要的时间是评价解释生成工作的重要参考指标。

5.3.1 实验设计

本次实验我们收集了10个本体，这些本体的部分数据已经在表5.4中罗列出来。注意其中#C 是概念名的数量，#R 是二元关系的数量，#A 是本体中公理的个数，#E 本体推理出的原子公理的个数，DL的表达能力由Pellet检测。第一个本体是著名的LUBM(Guo, Pan, and Heflin 2005)本体。之后两个是UOBM(Ma et al. 2006)本体。他们扩展了LUBM，在LUBM的基础上他们添加了OWL Lite特性以及OWL DL的特性。之后四个本体是从http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library。这四个本体主要描述家庭关系，

表 5.4: 测试本体数据

TBox	DL Expressivity	#C	#R	#A	#E
LUBM	$\mathcal{AL}\mathcal{E}\mathcal{H}\mathcal{I}_+(\mathbf{D})$	43	32	93	76
UOBM-Lite	$\mathcal{AL}\mathcal{E}\mathcal{H}\mathcal{I}\mathcal{N}_+(\mathbf{D})$	52	43	145	86
UOBM-DL	$\mathcal{SHOIN}(\mathbf{D})$	69	44	206	113
generations	\mathcal{ALCOIF}	18	4	38	45
wine	$\mathcal{SHOIN}(\mathbf{D})$	77	14	657	322
philosurfical	\mathcal{AL}	377	313	1465	2981
not-galen	$\mathcal{AL}\mathcal{E}\mathcal{H}\mathcal{F}_+$	3097	413	5771	32475
physiology	$\mathcal{AL}\mathcal{E}\mathcal{I}$	2129	134	2256	2146
drugs	$\mathcal{AL}\mathcal{E}\mathcal{H}\mathcal{I}\mathcal{F}(\mathbf{D})$	4258	89	5042	17103
pathology	$\mathcal{AL}\mathcal{E}\mathcal{I}(\mathbf{D})$	7378	258	7159	14235

哲学领域，酒以及早期的GALEN模型的原型。最后的三个本体是大型的本体，它们主要描述医疗相关领域的知识，我们是从<http://www.opengalen.org/download/opengalen8-owl-sources.zip>收集的。

为了展示我们计算基于决断集模版解释的方法的实用性，我们进行以下实验。我们在实验中根据生成的观察值以及决断集模版，分别计算基数为1和基数为2的解释。每一个生成的蕴含公理我们验证了它无法被测试本体蕴含，并且不会产生不一致。我们生成的决断集模版都是从本体中具有最高 $support$ 值的，因此基于决断集模版生成的解释有更大的可能性被成功实例化。对于每一个基数设置为 k (其中 $k \in \{1, 2\}$)的实验，我们对实验设置了10000秒的限制，即使时间到了我们也可以观察到已完成计算的任务数据。

5.3.2 实验结果与数据分析

在表5.5中，JP-Comp展示了对于每个测试本体计算最多1000个决断集模版的时间，如果计算出的决断集模版少于1000，那么表中的数据就是实际计算出的数据。表中报告的数据包括加载本体的时间。

需要注意的是在表中: #P 计算出的决断集模版; T 是计算最多1000个决断集模版所需要的时间(*表示所有的决断集都被计算出来); #TP 是在10000内得到的所有 $support$ 值最高的决断集模版; T1 (T2) 是使用这些决断集模版生成解释过程中

表 5.5: 在本体中生成解释的相关数据

TBox	JP-Comp		JPBE1-Comp			JPBE2-Comp		
	#P	T(s)	#TP	T1(s)	#E	#TP	T2(s)	#E
LUBM	17	*1.0	17	3.9	253	17	178.8	18898
UOBM-Lite	23	*1.3	23	5.5	342	23	250.5	39870
UOBM-DL	44	*2.3	44	6.1	255	44	290.7	40659
generations	31	*0.9	31	2.7	182	–	–	–
wine	894	*86.6	100	66.7	0	100	188.9	10948
philosurfical	12	*7.6	12	89.7	1834	7	7111.4	724961
not-galen	1000	1340.9	100	72.0	1384	3	5637.8	144626
physiology	274	*14.8	100	108.7	477	98	9996.8	398521
drugs	289	*69.6	100	193.3	617	4	2749.2	49650
pathology	1000	122.8	100	112.6	935	7	4161.8	29759

消耗的时间; #E 是在完全处理完决断集模版后对于生成的50个观察值的解释的总数。

在“JPBE k -Comp”部分展示的是对于每一个测试本体，基于 $support$ 值最高的100个决断集模版对生成的50个观察值计算所有的基数为 k 的解释的运行数据。当 k 等于1的时候，在所有的10个测试本体中，100个高频决断集模版都被处理完成。

但是 $k = 1$ 的设定无法保证至少一个解释被生成，所以我们又做了第二轮试验，计算所有基数为2的解释。当 k 被设置为2的时候，在 $generations$ 的本体中我们无法生成解释。因为在 $generations$ 中所有的解释基数都为1。对于其它的本体，当 $k = 2$ 且限时10000秒内，它们中至少有一部分的决断集模版已经被处理完，也就意味着相应的解释全部计算完成。这个实验展示了我们基于决断集模版生成解释的方法具有实用性。同时实验结果也表明对计算出来的解释集应该需要进一步的处理。

5.4 决断集模板解释修复知识库模型对比实验

本实验主要考虑的问题是基于决断集模板诊断求解得出的解释是否能够对知

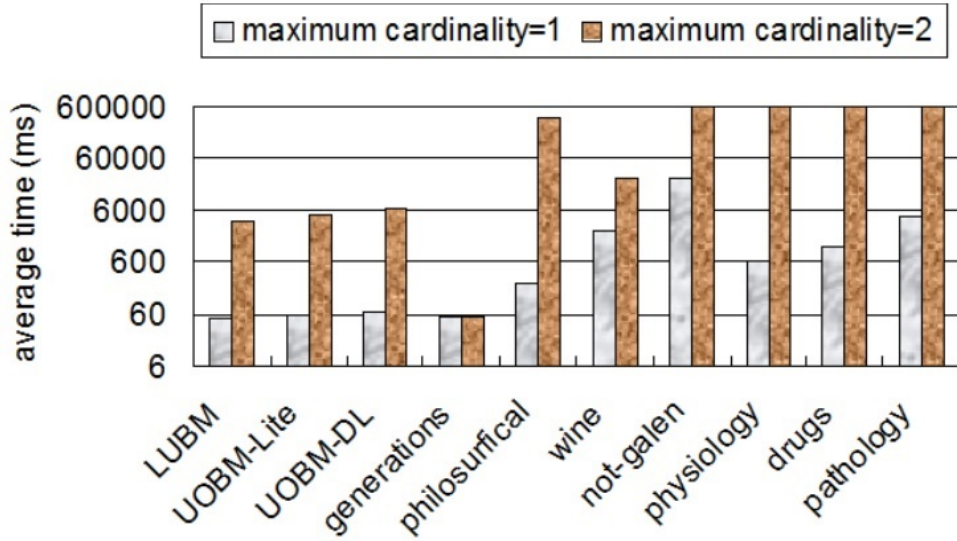


图 5.2: 对于一个观察值计算解释的平均耗时

知识库模型有一个较好的修复作用。知识库的训练集具有强的语义表达能力，如果对于一个观察值知识库模型无法正确地对它进行表达，我们尝试利用决断集模版对它进行修复，修复的方式就是利用求解得到的基于决断集模版的解释对训练集进行补充，对比诊断前后知识库模型对观察值的表达能力，我们可以验证解释集对知识库模型的修复能力。同时，由于基数的不同会对解释的修复能力产生影响，所以我们也对不同基数的解释的修复能力进行了比较。

5.4.1 实验设计

我们对诊断结果的评价方法由[21]中的方法调整而来。在利用现有的构造知识库模型的方法构造出向量模型之后，我们使用观察值集作为测试集来对模型进行测试。测试的时候我们对于测试集中的三元组(称为“有效三元组”)进行拆散操作，例如对于三元组 (h, r, t) ，我们首先移除三元组中的实体 h ，然后使用实体集合中的实体构造新的三元组，这些三元组我们称为“损坏三元组”，然后我们再把有效三元组和损坏三元组放在一起进行计算，分别计算这些三元组的能量函数，最后根据能量得分对它们进行由高到低排列。我们关注的指标有两个，一个是 $MeanRank$ ，表示测试集中有效三元组的平均排名，平均排名越高代表模型表

达该三元组的能力越强。另一个指标是 $hits@10$ ，它表示测试集中有效三元组进入排名前百分之10的比率，同 $MeanRank$ ， $hits@10$ 的比率也是越高越好。对比诊断前后的模型性能，我们就可以验证诊断结果对知识库模型的修复效果。

我们按照基数对实验中计算出的解释进行划分，直观上地，在其它条件不变的情况下，基数越低的解释意味着训练集的数据中与决断集模板的重合度越高，所以可以非常合理的推测基数越低的解释在语义上觉有更高的可信度。解释的生成条件虽然已经在定义中进行了限制，但是解释的数量仍然较大，因此我们对基数较低的解释赋予了更高的计算优先级，这样我们可以优先计算出语义可信度较高的解释。我们从有效三元组中抽取了12330个作为观察值，再根据观察值我们对训练数据进行了预处理，以使得训练集中不包含测试集中的数据。我们还使用了Du,Peng等提供的训练集本体，并利用决断集生成算法计算出的决断集。我们对处理后的数据进行转化，得到了决断集模版。最后我们使用训练集、决断集模版以及观察值作为PBA系统的输入进行了决断集模板解释修复知识库模型对比实验。

5.4.2 实验结果与数据分析

表 5.6: 诊断实验数据

Cardinity	TransE				TransR				TransH			
	$MeanRank$		$Hit@10$		$MeanRank$		$Hit@10$		$MeanRank$		$Hit@10$	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
No Exp	50.172	41.0	43.3	63.1	39.45	26.3	52.36	72.0	42.18	30.47	54.72	76.02
Exp.C = 1	27.37	16.39	54.78	82.8	22.13	15.98	63.7	81.4	32.03	24.77	58.13	86.02
Exp.C = 2	29.13	20.56	48.24	78.75	27.76	17.81	60.07	84.14	39.01	23.68	57.04	85.06
Exp.C = 3	46.0	40.76	41.33	68.46	25.0	19.88	57.12	81.35	38.88	31.3	55.0	80.42
Exp.C = 4	47.34	38.12	45.77	62.45	33.63	24.42	51.34	76.79	44.43	32.45	56.02	79.44

在本次实验中我们用来验证诊断效果的知识库模型包括以下：TransE(Bordes et al., 2013)，TransH (Wang et al., 2014)以及TransR (Lin et al., 2015)[37]。模型的实现代码来自https://github.com/mrlyk423/relation_extraction,

由Lin等人实现。在表5.6中记录了我们的实验数据，在表的左边列出了我们使用的知识库模型。在表的左侧 $No\ Exp$ ， $Exp.C = 1$ ， $Exp.C = 2$ ， $Exp.C = 3$ 以及 $Exp.C = 4$ ，表示我们根据生成的解释基数进行分类。根据解释的基数的不同我们共进行了五轮实验，其中实验 $No\ Exp$ 表示的是基准测试，在这个测试中我们使用三种知识库模型对生成的观察值集合进行链接预测任务。

在实验结果的第二行列出了上文提到的两种评价方法 $MeanRank$ ， $hits@10$ 。对于每一种评价方法，我们使用Lin等人的方法生成了两组测试数据分别是Raw测试组和Filter测试组[21]。Raw测试组的生成方式与Bordes等人及之前的生成方式一致，使用知识库个体字典中的随机个体替换测试三元组的头实体和尾实体。这种方式的不合理地方在于随机替换的个体有可能会生成训练集中的正向测例，这类测例通常会获得较高的正向得分，从而导致测试集中的数据排名下降。因此在Raw测试数据的基础上加上了过滤操作，剔除了所有训练集中包含的数据。另外值得注意的是，测试三元组的生成有替换头实体和替换尾实体两种方式，但是我们在生成解释时并没有对解释的头尾实体进行区分，因此最后的实验结果是对两种方式生成的测试三元组进行统一计算得出的。

从整体上看，诊断的结果除在极个别情况下，都使得训练出的模型产生了正向的效果，利用解释集重新训练出的模型，在对测试集的链接预测任务上取得了不同幅度上的提升，包括参与测试三个知识库模型TransE，TransH和TransR。纵向地看，在所有情况下，基数为1的解释集都取得了最好的结果，这也符合人们在语义上的推理习惯。一般情况下，基数为1的时候决断集模版与原知识库有着较高的匹配度，这也使得提出的假设有着更多的客观事实作为支撑。当基数继续的时候，解释集对知识库的修复能力开始下降，特别的，在经受测试的三种模型中，在解释集基数达到3和4时，重新构建的模型已经无法做到很好地对知识库模型进行修复。其中一个原因是当解释集基数增大时，知识库与决断集的重合度降低，从而降低了解释假设的合理性。横向的看，基数为2的时候，解释集对TransE模型的修复效果下降较快，但是对TransH和TransR仍然有着较好的修复效果，我们相信这与TransE无法很好地处理一对多、多对多关系的原因有关。

第6章 总结与展望

6.1 工作总结

词嵌入是由Bordes等人提出的近年来比较热门的一个研究方向[9]，词嵌入的目的是通过低纬度的稠密向量表征出知识库中的实体和关系的语义信息，在有效存储信息的基础上，能够实现实体预测、链接预测等推理任务。本文从知识库中难以被成功预测的三元组出发，遵循溯因推理的思想找出知识库模型无法成功预测的原因，并计算出用于修复数据集的解释。溯因推理是本体新知识获取的一个重要的推理方法[5]，最早的Colucci等人提出了基于表算法在 \mathcal{ALN} 语言上进行概念诊断的方法。Tu Anh T. Nguyen等人考虑了使用模版的概念寻找决断集。在Du等人2014的工作中利用提出的first-order rewritable TBoxes概念生成了数量可受限制的representative解释，实现对数据集的修复。另外还有一部分的研究集中在数据集的修复上，包括[5, 38, 39]。

我们在对知识库的诊断中，首先考虑了限制解释集的大小。解释集过于强大的表达能力会导致解空间的膨胀，而过度地压缩解的空间，不仅会极大地削弱解释集的表达能力，也会影响解释集的对缺漏公理的覆盖能力。为了平衡解释集的表达能力与解空间的矛盾，本文根据决断集合的特点提出决断集模版。利用决断集模版，我们有效地限制了解释集的空间大小。同时，利用 \subseteq_{ds} -minimal对解释的语义限制，我们进一步减小了解释的数量。在本体的溯因诊断中，观察值的表达是另外一个需要考虑的因素，但是在词向量模型的训练集的数据中，三元组是它的唯一表达方式，因此本文使用了OWL的可判定性片段DL *SROI*完成对观察值集合的表达，这样可以使得解释集的解空间缩小而不影响解释集的表达能力。

本文通过利用本体的表达能力对知识库进行表达，从而把优化词向量模型的问题转成本体研究中的溯因诊断问题。通过利用决断集模版的语义能力，我们从模版和集合中实例化出了目标解释集。然后，利用诊断出来的结果，我们对知识库进行修正，从而达到了优化词向量模型的目的。最后，我们还实现了一

个PBA的系统。在这个系统系下，我们的实验表明本文得出的解释规则具有良好的修复能力。

6.2 后续工作以及展望

本文的后续研究包括以下两个方向：

1. 解释排序

在本文提出的解释生成算法中，所有生成的解释并没有语义上的排序。实际上，生成的解释根据语义不同，背景知识的不同，对知识库的修复能力也不一样。因此，忽视解释合理性的差异有可能会影响对知识库的修复效果。在今后对解释语义相关性排序的研究中，解释的基数，解释的support以及解释所基于的决断集模版与背景知识的吻合程度都应该是重点需要考虑的因素。

2. 决断集的生成

决断集的生成依赖于本体的构建，目前在知识库上构建本体的手段还比较贫乏。Lin等人的工作中考虑了关系路径[21]，它们对多个关系组成的路径进行了建模。Yang等人利用学习出的向量模型挖掘出了逻辑规则，他们发现双线性实体的嵌入有利于对语义关系的捕获[40]。这些研究成果有助于在知识库上进行本体的自动构建，在计算机的介入下可以加速构建的速度，并相应地提高决断集模版的生成效率以及准确率。

参考文献

- [1] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, 2011.
- [2] Alexander Maedche and Steffen Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2):72–79, 2001.
- [3] Yankai Lin, Zhiyuan Liu, and Maosong Sun. Knowledge representation learning with entities, attributes and relations. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2866–2872, 2016.
- [4] Jianfeng Du, Guilin Qi, Yi-Dong Shen, and Jeff Z. Pan. Towards practical abox abduction in large OWL DL ontologies. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, 2011.
- [5] Júlia Pukancová and Martin Homola. Tableau-based abox abduction for description logics: Preliminary report. In *Proceedings of the 29th International Workshop on Description Logics, Cape Town, South Africa, April 22-25, 2016.*, 2016.
- [6] F. Wei-Kleiner, Z. Dragisic, and P. Lambrix. Abduction framework for repairing incomplete EL ontologies: Complexity results and algorithms. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1120–1127, 2014.

- [7] S. Staab and R. Studer, editors. *Handbook on Ontologies*. Springer-Verlag, 2009.
- [8] Szymon Klarman, Ulle Endriss, and Stefan Schlobach. Abox abduction in the description logic *ALC*. *J. Autom. Reasoning*, 46(1):43–80, 2011.
- [9] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 2787–2795, 2013.
- [10] P. Lambrix and Q. Liu. Debugging the missing is-a structure within taxonomies networked by partial reference alignments. *Data & Knowledge Engineering*, 86:179–205, 2013.
- [11] P. Lambrix, Z. Dragisic, and V. Ivanova. Get my pizza right: Repairing missing is-a relations in *ALC* ontologies. In *Proceedings of the 2nd Joint International Semantic Technology Conference (JIST)*, pages 17–32, 2012.
- [12] Jens Lehmann, Sören Auer, Lorenz Bühmann, and Sebastian Tramp. Class expression learning for ontology engineering. *J. Web Sem.*, 9(1):71–81, 2011.
- [13] Birte Glimm, Ian Horrocks, Boris Motik, Rob Shearer, and Giorgos Stoilos. A novel approach to ontology classification. *J. Web Sem.*, 14:84–101, 2012.
- [14] S. Colucci, T. Di Noia, E. Di Sciascio, F. M. Donini, and M. Mongiello. A uniform tableaux-based approach to concept abduction and contraction in *ALN*. In *Proceedings of the 17th International Workshop on Description Logics*, 2004.

- [15] S. Klarman, U. Endriss, and S. Schlobach. ABox abduction in the description logic *ALC*. *Journal of Automated Reasoning*, 46(1):43–80, 2011.
- [16] Ken Halland and Katarina Britz. Abox abduction in *ALC* using a DL tableau. In *2012 South African Institute of Computer Scientists and Information Technologists Conference, SAICSIT '12, Pretoria, South Africa, October 1-3, 2012*, pages 51–58, 2012.
- [17] Jianfeng Du, Hai Wan, and Huaguan Ma. Practical tbox abduction based on justification patterns. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 1100–1106, 2017.
- [18] Zhiyuan Liu, Maosong Sun, Yankai Lin, and Ruobing Xie. Knowledge representation learning: A review. *Journal of Computer Research and Development*, 53(2):247–261, 2016.
- [19] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From *SHIQ* and RDF to OWL: the making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.
- [20] L. Ma, Y. Yang, Z. Qiu, G. Xie, Y. Pan, and S. Liu. Towards a complete OWL ontology benchmark. In Y. Sure and J. Domingue, editors, *Proceedings of the 3rd European Semantic Web Conference (ESWC)*, pages 125–139, 2006.
- [21] Yankai Lin, Zhiyuan Liu, and Maosong Sun. Modeling relation paths for representation learning of knowledge bases. *CoRR*, abs/1506.00379, 2015.
- [22] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. Reasoning with neural tensor networks for knowledge base completion. In *27th Annual Conference on Neural Information Processing Systems 2013.*, pages 926–934, 2013.

- [23] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *CoRR*, abs/1412.6575, 2014.
- [24] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 1112–1119, 2014.
- [25] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
- [26] Guohui Xiao and Thomas Eiter. Inline evaluation of hybrid knowledge bases - phd description. In *Web Reasoning and Rule Systems - 5th International Conference, RR 2011, Galway, Ireland, August 29-30, 2011. Proceedings*, pages 300–305, 2011.
- [27] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the EL envelope. In *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, pages 364–369, 2005.
- [28] J. Du, G. Qi, and X. Fu. A practical fine-grained approach to resolving incoherent OWL 2 DL terminologies. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM)*, pages 919–928, 2014.
- [29] A. Kalyanpur, B. Parsia, M. Horridge, and E. Sirin. Finding all justifications of OWL DL entailments. In *Proceedings of the 6th International Semantic Web Conference (ISWC)*, pages 267–280, 2007.

- [30] Zhangquan Zhou, Guilin Qi, and Boontawee Suntisrivaraporn. A new method of finding all justifications in OWL 2 EL. In *2013 IEEE/WIC/ACM International Conferences on Web Intelligence, WI 2013, Atlanta, GA, USA, November 17-20, 2013*, pages 213–220, 2013.
- [31] B. Suntisrivaraporn, G. Qi, Q. Ji, and P. Haase. A modularization-based approach to finding all justifications for OWL DL entailments. In *Proceedings of the 3rd Asian Semantic Web Conference (ASWC)*, pages 1–15, 2008.
- [32] Jiang Yan. On the logical nature of pragmatic inference. In *Journal of Foreign Languages*, pages 18–29, 2002.
- [33] Qiu Ji, Zhiqiang Gao, Zhisheng Huang, and Man Zhu. An efficient approach to debugging ontologies based on patterns. In *The Semantic Web - Joint International Semantic Technology Conference, JIST 2011, Hangzhou, China, December 4-7, 2011. Proceedings*, pages 425–433, 2011.
- [34] K. M. Shchekotykhin and G. Friedrich. Query strategy for sequential ontology debugging. In *Proceedings of the 9th International Semantic Web Conference (ISWC)*, pages 696–712, 2010.
- [35] U. Junker. QUICKXPLAIN: Preferred explanations and relaxations for over-constrained problems. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI)*, pages 167–172, 2004.
- [36] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics*, 5(2):51–53, 2007.
- [37] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2181–2187, 2015.

-
- [38] Jianfeng Du, Kewen Wang, and Yi-Dong Shen. Towards tractable and practical abox abduction over inconsistent description logic ontologies. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 1489–1495, 2015.
- [39] J. Du, G. Qi, Y. Shen, and J. Z. Pan. Towards practical ABox abduction in large OWL DL ontologies. In *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI)*, pages 1160–1165, 2011.
- [40] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *CoRR*, abs/1412.6575, 2014.

在学期间论文发表情况

- [1]. XXX(作者名) xxx(论文名)// Proceedings of the 31th AAAI Conference on Artificial Intelligence (AAAI-17). 2017.

注：此论文的内容由本文第三、四章的工作扩展而来