

北京師範大學

# 硕士学位论文

论文题目：基于关系路径的知识库补全算法研究

作者：黄勇

导师：王志春 副教授

系别年级：信科2015级

学号：201521210022

学科专业：计算机软件与理论

完成日期：2018年 6月

北京师范大学研究生院

## 北京师范大学学位论文原创性声明

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。对本文的研究做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

## 学位论文使用授权书

学位论文作者完全了解北京师范大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京师范大学。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。保密的学位论文在解密后适用于本授权书。

本人签名： \_\_\_\_\_

日期： \_\_\_\_\_

导师签名： \_\_\_\_\_

日期： \_\_\_\_\_

# 基于关系路径的知识库补全算法研究

## 摘 要

知识库通常从非结构化、半结构化的数据中抽取事实，构建结构化的知识，常用三元组进行存储表示。知识库在信息检索、问答系统等领域具有重要作用，尽管一些知识库如Freebase、YAGO包含大量三元组，但这些知识库中的三元组是不完备的，很多实体对的关系存在缺失，预测知识库中实体对之间的关系变得十分重要。当前一种重要的知识库补全算法是基于符号逻辑的**路径排序算法**。路径排序算法抽取实体之间的**关系路径特征**，构建逻辑回归分类模型，对知识库中实体和实体之间关系进行预测。

本论文的研究在传统的**关系路径特征**基础上，提出了一种新的结合**关系路径特征**和**实体属性特征**的知识库补全模型，并将不同类型的**关系路径特征**、**头实体属性**、**尾实体属性特征**进行组合，显著提升了知识库补全效果。为了更好的进行特征工程和模型优化，在传统逻辑回归模型的基础上，本研究构建基于**学习排序模型**的知识库补全算法，通过训练学习排序模型的损失函数，优化正负实体对的排序，从而提高知识库补全模型效果。本论文的研究创新点包括以下三点：

- 抽取**关系路径特征**和**实体属性特征**作为知识库补全的特征。通过结合基于随机游走抽取**关系路径特征**、**头实体属性特征**、**尾实体属性特征**、**头尾比较实体属性特征**，将多种不同类型特征进行组合，构建逻辑回归预测模型，从特征工程方面极大拓展了知识库补全系统的维度，增强了知识库补全效果。
- 基于学习排序算法预测知识库补全中实体和实体的关系，通过优化排序模型损失函数，直接学习知识库补全中的实体对排序顺序，改进目标函数进行模型预测。将路径排序中的分类损失函数进行优化，改进为排序损失函数，改进排序的目标函数从而使得知识库补全预测效果更好。
- 构建基于逻辑回归、学习排序树模型方法的知识库补全模型，通过不同比例的正负候选实体对生成，深入研究了学习排序对知识库补全的影响。通过研究不同正负比例的实体对排序模型，结合树模型进行优化，显著提升了模型的预测效果，改进多对多关系模型预测排序结果。

**关键词：**知识库补全，路径排序算法，学习排序模型，关系路径，实体属性

# Knowledge Base Completion by relational path methods

## ABSTRACT

Knowledge base (KB) completion aims to predict new facts from the existing ones in KBs. There are many KB completion approaches, one of the state-of-art approaches is Path Ranking Algorithm (PRA), which predicts new facts based on relational path types connecting entities. PRA takes the relation prediction as a classification problem, and logistic regression or SVM is used as the classification model.

In this paper, our model use both literal facts and relational path types as feature matrix, which is much more comprehensive and complicated. We consider the relation prediction as a ranking problem, learning to rank model is trained on relations to predict new facts. Besides, we propose to extract literal features from literal facts, and incorporate them with path-based features extracted from relational facts; predictive model is then trained to infer new facts with assembly features and bring higher precision scores in classification metrics and ranking metrics. Experiments on YAGO show that our proposed approach outperforms approaches using relational path features and classification models. This paper has three main contribution, including:

- KB completion feature type is comprehensive including both relational path and literal features to generate feature matrix, and in different KBs our features help to generate higher scores both in ranking tasks and classification tasks.
- KB completion was considered as a ranking method, we use learning to ranking model to rank entity pairs rather than simple taking it as a classification model.
- KB completion methods are easily explainable, with combined literal and relational facts we can easily predict relation in KBs, tree models are used with assembly features, this helps us to rank entity pairs in different.

**KEY WORDS:** Knowledge Base Completion, Learning to Rank, Path Ranking Algorithm, Relational Path, Literal Facts

# 目 录

|                               |    |
|-------------------------------|----|
| 摘 要 .....                     | I  |
| ABSTRACT .....                | II |
| 第 1 章 绪论 .....                | 1  |
| 1.1 引言 .....                  | 1  |
| 1.2 知识库补全关键问题 .....           | 2  |
| 1.2.1 关系路径特征和实体属性特征 .....     | 3  |
| 1.2.2 基于分类的知识库补全模型 .....      | 4  |
| 1.3 论文研究工作 .....              | 5  |
| 1.3.1 结合关系路径和实体属性的知识库补全 ..... | 5  |
| 1.3.2 基于学习排序模型的知识库补全 .....    | 6  |
| 1.4 论文组织结构 .....              | 8  |
| 第 2 章 知识库补全研究现状 .....         | 9  |
| 2.1 知识库构及其应用 .....            | 10 |
| 2.2 基于表示学习的知识库补全 .....        | 12 |
| 2.3 基于关系路径的知识库补全 .....        | 14 |
| 2.4 知识库补全模型评价 .....           | 15 |
| 2.5 本章工作总结 .....              | 18 |
| 第 3 章 结合关系路径和实体属性的知识库补全 ..... | 19 |
| 3.1 问题引入 .....                | 19 |
| 3.2 提出模型 .....                | 19 |
| 3.2.1 模型框架 .....              | 21 |
| 3.2.2 关系路径特征计算 .....          | 22 |
| 3.2.2.1 关系路径类型集合 .....        | 23 |
| 3.2.2.2 关系路径特征向量 .....        | 23 |
| 3.2.3 实体属性特征计算 .....          | 23 |
| 3.3 实验结果 .....                | 24 |
| 3.3.1 YAGO知识库实验 .....         | 26 |
| 3.3.2 Freebase知识库实验 .....     | 29 |
| 3.4 本章工作总结 .....              | 32 |

|                            |    |
|----------------------------|----|
| 第 4 章 基于学习排序模型的知识库补全 ..... | 34 |
| 4.1 问题引入 .....             | 34 |
| 4.2 模型构建 .....             | 35 |
| 4.2.1 实体关系特征计算 .....       | 35 |
| 4.2.2 基于逻辑回归的补全模型 .....    | 36 |
| 4.2.3 基于学习排序的树模型 .....     | 37 |
| 4.3 实验结果 .....             | 38 |
| 4.4 本章工作总结 .....           | 42 |
| 第 5 章 总结和展望 .....          | 43 |
| 5.1 论文工作总结 .....           | 43 |
| 5.2 未来研究展望 .....           | 44 |
| 参考文献 .....                 | 46 |
| 学术成果 .....                 | 50 |
| 致谢 .....                   | 51 |

# 第 1 章 绪论

## 1.1 引言

近些年来，随着互联网的快速发展，人工智能和大数据技术的兴起和应用，知识、信息和数据大量增长，人们面对各种各样的信息、数据越来越多，而查找知识越来越困难。传统网页搜索的信息检索方式检索相关文档，越来越难于进行知识查找，而随着智能手机、移动互联网的兴起，人们对于信息检索、知识获取的要求也越来越高，面对各种数据信息，人们精确查找检索知识的需求越来越迫切，越来越期望能更好更快的检索了解学习知识。随之知识库构建、推理和应用变得流行起来，越来越多的大学、科研机构和商业公司开始构建大规模知识库，常见的知识库包括Google Knowledge Graph<sup>[1]</sup>，NELL<sup>[2]</sup>，YAGO<sup>[3]</sup>，Freebase<sup>[4]</sup>，DBpedia<sup>[5]</sup>等。这些知识库基于自动和半自动的信息抽取<sup>[6]</sup>、众包<sup>[7]</sup>、专家领域知识<sup>[8]</sup>等方式进行构建，把互联网、书籍、数据库等各处非结构化的文本中数据结构化、抽取构建知识库，获得了大量的实体、关系和属性信息，能为人们提供的方便快捷的信息查询、检索、知识学习途径。这类知识库的数据完整性、数据精确性和数据质量等衡量指标非常高，已经有很多基于知识库的系统被成功用于商业领域，如谷歌搜索引擎<sup>①</sup>，微软的必应搜索<sup>②</sup>等。除了这些通用领域的知识库之外，也有很多领域知识库被构建和应用，知识库也被用于生物学<sup>[9]</sup>、金融学、教育学等各个领域，一些较为成熟的问答系统、个人手机助手如苹果公司的Siri、谷歌公司的谷歌助手（Google Assistant）、微软公司的手机助手小冰、出门问问、亚马逊公司的Alexa<sup>③</sup>等也将知识库集成其中。

常见的知识库可以用有向的属性图进行可视化展示，图1是一个关于北京师范大学的知识库，描述了和北京师范大学相关的实体、属性和关系。图中展示了实体、实体和实体之间关系、实体属性类型、实体属性值等不同的信息，其中实体是描述现实中实际存在的事物、地点、人物、事件，如北京师范大学、北京可以看做是两个表示地点的实体，关系是描述实体和实体之间存在的某种联系和特征，北京师范大学位于北京，则北京师范大学和北京存在着位于的关系，这种关系是一种有方向的关系。除此之外，一个

① <https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html>

② <https://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/>

③ <https://developer.amazon.com/zh/alexa>

知识库中的三元组还可以分为关系型三元组和实体属性三元组。关系三元组典型的例子如：（北京师范大学，位于，北京），（北京，位于，中国），它描述了实体和实体之间的关系，其中“北京师范大学”和“北京”是实体，“位于”是关系。属性事实三元组如：（北京，人口，2150万），它描述实体具有的某种特征，其中“北京”是实体，“2150万”是描述实体属性的属性值，“人口”是北京这类实体具有的一种属性特征类型。通常实体关系三元组用来描述实体和实体之间的联系，实体属性三元组描述实体的属性事实特征，这些不同类型的三元组共同构成一个完整的知识库。但是尽管很多知识库中三元组数量很多，很多实体和实体之间的关系并未被发现，如（鲁迅，居住在，北京）、（董奇，居住在，北京）等三元组都未被发现。如何构建一个更加完整的知识库，发现知识库中潜藏的实体和实体之间的关系成为知识库研究的热点。

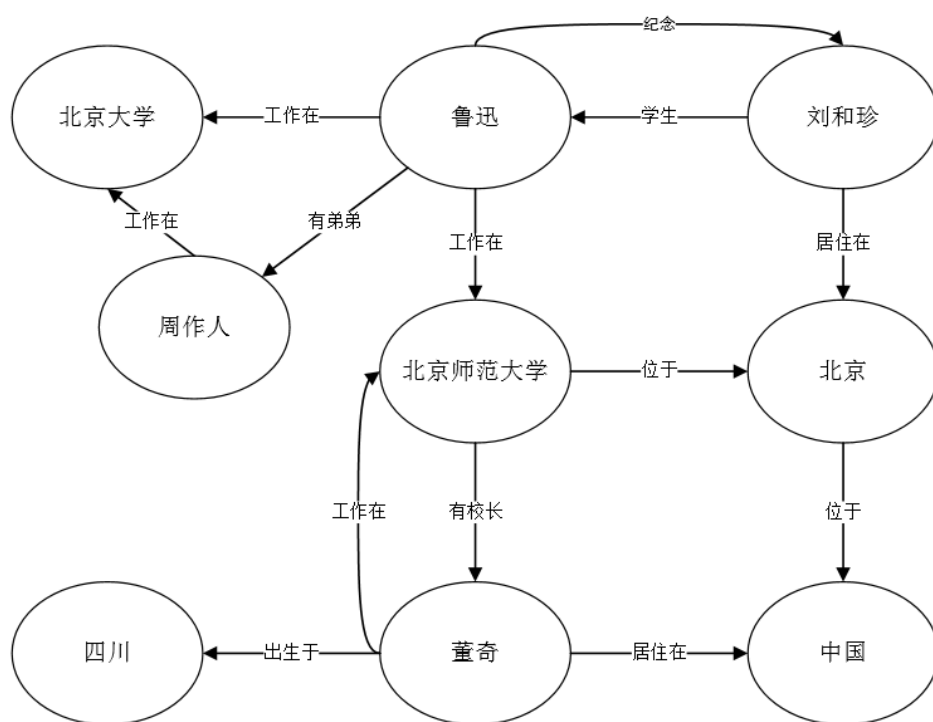


图1 “北京师范大学”相关的知识库实例

## 1.2 知识库补全关键问题

本节介绍知识库补全算法的主要研究问题，了解知识库补全算法相关的难点。尽管一些知识库的三元组数量巨大，但这些知识库仍然是不完备的，如很多人的出生地点并未包含在知识库中，一些演员是否出演过某些电影也是无法确定的，很多研究<sup>[10]</sup>表明知



识库中存在大量隐藏的关系未被发现。为了能发掘知识库中隐藏的实体和实体之间的关系，有很多知识库补全算法被提出。

### 1.2.1 关系路径特征和实体属性特征

表 1 YAGO和Freebase知识库部分关系类型

| YAGO关系类型            | Freebase关系类型   |
|---------------------|--|
| isCitizenOf         | /location/country/form_of_governme   |
| isAffiliatedTo      | /tv/tv/program/regular/cast./tv/regular/tv/appearance/acto                                 |
| wasBornIn           | /media_common/netflix_genre/titles   |
| playsFor            | /award/award_winner/awards_won./award/award_honor/award_wi                                 |
| isLocatedIn         | /soccer/football_team/current_roster./sports/sports_team_roster/position                   |
| influences          | /sports/sports_position/players./soccer/football_roster_position/t                         |
| hasWonPrize         | /film/film/starring./film/performance/acto   |
| dealsWith           | /soccer/football_team/current_roster./soccer/football_roster_position/posi                 |
| hasChild            | /film/actor/film./film/performance   |
| graduatedFrom       | /award/award_nominated_work/award_nominations./award/award_nomination/awar                 |
| isMarriedTo         | /award/award_category/nominees./award/award_nomination/nominated_f                         |
| worksAt             | /award/award_nominee/award_nominations./award/award_nomination/award_nomin                 |
| diedIn              | /olympics/olympic_sport/olympic_games_cont   |
| hasNeighbor         | /music/performance_role/regular_performances./music/group_membership/role                  |
| happenedIn          | /award/award_category/winners./award/award_honor/ceremony                                  |
| livesIn             | /film/film/release/date_s./film/film_regional_release_date/film_release_distribution_mediu |
| isPoliticianOf      | /people/marriage_union_type/unions_of_this_type./people/marriage/s                         |
| participatedIn      | /award/award_winning_work/awards_won./award/award_honor/award_winn                         |
| hasOfficialLanguage | /film/film/release_date_s./film/film_regional_release_date/film_release_re                 |
| owns                | /film/film/language  |
| .....               | .....  |
| actedIn             | /music/artist/genr   |

虽然许多知识库的规模很大，但这些知识库中仍然有很多实体和实体之间的关系未被发现。为了解决这个问题，很多知识库补全的方法被提出来，这些方法基于知识库中已有的关系三元组预测未知关系的三元组，通过知识库中的实体和实体之间的关系、图拓扑结构等信息构建模型，对可能具有某种关系的实体对进行模型预测。如果将现有的知识库数据看做是多种关系构成的图，图的顶点是实体，图的边是实体对之间的关系，知识库补全可以看成是图中关系的预测。表1展示了YAGO和Freebase知识库中部分关系

特征类型，和一些同构图中的链接预测模型<sup>[1]</sup>不同，知识库是一种由多种不同类型的关系组成的异构图，知识库补全需要处理多种不同关系类型的关系预测，而多数链接预测只需要预测一种单一关系，如预测某两个网站之间是否有链接关系。为了解决预测实体和实体之间关系的问题，已经出现了很多经典的知识库补全算法，这些算法可以分为两类：基于逻辑符号、关系路径的补全算法和基于表示学习的补全算法，这两种算法有时候也被称为基于图特征和基于隐藏特征的补全算法。

我们分析了常见的YAGO知识库和相关的知识库补全算法，发现共有四百多万实体关系三元组和三百多万属性事实的三元组。其中的实体关系三元组在以往基于关系路径、表示学习的知识库补全算法中被广泛使用，而属性事实数据尽管大量存在，却在知识库补全系统中并没有得到广泛应用。同时考虑到实体属性数据类型单位差别很大，难以进行统一有效的处理，将这些实体属性特征作为知识库补全的特征也十分困难。但可以预见在知识库关系预测中，实体属性特征起着重要的作用，如何将实体属性三元组有效用于知识库补全是本论文的关键点之一。

### 1.2.2 基于分类的知识库补全模型

在知识补全算法中，除了需要考虑不同类型的关系路径类型和实体属性类型外，如何构建合理有效的学习预测模型，预测知识库中实体和实体之间隐藏的关系，也是知识库补全算法的重要研究内容。通过分析知识库图的关系路径、实体属性信息可以发现，知识库中的图结构是稀疏的，每个存在的正例三元组实体对在训练模型中，可能生成上百组负例三元组实体对，如何解决正负实体对不匹配的问题很关键。在正负实体对比例悬殊时，关系预测中仅靠传统算法中的打分，比如一些逻辑符号推理中的逻辑回归算法是不够的，难以构建合理的学习预测模型，同事这种评价中并未考虑候选实体对的顺序对预测结果的影响，也不关注候选实体的排序信息，逻辑回归算法中基于熵的损失函数过于简单，在处理关系路径相关的特征时，不能有效的结合不同关系路径类型之间的组合关系，很难从多条关系路径类型中组合学习一些隐藏的关系路径特征。如何有效的结合不同类型特征，选择合适的目标函数，构建合理的模型，也是知识库补全算法中需要解决的重点。

通过研究知识库补全问题可以发现，对知识库补全是一种对正负三元组进行排序的过程，只有当正例三元组排序优于负例三元组的排序时，这样的知识库补全模型才是精确有效的，基于这种思考，本研究提出了基于学习排序的知识库补全算法，并通过基于树模型、深度神经网络的学习排序算法，研究如何设定知识库补全中目标函数和优化方

法。

在基于表示学习的知识库补全算法中，不同的模型选择不同的目标函数，通过神经网络、矩阵分解等技术，采用梯度下降的优化算法，学习不同实体、关系的低维度表示向量可以解决知识库预测问题。对于基于符号逻辑的知识库补全算法来说，常见的知识库补全算法如路径排序和子图特征抽取算法都是采用二分类模型进行知识库补全算法模型构建。如路径排序算法中，通常采用逻辑回归算法或者支持向量机回归，学习实体和实体之间具有某种关系的值。这种算法通常可以看做一种基于单文档排序（PointWise）的知识库补全算法，然而在一个知识库补全系统中，一组正负例实体对也可以当做一个整体进行优化，构建成对文档排序（PairWise）的知识库补全算法，进行模型优化，从而解决正负例实体对极其不平衡的问题。

### 1.3 论文研究工作

针对现有知识库补全技术不足，本研究将知识库中的关系路径特征和实体属性特征相结合，构建了一个更准确的知识库补全模型。首先，基于经典的路径排序模型抽取了关系路径特征；其次通过结合实体属性特征和关系路径特征，构建逻辑回归模型进行关系预测，从而进行知识库补全。除此之外，本研究提供一种基于学习排序算法的知识库补全技术，我们通过计算候选头实体和尾实体在关系预测中的位置排序，通过优化排序损失函数MAP来保证训练误差最小，从而获得最优的关系预测结果，并选择合适的模型评价指标来评估改进我们的预测结果。

#### 1.3.1 结合关系路径和实体属性的知识库补全

通过分析可以发现实体属性特征的复杂性较大。在表2中，我们展示了YAGO和Freebase知识库中部分实体的属性特征。以YAGO知识库为例，我们可以发现，常见的属性特征信息可以不仅仅有数量类型的如：hasNumberOfPeople、hasArea等属性类型，也有日期类型的特征信息如：wasCreatedOnDate、wasBornOnDate，也有比例型的实体属性特征如：hasInflation、hasUnemployment等。如何整合这些不同类型的特征信息，如何将属性特征和实体关系特征进行组合，都是本研究的重点和难点工作。

本部分研究主要介绍如何结合关系路径特征和实体属性特征进行知识库补全算法的构建，具体来说：（1）抽取关系路径类型特征，包括如何抽取关系路径类型构建特征集合，如何抽取计算每个关系下的的实体对的关系路径特征向量；（2）抽取实体属性特征，包括如何获取实体属性类型集合，如何抽取每个关系下的实体对的实体属性集合，

如何将实体对的实体属性集合进行标准化、归一化，如何结合不同类型的关系类型特征和实体属性特征，如何获取头实体实体属性特征、尾实体实体属性特征、实体对和实体对之间实体属性特征差值。

表 2 YAGO和Freebase知识库中的部分实体属性特征

| YAGO实体属性特征           | FB15K实体属性特征  |
|----------------------|--|
| hasNumberOfPeople    | /tv/tv_program/air_date_of_first_episode                         |
| hasArea              | /user/jg/default_domain/olympic_games/closing_date               |
| wasCreatedOnDate     | /user/jg/default_domain/olympic_games/opening_date               |
| hasLongitude         | /time/event/start_date   |
| hasPopulationDensity | /time/event/end_date   |
| hasLatitude          | /user/maxim75/default_domain/dbpedia_import/geocode_checked      |
| wasBornOnDate        | /tv/tv_program/air_date_of_final_episode                         |
| hasHeight            | /award/award_category/date_discontinued 16                       |
| wasDestroyedOnDate   | /user/ktrueman/default_domain/international_organization/founded |
| diedOnDate           | /base/usnris/nris_listing/significant_year                       |
| hasLength            | /sports/pro_athlete/career_start                                 |
| happenedOnDate       | /tennis/tennis_player/year_turned_pro                            |
| hasUnemployment      | /royalty/order_of_chivalry/date_founded 21                       |
| hasEconomicGrowth    | /business/defunct_company/ceased_operations                      |
| hasRevenue           | /government/legislative_session/date_began                       |
| hasGini              | /government/legislative_session/date_ended                       |
| hasExpenses          | /music/artist/active_start                                       |
| hasInflation         | /people/deceased_person/date_of_cremation                        |
| hasGDP               | /base/cdnpolitics/legislative_assembly/founded                   |
| hasImport            | /royalty/royal_line/ruled_to                                     |
| hasExport            | /royalty/royal_line/ruled_from                                   |
| hasPoverty           | /music/artist/active_end   |
| hasBudget            | /sports/pro_athlete/career_end                                   |
| hasWeight            | /base/lewisandclark/places_eastward/from                         |
| .....                | .....  |
| hasDuration          | /base/yalebase/secret_society/founded                            |

### 1.3.2 基于学习排序模型的知识库补全

本部分主要介绍如何构建预测模型学习知识库中的实体和实体之间的关系，通过借鉴表示学习中的目标函数选择，本部分研究了关系路径算法中基于学习排序中如何构建

损失函数。具体包括如何选择目标损失函数，如何构建优化算法，如何学习不同实体对的排列顺序。通过探索不同类型的目标函数、预测模型。本部分将知识库补全算法进行优化组合，期望能结合学习排序模型提高特征的组合能力和模型的泛化能力，优化正负实体对的排列顺序，使得知识库补全中，正例三元组排序结果高于负例三元组，而非简单的对知识库中的三元组进行打分。同时本部分研究借鉴了推荐系统中的思想，提出基于候选实体对生成-候选实体对排序两步进行知识库的思路。

本部分的研究的工作重点有两点：（1）提出了基于排序的知识库补全算法，将知识库补全算法从优化二分类模型改进为优化一组相关实体对排序的顺序问题，通过构建成对排序模型，进一步提升模型的预测能力。（2）研究了不同正负比例的实体对之间进行排序的效果，探索了如何基于匹配排序的思想构建知识库补全模型，通过表示学习构建知识库补全候选实体对，再构建基于关系路径的精确模型进行知识库补全构建。尽管基于学习排序的知识库补全算法选择很多，我们使用了最有效的树模型进行实体对排序，有效提高了模型的预测效果。

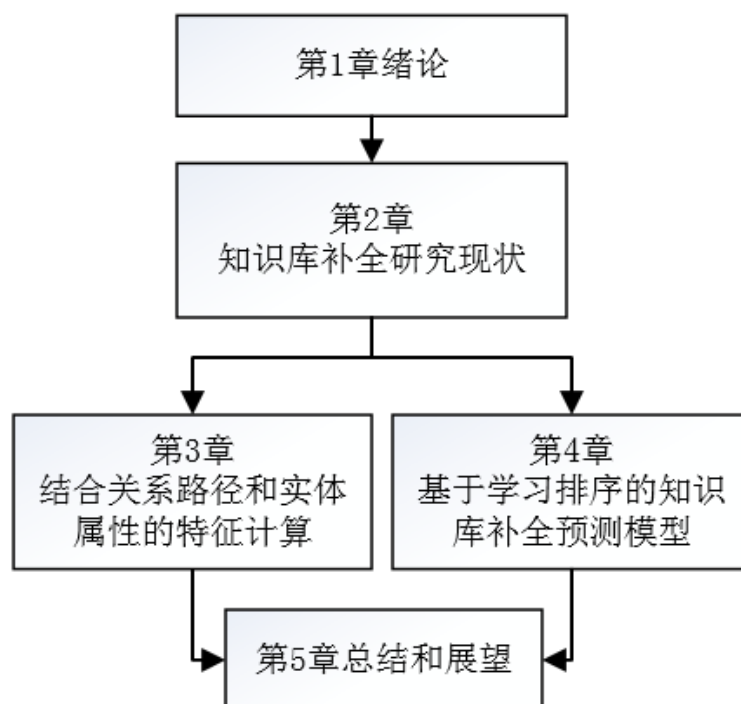


图2 论文组织结构图

## 1.4 论文组织结构

本论文的组织结构如图2所示。本研究首先提出知识库补全领域相关研究问题，介绍了知识库技术国内外研究现状，包括基于表示学习的知识库补全算法和基于符号逻辑的知识库补全算法。接着介绍本人的基于关系路径的知识库补全研究工作，包括结合实体属性的知识库补全算法和基于学习排序的知识库补全模型技术。最后对本人研究的相关工作和后续研究方向进行了总结和展望。

**第2章**介绍当前知识库技术的国内外研究现状。主要包括知识库构建、知识库应用、知识库补全和推理相关研究工作，并介绍了知识库补全工作中主要的两类算法：基于表示学习的知识库补全和基于关系路径的知识库补全。并介绍了知识库补全的评价函数和知识库补全的一些基本假设。

**第3章**介绍了基于关系路径的知识库补全算法研究框架，包括如何抽取计算关系路径类型和实体属性类型，构建预测模型的特征矩阵，并建立逻辑回归进行知识库补全算法模型预测，通过在YAGO和Freebase知识库中的实验，研究了如何组合不同类型特征向量进行模型预测。

**第4章**介绍了如何构建学习排序的算法进行知识库补全模型预测，包括如何选择目标函数，如何进行特征优化，选择合适的正负比例的实体对进行关系预测训练。通过在YAGO知识库中抽取不同比例的正负例实体对进行模型训练，获得不同的模型预测结果，从而阐释基于学习排序的知识库补全优势。

**第5章**总结了本研究工作的关键创新点，分析了本研究中基于关系路径的知识库补全算法研究的一些不足，并提出了未来研究的一些思路 and 方向。

## 第2章 知识库补全研究现状

当前知识库补全相关的研究主要分为三个方面：（1）知识库的构建，各种基于互联网服务知识库构建、各个领域内的知识库构建、通用的知识库构建。（2）知识库中关系机器学习，基于已经构建好的知识库预测实体和实体之间的关系，进行知识库补全推理等。（3）知识库系统应用。如基于知识库的问答系统，基于知识库的信息检索系统等，以及在各个领域内如生物、金融、教育等领域的本体知识库应用等。其中研究的热点之一就是知识库补全，即如何发掘知识库中实体和实体之间隐藏的关系。

现存的知识库系统如NELL、Knowledge Vault等，尽管已经包含了大量的实体、关系和属性，但这些知识库中很多实体和实体之间的关系仍然是缺失的。例如一个简单的知识库中存在一些三元组如：（北京师范大学，位于，北京）、（北京，位于，中国），但是当预测北师大是否位于中国时，由于系统中并未存在（北京师范大学，位于，中国）这样直接的三元组，通常计算机很难直接进行这类常识性的推理。如何构建一个辅助计算机进行常识推理的系统，发现知识库中隐藏的实体和实体之间的关系，是知识库补全系统的重要任务之一。当前有很多基于表示学习和基于符号逻辑的知识库补全算法被提出，这类算法都是基于知识库中现有的关系类型去预测知识库中未知的关系。但是在知识库中存在大量的属性事实型三元组未被使用，而这些属性事实三元组在进行关系预测时有很重要的作用，如对于已知的关系型三元组（A，isMarriedTo，B）、（A，HasChild，C），当预测B是否是C的父母时，这些相关实体的年龄、性别等实体属性也很重要，如何能处理不同类型的实体属性特征，将其用于知识库补全系统中，是知识库补全工作的一个重要探索点。

在构建知识库补全系统的过程中，基于符号逻辑的算法通常通过训练一个二分类的分类器来判断实体和实体之间是否具有某种关系。通常这个分类器的正例是通过抽取现有知识库中存在的三元组作为正例，或称之为正例三元组，而负例则是采用随机采样等方法生成负例三元组。通常在知识库补全系统中，每个正例三元组可以构建上百组不同种类的负例三元组，如何解决正负例三元组不均衡的问题，正确预测实体和实体之间的关系，也是知识库补全算法的一个重要优化方向。

## 2.1 知识库构及其应用

**知识库构建** 随着信息技术的发展和应用,过去几十年中,人们采用不同的方法构建知识库,将人们的经验、常识和现实世界中的事实知识用知识库的方法进行表示存储。通常知识库可以由三元组组成。知识库的构建可以有多种方法:(1)通过机器学习和自然语言处理技术自动抽取三元组<sup>[12]</sup>进行构建。(2)半自动抽取的方法,通过从维基百科等网站的infobox基于规则模式抽取三元组。(3)基于协同创作<sup>[13]</sup>的方法,通过众包的形式系统创建知识库。(4)基于专家知识构建的领域内知识库,这些领域知识库有较高的专业性。数据完整性、数据精确性和数据质量是考量这些构建知识库质量的重要指标,也是衡量知识库好坏的重要标准。

**YAGO知识库** YAGO<sup>[14]</sup>是一个从维基百科上抽取的、包含地理名词、WordNet<sup>[15]</sup>等数据的知识库。YAGO将WordNet的词汇定义与维基百科的分类体系进行了融合集成<sup>[16]</sup>,并集成了多种地理类型的数据,使得YAGO具有更加丰富的实体分类体系。YAGO还考虑了时间和空间知识<sup>[17]</sup>,为很多知识条目增加了时间和空间维度的属性描述。目前,YAGO包含超过1.2亿条结合关系三元组和实体属性三元组的知识。YAGO也是IBM Watson<sup>[18]</sup>的后端知识库之一。YAGO2<sup>[19]</sup>是YAGO的一个实例,当前YAGO2包括超过260万的实体和超过1.2亿的关系三元组知识,我们使用了其中实体的关系型三元组和属性型三元组共有4,484,914条、37种关系型三元组的事实描述,同时有3,353,659条、35种属性性三元组的事实描述。最新的YAGO3知识库<sup>[20]</sup>进一步拓展了YAGO知识库,集成了超过10种语言中的维基百科知识,属性值的准确率超过95%。

**Freebase知识库** Freebase<sup>[4]</sup>是一个协同创作的知识库系统,内容通过用户添加,所有的条目采用结构化的形式,将结构分为三次:领域-类型-主题,其中,每一个条目叫做一个主题,每个主题包含不同的属性类型,一些同类型的主题组成一个类型,所有相关的类型构成一个领域。这种通过协同创作方式创建了结构化的人类知识,截止2007年Freebase包含1.25亿条三元组,超过4000种类型和超过7000种属性知识。一些基于Freebase研究结构化知识的问答系统<sup>[21][22]</sup>,关注Freebase中的信息抽取和语义解析<sup>[23]</sup>,也有一些研究关注Freebase中的实体消除歧义问题<sup>[24]</sup>。

表3展示了一些不同类型的知识库的实体、关系类型和三元组等数量,其中的M表示百万数量。除了YAGO和Freebase知识库,也有很多重要的知识库如:Google Knowledge Graph, Wikidata, DBpedia等,这些不同类型的知识库都有着各自的特色和优势。通过分析表中的不同实体关系三元组、关系属性三元组分析可以发现,知识库中的实体数量、关系类型数量、以及三元组数量都很多。本研究中的实验考虑到同时使用属性三元组和



实体关系三元组，从而选择在YAGO和Freebase数据集中筛选属性特征较多的实体，构建两个子数据集进行模型训练。

表 3 不同知识库数据统计

| 知识库             | 实体数量 | 关系类型数量 | 三元组数量  |
|-----------------|------|--------|--------|
| Knowledge Graph | 570M | 35000  | 18000M |
| YAGO2           | 9.8M | 114    | 447M   |
| DBpedia         | 4.6M | 1367   | 538M   |
| Wikidata        | 18M  | 1632   | 66M    |
| Freebase        | 40M  | 35000  | 637M   |

**知识库应用** 在通过自动、半自动的信息抽取、专家知识构建知识库后，知识库库系统在信息检索、广告、推荐和金融领域等有着各种广泛的应用，知识库不仅能辅助人们检索、学习各种信息和知识，而且一些专家的领域内的知识有序化，规范化，能更好帮助人们去了解学习一个领域的专业知识。

知识库最广泛的应用是谷歌和微软等商业公司的信息检索系统，2014年谷歌提出的知识图谱<sup>[25]</sup>就是结合FreeBase、Wikidata以及各种互联网数据进行信息抽取、知识整合后的著名系统，谷歌也在其博客上讲述了如何将知识库应用在实际的检索中，微软的Satori也是一个类似的知识库系统,被必应搜索广泛应用。其他搜索公司如百度<sup>①</sup>、搜狗<sup>②</sup>等公司都在知识库构建和信息检索领域有各自的应用。对于每一个用户搜索的关键词，我们可以通过知识库系统来返回更丰富，更全面的信息。比如搜索一个企业法人的姓名，我们的智能搜索引擎可以返回与这个人相关的所有历史借款记录、联系人信息、行为特征等。另外我们通过可视化把复杂的信息以非常直观的方式呈现出来，使得用户对于对隐藏信息的来龙去脉一目了然。

基于知识库的问答系统也有广泛的发展，2000年就有人提出基于知识的问答系统<sup>[26]</sup>，通过构架知识库完成问答。也有很多方法基于知识库进行表示学习<sup>[27]</sup>，通过学习实体的低维度表示，学习知识库的实体和实体之间的关系，然后构建相关的知识库问答系统。还有一些基于标签语义解析的方法，构建知识库问答系统<sup>[28]</sup>。一些论文<sup>[29]</sup>研究结合深度学习、关系抽取和问答系统，从而获得更好的知识库问答效果。一些论文<sup>[30]</sup>研究基于开放的知识库构建开放领域的问答系统，期望能基于百科知识进行问答系统构建。也有基于知识库的系统通过结合智能计算<sup>[31]</sup>，在医疗诊断领域获得一些突破。也有一些

① <https://kgopen.baidu.com/index>

② <https://www.sogou.com/tupu/person.html>

研究结合了知识库和医疗图像信息<sup>[32]</sup>，对一些医疗疾病进行诊断治疗，期望能获得更好的治疗效果。

知识库在金融、教育等领域也可以有各种有效的应用。在进行金融风险检测时，我们可以使用不一致性验证可以用来判断一个借款人的欺诈风险，发现交易欺诈。比如借款人A和借款人B在同一地点、同一公司工作，但A和B的公司联系方式完全不一样，出现逻辑不通之处，则我们可以通过知识库补全推理发现这种交易欺诈。在金融领域除了贷前的风险控制，知识库也可以帮助我们在贷后进行催收。比如在贷后失联客户管理的问题上，知识库可以帮助我们挖掘出更多潜在的新的联系人，找到很多潜藏的知识库信息，发掘潜藏的关系等。

除了金融领域的知识库应用，在互联网和广告系统中，知识库也可以帮助我们分析用户和理解用户。通过结合多种数据源去分析实体之间的关系，发现用户的行为特征，更好的理解用户意图。比如一利用知识库来分析用户之间的关系，去发现一个组织的共同喜好，对于有相同特征的用户群体，制定合理有效的营销策略。通过对用户关系的分析，知识库帮助我们更好的、更深入的理解用户的需求，提升广告、搜索<sup>[33]</sup>、推荐<sup>[34]</sup>等相关领域的效果。

## 2.2 基于表示学习的知识库补全

当前知识库补全方法主要有两种方法：基于表示学习的知识库补全和基于符号逻辑的知识库补全。基于表示学习的方法是通过学习实体和关系的低维度向量表示，用向量的相似度计算预测实体之间的关系。常见的表示学习方法包括基于神经网络翻译的预测算法如TransE<sup>[35]</sup>、TransH<sup>[36]</sup>、TransR<sup>[37]</sup>等，也有基于矩阵张量分解、协同过滤的表示学习方法如RESCAL<sup>[38]</sup>等。

基于学习表示的知识库补全技术是近年来的研究热点之一，近年来获得了极大的关注热度，这类知识库补全技术通过不同的目标函数，希望能学习到知识库中实体和关系的低维度向量表示。这些实体的向量维度通常是50-300维度之间，将高维度稀疏的图数据张量，学习获得低维度的连续向量表示。获得实体和关系的向量表示后，可以通过向量之间余弦距离计算实体-关系-实体相似度。目前这类表示学习算法包括两类，基于神经网络翻译模型的表示学习和基于协同过滤的表示学习算法。

**协同过滤学习模型** 基于协同学习的知识库补全算法借鉴了推荐系统中的协同过滤模型，将基于用户-商品的协同矩阵分解技术进行拓展，构建了基于实体-关系-实体的三维张量模型，进行矩阵分解和实体关系学习预测。**RESCAL**是一种基于表示学习的潜藏特

征模型，基于矩阵分解、协同过滤进行知识库补全，RESCAL通过构建三维张量分解技术进行知识库补全，将实体-关系-实体构建三维的张量矩阵，提供了一种有效的表示学习方法。和其他表示学习模型不同，RESCAL这类模型更多借鉴推荐系统中张量分解算法，通过偏最小二乘算法，随机梯度下降优化算法，训练学习实体和关系的向量表示，学习实体和实体之间的相似度，进行对实体和实体之间未知的关系进行预测。详细来说，RESCAL基于R阶的分解模型， $\chi_k$  (2-1)被分解。

$$\chi_k \approx AR_k A^T, k = 1 \dots m \quad (2-1)$$

其中A表示一个 $n \times r$ 的矩阵，表示n个实体和r种关系组成矩阵， $R_k$ 是一个 $r \times r$ 的矩阵。则只基于随机梯度下降模型，最小化函数： $\min_{A, R_k} f(A, R_k) + g(A, R_k)$  其中 $f(A, R_k)$ 是度量 $\chi_k$ 和 $AR_k A^T$ 距离的函数。 $g(A, R_k)$ 是 $A, R_k$ 的复杂度惩罚项。通过梯度下降等学习算法可以计算获得模型的参数。

其他类似基于协同过滤模型表示学习进行知识库补全的方法还有张量分解机算法<sup>[39]</sup>，基于隐藏变量的张量分解机算法<sup>[40]</sup>，结构化的低维表示<sup>[41]</sup>、无结构化的低维表示等。基于张量分解模型和推荐系统等领域的协同过滤算法相似，都是从矩阵补全角度协同过滤学习向量模型。不同的是在知识库中，矩阵其实是由实体-关系-实体组成的三维张量，而推荐系统中的用户-商品矩阵是二维向量。一些实验<sup>[42]</sup>表明在稀疏的知识库图中，使用张量分解模型有较好的预测效果。

**神经网络翻译模型** 基于神经网络翻译模型的知识库补全算法近年来获得了极大的关注，很多基于神经网络翻译模型的知识库补全算法被提出如TransH、TransR等。TransE是一种典型的基于神经网络翻译模型的机器学习模型，研究考虑如何将知识库中的多种不同的关系、实体，学习获得它们的低维向量表示，期望能获得 $h + r \approx t$ 效果，其中h和t分别是头实体和尾实体学习到的低维向量，r是头实体和尾实体之间的关系向量表示。TransE模型定义了间隔损失函数L (2-2):

$$L = \sum_{(h,r,t) \in S} \sum_{(h',r',t') \in S'} [\gamma + d(h+r, t) - d(h'+r', t')]_+ \quad (2-2)$$

其中 $\gamma$ 是间隔超参数，s是真实存在的知识库， $(h, r, t)$ 是这个知识库中的三元组， $s'$ 是负例三元组组成的知识库实体， $(h', r, t')$ 表示负例三元组。 $d(h+r, t)$ 表示头实体加上关系向量和尾实体之间的欧氏距离。通过定义间隔损失函数，TransE期望学习的模型能保证正实例对比负实例对距离小，这样就能保证正实例对比负实例对余弦相似度更高。

其他的算法如TransH和TransR都是基于TransE模型的基础上，通过学习不同类型、不同目标的损失函数来获得低维度向量表示。TransH考虑一些关系中的映射属性知识，如

一对一、一对多、多对多关系等，训练学习这些关系的低维度向量表示。TransR<sup>[43]</sup>考虑实体和关系的多方面属性，在分割独立的向量空间中学习实体、关系的低维度向量表示。ProjE<sup>[44]</sup>基于神经网络的实体关系投影表示学习。这些不同类型的基于翻译模型的神经网络表示学习，能借助神经网络的优势，在CPU、GPU、TPU等不同的底层硬件中进行模型训练，从而能进行大规模的知识库表示学习和预测补全。

## 2.3 基于关系路径的知识库补全

基于关系路径的知识库补全算法也是多年来知识库补全、知识库推理中的关注热点之一。从90年代昆兰等人提出的规则归纳，将观察集数据中的知识以规则的形式提炼出来，到近年来热门的基于**路径排序算法（PRA）**和**子图特征抽取（SFE）**的知识库推理补全。这类进行知识库补全的算法多是利用符号逻辑，精确使用知识库中的关系路径特征，统计发现知识库图数据结构中存在的规则或规律，从而进行知识库补全预测。基于符号逻辑的方法主要包括AMIE<sup>[45]</sup>、PRA<sup>[46]</sup>和SFE<sup>[47]</sup>等；其中，AMIE方法通过从知识库中挖掘关联规则进行知识库补全，通过发掘知识库中频繁出现的关系路径，从而发现知识库中实体和实体之间隐藏的关系。路径排序算法PRA抽取连接实体和实体之间的关系路径，通过关系路径作为特征向量来预测实体之间的关系，子图特征抽取SFE在PRA框架的基础上，抽取更多隐藏的关系路径特征进行模型预测。

**规则挖掘算法** 规则挖掘是基于训练集知识库中的数据，通过挖掘知识库中隐藏的关联规则，进行知识发现和知识库补全。早期昆兰等人提出了一阶逻辑推理的算法FOIL<sup>[48]</sup>，通过学习知识库中的例子来构建霍顿子句。如公式 (2-3)就是一个典型的一阶逻辑推理。实体A是实体C的母亲，实体A和实体B结婚，则可以推断实体B是实体C的父亲，尽管这种推理并不是百分之百准确。

$$MotherOf(A, C) \wedge MarriedTo(A, B) \Rightarrow FatherOf(B, C) \quad (2-3)$$

FOCL<sup>[49]</sup>等算法也是基于一阶逻辑推理构建的知识库搜索算法。近年来随着互联网的快速发展，Schoenmackers<sup>[50]</sup>等人也将一阶逻辑推理用于互联网文本中。其他规则挖掘的典型算法包括AMIE<sup>[51]</sup>，AMIE受关联规则启发，基于开放世界假设，在一阶逻辑推理的基础上，能更快更高效的处理大规模的知识库数据，作者此后对AMIE算法进行改进获得AMIE+<sup>[52]</sup>，这些算法在 YAGO知识库中获得了很好的效果。RDF2Rules<sup>[53]</sup>等算法也是基于逻辑推理，构建频繁的谓词圈，获得了更加高效和准确的逻辑推理算法。

**路径排序算法** 路径排序算法<sup>[46]</sup>是2010年，劳逆等人提出的知识库补全算法。在传统

的一阶逻辑推理的基础上，路径排序算法基于随机游走，在由知识库构成的图数据中查找更多、更长的有效的路径进行知识库推理，和规则学习方法不同，路径排序算法在抽取图中的关系路径后，构建了分类学习器，通过分类器学习不同关系路径的权重。利用每种关系下三元组的不同关系路径权重，来进行新关系事实的预测，从而构建了新的知识库补全算法。此外，作者还构建了基于反向的随机游走图搜索算法<sup>[54]</sup>，进一步提升路径排序算法在知识库补全中的效果。考虑给定一个知识库关系 $r_i$ ，我们抽取所有具有这种关系的实体对构成集合 $R_i$  (2-4)

$$R_i = \{(h_{ij}, t_{ij}), (h_{ij}, r_i, t_{ij}) \in KB\} \quad (2-4)$$

随机游走查找路径后，我们抽取所有连接实体对 $(h_{ij}, t_{ij})$ 的路径类型 $p_i$  (2-5)构成分类特征：

$$P_i = \{p_i | (h_{ij}, p_i, t_{ij}) \in KB\} \quad (2-5)$$

从而我们可以选择合适的分类器模型进行关系路径的训练学习，利用合适的路径类型来进行关系预测，知识库补全。

子图特征抽取<sup>[47]</sup>提供了一种更加简便有效的关系路径计算方法。相比于路径排序算法，子图特征抽取能获得更多的潜藏路径，在进行关系预测的时候能获得更加显著有效的提升。此外，也有很多算法结合了子图特征抽取和表示学习算法<sup>[55]</sup>进行知识库补全。Coupled PRA<sup>[56]</sup>算法对相似的关系进行聚类，对于处理包括成百上千条关系的知识库，需要将相互影响的关系进行聚合，构建多任务的路径排序模型。

## 2.4 知识库补全模型评价

知识库补全的正例实体对可以选择知识库中已经存在的实体对，而构建负例实体对需要一些规则来生成负例三元组。此外，知识库补全模型一方面可以看做是正负例实体对分类的过程，从而可以选择分类器的模型评价函数，另一方面，知识库补全也可以看做对正负例实体对进行排序，从而可以选择排序模型评价函数。以往的知识库补全算法通常只选择分类评估指标和排序评估指标中的一类，本研究选择两种类型的评估指标进行模型评估。

**知识库补全假设** 知识库补全系统中，通常需要假定知识库中三元组正负实例对的正确性，或者在何种限制模式生成三元组负例。常用的有三种假设：（1）开放世界假设，（2）封闭世界假设，（3）局部封闭世界假设。我们给定一个三元组集合 $D^+$  (2-6)表示

正例:

$$D^+ = \{(h, r, t) | (h, t) \in KB\} \quad (2-6)$$

对于开放世界假说, 给定一个知识库中的三元组集合, 我们认为这些实际存在三元组是正例三元组, 对于在知识库中不存在的三元组, 开放世界假说认为这个三元组是不确定的, 可以通过概率大小预测这个三元组的正确性。

对于封闭世界假说, 给定一个知识库三元组集合, 我们认为只有知识库存在的三元组才是正例三元组。对于知识库中不存在的三元组, 封闭世界假说认为这些三元组是负例三元组。但通常这样会产生正负实例极其不平衡情况, 每个知识库中存在三元组都能生成数百例负例三元组。这些负例三元组组成负例集合 $D^-$  (4-1):

$$\begin{aligned} D^- = & \{(h_j, r, t_i) | (h_i \neq h_j \wedge (h_i, t_i) \in KB)\} \\ & \cup \{(h_i, r, t_j) | (t_j \neq t_i \wedge (h_i, t_i) \in KB)\} \end{aligned} \quad (2-7)$$

局部封闭世界假说对封闭世界假说进行了改进。给定一个关系, 知识库中这个关系的三元组记为正例三元组, 随机在这个关系中替换头尾实体, 生成这个关系下实体集合中错误的实体对, 这样就可以大大减少所有负例三元组个数。基于局部封闭世界假设构建的负例三元组集合 $D^-$  (2-8)可以用如下公式表示:

$$\begin{aligned} D^- = & \{(h_{ik}, r_i, t_{ij}) | (h_{ik} \neq h_{ij} \wedge (h_{ik}, t_{ik}) \in KB \wedge (h_{ij}, t_{ij}) \in KB)\} \\ & \cup \{(h_{ij}, r_i, t_{ik}) | (t_{ij} \neq t_{ik} \wedge (h_{ik}, t_{ik}) \in KB \wedge (h_{ij}, t_{ij}) \in KB)\} \end{aligned} \quad (2-8)$$

**模型评价方式** 知识库补全效果的评价方法借鉴了信息检索和分类器模型等研究中的模型评价方式, 主要采用MAP和MRR<sup>[55]</sup>指标进行模型评价。考虑到在一个给定关系下, 给定一个头实体和一系列的候选尾实体, 我们希望排在前面的实体对是正确的实体对, 如果正例实体对排在负例实体对后面, 则这个模型的预测效果有待改进。所以我们借鉴了信息检索领域相关的评价方法, 除了采用关键的MAP评价指标, 我们也采用了一些常见的AUC和Hit@1评价指标, 希望能详尽的分析模型的有效性和可预测性。本部分介绍了四类评价知识库补全的模型评估指标, 包括: (1) hit@1, 描述排序在第一位的正例三元组比例, (2) 平均准确率MAP, 衡量所有正例三元组在整个排序列表中的排序情况, 值越大说明整体排序情况越好, (3) 平均秩倒排序MRR, 每组三元组中, 正例三元组秩序排名, 值越大越好; (4) AUC, 使用二分类模型评价函数来预测知识库补全效果。

在信息检索中, hit@1表示每组三元组对应的正负实例中, 正实例在该组正负例三元组中排名第一的三元组比例, 尽管这种评价指标较为简单, 但可以有效衡量模型中正例

三元组排在第一位比例。对于每个关系，我们定义的 $hit@1$  (2-9)计算公式如下：

$$hit@1 = \frac{\sum_{q=1}^n hit_q}{n} \quad (2-9)$$

其中 $hit_q$ 表示该关系下第 $q$ 组实体对预测结果中，正例是否排名第一，如果正例排名第一则 $hit_q$ 为1，否则为0。

**平均准确率（mean average precision）** 给定一个头实体和一系列候选尾实体，头尾实体对之间的关系是否存在作为正负例，我们采用局部封闭世界假设，在知识库中存在的实体对标记为1，即为正例，否则为负例。**MAP**核心是利用头实体对应的相关的尾实体出现的位置来进行排序算法准确性的评估。它反映系统在全部相关文档上性能的单值指标。系统模型计算出来的实体对越靠前(rank 越高)，**MAP**就应该越高，否则准确率默认为0。**MAP** (2-10)的计算公式如下,对于每一组正负例实体对，**AP** (2-11)表示一个头实体和对应的候选尾实体（或者一个尾实体及其对于候选头实体）的评价得分。

$$MAP = \frac{\sum_{q=1}^n AP(q)}{n} \quad (2-10)$$

$$AP(q) = \frac{\sum_{i=1}^k P(i) \times rel(i)}{k} \quad (2-11)$$

其中**MAP**表示 $n$ 个头实体和其候选尾实体的平均准确度的均值。**AP**( $q$ )是一个头实体和其 $K$ 个候选尾实体排序结果的评价，如果正确的候选尾实体排在错误的候选尾实体之前，则**AP**值越高；如果所有的头实体和其候选尾实体**AP**值越高，则模型的**MAP**值也就越高。 $rel(i)$ 在第 $i$ 个实体对是预测结果时为1，否则为0。

**平均倒排秩序（mean reciprocal rank）** 通过定义第一个正确的候选尾实体来判断模型好坏，一个模型的**MRR**值越高，说明正确的候选尾实体在模型中排序越靠前。**MRR**通常和**MAP**一起综合评价一个模型的好坏。**MRR** (2-12)的定义公式如下：

$$MRR = \frac{1}{n} \sum_{i=1}^n \frac{1}{r_i} \quad (2-12)$$

其中 $n$ 表示一个关系下所有不同头实体的数目， $r_i$ 是每个头实体对应的正确尾实体的秩序。**MRR**在很多基于问答系统的知识库补全系统<sup>[57]</sup>中也是常见的评价方法之一。通常情况下，基于符号逻辑的知识库补全系统中**MRR**值较高，多数实验结果都在0.9左右，大部分关系中的**MRR**实验结果差别不显著，所以本研究的重点是关注如何显著提高知识库补全系统中关系补全的**MAP**指标。

除了常见的排序模型评价方式，我们在实验过程中也引入二分类分类器模型进行模型评价，同时也考虑模型对正负例排序能力。AUC（Area Under Curve）被定义为操作者曲线，或者称为ROC曲线（tpr和fpr曲线函数）下的面积，显然这个面积的数值不会大于1，同时这个值一般大于0.5。其中真阳性（tpr）(2-13)和假阳性（fpr）(2-14)的计算公式如下所示：

$$tpr = \frac{tp}{tp + fn} \quad (2-13)$$

$$fpr = \frac{fp}{fp + tn} \quad (2-14)$$

随机挑选一个正样本以及一个负样本，当前的分类算法根据计算得到的打分值将这个正样本排在负样本前面的概率就是AUC值。对所有实体对的预测得分进行排序，选择不同的阈值对正负例进行切分计算真阳性tpr和假阳性fpr的得分，从而绘制出完整的ROC曲线，并计算得到AUC的面积大小。AUC取值越大，当前的分类算法越有可能将正样本排在负样本前面，即能够更好的分类，整个模型的排序效果也更好。操作者曲线面积AUC (2-15)计算公式如下所示：

$$auc = \int_0^1 tpr \times d(fpr) \quad (2-15)$$

在知识库补全预测中，无论是逻辑回归模型、支持向量机模型还是学习排序模型，都需要关注模型的二分类的分类器性能问题。对于知识库中的实体对排序、实体对打分预测，在评估一组实体对集合的排序秩序性能之外，也关注模型的分类性能好坏，这些对于知识库补全算法是十分必要的。

## 2.5 本章工作总结

本部分介绍了当前知识库技术相关领域的一些重要研究内容，从知识库构建技术到知识库应用和发展，我们学习研究了各种重要的知识库构建、推理、应用相关内容，介绍了如何评价知识库补全算法的相关指标。本部分重点关注了知识库补全算法相关的研究现状，基于符号逻辑和基于表示学习的知识库补全算法一直是相关领域的研究热点，本部分抓住这些研究中的重要公式、推理和相关假设，为构建我的知识库补全算法框架提供了坚实的基础。



## 第3章 结合关系路径和实体属性的知识库补全

### 3.1 问题引入

大规模的知识库包含大量的实体、关系和属性值，并使用三元组的形式对现实世界中实体关系等各种知识进行表示，知识库中的三元组包括实体关系型和实体属性型两大类。实体关系型三元组形如（北京师范大学，位于，北京），其中“北京师范大学”和“北京”分别表示关系型三元组的头实体和尾实体，“位于”表示是关系路径特征；实体属性型三元组形如（北京师范大学，建校于，1902年），其中“北京师范大学”是头实体，“建校于”是实体属性特征，“1902年”是具体的实体属性特征值。虽然许多知识库的规模很大，但他们仍然是不完备的，如很多人的出生地点并未包含在知识库中，一些演员是否出演过某些电影也是未知的。为了解决这个问题，很多知识库补全的方法被提出来，这类方法的基于知识库中已有的三元组预测新的三元组，如提供一些三元组（北京师范大学，位于，北京）、（北京师范大学，有校长，董奇）、（董奇，居住在，北京）、（北京，位于，中国），我们希望能构建一种关系机器学习模型，预测如（北京师范大学，位于，中国）等知识库中目前不存在的三元组是否存在，即预测实体和实体之间是否具有某种关系，这也被称为关系预测。传统的基于关系路径的知识库补全算法通过抽取实体对之间的关系路径进行关系预测，本部分的研究主要解决如何结合大量的关系路径特征和实体属性特征进行知识库补全的模型预测。

如图3所示，本研究期望对于待预测的头尾实体对（ $h, t$ ），基于图中实体对之间的关系路径抽取关系类型特征，并结合头实体属性、尾实体属性特征、头尾实体组合属性特征等非关系型特征，获得头尾实体对的特征向量，构建目标关系 $r$ 的预测模型，从而判断（ $h, r, t$ ）这个三元组在知识库中是否存在。

### 3.2 提出模型

虽然存在很多的知识库补全模型，但是当前国内外研究的大部分知识库补全算法仅仅采用了知识库中实体和实体之间的关系路径构建算法，考虑到知识库中存在大量的实体属性知识并未被使用，而这些实体属性知识在知识库补全算法中有着重要的作用，本研究提供一种结合关系路径特征和实体属性特征的知识库补全方法。通过提取知识库中

实体的关系路径特征、头实体属性、尾实体属性特征、头尾实体属性组合特征，构建逻辑回归分类模型，进行知识库中的实体关系预测。本部分知识库补全模型主要分为两个步骤：**特征计算**和**模型预测**。特征计算将通过对实体对之间的关系型特征、非关系型特征抽取计算获得特征矩阵；模型预测通过将计算得到的特征矩阵进行训练并预测未知实体对之间的关系。

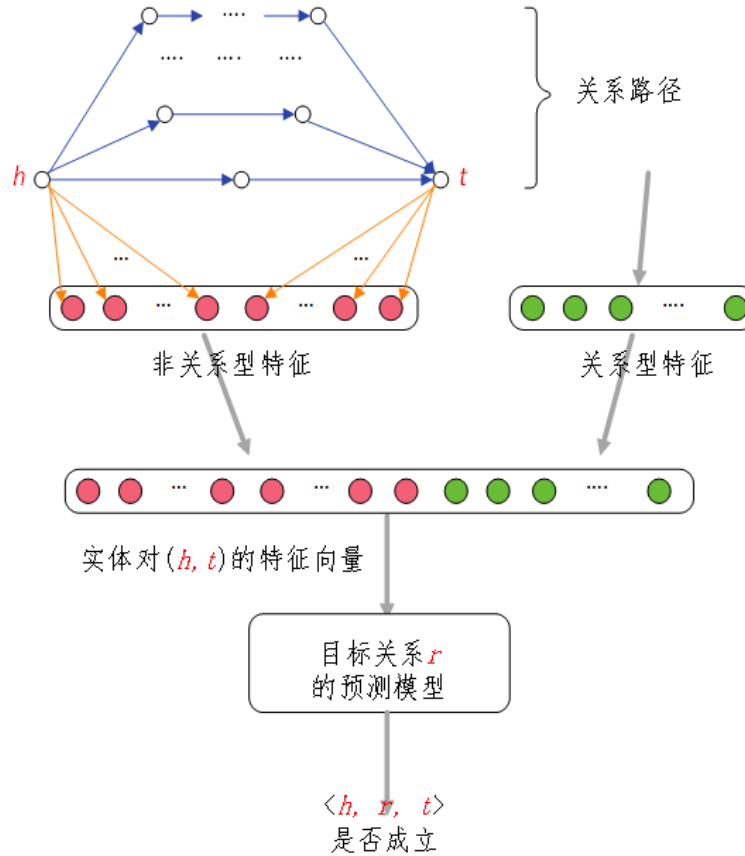


图3 结合实体属性和关系路径知识库补全计算过程

特征计算是知识库补全的特征工程阶段，特征的数量和特征有效性是决定知识库补全效果的关键点之一。给定一个完整的知识库，我们构建知识库中实体和实体之间的拓扑结构，将知识库中的实体和关系分别转化为图中的顶点和边，通过随机游走算法，计算获得知识库中的关系路径特征向量。对于知识库中每种待预测的关系，我们抽取对应关系下的实体对，将实体对的头实体和尾实体在图中进行随机游走<sup>[58]</sup>，获得连接头实体和尾实体的关系路径特征，从而获得了知识库补全的关系路径特征。

逻辑回归分类模型被用来构建预测模型，对于每个关系下的头尾实体对，预测这个

实体对是否具有某种关系。逻辑回归构建了一个二分类模型，对于特征计算中获得的特征矩阵，进行模型训练和预测，逻辑回归能有效的基于关系路径特征、实体属性特征进行模型分析，并帮助我们研究分析合理有效的关系路径特征、实体属性特征，进行模型效果分析。

### 3.2.1 模型框架

图4展示了本部分基于关系路径的知识库补全算法基本框架。其中对于每个给定关系，我们将这个关系下所有的三元组集合切分为训练集和测试集。模型的训练分为两个部分：（1）特征计算，包括关系路径特征计算和实体属性特征计算两个部分，包括如何获取关系路径特征类型和实体属性特征类型，如何计算关系路径特征和实体属性特征；（2）模型预测，包括基于分类模型的逻辑回归算法，基于树模型的学习排序算法，基于深度神经网络的排序算法等。根据对模型预测评价指标的不同，不断调整预测模型参数，使得模型能够达到最优的结果。

特征计算通过给定一个目标关系 $r$ 和这个关系对应的实体对集合 $I_s$  (3-1)，其中 $h_j$ 和 $t_j$ 分别表示关系 $s$ 下的头实体和尾实体。对于知识库中的关系实体组成的图，我们期望找出连接头尾实体对的关系路径特征。

$$I_s = \{(h_j, t_j) | < h_j, t_j > \in KB\} \quad (3-1)$$

由于连接头尾实体对之间的路径数量很大，通常需要限定关系路径的长度，并采用随机游走算法计算选择从头实体到尾实体的关系路径，将这些路径集合作为关系路径特征。对于能连接从头实体到尾实体的关系路径，我们记录这个关系路径的类型，作为模型预测的特征。我们计算0-1二值化后的的路径类型，作为关系路径特征的特征值。对于每个关系抽取的关系特征，我们最终记为 $V_r(h_i, t_i)$ 作为关系路径特征，表示从头实体到尾实体有哪些关系路径进行连接。例如对于实体对（北京师范大学，位于，北京）三元组，我们可以基于上述的关系路径抽取算法获得（北京师范大学，位于，北京市海淀区，位于，北京）、（北京师范大学，有校长，董奇，居住在，北京）等多条路径来获得关系“位于”对应的关系路径类型特征。

本论文通过枚举不同维度的实体属性特征，计算这些实体对在这些实体属性特征下的特征值。属性特征抽取过程较为复杂，不仅需要考虑不同类型的属性信息不一致问题，也要研究如何处理缺失值问题。首先对于每个实体，有着不同类型的描述特征，如一个人的信息，不仅有出生年月这种时间类型的信息，也有年龄、性别等不同种类的属性信息。我们采用了对实体信息进行标准化的方法，将实体对的头实体和尾实体这些不

同属性特征进行归一化处理范围限定在 $[0, 1]$ 之间。进行归一化后计算获得新的属性特征记为 $V_l(h_i)$ 和 $V_l(t_i)$ ，分别表示头实体和尾实体在所有熟悉特征下的实体属性特征向量，其中 $h_i$ 和 $t_i$ 表示给定关系 $l$ 的第 $i$ 个头实体和尾实体，同时我们对于头实体和尾实体进行相减计算获得 $V_l(h_i - t_i)$ 属性值。除此之外，对于很多实体在不同特征上的缺失值，我们将缺失值进行了补0处理，期望获得更优结果。如对于“北京师范大学”这个实体，我们抽取了他的“建校时间”、“占地面积”等所有的实体属性特征类型，并计算每个实体对在这些特征类型下标准化、归一化等特征工程转化后的特征值，而且将缺失属性特征补零。

除了抽取不同类型的关系路径和实体属性特征，构建实体对的特征向量，如何能更好结合实体属性特征和关系路径特征也是知识库补全的重要研究问题之一。最终，本研究通过组合不同类型特征，构建特征工程，抽取每个关系下实体对的头实体属性特征向量、尾实体属性特征向量、头尾实体组合属性特征向量集合。在每个关系下，将这些实体对向量集合构建逻辑回归模型，并选择对数损失函数，采用随机梯度下降算法，学习不同类型特征的权重，分析这些不同的关系路径特征、实体属性特征对于知识库补全效果的影响。

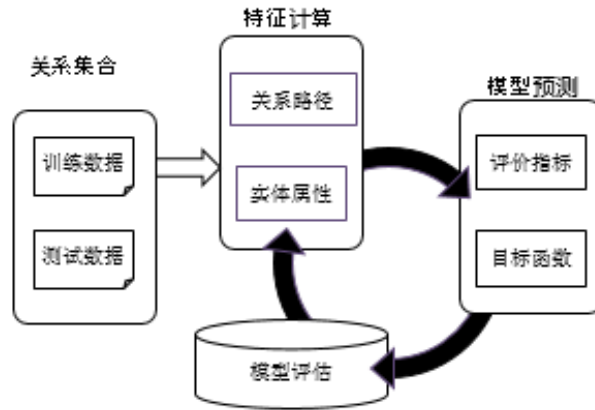


图4 知识库补全计算框架

### 3.2.2 关系路径特征计算

特征计算是知识库补全的特征抽取阶段，特征的数量和特征有效性是决定知识库补全效果的关键点之一。给定一个由三元组组成的知识库，我们将知识库中的实体和关系分别看做为图中的顶点和边，这样就能构建一个基于图模型的知识库补全系统。对于知识库中每种特定待预测的关系，我们抽取对应关系下的实体对，将实体对的头实体和尾实体在图中进行随机游走<sup>[58]</sup>，获得连接头实体和尾实体的关系路径特征，从而获得了知

识库补全的关系路径特征。给定一个目标关系 $r$ 和这个关系对应的实体对集合 $I_r$ ，我们期望找出给定关系下，所有连接头尾实体对的关系路径特征，最终获得一个关系路径类型的集合。在知识库中，由于连接头尾实体对之间的路径数量很大，通常需要限定关系路径的长度，一般限定图中的关系路径长度在2-6之间。

### 3.2.2.1 关系路径类型集合

对于能连接从头实体到尾实体的关系路径，我们记录这个关系路径的类型，作为模型预测的特征类型。我们采用随机游走算法（random walk）<sup>[59]</sup>计算选择从头实体到尾实体的关系路径，将这些路径集合作为关系路径类型特征集合。随机游走算法通过在知识库图的结构中进行随机游走，获取实体和实体之间的常见的关系路径，从而能获得一个关系下频繁出现的关系路径类型，并基于这些关系路径类型构建的特征矩阵，我们可以用来预测实体和实体之间的关系。例如对于实体对（北京师范大学，位于，北京）三元组，我们可以基于上述的关系路径抽取算法获得（北京师范大学，位于，海淀区，位于，北京）、（北京师范大学，有校长，董奇，居住在，北京）等多条路径来获得关系“位于”对应的关系路径特征。对于上述两条关系路径，我们抽取关系路径中的关系类型，生成一个关系路径类型集合，包括（位于-位于）和（有校长-居住在）等不同关系组成的关系路径类型，从将多种不同的关系路径类型结合而构成一个关系路径类型集合。对于反方向的关系路径我们采用 $relation^{-1}$ 表示。

### 3.2.2.2 关系路径特征向量

获取关系路径类型集合（记为 $S(r)$ ）后，我们将给定关系 $r$ 下的实体对 $(h_j, t_j)$ 计算所有关系路径类型集合下的特征值。如果实体对 $(h_j, t_j)$ 在集合 $S(r)$ 中的某个关系路径类型 $s_i$ 存在，则我们可以将该实体对的对应关系路径类型的特征值记为1，否则该实体对在这个关系路径类型下的特征值记为0，这种算法简化了不同关系类型在知识库中的权重，同时一些实验表明<sup>[55]</sup>进行0-1二值化不仅可以简化关系路径类型特征和特征值计算过程，同时对于实验结果的影响并不显著。为了能在不影响模型效果的前提下，加速我们的计算过程，我们对随机游走算法计算获得的关系路径类型特征值计算进行了0-1二值化。

### 3.2.3 实体属性特征计算

除了3.2.2中计算得到的关系路径类型，考虑到在知识库中仍然存在大量的实体属性特征并未被使用，本研究也对每个关系下的实体属性特征进行计算，获得这些实体属性

特征下的特征值。实体属性特征类型获取较为简单，只需要枚举知识库中存在的不同实体属性类型，即可将这些实体属性类型作为实体特征集合。而实体属性特征值的处理和计算是获取更有效特征、提升知识库补全算法的关键。

属性特征抽取过程较为复杂，不仅需要考虑不同维度的属性信息不一致问题，也要研究如何处理缺失值问题。本论文通过枚举不同维度的实体属性特征，计算这些实体对在这些实体属性特征下的特征值。首先对于每个实体，有着不同类型的描述特征，如一个人的信息，不仅有出生年月这种时间类型的信息，也有年龄、性别等不同种类的属性信息。我们采用了对实体信息进行标准化的方法，将实体对的头实体和尾实体这些不同属性特征进行归一化处理范围限定在 $[0, 1]$ 之间。进行标准化后计算获得新的属性特征记为 $V_l(h_i)$ 和 $V_l(t_i)$ ，分别表示头实体和尾实体在所有熟悉特征下的实体属性特征向量，其中 $h_i$ 和 $t_i$ 表示给定关系 $l$ 的第 $i$ 个头实体和尾实体，同时我们对于头实体和尾实体进行相减计算获得 $V_l(h_i - t_i)$ 属性值。除此之外，对于很多实体在不同特征上的缺失值，我们将缺失值进行了补零处理，期望获得更优结果。如对于“北京师范大学”这个实体，我们抽取了他的“建校时间”、“占地面积”等所有的实体属性特征，并计算每个实体对在这些特征下标准化后的特征值，而且将缺失属性特征补零。

在对每个关系下的实体对集合进行实体属性特征抽取计算后，我们获得了这个关系下头实体的实体属性特征、尾实体的实体属性特征、对头尾实体进行归一化后的归一化实体属性特征，以及头实体和尾实体差值计算得到的实体属性特征。

通过将3.2.2关系路径特征和3.2.3抽取的实体属性特征进行结合，构建结合实体属性特征和关系路径特征的组合特征，从而获得了由关系路径特征、头实体属性特征、尾实体属性特征、头尾实体结合属性特征、结合关系路径类型和实体属性类型特征组合的特征向量，从而获得每个关系下所有实体对组合构建的特征矩阵，并构建机器学习模型对生成的特征矩阵进行学习预测，获得知识库补全中的实体对的关系。

### 3.3 实验结果

本研究通过抽取关系路径特征和实体属性特征，并对抽取特征进行正则化特征工程后，构建了基于逻辑回归的预测模型，期望能进行研究分析，发掘如何更好结合关系路径和实体属性特征的方法。如图3所示，我们获取知识库中的关系型特征类型即关系路径特征类型和非关系型特征类型即实体属性特征类型，将两种特征相互融合，获得关系路径特征、头实体属性特征、尾实体属性特征、头尾实体结合属性特征组成的预测向量，从而构建预测模型进行知识库补全模型训练，获得每个关系下的知识库补全训练模型，

从而能更好进行知识库补全预测。

本研究的实验分为两组，包括基于YAGO知识库和基于Freebase知识库的模型预测评估。尽管有很多大规模的知识库，但考虑到这些知识库需要包含关系三元组和实体属性三元组，我们选择了这些知识库实体和关系丰富的YAGO2和Freebase知识库，并选择了YAGO和Freebase中实体属性丰富的实体关系组成集合，构建组成YAGO和FB15K数据集。对于这些知识库中的实体关系，我们抽取其中实体属性特征较多的三元组，作为一个训练预测数据，从而避免很多缺少实体属性三元组的实体在实体关系缺失时影响模型效果。

表 4 知识库基本数据统计

|        | FB15K  | YAGO   |
|--------|--------|--------|
| 实体     | 14951  | 58130  |
| 关系类型   | 1345   | 32     |
| 实体属性类型 | 336    | 25     |
| 属性三元组  | 25776  | 141606 |
| 关系三元组  | 592213 | 499350 |
| 训练集    | 483142 | 399480 |
| 测试集    | 59071  | 49132  |

表4中展示了YAGO和Freebase知识库的基本训练集合、测试集合、关系路径三元组、实体属性三元组的数据统计情况。我们选择了YAGO和Freebase中具有较多属性特征的实体关系三元组，并通过PRA、SFE算法抽取这些三元组的关系路径特征，同时，我们从知识库中抽取每个实体的实体属性特征类型，并计算这些实体属性特征值，获得每个关系下头实体属性特征向量、尾实体属性特征向量、头尾实体结合属性特征向量，并将关系路径特征向量和上述几种实体属性特征向量结合组成预测实体关系的特征向量。本章节的实验中我们选择三元组的正负实体对比例是1: 4，选择YAGO和Freebase中实体的要求是，每个实体在知识库中有超过5个及以上的实体属性值。

知识库补全模型采用MAP、MRR、AUC和hit@1多种不同的评估指标进行模型评价，其中MAP、MRR和hit@1是在排序算法中常见的模型评价指标，很多常见知识库补全算法如PRA、SFE和TransE等模型都采用这些指标进行模型评价。AUC是常见的分类和排序模型评价指标，可以很好的衡量二分类时模型预测的效果。这些指标也在各种常见知识库补全、信息检索、图链接预测任务作为评价函数。本研究全面深入的研究了结合实体属性和关系路径的知识库补全算法，为了能全面深入分析不同类型特征带来的模型效果影响，本部分实验使用了上述四种评价指标同时进行模型评估，期望采用客观公正的评价

指标，分析我们的算法是否能有效地提高模型效果。

### 3.3.1 YAGO知识库实验

本部分的效果展示了在YAGO知识库下进行知识库补全算法的结果，我们的方法和对比方法被分为三类。其中，SFE-literal（图中简称SFE-lit）和PRA-literal（图中简称PRA-lit）是结合关系路径特征和实体属性特征的方法，SFE-literal是基于子图特征抽取算法(SFE)抽取关系路径特征，并结合实体属性特征构建的特征向量，而PRA-literal是基于路径排序算法（PRA）抽取关系路径特征，并结合literal属性特征构建的特征向量模型。SFE和PRA是只基于关系路径特征进行知识库补全的两种算法，TransE和TransR是基于表示学习中进行知识库补全算法中典型的两种模型，通过学习特征的低维度向量表示，进行知识库中实体和实体之间的关系预测。本部分分析了表示学习算法中知识库补全算法，只基于关系路径的知识库补全算法，结合实体属性和关系路径的知识库补全算法的模型效果。

**知识库关系综合分析** 通过对所有的关系综合进行分析发现，无论是路径排序算法还是子图特征抽取模型，在结合关系路径特征和实体属性特征构建后模型，MAP、MRR和AUC预测结果最好，总体的预测结果稳定，能有效处理分类和排序问题。而仅仅基于关系路径特征，未加实体属性特征的PRA和SFE算法特征较好，使用表示学习的TransE和TransR模型效果在MAP中预测较差，但是通过AUC和hit@1进行模型的二分类效果结果和基于符号逻辑的效果相似。结果显示，加入实体属性特征后，结合实体属性特征和关系路径特征的补全技术，相比只采用关系路径特征的补全技术有更高的准确性，在进行排序相关的预测效果评测时，有较好的效果。在采用二分类模型进行评价时候，无论是符号逻辑的路径排序相关算法还是表示学习的低维嵌入算法，都能有效的进行模型预测。

从表5分析三类不同知识库补全算法的四种评估指标可以发现。相比只采用关系路径的知识库补全算法，结合实体属性特征和关系路径特征的知识库补全算法，在YAGO数据集集合上，MAP预测结果有较大的提升，相比原来的模型，SFE-literal和PRA-literal在计算MAP时，都获得了近1%的效果提升，模型提升效果统计显著。而传统的基于表示学习的知识库补全算法，由于进行了低维表示模型近似，无论是MAP还是MRR等各种指标都较基于关系路径的知识库补全算法差别较大。但基于二分类模型评价函数AUC指标进行分析时可以发现，基于关系路径、结合实体关系路径特征和基于表示学习的知识库补全算法，在模型效果上没有显著差异。



基于表5分析YAGO知识库补全实验可以获得如下结论：（1）预测知识库中新三元组通过结合关系路径特征和实体属性特征能更加的精确有效，无论是以分类为优化目标的分类模型，还是以排序为优化目标的排序问题。（2）由于在YAGO知识库集合中，有更多的属性事实进行关系预测，采用特征组合，能获得更多的不同类型组合特征，达到更好的优化结果，因此，结合属性事实和关系事实进行预测是非常重要的。（3）对于某些特殊的关系，进行标准化处理是非常有效的，但是并非对于所有的属性事实进行标准化有效，选择合适的属性特征和实体关系特征进行组合是十分必要的。（4）本研究的实验结果表明，对于多数YAGO2中的关系来说，我们的属性事实不仅可以用来预测关系事实，而且还能调整原来的关系特征的路径权重，使得模型预测更加合理。因此，结合属性事实和更丰富的关系特征能获得更好的知识库补全结果。

表 5 YAGO知识综合关系模型预测结果

|         | MAP   | MRR    | AUC   | hit@1 |
|---------|-------|--------|-------|-------|
| SFE-lit | 78.6% | 100.0% | 92.9% | 45.3% |
| PRA-lit | 78.1% | 97.8%  | 93.4% | 45.1% |
| PRA     | 77.6% | 97.8%  | 93.2% | 45.5% |
| SFE     | 77.3% | 97.8%  | 93.2% | 45.5% |
| transE  | 56.4% | 91.2%  | 92.7% | 45.6% |
| transR  | 46.4% | 65.4%  | 92.9% | 46.1% |

**知识库单个关系结果分析** 我们对YAGO知识库中的每个关系预测的平均准确率（MAP）得分结果进行了分析，如表6可知，对于知识库中的每一个关系，我们可以发现TransR模型的预测结果效果最差，其次是基于表示学习的TransE模型。而基于关系路径的预测结果都很好，其中，结合实体属性和关系路径特征的知识库补全模型效果最好，而无论关系路径特征选择PRA进行关系路径特征计算还是选择SFE关系路径特征计算，模型的预测结果都比基于表示学习的预测结果好。但是对于一些关系如worksAt、hasWonPrize等，在表示学习模型TransE上有相对较好的优势。总体来说，在超过15种的关系集合如actedIn、graduatedFrom等关系上，基于关系路径的模型预测结果比表示学习的模型预测结果更好，而结合关系路径特征和实体属性特征的模型相比只有关系路径特征的模型，平均准确率有显著的提高。其他的模型评价方式如平均倒排秩、AUC等指标差别不大，进行单个关系分析后发现结果差别不大，说明在进行二分类等模型预测时，这些方法都有较好的预测效果，因此本研究并没有对单个关系的上述指标进行详细的结果分析，重点关注了平均准确率在不同关系上的预测效果。

表 6 YAGO知识库单个关系平均准确率得分

| MAP-score          | SFE-lit | PRA-lit | PRA    | SFE    | transR | transE |
|--------------------|---------|---------|--------|--------|--------|--------|
| actedIn            | 79.3%   | 80.1%   | 75.3%  | 71.6%  | 18.0%  | 36.0%  |
| dealsWith          | 50.1%   | 53.3%   | 52.8%  | 53.6%  | 41.6%  | 42.4%  |
| diedIn             | 89.4%   | 89.9%   | 88.2%  | 87.9%  | 27.3%  | 37.7%  |
| graduatedFrom      | 62.8%   | 57.0%   | 54.9%  | 59.0%  | 26.4%  | 41.3%  |
| happenedIn         | 69.3%   | 63.1%   | 63.2%  | 62.1%  | 41.9%  | 65.2%  |
| hasCapital         | 96.6%   | 96.4%   | 94.7%  | 95.7%  | 71.8%  | 92.5%  |
| hasChild           | 95.8%   | 95.8%   | 95.8%  | 95.8%  | 37.5%  | 41.6%  |
| hasNeighbor        | 69.2%   | 67.7%   | 70.9%  | 65.8%  | 55.1%  | 41.8%  |
| hasOfficialLanguag | 61.3%   | 65.5%   | 65.7%  | 62.4%  | 29.7%  | 64.1%  |
| hasWonPrize        | 53.7%   | 55.3%   | 54.8%  | 54.8%  | 46.0%  | 57.8%  |
| influence          | 100.0%  | 100.0%  | 100.0% | 100.0% | 43.1%  | 34.3%  |
| isAffiliatedTo     | 98.2%   | 97.9%   | 98.0%  | 98.1%  | 78.8%  | 89.8%  |
| isCitizenOf        | 80.0%   | 80.5%   | 80.5%  | 79.6%  | 44.2%  | 70.6%  |
| isConnectedTo      | 57.4%   | 55.7%   | 63.3%  | 64.6%  | 60.7%  | 54.3%  |
| isLeaderOf         | 84.3%   | 78.7%   | 78.1%  | 78.0%  | 25.3%  | 48.5%  |
| isLocatedIn        | 94.6%   | 94.4%   | 94.1%  | 94.2%  | 85.4%  | 63.8%  |
| isMarriedTo        | 89.0%   | 90.5%   | 88.8%  | 87.6%  | 88.0%  | 69.3%  |
| livesIn            | 81.2%   | 78.9%   | 73.3%  | 74.8%  | 32.0%  | 50.1%  |
| own                | 100.0%  | 100.0%  | 100.0% | 100.0% | 16.9%  | 44.2%  |
| participatedIn     | 62.1%   | 66.2%   | 64.1%  | 64.3%  | 35.3%  | 60.1%  |
| playsFor           | 99.6%   | 99.7%   | 99.7%  | 99.6%  | 79.6%  | 91.2%  |
| wasBornIn          | 95.5%   | 95.0%   | 94.2%  | 94.4%  | 54.1%  | 63.5%  |
| worksAt            | 38.6%   | 36.3%   | 34.8%  | 35.1%  | 30.2%  | 36.8%  |

**知识库关系预测特征分析** 我们进一步分析了每个关系下关系路径特征和实体属性特征的关系路径特征权重变化、实体属性特征改变等情况，表7显示了在YAGO知识库中的集种关系典型：Export、GraduateFrom、HasAcademicAdvisor，通过结合实体属性特征和关系路径特征，学习获得的重要权重。相比于路径排序算法这种只使用关系特征进行预测的方式，结合属性事实特征不仅仅能增加模型预测的全面性具有更好的模型泛化能力，还能将模型预测结果精度提高，同时也能调整逻辑回归算法中不同关系路径特征和不同属性事实特征的权重，对于部分不重要的特征起到特征选择的作用。通过将关系路径特征和属性事实特征进行结合，使得模型的预测结果更加可靠。如表所示，我们分析关系“graduateFrom”可以发现，除了常见的关系路径特征： $isCitizenOf \rightarrow isCitizenOf^{-1} \rightarrow$

$livesIn \rightarrow isLocatedIn^{-1}$ 、 $isAffiliatedTo^{-1} \rightarrow isCitizenOf \rightarrow livesIn \rightarrow isLocatedIn^{-1}$  等，部分关系路径特征重要性发生变化，部分关系路径特征在模型训练中被特征选择筛出，从而获得了新的更具有预测能力的关系路径特征。同时我们也可以发现一些重要的实体属性特征如： $wasCreatedOnDate$ 、 $happenedOnDate$ 等都在关系预测结果中有着重要的作用。通过分析这些关系路径特征、实体属性特征的权重我们可以发现，实体属性特征不仅能提高关系预测中的精度，有较高的准确性，同时也能调整关系路径的权重，产生更多的有表达力的关系路径特征。这表明进行知识库补全模型预测的时候，结合关系路径和实体属性特征能有效进行组合从而获得更具有模型泛化能力的特征向量。

### 3.3.2 Freebase知识库实验

本部分的效果展示了在Freebase知识库下进行知识库补全算法的结果，我们的方法和对比方法也被分为三类：结合关系路径和实体属性的知识库补全、基于关系路径的知识库补全和基于表示学习的知识库补全算法，每组实验中都选择两种典型的算法进行实验分析。分析过程分为两部分，知识库关系综合结果分析和知识库单个关系结果分析。

**知识库关系综合结果分析** 和YAGO知识库补全实验分析过程相似，表8展示了在不同评价指标下Freebase知识库中部分关系的预测结果。其中，SFE-literal和PRA-literal是结合关系路径特征和实体属性特征的方法，PRA和SFE是只基于关系路径特征的知识库补全算法，TransE和TransR是基于表示学习、低维向量的知识库补全方法。其中结合实体属性和关系路径特征的知识库补全模型评价的MAP、MRR和AUC预测结果最好，尤其在采用AUC进行模型评价时候，相比于表示学习方法，基于符号逻辑的知识库补全算法特征有非常高的提升。从基于MAP和MRR的评价指标来看，加入实体属性特征的总体预测结果最稳定，效果最好，而未加实体属性特征的PRA和SFE特征较好，使用表示学习的TransE和TransR模型效果在MAP中预测较差，结果显示，加入实体属性特征后，结合实体属性特征和关系路径特征的补全技术，相比只采用关系路径特征的补全技术有更高的准确性，在进行排序相关的预测效果评测时，有较好的效果。在采用二分类模型进行评价时候，无论是符号逻辑的路径排序相关算法还是表示学习的低维嵌入算法，都能有效的进行模型预测。

总体上看，使用MAP、MRR、AUC以及hit@1评价指标时，基于符号逻辑的知识库补全算法效果较基于表示学习的算法性能有较大的提升，而结合实体属性特征和结合关系路径特征的知识库补全算法相比仅仅使用关系路径特征的模型效果更好。这说明无论采用分类作为模型的优化方向，还是采用实体对排序的序列作为知识库补全优化目标，使

表 7 YAGO知识库关系路径和实体属性特征比较

| Export             |   |   |
|--------------------|---|---|
| PRA                | imports $\rightarrow$ hasMusicalRole $^{-1}$ $\rightarrow$ hasMusicalRole<br>livesIn $^{-1}$ $\rightarrow$ diedIn $\rightarrow$ imports<br>livesIn $^{-1}$ $\rightarrow$ wasBornIn $\rightarrow$ dealsWith $^{-1}$ $\rightarrow$ imports<br>isCitizenOf $^{-1}$ $\rightarrow$ influences $^{-1}$ $\rightarrow$ isCitizenOf $^{-1}$ $\rightarrow$ imports<br>isCitizenOf $^{-1}$ $\rightarrow$ wasBornIn $\rightarrow$ dealsWith $^{-1}$ $\rightarrow$ imports   |   |
| PRA-lit            | hasCapital $\rightarrow$ hasCapital $^{-1}$ $\rightarrow$ exports<br>imports $\rightarrow$ hasMusicalRole $^{-1}$ $\rightarrow$ hasMusicalRole $^{-1}$<br>isCitizenOf $^{-1}$ $\rightarrow$ wasBornIn $\rightarrow$ hasCapital $^{-1}$ $\rightarrow$ exports<br>exports $\rightarrow$ hasMusicalRole $^{-1}$ $\rightarrow$ hasMusicalRole<br>isInterestedIn $^{-1}$ $\rightarrow$ wasBornIn $\rightarrow$ hasCapital $^{-1}$ $\rightarrow$ exports  | hasGini<br>hasInfaltion<br>hasEconomicGrowth<br>hasPoverty<br>wasDestroyedOnDate                    |
| GraduateFrom       |   |   |
| PRA                | isCitizenOf $\rightarrow$ isCitizenOf $^{-1}$ $\rightarrow$ livesIn $\rightarrow$ isLocatedIn $^{-1}$<br>diedIn $\rightarrow$ happenedIn $^{-1}$ $\rightarrow$ participatedIn $^{-1}$ $\rightarrow$ isLocatedIn $^{-1}$<br>hasWebsite $\rightarrow$ hasWebsite $^{-1}$ $\rightarrow$ livesIn $\rightarrow$ isLocatedin $^{-1}$<br>isAffiliatedTo $\rightarrow$ isAffiliatedTo $^{-1}$ $\rightarrow$ isCitizenOf $\rightarrow$ isLocatedIn $^{-1}$<br>isAffiliatedTo $^{-1}$ $\rightarrow$ isCitizenOf $\rightarrow$ livesIn $\rightarrow$ isLocatedIn $^{-1}$ |   |
| PRA-lit            | isAffiliatedTo $\rightarrow$ isAffiliatedTo $^{-1}$ $\rightarrow$ isCitizenOf $\rightarrow$ isLocatedIn $^{-1}$<br>hasAcademicAdvisor $\rightarrow$ hasAcademicAdvisor $^{-1}$ $\rightarrow$ graduated-<br>From<br>hasWebsite $\rightarrow$ hasWebsite $^{-1}$ $\rightarrow$ livesIn $\rightarrow$ isLocatedin $^{-1}$<br>isAffiliatedTo $\rightarrow$ isAffiliatedTo $^{-1}$ $\rightarrow$ isLeaderOf $\rightarrow$ isLocatedIn $^{-1}$<br>isCitizenOf $\rightarrow$ isCitizenOf $^{-1}$ $\rightarrow$ livesIn $\rightarrow$ isLocatedIn $^{-1}$             | happenedOnDate<br>wasDestroyedOnDate<br><br>wasDestroyedOnDate<br>wasCreatedOnDate<br>wasBornOnDate |
| HasAcademicAdvisor |   |   |
| PRA                | wasBornIn $\rightarrow$ happenedIn $^{-1}$ $\rightarrow$ participatedIn $^{-1}$ $\rightarrow$ livesIn $^{-1}$<br>diedIn $\rightarrow$ hasCapital $^{-1}$ $\rightarrow$ isLocatedIn $^{-1}$ $\rightarrow$ diedIn $^{-1}$<br>worksAt $\rightarrow$ graduatedFrom $^{-1}$ $\rightarrow$ livesIn $\rightarrow$ wasBornIn $^{-1}$<br>hasAcademicAdvisor $\rightarrow$ hasAcademicAdvisor $^{-1}$ $\rightarrow$ influences $^{-1}$<br>$\rightarrow$ hasAcademicAdvisor<br>livesIn $\rightarrow$ diedIn $\rightarrow$ hasAcademicAdvisor                             |   |
| PRA-lit            | hasGender $\rightarrow$ hasGender $^{-1}$<br>worksAt $\rightarrow$ graduatedFrom $^{-1}$ $\rightarrow$ livesIn $\rightarrow$ wasBornIn $^{-1}$<br>hasAcademicAdvisor $\rightarrow$ hasAcademicAdvisor $^{-1}$ $\rightarrow$ diedIn $\rightarrow$<br>diedIn $^{-1}$<br>diedIn $\rightarrow$ diedIn $^{-1}$ $\rightarrow$ livesIn $\rightarrow$ livesIn $^{-1}$<br>graduatedFrom $\rightarrow$ worksAt $\rightarrow$ worksAt $^{-1}$ $\rightarrow$ worksAt  | wasDestroyedOnDate<br>hasHeight<br>hasHeight<br><br>wasBornOnDate<br>diedOnDate                     |

表 8 Freebase知识库关系综合预测结果

|         | MAP   | MRR    | AUC   | hit@1 |
|---------|-------|--------|-------|-------|
| transR  | 66.4% | 95.5%  | 86.1% | 44.1% |
| transE  | 72.9% | 98.6%  | 89.1% | 47.3% |
| SFE     | 86.4% | 98.6%  | 97.0% | 55.5% |
| SFE-lit | 86.5% | 100.0% | 97.1% | 55.6% |
| PRA     | 86.9% | 98.2%  | 97.0% | 55.5% |
| PRA-lit | 87.1% | 100.0% | 97.1% | 55.6% |

用符号逻辑相比于近似的表示学习方法效果都有明显的提升。

**知识库单个关系结果分析** 我们对知识库预测结果的平均准确率分关系讨论，从表9中分析Freebase中的各个关系预测结果的MAP值。我们可以发现，在Freebase的15种重要关系中，我们通过采用结合关系路径特征和实体属性特征的知识库补全算法，相比于只采用关系路径的知识库补全算法，我们的模型效果在MAP排序指标上有很大的提升。如/tv/tv/genreprograms、/film/actor/film/film/performance/film和 /film/director/film、/media/commonnetflix/genretitles等关系使用MAP评测都有较大的模型效果提升。总体来说，超过10种关系都能获得超过60%平均准确率（MAP）。超过14种关系的平均准确率（MAP）超过50%。在关系路径特征上总体的平均准确率虽然高于表示学习的准确率，但是分关系来看，很多关系路径计算获得的平均准确率低于基于表示学习的平均准确率。各个关系的基于表示学习的平均准确率相差不大，说明表示学习能更稳定的进行模型预测。

在分关系的平均准确率（MAP）评价结果表中，为了能更好的介绍知识库中不同模型的预测结果，观察结合实体属性和关系路径的特征计算得到模型的优势。我们以关系/people/person/nationality和关系/film/actor/film/film/performance/film进行分析，研究关系路径特征和实体属性特征在路径排序算法中的特征表现。通过分析MAP指标来看，TransE和TransR这两个模型在各个关系上的效果较为稳定，符号逻辑相关算法在这些不同的关系上效果差别较大，通过分析相关的关系路径和实体属性权重，我们可以发现，这些关系路径特征相对于YAGO知识库来说，关系路径权重区别并不明显，甚至在结合实体属性特征后，对于/film/actor/film/film/performance/film这个关系来说，所有的关系路径和实体属性路径特征综合从44120减少到42787种特征。说明基于随机游走抽取的关系路径特征并不能很好的区分Freebase中实体和实体之间的关系，但是当加入关系路径特征后，能很好的惩罚这些加入区分度不好的关系路径特征。

其次尽管部分关系相对于表示学习来说，预测结果的效果较差，但是，基于图模型

表 9 FB15K部分关系MAP得分

| MAP-score                              | PRA-lit | PRA    | SFE-lit | SFE    | transE | transR |
|--|---------|--------|---------|--------|--------|--------|
| /people/person/profession              | 12.1%   | 11.9%  | 12.3%   | 12.0%  | 77.1%  | 71.7%  |
| /film/actor/film/film/performance/film | 28.3%   | 26.1%  | 22.1%   | 20.5%  | 87.0%  | 78.4%  |
| /media/common/netflix/genre/titles     | 50.1%   | 48.2%  | 49.9%   | 49.5%  | 63.7%  | 65.6%  |
| /people/ethnicity/people               | 57.1%   | 57.1%  | 57.1%   | 57.1%  | 60.4%  | 50.3%  |
| /film/film/genre/films/in/this/genre   | 57.2%   | 55.6%  | 41.4%   | 41.8%  | 68.7%  | 64.2%  |
| /music/instrument/instrumentalists     | 58.2%   | 58.2%  | 58.2%   | 58.2%  | 53.7%  | 52.4%  |
| /music/genre/artists                   | 69.5%   | 69.5%  | 69.5%   | 69.5%  | 77.7%  | 65.3%  |
| /location/location/time/zones          | 77.4%   | 77.1%  | 77.4%   | 77.4%  | 82.7%  | 72.2%  |
| /tv/tv/genre/programs                  | 80.9%   | 80.9%  | 80.9%   | 80.9%  | 78.1%  | 60.4%  |
| /people/person/nationality             | 81.1%   | 79.6%  | 81.7%   | 80.7%  | 75.4%  | 73.7%  |
| /people/cause/of/death/people          | 82.6%   | 82.6%  | 82.6%   | 82.6%  | 52.9%  | 44.5%  |
| /music/record/label/artist             | 83.7%   | 83.7%  | 83.7%   | 83.7%  | 55.0%  | 41.8%  |
| /film/production/company/films         | 85.5%   | 85.5%  | 85.5%   | 85.5%  | 68.7%  | 59.1%  |
| /film/director/film                    | 100.0%  | 100.0% | 100.0%  | 100.0% | 94.0%  | 88.3%  |
| /film/film/directed/by                 | 100.0%  | 100.0% | 100.0%  | 100.0% | 95.2%  | 85.6%  |

的知识库补全系统中，大部分关系的预测相对于基于表示学习效果提升很多。这一方面说明基于表示学习是一种近似的模型预测，而基于符号逻辑的关系路径学习排序则是一种精确而有效的学习方式。另一方面也说明，在知识库构建的图模型中，部分关系较为稀疏的实体之间使用表示学习，预测结果较为理想，而采用结合实体属性和关系路径的符号逻辑补全算法，则能基于有效的关系路径，获得十分有用的预测模型结果。

其次，由于Freebase知识库总共包含超过1345种不同类型的关系，这些关系覆盖了电影、人种、电视剧、音乐等不同类型的关系，这些不同的关系中部分关系具有对称性，一些关系和关系之间差别很大，如何能将这不同类型互不影响或者相互对称的关系区分出来，构建一个合理的图，基于这种合理的图模型进行关系路径的抽取和实体属性特征的计算，是未来研究的重要研究步骤。

### 3.4 本章工作总结

本部分通过抽取了实体属性类型、关系路径类型作为知识库补全的特征向量类型，并通过计算每个关系下实体对的实体属性特征值和关系路径特征值。获取关系路径特征值、头实体属性特征值、尾实体属性特征值，头尾结合实体属性值，将这些特征值进行组合，并构建了基于随机梯度下降的逻辑回归算法模型，通过再不同知识库中的实验，

评估了不同类型特征组合的情况下，我们结合关系路径特征和实体属性特征相比传统的关系路径特征计算问题，在分类预测、排序预测评估指标下的优势。

在接下来的章节中，本研究将通过构建不同的机器学习模型，比较逻辑回归分类算法、基于学习排序的树模型等来评估知识库补全算法应该如何选择预测模型。考虑到学习排序算法正负例会影响知识库补全效果，我们研究了不同比例的正负三元组进行模型构建，分析不同比例的实体对模型补全效果。

## 第4章 基于学习排序模型的知识库补全

本部分介绍基于学习排序模型的知识库补全算法，考虑到主流的路径排序算法采用基于逻辑回归、支持向量机等分类算法。然而知识库补全问题从理论上可以看做一类实体对的排序问题，基于此，本部分研究考虑使用学习排序算法进行模型构建，并通过实验比较不同比例正负例实体对情况下学习排序模型和分类模型的预测效果，研究分析了不同关系下实体对模型预测效果。

### 4.1 问题引入

当前的知识库基于分类打分模型进行知识库补全有很大不足。一是知识库中正负实体对比例差别很大，对于每个在知识库中实际存在的三元组正实例，可能有成千上万条不存在的三元组负实例相对应，如三元组（北京师范大学，位于，中国）这个三元组在知识库中实际存在，是一条正实例，而（北京师范大学，位于，美国）和（北京师范大学，位于，日本）等上百条负实例与之对应，如何解决正负实体对不匹配的问题很关键，正负实体对比例悬殊，关系预测中仅靠打分是不够的，很多基于排序的算法可以解决知识库中正负实例不匹配的问题。二是相关的方法都是通过评价三元组得分高低来预测结果的，是假设正例三元组中一个实体对之间的关系仅有一种，或者假设一个实体和关系的对应尾实体仅有一个，比如（美国，有总统，？）这个三元组中，尾实体通常被假设只有一个正确结果，而在现实的知识库中存在一对多的关系如：（美国，有总统，川普）和（美国，有总统，奥巴马）在知识库中都是实际存在的，并且通常来说，我们希望美国，有总统，川普）这个三元组比（美国，有总统，奥巴马）排序更高，这就需要我们不仅仅考虑实体对的得分问题，还要研究实体对相对排序情况，保证正实体对排序总在负实体对之前。而当前一些知识库补全算法并未考虑候选实体对的顺序对预测结果的影响，也不关注候选实体的秩序关系，而基于学习排序的算法可以解决候选实体的秩序关系。

本部分研究基于抽取知识库中的关系路径特征进行模型预测。模型预测主要分为两种方法：分类模型和排序模型。传统的基于逻辑回归的分类算法对正负实体对进行分类模型学习，利用二分类算法学习正负实体对的得分，依据得分大小将正负实体对进行划分，从而进行关系预测。排序模型通过构建排序算法，对一组正负实体对进行学习排序，通过学习实体对的秩序排名，期望能将正实体对排在负实体对之前，这样既可获得



比传统分类回归算法更好的结果。本研究构建了基于树排序方法的知识库补全算法和基于神经网络排序的知识库补全算法。

本部分研究构建了一种新的知识库补全的模型。其知识库补全的技术关键点在于：

(1) 给定一个输入关系，将输入关系切分为训练集（包括验证集）和测试集；(2) 对于训练集和测试集中的正实体对基于局部封闭世界的假设，生成对应比例的负实体对；(3) 将正负实体对构成的集合在由知识库构成的图中抽取特征，基于随机游走的算法抽取从头实体到尾实体的路径类型作为关系预测的特征；(4) 构建学习排序模型，将生成的正负实体对进行模型训练，并将训练获得的模型为新的关系特征提供预测；(5) 基于MAP和MRR进行模型评价，调整并改进模型的参数，更新模型获得更好的预测效果。考虑到结合关系路径和结合实体属性特征进行知识库补全算法的实现难度较大，本部分研究仅仅使用了关系路径特征作为知识库补全特征。除此之外，本研究将为了能更好分析正负例不匹配问题对于知识库补全问题的影响，本研究只使用MAP和MRR进行模型评估，对于不同比例的正负实体对进行排序模型效果预测，并未使用基于分类的评价函数。

## 4.2 模型构建

本部分包括如何获取知识库补全中的关系路径特征向量，并选择不同的模型预测方法构建模型，包括对每个关系构建不同比例的实体对集合、构建基于逻辑回归的知识库补全模型、基于学习排序的知识库补全预测模型。

### 4.2.1 实体关系特征计算

通过给定知识库中的某个 $r$ ，我们获取这个关系对应的实体对集合 $I_s$ ,

$$I_s = \{(h_j, t_j) | < h_j, t_j > \in KB\} \quad (4-1)$$

并对从这个知识库所有的关系和实体组成的图中抽取连接头尾实体的关系路径，由于连接头尾实体对之间的路径数量很大，通常需要限定关系路径的长度，并采用随机游走算法计算选择从头实体到尾实体的关系路径类型。对于能连接从头实体到尾实体的关系路径，我们记录这个关系路径的类型，作为模型预测的特征类型。对于知识库中的每一个实体对，我们计算实体对是否具有某些关系类型，从而获得实体对的关系路径特征向量，作为预测模型的特征向量。对于一个关系的下所有实体对，计算获得这些实体对的特征向量组成特征矩阵，从而进行下一步的模型预测。

#### 4.2.2 基于逻辑回归的补全模型

本研究的特征计算模块采用随机游走算法抽取给定关系下的路径类型特征。对于一个给定的关系下的实体对，我们需要抽取在知识库中，能在有限路径长度下，从头实体到达尾实体的路径信息。对于如图1知识库中实体对(北京师范大学，中国)，我们在知识库中可以抽取路径类型信息如：(校长-生活在-)、(位于-位于-)、(有大学-位于-位于-)、(校长-出生-相邻-位于)等关系路径信息，并将这些关系路径信息作为学习排序的特征。通常，路径长度被限制在2-6跳之间，过高则关系路径太多，计算复杂度太高，而小于2跳的关系路径则使得获得关系路径类型信息太少，不能有效提供特征。

对于抽取到的关系路径特征，传统方法构建了一个分类器模型，学习每个关系和这个关系包含的实体对集合，将预测关系问题转化成一个分类预测问题，学习每个实体对具有某种关系的概率。 $E_r$  (4-2)表示给定关系 $r$ 下的实体对集合，包括头实体、尾实体和头尾实体对是否存在的标签（label）。

$$E_r = \{(h_i, t_j), y_i\}_{i=1}^N \quad (4-2)$$

表示关系 $r$ 所有的实体对集合，其中 $y_i \in \{0, 1\}$ ，其中0表示负实体对，即知识库中并不是实际存在的三元组，1表示正实体对，表示在知识库中实际存在的实体对，通过对知识库中的实体对进行分类器模型学习，我们可以获得测试集中实体对的打分情况。通常这个分类器采用逻辑回归算法进行模型的训练。具体来说，对于每个关系的实体对，传统模型采用逻辑回归学习得到的关系路径特征向量 $V_r$ 和实体属性特征向量 $V_l$ 。并定义了如下的逻辑回归函数，对每个关系下的实体对集合进行评价打分。 $f(v, w)$  (4-3)表示给定关系下的逻辑回归打分函数。

$$f(v, w) = \frac{1}{1 + e^{w(V_r \oplus V_l)}} \quad (4-3)$$

其中 $w$ 表示关系路径特征和实体属性特征的学习权重参数。经典论文采用对数似然函数进行最大似然估计，并通过随机梯度下降算法学习这些模型的参数，除此之外，还考虑到模型的过拟合和参数正则化表示，本论文定义了如下的学习目标函数 $L_r$  (4-4)：

$$L_r = \frac{1}{N} \sum_{i=1}^N \{(y_i \log(f(v, w_r)) + (1 - y_i) \log(1 - f(v, w_r)))\} + \alpha \|w_r\| + \beta \|w_r^2\| \quad (4-4)$$

其中， $L_r$ 表示给定关系 $r$ 的目标函数， $\alpha$ 和 $\beta$ 分别是 $l_1$ 和 $l_2$ 正则化惩罚项的权重，对于每个关系我们采用随机梯度下降算法使得整个训练集对数损失最小，同时结合 $l_1$ 和 $l_2$ 防止过拟合。最终我们可以学习得到每个关系下的关系路径特征和实体属性特征的权重。对于学

习到的模型，我们调整不同的参数和梯度下降算法改进模型的效果。

### 4.2.3 基于学习排序的树模型

对于中抽取的关系路径特征，我们需要构建基于学习排序算法的知识库补全模型进行关系训练和关系预测。传统的路径排序算法中，当通过随机游走计算获得路径类型信息，并获得这些关系路径的值后，再通过基于分类或者回归的算法，计算的到每个实体对的打分值。如基于逻辑回归模型的方法，通过构建的特征向量，对每个关系中的实体对进行打分，并将打分结果通过逻辑回归函数（Sigmoid）正则化，最终使得模型的预测结果在 $[0,1]$ 之间，对于打分值越大的实体对，其具有这种关系的概率越大，反之具有这种关系的概率越小。这种模型实质上是将打分高的实体对排在打分低的实体对之前，表示更可能是实际存在的实体对，也可以看做是一种单文档排序Pointwise的实体对排序模型。而本论文中的研究不仅仅考虑实体对的打分高低，更关注实体对之间的排序关系，考虑一个关系下的一组实体对，其中部分实体对是正例，部分实体对是负例，在进行模型预测中，基于实体对排序的损失函数，其目标是期望正实体对总需要排序在负实体对前面，这样就能保证在预测的候选实体对中，排在前面的实体对是正确结果，这种对一组实体对进行整体的目标函数优化可以称为文档序列排序Listwise的排序模型。基于Listwise的排序模型通常构建的目标函数是非凹函数，难以进行模型优化，通常我们也可以考虑实体对和实体对之间的关系，正实体对的排序总在负实体对之前，则我们可以认为正实体对的打分比负实体对打分应该更高，基于正负实体对实体对构建的损失函数可以看做是基于Pairwise构建的目标函数，这是学习排序算法中常见的模型，很多基于支持向量机、树方法、神经网络排序算法的模型都可以基于Pairwise的学习排序模型构建。

具体来说，本论文中我们采用一种基于树模型Pairwise的学习排序算法进行知识库补全，除了基于树模型的学习排序，这种学习排序模型可以拓展为其他任何基于知识向量机的、基于神经网络的学习排序模型。同时，考虑到模型构建的复杂度和代码实现难度，本论文采用成对排序Pairwise学习排序模型，而其他基于Pointwise、Listwise的学习排序模型也可以进行知识库补全预测，传统的路径排序算法也可被看做是一种简单的Pointwise的学习排序模型。本研究通过学习正负实体对之间的Pairwise损失函数，直接优化MAP训练损失函数来进行模型参数更新，从而获得更好的模型预测结果。我们使用基于LambdaMART树模型的学习排序算法，对于一个给定的关系 $r$ ，定义目标函

数 $F(x_i|w, c)$  (4-5):

$$F(x_i|w, c) = \sum_{i=1}^K \alpha_i \pi(f_i) + \sum_{i=1}^N l(f_i(x), f'_i(x)) + \frac{C}{2} WW^T \quad (4-5)$$

其中第一项中的 $\sum_{i=1}^K \alpha_i \pi(f_i)$ 是描述树复杂度的函数，总共有K个树进行模型训练。而第二项中 $l(f_i(x), f'_i(x))$ 是模型的训练误差函数，其中 $f_i(x)$ 是每个实体对的实际分数，而 $f'_i(x)$ 是通过模型学习得到的预测值，共训练了N轮，训练误差函数可以根据实际需要改变，常见的训练误差函数可以选择MAP、AUC、NDCG等不同排序指标，通常学习排序中MAP评价指标最为常见。考虑到我们目标函数是Pairwise损失函数最小化，我们也选择MAP作为训练损失函数进行模型训练。第三部分 $\frac{C}{2} WW^T$ 是模型的惩罚项，是防止模型在训练数据中过拟合的L2惩罚函数。

基于树模型的构建函数相比于传统的逻辑回归等分类算法，树模型有更好的模型优化能力，通过构建单个树的分类回归模型，并将多个树模型进行叠加进行梯度下降算法，进一步优化模型的损失函数，通过重新将模型的损失函数选择为MAP、NDCG等排序预测指标，树模型能直接对这些指标进行预测。树模型能处理稀疏矩阵，同时能结合不同类型的关系路径特征进行模型预测。本部分选择基于学习排序的树模型进行模型构建，主要有以下考虑：（1）树模型相比于逻辑回归，更能有效的学习模型的真实分布，通过多个弱模型进行叠加改进参数，能保障树模型的损失函数下降结果更好。（2）相比于传统逻辑回归，树模型有更好的模型泛化效果，更能有效进行模型预测，相比之下逻辑回归更容易出现模型的过拟合。（3）树模型在处理稀疏向量，正负例比例差别很大的数据时，无论何种情况都有很好的模型优化性能。除此之外，本研究的学习排序模型也可以选择其他的线性的、非线性的、基于神经网络的各种模型。都能获得好的模型预测效果。

### 4.3 实验结果

本研究构建了一个面向YAGO2的知识库补全实例,YAGO是一个从维基百科上抽取的、包含地理名词、WordNet等数据的知识库。而YAGO2是YAGO的一个实例，当前YAGO2包括超过千万的实体和超过1.2亿的实体知识，我们使用了其中实体的关系型三元组共有4,484,914条、37种关系类型。对于YAGO2知识库的实验数据的构建，我们按照37种不同的关系类型，将知识库切分为37份不同关系的实体对集合。对于每个关系，首先基于知识库中的三元组抽取正实体对，对于每个由（头实体，尾实体）构成实体对，基于局部封闭世界假设，使用当前关系中的其他实体随机替换生成n个对应

的负实体对，其中 $\frac{n}{2}$ 个随机替换头实体， $\frac{n}{2}$ 个随机替换尾实体。生成正负实体对后，我们按照4: 1的比例将这些实体对切分为训练集合和测试集合，这样就完成了YAGO知识库中正负实体对、训练测试数据集的生成。同时，为了能研究不同比例的正负实体对下，模型预测的效果，我们设定了n的不同数值。本研究中分析了正负例三元组在比例为1: 4和1: 10下的模型预测效果，研究不同比例下模型的预测结果差别。

表 10 基于学习排序的YAGO知识库补全评价结果

|     | 1: 4    |         |        |        | 1:10    |         |        |        |
|-----|---------|---------|--------|--------|---------|---------|--------|--------|
|     | rankSFE | rankPRA | SFE    | PRA    | rankSFE | rankPRA | SFE    | PRA    |
| MAP | 80.68%  | 80.06%  | 70.31% | 69.35% | 47.33%  | 46.91%  | 30.83% | 31.27% |
| MRR | 100.00% | 100.00% | 97.83% | 97.83% | 91.95%  | 89.62%  | 90.48% | 92.14% |

**知识库综合关系评估**我们计算了基于YAGO2的总体MAP和MRR进行模型评价的结果。如表10可以看到在正负比例为1: 4和1: 10的不同比例下，模型的MAP预测指标变化较大，而MRR指标都超过80%。具体分析来说基于学习排序的知识库补全技术相比传统的路径排序算法在MAP上有很大的提升，在正负实体对比例为1: 10的情况下，基于学习排序的算法相比传统的算法在YAGO2数据集上有近50%的效果提升；而四种方法在MRR指标上效果相当，基于学习排序算法的MRR并未比传统路径排序算法有显著下降。在正负实体对比例为1: 4的情况下，基于学习排序算法相比于传统的在YAGO数据集上的逻辑回归算法，基于MAP预测的准确率提升超过10%，效果提升近15%。同时，对比不同比例的正负实体对分析可以发现，当负例三元组出现较多时，对预测结果有较大的影响，正负三元组比例超过1:10时，模型预测的MAP指标不足50%,这表明在进行学习排序模型，选择合适的正负例三元组作为预测模型的候选实体对也很重要，对于一个高效合理的模型，如何选择恰当的候选实体作为预测模型，也是知识库补全后续研究中的重要问题，本研究基于局部封闭世界假设，随机选择若干比例的候选正负实体对进行模型构建，而在后续的研究中，如何构建模型选择合适的候选正负实体对是十分必要的。

**知识库1: 10单个关系评估**我们详细分析了37种YAGO2中关系的MAP指标，并在表 11展示了部分关系在正负例为1: 10的情况下的MAP值。分析可以发现，大部分关系采用学习排序算法后，预测结果有较大的提高，而在不同的关系类型预测中，MAP差别较大，如playFor、dealsWith和isConnectedTo等关系预测MAP有较大的提高，而在isInterestedIn等关系中关系预测提升较差。总体来说，有超过30种关系的MAP预测获得了显著的提升，只有不到5种关系MAP预测结果并未有统计上的显著提升，这说明了基于学习排序算法的知识库补全技术相比

表 11 正负例1: 10的学习排序算法部分关系MAP值

| relation       | PRA-10 | SFE-10 | rankPRA-10 | rankSFE-10 |
|----------------|--------|--------|------------|------------|
| actedIn        | 33.7%  | 34.9%  | 62.2%      | 62.5%      |
| created        | 25.2%  | 25.3%  | 30.8%      | 31.2%      |
| dealsWith      | 17.2%  | 14.1%  | 12.6%      | 15.7%      |
| graduatedFrom  | 26.4%  | 27.2%  | 56.0%      | 57.2%      |
| hasCapital     | 56.3%  | 60.1%  | 72.7%      | 73.0%      |
| hasChild       | 50.0%  | 50.7%  | 66.7%      | 67.5%      |
| influences     | 29.4%  | 29.3%  | 57.7%      | 58.3%      |
| isAffiliatedTo | 63.6%  | 65.3%  | 78.1%      | 78.4%      |
| playsFor       | 65.3%  | 66.0%  | 80.2%      | 80.3%      |
| wasBornIn      | 36.6%  | 37.4%  | 60.5%      | 60.7%      |
| worksAt        | 23.4%  | 22.8%  | 50.2%      | 50.1%      |
| wroteMusicFor  | 34.8%  | 36.2%  | 61.4%      | 61.6%      |

传统的分类回归打分模型对于大多数知识库预测的关系，在模型效果上有非常大的效果提升。我们进一步分析了不同export、import等关系预测中效果较差的模型，通过分析这个模型构建中的三元组可以发现，在export和import等关系中，经常出现三元组如（China, export, Amercia）,（China, export, Europe）、（America, export,China）等多重三元组，如图5所示，这些类型的三元组可以看做是一种多对多的关系，在进行这类知识库补全预测模型构建中，很难利用国家和国家之间的关系路径、实体属性等信息进行模型构建，我们需要找到更好的模型预测方法，加入更多的背景知识进行知识库补全预测。

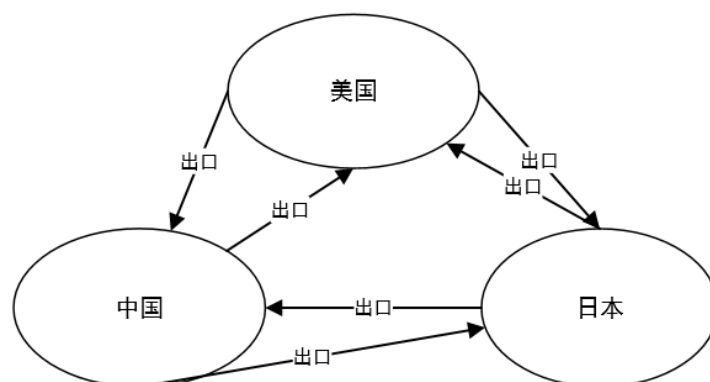


图 5 多对多实体关系的知识库

**知识库1：4**单个关系评估除了上述的1：10的知识库补全预测模型中MAP预测结果，表12展示了在知识库补全正负例为1：4的情况下，部分关系中知识库补全MAP预测结果。相比于1：10的正负例实体对，基于1：4的正负例实体对能显著增加MAP预测结果的准确率，总体的平均准确率MAP值从47%提升至80%，这说明在选择候选实体对作为学习排序的数据时，选择实体对的比例也很关键，对于部分关系，如worksAt、dealsWith等，这些关系提升超过100%，我们分析了相关关系下的三元组发现，这些关系下的三元组存在一对多，多对多的情况，这种情况下减少模型的候选实体对比例，能有效的提高模型的效果。通过对比不同正负比例的实体对预测情况，说明在进行知识库补全的学习排序模型构建中，如何选择候选实体对，构建合理有效的知识库补全正负实体对，也会影响知识库补全模型的结果好坏。这和很多推荐系统算法类似，如何从百万级别的数据中选择若干候选实体对给用户推荐出来，也是具有类似的过程，通常这在推荐系统中被分为候选集生成和候选集排序两个步骤。与这种“候选-排序”两步排序算法类似，如何从上万个负实体对中选择合适的候选负实体对，组成训练模型进行排序算法的训练，是知识库补全中即为重要的一步。后续研究可以借鉴表示学习如神经网络翻译模型等实现更好的知识库补全候选实体对生成。

表 12 正负例1：4的学习排序算法部分关系MAP值

|               | rankSFE-4 | rankPRA-4 | SFE-4 | PRA-4 |
|---------------|-----------|-----------|-------|-------|
| actedIn       | 79.2%     | 76.9%     | 74.5% | 69.4% |
| dealsWith     | 53.4%     | 55.7%     | 42.3% | 40.7% |
| diedIn        | 91.1%     | 91.3%     | 82.3% | 82.2% |
| graduatedFrom | 65.6%     | 66.9%     | 46.3% | 39.8% |
| hasChild      | 95.8%     | 95.8%     | 91.6% | 91.6% |
| hasNeighbor   | 76.6%     | 78.1%     | 69.0% | 75.0% |
| isCitizenOf   | 84.0%     | 83.9%     | 65.0% | 63.4% |
| isConnectedTo | 69.4%     | 69.3%     | 38.2% | 35.9% |
| isLeaderOf    | 94.4%     | 84.9%     | 71.7% | 79.0% |
| isLocatedIn   | 95.3%     | 94.9%     | 86.1% | 85.6% |
| isMarriedTo   | 88.4%     | 87.5%     | 81.1% | 78.8% |
| livesIn       | 80.8%     | 78.9%     | 67.2% | 68.9% |
| wasBornIn     | 95.9%     | 95.6%     | 85.4% | 84.0% |
| worksAt       | 50.5%     | 46.5%     | 42.7% | 38.4% |

本部分的实验结果有如下结论：（1）基于学习排序的知识库补全算法相比于传统的逻辑回归算法，能大幅提升模型预测的MAP指标，这在知识库补全算法中极为重要；

(2) 基于学习排序算法对于大部分关系都能产生较大的预测结果提升, 而对于一些频繁出现多对多关系的正负实体对中, 候选实体对选择较为困难, 生成正负例三元组难以区分, 如何构造有表达力的特征向量仍然是一个有挑战性的工作; (3) 不同正负比例的三元组预测模型效果差别很大, 这表明除了选择合理有效的排序算法之外, 如何构建模型, 生成合理有效的候选正负实体对也是十分重要的。后续可以借鉴推荐系统中的候选集合生成算法, 优化知识库补全模型中的实体对生成, 也可以考虑集成表示学习中的实体向量表示, 学习获得合理的候选实体对集合。

#### 4.4 本章工作总结

本章中, 我们主要比较了基于逻辑回归分类模型和基于学习排序的树模型, 通过优化知识库补全中的模型构建, 我们使用基于成对排序 (Pairwise) 的树排序模型, 更好的提升了知识库补全中实体对之间的排序问题, 将传统的基于逻辑回归的分类问题转化为基于学习排序的模型优化问题。

本研究通过对YAGO知识库进行试验发现, 不同知识库补全模型中, 基于排序的模型优化算法可以大幅提高正实体对在所有实体对中的秩序, 从而能改变原有的知识库补全优化中仅仅对实体对进行打分的模式, 能够获得更好的预测效果。对于多数关系而言, 基于学习排序的树模型算法效果明显优于传统的逻辑回归分类算法。



## 第5章 总结和展望

随着知识库在信息检索、手机助手、对话系统、人工智能等各个领域的广泛发展，知识库的完整性和真实性越来越重要，而如何对知识库进行补全，验证实体和实体之间关系的真实性，都是提高知识库可用性的重要问题。本研究从基于路径排序算法的知识库补全进行研究，选择实体和实体之间的关系路径、头尾实体中不同的属性值进行组合、并构建模型、选择合理有效的目标函数进行知识库补全。同时，本研究也结合了一些新的深度神经网络知识，将不同的特征进行深度融合，获得了更好的知识库补全效果。然而，基于符号逻辑的路径排序算法，补全效果和知识库图的拓扑结构有关，如何能在稀疏的图中进行知识库补全，如何更好的结合关系路径特征和实体属性特征以及基于表示学习获得的实体、关系向量表示特征，是未来进行知识库补全工作的重点。

### 5.1 论文工作总结

知识库补全算法是知识库构建和知识库应用的重要桥梁，基于信息抽取获得的三元组，需要通过知识库补全技术提高知识库的可用性，进而才能提供各种有效的知识库应用系统。本论文从知识库补全的特征构建开始，结合实体属性特征和关系路径特征，将不同类型的关系路径特征，不同类别的实体属性特征结合，能提高知识库补全的模型的可预测性，可拓展性。此外，本研究拓展了原有的知识库补全模型，将原来基于Pointwise的二分类模型，拓展为基于学习排序算法的Pairwise模型，提供了更多的模型优化方法，作为原有基于逻辑回归中的对数损失损失函数，或者基于支持向量机的间隔损失函数的补充。这些不同的损失函数作为机器学习的目标，在进行知识库补全工作中，可以依据具体任务不同而选择合适的模型进行预测。

本研究的实验发现了很多重要的结论，能为后续的知识库补全工作提供研究思路。

(1) 知识库中的关系路径是稀疏的，大部分实体和实体之间的关系较少，少数实体和实体之间的关系数量较多，将稀疏的关系路径处理好来预测实体和实体之间的关系，是知识库补全的重要步骤。(2) 基于符号逻辑和关系路径进行知识库补全，预测效果和图的拓扑结构密切相关。对于一些关系的实体，这些实体和其他实体之间关系很少，是一类比较孤立的顶点。这导致了基于随机游走过程中，部分实体对很难抽取有表达力的关系路径特征，同样对于部分实体对，他们的实体属性特征也很少，这样会导致知识库补全算法很难进行模型预测。如何结合能学习稀疏图中的表示实体向量模型，能有效将这些

实体关系模型进行预测。(3) 知识库中的实体属性特征和关系路径特征能相互影响, 对于部分关系路径表达力差的特征, 好的实体关系路径特征能进行模型惩罚, 减少这样关系特征的权重, 而对于关系路径表达力强的特征, 在加入实体属性特征后能增加这些特征的权重。(4) 除了传统的基于逻辑回归、支持向量机算法, 通过优化间隔损失函数 (hinge loss), 对数损失函数 (log loss) 学习二分类模型, 本研究展示了其他学习模型的优势。通过基于成对排序 (Pairwise) 的模型优化方法, 本研究能将一组正负实体对作为一个整体序列进行模型优化, 学习这些正负实体对的秩序, 从而将知识库补全作为一个排序结果进行训练。这种算法的另外一个优势就是能避免模型的正负例不平等问題, 传统的算法随机选择若干比例的负例进行模型优化, 而基于排序的算法可以将三元组中多种正例三元组排在负例三元组之前, 提高模型的可用性。(5) 基于学习排序的知识库补全算法, 在进行知识库补全过程中, 通常需要进行两个步骤, 生成候选实体对、对候选实体对进行排序。选择如何构建合理的实体对比例进行学习排序模型构建, 也会影响模型的效果, 后续需要结合表示学习等领域知识, 对知识库中的候选实体对进行选择, 生成候选实体对集合。

## 5.2 未来研究展望

尽管本研究基于关系路径和符号逻辑算法, 拓展了关系路径和实体属性等不同类型的特征, 增加了模型预测的优化方法, 优化了目标函数, 但是未来的研究还是有很多重要的工作, 如何结合符号逻辑算法和表示学习算法, 如何利用表示学习获得候选实体对, 如何将表示学习中的优化目标函数和符号逻辑中的优化目标函数相互统一, 都是研究的重要工作。

从知识库补全的关系路径开始拓展, 本研究在路径排序算法中加入子图特征路径, 拓展知识库的关系路径特征。同时也考虑了实体属性特征在知识库补全中的重要应用, 以及如何结合关系路径和实体属性特征, 从而增加模型的可预测性。另外, 本研究也拓展了不同的优化目标函数, 从实验中发现结合Pairwise的知识库补全算法可以有效的优化一组正负实体对集合, 很好的解决知识库中正负实例对不平衡的问题。而其他基于学习排序的算法模型构建知识库补全试验也是很值得尝试的, 结合成序列排序 (Listwise) 模型, 结合神经网络排序 (ranknet) 模型等方法, 都可能有效提升模型的预测效果, 在实际的工程项目中也可以进行试验, 根据不同的场景选择合适的学习排序模型。

本研究未来希望能关注如何有效结合关系路径特征、实体属性特征、实体向量特征, 不仅仅通过不同的算法抽取关系路径特征, 转换不同类型实体属性, 而且构建了深

度学习模型，将知识库补全算法的不同特征进行融合，更好提高了机器学习模型的效果。通过这些不同的实体、关系预测统一构建一个合理有效的目标函数，优化目标函数获得更好的模型系统，从而将知识库补全算法进行统一优化、改进，从而模型预测变得更加智能。

本研究未来关注的另外一个重要如何结合关系路径特征、头尾实体属性特征、实体向量特征预测实体的一些属性值。尽管很多知识库中，如何预测实体和实体之间的关系是一个重要的问题，但是我们在实验中也非常明显的发现，很多实体的属性值都是缺失的，如很多重要人物的出生日期、很多地点的地理位置等信息存在明显的缺失或者错误，如何从实体、实体属性类型预测实体属性值，对于解决实体属性值的缺失或错误问题，有着重要的影响。这个问题对于未来的知识库补全、知识库拓展和应用也都是十分重要的问题，构建合理而统一的学习模型，学习实体对之间的关系，学习实体的属性特征，对于知识库补全领域的未来研究方向，都是需要进一步思考和探索的。

除了在理论和算法方面的研究工作之外，进行关系路径的知识库补全算法的一个不足时对于实验机器要求较高。通常需要耗费大量的内存来构建图，查找图中的关系路径，如何能降低关系路径搜索中对于机器硬件的要求也是一个可以优化的方向，如何能有更高效，内存使用更合理的随机游走路径搜索算法，也是很多领域关注的热点问题。

## 参考文献

- [1] Xin Dong, Evgeniy Gabrilovich, Jeremy Heitz, et al. From Data Fusion to Knowledge Fusion[J]. PVLDB, 2014, 7:881–892.
- [2] T. Mitchell, W. Cohen, E. Hruschka, et al. Never-Ending Learning[C]. Proceedings of Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15), 2015.
- [3] Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum. Yago: A Core of Semantic Knowledge[C]. Proceedings of Proceedings of the 16th International Conference on World Wide Web, New York, NY, USA: ACM, 2007. 697–706.
- [4] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]. Proceedings of SIGMOD Conference, 2008.
- [5] Christian Bizer, Jens Lehmann, Georgi Kobilarov, et al. DBpedia - A Crystallization Point for the Web of Data[J]. Web Semant., 2009, 7(3):154–165.
- [6] Yanbin Gao, Gang Zhao. Knowledge-Based Information Extraction: A Case Study of Recognizing Emails of Nigerian Frauds[C]. Proceedings of NLDB, 2005.
- [7] Rui Meng, Lei Chen, Yongxin Tong, et al. Knowledge Base Semantic Integration Using Crowdsourcing[J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 29:1087–1100.
- [8] M M Shenassa, K. Khakpour. Knowledge base expert system for tuning PID controllers using wireless technology[J]. 2008 International Conference on Computer and Communication Engineering, 2008. 310–313.
- [9] Michel Dumontier, Alison Callahan, Jose Cruz-Toledo, et al. Bio2RDF Release 3: A Larger Connected Network of Linked Data for the Life Sciences[C]. Proceedings of Proceedings of the 2014 International Conference on Posters; Demonstrations Track - Volume 1272, Aachen, Germany, Germany: CEUR-WS.org, 2014. 401–404.
- [10] Xiang Li, Aynaz Taheri, Lifu Tu, et al. Commonsense Knowledge Base Completion[C]. Proceedings of ACL, 2016.
- [11] Linyuan Lu, Tao Zhou. Link Prediction in Complex Networks: A Survey[J]. CoRR, 2010, abs/1010.0725.
- [12] Gerhard Weikum, Martin Theobald. From information to knowledge: harvesting entities and relationships from web sources[C]. Proceedings of PODS, 2010.

- [13] Denny Vrandečić, Markus Krötzsch. Wikidata: a free collaborative knowledgebase[J]. Commun. ACM, 2014, 57:78–85.
- [14] Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum. Yago: a core of semantic knowledge[C]. Proceedings of WWW, 2007.
- [15] Gjergji Kasneci, Maya Ramanath, Fabian M. Suchanek, et al. The YAGO-NAGA approach to knowledge discovery[J]. SIGMOD Record, 2008, 37:41–47.
- [16] Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum. YAGO: A Large Ontology from Wikipedia and WordNet[J]. J. Web Sem., 2008, 6:203–217.
- [17] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, et al. YAGO2: exploring and querying world knowledge in time, space, context, and many languages[C]. Proceedings of WWW, 2011.
- [18] David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, et al. Building Watson: An Overview of the DeepQA Project[J]. AI Magazine, 2010, 31:59–79.
- [19] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, et al. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia[J]. Artif. Intell., 2013, 194:28–61.
- [20] Farzaneh Mahdisoltani, Joanna Biega, Fabian M. Suchanek. YAGO3: A Knowledge Base from Multilingual Wikipedias[C]. Proceedings of CIDR, 2015.
- [21] Xuchen Yao, Benjamin Van Durme. Information Extraction over Structured Data: Question Answering with Freebase[C]. Proceedings of ACL, 2014.
- [22] Xuchen Yao. Lean Question Answering over Freebase from Scratch[C]. Proceedings of HLT-NAACL, 2015.
- [23] Xuchen Yao, Jonathan Berant, Benjamin Van Durme. Freebase QA: Information Extraction or Semantic Parsing?[C]. 2014.
- [24] Zhicheng Zheng, Xiance Si, Fangtao Li, et al. Entity Disambiguation with Freebase[J]. 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, 2012, 1:82–89.
- [25] Xin Dong, Evgeniy Gabrilovich, Jeremy Heitz, et al. Knowledge vault: a web-scale approach to probabilistic knowledge fusion[C]. Proceedings of KDD, 2014.
- [26] Ulf HERMJAKOB, Eduard H. HOVY, Chin-Yew LIN. Knowledge-Based Question Answering[C]. 2000.
- [27] Min-Chul Yang, Nan Duan, Ming Zhou, et al. Joint Relational Embeddings for Knowledge-based Question Answering[C]. Proceedings of EMNLP, 2014.
- [28] Wen tau Yih, Matthew Richardson, Christopher Meek, et al. The Value of Semantic Parse Labeling for Knowledge Base Question Answering[C]. Proceedings of ACL, 2016.

- [29] Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, et al. Improved Neural Relation Detection for Knowledge Base Question Answering[C]. Proceedings of ACL, 2017.
- [30] Organisers Key-Sun, Choi, Jin-Dong Kim, et al. Open Knowledge Base and Question Answering Workshop (OKBQA)[C]. 2016.
- [31] Babita Pandey, R. B. Mishra. Knowledge and intelligent computing system in medicine[J]. Computers in biology and medicine, 2009, 39 3:215–30.
- [32] Ethan J Halpern, Eric L Gingold, Hugh White, et al. Evaluation of coronary artery image quality with knowledge-based iterative model reconstruction.[J]. Academic radiology, 2014, 21 6:805–11.
- [33] Zhenyu Liu, Wesley W. Chu. Knowledge-based query expansion to support scenario-specific retrieval of medical free text[J]. Information Retrieval, 2005, 10:173–202.
- [34] Robin D. Burke. Knowledge-based recommender systems[C]. 2000.
- [35] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, et al. Translating Embeddings for Modeling Multi-relational Data[C]. In: C. J. C. Burges, L. Bottou, M. Welling, et al., (eds.). Proceedings of Advances in Neural Information Processing Systems 26. Curran Associates, Inc., 2013: 2787–2795.
- [36] Zhen Wang, Jianwen Zhang, Jianlin Feng, et al. Knowledge Graph Embedding by Translating on Hyperplanes[C]. Proceedings of AAAI, 2014.
- [37] Wenhao Huang, Ge Li, Zhi Jin. Improved Knowledge Base Completion by the Path-Augmented TransR Model[C]. Proceedings of KSEM, 2017.
- [38] Maximilian Nickel, Volker Tresp, Hans-Peter Kriegel. A Three-Way Model for Collective Learning on Multi-Relational Data[J]. 28th International Conference on Machine Learning, 2011. 809—816.
- [39] Steffen Rendle. Factorization Machines[J]. 2010 IEEE International Conference on Data Mining, 2010. 995–1000.
- [40] Steffen Rendle. Factorization Machines with libFM[J]. ACM TIST, 2012, 3:57:1–57:22.
- [41] Embedding Large Subgraphs into Dense Graphs[C]. 2009.
- [42] Xin Luna Dong, Evgeniy Gabrilovich, Jeremy Heitz, et al. Knowledge Vault : A Web-Scale Approach to Probabilistic Knowledge Fusion[J]. 2014. 601–610.
- [43] Yankai Lin, Zhiyuan Liu, Maosong Sun, et al. Learning Entity and Relation Embeddings for Knowledge Graph Completion[C]. Proceedings of AAAI, 2015.
- [44] Baoxu Shi, Tim Weninger. ProjE: Embedding Projection for Knowledge Graph Completion[C]. Proceedings of AAAI, 2017.

- [45] Luis Galárraga, Christina Teflioudi, Katja Hose, et al. AMIE: association rule mining under incomplete evidence in ontological knowledge bases[C]. Proceedings of WWW, 2013.
- [46] Ni Lao, William W. Cohen. Relational retrieval using a combination of path-constrained random walks[J]. Machine Learning, 2010, 81(1):53–67.
- [47] Matt Gardner, Tom Mitchell. Efficient and Expressive Knowledge Base Completion Using Subgraph Feature Extraction[J]. Proceedings of EMNLP, 2015, (September):1488–1498.
- [48] J. Ross Quinlan, R. Mike Cameron-Jones. FOIL: A Midterm Report[C]. Proceedings of ECML, 1993.
- [49] MICHAEL J. PAZZANI, CLIFFORD A. BRUNK. Detecting and correcting errors in rule-based expert systems: an integration of empirical and explanation- based learning[C]. 1991.
- [50] Stefan Schoenmackers, Oren Etzioni, Daniel S. Weld, et al. Learning First-order Horn Clauses from Web Text[C]. Proceedings of Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. 1088–1098.
- [51] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, et al. AMIE: Association Rule Mining Under Incomplete Evidence in Ontological Knowledge Bases[C]. Proceedings of Proceedings of the 22Nd International Conference on World Wide Web, New York, NY, USA: ACM, 2013. 413–422.
- [52] Luis Galárraga, Christina Teflioudi, Katja Hose, et al. Fast rule mining in ontological knowledge bases with AMIE[J]. The VLDB Journal, 2015, 24:707–730.
- [53] Zhichun Wang, Juan-Zi Li. RDF2Rules: Learning Rules from RDF Knowledge Bases by Mining Frequent Predicate Cycles[J]. CoRR, 2015, abs/1512.07734.
- [54] Ni Lao, Einat Minkov, William W. Cohen. Learning Relational Features with Backward Random Walks[C]. Proceedings of ACL, 2015.
- [55] Matt Gardner, Partha P. Talukdar, Jayant Krishnamurthy, et al. Incorporating Vector Space Similarity in Random Walk Inference over Knowledge Bases[J]. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014. 397–406.
- [56] Quan Wang, Jing Liu, Yuanfei Luo, et al. Knowledge base completion via coupled path ranking[C]. Proceedings of Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016. 1308–1318.
- [57] Robert West, Evgeniy Gabrilovich, Kevin Murphy, et al. Knowledge Base Completion via Search-Based Question Answering[J]. 2014.
- [58] Ni Lao, Teruko Mitamura, Tom Mitchell, et al. Efficient Random Walk Inference with Knowledge Bases[J]. PhD thesis, 2012.
- [59] László Lovász. Random Walks on Graphs: A Survey[C]. 1993.

## 学术成果

1. Yong Huang, Zhichun Wang. Knowledge Base Completion by Learning to Rank Model. CCKS 2017:1-6
2. Yong Huang, Zhichun Wang. Knowledge Base Completion using both Literal and Relational Facts. (EMNLP2018在投)
3. Zhichun Wang, Danlu Wen, Yong Huang, Chu Li. Word, Mention and Entity Joint Embedding for Entity Linking. JIST (Workshops & Posters) 2016: 70-73
4. Zhichun Wang, Rongyu Wang, Danlu Wen, Yong Huang, Chu Li. Entity Linking in Queries Using Word, Mention and Entity Joint Embedding. JIST 2017: 138-150
5. Jun Li, Jinxian Pan, Chen Ye, Yong Huang, Danlu Wen, Zhichun Wang. Linking Entities in Chinese Queries to Knowledge Graph. NLPCC 2015: 590-597
6. 黄勇, 王志春. 基于学习排序算法的知识库补全方法及装置. (专利申请号: 201810059641.2)
7. 王志春, 黄勇. 一种知识库的补全方法及装置. (专利申请号: 201810085005.7)



## 致 谢

衷心感谢导师王志春教授对本人的精心指导，他的言传身教将使我终生受益。

在北京师范大学计算机软件与理论学习的三年中，承蒙王志春老师不辞劳苦的指导和帮助，让我从一个无知的学生，慢慢学到很多做人做事做学问的道理。导师不仅教我很多课本上的知识，带领我们读论文，做课题，教我学习到知识库、机器学习等算法原理，让我学习领悟到很多知识库补全研究的前沿和热点，找到自己喜欢和专心研究的问题。更是通过言传身教让我学会做事，学会做人，教了我很多读书做事的道理。这些知识不仅仅帮助我在学术的道路上探索向前，也帮我成长，学会发现更大的世界。

感谢我们实验室的所有师妹师弟们，感谢李楚同学不辞劳苦的帮助我解决Linux系统问题，感谢郑伟师妹帮我一起合作Java代码项目，感谢孙铭晨师妹经常为我们出谋划策，带给我们鲜美的水果零食，感谢吴妍蓉师妹帮我解决学术问题，带给我欢声笑语。最由衷的感谢我的父母，他们无私的为我付出，支持我在贫寒的学术道路上又坚持了三年，让我始终坚持自己的想法，做自己应该做的事情。

感谢清华的薛瑞尼及相关同学，他们制作维护的清华学位论文模板极大的方便了 $\text{\LaTeX}$ 用户的论文写作。

黄勇

2018年 6月