软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(10):2915−2930 [doi: 10.13328/j.cnki.jos.005549]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种元路径下基于频繁模式的实体集扩展方法*

郑玉艳, 田 莹, 石 川

(北京邮电大学 计算机学院,北京 100876)

通讯作者: 石川, E-mail: shichuan@bupt.edu.cn

摘 要: 实体集扩展是指已知某个特定类别的几个种子实体,根据一定的规则得到该类别的更多实体.作为一种经典的数据挖掘任务,实体集扩展已经有很多的应用,诸如字典建立、查询建议等.现有的实体集扩展主要是基于文本或网页信息,即实体之间的关系从其在文本或者网页中的共现来推断.随着知识图谱研究的兴起,根据知识图谱中知识的共现来研究实体集扩展也成为了一种可能.主要研究知识图谱中的实体集扩展问题,即:给定几个种子实体,利用知识图谱来得到更多的同类别的实体.首先,把知识图谱建模成一个异质信息网络,即含有多种实体类型或者关系类型的网络,提出了一种新的元路径下基于频繁模式的实体集扩展方法,称为 FPMP_ESE.FPMP_ESE 采用异质信息网络中的元路径来捕捉种子实体之间的潜在共同特征.为了找到种子实体之间重要的元路径,设计了一种新的基于频繁模式的元路径自动产生算法 FPMPG.之后,为了更好地给每条元路径分配相应的权重,设计了启发式的方法和 PU learning 的方法.最后,在真实数据集 Yago 上的实验结果表明,所提出方法较其他方法在实体集扩展任务上具有更好的性能和更高的效率.

关键词: 知识图谱;实体集扩展;异质信息网络;元路径;频繁模式;PU learning

中图法分类号: TP311

中文引用格式: 郑玉艳,田莹,石川.一种元路径下基于频繁模式的实体集扩展方法.软件学报,2018,29(10):2915−2930. http://www.jos.org.cn/1000-9825/5549.htm

英文引用格式: Zheng YY, Tian Y, Shi C. Method of entity set expansion based on frequent pattern under meta path. Ruan Jian Xue Bao/Journal of Software, 2018,29(10):2915−2930 (in Chinese). http://www.jos.org.cn/1000-9825/5549.htm

Method of Entity Set Expansion Based on Frequent Pattern Under Meta Path

ZHENG Yu-Yan, TIAN Ying, SHI Chuan

(School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: Entity set expansion (ESE) refers to getting a more complete set according to some rules, given several seed entities with specific semantic meaning. As a popular data mining task, ESE has many applications, such as dictionary construction and query suggestion. Contemporary ESE mainly utilizes text or Web information. That is, the intrinsic relations among entities are inferred from theirco-occurrences in text or Web. With the surge of knowledge graph in recent years, it is possible to extend entities according to their co-occurrences in knowledge graph. This paper studies the problem of the entity set expansion in knowledge graph. That is, given several seed entities, how to obtain more entities by leveraging knowledge graph. Firstly, the knowledge graph is modeled as a heterogeneous information network (HIN), which contains multiple types of entities or relationships. Next, a novel method of entity set expansion based

* 基金项目: 国家重点研究和发展计划(973)(2017YFB0803304); 国家自然科学基金(61772082, 61375058); 北京市自然科学基金(4182043)

Foundation item: National Key Research and Development Program of China (973) (2017YFB0803304); National Natural Science Foundation of China (61772082, 61375058); Beijing Municipal Natural Science Foundation of China (4182043)

本文由“本体工程与知识图谱”专题特约编辑李涓子教授推荐.

收稿时间: 2017-07-20; 修改时间: 2017-11-08; 采用时间: 2018-01-24; jos 在线出版时间: 2018-02-08

CNKI 网络优先出版: 2018-02-08 11:55:42, http://kns.cnki.net/kcms/detail/11.2560.TP.20180208.1155.005.html

on frequent pattern under Meta path, called FPMP_ESE, is proposed. FPMP_ESE employs Meta paths to capture the implicit common traits of seed entities. In order to find the important Meta paths between entities, an automatic Meta path generation method is designed based on frequent pattern called FPMPG. Then, two kinds of heuristic and PU learning methods are developed to distribute the weights of Meta paths. Finally, experiments on real dataset Yago demonstrate that the proposed method has better effectiveness and higher efficiency compared to other methods.

Key words: knowledge graph; entity set expansion; heterogeneous information network; meta path; frequent pattern; PU learning

实体集扩展指的是这样一类问题:已知某个特定类别的几个种子实体,根据一定的规则得到该类别的更多实体.比如给定种子集合{北京,华盛顿,莫斯科},利用数据确定种子集合潜在的类别语义即国家首都,然后找出更多的同类别的实体,诸如{汉城,东京,吉隆坡,...}.实体集扩展已经有很多的应用,例如字典建立[1]、词义辨析[2]等等.

目前,已经有很多实体集扩展的方法,这些方法的数据源大都是文本或者网页[3−6],并且使用实体的分布式信息或者上下文模式来进行扩展.例如,Wang 等人[4]提出了一种新颖的 SEAL 方法,该方法可以被用在任意标记语言或者人类语言的半结构文档中.最近,知识图谱已成为一种存储和检索事实信息的重要工具.随着知识图谱的流行,很多研究学者开始使用这一工具作为辅助信息来提高文本或者网页中的实体集扩展的准确性.例如,齐振宇等人[7]通过使用维基百科引入语义知识来解决种子歧义性的问题.

然而,利用知识图谱作为单独的数据源来进行实体集扩展的工作还很少.因为知识图谱不需要经过诸如文本等复杂的自然语言处理过程,并且包含大量的对象和丰富的语义关系,所以把知识图谱作为单独的数据源进行实体集扩展是可能的并且是很有必要的.知识图谱是由形如〈主体,谓语,客体〉的三元组构成,包含有多种类型的实体和关系,因此,我们可以把知识图谱建模成一个异质信息网络,即,包含有多种实体类型或关系类型的网络[8].异质信息网路中的一个很重要的概念是元路径[8],它是由实体类型和关系组成的一个序列,常被用来捕捉丰富的语义信息.基于这个思想,我们采用元路径来捕捉种子实体之间潜在的共同特征,从而进行实体集扩展.但是,我们会遇到以下两方面的挑战.

- 一是知识图谱中实体之间的元路径非常多,难以枚举.在传统的异质信息网络中,存在少量的实体类型和关系类型,可以人工列举出有意义的元路径.但是知识图谱中的实体类型和关系类型非常多,无法一一列举,因此,一种高效的自动寻找元路径的方法是非常有必要的;
- 二是即使我们可以自动地找到实体之间的重要元路径,如何对这些元路径进行组合从而进行实体集扩展,也是非常具有挑战性的.正如实体集扩展中有很少的种子实体,很难用传统的监督式的机器学习方法来建立一个分类或者排序模型,因此,我们需要设计一种方法来对这些元路径进行有效组合.

为了解决第 1 个方面的挑战,我们设计了一种基于频繁模式的元路径自动产生算法,称为 FPMPG(frequent pattern based meta path generation).具体地,FPMPG 算法利用了频繁模式挖掘技术,先将种子实体映射为实体事务,然后探测出所有种子实体的频繁模式(重要的 l-关系路径),并将任意两条 l-关系路径根据其结束关系所连接的实体是否相同进行连接,得到重要的元路径.此方法比文献[9]中提出的单向扩展的元路径产生算法效率更高.为了解决第 2 个方面的挑战,我们设计了两种策略为元路径分配适当的权重,进而把元路径整合在一起进行实体集扩展:一种是启发式方法,另一种是 PU learning(positive and unlabeled learning)的方法.最后,在真实数据集 Yago 上的实验,验证了本文提出方法较其他方法具有更好的有效性和更高的效率,并且进一步研究了不同的种子组合和种子数目等对实体集扩展性能的影响.

本文第 1 节介绍与知识图谱中的实体集扩展相关的工作.第 2 节介绍异质信息网络、知识图谱等基本概念知识.第 3 节详细描述本文提出的新颖的实体集扩展方法.第 4 节设计相关实验,验证提出方法的有效性和效率,并且进一步研究不同种子组合和种子数目等对性能的影响.第 5 节对全文进行概括总结,并对进一步的研究方向进行初步探讨.

1 相关工作

知识图谱中,基于频繁模式和元路径的实体集扩展的相关工作主要涉及以下4个方面:(1) 知识图谱;(2) 异质信息网络;(3) 频繁模式挖掘;(4) 实体集扩展.本节主要围绕这4个方面讨论已有的相关工作.

1.1 知识图谱

知识图谱是谷歌于2012年为优化搜索结果而提出来的[10],它是一个带有语义属性的知识库系统,是从诸如文本、维基百科等数据源中抽取相关知识建立而成.最近几年,已有大量的关于知识图谱的相关工作,本节主要从知识图谱的建立、知识图谱的精炼、知识图谱中的数据挖掘这3个角度来讨论.

知识图谱建立方面的工作主要包括诸如CYC[11]等专业的知识图谱、Freebase[12]等经过大众编辑过的知识图谱以及Yago[13]等抽取自大规模、半结构化网页知识库中的知识图谱.另外,对于那些非结构或者半结构化的信息,很多信息抽取的技术也被提出,采用这些抽取技术建立的知识图谱有KnowledgeVault[14]等等.

实际上,无论采用哪种方法建立起来的知识图谱都不一定具有完全的覆盖率和准确率.为了增加知识图谱的可使用性,相关研究学者提出了很多精炼方法,主要包括知识图谱的补全和纠错两大类.

知识图谱补全的目标是增加知识图谱的覆盖率,依据其补全的信息不同,又可以分为补全实体、实体类型以及实体之间的关系等.补全实体类型的方法通常是将其看作一个分类问题,利用已有的实体之间的关系或者不同的知识图谱中的链接关系来预测实体类型[15].补全实体之间的关系也是采用分类的方法,例如,Socher 等人[16]基于已有的其他关系来训练张量神经网络进而预测指定的关系.最近,有些研究者开始采用嵌入式的方法来预测关系[17].而对于实体补全的研究相对较少,可以采用不同语言版本的知识库补全其他语言中的实体[18].

知识图谱纠错的目标是改正已有的错误.依据其纠错的信息,可以分为类型纠错、关系纠错、属性值纠错以及不同知识图谱中的相同实体之间的对应关系的纠错.例如,Paulheim 等人[19]仅在数据本身上进行操作来探测属性和类型的统计分布,进而来识别知识图谱中错误的事实.

基于知识图谱的数据挖掘工作有问答、搜索、链接预测、决策支持等[20,21].对于问答系统,如何高效、准确地解析用户的问题是至关重要的.Zou 等人[20]从图数据驱动的角度来研究知识库中的问答,提出了一种系统框架来回答自然语言问题,并且给出了语义查询图来更好地建模作者的查询意图.对于链接预测,由于知识图谱中的很多事实或者关系的不正确性,使这一任务非常重要.Cao 等人[21]把知识图谱建模成异质信息网络,采用实体之间的元路径作为特征训练一个分类模型,进而对实体之间的关系进行预测.Maximilian 等人[22]将统计关系学习应用到知识图谱中进行事实的预测.而对于实体集扩展任务的研究相对较少.Zheng等人[9]将知识图谱建模成异质信息网络,采用元路径的方法来进行实体集扩展,但其效率相对来说较低,并且只是采用了启发式的方法进行元路径的整合.本文研究知识图谱中的实体集扩展问题,提出了更好、更高效的方法.

1.2 异质信息网络

异质信息网络是由不同类型的实体或关系构成的信息网络[23].自从数据挖掘权威韩家炜和Yu等人于2009年提出异质信息网络的概念之后,相关的概念和分析方法已经成为数据挖掘研究的热点.由于异质信息网络能够整合复杂的结构关系,同时包含丰富的语义信息,因而被广泛应用于各种数据挖掘问题,诸如相似性度量、聚类、分类等等.最近,Shi 等人[24]对异质信息网络方面的工作进行了比较全面的总结整理.异质信息网络中一个很重要的概念是元路径[8],元路径包含丰富的语义信息,基于不同的元路径,对象之间的语义是不同的.因此,异质信息网络中的很多数据挖掘任务都是基于元路径的.例如,Sun等人[8]提出了基于元路径的框架来捕捉异质信息网路中同种类型对象之间的语义,并且提出了 top-k 相似性搜索问题.但在实际生活中,我们可能会搜索不同类型的相关对象,如“作者和会议”“用户和电影”等.基于此,Shi 等人[25]提出了一种可以度量任意元路径下任意类型对象之间的相关性的通用方法.然而,实体集扩展问题还很少被研究,在本文中,我们把知识图谱建模成异质信息网络进行实体集扩展.

1.3 频繁模式挖掘

频繁模式挖掘是指发现数据集中出现频率超过一定阈值的模式,它是数据挖掘中的一项基础性工作,也是关联规则挖掘的一个关键步骤,可以应用于分类、聚类等数据挖掘任务.

为了发现大型超市中顾客购买行为之间的有趣联系,Agrawal 于 1994 年开创性地提出了关联规则挖掘问题,并提出了著名的 Apriori 算法[26],它用频繁项集产生关联规则.为了解决 Apriori 算法重复扫描数据库的问题,韩家炜等人提出了 FP-Growth 算法[27],它采用分治策略,将数据库压缩到频繁模式树(FP-tree)上,提高了挖掘效率.为了描述同一顾客在多次购物所购商品之间可能存在的某种关联关系,Agrawal 于 1995 年又发表了序列模式挖掘算法[28].之后,又出现了时间序列的频繁模式挖掘.时间序列数据存在于社会的各个领域,如科学研究记录、气象观测、股票等.传统时间序列分析的任务仅仅是为了对系统整体行为进行预测和控制,而时间序列的频繁模式挖掘则可以对时间序列的局部特征进行分析,发现经常出现的变化模式,对于时间序列的预测、相似性挖掘、周期模式挖掘等应用都具有十分重要的意义.随着 Internet 的发展,很多人开始对网络中的数据进行挖掘,称为 Web 挖掘,将关联规则应用于 Web 日志挖掘,可以发现用户的频繁访问模式,进而优化网络的拓扑结构和网页内容.此外,频繁模式挖掘技术也被运用到了知识图谱中.例如,Abedjan 等人[29]利用关联规则挖掘改进了 RDF 数据.Jiang 等人[30]提出了一种新的一般频繁模式挖掘(frequent generalized pattern mining)算法,用于挖掘 RDF 元数据中的一般关联关系(generalized pattern mining).知识图谱中实体的频繁模式可以揭露实体的某些共同特征,基于此,我们将频繁模式挖掘技术应用到知识图谱中的实体集扩展.

1.4 实体集扩展

最近几年,实体集扩展已经得到了学术界[4,5,31]和工业界(比如谷歌)的广泛关注,并且已有大量的实体集扩展的相关工作.根据数据源的不同,这些方法主要分为 3 类:基于文本信息、基于网页信息和其他类型.

基于文本数据源的实体集扩展方法,主要是基于这样一个假设,即,具有相似意义的单词往往出现在相似的上下文中,从而利用实体周围单词的分布信息来扩展特定类[3,6].基于网页数据源的方法首先根据给定的种子来抽取重要的模式,然后抽取满足固定模式的候选实体,最后根据相似性函数来进行实体扩展[4,5,31].最近,有很多研究者开始利用外部语义信息来提高文本或者网页中的实体集扩展的准确率.例如,齐振宇等人[7]利用维基百科引入语义知识来解决种子歧义性的问题,Jindal 等人[32]指定负例来限定扩展类的语义边界,等等.

近来,异质信息网络和知识图谱也已被应用在相关研究中.例如,Yu 等人[33]提出一种基于元路径的排序模型来进行实体查询,但是元路径需要事先被特定领域专家指定.Metzger 等人[34,35]研究知识图谱中的相似实体搜索的问题,Chen 等人[36]设计了一个实体扩展和纠错系统.然而,仅在知识图谱中进行实体集扩展还很少被研究.

2 基本知识

在这一节,我们介绍本文中用到的一些基本概念和基本知识.

定义 1(知识图谱[10,37]). 知识图谱被定义为一个 RDF 图,它是由 RDF 三元组⟨主体,谓语,客体⟩表示的.其中,主体是被描述的资源,客体是主体在谓语上的值或者是另一个资源,所有能使用 RDF 表示的对象都可以称为资源[38].

例如,在图 1 中,⟨斯皮尔伯格,导演,战马电影⟩(⟨Steven_Spielberg,directed,War_House(film)⟩)是一个 RDF 三元组的例子,斯皮尔伯格是主体,导演是谓语,战马电影是客体也是另一个资源.现有的知名的知识图谱有 Yago[13]和 Freebase[12]等.知识图谱中的每个实体都对应一或多个实体类型,同一对实体之间也可能有一或多种关系,如图 1 所示中,实体斯皮尔伯格(Steven_Spielberg)既是电影导演(film_director)类型也是电影制作人(film_maker)类型,实体斯皮尔伯格(Steven_Spielberg)和战马电影(War_House(film))之间有 *directed* 和 *created* 两种关系.

定义 2(异质信息网络[8]). 异质信息网络被定义为一个有向图 $G=(V,E)$,其中,V,E 分别是所有对象的集合和所有边的集合,并且存在着一个对象类型的映射函数 $\varphi: V\rightarrow A$ 和一个边类型的映射函数 $\psi: E\rightarrow R$,每个对象 $v\in V$ 属于一种特定的对象类型 $\varphi(v)\in A$,每条边 $e\in E$ 属于一种特定的关系类型 $\psi(e)\in R$,对象类型的种类 $|A|>1$ 或者关系类

型的种类$|R|>1$.

异质信息网络中一个很重要的概念是元路径[8],常被用来捕捉丰富的语义,它是由对象类型和关系类型组成的一个序列,其符号表示形式是 $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \ldots \xrightarrow{R_l} A_{l+1}$,$A_i$ 是对象类型,R_i 是关系类型,长度为 l.

例如,图 1 是知识图谱 Yago 的一个示例,存在多种类型的实体和关系,有演员类型的实体 Nigel_Havers,也有导演类型的实体斯皮尔伯格(Steven_Spielberg).很显然,知识图谱可以被看作是一个异质信息网络.实体 Nigel_Havers 和 Toby_Kebbell 不仅仅是演员类型的实例,而且也是参演 Steven_Spielberg 导演的电影的演员类的实例.为了更好地区分这两种实例,我们称前者为粗粒度的实体,后者为细粒度的实体.本文中,我们主要关注细粒度的实体集扩展.在实体 Nigel_Havers 和 Toby_Kebbell 之间存在一条元路径:

$$Person \xrightarrow{actedIn} Movie \xrightarrow{directed^{-1}} Person \xrightarrow{directed} Movie \xrightarrow{actedIn^{-1}} Person,$$

其中,$directed^{-1}$ 是关系 *directed* 的逆关系.这条元路径表明这两个实体演过被同一个导演导的电影,揭示了这两个实体之间潜在的共同特性.

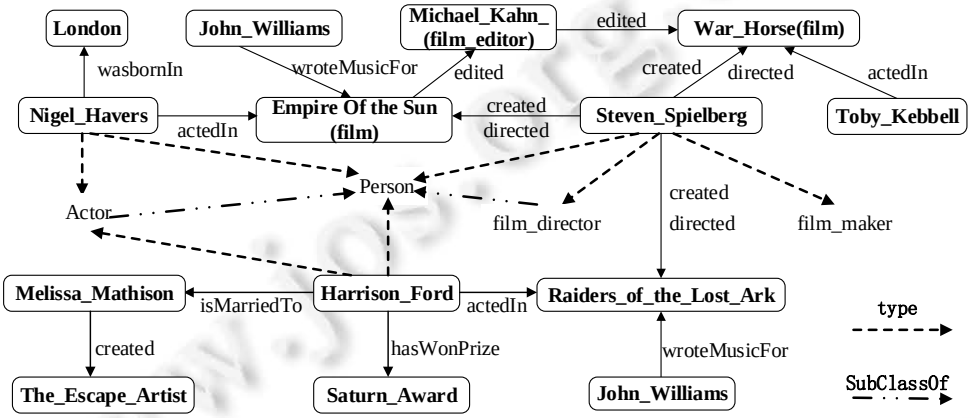


Fig.1 A tiny example of entity, relationship and type in Yago

图 1 Yago 中实体、关系及类型示例

定义 3(关系路径). 关系路径是一个仅由关系组成的序列,不包含实体,其表示形式是 $\xrightarrow{R_1} \xrightarrow{R_2} \ldots \xrightarrow{R_l}$. 这个序列的第 1 个关系称为开始关系,最后一个关系称为结束关系,其余的关系称为中间关系.关系的数目称为关系路径的长度,长为 l 的关系路径被记为 l-关系(l-relation).

定义 4(实体事务与 l-关系集). 对于一个事务数据库 $T=\{T_1,T_2,\ldots,T_n\}$[39],每个事务 $T_i(T_i \in T, \forall i \in \{1,2,\ldots,n\})$由一组项构成,即 $T_i=\{x_1,x_2,\ldots,x_l\}$.项集是一个集合 $P \in T_i$,它的大小就是它包含的项的数目.我们用 l-项集(l-itemset)(或 l-模式)来表示大小为 l 的项集.P 的支持数 $s(P)$是包含 P 的事务的数目.若 $s(P) \geqslant \sigma$,则 P 是频繁的,其中,σ是最小支持数阈值.相对应地,我们把知识图谱中的一个实体映射为一个事务,称为实体事务.每个实体事务由不同长度的 l-关系构成.实体事务的所有相同长度的 l-关系组成的集合称为最大 l-关系集(maximal l-relationset),最大 l-关系集的任意一个非空子集称为一个 l-关系集(l-relationset).

定义 5(l-关系集支持数与最大频繁 l-关系集). l-关系集的支持数是指满足这一关系集的实体事务的数目.若一个 l-关系集的支持数不小于最小支持数阈值,我们称这个 l-关系集是频繁的,所有频繁 l-关系组成的集合称为最大频繁 l-关系集(maximal frequent l-relationset).

例如,在图 1 中,$\xrightarrow{actedIn} \xrightarrow{directed^{-1}}$ 是满足实体 Nigel_Havers 的一条关系路径,$\xrightarrow{actedIn}$ 是开始关系,$\xrightarrow{directed^{-1}}$ 是结束关系,关系路径的长度是 2.Nigel_Havers 是一个实体事务,包含有 1-关系(如 $\xrightarrow{actedIn}$)、2-关系(如 $\xrightarrow{actedIn} \xrightarrow{directed^{-1}}$)、3-关系(如 $\xrightarrow{actedIn} \xrightarrow{directed^{-1}}$)等等.实体事务 Nigel_Havers 的最大 1-关系集是所有 1-关系组成的集合 $\{\xrightarrow{actedIn}, \xrightarrow{wasBornIn}\}$,最大 2-关系集是所有 2-关系组成的集合 $\{\xrightarrow{actedIn} \xrightarrow{directed^{-1}},$

$\xrightarrow{actedIn}\xrightarrow{wasMusicFor^{-1}}$, $\xrightarrow{actedIn}\xrightarrow{created^{-1}}$, $\xrightarrow{actedIn}\xrightarrow{edited^{-1}}$} 等等.其中,{$\xrightarrow{actedIn}$} 和 {$\xrightarrow{wasBornIn}$} 都是 1-关系集,{$\xrightarrow{actedIn}$} 的支持数是 3,因为有 3 个实体 Toby_Kebbell、Nigel_Havers 和 Harrison_Ford 都满足这一关系.{$\xrightarrow{wasBornIn}$} 的支持数是 1,因为只有实体 Nigel_Havers 满足这一关系.如果最小支持数阈值设置为 3,那么 1-关系集 {$\xrightarrow{actedIn}$} 是频繁的,因为其支持数不小于 3;{$\xrightarrow{wasBornIn}$} 不是频繁的,因为其支持数小于 3.在图 1 所示中,1-关系只有 $\xrightarrow{actedIn}$ 的支持数不小于 3,所以最大频繁 1-关系集也是{$\xrightarrow{actedIn}$}.

3 提出的方法

为了解决知识图谱中的实体集扩展问题,本文提出了一种新的元路径下基于频繁模式的实体集扩展方法,称为 FPMP_ESE(entity set expansion based on frequent pattern under meta path).如前所述,知识图谱本身是一个异质信息网络,我们采用元路径来表征种子实体之间潜在的共同特征.而知识图谱中的实体之间的元路径数目庞大,无法枚举,为了自动产生重要元路径,我们设计了一种基于频繁模式的元路径自动产生算法 FPMPG.之后,我们设计了两种权重学习方法对元路径进行组合:一种是启发式方法,另一种是 PU learning(positive and unlabeled learning)方法.下面详细阐述各个步骤.

3.1 基于频繁模式的元路径自动产生算法

受 Apriori 算法[26]的启发,我们设计了一种新的基于频繁模式的元路径自动产生算法 FPMPG.相比文献[9]提出的单向扩展的元路径产生算法,FPMPG 可以重复利用已有的频繁关系,因而可以更高效地发现种子实体之间的重要元路径.FPMPG 算法主要包括以下两个阶段:第 1 个阶段是将种子实体映射为实体事务,第 2 个阶段是利用频繁模式产生重要元路径.下面分别具体阐述各个阶段.

3.1.1 将种子实体映射为实体事务

受频繁模式挖掘的启发,我们将其应用在种子实体的特征挖掘上,将种子实体映射为相对应的实体事务.具体地,我们将知识图谱中的一个种子实体看作事务数据库中的一个事务,将实体的 l-关系(l-relation)看作对应事务的 l-项,任意一个 l-关系的集合都对应于一个 l-项集.为了更清晰地说明种子实体的映射过程,以图 1 所示的 3 个种子实体 Toby_Kebbell、Nigel_Havers 和 Harrison_Ford 为例(这 3 个实体都有一个共同的类标签,即,被同一导演(Steven_Spielberg)导的电影的演员).表 1 展示了映射后的实体事务,其中,第 1 列是种子实体,第 2 列是种子实体连接的所有 1-关系(如 $\xrightarrow{actedIn}$),第 3 列是种子实体连接的所有 2-关系(如 $\xrightarrow{actedIn}\xrightarrow{directed^{-1}}$),依次类推.

Table 1 An example of seed entity transaction in knowledge graph

表 1 知识图谱中的种子实体事务的例子

种子实体	1-关系	2-关系	33-关系
by_Kebbell	$\xrightarrow{actedIn}$	$\xrightarrow{actedIn}\xrightarrow{directed^{-1}}$,...	…
Nigel_Havers	$\xrightarrow{actedIn}$, $\xrightarrow{wasBornIn}$	$\xrightarrow{actedIn}\xrightarrow{directed^{-1}}$, $\xrightarrow{actedIn}\xrightarrow{wasMusicFor^{-1}}$,...	…
Harrison_Ford	$\xrightarrow{actedIn}$, $\xrightarrow{hasWonPrize}$,...	$\xrightarrow{actedIn}\xrightarrow{directed^{-1}}$, $\xrightarrow{actedIn}\xrightarrow{wasMusicFor^{-1}}$,...	…

3.1.2 利用频繁模式产生元路径

第 1 阶段将种子实体映射为相应的实体事务,然后,我们利用频繁模式挖掘技术来产生重要的元路径.其基本思路是:首先扫描所有种子实体事务,设定 l-关系中的 l 的最大值,记录每个 l-关系的支持数;然后,利用预先设定的最小支持数阈值 σ 寻找频繁 l-关系,并且根据设计的规则将这些频繁 l-关系进行连接;最后,考虑关系路径中的实体类型,得到最终的重要元路径.具体地,分为以下 5 步进行详细阐述.

(1) 第 1 步:记录实体事务的 l-关系.我们扫描所有种子实体事务,根据设定的 l 的最大值,记录种子实体的所有 l-关系;同时,记录每个 l-关系的支持数以及其满足的种子编号,记为候选 l-关系集合 C_l;

(2) 第 2 步:l-关系剪枝.对候选 l-关系集合 C_l 中的每一个 l-关系,将它的支持数与 σ 进行比较:若小于 σ,则剪掉这个 l-关系;若大于 σ,则这个 l-关系和它的支持数以及满足的种子编号将被保留,记为频繁 l-关系集

合 F_l;

(3) 第 3 步:l-关系连接.这一步包括自连接和互连接两种连接方式.自连接指的是 l-关系与自身进行连接,此时,l-关系对应的种子编号至少应该含有两个.互连接是指 l-关系与其他 l-关系进行连接.这里,针对第 2 步产生的频繁 l-关系集合 F_l,我们定义如下连接规则:考虑每个 l-关系的结束关系所连接的实体,若任意两个 l-关系的结束关系所连接的实体是相同的,那么就可以通过连接它们得到一条路径实例.同时,我们需要记录下每条路径实例连接的种子对数,记为候选关系路径 C_p.例如,通过连接一个频繁 1-关系和一个频繁 2-关系,我们可以得到长度为 3 的路径实例;

(4) 第 4 步:关系路径剪枝.针对 C_p 中的每条关系路径,如果关系路径连接的种子对数目小于设定的关系路径阈值 τ,则剪掉;否则,加入到频繁关系路径集 F_p;

(5) 第 5 步:添加实体类型得到重要元路径.根据第 4 步得到的频繁关系路径集,我们用实体类型取代实体本身,就可以得到重要的元路径,同时保留下连接的种子对数,记为重要元路径 P.这里,我们根据概念层次结构选择实体的最小共同祖先对应的实体类型.

特别地,为了清晰地记录每个状态的数据,我们给出一个新的数据结构,如图 2(b)所示.其中,l-关系记录相应的 1-关系、2-关系等等,支持数记录每个 l-关系满足的种子实体事务数,种子编号记录 l-关系满足的种子.另外,数据结构图 2(a)表示所有种子对的组合,假设有 m 个种子,则总的种子对数是 $m\times(m-1)$.

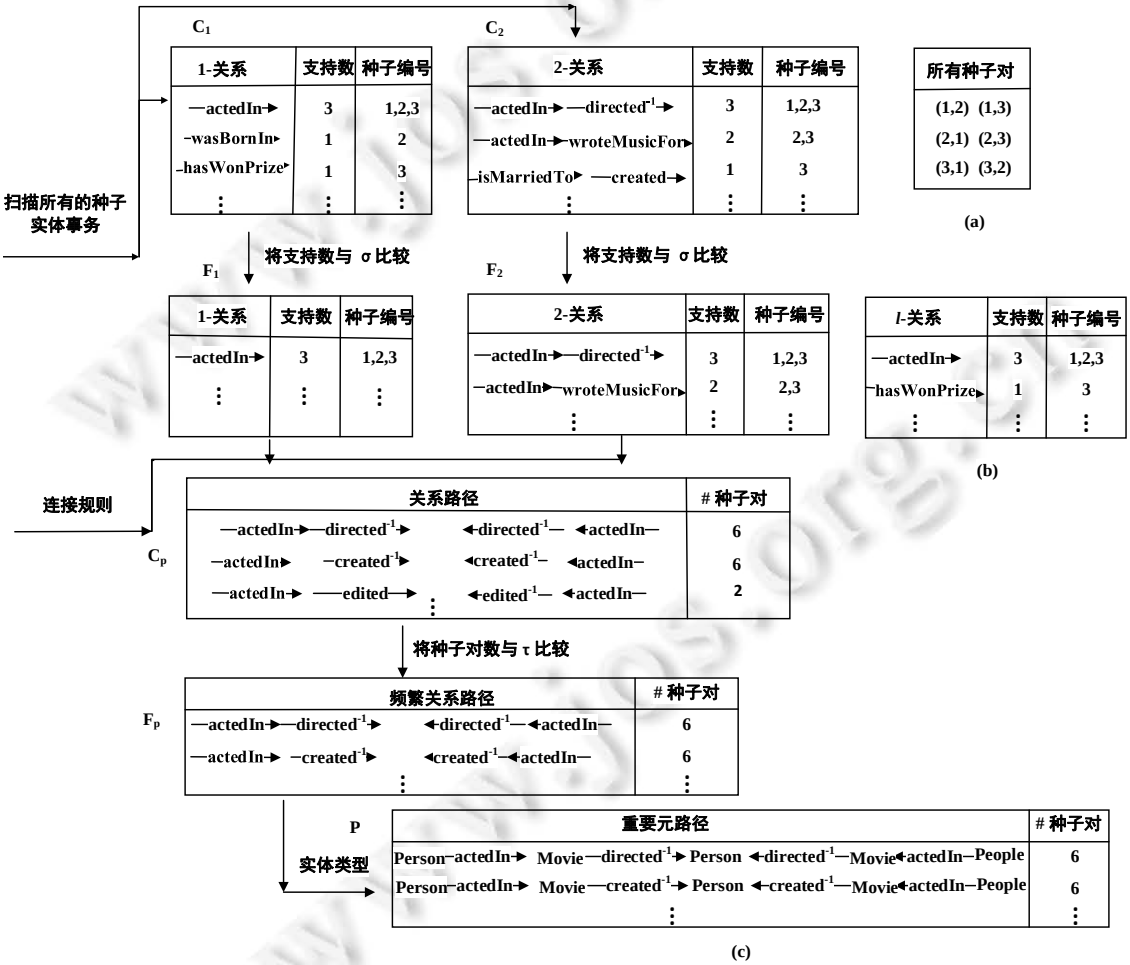


Fig.2 Frequent pattern based meta path generation algorithm

图 2 基于频繁模式的元路径产生算法

为了更清晰地阐述重要元路径的产生过程,我们给出如图 2(c)所示的一个例子.给定演员类别的 3 个种子 Toby_Kebbell、Nigel_Havers 和 Harrison_Ford,分别标记为 1,2,3.

(1) 第 1 步:记录种子实体满足的所有不同长度的 l-关系.这里,设定 l 的最大值为 2,即,我们仅仅记录 1-关系和 2-关系,因为长度大于 2 的关系含有很少的语义信息,同时记录下 3 个种子实体相应的 l-关系的支持数和满足的种子编号,得到候选 1-关系集合 C_1 和候选 2-关系集合 C_2,如图 2(c)中 C_1,C_2 所示.在 C_1 中可以看到,1-关系 $\xrightarrow{actedIn}$ 的支持数是 3,而 $\xrightarrow{wasBornIn}$ 的支持数只有 1,因此,关系 $\xrightarrow{actedIn}$ 比 $\xrightarrow{wasBornIn}$ 更重要,更能表现种子实体的潜在特征;

(2) 第 2 步:l-关系剪枝.对于 C_1 中的每个 1-关系,我们将它的支持数与 σ 进行比较.这里,我们设定 σ 为 2,则所有支持数低于 2 的 1-关系都被剪掉,得到了频繁 1-关系的集合 F_1.可以看到,F_1 中并没有关系 $\xrightarrow{wasBornIn}$,因为它的支持数低于 2 从而被剪掉.对 C_2 中的每一个 2-关系也进行相同的操作,得到了频繁 2-关系集合 F_2;

(3) 第 3 步:l-关系连接.根据 l-关系的结束关系所连接的实体是否相同,我们将 F_1 和 F_2 中的每个满足条件的 l-关系进行连接.例如,F_2 中的关系路径 $\xrightarrow{actedIn}\xrightarrow{directed^{-1}}$ 的结束关系所连接的实体为 Steven_Spielberg,这条关系路径满足 3 个种子实体 1,2,3.因此,我们可以对这条关系进行自连接得到关系路径 $\xrightarrow{actedIn}\xrightarrow{directed^{-1}}\xleftarrow{directed^{-1}}\xleftarrow{actedIn}$,它通过相同的实体 Steven_Spielberg 将 6 组种子对(1,2),(1,3),(2,1),(2,3),(3,1),(3,2)连接起来.对其他的 l-关系也按照连接规则进行操作,得到关系路径集 C_P;

(4) 第 4 步:关系路径剪枝.这一步剪掉种子对数小于阈值 τ 的关系路径,这里,设定 τ 为 4,则连接对数为 2 的关系路径 $\xrightarrow{actedIn}\xrightarrow{edited^{-1}}\xleftarrow{edited^{-1}}\xleftarrow{actedIn}$ 将被剪掉,得到频繁关系路径集 F_P;

(5) 第 5 步:我们利用实体类型的数据,得到 Steven_Spielberg 的实体类型为 Person,再添加其他的实体类型,得到这条关系路径对应的最终的重要元路径:

$$Person\xrightarrow{actedIn}Movie\xrightarrow{directed^{-1}}Person\xrightarrow{directed}Movie\xrightarrow{actedIn^{-1}}Person.$$

以同样的方式得到其他的重要元路径,如图 2(c)中 P 所示.

算法 1 是 FPMPG 的伪代码.

FPMPG 包括种子实体映射 TRANSFORMATION 和元路径自动产生 GENERATEPATH 两个过程.第 1 步~第 5 步是种子映射为实体事务的过程.第 6 步~第 25 步是产生重要元路径的过程,其中,第 7 步~第 9 步获得候选 l-关系集,第 10 步~第 12 步得到频繁 l-关系集;根据连接规则,第 13 步~第 18 步得到了候选关系路径集 C_P 和频繁关系路径集 F_P;第 19 步~第 23 步添加实体类型,得到最终的元路径.如果一条元路径连接的种子对数目大于阈值 τ,则此路径是重要的且能够更好地揭示种子的潜在特征,我们可以进一步用这些元路径扩展其他同类别的实体.

算法 1. 基于频繁模式的元路径自动产生算法 FPMPG.

输入:知识图谱 G,种子集 $S=\{s_1,s_2,\ldots,s_m\}$,最大关系长度 l,最小支持数阈值 σ,连接种子对数阈值 τ;

输出:元路径集合 P,元路径相对应的连接种子对数集合 SP.

```
1:  procedure TRANSFORMATION
2:      for each s∈S do
3:          获得实体事务 T;
4:      end for
5:  end procedure
6:  procedure GENERATEPATH
7:      for each t∈T do
8:          获得 C1,C2,…,Cl;
9:      end for
10:     for each in C1,C2,…,Cl do
```

11: 获得 $F_1,F_2,\ldots,F_l$;
12: end for
13: for each in $F_1,F_2,\ldots,F_l$ do:
14: 根据连接规则得到候选关系路径 C_P;
15: 记录连接的种子实体对数 n;
16: if $n\geqslant\tau$ then
17: 把相应的关系路径添加到频繁关系路径集 F_P 中;
18: end for
19: for each $path\in F_P$ do
20: 添加实体类型得到元路径 p;
21: $P\leqslant p\cup P$;
22: $SP\leqslant n\cup SP$;
23: end for
24: return P, SP;
25: end procedure

3.2 元路径的权重学习

算法 FPMPG 产生了重要元的路径 P,但是针对实体集扩展问题,不同元路径的重要性是不同的.因此,如何对这些元路径进行整合就变得尤为重要.实体集扩展可以看作是建立一个排序模型进而对候选实体进行排序,取恰当的前 k 个结果作为扩展集合,本文设计的排序模型如公式(1)所示:

$$R(c_i,S)=\frac{1}{m}\sum_{j=1}^{m}\sum_{k=1}^{l}w_k\times r\{(c_i,s_j)\mid p_k\} \tag{1}$$

其中,c_i 代表第 i 个候选实体;$S=\{s_1,s_2,\ldots,s_m\}$是种子集合;l 是元路径数目;$r\{(c_i,s_j)|p_k\}$表示在路径 p_k 下种子实体 s_j 和候选实体 c_i 的相关性;w_k 是元路径 p_k 的权重值,即需要学习的权重值.本文中,我们设计了两种权重学习方法:一种是启发式方法,一种是 PU learning 方法.下面详细介绍这两种方法.

3.2.1 启发式方法

直观上,若一条元路径连接的种子对越多,就越能反映种子实体之间的共同特征,也就越重要.基于这个思想,我们设计了一种与文献[9]类似的启发式方法,即,采用元路径连接的种子对数和所有种子对数的比值来衡量元路径的重要性.公式如下:

$$w_k=\frac{\dfrac{|SP_k|}{m\times(m-1)}}{\sum_{k=1}^{l}\dfrac{|SP_k|}{m\times(m-1)}}=\frac{|SP_k|}{\sum_{k=1}^{l}|SP_k|} \tag{2}$$

其中,$|SP_k|$代表元路径 P_k 连接的种子对数目,$m\times(m-1)$是总的种子对数,m 为种子数目,公式中的分母表示权重的归一化项.

3.2.2 PU learning 方法

PU learning 方法的主要思想是:利用少量的正例和没有标签的数据(包含潜在的正例和负例)来建立一个分类器进而用于没有标签的数据,判断其是否属于正例或者有多大概率属于正例[6].其主要特性是在训练模型时没有恰当的负例可以利用,而实体集扩展问题中有少量给定的种子正例和很多没有标签的数据,PU learning 可以用来对这样的数据进行学习.在本文中,我们采用文献[40]中的 PU learning 方法,它本质上是从这些非传统的输入数据中学习一个传统的分类器,并且基于这样一个假设,即,少量带标签的正例是从所有正例中随机选择的.这里,我们把种子对作为正例,把候选实体和种子组成的对作为没有标签的数据.另外,该 PU learning 方法可以调整训练的分类器,从而选择合适且效果好的分类器,因为实体集扩展问题中正例的数据是非常少的,诸如支

持向量机等方法并不适合,因此,这里我们选择 adaboost,它可以改变训练数据的分布,从而增加正例的重要性进而获得好的效果.

4 实 验

4.1 数据集

Yago 是一个大规模的知识图谱,它的数据主要来源于 Wikipedia、权威英文词典 WordNet 和著名数据库 GeoNames[13],以 RDF 数据结构描述.目前为止,它已经包含了超过 1 000 万的实体和超过 1.2 亿的事实记录.我们使用 Yago 中的“yagoFacts”“yagoSimpleTypes”和“yagoTaxonomy”这 3 部分,这些数据集中包含 35 种关系,超过 1 300 万的实体和 3 000 多种实体类型.表 2 是其数据描述.

Table 2 Description of the data

表 2 数据的描述

数据	三元组样式	#三元组
yagoFacts	⟨entity relation entity⟩	4 484 914
yagoSimpleTypes	⟨entity rdt:typewordnet_type⟩	5 437 149
yagoTaxonomy	⟨wordnet_typerdfs:subclassofwordnet_type_⟩	69 826

我们选择了 4 个具有代表性的实体集扩展任务来验证 FPMP_ESE 的性能.4 个扩展任务如下:配偶是演员且获得过艾美奖(Emmy award)的演员、在纽约的大学毕业的作家、获得过国家电影奖(national film award)奖项的导演导的电影、在位于马萨诸塞州剑桥(Cambridge of Massachusetts)的大学工作的科学家,分别记为 Actor*、Writer*、Movie*和 Scientist*,它们分别包含 193,60,653 和 202 个实例.

4.2 评价指标

实验中,我们采用 *percision*-at-*k*(*p@k*)和 Mean Average Precision(MAP)来评价算法的性能.*p@k* 是前 *k* 个结果中正确实例所占的比例,这里,我们使用 *p*@10、*p*@30 和 *p*@60.MAP 是 *p*@10、*p*@30 和 *p*@60 的平均准确度(average precision,简称 AP)的均值,这里, $AP=\frac{\sum_{i=1}^{k} p@i\times rel_i}{\#\text{of correct instances}}$,其中,若排在第 *i* 位的结果为正确实例,则 rel_i 为 1;否则为 0.

4.3 实验设置

本小节我们详细介绍实验的有关设置,将启发式和 PU learning 的权重方法相对应的实体集扩展方法分别记为 FPMP_ESE_He 和 FPMP_ESE_PU.因为已有的关于知识图谱中的实体集扩展问题的方法很少,因此我们设计了几种基本的方法 Link-Based、Neighbor、PCRW 和 MP_ESE.详细介绍如下.

- Link-Based:受文本或者网页中的基于模式的方法的启发[41],给出基于实体一跳链路关系的方法;
- Neighbor:受文献[34,35]的启发,给出同时考虑一跳链路和一跳实体的最近邻方法;
- PCRW:一种基于路径受限随机游走的相似性度量方法[42],这里,我们采用其广度优先搜索的策略,并且用是否连接来度量,设置路径长度是 1,2,3,分别记为 PCRW1,PCRW2,PCRW3;
- MP_ESE:最近,文献[9]提出了一种知识图谱中的实体集扩展方法,元路径是单向自动产生,然后采用简单的启发式方法进行整合的.

在算法 FPMP_ESE 中,我们根据经验设置支持数阈值 σ 为 $m-1$,关系路径阈值 τ 为 $m\times(m-1)/2+1$,最大路径长度为 4.其他算法分别设置最优参数.

4.4 有效性实验

在这一小节,我们将 FPMP_ESE 和其他基本方法进行比较,验证其在以上 4 个任务上的有效性.对每个任务,我们随机选择 3 个种子进行实验,实验运行 20 次取平均值,如图 3 所示.

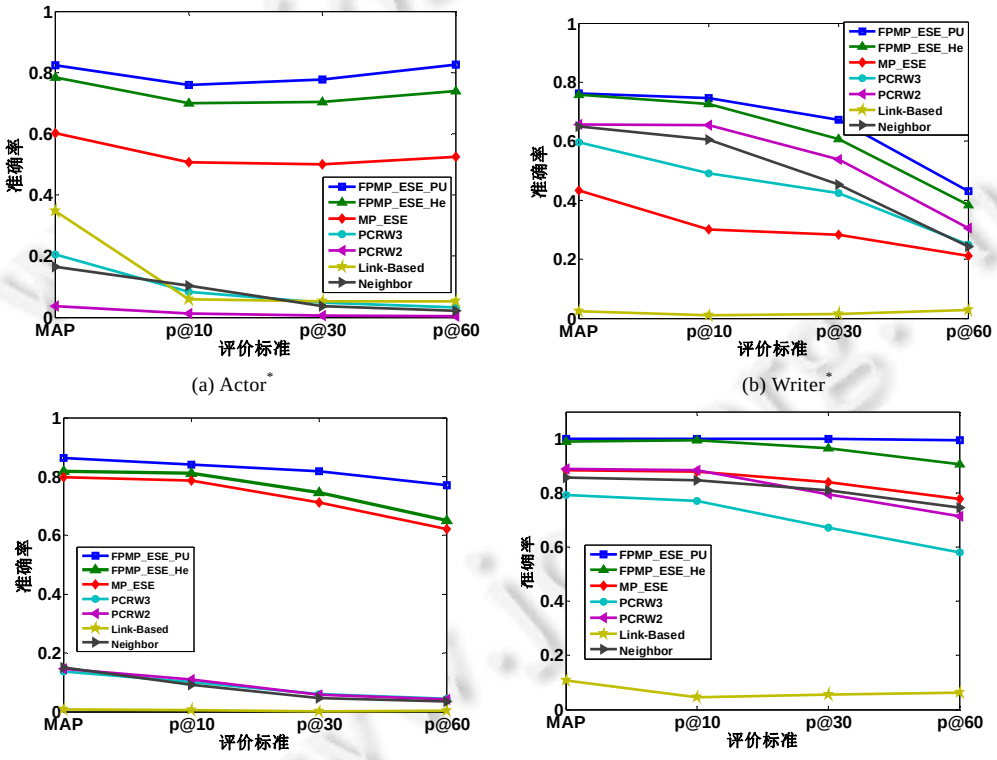
从图 3 中可以发现以下 3 种现象.

(1) 采用元路径的方法 MP_ESE 和 FPMP_ESE 较其他方法具有更好的性能.因为重要元路径可以捕捉种子实体之间潜在的共同特征,过滤掉一些噪音,从而进行更好的实体集扩展;

(2) 本文提出的方法 FPMP_ESE_He 和 FPMP_ESE_PU 较其他方法有更好的性能,因为 FPMP_ESE 可以尽可能全面地找到种子实体之间的重要元路径,不会因为一些潜在的因素剪掉某些重要元路径.例如,在 Actor*任务中,方法 MP_ESE 中寻找元路径的方法是单向搜索的,在搜索到第 3 跳时,其中的一条路径 $\xrightarrow{isMarriedTo}\xrightarrow{hasWonPrize}\xrightarrow{hasWonPrize^{-1}}$ 连接了种子对,之后在剪枝的步骤中,因设计的剪枝条件剪掉了这条路径.那么在后续的搜索过程中,我们就不可能搜索到长度为 4 且表达其语义(配偶是演员且获得过艾美奖(Emmy award)的演员)的路径 $\xrightarrow{isMarriedTo}\xrightarrow{hasWonPrize}\xrightarrow{hasWonPrize^{-1}}\xrightarrow{isMarriedTo^{-1}}$,从而会导致不好的扩展结果.而 FPMPG 算法由于是从每个种子实体进行扩展,然后进行有效连接,所以很好地避免了这一问题,因此会有更好的性能.对于 Link-Based 方法,在所有的任务上都有很差的性能,原因是它只考虑了一跳链路,具有很少的语义信息.对于 Neighbor 和 PCRW3 方法,性能也较差,原因是它们都只考虑了一跳链路和一跳实体,包含的信息也较少;

(3) FPMP_ESE_PU 较 FPMP_ESE_He 有更好的性能,说明与启发式的方法相比,PU learning 方法可以更好地学习到不同元路径的重要性,从而为不同的元路径分配更恰当的权重.

总之,FPMP_ESE 方法有最好的性能,因为它可以尽可能全面地找到种子实体之间重要的元路径,从而更好地捕捉种子实体之间潜在的共同特征,并且,PU learning 的方法可以学习到更加恰当的元路径权重,从而建立更恰当的实体集扩展模型.



(a) Actor* (b) Writer*

(c) Movie* (d) Scientist*

Fig.3 Results of entity set expansion on four tasks

图 3 在 4 个任务上的实体集扩展结果

为了更加直观地观察元路径的有效性,表 3 列出了在 Movie*任务上发现的前 3 条重要的元路径,其中,第 2 列 Gini 重要性表示进行 PU learning 训练后元路径的重要性,第 3 列表示采用启发式方法得到的元路径的重要性.从表 3 可以看出:第 1 条元路径很好地描述了 Movie*任务中种子实体的最重要的语义,即,这些电影都是获得过同一奖项的人导演的.其他元路径也揭示了种子实体的部分隐含信息,第 2 条元路径表明,某些导演也出演了自己导的电影,这在实际情况下也是很合理的.总之,提出的方法可以自动地找到这些有意义的元路径,并且分配恰当的权重,以便很好地发掘种子实体之间的重要语义关系,从而更好地进行实体集扩展.

Table 3 Top 3 meta paths for Movie*

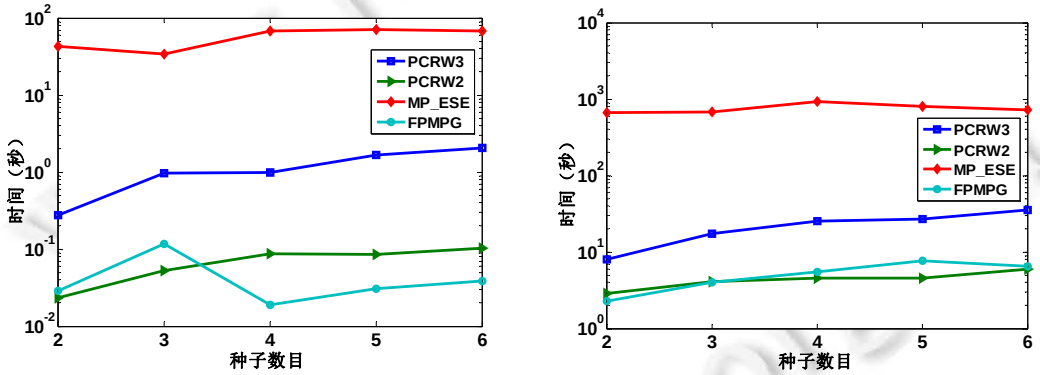
表 3 Movie*任务上最重要的前 3 条元路径

元路径	Gini 重要性	启发式权重
$Movie \xrightarrow{directed^{-1}} People \xrightarrow{hasWonPrize} Award \xrightarrow{hasWonPrize^{-1}} People \xrightarrow{directed} Movie$	0.120 77	0.026 77
$Movie \xrightarrow{actedIn^{-1}} People \xrightarrow{hasWonPrize} Award \xrightarrow{hasWonPrize^{-1}} People \xrightarrow{directed} Movie$	0.119 74	0.098 88
$Movie \xrightarrow{directed^{-1}} People \xrightarrow{hasWonPrize} Award \xrightarrow{hasWonPrize^{-1}} People \xrightarrow{actedIn} Movie$	0.112 56	0.098 88

4.5 效率实验

本小节我们比较采用不同方法寻找元路径的时间,主要从两个角度来研究,即,种子数目和不同的种子组合对寻找路径的效率的影响.

在种子数目对寻找路径时间的影响上,我们分别在 Movie*和 Scientist*任务上选取 2~6 个种子进行实验,对不同种子数目,我们分别在相应的任务上随机选取同等规模的种子进行实验,重复 20 次取平均值,结果如图 4 所示.



(a) Movie* (b) Scientist*

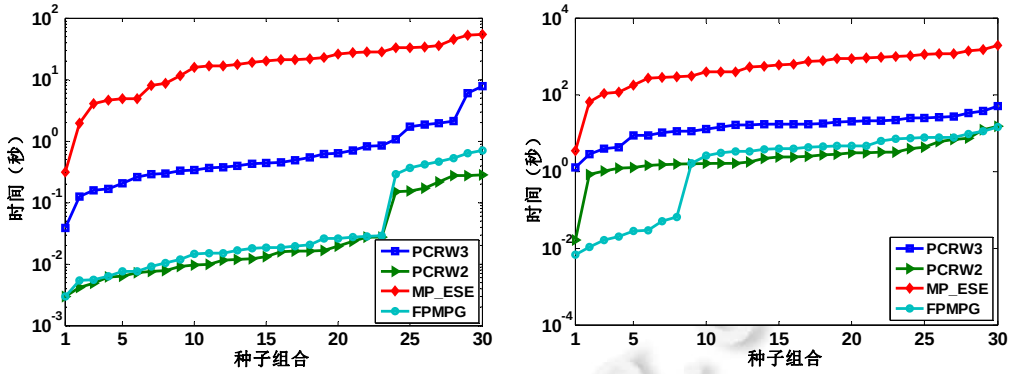
Fig.4 Running time of finding path of different methods with different seed size

图 4 不同的种子数目下采用不同方法寻找元路径的时间

从图中可以看出:随着种子数目的增加,寻找路径的时间整体上有增长的趋势,我们的方法 FPMPG 寻找路径的时间是最短的,因为 FPMPG 是基于种子实体进行路径扩展,然后进行路径连接,比其他单向的扩展方法要节省很多时间.PCRW2 也有较短的运行时间,这是因为它只找了最大长度为 2 的路径,这也导致了其不好的扩展性能.MP_ESE 方法寻找路径的时间是比较慢的,原因是它不仅采用的是单向搜索方式,而且在搜索过程中需要进行各种设定条件的判断,还需要进行剪枝等操作.

对于不同的种子组合对寻找路径时间的影响,我们也分别在 Movie*和 Scientist*任务上选取 3 个种子情况下不同的种子组合进行了 20 次实验取均值,结果如图 5 所示.从图中可以看出:在同样的种子数目下,不同的种子组合,其寻找路径的时间是不同的.可见,种子对寻找路径是有影响的.我们的方法 FPMPG 在不同的种子组合

下寻找路径的时间比 PCRW3 和 MP_ESE 方法都快.采用 PCRW2 方法寻找路径的时间比较快是因为其只找了最大长度为 2 的路径,但有效性很差.



(a) Movie* (b) Scientist*

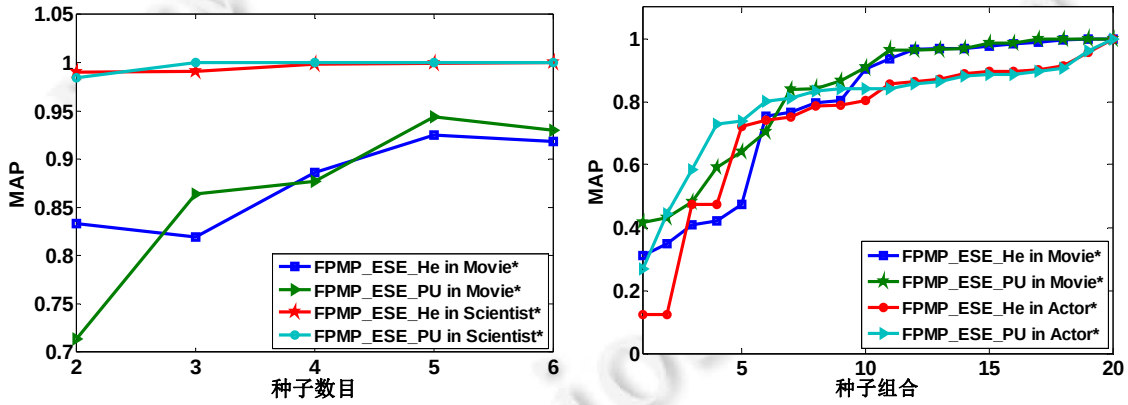
Fig.5 Running time of finding path of different methods with different seed combination

图 5 不同的种子组合下采用不同方法寻找元路径的时间

总之,种子数目和不同的种子组合对寻找路径的时间都是有影响的,因此在下一步工作中,我们可以进一步研究如何选择恰当数目的种子和最优的种子组合,进而得到最佳的寻找路径时间和最优的扩展性能.

4.6 种子个数和不同的种子组合对性能的影响

在这一小节,我们主要研究种子个数和不同的种子组合对实体集扩展性能的影响.为了研究种子个数对实体集扩展性能的影响,我们分别在 Movie*和 Scientist*任务上进行实验,从 2~6 变化种子数目,对不同种子数目,随机选择相同规模的种子进行实验 20 次取 MAP 的平均值,结果如图 6(a)所示.



(a) 种子数目对性能的影响 (b) 种子组合对性能的影响

Fig.6 Influence of seed size and different seed combinationon expansion performance

图 6 种子个数和不同的种子组合对实体集扩展性能的影响

从图中可以看出:在 Movie*任务上,随着种子数目的增加,FPMP_ESE_PU 的性能有稳定提升,FPMP_ESE_He 有一些震荡但整体性能是提升的,说明太少的种子数(如两个)包含较少的语义信息,其性能是较差的;当种子数增多时,语义信息比较丰富,性能就比较好.在 Scientist*任务上,FPMP_ESE_PU 和 FPMP_ESE_He 性能都比较好,原因可能是其语义类比较单一、明确.总之,越多的种子包含更多的信息,可以更好地表达潜在的语义,对算法

找到重要的元路径有更大的帮助;当种子数目增加到一定值时,性能趋于稳定.

为了研究不同的种子组合对性能的影响,我们分别在 Actor*和 Movie*任务上随机选择 3 个种子进行实验 20 次,取 MAP 的平均值,结果如图 6(b)所示.从图中可以看出:在两个任务上,最好和最差的性能之间有一个较大的差别.对 Actor*任务来说,最差的性能甚至不到 0.2,最好的性能接近 1.0.因此我们可以看出,不同的种子组合对结果有一个比较大的影响.可见:选择较好的种子对扩展性能是很重要的,那些劣势的种子应该被淘汰.接下来,我们将进一步研究如何选择最优的种子组合得到最佳的结果.

5 总 结

本文主要研究知识图谱中的实体集扩展问题,即:给定几个种子实体,利用知识图谱来得到更多的同类别的实体.具体地,我们把知识图谱建模成一个异质信息网络,采用元路径来探测种子实体之间潜在的共同特征.为了找到种子实体之间的重要的元路径,我们采用频繁模式挖掘技术,提出了一种新的自动寻找元路径的方法 FPMPG.FPMPG 把每个种子实体映射为一个实体事务,首先找到种子实体的频繁模式,然后连接频繁模式得到重要元路径.为了更好地组合元路径,我们设计了两种权重学习方法:一种是启发式方法,另一种是 PU learning 方法.最后,在 Yago 数据集上的实验,验证了所提方法较其他基本方法有更好的有效性以及更高的效率,并且研究了种子个数和不同的种子组合对实体集扩展性能的影响.在未来的工作中,我们将进一步研究实体集扩展问题中如何确定恰当的种子数目以及如何选取最优的种子.

References:

[1] Cohen WW, Sarawagi S. Exploiting dictionaries in namedentity extraction: Combining semi-Markov extraction processesand data integration methods. In: Proc. of the KDD. ACM Press, 2004. 89−98.

[2] Pantel P, Lin D. Discovering word senses from text. In: Proc. of the 8th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2002. 613−619.

[3] He Y, Xin D. Seisa: Set expansion by iterative similarity aggregation. In: Proc. of the WWW. ACM Press, 2011. 427−436.

[4] Wang RC, Cohen WW. Language-Independent set expansionof named entities using the Web. In: Proc. of the ICDM. IEEE, 2007. 342−350.

[5] Wang RC, Cohen WW. Iterative set expansion of named entities using the Web. In: Proc. of the ICDM. IEEE, 2008. 1091−1096.

[6] Li XL, Zhang L, Liu B, Ng SK. Distributional similarityvs. PU learning for entity set expansion. In: Proc. of the ACL. ACL Press, 2010. 359−364.

[7] Qi ZY, Liu K, Zhao J. A novel entity set expansion method leveraging entity semantic knowledge. Journal of Chinese Informantion Processing, 2013,27(2):1−10 (in Chinese with English abstract).

[8] Sun Y, Han J, Yan X, Yu PS, Wu T. Pathsim: Meta path-based top-*k* similarity search in heterogeneous information networks. Proc. of the VLDB Endowment, 2011,4(11):992−1003.

[9] Zheng Y, Shi C, Cao X, Li X, Wu B. Entity set expansion with meta path in knowledge graph. In: Proc. of the Pacific-Asia Conf. on Knowledge Discovery and Data Mining. Cham: Springer-Verlag, 2017. 317−329.

[10] Singhal A. Introducing the knowledge graph: Things, not strings. In: Proc. of the Official Google Blog. 2012.

[11] Lenat DB. CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM, 1995,38(11):33−38.

[12] Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: A collaboratively createdgraph database for structuring human knowledge. In: Proc. of the 2008 ACM SIGMOD Int'l Conf. on Management of Data. New York. ACM Press, 2008. 1247−1250.

[13] Suchanek FM, Kasneci G, Weikum G. YAGO: A core of semantic knowledge unifying word netand wikipedia. In: Proc. of the 16th Int'l Conf. on World Wide Web. New York: ACM Press, 2007. 697−706.

[14] Dong XL, Murphy K, Gabrilovich E, Heitz G, Horn W, Lao N, Strohmann T, Sun SH, Zhang W. Knowledge vault: A Web-scale approach to probabilisticknowledge fusion. In: Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2014. 601−610.

[15] Paulheim H, Bizer C. Type inference on noisy RDF data. In: Proc. of the Semantic Web (ISWC 2013). LNCS 8218, Berlin, Heidelberg: Springer-Verlag, 2013. 510−525.

[16] Socher R, Chen DQ, Manning CD, Ng A. Reasoning with neural tensor networks for knowledge base completion. In: Proc. of the Advances in Neural Information Processing Systems 26 (NIPS 2013). Curran Associates, Inc., 2013. 926−934.

[17] Zhao Y, Gao S, Gallinari P, Guo J. Knowledgebase completion by learning pairwise-interaction differentiate dembeddings. Data Mining and Knowledge Discovery, 2015,29(5):1486−1504.

[18] Bryl V, Bizer C. Learning conflict resolutionstrategies for cross-language wikipedia data fusion. In: Proc. of the Companion Publication of the 23rd Int'l Conf. on World Wide Web Companion. Geneva: Int'l World Wide Web Conf. Steering Committee, 2014. 1129−1134.

[19] Paulheim H, Bizer C. Improving the qualityof linked data using statistical distributions. Int'l Journal on Semantic Web and Information Systems (IJSWIS), 2014,10(2):63−86.

[20] Zou L, Huang R, Wang H, Yu JX, He W, Zhao D. Natural language question answering over RDF: A graph datadriven approach. In: Proc. of the SIGMOD. ACM Press, 2014. 313−324.

[21] Cao X, Zheng Y, Shi C, Li J, Wu B. Link prediction in schema-rich heterogeneous information network. In: Proc. of the Pacific-Asia Conf. on Knowledge Discovery and Data Mining. Springer Int'l Publishing, 2016. 449−460.

[22] Nickel M, Murphy K, Tresp V, Gabrilovich E. A review of relational machine learning for knowledge graphs. Proc. of the IEEE, 2016,104(1):11−33.

[23] Sun Y, Yu Y, Han J. Ranking-Based clustering of heterogeneous information networks with star network schema. In: Proc. of the KDD. 2009. 797−806.

[24] Shi C, Li Y, Zhang J, Sun Y, Yu PS. A survey on heterogeneous information network analysis. IEEE Trans. on Knowledge and Data Engineering, 2017,29(1):17−37.

[25] Shi C, Kong X, Huang Y, Philip SY, Wu B. HeteSim: A general framework for relevance measure in heterogeneous networks. IEEE Trans. on Knowledge & Data Engineering, 2014,26(10):2479−2492.

[26] Agrawal R, Srikant R, *et al*. Fast algorithms for mining associationrules. In: Proc. of the 20th Int'l Conf. Very Large Data Bases, Vol.1215. VLDB, 1994. 487−499.

[27] Han J, Pei J, Yin Y. Mining frequent patterns withoutcandidate generation. ACM SIGMOD Record, 2000,29(2):1−12.

[28] Rakesh A, Srikant R. Mining sequential patterns. In: Proc. of the 11th Int'l Conf. on Data Engineering. IEEE, 1995.

[29] Abedjan Z, Naumann F. Improving RDF data through associationrule mining. Datenbank-Spektrum, 2013,13(2):111−120.

[30] Jiang T, Tan AH. Mining RDF metadata for generalized association rules. In: Proc. of the Int'l Conf. on Database and Expert Systems Applications. Springer-Verlag, 2006. 223−233.

[31] Pasca M. Weakly-Supervised discovery of named entities using Web search queries. In: Proc. of the CIKM. ACM Press, 2007. 683−690.

[32] Jindal P, Roth D. Learning from negative examples in setexpansion. In: Proc. of the ICDM. IEEE, 2011. 1110−1115.

[33] Yu X, Sun Y, Norick B, Mao T, Han J. User guided entitysimilarity search using meta-path selection in heterogeneous information networks. In: Proc. of the CIKM. ACM Press, 2012. 2025−2029.

[34] Metzger S, Schenkel R, Sydow M. Qbees: Query by entityexamples. In: Proc. of the CIKM. ACM Press, 2013. 1829−1832.

[35] Metzger S, Schenkel R, Sydow M. Aspect-Based similar entity search in semantic knowledge graphs with diversity-awareness and relaxation. In: Proc. of the CWI and IAT. IEEE Computer Society, 2014. 60−69.

[36] Chen J, Chen Y, Du X, Zhang X, Zhou X. Seed: A systemfor entity exploration and debugging in large-scale knowledgegraphs. In: Proc. of the ICDM. IEEE, 2016. 1350−1353.

[37] Zhang J, Tang J. Focus of the next generation search engineer: Knowledge graph. Chinese Computer Society Communication, 2013, 9(4):64−68 (in Chinese with English abstract).

[38] Zou L, Chen YG. Massive RDF data management. Chinese Computer Society Communication, 2012,8(11):32−43 (in Chinese with English abstract).

[39] Aggarwal CC, Han J. Frequent Pattern Mining. Springer-Verlag, 2014.

[40] Elkan C, Noto K. Learning classifiers from only positive and unlabeled data. In: Proc. of the 14th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2008. 213−220.

[41] Shi B, Zhang ZZ, Sun L, Han XP. A probabilistic co-bootstrapping method for entity set expansion. In: Proc. of the 25th Int'l Conf. on Computational Linguistics (COLING 2014), Proc. of the Conf.: Technical Papers. Dublin, 2014. 2280−2290.

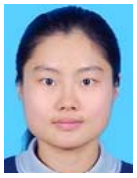
[42] Lao N, Cohen WW. Relational retrieval using a combination of path-constrained random walks. Machine Learning, 2010,81(1): 53−67.

附中文参考文献:

[7] 齐振宇,刘康,赵军.一种融合实体语义知识的实体集合扩展方法.中文信息学报,2013,27(2):1−10.

[37] 张静,唐杰.下一代搜索引擎的焦点:知识图谱.中国计算机学会通讯,2013,9(4):64−68.

[38] 邹磊,陈跃国.海量 RDF 数据管理.中国计算机学会通讯,2012,8(11):32−43.



郑玉艳(1992－),女,山东诸城人,博士生,主要研究领域为异质信息网络数据挖掘.

田莹(1996－),女,本科生,主要研究领域为数据挖掘.

石川(1978－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据挖掘,机器学习,演化计算.