基于随机游走的实体类型补全方法

张香玲[1),2)] 陈跃国[1),2)] 毛文祥[1),2)] 荣垂田[3)] 杜小勇[1),2)]

[1)](数据工程与知识工程教育部重点实验室(中国人民大学) 北京 100872)

[2)](中国人民大学信息学院 北京 100872)

[3)](天津工业大学计算机科学与软件学院 天津 300387)

摘 要 伴随着大数据的大量涌现以及开放链接数据(LOD)等项目的开展,语义网知识库的数量激增,语义网知识库正在引起学术界和工业界越来越多的关注,在信息检索系统中起着重要的作用,如实体搜索和问答系统等.实体类型信息在信息检索中扮演着重要的角色,例如,查询"汤姆·汉克斯所出演的电影",该查询限定了返回的实体类型是"电影",这对提高查询结果的精度具有重要作用.然而,知识库中实体类型信息的缺失是十分严重的,影响了知识库在信息检索等领域中使用的正确性和广泛性.据统计,在 DBpedia2014 中,8%的实体没有任何类型信息,28%的实体只有高度抽象的类型信息(比如类型为"Thing"),因此对于实体类型补全的研究尤其是实体细粒度类型的补全是十分重要的.目前已有的方法包括基于概率模型和表示学习两类.以基于概率模型的 SDType 算法为例.首先,SDType 为每个谓词计算对各个类型的区分能力得分,然后,在为实体做类型补全时,累加该实体所具有的谓词对各个类型的得分.此类方法没有考虑谓词与谓词之间的相互增强作用,在存在知识缺失的情况下会影响补全效果.以表示学习的类型补全方法 TransE 为例,此方法对于简单的关系(1-1 的关系)补全是可以的,但是对于补全实体类型这种复杂的关系效果并不理想.另外,表示学习的训练集尤其是负例难以获得.由于模型需要学习大量的参数,在大数据量的背景下,性能也是一个问题.文中提出一种基于谓词-类型推理图的随机游走方法来补全缺失的实体类型.首先对知识库中已有知识进行统计,包括具有某个谓词的实体数目、属于某个类型的实体数目以及属于某个类型并且具有某个谓词的实体数目.其次,基于得到的统计信息构建结点由谓词和类型组成的有向推理图,推理图的边包括谓词-谓词和谓词-类型两种.在构建推理图时,作者考虑了谓词之间的相互增强作用,在类型补全中是有效果的,尤其是在知识库存在知识缺失的背景下.最后,对于一个缺失类型信息的实体,根据该实体所具有的谓词在推理图上做随机游走来补全类型.为了解决由于知识库中存在错误知识等原因导致的类型语义漂移现象,文中使用 PMI(点互信息)技术对结果进行了进一步的优化.在真实 DBpedia 知识库上的实验,验证了文中提出的算法相比于已有的典型算法有更高的精确度.

关键词 知识库;类型补全;图模型;随机游走;大数据

中图法分类号 TP391 **DOI 号** 10.11897/SP.J.1016.2017.02352

An Entity Type Completion Algorithm Based on Random Walk

ZHANG Xiang-Ling[1),2)] CHEN Yue-Guo[1),2)] MAO Wen-Xiang[1),2)]

RONG Chui-Tian[3)] DU Xiao-Yong[1),2)]

[1)](*Key Laboratory of Data Engineering and Knowledge Engineering of Ministry of Education, Renmin University of China, Beijing* 100872)

[2)](*School of Information, Renmin University of China, Beijing* 100872)

[3)](*School of Computer Science and Software Engineering, Tianjin Polytechnic University, Tianjin* 300387)

Abstract Nowadays, semantic web knowledge bases are more and more prevalent because the wide usage of linking open data (LOD). They play an important role in IR systems, especially in entity search systems and question answering systems. An intuition is that the entity's type

收稿日期:2016-06-13;在线出版日期:2017-01-10.本课题得到国家自然科学基金(61472426,61402329)资助.**张香玲**,女,1983 年生,博士研究生,中国计算机学会(CCF)会员,主要研究方向为实体搜索、知识补全.E-mail:zhangxiangling@ruc.edu.cn.**陈跃国**(通信作者),男,1978 年生,博士,副教授,中国计算机学会(CCF)高级会员,主要研究方向为大数据实时分析系统、知识图谱和语义搜索.E-mail:chenyueguo@ruc.edu.cn.**毛文祥**,男,1994 年生,硕士研究生,主要研究方向为知识图谱、实体搜索.**荣垂田**,男,1981 年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为数据库与信息检索、大数据与云计算.**杜小勇**,男,1963 年生,博士,教授,博士生导师,中国计算机学会(CCF)会士,主要研究领域为数据库系统、智能信息检索等.

information is very important for IR tasks. For example, an entity search query "movies in which Tom hanks plays a role" requires results of the type movie. Unfortunately, the lack of type constraints for entities is very serious in knowledge bases, which affects the correctness and universality of the use of the knowledge base in the field of information retrieval etc. Our investigation shows that in DBpedia 2014, 8% entities do not have any type information and 28% entities only have coarse types (such as "Thing"). How to complete the type constraints especially the fine-grained types for entities in knowledge bases is a critical task. Some studies propose to complete entity's type constraints in the knowledge base, such as probabilistic distributional model-based methods and representation learning methods. Take a probabilistic-based approach SDType as an example. Firstly, SDType calculates the weight of each predicate for each type which describes the discriminability of a predicate for a type. Then, the score of a certain type for an entity is basically an aggregation of the scores of all predicates that the entity has. Such methods do not consider the mutually reinforcing effect between predicates, which may affects the accuracy of type completion in the absence of knowledge base. One typical method of representation learning is TransE which is suitable for simple relations but not for complex relations such as type. Another problem of representation learning methods is that the training data is difficult to obtain, especially the negatives. Moreover, due to the large number of parameters in the model, the efficiency is also a big problem for these kinds of methods. In this paper, we propose a novel way to complete type information of entities by using a random-walk-based iterative algorithm on a probabilistic graph. First of all, we calculate the number of entities with a certain predicate, the number of entities of a certain type, and the number of entities of a certain type and with a certain predicate within the knowledge base. Secondly, we build a predicate-type probabilistic graph. There are two classes of nodes in the probabilistic graph which are predicate nodes and type nodes. The edges can be classified into two categories: predicate-predicate ones and predicate-type ones. We consider the mutual enhancement between predicates. It is very helpful for the entity's type completion, especially when the knowledge base is not complete. Finally, an improved random walk strategy of the entity type completion in the predicate-type probabilistic graph is proposed. In order to solve the problem of semantic drift, which is caused by the reasons such as errors in the knowledge base and so on, we apply PMI (Point Mutual Information) technology to optimize the results. The experiments on a popular knowledge base show that our algorithm achieves higher accuracy compared with the existing typical methods.

Keywords　knowledge base; type completion; graph model; random walk; big data

1　引　言

语义网知识库(以下简称知识库)是对现有Web的延伸,信息被赋予良好的含义,从而使计算机可以更好地和人协同工作,它的基础是资源描述框架(Resource Description Framework,RDF)[1].随着开放链接数据(Linking Open Data,LOD)[2]等项目的开展,语义网知识库的数量激增,大量的RDF数据被发布.LOD项目的目的在于构建机器可理解的包含丰富语义关系的数据网,号召将数据按照一定规则发布到Web中,并将不同的数据源关联起来.从2007年10月第一次链接数据发布以来,其规模呈爆炸式增长.截止2014年4月,链接数据已经包含1014个数据集,涉及到的领域包括地理信息数据、政府数据、生命科学[3]等等.互联网正从传统的网页与网页之间超链接的文档万维网转变成包含丰富语义的实体与实体链接的数据万维网.具有代表性的知识库包括基于维基百科等构建的知识库,如DBpedia[4],Freebase[5],Yago2[6],一些企业也有自己的知识库服务于搜索引擎,如谷歌的知识图谱Knowledge Graph、Facebook推出的图谱搜索服务Graph Search、微软的概率知识库Probase及Bing搜索引擎的Satori等.在中文方面,百度的"知

心”和搜狗的“知立方”等，这些知识库在实体搜索、问答系统、推荐系统等领域都有所应用[7-9]，知识库在搜索任务中起着越来越重要的作用.

一般而言，RDF 知识库可以表示为三元组的集合，三元组的形式为〈主语，谓词，宾语〉，知识库也被称为知识图谱. 图 1 为知识库的一个示例，示例三元组〈Forrest_Gump，starring，Tom_Hanks〉表示实体 Forrest_Gump 和 Tom_Hanks 之间的关系是“starring”. 谓词具有方向性，实体 Forrest_Gump 具有的谓词是“starring”的出边，也称为实体 Forrest_Gump 具有谓词“starring”，实体 Tom_Hanks 具有的谓词是“starring”的入边，也称为实体 Tom_Hanks 具有谓词“$starring^{-1}$”.

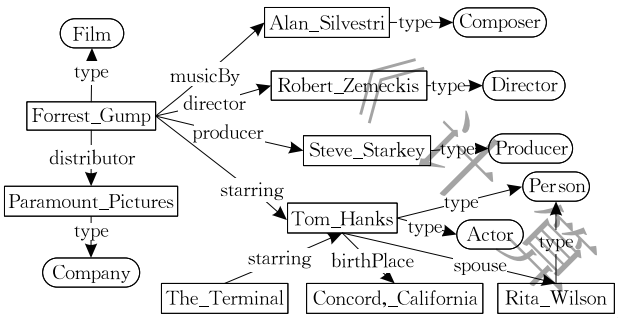

图 1 知识库示例

对于一个实体，类型是它最基本也是最重要的语义信息. 实体的类型信息在许多应用中发挥着非常重要的作用. 例如，实体的类型匹配对于实体搜索查询[10]至关重要，有时用户在输入查询时会限定答案的类型[11-12]，比如查询“movies which Tom hanks plays a role”，该查询是 INEX09[13]中的一个查询，查询限定了返回的实体类型是“Movie”. 另外在推荐系统中也会根据条目的类型给用户做出推荐，比如根据用户的浏览历史，分别推荐出电影、书籍等不同类型和浏览历史相关度较高的实体. 一个实体可能有许多粒度不同的类型. 例如，实体 Tom_Hanks 有“Thing”、“Person”、“Artist”、“Actor”、“Director”等几种类型. 其中，细粒度的类型如“Director”和“Actor”，粗粒度的类型如“Thing”、“Person”. 一个实体的类型越具体，其信息量越大，也越有意义. 然而在知识库中实体类型的缺失是非常严重的，尤其是实体细粒度类型的缺失，这在某种程度上影响了知识库应用的广泛性和有效性. 据我们统计，在 DBpedia 2014 版本中，实体总数约为 458 万个，8%的实体没有任何的类型信息，只有 64%的实体拥有细粒度的类型. 因此，进行实体类型补全的研究是重要的也是必要的.

实体类型补全现有的工作主要包括两类方法. 一类是根据实体所具有的不同谓词对实体类型做补全. 首先计算不同谓词对类型的区别能力，然后根据实体所拥有的谓词推断出该实体所属的类型[14]. 这种实体类型补全方法假设谓词之间相互独立，而现实世界中这种假设在很多情况下并不成立. 比如，很多实体既属于类型“Actor”同时也属于类型“Director”，同时具有“$starring^{-1}$”和“$director^{-1}$”两个谓词. 由于知识库中存在知识缺失，当需要补全的实体缺失某个谓词时，这种基于谓词相互独立的计算方法的准确性就会受到影响. 而知识库中的知识缺失是很严重的，比如在 DBpedia 2014 数据集中，类型为“Person”的实体有 1 445 104 个，但是具有“nationality”谓词的实体只有 104 913，也就是说，只有 7.26%的人具有“nationality”信息. 此外，该方法认为所有的类型也是相互独立的，没有考虑类型之间的层次结构. 具体地，如果实体具有谓词“$starring^{-1}$”，但是并没有可以表征类型为“Person”的谓词，比如“birthPlace”、“nationality”等，通过这个方法计算出该实体属于类型“Actor”的分值会很高，而属于类型“Person”的分值将会比较低，因为该实体缺失可以表征类型为“Person”的谓词，而且该方法没有用到类型的层次结构，不能根据类型“Person”与“Actor”的上下级关系补全类型“Person”.

另外一类是表示学习的方法. 其中基于张量分解的方法[15-19]是把知识库建模为一个三维张量，分别为实体-实体-关系三个维度，然后通过张量分解的方法实现类型补全. 根据封闭世界假设，知识库中存在的三元组即为正例，不存在的三元组视为负例. 另外，最近研究比较多的是将知识库的实体和谓词使用低维向量表示[20-24]. 然而，对于关系类型为 1-N、N-1 和 N-N 模式的复杂谓词（比如“birthPlace”是一个 N-1 的谓词，因为一个地方会有很多人都在此出生），由于 TransE[20]方法模型过于简单，实现出来的效果非常差[21-22]，其他方法例如 TransR[21]、PTransE[22]和 TransH[24]也在这一系列工作中被提出. 然而，这两种方法在训练模型时都依赖于训练数据，负例数据是难以获得的. 此外，这类方法缺乏解释性，因为这些基于向量操作的推理机制都是隐含的，并不能直观展现推理过程. 如何更好地综合考虑谓词和谓词之间、谓词和类型之间的相互作用进而获得更加准确的实体类型补全效果，并且显式地呈现推理过程是我们主要考虑的问题.

基于以上的相关工作和分析，我们发现根据实

体的谓词在一定程度上可以推理出实体具有的类型；另外，谓词和谓词之间并不都是相互独立的，有的谓词之间共现特别频繁，这对于存在知识缺失时的类型补全具有一定的帮助. 据此，我们提出了一种结点由谓词和类型构成的推理图模型，利用图模型中的随机游走算法为实体做类型补全.

本论文主要贡献有以下 3 点：

(1) 提出了一种由谓词和谓词及谓词和类型的相互作用补全实体类型的模型；

(2) 为了解决类型语义漂移，使用 PMI 技术设计了一个有效的谓词—类型推理图及基于图上的随机游走算法；

(3) 在开放的数据集上进行大量的实验，实验结果显示该方法优于目前已有的典型方法.

本文第 2 节介绍相关的研究工作；第 3 节阐述一些关于知识库、知识库模型和基本层次类型的相关概念；第 4 节介绍我们设计的 GBTC(Graph-Based Type Completion)算法及其优化算法 GBTC-PMI；第 5 节是对方法的评测分析；第 6 节是论文的总结和展望.

2　相关工作

目前实体类型补全最有代表性的方法是 SDType[14]. 该方法的基本思想是使用实体所具有的谓词推断实体所属的类型，可以看做是利用实体所具有的每个谓词对实体可能属于的类型进行投票，如式(1)：

$$s(c,e)=\sum_{\text{all properties } p \text{ of entity } e} w(p)\cdot P(c|p) \quad (1)$$

其中，$P(c|p)$表示具有谓词 p 的实体属于类型 c 的比率.

由于知识库中各个类型具有的实体数目不均衡，如果仅仅使用 $P(c|p)$补全实体类型，补全结果会偏向于返回包含更多实体的类型，尤其是当实体具有一些常见谓词，如“name”、“label”. 对具有同一个谓词的实体集合中，属于类型 c 的实体越多，$P(c|p)$越大. 基于此，定义谓词权重 $w(p)$，刻画谓词对于类型的预测能力，使用知识库中类型的先验概率分布与有谓词限定的条件概率分布的偏差来度量. 偏差越大，谓词的预测能力就越强. 另外，考虑谓词具有方向性，同一个谓词对于主语和宾语的类型区分能力可能也是不同的. 谓词权重计算方法如式(2).

$$w(p)=\sum_{\text{all types } c \text{ in } KB}(P(c)-P(c|p))^2 \quad (2)$$

其中 $P(c)$表示属于该类型的实体在整个知识库中占的比重.

根据如上对 SDType 方法的介绍，在计算某个实体 e 属于类型 c 的得分时，谓词是相互独立的. 具体地，如果实体 e 具有谓词“birthPlace”和“starring^{-1}”，那在计算该实体属于类型“Person”的得分时，两个谓词分别对类型“Person”贡献得分；另外的一个实体 s 由于知识缺失只具有谓词“starring^{-1}”，那在计算 s 与类型“Person”的得分时，就只有谓词“starring^{-1}”被用来计算属于类型“Person”的得分，这样 s 属于类型“Person”的得分就会很小. 而在知识库中，具有谓词“starring^{-1}”的实体集合中同时具有“birthPlace”这个谓词的比率是非常高的，SDType 没有利用这种谓词之间的相互增强作用做类型补全，在有知识缺失的情况下，类型补全的准确率就会受到影响. 另外，SDType 在计算某一实体类型时，没有考虑谓词出现的频率. 例如，Tom_Hanks 具有谓词“starring^{-1}”47 次，而谓词“director^{-1}”只出现了 5 次，表示 Tom_Hanks 主演了 47 部电影并导演了 5 部电影，这些都能强烈地表明 Tom_Hanks 的主要类型是一个“Actor”，然后是一个“Director”.

最近，受到 word2vec[25] 的启发，很多研究[20-24]将知识库中的实体和谓词转化为低维向量. 以第一个将表示学习应用在知识库中的 TransE 方法[20]为例. TransE 的目标函数为式(3)：

$$f(s,p,o)=\|s+p-o\|_2^2 \quad (3)$$

在为谓词和实体学习向量时，如果三元组$\langle s,p,o\rangle$存在，其中 s,o 为实体，p 为谓词，TransE 将 $f(s,p,o)$最小化，否则将其最大化. 可是对于复杂的谓词，例如 1-N，N-1 和 N-N 这样的谓词，TransE 的实现效果很差. 以 N-N 型的谓词“starring^{-1}”为例，三元组〈Catch_Me_If_You_Can, starring, Tom_Hanks〉和〈Catch_Me_If_You_Can, starring, Leonardo_DiCaprio〉都是知识库中的三元组条目，根据 TransE 的目标函数，会将实体 Catch_Me_If_You_Can 和谓词 starring 的向量之和与 Tom_Hanks 以及 Leonardo_DiCaprio 的向量接近，也就是说 Tom_Hanks 和 Leonardo_DiCaprio 的向量很接近，但是显然 Tom_Hanks 和 Leonardo_DiCaprio 是两个不同的实体，虽然他们共同参演了 Catch_Me_If_You_Can，但是他们的“birthDate”、“birthPlace”、

"alumni"、"Person/height"等等都各不相同.

以张量分解为基础的解决方法,将知识库视为一个张量,张量的三个维度分别是实体-实体-谓词.通过文献[15-19]提出的张量分解算法计算出知识库中不存在的三元组的得分.以 RESCAL[17] 为例,知识库三元组构成一个大的张量 Y,如果三元组 $\langle s,p,o\rangle$ 存在于知识库中,则 $Y_{spo}=1$,否则为 0. RESCAL 将高维张量分解成三个部分的乘积,一个是矩阵,一个是低维的核心张量以及第一个矩阵的转置矩阵.可是,张量分解对计算成本和内存要求都非常高,尤其是对于那些拥有数以百万计的实体和数以千计的谓词的知识库来说.

张量分解和使用低维向量表示实体和关系的两种方法的解释性都很差,因为所有的这些推理机制都是隐含的.

还有一类工作是在文献[26]中提出的实体概念化模型.这类工作主要是基于文本数据中的共现统计,对实体找到最合适的类型.文中提出实体 e 最合适的类型 $C(e)$ 计算方法为

$$C(e)=\arg\max_{c} Rep(e,c) \tag{4}$$

其中

$$Rep(e,c)=P(c|e)\cdot P(e|c),$$

$$P(c|e)=\frac{n(c,e)}{\sum_{e\in c_i} n(c_i,e)},$$

$$P(e|c)=\frac{n(c,e)}{\sum_{e_i\in c} n(c,e_i)}.$$

$n(c,e)$ 表示实体 e 与类型 c 同时出现的网页数量.比如,通过在文本中进行匹配,实体"Microsoft"具有 3 个类型"Company"、"Software_Company"和"The_largest_OS_vendor".提到"Microsoft"首先想到它是一个"Company",提到"The_largest_OS_vendor",首先想到的是"Microsoft".但是,文献[26]认为"Company"和"The_largest_OS_vendor"都不是描述实体"Microsoft"最好的类型,类型"Company"粒度太粗,而"The_largest_OS_vendor"粒度太细.因为,如果要找与"Microsoft"相似的实体,在"Company"中会找到很多比如"The_Coca-Cola_Company"、"Honda"等;但是如果在"The_largest_OS_vendor"中找,可能根本找不到其他任何实体(因为粒度太细,只有"Microsoft"一个实体属于该类型).而通过"Software_Company"则可以找到"IBM"、"Oracle"等与"Microsoft"类似的其他实体.实体概念化模型的工作[26]偏重于找出那些粒度不是太细、也不是太粗的类型,在第 3 节定义 3 中我们称之为基本层次类型[27-28].

实体概念化模型与实体类型补全具有如下区别:(1)文献[26]中的概念(类型)是从 16.8 亿网页中获得,在句子中提取匹配 Hearst[29] 模式的 isa 关系组成.例如,从句子"Albert Einstein is a world famous German-born American physicist."中,提取出"Albert Einstein"属于类型"world famous Germany-born American physicist".而本文提到的实体类型补全中的所有类型都是由知识库的本体提供的类型,无需再从文本中抽取;(2)基于第一点区别,知识库的本体提供的类型没有像"world famous Germany-born American physicist"这样特别细粒度的类型.而且由于网页的多样性、复杂性和丰富性,在 Probase 中,有 3024814 个概念(类型)[26];而在 DBpedia 2014 数据集中,仅有 685 个类型;(3)类型补全是希望将最细粒度的类型补全,同时结合类型的层次结构,将实体具有的抽象或者说上级类型也补全.而实体概念化模型中的共现统计都是在文本语料库中统计的,比如 $n(c,e)$ 统计的就是类型 c 与实体 e 共现的次数,而在知识库中由于没有重复的三元组,如果 $\langle e,type,c\rangle$ 存在于知识库中,则 $n(c,e)=1$,否则 $n(c,e)=0$,所以不能直接利用实体概念化模型补全实体的类型.

3 预备知识

在本节,给出论文中用到的概念和符号的定义.首先,我们对知识库做定义如下.

定义 1. 知识库.将知识库记作 $K_{kb}=\{E,U_e,P_e,\tau\}$,其中:$E$ 是实体集合,U_e 是连接实体与实体有向边的集合,P_e 是实体之间谓词的集合,τ 定义了从边映射到谓词的映射函数.每个谓词表示两个实体之间的关系.$\tau(s,o)\rightarrow p$,用三元组表示为 $\langle s,p,o\rangle$,即实体 s 和 o 的关系是 p,同时 $s\in E$,$o\in E$,$p\in P_e$,$\langle s,o\rangle\in U_e$.另外,谓词是具有方向性的,三元组 $\langle s,p,o\rangle$ 还可以表示为 $\langle o,p^{-1},s\rangle$.

如图 1 为知识库示例,图中的结点包括实体、实体的类型,谓词"type"表示实体所属的类型信息,图中出现的类型有:"Actor"、"Director"、"Producer"、"Film"、"Composer"以及"Company"和"Person".结点之间的连边表示实体之间的关系.

知识库提供了类型的本体,它是一个层次结构,如 DBpedia,以类型"Thing"作为根,"subclassof"

表示两个类型之间的上下位关系. 例如，〈Writer，subclassof，Person〉表示类型“Writer”是“Person”的子类. 另外，类型和类型之间还有其他关系，比如，〈Person，nationality，Country〉，我们称这样的本体为知识库模型，定义如下.

定义 2.　知识库模型. 将知识库模型记作 $K_{ks}=\{T,U_t,P_t,\pi\}$，其中，T 是类型集合，U_t 是类型与类型之间有向边的集合，P_t 是类型之间谓词的集合，π 定义了从边映射到谓词的映射函数. 每个谓词都表示两个类型之间的关系.

图 2 给出了一个知识库中部分类型的知识库模型. 在知识库模型中类型是一个层次结构，类型之间的关系是可以继承的. 比如〈Person，nationality，Country〉，而〈Writer，subclassof，Person〉，所以类型“Writer”也具有谓词“nationality”.

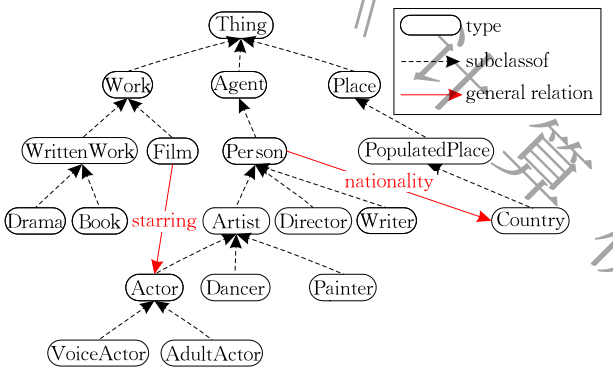

图 2　知识库中部分类型的知识库模型示例

在知识库类型层次结构中层级越深的类型越具体. 例如，类型“Person”比“Thing”更具体，而“Actor”比“Person”更具体. 在所有的类型中，有一些类型的子类型之间具有最大的相似性，而这些类型与其他类型差异性也最大，我们定义这种类型为基本层次类型.

定义 3.　基本层次类型. 在认知心理学范畴，类型大致分为 3 个级别[27-28]，分别为上级层次类型、基本层次类型和下级层次类型. 相对于上级层次类型和下级层次类型，人们优先选择基本层次类型. 而且，人们辨认基本层次类型的速度比其他两者都要快[27].

例如，当看到一只狗，大多数会向幼儿说“那是一条狗”，而不会说“腊肠狗”这样的下级层次类型或“动物”这样的上级层次类型. 在这个例子中，狗是基本层次类型. 图 3 是 3 种层次类型的示例. 在上级层次类型，我们很难识别两个类型之间的相似性，而在下级层次类型，我们又很难添加一些重要的特征来区分. 另外，一只“狗”的图片很容易画，但要画“动物”这种上级层次类型或者下级层次类型“腊肠狗”会比较困难，也就是说基本层次类型比上级层次类型和下级层次类型更容易图形化. 简而言之，在基本层次类型中，其子类型之间的相似性最大化，并且与其他基本层次类型的相似性最小化. 另外，类型的级别分类是相对的，比如，一般意义上说，“狗”是一个基本层次类型，但是对于动物学家来说“狗”可能是一个上级层次类型. 而且各个级别的层次类型也可能是多层，比如图 3 中，上级层次类型就包括两层的类型，下级层次类型“Artist”也包括两层的类型.

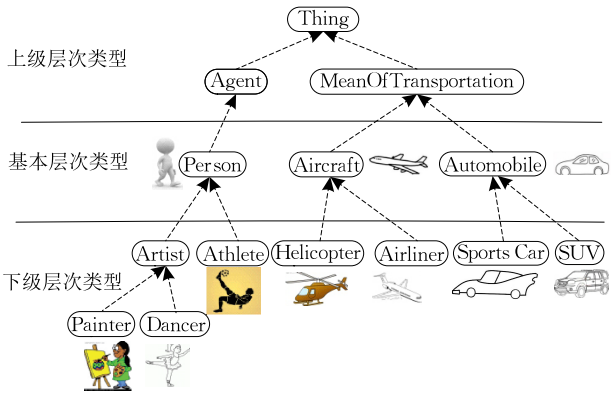

图 3　层次类型示例

引入基本层次类型的意义在于，对于实体类型补全问题，如果补全的类型是上级层次类型，这是没有意义的，因为它太抽象. 例如，补全实体 Tom_Hanks 的类型，基本层次类型“Person”和下级层次类型“Actor”可以更加精确地描述实体 Tom_Hanks，而上级层次类型“Thing”是没有意义的. 虽然一个实体的类型可以出现在上级层次、基本层次、下级层次，但是用下级层次类型或者至少是基本层次类型来对给定的实体进行类型补全是更有意义的，尤其是在推荐系统、实体搜索等实际应用场景中.

下面我们对实体类型补全问题做形式化定义.

定义 4.　实体类型补全. 给定知识库 K_{kb} 和知识库模型 K_{ks}，对于需要做类型补全的实体 $e\in E$，在知识库模型的类型集合 T 中找出与该实体最相关的类型并排序.

知识库中实体的类型具有如下特点：

(1) 由于知识库中的类型体系本身也有缺失，所以对于一个实体并不一定必须具有知识库模型 K_{ks} 中叶子结点类型或者说最细粒度的类型. 比如，在图 2 中，由于“Actor”的子类型没有“MovieActor”，所以 Tom_Hanks 最细粒度的类型就是“Actor”，而不是“VoiceActor”或“AdultActor”.

(2)有的实体可能具有多个下级层次类型.比如,赵薇具有多种身份,是一个导演也是一名演员.

(3)由于类型是一个层次结构,对实体做类型补全,最重要的是把该实体的下级层次类型或者至少是基本层次类型补全,而如果仅仅补的是上级层次类型是没有意义的.

为了便于理解下文中实体类型补全方法,表1中展示了一些常用的符号.

表1 常用符号

符号	描述
e	知识库中的一个实体
c	知识库中的一个类型
p	知识库中的一个谓词
$n(e)$	实体 e 拥有的边的个数
$n(e,p)$	实体 e 拥有出边是谓词 p 的边的总数
$n(c)$	属于类型 c 的实体总数
$n(p)$	出边具有谓词 p 的实体总数
$n(p,c)$	出边具有谓词 p 并且属于类型 c 的实体总数
$n(p_1,p_2)$	出边具有谓词 p_1 和谓词 p_2 的实体总数
$n_p(e)$	实体 e 具有的谓词总数
N	知识库中所有实体的总数

4 基于随机游走的实体类型补全

正如前面所提到的,利用谓词和谓词之间的相互增强作用以及谓词与类型的关联做实体类型补全是很有效的,尤其是在知识缺失的情况下.本节我们将给出方法的技术性细节,这包括构建谓词-类型推理图以及利用此推理图实现实体类型补全的方法.

4.1 构建谓词-类型推理图

在本节,我们将要介绍构建谓词-类型推理图的过程.我们的目标是用统计的方法获得谓词与谓词以及谓词与类型之间的推理.谓词-类型推理图有两种结点,类型结点和谓词结点.结点之间的边有两种类型:(1)谓词与谓词之间的连边;(2)谓词和类型之间的连边;我们用下面的计算方法获得的概率给每一条边赋予特定的权重.

谓词与类型的连边权重用 $P(c|p)$ 表示,其含义为具有谓词 p 的实体集合中,属于类型 c 的概率.谓词与谓词的连边权重用 $P(p_2|p_1)$ 表示,其含义为具有谓词 p_1 的实体集合中,有多少比例的实体同时具有谓词 p_2.

这些概率计算方法为

$$P(c|p)=\frac{n(c,p)}{n(p)} \tag{5}$$

$$P(p_2|p_1)=\frac{n(p_1,p_2)}{n(p_1)} \tag{6}$$

图4是一个推理图示例,可以发现,如果一个实体具有"birthPlace"这个谓词,那么它属于类型"Person"的概率为 $P(Person|birthPlace)=0.999$,而且该实体拥有"birthDate"这个谓词的概率 $P(birthDate|birthPlace)=0.860$.在知识缺失的情况下,即使实体只有谓词"starring^{-1}",我们依然可以利用"starring^{-1}"与"birthDate"、"birthPlace"的相互增强作用做类型补全,因为模型中考虑了他们之间的相互增强作用.另外,由图4可以看出,知识缺失是很严重的.比如,具有"birthPlace"这个谓词的实体集合中,同时具有"nationality"的比率仅为0.119,具有"starring^{-1}"这个谓词的实体集合中,同时具有"nationality"这个谓词的只占0.040.

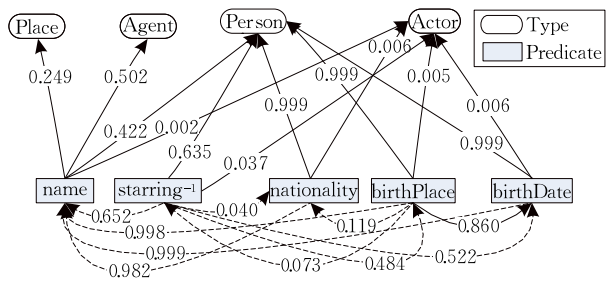

图4 谓词-类型推理图示例

如图4,在谓词-类型推理图中游走时,游走到类型的结点会被吸收(因为没有出边),每次迭代都是向类型结点注入增量的得分.这样做的目的就是为了防止走到上级层次类型时,出现类型语义漂移.如图4,谓词"starring^{-1}"、"nationality"、"birthPlace"和"birthDate"走到类型"Person"的概率要远大于"Actor",主要原因就是根据式(5),类型越抽象,$P(c|p)$ 的值越高,这样类型"Person"的得分就会很高,而如果考虑类型到谓词的出边,则由这种得分很高的抽象类型根据具体谓词出现的次数走到那些出现次数多的谓词上,这样就会偏离最初谓词对类型的判定,出现类型语义漂移问题.

4.2 类型补全

对于一个实体 e,它所具有的谓词为补全该实体可能具有的类型提供了很重要的提示.为了用谓词-类型推理图实现类型的补全,我们提出了一种基于随机游走的方法推理实体 e 可能属于的类型.本小节将介绍随机游走方法中用到的转移概率矩阵和初始概率分布的计算.

4.2.1 谓词-类型推理图的转移概率矩阵

我们用一个 $(m+n)\times(m+n)$ 的矩阵 $\boldsymbol{W}$ 表示谓词-类型推理图,其中 m 表示谓词的数目,n 表示类型的数目.如果两个结点 i 和 j 之间有边,那么矩阵

中对应元素 $w_{i,j}$ 的取值表示如式(7).

$$w_{i,j}=\begin{cases}P(p_2|p_1), & i=p_1 \text{且} j=p_2\\ P(c|p), & i=p \text{且} j=c\\ 0, & \text{其他}\end{cases} \tag{7}$$

我们定义从结点 i 到 j 的随机游走转移概率为 $P(j|i)$，它的大小为从结点 i 到 j 的边的权重的归一化值，由式(8)确定.

$$P(j|i)=\frac{w_{i,j}}{\sum_k w_{i,k}} \tag{8}$$

谓词-类型转移概率矩阵记作 $\boldsymbol{R}$，其中 $\boldsymbol{R}=[P(j|i)]_{ij}$，矩阵 $\boldsymbol{R}$ 的维数和矩阵 $\boldsymbol{W}$ 相同，也是 $(m+n)\times(m+n)$ 的规模，转移概率矩阵的每一行元素值的和为 1.

假设向量 $\boldsymbol{V}$ 表示谓词-类型推理图中每一个结点的权重，我们要应用随机游走的过程以便向量 $\boldsymbol{V}$ 可以收敛到向量 $\boldsymbol{V}^*$，从 $\boldsymbol{V}^*$ 中可以得到每个类型的得分值，这可以理解为给定一个实体所具有的谓词推出该实体所属类型的过程. $\boldsymbol{V}^n$ 表示在第 n 次迭代后所有结点权重的向量. 利用随机游走[30]更新结点的权重. 相应的，我们有

$$\boldsymbol{V}^n=\alpha\boldsymbol{R}\times\boldsymbol{V}^{n-1}+(1-\alpha)\boldsymbol{V}^0 \tag{9}$$

α 是一个调整初始向量 $\boldsymbol{V}^0$ 的权重的参数. 只要没有收敛或者迭代次数没有超过某个给定的阈值，这个过程将一直重复下去，如上过程就是一个标准的可以重新启动的随机游走，算法的收敛性也是可以得到证明的[31-32]. 此外，我们也可以获得该实体最相关的谓词或者说对实体缺失的谓词做补全.

4.2.2　谓词-类型概率图的初始概率分布

给定了谓词-类型转移概率矩阵之后，我们还需要建立谓词-类型概率图的初始概率分布. 初始向量 $\boldsymbol{V}^0$ 的值会影响着最终的收敛向量 $\boldsymbol{V}^*$ 的值. 因此，给 $\boldsymbol{V}^0$ 的每个元素赋予合理的权重是很重要的. 直接地，我们可以简单指定 $\boldsymbol{V}^0[v]=\frac{1}{n_p(e)}$，如果 v 是一个谓词结点且需要做类型补全的实体 e 具有谓词 v，$n_p(e)$ 指的是实体 e 所拥有的谓词的总数. 然而，这种简单的方法没有考虑谓词 v 相对于实体 e 的频率差异以及谓词 v 本身对于类型的区分度的差异. 我们在对初始向量 $\boldsymbol{V}^0$ 赋值时，考虑每个谓词在这个实体上出现的频率以及每个谓词对类型预测的不同区分度. 借用信息检索领域中经典 TF-IDF 模型，我们定义谓词的频率、谓词的逆实体频率.

谓词的频率 $f(e,p)$ 表示对于实体 e，谓词 p 是常见的还是罕见的，定义如式(10).

$$f(e,p)=\frac{n(e,p)}{n(e)} \tag{10}$$

这里 $n(e)$ 表示实体 e 具有边的数目，$n(e,p)$ 表示实体 e 具有谓词 p 的次数. 如果谓词对于实体 e 是频繁出现的，那这个谓词应该对实体的类型补全有更大的影响作用. 以 Tom_Hanks 为例，在 DBpedia 2014 数据集中，Tom_Hanks 具有“starring^{-1}”谓词 47 次，而谓词“director^{-1}”只出现了 5 次，很明显 $f(Tom\_Hanks, starring^{-1})>f(Tom\_Hanks, director^{-1})$，也就是 Tom_Hanks 具有更大的可能性属于与谓词“starring^{-1}”最相关的类型.

谓词对类型的区分能力是不同的，某些谓词对于实体类型几乎没有区分能力. 比如，知识库中几乎所有实体都有“name”这个谓词，“name”对类型就没有区分力. 为此，我们提出一种机制来降低这种出现次数过多的谓词在类型补全中的重要性，即谓词的逆实体频率，定义如式(11).

$$i(p)=\max\left\{0,\log\left(\frac{N-n(p)}{n(p)}\right)\right\} \tag{11}$$

这里 N 是知识库中所有实体的总数，$n(p)$ 是拥有谓词 p 的所有实体的总数.

基于上面所定义的谓词的频率、谓词的逆实体频率，迭代的初始值 $\boldsymbol{V}'$ 定义如下：

$$\boldsymbol{V}'[v]=\begin{cases}f(e,v)\times i(v), & v \text{是一个谓词}\\ \frac{1}{n(v)}, & v \text{是一个类型且实体} e \text{属于类型} v\\ 0, & \text{其他情况}\end{cases} \tag{12}$$

对向量 $\boldsymbol{V}'$ 做归一，最终迭代的初始分布 $\boldsymbol{V}^0$ 为

$$\boldsymbol{V}^0[v]=\frac{\boldsymbol{V}'[v]}{\sum_{v'}\boldsymbol{V}'[v']} \tag{13}$$

构建好谓词-类型推理图的转移概率矩阵和初始概率分布后，利用式(9)的随机游走算法得到实体的类型信息. 算法计算时间主要消耗在计算所有结点的向量上，时间复杂度为 $O(K(m+n)^2)$，其中 K 表示迭代次数，m 和 n 分别表示知识库中谓词和类型的数量.

4.3　改进模型 GBTC-PMI

本文提出的模型基本思想是基于实体具有的谓词补全实体的类型，比如实体具有谓词“starring^{-1}”，那实体很可能属于与“starring^{-1}”最相关的类型“Actor”. 然而，仍然存在类型语义漂移情况，分析原因主要包括 3 方面：

(1) 虽然谓词逆实体频率 $i(p)$ 可以打压类型区分度低的谓词对结果的影响，但是因为此类谓词与

知识库中其他谓词的共现非常高，会导致在游走时还是会走到这些区分度低的谓词上，从而影响补全效果. 如图 4 中，谓词“starring^{-1}”、“nationality”、“birthPlace”、“birthDate”与谓词“name”的出现频率都很高(因为知识库中几乎所有实体都具有谓词“name”)，虽然在计算初始概率分布时，考虑了类型区分度低的谓词对类型预测的影响，但是在图上随机游走时，由于知识库中的谓词与此类区分度低的谓词共现频率都很高，这些区分度低的谓词对结果还是会造成影响.

(2) 谓词本身对类型的区分就具有不确定性，比如谓词“country”，具有这个谓词的实体类型可能是“Book”、“Film”、“Software”等不同的类型. 如果根据这样的谓词推断实体类型，也会出现类型语义漂移的情况. 而对于这样的情况仅通过谓词区分度的定义是克服不了的，因为虽然该谓词与很多不同类型相关，但是具有该谓词的实体数目不一定很高.

(3) 知识库中存在错误的知识，会导致语义漂移. 比如 DBpedia 2014 数据集中有如下三元组〈*Winnie_Lau*, *spouse*, *Grasshopper_*(*band*)〉(中文含义为〈刘小慧，配偶，草蜢乐队〉，正确的知识为〈刘小慧，配偶，苏志威〉，其中苏志威是草蜢乐队中的一员). 由于这种错误知识的存在，在计算 $P(c|p)$ 和 $P(p_2|p_1)$ 的时候，就会存在有的概率值非常小，而如果把所有的不为 0 的概率都考虑进来，不仅会导致类型语义漂移，还会影响算法的时间开销.

针对上述分析的 3 个原因，我们在构造推理图时有两方面的修改.

4.3.1 引入 PMI

PMI[33] (Pointwise Mutual Information，点互信息)的概念，可以理解为一个随机变量中包含另一个随机变量的信息量，或者说是已知一个随机变量后另一个随机变量不确定性的减少程度. 其定义为

$$I(x;y)=\log\frac{P(x,y)}{P(x)P(y)} \tag{14}$$

其中 $P(x,y)$ 是随机变量(x,y)的联合分布，$P(x)$ 和 $P(y)$分别是两个随机变量的边缘分布. 根据点互信息的定义式(14)，我们可以观察到：

(1) 点互信息满足对称性，也就是 $I(x;y)=I(y;x)$.

(2) 点互信息的值可为正值也可为负值. 如果为负值，说明在已知某个随机变量后，不但使另一个随机变量的不确定性没有减少，反而增加，此时的点互信息为负值.

在谓词-类型推理图中，包括两种类型的边，谓词到谓词和谓词到类型. 我们分别计算谓词和谓词以及谓词和类型之间的 PMI. 计算公式如下：

$$I(c;p)=\log\frac{P(c,p)}{P(c)P(p)} \tag{15}$$

$$I(p_1;p_2)=\log\frac{P(p_1,p_2)}{P(p_1)P(p_2)} \tag{16}$$

其中，

$$P(c,p)=\frac{n(c,p)}{N},$$

$$P(p_1,p_2)=\frac{n(p_1,p_2)}{N},$$

$$P(c)=\frac{n(c)}{N},$$

$$P(p)=\frac{n(p)}{N}.$$

其中，N 是知识库中所有实体的总数，$n(c)$是属于类型 c 的实体数，$n(p)$具有谓词 p 的实体数，$n(c,p)$是属于类型 c 并且具有谓词 p 的实体数，$n(p_1,p_2)$是具有谓词 p_1和 p_2的实体数目.

计算出谓词和类型以及谓词和谓词之间的点互信息之后，将点互信息为 0 或者为负值的都丢弃，因为点互信息为 0 说明谓词与类型或者两个谓词之间是相互独立的，如果为负值则说明已知某个谓词后，另外的一个类型或者谓词的不确定性没有减少，这说明这种情况可能是由于数据噪声或者其他原因导致的. 比如上文中举的知识库中存在错误知识〈*Winnie_Lau*, *spouse*, *Grasshopper_*(*band*)〉，统计出 $P(Band|spouse^{-1})>0$，而根据点互信息可以得到 $I(Band;spouse^{-1})<0$，这样谓词“spouse^{-1}”和类型“Band”就可以不再有连边，在避免了类型语义漂移的同时也减少了图中边的数量.

我们也将文献[26]中给实体找到最典型类型的方法应用于构造推理图，公式为

$$Rep(c,p)=P(c|p)\cdot P(p|c) \tag{17}$$

$$Rep(p_1,p_2)=P(p_1|p_2)\cdot P(p_2|p_1) \tag{18}$$

表 2 对比给定谓词“starring^{-1}”，式(5)、式(15)、式(17)得出谓词“starring^{-1}”与类型的关联得分. 分别选取了上级层次类型“Agent”、基本层次类型“Person”、下级层次类型“Artist”及“Actor”.

表 2 3 种方法获得的与谓词“starring^{-1}”的相关类型比较

类型	式(5)	式(15)	式(17)
Agent	**0.642**	0.555	0.021
Person	0.635	0.701	**0.024**
Artist	0.096	1.519	0.008
Actor	0.037	**3.264**	0.018

从表 2 可以看出，式(5)的方法计算出来的与给定谓词“starring^{-1}”，最相关的类型是“Agent”，偏重于上级层次类型；式(15)得分最高的是“Actor”，偏重于下级层次类型，与“starring^{-1}”关系最紧密的就是“Actor”，符合预期；式(17)得分最高的是“Person”，属于基本层次类型.

表 3 对比给定谓词“starring^{-1}”，式(6)、式(16)、式(18)得出的谓词与谓词的关联得分. 分别选取了 name、birthPlace、director^{-1}. 由表 3 可以看出，式(6)得分最高的是谓词“name”，因为在整个知识库中几乎每个实体都有这个谓词“name”；式(16)得分最高的是“director^{-1}”，这是因为如果一个实体具有谓词“starring^{-1}”，则该实体同时具有谓词“director^{-1}”的不确定性程度降低值最大. 式(18)得分最高的是“birthPlace”.

表 3　3 种方法获得的与谓词“starring^{-1}”的相关谓词比较

谓词	式(6)	式(16)	式(18)
name	**0.652**	0.089	0.013
birthPlace	0.485	1.364	**0.036**
director^{-1}	0.046	**2.037**	0.007

表 4 给出的是根据式(6)、式(16)、式(18)，计算与谓词“starring^{-1}”最相关的 Top5 谓词的列表. 从结果中可以看出，式(6)列出的谓词偏重于描述上级层次类型或者基本层次类型的概念，式(16)列出的谓词偏重于描述下级层次类型的概念，也就是细粒度的概念，比如“director^{-1}”是“Director”类型所具有的谓词，式(18)偏重于描述基本层次类型的概念，比如“occupation”等都是属于“Person”类型的谓词. 所以 3 种不同的方法在应用于类型补全时，式(6)上级层次类型得分会很高，式(16)下级层次类型得分会很高，而式(18)则是基本层次类型得分会很高. 而我们期望的结果就是下级层次类型得分高，因为类型是一个层次结构，有了下级层次类型便可以根据类型的层次结构补全上级的类型，所以式(15)、式(16)是我们期望的一种形式.

表 4　3 种方法与谓词“starring^{-1}”相关的 Top5 谓词比较

公式	谓词
式(6)	name description birthYear birthDate birthPlace
式(16)	portrayer^{-1} narrator^{-1} voice^{-1} voiceType director^{-1}
式(18)	occupation birthName guest activeYearsStartYear birthPlace

4.3.2　谓词区分度 $i(p)$

4.2.2 节中为了计算谓词-类型推理图的初始概率分布，定义了谓词逆实体频率，用来打压对于类型区分度低的谓词对结果的影响. 然而在游走时，由于知识库中其他谓词与区分度低的谓词共现频率都很高，因此，这些区分度低的谓词对结果的负面影响还会被引入. 同时，考虑到谓词的区分度是一个全局的度量，与实体无关. 所以，我们在计算推理图边的权重时，不仅仅考虑两个节点的 PMI，还考虑谓词的区分度. 重新定义边的权重为

$$w_{i,j}=\begin{cases} I(p_2;p_1)\times i(p_1), & i=p_1 \text{ 且 } j=p_2 \\ I(c;p)\times i(p), & i=p \text{ 且 } j=c \\ 0, & \text{其他} \end{cases} \tag{19}$$

相应的概率图的初始概率分布只需要考虑谓词针对实体出现的频率就可以.

5　实　验

5.1　实验数据与评测指标

为了评估我们方法的有效性，我们在 DBpedia 2014 数据集上做了充分的实验. DBpedia 数据集是在 Wikipedia 中抽取出来的多领域结构化知识库，DBpedia 2014 中包含 458 万个实体，其中有 144.5 万个“Person”、73.5 万个“Place”以及 41.1 万个“Work”等，还有 5.83 亿条三元组. 在该数据集中，我们使用了 Mappingbased properties，Instance types 以及 Specific_mappingbased properties 三个子数据集. 数据集的一些基本描述信息如表 5.

表 5　DBpedia2014 数据集基本描述信息

特征	信息
实体数	4 580 000
类型数	685
谓词数	2679
类型层次结构中叶子类型的平均深度	3.5
每个实体平均具有的类型数	3.34
具有类型信息的实体数目	4 218 627
只具有上级层次类型的实体数目	1 276 147
具有基本层次类型的实体数目	2 942 446
具有下级层次类型的实体数目	1 240 191

5.1.1　实验数据的生成

我们采用同文献[14]相同的实验数据生成方法，在知识库中随机抽取 10 000 个实体，然后利用

知识库中这 10 000 个实体已有的类型作为标准答案来衡量方法的有效性.然而文献[14]这样抽取出来的实体很可能本身就有类型缺失,比如有的实体只有非常抽象的类型"Thing",缺少细粒度的类型信息,如果使用这些测试数据作为标准答案,会影响评测的有效性.

本文基于基本层次类型这个概念,通过人工标注的方法获得了每个类型所属的层次,我们的实验数据偏重于抽取具有细粒度类型的实体用于测试.抽取了两组测试数据,每组包含 10 000 个测试实体.第 1 组测试实体是在具有基本层次类型的实体集合中随机抽取,记为 $G1$.第 2 组测试实体是在具有下级层次类型的实体集合中随机抽取,这些实体具有更具体、更细粒度的类型,记为 $G2$.实验时,我们假设抽取的测试实体没有任何的类型信息.测试实体的一些基本描述性信息如表 6.

表 6 测试实体基本描述信息

特征	$G1$ 信息	$G2$ 信息
实体数	10 000	10 000
基本层次类型数	141	20
下级层次类型数	162	169
具有基本层次类型的实体数目	10 000	9999
具有下级层次类型的实体数目	4409	10 000
每个基本层次类型平均实体数	73.25	499.95
每个下级层次类型平均实体数	44.15	96.36
每个测试实体平均具有的谓词数	7.60	7.83

由表 6 可以看出:

(1)下级层次类型平均包含的实体数要比基本层次类型少,也就是说,类型越具体,该类型包含的实体就越少.

(2)从 $G1$ 的测试集中可以看出,具有基本层次类型的实体中,大概只有 44% 的具有下级层次类型,可以看出下级层次类型缺失严重.从 $G2$ 的测试集中可以观察出,具有下级层次类型的实体,99.9% 的具有基本层次类型.

(3)测试集 $G2$ 中每个实体平均具有的谓词是 7.83,多于 $G1$ 中的 7.60.

5.1.2 评价指标

抽取出两组测试集后,我们将这些测试实体在 DBpedia 2014 数据集中具有的类型作为标准答案(用的是原始数据,标准答案中也可能缺失类型),用如下指标对实验结果进行度量.

(1) MAP:平均正确率均值.AP(平均精度)是指对于一个查询,对不同召回率点上的正确率进行平均.平均正确率均值是所有查询 AP 的算术平均值.

(2) *Precision-Recall* 曲线:正确率-召回率曲线.对于每一次查询,我们在 11 个召回率值 0.0,0.1,…,1.0 上进行插值精度测试.对于每一个召回率值,计算在该召回率值下所有查询的插值正确率的算术平均值.

(3) *Mean Rank*:平均排名.我们希望实体类型补全后的结果中,细粒度的类型越靠前越好,也就是细粒度类型的平均排名越小越好.我们分别度量基本层次类型和下级层次类型的平均排序位置.

5.2 参数设置

随机游走步数的阈值选择,同文献[34],我们将其设置为 100.式(9)随机游走模型中有一个阻尼因子 α,该参数表示在随机游走时,返回初始结点的概率.设置参数 α 是为了在随机游走过程中能以一定的概率从起始点开始重新游走.我们在两组测试数据上对参数 α 做了性能测试,结果如表 7 所示.

表 7 不同 α 的 *MAP*

α	平均准确率($G1$)		平均准确率($G2$)	
	GBTC	GBTC-PMI	GBTC	GBTC-PMI
0.8	0.792	0.802	0.839	0.867
0.5	0.802	0.826	0.850	0.871
0.2	0.806	0.830	0.856	0.880
0.1	0.807	0.832	0.857	0.883
0.05	0.808	0.832	0.857	0.885
0.01	**0.808**	**0.832**	**0.857**	**0.886**

从表 7 可以看出,随着 α 变小,说明游走时以更大的概率停留在当前的结点上(实体在知识库中具有的谓词结点),以较小的概率向其他谓词或者类型结点游走.另外,从两组测试数据的平均正确率均值指标看,参数 α 越小,平均正确率均值越高.由于谓词-类型推理图模型走到类型结点会被吸收(类型结点没有出边),它们的得分不会分配给其他结点,只需要考虑谓词之间的相互增强作用结果即可提高.后续实验我们的参数取值为 $\alpha=0.01$.另外,由表 7 可以明显看出改进后的模型 GBTC-PMI 要明显好于 GBTC.分析原因主要是改进后的模型减少了游走到可以引起类型语义漂移的结点上,所以提高了平均准确率.

5.3 与目前典型方法性能的比较

我们在平均正确率均值、基本层次类型的平均排名和下级层次类型的平均排名 3 个指标上与其他方法进行了对比,如表 8 所示.

表 8　类型补全不同方法性能比较

方法	*G*1			*G*2		
	平均准确率	基本层次类型平均排名	下级层次类型平均排名	平均准确率	基本层次类型平均排名	下级层次类型平均排名
SDType	.762	4.420	5.517	.776	4.053	5.245
TransE	.004	31.870	131.290	.089	31.709	57.572
GBTC	.808	4.365	5.443	.857	3.404	5.428
GBTC-PMI	**.832**	**3.920**	**2.870**	**.886**	**2.570**	**1.910**

整体而言，4 种方法在 *G*2 测试集上的表现要好于 *G*1，分析其原因主要是 *G*2 测试集上测试实体的平均谓词数目要多于 *G*1，这样在实体类型补全时有更多的信息可以参考. 另外，方法 GBTC、SDType 及 TransE 对于实体做类型补全后，基本层次类型的平均排名要好于下级层次类型，原因也是因为基本层次类型有更多的谓词作为这种类型的特征，比如对于基本层次类型“Person”，有“birthPlace”、“nationality”、“birthDate”等等，但是对于下级层次类型比如“Chef”，在知识库 DBpedia 2014 中就没有标志类型“Chef”的谓词. 另外，也说明，根据谓词推理实体属于某个基本层次类型是相对容易的，但是仅仅根据谓词补全实体的细粒度类型是不够的. 改进后的模型 GBTC-PMI 在下级层次类型的平均排名要明显好于基本层次类型的平均排名，主要是因为该模型在由谓词游走到下级层次类型结点，相比于游走到上级层次类型或者基本层次类型的得分都要高.

另外，从表 8 也可以看出，TransE 方法在各个指标上补全效果都非常差，主要是因为 TransE 方法模型简单，不适用于解决复杂的关系类型（比如 1-*N*，*N*-1 和 *N*-*N*），而实体类型正是一种 *N*-*N* 的关系，因为对于一个实体可能会属于多个类型，而一个类型也会包含很多实体. 在 11 个召回率点上，TransE 在测试集 *G*1 上最高的正确率为 0.104，在 *G*2 上最高的正确率为 0.105. 为了更好地呈现我们的模型与 SDType 在正确率-召回率上的差别，TransE 的正确率-召回率曲线将不再呈现.

图 5、图 6 分别展示了 SDType 和本文提出的 GBTC 及改进的 GBTC-PMI 方法的正确率-召回率曲线. 由图可以看出 GBTC 的性能要好于 SD-Type 方法，随着召回率的增大，SDType 的正确率有明显的下降. 分析其原因就是由于知识库中知识缺失严重，导致类型补全正确率下降. 比如测试实体只有“starring^{-1}”这个谓词，而没有“birthPlace”、“birthDate”这种对于类型“Person”区分能力强的谓词，那根据“starring^{-1}”可以推出其类型为“Actor”的得分很高，但是由于 SDType 没有考虑谓词之间的相互增强作用，该实体在类型“Person”上的得分会很低. 这样就导致方法 SDType 的正确率随着召回率的提高而明显下降，同时也反应了知识库中知识缺失非常严重. 另外改进后的模型 GBTC-PMI 要优于原模型 GBTC，分析原因是由于 GBTC-PMI 减少了推理图中无效的连边，对类型语义漂移有一定抑制作用，提升了精度.

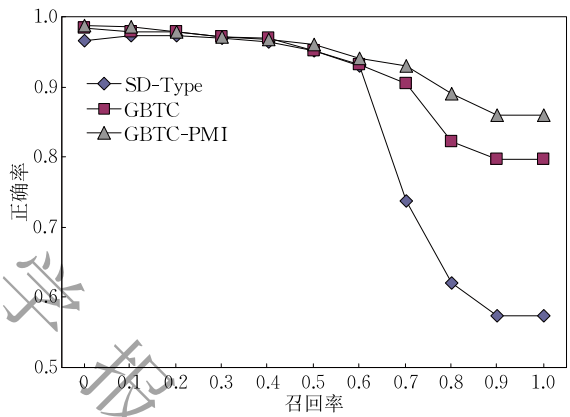

图 5　*G*1 测试集上的正确率-召回率曲线

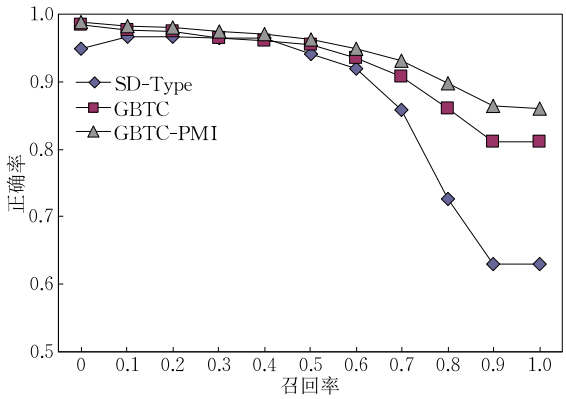

图 6　*G*2 测试集上的正确率-召回率曲线

5.4　GBTC-PMI 方法在不同类型上的比较

为了更好地分析实验效果，我们在两个测试集上将平均排名最好和最差的 5 个类型在表 9 中做了展示，从而对不同类型实体的补全效果做分析.

表 9 GBTC-PMI 方法在不同类型上的比较

	G1		G2	
	类型	平均排名	类型	平均排名
补全效果最好的前 5 个类型	CelestialBody	1.39	CelestialBody	1.63
	Chemical Substance	1.69	Satellite	2.00
	Language	2.00	ChemicalSubstance	2.12
	Colour	2.00	Tournament	2.75
	Currency	2.00	Mineral	2.94
补全效果最差的 5 个类型	Cave	120.00	Chef	88.38
	SiteOfSpecialScientificInterest	100.00	RollerCoaster	74.67
	BaseballLeague	83.33	Castle	66.14
	CricketGround	82.00	Lighthouse	58.33
	AmericanFootballLeague	77.00	Prison	57.667

从表 9 可以看出，在两个数据集上对于类型“CelestialBody”补全的效果最好，原因是“CelestialBody”是一个基本层次类型，有许多谓词都是描述这个类型所具有的属性，比如“periapsis”、“maxApparentMagnitude”、“absoluteMagnitude”、“orbitalEccentricity”、“apoapsis”等等，补全效果差的几个类型主要是因为没有明显用来区别该类型的谓词，比如类型“Chef”、“AmericanFootballLeague”. 因此，补全实体的类型时，仅靠谓词做补全正确率是有限的. 比如，类型“Physicist”和“Chemist”，属于这两个类型的实体都会具有“birthPlace”、“birthDate”、“alumni”、“field”、“award”等等这样的谓词，单从谓词是不能区分具体属于哪个类型的，还需要依靠谓词所具有的值来区分，比如“Physicist”和“Chemist”的“field”是不同的，即谓词“field”的值不同，“award”也是有所不同的. 这也是我们未来要研究的工作，就是在更细粒度的类型上做补全，尤其是仅用谓词无法区分的类型上的补全.

6 总结及展望

本文提出了一种基于谓词-类型推理图上的随机游走方法对实体类型进行补全. 推理图中包括谓词和类型两类结点，边也分为两类，一类是由谓词到谓词，另外一类是由谓词到类型. 本文提出的方法 GBTC-PMI 考虑了知识库中的知识缺失以及存在错误知识对类型补全效果的影响. 通过实验结果表明，相比于目前已有的典型方法，我们的方法可以更加有效地补全实体类型.

下一步的工作主要包括：

(1) 进一步提升对于实体细粒度类型补全的准确率. 从实验结果可以看出，仅仅使用实体的谓词补全实体类型是不够的. 比如类型“Scientist”下面还包括子类型“Physicist”、“Chemist”、“Mathematician”等等，这些子类型从所具有的谓词上并没有区别，都具有“birthPlace”、“birthDate”、“field”、“alumni”等谓词，还需要借助知识库中的其他信息.

(2) 目前对于知识库中类型的分类(上级层次类型、基本层次类型、下级层次类型)还是依据人工标注的方法，类型的自动分类也是后续的一个研究工作.

(3) 知识库的类型体系及知识库中的谓词也有缺失，考虑借助非结构化文本集，对知识库模式中的类型以及谓词做补全.

参 考 文 献

[1] Berners-Lee T, Hendler J, Lassila O. The semantic Web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific America, 2001, 284(5): 34-43

[2] Bizer C, Heath T, Idehen K, et al. Linked data on the web (LDOW2008)//Proceedings of the 17th International Conference on World Wide Web. Beijing, China, 2008: 1265-1266

[3] Jupp S, Malone J, Bolleman J, et al. The EBI RDF platform: Linked open data for the life sciences. Bioinformatics, 2014, 30(9): 1338-1339

[4] Auer S, Bizer C, Kobilarov G, et al. DBpedia: A nucleus for a web of open data//Proceedings of the 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference. Busan, Korea, 2007: 722-735

[5] Bollacker K, Evans C, Paritosh P, et al. Freebase: A collaboratively created graph database for structuring human knowledge//Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. Vancouver, Canada, 2008: 1247-1250

[6] Suchanek F M, Kasneci G, Weikum G. Yago: A core of semantic knowledge//Proceedings of the 16th International Conference on World Wide Web. Alberta, Canada, 2007: 697-706

[7] Zou Lei, Huang Ruizhe, Wang Haixun, et al. Natural language question answering over RDF: A graph data driven approach//Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. Snowbird, USA, 2014: 313-324

[8] Milne D N, Witten I H, Nichols D M. A knowledge-based search engine powered by Wikipedia//Proceedings of the 16th ACM Conference on Information and Knowledge Management. New York, USA, 2007: 445-454

[9] Wang Zhongyuan, Zhao Kejun, et al. Query understanding through knowledge-based conceptualization//Proceedings of the 24th International Joint Conference on Artificial Intelligence. Buenos Aires, Argentina, 2015: 3264-3270

[10] Chen Y, Gao L, Shi S, et al. Improving context and category matching for entity search//Proceedings of the 28th Conference on Artificial Intelligence. Quebec, Canada, 2014: 16-22

[11] Zhang Jun-San, Qu You-Li. Survey on related finding in information retrieval. Computer Engineering and Design, 2011, 32(12): 4035-4038(in Chinese)
(张俊三，瞿有利. 信息检索中相关实体发现综述. 计算机工程与设计，2011，32(12)：4035-4038)

[12] Bron M, Balog K, de Rijke M. Ranking related entities: Components and analyses//Proceedings of the 19th Conference on Information and Knowledge Management. Toronto, Canada, 2010: 1079-1088

[13] Demartini G, Iofciu T, de Vries A P. Overview of the INEX 2009 entity ranking track//Proceedings of the 8th International Workshop of the Initiative for the Evaluation of XML Retrieval. Brisbane, Australia, 2009: 254-264

[14] Paulheim H, Bizer C. Type inference on noisy RDF data//Proceedings of the 12th International Semantic Web Conference. Sydney, Australia, 2013: 510-525

[15] Drumond L, Rendle S, Schmidt-Thieme L. Predicting RDF triples in incomplete knowledge bases with tensor factorization //Proceedings of the ACM Symposium on Applied Computing. Trento, Italy, 2012: 326-331

[16] Franz T, Schultz A, Sizov S, Staab S. TripleRank: Ranking semantic web data by tensor decomposition//Proceedings of the 8th International Semantic Web Conference. Virginia, USA, 2009: 213-228

[17] Krompass D, Nickel M, Tresp V. Large-scale factorization of type-constrained multi-relational data//Proceedings of the International Conference on Data Science and Advanced Analytics. Shanghai, China, 2014: 18-24

[18] Nickel M, Tresp V, Kriegel H. A three-way model for collective learning on multi-relational data//Proceedings of the 28th International Conference on Machine Learning. Washington, USA, 2011: 809-816

[19] Nickel M, Tresp V, Kriegel H. Factorizing YAGO: Scalable machine learning for linked data//Proceedings of the 21st World Wide Web Conference 2012. Lyon, France, 2012: 271-280

[20] Bordes A, Usunier N, et al. Translating embeddings for modeling multi-relational data//Proceedings of the 27th Annual Conference on Neural Information Processing Systems. Nevada, USA, 2013: 2787-2795

[21] Lin Y, Liu Z, Luan H, et al. Modeling relation paths for representation learning of knowledge bases//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal, 2015: 705-714

[22] Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion//Proceedings of the 29th Conference on Artificial Intelligence. Austin, USA, 2015: 2181-2187

[23] Socher R, Chen D, Manning C D, Ng A Y. Reasoning with neural tensor networks for knowledge base completion// Proceedings of the 27th Annual Conference on Neural Information Processing Systems. Lake Tahoe, USA, 2013: 926-934

[24] Wang Z, Zhang J, Feng J, Chen Z. Knowledge graph embedding by translating on hyperplanes//Proceedings of the 28th Conference on Artificial Intelligence. Quebec, Canada, 2014: 1112-1119

[25] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality// Proceedings of the 27th Annual Conference on Neural Information Processing Systems. Lake Tahoe, USA, 2013: 3111-3119

[26] Wang Zhongyuan, Wang Haixun, Wen Ji-Rong, Xiao Yanghua. An inference approach to basic level of categorization// Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. Melbourne, Australia, 2015: 653-662

[27] Rosch E, et al. Basic objects in natural categories. Cognitive Psychology, 1976, 8(3): 382-439

[28] Mervis C B, Rosch E. Categorization of natural objects. Annual Review of Psychology, 1981, 32(1): 89-115

[29] Hearst M A. Automatic acquisition of hyponyms from large text corpora//Proceedings of the 14th International Conference on Computational Linguistics. Nantes, France, 1992: 539-545

[30] Tong H, Faloutsos C, Pan J. Fast random walk with restart and its applications//Proceedings of the 6th IEEE International Conference on Data Mining. Hong Kong, China, 2006: 613-622

[31] Fujiwara Y, Nakatsuji M, Onizuka M, Kitsuregawa M. Fast and exact top-k search for random walk with restart. Proceedings of the Very Large Data Bases Endowment, 2012, 5(5): 442-453

[32] Strang G. Introduction to Linear Algebra. Wellesley, USA: Wellesley-Cambridge Press, 1993

[33] Church K, Gale W, Hanks P, Kindle D. Using statistics in lexical analysis. Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon, 1991, 1(1): 115-164

[34] Hu Jian, Wang Gang, Lochovsky F, et al. Understanding user's query intent with Wikipedia//Proceedings of the 18th International Conference on World Wide Web. Madrid, Spain, 2009: 471-480

ZHANG Xiang-Ling, born in 1983, Ph. D. candidate. Her research interests include entity search, knowledge graph completion.

CHEN Yue-Guo, born in 1978, Ph. D., associate professor. His research interests include big data real-time analysis system, knowledge graph and semantic search.

MAO Wen-Xiang, born in 1994, M. S. candidate. His research interests include knowledge graph, entity search.

RONG Chui-Tian, born in 1981, Ph. D., associate professor. His research interests include database and information retrieval, cloud computing and big data analysis.

DU Xiao-Yong, born in 1963, Ph. D., professor, Ph. D. supervisor. His research interests include database system, intelligent information retrieval, etc.

Background

A number of knowledge graphs or structured knowledge bases, such as DBpedia, Freebase and YAGO2, have been constructed and released to public. Some proprietary ones such as Google's Knowledge Graph and Microsoft Bing's Satori have been applied in search engines to support important Web search tasks such as entity search and question answering. A fundamental and essential semantic information of an entity is its types. The type information of entities plays an important role in many applications. Unfortunately, the incompleteness of entity types is very serious in knowledge graphs which somehow affects the wide and effective applications of knowledge graphs. We observe that 8% entities are without any type information among 4 580 000 entities of DBpedia 2014 and only 64% have fine-grained types. Thus the study of type completion is important and necessary.

Traditionally, people solve this problem by assigning an entity to the type that the entity most likely belongs to. The likelihood can be estimated by the similarity between the entity and the entities already belong to the type. However, this method ignores the dependency between types and predicates. A number of representation learning methods have been recently proposed. However, these approaches rely on the training data that may be difficult to obtain (for negatives). Moreover, all these methods lack interpretation since the prediction is based on the vector operations and reasoning mechanisms are implicit.

In this paper we propose a novel approach to complete entity type using a random-walk-based iterative algorithm on a predicate-type probabilistic graph. The experiments results show that the proposed approaches have better performance.

This research is supported by the National Natural Science Foundation of China under Grant No. 61472426 and No. 61402329. This project focuses on the basic primitives, query language, interactive interface, query optimization and processing and other key technologies of the exploratory search. Our team has been studying and working on this research field for years, and has published a series of papers in various international conferences and journals. This work is also supported by a gift of Tencent.