使用桥接实体对知识库进行推理

Bhushan Kotnis
Indian Institute of Science
bkotnis@dese.iisc.ernet.in

Pradeep Bansal
Indian Institute of Science
pradeepb@ee.iisc.ernet.in

Partha Talukdar
Indian Institute of Science
ppt@serc.iisc.in

摘要

大规模知识库（例如 NELL，Yago，Freebase，等等）通常较为稀疏，即在实体间的大量合法关系丢失。最近的研究解决了这个问题。在知识图谱的结点不变的情况下，通过给知识图谱添加额外的从大型文本语料库中挖掘出的边，然后使用路径排序算法（Path Ranking Algorithm，PRA）来对这个扩展的图谱进行知识库推导。在这篇论文中，我们通过不仅仅向知识图谱中添加边，还添加桥接实体，两者都是从一个 500 000 000 大小的网络文本语料库中挖掘出来的。通过对真实世界的实际数据集合进行实验，我们证明了桥接实体在提高性能以及降低知识库推导中 PRA 算法的运行时间中的价值。

1 介绍

大规模知识库像 Freebase（Bollacker 等，2008），Yago（Suchanek 等，2007），NELL（Mitchell 等，2015）在各种应用中例如自然语言问答、语义搜索引擎等等都有作用。这些知识库由数百万的世界实体以及它们之间的关系所组成，以有向图的形式存储，结点表示实体，连接表示实体间的关系。尽管这样的知识库包含成千上万的实体，它仍然是稀疏的，即它在实体之间缺失了大量的关系（West 等，2014）。

在知识图谱中进行推导，找寻两个实体之间的关系，是一种使知识图谱密集化的一种方式。例如，从(Germany, playTournament, FIFA)和(FIFA, tournamentofSport, Soccer)，我们可以推出(Germany, playSport, Soccer)。路径排序算法（PRA）（Lao 和 Cohen，2010）通过在知识图谱上学习推导规则来进行推导。

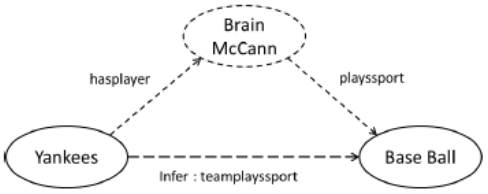

图 1：例子显示了桥接实体 Brian McCain，和两条边可以帮助 PRA 算法（Lao 和 Cohen，2010）推断出缺失的关系teamplayssport（Yankees, BaseBall）。原始知识图谱值包含两个结点Yankess 和Baseball，且没有边。

如果只是图谱是稀疏的，即在源实体与目标实体之间只有很少或者没有路径，那么 PRA 无法预测一个关系的存在性。为了解决这个缺陷，（Lao 等，2012）给知识图谱添加从一个外部语料库中获取的路径。添加的路径由从解析过从属外部语料库中得到的非词汇化的从属标签组成。为了增强添加的路径的表现性，（Gardner 等，2013）向知识图谱中添加谓语（表面关系），而不是非词汇化标签。谓词来自于超过 6 亿主语-谓语-宾语（Subject-Verb-Object，SVO）三元组。这些谓语以边的形式连接那些原本不连接的实体，于是增加了知识图谱的连接性，由此潜在地提高了 PRA 算法的性能。

然而，简单地添加这些边提高了稀疏性，降低了 PRA 中逻辑回归分类的判别能力。这个可以通过添加对表面关系聚类获取到的潜在关系来解决，而不是直接添加表面关系。这减少了稀疏型，并且已经被证明可以提高 PRA 推导能力（Gardner 等，2013），（Gardner 等，2014）。

在这篇文章中，我们提出一种方案，通过挖掘来自外部语料库的连接两个 SVO 三元组的名词短语来获取路径，以此扩展知识库。我们把这种名词短语叫做桥接实体，它桥接了两个知识库中的关系，形成了一条路径。这跟方案（Gardner 等，2013）与（Gardner 等，2014）不同，它通过从外部语料库中挖掘表面关系并向知识库结点中添加边。我们通过使用随需应变的方式在语料图中实现深度 DFS（深度优先搜索），在语料库中搜索这样的桥接实体。

我们将这个过程称作按需扩展（On-Demand Augmentation，ODA），因为搜索可以按照需求的方式在测试时间完成。相比之下，之前添加边或者嵌入知识库的方式（Gardner 等，2013），以及向量空间随机游走 PRA（Gardner 等，2014）是批处理程序。正如我们将要在第四部分看到的，由于有限的搜索空间，按需扩展比算法（Gardner 等，2013；Gardner 等，2014）更快。另外，由于边不是盲目添加的，按需扩展不增加稀疏性（致使性能下降的原因）。我们的实验表明，ODA 比（Gardner 等，2013）提供了更好的性能和与（Gardner 等，2014）几乎相同的预测能力，但是在这两种情况下，由于其在线和按需性质，增加了更快的运行时间与更大的灵活性的优势。代码以及结果可以从 https://github.com/malllabiisc/pra-oda 获取。

2 相关工作

使用表层关系和名词短语来提取出有意义的关系事实并不是一个新的想法（Hearst，1992）（Brin，1999）（Etzioni 等，2004）。然而，他们没有使用知识库来提高信息抽取。

第一次在（Lao 和 Cohen，2010）提出的路径排序算法（PRA）被使用在一个知识库推导（Lao 等，2011）。它被（Lao 等，2013）扩展，提高通过向知识库中添加从解析过的从属语料库中获取的语法信息。扩增知识库，使用从一个外部语料库中挖掘的表面关系与在表面关系中使用 PCA 获取的潜在的边标签，提高 PRA 推导，在（Gardner 等，2013）中探讨了。（Gardner 等，2014）采用了向量空间“软”映射，取代了表面关系到潜在关系的的硬映射。这允许了随机游走更频繁地遍历一条当前边类型的边。

尽管，像其他人一样，我们使用一个外部语料库来扩增知识库，在我们的方法中，关键的区别是，除了添加表面关系，我们也添加桥接实体，使我们能在知识库中添加新的路径。此外，改程序是有针对性的，因此，只有那些在推断关系中起到作用的路径会被添加。这样，用这种方式添加的路径的数量远远低于使用程序（Gardner 等，2013）添加的表面关系的数量。正如我们在第四节所见，这将导致更有效的算法和更快的运行时间。

3 方法

3.1 背景：路径排序算法（PRA）

本文首先简要概述了路径排序算法（PRA）（Lao 和 Cohen，2010）。PRA 使用路径特征的逻辑回归分类，预测给定的关系是否存在于一对实体之间。对于一对给定的实体 s 和 t，路径类型连接了 s 和 t，形成了向量特征。一个路径类型是一个有序的关系集合，具有相同有序关系但不同的中间或者末端实体的路径属于同一路径类型。例如，$s_1 \xrightarrow{v_0} x_1 \xrightarrow{v_1} t_1$和$s_2 \xrightarrow{v_0} x_2 \xrightarrow{v_1} t_2$属于路径类型$\xrightarrow{v_0} \xrightarrow{v_1}$。这个特征的值为 $P(s \to t;\pi)$，$P(s \to t;\pi)$是通过遍历路径π从 s 到 t 的可能性。PRA 通过在知识库中运行随机游走（RW）来近似得到这个可能性。令 $F=\{\pi_1, \pi_2, \dots, \pi_k\}$为所有路径类型的集合。为了预测实体 s 和 t 之间的关系 r 的存在性，逻辑回归分类输出一个分数值用来度量 s 和 t 之间存在关系 r 的置信度。它通过在训练阶段第一次分配权重给这项特征。分数值由一下公式给出：

$$S(s, t, r) = \sum_{\pi \in F} P(s \to t;\pi) \times \theta_\pi^r \qquad (1)$$

其中θ_π^r是权重，是在训练过程中，由逻辑回归分类学习关系 r 和路径类型π得到的。在测试阶段，由于目标不可用，PRA 收集候选目标进行随机游走，然后计算出向量特征，得到分数值。

3.2 PRA-SVO 和 PRA-VS

查询	候选答案	包括桥接实体（粗体表示）的路径
sportsteamPositionForSport(right handed pitcher, ?)	baseball	Right handed pitcher $\xrightarrow{\text{playsfor}}$ **Chicago Cubs** $\xrightarrow{\text{play}}$ baseball
riverFlowsThroughCity(Moselle, ?)	Koblenz	Mosell $\xrightarrow{\text{flows into}}$ **Rhine** $\xrightarrow{\text{meet at}}$ Koblenz
teamPlaysInLeague(Cleveland Indians, ?)	MLB	Cleveland Indians $\xrightarrow{\text{play}}$ **Detroit Tigers** $\xrightarrow{\text{blew}}$ MLB

表 1：使用 PRA-ODA 将包括桥接实体（粗体表示）的路径添加到知识库中。

PRA-SVO 和 PRA-VS 分别是（Gardner 等，2013）和（Gardner 等，2014）提出的系统，在知识库中添加了从一个大型主语-谓语-宾语（SVO）三元组语料库中挖掘的边。在这两个系统中，只有新的边被添加到固定的结点集合，这个添加过程是一个离线的批处理程序。相反，PRA-ODA，论文中提到的方法，还可以通过桥接实体扩展结点集，并按需进行扩展。

3.3 PRA 按需扩展（PRA-ODA）

训练：令 s 和 t 为任意两个知识库实体，$s^{(n)}$和$t^{(n)}$为他们对应的名词短语表示或别名。我们通过有限深度优先搜索（DFS）来搜索桥接实体$x_1, x_2, \dots, x_n$，从s^n开始，这样我们获取了一条路径s $\xrightarrow{ALIAS} s^{(n)} \xrightarrow{v_0} x_1 \xrightarrow{v_1} \dots \xrightarrow{v_{n-1}} x_n \xrightarrow{v_n} t^{(n)} \xrightarrow{ALIAS} t$，其中$\mathrm{v}_i$是在语料库图谱中出现的谓语。这些是在 n≤ $\mathrm{d}_{max} - 1$的条件下，d_{max}是 DFS 的最大深度。我们在知识库实体中添加“别名”边和它的名词表示。桥接实体的实用性如图 1 所示。

我们从一个来自 ClueWeb09（Callan 等，2009）的使用 MALT（Niver 等，2007）解析的超过 6 亿 SVO 三元组的语料库中挖掘桥接实体。我们使用 Mongo DB 来存储这些三元组，为一个邻接表。在训练期间，推断出的任何关系，源和其对应的目标实体是已知的。有限深度的 DFS 运行在所有深度小于d_{max}的 SVO 图中，使用主语实体的别名作为开始点。这样的别名在 NELL 和 Freebase 知识库中是可用的。如果路径的终端实体匹配任何目标实体的别名，DFS 就发现一条路径。我们选择使用别名来进行字符串匹配，是因为它容易通过简单地添加更多的别名来改变这种柔和度。这对于所有训练中的源目标对都是完成好的。表 1 中显示了添加路径的几个例子。

由于 SVO 图是通过解析从网络抓取到的 ClueWeb 语料库来获取的，因此它是嘈杂的。为了降低它的嘈杂度，我们添加 SVO 中发现的最频繁的 K 个路径类型，其中 K 是一个可调的参数。通过 SVO 路径类型，我们从 SVO 语料库中挖掘出有序谓语集合。有一种可能性是，从语料库中获取的桥接实体可能在知识库中存在。如果这个桥接实体匹配了任何别名，那么它被视作一个已存在的知识库实体的别名。如果不是，这个桥接实体被添加到知识库中作为新的实体。为了避免过度拟合，我们在训练集合中添加负面数据。另外，只有高质量的有表现力的桥接实体能够得到有意义且有判别力的路径。尽管桥接实体的质量依赖于语料库，低质量的桥接实体可以通过添加负面训练数据来过滤。低质量的桥接实体从正面和负面训练集合来连接源目标对，因此它被稀疏逻辑回归分类所去除。负面数据集是使用随机游走过程的封闭世界假设所产生的。

在知识库扩增之后，我们运行 PRA 算法的训练阶段来获取（路径）权重特征，它是由逻辑回归分类计算出的。

查询时间：对应于一个源实体和一个正在被预测的关系的目标实体集在查询（测试）时间内并不可用。我们使用包括在关系范围内的被预测为候选目标实体的所有实体。例如，关系是

KB 关系	PRA	PRA-SVO	PRA-VS	PRA-ODA
actorstarredinmovie	0.0	**1.0**	**1.0**	**1.0**
atheleteplaysforteam	1.0	1.0	1.0	1.0
citylocatedincountry	0.166	0.25	**1.0**	**1.0**
journalistwritesforpublication	1.0	1.0	1.0	1.0
riverflowsthroughcity	0.333	0.25	**1.0**	**1.0**
sportsteampositionforsport	1.0	1.0	1.0	1.0
stadiumlocatedincity	1.0	1.0	1.0	1.0
statehaslake	0.0	0.0	0.0	0.0
teamplaysinleague	1.0	1.0	1.0	1.0
writerwrotebook	1.0	1.0	1.0	1.0
平均（MRR）	0.649	0.75	**0.9**	**0.9**

表 2：NELL 中 10 个关系的 MRR 比较（值越大越好）。PRA-SVO，PRA-VS 是（Gardner 等，2013；Gardner 等，2014）提出的系统。PRA-ODA 是本文中提出的方法。与 PRA-SVO 上相比，PRA-ODA 的改善 $p < 0.007$ 有着统计学意义。

riverFlowsThroughCity，候选目标集将包括知识库中的城市实体。在训练期间，DFS 从源实体开始执行，但是这一次只限制路径为训练期间学习到的正权值。任何在这次搜索中找到的路径（连同桥接实体）被添加到知识库中，且 PRA 算法正运行于这个扩增的图谱中。

4 实验

我们使用（Gardner 等，2014）的作者提供的 PRA 算法的实现。在我们的实验中，我们使用相同的 10 个 NELL 关系数据，就像在（Gardner 等，2014）中。这个扩增在训练时期增加了 1086 条路径，在测试时期增加了 1430 条路径。

我们将 NELL 数据分为 60%训练数据，15%开发数据和 25%测试数据。d_{max}和最频繁路径 K 的值是通过在一个开发集上调整 4 个关系来获得的（athleteplaysforsport，actorstarredinmivie，citylocateincountry 和 journalistwritesforpulication）。超参数

时间（秒）	PRA	PRA-SVO	PRA-VS	PRA-ODA
训练	635.6	574.5	564.2	913.3
测试	354.3	322.0	301.2	436.7
批扩增	n/a	797	797	n/a
嵌入计算	n/a	n/a	812	n/a
总时间	989.9	1693.5	2474.4	1350

表 3：整个实验的运行时间比较（值越低越好）。PRA-SVO，PRA-VS 是（Gardner 等，2013；Gardner 等，2014）提出的系统。PRA-ODA 是本文中提出的方法。在两个系统 PRA-ODA 和 PRA-VS 中，PRA-ODA 快了 1.8 倍。

d_{max}=2，K=10 具有最高的 MRR，并给剩下的关系使用。在逻辑回归分类中的L_1和 L_2 正则化参数，我们使用与（Gardner 等，2013；Gardner 等，2014）相同的值，即 $L_1 = 0.005$，$L_2 = 1.0$。这是因为参数是稳健的，并能够很好地工作，即使是在知识库扩增的情况下。

我们比较（PRA-ODA）和 PRA 算法在 NELL 知识库上运行 的结果，NELL 知识库使用表面关系扩增（PRA-SVO）（Garnder 等，2013），和 PRA（PRA-VS）（Garnder 等，2014）向量空间随机游走时间。运行时间，即执行一整个 PRA-SVO 和 PRA-VS 实验，包括使用 SVO 边来扩增 NELL 知识库的时间。PRA-VS 运行时间也包括了生成嵌入的时间，用来执行向量空间随机游走。正如在表 2 和表 3 中所见，我们的方案，PRA-ODA，提供的性能相当于 PRA-VS，具有更快的运行时间（加快 1.8）。除了全部 SVO 扩增时间，PRA-VS 花费额外时间来从添加的谓语中生成嵌入（13 分钟）。我们注意到，在 PRA-SVO 和 PRA-VS 的批扩增，且以 PRA-VS 嵌入计算都是在特定评价集中的关系，这样就变成不能忽视的一个一次性线下成本。换句话说，这些成本可能会增加更多的关系（和它们的实例），包括训练和测试。在这样的设置下，PRA-ODA 运行时间会更加明显地增长。

该算法的另外一个优点就是，它可以运行在任何以 PRA 为基础的算法上，例如

PRA-SVO 和 PRA-VS。

5 结论

在这篇论文中，我们调查往知识库中添加路径的有用性，通过从外部语料库中挖掘桥接实体提高其连接性。虽然以前的知识库扩增方法只专注于使用挖掘出的表面谓语同时保持结点不变，我们扩展了这些方法，通过在线的方式添加桥接实体。我们使用一个 5 亿大小的大型网络文本语料库来挖掘这些添加的边和桥接实体。尽管在实际数据集中进行实验，我们证实了提出的方法不仅仅是相当或优于其他最先进的方法，而且更重要的是提供了比其他方案更快的整体运行速度。

致谢

这项工作由 Google 提供一定程度上的支持。

参考文献

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD '08, pages 1247–1250, New York, NY, USA. ACM.

Sergey Brin. 1999. Extracting patterns and relations from the world wide web. In Paolo Atzeni, Alberto Mendelzon, and Giansalvatore Mecca, editors, The World Wide Web and Databases, volume 1590 of Lecture Notes in Computer Science, pages 172–183. Springer Berlin Heidelberg.

J. Callan, M. Hoy, C. Yoo, and L. Zhao. 2009. Clueweb09 data set. boston.lti.cs.cmu.edu.

Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2004. Web-scale information extraction in knowitall: (preliminary results). In Proceedings of the 13th International Conference on World Wide Web, WWW '04, pages 100–110, New York, NY, USA. ACM.

Matt Gardner, Partha Pratim Talukdar, Bryan Kisiel, and Tom Mitchell. 2013. Improving learning and inference in a large knowledge-base using latent syntactic cues. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 833–838.

Matt Gardner, Partha Pratim Talukdar, Jayant Krishnamurthy, and Tom Mitchell. 2014. Incorporating vector space similarity in random walk inference over knowledge bases. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 397–406.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th Conference on Computational Linguistics - Volume 2, COLING '92, pages 539–545, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ni Lao and William W. Cohen. 2010. Relational retrieval using a combination ofpath constrained random walks. Machine Learning, 81(1):53–67.

Ni Lao, Tom Mitchell, and William W. Cohen. 2011. Random walk inference and learning in

a large scale knowledge base. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11, pages 529–539, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ni Lao, Amarnag Subramanya, Fernando Pereira, and William W. Cohen. 2012. Reading the web with learned syntactic-semantic inference rules. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLPCoNLL '12, pages 1017–1026, Stroudsburg, PA, USA. Association for Computational Linguistics.

T. Mitchell,W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2015. Never-ending learning. In Proceedings of AAAI.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, G¨ulsen Eryigit, Sandra K¨ubler, Svetoslav
Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. Natural Language Engineering, 13(02):95–135.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In Proceedings of the 16th International Conference on World Wide Web, WWW'07, pages 697– 706, New York, NY, USA. ACM.

Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge base completion via search-based question answering. In Proceedings of the 23rd International Conference on World Wide Web, WWW '14, pages 515–526, New York, NY, USA. ACM.