

多语义视图驱动的 OWL 知识图谱表示学习方法^{*}

杨建喜¹, 谢江村¹, 李 韧¹, 杨小霞², 肖 桥², 蒋仕新¹, 贺丽荣¹

¹(重庆交通大学 信息科学与工程学院, 重庆 400074)

²(重庆交通大学 交通运输学院, 重庆 400074)

通信作者: 李韧, E-mail: renli@cqjtu.edu.cn



摘 要: 针对当前 OWL 知识表示学习方法存在的概念层和实例层复杂语义信息联合表征能力不足等问题, 提出一种概念-属性-实例多语义视图驱动的 OWL 图谱知识表示学习方法 (MSV-KRL). 该方法采用“多语义视图划分、语义感知自监督进阶训练、多任务联合表示学习”的 3 阶段架构. 首先, MSV-KRL 在 OWL2Vec^{*} 的基础上, 优化 OWL 到 RDF 图结构的映射策略, 提出 5 类细粒度语义视图划分策略. 其次, 通过语义视图内随机游走和标注属性替换策略, 生成序列化进阶训练数据, 并开展预训练模型的自监督进阶训练, 以提升其面向多语义视图上下文的适配能力. 最后, 在多任务学习框架下, 通过多语义视图预测任务联合优化损失, 实现对 OWL 知识图谱中概念、属性和实例复杂语义有效表示学习. 实验结果表明, MSV-KRL 在多个基准数据集上的表现优于现有先进的知识表示学习方法, 且能适配于多种语言模型, 有效提升 OWL 复杂语义的知识表示能力.

关键词: 知识表示学习; OWL 知识图谱; 多语义视图; 进阶训练; 多任务学习

中图法分类号: TP182

中文引用格式: 杨建喜, 谢江村, 李韧, 杨小霞, 肖桥, 蒋仕新, 贺丽荣. 多语义视图驱动的 OWL 知识图谱表示学习方法. 软件学报, 2025, 36(12): 5644–5673. <http://www.jos.org.cn/1000-9825/7402.htm>

英文引用格式: Yang JX, Xie JC, Li R, Yang XX, Xiao Q, Jiang SX, He LR. OWL Knowledge Graph Representation Learning Method via Multi-semantic View. Ruan Jian Xue Bao/Journal of Software, 2025, 36(12): 5644–5673 (in Chinese). <http://www.jos.org.cn/1000-9825/7402.htm>

OWL Knowledge Graph Representation Learning Method via Multi-semantic View

YANG Jian-Xi¹, XIE Jiang-Cun¹, LI Ren¹, YANG Xiao-Xia², XIAO Qiao², JIANG Shi-Xin¹, HE Li-Rong¹

¹(School of Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China)

²(College of Traffic and Transportation, Chongqing Jiaotong University, Chongqing 400074, China)

Abstract: To address the issues in current OWL representation learning methods, which lack the ability to jointly represent complex semantic information across both the concept layer and the instance layer, an OWL representation learning approach using multi-semantic views of concepts, properties, and instances is proposed. The proposed method adopts a three-stage architecture including multi-semantic views partitioning, semantic-aware self-supervised post-training, and joint multi-task representation learning. First, MSV-KRL optimizes the mapping strategy from OWL to RDF graphs based on OWL2Vec^{*}, and five fine-grained semantic view partitioning strategies are proposed. Subsequently, serialized post-training data is generated through the random walk and annotated attribute replacement strategy. The self-supervised post-training of the pre-trained model is then carried out to enhance adaptability to multi-semantic views. Finally, by employing a multi-task learning strategy, the complex semantic representation learning of concepts, properties, and instances in OWL graphs is achieved through joint optimization loss of multi-semantic view prediction tasks. Experimental results demonstrate that MSV-KRL outperforms baseline representation learning methods on multiple benchmarks. MSV-KRL can be adapted to multiple language models, significantly improving the knowledge representation capability of OWL's complex semantics.

^{*} 基金项目: 国家自然科学基金 (62003063); 重庆市教委科学技术研究重大项目 (KJZD-M202300703); 重庆市自然科学基金面上项目 (CSTB 2023NSCQ-MSX0145); 重庆市教委科学技术研究青年项目 (KJQN202200720)

收稿时间: 2024-07-03; 修改时间: 2024-10-31, 2024-12-30; 采用时间: 2025-02-11; jos 在线出版时间: 2025-07-09

CNKI 网络首发时间: 2025-07-10

Key words: knowledge representation learning (KRL); OWL knowledge graph; multi-semantic view; post-training; multi-task learning

作为一种符号化的知识载体, 知识图谱 (knowledge graph, KG) 描述了现实世界中实体及其属性之间的语义关联^[1,2], 为智能问答^[3,4]、推荐系统^[5,6]等下游任务提供了知识表示基础。目前, 知识图谱受到了大数据和人工智能研究领域和产业界的广泛关注, 但业界尚未形成一种统一的定义。本文根据语义表征能力的不同, 将知识图谱分为基于资源描述框架 (resource description framework, RDF)^[7]的实例图谱和基于 RDFS (RDF schema)^[8,9]或 OWL (Web ontology language)^[10,11]的本体语义图谱。其中, RDF 实例图谱由描述实体对象及其属性之间简单语义关系的主谓宾三元组构成, 是 FreeBase^[12]等通用大规模知识图谱的主要表征方式。RDFS 语义图谱在实例图谱的基础上引入了概念、属性等语义模式, 能够对概念-实例关系及类层级关系等语义约束进行形式化定义。基于描述逻辑的 OWL 语义图谱在 RDFS 语义基础上增加了类之间交集、并集等重要算子, 同时将属性细分为对象属性和数据属性, 且为属性增加了逆关系等属性特性, 具有更强大的语义表示能力。OWL 本体语义图谱在垂直领域知识表征方面得到了广泛应用。目前, 业界已提出了基于 OWL 的通用领域图谱 DBpedia^[13], 以及面向医疗健康领域的 HeLis^[14]、FoodOn^[15]和生物信息领域的 GO^[16]等众多 OWL 语义图谱。

为实现符号化知识图谱信息在低维连续向量空间中的有效嵌入, 知识表示学习 (knowledge representation learning, KRL) 成为知识图谱研究领域中的重要新兴任务, 也是知识图谱补全或链接预测的基础^[17,18]。例如, 经典的 TransE 算法^[19]基于向量平移思想对三元组进行建模, 利用头、尾实体和关系的映射向量表达原图中的语义信息和链接关系, 然而, TransE 仅能对简单实体关系进行向量化表征, 在编码一对多、多对一和多对多等复杂实体语义关系时存在较大局限。之后, 业界又相继提出了 TransG^[20]和 TransRD^[21]等多种改进方法, 尝试解决实例图谱中复杂关系的语义向量表示、实体不对称等问题。近年来, 融合本体语义信息的知识表示学习成为该方向研究的重要趋势。例如, TransO^[22]基于 RDFS 语义模式, 在知识嵌入过程中融入实例类型、子类、子属性以及属性的定义域 (domain) 和值域范围 (range) 信息, 但仍缺乏概念、属性之间复杂语义公理的向量化知识表示能力。一些方法针对 OWL 图谱语义模式层中 EL 与 ALC 两种描述逻辑进行知识向量建模^[23,24]或利用优先搜索算法获取类层级关系^[25], 但忽略了图谱实例层相关语义信息的有效结合。OWL2Vec*^[26]面向 OWL 本体图谱的知识表示学习任务, 通过 Word2Vec 模型在基于 OWL 语义构造的序列文本文档上进行训练, 实现了 OWL 语义模式层和实例层联合的向量化知识嵌入。然而, OWL2Vec* 仍然难以有效处理复杂的概念术语层级语义关系, 在面向 OWL 图谱语义模式层和实例层信息融合的知识表示学习能力等方面仍有待提升。

为克服现有方法的复杂语义模式与实例信息联合嵌入表征不足这一问题, 本文提出一种新的多语义视图驱动的 OWL 图谱知识表示学习方法 (multiple semantic views-based knowledge representation learning, MSV-KRL)。该方法基于实体多语义空间联合认知思想, 即某一实体在多个语义空间中分别存在不同的嵌入表示, 而该实体的全量知识表示是对应多空间嵌入的有效融合。总体上, MSV-KRL 方法采用“多语义视图划分、语义感知进阶训练、多任务联合表示学习”的三阶段递进式架构。首先, 在多语义视图划分阶段, MSV-KRL 在 OWL2Vec* 本体映射方法基础上进行改进, 并提出类层级关系视图、类属性关系视图、类实例关系视图、实例对象属性视图和实例数据属性视图这 5 类细粒度子图划分策略, 构建概念-属性-实例在不同认知维度下的多语义视图。在语义感知进阶训练阶段, MSV-KRL 方法在各语义子图上进行随机游走, 并采用标注属性替换策略生成进阶训练数据, 实现 OWL 语义图谱中多种语义公理信息、对象/数据属性信息以及标注信息的充分利用。然后, 采用自动构建的进阶训练数据对预训练模型或大语言模型进行自监督训练微调, 使其适配于多语义视图上下文。在多任务联合表示学习阶段, MSV-KRL 在传统的类实例预测和类层级预测任务基础上, 引入类属性预测、实例对象属性预测以及实例数据属性预测 3 种新的任务, 通过适配于多语义视图的联合表示学习提升针对 OWL 语义图谱的语义模式层和实例层联合向量表征能力。本文的主要贡献可概括如下。

(1) 提出了一种 OWL 语义图谱概念-属性-实例细粒度语义感知的多语义视图划分策略, 以及基于该多语义视图的进阶训练数据自动构建方法, 并通过自监督进阶训练微调模型参数, 实现面向多语义视图上下文的适配能力提升。

(2) 提出了包含类实例预测、类层级关系预测、类属性预测、实例对象属性预测和实例数据属性预测的多任

务联合表示学习方法,提升模型在多语义视图驱动下的向量联合表征能力。

(3) 在多个基准数据集上开展实验对比分析,结果表明,所提出的 MSV-KRL 方法在绝大多数评测任务中性能优于当前先进的基线模型,且能适配多种语言模型,并有效提升 OWL 复杂语义的知识表示能力。

本文第 1 节介绍知识图谱表示学习的相关方法和研究现状。第 2 节首先阐述 OWL 知识表示学习任务定义,然后介绍提出方法的总体架构,详细讲述各个核心模块的关键内容。第 3 节给出相应的实验结果和分析。第 4 节总结全文,并展望后续工作。

1 相关工作

根据知识图谱语义表示能力和建模对象的不同,本文将现有的知识表示学习方法分为面向实例图谱的表示学习和面向本体语义图谱的表示学习。本节分析相关工作研究现状,并总结现有方法的不足。

1.1 面向实例图谱的知识表示学习

实例图谱以 RDF 三元组为基本结构,描述了头实体和尾实体之间的语义关系。面向实例图谱的知识表示学习旨在将实体和关系嵌入到低维连续的向量空间,实现实例关系事实 (h, r, t) 的向量化表达。目前,实例图谱表示学习方法可分为翻译距离模型 (translational distance model, TDM) 和语义匹配模型 (semantic matching model, SMM) 两类。其中, Bordes 等人^[19]提出的 TransE 模型是 TDM 的典型代表,并成为后续研究模型扩展的基础。TransE 方法将知识图谱语义关系视为实体之间的空间平移。对于实例断言三元组 (h, r, t) , 将关系 r 表示为翻译向量 t_r , 并将其视为头实体向量 l_h 和尾实体向量 l_t 的空间平移,即 $l_h + l_r \approx l_t$ 。然而, TransE 仅能对简单实体关系进行向量化表征,在编码一对多、多对一和多对多等复杂实体语义关系时存在较大局限。之后,国内外学者又提出了 TransH^[27]、TransR^[28]、TransD^[29]、TransG^[20]和 TransRD^[21]等多种改进方法,尝试解决复杂实体关系的语义向量表示和实体不对称等问题。RESCAL^[30]、DistMult^[31]和 ComplEx^[32]等是 SMMs 类模型的典型代表。RESCAL 将每个实体与向量关联,以捕获其潜在语义。每个关系都表示为一个矩阵,该矩阵对实体潜在语义之间的两两相互作用进行建模,并利用双线性函数对三元组事实陈述进行打分。然而,上述知识表示学习方法仅面向静态实例图谱,无法向量化表征动态图谱的语义关系。

近年来,面向大规模动态实例图谱的知识表示学习成为该领域研究的新兴任务之一。例如, puTransE^[33]使用了多个并行的嵌入空间对知识图谱的局部子图进行表示,并选择全局最高能量值进行多空间的链接预测。iTransA^[34]实现了实体嵌入空间间隔和关系嵌入空间间隔的在线优化,但在图谱嵌入学习过程中仍需重复训练所有三元组。Daruna 等人^[35]提出的 CKGE 模型应用了连续学习策略进行动态知识图谱嵌入。Wu 等人^[36]提出了上下文感知的动态知识图谱嵌入模型 DKGE,实现了子图嵌入在线更新。另一方面,将实例图谱中存在的实体描述文本信息与实体关系图结构信息相结合,用于提升实例图谱知识表示学习的效果,成为当前的又一研究趋势。例如,姚思雨等人^[37]利用卷积神经网络同时对实例图谱结构信息和文本信息进行建模,在链接预测任务上表现出不俗的性能。随着近年来预训练语言模型 (pretrained language model, PLM) 和大型语言模型 (large language model, LLM) 的快速发展,其强大的上下文理解能力与文本编码能力得到了充分展现,国内外学者开始研究将 PLM 和 LLM 引入知识表示学习领域。例如, Nayyeri 等人^[38]提出了融合多类型结构化知识和文本信息的统一框架。Xiao 等人^[39]提出了融合实体邻接信息及描述信息的知识表示学习模型。宁原隆等人^[40]提出了融合关系路径与实体描述信息的知识图谱表示学习方法。BERT-PKE^[41]将 BERT 预训练模型作为语义解析器,实现了实体和关系描述文本在知识表示学习任务中的有效利用。另外, KG-LLM^[42]通过微调大语言模型实现了多跳链接预测, Yao 等人^[43]将大语言模型应用于知识图谱补全。然而,上述工作均面向实体关系简单陈述的实例知识图谱,尚无法实现语义模式层概念、对象/数据属性及其全称量词、存在量词等复杂公理约束的知识表示学习。

1.2 面向本体语义图谱的知识表示学习

本体语义图谱在 RDF 三元组的实例关系断言声明的基础上,采用 RDFS、OWL 等不同语义表达能力的本体建模,实现概念、属性及其公理约束的形式化定义,形成涵盖语义模式层和实例层的复杂知识图谱。国内外学者已

开展较多的面向本体语义图谱的知识表示学习相关研究。

为实现本体语义知识图谱解析并使其适配于文本序列编码器或图编码模型, 现有方法通常采用 W3C 定义的映射规则^[44]将 OWL 图谱映射为基于 RDF 格式的三元组集合, 但在涉及到复杂公理时这种基础映射方式存在部分空白节点, 造成映射结果丢失部分语义信息, 或者在映射过程中会生成冗余部分。为克服上述不足, 业界提出了多种优化方式, 例如 Onto2Graph^[45]、DL2Vec^[46]以及 OWL2Vec*^[26]等。

近年来, 业界提出了一些针对语义模式层本体的知识表示学习方法。例如, BERTSubs^[25]提出单类、深度优先及广度优先这 3 种方式获取类层级关系的层次信息, 并利用 BERT 预训练语言模型对描述文本进行编码。Shiraishi 等人^[47]提出 Inverted-index 矩阵嵌入和 Co-occurrence 矩阵嵌入分别获取公理中每一个词的全局和局部信息, 并通过自编码器进行压缩, 但是该方法无法处理存在量词等复杂公理。ONTOLAMA^[48]通过语言模型实现 OWL 图谱类层级关系的推理, 能够同时实现原子概念以及复杂概念类层级关系推理。为了弥补仅使用描述文本表示概念语义能力的不足, Shi 等人^[49]通过引入概念的实例以增强其知识表示能力, 实现类层级关系推理。由于 OWL 图谱中的类层级关系通过 RDFS 语义进行定义, 因此这类方法虽然采用了 OWL 图谱中的部分信息, 但仍属于面向 RDFS 语义的知识表示学习范畴。同时, 上述方法并没有考虑实例层信息, 学习到的类嵌入包含的语义信息不足。最近一些工作重点关注于 OWL 语义图谱中基于描述逻辑的公理信息知识嵌入。例如, Box²EL^[23]面向存在量化逻辑 (existential quantification logic, EL), 利用 Box 嵌入的方式表示概念以及角色, 实例则为其中的一个点, 通过碰撞机制对关系进行建模。CatE^[24]是一种面向 ALC 描述逻辑的知识表示学习方法。然而, 上述现有方法忽略了 OWL 语义图谱中丰富的标注属性信息、关系层次信息以及实例层语义信息。

融合语义模式层与实例层的知识表示学习成为近年来的研究热点。JOIE^[50]利用跨视图关联模型桥接语义模式层实体和实例层实体, 并通过内视图模型分别在两个独立的空间中建模类和实例的结构信息。DGS^[51]利用非欧几里得几何空间的球面空间和双曲空间分别建模类及实例, 通过空间交汇桥接概念和实例视图, 实现概念和实例知识的联合嵌入表示。Concept2Box^[52]利用 Box 嵌入对类和实例进行建模, 并提出向量到 Box 的距离计算方法用以桥接概念和实例实体。TransO^[22]在翻译距离模型的基础上进行优化, 在对三元组结构建模时融入 RDFS 图谱类之间的层次信息 (subClassOf) 以及关系的层次信息 (subPropertyOf), 并依据相关公理进行了推理, 实现语义模式层信息融合的知识表示学习。在图表示学习领域也存在类似的方法, MV-HGSL^[53]通过元路径图和相似度图进行对比学习实现对异质图结构的学习。然而, 上述方法着重于 RDFS 语义模式与实例层信息的交互, 同时仅通过几何空间学习到的概念与实例层之间的联系并不完全, 而且难以利用 OWL 图谱中丰富的对象、数据和标注属性信息。RDFS 相对于 OWL 的表示能力有限, 面向复杂 OWL 语义公理的表示学习能力仍有待提升。

EL embedding^[54]和 Quantum embedding^[55]是两种端到端的 OWL 语义图谱表示学习方法。这两种方法通过将逻辑关系转换成几何关系来构建打分函数和损失函数。但这两种方法只利用到了类实例以及类属性这两种关系, 且依然没有引入 OWL 图谱中的文本属性信息。Onto2Vec^[56]以及 OPA2Vec^[57]这两种方法通过构建公理语料库, 然后利用 Word2Vec 词嵌入模型学习实体及属性的嵌入。OPA2Vec 在 Onto2Vec 方法的基础上利用了 OWL 图谱中的词汇信息。然而, 这两种方法将单个公理视为一个序列, 难以捕捉 OWL 中的图结构信息。OWL2Vec*^[26]在前两种方法基础上通过随机游走获取到公理之间的联系, 同时在使用了 *rdfs:comment* 标注属性基础上增加了更多的标注属性, 比如 *rdfs:label*。然而, OWL2Vec*在整个 OWL 图谱结构或本体映射后的 RDF 图谱上采用随机游走难以有针对性地捕捉具有高度语义相关性的子图信息, 且其基于 Word2Vec 的 URI 嵌入和词嵌入表示方式对上下文的特征建模能力较弱。另外, OWL2Vec*方法在对象属性的复杂定义域 (domain) 和值域 (range) 关系解析方面存在不足, 使得面向复杂语义公理的知识表示能力仍有待提升。

2 MSV-KRL 方法

本节首先对问题进行描述定义, 其次介绍本文所提 MSV-KRL 的总体架构, 并具体介绍关键模块。

2.1 问题定义

OWL 语义图谱形式化定义为 $\mathcal{O} = \{C, R, I, D, L\}$, 其中 C 和 I 分别表示原子概念 (concept) 集合和实例 (instance

或 individual) 集合. 关系 (relation) 集合 $R = \{r_i, r_s\} \cup R_o \cup R_d$, 其中 r_i 表示 $rdf:type$ 关系, r_s 表示 $rdfs:subClassOf$ 关系, R_o 表示对象属性 (object property) 关系集合, R_d 表示数据属性 (data property) 关系集合. 复杂概念可由描述逻辑算子合取 (例如 $c_i \sqcap c_j$)、析取 (例如 $c_i \sqcup c_j$)、存在量词 (例如 $\exists r.c_i$) 和全称量词 (例如 $\forall r.c_i$) 组合表征. D 表示数值、文本等数据属性对应的字面量集合, L 表示概念、实例或属性所对应的标注属性信息 (annotation) 集合. 针对 $\mathcal{O}$ 的知识表示学习旨在将所有的 $c \in C$ 、 $i \in I$ 和 $r \in R$ 嵌入为低维稠密特征空间中的向量 $v_c \in \mathbb{R}^d$, $v_i \in \mathbb{R}^d$, $v_r \in \mathbb{R}^d$, 使其尽可能地量化表征 OWL 图谱中各种概念、实例、属性及其公理的语义信息.

2.2 整体框架

本文所提出的 MSV-KRL 整体框架如图 1 所示. 该方法主要包含了 3 个阶段: 多语义视图划分、语义感知进阶训练、多任务联合表示学习.

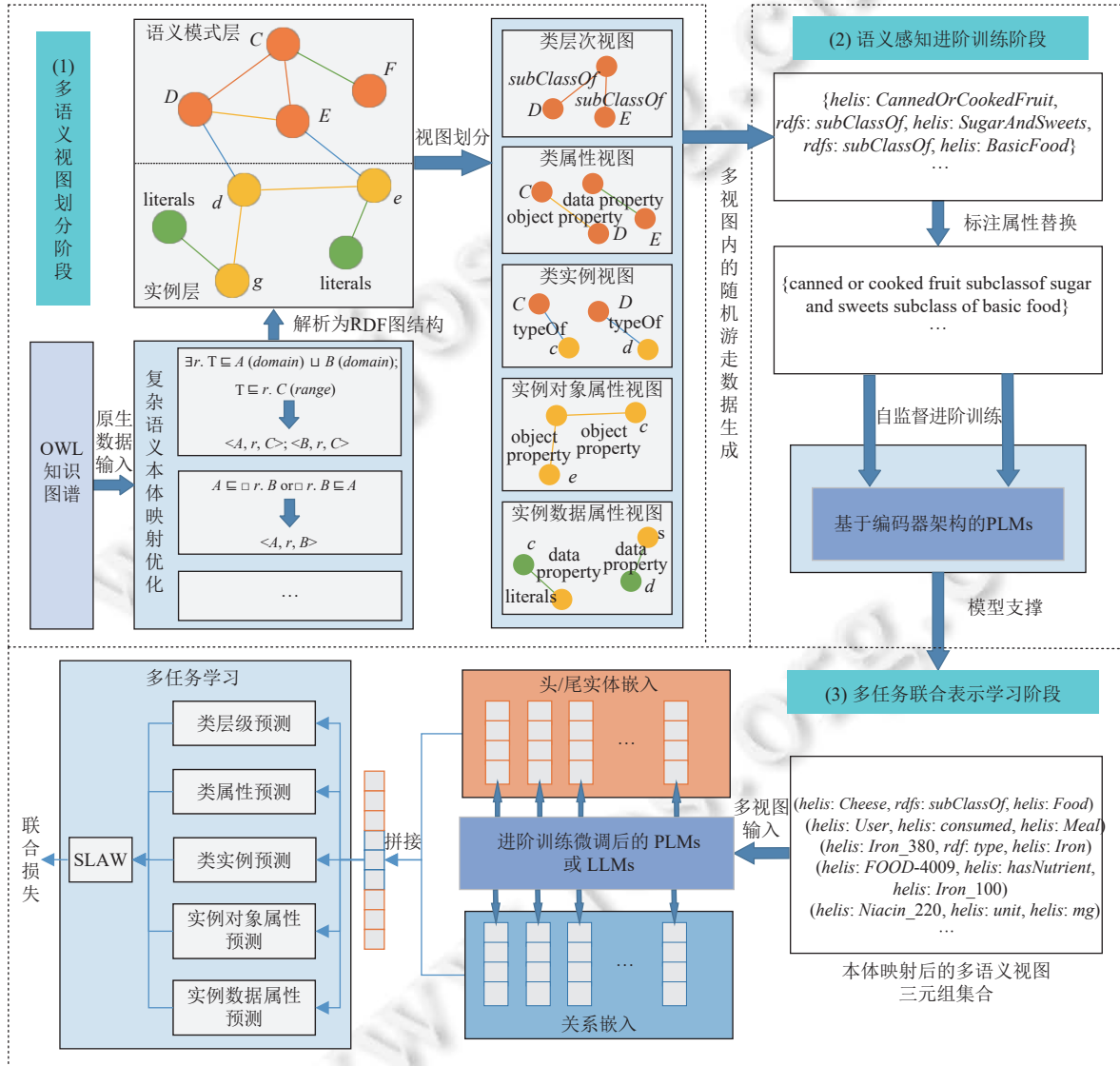


图 1 MSV-KRL 整体框架

在多语义视图划分阶段, MSV-KRL 首先在 OWL2Vec*定义的本体映射策略基础上进行优化, 尤其是针对对象属性的定义域和值域复杂语义约束, 实现 OWL 图谱到 RDF 图结构的更合理解析. 然后, 基于实体多语义空间

联合认知思想, 提出面向语义模式层和实例层语义信息联动的多语义视图划分策略, 分别将映射生成的 RDF 图进一步解析为类层级关系视图、类属性关系视图、类实例关系视图、实例对象属性视图和实例数据属性视图, 用以表示概念、实例及多类型属性的多维度语义关联。

在语义感知进阶训练阶段, 首先对上一阶段生成的 5 种类型语义视图分别采用随机游走策略构建进阶训练数据, 实现从图结构数据向序列化的三元组文本转换。然后, 为进一步引入概念、实例和各类型属性的标注文本描述信息, 设计了标注属性替换策略, 用于进一步丰富三元组序列数据的上下文语义信息。最后, 根据所采用的 PLM 或 LLM 模型基座, 根据生成的三元组序列特性设计掩码策略, 并进行自监督的模型进阶训练, 使得基座模型更适配于所构建的多语义视图上下文。

在多任务联合表示学习阶段, 采用映射生成的多语义视图三元组集合为数据基础, 将其输入进阶训练后的 PLM 或 LLM 模型, 获取头尾实体嵌入和关系嵌入, 拼接后通过类层级关系预测、类属性预测、类实例预测、实例对象属性预测以及实例数据属性预测展开多任务联合表示学习, 采用联合损失度量进行模型参数优化, 构建适配于多语义视图输入与多任务学习输出的端到端模型, 并获取概念-属性-实例联合感知的知识嵌入。

接下来, 将在第 2.3、2.4 和 2.5 节中详细介绍 MSV-KRL 中各阶段的核心方法。

2.3 概念-属性-实例多视图语义划分

为了实现在多语义视图中有效表征 OWL 语义图谱概念、实例及其对象属性、数据属性等复杂语义信息, MSV-KRL 根据 OWL 本体语义模式特性, 包含了复杂语义本体映射和多语义视图划分两个子任务。

2.3.1 OWL 本体到 RDF 图结构的语义映射优化

本文采用的本体映射方法总体上与 OWL2Vec*、Soylu 等人^[58]和 Holter 等人^[59]的方法相似, 如表 1 所示。给定 OWL 语义描述的条件 1 公理及条件 2 公理, 将其映射为对应的 RDF 三元组形式。例如, OWL 实例断言声明 $A(a)$ 可映射生成三元组 $\langle a, \text{rdf:type}, A \rangle$ 和 $\langle A, \text{rdf:type}^-, a \rangle$, 其中 A 是原子概念, a 是实例。对于对象属性 r , 同时声明有定义域 (*domain*) 为 A , 值域 (*range*) 为 B , 那么可映射为三元组 $\langle A, r, B \rangle$ 。另外, 表 1 中还指出了概念和属性的语义层级、量词约束等语义公理的映射规则。

表 1 本体映射规则

条件1公理	条件2公理	映射的三元组
$A \sqsubseteq \Box r.D$ or $\Box r.D \sqsubseteq A$ $r \text{ domain: } A$ $A \sqsubseteq \Box r.\{b\}$ $r \sqsubseteq r'$ $r' \equiv r^-$ $s_1 \circ \dots \circ s_n \sqsubseteq r$	$D \equiv B B_1 \sqcup \dots \sqcup B_n B_1 \sqcap \dots \sqcap B_n$ $r \text{ range: } B$ $B(b)$ $\langle A, r', B \rangle$ in graph $\langle B, r', A \rangle$ in graph $\langle A, s_1, C_1 \rangle \dots \langle C_n, s_n, B \rangle$ in graph	$\langle A, r, B \rangle$ or $\langle A, r, B_i \rangle$ for $i \in 1, \dots, n$
$B \sqsubseteq A$	—	$\langle B, \text{rdfs:subClassOf}, A \rangle$
	—	$\langle A, \text{rdfs:subClassOf}^-, B \rangle$
$A(a)$	—	$\langle a, \text{rdf:type}, A \rangle$
	—	$\langle A, \text{rdf:type}^-, a \rangle$
$r(a, b)$	—	$\langle a, r, b \rangle$

注: $\Box$ 是 $\geq$ 、 $\leq$ 、 $=$ 、 $\exists$ 及 $\forall$ 其中之一; A 、 B 、 B_i 和 C_i 表示原子概念 (类); s_i 、 r 和 r' 表示角色 (对象属性), r^- 是 r 的逆关系; a 和 b 是个体 (实例)

然而, 表 1 所示的本体映射规则仅能处理对象属性相应定义域和值域的原子概念语义映射, 对多原子概念交并等复杂语义的解析映射能力不足, 尤其是当某一对象属性的 *rdfs:domain* 是由多个原子概念通过交或并连接定义而成, 而 *rdfs:range* 是单个原子概念构成的情况, 就无法实现语义映射和三元组生成。另外, 在处理数据属性的值域映射时, 其三元组宾语处的字面量尚未关联对应的数据类型信息, 存在一定的语义映射缺失问题。因此, 本文在表 1 的映射规则基础上进行扩展, 分别针对对象属性定义域、值域复杂概念语义映射和数据属性值域字面量数据类型映射问题进行优化。算法 1 描述了针对对象属性定义域和值域复杂语义的映射优化算法。

算法 1. 对象属性定义域、值域复杂概念语义映射算法.

输入: 本体 $\mathcal{O}$ 中的对象属性集合 $R_o \subseteq R$;

输出: 映射后的三元组集合 P .

```

1.  $P \leftarrow \text{Initialize}(\cdot)$ ; //初始化一个空集合
2. for  $r_o \in R_o$  do //循环处理对象属性
3.    $C_{\text{domain}}, C_{\text{range}} \leftarrow$  分别执行 SPARQL 查询语句获取对象属性  $r_o$  的复杂定义域和复杂值域概念语义声明
4.   if  $\text{len}(C_{\text{domain}}) \neq 0$  and  $\text{len}(C_{\text{range}}) = 0$  do //若  $C_{\text{range}}$  集合为空则表明其为简单值域语义声明
5.      $C_{\text{range}} \leftarrow$  基于  $r_o$  的简单值域 SPARQL 查询语句更新  $C_{\text{range}}$  概念集合
6.   end if
7.   if  $\text{len}(C_{\text{domain}}) = 0$  and  $\text{len}(C_{\text{range}}) \neq 0$  do //若  $C_{\text{domain}}$  集合为空则表明其为简单定义域语义声明
8.      $C_{\text{domain}} \leftarrow$  基于  $r_o$  的简单定义域 SPARQL 查询语句更新  $C_{\text{domain}}$  集合
9.   end if
10.  for  $c_{\text{domain}} \in C_{\text{domain}}$  do
11.    for  $c_{\text{range}} \in C_{\text{range}}$  do
12.       $P.add((c_{\text{domain}}, r_o, c_{\text{range}}))$ ; //循环添加结果至三元组集合
13.    end for
14.  end for
15. end for
16. return  $P$ 

```

代码 1. 复杂定义域 SPARQL 查询示例.

```

1. SELECT DISTINCT ?d
2. WHERE
3. { { helis : hasNutrient rdfs : domain [ owl : intersectionOf [ rdf : rest* [ rdf : first ?d ] ] ] }
4. UNION
5. { helis : hasNutrient rdfs : domain [ owl : unionOf [ rdf : rest* [ rdf : first ?d ] ] ] }
6. FILTER (isIRI(?d)) }

```

代码 2. 简单值域 SPARQL 查询示例.

```

1. SELECT DISTINCT ?r
2. WHERE
3. { helis : hasNutrient rdfs : range ?r
4. FILTER (isIRI(?r)) }

```

在算法 1 中, 初始化一个空集合 P , 用于存储映射后生成的对象属性相关三元组, 并对输入图谱中的所有对象属性循环执行以下操作. 首先, 分别执行 SPARQL 语句获取 r_o 的复杂定义域和复杂值域概念声明集合 C_{domain} 和 C_{range} , 即获取定义域或值域声明中包含两个或两个以上概念的交并等语义声明. 其次, 判断 C_{domain} 集合与 C_{range} 集合是否为简单定义域或值域声明, 若是则重新通过简单定义域或值域 SPARQL 查询语句进行更新; 最后将 C_{domain} 和 C_{range} 中的每一个元素通过当前对象属性 r_o 进行链接构成三元组.

$$\text{Domain}(\text{helis} : \text{hasNutrient}) = \text{helis} : \text{Food} \cup \text{helis} : \text{Meal} \quad (1)$$

$$\text{Range}(\text{helis} : \text{hasNutrient}) = \text{helis} : \text{Nutrient} \quad (2)$$

例如公式 (1)、(2) 所示, HeLis 本体中对象属性 $\text{helis} : \text{hasNutrient}$ 同时存在复杂定义域声明: $\text{helis} : \text{Food} \cup \text{helis} : \text{Meal}$, 以及简单值域声明: $\text{helis} : \text{Nutrient}$. 当算法 1 中第 2 行循环到该对象属性后, 首先通过复杂定义域 SPARQL 查询 (如代码 1 所示) 和复杂值域 SPARQL 查询获取定义域概念集合以及值域概念集合, 复杂值域 SPARQL 只需将 $\text{rdfs} : \text{domain}$ 替换为 $\text{rdfs} : \text{range}$ 即可. 其次, 由于复杂定义域或值域 SPARQL 查询无法查询到简单值域或定义域概念声明, 因此当前对象属性无法查询到 $\text{helis} : \text{Nutrient}$, 即 $\text{len}(C_{\text{domain}}) \neq 0$ 但 $\text{len}(C_{\text{range}}) = 0$, 那么将满足算法 1 中第 4 行判断语句, 不满足第 7 行判断语句, 则在第 5 行通过简单值域 SPARQL 查询 (如代码 2 所示) 重新获取简单值域概念集合, 简单定义域 SPARQL 查询只需替换 $\text{rdfs} : \text{range}$ 为 $\text{rdfs} : \text{domain}$ 即可. 最后则需通过算法 1 第 10–14 行循环定义域、值域集合生成映射结果, 具体三元组如公式 (3) 和公式 (4) 所示.

$$\langle \text{helis} : \text{Food}, \text{helis} : \text{hasNutrient}, \text{helis} : \text{Nutrient} \rangle \quad (3)$$

$$\langle \text{helis} : \text{Meal}, \text{helis} : \text{hasNutrient}, \text{helis} : \text{Nutrient} \rangle \quad (4)$$

OWL2Vec* 方法为了能够综合利用本体中的公理信息、词汇信息以及图结构信息等, 同时使用 Word2Vec 词嵌入模型, 采用本体映射的方法将本体中的上述信息抽取出来成为一个语料库, 然后使用 Word2Vec 进行训练获取实体和关系嵌入. 当 OWL2Vec* 本体映射方法在使用代码 1 复杂 SPARQL 查询某一对象属性的 domain 或 range 时, 若其是由原子概念构成, 则会无法获取当前 domain 或 range , 造成当前三元组信息抽取失败, 导致语义信息的丢失. 然而, 通过改进后的算法 1 可以有效避免该情况, 当无法通过代码 1 获取到查询结果时, 则会触发判断语句重新获取由原子概念构成的 domain 或 range , 最后将新生成的三元组信息添加至三元组集合 P , 补充原来丢失的语义信息.

算法 2 描述了数据属性值域字面量数据类型语义映射算法.

算法 2. 数据属性值域字面量数据类型语义映射算法.

输入: 本体 $\mathcal{O}$ 中数据属性集合 $R_d \subseteq R$;

输出: 映射后的三元组集合 Q .

```

1.  $Q \leftarrow \text{Initialize}(\cdot)$ ; //初始化一个空集合
2. for  $r_d \in R_d$  do //循环处理数据属性
3.    $C_{\text{domain}_s}, C_{\text{range}_s} \leftarrow$  分别执行 SPARQL 查询语句获取  $r_d$  的简单定义域和简单值域语义声明
4.    $C_{\text{domain}_c} \leftarrow$  执行 SPARQL 查询语句获取  $r_d$  的复杂定义域语义声明
5.    $C_{\text{domain}} \leftarrow C_{\text{domain}_s} \cup C_{\text{domain}_c}$  //同时获取  $r_d$  的简单定义域和复杂定义域语义声明
6.   for  $c_{\text{domain}} \in C_{\text{domain}}$  do
7.     for  $c_{\text{range}} \in C_{\text{range}_s}$  do
8.        $Q.\text{add}((c_{\text{domain}}, r_d, c_{\text{range}}))$ ; //循环添加结果至三元组集合
9.     end for
10.  end for
11. end for
12. return  $Q$ 
```

代码 3. 数据属性简单定义域 SPARQL 查询示例.

```

1. SELECT DISTINCT ?d
2. WHERE
3. { helis : amountOfEnergy rdfs : domain ?d
4. FILTER (isIRI(?d)) }
```

代码 4. 数据属性复杂定义域 SPARQL 查询示例.

```

1. SELECT DISTINCT ?d
2. WHERE
3. { { helis : amountOfEnergy rdfs : domain [ owl : intersectionOf [ rdf : rest* [ rdf : first ?d ] ] ] }
4. UNION
5. { helis : amountOfEnergy rdfs : domain [ owl : unionOf [ rdf : rest* [ rdf : first ?d ] ] ] }
6. FILTER (isIRI(?d)) }

```

在算法 2 中, 初始化一个空集合 Q , 用于存储映射后生成的数据属性三元组, 并对输入图谱中的所有数据属性循环执行以下操作. 首先, 分别执行 SPARQL 查询语句获取数据属性 r_d 的简单定义域语义声明 C_{domain_s} 和简单值域语义声明 C_{range_s} , 其中, C_{range_s} 存储为字面量的数据类型 (例如 $xsd : double$). 然后, 再执行 SPARQL 查询语句获取数据属性 r_d 可能存在的复杂定义域语义声明 C_{domain_c} , 并执行 C_{domain_s} 和 C_{domain_c} 的合并运算得到 C_{domain} . 最后, 将 C_{domain} 和 C_{range_s} 中的每一个元素通过当前数据属性 r_d 进行链接, 映射生成三元组.

$$Domain(helis : amountOfEnergy) = helis : Food \cup helis : Meal \quad (5)$$

$$Range(helis : amountOfEnergy) = xsd : double \quad (6)$$

例如公式 (5)、(6) 所示, HeLis 本体中数据属性 $helis : amountOfEnergy$ 存在定义域语义声明 $helis : Food \cup helis : Meal$, 存在值域语义声明 $xsd : double$. 当算法 2 中第 2 行循环至该数据属性时, 首先在第 3 行通过简单定义域 SPARQL 查询 (如代码 3 所示) 和简单值域 SPARQL 查询分别获取简单定义域概念集合以及简单值域概念集合, 其中简单值域 SPARQL 查询即将代码 3 中 $rdfs : domain$ 替换为 $rdfs : range$ 即可. 其次, 算法 2 第 4 行通过复杂定义域 SPARQL 查询 (如代码 4 所示) 获取多个定义有交并操作的原子概念集合. 最后通过算法 2 中第 6–10 行映射生成三元组, 具体如公式 (7) 和公式 (8) 所示.

$$\langle helis : Food, helis : amountOfEnergy, xsd : double \rangle \quad (7)$$

$$\langle helis : Meal, helis : amountOfEnergy, xsd : double \rangle \quad (8)$$

OWL 本体中数据属性的 *domain* 通常是单独由原子概念或多个原子概念的交或并构成, 而 *range* 是由具体字面量值的类型构成, 例如浮点类型数据的类型为 $xsd : double$ 或 $xsd : float$ 、整型数据的类型为 $xsd : int$ 、中英文等字符类型数据的类型为 $xsd : string$. OWL2Vec* 本体映射方法在处理数据属性时, 仅抽取了其 *domain*, 并没有抽取 *range*, 且并没有将该类型三元组添加进语料库中, 直接损失了该类语义信息. 同时, OWL2Vec* 本体映射方法未能抽取由多个原子概念交或并构成的复杂 *domain*. 然而, 通过算法 2 可以有效同时解决上述两个问题. 首先, 算法 2 通过第 3–5 行能够同时获取数据属性简单和复杂的 *domain*, 并通过并集操作有效避免了原方法不能处理复杂 *domain* 的问题. 与此同时, 由于数据属性的 *range* 通常是由单个具体的 Datatype 构成, 则通过算法 2 第 3 行可以实现获取数据属性的 *range*. 最终, 将抽取出来的三元组添加进三元组集合. 通过算法 2 的优化, 可以有效地补充原本体映射方法所缺失的数据属性相关语义信息.

2.3.2 面向概念-属性-实例多视图的语义划分策略

经本体语义映射后, OWL 语义图谱 $\mathcal{O}$ 转换为由若干三元组构成的 RDF 图谱 $\mathcal{G}$. 现有的知识表示学习方法, 例如 OWL2Vec* 等, 大多采用在 $\mathcal{G}$ 整图上进行随机游走的策略获取其图结构信息, 然而这样的策略在单次执行中将会随机游走于多种关系, 难以捕捉到特定关系的层次信息, 反而可能在训练过程中引入噪声, 降低模型的嵌入表示学习能力. 与现有方法不同的是, 本文提出的 MSV-KRL 充分考虑图谱中概念、属性和实例之间的类层级关系、类实例关系、类属性关系、实例对象属性关系以及实例数据属性关系等多维度语义关联, 采用多语义视图划分机制对 $\mathcal{G}$ 进行进一步解析, 实现在不同认知维度下的知识联合表示学习.

具体而言, MSV-KRL 定义了 5 类互斥的语义视图, 分别是: 类层级关系视图 T_s 、类实例关系视图 T_i 、类属性关系视图 T_c 、实例对象属性视图 T_o 和实例数据属性视图 T_d , 具体定义如公式 (9)–公式 (13) 所示.

$$T_s = \{(c_m, r_s, c_n) \mid c_m, c_n \in C\} \quad (9)$$

$$T_t = \{(i, r_t, c) \mid i \in I, c \in C\} \quad (10)$$

$$T_c = \{(c_d, r_p, c_r) \mid c_d, c_r \in C, r_p \in R_o \cup R_d\} \quad (11)$$

$$T_o = \{(i_m, r_o, i_n) \mid i_m, i_n \in I, r_o \in R_o\} \quad (12)$$

$$T_d = \{(i_m, r_d, d_n) \mid i_m \in I, r_d \in R_d, d_n \in D\} \quad (13)$$

其中, r_s 表示 $rdfs:subClassOf$ 关系, r_t 表示 $rdf:type$ 关系, r_p 表示对象属性或数据属性关系, r_o 为对象属性, r_d 为数据属性. 因此, T_s 存储了 RDF 图谱 $\mathcal{G}$ 中所有的类之间层级语义声明三元组集合, T_t 为实例与类之间的类型定义语义声明三元组集合, T_c 为定义域、对象属性或数据属性、值域的语义关联三元组集合, T_o 中为实例、对象属性和实例的语义关联三元组集合, T_d 为实例、数据属性和字面量信息语义关联的三元组集合. 表 2 中描述了 HeLis 本体划分生成的各语义视图 RDF 三元组示例.

表 2 HeLis 多语义视图划分示例

语义视图名称	各视图中RDF三元组示例
类层级关系视图 T_s	$\langle helis:Beer, rdfs:subClassOf, helis:AlcoholicBeverages \rangle$
	$\langle helis:Eggs, rdfs:subClassOf, helis:BasicFood \rangle$
	$\langle helis:Meat, rdfs:subClassOf, helis:BasicFood \rangle$
类实例关系视图 T_t	$\langle helis:FOOD-4001, rdfs:type, helis:Beer \rangle$
	$\langle helis:AsparticAcid_800, rdfs:type, helis:AsparticAcid \rangle$
	$\langle helis:VitaminD_046, rdfs:type, helis:Nutrient \rangle$
类属性关系视图 T_c	$\langle helis:Beer, helis:hasNutrient, helis:AsparticAcid \rangle$
	$\langle helis:AsparticAcid, helis:amountOfNutrient, xsd:double \rangle$
	$\langle helis:ComposedFood, helis:contains, helis:Dressing \rangle$
实例对象属性视图 T_o	$\langle helis:FOOD-4001, helis:hasNutrient, helis:AsparticAcid_800 \rangle$
	$\langle helis:RECIPE-388-200, helis:hasRecipeFood, helis:RECIPEFOOD-FOOD-200 \rangle$
	$\langle helis:RECIPEFOOD-FOOD-1204, helis:hasFood, helis:FOOD-1204 \rangle$
实例数据属性视图 T_d	$\langle helis:AsparticAcid_800, helis:amountOfNutrient, "8.0" \rangle$
	$\langle helis:FOOD-1505, helis:amountEnergy, "833.0" \rangle$
	$\langle helis:Niacin_220, helis:unit, "mg" \rangle$

例如, 表 2 中示例的 $helis:FOOD-4001$ 的类型是 $helis:Beer$. 类属性关系包含了类对象属性关系与类数据属性关系, 例如表 2 中类属性关系视图类别第 1 个三元组是类对象属性, 第 2 个三元组是类数据属性. 实例对象属性是指头实体和尾实体是实例, 关系是对象属性. 而实例数据属性的关系是数据属性, 尾实体是字面量, 字面量可以是包括字符串、浮点型和整型等在内的多种数据类型. 在 5 个子视图中, 类层级关系视图与类实例关系视图仅包含一种关系, 而其余 3 种视图包含了对象属性和数据属性中的一类关系, 因此本文将前两个视图称为单关系视图, 将后 3 个视图称为多关系视图.

2.4 多语义视图感知的 PLMs 自监督进阶训练

为提升通用 PLMs 的多语义视图上下文特征编码能力, 需要在该场景下开展进阶训练并微调模型参数. 因此, 本文首先提出了面向多语义视图的序列化数据生成策略, 并在其基础上开展 PLMs 自监督进阶训练.

2.4.1 面向多语义视图的三元组数据序列化策略

第 1 阶段划分生成的多语义视图是由若干三元组构成的 RDF 图结构信息, 为使其适应于主流 PLMs 的文本序列化编码机制, 需将其按照特定的语义模式转换为序列化结构. 与 OWL2Vec*等方法在整个 RDF 图结构上随机游走生成文本序列不同, 本文面向多语义视图上下文特点, 采用单一视图内游走策略, 以实现在编码阶段加强模型的视图内语义上下文理解能力. 在三元组数据序列化转换过程中, 本文将三元组中的谓词也作为图节点, 根据不同视图的语义特点分别设计序列化策略.

具体而言, 针对类实例关系视图, 以实例或 $rdf:type$ 为原点, 经过预设的步数阈值抽取类实例节点序列. 针对类层级关系视图, 以类 (原子概念) 或 $rdfs:subClassOf$ 为原点, 抽取类层级关系节点序列. 由于类属性关系视图包含了对象属性关系和数据属性关系的定义域和值域语义声明, 本文以类、对象属性及数据属性为原点抽取类属性关系节点序列. 针对实例对象属性视图和实例数据属性视图, 分别以类和对象属性、类和数据属性为原点, 抽取其节点序列. 另外, 为进一步融合本体语义图谱中的文本标记信息, 本文方法采用实体和属性的 $rdfs:label$ 或 $rdfs:comment$ 等标注属性信息分别替换实体 URIs 和属性 URIs. 例如表 2 中的 $helis:FOOD-4001$ 实例实体可以使用其标注属性“Blonde Beer”将其替换. 当实体和属性中没有任何的文本标注属性时, 则采用其 URIs 中末尾部分所包含的文本信息进行代替, URIs 包含的其他非文本部分被丢弃. 例如, 假设表 2 中 $helis:AlcoholicBeverages$ 类实体不存在标注属性, 那么该类实体将会被表示成“Alcoholic Beverages”. 在最后生成的实体和属性的描述文本中, 所有的单词都将被转换成小写, 示例的对象属性和类将会分别变为“blonde beer”及“alcoholic beverages”. 表 3 中各语义视图类别对应的第 2 行示例了对应三元组生成的序列化数据.

表 3 HeLis 各语义视图中的三元组数据与序列化数据转换示例

语义视图类别	示例
类实例 关系视图	三元组: $\langle helis:FOOD-9030, rdf:type, helis:CoffeeAndSubstitutes \rangle$ 序列化数据: “american coffee in cups type coffee and substitutes” 进阶训练数据: “[CLS] american coffee in cups [SEP] type [SEP] coffee and substitutes [SEP]”
类层级 关系视图	三元组: $\langle helis:Beverages, rdfs:subClassOf, helis:BasicFood, rdfs:subClassOf, helis:Food \rangle$ 序列化数据: “berverages subclassof basic food subclassof food” 进阶训练数据: “[CLS] berverages [SEP] subclassof [SEP] basic food [SEP] subclassof [SEP] food [SEP]”
类属性 关系视图	三元组: $\langle helis:Meal, helis:hasConsumedFood, helis:ConsumedFood, helis:hasFood \rangle$ 序列化数据: “meal has consumed food consumed food has food food” 进阶训练数据: “[CLS] meal [SEP] has consumed food [SEP] consumed food [SEP] has food [SEP]”
实例对象 属性视图	三元组: $\langle helis:RECIPEFOOD-FOOD-419, helis:hasFood, helis:FOOD-419, helis:hasNutrient \rangle$ 序列化数据: “recipefood food 419 has food apples has nutrient protein 020” 进阶训练数据: “[CLS] recipefood food 419 [SEP] has food [SEP] apples [SEP] has nutrient [SEP]”
实例数据 属性视图	三元组: $\langle helis:ButyricCaproicCaprylicCapricAcid_000, helis:amountOfNutrient, “0.0” \rangle$ 序列化数据: “butyric caproic caprylic capric acid 000 amount of nutrient 0.0” 进阶训练数据: “[CLS] butyric caproic caprylic capric acid 000 [SEP] amount of nutrient [SEP] 0.0 [SEP]”

2.4.2 PLMs 自监督进阶训练

MSV-KRL 以基于 Transformer 编码器架构的 BERT 模型为基础编码器, 并对其开展基于 MLM 任务的无监督进阶训练. 首先, 本文将序列化数据中对应的实体或属性都通过 [SEP] 标记进行分割, 以便于模型在进阶训练过程中能够正确区分不同的实体或属性, 如公式 (14) 所示, 其中 n 表示序列中节点数量, m_n 表示第 n 个节点对应的标注属性字符数. 在语料库中包含了与 5 种视图对应的节点序列, 都将对其采用如上操作, 如表 3 中各类别第 3 行.

$$sequence = \{[CLS], x_1^1, \dots, x_{m_1}^1, [SEP], \dots, [SEP], x_1^n, \dots, x_{m_n}^n, [SEP]\} \quad (14)$$

随后, 对 $sequence$ 进行 Tokenization, 结果如公式 (15) 所示, 其中 $t_{p_n}^n$ 表示第 n 个节点对应的第 p_n 个描述文本 Token. 其次, MSV-KRL 以 0.15 的概率随机对公式 (15) 中 Token 进行掩码, 被掩码的 Token 在公式 (16) 中表示为 $t_{[MASK]}$. 最后, 将其输入至 BERT 模型让其自行预测掩码 Tokens, 以实现预训练模型的自监督进阶训练. MSV-KRL 采用了文献 [60] 的方法开展自监督进阶训练策略, 定义损失函数为交叉熵损失函数, 其真实标签如公式 (17) 所示, -100 表示不参与损失计算.

$$tokens = \{t_{[CLS]}, t_1^1, \dots, t_{p_1}^1, t_{[SEP]}, \dots, t_{[SEP]}, t_1^n, t_2^n, \dots, t_{p_n}^n, t_{[SEP]}\} \quad (15)$$

$$tokensMasked = \{t_{[CLS]}, t_1^1, t_2^1, \dots, t_{[MASK]}, t_{[SEP]}, \dots, t_{[SEP]}, t_1^n, t_{[MASK]}, \dots, t_{p_n}^n, t_{[SEP]}\} \quad (16)$$

$$labels = \{-100, -100, -100, \dots, t_{p_1}^1, -100, \dots, -100, -100, t_2^n, \dots, -100, -100\} \quad (17)$$

2.5 面向多语义视图的多任务联合表示学习

MSV-KRL 多任务联合表示学习以第 1 阶段生成的多语义视图三元组 $\langle s, p, o \rangle$ 为数据输入, 并使用经过进阶训练参数微调后的模型为编码器, 用于获取初始嵌入. 具体而言, 首先将 URIs 定义的各个三元组头实体、关系和尾实体用第 2.4.1 节中标注属性替换策略替换为文本序列, 并使用 [SEP] 标签对其进行分隔, 最后对其进行 Tokenization, 得到如公式 (18) 所示的三元组 Token 序列 X .

$$X = \{t_{[CLS]}, t_{x_1^1}, \dots, t_{x_i^1}, t_{[SEP]}, t_{x_1^2}, \dots, t_{x_j^2}, t_{[SEP]}, t_{x_1^3}, \dots, t_{x_k^3}, t_{[SEP]}\} \quad (18)$$

将 X 输入到进阶训练后的模型 PT_BERT 进行编码, 并输出所有 Encoder 层 (隐藏层) 的编码结果. 设 PT_BERT 中 Encoder 层数为 n , 嵌入维度为 d , 且其中某一层的编码结果为 V , 如公式 (19), 其中 $V \in \mathbb{R}^{(i+j+k+4) \times d}$, $v \in \mathbb{R}^{1 \times d}$, i, j, k 分别表示当前三元组头实体、关系和尾实体对应 Token 数量.

$$V = \{v_{[CLS]}, v_1^1, \dots, v_i^1, v_{[SEP]}, v_1^2, \dots, v_j^2, v_{[SEP]}, v_1^3, \dots, v_k^3, v_{[SEP]}\} \leftarrow PT_BERT(X) \quad (19)$$

然后, 依据公式 (20)–公式 (22) 以均值池化的方式, 分别获取所有编码层中头实体嵌入 h_s 、关系嵌入 h_p 和尾实体嵌入 h_o , 其中 $v_{a,b}^1$ 表示头实体在第 a 层 Encoder 中第 b 个 Token 对应的嵌入向量, $v_{a,b}^2$ 、 $v_{a,b}^3$ 同理.

$$h_s = (1/(n \times i)) \sum_{a=1}^n \sum_{b=1}^i v_{a,b}^1 \quad (20)$$

$$h_p = (1/(n \times j)) \sum_{a=1}^n \sum_{b=1}^j v_{a,b}^2 \quad (21)$$

$$h_o = (1/(n \times k)) \sum_{a=1}^n \sum_{b=1}^k v_{a,b}^3 \quad (22)$$

由于同一实体或关系可能存在于多个语义视图和多个三元组中, 例如, 表 3 中 $\langle helis : Beer, rdfs : subClassOf, helis : AlcoholicBeverages \rangle$ 、 $\langle helis : Beer, helis : hasNutrient, helis : AsparticAcid \rangle$ 同时包含了 $helis : Beer$ 实体, 因此本文依据公式 (23) 对在不同三元组中的相同实体的所有嵌入做均值池化, 得到嵌入结果 h'_s 、 h'_p 与 h'_o , 将该结果作为实体和属性的初始嵌入结果.

$$h'_q = (1/l) \sum_{k=1}^l h_q^k, q \in \{s, p, o\} \quad (23)$$

在公式 (23) 中, $h'_q \in \mathbb{R}^{1 \times d}$, l 表示包含当前实体或属性的三元组数量. 最后本文将包含所有实体的嵌入矩阵定义为 $H^e \in \mathbb{R}^{(|C|+|I|+|D|) \times d}$, 将包含所有关系的嵌入矩阵定义为 $H^r \in \mathbb{R}^{|R| \times d}$.

其次, 面向映射生成的 5 种语义视图, MSV-KRL 设计了相应的预测任务用于学习不同语义上下文中的实体和关系嵌入. 这 5 种预测任务分别是类实例关系预测任务 $\mathcal{T}_i$ 、类层级关系预测任务 $\mathcal{T}_s$ 、类属性关系预测任务 $\mathcal{T}_c$ 、实例对象属性关系预测任务 $\mathcal{T}_o$ 以及实例数据属性关系预测任务 $\mathcal{T}_d$. 所有任务统一采用多层感知机 (multi-layer perception, MLP) 作为分类器, 使用二分类交叉熵损失函数作为单个任务的损失函数, 如公式 (24) 所示.

$$\mathcal{L}_{BCE} = -(y^s \log(y^p) + (1 - y^s) \log(1 - y^p)) \quad (24)$$

其中, $\mathcal{L}_{BCE}$ 表示某一任务的损失值, y^p 表示模型预测值, y^s 表示真实标签. 同时, 在每一个分类器中的线性层后使用 ReLU 激活函数, 且在其最后使用 Sigmoid 函数将模型输出值映射到 0–1 之间.

针对任务 $\mathcal{T}_i$, 给定一个类实例关系三元组 (i, r_i, c) 作为正例, 其中 $i \in I$, $c \in C$, $r_i \in R$ 表示 $rdf : type$, 同时通过将尾实体 c 替换为 $c' \in C$ 且 $c' \neq c$ 构造负例三元组 (i, r_i, c') , 从而保证新生成的负例从未在数据集中出现过. 两种三元组的标签为 $y_i^s \in \{0, 1\}$, 其中正例三元组对应标签为 1, 负例三元组标签为 0, $\mathcal{T}_i$ 的任务目标就是让模型能够区分正确和错误的类实例关系三元组. 令正例三元组 (i, r_i, c) 中 3 个元素对应的嵌入分别为 h_i 、 h_{r_i} 与 h_c , 将其按序拼接在一起为一个向量作为模型输入, 如公式 (25) 所示, 其中 $h_i, h_c \in H^e$, $h_{r_i} \in H^r$, $x \in \mathbb{R}^{3d}$. 其次, 将向量 x 输入至分类器得到模型预测值概率 y_i^p , 如公式 (26) 所示, 其中 W_i 表示当前任务分类器的权重, b_i 则表示偏置量. 最后, 依据模型预测值 y_i^p 以及真实标签 y_i^s 计算二分类交叉熵损失函数值 $\mathcal{L}_{BCE}^i$, 如公式 (27) 所示.

$$x = [h_i, h_{r_i}, h_c] \quad (25)$$

$$y_i^p = \text{Sigmoid}(\text{ReLU}(W_i x + b_i)) \quad (26)$$

$$\mathcal{L}_{\text{BCE}}^s = -(y_i^s \log(y_i^p) + (1 - y_i^s) \log(1 - y_i^p)) \quad (27)$$

针对任务 $\mathcal{T}_s$, 给定一个类层级关系三元组 (c_1, r_s, c_2) 作为正例, 其中 $c_1, c_2 \in C, r_s \in R$ 表示 *rdfs: subclassOf*, 而负例构造方式同 $\mathcal{T}_r$, 构造结果为 (c_1, r_s, c_2') , 同时正负例的标签为 $y_s^s \in \{0, 1\}$, $\mathcal{T}_s$ 的任务目标就是让模型能够区分正确和错误的类层级关系三元组. 令正例三元组 (c_1, r_s, c_2) 中 3 个元素对应的嵌入分别为 $\mathbf{h}_{c_1}$ 、 $\mathbf{h}_{r_s}$ 与 $\mathbf{h}_{c_2}$, 将其按序拼接为一个向量作为模型输入, 如公式 (28) 所示, 其中 $\mathbf{h}_{c_1}, \mathbf{h}_{c_2} \in \mathbf{H}^e, \mathbf{h}_{r_s} \in \mathbf{H}^r, \mathbf{x} \in \mathbb{R}^{3d}$. 其次, 将拼接的输入向量 $\mathbf{x}$ 输入至分类器得到概率值 y_s^p , 如公式 (29) 所示, 其中 $\mathbf{W}_s$ 表示类层级关系分类器的权重, b_s 表示对应偏置量. 最后, 通过公式 (30) 计算得到类层级关系三元组的损失值 $\mathcal{L}_{\text{BCE}}^s$, 公式 (30) 中 y_s^s 表示类实例关系真实标签.

$$\mathbf{x} = [\mathbf{h}_{c_1}, \mathbf{h}_{r_s}, \mathbf{h}_{c_2}] \quad (28)$$

$$y_s^p = \text{Sigmoid}(\text{ReLU}(\mathbf{W}_s \mathbf{x} + b_s)) \quad (29)$$

$$\mathcal{L}_{\text{BCE}}^s = -(y_s^s \log(y_s^p) + (1 - y_s^s) \log(1 - y_s^p)) \quad (30)$$

针对任务 $\mathcal{T}_c$, 给定一个类属性关系三元组 (c_1, r_c, c_2) , 其中 $c_1, c_2 \in C, r_c \in (R_o \cup R_d) \subset R$, 负样本构造方式同 $\mathcal{T}_r$, 可以表示为 (c_1, r_c, c_2') , 同时正负例的标签为 $y_c^s \in \{0, 1\}$, $\mathcal{T}_c$ 的任务目标就是让模型能够区分正确和错误的类属性关系三元组. 令正例三元组 (c_1, r_c, c_2) 中三个元素对应的嵌入分别为 $\mathbf{h}_{c_1}$ 、 $\mathbf{h}_{r_c}$ 与 $\mathbf{h}_{c_2}$, 将其按序拼接在一起为一个向量作为模型输入, 如公式 (31) 所示, 其中 $\mathbf{h}_{c_1}, \mathbf{h}_{c_2} \in \mathbf{H}^e, \mathbf{h}_{r_c} \in \mathbf{H}^r, \mathbf{x} \in \mathbb{R}^{3d}$. 其次, 将拼接好的输入向量 $\mathbf{x}$ 输入至分类器得到概率值 y_c^p , 如公式 (32) 所示, 其中 $\mathbf{W}_c$ 表示类属性关系分类器的权重, b_c 表示对应偏置量. 最后, 通过公式 (33) 计算得到类属性关系三元组的损失值 $\mathcal{L}_{\text{BCE}}^c$, 公式 (33) 中 y_c^s 表示类属性关系真实标签.

$$\mathbf{x} = [\mathbf{h}_{c_1}, \mathbf{h}_{r_c}, \mathbf{h}_{c_2}] \quad (31)$$

$$y_c^p = \text{Sigmoid}(\text{ReLU}(\mathbf{W}_c \mathbf{x} + b_c)) \quad (32)$$

$$\mathcal{L}_{\text{BCE}}^c = -(y_c^s \log(y_c^p) + (1 - y_c^s) \log(1 - y_c^p)) \quad (33)$$

针对任务 $\mathcal{T}_o$, 给定一个实例对象属性关系三元组 (i_1, r_o, i_2) , 其中 $i_1, i_2 \in I, r_o \in R_o$, 实例对象属性负样本构造方式不同于前 3 个任务, $\mathcal{T}_o$ 从所有的实例中选择一个 i_1' 用来替换头实体 i_1 , i_1' 满足条件 $i_1' \in I$ 且 $i_1' \neq i_1$, 同时正负例的标签为 $y_o^s \in \{0, 1\}$, $\mathcal{T}_o$ 的任务目标就是让模型能够区分正确和错误的实例对象属性关系三元组. 令正例三元组 (i_1, r_o, i_2) 中 3 个元素对应的嵌入分别为 $\mathbf{h}_{i_1}$ 、 $\mathbf{h}_{r_o}$ 与 $\mathbf{h}_{i_2}$, 将其按序拼接在一起为一个向量作为模型输入, 如公式 (34) 所示, 其中 $\mathbf{h}_{i_1}, \mathbf{h}_{i_2} \in \mathbf{H}^e, \mathbf{h}_{r_o} \in \mathbf{H}^r, \mathbf{x} \in \mathbb{R}^{3d}$. 其次, 将拼接好的输入向量 $\mathbf{x}$ 输入至分类器得到概率值 y_o^p , 如公式 (35) 所示, 其中 $\mathbf{W}_o$ 表示实例对象属性关系分类器的权重, b_o 表示对应偏置量. 最后, 通过公式 (36) 计算得到实例对象属性关系三元组的损失值 $\mathcal{L}_{\text{BCE}}^o$, 公式 (36) 中 y_o^s 表示实例对象属性关系真实标签.

$$\mathbf{x} = [\mathbf{h}_{i_1}, \mathbf{h}_{r_o}, \mathbf{h}_{i_2}] \quad (34)$$

$$y_o^p = \text{Sigmoid}(\text{ReLU}(\mathbf{W}_o \mathbf{x} + b_o)) \quad (35)$$

$$\mathcal{L}_{\text{BCE}}^o = -(y_o^s \log(y_o^p) + (1 - y_o^s) \log(1 - y_o^p)) \quad (36)$$

针对任务 $\mathcal{T}_d$, 给定一个实例数据属性关系三元组 (i, r_d, d) , 其中 $i \in I, d \in D, r_d \in R_d$, 实例数据属性负样本构造方式同 $\mathcal{T}_o$, $\mathcal{T}_d$ 从所有的实例中选择一个 i' 用来替换头实体 i , i' 满足条件 $i' \in I$ 且 $i' \neq i$, 同时正负例的标签为 $y_d^s \in \{0, 1\}$, $\mathcal{T}_d$ 的任务目标就是让模型能够区分正确和错误的实例数据属性关系三元组. 令正例三元组 (i, r_d, d) 中 3 个元素对应的嵌入分别为 $\mathbf{h}_i$ 、 $\mathbf{h}_{r_d}$ 与 $\mathbf{h}_d$, 将其按序拼接在一起为一个向量作为模型输入, 如公式 (37) 所示, 其中 $\mathbf{h}_i, \mathbf{h}_d \in \mathbf{H}^e, \mathbf{h}_{r_d} \in \mathbf{H}^r, \mathbf{x} \in \mathbb{R}^{3d}$. 其次, 将拼接好的输入向量 $\mathbf{x}$ 输入至分类器得到概率值 y_d^p , 如公式 (38) 所示, 其中 $\mathbf{W}_d$ 表示实例数据属性关系分类器的权重, b_d 表示对应偏置量. 最后, 通过公式 (39) 计算得到实例数据属性关系三元组的损失值 $\mathcal{L}_{\text{BCE}}^d$, 公式 (39) 中 y_d^s 表示实例数据属性关系真实标签.

$$\mathbf{x} = [\mathbf{h}_i, \mathbf{h}_{r_d}, \mathbf{h}_d] \quad (37)$$

$$y_d^p = \text{Sigmoid}(\text{ReLU}(\mathbf{W}_d \mathbf{x} + b_d)) \quad (38)$$

$$\mathcal{L}_{\text{BCE}}^d = -(y_d^s \log(y_d^p) + (1 - y_d^s) \log(1 - y_d^p)) \quad (39)$$

联合损失函数采用加权相加的方式计算多任务联合损失值, 如公式 (40) 所示, 其中 $\mathcal{L}_{\text{joint}}$ 表示多任务联合损失值, w_t 、 w_s 、 w_c 、 w_d 以及 w_o 分别表示任务 $\mathcal{T}_t$ 、 $\mathcal{T}_s$ 、 $\mathcal{T}_c$ 、 $\mathcal{T}_d$ 以及 $\mathcal{T}_o$ 对应损失函数的权重值.

$$\mathcal{L}_{\text{joint}} = w_t \mathcal{L}_{\text{BCE}}^t + w_s \mathcal{L}_{\text{BCE}}^s + w_c \mathcal{L}_{\text{BCE}}^c + w_o \mathcal{L}_{\text{BCE}}^o + w_d \mathcal{L}_{\text{BCE}}^d \quad (40)$$

公式 (40) 中的各个任务对应权重值通过 SLAW (scaled loss approximate weighting) 算法^[61]依据模型训练过程中各任务损失值变化自动进行更新, 如公式 (41)–公式 (44) 所示.

$$a_i \leftarrow \beta a_i + (1 - \beta) \mathcal{L}_i^2 \quad (41)$$

$$b_i \leftarrow \beta b_i + (1 - \beta) \mathcal{L}_i \quad (42)$$

$$s_i \leftarrow \sqrt{a_i - b_i^2} \quad (43)$$

$$w_i = (n/s_i) / \sum_{j=1}^n (1/s_j) \quad (44)$$

其中, β 为超参数, 在模型开始训练前定义, $\mathcal{L}_i$ 表示第 i 个任务损失值, a_i 与 b_i 的值在模型开始训练前自定义值进行初始化, $n = 5$ 表示任务数量, s_i 与 s_j 分别表示第 i 个和第 j 个任务损失值的估计标准差.

3 实验分析

3.1 数据集与实验设置

3.1.1 数据集

参照 OWL2Vec^[26] 的实验数据集选择, 本文首先在 HeLis^[14]、FoodOn^[15]以及 GO^[16]这 3 个 OWL 语义图谱上针对类实例预测、类层级关系预测和类属性预测等多个任务展开对比实验. 表 4 描述了实验相关数据集的统计信息.

表 4 HeLis、FoodOn、GO、IB2M 及 DBpedia-20k 相关统计信息

OWL图谱	实例	类	实例对象属性	实例数据属性	公理	类实例	类层级关系
HeLis	20318	277	83210	39808	172213	20318	261
FoodOn	359	28182	0	0	241581	0	29778
GO	0	44244	0	0	513306	0	72601
DBpedia-20k	19996	1227	0	0	71692	20001	813
IB2M	22427	245	36466	14739	120346	22435	348

注: “实例对象属性”表示实例间对象属性声明的数量, “实例数据属性”表示实例间数据属性声明的数量, 其余各列表示其本身在 OWL 图谱中的数量

为了进一步验证本文所提方法面向不同领域知识图谱的表示学习性能, 本文在 DBpedia 和 IB2M 两个 OWL 语义图谱上展开实验. 其中, DBpedia^[13]来自于 Wikipedia, 包含了大量通用世界知识. DBpedia 有超过 40 万个实例, 也包含了 OWL 概念层语义定义. 本文从 DBpedia 实例层中随机抽取 20000 实例, 将其与概念层进行融合, 合并为一个完整的 OWL 图谱, 在后续实验中将该 OWL 图谱称为 DBpedia-20k. IB2M (intelligent bridge management and maintenance)^[62]是在 BSHM (bridge structure and health monitoring)^[63] OWL 语义基础上构建的桥梁智能管养领域知识图谱, 表征了桥梁结构体系、材料属性、健康监测、结构检测和养护等多维度语义信息. IB2M 图谱也被应用于知识图谱问答任务^[4]. 图 2 描述了 IB2M 知识图谱中的部分概念、实例及其语义关系.

从表 4 中各 OWL 图谱的各项指标参数可以发现, HeLis 和 IB2M 中包含了 5 种预测任务所需要的全部公理信息. 而 FoodOn 和 GO 中对应的类实例关系、实例对象属性关系与实例数据属性关系数量为 0, 表示只包含了两种任务公理信息. 同时, DBpedia-20k 的概念层和实例层分别属于两个文件, 且实例文件中仅包含了实例的类型信息, 尚未定义实例对象属性关系和实例数据属性关系. 综合分析其统计结果, 上述 5 种数据集反映了不同领域 OWL 语义图谱的差异化上下文场景, 可为本文提出方法在不同 OWL 语义知识表示学习性能评估提供支持. 本文实验所采用的上述数据集及相关代码可在下列地址获取: <https://github.com/Institute-of-BDKE-CQJTU/MSV-KRL>.

表示完整数据集, $\mathcal{D}_{\text{pos}}$ 表示正例三元组集合, 若该三元组为类实例、类层级或类属性关系, 则利用公式 (47) 生成负例, 否则利用公式 (48) 生成负例, 其中 $\mathcal{D}_{\text{neg}}$ 表示负例三元组集合.

$$\mathcal{D}_{\text{neg}} = \{(s, p, o') \mid o' \in C, (s, p, o') \notin \mathcal{D}\} \quad (47)$$

$$\mathcal{D}_{\text{neg}} = \{(s', p, o) \mid s' \in I, (s', p, o) \notin \mathcal{D}\} \quad (48)$$

在模型测试阶段, 需要为每一个测试三元组构建候选实体集, 若三元组为类实例、类层级或类属性关系, 与负例构造方法类似, 以整个概念集合 C 为候选实体集替换其尾实体, 否则以实例集合 I 为候选实体集替换头实体, 但需要保证新生成的三元组不能是训练集和验证集中的已有三元组, 即 TransE^[19] 和 OWL2Vec* 中采用的过滤设置 (filter settings). MSV-KRL 将对由候选实体集替换后生成的三元组集合进行打分并进行排序, 依据正确实体的位次计算 $H@k$ 以及 MRR .

在多语义视图划分阶段, 本文参照 OWL2Vec*^[26] 方法中所采用的最优设置, 将随机游走深度设置为 4. 在语义感知进阶训练阶段, 各项超参数参照 BERT^[60] 论文中进行预训练的最优参数设置, 统一设置如下: 学习率为 10^{-5} 、批次大小为 128、嵌入维度大小为 768, 在模型损失值不再出现明显下降时停止训练. 在多任务联合表示学习阶段, 经过多次预实验对比分析, 得到如下最优参数: 学习率为 10^{-3} 、批次大小为 512、训练轮数为 60 轮. 针对 MLP 分类器的隐藏层大小, 由于 DBpedia-20k 和 IB2M 两个本体较为复杂, 对模型参数量的要求更大, MSV-KRL 将其设置为 400, 其余 3 个本体统一设置为 200. 最后, 将公式 (41) 和公式 (42) 中的超参数 β 设置为 0.999, 与 SLAW 算法^[61]中原始值一致.

3.2 整体实验结果

3.2.1 类实例预测任务

HeLis 类实例预测任务的整体实验结果如表 5 中所示. 实验结果表明, MSV-KRL 在所有指标上都要优于对比的基线方法, 且与 OWL2Vec* 等 OWL 图谱表示学习方法相比, MSV-KRL 各项指标都有较大幅度提升, 证明面向多语义视图的多任务联合学习方法能够有效捕捉概念层和实例层之间的语义关联, 提升知识嵌入学习能力. 此外, MSV-KRL 在各项指标上也能远超 TransE 等传统知识图谱表示学习方法, 这是由于这类方法需要通过多种不同关系对实体进行建模, 而在当前任务中仅基于类实例关系展开训练测试, 造成这类方法无法有效区分不同的类和实例. Box²EL 能够有效建模 OWL 图谱中概念层描述逻辑公理信息, 但无法处理 OWL 图谱中实例层信息, 因此该方法在当前任务中的性能有待提升. DGS 能够通过概念空间与实例空间的融合建模实例的类型, 但该方法无法仅通过几何空间有效捕捉公理信息, 因此该方法虽在类实例预测任务上要优于 Box²EL, 但仍值得优化.

表 5 HeLis 类实例预测任务实验结果

方法	MRR	$H@1$	$H@5$	$H@10$
TransE ^[19]	0.181	0.090	0.232	0.355
DistMult ^[31]	0.253	0.166	0.304	0.437
TransR ^[28]	0.298	0.184	0.391	0.559
Onto2Vec ^[56]	0.211	0.108	0.268	0.397
OPA2Vec ^[57]	0.237	0.146	0.286	0.408
OWL2Vec v1 ^[59]	0.335	0.215	0.397	0.601
RDF2Vec ^[64]	0.345	0.219	0.460	0.655
Transformer (all text)	0.599	0.390	0.870	0.912
Transformer (label)	0.657	0.515	0.824	0.897
Pre-trained Word2Vec	0.899	0.877	0.923	0.933
OWL2Vec* ^[26]	0.953	0.932	0.978	0.987
DGS ^[51]	0.101	0.021	0.119	0.267
Box ² EL ^[23]	0.033	0.014	0.032	0.040
MSV-KRL	0.969	0.951	0.992	0.997

3.2.2 类层级关系预测任务

面向 HeLis、FoodOn 和 GO 的类层级关系预测实验结果如表 6 和表 7 所示. 从整体结果来看, MSV-KRL 在 3 个 OWL 图谱上的绝大部分实验结果均远超其他基线方法, 且在 FoodOn、GO 和 HeLis 上的 *Hits@10* 指标相比次优方法分别提升约 23%、18% 和 36%. 此外, 与 OWL2Vec*等知识表示学习方法相比, MSV-KRL 各项指标均有较大提升. 例如, 与 OWL2Vec*相比, HeLis 在 *Hits@1* 指标上的性能大幅提升, 证明本文所提面向多语义视图的三元组数据序列化策略能够很好地捕捉类层级关系信息.

表 6 FoodOn 及 GO 类层级预测任务实验结果

方法	FoodOn				GO			
	<i>MRR</i>	<i>H@1</i>	<i>H@5</i>	<i>H@10</i>	<i>MRR</i>	<i>H@1</i>	<i>H@5</i>	<i>H@10</i>
TransE ^[19]	0.029	0.011	0.044	0.065	0.015	0.005	0.018	0.030
DistMult ^[31]	0.076	0.045	0.099	0.134	0.046	0.018	0.680	0.097
TransR ^[28]	0.072	0.044	0.093	0.13	0.048	0.016	0.076	0.113
EL_Embedding ^[54]	0.040	0.014	0.067	0.099	0.018	0.005	0.021	0.036
Onto2Vec ^[56]	0.034	0.014	0.047	0.064	0.024	0.008	0.031	0.053
OPA2Vec ^[57]	0.093	0.058	0.117	0.156	0.075	0.032	0.106	0.157
OWL2Vec v1 ^[59]	0.091	0.052	0.121	0.152	0.031	0.012	0.040	0.067
RDF2Vec ^[64]	0.078	0.053	0.097	0.119	0.043	0.017	0.057	0.087
Transformer (all text)	0.022	0.011	0.032	0.050	0.014	0.008	0.017	0.019
Transformer (label)	0.016	0.005	0.027	0.046	0.009	0.001	0.009	0.018
Pre-trained Word2Vec	0.136	0.089	0.175	0.227	0.123	0.055	0.177	0.260
OWL2Vec* ^[26]	0.213	0.143	0.287	0.357	0.170	0.076	0.258	0.376
Box ² EL ^[23]	0.021	0.003	0.027	0.051	0.066	0.023	0.094	0.158
InME/CoME ^[47]	0.306	0.205	0.419	0.501	0.253	0.148	0.362	0.478
MSV-KRL	0.345	0.212	0.492	0.618	0.277	0.146	0.412	0.565

表 7 HeLis 类层级关系预测任务实验结果

方法	<i>MRR</i>	<i>H@1</i>	<i>H@5</i>	<i>H@10</i>
Onto2Vec ^[56]	0.193	0.073	0.291	0.418
OPA2Vec ^[57]	0.208	0.073	0.345	0.491
RDF2Vec ^[64]	0.336	0.236	0.436	0.600
OWL2Vec* ^[26]	0.236	0.127	0.309	0.455
DGS ^[51]	0.013	0.000	0.000	0.000
Box ² EL ^[23]	0.150	0.073	0.182	0.309
MSV-KRL	0.509	0.364	0.709	0.818

从具体实验结果来看, InME/CoME 方法虽能利用类层级关系文本信息的全局和局部特征完成类层级关系的建模, 但没有较好地处理复杂公理信息, 而 MSV-KRL 能够通过本体映射处理复杂公理信息, 因此在 FoodOn 和 GO 上的整体表现要优于 InME/CoME. Box²EL 能够处理 OWL 图谱中概念层描述逻辑公理信息, 但在建模层级关系信息能力上较弱, 因此在该任务中表现有待提升. RDF2Vec 仅在类层级关系数据集上训练测试, 仅能学习到概念层中的局部信息, 导致学习到的嵌入表示能力有限, 因此在该任务上的表现与 MSV-KRL 有一定差距. 而对 DGS 而言, 该方法将概念的类层级关系、类实例关系都建模为“is_a”关系, 造成其难以将类层级关系与类实例关系区分开, 因此在该任务上的性能表现不佳.

3.2.3 类属性预测任务

类属性预测任务在 3 个 OWL 图谱上的实验结果如表 8 所示, 由于 DGS 方法需要实例层数据, 因此无法在 FoodOn 和 GO 上展开测试. 从实验结果可知, 本文所提 MSV-KRL 方法能够较为准确地预测类属性信息, 仅在

FoodOn 上略弱于 Box²EL 方法. 然而, 当 Box²EL 遇到更加复杂的 OWL 图谱如 GO 时, 却难以充分捕获更加复杂的概念之间的公理信息. DGS 方法虽无法处理类属性关系, 本文仍将其作为概念层部分同实例部分一起进行训练测试, 通过结果可以发现, 该方法能够通过双曲空间学习到多关系信息. 但是, 由于 DGS 无法使用文本信息, 也没有深度利用前面所述公理信息, 因此其性能相比于 Box²EL 和 MSV-KRL 仍有待提升. RDF2Vec 方法在单独处理简单 OWL 图谱时, 效果要优于 Onto2Vec 和 OPA2Vec, 但难以处理 GO 等复杂图谱. 同时, 由于 OPA2Vec 加入了文本信息, 因此其表现整体要好于 Onto2Vec. 而 OWL2Vec*在上述几种方法基础上能够充分捕获多种公理信息, 所以其效果整体上要优于其他 3 种方法, 但 OWL2Vec*仍不能有效处理所有类属性信息.

表 8 HeLis、FoodOn 及 GO 类属性预测任务实验结果

方法	HeLis				FoodOn				GO			
	MRR	H@1	H@5	H@10	MRR	H@1	H@5	H@10	MRR	H@1	H@5	H@10
Onto2Vec ^[56]	0.145	0.098	0.166	0.217	0.044	0.014	0.061	0.101	0.011	0.003	0.017	0.025
OPA2Vec ^[57]	0.147	0.111	0.167	0.195	0.046	0.013	0.064	0.112	0.034	0.013	0.045	0.072
RDF2Vec ^[64]	0.218	0.083	0.333	0.500	0.069	0.032	0.096	0.148	0.014	0.005	0.019	0.031
OWL2Vec* ^[26]	0.139	0.083	0.167	0.250	0.088	0.039	0.121	0.185	0.021	0.007	0.028	0.040
DGS ^[51]	0.083	0.250	0.250	0.250	—	—	—	—	—	—	—	—
Box ² EL ^[23]	0.272	0.083	0.417	0.667	0.437	0.398	0.466	0.501	0.363	0.282	0.451	0.499
MSV-KRL	0.396	0.333	0.500	0.583	0.354	0.256	0.465	0.548	0.421	0.301	0.554	0.657

3.2.4 实例对象属性预测任务

HeLis 实例对象属性预测任务实验结果如表 9 所示. 从实验结果可知, MSV-KRL 性能表现远超所有基线方法, 在 MRR 指标上相比次优方法 DGS 提升了约 465.9%, 证明 MSV-KRL 相比于 DGS 等方法更加擅长处理实例层信息. 此外, DGS 方法的表现整体上要优于其他基线方法, 其原因是 DGS 训练时联合 OWL 图谱概念层与实例层信息, 能够充分利用实例的类型信息. 虽然本文将整个 OWL 图谱用于训练 Box²EL, 并使用实例对象属性三元组测试得到了结果, 但可以发现 Box²EL 仅适用于学习 OWL 图谱概念层公理信息. 在其余 4 种基线方法中 OWL2Vec*取得的效果最好, 其原因是 OWL2Vec*能够较好地处理类实例关系和类层级关系. 而 Onto2Vec 和 OPA2Vec 由于利用了 OWL 图谱中的所有信息, 因此性能表现要优于仅使用当前任务数据集的 RDF2Vec 方法.

表 9 HeLis 实例对象属性预测任务实验结果

方法	MRR	H@1	H@5	H@10
Onto2Vec ^[56]	0.027	0.006	0.027	0.057
OPA2Vec ^[57]	0.027	0.005	0.028	0.057
RDF2Vec ^[64]	0.023	0.004	0.023	0.045
OWL2Vec* ^[26]	0.029	0.007	0.029	0.056
DGS ^[51]	0.047	0.000	0.099	0.206
Box ² EL ^[23]	0.004	0.002	0.004	0.005
MSV-KRL	0.266	0.206	0.316	0.377

3.2.5 实例数据属性预测任务

HeLis 实例数据属性预测任务实验结果如表 10 所示, 得益于 MSV-KRL 能够充分利用 OWL 图谱中的标注属性信息与类数据属性信息, 同时利用面向多语义视图的多任务联合表示学习加强两者之间的联系, 其在所有指标上均取得了最佳性能. 除 MSV-KRL 方法外, 其他方法均在该任务上表现较差, 具体分析而言, Box²EL 难以处理 ABox 部分信息, 性能表现有待提升. 此外, 数据属性三元组中尾实体多是由文本或数字构成的字面量, OWL2Vec*、RDF2Vec 和 OPA2Vec 方法虽然能够处理这类文本信息, 但却都无法与字面量的类型信息建立联系, 造成其在该任务上的表现仍然欠佳.

表 10 HeLis 实例数据属性预测任务实验结果

方法	<i>MRR</i>	<i>H@1</i>	<i>H@5</i>	<i>H@10</i>
Onto2Vec ^[56]	0.001	0.000	0.000	0.001
OPA2Vec ^[57]	0.002	0.000	0.001	0.002
RDF2Vec ^[64]	0.003	0.000	0.002	0.003
OWL2Vec* ^[26]	0.005	0.001	0.003	0.006
DGS ^[51]	0.002	0.000	0.003	0.005
Box ² EL ^[23]	0.001	0.000	0.000	0.000
MSV-KRL	0.598	0.545	0.653	0.700

3.3 验证性实验结果

3.3.1 改进本体映射方法验证

MSV-KRL 基于 OWL2Vec*中所使用的本体映射方法进行改进,因此在本节中对其改进前后的效果展开实验对比,本体映射方法验证实验结果如表 11 所示.与 OWL2Vec*相比,本文所提多语义视图划分策略在 HeLis、FoodOn 和 GO 本体上均取得最佳性能,且在公理信息较少的 FoodOn 和 GO 本体上差异尤为明显.在具体任务方面,从 HeLis 各项任务的实验结果可以发现,类实例预测和实例对象数据属性预测不同映射方法的差距较小,而类层级关系预测、类属性预测和实例数据属性预测由于抽取信息的缺失性能差距较大.因此,经过综合分析可知,本文所增加的各项语义信息能够有效弥补原本本体映射方法信息抽取不足这一问题.

表 11 改进本体映射方法验证实验结果

数据集	任务	本体映射方法	<i>MRR</i>	<i>H@1</i>	<i>H@5</i>	<i>H@10</i>
HeLis	类实例预测	MSV-KRL	0.969	0.951	0.992	0.997
		OWL2Vec*	0.942	0.915	0.976	0.991
	类层级关系预测	MSV-KRL	0.509	0.364	0.709	0.818
		OWL2Vec*	0.423	0.327	0.527	0.745
	类属性预测	MSV-KRL	0.396	0.333	0.500	0.583
		OWL2Vec*	0.290	0.200	0.300	0.500
	实例对象属性预测	MSV-KRL	0.266	0.206	0.316	0.377
		OWL2Vec*	0.202	0.144	0.255	0.303
	实例数据属性预测	MSV-KRL	0.598	0.545	0.653	0.700
		OWL2Vec*	0.508	0.460	0.551	0.598
FoodOn	类层级关系预测	MSV-KRL	0.345	0.212	0.492	0.618
		OWL2Vec*	0.058	0.018	0.081	0.126
	类属性预测	MSV-KRL	0.354	0.256	0.465	0.548
		OWL2Vec*	0.067	0.029	0.089	0.131
GO	类层级关系预测	MSV-KRL	0.277	0.146	0.412	0.565
		OWL2Vec*	0.153	0.065	0.219	0.340
	类属性预测	MSV-KRL	0.421	0.301	0.554	0.657
		OWL2Vec*	0.297	0.209	0.389	0.468

3.3.2 消融实验

为了验证 MSV-KRL 各个模块的有效性,本文开展了消融实验,结果如表 12、表 13 所示.

表 12 HeLis 消融实验结果

任务	不同模块	<i>MRR</i>	<i>H@1</i>	<i>H@5</i>	<i>H@10</i>
类实例预测	MSV-KRL	0.969	0.951	0.992	0.997
	w/o 多视图语义划分	0.910	0.863	0.962	0.980
	w/o 自监督进阶训练	0.949	0.919	0.991	0.995
	w/o 多任务联合表示学习	0.946	0.915	0.984	0.994

表 12 HeLis 消融实验结果 (续)

任务	不同模块	MRR	H@1	H@5	H@10
类层级关系预测	MSV-KRL	0.509	0.364	0.709	0.818
	w/o 多视图语义划分	0.460	0.363	0.600	0.745
	w/o 自监督进阶训练	0.394	0.263	0.526	0.737
	w/o 多任务联合表示学习	0.203	0.091	0.273	0.509
类属性预测	MSV-KRL	0.396	0.333	0.500	0.583
	w/o 多视图语义划分	0.257	0.200	0.300	0.400
	w/o 自监督进阶训练	0.201	0.219	0.300	0.500
	w/o 多任务联合表示学习	0.309	0.200	0.400	0.500
实例对象属性预测	MSV-KRL	0.266	0.206	0.316	0.377
	w/o 多视图语义划分	0.195	0.143	0.233	0.291
	w/o 自监督进阶训练	0.177	0.119	0.216	0.286
	w/o 多任务联合表示学习	0.201	0.165	0.221	0.269
实例数据属性预测	MSV-KRL	0.598	0.545	0.653	0.700
	w/o 多视图语义划分	0.487	0.447	0.522	0.570
	w/o 自监督进阶训练	0.469	0.429	0.499	0.544
	w/o 多任务联合表示学习	0.468	0.433	0.495	0.531

表 13 FoodOn 和 GO 消融实验结果

数据集	任务	不同模块	MRR	H@1	H@5	H@10
FoodOn	类层级关系预测	MSV-KRL	0.345	0.212	0.492	0.618
		w/o 多视图语义划分	0.303	0.179	0.434	0.569
		w/o 自监督进阶训练	0.218	0.113	0.314	0.438
		w/o 多任务联合表示学习	0.274	0.156	0.396	0.522
	类属性预测	MSV-KRL	0.354	0.256	0.465	0.548
		w/o 多视图语义划分	0.323	0.223	0.426	0.520
		w/o 自监督进阶训练	0.219	0.133	0.301	0.392
		w/o 多任务联合表示学习	0.341	0.243	0.444	0.542
GO	类层级关系预测	MSV-KRL	0.277	0.146	0.412	0.565
		w/o 多视图语义划分	0.262	0.129	0.396	0.553
		w/o 自监督进阶训练	0.209	0.096	0.318	0.457
		w/o 多任务联合表示学习	0.243	0.118	0.368	0.518
	类属性预测	MSV-KRL	0.421	0.301	0.554	0.657
		w/o 多视图语义划分	0.406	0.288	0.537	0.628
		w/o 自监督进阶训练	0.375	0.270	0.483	0.581
		w/o 多任务联合表示学习	0.333	0.223	0.443	0.549

从表 12 和表 13 中 3 个 OWL 图谱中各个模块的消融实验结果整体来看, 各个模块均有助于 MSV-KRL 性能提升. 具体而言, 当去掉多视图语义划分模块后, HeLis、FoodOn 以及 GO 本体中 9 个任务的结果都有所下降. 首先, HeLis 中类实例预测以及类层级关系预测虽然下降幅度较小, 但仍表明多视图语义划分能够有效提升 MSV-KRL 整体性能. 其次, 类属性预测实验结果和 MSV-KRL 相比性能差距明显, 同时实例对象属性预测和实例数据属性预测也有较大提升, 其原因是经过多视图语义划分后 MSV-KRL 能够对各类关系进行针对性的表示学习, 有效降低噪声. 而 FoodOn 和 GO 中共 4 个任务中 GO 的类层级预测和类属性预测性能下降幅度相比于 FoodOn 的较小, 其原因是 GO 本体的比较复杂, 因此 GO 本体中这个任务的性能差异较小, 而 FoodOn 较大. 当去掉自监督进阶训练模块后, HeLis 中类实例预测任务差异较小, 其原因是 HeLis 中类与实例的标注属性内容自身已非常相似, 因此性

能差异较小, 而其余 4 个任务性能差异明显, 其原因是经过自监督进阶训练后能够使得获取的初始嵌入能够包含更多的语义信息. 针对 FoodOn 与 GO 中的 4 个任务而言, 性能差异都非常明显, 表明自监督进阶训练能够有效处理更加复杂的语义信息. 当去掉多任务联合表示学习模块后, HeLis 的类实例关系预测和 FoodOn 中的类属性能差异较小, 其原因也是 HeLis 中类和实例的标注属性非常相似, FoodOn 类属性预测性能差距较小则也是由于各类之间实体标注属性信息过于相似导致. 而 HeLis、FoodOn 和 GO 中除前述两个任务外的任务性能差异都较为明显, 其原因是 HeLis 同时包含概念层和实例层的信息, 因此 HeLis 的性能提升幅度最为明显, 表明多任务联合表示学习模块能够有效实现不同语义视图信息的相互补充, 有助于提升 MSV-KRL 整体性能.

针对类层级关系预测任务而言, 当去掉多视图语义划分模块后, 导致在 HeLis 上指标略微下降, 说明该模块能捕捉层级关系细节, 但影响程度较小. 然而当去掉多视图语义划分模块以及自监督进阶训练模块后, FoodOn 和 GO 各项评价指标如 *MRR* 等下降明显, 说明这两个模块能够捕捉层级关系的细节并丰富初始嵌入表示. 同时, 自监督进阶训练模块对 HeLis 的影响程度也较大. 当在 3 个本体上都多任务表示学习模块去掉后, *MRR* 指标和 *Hits@1* 指标在 FoodOn 和 GO 上都出现了明显下降, 在 HeLis 本体上大幅下降, 表明该模块在类层级关系任务中以及 MSV-KRL 方法中在任务间语义互补上有具有重要作用. 针对类属性关系预测任务而言, 当去掉多视图语义划分模块以及自监督进阶训练模块后, HeLis 和 FoodOn 上的 *MRR* 和 *Hits@1* 指标出现大幅下降, 而在 GO 本体中去掉自监督进阶训练模块以及多任务联合表示学习模块后 *MRR* 指标出现大幅下降, 说明当本体更加复杂时多任务联合表示学习以及自监督进阶训练能够更加有效地让 MSV-KRL 捕捉语义关系细节并丰富嵌入表示.

针对类实例预测任务而言, 当 HeLis 去掉自监督进阶训练与多任务联合表示学习后 *MRR* 指标下降幅度较小, 而当去掉多语义视图划分后该指标就出现大幅下降, 说明在处理类与实例之间的关系时概念层和实例层语义信息的单独处理是有效且重要的. 针对实例对象属性而言, 当 HeLis 去掉自监督进阶训练模块之后 *MRR* 指标和 *Hits@1* 指标出现大幅下降, 说明对当前任务而言通过随机游走获取节点序列并进行自监督进阶训练可以有效捕获上下文语义信息, 从而增强嵌入结果的语义表示能力. 针对实例数据属性而言, 当 HeLis 去掉 3 个模块后的性能下降幅度相当, 说明 3 个模块很对当前任务而言对 MSV-KRL 同等重要, 但三者的 *MRR* 等指标下降幅度都较大. 综合来看, 多视图语义划分模块能够有效捕捉不同视图下的复杂关系, 提高各任务的预测准确性. 而自监督进阶训练模块通过引入更多的语义信息, 显著提升了模型的初始嵌入表示能力, 尤其是在处理复杂语义信息时效果显著. 多任务联合表示学习模块通过任务间的语义互补, 进一步提高了整体表示学习能力, 对各任务的性能提升均有显著贡献.

为了综合对比 MSV-KRL 各模块对整体框架的性能提升, 本文依据 *MRR* 指标对整体框架的性能提升幅度展开数据可视化分析, 结果如后文图 3 所示. 从图中可以发现, 在 FoodOn 中的类层级关系预测和类属性预测任务、GO 的类层级关系预测任务以及 HeLis 的类属性预测任务和实例对象属性预测任务上的 *MRR* 指标在各项消融实验中性能提升最高, 占 9 个任务中的 5 个. 通过面向多语义视图的多任务联合表示学习后, 对 MSV-KRL 性能提升最大的任务数量为 3 个, 占 9 个任务的 1/3. 此外, 针对多视图语义划分模块, HeLis 中的类层级关系预测和类实例预测、FoodOn 中的类属性关系预测以及 GO 中的类层级关系预测和类属性预测任务, 在 3 个模块中的性能提升幅度最低, 占 9 个任务中的 5 个. 综上, 可以得知 MSV-KRL 可以在本体映射后通过概念-属性-实例多视图语义划分有效抽取 OWL 图谱中的多种语义信息与公理信息, 进而提升语义视图感知的 PLMs 自监督进阶训练效果, 最后通过面向多语义视图的多任务联合表示学习显著提升整体表示学习能力.

3.3.3 多种语言模型适用性验证

为分析本文所提方法在模型选择方面的适用性, 参照 Shu 等人^[42]和 Yao 等人^[43]的 LLMs 实验设置以及 BERTSubs^[25]中提到的 PLMs, 本文选择 BiomedBERT^[65]、RoBERTa^[66]两个预训练语言模型, 以及 LLaMA-2-7B^[67]及 ChatGLM3-6B^[68]两个大语言模型开展相应实验. 其中, 对于 BiomedBERT 和 RoBERTa 两个轻量级预训练语言模型, 本文对其进行自监督进阶训练后进行任务预测, 对于 LLaMA-2-7B 和 ChatGLM3-6B 两个大语言模型, 本文直接使用其特征表示. 最终实验结果如表 14 和表 15 所示.

在轻量级预训练语言模型方面, 实验结果表明在多数任务上 BiomedBERT 和 RoBERTa 均能取得较为优异的性能表现. 此外, 由于 BiomedBERT 使用了大量生物医疗领域数据集进行训练, 其在医疗健康领域本体 HeLis 以及 GO 图谱上更是取得了最佳性能. 在大语言模型方面, 得益于它们所拥有的巨大参数量, 使得大语言模型能够通过 在超大规模语料库学习得到更多的知识, 因此即使在本文中未对其进行自监督进阶训练, 它们仍能够在各 OWL 图谱中的多个任务上表现出不俗的性能. 综上, 本文所提 MSV-KRL 方法可适应于多种不同形式的预训练语言模型, 证明所提方法能够适配于多种不同的 PLMs 或 LLMs.

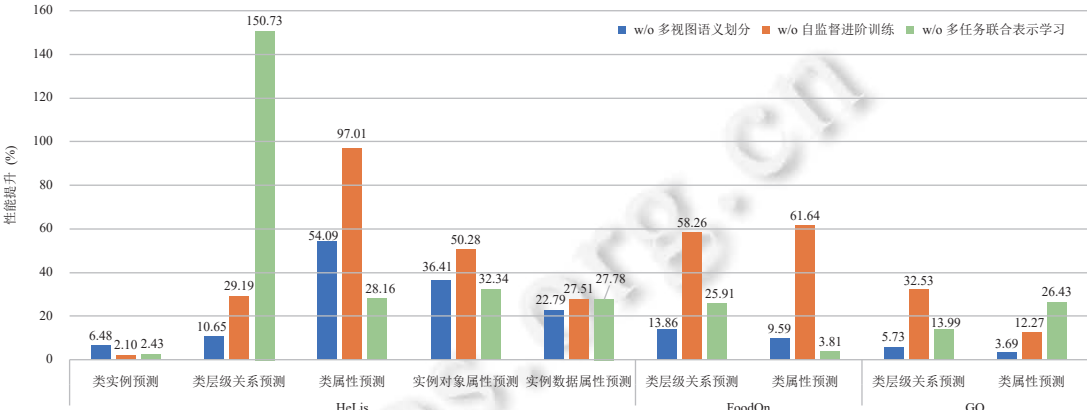


图 3 MSV-KRL 不同模块在各数据集所有任务中 MRR 指标的性能提升

表 14 HeLis 多语言模型适用性验证实验结果

任务	语言模型	MRR	H@1	H@5	H@10
类实例预测	RoBERTa	0.956	0.936	0.981	0.987
	BiomedBERT	0.972	0.958	0.992	0.996
	chatGLM3-6B	0.956	0.933	0.986	0.990
	LLaMA-2-7B	0.966	0.950	0.986	0.993
	BERT	0.969	0.951	0.992	0.997
类层级关系预测	RoBERTa	0.523	0.423	0.692	0.731
	BiomedBERT	0.529	0.400	0.720	0.880
	chatGLM3-6B	0.016	0.000	0.000	0.043
	LLaMA-2-7B	0.424	0.276	0.724	0.759
	BERT	0.509	0.364	0.709	0.818
类属性预测	RoBERTa	0.359	0.300	0.400	0.600
	BiomedBERT	0.437	0.333	0.500	0.667
	chatGLM3-6B	0.027	0.000	0.000	0.167
	LLaMA-2-7B	0.338	0.273	0.364	0.545
	BERT	0.396	0.333	0.500	0.583
实例对象属性预测	RoBERTa	0.174	0.113	0.227	0.287
	BiomedBERT	0.352	0.292	0.401	0.468
	chatGLM3-6B	0.282	0.220	0.338	0.384
	LLaMA-2-7B	0.250	0.194	0.297	0.362
	BERT	0.266	0.206	0.316	0.377
实例数据属性预测	RoBERTa	0.539	0.488	0.586	0.640
	BiomedBERT	0.683	0.636	0.733	0.774
	chatGLM3-6B	0.641	0.613	0.665	0.699
	LLaMA-2-7B	0.455	0.423	0.472	0.516
	BERT	0.598	0.545	0.653	0.700

表 15 FoodOn 与 GO 多语言模型适用性验证实验结果

数据集	任务	语言模型	MRR	H@1	H@5	H@10
FoodOn	类层级关系预测	RoBERTa	0.175	0.082	0.259	0.364
		BiomedBERT	0.326	0.195	0.474	0.596
		chatGLM3-6B	0.192	0.097	0.281	0.394
		LLaMA-2-7B	0.211	0.112	0.299	0.416
		BERT	0.345	0.212	0.492	0.618
	类属性预测	RoBERTa	0.199	0.130	0.262	0.332
		BiomedBERT	0.342	0.235	0.456	0.552
		chatGLM3-6B	0.260	0.172	0.346	0.442
		LLaMA-2-7B	0.231	0.149	0.313	0.390
		BERT	0.354	0.256	0.465	0.548
GO	类层级关系预测	RoBERTa	0.155	0.060	0.232	0.358
		BiomedBERT	0.342	0.181	0.532	0.693
		chatGLM3-6B	0.222	0.108	0.325	0.475
		LLaMA-2-7B	0.056	0.016	0.071	0.131
		BERT	0.277	0.146	0.412	0.565
	类属性预测	RoBERTa	0.329	0.231	0.438	0.522
		BiomedBERT	0.488	0.348	0.658	0.768
		chatGLM3-6B	0.285	0.169	0.413	0.520
		LLaMA-2-7B	0.321	0.205	0.449	0.554
		BERT	0.421	0.301	0.554	0.657

3.3.4 多类型 OWL 图谱适用性验证

为验证本文所提方法在面对不同垂直或通用领域 OWL 图谱时的性能表现, 本文通过知识表示学习任务对其进行测试, 本节选取 IB2M 和 DBpedia-20k 开展相应实验, 结果如表 16、表 17 所示. 其中, IB2M 中的大多数标注属性以中文为主, DBpedia-20k 以英语为主, 但包含了多种语言. 为了对比的公平性, 本节仅选取能够适用于多种类型 OWL 图谱 (多语言、多领域), 同时以原始 OWL 图谱作为输入的 OWL2Vec*方法作为基线方法, 其他实验设置参照第 3.1.2 节.

表 16 DBpedia-20k 实验结果

任务	方法	MRR	H@1	H@5	H@10
类实例预测	OWL2Vec*	0.882	0.856	0.912	0.929
	MSV-KRL	0.905	0.848	0.976	0.990
类层级关系预测	OWL2Vec*	0.217	0.126	0.344	0.437
	MSV-KRL	0.372	0.258	0.503	0.656
类属性预测	OWL2Vec*	0.209	0.098	0.341	0.472
	MSV-KRL	0.460	0.303	0.644	0.734

表 17 IB2M 实验结果

任务	方法	MRR	H@1	H@5	H@10
类实例预测	OWL2Vec*	0.351	0.131	0.681	0.949
	MSV-KRL	0.868	0.762	0.989	0.996
类层级关系预测	OWL2Vec*	0.216	0.073	0.317	0.488
	MSV-KRL	0.380	0.268	0.488	0.610
类属性预测	OWL2Vec*	0.100	0.001	0.161	0.323
	MSV-KRL	0.158	0.065	0.160	0.355
实例对象属性预测	OWL2Vec*	0.155	0.123	0.192	0.227
	MSV-KRL	0.240	0.175	0.289	0.380
实例数据属性预测	OWL2Vec*	0.012	0.002	0.009	0.019
	MSV-KRL	0.038	0.007	0.051	0.095

从实验结果可知, 由于 MSV-KRL 能够适配多种语言模型, 其能够利用面向多种语言的 PLMs 实现对实体和属性的编码, 在 2 个 OWL 图谱中的绝大多数任务上, 均取得了最佳性能. 与 MSV-KRL 相比, 虽然 OWL2Vec* 方法使用的 Word2Vec 编码器也能获取多语言特征表示, 但其初始设计目标并不针对多语言 OWL 图谱, 因此在 IB2M 和 DBpedia-20k 上性能表现有待提升.

3.3.5 实体嵌入可视化分析

为了进一步探究 MSV-KRL 实体嵌入的效果, 本文参照 OWL2Vec*[26] 与 InME/CoME[47] 中的可视化实验方法, 利用 t-SNE[69] 对 HeLis 和 FoodOn 中部分类以及类的实例进行可视化. 其中, 本文以 HeLis 中的 Histidine、Oleic Acid、Recipe Food 和 Monounsaturated Lipid 以及它们的实例为例, 分别使用 OWL2Vec* 方法以及 MSV-KRL 获取嵌入表示, 最后通过 t-SNE 进行可视化, 结果如图 4 所示. 此外, 本文以 FoodOn 中的 SFP (spice or flavor-producing plant)、FSC (27920-firm/semi-hard cheese (gouda and edam type) (efsa foodex2))、CFF (codex alimentarius food additive functional class) 以及 Obsolete Class 这 4 个类的所有子类为例进行可视化, 结果如图 5 所示.

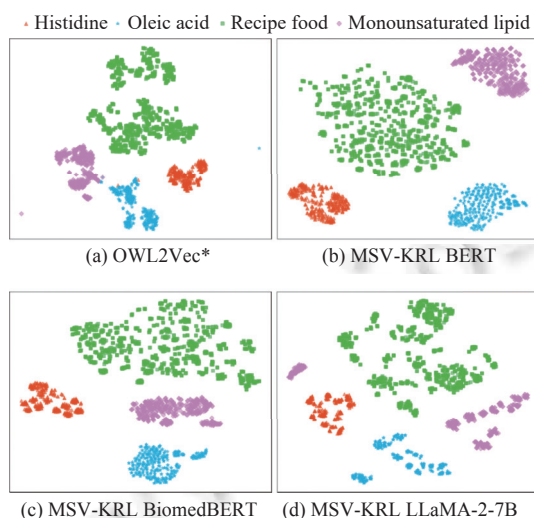


图 4 利用 t-SNE 对 HeLis 中类的实例嵌入进行可视化

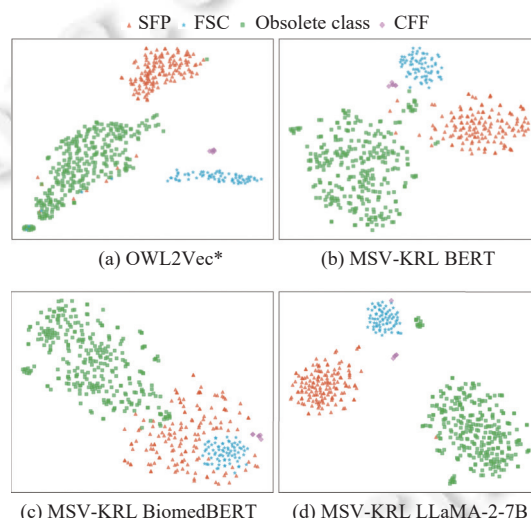


图 5 利用 t-SNE 对 FoodOn 中类的类层级关系嵌入进行可视化

通过图 4 可以发现, OWL2Vec* 中除 Histidine 类的实例较为聚集外, 其他 3 个类分散情况较为严重, 且除 Recipe Food 类之外其余各类均存在个别离散点. 与 OWL2Vec* 相比, 以 BERT 和 BiomedBERT 为编码模型的 MSV-KRL 方法中则不存在离散情况, 各个分散簇相比 OWL2Vec* 更加聚集, 彼此之间联系更紧密. 此外, 虽然 LLaMA-2-7B 在 HeLis 本体各任务上性能表现较为优异, 但由于未对其进行自监督进阶训练, 通过它获取的实例嵌入分散情况十分严重. 通过图 5 中可以发现, OWL2Vec* 中类嵌入除 CFF 的子类聚集效果较好之外, 其余类的子类都成条状分布, 较为分散. 与图 4 表现的结果相反, MSV-KRL 中 LLaMA-2-7B 与 BERT 的表现均比较好, BiomedBERT 的聚类情况反而在测试中表现一般, 例如 SFP 的子类将 FSC 的子类包围了起来, 但原本两者联系并不大. 综上, 本文所提以 BERT 模型为编码器的 MSV-KRL 方法能够有效学习概念与实例的嵌入, 同时准确理解如公理、文本等多种语义信息.

从整体上看, 图 5 中 FoodOn 各类与其子类的分散程度相比较于图 4 中 HeLis 各类与其实例之间更加分散, 这是因为 FoodOn 本体比 HeLis 本体拥有更多的实体数量以及更加复杂的公理信息, 部分也是由于 HeLis 中实例标注信息与其所属类的标注信息非常相似. 具体来看, 图 4 中 OWL2Vec* 方法对应的 Oleic Acid、Recipe Food 以及 Monounsaturated Lipid 这 3 个类的实例相比较与 MSV-KRL 的更加分散, 这是因为不同于 OWL2Vec* 在完整 RDF 图上展开随机游走以及在单任务上展开训练与预测, MSV-KRL 的多语义视图划分通过分割概念层与语义层

信息, 将类的信息与实例的信息进行单独处理, 但又通过类实例视图与其他视图一起通过进阶训练和多任务表示学习将两者信息给联系起来, 加强类与实例之间的语义联系程度. Histidine 在 OWL2Vec*方法中显示较为聚集, 这是由于其实例文本信息与 Histidine 类的文本信息几乎完全一样, 例如 Histidine_000, 而其他 3 个类与其实例有些许区别, 例如 RECIPEFOOD-FOOD-018-12.5 是 Recipe Food 的实例. 图 5 中, 由于 CCF (codex alimentarius food additive functional class) 的子类数量较少, 且其子类的标注属性包含其标注属性中的部分内容, 因此在 OWL2Vec*与 MSV-KRL 中都表现的比较聚集, 没有表现出较大差距. 针对 SFP (spice or flavor-producing plant), 由于 BiomedBERT 采用生物医学领域数据进行预训练, 而 SFP 属于一种植物, 因此其表现相较于其他几个类别较差, 这也与第 3.3.3 节中 BiomedBERT 在 FoodOn 与 HeLis 两个本体上的结果一致. Obsolete class 在 BiomedBERT 上的表现也是同理, 而 FSC (27920-firm/semi-hard cheese (gouda and edam type) (efsa foodex2)) 由于属于食物, 因此在 BiomedBERT 表现良好. 然而, 由于 OWL2Vec*针对完整的 RDF 图进行处理, 难以针对类之间子类关系以外的其他关系进行有效处理, 导致 Obsolete class 与 FSC 的子类并非如 MSV-KRL 那样聚集. 而 MSV-KRL 通过多语义视图划分和多任务联合表示学习, 使其不仅可以处理子类关系, 还可以处理类属性关系, 增强语义联系. LLaMA-2-7B 在此处虽然较为集中, 但离散点数量较多.

3.3.6 时间复杂度分析

为了能够综合分析 MSV-KRL 的时间复杂度, 将在本节中针对其各个模块的时间消耗做出针对性的实验分析. 首先本节的实验所采用的显卡为 A6000, CPU 为 Intel(R) Xeon(R) Gold 6338 CPU @ 2.00 GHz, 内存大小为 256 GB. 针对多语义视图划分模块采用直接计算其运行时间的方法, 将其结果作为时间复杂度, 同时对第 2.3 节中的算法 1 和算法 2 做理论性分析. 多任务联合表示学习针对经过进阶训练的语言模型获取嵌入的方法的时间复杂度, 采用与第 1 个模块相同的方法, 然后取平均值作为其时间消耗. 而针对具体的多任务推理, 本文选择统计同一本体不同任务的平均推理时间消耗作为该方法在某一数据集上的时间复杂度, 同时为了增加测试效率和减少不必要的消耗, 本文从测试集中选择 100 条数据在 MSV-KRL 和各基线方法上进行测试, 然后计算单个样本所花的时间. 为了能够和 MSV-KRL 的时间复杂度实验结果有效地进行对比, 本文选择的基线方法为 Box²EL 和 OWL2Vec*, 这是由于这两种方法能够与 MSV-KRL 一样同时处理 HeLis、FoodOn 和 GO 这 3 个本体, 且代表着不同类型的方法.

MSV-KRL 与 OWL2Vec*的本体映射时间消耗, 与 MSV-KRL 的多语义视图划分时间消耗, 以及 HeLis、FoodOn 和 GO 的实体数量以及公理数量都在表 18 中列出. 首先, 从表中可以发现 MSV-KRL 与 OWL2Vec*的本体映射时间相差无几, 甚至在 GO 上还要略低于 OWL2Vec*, 表明本体映射时间在合理范围内. 其次, 可以发现多语义视图划分时间花费最多的是 GO, 其次是 HeLis, 最后是 FoodOn. 这主要是由于 FoodOn 的规模只略大于 HeLis, 且 FoodOn 只具有类层次关系和类属性关系两个视图, 而 GO 在实体数量以及公理数量上都要远超 HeLis 与 FoodOn. 同时, MSV-KRL 各模块之间的耦合度较低, 可以预先进行处理从而降低整体框架的时间消耗.

表 18 多语义视图划分阶段时间复杂度

数据集	方法	本体映射时间 (s)	视图划分时间 (s)	实体数量	公理数量
HeLis	MSV-KRL	65.7	19.6	20 595	4 984
	OWL2Vec*	60.9	—		
FoodOn	MSV-KRL	3 300.7	9.3	28 182	142 536
	OWL2Vec*	3 318.6	—		
GO	MSV-KRL	28 350.1	35.5	44 244	452 028
	OWL2Vec*	28 476.5	—		

针对第 2.3.1 节中的算法 1、算法 2, 本文在此理论上对其进行时间复杂度分析. 首先假设算法 1 中对象属性集合 R_o 的大小为 n , 即 $|R_o| = n$, 那么算法 1 外层循环应该执行 n 次. 初始化一个空集合的时间复杂度为 $O(1)$, 然后分别执行 SPARQL 查询语句获取对象属性 r_o 的复杂定义域和复杂值域概念语义声明, 假设每次 SPARQL 查询需要时间 T_s , 而因为一次循环中进行了两次 SPARQL 查询 (分别查询定义域和值域), 每次循环的查询时间为 $2T_s$, 总

共的查询时间为 $O(n \times T_s)$. 针对条件判断语句而言, 条件分支的时间复杂度是常数 $O(1)$, 它们的影响可以忽略, 若条件满足, 会执行额外的 SPARQL 查询, 用来更新 C_{domain} 或 C_{range} , 假设这些查询的时间仍为 T_s , 则更新操作的时间复杂度为 $O(T_s)$, 可能在某些循环中被触发. 后续双层嵌套循环的时间复杂度为 $O(|C_{domain}| \times |C_{range}|)$, 又由于外层循环对 R_o 遍历了 n 次, 总时间复杂度为 $O(n \times |C_{domain}| \times |C_{range}|)$. 每次添加操作的时间复杂度为 $O(1)$, 然后添加操作在双层嵌套循环内执行, 总共执行 $n \times |C_{domain}| \times |C_{range}|$ 次, 因此其总体复杂度为 $O(n \times |C_{domain}| \times |C_{range}|)$. 综上所述算法 1 的时间复杂度为 $O(n \times (T_s + |C_{domain}| \times |C_{range}|))$. 同理对算法 2 进行分析, 可以得出算法 2 的时间复杂度为 $O(n \times (T_s + (m_s + m_c) + m_d \times m_r))$, 其中 m_s 、 m_c 是简单和复杂定义域集合的平均大小, m_d 和 m_r 表示定义域集合和简单值域集合的平均大小.

MSV-KRL 单个样本嵌入抽取时间, 以及 MSV-KRL、Box²EL、OWL2Vec* 的单个样本的推理时间消耗如表 19 中所示. 首先可以发现 MSV-KRL 获取 3 个数据集上单个样本嵌入时间消耗中, FoodOn 的值最大, HeLis 次之, GO 最小, 这主要是由于 FoodOn 中的实体数量略高于 HeLis, 而 GO 虽然获取嵌入的整体时间消耗要大于 FoodOn, 但 GO 中的实体数量要远远大于 FoodOn. 其次, 可以发现 OWL2Vec* 和 Box²EL 方法在单个样本推理时间上相差不大, 但 MSV-KRL 却要远大于前两个基线方法. 这主要是由于 MSV-KRL 在进行推理时, 是将所有的概念或实例都作为候选实体, 而 OWL2Vec* 则是经过了筛选. 与此同时, MSV-KRL 通过牺牲一定的时间复杂度, 使其在多个任务上的性能表现得到了极大的提升.

表 19 HeLis、FoodOn 及 GO 本体所有任务各方法时间复杂度分析

数据集	方法	嵌入抽取时间消耗 (ms)	推理时间消耗 (s)
HeLis	MSV-KRL	9.6	6.9
	Box ² EL	—	1.8
	OWL2Vec*	—	1.3
FoodOn	MSV-KRL	10.3	12.9
	Box ² EL	—	2.1
	OWL2Vec*	—	1.4
GO	MSV-KRL	6.3	33.1
	Box ² EL	—	2.3
	OWL2Vec*	—	1.2

4 总结与展望

本文提出一种新的多语义视图联合 OWL 图谱知识表示学习方法 (MSV-KRL). MSV-KRL 能够通过 OWL 到 RDF 图结构的语义映射优化、概念-属性-实例多视图语义划分以及多语义视图感知的自监督进阶训练表征概念之间的层级关系信息、多种公理信息、异质属性信息以及标注属性信息, 同时利用面向多语义视图的多任务联合表示学习加强 OWL 图谱概念层与实例层之间联系. 本文基于多个 OWL 评测数据集开展的多个预测任务对比实验, 实验结果表明 MSV-KRL 在绝大多数任务上优于当前先进的 OWL 图谱表示学习方法、联合概念层与实例层的表示学习方法以及针对 OWL 图谱描述逻辑公理的表示学习方法. 同时, 本文通过大量的消融实验、验证性实验与可视化分析充分证明了 MSV-KRL 各模块的有效性, 以及 MSV-KRL 不受限于语言模型且能够适用于多种类型的 OWL 图谱.

在未来的工作中将考虑引入时态信息, 面向通用或垂直领域时态知识图谱开展动态知识图谱表示学习. 同时, 鉴于垂直领域知识图谱中存在着多模态信息, 例如存在着部分病害图片与 IB2M 本体中的病害信息相关联, 而本文方法尚未考虑多模态信息, 将时态与多模态信息融合是未来的知识图谱相关研究中至关重要的一环, 需要进一步深入研究. 除上述未来工作以外, 本文还将针对 MSV-KRL 的多项不足在未来的研究工作中作出针对性的优化. 首先, 将考虑增强 MSV-KRL 对本体不同复杂程度公理信息的处理能力, 以及针对 OWL 不同子语言的适配能力. 其次, 针对 MSV-KRL 目前无法处理知识图谱中可能存在的语义模糊等数据不一致问题, 以及关系标注错误等

噪声数据的问题, 作者团队也将在未来的工作中对其展开研究, 提升模型方法的鲁棒性以及多种问题的处理能力。

References:

- [1] Wang M, Wang HF, Li BH, Zhao X, Wang X. Survey on key technologies of new generation knowledge graph. *Journal of Computer Research and Development*, 2022, 59(9): 1947–1965 (in Chinese with English abstract). [doi: [10.7544/jssn1000-1239.20210829](https://doi.org/10.7544/jssn1000-1239.20210829)]
- [2] Peng CY, Xia F, Naseriparsa M, Osborne F. Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review*, 2023, 56(11): 13071–13102. [doi: [10.1007/s10462-023-10465-9](https://doi.org/10.1007/s10462-023-10465-9)]
- [3] Chen Y, Wu LF, Zaki MJ. Toward subgraph-guided knowledge graph question generation with graph neural networks. *IEEE Trans. on Neural Networks and Learning Systems*, 2024, 35(9): 12706–12717. [doi: [10.1109/TNNLS.2023.3264519](https://doi.org/10.1109/TNNLS.2023.3264519)]
- [4] Yang JX, Yang XX, Li R, Luo MT, Jiang SX, Zhang Y, Wang D. BERT and hierarchical cross attention-based question answering over bridge inspection knowledge graph. *Expert Systems with Applications*, 2023, 233: 120896. [doi: [10.1016/j.eswa.2023.120896](https://doi.org/10.1016/j.eswa.2023.120896)]
- [5] Zhao N, Long Z, Wang J, Zhao ZD. AGRE: A knowledge graph recommendation algorithm based on multiple paths embeddings RNN encoder. *Knowledge-based Systems*, 2023, 259: 110078. [doi: [10.1016/j.knosys.2022.110078](https://doi.org/10.1016/j.knosys.2022.110078)]
- [6] Xia LQ, Liang YS, Leng JW, Zheng P. Maintenance planning recommendation of complex industrial equipment based on knowledge graph and graph neural network. *Reliability Engineering & System Safety*, 2023, 232: 109068. [doi: [10.1016/j.res.2022.109068](https://doi.org/10.1016/j.res.2022.109068)]
- [7] World-Wide Web Consortium. RDF 1.1 Primer. Boston: World-Wide Web Consortium, 2014.
- [8] Breitman KK, Casanova MA, Truszkowski W. RDF and RDF schema. *Semantic Web: Concepts, Technologies and Applications*, 2007, 57–79.
- [9] Hogan A. RDF schema and semantics. In: Hogan A. *The Web of Data*. Cham: Springer, 2020: 111–183. [doi: [10.1007/978-3-030-51580-5_4](https://doi.org/10.1007/978-3-030-51580-5_4)]
- [10] McGuinness DL, van Harmelen F. OWL Web ontology language—Overview. 2004. <https://www.w3.org/TR/owl-features/>
- [11] Antoniou G, van Harmelen F. Web ontology language: OWL. In: Staab S, Studer R. *Handbook on Ontologies*. Heidelberg: Springer, 2009: 91–110. [doi: [10.1007/978-3-540-92673-3_4](https://doi.org/10.1007/978-3-540-92673-3_4)]
- [12] Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: A collaboratively created graph database for structuring human knowledge. In: *Proc. of the 2008 ACM SIGMOD Int'l Conf. on Management of Data*. Vancouver: ACM, 2008. 1247–1250. [doi: [10.1145/1376616.1376746](https://doi.org/10.1145/1376616.1376746)]
- [13] Lehmann J, Isele R, Jakob M, Jentzsch A, Kontokostas D, Mendes PN, Hellmann S, Morsey M, van Kleef P, Auer S, Bizer C. DBpedia—A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 2015, 6(2): 167–195. [doi: [10.3233/SW-140134](https://doi.org/10.3233/SW-140134)]
- [14] Dragoni M, Bailoni T, Maimone R, Eccher C. HeLiS: An ontology for supporting healthy lifestyles. In: *Proc. of the 17th Int'l Semantic Web Conf. on the Semantic Web*. Monterey: Springer, 2018. 53–69. [doi: [10.1007/978-3-030-00668-6_4](https://doi.org/10.1007/978-3-030-00668-6_4)]
- [15] Dooley DM, Griffiths EJ, Gosal GS, Buttigieg PL, Hoehndorf R, Lange MC, Schriml LM, Brinkman FSL, Hsiao WWL. FoodOn: A harmonized food ontology to increase global food traceability, quality control and data integration. *npj Science of Food*, 2018, 2(1): 23. [doi: [10.1038/s41538-018-0032-6](https://doi.org/10.1038/s41538-018-0032-6)]
- [16] The Gene Ontology Consortium. The gene ontology project in 2008. *Nucleic Acids Research*, 2008, 36(S1): D440–D444. [doi: [10.1093/nar/gkm883](https://doi.org/10.1093/nar/gkm883)]
- [17] Zhong LF, Wu J, Li Q, Peng H, Wu XD. A comprehensive survey on automatic knowledge graph construction. *ACM Computing Surveys*, 2024, 56(4): 94. [doi: [10.1145/3618295](https://doi.org/10.1145/3618295)]
- [18] Du XY, Liu MW, Shen LW, Peng X. Survey on representation learning methods of knowledge graph for link prediction. *Ruan Jian Xue Bao/Journal of Software*, 2024, 35(1): 87–117 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/6902.htm> [doi: [10.13328/j.cnki.jos.006902](https://doi.org/10.13328/j.cnki.jos.006902)]
- [19] Bordes A, Usunier N, Garcia-Durán A, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. In: *Proc. of the 27th Int'l Conf. on Neural Int'l Processing Systems—Vol. 2*. Lake Tahoe: Curran Associates Inc., 2013. 2787–2795. [doi: [10.5555/2999792.2999923](https://doi.org/10.5555/2999792.2999923)]
- [20] Xiao H, Huang ML, Hao Y, Zhu XY. TransG: A generative mixture model for knowledge graph embedding. *arXiv:1509.05488*, 2017.
- [21] Zhu YL, Yang XP, Wang L, Zhang ZY. TransRD: Embedding of knowledge graph with asymmetric features. *Journal of Chinese Information Processing*, 2019, 33(11): 73–82 (in Chinese with English abstract). [doi: [10.3969/j.issn.1003-0077.2019.11.009](https://doi.org/10.3969/j.issn.1003-0077.2019.11.009)]
- [22] Li Z, Liu X, Wang X, Liu PK, Shen YX. TransO: A knowledge-driven representation learning method with ontology information constraints. *World Wide Web*, 2023, 26(1): 297–319. [doi: [10.1007/s11280-022-01016-3](https://doi.org/10.1007/s11280-022-01016-3)]
- [23] Jackermeier M, Chen JY, Horrocks I. Dual box embeddings for the description logic EL^{++} . In: *Proc. of the 2024 ACM Web Conf.*

- Singapore: ACM, 2024. 2250–2258. [doi: [10.1145/3589334.3645648](https://doi.org/10.1145/3589334.3645648)]
- [24] Zhapa-Camacho F, Hoehndorf R. Lattice-preserving ALC ontology embeddings. In: Proc. of the 18th Int'l Conf. on Neural-symbolic Learning and Reasoning, Part I. Barcelona: Springer, 2024. 355–369. [doi: [10.1007/978-3-031-71167-1_19](https://doi.org/10.1007/978-3-031-71167-1_19)]
 - [25] Chen JY, He Y, Geng YX, Jiménez-Ruiz E, Dong H, Horrocks I. Contextual semantic embeddings for ontology subsumption prediction. World Wide Web, 2023, 26(5): 2569–2591. [doi: [10.1007/s11280-023-01169-9](https://doi.org/10.1007/s11280-023-01169-9)]
 - [26] Chen JY, Hu P, Jimenez-Ruiz E, Holter OM, Antonyrajah D, Horrocks I. OWL2Vec*: Embedding of OWL ontologies. Machine Learning, 2021, 110(7): 1813–1845. [doi: [10.1007/s10994-021-05997-6](https://doi.org/10.1007/s10994-021-05997-6)]
 - [27] Wang Z, Zhang JW, Feng JL, Chen Z. Knowledge graph embedding by translating on hyperplanes. In: Proc. of the 28th AAAI Conf. on Artificial Intelligence. Québec City: AAAI Press, 2014. 1112–1119. [doi: [10.5555/2893873.2894046](https://doi.org/10.5555/2893873.2894046)]
 - [28] Lin YK, Liu ZY, Sun MS, Liu Y, Zhu X. Learning entity and relation embeddings for knowledge graph completion. In: Proc. of the 29th AAAI Conf. on Artificial Intelligence. Austin: AAAI Press, 2015. 2181–2187. [doi: [10.5555/2886521.2886624](https://doi.org/10.5555/2886521.2886624)]
 - [29] Ji GL, He SZ, Xu LH, Liu K, Zhao J. Knowledge graph embedding via dynamic mapping matrix. In: Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th Int'l Joint Conf. on Natural Language Processing (Vol. 1: Long Papers). Beijing: Association for Computational Linguistics, 2015. 687–696. [doi: [10.3115/v1/P15-1067](https://doi.org/10.3115/v1/P15-1067)]
 - [30] Nickel M, Tresp V, Kriegel HP. A three-way model for collective learning on multi-relational data. In: Proc. of the 28th Int'l Conf. on Int'l Conf. on Machine Learning. Bellevue: Omnipress, 2011. 809–816. [doi: [10.5555/3104482.3104584](https://doi.org/10.5555/3104482.3104584)]
 - [31] Yang BS, Yih WT, He XD, Gao JF, Deng L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In: Proc. of the 3rd Int'l Conf. on Learning Representations. San Diego, 2015.
 - [32] Trouillon T, Welbl J, Riedel S, Gaussier É, Bouchard G. Complex embeddings for simple link prediction. In: Proc. of the 33rd Int'l Conf. on Int'l Conf. on Machine Learning. New York: JMLR, 2016. 2071–2080. [doi: [10.5555/3045390.3045609](https://doi.org/10.5555/3045390.3045609)]
 - [33] Tay Y, Luu A, Hui SC. Non-parametric estimation of multiple embedding for link prediction on dynamic knowledge graphs. In: Proc. of the 2017 AAAI Conf. on Artificial Intelligence. San Francisco: AAAI Press, 2017. 1243–1249. [doi: [10.1609/aaai.v31i1.10685](https://doi.org/10.1609/aaai.v31i1.10685)]
 - [34] Jia YT, Wang YZ, Jin XL, Lin HL, Cheng XQ. Knowledge graph embedding: A locally and temporally adaptive translation-based approach. ACM Trans. on the Web, 2017, 12(2): 8. [doi: [10.1145/3132733](https://doi.org/10.1145/3132733)]
 - [35] Daruna A, Gupta M, Sridharan M, Chernova S. Continual learning of knowledge graph embeddings. IEEE Robotics and Automation Letters, 2021, 6(2): 1128–1135. [doi: [10.1109/LRA.2021.3056071](https://doi.org/10.1109/LRA.2021.3056071)]
 - [36] Wu TX, Khan A, Yong M, Qi GL, Wang M. Efficiently embedding dynamic knowledge graphs. Knowledge-based Systems, 2022, 250: 109124. [doi: [10.1016/j.knosys.2022.109124](https://doi.org/10.1016/j.knosys.2022.109124)]
 - [37] Yao SY, Zhao TZ, Wang RJ, Liu J. Rule-guided joint embedding learning of knowledge graphs. Journal of Computer Research and Development, 2020, 57(12): 2514–2522 (in Chinese with English abstract). [doi: [10.7544/issn1000-1239.2020.20200741](https://doi.org/10.7544/issn1000-1239.2020.20200741)]
 - [38] Nayyeri M, Wang ZH, Akter MM, Alam MM, Rony MRAH, Lehmann J, Staab S. Integrating knowledge graph embeddings and pre-trained language models in hypercomplex spaces. In: Proc. of the 22nd Int'l Semantic Web Conf. on the Semantic Web (ISWC 2023). Athens: Springer, 2023. 388–407. [doi: [10.1007/978-3-031-47240-4_21](https://doi.org/10.1007/978-3-031-47240-4_21)]
 - [39] Xiao L, Shan X, Wang YH, Deng ML. Research on joint representation learning methods for entity neighborhood information and description information. In: Proc. of the 8th China Conf. on Knowledge Graph and Semantic Computing: Knowledge Graph Empowers Artificial General Intelligence. Shenyang: Springer, 2023. 41–53. [doi: [10.1007/978-981-99-7224-1_4](https://doi.org/10.1007/978-981-99-7224-1_4)]
 - [40] Ning YL, Zhou G, Lu JC, Yang DW, Zhang T. A representation learning method of knowledge graph integrating relation path and entity description information. Journal of Computer Research and Development, 2022, 59(9): 1966–1979 (in Chinese with English abstract). [[10.7544/issn1000-1239.20210651](https://doi.org/10.7544/issn1000-1239.20210651)]
 - [41] Hu XY, Wang ZZ, Sun YY, Xu B, Lin HF. Knowledge graph representation method combined with semantic parsing. Journal of Computer Research and Development, 2022, 59(12): 2878–2888 (in Chinese with English abstract). [doi: [10.7544/issn1000-1239.20210849](https://doi.org/10.7544/issn1000-1239.20210849)]
 - [42] Shu D, Chen TL, Jin MY, Zhang C, Du MN, Zhang YF. Knowledge graph large language model (KG-LLM) for link prediction. In: Proc. of the 16th Asian Conf. on Machine Learning. Hanoi: PMLR, 2025. 143–158.
 - [43] Yao L, Peng JZ, Mao CS, Luo Y. Exploring large language models for knowledge graph completion. In: Proc. of the 2025 IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP 2025). Hyderabad: IEEE, 2025. 1–5. [doi: [10.1109/ICASSP49660.2025.10889242](https://doi.org/10.1109/ICASSP49660.2025.10889242)]
 - [44] Hitzler P, Krötzsch M, Parsia B, Patel-Schneider PF, Rudolph S. OWL 2 Web ontology language primer. W3C Recommendation, 2009, 27(1): 123.
 - [45] Rodríguez-García MÁ, Hoehndorf R. Inferring ontology graph structures using OWL reasoning. BMC bioinformatics, 2018, 19(1): 7.

- [doi: [10.1186/s12859-017-1999-8](https://doi.org/10.1186/s12859-017-1999-8)]
- [46] Chen J, Althagafi A, Hoehndorf R. Predicting candidate genes from phenotypes, functions and anatomical site of expression. *Bioinformatics*, 2021, 37(6): 853–860. [doi: [10.1093/bioinformatics/btaa879](https://doi.org/10.1093/bioinformatics/btaa879)]
 - [47] Shiraishi Y, Kaneiwa K. A self-matching training method with annotation embedding models for ontology subsumption prediction. *arXiv:2402.16278*, 2024.
 - [48] He Y, Chen JY, Jimenez-Ruiz E, Dong H, Horrocks I. Language model analysis for ontology subsumption inference. In: *Proc. of the 2023 Association for Computational Linguistics*. Toronto: Association for Computational Linguistics, 2023. 3439–3453. [doi: [10.18653/v1/2023.findings-acl.213](https://doi.org/10.18653/v1/2023.findings-acl.213)]
 - [49] Shi JC, Chen JY, Dong H, Khan I, Liang L, Zhou QZ, Wu Z, Horrocks I. Subsumption prediction for E-commerce taxonomies. In: *Proc. of the 20th Int'l Conf. on the Semantic Web*. Herssonissos: Springer, 2023. 244–261. [doi: [10.1007/978-3-031-33455-9_15](https://doi.org/10.1007/978-3-031-33455-9_15)]
 - [50] Hao JH, Chen MH, Yu WC, Sun YZ, Wang W. Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts. In: *Proc. of the 25th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining*. Anchorage: ACM, 2019. 1709–1719. [doi: [10.1145/3292500.3330838](https://doi.org/10.1145/3292500.3330838)]
 - [51] Iyer RG, Bai YS, Wang W, Sun YZ. Dual-geometric space embedding model for two-view knowledge graphs. In: *Proc. of the 28th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*. Washington: ACM, 2022. 676–686. [doi: [10.1145/3534678.3539350](https://doi.org/10.1145/3534678.3539350)]
 - [52] Huang ZJ, Wang DH, Huang BX, Zhang CW, Shang JB, Liang Y, Wang ZY, Li X, Faloutsos C, Sun YZ, Wang W. Concept2Box: Joint geometric embeddings for learning two-view knowledge graphs. In: *Proc. of the 2023 Association for Computational Linguistics*. Toronto: Association for Computational Linguistics, 2023. 10105–10118. [doi: [10.18653/v1/2023.findings-acl.642](https://doi.org/10.18653/v1/2023.findings-acl.642)]
 - [53] Bing R, Yuan G, Meng FR, Wang SZ, Qiao SJ, Wang ZX. Multi-view contrastive enhanced heterogeneous graph structure learning. *Ruan Ruan Jian Xue Bao/Journal of Software*, 2023, 34(10): 4477–4500 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/6883.htm> [doi: [10.13328/j.cnki.jos.006883](https://doi.org/10.13328/j.cnki.jos.006883)]
 - [54] Kulmanov M, Wang LW, Yuan Y, Hoehndorf R. EL embeddings: Geometric construction of models for the description logic $\overline{\mathcal{EL}}^{++}$. In: *Proc. of the 28th Int'l Joint Conf. on Artificial Intelligence*. Macao: IJCAI, 2019. 6103–6109. [doi: [10.24963/ijcai.2019/845](https://doi.org/10.24963/ijcai.2019/845)]
 - [55] Garg D, Ikbāl S, Srivastava SK, Vishwakarma H, Karanam H, Subramaniam LV. Quantum embedding of knowledge for reasoning. In: *Proc. of the 33rd Int'l Conf. on Neural Int'l Processing Systems*. Vancouver: Curran Associates Inc., 2019. 502. [doi: [10.5555/3454287.3454789](https://doi.org/10.5555/3454287.3454789)]
 - [56] Smaili FZ, Gao X, Hoehndorf R. Onto2Vec: Joint vector-based representation of biological entities and their ontology-based annotations. *Bioinformatics*, 2018, 34(13): i52–i60. [doi: [10.1093/bioinformatics/bty259](https://doi.org/10.1093/bioinformatics/bty259)]
 - [57] Smaili FZ, Gao X, Hoehndorf R. OPA2Vec: Combining formal and informal content of biomedical ontologies to improve similarity-based prediction. *Bioinformatics*, 2019, 35(12): 2133–2140. [doi: [10.1093/bioinformatics/bty933](https://doi.org/10.1093/bioinformatics/bty933)]
 - [58] Soylu A, Kharlamov E, Zheleznyakov D, Jimenez-Ruiz E, Giese M, Skjæveland MG, Hovland D, Schlatte R, Brandt S, Lie H, Horrocks I. OptiqueVQS: A visual query system over ontologies for industry. *Semantic Web*, 2018, 9(5): 627–660. [doi: [10.3233/SW-180293](https://doi.org/10.3233/SW-180293)]
 - [59] Holter OM, Myklebust EB, Chen JY, Jimenez-Ruiz E. Embedding OWL ontologies with OWL2Vec. In: *Proc. of the 18th Int'l Semantic Web Conf. (ISWC 2019)*. Auckland, 2019. 33–36.
 - [60] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional Transformers for language understanding. In: *Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1 (Long and Short Papers)*. Minneapolis: Association for Computational Linguistics, 2019. 4171–4186. [doi: [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423)]
 - [61] Crawshaw M, Košecká J. SLAW: Scaled loss approximate weighting for efficient multi-task learning. *arXiv:2109.08218*, 2021.
 - [62] Yang JX, Xiang FY, Li R, Zhang LY, Yang XX, Jiang SX, Zhang HY, Wang D, Liu XL. Intelligent bridge management via big data knowledge engineering. *Automation in Construction*, 2022, 135: 104118. [doi: [10.1016/j.autcon.2021.104118](https://doi.org/10.1016/j.autcon.2021.104118)]
 - [63] Li R, Mo TJ, Yang JX, Jiang SX, Li T, Liu YM. Ontologies-based domain knowledge modeling and heterogeneous sensor data integration for bridge health monitoring systems. *IEEE Trans. on Industrial Informatics*, 2021, 17(1): 321–332. [doi: [10.1109/TII.2020.2967561](https://doi.org/10.1109/TII.2020.2967561)]
 - [64] Ristoski P, Paulheim H. RDF2VEC: RDF graph embeddings for data mining. In: *Proc. of the 15th Int'l Semantic Web Conf. on the Semantic Web (ISWC 2016)*. Kobe: Springer, 2016. 498–514. [doi: [10.1007/978-3-319-46523-4_30](https://doi.org/10.1007/978-3-319-46523-4_30)]
 - [65] Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 2020, 36(4): 1234–1240. [doi: [10.1093/bioinformatics/btz682](https://doi.org/10.1093/bioinformatics/btz682)]
 - [66] Liu YH, Ott M, Goyal N, Du JF, Joshi M, Chen DQ, Levy O, Lewis M, Zettlemoyer L, Stoyanov V. RoBERTa: A robustly optimized bert pretraining approach. *arXiv:1907.11692*, 2019.
 - [67] Touvron H, Martin L, Stone K, *et al.* LLaMA 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023.

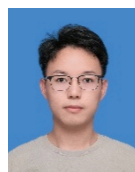
- [68] Zeng AH, Liu X, Du ZX, *et al.* GLM-130B: An open bilingual pre-trained model. In: Proc. of the 11th Int'l Conf. on Learning Representations. Kigali: OpenReview.net, 2022. 1–56.
- [69] van der Maten L, Hinton G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 2008, 9: 2579–2605.

附中文参考文献:

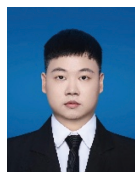
- [1] 王萌, 王昊奋, 李博涵, 赵翔, 王鑫. 新一代知识图谱关键技术综述. *计算机研究与发展*, 2022, 59(9): 1947–1965. [doi: [10.7544/j.issn1000-1239.20210829](https://doi.org/10.7544/j.issn1000-1239.20210829)]
- [18] 杜雪盈, 刘名威, 沈立炜, 彭鑫. 面向链接预测的知识图谱表示学习方法综述. *软件学报*, 2024, 35(1): 87–117. <http://www.jos.org.cn/1000-9825/6902.htm> [doi: [10.13328/j.cnki.jos.006902](https://doi.org/10.13328/j.cnki.jos.006902)]
- [21] 朱艳丽, 杨小平, 王良, 张志宇. TransRD: 一种不对等特征的知识图谱嵌入表示模型. *中文信息学报*, 2019, 33(11): 73–82. [doi: [10.3969/j.issn.1003-0077.2019.11.009](https://doi.org/10.3969/j.issn.1003-0077.2019.11.009)]
- [37] 姚思雨, 赵天哲, 王瑞杰, 刘均. 规则引导的知识图谱联合嵌入方法. *计算机研究与发展*, 2020, 57(12): 2514–2522. [doi: [10.7544/j.issn1000-1239.2020.20200741](https://doi.org/10.7544/j.issn1000-1239.2020.20200741)]
- [40] 宁原隆, 周刚, 卢记仓, 杨大伟, 张田. 一种融合关系路径与实体描述信息知识图谱表示学习方法. *计算机研究与发展*, 2022, 59(9): 1966–1979. [doi: [10.7544/j.issn1000-1239.20210651](https://doi.org/10.7544/j.issn1000-1239.20210651)]
- [41] 胡旭阳, 王治政, 孙媛媛, 徐博, 林鸿飞. 融合语义解析的知识图谱表示方法. *计算机研究与发展*, 2022, 59(12): 2878–2888. [doi: [10.7544/j.issn1000-1239.20210849](https://doi.org/10.7544/j.issn1000-1239.20210849)]
- [53] 邴睿, 袁冠, 孟凡荣, 王森章, 乔少杰, 王志晓. 多视图对比增强的异质图结构学习方法. *软件学报*, 2023, 34(10): 4477–4500. <http://www.jos.org.cn/1000-9825/6883.htm> [doi: [10.13328/j.cnki.jos.006883](https://doi.org/10.13328/j.cnki.jos.006883)]



杨建喜(1977—), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为桥梁健康监测, 大数据知识工程.



肖桥(2000—), 男, 博士生, CCF 学生会会员, 主要研究领域为自然语言处理.



谢江村(1998—), 男, 硕士生, CCF 学生会会员, 主要研究领域为知识图谱表示学习, 大数据知识工程.



蒋仕新(1992—), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为图像处理, 多模态大模型.



李韧(1985—), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为大数据知识工程, 自然语言处理.



贺丽荣(1994—), 女, 博士, 讲师, CCF 专业会员, 主要研究领域为知识图谱.



杨小霞(1996—), 女, 博士生, CCF 学生会会员, 主要研究领域为大数据知识工程.