

结合注意力与张量分解的依赖感知链接预测方法

丁里超^{1,2,3}, 赵晶^{1,2,3}

(1. 齐鲁工业大学(山东省科学院) 山东省计算中心(国家超级计算济南中心) 算力互联网与信息安全教育部重点实验室; 2. 齐鲁工业大学(山东省科学院) 计算机科学与技术学部 大数据应用技术山东省工程研究中心; 3. 山东省工业网络和信息网络安全重点实验室 山东省基础科学研究中心(计算机科学), 山东 济南 250353)

摘要: 知识图谱(KGs)通过实体—关系三元组实现现实世界的结构化表示。但现有的KGs通常是不完整的,需要从已知信息中推断缺失内容,也即进行链接预测。知识图谱嵌入(KGE)是用来解决链接预测问题的主流方法,但KGE方法存在使用单一语义信息的局限性,难以捕捉KGs中复杂的依赖关系。为此,提出一种结合注意力与张量分解的依赖感知链接预测方法(DALPM)。DALPM通过注意力机制和张量分解技术获取对KGs的语义增强表示。具体而言,DALPM采用Transformer作为编码器,通过多头自注意力机制捕获实体和关系之间的全局依赖信息,并使用非线性增强的Tucker分解即NeTu作为解码器进行嵌入得分计算。在标准数据集FB15k-237和WN18RR上的实验结果表明,DALPM相比其他先进方法展现出具有竞争力的性能。

关键词: 知识图谱; 链接预测; 知识图谱嵌入; 注意力机制; 张量分解

DOI: 10.11907/rjdk.241636

扫描二维码阅读全文: 

中图分类号: TP391.1

文献标识码: A

文章编号: 1672-7800(2025)010-0082-06

A Dependency-aware Link Prediction Method Combining Attention and Tensor Decomposition

DING Lichao^{1,2,3}, ZHAO Jing^{1,2,3}

(1. Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences); 2. Shandong Engineering Research Center of Big Data Applied Technology, Faculty of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences); 3. Shandong Fundamental Research Center for Computer Science, Shandong Provincial Key Laboratory of Industrial Network and Information System Security, Jinan 250353, China)

Abstract: Knowledge graphs (KGs) provide a structured representation of the real world through entity - relation triples. However, existing KGs are often incomplete, necessitating the inference of missing information, i.e., link prediction. Knowledge graph embedding (KGE) is the mainstream approach for addressing link prediction, but conventional KGE methods suffer from the limitation of relying on single-source semantic information, making it difficult to capture the complex dependencies within KGs. To overcome this limitation, we propose a Dependency-aware Link Prediction Method combining Attention and Tensor Decomposition (DALPM). DALPM learns semantically enhanced representations of KGs by integrating attention mechanisms with tensor decomposition. Specifically, DALPM employs a Transformer-based encoder that leverages multi-head self-attention to capture global dependencies between entities and relations, while the decoder adopts a nonlinearly enhanced Tucker decomposition, termed NeTu, to compute embedding scores. Experimental results on the standard datasets FB15k-237 and WN18RR demonstrate that DALPM achieves competitive performance compared with state-of-the-art methods.

Key Words: knowledge graphs; link prediction; knowledge graph embedding; attention mechanisms; tensor factorization

收稿日期: 2024-07-27

基金项目: 山东省自然科学基金—联合基金项目(ZR2022LZH008); 山东省本科教学改革研究重点项目(Z2023314); 济南市市校融合发展工程战略工程项目(JNSX2023045); 齐鲁工业大学(山东省科学院)研究生教改项目(YJG23YB002)

作者简介: 丁里超(2001-), 男, 齐鲁工业大学(山东省科学院)计算机科学与技术学部硕士研究生, 研究方向为知识图谱嵌入、知识图谱补全、链接预测; 赵晶(1979-), 女, 博士, CCF会员, 齐鲁工业大学(山东省科学院)计算机科学与技术学部副教授、硕士生导师, 研究方向为智能优化算法、机器学习、图像处理。本文通讯作者: 赵晶。

0 引言

知识图谱(Knowledge Graphs, KGs)通过捕捉实体之间的关系,构建了现实世界事实的结构化表示^[1]。知识图谱的基本单元是 (h, r, t) 形式的三元组,其中 h, t 分别表示头实体和尾实体, r 表示实体之间的关系。作为有组织的人类知识库,知识图谱被广泛应用于各种下游任务中,包括医学、图像处理任务,以及教育系统、咨询系统,特别是推荐系统中^[2-3]。尽管如今已有多个大型知识图谱被构建并持续维护,如Yago3和Freebase,但其仍然面临不完整的问题。例如,Freebase中近80%的人物没有已知的国籍。知识图谱的不完整性会对下游任务产生负面影响。因此,链接预测在知识图谱领域具有重要意义。

链接预测旨在利用知识图谱中已知的信息来推断可能存在的实体之间的关系。知识图谱链接预测领域目前主要可分为两个方向:基于嵌入的方法和基于文本的方法^[4]。

基于嵌入(Knowledge Graph Embedding, KGE)的方法具有简单、直接的模型结构,易于实现与分析。基于嵌入的方法将知识图谱中的实体和关系转换为连续的向量空间,主要分为3类:①基于翻译的方法。如TransE^[5]和RotatE^[6],通过平移实体和关系向量来建模语义相似性与相关性;②基于张量分解的方法,如RESKAL^[7]和TuckER^[8],将高阶关系转换为低阶张量,并将其分解为向量表示;③基于神经网络的方法。通过神经网络,使用如ConvE^[9]来学习实体和关系的表示。基于嵌入的方法主要关注三元组自身结构的建模^[10]。尽管TuckER等模型的性能已经获得了显著提升,但仍存在难以捕捉全局依赖信息的问题^[11]。

相比之下,基于文本的方法主要采用基于Transformer的预训练模型(Pre-trained Models, PTMs),目前已取得显著进展^[12]。这是由于自注意力机制能够有效捕捉知识图谱中的依赖关系。然而,现有基于文本的方法一是面临网络架构过于复杂的问题,二是该方法虽然能处理知识图谱中的全局信息,但是通常要将三元组处理成文本序列的形式,相对损失了更为重要的三元组自身的结构信息。

综上,针对知识图谱中的链接预测问题,本文提出结合注意力与张量分解的依赖感知链接预测方法(A Dependency-aware Link Prediction Method Combining Attention and Tensor Decomposition, DALPM),表现为一个编解码架构的模型。该模型主要由编码器部分的多头自注意力和解码器部分的张量分解构成。具体来讲,考虑到注意力头的数量对于捕捉知识图谱中全局依赖信息的重要性,编码器由具有多个注意力头的Transformer编码器块构成。针对知识图谱的结构特性,原始Transformer中针对序列任务的位置编码被舍弃。编码器块中前馈神经网络(Feedforward Neural Network, FFN)传统的多层感知机(Multilayer Perceptron, MLP)使用卷积进行替换,以增强FFN的参数

效率和泛化能力。为避免丢失三元组的初始嵌入信息,模型将初始嵌入与编码器获得的全局依赖特征进行加权融合,从而实现局部与全局信息的多尺度特征整合。解码器部分采用非线性增强的TuckER(即NeTu),由于TuckER是对Tucker分解的实现,而Tucker分解本质上是一种线性变换,这在一定程度上限制了模型的表达能力^[13]。为此,在TuckER的基础上引入非线性变换构建NeTu,以提升模型的表示能力。

本文的主要贡献如下:

(1)提出一种编码器—解码器架构的网络模型DALPM,能够解决KGE方法缺乏全局依赖信息的问题,从而完成知识图谱链接任务。

(2)编码器由多头自注意力的Transformer编码器块构成,在保留注意力机制的同时舍弃了原始Transformer中的位置编码,并通过卷积调整前馈网络,以增强对全局依赖信息的建模能力。同时,考虑到三元组自身结构的重要性,引入初始嵌入信息,并与全局依赖特征进行加权融合,从而在引入全局信息的同时保留局部信息,实现全局与局部的多尺度特征融合。

(3)在Tucker分解中引入非线性变化实现了NeTu,从而在一定程度上提升了解码器性能。

1 相关工作

在链接预测领域,学者们已经进行了广泛研究^[1,14-15]。本节将概述该领域相关研究,并进一步说明提出本文方法的原因。现有的链接预测方法大致可分为5类:基于翻译的方法、基于张量分解的方法、基于CNNs的方法、基于GNNs的方法和基于Transformer的方法。

1.1 基于翻译的方法

基于翻译的方法通过将实体和关系转换为向量空间对知识图谱进行映射。TransE利用Minkowski距离在实空间中定义评分函数^[5]。具体来说,给定嵌入 h, r 和 t ,TransE的评分函数为:

$$\phi_{\text{TransE}}(h, r, t) = \|h + r - t\|_{1/2} \quad (1)$$

其中, $1/2$ 表示L1或L2范数, h, r 和 t 分别表示头实体、关系与尾实体。TransE之后诞生了一系列Trans方法,基本构成了基于翻译的方法^[16]。这些方法拥有直观的几何解释并且比较高效,但在复杂三元组的应用中表现较差,容易出现嵌入向量的混淆。

1.2 基于张量分解的方法

基于张量分解的方法本质上是基于语义相似度的方法,与基于翻译的方法共同构成了传统的知识图谱嵌入方法。基于张量分解的方法通过将高维张量分解为多个低维张量的乘积,以获得实体和关系的嵌入表示。RESKAL通过将知识图谱中的高阶关系转换为低阶张量,以表示实体和关系之间的关联^[7]。其评分函数为:

$$\phi_{\text{RESCAL}}(h, r, t) = h^T W_r t \quad (2)$$

其中, h, t 分别为头实体和尾实体的嵌入, W_r 为表示关系 r 的矩阵。基于张量分解的方法相比基于翻译的方法, 可以更灵活地捕捉复杂的语义关系。但基于翻译的方法和基于张量分解的方法都主要关注三元组自身的结构, 相对忽略了三元组间的交互信息。

1.3 基于 CNNs 的方法

卷积神经网络 (Convolutional Neural Network, CNNs) 在深度学习中被广泛应用于学习深层表达特征, 在图像识别和自然语言处理等领域表现出色。ConvE 是一种基于 CNNs 的链接预测方法, 其利用卷积操作来捕捉实体和关系的复杂特征^[9]。ConvE 的评分函数如下:

$$\phi_{\text{ConvE}}(h, r, t) = \text{cat} \left(\text{mp} \left(\sigma \left(W * [h; r] \right)^T \right) \right) \cdot t \quad (3)$$

其中, W 表示卷积核, σ 表示激活函数, $*$ 表示二维卷积操作, $[h; r]$ 表示头实体和关系嵌入的拼接, mp 表示最大池化操作, cat 表示拼接。基于 CNNs 的方法是 CNNs 在链接预测领域的尝试, 虽然基于 CNNs 的方法能在一定程度上提取出深层特征信息, 但整体表现仍显不足。

1.4 基于 GNNs 的方法

GNNs 在编码器-解码器框架中被用来聚合知识图谱的结构信息。R-GCN 通过引入关系特定的变换来建模知识图谱的方向特性^[17]。其前向传播定义如下:

$$h_i^{(l+1)} = \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(l)} f_r(h_i^{(l)}, h_j^{(l)}) \quad (4)$$

其中, $h_i^{(l+1)}$ 表示第 $l+1$ 节点 i 的更新隐藏状态, $\mathcal{N}_i$ 表示节点 i 的邻域, $\alpha_{ij}^{(l)}$ 表示第 l 层节点 i 和 j 之间的归一化注意力权重, f_r 表示关系特定的变换函数, r 表示节点 i 和 j 之间的关系类型。基于 GNNs 的方法从理论上十分契合知识图谱的图结构, 但在实际应用中存在冗余信息聚合以及模型太过复杂的问题^[18]。

1.5 基于 Transformer 的方法

基于 Transformer 的方法显著提升了上下文表示学习的效果。KGBERT (BERT for Knowledge Graph Completion) 创造性地将预训练模型整合到知识图谱中, 利用 Transformer 的双向编码器表示作为实体和关系的编码器^[19]。随着大语

言模型的发展, 不少研究者尝试利用大模型的丰富知识来补全知识图谱, 但 SimKGC 指出, 此类方法丧失了建模三元组结构的能力, 致使性能达不到预期目标^[4]。此外, 基于 Transformer 的方法也普遍存在模型相对复杂的问题。

本文提出的 DALPM 同时利用注意力机制和张量分解技术, 进而解决基于张量分解的方法缺乏对知识图谱交互信息的关注, 以及基于 Transformer 的方法, 尤其是基于 Transformer 的文本方法丧失三元组自身结构处理能力的问题。

2 DALPM 方法

2.1 问题定义

在知识图谱中, $\mathcal{E}, \mathcal{R}$ 分别表示所有实体和关系的集合。对于一个三元组 (h, r, t) , 其中 h, r 和 t 分别表示头实体、关系与尾实体, 即 $h, t \in \mathcal{E}, r \in \mathcal{R}$ 。 $\mathcal{G}$ 表示一个真实的知识图谱, 其是所有真实三元组集合的一个子集。

链接预测旨在从 $\mathcal{G}$ 中的现有信息推断潜在的正确三元组, 并将这些三元组添加到 $\mathcal{G}$ 中, 以完成知识图谱补全的任务。

2.2 DALPM 框架

DALPM 的框架如图 1 所示。具体流程如下: 首先, 将知识图谱中的实体和关系嵌入到向量空间中, 并将每个三元组中的头实体和关系对 (e_h, w_r) 输入基于 Transformer 的编码器进行嵌入表示建模, 编码器通过聚合实体-关系对的交互信息生成具有丰富表达能力的表示; 之后, 在嵌入评分生成阶段, NeTu 解码器利用编码器生成的表示对头实体-关系对及其对应的尾实体 e_t 进行评分, 并通过 sigmoid 函数计算最终得分。

在编码器之前, 首先需要初始化嵌入向量。对于 $h, t \in \mathcal{E}$ 和 $r \in \mathcal{R}$, 分别有嵌入向量 $e_h, e_t \in R^d$ 和 $w_r \in R^d$ 。

2.3 嵌入表示建模

标准的 Transformer 使用固定或可学习的位置编码对序列进行编码以保留位置信息。然而, 与图像中的视觉块或文本中的词语不同, 知识图谱没有显式的序列信息。因此, DALPM 没有使用位置嵌入。

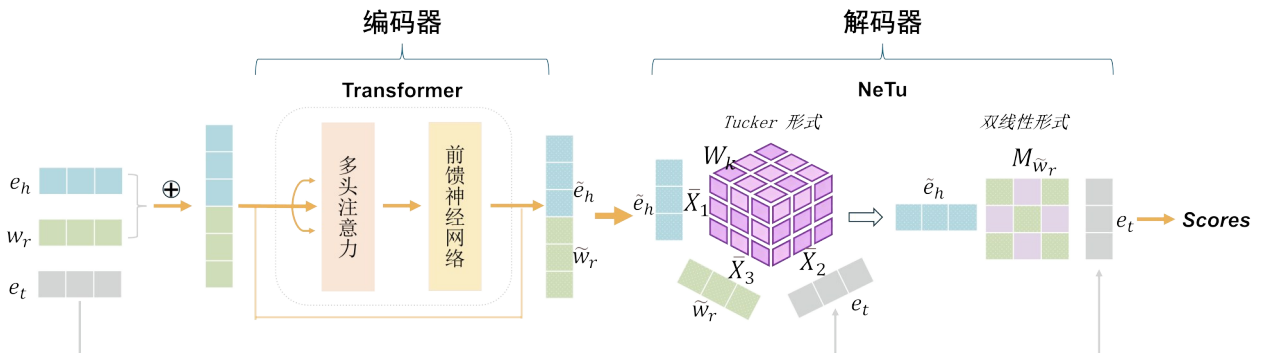


Fig. 1 The architecture of DALPM

图 1 DALPM 架构

在 DALPM 中, 通过具有多个注意力头的 Transformer 编码器实现知识图谱中的依赖感知和信息聚合能力。此外, 前馈神经网络(Feedforward Neural Network, FFN)使用卷积代替原始的多层感知机(Multilayer Perceptron, MLP)。由于自注意力机制可能导致初始嵌入信息弱化, DALPM 通过加权融合初始嵌入与编码后的表示, 从而获得更具表达力的嵌入表示。

假设一个简单的单头注意力机制, 对于一个三元组 (h, r, t) , 其中 $e_h, w_t \in R^d$, 当嵌入向量 (e_h, w_t) 被输入到 Transformer 编码器时, 注意力矩阵 A_{hr} 的计算公式如下:

$$A_{hr} = \sigma_{\text{softmax}} \left(\frac{QK^\top}{\sqrt{d_k}} \right) \quad (5)$$

其中, $Q \in R^{2 \times d}$, $K \in R^{2 \times d}$, 注意力矩阵 $A_{hr} \in R^{2 \times 2}$ 。

可以发现, 注意力矩阵 A_{hr} 是整合实体与关系信息的有效方式, 这是因为 A_{hr} 决定了头实体在多大程度上关注自身信息与关系信息。换句话说, A_{hr} 可被视为作用于查询相关输入的线性投影, 从而促进输入元素之间交互信息的传播。

考虑到注意力矩阵 A_{hr} 的重要性, 多头注意力在获取富有表现力的表示上显得至关重要。基本上通过对查询、键和值使用不同参数集的多重线性投影, 生成了不同的查询、键和值集。自注意力机制应用于每一组查询、键和值, 然后连接最终的值向量并进行线性变换。数学上表示为:

$$MH(Q, K, V) = \left[\sigma \left(\frac{Q_1 K_1^\top}{\sqrt{d_k}} \right) V_1, \dots, \sigma \left(\frac{Q_h K_h^\top}{\sqrt{d_k}} \right) V_h \right] M^P \quad (6)$$

式中, $Q_i = QW_i^Q$, $K_i = KW_i^K$, $V_i = VW_i^V$ 。其中, $W_i^Q \in R^{d \times d_k}$, $W_i^K \in R^{d \times d_k}$, $W_i^V \in R^{d \times d_v}$, $M^P \in R^{hd_k \times d}$ 是投影矩阵, h 是注意力头数量。

一维卷积可以在结构上实现与前馈网络(FFN)中多层感知机(MLP)相同的功能。当卷积核大小设置为 1 时, 卷积操作等价于对每个 token 的特征维度进行逐点线性变换, 因此可视为一种参数共享的 MLP 形式。通过这种替换, 可在保持表达能力的同时提升网络的参数效率与泛化性能。基于 token 的卷积前馈网络生成最终的输出表示矩阵:

$$H_o = \max(0, MH(Q, K, V)C_1 + b_1)C_2 + b_2 \quad (7)$$

其中, $C_1 \in R^{d \times d_h \times 1}$, $C_2 \in R^{d_h \times d \times 1}$, $b_1 \in R^{d_h}$, $b_2 \in R^d$ 。

最后将 FFN 产生的表示矩阵与初始嵌入矩阵融合, 生成最终的嵌入表示:

$$F = a \cdot H_o + E \quad (8)$$

其中, F 是最终的输出特征矩阵, H_o 、 E 分别是经过 Transformer 编码后的嵌入矩阵和初始嵌入矩阵, a 是加权融合参数。

2.4 嵌入评分生成

TuckER 是一种基于张量分解的模型, 通过共享核心张量的参数来控制参数数量^[8]。然而, 由于其具有的线性性质, TuckER 的表达能力受到一定限制。DALPM 尝试通

过引入非线性来增强 TuckER 的能力, 并将非线性增强后的 TuckER 称为 NeTu。

首先介绍 TuckER。如上文所述, TuckER 基于三阶张量 $\mathcal{X}$ 的 Tucker 分解。具体公式如下:

$$\mathcal{X} \approx \mathcal{Z} \times_1 A \times_2 B \times_3 C \quad (9)$$

其中, $\mathcal{X} \in R^{L \times M \times N}$, $\mathcal{Z} \in R^{I \times J \times K}$, $A \in R^{L \times I}$, $B \in R^{M \times J}$, $C \in R^{N \times K}$, $\times_n$ 表示沿第 n 个模维度进行的张量乘积运算。TuckER 的评分函数可以表示为:

$$\Phi_{\text{TuckER}}(h, r, t) = \mathcal{W} \times_1 e_h \times_2 e_t \times_3 w_r \quad (10)$$

其中, $e_h, e_t \in R^d$ 是 $\mathcal{E}$ 的行, 分别表示头实体和尾实体的嵌入向量, $w_r \in R^d$ 是 $\mathcal{R}$ 的行, 表示关系的嵌入向量, $\mathcal{W} \in R^{d_e \times d_e \times d_r}$ 是核心张量。

神经张量网络(Neural Tensor Network, NTN)的评分函数与 TuckER 类似, 但 NTN 实现了非线性变换, 赋予了模型更强的表达能力^[20]。然而, NTN 存在一个主要缺点: 其使用不同的权重矩阵来建模每种关系, 导致随着知识图谱规模的增长, 模型参数数量急剧增加。相比之下, TuckER 通过共享核心张量来抑制参数数量, 但由于其仅基于线性变换, 性能相对被限制。通过结合这两者的优点实现了一种非线性增强的张量分解方法, 称为 NeTu。NeTu 的评分函数为:

$$\Phi_{\text{E-TuckER}}(h, r, t) = e_t^\top \cdot \sigma(\mathcal{W} \times_1 e_h \times_3 w_r) \quad (11)$$

其中, $w_r \in R^d$ 是关系的嵌入向量, σ 是非线性激活函数(如 ELU 或 ReLU), $\cdot$ 是点积。

在 DALPM 中, NeTu 的评分函数表示为:

$$\Phi_{\text{NeTu}}(h, r, t) = e_t^\top \cdot \sigma(\mathcal{W} \times_1 \widetilde{e}_h \times_3 \widetilde{w}_r) \quad (12)$$

2.5 训练策略

DALPM 采用标准的数据增强技术, 最初由 Dettmers 等^[9]进行形式化, 后由 Lacroix 等^[21]作进一步完善。这项技术涉及为数据集中的每个三元组添加互逆关系, 即对每个 (h, r, t) 添加 (t, r', h) , 使得模型能够仅关注三元组关系的正向传播。同时, DALPM 使用 1-N 评分机制来加速训练。DALPM 采用二元交叉熵(Binary Cross Entropy Loss, BCE)损失作为损失函数, 使模型能够在训练过程中有效区分正样本与负样本。DALPM 损失函数具体表达式如下:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{|\mathcal{E}|} \sum_{e' \in \mathcal{E}} [y_{e'} \log p_{e'} + (1 - y_{e'}) \log(1 - p_{e'})] \quad (13)$$

其中, $p \in R^d$ 是预测概率向量, $y \in R^d$ 是二元标签向量, $\mathcal{E}$ 是所有实体的集合。

3 实验与结果分析

3.1 数据集

为了评估 DALPM 的性能, 本文采用两个主流标准数据集 FB15k-237 和 WN18RR。数据集统计信息如表 1 所示。

(1) FB15k-237。FB15k-237 数据集是从 Freebase 知识图谱中构建的一个子集。FB15k-237 数据集去除了 FB15k 数据集中的冗余实体和关系, 同时保留了原始数据

Table 1 Datasets statistics

表1 数据集统计数据

数据集	$ \mathcal{E} $	$ \mathcal{R} $	训练集	验证集	测试集
FB15k-237	14 541	237	272 115	17 535	20 466
WN18RR	40 943	11	86 835	3 034	3 134

集中最频繁出现的实体和关系,使简单模型在该数据集上难以获得良好性能。

(2) WN18RR。WN18RR 数据集是 WN18 数据集的重新划分版本,构建自 WordNet 知识图谱的一个子集。与原始 WN18 数据集相比,WN18RR 数据集去除了数据中不正确的负样本,并重新分配了训练集、验证集和测试集。

3.2 基线模型

本文将 DALPM 与不同类别的先进模型进行了比较。在欧几里得空间中,选择 TransE^[5]作为基于平移的模型,ConvE^[9]、A2N^[22]、QuatE^[23]作为基于神经网络的模型,TuckER^[8]和 Hyper^[24]作为张量分解模型。另外,选择 ComplEx^[25]和 RotatE^[6]作为使用复数空间的模型。在双曲空间中,选择 MuRP^[26]和 ATTH^[27]作为基于双曲空间的模型。

3.3 评价指标

在链接预测任务中,MRR (Mean Reciprocal Rank)和 Hits@K 是评估模型性能的常用指标。MRR 是所有测试样本倒数排名的平均值,反映了模型对所有正确答案排名的总体表现。MRR 值越高,表示模型在预测正确答案时的排名越靠前。Hits@K (K=1, 3, 10)则表示在所有测试样本中,正确答案出现在前 K 名中的比例,反映了模型在不同宽容度下的预测能力。综合使用 MRR 和 Hits@K 可以全面评估模型在链接预测任务中的表现。

3.4 实验结果

将 DALPM 在 FB15K-237 和 WN18RR 两个标准数据集上与多个基线方法进行比较,结果如表 2 所示。其中,† 引自文献[28],‡ 引自文献[25],其余结果来自各自的原始文献。最好的结果用粗体表示,排名第二的结果用下划线表示。

由表 2 可以看出,DALPM 在这两个数据集上都取得了具有竞争力的结果。在 FB15K-237 数据集上,TuckER 的结

Table 2 Results of link prediction on two standard datasets

表2 在两个标准数据集上链接预测结果

模型	FB15K-237				WN18RR			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE [†] ^[5]	0.257	0.174	0.284	0.420	0.182	0.027	0.295	0.444
ComplEx [‡] ^[25]	0.247	0.158	0.275	0.428	0.440	0.410	0.460	0.510
ConvE ^[9]	0.325	0.237	0.356	0.501	0.430	0.400	0.440	0.520
A2N ^[22]	0.317	0.232	0.348	0.486	0.450	0.420	0.460	0.510
MURP ^[26]	0.335	0.243	0.367	0.518	0.481	0.440	0.495	0.566
RotatE ^[6]	0.338	0.241	0.375	0.533	0.476	0.428	0.492	0.571
Hyper ^[24]	0.341	0.252	0.376	0.520	0.465	0.436	0.477	0.522
QuatE ^[23]	0.348	0.248	0.382	0.550	<u>0.488</u>	0.438	0.508	0.582
TuckER ^[8]	<u>0.358</u>	<u>0.266</u>	<u>0.394</u>	0.544	0.470	<u>0.444</u>	0.482	0.526
ATTH ^[27]	0.348	0.252	0.384	0.540	0.486	0.443	0.499	<u>0.573</u>
DALPM	0.363	0.268	0.401	<u>0.545</u>	0.489	0.451	<u>0.503</u>	0.555

果最逼近 DALPM 的方法。通过在关系之间共享知识进行多任务学习,使得 TuckER 在具有相对较多关系的数据集上表现出色。然而,在像 WN18RR 这种具有更多实体和较少关系的数据集上,TuckER 的性能会受到影响。相比之下,SAAttLE 中的自注意力机制捕获了实体与关系之间的依赖关系和交互作用,NeTu 非线性的引入进一步提升了建模知识图谱结构的能力。QuatE 和 ATTH 在 WN18RR 数据集上取得了很好的效果,特别是 QuatE 在 WN18RR 数据集上的 Hits@10 指标达到了 0.582。不过这是因为此类方法预先假设了数据集的多层次结构,并进行了网络的针对性调整,因此在 WN18RR 这种多层次结构的数据集上表现较好。而 DALPM 并未预先假设数据集,因此在 FB15K-237 和 WN18RR 两个情况完全不同的数据集上都表现良好。

4 分析与讨论

4.1 时间复杂度分析

DALPM 由一个基于 Transformer 的编码器和一个基于 Tucker 分解的嵌入评分解码模块组成。模型中的多头注意力块分为 3 个步骤:首先,为了在每个头部生成矩阵 $Q \in R^{2 \times d}$ 、 $K \in R^{2 \times d}$ 和 $V \in R^{2 \times d}$,输入矩阵 $[e_h; w_i] \in R^{2 \times d}$ 需要通过 3 个维度对 $R^{d \times d}$ 的可学习矩阵进行线性变换,该过程的计算复杂度为 $\mathcal{O}(d^2)$ 。其次,要计算公式(6)中每

个头的输出表示矩阵 $\sigma\left(\frac{Q_i K_i^\top}{\sqrt{d_k}}\right) V_i \in R^{2 \times d}$,则要根据公式

(5) 计算注意力矩阵 $A_{hr} \in R^{2 \times 2}$,总体时间复杂度为 $\mathcal{O}(d)$ 。最后,根据公式(7)计算 $H_o \in R^{2d}$,该步骤的复杂度同样为 $\mathcal{O}(d)$ 。因此,编码器部分的总体时间复杂度为 $\mathcal{O}(d^2)$ 。在解码器 NeTu 部分,根据公式(12),主要涉及核心张量的 n 模计算,其复杂度为 $\mathcal{O}(d^3)$ 。因此,DALPM 的总体时间复杂度为 $\mathcal{O}(d^3)$,与 TuckER 模型处于同一量级。

4.2 消融实验

针对模型的主要模块进行消融实验,结果如表 3 所示。可以看出,NeTu 的非线性引入一定程度提升了网络对知识图谱的建模能力,这与前文对于 Tucker 分解线性变化本质的分析保持一致。其次,引入初始嵌入信息也有助于提升模型性能,这可能与模型在丧失初始嵌入信息后更容易过拟合有关。此外,初始嵌入信息的引入可视为一种多层次特征融合,从而增强了模型的表达能力。总之,完整的 DALPM 模型取得了最佳性能,验证了本文方法的有效性。

Table 3 Results of ablation experiment

表3 消融实验结果

模型	FB15K-237		WN18RR	
	MRR	Hits@10	MRR	Hits@10
w/o Encoder	0.362	0.545	0.485	0.548
w/o Initial Embedding	0.359	0.541	0.483	0.542
DALPM	0.363	0.545	0.489	0.555

5 结语

本文提出一种结合了自注意力和张量分解的编解码模型 DALPM, 以完成知识图谱中的链接预测任务。DALPM 在一定程度上解决了 KGE 方法缺乏捕捉全局依赖信息能力的问题, 以及非 KGE 方法难以建模三元组自身结构的问题。最近, 有研究人员探索了多分区嵌入, 也许非线性增强的 NeTu 可以与多分区嵌入相结合^[29-30]。将 Transformer 扩展为 Graph Transformer, 以引入节点领域信息来增强局部表示, 也是一个值得探索的方向^[31-32]。

参考文献:

- [1] JI S, PAN S, CAMBRIA E, et al. A survey on knowledge graphs: representation, acquisition, and applications[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(2): 494-514.
- [2] SHEN T, ZHANG F, CHENG J. A comprehensive overview of knowledge graph completion[J]. Knowledge-Based Systems, 2022, 255: 109597.
- [3] DU X Y, LIU M W, SHEN L W, et al. Survey on representation learning methods of knowledge graph for link prediction[J]. Journal of Software, 2024, 35(1): 87-117.
杜雪盈, 刘名威, 沈立伟, 等. 面向链接预测的知识图谱表示学习方法综述[J]. 软件学报, 2024, 35(1): 87-117.
- [4] WANG L, ZHAO W, WEI Z, et al. SimKGC: simple contrastive knowledge graph completion with pre-trained language models[C]//Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, 2022: 4281-4294.
- [5] BORDES A, USUNIER N, GARCIA-DURAN A, et al. Translating embeddings for modeling multi-relational data[C]//Proceedings of the 27th International Conference on Neural Information Processing Systems, 2013: 2787-2795.
- [6] SUN Z, DENG Z H, NIE J Y, et al. Rotate: knowledge graph embedding by relational rotation in complex space[DB/OL]. <https://arxiv.org/abs/1902.10197>.
- [7] NICKEL M, TRESP V, KRIEGEL H P. A three-way model for collective learning on multi-relational data[C]//Proceedings of the 28th International Conference on Machine Learning, 2011: 809-816.
- [8] BALAZEVIC I, ALLEN C, HOSPEDALES T. TuckER: tensor factorization for knowledge graph completion[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019: 5185-5194.
- [9] DETTMERS T, MINERVINI P, STENETORP P, et al. Convolutional 2D knowledge graph embeddings[C]//Proceedings of the Thirty-second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, 2018: 1811-1818.
- [10] DAI Y, WANG S, XIONG N N, et al. A survey on knowledge graph embedding: approaches, applications and benchmarks[J]. Electronics, 2020, 9(5): 750.
- [11] BAGHERSHAHI P, HOSSEINI R, MORADI H. Self-attention presents low-dimensional knowledge graph embeddings for link prediction[J]. Knowledge-based Systems, 2023, 260: 110124.
- [12] PAN S, LUO L, WANG Y, et al. Unifying large language models and knowledge graphs: a roadmap[J]. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(7): 3580-3599.
- [13] SONKAR S, KATIYAR A, BARANIUK R. NePTuNe: neural powered tucker network for knowledge graph completion[C]// Proceedings of the 10th International Joint Conference on Knowledge Graphs, 2021: 177-180.
- [14] GUO Q, ZHUANG F, QIN C, et al. A survey on knowledge graph-based recommender systems[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 34(8): 3549-3568.
- [15] HOGAN A, BLOMQUIST E, COCHEZ M, et al. Knowledge graphs[J]. ACM Computing Surveys, 2022, 54(4): 1-37.
- [16] ZHANG T C, TIAN X, SUN X H, et al. Overview on knowledge graph embedding technology research[J]. Journal of Software, 2023, 34(1): 277-311.
张天成, 田雪, 孙相会, 等. 知识图谱嵌入技术研究综述[J]. 软件学报, 2023, 34(1): 277-311.
- [17] SCHLICHTKRULL M, KIPF T N, BLOEM P, et al. Modeling relational data with graph convolutional networks[C]//The Semantic Web, 2018: 593-607.
- [18] YANG D H, HE T, WANG H Z, et al. Survey on knowledge graph embedding learning[J]. Journal of Software, 2022, 33(9): 3370-3390.
杨东华, 何涛, 王宏志, 等. 面向知识图谱的图嵌入学习研究进展[J]. 软件学报, 2022, 33(9): 3370-3390.
- [19] YAO L, MAO C, LUO Y. KG-BERT: BERT for knowledge graph completion[DB/OL]. <https://arxiv.org/abs/1909.03193>.
- [20] SOCHER R, CHEN D, MANNING C D, et al. Reasoning with neural tensor networks for knowledge base completion[C]//Proceedings of the 27th International Conference on Neural Information Processing Systems, 2013: 926-934.
- [21] LACROIX T, USUNIER N, OBOZINSKI G. Canonical tensor decomposition for knowledge base completion[C]//Proceedings of International Conference on Machine Learning, 2018: 2863-2872.
- [22] BANSAL T, JUAN D C, RAVIS, et al. A2N: attending to neighbors for knowledge graph inference[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019: 4387-4392.
- [23] ZHANG S, TAY Y, YAO L, et al. Quaternion knowledge graph embeddings[C]//Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019: 2735-2745.
- [24] BALAZEVIC I, ALLEN C, HOSPEDALES T M. Hypernetwork knowledge graph embeddings[C]// Proceedings of 28th International Conference on Artificial Neural Networks, 2019: 553-565.
- [25] TROUILLON T, WELBL J, RIEDEL S, et al. Complex embeddings for simple link prediction[C]//Proceedings of International Conference on Machine Learning, 2016: 2071-2080.
- [26] BALAZEVIC I, ALLEN C, HOSPEDALES T. Multi-relational poincaré graph embeddings[C]//Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019: 4463-4473.
- [27] CHAMI I, WOLF A, JUAN D C, et al. Low-dimensional hyperbolic knowledge graph embeddings[C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020: 6901-6914.
- [28] NICKEL M, ROSASCO L, POGGIO T. Holographic embeddings of knowledge graphs[C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2016: 1955-1961.
- [29] TRAN H N, TAKASU A. Multi-partition embedding interaction with block term format for knowledge graph completion[DB/OL]. <https://arxiv.org/abs/2006.16365>.
- [30] TRAN H N, TAKASU A. MEIM: multi-partition embedding interaction beyond block term format for efficient and expressive link prediction[C]// Proceedings of the Thirty-first International Joint Conference on Artificial Intelligence, 2022: 2262-2269.
- [31] YUN S, JEONG M, KIM R, et al. Graph transformer networks[C]//Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019: 11983-11993.
- [32] MIN E, CHEN R, BIAN Y, et al. Transformer for graphs: an overview from architecture perspective [DB/OL]. <https://arxiv.org/abs/2202.08455>.