中图分类号：**TP391.4**

论 文 编 号：**10006SY2121127**

北京航空航天大學

硕 士 学 位 论 文

基于 Transformer 的知识图谱补全算法研究

作 者 姓 名　朱 桐

学 科 专 业　软 件 工 程

指 导 教 师　谭 火 彬　副 教 授

培 养 学 院　软 件 学 院

Research of Knowledge Graph Completion Algorithm Based on Transformer

A Dissertation Submitted for the Degree of Master

Candidate : Zhu Tong

Supervisor : Assoc. Prof. Tan Huobin

School of Software

Beihang University, Beijing, China

中图分类号：**TP391.4**

论 文 编 号：**10006SY2121127**

硕 士 学 位 论 文

基于 Transformer 的知识图谱补全算法研究

作者姓名	朱桐	申请学位级别	工学硕士
指导教师姓名	谭火彬	职　　称	副教授
学科专业	软件工程	研究方向	软件工程
学习时间自	2021 年　09 月　01 日	起至	2024 年　05 月　16 日止
论文提交日期	2024 年　01 月　10 日	论文答辩日期	2024 年　03 月　01 日
学位授予单位	北京航空航天大学	学位授予日期	年　　月　　日

关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中作出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名：_____ 日期： 年 月 日

学位论文使用授权

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名：_____ 日期： 年 月 日

指导教师签名：_____ 日期： 年 月 日

摘　要

知识图谱多个人工智能领域中得到了广泛的应用。但是，目前大多数的知识图谱是不完全的，因此知识图谱补全任务成为了热门的研究领域。知识图谱补全任务旨在预测知识图谱中缺失的三元组，目前的主流方案是知识图谱嵌入，将实体和关系投影到低维向量空间中以学习知识图谱中的隐含规律。许多现有的方法通过利用图谱的图结构信息获得优异的性能，其中最具代表性的为利用图谱中局部邻域结构的基于图神经网络的方法。但是一方面图神经网络其浅层的网络结构限制了模型的表达能力，另一方面存在的过度平滑问题也导致模型无法利用长距离的信息。Transformer 是注意力机制方面里程碑式的工作，基于 Transformer 的模型变体在计算机视觉和编程语言领域中表现出了出色的性能。因此针对上述不足，本文研究基于 Transformer 的知识图谱补全方法，提出了基于领域感知的 Transformer 模型 NATLP 以及结合图路径和局部邻域的 Transformer 模型 TKGE-PN。本文的主要工作内容如下：

（1）针对基于图神经网络的知识图谱嵌入方法表达能力不足、无法捕捉邻居实体之间的相互依赖的问题，本文提出了基于领域感知的 Transformer 模型 NATLP，通过关系特定的邻居实体信息构造建模了不同关系对于实体传递消息的影响，并对 Transformer 的自注意力计算机制进行了改造，使 Transformer 能够感知到局部邻域内的图结构数据，学习到邻居实体之间的相互依赖。

（2）针对基于图神经网络的方法以及 NATLP 无法学习到长距离依赖的问题，本文提出了结合图路径和局部领域的 Transformer 模型 TKGE-PN。模型首先通过有偏随机游走算法以中心实体为起点采样多条图路径，其次通过基于 Transformer 的图路径编码模块学习图路径中的长距离依赖并转化为向量表示，最后局部邻域编码模块集合所有图路径向量结合局部邻域信息表示对三元组打分。在图路径编码的过程中，利用掩蔽实体关系预测任务增强模型对于长距离信息的学习能力。

（3）在两个主流基准数据集 FB15K-237 和 WN18RR 上对提出的 NATLP 和 TKGE-PN 模型进行了实验，证明了本文提出的模型以及关键设计的有效性。

关键词：知识图谱，知识图谱补全，知识图谱嵌入，Transformer，图路径，局部邻域，

Abstract

Given the widespread application of Knowledge Graphs (KGs) across multiple Artificial Intelligence domains, the incompleteness of most current knowledge graphs has made Knowledge Graph Completion (KGC) a hot research area. KGC aims to predict missing triples in a knowledge graph, and the current mainstream solution is Knowledge Graph Embeddings (KGE), which projects entities and relations into a low-dimensional vector space to learn the hidden patterns within the knowledge graph. Many existing methods achieve excellent performance by utilizing the graph structure information of the graph, among which the most representative is the method based on Graph Neural Networks (GNNs) that utilizes the local neighborhood structure in the graph. However, the shallow network structure of GNNs limits the model's expressive power on one hand, and on the other hand, the issue of over-smoothing also prevents the model from utilizing long-distance information. Transformer, a milestone work in the field of attention mechanisms, and its variants have shown outstanding performance in the fields of Computer Vision and Programming Languages. Therefore, to address the aforementioned deficiencies, this paper investigates Transformer-based methods for Knowledge Graph Completion, introducing a domain-aware Transformer model, NATLP, and a Transformer model that combines graph paths and local neighborhoods, TKGE-PN. The main contributions of this paper are as follows:

(1) To address the issue of limited expressive power and inability to capture mutual dependencies among neighboring entities in knowledge graph embedding methods based on GNNs, this paper proposes NATLP, a neighborhood aware transformer for Link Prediction that constructs and models the impact of different relations on entity message passing based on relation-specific neighboring entity information. The Transformer's self-attention mechanism is also modified to enable it to perceive local neighborhood graph structure data and learn mutual dependencies among neighboring entities.

(2) To address the issue that methods based on Graph Neural Networks, as well as NATLP, fail to learn long-distance dependencies, this paper introduces a transformer-based knowledge graph embedding Model combining Graph Paths and Local Neighborhood, TKGE-PN. The model first samples multiple graph paths centered on the target entity using a biased random

walk algorithm. Then, it employs a Transformer-based graph path encoding module to learn and vectorize the long-distance dependencies within these paths. Lastly, a local neighborhood encoding module aggregates all the graph path vectors along with local neighborhood information to score triples. During the graph path encoding process, the model's ability to learn long-distance information is enhanced by employing a masked entity relation prediction task.

(3) Experiments on two mainstream benchmark datasets, FB15K-237 and WN18RR, demonstrated the effectiveness of the proposed NATLP and TKGE-PN models along with the key designs.

Key words: Knowledge Graph, Knowledge Graph Completion, Knowledge Graph Embedding, Transformer, Graph Path, Local Neighborhood

目 录

图 清 单

表 清 单

第一章　绪论

1.1　研究背景与意义

论文选题来源于实验室承担的国家重点研发计划课题，本文主要研究高效的知识图谱补全方法，为知识图谱构建和和下游任务的应用提供技术支撑。

知识图谱 (Knowledge Graph, KG) 是知识库的一种主要形式，是由事实三元组（头实体、关系、尾实体）表示的结构化知识的集合，在多个人工智能领域中得到了广泛的应用，例如语义搜索、问答和推荐系统。主流的开放知识图谱包括 FreeBase，Wikidata，DBpedia 和 YAGO 等，它们通常包含由数十亿个实体和关系所构建的海量事实。然而，由于现实世界不断发展带来的知识的动态变化，大多数知识图谱、即使是大规模知识图谱也难以囊括所有的知识，这限制了知识图谱在现实世界中的应用。因此，近年知识图谱补全 (Knowledge Graph Completion, KGC)，又称链路预测任务，成为了知识图谱领域的热门研究方向，尝试在给定事实三元组中的头（尾）实体和关系的情况下，自动预测缺失的尾（头）实体。

知识图谱补全需要挖掘图谱中隐藏的语义信息，但是知识图谱中的事实三元组一般是以文本形式进行储存的，无法直接利用，需要首先寻找一种合适的方式来对语义信息进行表达。传统方法一般通过特征工程进行，效率低且可移植性较差，因此不少研究者投入了自动化知识补全的研究。

目前，知识图谱补全的主流解决方案是知识图谱嵌入 (Knowledge Graph Embedding, KGE)，又称知识图谱表示学习[1](Knowledge Graph Representation Learning)。它的核心思想是将图谱中的实体和关系投影到低维向量空间中，通过预先设计好的得分函数 (Scoring Function) 评估事实三元组的合理性，并基于知识图谱中的已有事实，最大化对正确事实三元组的预测概率。通过这种方式获得的嵌入表示不仅可以用于知识图谱补全，还能够用于语义搜索、问答和推荐系统等下游任务中。

传统的知识图谱表示学习方法主要考虑如何在单纯的三元组上进行学习，但这种方式存在较大的缺陷：忽略了知识图谱本身的图结构信息。基于图神经网络的模型通过学习中心实体的局部邻域结构一定程度上解决了以上的问题，获得了更加优秀的性能，但依然存在不足：首先图神经网络的网络结构较浅，限制了模型的表达能力；另外，基于

图神经网络的方法随着网络层数的提升会遭遇过度平滑[2] 的问题，导致其只能捕捉单个实体附近 1-2 跳内的局部邻域信息，缺乏利用长距离依赖的能力。针对以上问题，本文研究基于 Transformer 的知识图谱补全方法。Transformer[3] 被公认为是建模序列数据的最强大的神经网络，不少工作致力于研究将 Transformer 网络应用到知识图谱嵌入工作中。本文研究利用 Transformer 强大的表达能力，结合知识图谱中图结构信息，实现更加准确的知识图谱补全，支持稀土催化材料知识图谱构建和应用。

1.2　国内外研究现状

1.2.1　相关研究发展现状

知识图谱 (Knowledge Graph, KG) 的现代含义由 2012 年谷歌知识图谱[4] 的发布而确立，它是知识库的一种主要表现形式，是由事实三元组（头实体、关系、尾实体）表示的结构化知识的集合。图谱中的节点为实体，表示现实世界中的具体事物；图谱中的节点为关系，表示实体之间的联系。目前知识图谱已经在多个人工智能领域中得到了广泛的应用，例如语义搜索[5]、问答[6] 和推荐系统[7]。主流的开放知识图谱包括 FreeBase[8]，Wikidata[9]，DBpedia[10] 和 YAGO[11] 等，它们通常包含使用数十亿个实体和关系构建的大量事实。然而，即使是大规模知识图谱也不可避免的是不完全的，缺乏部分事实，这限制了知识图谱在现实世界中的应用。因此，近年知识图谱补全又称链路预测任务，成为了知识图谱领域的热门研究方向，尝试在给定事实三元组中的头（尾）实体和关系的情况下，自动预测缺失的尾（头）实体。

目前，知识图谱嵌入是知识图谱补全任务的主流解决方案, 它将图谱中的实体和关系转化为低维向量空间中的向量，尽可能地保留其原始的结构性质，并用得分函数估计事实三元组正确的概率。现阶段对于知识图谱嵌入算法的研究，根据方法的核心思想和实现方式的不同，可以划分为传统的知识图谱嵌入方法，基于图神经网络的知识图谱嵌入方法，基于 Transformer 的知识图谱嵌入方法和融合多源信息的知识图谱嵌入方法。

传统的知识图谱嵌入方法仅独立研究知识图谱中的事实三元组，主要包含基于翻译的方法、基于张量分解的方法和引入神经网络后的基于多层神经网络方法、基于卷积神经网络的方法。

基于翻译的方法是最早被提出的一类知识图谱嵌入方法，最早起源于 2013 年的

TransE[1] 模型，核心思想是将知识图谱中的关系视为一个实体到另一个实体的翻译，又被称为平移距离模型。由于 TransE 无法有效建模知识图谱中的一对多、多对一、多对多关系，后续基于 TransE 进行改进并衍生出了如 TransH[12]、TransR[13]、TransD[14] 等模型，不断丰富模型的表达能力。基于翻译的方法最大的优点在于其模型结构简单、计算速度快、易于理解且可解释性较强，但另一方面浅层的模型结构也限制了该类方法的表达能力。

以 RESCAL[15] 为代表的基于张量分解的方法则将整个知识图谱表示为一个高维的稀疏张量，通过对其进行张量分解来获得实体和关系的嵌入。RESCAL 用一个维度为 $N \times N \times M$ 的张量来表示一个实体数量为 N, 关系数量为 M 的知识图谱，其中第 i 行 j 列深度为 k 的元素值为 1 时表示实体 i 和实体 j 之间存在关系 k。通过张量分解，模型最终能够得到用一维向量表示的实体嵌入和用二维矩阵表示的关系嵌入。继 RESCAL 之后，基于张量分解的思想提出的 DistMult[16]、ComplEx[17]、ANALOGY[18] 等一系列模型分别从强化模型表达能力以及压缩模型参数两方面对 RESCAL 模型进行了改进。DistMult 将关系嵌入进行了简化，选用了对角矩阵替代了 RESCAL 的二维矩阵，降低了模型的复杂度并获得了更优的性能；ComplEx 则将模型从实数域扩展到了复数域，提高了模型的表达能力。总的来说，基于张量分解的方法可解释性较强，并能够捕捉到实体和关系之间的双线性关系，但和基于翻译的方法类似，浅层的模型结构很难有效的学习图谱中蕴含的复杂信息，模型表达能力较弱。

而随着神经网络的发展，大量基于神经网络的知识图谱嵌入方法开始涌现。使用神经网络进行知识图谱嵌入能够建立更加复杂的模型，自动学习知识图谱当中蕴含的特征，模型的表达能力更强，更加充分地学习和表达知识图谱中的信息。这其中最早提出的是基于多层神经网络的方法，SME[19]、NTN[20]、MLP[21] 等模型直接使用多层的神经网络去拟合知识图谱，以事实三元组的嵌入作为模型的输入，输出三元组正确的概率。这类方法相较之前没有神经网络结构的方法在性能上有了提升，但网络结构相对简单，可解释性较差。

而受到计算机视觉领域的研究方法的启发，随后有不少工作尝试将卷积引入知识图谱嵌入领域，大量基于卷积神经网络的方法被提出，其中最具代表性的方法为 ConvE[22]。ConvE 将事实三元组中的头实体和关系的一维向量嵌入，重组为二维张量并对其进行卷积操作，将结果向量化之后经过神经网络层，随后和候选实体的嵌入进行点乘，输出事

实三元组的正确概率。基于 ConvE 的思想，有不少方法提出了进一步的改进。ConvR[23] 使用关系嵌入构造卷积核，减少了网络的参数；ConvKB[24] 通过在实体和关系的相同维度上进行卷积，能够捕获在实体和关系之间相同维度上的联系；InteractE[25] 则将重组后二维张量修改为棋盘式，大大提升了头实体和关系之间的交互。

以上提到的知识图谱嵌入方法研究的对象是知识图谱中独立的三元组，这导致这些模型忽略了知识图谱的结构信息，因此被统一归类为传统的知识图谱嵌入方法。例如，这些方法没有办法感知到头实体的邻居实体，无法充分利用每个实体丰富的邻域结构，不仅链路预测的性能受到限制，而且也缺乏嵌入空间的可解释性。而基于图神经网络 (Graph Neural Network, GNN) 的知识图谱嵌入方法则利用图卷积神经网络来捕获图谱中的图结构信息，中心实体接受来自邻居实体与邻居关系的消息，并依此对实体和关系的嵌入表示进行更新。

R-GCN[26] 是第一个利用图卷积神经网络学习知识图谱表示的方法，整体采用编码器-解码器架构。编码器部分通过图神经网络对图结构进行建模，在 R-GCN 的信息传播过程中，中心实体会接受来自出边、入边和自循环边三个方向的信息；通过多次信息传播模型能够获得多阶邻居的信息。解码器部分则基于编码的信息对三元组进行打分。后续提出的基于图神经网络的方法沿用了 R-GCN 的编码器-解码器架构，并在此基础上进行改进。SACN[27] 模型基于关系类型将实体的邻域划分为带权值的子图进行聚合。TransGCN[28] 提出了两种基于翻译的思想的编码器同时学习实体和关系嵌入，分别用于实数域和复数域。

而收到自然语言处理和计算机视觉领域中注意力机制的成功的启发，有不少工作尝试将注意力机制引入到了基于图神经网络的知识图谱嵌入方法中来并取得了不错的效果。KBGAT[29] 是首个在知识图谱嵌入领域应用图注意力网络的方法，模型能够自动分辨出哪些邻居实体的信息对于中心实体是更加重要的。RGHAT[30] 将注意力机制进行了进一步的细分，引入了关系注意力机制和实体注意力机制，实现了更细粒度的建模。EIGAT[31] 则通过随机游走算法引入了全局实体重要性，将局部注意力机制和知识图谱的全局信息进行了结合。

基于图神经网络的方法通过对实体的邻域结构进行学习从而获得了阶段性的成功，性能普遍优于传统的知识图谱嵌入模型。但是图神经网络的表达能力虽然相较于传统方法的多层神经网络和卷积神经网络有了较大的提升，但是依然不足以充分学习知识图谱

中的语义信息。针对这个问题，许多研究者尝试引入表达能力更强的架构。Transformer[3]是注意力机制方面里程碑式的工作，被认为是建模序列数据的最强大的神经网络，基于 Transformer 的模型变体在计算机视觉和编程语言领域中也表现出了出色的性能，因此目前有不少工作致力于研究将 Transformer 结构应用到知识图谱嵌入工作中，这些方法的特点是对 Transformer 的编码方式和注意力机制进行改造，使得模型能够学习到知识图谱中的事实三元组和结构信息并进行预测，典型方法有 HittER[32] 和 Relphormer[33]。HittER 采用分层 Transformer 架构对实体的局部邻域进行了建模。Relphormer 提出了一种用于知识图谱嵌入的 Transformer 架构变体，并提出了一种 Triple2Seq 序列化算法来解决知识图谱中边和节点的异构性问题。

融合多源信息的知识图谱嵌入方法则是在以上几类算法的基础上利用更多的额外信息来进行知识图谱嵌入，例如图路径、文本描述、实体类别或者时间顺序等。这些信息能够帮助模型从不同的维度对知识图谱进行建模，提高知识图谱补全的效果。

基于图路径的方法尝试利用知识图谱中的图路径信息来捕获实体与实体之间的长距离依赖。在知识图谱中，图路径被定义为图谱中的实体-关系链，例如 (Yao Ming, Born In, Shanghai, City Of, China)。对于每个待预测的事实三元组，这类方法一般通过随机游走等方式获得若干条图路径，并基于图路径学习实体和关系的嵌入。TransE-Comp[34] 和 PTransE[35] 尝试建模两个实体之间的图路径上多跳关系构成的复合关系。Chain[36] 和 RSN[37] 则对循环神经网络 (Recurrent Neural Network, RNN) 进行了改造，以学习图路径上的所有相邻实体和关系之间的依赖。Interstellar[38] 分析了图路径信息对知识图谱嵌入的重要性，并将图路径学习问题定义为循环神经网络架构搜索问题，并设计了一种特定于知识图谱嵌入领域的混合搜索算法以及搜索空间。

除了结构信息之外，知识图谱中的每个实体和关系一般都有名称和对应的文本描述，蕴含对应的自然语言语义。NTN[20] 对文本描述的词向量进行平均来初始化实体的向量表示。而随着最近几年预训练语言模型 (Pre-trained language models, PLM) 的火热发展，也有不少方法探究利用 PLM 来完成知识图谱补全任务。KG-BERT[39] 通过 BERT[40] 来利用知识图谱中的三元组的文本信息进行知识图谱补全。LMKE[41] 提出了一种对比学习框架，提高了负采样的效率，大大缩短了基于预训练语言模型的方法的训练和推理的时间，并提高了补全性能。TagReal[42] 利用 PLM 结合语料库信息搜索进行知识图谱补全，并开发了自动的提示 (Prompts) 生成和信息检索方法，使 TagReal 能够自动生成高

质量提示支持 PLM 搜索相关信息，这使得模型在 PLM 缺乏某些领域知识时更加实用。

此外，考虑到知识的时效性，部分方法在知识图谱中引入了时序信息。TTransE[43] 模型在传统的基于翻译的方法 TransE 的基础上进行了改进，引入了额外的时序信息；TeAST[44] 采用了阿基米德螺旋时间线来对时序知识图谱进行编码，将时序知识图谱的四元组补全问题转化为了三阶张量补全问题，降低了复杂度。

总的来说，融合多源信息的知识图谱嵌入方法通过引入额外的信息获得了更好的知识图谱嵌入效果，但是往往需要额外的数据准备工作，成本较高。有些知识图谱甚至无法获取对应的信息，可移植性较差；另外数据的质量也会对模型的性能造成影响。

1.2.2　对比分析

通过调研国内外知识图谱嵌入方法，本文对于各类知识图谱嵌入方法进行了总结与对比，各种方法的优缺点如表1所示。

表 1 各类知识图谱嵌入方法对比分析

方法类型	优点	缺点
基于翻译	结构简单，计算速度快，可解释性较强	模型表达能力弱
基于张量分解	可解释性较强，能够捕捉实体与关系之间的双线性关系	模型表达能力较弱
基于多层神经网络	表达能力相比于之前的方法更强	容易出现过拟合的问题，嵌入的维度对性能的影响较大
基于卷积神经网络	实体和关系之间的交互得到了增强，参数数量较少	没有利用到知识图谱的图结构信息
基于图神经网络	能够学习到实体的局部邻域信息，模型性能相较于传统方法得到了提高	模型的表达能力不足以充分学习知识图谱的语义信息，另外捕获长距离信息的能力不足
基于 Transformer	通过自注意力机制和更复杂的网络结构获得了更强大的模型表达能力	模型复杂度高，不适用于大规模知识图谱，无法直接利用图结构信息
融合多源信息	将现有方法与额外的信息进行结合，获得了更好的知识图谱嵌入效果	需要额外的数据准备工作，成本较高；信息的质量对模型的性能影响较大，可移植性相对较差

1.3 研究目标及内容

1.3.1 研究目标

本课题的研究目标是设计基于 Transformer 的知识图谱嵌入模型，利用 Transformer 模型的强大表达能力来学习实体和关系的合适嵌入表示，对知识图谱进行自动化补全。本课题针对传统知识图谱嵌入和基于图神经网络的方法表达能力弱、图信息利用不足、无法捕获长距离信息乃至全局信息的问题，研究如何基于 Transformer 网络和知识图谱的特点，采用合适的方式采样和编码知识图谱中局部邻域和图路径两类图结构，并进行综合利用以充分发挥 Transformer 网络强大的表达能力，最终得到能够尽可能拟合现有图谱的合适表示。

1.3.2 研究内容

针对本课题的研究目标，本课题的主要研究路线如图1所示。本课题的研究内容主要包括以下几个方面：

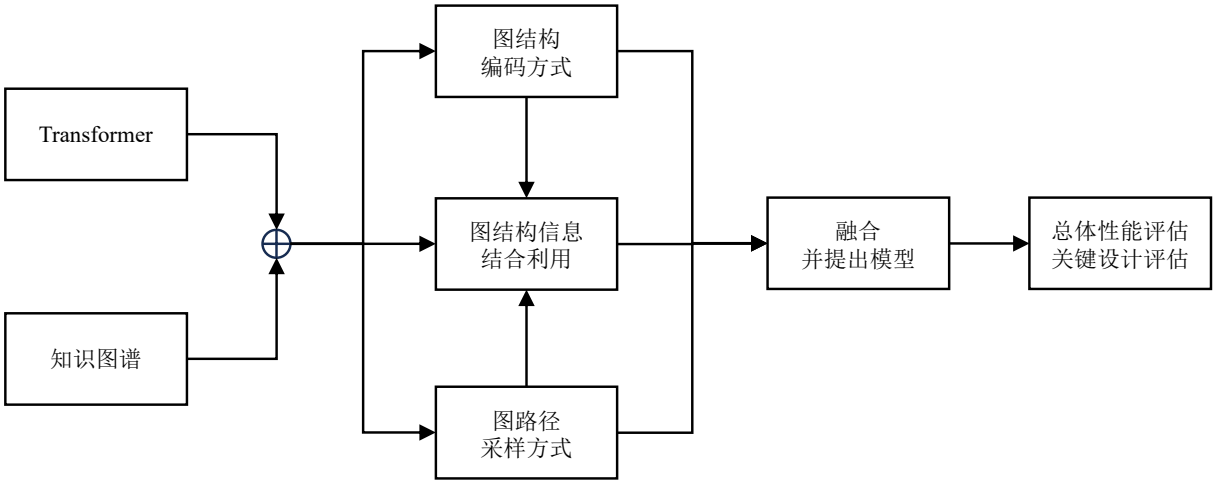

图 1 研究路线示意图

（1）基于 Transformer 的模型对于图结构的捕获研究

在 Transformer 中，任意一个位置都能直接感知到其他位置的输入信息，这导致模型无法直接捕捉到输入之间的相对位置关系，因此在处理序列数据时，采用的方式一般是为每个位置的输入添加对应的位置编码，标识输入与输入之间的前后位置关系。但在知识图谱中节点并不是顺序排列的，因此本文的主要研究内容之一就是设计一种合适方案让 Transformer 模型能够学习到知识图谱的拓扑结构，实现对知识图谱结构的感知。

（2）图路径采样算法研究

本课题计划通过对知识图谱中的图路径信息学习来挖掘实体与实体之间的长距离依赖，因此为了提升模型性能，对于当前的待遇测事实三元组，如何采样到高质量的图路径是首先需要解决的问题。因此本文计划研究设计合适的采样策略，实现高效的图路径采样，提高模型捕获长距离依赖的能力。

（3）不同图结构信息的结合方案研究

基于图神经网络的模型通过聚合消息的方式实现了对于中心实体局部邻域结构的感知，但无法捕捉实体之间长距离的依赖；基于图路径的方法能够挖掘到更远距离的依赖，但忽略了实体丰富的局部邻域。因此，本文的主要研究内容之一是设计合适的模型结构实现对于以上两类图结构信息的综合利用，实现对于图谱中长短距离信息的捕捉。

（4）实验与验证

在完成以上研究内容，实现完整的知识图谱补全模型之后，设计相应实验方案，通过平均排名、平均倒数排名等指标在主流公开数据集上与基线模型进行性能对比，验证本文提出的模型的有效性；并且通过设计合适的消融实验，验证模型关键设计的有效性。

1.3.3　论文组织安排

本文对基于 Transformer 的知识图谱补全方法进行研究，论文内容总共分为五个章节以及总结与展望部分，各个章节的内容安排组织如下：

第一章绪论首先介绍了论文的背景与意义，随后对知识图谱以及知识图谱算法的国内外研究进展进行了简单介绍，并进行了各类方法的对比与总结。随后明确了论文的研究目标与研究路线，概述了论文的主要研究内容。最后介绍了论文的组织安排。

第二章介绍了论文中方法所涉及到的相关理论基础。首先介绍了注意力机制与 Transformer 模型架构，其次对 Transformer 模型在图学习领域中的应用进行了概述，最后对不同类别的知识图谱嵌入方法进行了介绍，包括部分模型的核心思想以及数学公式。

第三章首先对 Transformer 模型在知识图谱嵌入领域的应用存在的困难进行了分析，随后介绍了提出的基于邻域感知的 Transformer 模型，给出了符号定义以及模型的总体架构，并对其中的关键设计结构强化的自注意力机制进行了说明。

第四章首先指出了现有的基于图结构信息的方法的缺点，随后介绍了提出的结合

图路径和局部邻域的 Transformer 知识图谱嵌入模型，说明了模型的总体架构以及各个模块的设计方案，包括基于有偏随机游走的图路径采样算法、图路径信息与局部邻域信息的结合方案以及掩蔽实体关系预测任务。

第五章为实验与验证部分，首先说明了实验采用的数据集、选取的进行对比的基线模型、实验环境以及采用的评估策略等基本情况，随后对实验结果进行了介绍和分析，包括本文提出的模型与基线模型的总体性能对比、关键模块消融实验的结果，超参数对于模型的性能影响等。

总结与展望部分对本文的研究内容进行了回顾与总结，并对未来可能的研究方向进行了展望。

第二章　相关理论基础

本章对论文中所涉及到的相关理论基础进行了介绍。首先介绍了注意力机制与Transformer模型架构，其次对Transformer模型在图表示学习领域中的应用进行了概述，最后对不同类型的知识图谱嵌入方法的核心思想和数学公式进行了说明，包括传统的知识图谱嵌入方法、基于图神经网络的知识图谱嵌入方法、基于图路径的知识图谱嵌入方法以及基于Transformer的知识图谱嵌入方法。

2.1　注意力机制与Transformer网络

深度学习中的注意力机制(Attention Mechanism)灵感来源于人类的视觉和认知系统。在推理过程中，注意力机制动态的为输入数据分配不同的权重，使模型能够自动地学习并选择性地关注输入中的重要信息，提高模型的性能和泛化能力。注意力机制最早被用于处理计算机视觉任务，后来在多个领域中得到了应用，例如自然语言处理和推荐系统等。

谷歌的研究团队于2017年提出的Transformer[3]网络则是注意力机制方面里程碑式的工作。Transformer网络设计之初主要用于处理序列数据，在Transformer出现之前，序列数据的处理通常依赖于循环神经网络(RNN)及其变体，例如长短期记忆网络(LSTM)和门控递归单元(GRU)。RNN及其变体在处理序列数据时能够保持一定程度的历史信息，但存在一定的问题：由于对序列数据进行逐步处理，RNN在训练过程中容易出现梯度消失或者梯度爆炸的问题，特别是在处理长序列时；逐步处理也限制了模型的并行计算能力，导致训练效率低下；此外，尽管LSTM和GRU通过特殊的门控机制改善了长距离依赖问题，但当序列长度过高时，模型依然难以捕捉到距离较远的依赖关系。

Transformer网络通过使用自注意力(Self-Attention)机制解决了上述问题，在自注意力机制中，输入数据中的任意一个位置都能够直接感知到其它位置的信息，因此相比于传统的RNN结构，自注意力能够更加直接地捕捉到序列中长距离的依赖关系；自注意力机制允许模型在处理数据时并行计算各个位置的注意力分数，与RNN逐步计算的方式相比，可以显著提高模型的计算效率；自注意力机制通过学习输入序列中不同位置之间的动态相关性，能够根据特定的任务自适应地调整注意力分布。

Transformer 网络中的自注意力机制的核心为缩放点积注意力机制，结构如图2所示。

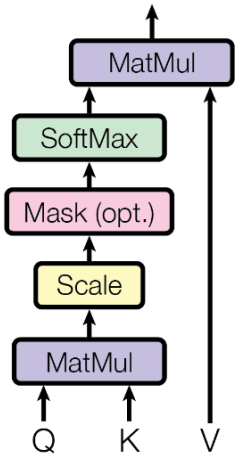

图 2 缩放点积注意力机制

具体来说，假设模型的输入为 X，首先模型将会通过线性变化生成输入对应的查询向量 Q，键向量 K 以及值向量 V:

$$Q = XW^Q, K = XW^K, V = XW^K \tag{2.1}$$

其中 W^Q、W^K、W^K 是可学习的参数矩阵。

随后模型会将查询向量 Q 和键向量 K 进行点积并乘以缩放因子 $\frac{1}{\sqrt{d_k}}$ 获得注意力分数，将其进行归一化处理转化为概率分布，用作权重对值向量 V 进行加权平均和，最终得到缩放点积注意力机制对应的输出，其中 d_k 为键向量的维度：

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.2}$$

进一步的，为了让模型能够同时关注来自不同维度的信息，并稳定自注意力的学习过程，Transformer 采用了多头注意力机制，通过不同的参数矩阵将 Q、K、V 映射到不同的向量空间下并计算缩放点积注意力，将结果进行拼接获得最终的输出，如图3所示。

具体来说，对于 h 个独立的注意力头，有：

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(head_1, ..., head_h)W^O \\ \text{where } head_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.3}$$

其中 QW_i^Q、KW_i^K、VW_i^V 为将 Q、K、V 映射到第 i 个向量空间的参数矩阵，Concat 为拼接操作。

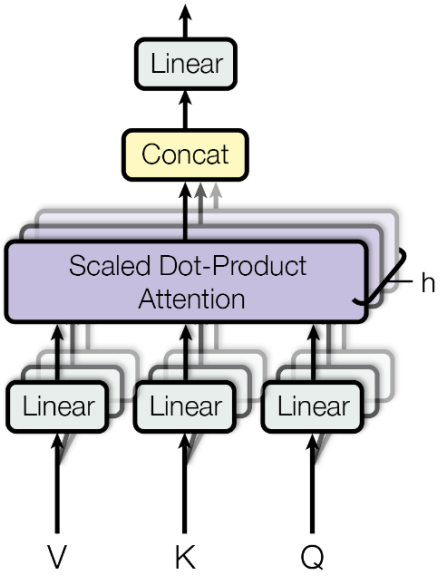

图 3　多头注意力机制

2.2　基于 Transformer 的图表示学习方法

图被广泛用于连接数据的网络结构表示，在社交系统、生态系统、生物网络、知识图谱等领域中都有广泛的应用。图表示学习方法将图的特征转化为低维嵌入空间中的向量。由于 Transformer 在计算机视觉和自然语言处理等领域展现除了出色的性能，近来，已经有大量的基于 Transformer 的模型被用于编码图结构数据。GraphTrans[45] 利用 Transformer 的自注意力机制学习图中的长距离的成对关系，并设计了一种读出机制以获得全局图嵌入。Grover[46] 设计了节点级、边级和图级的自监督任务，能够从未标记的数据中学习图的结构和语义信息。Graphormer[47] 对 Transformer 的注意力计算方式进行了改造，并从数学上证明了许多流行的图神经网络变体可以被视为 Graphormer 的特殊情况。

Graphormer 认为 Transformer 设计之初是为了建模序列数据，为了让其能够在图结构上实现最好效果，关键是要将图的结构信息恰当的融合到模型之中。Graphormer 结合了几种有效的结构编码方法来利用这些信息。

Graphormer 认为 Transformer 模型中的注意力机制会基于节点的语义相似度计算注意力分布，因此提出了基于节点的出度和入度中心性编码来捕获知识图谱的节点重要

性，其中 x_i 是节点表征，$z^-_{deg^-(v_i)}$ 和 $z^+_{deg^+(v_i)}$ 是代表节点出度和入度的可学习表征：

$$h_i^{(0)} = x_i + z^-_{deg^-(v_i)} + z^+_{deg^+(v_i)} \tag{2.4}$$

Transformer 模型在处理长序列文本时，会通过位置编码来学习文本之间的相对位置信息，但图不是序列数据，因此需要重新设计空间编码，加入到注意力运算中：

$$A_{ij} = \frac{(h_i W_Q)(h_j W_K)^T}{\sqrt{d}} + b_{\Phi(v_i,v_j)} \tag{2.5}$$

这种编码方式的优势在于 $b_{\Phi(v_i,v_j)}$ 提供了一个对于图中的每个节点的全局的空间信息。

关系信息对于图中的节点表征至关重要，Graphormer 模型为了将关系信息加入到注意力中，引入了关系编码 c_{ij}, 表示的是节点 v_i 和 v_j 之间最短路径上所有关系表征的平均值，如下图公式所示：

$$A_{ij} = \frac{(h_i W_Q)(h_j W_K)^T}{\sqrt{d}} + b_{\Phi(v_i,v_j)} + c_{ij}, \text{ where } c_{ij} = \frac{1}{N}\sum_{n=1}^{N} x_{e_n}(w_n^E)^T \tag{2.6}$$

2.3 知识图谱嵌入方法

2.3.1 传统的知识图谱嵌入方法

传统知识图谱嵌入方法的研究对象是知识图谱中独立的三元组，集中于利用嵌入空间中的显式几何特性来捕捉实体之间的不同关系，基于翻译的方法、基于张量分解的方法和神经网络中的基于多层神经网络的、基于卷积神经网络的方法均属于此类。

基于翻译和基于张量分解的表示学习属于较早提出的方法，它们模型结构简单，没有神经网络结构，计算速度快，可解释性较高。因此，在各个领域中的应用都十分广泛。

TransE[1] 模型于 2013 年被提出，是基于翻译的知识图谱嵌入方法的起源。TransE 的核心思想是将图谱中的关系视为嵌入空间内实体到实体的翻译，具体来说，TransE 将实体和关系投影到相同的向量空间中，对于正确的事实三元组 (h, r, t)，头实体嵌入 $\boldsymbol{h}$ 和关系嵌入 $\boldsymbol{r}$ 的相加结果应该尽可能得接近尾实体嵌入 $\boldsymbol{t}$，即：

$$\boldsymbol{h} + \boldsymbol{r} \approx \boldsymbol{t} \tag{2.7}$$

而在 TransE 的训练和评估过程中，对于训练集中的每一个正样本事实三元组 (h,r,t)，TransE 会通过随机替换头尾实体的方式，生成对应的负样本参与训练，这样的策略也成为了后续许多知识图谱嵌入方法的训练和评估策略。最终，对于扩充后的训练集中的每一个三元组 (h,r,t)，TransE 模型会计算 $\boldsymbol{h}+\boldsymbol{r}$ 和 $\boldsymbol{t}$ 之间的距离的 L2 范数作为衡量标准：

$$d_r(h,t)=\|\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{t}\| \tag{2.8}$$

对于正样本，TransE 期望得到的距离尽可能得小，负样本得到的距离尽可能地大，最终得到 TransE 模型的损失函数：

$$\mathcal{L}=\sum_{(h,r,t)\in S}\sum_{(h',r,t')\in S'_{(h,r,t)}}[\gamma+d_r(h,t)-d_r(h',t')] \tag{2.9}$$

其中 S 代表正样本集，$S'_{(h,r,t)}$ 为生成的对应的负样本集。TransE 方法最大的缺点对复杂关系建模效果不佳，例如一对多、多对一以及多对多关系，容易把不同实体学习成相近的嵌入，随后的一系列基于翻译的方法针对这个缺点提出了很多的改进方式。

在基于张量分解的方法中，最为经典的是 RESCAL[15] 方法。给定一个知识图谱，RESCAL 将其形式化为一个三阶张量 $\mathcal{X}\in\mathbb{R}^{N\times N\times M}$，其中 N 是实体的数量，M 为关系种类的数量，张量的每一个切片 $\mathcal{X}_k$ 对应于第 k 种关系的邻居矩阵，代表知识图谱中该种关系下实体之间的连接情况，如果实体 i 与实体 j 之间存在关系 k，那么 $\mathcal{X}_{ijk}$ 的值为 1，否则为 0。

RESCAL 假设每个实体 i 都可以通过一个向量 $a_i\in\mathbb{R}^R$ 来表示，每类关系 k 由一个二维矩阵 $R_k\in\mathbb{R}^{R\times R}$ 表示，其中 R 为预定义的嵌入维度。RESCAL 通过以下公式对张量 $\mathcal{X}$ 进行分解：

$$\mathcal{X}_k\approx AR_kA^T,\ \text{for } k=1,\ldots,m \tag{2.10}$$

其中 $A=[a_1,\ldots,a_N]$ 是实体嵌入矩阵。

RESCAL 模型的训练目标是最小化张量 $\mathcal{X}$ 与通过学习到的实体和关系表示重建的张量之间的差异，因此模型的损失函数为：

$$\min_{A,R_k}=f(A,R_k)+g(A,R_k) \tag{2.11}$$

其中有：

$$f(A, R_k) = \frac{1}{2}\left(\sum_k \left\|\mathcal{X}_k - AR_kA^T\right\|_F^2\right) \tag{2.12}$$

g 为模型的正则项：

$$g(A, R_k) = \frac{1}{2}\lambda\left(\|A\|_F^2 + \sum_k \|R_k\|_F^2\right) \tag{2.13}$$

其中 $\|\cdot\|_F$ 为 Frobenius 范数，λ 为正则化参数，用于防止过拟合。RESCAL 首次提出了基于张量分解的知识图谱嵌入方法，但也存在缺陷，用二维矩阵表示关系的方法使得 RESCAL 在处理大规模知识图谱时的计算成本和存储需求可能非常巨大，因此后续提出的一系列基于张量分解的方法在 RESCAL 的基础上进行了改进。

早期的基于神经网络的方法主要是采用多层神经网络来尝试直接拟合知识图谱。NTN[20] 模型是一种用于知识图谱嵌入的神经网络架构，于 2013 年被提出。NTN 通过引入张量运算和多层神经网络进行非线性特征变换来学习实体关系的语义信息，提高了模型的表达能力，克服了之前的知识图谱嵌入方法的限制。

NTN 模型采用低维向量代表实体，实体之间的关系则用一个三维张量进行表示。和 RESCAL 模型类似，NTN 引入了双线性张量操作。它通过在实体之间进行张量运算来捕捉实体之间的复杂交互关系，但是在 NTN 中还使用了基于多层神经网络的框架，相比于 RESCAL，提供了更加丰富的线性特征表达能力。

在训练和评估过程中，NTN 通过以下公式来对事实三元组进行打分：

$$f(h, r, t) = u_r^T \tanh\left(v_h^T M_r v_t + W_r^1 v_h + W_r^2 v_t + b_r\right) \tag{2.14}$$

其中，v_h 和 v_t 分别为头实体和尾实体的嵌入向量，u_r 为关系 r 的权重向量，M_r 为关系 r 对应的三维张量，W_r^1 和 W_r^2 为多层神经网络对应的线性变换矩阵，b_r 为偏置项，tanh 为引入非线性的双曲正切激活函数。此外，NTN 还采用了预训练的词向量来对实体和关系嵌入进行初始化，提升了模型的效果。

而随着卷积神经网络在计算机视觉领域大获成功，在知识图谱嵌入领域中也涌现出了以 ConvE[22] 为代表的基于卷积神经网络的知识图谱嵌入方法。ConvE 将设计和关系的嵌入表示重塑为二维矩阵，并用卷积神经网络来捕获实体和关系之间的复杂交互。具体来说，ConvE 模型的具体计算步骤如下：

首先，对于给定的头实体 h 和关系 r，ConvE 将它们的嵌入重塑为二维形式，并拼接成一个二维矩阵：

$$\left[\overline{\boldsymbol{h}}; \overline{\boldsymbol{r}}\right] \tag{2.15}$$

其中 $[;]$ 表示拼接操作，$\overline{\boldsymbol{h}}$ 和 $\overline{\boldsymbol{t}}$ 表示头实体嵌入 $\boldsymbol{h}$ 和关系嵌入 $\boldsymbol{t}$ 重塑后的二维形式。

接下来 ConvE 对得到的二维矩阵进行卷积操作，应用 ReLU 激活函数 f 后得到的特征图为：

$$f\left(\left[\overline{\boldsymbol{h}}; \overline{\boldsymbol{r}}\right] * \omega\right) \tag{2.16}$$

其中 $*$ 表示卷积操作。随后 ConvE 将卷积层得到的特征图进行向量化，将得到的向量经过全连接层和 ReLU 激活函数后，形成最终的特征表示：

$$f\left(\text{vec}\left(f\left(\left[\overline{\boldsymbol{h}}; \overline{\boldsymbol{r}}\right] * \omega\right)\right) W\right) \tag{2.17}$$

在预测阶段，模型将得到的特征表示与每个候选尾实体的嵌入进行点积得到对应的相似度得分：

$$\psi_r(\boldsymbol{h}, \boldsymbol{t}) = f\left(\text{vec}\left(f\left(\left[\overline{\boldsymbol{h}}; \overline{\boldsymbol{r}}\right] * \omega\right)\right) W\right) \boldsymbol{t} \tag{2.18}$$

其中 $\boldsymbol{t}$ 为尾实体嵌入。

最后，ConvE 模型采用交叉熵损失函数对模型进行训练：

$$\mathcal{L}(p, t) = -\frac{1}{N} \sum_i (t_i \cdot \log(p_i) + (1 - t_i) \cdot \log(1 - p_i)) \tag{2.19}$$

其中 p 为事实三元组正确的概率，有：

$$p = \text{sigmoid}(\psi_r(\boldsymbol{h}, \boldsymbol{t})) \tag{2.20}$$

2.3.2　基于图神经网络的知识图谱嵌入方法

基于图神经网络的知识图谱嵌入方法是近些年知识图谱嵌入领域的一个重要发展方向，这类方法主要利用图神经网络的能力来学习图谱中实体和关系的嵌入表示。相比于传统的知识图谱嵌入方法，图神经网络天然适合处理图结构类型的数据，通过聚合一

个实体周围的邻居节点信息来学习中心嵌入，这一过程能够捕捉到图谱中的局部拓扑信息，从而能够更好地表达实体之间的关系。此外，传统的嵌入模型往往只能考虑直接的实体关系，而图神经网络可以通过多层网络堆叠来捕捉实体之间的多跳路径信息，从而实现对更远距离实体关系的建模。

R-GCN[26] 第一个在知识图谱嵌入领域应用图神经网络的方法。传统的图卷积网络主要设计用来处理无向图或者单一关系类型的图，但是这种方法无法直接应用于知识图谱，因为其忽略了图谱中边上的多种关系类型信息。而 R-GCN 则将图卷积神经网络扩展到了可以处理具有多种关系类型的图数据。R-GCN 为图谱中的每种关系类型引入了一个单独的权重矩阵，每个关系类型在聚合邻居信息时能产生不同的影响，从而能够学习到每种关系特定的模式。R-GCN 还在知识图谱中为每个实体添加了一个特殊类型的自环边，允许每个节点保留自身的信息，自环边在更新节点表示时作为单独的一种关系处理，也有自己的权重矩阵。

R-GCN 整体为编码器-解码器架构，使用改造后的图卷积神经网络作为编码器，使用 DistMult 方法作为链路预测任务的解码器，使用 softmax 函数来作为实体分类任务的解码器。

具体来说，R-GCN 的传播层用数学公式表示如下：

$$h_i^{l+1} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^l h_j^l + W_0^l h_i^l\right) \tag{2.21}$$

其中 h_i^{l+1} 是第 i 个节点在经过第 $l+1$ 层传播层更新后的表示，$\mathcal{N}_i^r$ 是节点 i 通过关系 r 相连接的邻居节点集合，$c_{i,r}$ 是归一化因子，一般设置为 $c_{i,r} = |\mathcal{N}_i^r|$。

获得聚合后的实体表示后，R-GCN 使用 DistMult 方法对三元组进行打分:

$$f(s,r,o) = e_s^T R_r e_o \tag{2.22}$$

训练用的损失函数为：

$$\mathcal{L} = -\frac{1}{(1+\omega)\,|\hat{\varepsilon}|}\sum_{(s,r,o,y)\in\mathcal{T}} y\log l(f(s,r,o)) + (1-y)\log(1-l(f(s,r,o))) \tag{2.23}$$

其中 ω 为 R-GCN 为每个正样本生成的负样本数量。后续提出的基于图神经网络的方法在 R-GCN 的基础上进行了改进，基本沿用了 R-GCN 的编码器-解码器架构。SACN[27] 将实体的邻域划分为带权值的子图进行聚合。TransGCN[28] 提出了两种基于翻译的思想的编码器，分别用于实数域和复数域。

受到注意力机制在计算机视觉领域和自然语言处理领域的成功的启发，还有一部分基于图神经网络的方法尝试融合注意力机制。KBGAT[29] 首次将图注意力网络用于知识图谱嵌入任务，RGHAT[30] 使用实体和关系分层的方式对邻居的注意力进行了细分。

RGHAT 采用了一个创新的层次化注意力机制来充分利用实体的本地邻域信息，主要分为两层：关系级注意力根据不同关系对于中心实体重要性的不同为实体的每个邻接关系分配不同的权重；实体级注意力在关系级注意力的基础上进一步评估每个关系下邻居实体的重要性并分配对应的注意力分数。

对于中心实体 h，邻接关系 r 的关系级注意力 $\alpha_{h,r}$ 的计算方式如下：

$$\mathbf{a}_{h,r} = \mathbf{W}_1 \left[\mathbf{h} \| \mathbf{v}_r\right] \tag{2.24}$$

$$\alpha_{h,r} = \text{softmax}_r(\mathbf{a}_{h,r}) = \frac{\exp(\sigma(\mathbf{p} \cdot \mathbf{a}_{h,r}))}{\sum_{r' \in \mathcal{N}_h} \exp(\sigma(\mathbf{p} \cdot \mathbf{a}_{h,r'}))} \tag{2.25}$$

其中 $\mathbf{h}$ 为实体 h 的嵌入，$\mathbf{W}_1$、$\mathbf{v}_r$ 和 $\mathbf{p}$ 为可训练的参数，其中 $\mathbf{v}_r$ 是关系特定的。

在关系级注意力的基础上，RGHAT 进一步计算邻居节点 t 在关系 r 下对于中心实体 h 的实体级注意力 $\beta_{r,t}$：

$$\mathbf{b}_{h,r,t} = \mathbf{W}_2 \left[\mathbf{a}_{h,r} \| \mathbf{t}\right] \tag{2.26}$$

$$\beta_{r,t} = \text{softmax}_t(\mathbf{b}_{h,r,t}) = \frac{\exp(\sigma(\mathbf{q} \cdot \mathbf{b}_{h,r,t}))}{\sum_{t' \in \mathcal{N}_{h,r}} \exp(\sigma(\mathbf{q} \cdot \mathbf{b}_{h,r,t'}))} \tag{2.27}$$

之后将关系级注意力与实体级注意力相乘，RGHAT 得到 (h, r, t) 在所有邻居三元组中的注意力得分 $\mu_{h,r,t}$：

$$\mu_{h,r,t} = \alpha_{h,r} \cdot \beta_{r,t} \tag{2.28}$$

获得注意力得分之后，RGHAT 在每一层中对邻居信息进行聚合，结合中心实体自

身的嵌入得到该层的输出：

$$\hat{\mathbf{h}} = \sum_{r \in \mathcal{N}_h} \sum_{t \in \mathcal{N}_{h,r}} \mu_{h,r,t} \mathbf{b}_{h,r,t} \tag{2.29}$$

$$\mathbf{h}^{'} = \frac{1}{2} \left(\sigma(\mathbf{W}_3(\mathbf{h} + \hat{\mathbf{h}})) + \sigma(\mathbf{W}_4(\mathbf{h} \odot \hat{\mathbf{h}})) \right) \tag{2.30}$$

RGHAT 还采用了多头注意力机制：

$$\mathbf{h}^{'} = \Big\Vert_{k=1}^{K} \mathbf{h}_k^{'} \tag{2.31}$$

通过采用层次化的注意力机制，RGHAT 能够更精细地对实体的邻域进行建模，不仅聚焦于关系的重要性，还考虑了实体之间的语义贡献，提供了模型的可解释性。

2.3.3 基于图路径的知识图谱嵌入方法

和基于图神经网络的方法利用中心实体的局部邻域进行链路预测不同，基于图路径的方法则尝试学习图谱中的路径信息来捕获实体与实体之间的长距离依赖，PTransE[35]、RSN[37] 和 Interstellar[38] 模型均属于此类，其中 PTransE 是其中较早的方法。

PTransE 建立在 TransE 模型的基础上，并增加了考虑实体间多步关系路径的能力。PTransE 的主要贡献在于它不仅仅考虑实体之间的直接关系，还考虑通过其他实体间接连接的路径，从而捕获更丰富的语义信息。

在 PTransE 中，路径由一系列关系构成，定义为：

$$\mathbf{p} = \mathbf{r}_1 \circ \cdots \circ \mathbf{r}_l \tag{2.32}$$

其中 $\circ$ 表示关系的串联。PTransE 模型尝试学习路径 $\mathbf{p}$ 的表示，可以通过多种方式实现，例如通过关系向量的相加、乘积或者利用循环神经网络来进行学习，目的是最小化以下距离函数：

$$E(h,p,t) = \|\mathbf{p} - (\mathbf{t} - \mathbf{h})\| = \|\mathbf{p} - \mathbf{r}\| = E(p,r) \tag{2.33}$$

PTransE 模型将直接关系和间接关系一同考虑，所以基于翻译的思想，以下距离函

数也需要同时最小化：

$$E(h,r,t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\| \tag{2.34}$$

最终，PTransE 模型的损失函数为：

$$L(\mathbf{S}) = \sum_{(h,r,t)\in\mathbf{S}} \left[L(h,r,t) + \frac{1}{Z} \sum_{p\in P(h,r)} R(p|h,t)L(p,r) \right] \tag{2.35}$$

其中 $R(p|h,t)$ 为路径的置信度，$L(h,r,t)$ 和 $L(p,r)$ 为基于间隔的损失函数：

$$L(h,r,t) = \sum_{(h',r',t')\in\mathbf{S}^-} \left[\gamma + E(h,r,t) - E(h',r',t') \right]_+ \tag{2.36}$$

$$L(p,r) = \sum_{(h,r',t)\in\mathbf{S}^-} \left[\gamma + E(p,r) - E(p,r') \right]_+ \tag{2.37}$$

通过结合直接关系和通过多步路径发现的间接关系，PTransE 能够捕捉实体之间更为复杂的交互模式，从而提高知识图谱补全任务的性能。但是，PTransE 没有考虑到路径中的实体信息，因此后面提出的基于路径的知识图谱补全方法对 PTransE 进行了改进。

2.3.4　基于 Transformer 的知识图谱嵌入方法

基于 Transformer 的知识图谱嵌入方法的核心思路是利用 Transformer 强大的表达能力来挖掘图谱中的语义和结构信息，以学习知识图谱的嵌入表示。Transformer 模型的自注意力机制能够有效地捕捉实体之间的复杂关系和交互。还可以通过在 Transformer 中集成额外信息进一步加强实体和关系的表示。代表方法有 HittER[32] 和 Relphormer[33]。

Relphormer 是一种为知识图谱嵌入任务而设计的基于 Transformer 的神经网络架构。为了解决知识图谱中实体与关系节点的异构性，Relphormer 把实体和关系均视为相同的节点，有 $V = \mathcal{E} \cup \mathcal{R}$ 为节点集合，其中 $\mathcal{E}$ 为实体集合，$\mathcal{R}$ 为关系集合，并用邻接矩阵 $\mathbf{A} \in \{0,1\}^{|V|\times|V|}$ 来表示节点之间的连接关系，事实三元组表示为：$\mathcal{T} = (v_s, v_p, v_o)$，中心节点的局部邻域为：

$$\mathcal{T}_G = \mathcal{T}_c \cup \mathcal{T}_{context}, \text{ where } \mathcal{T}_{context} = \{\mathcal{T}_i \in \mathcal{N}\} \tag{2.38}$$

由于 Transformer 是序列数据模型，Relphormer 利用 Triple2Seq 算法对中心节点的局部邻域进行随机采样，抽取一部分邻接三元组并转化对应的为序列数据。

由于知识图谱是图结构，为了防止局部邻域转化为序列数据后损失知识图谱的结构信息，Relphormer 提出了一种结构强化的自注意力机制，当计算注意力分数时，额外添加一个偏差项：

$$a_{ij} = \frac{(\mathbf{h}_i \mathbf{W}_Q)(\mathbf{h}_j \mathbf{W}_K)}{\sqrt{d}} + \phi(i,j) \tag{2.39}$$

$$\phi(i,j) = f_{structure}(\tilde{\mathbf{A}}^1, \tilde{\mathbf{A}}^2, \cdots, \tilde{\mathbf{A}^m}) \tag{2.40}$$

其中 $\tilde{\mathbf{A}}$ 是归一化后的邻接矩阵，$\tilde{\mathbf{A}^m}$ 是 $\tilde{\mathbf{A}}$ 的 m 次方，指在 m 步内节点的连通性。

对于一些较为密集的知识图谱，每步训练中对局部邻域的随机采样过程随机性较大，可能会导致训练过程中的不一致性。为了避免这个问题，Relphormer 利用上下文对比策略来克服不稳定性，使用不同训练步骤中相同三元组的采样的局部邻域的内容来强制模型进行类似的预测，最小化以下损失函数：

$$\mathcal{L}_{contextual} = -\log \frac{\exp(\text{sim}(\boldsymbol{c}_t, \boldsymbol{c}_{t-1})/\tau)}{\exp(\text{sim}(\boldsymbol{c}_t, \boldsymbol{c}_{t-1})/\tau) + \sum_j \exp(\text{sim}(\boldsymbol{c}_t, \boldsymbol{c}_j)/\tau)} \tag{2.41}$$

其中 τ 为温度系数，$\boldsymbol{c}_t$ 为第 t 步采样到的样本的表示，$\text{sim}(\boldsymbol{c}_t, \boldsymbol{c}_{t-1})$ 为 $\boldsymbol{c}_t$ 与 $\boldsymbol{c}_{t-1}$ 之间的余弦相似度：$\frac{\boldsymbol{c}_t^T \boldsymbol{c}_{t-1}}{\|\boldsymbol{c}_t\| \cdot \|\boldsymbol{c}_{t-1}\|}$。

受掩码语言模型例如 BERT 的启发，和之前的知识图谱嵌入方法不同，Relphormer 采用了一种新的训练策略：随机掩盖输入序列中的特定节点，然后对其进行预测。具体来说，在训练过程中，随机遮掩中心三元组头实体或者尾实体，并利用剩余的节点序列 $\mathcal{T}_M$ 和上下文邻接矩阵 $\mathbf{A}_G$ 的情况下，预测缺失的部分 Y:

$$\begin{aligned} \mathcal{T}_M &= MASK(\mathcal{T}_G) \\ Relphormer(\mathcal{T}_M, \mathbf{A}_G) &\rightarrow Y \end{aligned} \tag{2.42}$$

最终，Relphormer 模型的损失函数为：

$$\mathcal{L}_{all} = \mathcal{L}_{MKM} + \lambda \mathcal{L}_{contextual} \tag{2.43}$$

其中 $\mathcal{L}_{MKM}$ 和 $\mathcal{L}_{contextual}$ 分别为掩盖预测任务和上下文对比任务的损失函数。

2.4　本章小结

本章对论文中所涉及到的相关理论基础和关键技术进行了介绍，方便后续章节的说明。首先对注意力机制与 Transformer 模型的原理进行介绍，然后介绍了 Transformer 网络在图表示学习中的应用，最后介绍了知识图谱嵌入方法，包括不利用图结构的传统的知识图谱嵌入方法，以及基于图神经网络的知识图谱嵌入方法、基于图路径的知识图谱嵌入方法和基于 Transformer 的知识图谱嵌入方法。

第三章　基于邻域感知的 Transformer 模型

本章主要对基于邻域感知的 Transformer 模型 NATLP 的总体设计和模块的具体实现进行了介绍。主要包括对现存基于图神经网络方法存在问题的分析、模型的总体框架设计、关系特定的邻居实体信息构造设计以及融合图结构信息的自注意力机制的改进。

3.1　现有问题描述和分析

图卷积神经网络 GCN 于 2017 年被提出，对原先图神经网络中基于谱空间的的图卷积算子进行了优化，降低了模型的复杂度，由此引发了图神经网络的研究热潮。图神经网络迅速成为了图结构数据处理的重要方式，在社交网络[48]、推荐系统、知识图谱等多个领域都有着重要应用。近几年，图神经网络的应用是知识图谱嵌入领域非常重要的进展。图卷积神经网络能够直接处理图结构数据并捕捉知识图谱中的拓扑结构，通过聚合邻居节点的信息，图神经网络可以有效地学习知识图谱中节点（实体）和边（关系的嵌入表示）。相较于之前的方法，基于图神经网络的知识图谱嵌入方法获得了很大的性能提升。

然而，受限于本身的网络结构，在进行知识图谱嵌入时，基于图神经网络的方法依然存在不足，导致其性能受限。首先，图卷积神经网络采用聚合邻居节点的方法来更新中心实体的表示，在这个过程中，模型只考虑了邻居节点和中心实体之间的连通性，却忽略了不同邻居节点之间的可能也存在直接连接，各个邻居节点传递的信息之间是互不感知、互相独立的，这样的聚合方式没有将邻居节点信息之间的相互依赖纳入考虑；其次，图神经网络采用的消息传递模式整体模型结构比较简单，使模型的表达能力受到了限制，在挖掘图谱中实体和实体、实体与关系之间的复杂交互上存在困难。

而在以上两个方面，Transformer 网络存在巨大的优势。首先，通过构造查询向量、键向量和值向量来进行注意力的计算以及采用多头注意力机制，相比于图神经网络，Transformer 能够更加高效地挖掘输入之间各个维度的复杂交互；同时，通过调整模型的层数、头的数量或是隐层的维度大小，Transformer 可以很容易地适应处理不同规模和复杂度的知识图谱的需求。

其次，Transformer 网络的自注意力机制能够有效地捕捉序列中任意两个元素之间

的全局依赖关系。在知识图谱嵌入的场景之中，这意味着模型在挖掘局部邻域的结构信息时，除了邻居节点和中心节点之间的依赖之外，还能够同时学习到邻居节点之间的长距离依赖，捕获邻居节点传递的信息之间的相互影响。此外，Transformer 架构还支持模型的预训练和迁移学习，可以首先在一个大规模的综合性知识图谱上进行预训练，然后迁移到特定领域的知识图谱上，通过这样的方式，可以减少模型对标注数据的依赖，提高模型在特定任务上的表现。

但是，虽然 Transformer 网络在自然语言处理（NLP）领域已经取得了巨大成功，但将 Transformer 网络直接应用到知识图谱嵌入领域时依旧会遇到一系列挑战和困难。图具有非欧几里得结构，是一种无序的数据结构，这与 NLP 中处理的序列数据（一维结构）有本质区别。Transformer 网络原本是为处理序列数据设计的，它使用位置编码来保留序列中元素的顺序信息。但是，在知识图谱中，节点之间的关系是通过边来定义的，并且没有固定的顺序，因此 Transformer 网络无法直接使用位置编码来捕捉节点间的结构关系。部分方法例如 MAGNN[49] 选择通过随机游走的方式来将图数据转化为序列数据来处理，但这样的方式会导致图结构信息的失真。

此外，在知识图谱中，边，即关系，反映着实体和实体之间不同的交互方式，蕴含着丰富的语义信息。两个实体之间连接的关系不同，传递的信息可能是千差万别的，因此如何采用合适的方式来对利用关系信息，体现关系对于消息传递的影响十分重要。但标准的 Transformer 模型并没有直接的方式来编码和使用边的信息。部分利用 Transformer 来进行知识图谱表示学习的模型例如 Relphormer[33] 将知识图谱中的实体和关系视为地位相同的节点，采用同样的方式进行处理，这样的方式虽然解决了边的表示问题，但没有考虑到知识图谱中实体和关系的差异性，没有考虑到关系对于实体消息传递的独特作用。

为了解决上述问题，本章提出了一种基于邻域感知的 Transformer 模型用于链路预测任务 (Neighborhood Aware Transformer for Link Prediction, NATLP)。首先，在模型输入信息构造阶段，为了充分建模不同关系对于实体传递消息的影响，模型基于关系生成特定的网络参数，实现关系特定的邻居信息构造。其次，为了让 Transformer 能够更好地处理图结构数据，模型对 Transformer 的自注意力机制进行了改造，提出了一种融合图结构的自注意力机制，使得 Transformer 能够学习到输入消息之间的互相依赖。

3.2 NATLP 模型设计

3.2.1 符号定义

为了方便说明论文提出的 NATLP 模型的实现细节，本节首先对模型中的关键概念和相关的数学符号进行了定义，具体内容参见表2。

表 2 NATLP 模型中的符号定义

符号	说明
$\mathcal{G}$	知识图谱
$\mathcal{E},\mathcal{R},\mathcal{T}$	实体集合、关系集合、边集合
$\mathcal{G}'$	拓展后的知识图谱
$\mathcal{R}'$	拓展后的关系集合
$\mathcal{T}^{-1}$	逆关系边集合
$\mathcal{T}'$	拓展后的边集合
$(s,r,?)$	待遇测的三元组
s	头实体即中心实体
o	尾实体即目标实体
e	实体
r	关系
r^{-1}	关系 r 的逆关系
$\boldsymbol{s},\boldsymbol{o}$	头实体嵌入和尾实体嵌入
$\boldsymbol{e},\boldsymbol{r}$	实体嵌入和关系嵌入
d	嵌入维度
ϕ_{chk}	棋盘式特征重组
$\circledast$	循环卷积操作
$f(\cdot)$	ReLU 激活函数
$vec(\cdot)$	二维张量转化为一维向量
k_{size}	卷积核的边长
n_{conv}	卷积核的数量
ω_r	特定于关系 r 的卷积层参数
$\mathbf{W}_r$	特定于关系 r 的全连接层参数
$\mathbf{W}_{conv}$	卷积层参数生成网络
$\mathbf{W}_{fc}$	全连接层参数生成网络
$\boldsymbol{r}_{global}$	全局关系嵌入
Re3D$(\cdot)$	一维向量转化为三维张量
Re2D$(\cdot)$	一维向量转化为二维张量
$\boldsymbol{m}_{e,r}$	邻居实体 e 通过关系 r 传递的信息

表 2 NATLP 模型中的符号定义

符号	说明
$\boldsymbol{e}_{cls}$	特殊嵌入 Class Token
$\mathbf{TE}$	类型嵌入
a_{ij}	第 i 个输入和第 j 个输入之间的注意力得分
$dis(e_i, e_j)$	实体 e_i 与实体 e_j 之间最短路径的距离
$deg(e)$	实体 e 的节点度数
$\boldsymbol{o}_t$	模型预测的候选实体的嵌入
$*$	普通卷积操作
σ	sigmoid 激活函数
p	三元组正确概率
L	模型损失
t_i	第 i 个三元组的标签

知识图谱是表示为 (s, r, o) 的“头实体-关系-尾实体”事实三元组的集合，所有的这些事实三元组连接起来构成了一个异构图，即为知识图谱，表示为 $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$，其中 $\mathcal{E}$ 集合，$\mathcal{R}$ 为关系集合，$\mathcal{R}$ 为实体和关系构成的边集合。由于知识图谱中的关系具有方向性，为了确保信息能够在两个相连的实体之间进行双向的流通，本文在知识图谱中为每个事实三元组 (s, r, o) 创建了对应的逆三元组 (o, r^{-1}, s)，其中 r^{-1} 是关系 r 对应的逆关系。因此，关系集合、边集合以及知识图谱被拓展为：

$$\mathcal{R}' = \mathcal{R} \cup \{r^{-1} | r \in \mathcal{R}\} \tag{3.1}$$

$$\mathcal{T}^{-1} = \{(o, r^{-1}, s) | (s, r, o) \in \mathcal{T}\} \tag{3.2}$$

$$\mathcal{T}' = \mathcal{T} \cup \mathcal{T}^{-1} \tag{3.3}$$

$$\mathcal{G}' = (\mathcal{E}, \mathcal{R}', \mathcal{T}') \tag{3.4}$$

知识图谱补全任务，即链路预测任务，是在给定待遇测三元组中的头实体 s 以及关系 r 的情况下预测缺失的尾实体 o，表示为 $(s, r, ?)$，或者是在给定尾实体 o 以及逆关系 r^{-1} 的情况下预测缺失的头实体 s，表示为 $(?, r^{-1}, o)$。为了方便说明，论文随后统一采用 $(s, r, ?)$ 的形式进行表述。

3.2.2 模型总体结构

本节主要对提出的用于链路预测的基于邻域感知的 Transformer 模型 NATLP 的总体结构进行介绍。NATLP 整体为编码器-解码器架构，编码器部分主要由关系特定的邻居信息构造模块和邻域感知 Transformer 模块组成，解码器部分则采用了基于卷积神经网络的知识图谱嵌入方法进行了实现。模型整体架构如图4所示。

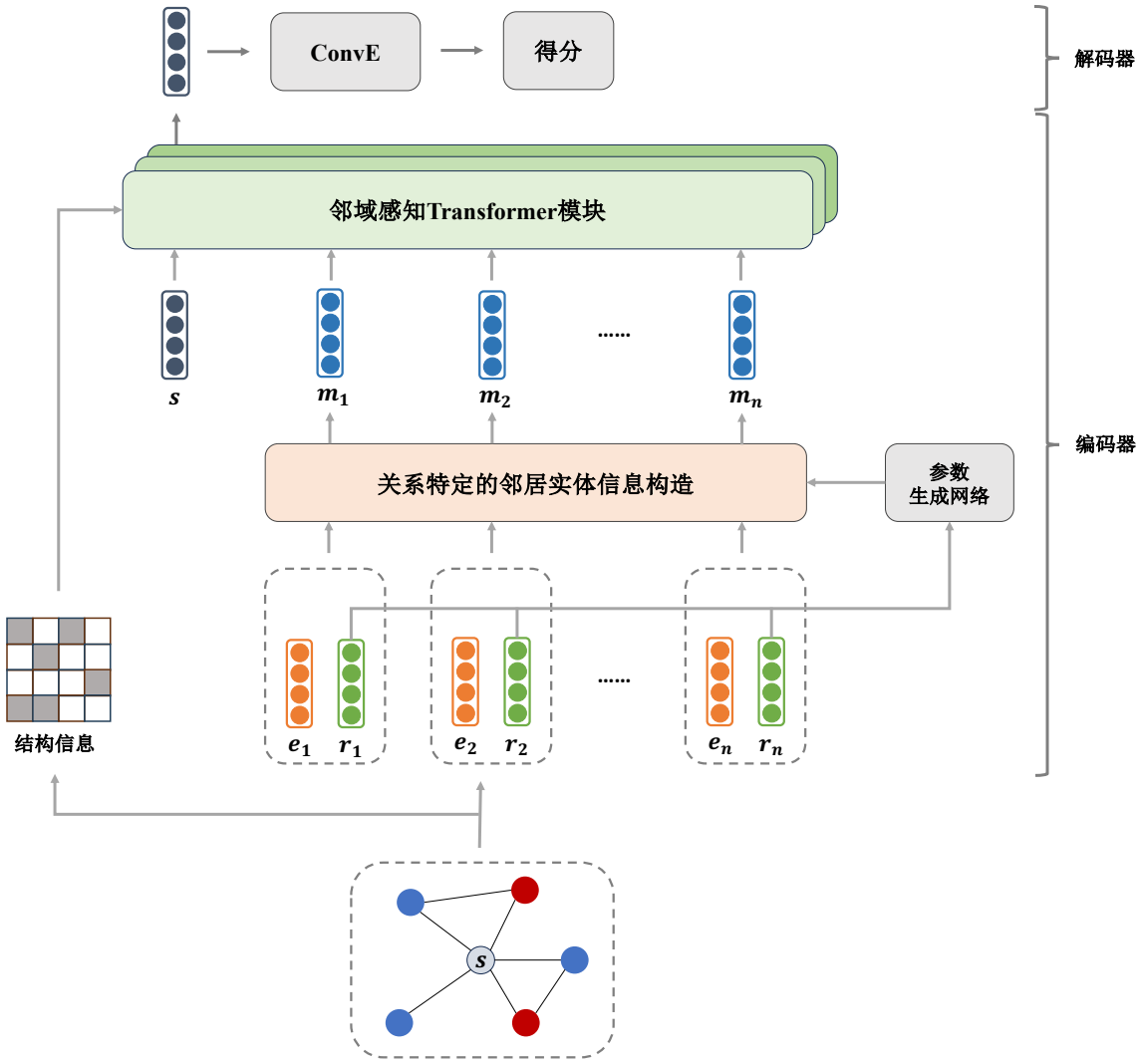

图 4 NATLP 模型整体架构

编码器的主要作用是将模型输入的实体和关系转化为对应的嵌入，并学习其中蕴含的语义信息和结构信息并编码成向量形式，是模型的核心部分。在 NATLP 中，模型的输入主要包括待预测的三元组及其局部邻域，编码器首先会根据中心实体和邻居之间相连的关系种类，为每一个邻居实体构造关系特定的邻居消息；随后的邻域感知 Transformer 模块综合学习构造的邻居信息、中心实体本身的信息以及局部邻域的结构信息，并完成编码。

解码器的主要任务则是根据编码器得到的知识表示，对下游任务的各项性能指标进行评测。根据下游任务的不同，模型可以采用不同的解码器进行解码。NATLP 采用了基于卷积神经网络的知识图谱嵌入方法 ConvE 作为解码器，来对事实三元组的正确概率进行评估，完成知识图谱补全。

3.2.3　关系特定的邻居实体信息构造

链路预测任务的目标是利用知识图谱中已有的事实去预测未知事实在知识图谱中的存在概率。为了能够充分利用邻域信息来帮助预测三元组中缺失的尾实体，NATLP 需要获得局部邻域中邻居实体向中心实体传递的信息。知识图谱中的关系反映着实体和实体之间不同的交互方式，但标准的 Transformer 模型没有办法直接对关系进行编码。为了解决这个问题，受到基于图神经网络的知识图谱嵌入方法中的消息传递模型的启发，NATLP 首先基于连接的关系完成邻居实体的消息构造后，再将消息传递到 Transformer 模型中进行学习。

但是，目前基于图神经网络的知识图谱嵌入方法中采用的消息构造函数存在着一些不足。本文调研了部分基于图神经网络的知识图谱嵌入方法采用的消息构造函数，具体内容见表3。

表 3 部分基于图神经网络的知识图谱嵌入方法采用的消息构造函数

知识图谱嵌入方法	采用的消息构造函数
R-GCN[26]	$\mathbf{W}_r e$
SACN[27]	$\mathbf{W} e$
Graph2Seq[50]	$\mathbf{W}_{in}\left[e_i, r_k, e_j\right]$ or $\mathbf{W}_{out}\left[e_i, r_k, e_j\right]$
CompGCN[51]	$\mathbf{W}_{dir(r)} e_i \star e_k$
KBGAT[29]	$\mathbf{W}\left[e_i, e_j, r_k\right]$
RGHAT[30]	$\mathbf{W}_2\left[\mathbf{W}_1\left[e_i, r\right], e_j\right]$

可以发现，除了 R-GCN[26] 之外，其余的方法对于实体通过不同关系传递的信息，采用的都是同样的网络参数进行编码。但是，同一个实体和不同的关系相连，表达的语义信息可能完全不一样。例如（姚明，出生于，上海）和（姚明，职业，篮球运动员），传递的信息就有着很大不同。采用同样的参数进行编码，会导致模型难以捕获实体中和不同关系相关的特定特征。针对关系的这个特点，R-GCN 模型为每个关系都定义了单独的网络参数，但是这样的方法也存在问题：一方面，每类关系的网络参数需要单独进

行学习，对于数量较少的关系可能会出现训练不充分的情况；另一方面，这样的方法会容易导致关系之间的内在相关性被忽略。TransCoRe[52] 对 TransE/TransH/TransR 学习到的关系嵌入进行了分析，发现关系之间的相关性通过嵌入表示上的低秩结构显示出来，即不同种类的关系之间存在某种共同的特点。

为了解决上述问题，实现捕获邻居实体中关系相关的特定特征的同时，兼顾不同类别关系之间的共通特征，NATLP 提出了一种关系特定的邻居实体信息构造方法，具体如图5所示。

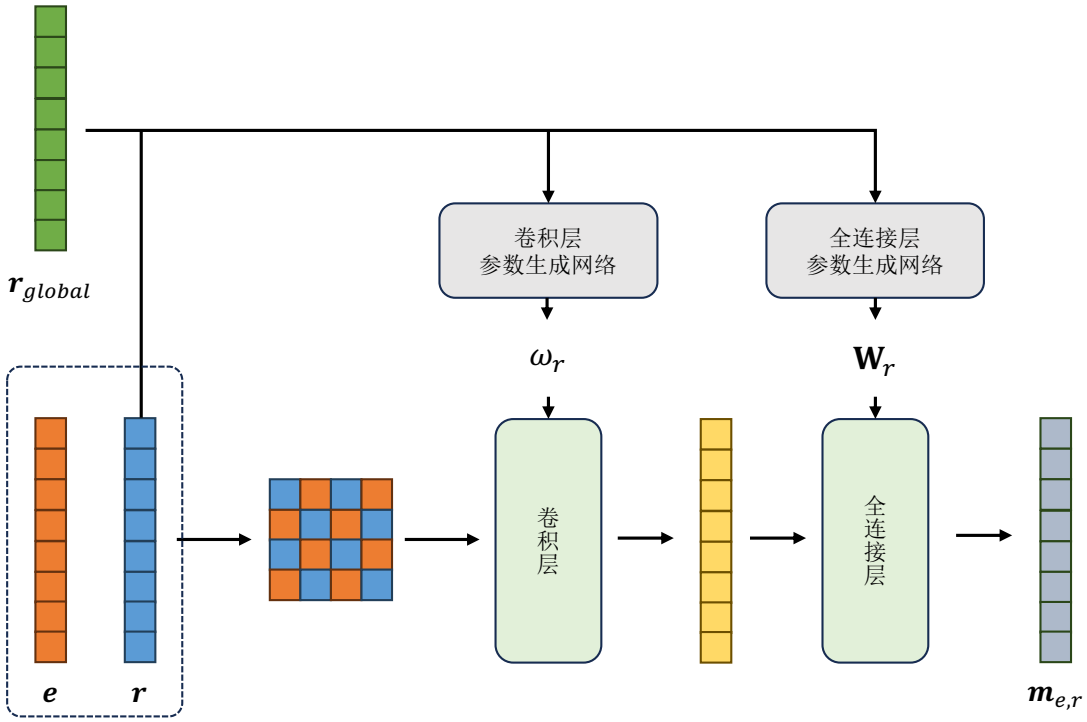

图 5 关系特定的邻居实体信息构造

首先，相比于大多数方法采用的实体和关系嵌入拼接之后再线性转换的构造方式，NATLP 模型选择将实体和嵌入重塑为二维张量之后再对其进行卷积操作。相比于线性转换，卷积神经网络更擅长捕捉局部模式，通过卷积操作，模型可以有效地提取实体和关系之间的局部交互特征；此外，由于权重共享的特性，卷积神经网络在模型参数上更加高效，使得模型能够用更少的参数完成信息的构建，减少模型过拟合的风险，加快模型训练的过程。而相比于一维卷积，二维卷积能够提升实体和关系嵌入之间的特征交互，从而更丰富的特征。为了进一步的提升实体和关系之间的交互，NATLP 采用了棋盘式的特征重组方式来进行二维张量的重塑，实体中的每一个特征分量能够和四个关系的特征分量进行交互，如图6所示:

$$\begin{matrix}[a & a & a & a & a & a & a & a] \\ & & & \times & & & & \\ [b & b & b & b & b & b & b & b]\end{matrix} = \begin{bmatrix} a & b & a & b \\ b & a & b & a \\ a & b & a & b \\ b & a & b & a \end{bmatrix}$$

图 6 棋盘式特征重组

具体的，给定一个邻居实体 e 和相连的关系 r，模型先采用棋盘式特征重组的方式将实体嵌入和关系嵌入重塑为二维张量：

$$\phi_{chk}\left(\boldsymbol{e}, \boldsymbol{r}\right) \tag{3.5}$$

其中 ϕ_{chk} 代表棋盘式特征重组，$\boldsymbol{e} \in \mathbb{R}^d$ 为实体 e 的嵌入表示，$\boldsymbol{r} \in \mathbb{R}^d$ 为关系 r 的嵌入表示，d 为嵌入的维度。

将实体和关系嵌入表示重塑为二维张量之后，NATLP 会对其进行循环卷积操作:

$$\phi_{chk}\left(\boldsymbol{e}, \boldsymbol{r}\right) \circledast \omega_r \tag{3.6}$$

其中 $\circledast$ 代表循环卷积。相比于普通的卷积操作，循环卷积能够捕捉更多的特征交互。卷积完成后，NATLP 将卷积的输出重组为一维向量，再经过线性变化之后就可以得到邻居实体向中心实体传递的信息 $\boldsymbol{m}_{e,r} \in \mathbb{R}^d$：

$$\boldsymbol{m}_{e,r} = f\left(vec\left(f\left(\phi_{chk}\left(\boldsymbol{e}, \boldsymbol{r}\right) \circledast \omega_r\right)\right) \mathbf{W}_r\right) \tag{3.7}$$

其中 $f(\cdot)$ 代表 ReLU 激活函数，$vec(\cdot)$ 代表将卷积的输出重整为一维向量的操作。

为了让模型能够充分捕获关系对于实体消息传递的影响，NATLP 采用参数生成网络来为卷积层和全连接层生成关系对应的特定参数 ω_r 和 $\mathbf{W}_r$。同时，为了显示地捕捉不同关系之间的共性，NATLP 引入了一个全局关系嵌入 $\boldsymbol{r}_{global}$ 参与网络参数的生成:

$$\omega_r = \text{Re3D}\left(\mathbf{W}_{conv}\left[\boldsymbol{r}; \boldsymbol{r}_{global}\right]\right) \tag{3.8}$$

$$\mathbf{W}_r = \text{Re2D}\left(\mathbf{W}_{fc}\left[\boldsymbol{r}; \boldsymbol{r}_{global}\right]\right) \tag{3.9}$$

其中 $\mathbf{W}_{conv} \in \mathcal{R}^{n_{conv} \times k_{size} \times k_{size} \times d}$ 为卷积层参数生成网络，n_{conv} 为卷积核的数量，k_{size} 为卷积核的边长，$\mathbf{W}_{fc}$ 为全连接层参数生成网络，Re3D$(\cdot)$ 和 Re2D$(\cdot)$ 代表将参数生成网络的输出重整为卷积层和全连接层参数需要的三维张量和二维张量的形式。$\boldsymbol{r}_{global}$ 为全局关系嵌入。通过全局关系嵌入 $\boldsymbol{r}_{global}$，模型能够捕捉到不同关系之间的共同特征，当部分关系种类训练数据较少时，模型通过全局关系嵌入也能获得不错的泛化能力。

3.2.4 邻域感知 Transformer 模块

完成关系特定的邻居实体信息构造之后，NATLP 下一步的任务是利用 Transformer 模型来挖掘局部邻域蕴含的语义和结构信息，并编码成向量形式，提供给解码器进行解码。

给定一个待遇测的事实三元组 $(s, r, ?)$，其中头实体即中心实体 s 有 n 个邻居实体，e_i、r_i 为中心实体的邻居实体和对应的相连的关系，有 $\forall i \in [1, n]$，$(s, r_i, e_i) \in \mathcal{T}'$，则在完成关系特定的邻居实体信息构造之后，邻域感知 Transformer 模块的输入可以表示为以下形式：

$$\mathbf{M}_{input} = [\boldsymbol{s}, \boldsymbol{m}_{e_1,r_1}, \boldsymbol{m}_{e_2,r_2}, ..., \boldsymbol{m}_{e_n,r_n}] \tag{3.10}$$

$$\Phi(e, r) = f\left(vec\left(f\left(\phi_{chk}\left(\boldsymbol{e}, \boldsymbol{r}\right) \circledast \omega_r\right)\right)\mathbf{W}_r\right) \tag{3.11}$$

$$\boldsymbol{m}_{e_i,r_i} = \Phi(e_i, r_i) \tag{3.12}$$

其中 s 为中心实体的嵌入表示，$\boldsymbol{m}_{e_i,r_i}$ 为实体 e_i 通过关系 r_i 传递的信息。

此外，为了防止在解码的时候模型对某个特定的输入具有偏向性，模型在输入序列的头部添加了一个特殊的嵌入 Class Token，表示为 $\boldsymbol{e}_{cls}$。Class Token 不基于任意的输入内容，在训练之前进行随机的初始化，并且随着网络的训练不断更新，能够在一定程度上编码整个知识图谱的统计特性。最终，在 Transformer 的输出中，Class Token 对应的输出向量被用作代表整个输入序列的特征表示，传递给解码器。为了帮助模型在 Class Token、中心实体嵌入表示和邻居实体传递的消息之间进行区分，受到 BERT[40] 模型的启发，NATLP 为以上三类输入分配了可学习的类型嵌入，则邻域感知 Transformer 模块

的最终输入可以表示为：

$$\mathbf{M}'_{input} = [\boldsymbol{e}_{cls}, \boldsymbol{s}, \boldsymbol{m}_{e_1,r_1}, \boldsymbol{m}_{e_2,r_2}, ..., \boldsymbol{m}_{e_n,r_n}] \tag{3.13}$$

$$\mathbf{M}_{input} = \mathbf{M}'_{input} + \mathbf{TE} \tag{3.14}$$

其中 $\mathbf{TE}$ 代表可学习的类型嵌入。

在完成输入的构造之后，NATLP 将利用 Transformer 的自注意力机制来学习输入中的信息。原始版本的 Transformer 的第 i 个输入和第 j 个输入之间的注意力分数 a_{ij} 计算公式为：

$$a_{ij} = \frac{(\boldsymbol{m}_i W_Q)(\boldsymbol{m}_j W_K)^T}{\sqrt{d}} \tag{3.15}$$

这样的计算方式给 Transformer 带来的最大优势是让其具备了捕捉输入中全局信息的能力。在 Transformer 的每一层中，所有的输入都能够接收并处理来自输入序列中任何位置的信息。然而，这样的方式也带来了副作用：输入序列中的结构信息丢失了，在处理邻居实体传递的信息时，模型无法捕捉到邻居实体之间的直接联系，因此模型需要想办法明确区分不同的位置信息或者分辨不同输入之间的位置相关性。在处理序列数据时，可以采用为不同位置的输入分配不同的位置向量的方法解决这个问题，但这种方法并不适合知识图谱这种非欧结构的数据。

为了让 Transformer 能够捕获中心实体局部邻域中的图结构数据，本文提出了一种节点距离编码，模型根据节点之间在图谱中的最短距离来辅助计算输入之间的注意力分数。一般来说，实体应该更关注距离较近的其他实体。具体来说，当计算注意力时，模型额外添加一个基于实体节点间最短距离的偏置项：

$$a_{ij} = \frac{(\boldsymbol{m}_i W_Q)(\boldsymbol{m}_j W_K)^T}{\sqrt{d}} + \frac{1}{dis(e_i, e_j)} \tag{3.16}$$

其中 $dis(e_i, e_j)$ 为知识图谱中实体 e_i 和 e_j 之间的最短路径的距离。通过添加额外的辅助项，距离越近的实体之间计算得到的注意力得分越高。

此外，在公式3.15中注意力分数是基于输入信息之间的语义相关性计算的，但是知识图谱中实体的节点度数也是重要的结构信息，它衡量了实体在知识图谱中的重要性。例如，在社交网络知识图谱中，拥有大量关注者的明星的权重应该更高。因此在注意力

计算中，节点度数也应该被纳入考虑。具体来说，NATLP 在注意力计算中额外添加一个节点度数的辅助项：

$$a_{ij} = \frac{(\boldsymbol{m}_i W_Q)(\boldsymbol{m}_j W_K)^T}{\sqrt{d}} + 1 - \frac{1}{\lg(deg_{e_i}) \cdot \lg(deg_{e_j})} \tag{3.17}$$

其中 deg_{e_i} 为实体 e_i 的节点度数。两个实体的节点度数越高，注意力得分越高。通过这样的方式，模型能够在注意力机制中同时捕获语义相关性和节点的重要性。

最终，领域感知 Transformer 模块的自注意力计算方式为：

$$a_{ij} = \frac{(\boldsymbol{m}_i W_Q)(\boldsymbol{m}_j W_K)^T}{\sqrt{d}} + \frac{1}{dis(e_i, e_j)} + 1 - \frac{1}{\lg(deg_{e_i}) \cdot \lg(deg_{e_j})} \tag{3.18}$$

模型取 Transformer 最后一层的输出中 Class Token 对应位置的输出向量 $\boldsymbol{T}_{cls}$ 作为整个编码器的最终输出。

3.2.5 基于卷积神经网络的解码器

解码器的主要任务是根据邻域感知 Transformer 模块的输出来计算待遇测三元组正确的概率，对链路预测任务的效果进行评估。在知识图谱补全任务中，一般采用传统的知识图谱嵌入方法作为解码器，它们结构简单，计算效率高，可解释性强。常见的解码器有基于翻译的方法如 TransE[1]、基于张量分解的方法 DistMult[16] 以及基于卷积神经网络的方法 ConvE[22]。在这之中性能最好，最常被使用的解码器是 ConvE，因此 NATLP 也采用 ConvE 作为解码器。

给定待遇测的事实三元组 $(s, r, ?)$ 和编码器的输出 $\boldsymbol{T}_{cls}$，ConvE 解码器先计算得到模型预测的候选实体的嵌入 $\boldsymbol{o}_t$:

$$\boldsymbol{o}_t = f\left(vec\left(f\left([\boldsymbol{T}_{cls}; \boldsymbol{r}] * \omega\right)\right) \mathbf{W}\right) \tag{3.19}$$

其中 $*$ 代表卷积操作。随后对于任意一个候选实体 e_t，模型将 $\boldsymbol{o}_t$ 与 e_t 的嵌入 $\boldsymbol{e}_t$ 进行点积后并经过 sigmoid 激活函数后得到 e_t 正确的概率：

$$p_{e_t} = \sigma(\boldsymbol{o}_t \cdot \boldsymbol{e}_t^T) \tag{3.20}$$

其中 $\sigma(\cdot)$ 为 sigmoid 激活函数。

获得所有候选实体的得分后，模型采用交叉熵损失函数计算任务损失：

$$L = -\frac{1}{N}\sum_{i} t_i log(p_i) + (1 - t_i) log(1 - p_i) \tag{3.21}$$

t_i 为第 i 个候选实体组成的三元组是否正确的标签，p_i 是模型预测的第 i 个候选实体组成的三元组是否正确的概率。

3.3　本章小结

本章对 NATLP 模型的整体架构和实现细节进行了详细介绍。首先对当前基于图神经网络的方法存在的部分问题以及 Transformer 在知识图谱嵌入领域中应用的限制进行了介绍；随后给出了模型中涉及到的数学符号的详细定义；之后介绍了模型的整体架构组成；最后，对 NATLP 中的关键设计细节进行了具体的说明，包括（1）关系特定的邻居实体信息构造，利用参数生成网络生成关系特定的网络参数，学习关系对于实体信息传递的作用，并利用全局关系嵌入捕捉不同关系之间的共性。（2）邻域感知 Transformer 模块，通过最短距离编码和度数编码在自注意力机制计算时融合图结构信息，捕捉邻居消息之间的互相依赖，更好地适应知识图谱的图结构形式。（3）基于卷积神经网络的解码器。利用基于卷积神经网络的方法 ConvE 进行解码，并计算任务的交叉熵损失。

第四章　NALTP 模型实验与验证

为了验证本课题提出的 NATLP 算法的有效性，本文进行了实验验证。本章首先对 NALTP 的实验方案设计进行了介绍，包括实验采用的数据集、实验环境、评估策略；用来对比的基线算法以及模型在两个数据集上的超参数设置，并对实验结果进行了详细的分析，包括整体性能分析、关键设计分析。

4.1　实验方案设计

4.1.1　实验数据集

为了对 NATLP 模型进行评估，本文在两个标准基准数据集 FB15k-237[53] 和 WN18RR[22] 上进行了实验。FB15k-237 是开源知识图谱 Freebase[8] 的子集，存储了有关电影、演员、奖项等等现实世界的常识信息的。WN18RR 则是开源知识图谱 WordNet[54] 的子集，包含了英文单词中的语义信息，例如同义、反义、单词概念的上下层等多种单词语义关系。为了避免测试集出现逆关系泄露的问题，两个数据集中所有的逆关系都已经被去除。两个数据集的统计数据如表4所示，其中训练集用于模型参数训练，验证集用于模型超参数调优，测试集用于模型性能评估。值得注意的是，这两个数据集中的关系数量存在差异，FB15k-237 包含了 237 种 Freebase 中的不同关系，而 WN18RR 的关系种类数量为 11，总的来说，相比于 FB15k-237，WN18RR 数据集更加的稀疏。

表 4 数据集统计信息

数据集	FB15k-237	WN18RR
# 实体数量	10541	40943
# 关系数量	237	11
# 训练集数量	272115	86835
# 验证集数量	17535	3034
# 测试集数量	20466	3134
# 实体平均度数	42.7	4.5

4.1.2　实验评估策略

知识图谱的链路预测任务被定义为实体排序预测任务。测试集中的每个三元组将在两种不同的场景中进行链路预测评估：给定头实体和关系下的尾实体预测 $(s, r, ?)$，以及给定尾实体和关系下的头实体预测 $(?, r, o)$。在实践中，头实体预测以 $(o, r^{-}1, ?)$ 的形式执行。预测时，待预测的头实体或者尾实体将被每个候选实体替换，并计算每个候选三元组的得分，随后所有的候选三元组将按照分数降序进行排序，以获得基本事实三元组的准确排名，并根据排名来对评估指标进行计算。

在知识图谱补全任务中，平均排名 (Mean Rank，MR)、倒数平均排名 (Mean Reciprocal Rank，MRR) 和前 N 名百分比 (Hits@n) 三种评估指标将用来评估模型的性能。假设测试集中待预测三元组的数量为 K，$rank_i$ 为第 i 个三元组的正确实体在所有候选实体中的排序位置，则平均排序 MR 的计算方式为：

$$MR = \frac{\sum_i rank_i}{K} \tag{4.1}$$

MR 代表了所有正确实体的平均排序位置，MR 越小，说明正确实体的得分越高，排名越靠前，模型的性能更好。但是 MR 存在的最大问题则是它对于不同排序位置的预测效果投入的关注是一样的，例如，假设有两个三元组，其中一个三元组的正确实体的排名从 105 上升到了 100，另一个三元组的正确实体的排名从 5 上升到了 1，他们对于 MR 指标的贡献是一致的，但是一般来说，在实验中我们认为从 5 到 1 的提升是更有意义的，平均排名指标 MR 则忽略了这一点。

平均倒数排名 MRR 对于这个问题进行了改进，MRR 计算的时候正确实体排名的倒数的平均值，而不是排名的平均值，具体计算公式为：

$$MRR = \frac{1}{K} \sum_i \frac{1}{rank_i} \tag{4.2}$$

通过这样的计算方式，排名越靠前的项会获得更大的权重占比，对于整体性能指标的贡献越大。和 MR 不同，MRR 越高，代表模型的性能越好。

前 N 名百分比 Hits@n 指的则是所有测试三元组中正确实体的排序处于前 n 名的比

例，计算方式为：

$$Hits@n = \frac{\sum_i [1\ if\ rank_i \leq n\ or\ 0]}{K} \tag{4.3}$$

指标越高，代表模型的性能越好。在知识图谱补全任务中，常用的前 N 名百分比指标包括 Hits@1，Hits@3 和 Hits@10。

此外，注意到对于一个待预测的三元组 $(s, r, ?)$，正确实体 o 的选项可能不止一个，例如（姚明，出生于，上海）和（姚明，出生于，中国）都是正确的事实。在计算评价指标时，其余的正确实体可能会导致当前测试三元组中的正确实体排名下降，对模型评估造成影响，因此本文和大多数基线模型类似，计算排序时除了基本事实三元组之外的所有正确三元组都被排除在排名之外。

4.1.3 实验环境

本文采用的实验环境中的服务器硬件和软件配置如表5所示。硬件配置方面，实验所采用的服务器的处理器规格为 Intel(R) Core(TM) i9-10920X CPU @ 3.50GHz，内存大小为 128G，使用的显卡型号为 GeForce RTX 4090，对应的显存大小为 24G；软件配置方面，使用的操作系统为 Ubuntu 5.4.0-74，采用的编程语言为 Python，版本为 3.10.9，使用的深度学习框架及其对应的版本为 Pytorch 1.13.1。

表 5 采用的实验环境

配置项目	版本/内容
处理器	Intel(R) Core(TM) i9-10920X CPU @ 3.50GHz
内存	128G
显卡	GeForce RTX 4090
显存	24G
操作系统	Ubuntu 5.4.0-74
编程语言	Python 3.10.9
深度学习框架	Pytorch 1.13.1

4.1.4 对比算法

为了验证本文提出的 NATLP 方法在知识图谱补全任务上的有效性，实验部分本文选取了一些最具代表性的以及最先进的知识图谱补全方法作为基线模型与 NATLP 模型

进行了对比实验，主要包含以下几类方法：

(1) 基于翻译的知识图谱嵌入模型，包括 TransE[1]，RotatE[55]。

(2) 基于张量分解的知识图谱嵌入模型，包括 DistMult[16]，ComplEx[17]。

(3) 基于卷积神经网络的模型，包括 ConvE[22]，ConvR[23]。

(4) 基于图神经网络的知识图谱嵌入模型，包括 R-GCN[26]，CompGCN[51]，KBGAT[29]，HKGN[56]，SE-GNN[57] 以及 MRGAT[58]。

4.1.5　超参数设置

NATLP 模型在实验中采用的超参数是通过网格搜索在验证集上进行评估后得出的，主要涉及到的超参数有实体和关系嵌入的维度、关系特定的邻居实体信息构造中使用的卷积核的大小、卷积核的数量、不同神经网络层的 Dropout 概率、训练批次的大小、训练中的总迭代次数、训练中的最大学习率以及标签的平滑比例等。

实验中采用了 Adamax[59] 优化器结合动态学习率调整策略进行模型的训练。在总迭代次数的前 10% 内，模型的学习率将从 0 线性提升到最高，并在剩余迭代次数内线性下降到 0。为了防止模型出现过度自信的现象，训练过程中以 0.1 的比率进行了标签平滑。NATLP 模型在两个数据集上的具体的超参数设置见表6所示。

表 6 NATLP 模型超参数设置

超参数	FB15k-237	WN18RR
实体和关系嵌入大小	320	320
卷积核大小	3×3	3×3
卷积核数量	32	32
Transformer 网络层数	4	4
嵌入层 Dropout 概率	0.2	0.2
Transformer 中的 Dropout 概率	0.6	0.6
训练批次大小	512	512
总迭代次数	300	500
最大学习率	0.001	0.02
标签平滑比例	0.1	0.1

4.2 实验结果与分析

4.2.1 整体实验结果分析

不同模型在 WN18RR 数据集和 FB15k-237 数据集上的链路预测实验结果如表7所示。文献 [60] 在充分大的超参数搜索空间下采用统一的训练框架进行了大量的实验，探究了不同训练策略、模型架构和超参数搜索方法对于知识图谱嵌入模型性能的影响，发现如果选择合适的训练方法，许多浅层的知识图谱嵌入模型能够获得比原始文献中优秀得多的性能。因此 TransE，RotatE，DistMult，ComplEx 和 ConvE 模型在两个数据集上的实验结果直接选取文献 [60] 中经过大量调参后的最优结果。另外，文献 [61] 指出 KBGAT 中存在评估策略不当以及测试集数据泄露的问题，因此 KBGAT 的实验结果选取自文献 [61] 中经过修正后的结果。

表 7 NATLP 实验结果

模型	WN18RR					FB15k-237				
	MRR	MR	Hits@1	Hits@3	Hits@10	MRR	MR	Hits@1	Hits@3	Hits@10
TransE	0.228	-	0.053	0.368	0.520	0.313	-	0.221	0.347	0.497
RotatE	0.478	-	0.439	0.494	0.553	0.333	-	0.240	0.368	0.522
DistMult	0.452	-	0.413	0.466	0.530	0.343	-	0.250	0.378	0.531
ComplEx	0.475	-	0.438	0.490	0.547	0.348	-	0.253	0.384	0.536
ConvE	0.442	-	0.411	0.451	0.504	0.339	-	0.248	0.369	0.521
ConvR	0.475	-	0.443	0.489	0.537	0.350	-	0.261	0.385	0.528
R-GCN	-	-	-	-	-	0.248	-	0.153	0.258	0.414
CompGCN	0.479	3533	0.443	0.494	0.546	0.355	197	0.264	0.390	0.535
KBGAT	0.412	**1921**	-	-	0.554	0.157	270	-	-	0.331
SE-GNN	0.484	3211	0.446	<u>0.509</u>	<u>0.572</u>	<u>0.365</u>	**157**	<u>0.271</u>	<u>0.399</u>	<u>0.549</u>
MRGAT	0.481	-	0.443	0.501	0.568	0.358	-	0.266	0.386	0.542
HKGN	<u>0.487</u>	<u>2468</u>	<u>0.448</u>	0.505	0.561	<u>0.365</u>	194	<u>0.271</u>	0.397	0.544
NATLP	**0.505**	2687	**0.465**	**0.519**	**0.576**	**0.374**	<u>181</u>	**0.281**	**0.411**	**0.560**

表中加粗项为每项指标的最高值，下划线项为每项指标的次高值。从表7中可以观察到，在两个基准数据集的绝大多数指标上，NATLP 都取得了最优的效果，相对于次优结果，在 FB15k-237 数据集上，NATLP 在 MRR、Hits@1、Hits@3 和 Hits@10 四个指标

上分别取得了 3.6%，3.7%，1.9%，0.6% 的性能提升；在 WN18RR 数据集上，NATLP 在 MRR、Hits@1、Hits@3 和 Hits@10 四个指标上分别取得了 2.4%，3.6%，3.0% 和 2.0% 的性能提升，这表明本文提出的 NALTP 方法在链路预测任务上有着很好的表现，证明了 NATLP 模型的有效性。同样使用了注意力机制，相比于 HKGN 以及 MRGAT 等基于图神经网络的方法，NATLP 有着可观的性能提升，证明了 Transformer 网络强大的表达能力以及在知识图谱嵌入领域的应用潜力。

4.2.2　模型关键设计分析

为了验证 NATLP 模型中关键设计：关系特定的邻居实体信息构造以及邻域感知 Transformer 模块的有效性，本文进行了多方面的对比实验来验证关键设计的效果。

首先，为了验证关系特定的邻居实体信息构造中参数生成网络以及全局关系嵌入对于链路预测任务的作用，本文在两个数据集上分别以三种设置进行了消融实验，分别是（1）原始 NATLP 模型，未对模型进行任何改动；（2）消融了参数生成网络的 NATLP 模型，同一实体在不同关系下进行信息构造中共享完全相同的网络参数；（3）消融了全局关系嵌入的 NATLP 模型，参数生成网络仅依赖于关系嵌入进行参数生成。实验结果如表8所示。

表 8 关系特定的邻居实体信息构造消融实验

数据集	模型设置	MRR	MR	Hits@1	Hits@3	Hits@10
WN18RR	NATLP	0.505	2504	0.466	0.519	0.578
	消融参数生成网络	0.496	2399	0.456	0.512	0.575
	消融全局关系嵌入	0.503	2519	0.461	0.518	0.578
FB15k-237	NATLP	0.376	161	0.284	0.411	0.561
	消融参数生成网络	0.367	145	0.274	0.404	0.555
	消融全局关系嵌入	0.373	175	0.279	0.411	0.558

从表8中实验结果可以观察到，在对参数生成网络进行消融后，模型的在两个数据集上的链路预测性能都有了明显的下降，这证明了相比于采用统一的网络参数进行消息构造，关系特定的网络参数能够更好地捕捉实体中关系相关的特定特征，挖掘相关的语义信息，提高任务性能；注意到消融了参数生成网络之后，相比于 WN18RR 数据集上的表现，NATLP 模型在 FB15k-237 数据集上的性能下降更为严重，本文认为这样的

差异主要是由于两个数据集上的关系种类差异所导致的，FB15k-237 数据集中的关系种类为 237 种，远远超过 WN18RR 数据集上的 11 种，因此采用统一的网络参数时模型在 FB15k-237 数据集上损失的信息更多。此外，去除全局关系嵌入后模型在两个数据集上的性能也有所下降，证明了捕捉关系之间共同特征的有效性。

此外，为了验证邻域感知 Transformer 模块中基于实体节点间最短距离的偏置项以及基于节点度数的偏置项的有效性，本文在两个数据集上以四种设置进行了 NATLP 模型的消融实验，分别是（1）原始 NATLP 模型，未对模型进行任何改动；（2）消融了基于实体节点间最短距离的偏置项的 NATLP 模型；（3）消融了基于节点度数的偏置项的 NATLP 模型；（4）同时对两个偏置项进行了消融的 NATLP 模型。具体实验结果如表9所示。

表 9 邻域感知 Transformer 模块消融实验

数据集	模型设置	MRR	MR	Hits@1	Hits@3	Hits@10
WN18RR	NATLP	0.505	2504	0.466	0.519	0.578
	消融节点最短距离的偏置项	0.500	2603	0.457	0.515	0.573
	消融节点度数偏置项	0.501	2570	0.459	0.517	0.578
	消融两个偏置项	0.497	2856	0.454	0.515	0.570
FB15k-237	NATLP	0.376	161	0.284	0.411	0.561
	消融节点最短距离的偏置项	0.373	180	0.280	0.409	0.557
	消融节点度数偏置项	0.372	184	0.281	0.409	0.554
	消融两个偏置项	0.369	179	0.277	0.409	0.550

通过表中结果可以发现，无论是消融了节点间最短距离的偏置项还是节点度数的偏置项，模型在两个数据集上的性能都会出现下降，证明了两种结构信息对于链路预测任务的重要性。

4.3 本章小结

本章对于基于领域感知的 Transformer 模型 NALTP 的实验部分进行了介绍，首先对本文中实验的基本方案设计进行了说明，包括实验采用的数据集，模型的评估策略以及实验的硬件和软件环境，并介绍了用来对比 NATLP 的基线模型。随后论文对 NALTP 模型的整体实验结果进行了介绍，证明了模型的有效性；最后通过消融实验对 NATLP

模型的中的关键设计：关系特定的邻居实体信息构造以及邻域感知 Transformer 模块进行了具体的探究和分析。

第五章　结合图路径和局部邻域的 Transformer 模型

本章主要对结合图路径和局部邻域的 Transformer 模型 TKGE-PN 的总体设计和模块的具体实现进行了介绍。主要包括对基于图神经网络的方法以及 NATLP 模型存在问题的分析、模型的总体框架设计、基于有偏随机游走的图路径采样算法的设计、Path-Transformer 路径编码模块以及 Neighbor-Transformer 局部邻域编码模块的具体设计。

5.1　现有问题分析

NATLP 模型对传统的 Transformer 模型的自注意力机制的计算方式进行改进，捕获了中心实体局部邻域内的结构信息，将 Transformer 模型应用到了知识图谱补全中，解决了先前基于图神经网络的方法模型表达能力差、对邻居实体之间相互依赖学习不足的问题。

但是，NATLP 模型依然存在缺陷，由于依然基于中心实体的局部邻域进行推理，NATLP 并没有解决基于图神经网络的知识图谱嵌入方法对于于长距离依赖学习不足的缺点。在知识图谱中，长距离的含义是实体间通过边连接需要经过多个中间节点。长距离依赖反映出实体之间较为间接的联系，这种联系在理解实体间复杂关系的层面上是非常重要的。例如，某一个历史人物与某一个现代组织之间可能存在长距离依赖，尽管他们之间直接的联系很少，但通过一系列历史事件和影响，可以构建出两者之间的联系，因此捕捉长距离依赖有益于推理和查询知识图谱的高阶模式。

基于图神经网络的方法主要是通过利用中心实体的局部邻域中蕴含的信息来完成知识图谱的补全。每层图神经网络只能学习到中心实体的一跳邻居的信息，这导致模型能够很好的捕捉知识图谱中的短距离依赖，而对于实体和实体之间的长距离依赖的学习不够充分。虽然其可以通过堆叠多层图神经网络让中心实体感知到距离更远的其他实体，但这样的方式只在层数较低（例如一到两层）时有效，之后随着层数的进一步增加，模型的性能反而会出现快速的下降，这主要是由过平滑问题（over-smoothing）导致的，即因为感受野的重叠而导致不同节点的表示过于相似无法进行区分。文献 [2] 对这种现象进行了系统和定量的研究，通过引入定量指标，发现导致过平滑的关键因素是

节点接收到的有效信息与噪声比例过低，结果如图7所示，该图来自于文献 [2] 原文。在每层神经网络中，节点都会聚合其所有邻居节点的信息并传递给下一层，这导致模型引入了大量的噪声信息。因此基于图神经网络的方法一般只能捕捉单个实体附近 1-2 跳内的局部信息，而缺乏利用长距离乃至全局信息的能力。

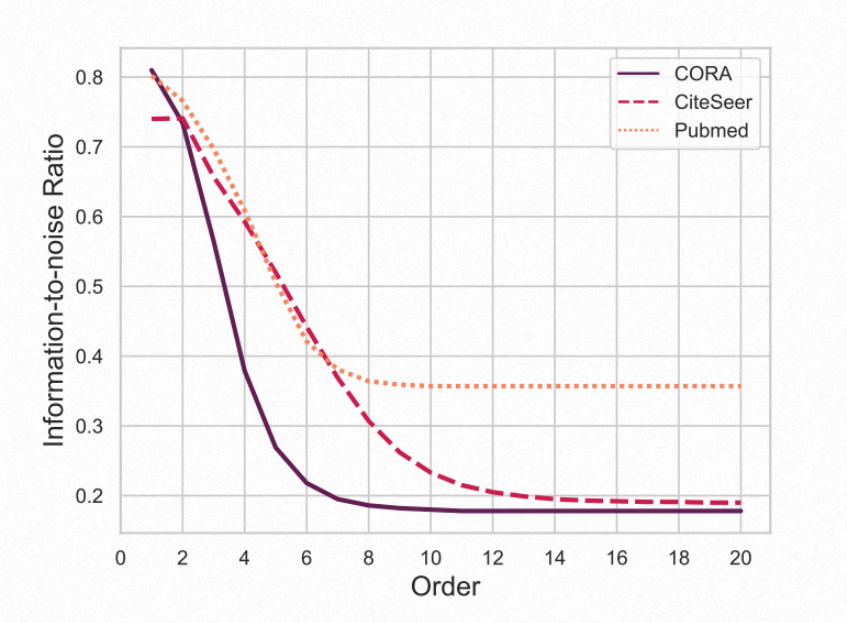

图 7 网络层数对信噪比的影响

为了解决以上问题，本文提出了一种基于 Transformer 的结合图路径和局部邻域的知识图谱嵌入方法 (A Transformer-based Knowledge Graph Embedding Model Combining Graph Paths and Local Neighborhood，TKGE-PN)。基于图神经网络的方法的成功证明了实体的局部邻域蕴含了丰富的信息，但知识图谱的结构信息除了图神经网络使用的局部邻域之外还有多种表达形式，例如图路径以及子图。在知识图谱中，图路径被定义为图谱中的实体-关系链，由不同三元组收尾相连组成，例如 (Yao Ming, Born In, Shanghai, City Of, China)。相对于局部邻域，图路径能够帮助模型更好地捕获实体和实体之间长距离的依赖，如图8所示。结合图路径和邻域信息，模型能够更好地学习长短距离依赖的同时避免过度平滑问题的出现。同样的，和 NATLP 类似，和 GNN 浅层的神经网络结构相比，Transformer 的自注意力机制能够给模型带来更强大的表达能力。本文提出的 TKGE-PN 以中心实体作为起点，采用有偏随机游走算法对图路径进行采样，并通过基于 Transformer 的图路径编码模块 Path-Transformer 和邻域信息编码模块 Neighbor-Transformer 对图谱中的长距离和短距离结构信息进行编码。此外针对相比于近距离的

信息，长距离信息学习更为困难的问题，本文为图路径编码模块设计了一个掩蔽实体关系预测任务，以确保模型能够充分学习图路径之间的长距离依赖。

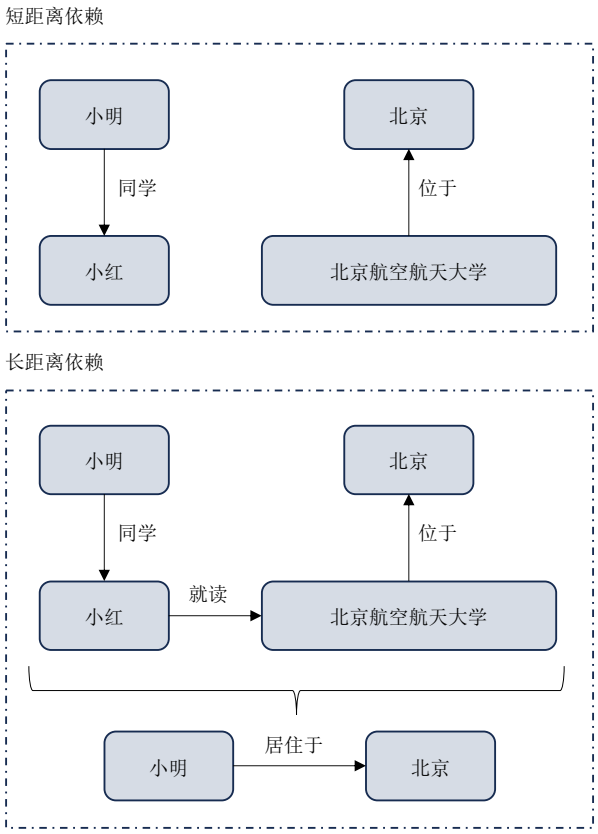

图 8 知识图谱中的短距离距离信息和长距离信息

5.2 TKGE-PN 模型设计

5.2.1 符号定义

为了方便说明论文提出的 TKGE-PN 模型的实现细节，本节对 TKGE-PN 模型中的关键概念和相关的数学符号进行了定义，具体内容参见表10。

表 10 TKGE-PN 模型中的符号定义

符号	说明
$\mathcal{G}$	知识图谱
$\mathcal{E},\mathcal{R},\mathcal{T}$	实体集合、关系集合、边集合
$\mathcal{G}'$	拓展后的知识图谱
$\mathcal{R}'$	拓展后的关系集合
$\mathcal{T}^{-1}$	逆关系边集合
$\mathcal{T}'$	拓展后的边集合

表 10 TKGE-PN 模型中的符号定义

符号	说明
$(s, r, ?)$	待遇测的三元组
s	头实体即中心实体
o	尾实体即目标实体
e	实体
r	关系
r^{-1}	关系 r 的逆关系
$\boldsymbol{s}, \boldsymbol{o}$	头实体嵌入和尾实体嵌入
$\boldsymbol{e}, \boldsymbol{r}$	实体嵌入和关系嵌入
T	图路径的长度
P	知识图谱中的图路径
$\mathcal{N}_s$	实体 s 的一阶邻居节点集合
$\mu_{depth}(e_{i+1})$	候选实体 e_{i+1} 的深度偏差
$\mu_{degree}(e_{i+1})$	候选实体 e_{i+1} 的度数偏差
α	控制深度偏差的权重
β	控制度数偏差的权重
p_{sample}	采样过程中实体被选中的概率
e_m, r_m	掩蔽实体关系预测任务中被掩蔽的实体或关系
$\mathbf{M}^P_{input}, \mathbf{M}^P_{output}$	Path-Transformer 模块的输入和输出
$\boldsymbol{e}_{mask}$	掩蔽占位嵌入
$\boldsymbol{e}_{query}$	查询向量
d	嵌入维度
ϕ_{chk}	棋盘式特征重组
$\circledast$	循环卷积操作
$f(\cdot)$	ReLU 激活函数
$vec(\cdot)$	二维张量转化为一维向量
ω_r	特定于关系 r 的卷积层参数
$\mathbf{W}_r$	特定于关系 r 的全连接层参数
$\boldsymbol{e}_{cls}$	Path-Transformer 特殊嵌入 Class Token
$\boldsymbol{e}_{gcls}$	Neighbor-Transformer 特殊嵌入 Global Class Token
$\mathbf{TE}$	类型嵌入
$\mathbf{PE}$	位置嵌入
a_{ij}	第 i 个输入和第 j 个输入之间的注意力得分
$dis(e_i, e_j)$	实体 e_i 与实体 e_j 之间最短路径的距离
$deg(e)$	实体 e 的节点度数
$\boldsymbol{o}_t$	模型预测的候选实体的嵌入

表 10 TKGE-PN 模型中的符号定义

符号	说明
σ	sigmoid 激活函数
p	三元组正确概率
L_{MERP}	掩蔽实体关系预测任务的损失
L_{LP}	链路预测任务损失
L	模型损失
t_i	第 i 个三元组的标签

和 NATLP 中的处理方式类似，为了确保实体之间信息的双向流动，TKGE-PN 会对原始的知识图谱进行拓展，为知识图谱中的每个事实三元组 (s, r, o) 添加对应的逆关系 r^{-1} 和逆三元组 (o, r^{-1}, s)：

$$\mathcal{R}' = \mathcal{R} \cup \{r^{-1} | r \in \mathcal{R}\} \tag{5.1}$$

$$\mathcal{T}^{-1} = \{(o, r^{-1}, s) | (s, r, o) \in \mathcal{T}\} \tag{5.2}$$

$$\mathcal{T}' = \mathcal{T} \cup \mathcal{T}^{-1} \tag{5.3}$$

$$\mathcal{G}' = (\mathcal{E}, \mathcal{R}', \mathcal{T}') \tag{5.4}$$

5.2.2 模型总体结构

本节主要对提出的基于 Transformer 的结合图路径和局部邻域的知识图谱嵌入方法 TKGE-PN 的总体结构进行介绍。模型架构如图9所示。

和 NATLP 不同，TKGE-PN 并没有采用编码器-解码器架构，而是利用 Transformer 自身的强大表达能力直接对目标实体的嵌入进行预测，这样的方式的优点是模型可以充分发挥自注意力机制的强大表达能力，其性能不会受限于用作解码器的基于图神经网络的知识图谱嵌入方法的限制。

TKGE-PN 模型主要由三个核心部分组成。首先第一部分是基于有偏随机游走的图路径采样算法，主要职责是以中心实体为起点在图谱中采样多条图路径；随后第二部分 Path-Transformer 路径编码模块负责学习采样到的图路径中蕴含的长距离的语义信息并将其转换为向量表示；最后 Neighbor-Transformer 局部邻域编码模块接收来自

Path-Transformer 的输入，整合待遇测事实三元组中的信息以及多条图路径所构成上下文邻域信息，并以此预测三元组得分。

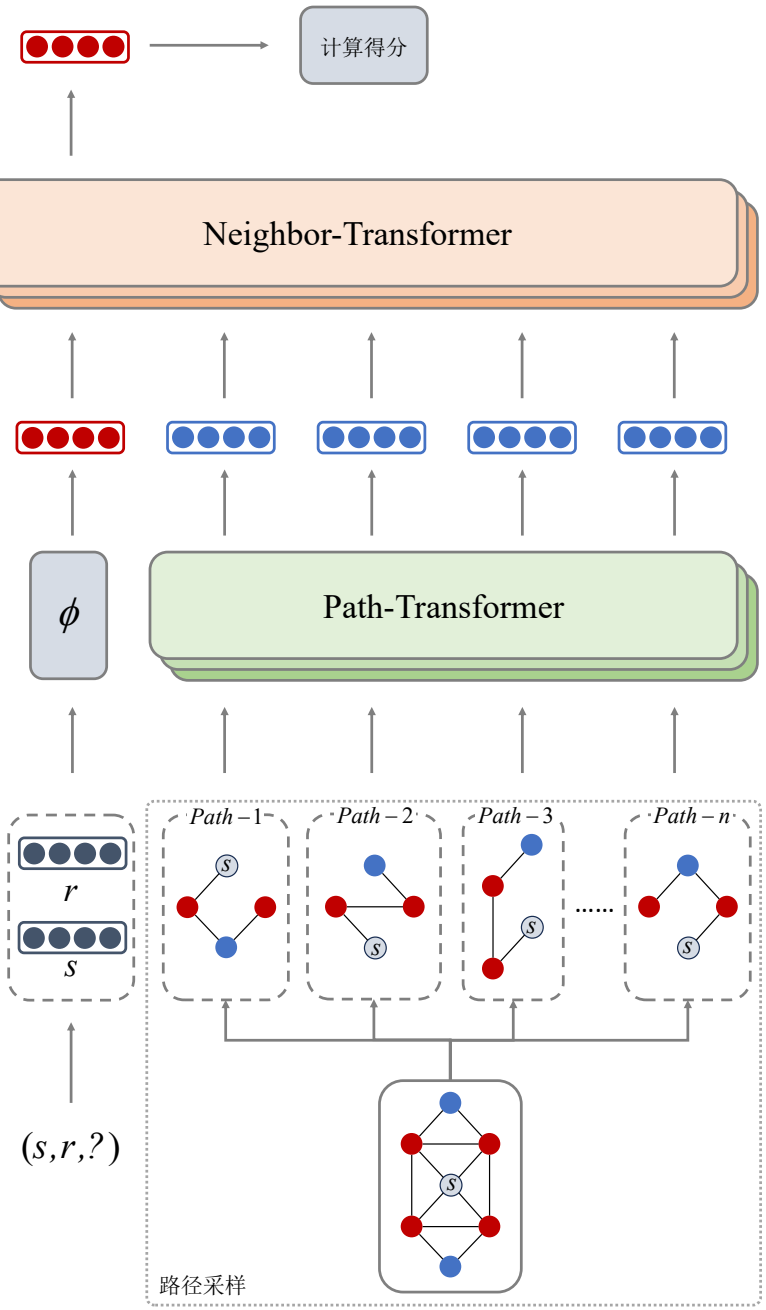

图 9　TKGE-PN 模型整体架构

5.2.3　基于有偏随机游走的图路径采样算法

为了通过图路径来学习实体之间的长距离依赖，TKGE-PN 模型的第一个任务是获得知识图谱中的图路径信息。由于知识图谱的规模往往十分庞大，因此遍历图谱中所有可能的图路径组合是一件不可能的工作。因此为了让模型能够充分利用图路径信息，提高链路预测的准确性，如何采样到高质量的知识图谱图路径是 TKGE-PN 首先需要解决

的问题。

给定一个待预测的三元组 $(s,r,?)$，图路径采样模块的主要任务是获得一条或者若干条以头实体 s 为起点的图路径用于链路预测任务。在 TKGE-PN 模型中，知识图谱中的图路径被定义为图谱中的实体-关系链，在链中实体和关系交替出现，链的第一个元素和最后一个元素必须为实体。在图谱中，以节点 s 为起点，长度为 T 的图路径 P 表示为：

$$\begin{aligned} &P =< s, r_1, e_1, r_2, e_2, ..., r_T, e_T >, \\ &\forall i \in (0,T), e_i \in \mathcal{E}, r_i \in \mathcal{R}', (e_i, r_{i+1}, e_{i+1}) \in \mathcal{T}' \end{aligned} \tag{5.5}$$

以往的基于图路径的知识图谱嵌入方法在给事实三元组打分时利用到的图路径数量往往只有 1-2 条，例如 RSN[37] 和 Interstellar[38]。这些方法采样到的图路径数量相比于知识图谱中可能的图路径数量是十分有限的，因此模型从这些图路径中学习得到的信息往往是片面的，很难全面地挖掘到中心实体对其他实体的长距离依赖，特别是当中心实体的节点度数比较高时，这样的问题会更加严重。此外，并不是所有的路径都对高质量的知识图谱嵌入有意义，低质量的图路径信息可能为引入额外的噪声，反而降低模型的性能。为了解决以上提到这些问题，本文提出了一种基于有偏随机游走的图路径采样算法。

首先，为了解决图路径采样数量不足导致模型学习到的信息不够全面的问题，对于一个待预测的三元组 $(s,r,?)$，TKGE-PN 的图路径采样模块不再采样固定数量的图路径，而是采样等于头实体 s 的节点度数数量的图路径。一般来说，实体的节点度数越高，以实体作为起点可能采样到的图路径就越多，蕴含的信息就越丰富；而当节点度数较小时，采样一到两条图路径能够充分学习中心实体的长距离依赖。因此随实体节点度数动态变化的图路径采样条数有利于更加全面、有效的捕捉图谱中的长距离依赖信息。

此外，为了避免随机采样的图路径之间出现路径重复而导致信息冗余的情况，对于同一个头实体节点 s，采样模块确保采样到的不同图路径中的前三个元素组成的事实三元组唯一，即对于不同图路径 $P_i =< s, r_1^i, e_1^i, ..., r_T^i, e_T^i >$ 和 $P_j =< s, r_1^j, e_1^j, ..., r_T^j, e_T^j >$，有 $(s, r_1, e_1) \neq (s, r_2, e_2)$，其中 $e_1^i, e_1^j \in \mathcal{N}_s$，$\mathcal{N}_s$ 为实体 s 的一阶邻居节点组成的集合。这样的采样方式不仅保证了采样到的图路径之间有足够的区分度，还让采样到的路径覆盖了中心实体的一阶邻域，通过所有采样的图路径，模型就能学习到中心实体的局部邻

域信息，实现长短距离依赖的同时学习。

为了确保采样到的图路径能够帮助模型进行链路预测任务，TKGE-PN 的图路径采样模块采用有偏随机游走算法来决定图路径中每一跳选中的实体和关系，实现高质量的图路径采样。为了能够更好的捕获长距离的依赖，TKGE-PN 希望采样到的路径能够更加的远离起始的头实体，即采样深度更大一些。为了实现这一点，在进行路径采样的过程中，模型提出了深度偏差来控制路径采样的过程。具体来说，假设在图路径采样的过程中，起始节点为 s，当前选中的实体为 e_i，上一跳实体为 e_{i-1}，下一跳的候选实体 e_{i+1} 为 e_i 的一阶邻居，则深度偏差 $\mu_{depth}(e_{i+1})$ 的计算方式为:

$$\mu_{depth}(e_{i+1}) = 1 - \alpha \cdot \frac{1}{dis(s, e_{i+1})} \tag{5.6}$$

其中 $dis(s, e_{i+1})$ 为实体 s 与实体 e_{i+1} 之间最短路径的距离，α 为控制采样过程中路径深度的超参数，为了获得深度更大的路径，一般设置为 $\alpha > 0$，这样候选实体距离起始实体的最短距离越大，则在计算采样概率时的深度偏差越大。图10说明了一个具体的实例。

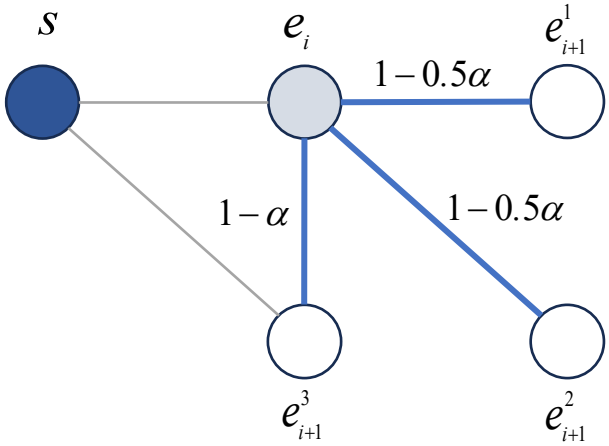

图 10 图路径采样过程中的深度偏差

除了路径深度之外，在图路径采样的过程中，TKGE-PN 还将实体的节点度数纳入了考虑。作为知识图谱中一种重要的结构信息，实体的节点度数反映了实体在整个知识图谱中的全局重要性，实体的节点度数越高，说明该实体连接的其他实体和关系越多，蕴含的信息越丰富，应该越重要，为了提高链路预测的准确性，在采样的过程中，节点度数较高的实体采样的权重应该更高。为了实现这一点，模型在路径采样过程中引入了

度数偏差。对于候选实体 e_{i+1}，度数偏差 $\mu_{degree}(e_{i+1})$ 的计算方式如下：

$$\mu_{degree}(e_{i+1}) = \lg(deg_{(e_{i+1})}) \tag{5.7}$$

其中 $deg_{(e_{i+1})}$ 为候选实体 e_{i+1} 的度数。此外，我们认为，如果一个实体相邻的实体的节点度数很高，那么该实体的重要性应该也会受到邻居实体的影响随之增加，因为经过该节点可以获取到重要性较高的实体节点的信息，因此在计算度数偏差时 TKGE-PN 将相邻实体的节点度数也纳入了考虑，有：

$$\mu_{degree}(e_{i+1}) = \frac{1}{2}\lg(deg_{e_{i+1}}) + \frac{1}{2\|\mathcal{N}_{e_{i+1}}\|}\sum_{e_k \in \mathcal{N}_{e_{i+1}}} \lg(deg_{e_k}) \tag{5.8}$$

最终，我们可以得到在图路径采样的过程中，下一跳候选实体 e_{i+1} 被选中的未正则化的概率计算公式为：

$$\mu_i = \mu_{depth}(e_{i+1}) + \beta \cdot \mu_{degree}(e_{i+1}) \tag{5.9}$$

$$p_{sample}(e_{i+1}) = \begin{cases} \mu_i & (e_i, r_i, e_{i+1}) \in \mathcal{T}' \\ 0 & otherwise \end{cases} \tag{5.10}$$

其中 β 为控制采样过程中度数偏差的权重。

5.2.4 Path-Transformer 路径编码模块

通过基于有偏随机游走的采样算法，TKGE-PN 模型获得了若干条图路径用于链路预测任务。但是这些采样到的图路径无法直接使用，还需要 Path-Transformer 路径编码模块学习其中蕴含的语义信息并转换为对应的向量表示。和处理局部邻域结构不一样的是，图路径天然具有序列数据的形式，而 Transformer 被认为是建模序列数据的最强大的神经网络，因此原始的 Transformer 网络就能够很好的挖掘路径中实体与实体、实体与关系之间的长短距离依赖，而无需对网络结构进行额外的改造。

给定一条长度为 T 的图路径 $P =< s, r_1, e_1, ..., r_T, e_T >$，Path-Transformer 路径编码模块的输入主要由以下几个部分组成：图路径嵌入表示 $\mathbf{M}_{path}$，以及一个可学习的特殊嵌入 Class Token 的向量表示 $\mathbf{e}_{cls}$ 用于获取所有输入的统计特性，因此路径编码模块的

输入表示为：

$$\mathbf{M}_{path} = [\boldsymbol{e}_s, \boldsymbol{r}_1, \boldsymbol{e}_1, ..., \boldsymbol{r}_T, \boldsymbol{e}_T] \tag{5.11}$$

$$\mathbf{M}'_{path} = [\boldsymbol{e}_{cls}, \mathbf{M}_{path}] \tag{5.12}$$

其中 $\boldsymbol{e}_{cls} \in \mathbb{R}^d$，$\mathbf{M}_{path} \in \mathbb{R}^{(2T+1)\times d}$。$\mathbf{M}_{path}$ 由图路径中包含的所有的实体及关系对应的嵌入构成，$\boldsymbol{e}, \boldsymbol{r} \in \mathbb{R}^d$ 代表分别代表实体和关系嵌入，d 则为嵌入向量维度的大小。同样的，和 NATLP 类似，Path-Transformer 采用类型嵌入来帮助模型区分实体嵌入，关系嵌入以及特殊嵌入向量 $\boldsymbol{e}_{cls}$。此外，由于图路径是序列数据，Path-Transformer 采用可学习的位置位置来识别图路径中实体和关系之间的顺序关系，因此 Path-Transformer 的最终输入表示为：

$$\mathbf{M}^P_{input} = \mathbf{M}'_{path} + \mathbf{TE} + \mathbf{PE} \tag{5.13}$$

其中 $\mathbf{TE}$ 为类型嵌入，$\mathbf{PE}$ 为位置嵌入。

在完成输入构造后，Path-Transformer 会利用自注意力机制学习图路径中的语义信息。设 Path-Transformer 模块最后一层的输出为 $\mathbf{M}^P_{output} \in \mathbb{R}^{(2T+1)\times d}$，TKGE-PN 取输出中特殊嵌入向量 $\boldsymbol{e}_{cls}$ 对应位置的嵌入 $\boldsymbol{e}_{path} \in \mathbb{R}^d$ 作为当前图路径的向量表示。

Path-Transformer 路径编码模块尝试从知识图谱图路径中挖掘长距离的依赖用于链路预测任务，但是相比于近距离的局部邻域，图路径中蕴含的长距离依赖反应的是实体之间间接的联系，容易在学习的过程中丢失，模型学习起来是更加困难的。针对这样的挑战，受到 BERT[40] 中掩蔽语言建模 (Masked Language Model，MLM) 预训练任务的启发,TKGE-PN 提出了掩蔽实体关系预测 (Masked Entity and Relation Prediction，MERP) 任务，以加强 Path-Transformer 从图路径上下文中挖掘信息的能力。

具体来说，对于输入的每一条图路径 P，模型将会随机选择遮掩或者替换掉某个关系或者实体进行预测任务。以实体为例，在掩蔽替换阶段，被选中的实体 e_m 将会以一定概率被替换成特殊的掩蔽占位嵌入 e_{mask}、其他的随机实体或者维持不变，模型使用超参数来调节以上三种情况的概率。而在预测阶段，模型取 Path-Transformer 输出嵌入矩阵 $\mathbf{M}^P_{output}$ 中被选中的实体对应的嵌入 $\boldsymbol{e}'_m$ 来尝试辨认出被遮掩的正确实体 e_m。具体来说，对于一个候选的实体 e_t，模型将 $\boldsymbol{e}'_m$ 通过一个双层全连接层并计算 $\boldsymbol{e}'_m$ 与 $\boldsymbol{e}_t$ 之间

的余弦相似度来获得候选实体 e_t 的正确概率 p_m：

$$p_m(e_t) = \sigma((f(\boldsymbol{e}'_m\mathbf{W}')\mathbf{W}'')\boldsymbol{e}_t) \tag{5.14}$$

其中 $f(\cdot)$ 和 $\sigma(\cdot)$ 分别代表 ReLU 和 sigmoid 激活函数，$\mathbf{W}'$ 和 $\mathbf{W}''$ 分别为两个全连接层的参数。当被选中的元素为关系时，处理方法类似。最后 TKGE-PN 通过交叉熵损失函数，计算得到掩蔽实体关系预测任务的分类损失为:

$$L_P = -\frac{1}{N}\sum_i t_m^i log(p_m^i) + (1-t_m^i)log(1-p_m^i) \tag{5.15}$$

其中 t_m^i 为 MERP 任务中的分类标签，p_m^i 为对应候选实体的正确概率。

掩蔽实体关系预测任务加强了 Path-Transformer 挖掘图路径中语义信息的能力。在原本的训练过程中，由于近距离的依赖学习难度较低，模型可能会倾向于挖掘近距离的实体中蕴含的信息，而忽略了远距离的部分。而通过随机地对图路径中的元素进行掩蔽，模型无法再单纯地依赖某个特定的图路径元素；通过对掩蔽的元素进行预测，模型也确保了信息不会丢失。基于掩蔽实体关系预测任务，模型实现了对图路径中长短距离依赖信息的平衡。

5.2.5 Neighbor-Transformer 局部邻域编码模块

给定一个待遇测的三元组 $(s, r, ?)$，通过基于有偏随机游走的图路径采样算法，模型已经获得了以头实体 s 为起点的多条图路径，数量等于头实体 s 的节点度数。通过对采样的规则进行限制，这些图路径覆盖了头实体 s 的一阶局部邻域。因此，Neighbor-Transformer 局部邻域编码模块通过基于 Path-Transformer 学习到的所有的图路径表示，便能够同时结合知识图谱的局部邻域结构信息和图路径结构信息，在避免过平滑问题的同时实现对于长短距离依赖的同时学习，提高知识图谱补全任务的性能。

Neighbor-Transformer 局部邻域编码模块的输入主要包含以下几个部分：图路径嵌入矩阵 $\mathbf{M}_{paths}$，查询嵌入 $\boldsymbol{e}_{query}$ 以及一个可学习的特殊嵌入 $\boldsymbol{e}_{gcls}$。$\mathbf{M}_{paths} \in \mathbb{R}^{N_p \times d}$ 由所有采样到的以头实体 s 为起点的图路径的嵌入表示组成，其中 N_p 为采样到的图路径的数量，有：

$$\mathbf{M}_{paths} = \left[\boldsymbol{e}_{P_1}, \boldsymbol{e}_{P_2}, ..., \boldsymbol{e}_{P_{N_p}}\right] \tag{5.16}$$

其中 $\boldsymbol{e}_{P_i} \in \mathbb{R}^d$ 为第 i 条图路径的向量表示。为了能够发挥 Transformer 模型的表达能力，避免模型的性能被基于卷积神经网络的解码器限制，TKGE-PN 抛弃了 NALTP 中采用的编码器-解码器架构，而是利用 Transformer 网络尝试直接拟合待遇测尾实体 o 的嵌入。查询向量 $\boldsymbol{e}_{query}$ 的主要作用是向 Transformer 传递待遇测三元组的信息，让模型能够基于当前输入调整注意力分数的计算，提高模型链路预测的性能。$\boldsymbol{e}_{query}$ 由待预测三元组中头实体 s 和关系 r 对应的嵌入向量 $\boldsymbol{s}$ 和 $\boldsymbol{r}$ 计算得到，有：

$$\boldsymbol{e}_{query} = \Phi(\boldsymbol{s}, \boldsymbol{r}) \tag{5.17}$$

其中 $\boldsymbol{s}, \boldsymbol{r} \in \mathbb{R}^d$，$\Phi(\cdot)$ 为对应的运算函数，可以根据不同的场景选用不同的形式，例如向量相加，向量相乘，或者利用神经网络完成查询向量的构造。本文中采用的是 NATLP 中基于特定关系的消息构造方式，有：

$$\Phi(s, r) = f\left(vec\left(f\left(\phi_{chk}\left(\boldsymbol{s}, \boldsymbol{r}\right) \circledast \omega_r\right)\right)\mathbf{W}_r\right) \tag{5.18}$$

特殊嵌入向量 $\mathbf{e}_{gcls}$ 用于获取所有输入的统计特性，并且对应位置的输出将会被用于链路预测任务最后的概率计算。和 NALTP 类似，三种不同的类型嵌入被用来区分图路径嵌入矩阵 $\mathbf{M}_{paths}$，查询嵌入 $\boldsymbol{e}_{query}$ 以及 $\boldsymbol{e}_{gcls}$。由于不同的图路径信息之间没有明显的相对位置关系，因此 Neighbor-Transformer 局部邻域编码模块中没有使用位置嵌入。最终，Neighbor-Transformer 模块的输入表示为：

$$\mathbf{M}'_{neighbor} = [\boldsymbol{e}_{gcls}, \boldsymbol{e}_{query}, \mathbf{M}_{paths}] \tag{5.19}$$

$$\mathbf{M}^N_{input} = \mathbf{M}'_{neighbor} + \mathbf{TE} \tag{5.20}$$

和 NATLP 类似，考虑到图路径上的实体度数之和能够一定程度上反应图路径信息的重要性，在计算注意力分数时，Neighbor-Transformer 将图路径的整体节点度数也纳入了考虑。具体的，Neighbor-Transformer 在计算注意力得分时额外添加一个图路径节

点度数的辅助项：

$$a_{ij} = \frac{(\boldsymbol{e}_{P_i} W_Q)(\boldsymbol{e}_{P_j} W_K)^T}{\sqrt{d}} + 1 - \frac{1}{\lg(deg_{P_i}) \cdot \lg(deg_{P_j})} \tag{5.21}$$

$$deg_{P_i} = \sum_{n=1}^{T} deg_{e_n} \tag{5.22}$$

最后，TKGE-PN 取最后一层输出中对应特殊嵌入向量 $\boldsymbol{e}_{gcls}$ 对应位置的输出向量 $\boldsymbol{T}_{GCLS}$ 来进行链路预测任务。对于任意一个候选实体 e_t，模型将 $\boldsymbol{T}_{GCLS}$ 与 e_t 的嵌入 $\boldsymbol{e}_t$ 进行点积后并经过 sigmoid 激活函数后得到三元组 (s, r, e_t) 的得分 p。获得所有候选实体的得分后，计算得到链路预测任务的交叉熵损失为：

$$L_{LP} = -\frac{1}{N} \sum_i t_i log(p_i) + (1 - t_i) log(1 - p_i) \tag{5.23}$$

最终，通过将链路预测任务的损失和掩蔽实体关系预测任务的损失相加，我们可以得到用于模型训练的最终损失函数：

$$L_{MERP} = -\frac{1}{N_p} \sum_i L_{p_i} \tag{5.24}$$

$$L = L_{LP} + L_{MERP} \tag{5.25}$$

其中 L_{p_i} 是第 i 条图路径上掩蔽实体预测任务的损失。

此外，为了避免出现模型训练和模型预测时的数据分布的不一致，在训练过程中模型将从头实体的邻域内移除真实的尾实体，避免训练期间模型总是可以从头实体的邻域内感知到真实的尾实体，而这种情况和预测时的实际情况不符。

5.3 本章小结

本章对 TKGE-PN 模型的整体架构和实现细节进行了详细介绍。首先对基于图神经网络的方法以及前文提出的 NATLP 模型依然存在的问题进行了介绍；随后给出了模型中涉及到的数学符号的详细定义；之后介绍了模型的整体架构组成；最后，对 TKGE-PN 中的关键设计细节进行了具体的说明，包括（1）基于有偏随机游走的图路径采样算法，将图路径的路径深度以及实体的节点度数纳入了考虑，用于采样高质量的图路径（2）

Path-Transformer 路径编码模块，通过掩蔽实体关系预测任务加强模型学习图路径中语义信息的能力，实现了长短距离依赖信息的平衡（3）Neighbor-Transformer 局部邻域编码模块，结合待遇测的事实三元组，综合图路径结构信息和局部邻域结构信息完成链路预测任务。

第六章　TKGE-PN 模型实验与验证

为了验证本课题提出的 TKGE-PN 算法的有效性，本文在两个基准数据集上对 TKGE-PN 进行了实验验证。TKGE-PN 模型实验时采用的数据集、实验环境和评估策略和 NATLP 模型实验时完全一致，因此本章不再另行介绍。本章首先说明了用来对比的基线算法以及模型在两个数据集上的超参数设置，随后介绍了模型的整体实验结果并进行了深入的探究，最后通过消融实验对模型的关键设计进行了分析和讨论。

6.1　对比算法

为了验证本文提出的 TKGE-PN 模型在链路预测任务上的有效性，在实验部分，我们将其与一些最具代表性的以及最先进的知识图谱嵌入模型进行了比较，主要包含以下几类：

（1）传统的知识图谱嵌入模型，时间和空间复杂度较低，包括基于翻译的模型 TransE[1]，RotatE[55]，基于张量分解的模型 DistMult[16]，ComplEx[17]，以及基于卷积神经网络的模型 ConvE[22] 和 ConvR[23]。

（2）基于图神经网络的嵌入模型, 主要学习图谱局部邻域结构进行知识图谱嵌入，包括 R-GCN[26]，CompGCN[51]，KBGAT[29]，HKGN[56]，SE-GNN[57] 以及 MRGAT[58]。

（3）基于图路径的嵌入模型，利用知识图谱的图路径结构进行嵌入工作，包括 RSN[37] 和 Interstellar[38]。

（4）基于 Transformer 的嵌入模型，主要利用 Transformer 的表达能力进行链路预测，包括 KG-BERT[39]，HittER[32]，StAR[62] 和 Relphormer[33]，以及本文先前提出的 NALTP 模型。

6.2　模型超参数设置

实验采用的超参数是通过网格搜索在验证集上进行评估后得出的。实验中，Path-Transformer 图路径编码模块以及 Neighbor-Transformer 局部邻域编码模块分别由 3 层和 4 层 Transformer 编码层组成，多头注意力机制的头数为 8，采用的实体和关系嵌入维度为 320。在路径采样算法的设置上，两个数据集上路径采样的深度偏差权重 $\alpha = 0.7$，度

数偏差权重 $\beta = 0.1$，WN18RR 数据集上采样路径的最大长度 $T = 4$，FB15k-237 数据集上 $T = 3$。在掩蔽实体关系预测任务中，每个实体和关系有 30% 的概率被特殊的掩蔽占位嵌入 e_{mask} 替换，20% 被其余随机实体和关系替换，50% 的概率维持不变。考虑到数据集中存在极小一部分节点度数极高的实体以及 FB15k-237 和 WN18RR 数据集在节点平均度数上的差异，实验对每个实体采样得到的图路径的最大数量进行了限制，在 FB15k-237 和 WN18RR 数据集上分别为 50 和 12，这样的设置能够保证采样的图路径可以完全覆盖数据集中 80% 以上实体的一阶邻域。

实验中采用了 Adamax[59] 优化器结合动态学习率调整策略进行模型的训练。在总迭代次数的前 10% 内，模型的学习率将从 0 线性提升到最高，并在剩余迭代次数内线性下降到 0。在训练过程中，除了嵌入层以外的神经网络层的随机失活概率为 0.65，嵌入层的随机失活概率为 0.2。模型训练时在 FB15K-237 数据集上的总迭代次数为 300，训练批次的大小为 512，最大学习率为 0.0011；而对于 WN18RR 数据集，总迭代次数为 500，批处理大小为 512，最大学习率为 0.002。为了防止模型出现过度自信的现象，训练过程中以 0.1 的比率进行了标签平滑。模型使用的具体超参数参见表11。

表 11 TKGE-PN 模型超参数设置

超参数	FB15k-237	WN18RR
实体和关系嵌入大小	320	320
Path-Transformer 网络层数	3	3
Neighbor-Transformer 网络层数	4	4
嵌入层 Dropout 概率	0.2	0.2
Transformer 中的 Dropout 概率	0.65	0.65
采样路径的最大长度	3	4
深度偏差权重	0.7	0.7
度数偏差权重	0.1	0.1
路径采样最大数量	50	12
训练批次大小	512	512
总迭代次数	300	500
最大学习率	0.001	0.02
标签平滑比例	0.1	0.1

6.3 整体实验结果分析

不同模型在 WN18RR 数据集和 FB15k-237 数据集上的链路预测实验结果如表所示。和 NATLP 类似，TransE、RotatE、DistMult、ComplEx 和 ConvE 在两个数据集上的实验结果直接选取文献 [60] 中经过大量调参的最优结果。由于存在评估策略不当以及测试集数据泄露的问题，KBGAT 的实验结果选取文献 [61] 中经过修正后的结果。

表 12 TKGE-PN 实验结果

模型	WN18RR					FB15k-237				
	MRR	MR	Hits@1	Hits@3	Hits@10	MRR	MR	Hits@1	Hits@3	Hits@10
TransE	0.228	-	0.053	0.368	0.520	0.313	-	0.221	0.347	0.497
RotatE	0.478	-	0.439	0.494	0.553	0.333	-	0.240	0.368	0.522
DistMult	0.452	-	0.413	0.466	0.530	0.343	-	0.250	0.378	0.531
ComplEx	0.475	-	0.438	0.490	0.547	0.348	-	0.253	0.384	0.536
ConvE	0.442	-	0.411	0.451	0.504	0.339	-	0.248	0.369	0.521
ConvR	0.475	-	0.443	0.489	0.537	0.350	-	0.261	0.385	0.528
R-GCN	-	-	-	-	-	0.248	-	0.153	0.258	0.414
CompGCN	0.479	3533	0.443	0.494	0.546	0.355	197	0.264	0.390	0.535
KBGAT	0.412	1921	-	-	0.554	0.157	270	-	-	0.331
SE-GNN	0.484	3211	0.446	0.509	0.572	0.365	157	0.271	0.399	0.549
MRGAT	0.481	-	0.443	0.501	0.568	0.358	-	0.266	0.386	0.542
HKGN	0.487	2468	0.448	0.505	0.561	0.365	194	0.271	0.397	0.544
RSN	0.400	-	0.380	-	0.448	0.280	-	0.202	-	0.453
Interstellar	0.480	-	0.438	-	0.546	0.320	-	0.233	-	0.508
KG-BERT	0.216	97	0.041	0.302	0.524	-	153	-	-	0.420
HittER	0.503	-	0.462	0.516	0.584	0.373	-	0.279	0.409	0.558
StAR	0.40	**51**	0.243	0.491	0.709	0.296	**117**	0.205	0.322	0.482
Relphormer	0.495	-	0.448	-	**0.591**	0.371	-	**0.314**	-	0.481
NATLP	0.505	2687	0.465	0.519	0.576	0.374	181	0.281	0.411	0.560
TKGE-PN	**0.510**	2540	**0.467**	**0.522**	0.590	**0.379**	160	0.288	**0.414**	**0.562**

表中加粗项为每项指标的最高值，下划线项为每项指标的次高值。从表12中我们可以观察到，（1）在两个基准数据集中的大多数评价指标上，TKGE-PN 模型优于其

他所有的基准模型，这证明了本文提出的方法的有效性。TKGN-PN 和 NATLP 等基于 Transformer 的嵌入方法相比于基于图卷积神经网络的嵌入方法如 MRGAT 取得了较大的性能提升，这证明了 Transformer 结构在知识图谱嵌入领域的巨大潜力。（2）而同样是基于 Transformer 的知识图谱嵌入方法，NALTP 利用 Transformer 来学习中心实体的一阶邻域信息，没有利用路径信息而缺乏捕捉图谱中长距离依赖的能力。TKGE-PN 相对于 NALTP 的性能提升表明了图路径结构对于知识图谱嵌入的帮助以及 TKGE-PN 从图路径结构中挖掘长距离依赖的能力。

为了进一步研究图路径采样策略对于 TKGE-PN 链路预测性能的影响，本文探究了在不同图路径采样长度下 TKGE-PN 在两个数据集上的性能表现，实验结果如表13所示。

表 13 验证集上不同路径采样长度下的链路预测结果

采样路径的最大长度	WN18RR		FB15k-237	
T	MRR	Hits@10	MRR	Hits@10
1	0.498	0.571	0.373	0.559
2	0.503	0.580	0.376	0.560
3	0.506	0.584	**0.380**	**0.563**
4	**0.509**	**0.588**	0.377	0.561
5	0.507	0.585	0.375	0.557
6	0.503	0.578	0.372	0.559
7	0.501	0.576	0.370	0.555

可以发现，随着采样长度的增加，模型在两个数据集上的性能总体呈现一个先增加后降低的趋势。一方面，路径长度的增加能够帮助模型学习到长距离的依赖，因此在初期能够有效提升模型嵌入效果；而另一方面，并不是所有的图路径信息都是有意义的，采样长度过高时采样到的路径随机性会增大，距离过长时两个实体之间的依赖也会减弱，引入的噪声信息随之增加，反而对模型造成了干扰，导致性能下降。此外，我们可以发现在不同数据集上最优的采样长度也不同，相比较之下，WN18RR 数据集上的最优长度更长。我们认为这样的差异是由于不同数据集之间的稀疏性差距导致的，WN18RR 数据集相对更加稀疏，使得采样的随机性降低，引入的噪声数据减少，因此最优的采样长度相比于 FB15k-237 数据集更长。

此外，我们还探究了不同路径深度偏差 α 下 TKGE-PN 模型的性能表现。深度偏差

α 较小时路径采样偏向于围绕中心实体进行，深度偏差 α 较大时则更注重于捕捉远距离的依赖，我们发现深度偏差 $\alpha \in [0.6, 0.8]$ 时模型的性能表现最好。我们认为当深度偏差控制在这个区间内时，模型能够比较好的兼顾长短距离的信息。

6.4 模型关键设计分析

为了验证 TKGE-PN 模型中局部邻域、图路径信息以及掩蔽实体关系预测任务对于知识图谱嵌入的作用，论文在两个数据集上分别以四种设置进行了消融实验，分别是未消融任何部分的 TKGE-PNM 模型、去除了局部邻域编码模块、去除掩蔽实体关系预测任务以及同时去除图路径编码模块和掩蔽实体关系预测任务，实验结果如表14所示。

表 14 TKEG-PN 消融实验结果

数据集	模型	MRR	MR	Hits@1	Hits@3	Hits@10
WN18RR	TKGE-PN	0.509	2610	0.467	0.525	0.588
	Path+MERP	0.487	2752	0.445	0.504	0.567
	Neighbor	0.497	2806	0.459	0.512	0.570
	Neigbor+Path	0.502	2636	0.461	0.521	0.579
FB15k-237	TKGE-PN	0.380	156	0.289	0.415	0.563
	Path+MERP	0.367	144	0.278	0.406	0.555
	Neighbor	0.373	175	0.279	0.410	0.559
	Neigbor+Path	0.376	169	0.282	0.413	0.562

具体来说，去除局部邻域编码模块是通过限制路径采样数量为 1 来实现，而去除图路径编码模块则是通过限制路径采样长度为 1 实现，此时路径信息内只包含中心实体的一阶邻居，即一阶局部邻域信息。可以看到，无论是去除局部邻域编码模块，还是去除图路径编码模块，模型的效果都会出现明显的降低。实验证明了知识图谱中的这两种结构信息都能够帮助进行链路预测，仅依赖路径信息，模型无法从局部邻域内的丰富的实体和关系中学习到中心实体的综合性质；仅依赖局部邻域，则无法挖掘到中心实体和其他实体之间的远距离依赖。相对来说，去除局部邻域编码模块带来的性能下降更加的严重，这证明了局部邻域信息的重要性，也符合我们的直觉：距离中心实体更近的实体和关系更能反映中心实体的性质，蕴含的信息也更加容易学习。此外，论文发现在脱离掩蔽实体关系预测任务的情况下添加图路径信息而并不能带来明显的性能提升，这说明

了图路径上长距离信息学习的困难性以及掩蔽实体关系预测对提升模型长距离信息学习能力的作用。总体来看，TKGE-PN 模型任一部分的缺失都会导致模型的最终结果受到负面影响。

为了进一步验证图路径信息在挖掘知识图谱中长距离依赖的作用，以及长距离依赖信息对于链路预测任务的帮助，本文将 WN18RR 验证集中的事实三元组按照头实体和尾实体之间的最短距离进行了划分，并在划分后的验证集上评估了消融了路径编码模块的 TKGE-PN 模型以及原始 TKGE-PN 模型的链路预测任务性能，实验结果如图11所示。

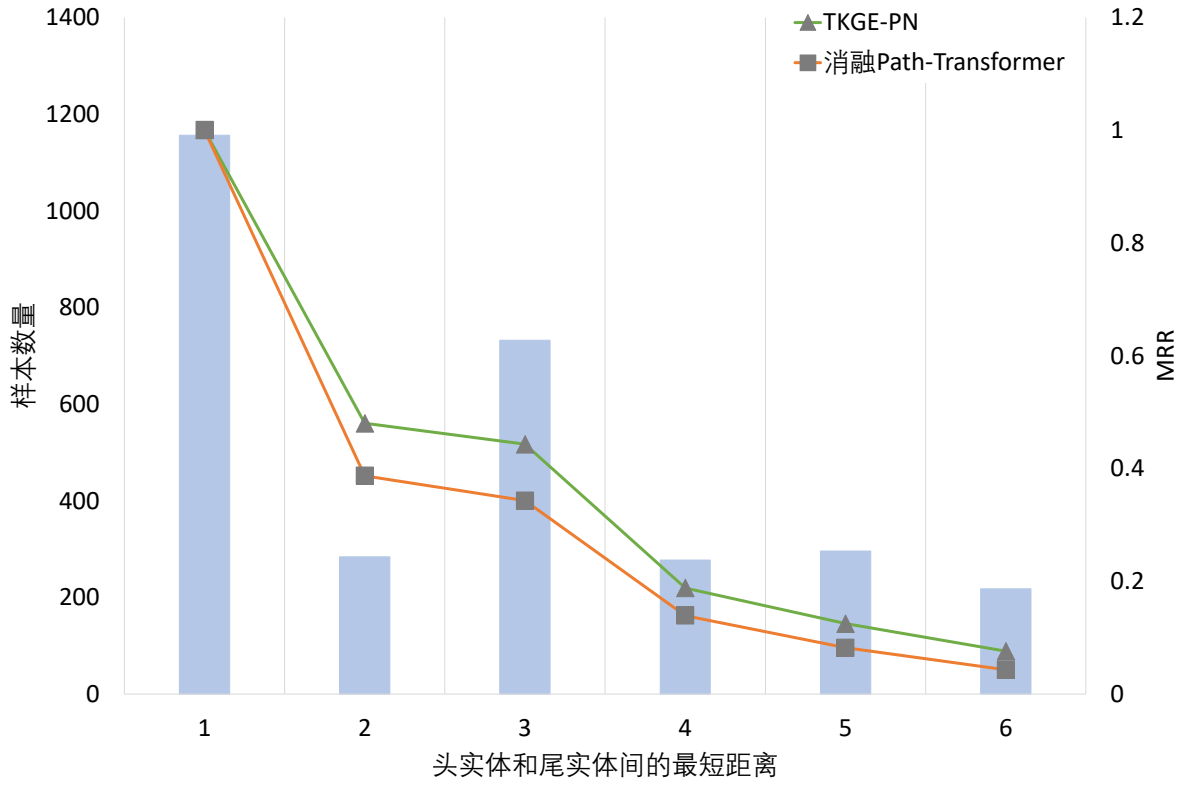

图 11　WN18RR 分组实验结果

从图11中实验结果可以发现，随着头实体和尾实体之间最短距离的增加，模型的 MRR 指标出现了明显的下降，说明推断图谱之间的长距离依赖是非常困难的。此外，我们可以注意到，相比于消融了图路径编码模块的模型，学习了图路径的 TKGE-PN 模型在长距离的样本上性能表现更好，特别是在头尾实体间最短距离为 2-4 的样本上，证明了图路径信息能够有效捕捉实体间的长距离依赖，提高链路预测任务的性能表现。

6.5 本章小结

本章对于结合图路径和局部邻域的 Transformer 模型 TKGE-PN 的实验部分进行了介绍，首先论文对 NALTP 模型的整体实验结果进行了介绍，证明了 TKGE-PN 模型性能表现优于绝大部分现有的模型，证明了 TKGE-PN 模型的有效性，并探究了路径长度以及路径采样策略对于性能的影响；最后通过实验对 TKGE-PN 模型的中的关键设计进行了具体的探究和分析，包括图路径编码模块、局部邻域编码模块以及掩蔽实体关系预测任务，并通过实验进一步证明了图路径在长距离依赖信息学习中的重要作用。

总结与展望

自从2017年被提出，Transformer已经在计算机视觉、自然语言处理等多个领域大放异彩，展现出了巨大的潜力，收到了许多的关注，因此也有不少工作尝试将Transformer网络引入到知识图谱补全领域中。本课题针对传统知识图谱嵌入方法和基于图神经网络的方法的缺点，研究如何利用Transformer模型来学习知识图谱中的语义和结构信息，提升知识图谱补全任务的性能，提出了两种新型的基于Transformer的知识图谱补全方法：基于邻域感知的Transformer模型NATLP和结合图路径和局部邻域的Transformer模型TKGE-PN。

针对基于图神经网络的知识图谱嵌入方法表达能力不足的问题，NATLP研究利用Transformer网络来完成知识图谱补全。针对Transformer无法直接学习图结构的问题等问题，NALTP对Transformer的自注意力机制进行了改造，提出了一种融合图结构的自注意力机制，使得模型能够学习到输入消息之间的互相依赖；针对关系和实体间的交互建模不足的问题，NATLP在模型输入信息构造阶段，基于关系生成特定的网络参数，实现关系特定的邻居信息构造，建模不同关系对于实体传递消息的影响。

针对NATLP模型和基于图神经网络的方法缺乏捕捉图谱中长距离依赖的缺点，TKGE-PN通过对知识图谱中的局部邻域和图路径两种结构信息的融合，完成了在利用丰富的邻域信息的同时对于图谱中长距离信息的挖掘，提高了知识图谱补全任务的性能。TKGE-PN首先通过基于有偏随机游走的采样算法对图路径进行采样，随后通过基于Transformer的图路径编码模块学习其中的长距离依赖，并通过掩蔽实体关系任务实现长短距离信息的平衡；最后通过局部邻域编码模块实现了图路径和局部邻域结构信息的综合应用。在两个标准数据集WN18RR和FB15k-237上，TKGE-PN链路预测任务的性能表现超越绝大多数现有的嵌入模型，证明了方法的有效性。

未来，我们计划进一步探索对于知识图谱中结构信息的利用，包括更有效率的利用方式和更加丰富的结构信息种类，同时尝试将TKGE-PN应用到除链路预测之外的其他知识图谱表示学习任务中。此外，TKGE-PN基于Transformer的结构带来的大量资源开销如何进行优化也是一个值得研究的方向。

参考文献

[1] Bordes A, Usunier N, Garcia-Durán A, et al. Translating embeddings for modeling multi-relational data[C]//NIPS'13: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. Red Hook, NY, USA: Curran Associates Inc., 2013: 2787–2795

[2] Chen D, Lin Y, Li W, et al. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(04): 3438–3445

[3] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc., 2017: 6000–6010

[4] Singhal A. Introducing the knowledge graph: things, not strings[EB/OL]. https://www.blog.google/products/search/introducing-knowledge-graph-things-not/, 2012

[5] Xiong C, Power R, Callan J. Explicit semantic ranking for academic search via knowledge graph embedding[M]//WWW '17: Proceedings of the 26th International Conference on World Wide Web. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017: 1271–1279

[6] Kaiser M, Saha Roy R, Weikum G. Reinforcement learning from reformulations in conversational question answering over knowledge graphs[C]//SIGIR '21: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY, USA: Association for Computing Machinery, 2021: 459–469

[7] Wang X, Huang T, Wang D, et al. Learning intents behind interactions with knowledge graph for recommendation[C]//WWW '21: Proceedings of the Web Conference 2021. New York, NY, USA: Association for Computing Machinery, 2021: 878–887

[8] Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]//SIGMOD '08: Proceedings of the 2008

ACM SIGMOD International Conference on Management of Data. New York, NY, USA: Association for Computing Machinery, 2008: 1247–1250

[9] Vrandečić D, Krötzsch M. Wikidata: a free collaborative knowledgebase[J]. Commun. ACM, 2014, 57(10): 78–85

[10] Lehmann J, Isele R, Jakob M, et al. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia[J]. Semantic Web, 2015, 6(2): 167–195

[11] Suchanek F M, Kasneci G, Weikum G. Yago: a core of semantic knowledge[C]//WWW '07: Proceedings of the 16th International Conference on World Wide Web. New York, NY, USA: Association for Computing Machinery, 2007: 697–706

[12] Wang Z, Zhang J, Feng J, et al. Knowledge graph embedding by translating on hyperplanes[C]//AAAI'14: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. : AAAI Press, 2014: 1112–1119

[13] Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion[C]//AAAI'15: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. : AAAI Press, 2015: 2181–2187

[14] Ji G, He S, Xu L, et al. Knowledge graph embedding via dynamic mapping matrix[C]// Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Beijing, China: Association for Computational Linguistics, 2015: 687–696

[15] Nickel M, Tresp V, Kriegel H P. A three-way model for collective learning on multi-relational data[C]//ICML'11: Proceedings of the 28th International Conference on International Conference on Machine Learning. Madison, WI, USA: Omnipress, 2011: 809–816

[16] Yang B, tau Yih W, He X, et al. Embedding entities and relations for learning and inference in knowledge bases[J]. CoRR, 2014, abs/1412.6575

[17] Trouillon T, Welbl J, Riedel S, et al. Complex embeddings for simple link prediction[C]// ICML'16: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. 2016: 2071–2080

[18] Liu H, Wu Y, Yang Y. Analogical inference for multi-relational embeddings[C]// ICML'17: Proceedings of the 34th International Conference on Machine Learning - Volume 70. 2017: 2168–2178

[19] Bordes A, Glorot X, Weston J, et al. A semantic matching energy function for learning with multi-relational data[J]. Mach. Learn., 2014, 94(2): 233–259

[20] Socher R, Chen D, Manning C D, et al. Reasoning with neural tensor networks for knowledge base completion[C]//NIPS'13: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1. Red Hook, NY, USA: Curran Associates Inc., 2013: 926–934

[21] Dong X, Gabrilovich E, Heitz G, et al. Knowledge vault: a web-scale approach to probabilistic knowledge fusion[C]//KDD '14: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: Association for Computing Machinery, 2014: 601–610

[22] Dettmers T, Minervini P, Stenetorp P, et al. Convolutional 2d knowledge graph embeddings[C]//AAAI'18/IAAI'18/EAAI'18: 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. : AAAI Press, 2018

[23] Jiang X, Wang Q, Wang B. Adaptive convolution for multi-relational learning[C]// Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 978–987

[24] Nguyen D Q, Nguyen T D, Nguyen D Q, et al. A novel embedding model for knowledge base completion based on convolutional neural network[C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). New Orleans, Louisiana: Association for Computational Linguistics, 2018: 327–333

[25] Vashishth S, Sanyal S, Nitin V, et al. Interacte: Improving convolution-based knowledge

graph embeddings by increasing feature interactions[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(03): 3009–3016

[26] Schlichtkrull M, Kipf T N, Bloem P, et al. Modeling relational data with graph convolutional networks[C]//The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings. Berlin, Heidelberg: Springer-Verlag, 2018: 593–607

[27] Shang C, Tang Y, Huang J, et al. End-to-end structure-aware convolutional networks for knowledge base completion[C]//AAAI'19/IAAI'19/EAAI'19: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence. : AAAI Press, 2019

[28] Cai L, Yan B, Mai G, et al. Transgcn: Coupling transformation assumptions with graph convolutional networks for link prediction[C]//K-CAP '19: Proceedings of the 10th International Conference on Knowledge Capture. New York, NY, USA: Association for Computing Machinery, 2019: 131–138

[29] Nathani D, Chauhan J, Sharma C, et al. Learning attention-based embeddings for relation prediction in knowledge graphs[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 4710–4723

[30] Zhang Z, Zhuang F, Zhu H, et al. Relational graph neural network with hierarchical attention for knowledge graph completion[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(05): 9612–9619

[31] Zhao Y, Feng H, Zhou H, et al. Eigat: Incorporating global information in local attention for knowledge representation learning[J]. Knowledge-Based Systems, 2022, 237: 107909

[32] Chen S, Liu X, Gao J, et al. HittER: Hierarchical transformers for knowledge graph embeddings[C]//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021: 10395–10407

[33] Bi Z, Cheng S, Chen J, et al. Relphormer: Relational graph transformer for knowledge

graph representations[J]. Neurocomputing, 2024, 566: 127044

[34] Guu K, Miller J, Liang P. Traversing knowledge graphs in vector space[C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, 2015: 318–327

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

[36] Das R, Neelakantan A, Belanger D, et al. Chains of reasoning over entities, relations, and text using recurrent neural networks[C]//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. Valencia, Spain: Association for Computational Linguistics, 2017: 132–141

[37] Guo L, Sun Z, Hu W. Learning to exploit long-term relational dependencies in knowledge graphs[C]//Proceedings of Machine Learning Research: volume 97 Proceedings of the 36th International Conference on Machine Learning. 2019: 2505–2514

[38] Zhang Y, Yao Q, Chen L. Interstellar: searching recurrent architecture for knowledge graph embedding[C]//NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc., 2020

[39] Yao L, Mao C, Luo Y. KG-BERT: BERT for knowledge graph completion[J]. CoRR, 2019, abs/1909.03193

[40] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 4171–4186

[41] Wang X, He Q, Liang J, et al. Language models as knowledge embeddings[C]// Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22. 2022: 2291–2297

[42] Jiang P, Agarwal S, Jin B, et al. Text augmented open knowledge graph completion via pre-trained language models[C]//Findings of the Association for Computational Linguistics: ACL 2023. Toronto, Canada: Association for Computational Linguistics, 2023: 11161–11180

[43] Leblay J, Chekol M W. Deriving validity time in knowledge graph[C]//WWW '18: Companion Proceedings of the The Web Conference 2018. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018: 1771–1776

[44] Li J, Su X, Gao G. TeAST: Temporal knowledge graph embedding via archimedean spiral timeline[C]//Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Toronto, Canada: Association for Computational Linguistics, 2023: 15460–15474

[45] Wu Z, Jain P, Wright M, et al. Representing long-range context for graph neural networks with global attention[C]//Advances in Neural Information Processing Systems: volume 34. 2021: 13266–13279

[46] Rong Y, Bian Y, Xu T, et al. Self-supervised graph transformer on large-scale molecular data[C]//Advances in Neural Information Processing Systems: volume 33. 2020: 12559–12571

[47] Ying C, Cai T, Luo S, et al. Do transformers really perform badly for graph representation? [C]//Advances in Neural Information Processing Systems: volume 34. 2021: 28877–28888

[48] Yang L, Liu Z, Dou Y, et al. Consisrec: Enhancing gnn for social recommendation via consistent neighbor aggregation[C]//SIGIR '21: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY, USA: Association for Computing Machinery, 2021: 2141–2145

[49] Xu H, Bao J, Liu W. Double-branch multi-attention based graph neural network for knowledge graph completion[C]//Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Toronto, Canada: Association for Computational Linguistics, 2023: 15257–15271

[50] Xu K, Wu L, Wang Z, et al. Graph2seq: Graph to sequence learning with attention-based neural networks[M]. 2018

[51] Vashishth S, Sanyal S, Nitin V, et al. Composition-based multi-relational graph convolutional networks[C]//8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. 2020

[52] Zhu J Z, Jia Y T, Xu J, et al. Modeling the correlations of relations for knowledge graph embedding[J]. Journal of Computer Science and Technology, 2018, 33(2): 323–334. DOI: 10.1007/s11390-018-1821-8

[53] Toutanova K, Chen D. Observed versus latent features for knowledge base and text inference[C]//Proceedings of the 3rd workshop on continuous vector space models and their compositionality. 2015: 57–66

[54] Miller G A. Wordnet: a lexical database for english[J/OL]. Commun. ACM, 1995, 38 (11): 39–41. https://doi.org/10.1145/219717.219748

[55] Sun Z, Deng Z, Nie J, et al. Rotate: Knowledge graph embedding by relational rotation in complex space[C]//7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. 2019

[56] Liu X, Zhu T, Tan H, et al. Heterogeneous graph neural network with hypernetworks for knowledge graph embedding[C]//The Semantic Web –ISWC 2022: 21st International Semantic Web Conference, Virtual Event, October 23 – 27, 2022, Proceedings. Berlin, Heidelberg: Springer-Verlag, 2022: 284–302

[57] Li R, Cao Y, Zhu Q, et al. How does knowledge graph embedding extrapolate to unseen data: a semantic evidence view[C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 36. 2022: 5781–5791

[58] Li Z, Zhao Y, Zhang Y, et al. Multi-relational graph attention networks for knowledge graph completion[J]. Knowledge-Based Systems, 2022, 251: 109262

[59] Kingma D P, Ba J. Adam: A method for stochastic optimization[C]//3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. 2015

[60] Ruffinelli D, Broscheit S, Gemulla R. You CAN teach an old dog new tricks! on

training knowledge graph embeddings[C]//8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. 2020

[61] Sun Z, Vashishth S, Sanyal S, et al. A re-evaluation of knowledge graph completion methods[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020. 2020: 5516–5522

[62] Wang B, Shen T, Long G, et al. Structure-augmented text representation learning for efficient knowledge graph completion[C]//Proceedings of the Web Conference 2021. 2021: 1737–1748

攻读硕士学位期间取得的学术成果

[1] Liu X, **Zhu T**, Tan H, et al. Heterogeneous graph neural network with hypernetworks for knowledge graph embedding [C]//The Semantic Web–ISWC 2022: 21st International Semantic Web Conference, Virtual Event, October 23–27, 2022, Proceedings. Berlin, Heidelberg: Springer-Verlag, 2022: 284–302

[2] 郭子溢, **朱桐**, 林广艳, 等. 球面坐标下基于语义分层的知识图谱补全方法 [J]. 应用科学学报, 2024, 42 (01): 119-133.

致 谢

研究生三年生涯转瞬即逝，我的学生生涯也马上要走到了尽头。回首过去的三年，我初次叩开了科研的大门，虽然未能深入探索，但也收获良多，受益匪浅。

在三年的研究生生活中，我首先要感谢的是谭火彬老师和林广艳老师对我的学术和生活上给予的指导和帮助。两位老师不仅会在学术上为我提供建议，指明方向，对我的论文提出了许多宝贵的修改意见，生活中，还时刻关心着我们的身体健康和心理状况，在我的职业规划上也提供了大力的支持。实验室的良好氛围也为我提供了非常好的学习和研究的环境。能够接受两位老师的指导是我的幸运。还要感谢我的父母，他们的默默付出是我最坚强的后盾，父母的理解和爱护是我低谷时最重要的慰藉和支持。

此外，我还要感谢实验室的各位同门。感谢刘希阳学长在我初入实验室时提供的指导，帮助我迈出了学术研究的第一步，在我论文工作遇到困难时也对我提出了非常宝贵的意见和建议。还要感谢柳啸峰、郭子溢、朱伯同三位学长在我的学术研究中为我答疑解惑，学长们丰富的经验帮助我少走了不少弯路。感谢任博林和蒋沛宇两位学弟，论文实验的顺利完成离不开他们对于实验室服务器的维护。

最后，我要感谢我的小伙伴们，三年的时间我们一起度过了一段难忘的时光。