中图分类号：**TP391.4**

论 文 编 号：**10006SY2121127**

内 部

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

摘要是学位论文内容的简短陈述，应体现论文工作的核心思想。论文摘要应力求语言精炼准确。博士学位论文的中文摘要一般约 800～1200 字；硕士学位论文的中文摘要一般约 500 字。摘要内容应涉及本项科研工作的目的和意义、研究思想和方法、研究成果和结论。博士学位论文必须突出论文的创造性成果，硕士学位论文必须突出论文的新见解。

关键字是为用户查找文献，从文中选取出来揭示全文主体内容的一组词语或术语，应尽量采用词表中的规范词（参考相应的技术术语标准）。关键词一般 3～5 个，按词条的外延层次排列（外延大的排在前面）。关键词之间用逗号分开，最后一个关键词后不打标点符号。

为了国际交流的需要，论文必须有英文摘要。英文摘要的内容及关键词应与中文摘要及关键词一致，要符合英语语法，语句通顺，文字流畅。英文和汉语拼音一律为 Times New Roman 体，字号与中文摘要相同。

关键词：北航，学位论文，博士，硕士，中文，LaTeX 模板，BUAATHESIS

Abstract

What were you doing 500 years ago? Oh, that's right nothing, because you didn't exist yet. In fact, several generations of your family had yet to leave their mark on the world, but one very special shark may already have been swimming in the chilly North Atlantic at that time, and the incredible animal is somehow still alive today.

Scientists studying Greenland sharks observed the particularly old specimen just recently, and after studying it they've determined that the creature is approximately 272 to 512 years old. That's an absolutely insane figure, and if its age lands towards the higher end, it makes the animal the oldest observed living vertebrate on the entire planet.

Greenland sharks are an incredible species in a number of ways, but most notable is its longevity. The sharks are well over 100 years old before even reaching sexual maturity, and regularly live for centuries. This particularly old specimen, along with 27 others, were analyzed using radiocarbon dating. The reading came back at around 392 years, but potential margin of error means the animal's true age is somewhere between 272 and 512.

The shark, which is a female, measures an impressive 18 feet long. That's pretty large, but it might not sound particularly large for an ocean-dwelling creature that lives hundreds of years. That is, until you consider that the Greenland shark only grows around one centimeter per year. With that in mind, 18 feet is actually downright massive.

As for how this particular shark species manages to live so incredibly long, scientists attribute a lot of its longevity to its sluggish metabolism, as well as its environment. The frigid waters where the sharks thrive is thought to increase overall lifespan in a variety of ways. Past research has shown that cold environments can help slow aging, and these centuries-old sharks are most certainly benefiting from their chilly surroundings.

— Online news *Scientists find incredible shark that may be over 500 years old and still kicking*, 12.16.2017. (http://bgr.com/2017/12/14/oldest-shark-greenland-512-years-old/).

Key words: News, BGR, Shark

目 录

图 清 单

表 清 单

第一章　绪论

1.1　研究背景与意义

论文选题来源于国家重点研发计划课题“稀土催化材料专用数据库及全流程数字化研发平台”，本文研究高效的知识图谱补全方法，为稀土催化材料知识图谱构建和应用提供技术支撑。

稀土是由镧系元素和与其密切相关的钪和钇等化学元素组成，稀土的存在能有效提高催化剂的储氧能力、提高活性金属的分散度、降低贵金属用量、促进水气转化和蒸汽重整反应等性质，在催化材料领域中有着重要的应用。稀土催化材料知识图谱存储着稀土催化材料的合成方式、理化性质、组成结构等重要信息，被应用于稀土催化材料合成等任务中，能够帮助稀土催化材料降低开发成本、减少开发周期。然而，由于稀土材料领域的不断研究发展带来的知识的动态变化，和大多数知识图谱一样，稀土催化材料知识图谱往往是不完全的，难以囊括领域内的所有知识，对其在上层任务中的应用造成了阻碍。因此，为了挖掘图谱内蕴含的丰富信息，完善稀土催化材料知识图谱，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。

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，结合知识图谱中的局部邻域和图路径两种图结构来学习图谱中的短距离依赖、长距离依赖乃至全局信息，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：

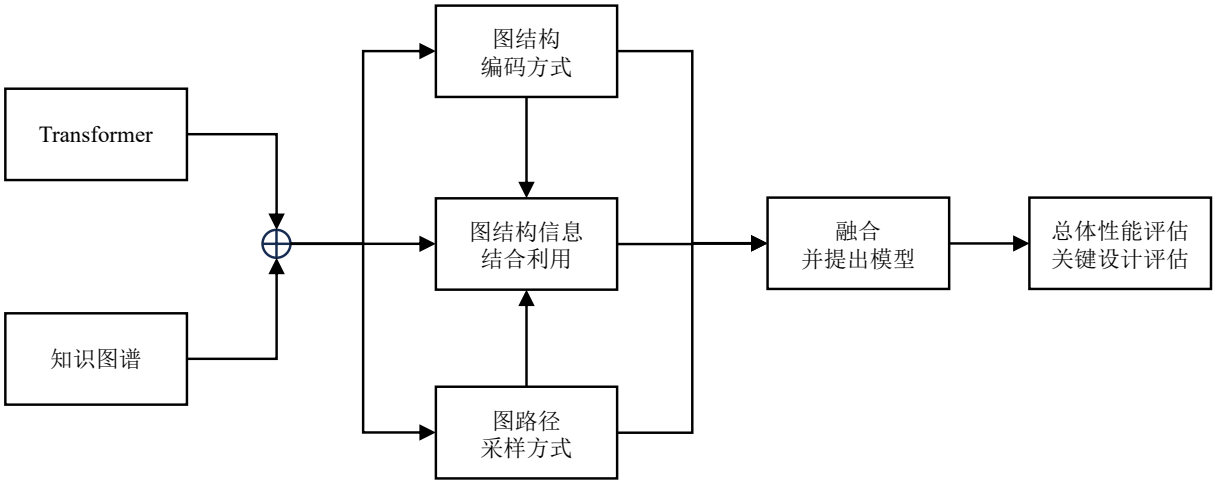

图 1 研究路线示意图

（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。

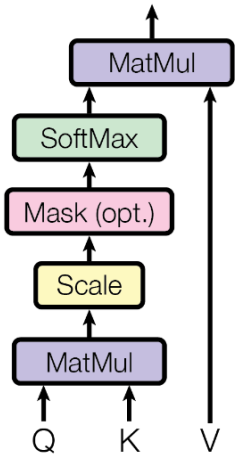

图 2 缩放点积注意力机制

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。

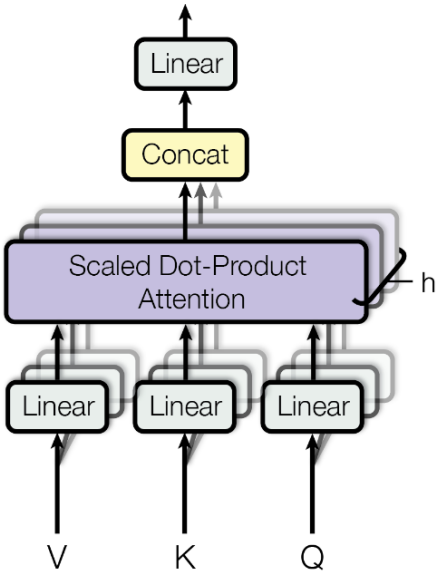

图 3　多头注意力机制

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：$\mathcal{T} = (v_s, v_p, v_o)$，中心节点的局部领域为：

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

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 NATLP 模型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。

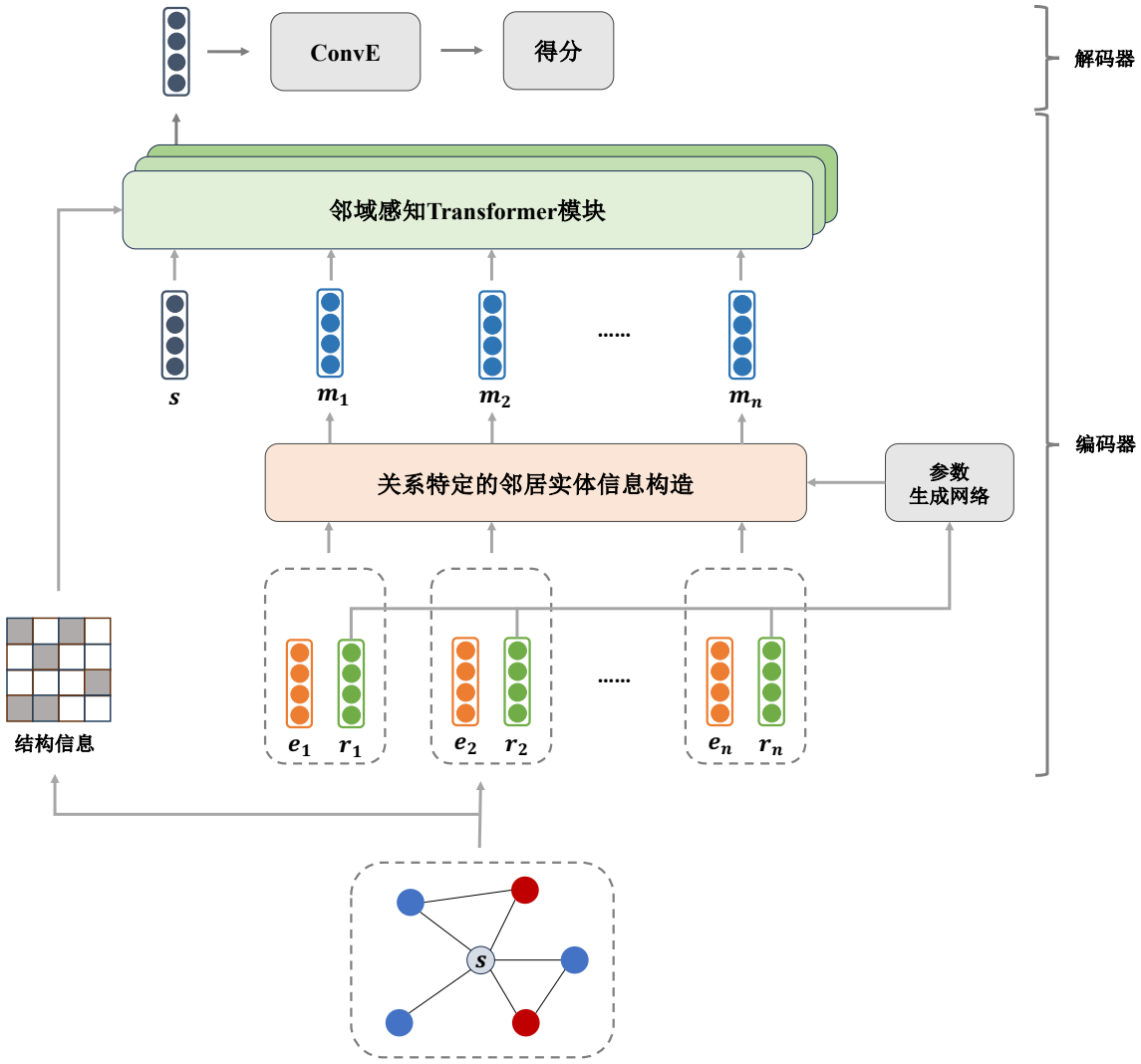

图 4 关系特定的邻居实体信息构造

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。

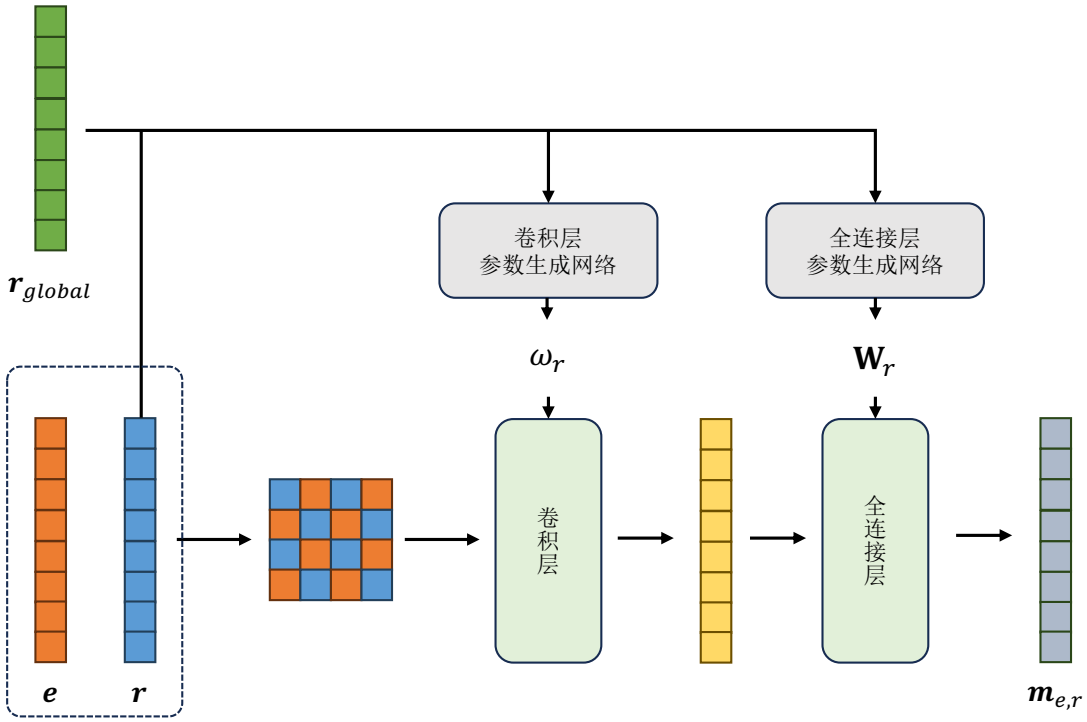

图 5 关系特定的邻居实体信息构造

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）关系特定了邻居实体信息构造，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。

总结与展望

学位论文的结论单独作为一章，但不加章号。如果不可能导出应有的结论，也可以没有结论而进行必要的讨论。

* 嗯，这就是你的论文了 *

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

攻读硕士学位期间取得的学术成果

对于博士学位论文，本条目名称用“攻读博士学位期间取得的研究成果”，一般包括：

攻读博士学位期间取得的学术成果：攻读博士学位期间取得的学术成果：列出攻读博士期间发表（含录用）的与学位论文相关的学位论文、发表专利、著作、获奖项目等，书写格式与参考文献格式相同；

攻读博士期间参与的主要科研项目：列出攻读博士学位期间参与的与学位论文相关的主要科研项目，包括项目名称，项目来源，研制时间，本人承担的主要工作。

对于硕士学位论文，本条目名称用“攻读硕士学位期间取得的学术成果”，只列出攻读硕士学位期间发表（含录用）的与学位论文相关的学位论文、发表专利、著作、获奖项目等，书写格式与参考文献格式相同。

* 嗯，研究生不列科研项目 *

致 谢

致谢中主要感谢指导教师和在学术方面对论文的完成有直接贡献及重要帮助的团体和人士，以及感谢给予转载和引用权的资料、图片、文献、研究思想和设想的所有者。致谢中还可以感谢提供研究经费及实验装置的基金会或企业等单位和人士。致谢辞应谦虚诚恳，实事求是，切记浮夸与庸俗之词。

* 嗯，感谢完所有人之后，也请记得感谢一下自己 *