重庆师范大学硕士学位论文

提高长尾数据知识图谱补全性能的算法研究

硕士研究生：何苗惠
指 导 老 师：吴至友 教授
学 科 专 业：运筹学与控制论
所 在 学 院：数学科学学院

重庆师范大学

二〇二二年五月

A Thesis Submitted to Chongqing Normal University in Partial Fulfillment of the Requirements for the Degree of Master

Research on Algorithm for Improving Knowledge Graph Completion Performance for Long-tail Data

Candidate: He Miaohui
Supervisor: Professor Wu Zhiyou
Major: Operational Research & Cybernetics
College: School of Mathematical Sciences

Chongqing Normal University

May, 2022

提高长尾数据知识图谱补全性能的算法研究

摘 要

知识图谱是通过结构化的形式表示现实世界事实的集合，是信息智能化的基础。但知识图谱往往只包含部分事实，数据具有不完备性，该缺陷使得很多知识图谱在实际应用中受到了极大的限制，例如智能问答。因此需要对知识图谱进行补全，而基于知识图谱嵌入进行链接预测任务就是实现自动补全的重要方法之一。因此，如何通过知识图谱嵌入使其能够更加精确地补全知识图谱，提高其推理的准确性具有极其重要的研究价值。

随着众多学者和行业对知识图谱的关注和研究，许多知识图谱嵌入方法相继被提出。但知识图谱的数据常呈长尾分布，当前的大部分方法对非长尾数据补全性能较好，对长尾数据的补全性能较差。因此本文主要研究如何通过改进现有方法，进一步提高长尾数据的补全性能，主要研究内容如下：

1. 提出了一种融入期望最大化算法思想的知识图谱嵌入方法 (EM-KGE)：将非长尾数据中包含的丰富的语义知识作为监督知识，将该知识转移到长尾数据中，以此来增强长尾数据的语义表达，提高其补全性能。因此 EM-KGE 方法将期望最大化算法思想融入知识图谱嵌入方法中，利用 EM 算法中的隐变量来建立非长尾数据和长尾数据的联系，把非长尾数据中的冗余实体作为隐变量，通过交替更新实现知识的转移，该做法在一定程度上能够提高长尾数据的补全性能。并通过数值实验验证了 EM-KGE 方法的有效性。

2. 提出了一种融入期望最大化算法思想的双重嵌入方法 (DEM)：针对 EM-KGE 方法仅利用冗余实体无法准确传递非长尾数据的语义知识的问题，DEM 方法进一步提出双重嵌入技术，包括原实体、原关系嵌入和潜在语义嵌入，将潜在语义嵌入作为隐变量来实现非长尾数据的知识转移，从而进一步达到提高长尾数据补全性能的目的。并通过数值实验验证了 DEM 方法的有效性。

3. 提出了一种在 DEM 方法中引入相似度计算的 SDEM 方法：该方法引入相似度计算来改进 DEM 方法的目标函数，通过极小化该目标函数，使得在嵌入空间中，长尾数据中的三元组与其在非长尾数据中相似的三元组的向量表示能够更加接近，从而促使长尾数据补全性能进一步提升。并通过数值实验验证了 SDEM 方法的有效性。

关键词: 知识图谱补全; 知识图谱嵌入; 长尾数据; 期望最大化算法; 双重嵌入方法

Research on Algorithm for Improving Knowledge Graph Completion Performance of Long-tail Data

ABSTRACT

Knowledge graphs represent the facts of the real world, and is the basis for information intelligence. However, knowledge graphs often only contain part of the facts, and the data is incomplete. This defect greatly limits practical applications of many knowledge graphs, such as intelligent question answering. Therefore, it is necessary to complete the knowledge graph, and the link prediction task based on knowledge graph embedding methods is one of the important methods to achieve automatic completion. Therefore, how to use the knowledge graph embedding methods to make it more accurate to complete the knowledge graphs and improve the accuracy of its reasoning has extremely important research value.

Many scholars and industries have paid attention to and studied knowledge graphs, and proposed many knowledge graph embedding methods. However, the data distribution of the knowledge graphs trends long-tail. Most of the existing methods have good completion performance for non-long-tail data, but poor completion performance for long-tail data. Therefore, this paper focus on how to further improve the completion performance of long-tail data for existing methods. The main research contents are as follows:

1. This paper proposes a knowledge graph embedding method incorporating the idea of the Expectation Maximization algorithm (EM-KGE): The rich semantic knowledge contained in the non-long-tail data is used as supervised knowledge, and the knowledge is transferred to the long-tail data, so as to improve the long-tail data completion performance. Therefore, the EM-KGE method integrates the Expectation Maximization algorithm (EM) idea into the knowledge graph embedding method, and uses the latent variables in the EM algorithm to establish the connection between non-long-tail data and long-tail data. Taking redundant entities in non-long-tail data as latent variables, and realizing knowledge transfer through alternate updating, can improve the completion performance of long-tail data to a certain extent. The effectiveness of the EM-KGE method is verified by numerical experiments.

2. This paper proposes a dual embedding method that incorporates the idea of the Expectation Maximization algorithm (DEM): The EM-KGE method cannot accurately transfer the semantic knowledge of non-long-tail data by only using redundant entities. Aiming at this problem, the DEM method proposes dual embedding technology, including original entity embedding, original relation embedding and latent semantic embedding. This method uses latent semantic embedding as a latent variable to realize the knowledge transfer of non-long-tail data, thereby further improving the performance of long-tail data completion. The effectiveness of the DEM method is verified by numerical experiments.

3. This paper proposes an SDEM method that introduces similarity calculation into the DEM method: The method introduces similarity calculation to improve the objective function of the DEM method. In the embedding space, the vector representation of triples in long-tail data is closer to that of similar triples in non-long-tail data, which further improves the performance of long-tail data completion. The effectiveness of the SDEM method is verified by numerical experiments.

Keywords: Knowledge Graph Completion; Knowledge Graph Embedding; Long-tail data; Expectation Maximization Algorithm; Dual embedding method

目 录

1 绪论

1.1 研究背景及意义

近年来，人工智能和深度学习以及大数据的发展极其迅猛，互联网上的数据和知识呈爆炸式增长，人类开始进入大数据时代[1]。随着信息技术的更新和发展，人们也开始不断地追求着智能化的生活。知识图谱 (Knowledge Graphs，简称 KGs) 是大数据时代中储存、管理、理解知识数据的有效方式之一[2]，该技术使得知识信息变得结构化和智能化，且能够给机器的智能认知提供有效的知识基础。因此，掀起了一股知识图谱[3–7]的研究热潮，是否掌握成熟的知识图谱技术也成为各行业能否走上智能化道路和能否进一步升级的重要环节。

图1.1: 知识图谱在Google搜索上应用示例 1

在 2012 年，Google 首次提出知识图谱的概念[8]，并将该技术应用至 Google Chrome 引擎中，使得人们在进行搜索后不仅可以获得直接的结果，还可以得到与之相关的信息，使得获得的信息更全面。融入该技术到搜索引擎中极大满足了人们对搜索的需求，提高了用户的使用感受和搜索质量。如图 1.1 所示，当在 Google Chrome 引擎中搜索“比尔·盖茨”时，搜索的整个界面被分为了两块，其中左侧展示了与“比尔·盖茨”相关联的网页链接，右侧展示了他的出生地、出生时间等事实信息。这些客观的信息的获得得益

于知识图谱的技术，使得用户在进行主题搜索时不需要链接到其他网页便可获得关联的客观主题信息。除此之外，若用户想查询更为复杂的问题，如图 1.2 所示，搜索“比尔·盖茨的前妻是谁”，Google 便能精准的返回“梅琳达”。由此可见利用知识图谱可以辅助搜索引擎解析知识间的联系。随后，知识图谱技术便开始运用至人工智能领域的众多应用中，例如智能问答 [9,10]、机器阅读 [11,12]、智能医疗 [13,14]、推荐系统 [15,16]等。

图1.2: 知识图谱在Google搜索上应用示例 2

知识图谱是一种结构化的语义知识库，通过网状结构来描述现实世界中的事物和它们之间存在的关系 [17]，网状结构的两个基本元素就是节点和边。其中节点代表实体 (entity)，该实体或概念既可以是具体实体，例如“银杏树”，也可以是抽象概念，例如“植物”；节点与节点之间的边代表实体之间存在的关系 (relation)，例如夫妻关系、姐弟关系，也可以是实体属性，例如民族、籍贯。知识图谱通过三元组 (h, r, t) 的形式来存储客观世界中的知识，其中 h 和 t 分别表示头实体和尾实体，r 表示头实体和尾实体之间的关系。三元组主要包含两种形式：(实体，属性，属性值) 和 (实体，关系，实体)。如图 1.3 所示，三元组 (比尔 • 盖茨，出生地，西雅图) 描述了“比尔 • 盖茨出生于西雅图”的事实，其中头实体 h 是节点“比尔 • 盖茨”，尾实体 t 是节点“西雅图”，头实体与尾实体之间的关系就是连接节点“比尔 • 盖茨”与节点“西雅图”之间的边——“出生地”。三元组可用于描述客观世界中的大量事实，这些众多三元组聚集在一起就构成了复杂而庞大的知识图谱。

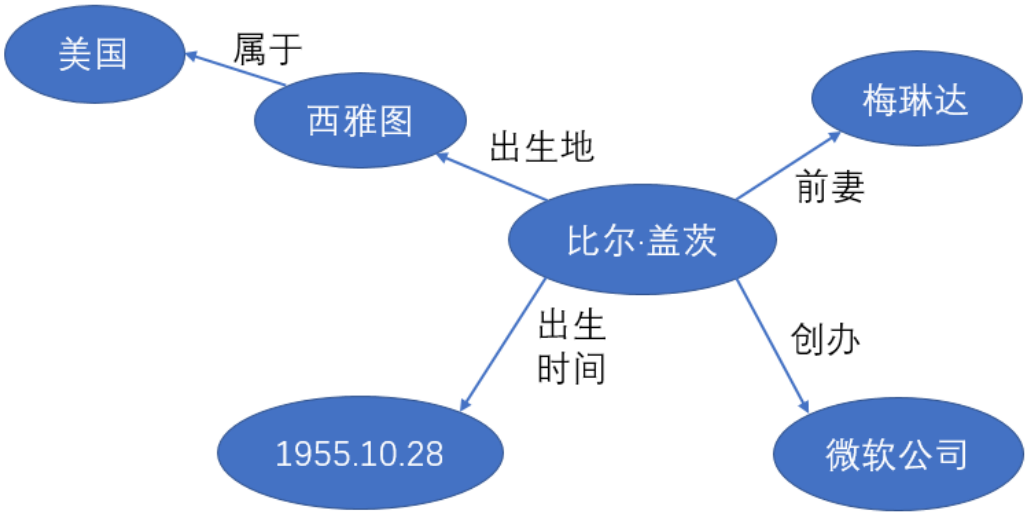

图1.3: 知识图谱示例

随着对知识图谱的深入研究，越来越多的 KGs 被人们所构建，根据组织或社区的不同将其分为两类 [18]：开放知识图谱 (Open Knowledge Graph) 和企业知识图谱 (Enterprise Knowledge Graph)。开放知识图谱通常是在线发布，其内容为公众所用，例如 DBpedia [19,20]、BabelNet [21]、Freebase [22]、YAGO [23–25]、Wikidata [26] 以及 ConceptNet [27]等。企业知识图谱常常是公司内部发布，其内容为商用，例如亚马逊 [28]、LinkedIn [29]、Facebook [30]、淘宝 [31]等。这些 KGs 构建了真实世界中的知识结构，成为支撑人工智能发展的基础技术，支持智能决策、智能推荐、智能问答等智能信息服务的应用。然而知识图谱在实际运用中仍存在缺陷，其中最主要的问题是内部信息不完备。例如，即使是包含数千个关系、数千万个实体、超过 20 亿个三元组的庞大知识图谱 [32]——Freebase，也依旧有 75% 的人缺少关于国籍的信息，约 70% 的人缺少他们的出生地信息。然而，国籍和出生地还是常见的关系，那些少见的关系必然缺乏更多信息 [33]。这极大限制了知识图谱在实际中的应用，例如在问答场景中无法回答缺失的知识或只能给出错误答案。因此为了使得 KGs 包含的知识更加完整，需对其中的知识进行完善补充，即知识图谱补全 (Knowledge Graph Completion)。知识图谱补全是指向不完整的知识图谱中增添未知的知识，同时保证增加的知识是正确的。例如图 1.4 中，实线的边代表该知识图谱中已存在的关系，虚线的边代表可以通过知识图谱补全来获得的未知的关系，从而增添新的知识。基于知识图谱嵌入进行链接预测任务便是实现知识图谱补全的一种重要方法。知识图谱嵌入 (Knowledge Graph Embedding，简称 KGE) 是指利用稠密的低维实值向量对 KGs 中的实体和关系进行编码，从而实现将知识图谱嵌入到一个低维空间中，既能保留知识图谱的结构信息和语义信息，也可以进行高效的计算，实现知识图谱补全的自动化。由于实体和关系向量包含了大量的语义信息，便可对向量进行推理，从而预测出知识图谱中缺失的知识。

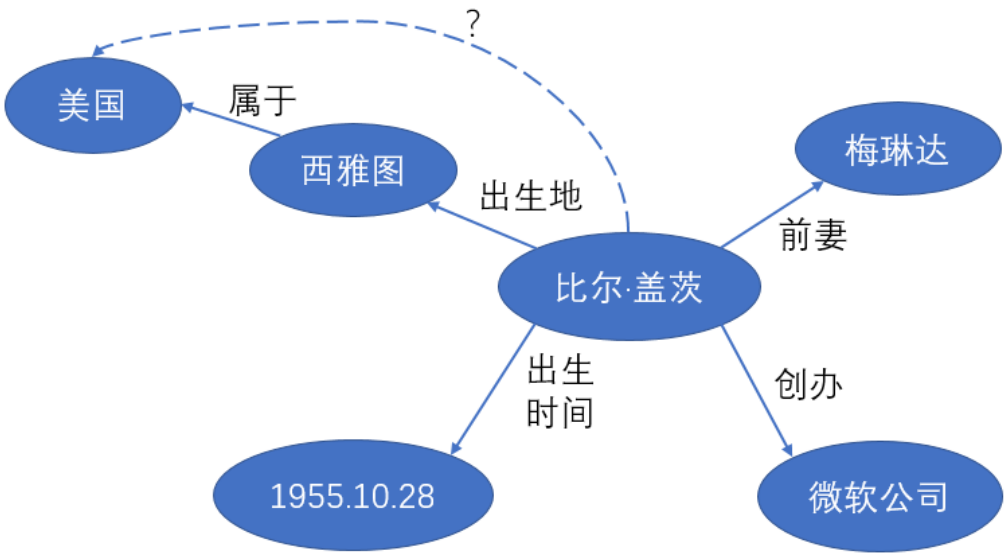

图1.4: 知识图谱补全示例

知识图谱仅含有真实世界中的一小部分知识，如何通过已包含的知识去捕获未知的知识，如何通过改进算法和模型去进一步改善向量表示，使得实体和关系向量能够表达真实的语义信息，从而提高知识图谱的补全性能是一个十分重要的研究课题。

1.2 研究现状及挑战

知识图谱嵌入作为知识图谱领域中的重要研究之一，目前已有众多成果。这类知识图谱嵌入方法可以分为两大类：融入事实信息的知识图谱嵌入方法和融入附加信息的知识图谱嵌入方法 [32,34]。

融入事实信息的知识图谱嵌入方法只利用知识图谱中的结构信息，包括基于翻译的 KGE 方法、双线性 KGE 方法、基于神经网络的 KGE 方法、基于旋转的 KGE 方法等。基于翻译的 KGE 方法包括 TransE [36]、TransH [37]、TransR [38] 和 TransD [39] 等改进模型 [40–44]，这类方法将三元组 (h, r, t) 中的 r 看作是 h 到 t 的翻译操作。双线性 KGE 方法包括 RESCAL [45]、DistMult [46]、ComplEx [47] 等，通过捕获向量空间中的实体和关系之间的潜在语义相似度来得到三元组得分。基于神经网络的 KGE 方法 [48–52]包括 ConvE [48]、ConvKB [49] 和 CapsE [50] 等，通过直接利用不同的神经网络来计算三元组的置信度。基于旋转的 KGE 方法，包括 RotatE [53] 和 QuatE [54] 等，它们通过利用欧拉公式将 r 作为 h 到 t 的旋转来定义三元组的评分函数。除此之外，融入附加信息的知识图谱嵌入方法还利用其他有效信息来进一步改进评分函数。例如，实体所属的语义类别 [55,56]，包含语义线索的关联路径 [57–59]，丰富的文本信息 [60–62] 以及逻辑规则 [63] 等。

文献 [64] 对当前的 KGE 方法进行了大量的分析实验，发现现有知识图谱中存在较多的冗余关系。本文将其中关系是冗余关系的数据定义为非长尾数据，该数据之间的相关性较大、语义重复度较高且相同实体和关系出现频率较高；将关系不是冗余关系的数

据定义为长尾数据，该数据之间的相关性较小、语义重复度较低且相同实体和关系出现频率较低。作者指出，当三元组的关系互为反向关系，具体地，在三元组 (h_1, r_1, t_1) 和 (h_2, r_2, t_2) 中，若 $h_2 = t_1$、$t_2 = h_1$，则关系 r_1 和 r_2 互为反向关系。模型便会偏向去判断关系 r_1 与 r_2 是否具有反向关系的特征，而无法表达其中的语义知识，使得不具有冗余关系的长尾数据的补全效果较差。文章中列举了较多的实验结果对比：(1) 数据集 FB15K 中包含反向关系，如果仅使用统计方法，预测排序在前 10 的百分比约 71.3% ，而使用当前较好的 RotatE 方法，其百分比也仅有 73.8%，提升并不大；(2) 对比数据集 FB15K 和数据集 FB15K-237 的实验结果，其中 FB15K 包含冗余关系，FB15K-237 除去了冗余关系，发现大部分当前的 KGE 方法在 FB15K-237 上的结果差于 FB15K 的结果，如 TransE 方法预测排序在前 10 的百分比结果从 62.4% 降至 47.5%，RotatE 方法预测排序在前 10 的百分比结果从 88.1% 降至 53.2%。由此可以得出大部分当前的 KGE 方法针对非长尾数据的补全效果较好，但对长尾数据的补全效果较差。因此如何通过改进算法进一步改善长尾数据的向量表示，提高长尾数据自动化知识图谱补全的性能，从而提高下游任务如智能问答，推荐系统的性能是一项非常有意义的工作。

本文将利用非长尾数据中丰富的语义信息来进一步增强长尾数据的语义表达，提高其补全性能。其中第一个问题是如何设计合理的算法框架，使得可以将非长尾数据的语义知识转移到长尾数据中；第二个问题是如何建立非长尾数据和长尾数据之间的联系，使得可以更好、更准确地实现知识转移；第三个问题是如何在目标函数方面更好地描述两个数据之间的结构信息，从而利用非长尾数据良好的补全性能来促进长尾数据补全性能的提升。为解决以上几个问题，本文将展开以下具体研究。

1.3 研究内容

针对现有的大多数知识图谱嵌入方法对长尾数据的补全性能较差的问题，本文基于现有的方法进一步改进其算法，实现提高知识图谱嵌入方法对长尾数据补全性能的目的。所以，本文的主要贡献如下：

- 本文将非长尾数据中蕴含的丰富的语义知识作为不可观测的监督知识，将长尾数据作为无监督数据，融入期望最大化算法思想来设计算法框架，利用其中的隐变量建立非长尾数据和长尾数据的联系，并将非长尾数据中的冗余实体作为隐变量，通过交替更新将非长尾数据中的知识迁移到长尾数据中，从而提高长尾数据的补全性能。

- 本文在内容一的基础上，进一步提出双重嵌入技术，包括原实体、原关系嵌入和潜在语义嵌入，并融入期望最大化算法思想，利用其中一重嵌入——潜在语义嵌入，作为隐变量建立非长尾数据和长尾数据之间的联系，以此达到更好的非长尾数据的信息转移效果，进而提高长尾数据的补全性能。

- 本文在内容二的基础上，进一步引入相似度计算来改进长尾数据的目标函数，通过使长尾数据中的三元组与其在非长尾数据中相似的三元组的表示能够更加接近，从而

促使长尾数据的补全性能进一步提高。

- 在基准数据集 FB15K 上，在链接预测任务上利用常用的评估指标对方法的效果进行评估。将 TransE，TransH 和 TransD 作为基准方法进行相应改进，并进行对比和分析，验证本文工作的有效性。

1.4 文章结构安排

本文共分为六个章节，本文的结构编排如下：

第一章：一方面，简述本文的研究背景，介绍知识图谱，进而通过知识图谱数据的不完备问题，引出知识图谱补全和知识图谱嵌入；另一方面，介绍知识图谱嵌入方法的研究现状，根据目前的相关研究存在的问题，阐述本文的研究内容，并对本文的具体研究内容进行简述。

第二章：对知识图谱嵌入方法进行综述；对知识图谱中存在的长尾数据和非长尾数据进行介绍；对本文研究内容所采用的期望最大化算法进行介绍。

第三章：介绍本文提出的融入期望最大化算法思想的 KGE 方法 (EM-KGE)，并通过丰富的数值实验验证该方法的有效性。

第四章：介绍本文提出的融入期望最大化算法思想的双重嵌入方法 (DEM)，并通过实验验证 DEM 方法的有效性。

第五章：介绍本文提出的在 DEM 方法中引入相似度计算的 SDEM 方法，并通过数值实验验证该方法的有效性。

第六章：对本文工作的总结与对未来工作的展望。

2 相关工作

2.1 知识图谱嵌入方法

作为知识图谱领域的重要研究方向之一，知识图谱嵌入的工作目前已取得了许多成果。这些嵌入方法大致可以分为两类：融合事实信息的嵌入方法和融合附加信息的嵌入方法。本章节将主要介绍这两类嵌入方法的研究现状。

先简单介绍知识图谱嵌入中常用的符号表示：使用 $\mathcal{G}$ 表示一个知识图谱，三元组 (h,r,t) 表示知识图谱中的数据，其中 $h,t \in \mathcal{E}$ 分别表示头实体和尾实体，$\mathcal{E}$ 表示实体集，$r \in \mathcal{R}$ 表示实体间的关系，$\mathcal{R}$ 表示关系集。

2.1.1 融合事实信息的 KGE 方法

融入事实信息的知识图谱嵌入方法主要通过知识图谱本身的事实信息构建模型，分为以下几种：基于翻译的 KGE 方法、双线性 KGE 方法、基于神经网络的 KGE 方法和基于旋转的 KGE 方法等。

2.1.1.1 基于翻译的 KGE 方法

基于翻译的 KGE 方法，通过将 r 看作是从 h 到 t 的翻译，并使用基于距离的函数来作为评判三元组置信度的评分函数。最初，Bordes 等人 [36]提出 TransE 模型，他们将实体和关系嵌入到同一个空间中。给定一个三元组 (h,r,t)，TransE 方法将 r 当作是 h 到 t 的一个翻译操作，认为 $h+r$ 与 t 在空间中应尽可能相似，即 $h+r \approx t$。例如：$v\,(king) - v\,(man) \approx v(royal)$ 和 $v\,(queue)\ - v\,(woman) \approx v(royal)$。因此 TransE 将 $h+r$ 与 t 之间的距离定义为评分函数：

$$f_r(h,t) = \|h + r - t\|_{L_n}. \tag{2.1}$$

其中 L_n 可以选择 L_1 范数或 L_2 范数。该分数越低代表该三元组的置信度越高。

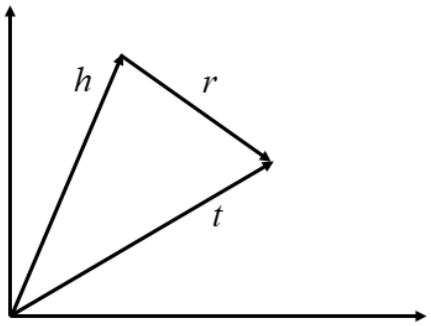

图2.1: TransE实体和关系空间

但 TransE 对复杂关系的处理效果不好，例如在 (中国，直辖市，北京市)、(中国，直辖市，重庆市) 两个三元组中，当头实体 h 和关系 r 一样时，TransE 便会认为尾实体 t 也应该是一样，显然这在实际情况中并不成立。因此 Wang 等人 [37]进一步提出 TransH 模型，它将头实体和尾实体映射在关系所在的超平面内。在进行投影操作后，当头实体 h 和关系 r 是一样时，只需要保证尾实体 t 在关系所在的超平面的投影相同即可，从而可以使得尾实体 t 具有不同的嵌入向量。因此该做法可以改善对复杂关系的处理效果。TransH 的评分函数为：

$$f_r(h,t) = \|h_\perp + r - t_\perp\|_{L_n}. \tag{2.2}$$

其中 $h_\perp = h - w_r^T h w_r$，$t_\perp = t - w_r^T t w_r$，$w_r$ 表示关系特定的超平面的法向量，$h_\perp$ 表示头实体在关系所在超平面上的投影，$t_\perp$ 表示尾实体在关系所在超平面上的投影。

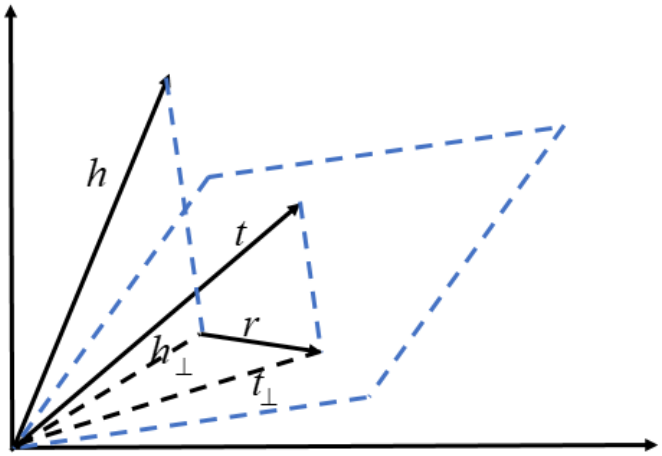

图2.2: TransH实体和关系空间

但是，TransE 模型和 TransH 模型都将实体和关系嵌入到相同的空间进行模型构建，但是在实际中，实体会存在不同的属性。当关系不同时，实体所表现的属性也不尽相同，故将其嵌入到同一空间是有偏差的。因此 TrnasR 模型 [38]分别为实体和关系构建了两个不同的嵌入空间，如图 2.3 所示。具体地，针对每一个三元组 (h,r,t)，h 和 t 都在实体空间中，然后定义一个关于关系 r 的投影矩阵 M_r，根据这个投影矩阵将实体 h 和 t 投影到关系空间中，得到对应的 h_r 和 t_r，则：

$$h_r = hM_r. \tag{2.3}$$

$$t_r = tM_r. \tag{2.4}$$

因此 TransR 将评分函数定义为：

$$f_r(h,t) = \|h_r + r - t_r\|_{L_n}. \tag{2.5}$$

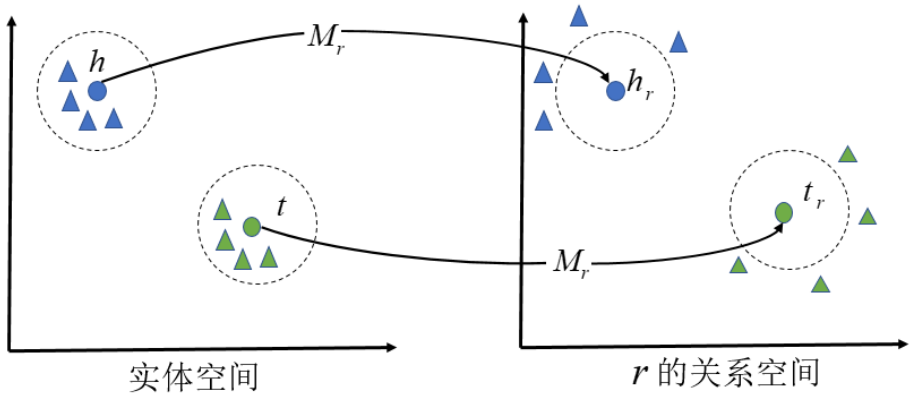

图2.3: TransR实体和关系空间

与 TransE 和 TransH 相比，TransR 有了较大的改进，但它依旧存在一些弊端：(1) 在三元组 (h,r,t) 中，h 和 t 都使用的同一个投影矩阵进行投影，但头、尾实体可能具有不同的属性。例如在三元组 (霸王别姬，导演，陈凯歌) 中，实体“霸王别姬”和实体“陈凯歌”的属性是不一样的，前者是电影，后者是人物。(2) TransR 的投影矩阵仅根据关系来设置，但实际上应该与实体和关系都有关联性，因此 TransR 的投影矩阵设置不合理。(3) TransR 模型的复杂度较高，训练耗时教久，成本较高。因此 Ji 等人 [39]提出改进模型 TransD，在一定程度上解决了以上问题。该模型针对头实体和尾实体的不同投影过程，分别定义了 2 个投影矩阵：

$$M_{rh} = r_p h_p^T + I. \tag{2.6}$$

$$M_{rt} = r_p t_p^T + I. \tag{2.7}$$

其中，$h_p \in R^n$、$t_p \in R^n$ 和 $r_p \in R^m$ 分别表示与 h、t 和 r 相关的投影向量，$I \in R^{m\times n}$ 是单位矩阵，由此可见，该投影矩阵由实体和关系共同决定。然后利用对应的投影矩阵将 h 和 t 从实体空间投影到关系空间中：

$$h_\perp = M_{rh} h. \tag{2.8}$$

$$t_\perp = M_{rt} t. \tag{2.9}$$

最终 TranD 将评分函数定义为：

$$f_r(h,t) = \|h_\perp + r - t_\perp\|_{L_n}. \tag{2.10}$$

TransD 模型既考虑了实体和关系之间的交互，也减少了参数的数量，提高了模型训练的效率。

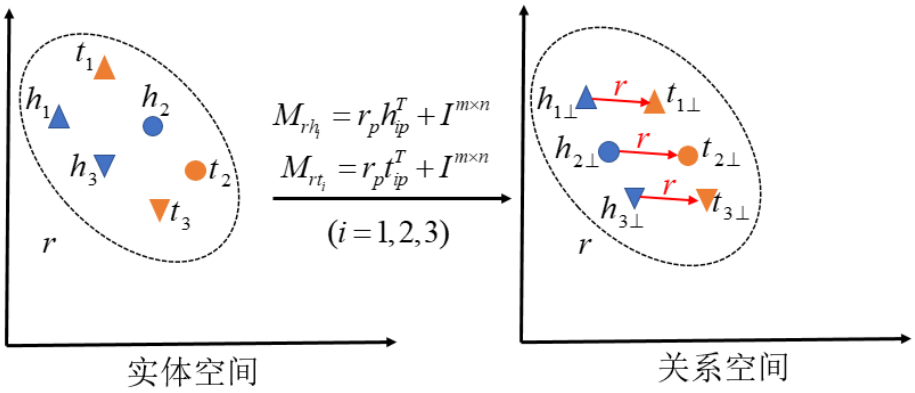

图2.4: TransD实体和关系空间

因为 TransE 模型较简单，且对大部分 KGs 都有效，因此许多学者针对 TransE 模型都进行了丰富的研究工作，除了以上介绍的几种较常见的 Trans 系列的模型，还有其余基于 TransE 模型的改进方法，例如：TransA [40]、TransG [41]、KG2E [42]、TranSparse [43]以及 TransM [44]。它们都从不同的角度去进一步优化模型，提高模型的语义表达能力。

2.1.1.2 双线性 KGE 方法

双线性 KGE 方法通过捕获向量空间中的实体和关系之间的潜在语义相似度来得到三元组得分。RESCAL 模型 [45]使用满秩矩阵来表示关系，通过实体和关系的矩阵运算去获得三元组潜在的置信度。它的评分函数为：

$$f_r(h,t) = h^T M_r t. \tag{2.11}$$

其中 h，t 表示头尾实体，M_r 表示关系特定的满秩矩阵。

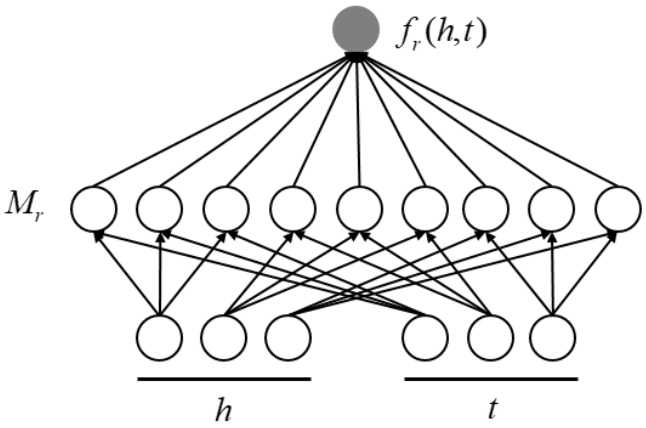

图2.5: RESCAL

由于 RESCAL 模型使用满秩矩阵表示关系，随着该矩阵维度的增长会导致计算复杂度过高，对大规模知识图谱的适用度较低，且易过拟合。因此 DisMult 模型 [46]使用对

角矩阵表示关系，从而达到简化 RESCAL 模型的效果，其评分函数为：

$$f_r(h,t) = h^T diag(M_r)t. \tag{2.12}$$

其中 $diag(M_r)$ 表示 M_r 为对角矩阵。

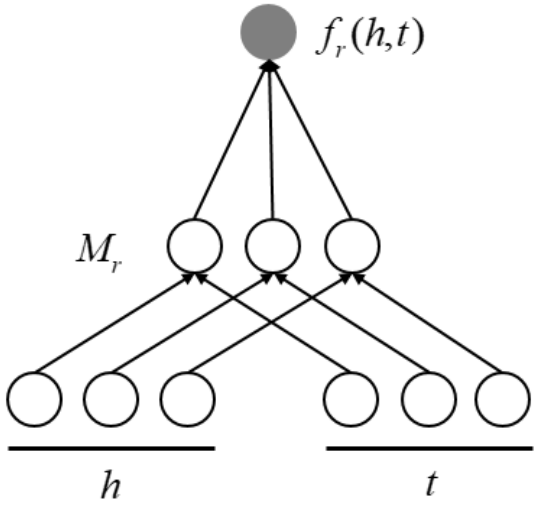

图2.6: DistMult

然而， DisMult 模型简化后的关系对角矩阵使得该模型只能处理对称关系。为此，Trouillon 等人 [47]提出 ComplEx 模型，该模型在复数空间中进行建模，利用复值向量和复对角矩阵表示实体和关系，该做法使其能进一步处理非对称关系。ComplEx 的评分函数定义为：

$$f_r(h,t) = \mathrm{Re}(h^T diag(M_r)\bar{t}). \tag{2.13}$$

其中 $h \in \mathbb{C}^d$，$t \in \mathbb{C}^d$，$M_r \in \mathbb{C}^{d\times d}$，$\bar{t}$ 是 t 的共轭，$\mathrm{Re}(\cdot)$ 表示复数的实部。

2.1.1.3 基于神经网络的 KGE 方法

近年来，随着神经网络越来越火热，不少学者开始利用神经网络来处理知识图谱嵌入问题。2018 年，Dettmers 等人 [48]提出 ConvE 模型，该模型将高效且快速的卷积神经网络运用到知识图谱嵌入中，具体的评分函数定义如下：

$$f_r(h,t) = \sigma(\varphi(vec(\varphi([\hat{h};\hat{r}] * \omega))W)t). \tag{2.14}$$

其中 $\hat{h}$ 和 $\hat{r}$ 分别表示 h 和 r 重构后的二维向量表示，即 $h,t \in \mathbb{R}^d$，$\hat{h},\hat{t} \in \mathbb{R}^{d_w\times d_h}$ 且 $d_w \times d_h = d$，$[\hat{h};\hat{r}]$ 表示将 $\hat{h}$ 和 $\hat{r}$ 进行拼接，并将其结果作为卷积层的输入，$*$ 表示卷积操作，ω 表示卷积核，$vec(X^{c\times m\times n})$ 表示将张量 X 重构为一个向量，其向量的维度是 cmn，$\varphi(x) = \max(0,x)$ 是一个非线性激活函数，W 表示矩阵，$\sigma(x) = \frac{1}{1+e^{-x}}$。

随后，Nguyen 等人 [49]提出 ConvKB 模型，依旧采用卷积神经网络进行建模，该模型可以捕获知识库中实体和关系之间的全局关系和过渡特征。在 ConvKB 中，每个三元组 (h,r,t) 都表示为一个 3 列矩阵，其中每个列向量代表一个三元组元素。然后将这个 3 列矩阵馈送到卷积层，在该卷积层上对矩阵进行多个过滤器操作以生成不同的特征。然后将这些特征连接成单个特征向量，再通过点积与权重向量相乘以返回分数。然后使用该分数来预测三元组是否有效，具体如下：

$$f(h,r,t) = concat(g([v_h, v_r, v_t] * \Omega)) \cdot w. \tag{2.15}$$

其中 v_h、v_r 和 v_t 分别表示头实体、关系和尾实体的嵌入向量，$[v_h, v_r, v_t]$ 表示将 v_h、v_r 和 v_t 进行拼接得到 $d \times 3$ 的矩阵，d 表示嵌入维度，$g(x) = \max(0, x)$ 是一个非线性激活函数，$*$ 表示卷积操作，Ω 表示卷积核。

2019 年，文献 [44] 提出了 CapsE 方法，该模型进一步将胶囊神经网络运用至知识图谱嵌入建模中，其评分函数为：

$$f_r(h,t) = \|capsnet(g([h,r,t] * \Omega))\|. \tag{2.16}$$

其中 h、r、t 是 k 维向量表示，$[h,r,t]$ 表示将 h、r、t 进行拼接得到 $k \times 3$ 维的矩阵，$*$ 表示卷积操作，Ω 表示卷积核，$g(x) = \max(0, x)$ 是一个非线性函数，$capsnet()$ 表示一个胶囊网络。

当然，还有利用其他神经网络来解决知识图谱嵌入问题的模型，例如 R-GCN [51] 和 KGAT [52] 则利用了图神经网络。

2.1.1.4 基于旋转的 KGE 方法

基于旋转的 KGE 方法将 r 看作是 h 到 t 的旋转操作。2019 年，Sun 等人 [53] 提出了 RotatE 方法，该方法在复数空间内进行建模，并将评分函数定义为：

$$f_r(h,t) = \|h \circ r - t\|. \tag{2.17}$$

其中 $\{h,r,t\} = e^{i\theta} = \cos\theta + i\sin\theta$，$(\circ)$ 表示哈达玛积 (Hadamard product)。通过理论证明，RotatE 可以有效处理组合关系、翻转和对称/反对称。

RotatE在二维复平面空间中构建模型，Zhang 等人 [54]进一步扩展到三维复平面空间，从而提出 QuatE 模型。该模型利用四元数和欧拉角等方法进行建模，其中四元数可表示为 $Q = a + bi + cj + dk$，且评分函数被定义为：

$$f_r(h,t) = h \otimes r^{\triangleleft} \bullet t. \tag{2.18}$$

其中 h、r、t 均为四元数，$r^{\triangleleft} = \frac{r}{\|r\|_2}$，$\otimes$ 表示哈密尔顿乘积 (Hamilton product)，$\bullet$ 表示内积。

2.1.2 融合附加信息的 KGE 方法

融入附加信息的知识图谱嵌入方法还利用其他有效信息来进一步改进评分函数提高模型的语义表达能力。第一种可以融入的额外信息是实体的语义类别，大部分 KGs 中的该类信息都是有价值的。例如“陈凯歌”的类别是人物，“霸王别姬”的类别是电影作品。2015 年，Guo 等人 [55]提出了语义平滑嵌入 (SSE) 模型，它要求属于同一语义类别的实体将在嵌入空间中彼此靠近，例如，“霸王别姬”应与“十面埋伏”更靠近，而不是靠近“张艺谋”。SSE 使用了两种不同的流形学习算法对对光滑度假设进行建模。第一种是利用了拉普拉斯特征映射，目的是让每个实体靠近其相同类别的其他实体，其平滑度量被定义为：

$$\mathcal{R}_1 = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{e}_i - \mathbf{e}_j\|_2^2 w_{ij}^{(1)}. \tag{2.19}$$

其中 $\mathbf{e}_i$ 和 $\mathbf{e}_j$ 分别是实体 e_i 和 e_j 的向量表示，$w_{ij}^{(1)} = \begin{cases} 1, if c_{e_i} = c_{e_j}, \\ 0, otherwise, \end{cases}$，$c_{e_i}/c_{e_j}$ 表示实体 e_i/e_j 的类别标签。通过最小化 $\mathcal{R}_1$，当两个实体 e_i 和 e_j 属于同一范畴时，期望 $\mathbf{e}_i$ 和 $\mathbf{e}_j$ 之间距离极小。

第二种是局部线性嵌入，其思想是利用相同类别里的实体的线性组合来表示每个实体，其平滑度量被定义为：

$$\mathcal{R}_2 = \sum_{i=1}^{n}\left\|\mathbf{e}_i - \sum_{\mathbf{e}_j\in\mathcal{N}(e_i)} w_{ij}^{(2)}\mathbf{e}_j\right\|_2^2. \tag{2.20}$$

其中 $w_{ij}^{(2)} = \begin{cases} 1, if e_j \in \mathcal{N}(e_i), \\ 0, otherwise, \end{cases}$ 且 $\sum_{j=1}^{n} w_{ij}^{(2)} = 1$，$\mathcal{N}(e_i)$ 表示 e_i 的最近邻集，即从 e_i 所属的类别中均匀地随机抽取 K 个实体构成的集合。通过最小化 R_2，每个实体都可以从其最近的邻居以低误差进行线性重建。最后将 R_1 和 R_2 作为正则化项合并到目标函数中来获得两个 SSE 模型。

SSE 的主要考虑每个实体仅属于一个类别，但实际上并不一定。因此 Xie 等人 [56]进一步提出 TKRL 模型，它建议实体应该有不同类型的多种表示，充分利用分层实体类型。具体地，它将层次类型视为实体的投影矩阵，用两种类型的编码对分层结构进行建模。同时，类型信息也被用作关系特定的类型约束。

第二种是添加额外的关系路径信息。关系路径可以被定义为一个序列 $(r_1, r_2, ..., r_l)$，较远的两个实体可以利用该序列进行连接。如图 2.7 所示，序列 (BornIn，LocatedIn) 可以将实体“AlfredHitchcock”和实体“England”连接起来，作为这两个实体的一个关系序列。关系路径蕴含着丰富的线索和信息，有利于知识图谱的自动化补全，例如 (BornIn，LocatedIn) 可以推断出 AlfredHitchcock 与 England 之间存在国籍 (Nationality) 关系，所以可以使用关系路径来推测出两个实体之间存在的关系。因此常利用该信息来

进一步增强模型的推理能力 [57–59]。

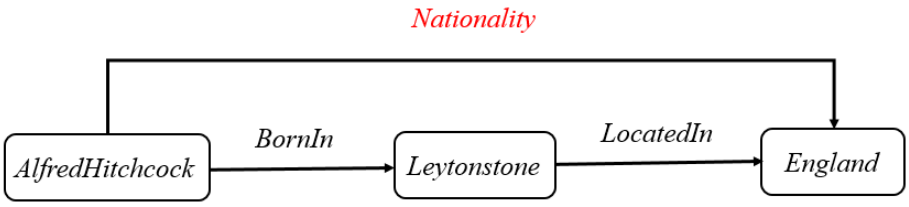

图2.7: 关联路径示例

第三种是附加文本描述的信息。在部分知识图谱中，通常会简单描述部分语义信息丰富的实体。如图 2.8 所示，该图展示了在知识图谱 FreeBase 中，实体“ AlfredHitchcock”和“Psycho”的描述信息。除此之外，还可以通过维基百科文章或新闻等来获取更多的有效的文本信息。

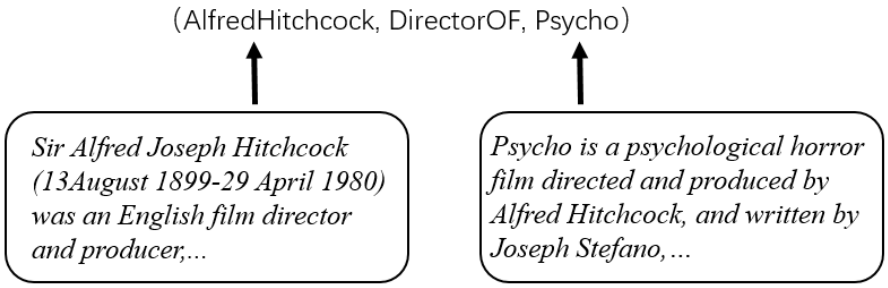

图2.8: 实体描述示例

Socher 等人 [60]提出 NTN 模型，该模型首先利用大型语料库来训练单词的嵌入向量，然后把实体表示为它们构成的词向量的平均值。Xie 等人 [61]提出 DKRL 模型，该模型探索了两个编码器，包括连续词袋和深度卷积神经模型来编码实体描述的语义，从而更好地处理实体的描述。Wang 等人 [62]利用文本语料库中丰富的上下文信息，提出了一种新颖的 KGE 方法——TEKE，极大地扩展了知识图谱的语义结构。

第四种是考虑融入逻辑规则，例如由关系 HasWife 连接的两个实体也可以使用关系 HasSpouse 连接。这些逻辑规则包含丰富的背景信息，因此学者们也利用该额外信息来改善知识图谱补全 [63]，提高模型的知识获取和推理。

2.2 长尾数据和非长尾数据

在部分知识图谱中，存在较多的冗余关系，例如反向关系、近似冗余关系和笛卡尔积

关系等，定义分别如下：

反向关系 在 KGs 中的三元组对 (h_1, r_1, t_1) 和 (h_2, r_2, t_2) 中，若 $h_2 = t_1$，$t_2 = h_1$，则关系 r_1 与 r_2 称为反向关系，r_2 可记为 ${r_1}^{-1}$，称为 r_1 的逆。

近似冗余关系 记 $|r|$ 为 KGs 中关系是 r 的三元组的数量，P_r 为关系是 r 的头-尾实体对集合，即 $P_r = \{(h,t)\,|(h,r,t) \in \mathcal{G}\}$。假设 θ_1 和 θ_2 是设置的阈值，如果 $\frac{|P_{r_m} \cap P_{r_n}|}{|r_m|} > \theta_1$ 且 $\frac{|P_{r_m} \cap P_{r_n}|}{|r_n|} > \theta_2$，则称关系 r_m 和 r_n 是近似冗余关系。记 R_{near} 为这种近似冗余关系的集合，即 $R_{near} = \{r_m, r_n | \frac{|P_{r_m} \cap P_{r_n}|}{|r_m|} > \theta_1, \frac{|P_{r_m} \cap P_{r_n}|}{|r_n|} > \theta_2\}$。

笛卡尔积关系 记 H_r 和 T_r 分别为 KGs 中关系是 r 的三元组的头实体集合和尾实体集合，即 $H_r = \{h|\exists(h,r,t) \in \mathcal{G}\}$，$T_r = \{t|\exists(h,r,t) \in \mathcal{G}\}$。设 θ 是设置的阈值，如果 $\frac{|r|}{|H_r| \times |T_r|} > \theta$，则称关系 r 是笛卡尔积关系。记 R_{Car} 为笛卡尔积关系的集合，即 $R_{\text{Car}} = \{r | \frac{|r|}{|H_r| \times |T_r|} > \theta\}$。

非长尾数据 若三元组中的关系是反向关系、近似冗余关系或笛卡儿积关系等，则该三元组数据称为非长尾数据。记 S_{non} 为非长尾数据构成的集合，称为非长尾数据集，即 $S_{non} = S_{reverse} \cup S_{near} \cup S_{\text{Car}}$，其中 $S_{reverse} = \{(h,r,t)|\exists(t, r^{-1}, h) \in \mathcal{G}\}$、$S_{near} = \{(h,r,t)|r \in R_{near}\}$ 和 $S_{\text{Car}} = \{(h,r,t)|r \in R_{\text{Car}}\}$。

长尾数据 若三元组数据不是非长尾数据，称为长尾数据。记 S_{long} 为长尾数据构成的集合，称为长尾数据集，即 $S_{long} = \{(h,r,t)|(h,r,t) \notin S_{non}\}$。

非长尾数据之间的相关性较大、语义重复度较高且相同实体和关系出现频率较高；而长尾数据之间的相关性较小、语义重复度较低且相同实体和关系出现频率较低。当前大部分知识图谱嵌入方法对非长尾数据的补全效果较好，而对长尾数据的补全效果较差，因此本文重点研究提高长尾数据补全效果的算法。

2.3 期望最大化算法

期望最大算法 (Expectation Maximization Algorithm，简称 EM 算法) 是一种迭代算法，该算法主要解决包含隐变量的概率参数模型的最大似然估计问题。

首先，最大似然估计的目标是根据样本推断出模型最有可能的分布参数。对于样本集 $X = \{x_1, x_2, ..., x_N\}$，为解出该模型的参数值 θ，需要极大化以下这个对数似然函数：

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{N} \log P(x_i; \theta) \tag{2.21}$$

但是，在一些实际情况中，给出的数据中可能包含某些无法观测的隐变量 $Z = \{z_1, z_2, ..., z_N\}$，即上文中每个样本属于哪个分布是未知的，因此现需极大化的对数似然函数有所改变：

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{N} \log P(x_i; \theta) = \arg\max_{\theta} \sum_{i=1}^{N} \log \sum_{z_i} P(x_i, z_i; \theta) \tag{2.22}$$

在 (2.22) 式中，θ 的值无法直接计算得到，因此对该式先进行缩放：

$$\begin{aligned}\sum_{i=1}^{N}\log\sum_{z_i}P(x_i,z_i;\theta)&=\sum_{i=1}^{N}\log\sum_{z_i}Q_i(z_i)\frac{P(x_i,z_i;\theta)}{Q_i(z_i)}\\&\geq\sum_{i=1}^{N}\sum_{z_i}Q_i(z_i)\log\frac{P(x_i,z_i;\theta)}{Q_i(z_i)}\end{aligned}\tag{2.23}$$

在 (2.23) 式中引入一个新的分布 $Q_i(z_i)$，然后利用 Jensen 不等式进行缩放：

$$\log\sum_i\lambda_i y_i\geq\sum_i\lambda_i\log y_i,\lambda_i\geq 0,\sum_i\lambda_i=1\tag{2.24}$$

当给定 θ 时，对数似然函数 $\sum\limits_{i=1}^{N}\log P(x_i;\theta)$ 的值与 $Q_i(z_i)$ 和 $P(x_i,z_i)$ 有关，因此可以通过不断调整 $Q_i(z_i)$，$P(x_i,z_i;\theta)$ 这两个概率使下界不断上升，达到逼近对数似然函数真实值的目的。当不等式取等号时，调整后的概率等价于对数似然函数。因此根据 Jensen 不等式，不等式取等号时有：

$$\frac{P(x_i,z_i;\theta)}{Q_i(z_i)}=c.\tag{2.25}$$

其中 c 为常数。

由于 $Q_i(z_i)$ 是一个分布，则满足：

$$\sum_i Q_i(z_i)=1.\tag{2.26}$$

根据 (2.25) 式和 (2.26) 式，整理得到：

$$Q_i(z_i)=\frac{p(x_i,z_i;\theta)}{\sum\limits_{z_i}p(x_i,z_i;\theta)}=\frac{p(x_i,z_i;\theta)}{p(x_i;\theta)}=p(z_i|x_i;\theta).\tag{2.27}$$

上式推出在固定其他参数 θ 后，$Q_i(z_i)$ 的计算公式就是后验概率，建立了对数似然函数的下界，此步骤就是 EM 算法中的 E 步。接下来的 M 步中，给定 $Q_i(z_i)$ 后，调整 θ，去极大化对数似然函数的下界。

EM 算法步骤如下：

Algorithm 1 EM算法

1: 初始化模型参数 θ；
2: (E步)：根据参数 θ 初始值或上一次迭代所得参数值 θ 来计算隐变量的后验概率，作为隐变量的现估计值：

$$Q_i(z_i):=p(z_i|x_i;\theta)$$

3: (M步)：通过最大化似然函数得到新的参数值θ：

$$\theta:=\arg\max_{\theta}\sum_{i=1}^{N}\sum_{z_i}Q_i(z_i)\log\frac{P(x_i,z_i;\theta)}{Q_i(z_i)}$$

4: 重复步 2(E步)、步 3(M步)直至收敛。

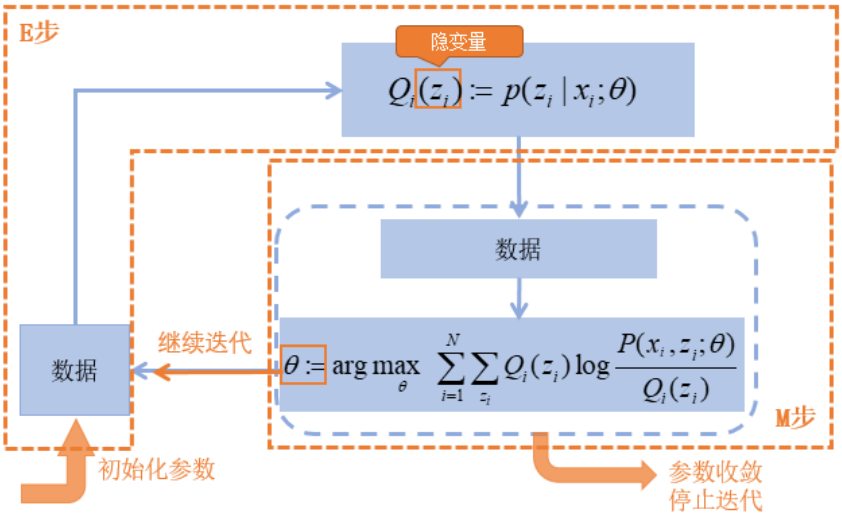

图2.9: EM算法概述

EM 算法是一种解决数据缺失情况下的参数估计算法，已知的是观察数据，未知的是隐变量和模型参数，其基本思想就是：在 E 步固定模型参数的值，计算隐变量的分布；在 M 步固定隐变量的分布，优化模型的参数。由于 EM 算法思想简单，其发展迅速，这种思想也被用于处理更加广泛的问题，例如高斯混合模型，用于数据聚类的 k-means 算法，坐标轴下降法等都包含了类似的思想。本文借助该思想，将非长尾数据中不可观测的语义信息作为隐变量，通过相似的交替更新将非长尾数据的知识转移到长尾数据中，以此提高长尾数据的补全性能。

3 融入期望最大化算法思想的 KGE 方法 (EM-KGE)

对于长尾数据，当前的大部分 KGE 方法对其的补全效果都不太好且推理的精度较差，而对于非长尾数据，当前的大部分 KGE 方法对其的补全效果都较好且推理的精度也较好，本章主要针对当前方法存在的该问题，设计了一种融入期望最大化算法思想的 KGE 方法 (EM-KGE)。以下将从 EM-KGE 方法介绍及实验评估两个方面进行详细介绍。

3.1 融入期望最大化算法思想的 KGE 方法 (EM-KGE) 介绍

Google Brain [66]团队通过直接针对一个有监督任务，利用元学习来更新无监督的学习规则将有利于无监督学习，这说明监督信息有助于无监督学习。受该工作的启发，本章将采用类似的思想将当前的 KGE 方法加以改进：由于非长尾数据蕴含充足的语义信息，因此将非长尾数据当作具有监督信息的数据，将长尾数据当作无监督信息的数据，利用非长尾数据里的语义信息来增强长尾数据的语义表达，帮助长尾数据补全效果能够进一步得到提升。为实现这一目标，本文提出在原有 KGE 方法基础上，融合期望最大化算法的框架模式，借助该框架中的隐变量搭建非长尾数据和长尾数据间的桥梁。该框架把非长尾数据中蕴含的监督信息看作无法观测的隐变量，采取与 EM 算法框架相似的异步更新方式，实现监督信息从非长尾数据到长尾数据的迁移，继而达到长尾数据的补全性能进一步提高的目的。为确定隐变量，下面先给出冗余实体和非冗余实体的定义：

冗余实体 若实体属于非长尾数据，则该实体称为冗余实体。记 $\mathcal{E}_{redun}$ 为冗余实体构成的集合，称为冗余实体集，即 $\mathcal{E}_{redun} = \{h, t \,|h, t \in S_{non}\}$。

非冗余实体 若实体不是冗余实体，称为非冗余实体。记 $\mathcal{E}_{non}$ 为非冗余实体构成的集合，称为非冗余实体集，即 $\mathcal{E}_{non} = \mathcal{E} - \mathcal{E}_{redun}$。

最终本章提出一种融入 EM 算法思想的 KGE 方法 (EM-KGE)，如图 3.1 所示，该方法将利用非长尾数据中的冗余实体来存储监督知识，因此把冗余实体的嵌入向量当作隐变量，等价于 EM 算法中的 Z，把长尾数据中的其余非冗余实体嵌入作为模型参数，等价于 EM 算法中的 θ，算法更新过程类似于 EM 算法的 E 步和 M 步，并进行异步交错更新。而关系嵌入不融入 EM 算法框架，既不看作隐变量，也不看作模型参数，它将在 E 步和 M 步都进行更新。

E步 此步的目的是期望能够获得非长尾数据中包含的监督信息，所以在该步只关注非长尾数据。首先，EM-KGE 方法直接把非长尾数据中的冗余实体嵌入参数当作隐变量，然后计算非长尾数据部分的目标函数，再将此时的非冗余实体嵌入参数固定不变，利用梯度优化算法更新冗余实体嵌入参数和关系嵌入参数。

M步 此步的目的是希望把非长尾数据中包含的监督信息通过隐变量迁移到长尾数据中，所以在该步会固定隐变量的参数，继而指导其余参数的更新。首先，EM-KGE 方

法会计算长尾数据部分的目标函数，再将此时的隐变量，即冗余实体嵌入参数固定不变，利用梯度优化算法更新非冗余实体嵌入参数和关系嵌入参数。

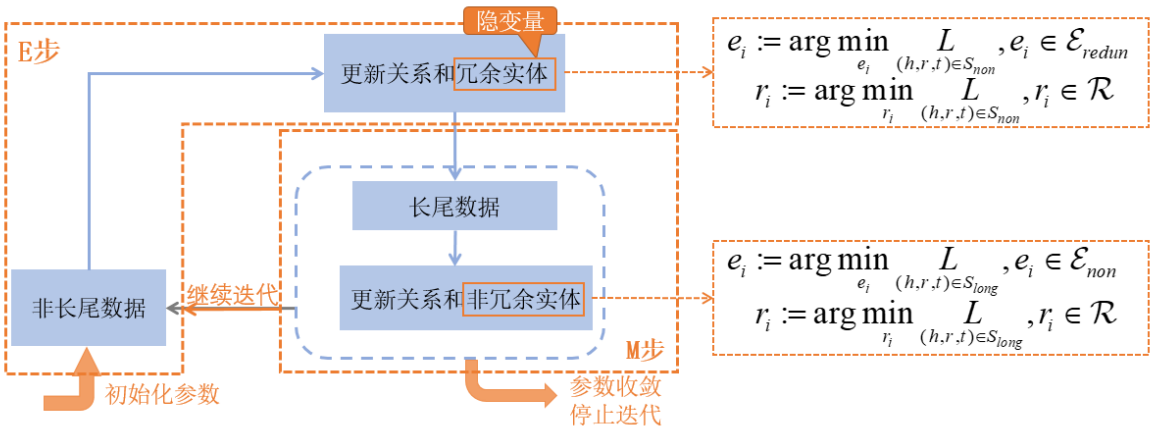

图3.1: EM-KGE方法

下面对目标函数及具体的算法步骤进行详细介绍。

3.1.1　目标函数

对于每个三元组，本文采用以下函数作为目标函数来训练实体和关系的向量表示，该目标函数也是基于翻译的 KGE 方法常采用的目标函数:

$$L = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} [\gamma + f_r(h,t) - f_r(h',t')]_+. \tag{3.1}$$

其中 $[x]_+$ 表示 $\max(0,x)$，$\gamma > 0$ 是一个超参数。$f_r(h,t)$ 表示评分函数，用于衡量该三元组的置信度，该评分函数在不同的方法中有不同的设计。例如，TransE 将评分函数定义为 $f_r(h,t) = \|h + r - t\|_{L_n}$。$S'$ 表示负采样集合，具体表示如下：

$$S' = \{(h',r,t)|h' \in \mathcal{E}\} \cup \{(h,r,t')|t' \in \mathcal{E}\}. \tag{3.2}$$

训练 EM-KGE 的目标就是最小化上述目标函数并迭代更新实体和关系的嵌入向量。

3.1.2　算法步骤

首先初始化冗余实体 $e_i \in \mathcal{E}_{redun}$、非冗余实体 $e_i \in \mathcal{E}_{non}$ 和关系 $r_i \in \mathcal{R}$ 的向量表示参数。

3.1.2.1　EM-KGE 方法中的 E 步更新

首先，根据 (3.1) 式计算非长尾数据部分的目标函数 $L_{(h,r,t)\in S_{non}}$，为了最小化该目标函数，利用梯度优化算法更新隐变量的参数，即冗余实体 $e_i \in \mathcal{E}_{redun}$ 的向量参数和关系

$r_i \in \mathcal{R}$ 的向量参数:

$$e_i := \arg\min_{e_i} \underset{(h,r,t)\in S_{non}}{L}, e_i \in \mathcal{E}_{redun}, i = 1, ..., |\mathcal{E}_{redun}|. \tag{3.3}$$

$$r_i := \arg\min_{r_i} \underset{(h,r,t)\in S_{non}}{L}, i = 1, ..., |\mathcal{R}|. \tag{3.4}$$

该过程对应算法 2 中的步 2 至步 5 。

3.1.2.2 EM-KGE 方法中的 M 步更新

首先，重新使用在 E 步中更新得到的冗余实体和关系的向量表示，根据 (3.1) 式再次计算长尾数据部分的目标函数 $\underset{(h,r,t)\in S_{\text{long}}}{L}$。为了将最小化该目标函数，利用梯度优化算法继续更新非冗余实体和关系的嵌入参数，即 $e_i \in \mathcal{E}_{non}$ 和 $r_i \in \mathcal{R}$ 的参数。

$$e_i := \arg\min_{e_i} \underset{(h,r,t)\in S_{long}}{L}, e_i \in \mathcal{E}_{non}, i = 1, ..., |\mathcal{E}_{non}|. \tag{3.5}$$

$$r_i := \arg\min_{r_i} \underset{(h,r,t)\in S_{long}}{L}, i = 1, ..., |\mathcal{R}|. \tag{3.6}$$

该过程对应算法 2 中的步 6 至步 9。

EM-KGE 的算法设计如下：

Algorithm 2 EM-KGE算法步骤

输入: 非长尾训练集S_{non};
长尾训练集S_{long};
冗余实体集$\mathcal{E}_{redun}$;
非冗余实体集$\mathcal{E}_{non}$;
关系集$\mathcal{R}$;
批次数量T;
迭代次数N;
负采样样本的数量neg。
输出: 冗余实体表示$e_i \in \mathcal{E}_{redun}$;
非冗余实体表示$e_i \in \mathcal{E}_{non}$;
关系表示$r_i \in \mathcal{R}$.
1: 初始化$e_i \in \mathcal{E}_{redun}$，$e_i \in \mathcal{E}_{non}$，$r_i \in \mathcal{R}$，$n = 1$。
2: 令$t = 1$。
3: 采集小批次非长尾训练集并根据负采样样本的数量neg进行负采样得到S_{non,b_1}。
4: 根据(3.1)式计算非长尾数据部分的目标函数 $\underset{(h,r,t)\in S_{non,b_1}}{L}$，通过梯度下降法更新$e_i \in \mathcal{E}_{redun}$，$r_i \in \mathcal{R}$。
5: 若$t > T$，则转步6；否则，令$t := t + 1$转步3。
6: 令$t = 1$。
7: 采集小批次长尾训练集并根据负采样样本数量neg进行负采样得到S_{b_2}。
8: 根据(3.1)式计算长尾数据部分的目标函数 $\underset{(h,r,t)\in S_{long,b_2}}{L}$，固定参数$e_i \in \mathcal{E}_{redun}$，通过梯度下降法更新$e_i \in \mathcal{E}_{non}$，$r_i \in \mathcal{R}$。
9: 若$t > T$，则转步10；否则，令$t := t + 1$转步7。
10: 若$n > N$，则算法终止；否则，令$n := n + 1$转步2。
11: **返回** 冗余实体表示$e_i \in \mathcal{E}_{redun}$；非冗余实体表示$e_i \in \mathcal{E}_{non}$；关系表示$r_i \in \mathcal{R}$。

3.2　实验与评估

3.2.1　数据集

为了验证本章提出的 EM-KGE 方法的有效性，本章选择了基准数据集 FB15K，它是从 Freebase 中提取的一个数据集，Freebase 提供了世界的一般事实，例如，三元组 (Steve Jobs, founded, Apple Inc.) 在人名实体 Steve Jobs 和组织实体 Apple Inc. 之间建立了 founded 关系。数据集 FB15K 中涵盖了 14951 个实体和 1345 个关系，包括 483142 条训练集、59071 条测试集和 50000 条验证集，具体的统计信息如表 3.1 所示。同时，FB15K 数据集中有大量本文提到的冗余关系，例如笛卡尔积关系、反向关系和近似冗余关系。

表 3.1: FB15K的统计数据

数据集	实体	关系	训练集	验证集	测试集
FB15K	14951	1345	483142	50000	59071

3.2.2　链接预测

链接预测是用于评估知识图谱嵌入方法性能的一个常见任务，它是指当三元组中的其中一个实体和关系已知时，预测出未知的实体。具体地，对于三元组 $(?, r, t)$，需预测出未知的头实体 h，对于三元组 $(h, r, ?)$，需预测出未知的尾实体 t。例如，预测三元组 $(?, founded, AppleInc.)$ 中缺失的实体，就是预测谁创立了苹果公司，预测三元组 $(SteveJobs, founded, ?)$ 中缺失的实体，就是预测 Steve Jobs 创立了哪个公司？当模型训练完成后，就可以使用测试集进行相应的评估。首先，取出测试集中的一个三元组 (h, r, t)，该三元组 (h, r, t) 是正确三元组。然后对于该三元组 (h, r, t)，使用实体集 $\mathcal{E} - \{h\}$ 中的每个实体对头实体 h 或尾实体 t 进行逐一替换，从而构造新的错误三元组，这样的错误三元组有 $|\mathcal{E}| - 1$ 个。再利用评分函数计算该正确三元组以及 $|\mathcal{E}| - 1$ 个错误三元组的分数，将分数按升序进行排列。若是进行头替换，则将其正确三元组的排序记为 $rank_h$，若是进行尾替换，则将其正确三元组的排序记为 $rank_t$，该排序越小越好。

本章将使用以下评估指标：MR、MRR 和 Hit@N。MR 表示该测试集中所有正确三元组的排名平均值，MRR 表示该测试中所有正确三元组的排名的倒数平均值，Hit@N 表示该测试集中正确三元组的排序在前N的百分比。由此可见，MR的值越小代表方法越有效，MRR 和 Hit@N 的值越大代表方法越有效。计算公式分别如下：

$$MR = \frac{1}{2|S_{test}|} \sum_{(h,r,t) \in S_{test}} (rank_h + rank_t). \tag{3.7}$$

$$MRR = \frac{1}{2|S_{test}|} \sum_{(h,r,t) \in S_{test}} (\frac{1}{rank_h} + \frac{1}{rank_t}). \tag{3.8}$$

$$Hit@N = \frac{1}{2|S_{test}|} \sum_{(h,r,t)\in S_{test}} I(rank_h \leq N) + I(rank_t \leq N). \tag{3.9}$$

$$I(x) = \left\{ \begin{array}{l} 1,\ if\ x\ is\ Ture \\ 0,\ if\ x\ is\ False \end{array} \right\}. \tag{3.10}$$

其中 $|S_{test}|$ 表示测试集中三元组的数量。

但是，在错误三元组构造的过程中，可能构造的三元组在训练集、测试集或验证集中已经存在，那么该三元组不能被认定为错误三元组，并且该三元组的分数可能会低于测试的正确三元组的分数，导致正确三元组的预测排序增大，从而影响评估结果，因此在测试时可进行过滤设置，将构造的这类三元组过滤掉，不参与排序。将过滤后得到的相应指标记为 FMR、FMRR 和 FHit@N。

3.2.3　实验配置

本节选择 TransE、TransH 和 TransD 作为实验的原始方法，并在此基础上用 EM-KGE 方法进行改进。实验环境为 64GB 内存和 TiTAN XP GPU 的 Intel Xeon(R) Silver 4114 CPU 的个人工作站，使用 Python 中的 Pytorch 实现代码，并可以在以下网址查看相关的数据和源代码：`https://github.com/HMH-123/KGE-Code.git`。

为更好地训练模型，本文设置了以下具体的参数范围。

表 3.2: 实验参数配置表

参数名	参数值
α_1	{0.1,0.2,0.3}
α_2	{0.1,0.2,0.3}
T	{100,200,300}
γ	{3,4,5}
neg	{10,15,20,25}
$optimizer$	{“SGD”,“Adam”}
d	50
N	50

以上的 α_1 是改进方法 EM-KGE 方法的学习率，α_2 是原始方法的学习率，T 是批次数量，neg 是负采样数量，d 是嵌入向量的维度，N 是迭代次数，$optimizer$ 是 pytoch 中的优化器。

3.2.4　实验结果

3.2.4.1　数据分析

根据前面对冗余关系的定义，首先简单分析一下 FB15K 训练集中关系是反向关系、笛卡尔积关系和近似冗余关系得三元组得数量，如表 3.3 所示。从该表中可以看出 FB15K

训练集中关系是冗余关系的三元组的数量较大，其中最多的是反向关系，有 429543 条数据，占比为 88.9%，可见该数据中确实存在大量的冗余关系。

表 3.3: FB15K训练集中每个冗余关系的三元组数量

冗余关系	reverse relation	Cartesian product relation	nearduplicate relation
数量	429543	983	29288
百分比	0.889	0.002	0.060

3.2.4.2　实验结果分析

本章选择了经典的基于翻译的 KGE 方法——TransE、TransH、TransD，作为基准线，然后利用 EM-KGE 方法对原始方法进行改进，改进后的方法相应地记为 EM-X，例如利用 EM-KGE 方法改进 TransE 方法后记为 EM-TransE，并将实验结果分为三组。粗体的结果表示组中的最佳结果，下划线的结果表示列中的最佳结果。

表 3.4: FB15K测试数据的链路预测结果

	MR	FMR	MRR	FMRR	Hit@10	FHit@10
TransE	**237.27**	**143.65**	0.226	0.343	0.438	0.561
EM-TransE	247.88	154.90	**0.229**	**0.344**	**0.442**	0.561
TransH	**232.92**	**139.22**	0.226	**0.343**	0.439	**0.565**
EM-TransH	250.15	157.12	**0.228**	0.342	**0.442**	0.562
TransD	**234.10**	**140.66**	0.227	0.342	0.438	0.564
EM-TransD	247.24	153.13	**0.228**	0.342	**0.441**	0.564

从表 3.4 中可以发现：(1) 从指标 Hit@10 得结果可以看出，大多数改进方法的结果优于原始方法或达到同等效果，表明 EM-KGE 方法能够进一步提高预测的准确性；(2) 对照指标 MR 和 MRR，MR 的值没有提升，MRR 的值大部分提升，表明 EM-KGE 方法提高了前面的大部分排序，但也导致后面的排序大幅度降低。由此可见，EM-KGE 方法有一定的提升作用，能够提升预测准确性和前面的排序结果，但其作用有限，也会造成部分后面的排序结果有所劣化。

4 融入 EM 算法思想的双重嵌入方法 (DEM)

针对 EM-KGE 方法仅依靠冗余实体来传递非长尾数据中包含的语义信息的效果具有一定局限性的问题，本章对该方法进一步改进，寻找更加合适的隐变量来实现信息传递，设计了一种融入期望最大化算法思想的双重嵌入方法 (DEM)。以下将从 DEM 方法介绍及实验评估两个方面进行详细介绍。

4.1 融入 EM 算法思想的双重嵌入方法(DEM)介绍

Zhang 等人 [67]提出了潜在语义单元，并将它作为实体和关系嵌入的子组件，从而利用该潜在语义单元实现知识在实体或关系之间转移。本章借助该思想，在融入期望最大化算法思想的 KGE 方法基础上，提出了双重嵌入技术，包括原实体、原关系的嵌入和潜在语义嵌入，利用原实体、原关系的嵌入存储原本的语义知识，将潜在语义嵌入作为隐变量，保存非长尾数据中的监督知识，实现该知识向长尾数据的迁移。最终提出一种融入期望最大化算法思想的双重嵌入方法 (DEM)，如图 4.1 所示，该方法将原实体、原关系的嵌入作为模型参数，即 EM 算法中的参数 θ，将潜在语义嵌入作为无法观测的隐变量，即 EM 算法中的 Z，并分为 E 步和 M 步进行交替更新。

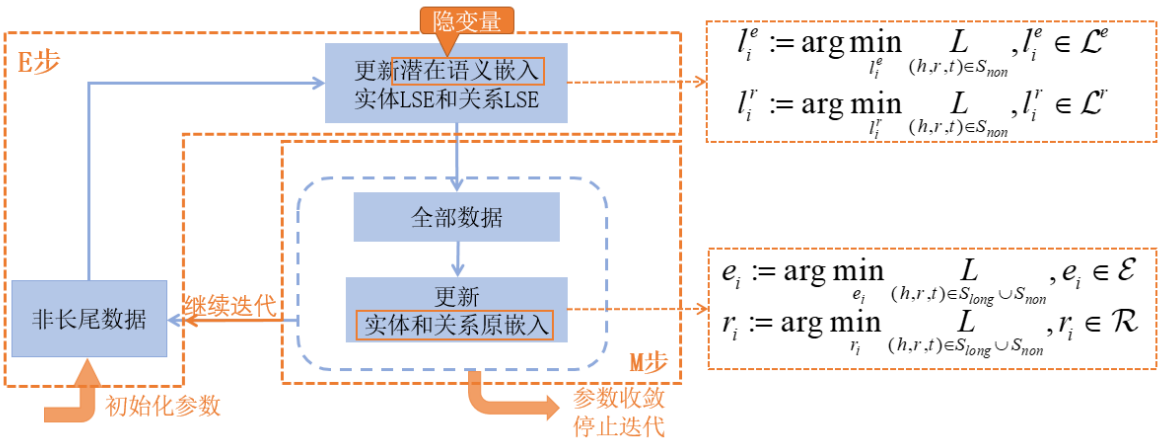

图4.1: 融入EM算法思想的双重嵌入方法

E步 此步的目的是期望能够获得非长尾数据中包含的监督信息，所以在该步只关注非长尾数据。首先，DEM 方法直接把潜在语义嵌入参数当作隐变量，然后计算非长尾数据部分的目标函数，再将此时的原实体、原关系嵌入参数固定不变，利用梯度优化算法更新潜在语义嵌入参数。

M步 此步的目的是希望把非长尾数据中包含的监督信息通过隐变量迁移到长尾数

据中，所以在该步会固定隐变量的参数，继而指导其余参数的更新。首先，DEM 方法会计算所有数据的目标函数，再将此时的隐变量，即潜在语义嵌入参数固定不变，利用梯度优化算法更新原实体、原关系嵌入参数。

下面将详细介绍双重嵌入、目标函数及具体的算法步骤。

4.1.1　双重嵌入

本章提出新的双重嵌入技术，第一重是原实体、原关系的嵌入，第二重是潜在语义嵌入 (Latent semantic embedding，简称 LSE)，其中潜在语义嵌入包括实体潜在语义嵌入和关系潜在语义嵌入。潜在语义嵌入由其子组件 LSEs 根据相应的权重计算得到，该权重由子组件 LSEs 分别与实体嵌入或关系嵌入之间的相似度所决定，那么相似的实体或关系嵌入能够共享相同的 LSEs，达到语义共享的目的。因此长尾数据能够获得非长尾数据中的语义信息。图 4.2 呈现了一个实体的双重嵌入例子，该例子中设置了 6 个子组件 LSEs，展示了两个实体 e_1 和 e_2 的具体构成。实体 e_1 和 e_2 的最终嵌入表示由潜在语义嵌入和原嵌入两个部分构成，其中潜在语义嵌入部分由 6 个 LSEs 及其对应的权重求和得到。根据示例可知，e_1 和 e_2 的潜在语义嵌入部分共享第 2 和第 3 个 LSE，则 e_1 和 e_2 能够利用共享的 LSE 实现共享语义知识。

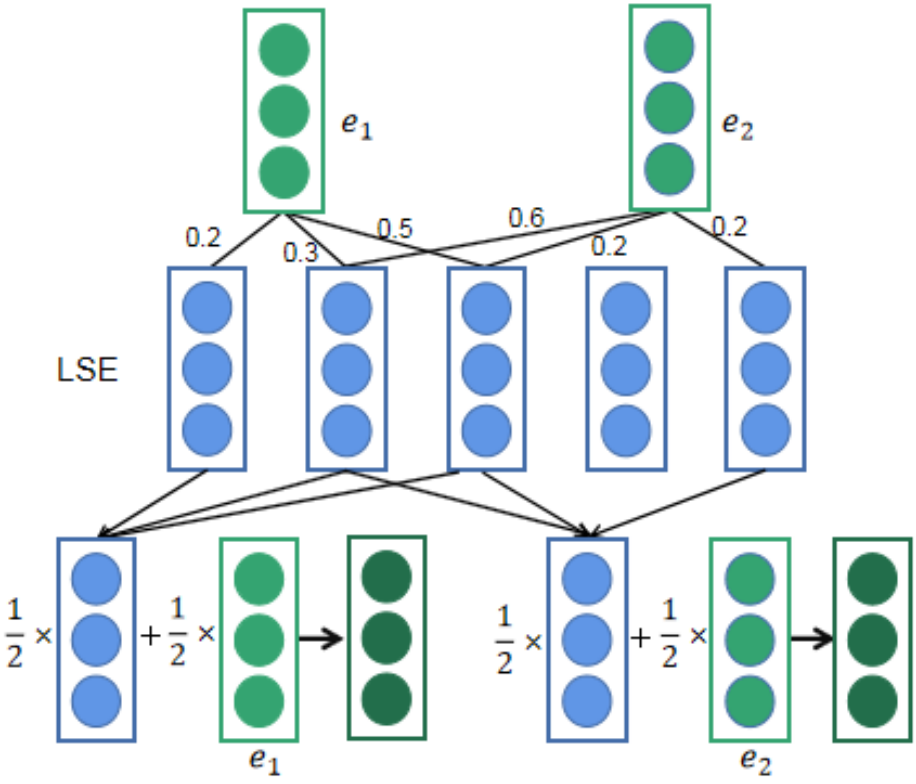

图4.2: 双重嵌入的一个例子

本章节提出的潜在语义嵌入 LSE，包括实体 LSEs $\mathcal{L}^e = \left\{l_1^e, l_2^e, l_3^e, ..., l_{|\mathcal{L}^e|}^e\right\}$ 以及关系 LSEs $\mathcal{L}^r = \left\{l_1^r, l_2^r, l_3^r, ..., l_{|\mathcal{L}^r|}^r\right\}$，其中 $l_i^e \in R^d$ 是 $\mathcal{L}^e$ 的第 i 个实体LSE，$l_i^r \in R^d$ 是 $\mathcal{L}^r$ 的第 i 个关系LSE，d 表示嵌入维度，$|\mathcal{L}^e|$ 是实体LSE的数量，$|\mathcal{L}^r|$ 是关系LSE的数量。接下来将对实体嵌入和关系嵌入的计算步骤作具体介绍。

在双重嵌入方法中，假设每个实体由原实体嵌入和实体潜在语义嵌入组成。实体 LSEs 的相应权重通过实体 LSE 与实体间的相似度所决定，具体地，第 i 个实体 e_i 的实体潜在语义部分的权重计算公式如下：

$$s_{i,j}^e = e_i \bullet l_j^e \tag{4.1}$$

$$\alpha_{i,j}^e = \frac{\exp(s_{i,j}^e)}{\sum_{p=1}^{|\mathcal{L}^e|} \exp(s_{i,p}^e)} \tag{4.2}$$

其中 $[\bullet]$ 表示点积，$s_{i,j}^e \in R$ 表示实体 e_i 与实体 LSEs$\mathcal{L}^e$ 中第 j 个实体 LSEl_j^e 之间的相似度，$\alpha_{i,j}^e$ 是权重向量 $\alpha_i^e \in R^{|\mathcal{L}^e|}$ 的第 j 个元素。

则最终的实体嵌入向量计算如下：

$$e_i := \frac{\sum_{j=1}^{|\mathcal{L}^e|} \alpha_{i,j}^e l_j^e}{2} + \frac{e_i}{2} \tag{4.3}$$

下面以相同地方式得到关系嵌入，具体地，第 i 个关系 r_i 的嵌入向量计算步骤如下：

$$s_{i,j}^r = r_i \bullet l_j^r \tag{4.4}$$

$$\alpha_{i,j}^r = \frac{\exp(s_{i,j}^r)}{\sum_{p=1}^{|\mathcal{L}^r|} \exp(s_{i,p}^r)} \tag{4.5}$$

$$r_i := \frac{\sum_{j=1}^{|\mathcal{L}^r|} \alpha_{i,j}^r l_j^r}{2} + \frac{r_i}{2} \tag{4.6}$$

4.1.2 目标函数

DEM 方法依旧采用以下函数作为目标函数来训练相关的向量表示:

$$L = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} [\gamma + f_r(h,t) - f_r(h',t')]_+. \tag{4.7}$$

其中 $[x]_+$ 表示 $\max(0,x)$，$\gamma > 0$ 是一个超参数。$f_r(h,t)$ 表示评分函数。S' 表示负采样集合，采样方法如下：

$$S' = \{(h',r,t)|h' \in \mathcal{E}\} \cup \{(h,r,t')|t' \in \mathcal{E}\}. \tag{4.8}$$

该方法的目标就是最小化上述函数并迭代更新原实体、原关系的嵌入向量和潜在语义嵌入向量。

4.1.3　算法步骤

首先，初始化实体潜在语义嵌入 $l_i^e \in \mathcal{L}^e$、关系潜在语义嵌入 $l_i^r \in \mathcal{L}^r$、原实体 $e_i \in \mathcal{E}$ 和原关系 $r_i \in \mathcal{R}$ 嵌入的参数。

4.1.3.1　DEM 方法中的 E 步更新

此步的目的是获得非长尾数据中的监督信息，因此将潜在语义嵌入参数当作隐变量，计算非长尾数据部分的目标函数，利用梯度优化算法仅更新隐变量——潜在语义嵌入参数，用以存储非长尾数据中准确的的监督信息。

首先，根据 4.1.1 节双重嵌入的 (4.1-4.3) 式计算实体集中每个实体 $e_i \in \mathcal{E}$ 的嵌入表示，根据 (4.4-4.6) 式计算关系集中每个关系 $r_i \in \mathcal{R}$ 的嵌入表示。

然后根据 (4.7) 式计算非长尾数据部分的目标函数 $\underset{(h,r,t)\in S_{non}}{L}$，为最小化该目标函数，利用梯度优化算法更新隐变量，即实体潜在语义嵌入 $l_i^e \in \mathcal{L}^e$ 和关系潜在语义嵌入 $l_i^r \in \mathcal{L}^r$ 的参数:

$$l_i^e := \arg\min_{l_i^e} \underset{(h,r,t)\in S_{non}}{L}, i = 1, ..., |\mathcal{L}^e|. \tag{4.9}$$

$$l_i^r := \arg\min_{l_i^r} \underset{(h,r,t)\in S_{non}}{L}, i = 1, ..., |\mathcal{L}^r|. \tag{4.10}$$

该过程对应算法 3 中的步 2 至步 5。

4.1.3.2　DEM 方法中的 M 步更新

此步的目的是将非长尾数据中的信息通过隐变量迁移到长尾数据中，因此在该步固定隐变量的参数，利用隐变量指导其余参数的更新。因此该步让潜在语义嵌入参数固定不变，利用梯度优化算法更新原实体、原关系嵌入参数，实现知识的迁移。

首先，利用在 E 步中得到的实体潜在语义嵌入 $l_i^e \in \mathcal{L}^e$ 和关系潜在语义嵌入 $l_i^r \in \mathcal{L}^r$，根据 4.1.1 节双重嵌入的 (4.1-4.3) 式重新计算实体集中每个实体 $e_i \in \mathcal{E}$ 的嵌入表示，根据 (4.4-4.6) 式重新计算关系集中每个关系 $r_i \in \mathcal{R}$ 的嵌入表示。

然后根据 (4.7) 式计算所有数据的目标函数 $\underset{(h,r,t)\in S_{non}\cup S_{long}}{L}$，为最小化该目标函数，利用梯度优化算法更新原实体、原关系嵌入的参数，即 $e_i \in \mathcal{E}$，$r_i \in \mathcal{R}$ 的参数。

$$e_i := \arg\min_{e_i} \underset{(h,r,t)\in S_{non}\cup S_{long}}{L}, i = 1, ..., |\mathcal{E}|. \tag{4.11}$$

$$r_i := \arg\min_{r_i} \underset{(h,r,t)\in S_{non}\cup S_{long}}{L}, i = 1, ..., |\mathcal{R}|. \tag{4.12}$$

该过程对应算法 3 中的步 6 至步 9。

DEM 的算法设计如下：

Algorithm 3 DEM 算法框架

输入: 非长尾训练集S_{non};
长尾训练集S_{long};
实体集$\mathcal{E}$;
关系集$\mathcal{R}$;
批次数量T;
迭代次数N;
负采样样本的数量neg。
输出: 实体表示$e_i \in \mathcal{E}$;
关系表示$r_i \in \mathcal{R}$;
实体潜在语义表示$\mathcal{L}^e$;
关系潜在语义表示$\mathcal{L}^r$。
1: 初始化 $e_i \in \mathcal{E}$，$r_i \in \mathcal{R}$，$\mathcal{L}^e$，$\mathcal{L}^r$，$n = 1$。
2: 令$t = 1$。
3: 采集小批次非长尾训练集并根据负采样样本的数量neg进行负采样得到S_{non,b_1}。
4: 根据(4.7)式计算非长尾数据的目标函数 $\underset{(h,r,t)\in S_{non,b_1}}{L}$，通过梯度下降法更新$l_i^e \in \mathcal{L}^e$，$l_i^r \in \mathcal{L}^r$。
5: 若$t > T$，则转步6；否则，令$t := t + 1$转步3。
6: 令$t = 1$。
7: 采集小批次非长尾数据和长尾训练集并根据负采样样本数量neg进行负采样得到S_{non,b_1}和S_{long,b_2}。
8: 根据(4.7)式计算所有数据的目标函数 $\underset{(h,r,t)\in S_{non,b_1}\cup S_{long,b_2}}{L}$，固定参数$\mathcal{L}^e$和$\mathcal{L}^r$，通过梯度下降法更新$e_i \in \mathcal{E}$，$r_i \in \mathcal{R}$。
9: 若$t > T$，则转步10；否则，令$t := t + 1$转步7。
10: 若$n > N$，则算法终止；否则，令$n := n + 1$转步2。
11: **返回** 实体表示$e_i \in \mathcal{E}$;关系表示$r_i \in \mathcal{R}$;实体潜在语义表示$\mathcal{L}^e$;关系潜在语义表示$\mathcal{L}^r$。

4.2 实验与评估

4.2.1 数据集

为了验证本章提出的 DEM 方法的有效性，本章选择了基准数据集 FB15K，它是从 Freebase 中提取的一个数据集，Freebase 提供了世界的一般事实，例如，三元组 (Steve Jobs, founded, Apple Inc.) 在人名实体 Steve Jobs 和组织实体 Apple Inc. 之间建立了 founded 关系。数据集 FB15K 中涵盖了 14951 个实体和 1345 个关系，包括 483142 条训练集、59071 条测试集和 50000 条验证集。同时，FB15K 数据集中有大量本文提到的冗余关系，例如笛卡尔积关系、反向关系和近似冗余关系。

4.2.2 链接预测

链接预测是用于评估知识图谱嵌入方法性能的一个常见任务，它是指当三元组中的其中一个实体和关系已知时，预测出未知的实体。具体地，对于三元组 $(?, r, t)$，需预测出未知的头实体 h，对于三元组 $(h, r, ?)$，需预测出未知的尾实体 t。首先，取出测试集中的一个三元组 (h, r, t)，该三元组 (h, r, t) 是正确三元组。然后对于该三元组 (h, r, t)，使用实体集 $\mathcal{E} - \{h\}$ 中的每个实体对头实体 h 或尾实体 t 进行逐一替换，从而构造新的错误三元组，这样的错误三元组有 $|\mathcal{E}| - 1$ 个。再利用评分函数计算该正确三元组以及 $|\mathcal{E}| - 1$

个错误三元组的分数，将分数按升序进行排列。若是进行头替换，则将其正确三元组的排序记为 $rank_h$，若是进行尾替换，则将其正确三元组的排序记为 $rank_t$，该排序越小越好。

本章将使用以下评估指标：MR、MRR 和 Hit@N。MR 表示该测试集中所有正确三元组的排名平均值，MRR 表示该测试中所有正确三元组的排名的倒数平均值，Hit@N 表示该测试集中正确三元组的排序在前 N 的百分比。由此可见，MR 的值越小代表方法越有效，MRR 和 Hit@N 的值越大代表方法越有效。

但是，在错误三元组构造的过程中，可能构造的三元组在训练集、测试集或验证集中已经存在，那么该三元组不能被认定为错误三元组，并且该三元组的分数可能会低于测试的正确三元组的分数，导致正确三元组的预测排序增大，从而影响评估结果，因此在测试时可进行过滤设置，将构造的这类三元组过滤掉，不参与排序。将过滤后得到的相应指标记为FMR、FMRR 和 FHit@N。

4.2.3 实验配置

本节选择 TransE、TransH 和 TransD 作为实验的原始方法，并在此基础上用 DEM 方法进行改进。实验环境为 64GB 内存和 TiTAN XP GPU 的 Intel Xeon(R) Silver 4114 CPU 的个人工作站，使用 Python 中的 Pytorch 实现代码，并可以在以下网址查看相关的数据和源代码：`https://github.com/HMH-123/KGE-Code.git`。

为更好地训练模型，本文设置以下参数范围：

表 4.1: 实验参数配置表

参数名	参数值
α_1	{0.0001，0.0005，0.0008}
α_2	{0.1,0.2,0.3}
T	{100,200,300}
γ	{3,4,5}
neg	{10,15,20,25}
$\|\mathcal{L}^e\|$	{50，100，200，300}
$\|\mathcal{L}^r\|$	{50，100，200，300}
$optimizer$	{“SGD”,“Adam”}
d	100
N	50

以上 α_1 是 DEM 方法的学习率，α_2 是原始方法的学习率，T 是批次数量，neg 是负采样数量，$|\mathcal{L}^e|$ 表示实体 LSEs 的数量，$|\mathcal{L}^r|$ 表示关系 LSEs 的数量，d 是嵌入向量的维度，N 是迭代次数，$optimizer$ 是 pytoch 中的优化器。

4.2.4 实验结果

本章选择了经典的基于翻译的 KGE 方法——TransE、TransH、TransD，作为基准线，然后利用 DEM 方法对原始方法进行改进，改进后的方法相应地记为 X-DEM，例如利用

DEM 方法改进 TransE 方法后记为 TransE-DEM，并将实验结果分为三组。粗体的结果表示组中的最佳结果，下划线的结果表示列中的最佳结果。

从表 4.2 中可以发现：(1) 在几组结果中，大多数改进方法的结果都优于原始方法，例如原始方法 TransH 与改进方法 TransH-DEM 相比，FMRR 提高了0.018，FHit@10 提高了 0.011，这表明改进方法推理的准确性大幅度提高；原始方法 TransE 与改进方法 TransE-DEM相比，MR 降低了 22.63，FMR 降低了 23.87，表明改进方法对正确三元组的预测排名整体上有所提升。(2) 改进方法在在 14 个指标上都具有优于原始方法的结果，占总指标的 78%，同时在 6 个指标上都是该列的最优值，占总指标的 100%，因此改进方法 DEM 能够提高知识图谱的补全性，提高推理的准确率。

表 4.2: FB15K测试数据的链路预测结果

	MR	FMR	MRR	FMRR	Hit@10	FHit@10
TransE	224.81	129.40	0.248	0.418	0.498	0.658
TransE-DEM	**202.18**	**105.53**	**0.256**	**0.421**	**0.500**	**0.666**
TransH	**221.45**	**126.64**	0.245	0.413	0.495	0.657
TransH-DEM	222.94	125.78	**0.258**	**0.431**	**0.504**	**0.668**
TransD	**221.82**	**126.62**	0.246	0.417	0.497	0.660
TransD-DEM	228.43	132.53	**0.257**	**0.427**	**0.501**	**0.663**

为了进一步比较原始方法和改进方法，将测试集划分为非长尾测试集和长尾测试集，并计算每个指标的值。表 4.3 展示了在非长尾测试集中的测试结果，表 4.4 展示了在长尾测试集中的测试结果，通过表 4.3 和表 4.4 可以发现：(1) 表 4.3 表明 DEM 方法对非长尾数据的补全性能有较好的提升。例如，在 TransE 上，改进方法 TransE-DEM 的 MR 和 FMR 分别降低了 22.61 和 22.11，说明改进方法对正确三元组的预测排名有较大的提高，指标 Hit@10 和 FHit@10 上分别增加了 0.002 和 0.005，说明改进方法预测的准确率进一步提高。(2) 表 4.4 表明改进的方法在长尾数据上也取得了更好的效果，例如，TransE 进行改进后，MR 和 FMR 分别降低了 22.78 和 41.23，比非长尾数据降低得更多，说明改进方法对长尾数据的正确三元组的总体预测排名有较大的提高，Hit@10 和 FHit@10 分别增加了 0.003 和 0.026，比非长尾数据对应的增长值更大，说明改进方法对长尾数据预测的准确率有较大的提高。由此可见，本章提出的 DEM 方法不仅可以有效提高长尾数据的补全性能，同时也能进一步提高非长尾数据的补全性能，提高其知识的推理准确率。

表 4.3: FB15K 非长尾测试集的链路预测结果

	MR	FMR	MRR	FMRR	Hit@10	FHit@10
TransE	203.17	119.80	0.250	0.427	0.515	0.674
TransE-DEM	**180.56**	**97.69**	**0.260**	**0.429**	**0.517**	**0.679**
TransH	**200.73**	117.56	0.246	0.422	0.511	0.671
TransH-DEM	200.86	**117.53**	**0.261**	**0.438**	**0.521**	**0.682**
TransD	**199.81**	**116.63**	0.249	0.427	0.514	**0.676**
TransD-DEM	208.94	126.57	**0.259**	**0.432**	**0.517**	0.675

表 4.4: FB15K 长尾测试集的链路预测结果

	MR	FMR	MRR	FMRR	Hit@10	FHit@10
TransE	438.86	224.38	**0.219**	0.328	0.329	0.504
TransE-DEM	**416.08**	**183.15**	0.215	**0.340**	**0.332**	**0.530**
TransH	**426.43**	216.45	0.226	0.331	0.332	0.510
TransH-DEM	441.35	**207.40**	**0.229**	**0.363**	**0.336**	**0.534**
TransD	439.65	225.48	0.217	0.326	0.325	0.507
TransD-DEM	**421.33**	**191.45**	**0.236**	**0.372**	**0.342**	**0.549**

5 在 DEM 方法中引入相似度计算的 SDEM 方法

本章继续对 DEM 方法的目标函数进行改进，从而使得进一步提高长尾数据的补全性能。本章设计了一种在 DEM 方法中引入相似度计算的 SDEM 方法。以下将从 SDEM 方法介绍及实验评估两个方面进行详细介绍。

5.1 在 DEM 方法中引入相似度计算的 SDEM 方法介绍

Guo 等人 [55]在模型中融入实体类别，认为具有相同类型的实体在嵌入空间中是彼此邻近的；Zhang 等人 [68]考虑语义上相似的关系，将其构成关系集群，在层次类别建模上有较好效果；Wen 等人 [69]提出基于实体相似性的 KGE 方法 SimE，将实体的相似性和拉普拉斯特征映射相结合，取得了较好效果。基于以上想法，本章节考虑长尾数据中的三元组与非长尾数据中相似三元组的相似度，期望长尾数据和非长尾数据间相似的三元组表示能够更加接近。因此引入三元组的相似度计算来改进 DEM 方法中的长尾数据目标函数，从而使得长尾数据的补全性能进一步提升。

5.1.1 目标函数

5.1.1.1 非长尾数据的目标函数

对于非长尾数据，依旧采用以下函数作为目标函数来训练实体和关系的向量表示:

$$L_{non} = \sum_{(h,r,t)\in S_{non}} \sum_{(h',r,t')\in S'} [\gamma + f_r(h,t) - f_r(h',t')]_+. \tag{5.1}$$

其中 $[x]_+$ 表示 $\max(0,x)$，$\gamma > 0$ 是一个超参数。$f_r(h,t)$ 表示评分函数，在不同的方法中可以使用不同的评分函数。例如，在 TransE 中，评分函数被定义为 $f_r(h,t) = \|h+r-t\|_{L_n}$。$S'$ 表示负采样集合，采样方法如下：

$$S' = \{(h',r,t)|h' \in \mathcal{E}\} \cup \{(h,r,t')|t' \in \mathcal{E}\}. \tag{5.2}$$

5.1.1.2 长尾数据的目标函数

针对长尾数据，引入三元组的相似度计算改进其目标函数，希望模型训练后，长尾数据的三元组表示与非长尾数据的三元组表示更加接近，从而促使长尾数据的补全效果进一步提升。

首先对于长尾数据中的每一个三元组 (h,r,t)，需要在非长尾数据中确定一个与 (h,r,t) 相似的三元组 (h'',r'',t'')，该对应三元组在头实体、关系以及尾实体上应尽可能一样。除次之外，确定的相似三元组还应该有较好的向量表示，这样才能帮助对应的长尾数据改善其三元组表示，从而提升补全性能。具体的搜索过程见算法 4。步 1 中的 $rank_h$ 表示：首先使用实体集 $\psi(\psi = \mathcal{E} - \{h\})$ 中的实体逐个替换 (h,r,t) 中的 h，得到重构的错误三元组集 $\{(h',r,t)\,|h' \in \psi\}$，然后计算 $\{(h',r,t)\,|h' \in \psi\} \cup \{(h,r,t)\}$ 中所有

三元组得分，再将三元组得分按升序排列，(h,r,t) 排名的数值即为 $rank_h$；$rank_t$ 也按相似的计算方式得到。较小的排序代表该三元组的向量表示更良好。从第 4-7 步可以看出，该搜索算法尽可能保证搜索的三元组更加相似，第 8 步尽可能保证从搜索出的众多三元组中确定一个排序最好的三元组。

Algorithm 4 针对长尾数据对应非长尾数据的搜索算法框架

输入： 长尾数据集$S_{long}=\{(h_i,r_i,t_i)\,|i=1,...,N\}$；
非长尾数据$S_{non}=\{(h_j,r_j,t_j)\,|j=1,...,M\}$。
输出： 数据集$S_{non-long}$。
1: 固定最优超参数，利用TransE训练得到实体和关系的嵌入向量，再计算S_{non} 中每个三元组的排序结果$rank_h$和$rank_t$。
2: 令$k=1$。
3: 取出$(h_k,r_k,t_k)\in S_{long}$。
4: 取出数据集S_{non}中头实体等于h_k且尾实体等于t_k的所有三元组，构成集合S'_k。若$S'_k\neq\emptyset$，转步8；否则转步5。
5: 取出数据集S_{non}中头实体等于h_k或尾实体等于t_k的所有三元组，构成集合S'_k。若$S'_k\neq\emptyset$，转步8；否则转步6。
6: 取出数据集S_{non}中关系等于r_k的所有三元组，构成集合S'_k。若$S'_k\neq\emptyset$，转步8；否则转步7。
7: 在S_{non}中随计选取一个三元组并写入数据集$S_{non-long}$，令$k:=k+1$转步3。
8: 取出S'_k中排序$rank_h+rank_t$最小的一个三元组并写入数据集$S_{non-long}$（当出现有相同最小排序和$rank_h+rank_t$时，则随机选取一个）。
9: 令$k:=k+1$转步3。

为了能够衡量长尾数据中三元组的向量表示与对应非长尾数据中三元组的向量表示之间的相似程度，本节设计了函数 L_{sim}，具体如下：

$$L_{sim}=\sum_{\substack{(h,r,t)\in S_{long}\\(h'',r'',t'')\in S_{non-long}}}\frac{\exp(\cos((h+r-t),(h''+r''-t'')))}{\sum\limits_{(h_j,r_j,t_j)\in S_{non}}\exp(\cos((h+r-t),(h_j+r_j-t_j)))}\tag{5.3}$$

对于长尾数据中的每一个三元组 (h,r,t)，(h'',r'',t'') 是 (h,r,t) 在非长尾数据中对应的相似三元组，$S_{non-long}$ 是长尾数据 S_{long} 中的三元组在非长尾数据中对应的相似三元组所构成的数据集，具体的搜索过程见算法 4。$(h+r-t)$ 表示三元组 (h,r,t) 的向量表示，$\cos((h+r-t),(h''+r''-t''))$ 近似表示三元组 (h,r,t) 与其对应的相似三元组 (h'',r'',t'') 向量表示之间的相似度。为进一步保证该相似度值是一个正数，在外面嵌套一个指数函数得到 $\exp(\cos((h+r-t),(h''+r''-t'')))$，$\sum\limits_{(h_j,r_j,t_j)\in S_{non}}\exp(\cos((h+r-t),(h_j+r_j-t_j)))$ 计算了三元组 (h,r,t) 与非长尾数据中所有三元组的相似度之和，再通过计算两者之间的比值来衡量长尾数据中三元组 (h,r,t) 的向量表示与非长尾数据中对应的相似三元组 (h'',r'',t'') 的向量表示的相似程度，该比值越大代表三元组 (h,r,t) 与三元组 (h'',r'',t'') 向量表示之间的相似度越大。

因此最终将长尾数据部分的目标函数定义为：

$$L_{long}=\sum_{(h,r,t)\in}\sum_{(h',r,t')\in S'}[\gamma+f_r(h,t)-f_r(h',t')]_+-L_{sim}\tag{5.4}$$

其中，$\gamma>0$ 是一个超参数。

5.1.2　算法步骤

首先，初始化实体潜在语义嵌入 $l_i^e \in \mathcal{L}^e$、关系潜在语义嵌入 $l_i^r \in \mathcal{L}^r$、原实体 $e_i \in \mathcal{E}$ 和原关系 $r_i \in \mathcal{R}$ 嵌入的参数。

5.1.2.1　SDEM方法中的E步更新

此步的目的是获得非长尾数据中的监督信息，因此将潜在语义嵌入参数当作隐变量，计算非长尾数据部分的目标函数，利用梯度优化算法仅更新隐变量——潜在语义嵌入参数，用以存储非长尾数据中准确的的监督信息。

首先，根据双重嵌入方法计算实体集中每个实体 $e_i \in \mathcal{E}$ 的嵌入表示和关系集中每个关系 $r_i \in \mathcal{R}$ 的嵌入表示:

$$s_{i,j}^e = e_i \bullet l_j^e \tag{5.5}$$

$$\alpha_{i,j}^e = \frac{\exp(s_{i,j}^e)}{\sum_{p=1}^{|\mathcal{L}^e|} \exp(s_{i,p}^e)} \tag{5.6}$$

$$e_i := \frac{\sum_{j=1}^{|\mathcal{L}^e|} \alpha_{i,j}^e l_j^e}{2} + \frac{e_i}{2} \tag{5.7}$$

$$s_{i,j}^r = r_i \bullet l_j^r \tag{5.8}$$

$$\alpha_{i,j}^r = \frac{\exp(s_{i,j}^r)}{\sum_{p=1}^{|\mathcal{L}^r|} \exp(s_{i,p}^r)} \tag{5.9}$$

$$r_i := \frac{\sum_{j=1}^{|\mathcal{L}^r|} \alpha_{i,j}^r l_j^r}{2} + \frac{r_i}{2} \tag{5.10}$$

然后根据 (5.1) 式计算非长尾数据部分的目标函数 L_{non}，为最小化该目标函数，利用梯度优化算法更新隐变量，即实体潜在语义嵌入 $l_i^e \in \mathcal{L}^e$ 和关系潜在语义嵌入 $l_i^r \in \mathcal{L}^r$ 的参数:

$$l_i^e := \arg\min_{l_i^e} L_{non}, i = 1, ..., |\mathcal{L}^e| \tag{5.11}$$

$$l_i^r := \arg\min_{l_i^r} L_{non}, i = 1, ..., |\mathcal{L}^r| \tag{5.12}$$

该过程对应算法 5 中的步 2 至步 5。

5.1.2.2 SDEM方法中的M步更新

此步的目的是将非长尾数据中的信息通过隐变量迁移到长尾数据中，因此在该步固定隐变量的参数，利用隐变量指导其余参数的更新。因此该步让潜在语义嵌入参数固定不变，利用梯度优化算法更新原实体、原关系嵌入参数，实现知识的迁移。

首先，利用在E步中得到的实体潜在语义嵌入 $l_i^e \in \mathcal{L}^e$ 和关系潜在语义嵌入 $l_i^r \in \mathcal{L}^r$，根据 (5.5-5.7) 式重新计算实体集中每个实体 $e_i \in \mathcal{E}$ 的嵌入表示，根据 (5.8-5.10) 式重新计算关系集中每个关系 $r_i \in \mathcal{R}$ 的嵌入表示。

然后计算所有数据的目标函数：

$$L_{all} = L_{long} + L_{non} \tag{5.13}$$

其中 L_{non}、L_{long} 在 5.1.1.2 节的 (5.1) 式和 (5.4) 式已介绍。

为最小化以上目标函数，利用梯度优化算法更新原实体、原关系嵌入的参数，即 $e_i \in \mathcal{E}$，$r_i \in \mathcal{R}$ 的参数。

$$e_i := \arg\min_{e_i} L_{all}, i = 1, ..., |\mathcal{E}| \tag{5.14}$$

$$r_i := \arg\min_{r_i} L_{all}, i = 1, ..., |\mathcal{R}| \tag{5.15}$$

该过程对应算法 5 中的步 6 至步 9。

SDEM 的算法设计如下：

Algorithm 5 SDEM 算法步骤

输入： 非长尾训练集S_{non};
长尾训练集S_{long};
实体集$\mathcal{E}$;
关系集$\mathcal{R}$;
批次数量T;
迭代次数N;
负采样样本的数量neg。
输出： 实体表示$e_i \in \mathcal{E}$;关系表示$r_i \in \mathcal{R}$;实体潜在语义表示$\mathcal{L}^e$;关系潜在语义表示$\mathcal{L}^r$。
1: 初始化 $e_i \in \mathcal{E}$，$r_i \in \mathcal{R}$，$\mathcal{L}^e$，$\mathcal{L}^r$，$n = 1$。
2: 令$t = 1$。
3: 采集小批次非长尾训练集并根据负采样样本的数量neg进行负采样得到S_{non,b_1}。
4: 根据(5.1)式计算非长尾数据的目标函数L_{non,b_1}，通过梯度下降法更新$l_i^e \in \mathcal{L}^e$，$l_i^r \in \mathcal{L}^r$。
5: 若$t > T$，则转步6；否则，令$t := t + 1$转步3。
6: 令$t = 1$。
7: 采集小批次非长尾数据和长尾训练集并根据负采样样本数量neg进行负采样得到S_{non,b_1}和S_{long,b_2}。
8: 根据(5.1)和(5.4)式计算所有数据的目标函数$L_{non,b_1} + L_{long,b_2}$，固定参数$\mathcal{L}^e$和$\mathcal{L}^r$，通过梯度下降法更新$e_i \in \mathcal{E}$，$r_i \in \mathcal{R}$。
9: 若$t > T$，则转步10；否则，令$t := t + 1$转步7。
10: 若$n > N$，则算法终止；否则，令$n := n + 1$转步2。
11: **返回** 实体表示$e_i \in \mathcal{E}$;关系表示$r_i \in \mathcal{R}$;实体潜在语义表示$\mathcal{L}^e$;关系潜在语义表示$\mathcal{L}^r$。

5.2 实验与评估

5.2.1 数据集

为了验证本章提出的 EM-KGE 方法的有效性，本章选择了基准数据集 FB15K，它是从 Freebase 中提取的一个数据集，Freebase 提供了世界的一般事实，例如，三元组 (Steve Jobs, founded, Apple Inc.) 在人名实体 Steve Jobs 和组织实体 Apple Inc. 之间建立了 founded 关系。数据集 FB15K 中涵盖了 14951 个实体和1345 个关系，包括 483142 条训练集、59071 条测试集和 50000 条验证集，具体的统计信息如表 3.1 所示。同时，FB15K 数据集中有大量本文提到的冗余关系，例如笛卡尔积关系、反向关系和近似冗余关系。

5.2.2 链接预测

链接预测是用于评估知识图谱嵌入方法性能的一个常见任务，它是指当三元组中的其中一个实体和关系已知时，预测出未知的实体。具体地，对于三元组 $(?, r, t)$，需预测出未知的头实体 h，对于三元组 $(h, r, ?)$，需预测出未知的尾实体 t。首先，取出测试集中的一个三元组 (h, r, t)，该三元组(h, r, t) 是正确三元组。然后对于该三元组 (h, r, t)，使用实体集 $\mathcal{E} - \{h\}$ 中的每个实体对头实体 h 或尾实体 t 进行逐一替换，从而构造新的错误三元组，这样的错误三元组有 $|\mathcal{E}| - 1$ 个。再利用评分函数计算该正确三元组以及 $|\mathcal{E}| - 1$ 个错误三元组的分数，将分数按升序进行排列。若是进行头替换，则将其正确三元组的排序记为 $rank_h$，若是进行尾替换，则将其正确三元组的排序记为 $rank_t$，该排序越小越好。

本章将使用以下评估指标：MR、MRR 和 Hit@N。MR 表示该测试集中所有正确三元组的排名平均值，MRR 表示该测试中所有正确三元组的排名的倒数平均值，Hit@N 表示该测试集中正确三元组的排序在前 N 的百分比。由此可见，MR 的值越小代表方法越有效，MRR 和 Hit@N 的值越大代表方法越有效。

但是，在错误三元组构造的过程中，可能构造的三元组在训练集、测试集或验证集中已经存在，那么该三元组不能被认定为错误三元组，并且该三元组的分数可能会低于测试的正确三元组的分数，导致正确三元组的预测排序增大，从而影响评估结果，因此在测试时可进行过滤设置，将构造的这类三元组过滤掉，不参与排序。将过滤后得到的相应指标记为 FMR、FMRR 和 FHit@N。

5.2.3 实验配置

本节选择 TransE、TransH 和 TransD 作为实验的原始方法，并在此基础上用 SDEM 方法进行改进。实验环境为 64GB 内存和 TiTAN XP GPU 的 Intel Xeon(R) Silver 4114 CPU 的个人工作站，使用 Python 中的 Pytorch 实现代码，并可以在以下网址查看相关的数据和源代码：`https://github.com/HMH-123/KGE-Code.git`。

为更好地训练模型，本节设置的参数范围见表 5.1。其中 α_1 是 SDEM 方法的学习率，α_2 是原始方法的学习率，T 是批次数量，neg 是负采样数量，$|\mathcal{L}^e|$ 表示实体 LSEs 的数量，$|\mathcal{L}^r|$ 表示关系 LSEs 的数量，d 是嵌入向量的维度，N 是迭代次数，$optimizer$ 是

pytoch 中的优化器。

表 5.1: 实验参数配置表

参数名	参数值
α_1	{0.0001，0.0005，0.0008}
α_2	{0.1,0.2,0.3}
T	{100,200,300}
γ	{3,4,5}
neg	{10,15,20,25}
$\|\mathcal{L}^e\|$	{50，100，200，300}
$\|\mathcal{L}^r\|$	{50，100，200，300}
$optimizer$	{“SGD”,“Adam”}
d	100
N	50

5.2.4　实验结果

本章选择了经典的基于翻译的 KGE 方法——TransE、TransH、TransD，作为基准线，然后利用 DEM 方法和 SDEM 方法对原始方法进行改进，改进后的方法相应地记为 X-DEM 和 X-SDEM，例如利用 DEM 方法和 SDEM方法改进 TransE 方法后分别记为 TransE-DEM 和 TransE-SDEM，并将实验结果分为三组。

从表 5.2 中可以发现：(1) 在三组结果中，改进方法 SDEM 的结果都优于原始方法，既能提高正确三元组的整体预测排名，也可以提高推理的准确率，证实了 SDEM 方法的有效性；(2) 在三组结果中，改进方法 SDEM 方法的结果几乎都优于 DEM 方法，表明 SDEM 方法通过引入相似度计算来改进目标函数是有效的。

表 5.2: FB15K 测试数据的链路预测结果

	MR	FMR	MRR	FMRR	Hit@10	FHit@10
TransE	224.81	129.40	0.248	0.418	0.498	0.658
TransE-DEM	202.18	105.53	**0.256**	0.421	0.500	0.666
TransE-SDEM	**201.88**	**105.03**	**0.256**	**0.425**	**0.502**	**0.670**
TransH	221.45	126.64	0.245	0.413	0.495	0.657
TransH-DEM	222.94	125.78	0.258	0.431	0.504	0.668
TransH-SDEM	**218.20**	**121.34**	**0.259**	**0434**	**0.506**	**0.675**
TransD	221.82	126.62	0.246	0.417	0.497	0.660
TransD-DEM	228.43	132.53	0.257	0.427	0.501	0.663
TransD-SDEM	**209.84**	**113.41**	**0.258**	**0.430**	**0.505**	**0.670**

为了进一步比较原始方法和改进方法 DEM 和 SDEM，同样将测试集划分为非长尾测试集和长尾测试集，并计算每个指标的值。表 5.3 展示了在非长尾测试集中的测试结果，表 5.4 展示了在长尾测试集中的测试结果，通过表 5.3 和表 5.4 可以发现：(1) 表 5.3 说明 SDEM 方法在所有指标上都优于原始方法，几乎优于 DEM 方法，只有在极个别指标上略差于 DEM 方法，且在多个指标上都能达到最优值，证实了 SDEM 方法对非长尾数据的补全性能有一定的提升作用，且在引入相似度计算后能够进一步提高方法的有效

性。(2) 表 5.4 说明 SDEM 方法几乎在所有指标上都优于原始方法，在绝大多数结果上都优于 DEM 方法，且在多个指标上能达到最优值，证实了 SDEM 方法能够进一步提升长尾数据的补全性，在引入相似度计算后能达到提高长尾数据知识的推理能力的目的。

表 5.3: FB15K 非长尾测试集的链路预测结果

	MR	FMR	MRR	FMRR	Hit@10	FHit@10
TransE	203.17	119.80	0.250	0.427	0.515	0.674
TransE-DEM	**180.56**	**97.69**	**0.260**	0.429	0.517	0.679
TransE-SDEM	180.61	97.79	**0.260**	**0.433**	**0.519**	**0.683**
TransH	200.73	117.56	0.246	0.422	0.511	0.671
TransH-DEM	200.86	117.53	0.261	0.438	0.521	0.682
TransH-SDEM	**195.94**	**112.51**	**0.262**	**0.441**	**0.523**	**0.688**
TransD	199.81	116.63	0.249	0.427	0.514	0.676
TransD-DEM	208.94	126.57	0.259	0.432	0.517	0.675
TransD-SDEM	**188.55**	**105.68**	**0.260**	**0.436**	**0.522**	**0.682**

表 5.4: FB15K 长尾测试集的链路预测结果

	MR	FMR	MRR	FMRR	Hit@10	FHit@10
TransE	438.86	224.38	**0.219**	0.328	0.329	0.504
TransE-DEM	416.08	183.15	0.215	0.340	0.332	0.530
TransE-SDEM	**412.31**	**176.66**	**0.219**	**0.349**	**0.334**	**0.538**
TransH	426.43	216.45	0.226	0.331	0.332	0.510
TransH-DEM	441.35	**207.40**	**0.229**	0.363	**0.336**	0.534
TransH-SDEM	**438.51**	208.68	**0.229**	**0.368**	0.335	**0.543**
TransD	439.65	225.48	0.217	0.326	0.325	0.507
TransD-DEM	421.33	191.45	**0.236**	0.372	**0.342**	**0.549**
TransD-SDEM	**420.48**	**189.92**	0.234	**0.370**	0.334	0.544

6 结论及展望

6.1 本文工作总结

本文针对现有的大多数知识图谱嵌入模型对非长尾数据有较好的补全性能，而对长尾数据的补全性较差的问题做了进一步研究，提出利用非长尾数据中蕴含丰富语义信息来增强长尾数据的语义信息，进而提升长尾数据的补全性能。本文共提出了三个改进方法，EM-KGE 方法在原始知识图谱嵌入方法中融入 EM 算法思想，简单的将冗余实体作为隐变量建立数据之间的关系，通过交替更新，通过冗余实体的向量表示来传递非长尾数据中的监督信息；DEM 方法进一步引入双重嵌入技术来改进隐变量，将其中的潜在语义嵌入作为隐变量，更好的传递非长尾数据中的知识；SDEM 方法在 DEM 方法的基础上引入三元组的相似度计算来改进目标函数，通过使长尾数据中的三元组与其在非长尾数据中相似的三元组的表示更加接近来进一步提高长尾数据的补全性能。通过数值实验验证了本文提出的方法与部分现有方法相比，取得了非常好的效果。

6.2 未来工作展望

尽管本文提出的方法能够在一定程度上解决现有大多数知识图谱嵌入方法存在的问题，也能够得到良好的结果，但对于提高长尾数据的补全性能的问题，是否有更好更一般的方法来实现，也是值得继续思考的一个问题。同时，是否可以进一步改进评分函数，使得该评分函数更能描述知识图谱内的结构关系，从而使得向量表示更能表达语义知识，也是未来值得继续开展研究的方向。

参考文献

[1] 张海燕. 网络大数据的现状与展望研究[J]. 中国新通信, 2020, 22(18): 29-30.

[2] 姚萍, 李坤伟, 张一帆. 知识图谱构建技术综述[J]. 信息系统工程, 2020(05): 121-123.

[3] 漆桂林, 高桓, 吴天星. 知识图谱研究进展[J]. 情报工程, 2017, 3(1): 22.

[4] Chen X, Jia S, Xiang Y. A review: Knowledge reasoning over knowledge graph[J]. *Expert Systems with Applications*, 2020, 141: 112948.

[5] Zhang Z, Cai J, Zhang Y, et al. Learning hierarchy-aware knowledge graph embeddings for link prediction[C]. *Proceedings of the Association for the Advance of Artificial Intelligence Conference on Artificial Intelligence*, 2020, 34(03): 3065-3072.

[6] Zou X. A survey on application of knowledge graph[C]. *Journal of Physics: Conference Series*, 2020, 1487(1): 012016.

[7] Rossi A, Barbosa D, Firmani D, et al. Knowledge graph embedding for link prediction: A comparative analysis[J]. *ACM Transactions on Knowledge Discovery from Data*, 2021, 15(2): 1-49.

[8] Singhal A. Introducing the knowledge graph: things, not strings. *Official Google Blog*, 2012. URL https://googleblog. blogspot.co.za/2012/05/introducing-knowledge-graph-things-not.html.

[9] Huang X, Zhang J, Li D, et al. Knowledge graph embedding based question answering[C]. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 2019: 105-113.

[10] Saxena A, Tripathi A, Talukdar P. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings[C]. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020: 4498-4507.

[11] Qiu D, Zhang Y, Feng X, et al. Machine reading comprehension using structural knowledge graph-aware network[C]. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 2019: 5896-5901.

[12] Ding M, Zhou C, Chen Q, et al. Cognitive graph for multi-hop reading comprehension at scale[J]. arXiv preprint arXiv:1905.05460, 2019.

[13] Li L, Wang P, Yan J, et al. Real-world data medical knowledge graph: construction and applications[J]. *Artificial Intelligence in Medicine*, 2020, 103: 101817.

[14] Gong F, Wang M, Wang H, et al. SMR: medical knowledge graph embedding for safe medicine recommendation[J]. *Big Data Research*, 2021, 23: 100174.

[15] Wang Y, Dong L, Zhang H, et al. An enhanced multi-modal recommendation based on alternate training with knowledge graph representation[J]. *Ieee Access*, 2020, 8: 213012-213026.

[16] Zhou K, Zhao W X, Bian S, et al. Improving conversational recommender systems via knowledge graph based semantic fusion[C]. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020: 1006-1014.

[17] 田玲, 张谨川, 张晋豪, 周望涛, 周雪. 知识图谱综述——表示、构建、推理与知识超图理论[J]. 计算机应用, 2021, 41(08): 2161-2186.

[18] Hogan A, Blomqvist E, Cochez M, et al. Knowledge graphs[J]. *Synthesis Lectures on Data, Semantics, and Knowledge*, 2021, 12(2): 1-257.

[19] Auer S, Bizer C, Kobilarov G, et al. Dbpedia: A nucleus for a web of open data[M]. *The Semantic Web*, 2007: 722-735.

[20] Bizer C, Lehmann J, Kobilarov G, et al. Dbpedia-a crystallization point for the web of data[J]. *Journal of Web Semantics*, 2009, 7(3): 154-165.

[21] Navigli R, Ponzetto S P. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network[J]. *Artificial Intelligence*, 2012, 193: 217-250.

[22] Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]. *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. 2008: 1247-1250.

[23] Suchanek F M, Kasneci G, Weikum G. Yago: a core of semantic knowledge[C]. *Proceedings of the 16th International Conference on World Wide Web*. 2007: 697-706.

[24] Hoffart J, Suchanek F M, Berberich K, et al. Yago2: A spatially and temporally enhanced knowledge base from Wikipedia[J]. *Artificial Intelligence*, 2013, 194: 28-61.

[25] Mahdisoltani F, Biega J, Suchanek F. Yago3: A knowledge base from multilingual wikipedias[C]. *7th Biennial Conference on Innovative Data Systems Research*, 2014.

[26] Vrandečić D, Krötzsch M. Wikidata: a free collaborative knowledgebase[J]. *Communications of the ACM*, 2014, 57(10): 78-85.

[27] Speer R, Havasi C. Representing general relational knowledge in conceptnet 5[C]. *LREC*. 2012, 2012: 3679-3686.

[28] Krishnan A. Making search easier: How Amazon' s Product Graph is helping customers find products more easily. *ed. Amazon Blog*, 2018. URL https://blog. aboutamazon. com/innovation/making-search-easier.

[29] He Q, Chen B, Argawal D. Building the linkedin knowledge graph. *Engineering. Linkedin. Com*, 2016. URL https://engineering.linkedin.com/blog/2016/10/building-the-linkedin-knowledge-graph.

[30] Noy N, Gao Y, Jain A, et al. Industry-scale knowledge graphs: lessons and challenges[J]. *Communications of the ACM*, 2019, 62(8): 36-43.

[31] Luo X, Liu L, Yang Y, et al. AliCoCo: Alibaba e-commerce cognitive concept net[C]. *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 2020: 313-327.

[32] 王桢. 基于嵌入模型的知识图谱补全[D]. 中山大学, 2017.

[33] Dong X, Gabrilovich E, Heitz G, et al. Knowledge vault: A web-scale approach to probabilistic knowledge fusion[C]. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2014: 601-610.

[34] 刘知远, 孙茂松, 林衍凯, 谢若冰. 知识表示学习研究进展[J]. 计算机研究与发展, 2016, 53(02): 247-261.

[35] Wang Q, Mao Z, Wang B, et al. Knowledge graph embedding: A survey of approaches and applications[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2017, 29(12): 2724-2743.

[36] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data[J]. *Advances in Neural Information Processing Systems*, 2013, 2: 2787-2795.

[37] Wang Z, Zhang J, Feng J, et al. Knowledge graph embedding by translating on hyperplanes[C]. *Proceedings of the Association for the Advance of Artificial Intelligence Conference on Artificial Intelligence*. 2014, 28(1): 1112-1119.

[38] Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion [C]. *Twenty-ninth the Association for the Advance of Artificial Intelligence Conference on Artificial Intelligence*. 2015: 2181-2187.

[39] 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*. 2015: 687-696.

[40] Xiao H, Huang M, Hao Y, et al. TransA: An adaptive approach for knowledge graph embedding[J]. arXiv preprint arXiv:1509.05490, 2015.

[41] Xiao H, Huang M, Hao Y, et al. Transg: A generative mixture model for knowledge graph embedding[J]. arXiv preprint arXiv:1509.05488, 2015.

[42] He S, Liu K, Ji G, et al. Learning to represent knowledge graphs with gaussian embedding[C]. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 2015: 623-632.

[43] Ji G, Liu K, He S, et al. Knowledge graph completion with adaptive sparse transfer matrix[C]. *Thirtieth the Association for the Advance of Artificial Intelligence Conference on Artificial Intelligence*. 2016: 985-991.

[44] Fan M, Zhou Q, Chang E, et al. Transition-based knowledge graph embedding with relational mapping properties[C]. *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing*. 2014: 328-337.

[45] Nickel M, Tresp V, Kriegel H P. A three-way model for collective learning on multi-relational data[C]. *International Conference on Machine Learning*. 2011: 809-816.

[46] Yang B, Yih W, He X, et al. Embedding entities and relations for learning and inference in knowledge bases[J]. arXiv preprint arXiv:1412.6575, 2014.

[47] Trouillon T, Welbl J, Riedel S, et al. Complex embeddings for simple link prediction[C]. *International Conference on Machine Learning*, 2016: 2071-2080.

[48] Dettmers T, Minervini P, Stenetorp P, et al. Convolutional 2d knowledge graph embeddings[C]. *Thirty-econd Thirtieth the Association for the Advance of Artificial Intelligence Conference on Artificial Intelligence Conference on Artificial Intelligence*. 2018, 32(1): 1811-1818.

[49] Nguyen D Q, Nguyen T D, Nguyen D Q, et al. A novel embedding model for knowledge base completion based on convolutional neural network[J]. arXiv preprint arXiv:1712.02121, 2017.

[50] Vu T, Nguyen T D, Nguyen D Q, et al. A capsule network-based embedding model for knowledge graph completion and search personalization[C]. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2019: 2180-2189.

[51] Schlichtkrull M, Kipf T N, Bloem P, et al. Modeling relational data with graph convolutional networks[C]. *European Semantic Web Conference*, 2018: 593-607.

[52] Nathani D, Chauhan J, Sharma C, et al. Learning attention-based embeddings for relation prediction in knowledge graphs[J]. arXiv preprint arXiv:1906.01195, 2019.

[53] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie,and Jian Tang. RotatE: knowledge graph embedding by relational rotation in complex space[C]. *Proceedings of the International Conference on Learning Representations*. 2019: 926-934.

[54] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. Quaternion knowledge graph embeddings[C]. *In Advances in Neural Information Processing Systems*, 2019: 2731-2741.

[55] Guo S, Wang Q, Wang B, et al. Semantically smooth knowledge graph embedding[C]. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 2015: 84-94.

[56] Xie R, Liu Z, Sun M. Representation learning of knowledge graphs with hierarchical types[C]. *International Joint Conference on Artificial Intelligence*, 2016: 2965-2971.

[57] Lin Y, Liu Z, Luan H, et al. Modeling relation paths for representation learning of knowledge bases[J]. arXiv preprint arXiv:1506.00379, 2015.

[58] Lao N, Cohen W W. Relational retrieval using a combination of path-constrained random walks[J]. *Machine Learning*, 2010, 81(1): 53-67.

[59] Lao N, Mitchell T, Cohen W. Random walk inference and learning in a large scale knowledge base[C]. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011: 529-539.

[60] Socher R, Chen D, Manning C D, et al. Reasoning with neural tensor networks for knowledge base completion[J]. *Advances in Neural Information Processing Systems*, 2013, 26.

[61] Xie R, Liu Z, Jia J, et al. Representation learning of knowledge graphs with entity descriptions[C]. *Proceedings of Thirtieth the Association for the Advance of Artificial Intelligence Conference on Artificial Intelligence Conference on Artificial Intelligence*, 2016, 30(1): 2659-2665.

[62] Wang Z, Li J, Liu Z, et al. Text-enhanced representation learning for knowledge graph[C]. *Proceedings of International Joint Conference on Artificial Intelligent*, 2016: 4-17.

[63] Guo S, Wang Q, Wang L, et al. Jointly embedding knowledge graphs and logical rules[C]. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016: 192-202.

[64] Akrami F, Saeef M S, Zhang Q, et al. Realistic re-evaluation of knowledge graph completion methods: An experimental study[C]. *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 2020: 1995-2010.

[65] Dempster A P, Laird N M, Rubin D B. Maximum likelihood from incomplete data via the EM algorithm[J]. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1977, 39(1): 1-22.

[66] Metz L, Maheswaranathan N, Cheung B, et al. Meta-learning update rules for unsupervised representation learning[J]. arXiv preprint arXiv:1804.00222, 2018.

[67] Zhang Z, Zhuang F, Qu M, et al. Knowledge graph embedding with shared latent semantic units[J]. *Neural Networks*. 2021, 139: 140-148.

[68] Zhang Z, Zhuang F, Qu M, et al. Knowledge graph embedding with hierarchical relation structure[C]. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018: 3198-3207.

[69] 文洋, 张茂元, 周礼全, 张洁琼, 袁贤其. 基于实体相似性的知识表示学习方法[J]. 计算机应用研究, 2021, 38(04): 1008-1012.

附录A: 作者攻读硕士学位期间发表论文及科研情况

发表论文:

1. 何苗惠, 段旭祥, 吴至友. 提高长尾数据知识图谱补全性能的一种新算法[J/OL]. 运筹学学报: 1-14[2022-05-16]. http://kns.cnki.net/kcms/detail/31.1732.O1.20220424.1147.012.html.

致　谢

光阴似箭，日月如梭，转眼间我的研究生生活即将结束。回首初进校园的青涩、茫然、浮躁，如今的我多了一分成熟、坚韧、稳重。三年的生活和学习也成为我人生路程中可贵的一段经历，每一个收获，每一次进步，都离不开导师的教导，老师的教诲，同门的帮助，朋友的鼓励，家人的支持。在此，我谨向所有关心、帮助我的人们致以最诚挚的谢意。

首先，我要衷心的感谢我的指导老师吴至友教授，在我们学习的过程中，吴老师总是为我们解惑，提出指导性意见，教导我们做科研要苦心钻研，敢于质疑，善于思考。吴老师渊博的学识，严谨求实、认真负责的工作作风，严谨的治学精神不断地感染着我，激励着我。能够成为吴老师的学生是我人生中的一件幸事!

其次，衷心感谢高桓博士在研究生阶段对我的关心和帮助，指导我进行知识图谱的相关研究。高桓博士为人十分谦逊，对人热情，目标明确，做事严谨。无论是在科研上、生活上还是人生道路上都给予我很多的指导和影响，在此对他表示真心的感谢!

感谢学院的所有老师和领导，提供学习环境，营造良好学风。感谢我的师兄、师姐、师妹和师弟以及我的室友和同学们，感谢他们在学习上帮助我，在生活上关心我，在我迷茫和感到压力时给予我鼓励和建议。

感谢我的家人，正是因为他们对我极大的支持，无私的爱和付出，给予我受教育的机会，我才可以投入到学习中去，并最终完成学业!

最后，由衷的感谢评审专家在百忙之中抽出时间审阅我的论文，感谢您的辛勤劳动。感谢各位评委老师的指正和赐教，谢谢您们!

2022 年 5 月

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含他人已经发表或撰写过的研究成果，也不包含为获得重庆师范大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明。

学位论文作者签名：　　签字日期：　　年　月　日

学位论文版权使用授权书

本学位论文作者完全了解重庆师范大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权 重庆师范大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

学位论文作者签名：　　签字日期：　　年　月　日