计算机科学
COMPUTER SCIENCE

基于关系约束的上下文感知时态知识图谱补全

汪璟玢, 赖晓连, 林新宇, 杨心逸

引用本文

汪璟玢，赖晓连，林新宇，杨心逸.基于关系约束的上下文感知时态知识图谱补全[J]．计算机科学, 2023, 50(3): 23-33.

WANG Jingbin, LAI Xiaolian, LIN Xinyu, YANG Xinyi. Context-aware Temporal Knowledge Graph Completion Based on Relation Constraints [J]. Computer Science, 2023, 50(3): 23-33.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220400255

基于关系约束的上下文感知时态知识图谱补全

汪璟玢　赖晓连　林新宇　杨心逸
福州大学计算机与大数据学院　福州 350108

摘　要　现有的时间知识图谱补全模型仅考虑四元组自身的结构信息,忽略了实体隐含的邻居信息和关系对实体的约束,导致模型在时态知识图谱补全任务上表现不佳。此外,一些数据集在时间上呈现不均衡的分布,导致模型训练难以达到一个较好的平衡点。针对这些问题,提出了一个基于关系约束的上下文感知模型(CARC)。CARC通过自适应时间粒度聚合模块来解决数据集在时间上分布不均衡的问题,并使用邻居聚合器将上下文信息集成到实体嵌入中,以增强实体的嵌入表示。此外,设计了四元组关系约束模块,使具有相同关系约束的实体嵌入彼此相近,不同关系约束的实体嵌入彼此远离,以进一步增强实体的嵌入表示。在多个公开的时间数据集上进行了大量实验,实验结果证明了所提模型的优越性。

关键词:时间知识图谱;链路预测;时间区间预测;关系约束;邻居信息;时间粒度
中图法分类号　TP391

Context-aware Temporal Knowledge Graph Completion Based on Relation Constraints

WANG Jingbin,LAI Xiaolian,LIN Xinyu and YANG Xinyi
College of Computer and Data Science,Fuzhou University,Fuzhou 350108,China

Abstract　The existing temporal knowledge graph completion models only consider the structural information of the quadruple itself,ignoring the implicit neighbor information and the constraints of relationships on entities,which leads to the poor performance of the models on the temporal knowledge graph completion task. In addition,some datasets exhibit unbalanced distribution in time,which makes it difficult for model training to achieve a good balance. To address these problems,the paper proposes a context-aware model based on relation constraints(CARC). CARC solves the problem of an unbalanced distribution of datasets in time through an adaptive time granularity aggregation module and uses a neighbor-aggregator to integrate contextual information into entity embeddings to enhance the embedding representation of the entity. In addition,the quadruple relation constraint module is designed to make the embeddings of entities with the same relational constraints close to each other,while those with different relational constraints are far away from each other,which further enhances the embedding representation of entities. Extensive experiments are conducted on several publicly available temporal datasets,and the experimental results prove the superiority of the proposed model.

Keywords　Temporal knowledge graph,Link prediction,Time interval prediction,Relation constraint,Neighbor information,Time granularity

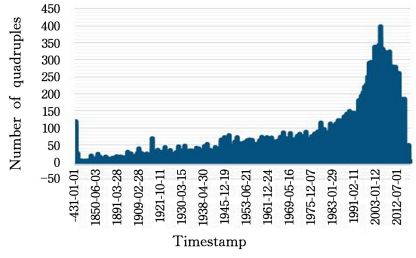
1　引言

时间知识图谱(Temporal Knowledge Graph,TKG)是将现实世界的知识抽象成一个由数十亿个四元组组成的复杂网络图。时间知识图谱中的知识以四元组(s,r,o,t)的形式表示,其中s和o分别表示头实体和尾实体,r表示关系,t表示时间戳。因为大多数时间知识图谱都是不完整的,所以时间知识图谱补全(Temporal Knowledge Graph Completion,TKGC)成为了时间知识图谱领域的主要挑战之一。为了解决这个问题,时间知识图谱嵌入(Temporal Knowledge Graph Embedding,TKGE)方法将实体、关系和时间嵌入到一个低维的向量空间中以获得低维表示,然后将低维表示输入到得分函数中来衡量四元组的合理性。

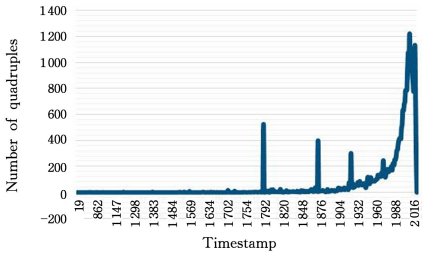
现实世界中,数据在不同时间上的不均衡分布是一个普遍的现象。比如,2021年7月1日是建党100周年,在这一天发生的与中国共产党相关的重要事实显然会比平常多。类似地,在现有的时间知识图谱中,也存在严重的数据分布不均衡问题,如YAGO11k和Wikidata12k(见图1)。从图中可以

到稿日期:2022-04-25　返修日期:2023-01-09
基金项目:国家自然科学基金(61672159);福建省自然科学基金(2021J01619)
This work was supported by the National Natural Science Foundation of China(61672159) and Natural Science Foundation of Fujian Province (2021J01619).
通信作者:汪璟玢(wjb@fzu.edu.cn)

看出，这两个数据集在各个时间戳的数据分布非常不均衡，如果直接训练，在数据量较多的时间上，模型训练容易欠拟合，对这类实例的预测将会变得很困难；在仅有少量实例的时间上，模型训练容易过拟合，因此模型训练无法达到一个较好的平衡点，导致模型在补全任务上的效果不佳。此外，YAGO11k 和 Wikidata12k 还存在大量时间戳缺失的问题，这也是导致模型补全性能下降的一个因素。



(a) YAGO11k



(b) Wikidata12k

图 1 YAGO11k 和 Wikidata12k 在各个时间戳上的数据分布

Fig. 1 Data distribution of YAGO11k and Wikidata12k in each timestamp

时间旋转模型(Temporal Rotation，TeRo)[1]通过设置时间粒度将出现频率低的时间戳合并为一个时间，能够在一定程度上缓解数据不均衡分布的问题。但是 TeRo 仅使用时间戳中的年份信息，容易导致合并不合理的问题。比如，对于实体 e，在时间戳为 t_1=2005-01-05，t_2=2005-12-12 和 t_3=2006-02-01 时，我们期望它在 t_2 时的嵌入表示应该与 t_3 时的嵌入表示更为接近，与 t_1 时的嵌入表示应差别较大。对于这种情况，TeRo 仅使用年份信息的方式是无法满足要求的，因为 TeRo 会将 t_1 和 t_2 当作同一个时间点(即 2005)，实体在 t_1 和 t_2 下具有相同的嵌入表示。此外，YAGO11k 中的一些时间戳包含了完整的年月日信息，如果丢弃月和日信息容易导致重要信息丢失。

在知识图谱中，实体除了自身的结构信息外，还蕴含着丰富的潜在信息，比如实体的上下文信息(邻居信息)和类型信息。实体的邻居信息是知识图谱中的重要信息，对邻居信息的合理利用，可以提高实体嵌入的质量，从而提升模型性能。现有模型中融合了邻居信息的模型包括多关系图神经网络模型(Graph Convolutional Network Model for Multi-relational Graphs，CompGCN)[2]和结合实体邻居信息的知识表示模型(Knowledge Representation Model That Combining Entity Neighbor Information，CombiNe)[3]，它们在补全任务中表现出了良好的性能。在直觉上实体邻居的重要性程度应与时间距离成反比，如图 2 所示，当预测(A，Make a visit，?，1987)时，(A，Make a visit，C，1902)与预测时间 1987 相对距离较远，因此分配的权重也较小。并且，由于 Threaten，Criticize 和 Cooperate economically 这 3 个邻居关系中 Cooperate economically 与预测关系 Make a visit 有最相近的语义，因此(A，Cooperate economically，B，1986)应该分配最大的权重。根据实体 A 的邻居信息可知，A 应该有较大的概率访问 B，而不太可能访问 C 或 D。但 CompGCN 和 CombiNe 都建立在静态知识图谱中，在聚合邻居信息时，会为(A，Make a visit，C，1902)和(A，Threaten，C，1987)分配统一的权重，忽略了实体邻居的重要性不同的问题，导致模型得到错误的答案。

实体关联的关系可以约束实体的类型，对具有相同关系约束的实体进行相似性建模，可以显著地改进实体嵌入表示并提高预测精度。对于一个实体“Apple”，如果仅靠实体自身的结构信息，我们无法判断“Apple”是水果还是 Apple 公司。但如果实体的关联关系是‘吃’，那么我们就可以很明确地知道“Apple”指的是苹果。现有的一些利用实体类型信息的模型，如类型嵌入知识表示学习模型(Type-embodied Knowledge Representation Learning，TKRL)[4]和基于类型的多重嵌入模型(Type-based Multiple Embedding Model，TransT)[5]，它们需要显式的实体类型输入，对没有显式提供实体类型信息的知识图谱的补全任务具有一定的局限性。自动实体类型表示模型(Automated Entity Type Representation Model，AutoETER)[6]能够自动编码实体类型信息，但是它建立在静态知识图谱上，忽视了事实的动态性。本文认为，实体的类型信息应受关系和时间的共同约束，比如，对于实体“鲁迅”，当相连的关系为“写作”时，他的实体类型是“作家”。但鲁迅于 1906 年弃医从文，因此在 1906 年前他的实体类型就不可能是“作家”，即 1906 年前，鲁迅与关系“写作”构成的四元组都是错误的。

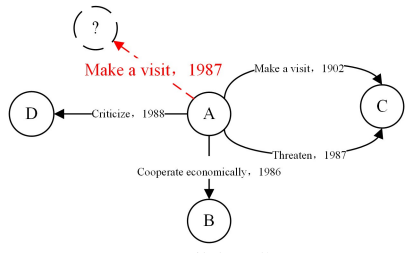


图 2 实体邻居信息

Fig. 2 Entity neighbor information

针对上述问题，本文提出了一个基于关系约束的上下文感知模型(Context-Aware model based on Relation Constraints，CARC)。首先，通过设定时间粒度，对数据集进行预处理，使得数据在时间分布上尽可能均衡，以解决数据不均衡分布给模型带来的问题。其次，提出了两个评估四元组有效性的模块——四元组结构模块和四元组关系约束模块。在四元组结构模块中，考虑到现有的一些模型通常将实体和关系嵌入到复数空间而不是嵌入常用的实数空间来提高表示学习能力，如基于张量分解的复数嵌入模型(Complex Embedding

Model based on Tensor Decompositions，TNT-ComplEx)[7]和TIMEPLEX模型[8]，因此我们将四元组嵌入到复数空间中获得初始化嵌入，然后通过邻居编码器聚合实体上下文信息，以增强实体的嵌入表示。在四元组关系约束模块中，为了避免模型参数量过大，四元组被嵌入到实数空间，该模块能够在未提供实体类型信息的情况下，通过关系对实体的约束来进一步增强实体嵌入表示，提高模型补全的能力。最后，将四元组结构模块的得分与四元组关系约束模块的得分按照一定的权重相加，获得四元组最终的得分。本文在4个公开的无类型信息的时间数据集上进行了链路预测实验和时间区间预测实验，实验结果表明，本文模型在MRR，Hits@N，aeIOU等多个评估指标上优于最先进的基线模型。

2 相关工作

知识表示学习是一种有效且可靠的知识补全技术，近年来有关知识图谱的表示学习方法层出不穷。复数嵌入模型(Complex Embeddings，ComplEx)[9]将三元组嵌入到复数空间中，能够建模对称/反对称、自反两种关系模型，在静态知识图谱补全任务上取得了一定的成效。AutoETER[6]将四元组嵌入到实数空间中，自动学习实体的类型嵌入，丰富了实体的一般特征，此外，它还可以推断和建模所有对称/反对称、自反和组合这3种关系模式，以及复杂的一对多、多对一和多对多关系。CompGCN利用图卷积神经网络(Graph Convolutional Network，GCN)[10]聚合实体的邻居信息，增强了实体的嵌入表示，该模型还通过设置基向量解决了过度参数化的问题。上述模型都建立在静态知识图谱上，在静态知识图谱补全任务上表现良好，但由于没有考虑时间信息，在时间知识图谱补全任务上性能不佳。

近年来，许多研究努力将静态知识图谱补全模型扩展到时间知识图谱中。比如，基于超平面的时间感知知识图谱嵌入模型(Hyperplane-based Temporally Aware Knowledge Graph Embedding，HyTE)[11]为每个时间戳定义了一个时间超平面，并将实体和关系投影到时间超平面中，然后将投影的嵌入用翻译模型(Translating Embedding，TransE)[12]进行处理以获得四元组的得分。García-Durán等[13]将时间戳划分为token序列，与关系一并输入到长短期记忆网络(Long Short-term Memory，LSTM)中，以获得不同时间下的关系表示，获得的关系表示可以应用于多个模型中。Jain等[8]将该方法应用于ComplEx中，得到了时间感知的ComplEx模型(Temporal-aware ComplEx，TA-ComplEx)。受历时词的启发，Goel等[14]将时间信息整合到实体嵌入中，提出了基于历时词嵌入的简单嵌入模型(Simple Embedding Based On Diachronic Embedding，DE-SimplE)。DE-SimplE认为实体嵌入中可能有一些随时间变化的特性和一些固定保持的特性，因此设置了用于控制时间特征百分比的超参数 $\gamma\in[0,1]$，在时间知识图谱补全任务中取得了显著的效果。TeRo将实体嵌入的时间演化定义为实体在复数空间中从初始时间到当前时间的旋转，该模型结合了旋转模型(Rotation Embedding，RotatE)[15]的优势，可以建模实体间的复杂关系(如自反关系)；此外，TeRo采用时间粒度合并部分时间戳，可以缓解数据集在时间上分布不均衡的问题。受四阶张量分解的启发，Lacroix等[7]将四元组嵌入到复数空间中，提出了TNT-ComplEx模型，该模型通过四元组的内积操作得到四元组的得分，得分函数定义为 $f(s,r,o,t)=[\boldsymbol{e}_s,\boldsymbol{r},\bar{\boldsymbol{e}}_o,\boldsymbol{t}]$。Jain等[8]提出的TIMEPLEX根据时间的特有性质，定义了3种类型的时间约束：关系的重复性、关系间的顺序、关系间的时间间隔。其中，关系的重复性即许多关系对于特定实体不会重复出现(例如，一个人只出生一次)，有些关系在固定周期内重复(例如，奥运会每4年出现一次)。关系间的顺序指对于一个给定的实体，一个关系先于另一个关系，比如PersonBornYear应在给定实体的PersonDiedYear之前。关系间的时间间隔指对于一个给定的实体，两个关系间的时间差值分布在一个平均值周围，例如PersonDiedYear减去PersonBornYear的平均值约为70。TIMEPLEX在没有额外时间约束输入的情况下，使用高斯分布建模这3种时间约束，在时间知识图谱补全任务中表现出了良好的性能。上述时间知识图谱补全模型，虽然在时间知识图谱补全任务中取得了一定的成效，但忽略了实体的邻居信息和关系对实体的约束，在时间知识图谱补全任务中具有一定的局限性。

3 CARC模型

3.1 相关定义

给定实体集合 E，关系集合 R，时间戳集合 $\mathfrak{T}$，四元组可以表示为 (s,r,o,t) 和 $(s,r,o,T=[t_b,t_e])$ 两种形式。TKG可以表示为四元组的集合 $\{(s,r,o,t)\mid s,o\in E,r\in R,t\in\mathfrak{T}\}$，其中 $s,o\in E$ 表示头实体和尾实体，$r\in R$ 表示关系，$t\in\mathfrak{T}$ 是形如“2021-11-12”的时间戳，T 表示时间区间，$t_b,t_e\in\mathfrak{T}$ 分别是开始时间和结束时间。TKGC可以表示为 (s,r,o,t)，$(s,r,?,t)$ 或 $(s,r,o,?)$，即基于已知的3个元素预测缺失的元素。本文用 $\boldsymbol{e}_s,\boldsymbol{e}_o,\boldsymbol{r},\boldsymbol{\tau}\in\mathbb{C}^{d_c}$ 表示头实体 s、尾实体 o、关系 r 和时间 τ 在复数空间中的嵌入，d_c 是复数空间的嵌入维度；使用 $\boldsymbol{c}_s,\boldsymbol{c}_o,\boldsymbol{c}_r,\boldsymbol{c}_\tau\in\mathbb{R}^d$ 表示头实体 s、尾实体 o、关系 r 和时间 τ 在实数空间中的嵌入，d 是实数空间的嵌入维度。

3.2 网络组成部分

基于关系约束的上下文感知模型CARC的整体结构如图3所示。首先，为了解决某些数据集在时间分布上不均衡导致模型性能不佳的问题，本文引入了自适应时间粒度聚合模块(见3.2.1节)；其次，为了充分利用实体邻居信息，在四元组结构模块中借助邻居编码器，获得头尾实体聚合邻居后的增强表示 $\boldsymbol{e}_s'$，$\boldsymbol{e}_o'$(见3.2.2节)；此外，考虑到同一关系连接的头(尾)实体应属于同一类型这一特点，本文提出了关系特定的实体相似性约束，以进一步优化关系约束下的实体嵌入(见3.2.3节)；最后，将四元组结构模块的得分 f_{stru} 与四元组关系约束模块的得分 f_{relation} 按照一定的权重聚合，获得四元组的最终得分 f_{final}(见3.2.4节)。

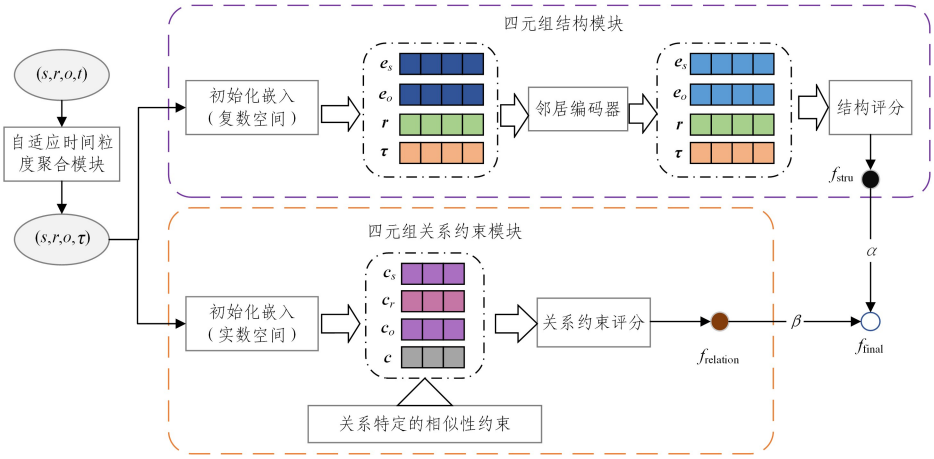


图 3 CARC 的总体框架图

Fig. 3 Overall framework of CARC

3.2.1 自适应时间粒度聚合模块

本文针对一些数据集在时间上分布不均衡的问题，为模型设置了一个超参数：时间粒度 *thre*。通过时间粒度，将出现频率较低的时间戳合并成一个时间戳，出现频率较高的时间戳则形成单独的时间戳，使得数据在时间分布上尽可能地均衡，以有效缓解数据不均衡分布对模型的影响。自适应时间粒度聚合模块的具体流程为：首先计算数据集中每个时间戳下的四元组数量 $num[t], t\in\mathfrak{T}$；其次，将时间戳按时间先后顺序排序；然后，根据时间粒度 *thre* 合并时间戳，当满足条件(1)时，$t_1, t_2, \cdots, t_i$ 被合并为同一时间 τ。

$$\begin{cases} num[t_1]+num[t_2]+\cdots+num[t_{i-1}]<\text{thre} \\ num[t_1]+num[t_2]+\cdots+num[t_{i-1}]+num[t_i]\geqslant\text{thre} \end{cases} \tag{1}$$

经过上述步骤，四元组 (s,r,o,t) 变为 (s,r,o,τ)，$(s,r,o,T=[t_b,t_e])$ 变为 $(s,r,o,T=[\tau_b,\tau_e])$。对于 $(s,r,o,T=[\tau_b,\tau_e])$，本文通过枚举操作，将其映射为时间点形式的四元组，比如 $(s,r,o,T=[\tau_0,\tau_3])$ 被映射为 $(s,r,o,\tau_0)(s,r,o,\tau_1)(s,r,o,\tau_2)$ 和 (s,r,o,τ_3)。

3.2.3 四元组结构模块

将自适应时间粒度聚合模块处理后的四元组 (s,r,o,τ) 输入到四元组结构模块中，经过初始化嵌入模块，获得头实体 s、尾实体 o、关系 r 和时间 τ 在复数空间的初始向量表示 $\boldsymbol{e}_s$，$\boldsymbol{e}_o, \boldsymbol{r}, \boldsymbol{\tau}\in\mathbb{C}^{d_c}$。为了捕获实体的邻域信息，本文将 $\boldsymbol{e}_s, \boldsymbol{e}_o, \boldsymbol{r}, \boldsymbol{\tau}$ 输入到邻居编码器中，获得头尾实体聚合邻居后的增强表示 $\boldsymbol{e}_s', \boldsymbol{e}_o'$。然后，通过结构评分模块获得四元组结构得分 f_{stru}。

(1)邻居编码器

如图 4 所示，邻居编码器可以划分成 3 部分：1)基于相似性的邻居采样，即从实体邻居集合中选取固定数量的邻居；2)邻居聚合器，即将采样后的邻居按照一定的方式聚合，获得实体 e 的邻居表示 $\boldsymbol{e}_{nei}$；3)融合，即将实体 e 的自身结构表示 $\boldsymbol{e}_e$ 与邻居表示 $\boldsymbol{e}_{nei}$ 融合，得到最终的实体表示 $\boldsymbol{e}_e'$。

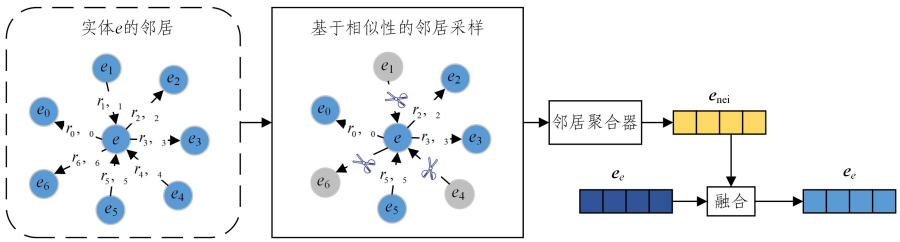


图 4 邻居编码器结构图

Fig. 4 Diagram of neighborhood encoder structure

1)基于相似性的邻居采样

考虑到随机选取的邻居具有较强的随机性，选取的邻居质量无法保证，因此本文提出了一种基于相似性的邻居采样方式。

首先，需要对实体名称进行预处理，去除实体名称中包含的停顿词以及“(”“)”“!”“-”等特殊符号。然后，利用 word2vec[16] 预训练的词向量对实体名称编码，因为实体名称可能由多个单词组成，所以本文将实体名称嵌入 $\boldsymbol{name}_e$ 定义为：

$$\boldsymbol{name}_e=\frac{1}{K}\sum_{i=0}^{K-1}\boldsymbol{word}_i \tag{2}$$

其中，$word_i$ 表示实体名称中第 i 个单词经过 word2vec 预训练后的嵌入，K 表示实体名称中单词的数量。最后，计算实体 e 的名称嵌入与邻居实体名称嵌入的余弦相似度，选取出与实体 e 相似度最高的前 n 个邻居实体，获得最终的实体邻居集合 $N_e=\{(r_0,e_0,\tau_0),(r_2,e_2,\tau_2),\cdots\}$，$|N_e|=n$。

2)邻居聚合器

本文采用两种邻居聚合器聚合实体的邻居信息：基于 LSTM 的邻居聚合器和基于注意力机制的邻居聚合器。

鉴于 LSTM 在处理长序列数据时表现出了良好的性能，本文将 N_e 中的实体按照时间先后顺序输入到 LSTM 中，得到实体 e 的邻居表示 $\boldsymbol{e}_{nei}$：

$$\boldsymbol{e}_{\text{nei}}=\text{LSTM}(\boldsymbol{x}),\boldsymbol{x}=\{\boldsymbol{e}_2,\boldsymbol{e}_0,\cdots\} \tag{3}$$

其中，$\{\boldsymbol{e}_2,\boldsymbol{e}_0,\cdots\}$是按照时间先后顺序排序后的邻居实体嵌入集合，$|\boldsymbol{x}|=n$。

考虑到每个邻居对实体 e 具有不同的重要性，如果不加区分地聚合，容易引入无用信息，导致模型性能下降。因此，本文引入注意力机制加权聚合邻居信息。首先，需要计算每个邻居对实体 e 的重要程度：

$$att_i=\frac{\exp(\varphi(e,r_i,e_i,\tau_i))}{\sum\limits_{(r_j,e_j,\tau_j)\in N_e}\exp(\varphi(e,r_j,e_j,\tau_j))}$$
$$\varphi(e,r_i,e_i,\tau_i)=\text{Re}(\sum_{k=0}^{d_c-1}\boldsymbol{e}_e[k]\boldsymbol{r}_i[k]\boldsymbol{\tau}_i[k]\bar{\boldsymbol{e}}_i[k]) \tag{4}$$

其中，N_e 是实体 e 的邻居集合，$\boldsymbol{e}_e$ 是实体 e 的嵌入表示，$\boldsymbol{e}_e[k]$表示 $\boldsymbol{e}_e$ 的第 k 个元素，$\bar{\boldsymbol{e}}_i$ 是 $\boldsymbol{e}_i$ 的共轭表示，$\text{Re}(\cdot)$表示取实部。

根据上述公式得到每个邻居的权重后，将邻居按权重聚合，最终得到实体邻居表示 $\boldsymbol{e}_{\text{nei}}$：

$$\boldsymbol{e}_{\text{nei}}=\sum_{(r_i,e_i,\tau_i)\in N_e}att_i\boldsymbol{e}_i \tag{5}$$

3）融合

获得实体的邻居表示 $\boldsymbol{e}_{\text{nei}}$后，需要将其与实体自身结构表示 $\boldsymbol{e}_e$ 融合，本文使用了加法融合、乘法融合和门控融合 3 种融合方式。

加法融合：将实体自身结构表示与邻居表示简单相加，如图 5(a)所示。实体最终表示定义为：

$$\boldsymbol{e}_e'=\boldsymbol{e}_e\oplus\boldsymbol{e}_{\text{nei}} \tag{6}$$

乘法融合：将实体自身结构表示与邻居表示相乘，如图 5(b)所示，实体最终表示定义为：

$$\boldsymbol{e}_e'=\boldsymbol{e}_e\otimes\boldsymbol{e}_{\text{nei}} \tag{7}$$

门控融合：上述两种方式平等地对待实体结构信息与邻居信息，容易引入邻居信息中无效的信息，降低模型性能。因此，本文采用门控机制为实体筛选出重要信息，如图 5(c)所示。实体最终表示定义为：

$$\boldsymbol{e}_e'=\gamma\boldsymbol{e}_e\oplus(1-\gamma)\boldsymbol{e}_{\text{nei}} \tag{8}$$

其中，$\gamma\in[0,1]$是门控因子。

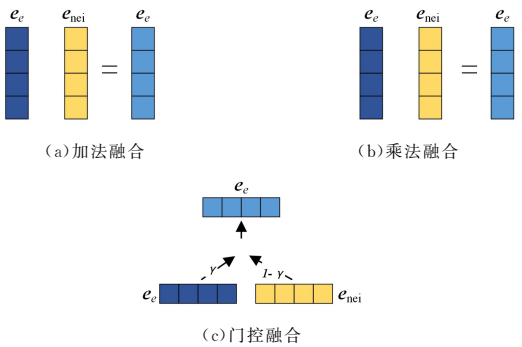


图 5 融合方式

Fig. 5 Fusion methods

(2)结构评分

在利用邻居编码器获得头尾实体的增强表示 $\boldsymbol{e}_s'$ 和 $\boldsymbol{e}_o'$ 后，需使用评分函数来评估四元组(s,r,o,τ)成立的概率。得益于时间信息，四元组中的关系表现出了特有的时间约束，比如关系的重复性、关系间的顺序和关系间的时间间隔。为了捕获这些时间约束，本文使用文献[8]的评分函数对四元组进行评分，具体得分函数的定义如下：

$$\begin{aligned}f_{\text{stru}}(s,r,o,\tau)&=\text{TIMEPLEX}(s,r,o,\tau)\\&=f^{TX}(s,r,o,\tau)+\kappa f^{\text{Pair}}(s,r,o,\tau)+\lambda f^{\text{Rec}}(s,r,o,\tau)\\f^{TX}(s,r,o,\tau)&=\langle\boldsymbol{e}_s',\boldsymbol{r},\bar{\boldsymbol{e}}_o'\rangle+\langle\boldsymbol{e}_s',\boldsymbol{r}^{s\tau},\bar{\boldsymbol{\tau}}\rangle+b\langle\boldsymbol{e}_o',\boldsymbol{r}^{o\tau},\bar{\boldsymbol{\tau}}\rangle+\\&\quad c\langle\boldsymbol{e}_s,\boldsymbol{e}_o',\bar{\boldsymbol{\tau}}\rangle\end{aligned} \tag{9}$$

其中，$f^{TX}(s,r,o,\tau)$是四元组(s,r,o,τ)的得分，$\boldsymbol{r}^{s\tau},\boldsymbol{r}^{o\tau}\in\mathbb{C}^{d_c}$是特定于关系 r 的嵌入表示。$\langle\boldsymbol{x},\boldsymbol{y},\bar{\boldsymbol{z}}\rangle=\text{Re}(\sum\limits_{k=0}^{d_c-1}\boldsymbol{x}[k]\boldsymbol{y}[k]\bar{\boldsymbol{z}}[k])$，$\kappa,\lambda,a,b,c$ 均为超参数。$f^{\text{Rec}}(s,r,o,\tau)$是时间约束(a)的得分，$f^{Pair}(s,r,o,\tau)$是时间约束(b)和时间约束(c)的得分，相关定义可见文献[8]。

3.2.3 四元组关系约束模块

对于四元组(s,r,o,τ)，本模块的目标是结合四元组特定的关系和时间，学习实体在关系约束下的嵌入。本文将四元组(s,r,o,τ)嵌入到实数空间中，获得头实体 s、尾实体 o、关系 r 和时间 τ 在关系约束模块中的初始嵌入 $\boldsymbol{c}_s,\boldsymbol{c}_o,\boldsymbol{c}_r,\boldsymbol{c}_\tau\in\mathbb{R}^d$。考虑到一些实体也受时间影响，因此本文定义四元组关系约束的评分函数为：

$$\begin{aligned}f_{\text{relation}}(s,r,o,\tau)&=\eta\langle\boldsymbol{c}_s,\boldsymbol{c}_r,\boldsymbol{c}_o\rangle+\mu[\boldsymbol{c}_s,\boldsymbol{c}_r,\boldsymbol{c}_o,\boldsymbol{c}_\tau\rangle\\&=\eta(\sum_{k=0}^{d-1}\boldsymbol{c}_s[k]\boldsymbol{c}_r[k]\boldsymbol{c}_o[k])+\mu(\sum_{k=0}^{d-1}\boldsymbol{c}_s[k]\\&\quad\boldsymbol{c}_r[k]\boldsymbol{c}_o[k]\boldsymbol{c}_\tau[k])\end{aligned} \tag{10}$$

其中，η 和 μ 是权重因子，$\boldsymbol{c}_s[k]$表示 $\boldsymbol{c}_s$ 的第 k 个元素。

一般而言，一个关系连接的头(尾)实体应属于同一类别。比如，关系“*presidentOf*”连接的头实体都属于“人”这一类别，而尾实体一般是“国家”。类别相同，显然它们头实体的嵌入也应该彼此接近，尾实体同理。

受此启发，本文在学习实体受关系约束的嵌入的过程中引入了关系特定的实体相似性约束，具体定义为：具有相同关系的四元组中涉及到的头(尾)实体在关系约束下的嵌入应该彼此接近，否则应远离。为了便于后续描述，本文将实体 s 在关系 r 下的受关系约束嵌入表示 $\boldsymbol{c}_{s,r}$ 定义为：

$$\boldsymbol{c}_{s,r}=\boldsymbol{c}_s\cdot\boldsymbol{c}_r \tag{11}$$

关系特定的实体相似性约束示例图如图 6 所示。

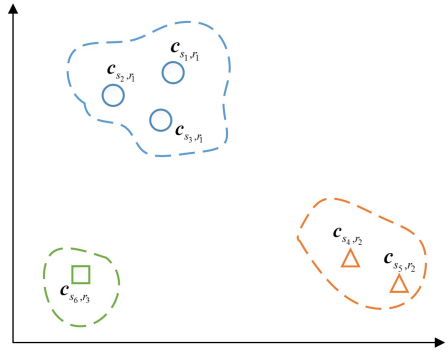


图 6 关系特定的实体相似性约束示例

Fig. 6 Example of relation-specific entity similarity constraints

给定四元组$(s_1,r_1,o_1,\tau_1)(s_2,r_1,o_2,\tau_2)(s_3,r_1,o_3,\tau_3)(s_4,r_2,o_4,\tau_4)(s_5,r_2,o_5,\tau_5)$和$(s_6,r_3,o_6,\tau_6)$，对于四元组$(s_1,r_1,o_1,\tau_1)(s_2,r_1,o_2,\tau_2)$和$(s_3,r_1,o_3,\tau_3)$，头实体 s_1,s_2 和 s_3 在

关系 r_1 下的受关系约束嵌入 $\boldsymbol{c}_{s_1,r_1}$，$\boldsymbol{c}_{s_2,r_1}$ 和 $\boldsymbol{c}_{s_3,r_1}$ 应该彼此接近；对于 (s_1,r_1,o_1,τ_1) 和 (s_4,r_2,o_4,τ_4)，头实体 s_1 在关系 r_1 下的受关系约束嵌入 $\boldsymbol{c}_{s_1,r_1}$ 与 s_4 在关系 r_2 下的受关系约束嵌入 $\boldsymbol{c}_{s_4,r_2}$ 应彼此远离。同理，尾实体在关系约束下的嵌入也应该满足上述要求。

为了评估在特定关系约束下，实体嵌入间的相似性，对于任意两个四元组 (s_i,r_i,o_i,τ_i) 和 (s_j,r_j,o_j,τ_j)，本文定义实体嵌入间的相似性得分为：

$$f_{\text{sim}}((s_i,r_i,o_i,\tau_i),(s_j,r_j,o_j,\tau_j))=\frac{1}{2}(\|\boldsymbol{c}_{s_i,r_i}-\boldsymbol{c}_{s_j,r_j}\|+\|\boldsymbol{c}_{o_i,r_i}-\boldsymbol{c}_{o_j,r_j}\|) \tag{12}$$

当 $r_i=r_j$ 时，f_{sim} 应较小，反之，则应较大。

3.2.4 模型最终评分

本文将四元组结构模块的得分 $f_{\text{stru}}(s,r,o,\tau)$ 与四元组关系约束模块的得分 $f_{\text{relation}}(s,r,o,\tau)$ 按照一定的权重相加作为四元组 (s,r,o,τ) 的最终得分，具体定义如下：

$$f_{\text{final}}(s,r,o,\tau)=\alpha f_{\text{stru}}(s,r,o,\tau)+\beta f_{\text{relation}}(s,r,o,\tau) \tag{13}$$

其中，α 和 β 是权重因子。

3.3 模型优化

在 CARC 模型的优化过程中，本文使用如下损失函数来训练模型：

$$Loss=\sum_{(s,r,o,\tau)\in G}(Loss_1+\alpha_{\text{con}}Loss_2) \tag{14}$$

其中，$Loss_1$ 是四元组结构模块的损失，$Loss_2$ 是四元组关系约束模块中关系特定的实体相似性约束的损失，α_{con} 是权重因子，G 是正确四元组的集合。$Loss_1$ 和 $Loss_2$ 的详细定义为：

$$Loss_1=-(\log\Pr(o|s,r,\tau)+\log\Pr(s|o,r,\tau)+\log\Pr(\tau|s,r,o)) \tag{15}$$

$$Loss_2=-\sum_{(sp,r,op,\tau)\in Y}\sum_{(sn,r',on,\tau)\in Y'}\max[0,f_{\text{sim}}((s,r,o,\tau),(sp,r,op,\tau))+\gamma_{\text{con}}-f_{\text{sim}}((s,r,o,\tau),(sn,r',on,\tau))] \tag{16}$$

式(15)中，$\Pr(o|s,r,\tau)$ 是尾实体预测时实体 o 的概率，计算式如式(17)所示，同理可知 $\Pr(s|o,r,\tau)$ 和 $\Pr(\tau|s,r,o)$。式(16)中，$Y=\{(\tilde{s},\tilde{r},\tilde{o},\tilde{\tau})\in G|\tilde{r}=r,(\tilde{s},\tilde{r},\tilde{o},\tilde{\tau})\neq(s,r,o,\tau)\}$ 是除了 (s,r,o,τ) 外，关系也为 r 的其他四元组的集合；$Y'=\{(\tilde{s},\tilde{r},\tilde{o},\tilde{\tau})\in G|\tilde{r}\neq r\}$ 是关系不为 r 的四元组集合；$\max[0,x]$ 表示选择 0 和 x 之间较大的值；γ_{con} 是固定边距。

$$\Pr(o|s,r,\tau)=\frac{\exp(f_{\text{final}}(s,r,o,\tau))}{\sum_{(s,r,o',\tau)\in G^-}\exp(f_{\text{final}}(s,r,o',\tau))} \tag{17}$$

其中，$G^-=\{(s,r,o',\tau)\notin G\}$ 是经过随机替换尾实体和过滤操作的四元组集合。

4 实验

4.1 数据集

本文在 4 个常用的时间数据集上进行实验。García-Durán 等[13]生成的 ICEWS14 和 ICEWS05-15 数据集中，每个事实的时间戳都是一个完整的时间点，四元组表现形式形如(Obama, visit, Japan, 2014-03-12)；Dasgupta 等[11]生成的 YAGO11k 和 Wikidata12k 主要包含具有时间间隔的事件，四元组表现形式形如(David Beckham, plays for, Manchester United, [1992-##-##, 2003-##-##])。本文使用的 4 个数据集都不包含类型信息，各个数据集的统计信息如表 1 所列。

表 1 实验数据集

Table 1 Experimental datasets

数据集	实体	关系	时间戳	训练集	验证集	#测试集
ICEWS14	7128	230	365	72826	8941	8963
ICEWS05-15	10488	251	4017	386962	46275	46092
YAGO11k	10623	10	251	16406	2050	2051
Wikidata12k	12554	24	237	32497	4062	4062

4.2 实验设置

4.2.1 实验参数

本文采用自适应低阶矩估计优化器(Adam)优化算法，并利用网格搜索策略为模型寻找最优的超参数。经过多次实验，在 ICEWS14 上的最优参数组合为：$d=d_c=200$，$\kappa=0$，$\lambda=0$，$a=b=c=5.0$，$\eta=2.0$，$\mu=2.0$，$\gamma=0.7$，$\alpha=1$，$\beta=1$，$\alpha_{\text{con}}=0.1$，$\gamma_{\text{con}}=1.0$，$n=10$，$lr=0.1$，$epoch=150$，$thre=1$。ICEWS05-15 上的最佳参数组合为：$d=200$，$d_c=400$，$\kappa=0$，$\lambda=5.0$，$a=b=c=5.0$，$\eta=5.0$，$\mu=2.0$，$\gamma=0.7$，$\alpha=1$，$\beta=1$，$\alpha_{\text{con}}=0.1$，$\gamma_{\text{con}}=2.0$，$n=10$，$lr=0.1$，$epoch=250$，$thre=1$。在 YAGO11k 上的最优参数组合为：$d=d_c=200$，$\kappa=3.0$，$\lambda=5.0$，$a=b=5.0$，$c=0$，$\eta=0.1$，$\mu=0$，$\gamma=0.9$，$\alpha=1$，$\beta=0.9$，$\alpha_{\text{con}}=0.4$，$\gamma_{\text{con}}=2.0$，$n=10$，$lr=0.1$，$epoch=100$，$thre=300$。在 Wikidata12k 上的最优参数组合为：$d=d_c=200$，$\kappa=0$，$\lambda=5.0$，$a=b=c=5.0$，$\eta=2.0$，$\mu=0.2$，$\gamma=0.9$，$\alpha=1$，$\beta=1$，$\alpha_{\text{con}}=0.2$，$\gamma_{\text{con}}=1.0$，$n=10$，$lr=0.1$，$epoch=100$，$thre=400$。

4.2.2 过滤与预测策略

本文采用文献[8]中的过滤策略：对于每一幅四元组 $(s,r,o,T=[\tau_b,\tau_e])$，首先枚举每个时间戳 $\tau\in[\tau_b,\tau_e]$，从而生成枚举四元组集合 $Q=\{(s,r,o,\tau_b),(s,r,o,\tau_{b+1}),\cdots,(s,r,o,\tau_e)\}$，然后通过随机替换头尾实体和随机替换时间戳生成新的四元组集合 Q'，并过滤集合 Q' 中已存在于训练集、验证集和测试集中的四元组。时间戳为时间点的四元组 (s,r,o,τ) 可以看作是时间区间的一个特例 $(s,r,o,T=[\tau,\tau])$。因此，(s,r,o,τ) 类型的四元组的过滤方式与上述内容相同。

在链路预测任务中，本文采用文献[8]的排名策略：对于四元组 $(s,r,o,T=[\tau_b,\tau_e])$，计算四元组集合 Q 中每个四元组的排名，之后将 Q 中每个四元组的排名求和取平均作为四元组 $(s,r,o,T=[\tau_b,\tau_e])$ 的最终排名，即：

$$rank(s,r,o,T=[\tau_b,\tau_e])=\frac{\sum_{(s,r,o,\tau')\in Q}rank(s,r,o,\tau')}{|Q|} \tag{18}$$

在时间区间预测任务中，与文献[8]一样，我们首先计算 $(s,r,o,?)$ 在所有时间戳上的概率分布 $\Pr(\tau|s,r,o)$，以最高概率的时间戳为起点，不断向左或向右扩展(若左边时间戳概率较高，则向左边扩展，反之亦然)，直至区间长度达到 θ_r(模型训练阶段为每个关系学习的阈值)，从而获得预测的时间区间 $\tau^{\text{pre}}=[\tau_b^{\text{pre}},\tau_e^{\text{pre}}]$。

4.2.3 基线

本文模型对比的基线可以划分为两大类：基于静态知识图谱的知识表示学习模型和基于时间知识图谱的知识表示学习模型。

基于静态知识图谱的知识表示学习模型包括：基于复数空间的 ComplEx[9]、能够自动编码实体类型的 AutoETER[6]、融合实体邻居信息的 CompGCN[2]。

基于时间知识图谱的知识表示学习模型包括：时间感知

的 ComplEx 模型 TA-ComplEx[8]、基于时间超平面的 HyTE[11]、时间融合框架的代表模型 DE-SimplE[14]、基于时间旋转的 TeRo[1]、基于四阶张量分解的 TNT-ComplEx[7]、利用高斯分布建模 3 种时间约束的 TIMEPLEX[8]。

4.2.4 评估指标

本文使用链路预测[16]和时间区间预测[8]来评估模型的性能。

(1)链路预测

链路预测任务可以拆分为头实体预测和尾实体预测两个子任务。其中头实体预测即给定关系、尾实体和时间戳，预测头实体表示为$(?,r,o,\tau)$。同理，尾实体预测可以表示为$(s,r,?,\tau)$。本文使用 MRR，Hits@1，Hits@10 等指标来评估链路预测任务。其中，MRR 是正确四元组的平均倒数排名，Hits@N 是前 N 个候选四元组中正确四元组的比例。MRR 和 Hits@N 越高，说明模型的补全性能越好。

(2)时间区间预测

时间预测任务即给定头实体、关系和尾实体，预测三元组发生的时间区间，表示为$(s,r,o,?)$。本文将模型预测的时间区间表示为 $\tau^{\text{pre}}=[\tau_b^{\text{pre}},\tau_e^{\text{pre}}]$，实际的时间区间表示为 $\tau^{\text{gold}}=[\tau_b^{\text{gold}},\tau_e^{\text{gold}}]$，使用 Prachi 等[8]使用的 TAC，IOU，gIOU 和 aeIOU 等指标评估时间区间的预测能力，各指标的定义如下：

$$gIOU(\tau^{\text{gold}},\tau^{\text{pre}})=IOU(\tau^{\text{gold}},\tau^{\text{pre}})-\frac{vol((\tau^{\text{gold}}\overset{\triangle}{\cup}\tau^{\text{pre}})(\tau^{\text{gold}}\cup\tau^{\text{pre}}))}{vol(\tau^{\text{gold}}\overset{\triangle}{\cup}\tau^{\text{gold}})}\in(-1,1] \tag{19}$$

$$IOU(\tau^{\text{gold}},\tau^{\text{pre}})=\frac{vol(\tau^{\text{gold}}\cap\tau^{\text{pre}})}{vol(\tau^{\text{gold}}\cup\tau^{\text{pre}})}\in(0,1] \tag{20}$$

$$gIOU(\tau^{\text{gold}},\tau^{\text{pre}})=IOU(\tau^{\text{gold}},\tau^{\text{pre}})-\frac{vol((\tau^{\text{gold}}\overset{\triangle}{\cup}\tau^{\text{pre}})(\tau^{\text{gold}}\cup\tau^{\text{pre}}))}{vol(\tau^{\text{gold}}\overset{\triangle}{\cup}\tau^{\text{pre}})}\in(-1,1] \tag{21}$$

$$aeIOU(\tau^{\text{gold}},\tau^{\text{pre}})=\frac{\max\{1,vol(\tau^{\text{gold}}\cap\tau^{\text{pre}})\}}{vol(\tau^{\text{gold}}\overset{\triangle}{\cup}\tau^{\text{pre}})} \tag{22}$$

其中，$vol(\cdot)$是区间的大小；$\tau^{\text{gold}}\overset{\triangle}{\cup}\tau^{\text{pre}}$是包含了 τ^{gold}和 τ^{pre}的最小单一连续区间，如$[1,2]\overset{\triangle}{\cup}[5,20]=[1,20]$。TAC，IOU，gIOU 和 aeIOU 越高，说明模型在时间区间预测任务上的表现越好。

4.3 链路预测

链路预测结果如表 2 所列。其中，* 表示将源码应用于新的数据集得到的结果，♣表示结果取自文献[1]，本文模型 CARC 的结果采用 4.2.2 节所述的过滤方式得到，其余结果取自文献[8][1)]。如 3.2.2 节所述，本文采用基于 LSTM 或基于注意力的邻居聚合器，LSTM 表示基于 LSTM 的邻居聚合器，Atten 表示基于注意力机制的邻居聚合器；采用 3 种特征融合方式，其中 Add 表示加法融合方式，Mul 表示乘法融合方式，Gate 表示门控融合方式。CARC(LSTM+Gate)表示使用基于 LSTM 的邻居聚合器和门控融合方式的模型，同理可知其余组合方式。表 2 中粗体表示最优结果，下划线表示次优结果，文中实验结果取 5 次实验的平均值。

表 2 各个数据集的链路预测结果

Table 2 Link prediction results of each dataset

(单位：%)

模型	ICEWS14			ICEWS05-15			YAGO11k			Wikidata12k		
	MRR	*Hits*@1	*Hits*@10	*MRR*	*Hits*@1	*Hits*@10	*MRR*	*Hits*@1	*Hits*@10	*MRR*	*Hits*@1	*Hits*@10
ComplEx	45.15	33.95	67.22	48.25	36.65	70.84	18.14	11.46	29.96	24.82	14.30	47.46
AutoETER *	38.89	23.30	69.55	31.56	11.05	68.47	14.82	9.53	25.79	22.08	12.14	44.21
CompGCN *	46.91	35.69	69.00	47.59	36.26	69.39	15.39	9.46	26.91	24.50	14.76	45.26
TA-ComplEx	40.97	29.58	—	49.23	37.6	—	15.24	9.36	—	22.78	12.69	—
HyTE	24.91	2.98	63.56	23.73	3.11	62.76	13.55	3.32	29.01	25.28	14.70	46.57
DE-SimplE	52.60	41.80	72.50	51.30	39.20	74.80	15.12	8.75	25.69	25.29	14.68	47.34
TeRo♣	56.20	46.80	75.00	58.60	46.90	79.50	14.67	9.46	25.09	26.69	16.16	49.06
TNT-ComplEx	56.72	47.04	74.47	61.21	51.17	79.91	18.01	11.02	30.15	30.10	19.73	49.56
TIMEPLEX	60.40	51.50	76.46	63.99	54.51	80.77	23.64	16.92	36.05	33.35	22.78	52.08
CARC(LSTM+Add)	59.79	50.71	76.01	65.65	56.27	82.27	21.05	14.51	32.52	33.31	24.04	48.47
CARC(LSTM+Mul)	60.60	51.80	76.26	65.53	56.50	81.79	**23.90**	16.63	**36.98**	34.22	23.92	52.20
CARC(LSTM+Gate)	**61.23**	**52.48**	**77.15**	**66.49**	**57.29**	**82.94**	23.76	**17.28**	35.30	**35.20**	**24.32**	**54.65**
CARC(Atten+Add)	60.69	51.79	76.78	65.49	55.99	82.39	23.45	16.50	35.84	35.08	23.98	54.62
CARC(Atten+Mul)	59.07	50.13	75.32	62.78	53.46	79.67	17.46	10.48	30.33	32.49	22.21	50.92
CARC(Atten+Gate)	60.56	51.58	76.77	65.90	56.60	82.58	23.54	16.70	35.88	34.96	24.15	54.19

经过分析，可以得出以下结论：

(1)CARC 模型总体上优于其他基线，有效证明了本文模型在链路预测上的优势。具体地，在 ICEWS05-15 数据集上，相比 TIMEPLEX，本文最优模型 CARC(LSTM+Gate)在 MRR 上提升了 2.5%，在 Hits@1 上提升了 2.78%，在 Hits@10 上提升了 2.17%。这得益于 CARC 使用了实体的邻居信息和关系约束，有效地增强了实体的嵌入表示，有助于模型更精准地预测实体。相比之下，TIMEPLEX 仅依靠实体自身的结构信息，实体嵌入中包含的信息不够丰富，一定程度上影响了模型性能。

(2)本文模型在各个数据集上的实验结果都显著优于融合了邻居信息的 CompGCN 模型和利用了实体类型相似性信息的 AutoETER 模型。这是因为，本文模型同时使用了实体的邻居信息和关系约束下实体类型的相似性，两者相辅相成，

1) https://github.com/dair-iitd/tkbi

在一定程度上改善了实体嵌入表示，提高了模型预测精度。此外，CompGCN 和 AutoETER 没有利用时间信息，导致它们在时间知识图谱的补全任务上性能不佳。

(3)对比 CARC(LSTM+)模型与 CARC(Atten+)模型，可以发现使用 LSTM 作为邻居聚合器的实验结果大多比使用注意力机制的更好，这是因为 LSTM 在处理长序列数据时性能较好。对于时间距离较长的邻居信息，LSTM 中的门控机制能够对其进行有效的过滤，避免为模型引入噪声。

(4)与 Add 和 Mul 相比，Gate 融合方式的实验效果相对较好。这是因为 Add 和 Mul 只是简单地将邻居嵌入与实体结构嵌入相加或者相乘，弱化了实体结构嵌入在预测中的重要性。实体的邻居信息虽然也携带对补全任务有用的信息，但它只是附加信息，用于丰富实体嵌入，实际上四元组自身的结构信息更重要。

4.4 时间区间预测

YAGO11k 和 Wikidata12k 数据上各模型的时间区间预测结果如表 3 所列。从表中可以发现，本文模型总体上优于基线模型，在 aeIOU 这个更有说服力的指标上[8]，本文最优模型在 YAGO11k 和 Wikidata12k 上相比 TIMEPLEX 分别提升了 1.35%和 1.05%。这进一步说明了邻居信息和类型信息的有效性，对它们的合理利用能够显著提升知识补全的能力。

表 3 YAGO11k 和 Wikidata12k 数据集上各模型的时间区间预测结果

Table 3 Time interval prediction results for each model on YAGO11k and Wikidata12k datasets

(单位：%)

模型	YAGO11k				Wikidata12k			
	TAC	*gIOU*	*IOU*	*aeIOU*	*gIOU*	*IOU*	*aeIOU*	
HyTE	5.59	15.96	1.91	5.41	6.13	14.55	1.40	5.41
TNT-ComplEx	9.90	20.78	3.99	8.40	26.98	36.63	11.68	23.25
TIMEPLEX	22.66	32.64	8.24	20.03	30.71	39.34	13.15	26.36
CARC(LSTM+Add)	22.71	33.05	8.67	20.09	22.19	32.25	8.70	18.79
CARC(LSTM+Mul)	23.10	34.00	9.08	20.44	29.28	36.44	10.25	25.56
CARC(LSTM+Gate)	**23.72**	**34.15**	**9.96**	**21.38**	30.63	39.64	13.21	26.68
CARC(Atten+Add)	23.32	33.66	8.69	20.63	30.93	39.97	13.59	26.93
CARC(Atten+Mul)	22.88	33.82	9.65	21.05	29.56	36.64	10.34	25.75
CARC(Atten+Gate)	22.84	33.18	8.32	20.07	**31.29**	**40.02**	**13.64**	**27.41**

4.5 复杂关系类型实验

本文在 ICEWS14 和 ICEWS05-15 两个数据集上进行不同关系类型的链路预测实验，以进一步分析模型性能。首先根据文献[12]中的划分策略，本文将 ICEWS14 和 ICEWS05-15 中的测试集划分为一对一(1-to-1)、一对多(1-to-*N*)、多对一(*N*-to-1)、多对多(*N*-to-*N*)4 种关系类型。两个数据集在 4 种关系类型下的四元组数量占比情况如图 7 所示，可以看到在这两个数据集中 *N*-to-*N* 这种关系类型的占比最大。

在 ICEWS14 和 ICEWS05-15 上不同关系类型链路预测结果如表 4 所列。总体上，CARC(LSTM+Gate)优于其他模型，尤其在 *N*-to-*N* 这种较为复杂且数据量较大的关系类型上的链路预测结果都优于其他基线模型。与次优的 TIMEPLEX 相比，本文模型在 ICEWS14 和 ICEWS05-15 数据集 *N*-to-*N* 关系类型链路预测任务上，MRR 提升了 0.8%和 2.93%，Hits@1 提升了 0.83%和 3.28%，Hits@10 提升了 0.58%和 2.29%。值得注意的是，ICEWS14 和 ICEWS05-15 在 1-to-*N*/*N*-to-1 关系类型上链路预测的效果不佳，但是从图 7 所示的统计结果可知，1-to-*N*/*N*-to-1 关系类型下的四元组数量占比很小，尽管效果不如其他模型，但对模型总体效果的影响并不大。

表 4 ICEWS14 和 ICEWS05-15 不同关系类型链路预测结果

Table 4 Link prediction results of different relation types on ICEWS14 and ICEWS05-15

(单位：%)

关系类型	模型	ICEWS14			ICEWS05-15		
		MRR	*Hits*@1	*Hits*@10	*MRR*	*Hits*@1	*Hits*@10
1-to-1	DE-SimplE	34.40	22.20	42.50	41.40	29.80	65.30
	TNT-ComplEx	37.20	25.94	59.43	35.64	25.00	56.45
	TIMEPLEX	45.97	35.85	65.09	43.84	33.87	63.71
	CARC(LSTM+Gate)	**46.45**	**36.32**	**69.34**	**45.79**	**34.68**	**66.13**
1-to-*N*	DE-SimplE	40.20	32.10	58.30	41.40	28.90	**69.10**
	TNT-ComplEx	39.61	30.95	60.71	41.69	31.96	61.34
	TIMEPLEX	**46.64**	**39.29**	**66.67**	43.69	33.51	63.40
	CARC(LSTM+Gate)	44.67	35.71	63.10	**50.70**	**40.72**	67.53
N-to-1	DE-SimplE	46.20	35.00	61.70	34.60	27.30	50.00
	TNT-ComplEx	52.00	40.00	**71.67**	23.80	18.18	31.82
	TIMEPLEX	47.99	38.33	63.33	**40.09**	**36.36**	**63.64**
	CARC(LSTM+Gate)	**54.67**	**46.67**	**71.67**	39.06	31.82	59.09
N-to-*N*	DE-SimplE	52.70	41.70	73.20	51.10	40.00	74.70
	TNT-ComplEx	56.71	47.36	74.19	60.30	51.09	77.47
	TIMEPLEX	60.71	51.94	76.75	63.62	54.08	80.71
	CARC(LSTM+Gate)	**61.51**	**52.77**	**77.33**	**66.55**	**57.36**	**83.00**

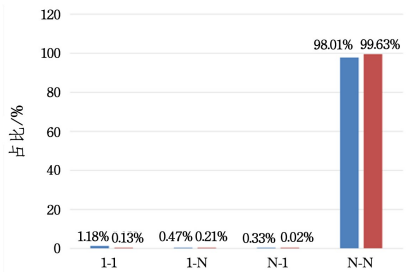


图 7 ICEWS14 和 ICEWS05-15 中不同关系类型所占的比例

Fig. 7 Proportion of different relation types in ICEWS14 and ICEWS05-15

4.6 消融实验

为了证明 CARC 中各个模块的重要性，本文在 ICEWS14 和 ICEWS05-15 上进行了消融实验。本文将去除了邻居编码器的模型记为 CARC(-Neighbor)，将去除了四元组关系约束模块的模型记为 CARC(-Relation)，将去除了四元组关系约束模块中有关时间的得分(即式(9)中的 $\mu=0$)的模型记为 CARC(-Time)。

实验结果如表 5 所列，可以看到移除邻居信息、关系约束以及关系约束模块中有关时间的得分都会一定程度地造成模型性能下降，这说明它们都是模型至关重要的一部分。从实验结果可以发现，关系约束给模型性能带来的影响较为显著，这是因为关系对实体的相似性约束可以在模型预测时起到一定的作用，能够帮助模型实现更精准的预测，从而提高模型性能。去除了关系约束模块中有关时间的得分也会降低模型的精度，这验证了本文提出的“实体受时间信息的影响”的猜想。实体邻居信息为模型带来的影响虽然稍逊于关系约束，但也提升了模型的预测精度，说明实体的邻域信息对模型也有一定帮助。

表 5 消融实验结果

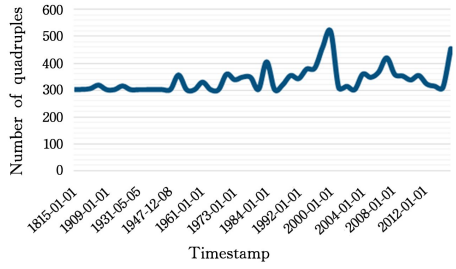
Table 5 Abalation experiment results

(单位：%)

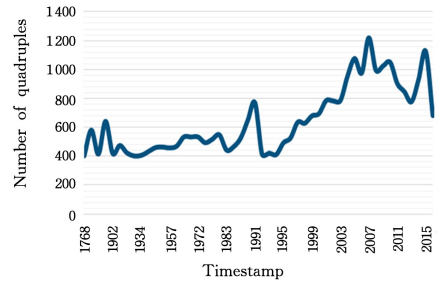
模型	ICEWS14			ICEWS05-15		
	MRR	*Hits*@1	*Hits*@10	*MRR*	*Hits*@1	*Hits*@10
CARC (LSTM+Gate)	**61.23**	**52.48**	**77.15**	**66.49**	**57.29**	**82.94**
CARC (-Neighbor)	60.89	52.08	76.74	66.06	57.07	82.28
CARC (-Relation)	60.59	51.40	76.85	65.64	56.35	82.42
CARC(-Time)	61.07	52.15	76.99	65.98	56.75	82.53

4.7 自适应时间粒度的有效性实验

在自适应时间粒度聚合模块中，本文在 YAGO11k 数据集中采用了时间戳中完整的年月日信息。对于 Wikidata12k，我们仅采用了时间戳中的年份信息，因为 Wikidata12k 的所有时间戳都只包含年份信息，额外填充月和日信息容易为模型带来噪声。YAGO11k 与 Wikidata12k 经过自适应时间粒度聚合模块处理后，在时间上的数据分布如图 8 所示。从图中可以看出，数据在各个时间戳下的分布相对比较均衡，有利于训练出一个性能相对平衡的模型。此外，通过这种方式，也可以在一定程度上缓解时间戳缺失造成的问题。



(a) YAGO11k



(b) Wikidata12k

图 8 YAGO11k 和 Wikidata12k 时间戳合并后的数据分布

Fig. 8 Data distribution of YAGO11k and Wikidata12k after timestamp merging

为了进一步证明本文提出的自适应时间粒度聚合模块的有效性，本文选取 CARC(LSTM+Gate)模型在 YAGO11k 和 Wikidata12k 上进行进一步的实验。本文将不使用自适应时间粒度模块的模型记为 CARC-TG(LSTM+Gate)，将自适应时间粒度模块中不考虑时间戳中的月和日信息的模型记为 CARC-MD(LSTM+Gate)(不在 Wikidata12k 上做该实验，因为该数据集时间戳中只有年份信息)。实验结果如表 6 所列，从结果中可以看出，本文提出的自适应时间粒度聚合模块为模型性能带来了显著的提升。使用了时间戳中完整年月日信息的模型性能比只使用年份信息的模型更好，这说明时间信息越准确，为模型带来的效果就越好，模型能够以更细粒度的方式合并时间戳，合并方式更合理。值得注意的是，在 YAGO11k 的实验结果中，CARC-TG(LSTM+Gate)的实验结果优于 CARC-MD(LSTM+Gate)，这是因为 CARC-MD(LSTM+Gate)仅使用时间戳中的年份信息，容易导致时间戳的不合理合并，从而导致模型性能明显下降。

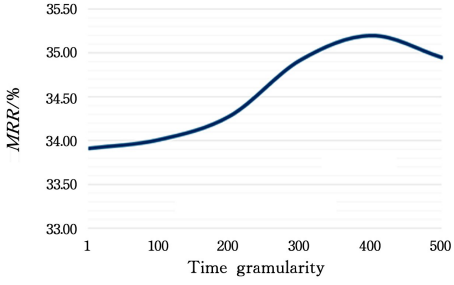
表 6 自适应时间粒度的实验结果

Table 6 Adaptive time granularity experimental results

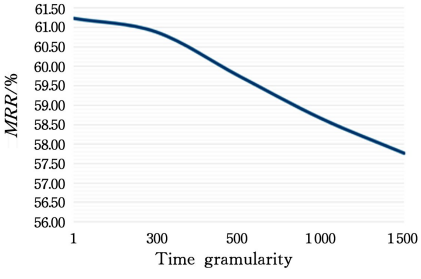
(单位：%)

模型	YAGO11k			Wikidata12k		
	MRR	*Hits*@1	*Hits*@10	*MRR*	*Hits*@1	*Hits*@10
CARC (LSTM+Gate)	**23.76**	**17.28**	**21.38**	**35.20**	**24.32**	**26.68**
CARC-TG (LSTM+Gate)	23.48	16.92	20.40	33.92	22.67	24.70
CARC-MD (LSTM+Gate)	23.31	16.46	20.21	—	—	—

此外，本文还分析了时间粒度 *thre* 的变化对模型性能的影响，如图 9 所示。从图 9(a)中可以看出，在 Wikidata12k 这个数据分布不均衡的数据集上，时间粒度设置太大或者太小对模型性能都有一定的影响，这是因为太小的时间粒度并不能很好地解决数据不均衡分布带来的问题，而太大的时间粒度会造成过多的时间信息合并为一个，时间信息没有完全表达，从而导致模型性能降低。从图 9(b)中可以看到，在 ICEWS14 数据集上，模型性能随着时间粒度的增大而减小，这是因为在 ICEWS14 上，数据在时间上的分布相对均匀，如图 10 所示，使用较小的时间粒度可以为模型提供更丰富和准确的时间信息。



(a) Wikidata12k



(b) ICEWS14

图 9 不同时间粒度 *thre* 下 CARC 模型在各数据集上的 MRR 结果

Fig. 9 MRR of CARC model on each dataset under different time granularity *thre*

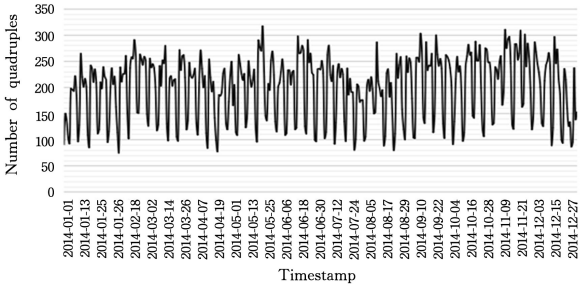


图 10 ICEWS14 在各个时间戳上的数据分布

Fig. 10 Data distribution of ICEWS14 on each timestamp

4.8 采样方式对比实验

为了验证本文采样方式的有效性，本文将 CARC(LSTM+Gate)中基于相似性的邻居采样方式替换为随机采样方式，记为 CARC-Random(LSTM+Gate)。本文在时间上数据分布均衡的 ICEWS05-15 数据集以及在时间上数据分布不均衡的 YAGO11k 数据集上进行了该项实验，实验结果如表 7 所列。

表 7 不同采样方式的结果

Table 7 Results of different sampling methods

(单位：%)

模型	ICEWS05-15			YAGO11k		
	MRR	*Hits*@1	*Hits*@10	*MRR*	*Hits*@1	*Hits*@10
CARC (LSTM+Gate)	**66.49**	**57.29**	**82.94**	**23.76**	**17.28**	**35.30**
CARC-Random (LSTM+Gate)	66.24	57.04	82.78	23.34	16.38	35.11

结果表明，本文提出的基于相似性的邻居采样方式效果更好，主要原因在于它可以在邻居采样过程中选取出与当前实体较为相关的邻居信息，避免为模型引入噪声。而随机采样的方式存在一定的随机性，有可能选取到与实体当前预测任务不相关的邻居，从而降低模型性能。

结束语 本文提出了一个基于关系约束的上下文感知模型 CARC。CARC 中的自适应时间粒度聚合模块通过设置时间粒度，能够解决数据集在时间上分布不均衡的问题，从而训练出一个性能良好的模型；四元组结构模块通过邻居编码器为实体聚合上下文信息，从而增强实体的嵌入表示；四元组关系约束模块在关系约束下学习实体的嵌入表示，在知识补全任务中发挥了关键作用。在链路预测和时间区间预测上的大量实验表明，本文模型在时态知识图谱补全任务上具有较大的优势。在未来的工作中，我们将进一步改进邻居编码器，从实体邻居中捕获更为关键有效的信息。此外，我们也考虑将关系上下文纳入模型中，以便更好地利用图上下文信息。

参 考 文 献

[1] XU C, NAYYERI M, ALKHOURY F, et al. TeRo: A Time-aware Knowledge Graph Embedding via Temporal Rotation [C]// Proceedings of the 28th International Conference on Computational Linguistics. Barcelona: International Committee on Computational Linguistics, 2020: 1583-1593.

[2] VASHISHTH S, SANYAL S, NITIN V, et al. Composition-based Multi-Relational Graph Convolutional Networks[C]// 8th International Conference on Learning Representations. Addis Ababa: OpenReview. net, 2020: 1-16.

[3] HONG J D, CHEN W, ZHAO L. Knowledge Representation Model That Combining Entity Neighbor Information[J]. Journal of Chinese Computer Systems, 2020, 41(8): 1596-1601.

[4] XIE R, LIU Z, SUN M. Representation Learning of Knowledge Graphs with Hierarchical Types[C]// Proceedings of the 25th International Joint Conference on Artificial Intelligence. New York: Morgan Kaufmann, 2016: 2965-2971.

[5] MA S, DING J, JIA W, et al. Transt: Type-based multiple embedding representations for knowledge graph completion[C]// Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Skopje: Cham: Springer, 2017: 717-733.

[6] NIU G, LI B, ZHANG Y, et al. AutoETER: Automated Entity Type Representation for Knowledge Graph Embedding[C]//

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. Punta Cana: ACL, 2020: 1172-1181.

[7] LACROIX T, OBOZINSKI G, USUNIER N. Tensor Decompositions for temporal knowledge base completion[C]// 8th International Conference on Learning Representations. Addis Ababa: OpenReview. net, 2020: 1-12.

[8] JAIN P, RATHI S, CHAKRABARTI S. Temporal Knowledge Base Completion: New Algorithms and Evaluation Protocols [C]// Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. Punta Cana: ACL, 2020: 3733-3747.

[9] TROUILLON T, WELBL J, RIEDEL S, et al. Complex embeddings for simple link prediction[C]// Proceedings of International Conference on Machine Learning. Sydney: PMLR Press, 2016: 2071-2080.

[10] MARCHEGGIANI D, TITOV I. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling[C]// Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Brussels: ACL, 2017: 1506-1515.

[11] DASGUPTA S S, RAY S N, TALUKDAR P. Hyte: Hyperplane-based temporally aware knowledge graph embedding [C]// Proceedings of the 2018 Conference on Empirical Methods in Matural Language Processing. Brussels: ACL, 2018: 2001-2011.

[12] BORDES A, USUNIER N, GARCIA-DURAN A, et al. Translating embeddings for modeling multi-relational data [C]// Advances in Neural Information Processing Systems 26. Cambridge: The MIT Press, 2013: 2787-2795.

[13] GARCÍA-DURÁN A, DUMANČIĆ S, NIEPERT M. Learning sequence encoders for temporal knowledge graph completion [C]// Proceedings of the 2018 Conference on Empirical Methods in Matural Language Processing. Brussels: ACL, 2018.

[14] GOEL R, KAZEMI S M, BRUBAKER M, et al. Diachronic embedding for temporal knowledge graph completion[C]// Proceedings of the 34th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2020: 3988-3995.

[15] SUN Z, DENG Z, NIE J, et al. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space[C]// In 8th International Conference on Learning Representations. New Orleans: OpenReview. net, 2019: 1-18.

[16] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality [C]// Advances in Neural Information Processing Systems. Cambridge: The MIT Press, 2013: 3111-3119.

WANG Jingbin, born in 1973, master, associate professor, is a member of China Computer Federation. Her main research interests include knowledge graph, relation reasoning, distributed data management and knowledge repesen-tation.

（责任编辑：何杨）