

Logic and Commonsense-Guided Temporal Knowledge Graph Completion

魏仕龙

摘要

时间知识图谱 (TKG) 存储涉及时间的数据生成的事件。预测事件非常具有挑战性, 因为事件具有时间敏感性质。此外, 先前的时间知识图谱补全 (TKGC) 方法无法同时表示事件的及时性和因果性特征。为了解决这些挑战, 论文中提出了一个逻辑和常识引导的嵌入模型 (LCGE), 以共同学习涉及事件的时间敏感表示, 同时从常识的角度学习事件的时间独立表示。具体而言, 论文中设计了一个时间规则学习算法来构建规则引导的谓词嵌入正则化策略, 用于学习事件之间的因果关系。此外, 论文中通过辅助常识知识准确评估事件的合理性。TKGC任务的实验结果显示, 与现有方法相比, 该模型显著提升了性能, 并且模型能够从因果推断的角度提供对预测结果的解释性。

关键词: 时间知识图谱; 规则学习; 知识图谱嵌入

1 引言

知识图谱之前主要以 (主语, 谓语, 宾语) 的形式存储事实; 近些年来, 新的技术中提出了时间知识图谱 (TKG) 存储了与时间相关的数据中提取的事件。将时间作为一个新的维度加入存储之中, 但是有时候却没有完全的发挥出多储存的作用, 先前的 TKG 方法不能同时表示事件的及时性和因果关系属性。为了解决这些挑战, 原论文中提出了一个名为”逻辑和常识引导嵌入” (LCGE) 的模型, 它共同学习了与事件的时间敏感性、及时性和因果关系相关的表示, 以及从常识角度来看事件的时间无关表示。原论文中设计了一个时间规则学习算法, 时间规则引导的谓词嵌入正则化来学习事件的因果关系。提出了一种基于联合事件和常识的 KG 嵌入策略, 通过学习涉及及时性和因果关系的时间敏感表示以及常识视图中与时间无关的表示来对每个事件进行评分。改进方向主要围绕修改损失函数和对其中compleX模型中添加注意力机制, 尝试从多个方面提高模型的基于常识的推理能力。

2 相关工作

时间知识图谱补全 (TKGC) 是一种重要技术, 用于预测TKG中缺失的潜在事件是否会发生, 即图1中显示的 (Xi Jinping, Consult, Barack Obama, 2014-06-15)。特别地, 事件只在特定时间有效, 即及时性。现有的TKGC方法可以分为两类: (1) 基于演变的模型能够表示事件之间的因果关系以推理未来事件, 例如Know-Evolve^[1]、RE-NET^[2]和CyGNet^[3]。如图1所

示，当两个事件按照一定的时间顺序发生时，一个事件会影响另一个事件。先发生的事件是原因，后发生的事件是结果。（2）本文关注的TKG嵌入（TKGE）模型评估事件的合理性，通过利用实体和谓词的嵌入以及时间戳对事件进行评分，包括TTransE^[4]、HyTE^[5]和DE-Simple^[6]。TKGE模型认为不同时刻发生的事件完全独立，这些方法预测已知时间的事件。

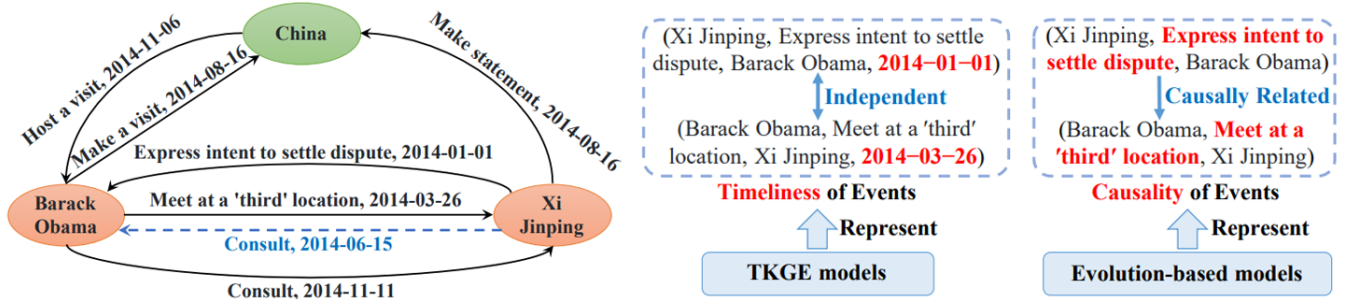


图 1. 时空知识图谱构成

2.1 传统的知识图谱嵌入模型

知识图谱嵌入技术旨在通过学习实体和谓词的嵌入来评分事实的合理性。TransE^[7](Bordes等人, 2013)模拟了三元组事实之间的交互，将谓词视为从主语到客体的平移操作。更先进的方法将谓词表示为旋转操作，用于建模对称和反对称谓词，比如RotatE^[8](Sun等人, 2019)、QuatE^[9](Zhang等人, 2019)和DualE^[10](Cao等人, 2022)。RESCAL^[11](Nickel, Tresp, and Kriegel, 2011)利用三阶张量分解来计算事实的真值。DistMult^[12](Yang等人, 2014)和CompLex^[13](Trouillon等人, 2016)简化了RESCAL模型，减少了参数并提高了知识图谱补全的性能。

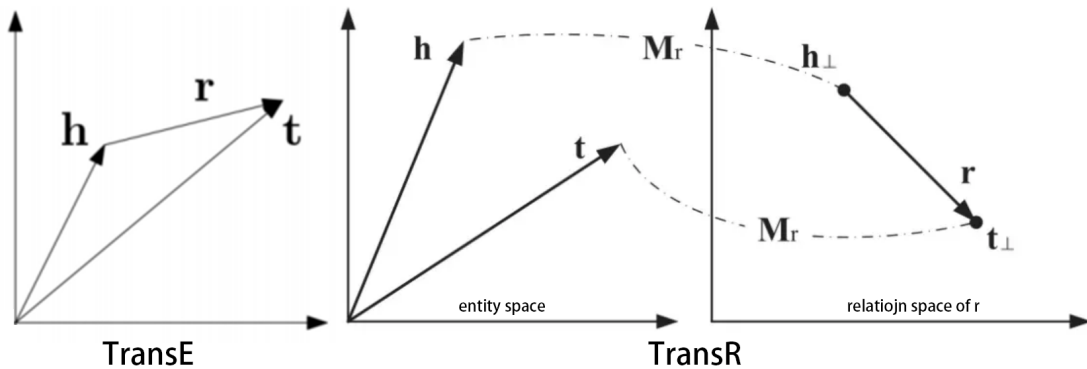


图 2. transE系列模型示意图

2.2 时间感知的知识图谱嵌入模型

时间感知的知识图谱嵌入模型通过增加时间嵌入来扩展传统的知识图谱嵌入方法，以表示时间感知事件。TTransE^[4](Leblay和Chekol, 2018)在基于平移的评分函数中添加了额外的时间嵌入，以表示事件的及时性。受TransH^[14](Wang等人, 2014)中提出的与关系特定的超平面启发，HyTE^[5](Dasgupta, Ray, and Talukdar, 2018)将实体和谓词投影到特定时间的超平面上。DE-Simple^[6](Goel等人, 2020)利用时序实体嵌入来表示不同时间戳下的每个实体。ATiSE^[15](Xu等人, 2019)学习实体和谓词的时间感知嵌入，将其表示为高斯分布以表示时间

的不确定性。TeRo^[16](Xu等人, 2020)扩展了HyTE, 通过特定于各个时间戳的旋转操作学习时间敏感的实体和谓词嵌入^[8](Sun等人, 2019)。TComplEx^[17](Lacroix, Obozinski, and Usunier, 2020)将ComplEx升级为通过引入时间信息的四阶张量分解来评分每个事件。

2.3 规则学习

根据知识图谱的符号特性, 逻辑规则自然适用于知识图谱补全任务。Horn规则是一种典型的逻辑规则, 形式为 $a_1 \Leftarrow a_2 \wedge a_3 \wedge \dots \wedge a_n$, 其中 a_1 表示规则头部的原子(即头部原子), a_2, \dots, a_n 表示规则主体的原子(即主体原子)。一些有效的规则学习算法专门用于大规模知识图谱, 依赖于规则搜索和规则质量评估, 例如AMIE+^[18](Galarraga等人, 2015)、ScaLeKB^[19](Y.Chen, D.Z.Wang, and S.Goldberg, 2016)、RuLES^[20](Ho等人, 2018)、Anyburl^[21](Meilicke等人, 2019)、DRUM^[22](Sadeghian等人, 2019)、RLvLR^[23](Omran, Wang, and Wang, 2019a)和RNNLogic^[24](Qu等人, 2020)。然而, 这些方法设计用于静态知识图谱而非时间感知的知识图谱。StreamLearner^[25](Omran, Wang, and Wang, 2019b)是唯一已知的算法, 用于挖掘限制所有主体原子的时间规则。

3 原文方法

3.1 本文方法概述

然而, 先前的模型面临几个挑战:(1) 现有的TKGC模型认为任何TKG简单包含涉及时间的事件, 但它们忽略了TKG中蕴含的长期有效的常识知识。(2) 基于演变的模型难以推理与过去事件弱相关的事件, 而TKGE模型无法利用事件之间的因果关系。总之, 所有现有的TKGC方法无法共同表示事件的及时性和因果关系。(3) 几乎所有先前的TKGC技术都是数据驱动的, 缺乏可解释性。此外, StreamLearner^[25](Omran, Wang, and Wang, 2019b)是唯一已知的从TKG自动挖掘时间规则的方法。然而, 它仅探索了规则主体中所有原子同时受限的单一模式, 但忽略了事件之间多样化的时间规则模式。

时间知识图谱(TKG)是一组带有时间信息的事件集合。每个事件表示为一个四元组 (s, p, o, t) , 其中 s 和 o 是主体和客体, p 表示谓词, t 表示时间戳或时间间隔。特别地, 具有时间间隔 $[ts, te]$ 的事件可以转换为两个具有时间戳的事件, 即 (s, p, o, ts) 和 (s, p, o, te) 。

时间规则被定义为附加时间标签的原子的合取式。在我们的工作中, 我们关注形式为:

$$p_{n+1}(x, y, t) \Leftarrow p_1(x, z_1, t_1) \wedge \dots \wedge p_n(z_{n-1}, y, t_n)$$

的时间规则, 其中 $p_i(i = 1, \dots, n + 1)$ 是谓词。 x, y 和 $z_j(j = 1, \dots, n - 1)$ 表示实体变量。特别地, t 和 $t_l(l = 1, \dots, n)$ 表示满足约束条件 $t_1 \leq t_2 \leq \dots \leq t_n \leq t$ 的时间变量。时间规则表示当规则体成立时规则头部将发生。时间规则的一个实例是通过用特定实体和时间戳替换变量得到的。

3.2 时间规则学习

我们开发了一种新颖的静态到动态策略来挖掘具有各种模式的时间规则。在静态规则学习阶段，我们首先将训练集中的所有四元组事件通过屏蔽每个事件的时间信息转换为三元组，如图2所示。然后，我们获得一个全局静态知识图谱（GSKG），其中包含所有三元组，并使用任何现有的规则学习算法（如AMIE+^[18]（Galarraga等人，2015）或Anyburl^[21]（Meilicke等人，2019））从GSKG中挖掘静态规则。值得注意的是，时间规则可以被视为具有各种时间规则模式的静态规则的扩展。

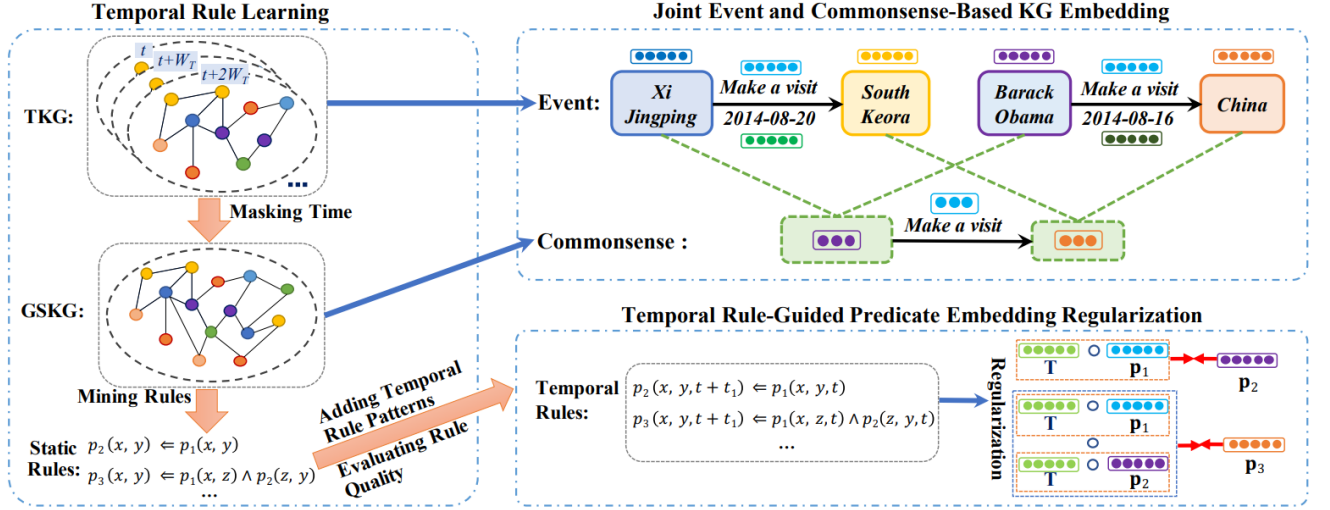


图 3. LCGE模型整体框架

时间规则模式的构建。在动态规则学习阶段，我们根据原子之间的多样化时间序列设计了五种时间规则模式，如图3所示，分别是：

1. 一个长度为1的规则，其中两个原子具有不同的时间戳： $p_2(x, y, t + t_1) \Leftarrow p_1(x, y, t)$ 。
2. 一个长度为1的规则，其中两个原子在同一时间有效： $p_2(x, y, t) \Leftarrow p_1(x, y, t)$ 。
3. 一个长度为2的规则，其中三个原子的时间戳互不相同： $p_3(x, y, t + t_1 + t_2) \Leftarrow p_1(x, z, t) \wedge p_2(z, y, t + t_1)$ 。
4. 一个长度为2的规则，其中规则体的时间戳与规则头部的时间戳不同： $p_3(x, y, t + t_1) \Leftarrow p_1(x, z, t) \wedge p_2(z, y, t)$ 。
5. 一个长度为2的规则，其中三个原子在同一时间有效： $p_3(x, y, t) \Leftarrow p_1(x, z, t) \wedge p_2(z, y, t)$ 。

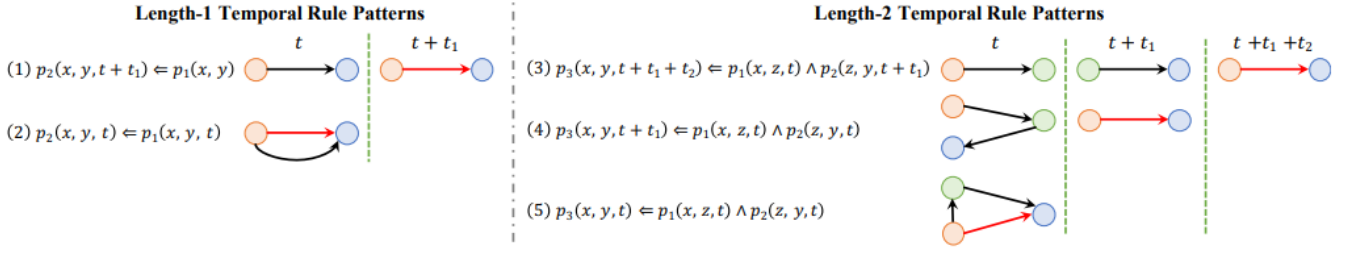


图 4. 时间规则示意图

时间规则的质量评估。根据现有规则学习算法挖掘的静态规则和提出的五种时间规则模式，我们将每个静态规则扩展为相应的候选时间规则。为了评估每个候选时间规则的质量，我们首先将同一时间窗口内的事件合并成子图，因为一些因果相关的事件可能发生在同一子图或相邻的子图中。然后，我们搜索满足候选时间规则实例化的所有事件。对于长度为1的时间规则，例如形式为 $p_2(x, y, t + T_1) \Leftarrow p_1(x, y, t)$ 的规则，当 $T_1 = 0$ 时符合模式（1），当 $T_1 > 0$ 时符合模式（2）。因此，根据以前规则学习模型的评价标准（Sadeghian等人，2019），我们提出支持度（SD）、标准置信度（SC）和头部覆盖（HC）作为时间规则的评价标准：

$$\begin{aligned}
 SD &= \begin{cases} \#(a, b) : p_1(a, b, t) \wedge p_2(a, b, t), T_1 = 0 \\ \#(a, b) : \sum_{t_{1i} \in [1, W_T]} p_1(a, b, t) \wedge p_2(a, b, t + t_{1i}), T_1 > 0 \end{cases} \\
 SC &= \begin{cases} \frac{SD}{\#(a, b) : p_1(a, b, t)}, T_1 = 0 \\ \frac{1}{|W_T|} \cdot \frac{SD}{\#(a, b) : p_1(a, b, t)}, T_1 > 0 \end{cases} \\
 HC &= \begin{cases} \frac{SD}{\#(a, b) : p_2(a, b, t)}, T_1 = 0 \\ \frac{SD}{\#(a, b) : \sum_{t_{1i} \in [1, W_T]} p_2(a, b, t + t_{1i})}, T_1 > 0 \end{cases}
 \end{aligned}$$

其中 W_T 表示预定义时间窗口的大小。我们按照时间规则的评价标准遍历时间戳 t ，获得每个候选时间规则的SC和HC。满足SC和HC阈值的候选时间规则将用于后续的正则化策略。

3.3 设计损失函数

3.3.1 设计的RGPR机制

基于暗示原子间因果关系的生成的时间规则，我们定义了一个时间传输算子 T ，确保规则中的所有原子在同一时间计算它们之间的相关性。以时间规则 $p_3(x, y, t + t_1 + t_2) \Leftarrow p_1(x, z, t) \wedge p_2(z, y, t + t_1)$ 为例，将规则中的主体原子 $p_1(x, z, t)$ 投影到头部原子的相同时间窗口需要进行两次时间传输操作。此外，我们提出了基于时间规则的谓词嵌入正则化RGPR，

将蕴含在各种时间规则模式中的因果关系注入到相应时间规则模式对应的谓词嵌入中：

$$\begin{aligned}
(1) \quad & p_2(x, y, t + t_1) \Leftarrow p_1(x, y, t) : \\
& G = \|T \circ p_{r1} - p_{r2}\| \\
(2) \quad & p_2(x, y, t) \Leftarrow p_1(x, y, t) : \\
& G = \|p_{r1} - p_{r2}\| \\
(3) \quad & p_3(x, y, t + t_1 + t_2) \Leftarrow p_1(x, e, t) \wedge p_2(e, y, t + t_1) : \\
& G = \|T \circ T \circ p_{r1} \circ (T \circ p_{r2}) - p_{r3}\| \\
(4) \quad & p_3(x, y, t + t_1) \Leftarrow p_1(x, e, t) \wedge p_2(e, y, t) : \\
& G = \|T \circ p_{r1} \circ (T \circ p_{r2}) - p_{r3}\| \\
(5) \quad & p_3(x, y, t) \Leftarrow p_1(x, e, t) \wedge p_2(e, y, t) : \\
& G = \|p_{r1} \circ p_{r2} - p_{r3}\|
\end{aligned}$$

其中， p_{r1} 、 p_{r2} 和 p_{r3} 分别表示谓词 p_1 、 p_2 和 p_3 的嵌入。 \circ 表示Hadamard积。

3.3.2 联合事件和基于常识的KGE

我们充分利用常识的长期有效性来准确评估事件的可信度，因为一些违反常识的事件永远不会发生。因此，我们从既具有时间敏感性又具有时间无关性的角度对每个事件进行建模。

为了学习事件的时间敏感性表示，我们受到了TKGE模型TComplEx^[17](Lacroix, Obozinski, and Usunier 2020)的启发，通过四阶张量分解学习了每个事件与时间戳相关的及时性。此外，事件之间的因果关系可以通过我们的RGPR机制与主体和客体嵌入一起表示。对于给定的事件四元组 (s, p, o, t) ，时间敏感评分函数定义为：

$$E_1(s, p, o, t) = \text{Re}(\mathbf{s}^\top \text{diag}(\mathbf{p}_t + \mathbf{p}_r) \bar{\mathbf{o}}) = \text{Re}\left(\sum_{i=1}^d [\mathbf{s}]_i \cdot [\mathbf{p} \circ \mathbf{t} + \mathbf{p}_r]_i \cdot [\bar{\mathbf{o}}]_i\right)$$

其中， $s \in \mathbb{C}^d$ ， $p \in \mathbb{C}^d$ 和 $o \in \mathbb{C}^d$ 表示分别与主体 s 、谓词 p 和客体 o 相关的 d 维复向量空间中的嵌入。 o^* 是 o 的共轭。特别地， $pt = p \circ t$ 表示受时间戳 t 约束的谓词嵌入。与前述的正则化谓词嵌入相比，根据公式5-9，如果一些事件成立，那么与这些事件有因果关系的其他事件将根据谓词嵌入的正则化获得更高的分数。 $[x]_i$ 表示复向量 x 的第 i 个值。基于公式10中的开发的评分函数，我们能够共同表示事件的及时性和因果关系，有助于更充分地表示事件的时间敏感性。

为了学习与事件相关的时间无关的常识表示，我们将每个事件的时间戳掩盖，将事件四元组 (s, p, o, t) 转换为事实三元组 (s, p, o) 。受到一些典型常识知识图谱（例如ConceptNet, Speer, Chin, and Havasi 2017）的启发，常识表示为由谓词连接的两个概念。因此，我们通过可学习的概念和谓词嵌入以及基于常识的时间无关性评分函数来对常识视角下的每个事件进行评分：

$$E_2(s, p, o) = \text{Re}\left(\sum_{i=1}^k [\mathbf{s}_c]_i \cdot [\mathbf{p}_c]_i \cdot [\bar{\mathbf{o}}_c]_i\right)$$

其中, $sc \in \mathbb{C}^k$, $pc \in \mathbb{C}^k$, $oc \in \mathbb{C}^k$ 分别表示与主体 s 、谓词 p 和客体 o 相关的 k 维复向量空间中的概念嵌入。特别地, k 应该设置得比 d 小, 以增强实体概念嵌入的抽象特征。

3.3.3 优化目标

我们使用对数softmax损失函数和N3正则化设计了用于训练的优化目标:

$$\begin{aligned}
 L &= \sum_{(s,p,o,t) \in \mathcal{T}} (L_1 + \lambda \cdot L_2) \\
 L_1 &= -\log \left(\frac{\exp(E_1(s, p, o, t))}{\sum_{e_i \in \mathcal{E}} \exp(E_1(e_i, p, o, t))} \right) \\
 &\quad -\log \left(\frac{\exp(E_1(s, p, o, t))}{\sum_{e_i \in \mathcal{E}} \exp(E_1(s, p, e_i, t))} \right) \\
 &\quad + \alpha_1 (\|s\|_3^3 + \|p_t\|_3^3 + \|p_r\|_3^3 + \|o\|_3^3) \\
 L_2 &= -\log \left(\frac{\exp(E_2(s, p, o))}{\sum_{e_i \in \mathcal{E}} \exp(E_2(e_i, p, o))} \right) \\
 &\quad -\log \left(\frac{\exp(E_2(s, p, o))}{\sum_{e_i \in \mathcal{E}} \exp(E_2(s, p, e_i))} \right) \\
 &\quad + \alpha_2 (\|s_c\|_3^3 + \|p_c\|_3^3 + \|o_c\|_3^3)
 \end{aligned}$$

其中, L_1 和 L_2 分别表示特定于时间敏感事件和时间无关常识的损失函数。 T 和 E 分别表示事件集和时间知识图谱中的实体集。 e_i 表示实体集 E 中的一个实体。 α_1 和 α_2 分别定义为对实体嵌入和谓词嵌入进行N3正则化的权重。 λ 是整体损失函数中用于权衡时间敏感性和时间无关性表示的权重, 用于在每个事件的时间敏感性和时间无关性表示之间进行权衡。

整体优化目标是损失函数 L 和谓词嵌入正则化的组合。此外, 我们的模型使用Adam优化器^[26] (Kingma and Ba 2015) 进行训练, 以学习实体、谓词、概念和时间戳的嵌入。

4 复现细节

4.1 与已有开源代码对比

论文中原有代码链接: <https://github.com/ngl567/LCGE> 在尝试改进优化器方面, 采用L2正则化作为一项惩罚项, 我们期望通过限制模型参数的大小, 减少过拟合现象, 从而提升模型的泛化性能。然而, 实验结果显示改进后的优化器效果不如原始结果, 这可能表明在当前问题上使用L2正则化并没有产生预期的正面影响, 或者调整的正则化强度可能需要进一步优化。

另一方面, 引入自注意力机制, 即在模型内部建立每个元素与其他元素之间的关联, 可以使模型更加关注输入的不同部分, 从而提高模型对信息的利用效率。自注意力机制允许模型为每个输入元素分配不同的权重, 根据其在上下文中的重要性来调整表示, 这有助于更好地捕捉输入之间的复杂关系。

实验结果显示, 加入自注意力机制后, 模型的性能得到提升, 这可能表明在当前任务中, 模型更适合通过自注意力机制来学习输入数据之间的关联, 而不是简单地通过L2正则化进行参数约束。这也强调了在调整模型时, 根据任务特性选择合适的改进手段的重要性。

```

def forward(self, x):
    lhs = self.embeddings[0](x[:, 0])
    rel = self.embeddings[1](x[:, 1])
    rel_no_time = self.embeddings[3](x[:, 1])
    rhs = self.embeddings[0](x[:, 2])
    time = self.embeddings[2](x[:, 3])
    transt = self.embeddings[4](torch.LongTensor([0]).cuda())
    # 提取输入数据的embedding:
    # lhs、rel、rhs、time和transt分别提取了左实体、关系、右实体、时间和时间转移的embedding。
    # 这些embedding是通过调用nn.Embedding层得到的。
    # print('lhs:', lhs.shape, '\n', 'rel:', rel.shape, '\n', 'rhs:', rhs.shape, '\n', 'time:', time.shape)
    # embedding后是(1000, 4000) 1000为epoch的限制, 每个单词转化为4000维的向量

    input_sequence = torch.cat([lhs, rel, rhs, time], dim=0)
    # 将左实体、关系、右实体拼接成一个序列
    attended_output, _ = self.self_attention(input_sequence, input_sequence, input_sequence)

    lhs, rel, rhs, time = torch.split(attended_output, [lhs.size(0), rel.size(0), rhs.size(0), time.size(0)], dim=0)

    lhs = lhs[:, :self.rank], lhs[:, self.rank:]
    rel = rel[:, :self.rank], rel[:, self.rank:]
    rhs = rhs[:, :self.rank], rhs[:, self.rank:]
    time = time[:, :self.rank], time[:, self.rank:]
    transt = transt[:, :self.rank], transt[:, self.rank:]
    # print(transt[1])
    # print('lhs[0]:', lhs[0].shape, '\n', 'rel[0]:', rel[0].shape, '\n', 'rhs[0]:', rhs[0].shape, '\n', 'time[0]:'

```

图 5. 修改代码示意图（图中是部分添加注意力机制的示意图，还在相对应的别的位置也增加了变化，详情请见代码附件）

4.2 实验环境搭建

实验使用RTX3090单机进行训练，训练一次时间约为10h，使用pycharm+jupyter环境。

4.2.1 数据集使用

ICEWS（Integrated Conflict Early Warning System）是一个旨在监测国际冲突并提供早期警告的系统。其中，ICEWS14和ICEWS04-15是两个时间段内的数据集，为研究国际事务提供了丰富的信息。

4.2.2 ICEWS14数据集

ICEWS14数据集涵盖了2014年的国际事件数据。该数据集包括有关冲突、外交和合作等方面的详细信息。研究人员可以利用这一数据集来深入了解2014年全球事务的动态和国家间关系。

4.2.3 ICEWS04-15数据集

ICEWS04-15数据集扩展了时间范围，覆盖了2004年至2015年的数据。与ICEWS14类似，这个数据集提供了关于国际事件的广泛信息，涉及军事冲突、外交关系、经济合作等多个领域。由于其更长的时间跨度，可以进行更广泛和深入的分析，了解长期趋势和模式。通过使用ICEWS14和ICEWS04-15，进行详尽的分析，揭示国际事务的演变、国家之间相互作用的模式，并为决策者提供有关全球动态的重要见解。

4.3 创新点

添加自注意力机制和多头注意力机制在深度学习任务中都具有独特的优点，它们分别为模型提供了灵活性和表示能力的提升。自注意力机制的优点：自注意力机制允许模型在处理序列或集合数据时，为每个元素分配不同的权重，以反映其在上下文中的重要性。这种全局关联性使得模型能够捕捉输入之间更为复杂的依赖关系，尤其对于长距离的依赖关系更为有效。自注意力机制允许模型动态地调整注意力权重，使其适应不同任务和输入模式。这种灵活性使得模型能够更好地适应不同的输入数据分布，而无需手动设计特征。多头注意力机制通过并行地学习多组注意力权重，能够更全面地捕捉输入数据的不同子空间中的信息。这提高了模型的表示能力，使其更好地捕获输入数据的复杂结构和特征。多头注意力机制在学习中可以关注输入的不同方面，这有助于减轻过拟合问题。通过同时关注多个方面，模型更有可能产生更为泛化的表示，从而提高对未见数据的泛化能力。

在前述实验中选择这两种注意力机制的理由可能是为了综合利用它们的优势。自注意力机制可以捕捉全局的关系，而多头注意力机制则可以通过并行学习多组权重提高模型的表示能力。实验的初衷可能是通过这种组合，使模型更好地理解输入数据的复杂结构，提高对任务的性能。然而，实验结果表明改进优化器的尝试效果不佳，而引入自注意力机制后获得了性能提升，这强调了对于不同任务和数据，适当选择注意力机制的重要性。

4.4 实验结果

实验在上述两个数据集上进行训练，经过训练后得到的结果如下图所示，这里以ICEWS14数据集为例。其中图5-图8是原论文中作者的复现。图9-图12为添加注意机制的Mrr分数和HIT@k指数。由于仅仅修改优化函数没有改进效果，故没有展示。

4.4.1 MRR (Mean Reciprocal Rank)

MRR是在知识图谱领域中常用的排序质量评估指标之一。对于查询 q ，其相关文档排名为 r ，MRR的计算方式为：

$$\text{MRR}(q) = \frac{1}{r}$$

这里，MRR越高表示相关文档越早出现在排序结果中，反映了系统更好地定位相关信息的能力。

4.4.2 HIT@k (Top-k Accuracy)

HIT@k用于衡量模型在前 k 个推荐或检索结果中是否包含了真实相关的项目。对于查询 q ，HIT@k的计算方式为：

$$\text{HIT@k}(q) = \begin{cases} 1, & \text{如果真实相关项目出现在前k个结果中} \\ 0, & \text{否则} \end{cases}$$

HIT@k表示模型在前 k 个推荐中的准确性，即模型是否能够在前 k 个结果中包含真实相关的项目。

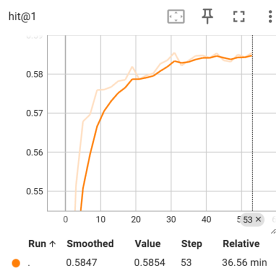


图 6. 原文hit1

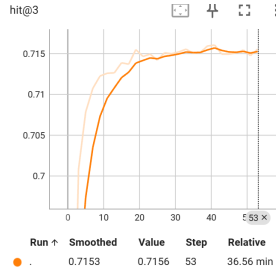


图 7. 原文hit3

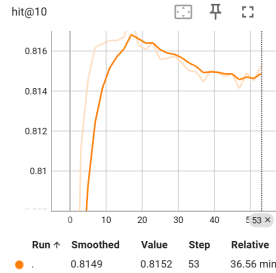


图 8. 原文hit10

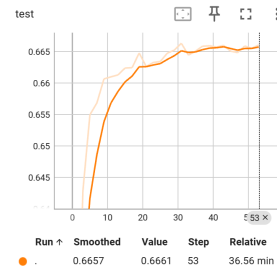


图 9. 原文MRR



图 10. 修改后hit1

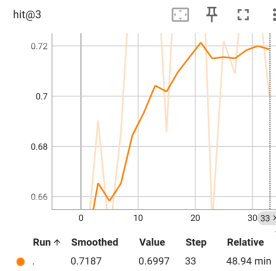


图 11. 修改后hit3

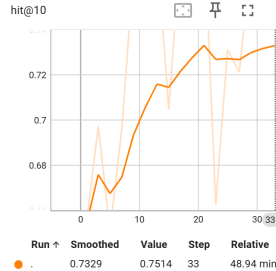


图 12. 修改后hit10

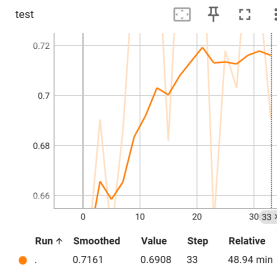


图 13. 修改后MRR

通过实验结果的数据，我们可以观察到引入注意力机制在模型性能上带来了一定程度的提升。注意力机制的引入使得在三元组中多了一个权重因素，这有助于模型更加关注重要的信息，提高了模型在某些任务上的准确性。

然而，值得注意的是，虽然注意力机制在性能上带来了提升，但也伴随着推理时间的增加。引入额外的权重计算和注意力机制的处理过程可能导致模型的推理时间延长。在实际应用中，我们需要根据具体需求权衡性能和推理时间，并根据任务的要求进行取舍。

因此，在使用注意力机制时，需要权衡其在性能提升和推理时间上的投入。根据具体场景的需求，可以灵活选择是否引入注意力机制，以达到更好的平衡。这种灵活性允许在不同情境下调整模型的配置，以满足实际应用中的不同要求。

5 总结与展望

本文通过对代码实验中的改进尝试以及对注意力机制的引入进行了详细描述。实验中，优化器的改进效果未达到预期，表明当前的调整可能并不适用于解决问题。然而，在加入自注意力机制后，实验结果取得了一定的提升。

不足与改进方向：

1. 优化器改进不理想：实验中改进的优化器效果未如预期，可能是因为L2正则化并不适用于当前问题，或者正则化的参数需要更精细的调整。未来工作可以考虑尝试其他优化器调整策略，如学习率调整、不同的正则化方法等。

2. 未来的改进方向：为了进一步提高实验效果，可以考虑引入更先进的预训练模型，例如K-BERT。通过使用预训练模型，可以在更大规模的数据上进行训练，提高模型的表示能力和泛化性能。将K-BERT引入当前任务，探索其对知识图谱和规则化推理的效果。K-BERT结合了BERT的预训练能力和知识图谱的信息，可能对当前任务的改进具有积极影响。

尽管在实验中未能取得理想的结果，但这个经历对我而言仍然是一次收获颇丰的探索。

通过这个项目，我深入了解了在编辑过程中确保项目顺序和结构的重要性。这对于我作为一个初学者来说具有重要的启发和帮助，使我对项目管理和结构设计有了更清晰的认识。

未来，我的研究方向可能会集中在知识图谱和预训练大模型结合的领域。考虑到知识图谱的丰富信息和预训练大模型在语义理解上的优势，我计划进一步探索将这两者结合的可能性。这包括将知识图谱引入到大型预训练模型中，以更全面地利用大规模语料库和领域知识。

这种结合有望提升模型对于知识图谱信息的理解和推理能力。通过整合领域专业知识和广泛的语料库，我期望我的研究可以为提高模型在知识图谱相关任务上的性能做出贡献。这一步可能推动对于知识图谱应用更深层次理解的研究，并在实际应用中取得更为显著的成果。

参考文献

- [1] Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *international conference on machine learning*, pages 3462–3471. PMLR, 2017.
- [2] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. *arXiv preprint arXiv:1904.05530*, 2019.
- [3] Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, page 4732–4740, Sep 2022.
- [4] Julien Leblay and Melisachew Wudage Chekol. Deriving validity time in knowledge graph. In *Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW ' 18*, Jan 2018.
- [5] Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Jan 2018.
- [6] Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. Diachronic embedding for temporal knowledge graph completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, page 3988–3995, Jun 2020.
- [7] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Le Centre pour la Communication Scientifique Directe - HAL - Inria, Le Centre pour la Communication Scientifique Directe - HAL - Inria*, Dec 2013.
- [8] Zhiqing Sun, Zhihong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv: Learning, arXiv: Learning*, Feb 2019.

- [9] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. Quaternion knowledge graph embeddings. *arXiv: Learning, arXiv: Learning*, Apr 2019.
- [10] Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. Dual quaternion knowledge graph embeddings. *Proceedings of the AAAI Conference on Artificial Intelligence*, page 6894–6902, Sep 2022.
- [11] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. *International Conference on Machine Learning, International Conference on Machine Learning*, Jun 2011.
- [12] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.
- [13] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. *International Conference on Machine Learning, International Conference on Machine Learning*, Jun 2016.
- [14] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014.
- [15] Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. Temporal knowledge graph completion based on time series gaussian embedding. Nov 2019.
- [16] Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. Tero: A time-aware knowledge graph embedding via temporal rotation. *Cornell University - arXiv, Cornell University - arXiv*, Oct 2020.
- [17] Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. Tensor decompositions for temporal knowledge base completion. *arXiv: Machine Learning, arXiv: Machine Learning*, Apr 2020.
- [18] Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. Fast rule mining in ontological knowledge bases with amie
+
+. *The VLDB Journal*, 24(6):707–730, 2015.
- [19] Yang Chen, Daisy Zhe Wang, and Sean Goldberg. Scalekb: scalable learning and inference over large knowledge bases. *The VLDB Journal*, 25(6):893–918, Dec 2016.
- [20] Vinh Thinh Ho, Daria Stepanova, Mohamed H. Gad-Elrab, Evgeny Kharlamov, and Gerhard Weikum. *Rule Learning from Knowledge Graphs Guided by Embedding Models*, page 72–90. Jan 2018.

- [21] Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. Anytime bottom-up rule learning for knowledge graph completion. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Aug 2019.
- [22] Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and DaisyZhe Wang. Drum: End-to-end differentiable rule mining on knowledge graphs. *Cornell University - arXiv, Cornell University - arXiv*, Oct 2019.
- [23] Pouya Ghiasnezhad Omran, Kewen Wang, and Zhe Wang. An embedding-based approach to rule learning in knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1348–1359, 2019.
- [24] Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. *Cornell University - arXiv, Cornell University - arXiv*, Oct 2020.
- [25] PouyaGhiasnezhad Omran, Kewen Wang, and Zhe Wang. Learning temporal rules from knowledge graph streams. *AAAI Spring Symposium Combining Machine Learning with Knowledge Engineering, AAAI Spring Symposium Combining Machine Learning with Knowledge Engineering*, Jan 2019.
- [26] DiederikP. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv: Learning, arXiv: Learning*, Dec 2014.