文章编号:1674-2974(2023)08-0213-10 DOI:10.16339/j.cnki.hdxbzkb.2023293

面向脚本事件预测的稠密事件图嵌入

宁佐廷[1],贾明颐[2],安莹[3],段俊文[2†]
(1. 湖南警察学院 网络侦查技术湖南省重点实验室,湖南 长沙 410138;
2. 中南大学 计算机学院,湖南 长沙 410083;
3. 中南大学 大数据研究院,湖南 长沙 410083)

摘 要:脚本事件预测是指在给定现有上下文事件链的情况下预测后续事件.在现实世界中,不同事件的关系可以自然地表示为图结构,以事件为节点,以时间或因果关系为边.由于语料库规模有限和信息提取工具的能力不足,先前工作中自动构建的事件图会存在稀疏性问题,并且无法集成来自高阶节点的信息以支持多步推理.为了解决这个问题,本文提出使用可学习的多维加权邻接矩阵的稠密事件图(DEG)来解决之前事件图存在的稀疏性问题并表征事件之间的关系强度.为了实现DEG的嵌入表示,本文同时提出了一个通用框架,该框架能够将高阶事件演化信息组合到事件表示中.在多选叙事完形填空(multiple choice narrative cloze,MCNC)和连贯多选叙事完形填空(coherent multiple choice narrative cloze ,CMCNC)数据集上进行了实验,结果证明了此框架的有效性.

关键词:脚本事件预测;稠密事件图;图卷积网络;事件抽取
中图分类号:TP399 **文献标志码**:A

Embedding Dense Event Graph for Script Event Prediction

NING Zuoting[1], JIA Mingyi[2], AN Ying[3], DUAN Junwen[2†]
(1. Key Laboratory of Network Crime Investigation of Hunan Provincial Colleges, Hunan Police Academy, Changsha 410138, China;
2. School of Computer Science and Engineering, Central South University, Changsha 410083, China;
3. Big Data Institute, Central South University, Changsha 410083, China)

Abstract: Script Event Prediction refers to predicting the subsequent event based on a given existing chain of context events. In the real world, the relationship of different events can be naturally represented as a graph structure, where events serve as nodes and their temporal or causal relations are depicted as edges. However, previous approaches that automatically constructed event graphs suffer from sparsity problem due to the limited scale of corpus and the incapability of information extraction tools. Moreover, they fail to integrate information from higher order nodes to support multi-step reasoning. To remedy this, we propose a Dense Event Graph (DEG) approach which use a learnable multi-dimensional weighted adjacency matrix to address the sparsity issue and characterize the relation strengths between events. To embed the DEG, we propose a general framework capable of combining

* **收稿日期**:2022-12-15
基金项目:网络犯罪侦查湖南省普通高校重点实验室开放基金资助项目(2020WLFZZC004), Open Research Fund of Key Laboratory of Network Crime Investigation of Hunan Provincial Colleges (2020WLFZZC004);湖南省自然科学基金资助项目(2021JJ40783), Hunan Provincial Natural Science Foundation of China (2021JJ40783)
作者简介:宁佐廷(1982—),男,湖南隆回人,湖南警察学院副教授,博士
† 通信联系人,E-mail: jwduan@csu.edu.cn

high-order event evolution information into the event representations. Experimental results on the multiple choice narrative cloze (MCNC) and coherent multiple choice narrative cloze (CMCNC) demonstrate the effectiveness of our approach.

Key words: script event prediction; dense event graph; graph convolutional networks; event extraction

叙述性脚本由一系列事件组成,并描述这些事件之间的时间和因果关系.最近,从非结构化文本中自动归纳脚本受到了越来越多的关注,因为脚本蕴含了重要的现实世界知识,这对于常识推理至关重要.图 1 给出了针对用餐场景的脚本事件预测的示例.对于人类来说,凭借我们拥有的背景知识,我们可以很容易地推断出下一个事件为 X 离开了饭店[$Leave(X, restaurant)$].了解事件如何随时间演变有利于改善各种下游人工智能应用程序,例如推荐系统、用户意图理解和对话生成.然而,根据现有事件预测未来可能发生的事情仍然是一个具有挑战性的问题.

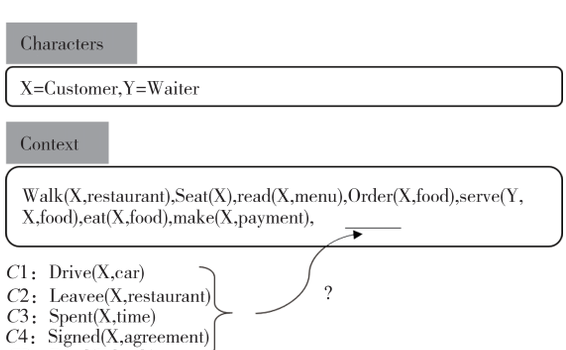

图 1 脚本事件预测的一个示例

Fig.1 An example forscript event prediction

本文遵循最近关于脚本学习的研究工作[1-2],其任务是根据现有事件从候选列表中预测最有可能发生的后续事件.在早期工作中[1-3],事件使用词袋表示,并且仅对成对关系建模,这不足以支持鲁棒的预测.早期的工作通过浅层的词汇或句法特征来表示事件,例如单词、词根、位置标签和依存关系[1,4].但由于它们的稀疏性,这种方法往往缺乏语义表示能力.

随着词嵌入表示方法的流行,许多方法开始将事件中的实体和参数语义组合成向量表示.然而,属性之间的丰富联系并没有得到充分利用.基于神经网络的模型的引入为这项任务带来了重大进展.Ding 等人[5]首先提出利用神经张量网络的表达能力来直接捕捉参数之间的相互作用.Weber 等人[6]提出了一个类似的网络来对事件的场景级语义进行建模.Granthorh-Wilding 等人[2]使用神经组合模型将事件组合成密集向量.Pichotta 等人[7]以及 Wang 等人[8]利用 seq-seq 模型来捕获事件链中的时间信息.

另一方面,如图 2(a) 所示,现实世界中的事件可以自然地构成图形结构.这也使得图神经网络成为一个值得深入研究的课题.图神经网络最早由 Gori 等人[9]和 Scarselli 等人[10]提出,作为递归神经网络的推广.GNN 的递归性质允许更新其邻居的节点输出.这个想法后来被扩展到用 GRU 代替递归神经网络[11-12]、卷积神经网络[13]和掩码多头自注意力[14].无监督方法通常依赖于图结构和节点连接性,Tang 等人[15]通过一阶和二阶邻近度学习节点表示,Perozzi 等人[16]基于随机游走理论获得网络节点嵌入.然而,图神经网络的一个基本假设是图中节点之间的连接关系是已知的,这不适用于我们的场景.

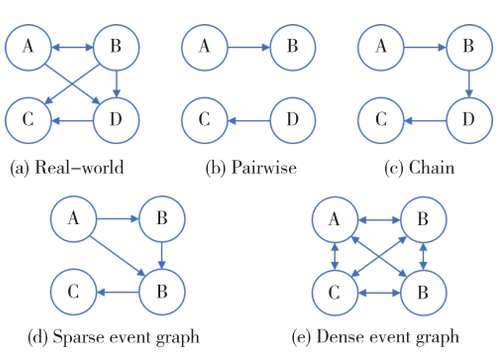

图 2 针对事件建模的不同结构

Fig.2 Different structures for event modeling

同时,从事件图到事件对或事件链的转换有可能会丢失支持后续事件预测的重要结构信息,如图 2(b) 和 2(c) 所示.为了克服这个问题,Li 等人[17]提出构建一个叙事事件演化图作为外部知识来支持脚本事件预测.由于语料库大小有限和信息提取工具的不完善,他们的图可能会丢失事件之间的重要联系并出现稀疏性问题,如图 2(d) 所示.

为了解决上述问题,本文提出使用稠密图[图 2(e)]来表示事件,其中节点是事件,边是它们的关系,事件彼此完全连接.使用一个可学习的多维邻接

矩阵用于定义事件之间的关系及其对应的关系强度,该矩阵通过从数据中学习从而进行更新.与之前的邻接矩阵不同,多维邻接矩阵是每个事件对的密集向量,它可以在细粒度级别捕获事件之间的关系.

为了得到正确的答案,通常需要进行多步推理.然而,以前的工作仅利用一阶邻接点(直接连接的事件)的信息来更新事件表示,未能捕获高阶事件演化模式,因此不足以支持多步推理.为了克服这个问题,本文提出了一个嵌入稠密事件图(DEG)的通用框架,它允许来自更高阶事件的信息影响事件表示.

在 benchmark 数据集上的实验结果表明,基于 DEG 的模型可以有效地利用丰富的图结构信息和高阶事件演化信息.本文提出的基于多维加权邻接矩阵和稠密事件图嵌入模型在此任务上实现了最新的性能.本文的主要贡献可以概括为:

1)本文提出利用事件之间的丰富联系构建稠密事件图并用于脚本事件预测任务的;

2)提出了一个稠密事件图嵌入框架,能够将高阶事件演化信息集成到事件表示中;

3)提出了一个可学习的多维加权邻接矩阵,它可以通过从数据中学习来更新,以在细粒度的特征级别上描述事件之间的关系强度.

1　问题定义

本节给出事件和脚本事件预测任务的正式定义.

事件:根据具体任务,将事件表示为两种不同的结构,即 $v(s, o, p)$ 或 (s, v, o),其中 v 是谓语动词,s、o 和 p 分别是主语、直接宾语和介词宾语.它们合称为事件参数.例如,可以从句子 *he opens the door with the key* 中提取事件 $opens(he, the\ door, with\ the\ key)$ 或者 $(he, opens, the\ door)$,其中 v = "*opens*", s = "*he*", o = "*the door*", p = "*with the key*".这仅仅是每个任务中事件结构的其中一种形式.

稠密事件图:本文从语料库中提取了一组叙事事件链 $\boldsymbol{C} = \{\boldsymbol{C}_1, \boldsymbol{C}_2, \cdots, \boldsymbol{C}_n\}$,事件链 $\boldsymbol{C}_i$ 的每个事件 e_i^k 均以 $v_i^k\left(s_i^k, o_i^k, p_i^k\right)$ 或 $\left(s_i^k, o_i^k, p_i^k\right)$ 的形式表示,取决于具体的任务.本文为每个事件链构建一个稠密事件图 $\boldsymbol{G}_i = \left(\boldsymbol{V}_i, \boldsymbol{A}_i\right)$,其中 $\boldsymbol{V}_i$ 表示链 $\boldsymbol{C}_i$ 中的节点,$\boldsymbol{A}_i$ 是含有 m 个事件的事件链的 $m \times m$ 有向加权邻接矩阵,它定义了事件之间的关系及事件之间的关系强度.

脚本事件预测:给定一个由 m 个上下文事件 $\{e_1, e_2, \cdots, e_m\}$ 和 n 个候选后续事件 $\{c_1, c_2, \cdots, c_n\}$ 组成的事件链,任务是根据给定的条件预测合理的后续事件上下文事件.注意,在给定的候选后续事件中,只有一个候选事件是正确的.

2　模型

本文提出了一个稠密事件图表示网络(DegNet),它由事件表示模块、多维加权矩阵生成模块和稠密事件图表示模块三个主要模块组成.该框架的整体架构如图 3 所示.基于该框架,本文进一步提出了 DegNet 的两个变体,即 DegNet-G 和 DegNet-R.两个变体之间的区别在于稠密事件图表示模块,其中 DegNet-G 利用了门控图神经网络,而 DegNet-R 结合了随机游走理论.

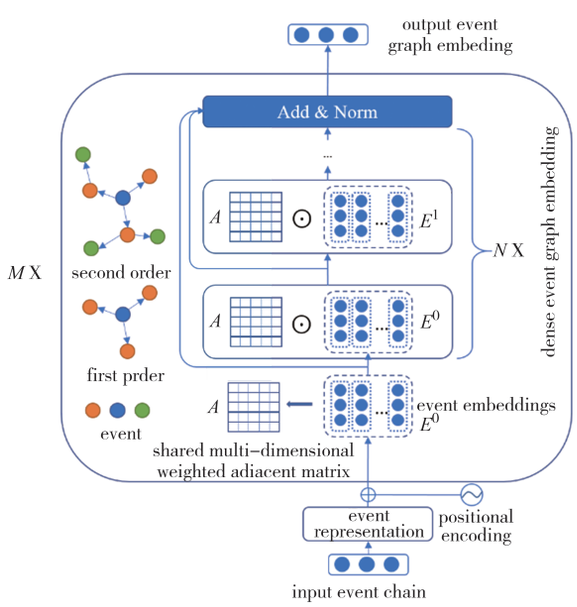

图 3　本文所采用方法的框架

Fig.3　The framework of the proposed architecture

2.1　事件表示

事件表示模块旨在学习事件元组的向量表示,用于初始化 DEG 中的节点表示.给定事件元组 $v_i(s_i, o_i, p_i)$,本文首先从预训练的词嵌入矩阵中检索动词 $e(v_i)$ 和参数 $e(s_i), e(o_i), e(p_i)$ 对应的词向量.然后学习一个映射函数 f,它将词向量语义组合成事件表示 $e_i = f(e(v_i), e(s_i), e(o_i), e(p_i))$.

本文使用具有非线性激活的连接组合,从而实现了最佳性能.同时也尝试了其他的事件表示模型,

例如：

$$e_i = \tanh\left(\left[e(v_i):e(s_i):e(o_i):e(p_i)\right]\right) \tag{1}$$

其中[:]表示向量拼接操作.在本文的其余部分将使用 E 来指代链中事件的向量表示，即 $\boldsymbol{E}=[e_1, e_2, \cdots, e_n]$.

事件序列的时间顺序对于叙事事件链建模非常重要.为了保留这些信息,本文采用与 Transformer[18] [等式(2)] 相同的位置编码策略,位置编码的每个维度都是根据它们的位置生成的.

$$\begin{aligned}\mathrm{PE}(\mathrm{pos}, 2i) &= \sin\left(\frac{\mathrm{pos}}{10\,000^{\frac{2i}{d}}}\right)\\ \mathrm{PE}(\mathrm{pos}, 2i+1) &= \cos\left(\frac{\mathrm{pos}}{10\,000^{\frac{2i}{d}}}\right)\end{aligned} \tag{2}$$

其中 d 是事件嵌入的维度,位置编码嵌入在训练期间是固定的.

2.2 多维加权邻接矩阵 (MWDM)

加权邻接矩阵(WAM)描述了事件之间的关系及其对应的关系强度.对于指向事件 e_j 的事件 e_i,它们的关系强度 r_{ij} 可以从两个角度来解释:1)从 e_i 到 e_j 的转移概率,2)e_i 和 e_j 在语义上相互作用.

之前的工作[17]分别使用静态的动词-动词连接和共现频率作为事件-事件关系及其关系强度,存在“语义鸿沟”问题.例如,“*throw*”和“*catch*”经常同时出现,但是,如果采用“*throw-catch*”的共现频率来定义“*he throws a ball*”和“*he catches cold*”的关系强度,则会不合适.为了克服这个问题,本文建议从数据中自动学习加权邻接矩阵并动态更新矩阵的值.

事件关系建模的另一个挑战是多义现象.多义词是自然语言中非常普遍的现象,在事件中也存在.例如,事件“*he went to the bank*”中,“*bank*”可以指“*riverside*”(河堤),也可以指“*financial bank*”(银行).通常,不同含义的同一个词对应同一个词向量.但是词向量的特征(即每个维度)应该携带不同的句法和语义信息[19-20].受此启发,为了减少歧义并更好地表征事件之间的关系,本文提出了多维加权邻接矩阵,它从细粒度特征级别衡量事件之间的关系强度.

图 4 说明了事件链 $\boldsymbol{E}=\{e_1, e_2, \cdots, e_n\}$ 的 MWDMA 是如何生成的.

给定来自 DEG 的两个事件的表示 e_i, e_j,首先将它们输入非线性矩阵并获得对齐向量 $\boldsymbol{a}_{i,j} \in \mathbb{R}^d$ [等式(3)]:

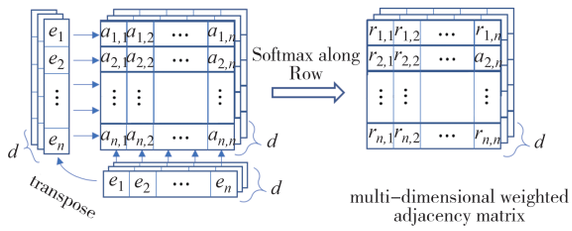

图 4 事件链 $\boldsymbol{E}=\{e_1, e_2, \cdots, e_n\}$ 生成多维权重邻接矩阵 $\boldsymbol{A}$ 的示例

Fig.4 Example for generating the multi-dimensional weighted adjacency matrix $\boldsymbol{A}$ for event chain $\boldsymbol{E}=\{e_1, e_2, \cdots, e_n\}$

$$\boldsymbol{a}_{i,j} = W_a^{\top}\tanh\left(W_1 e_i \odot W_2 e_j\right) \tag{3}$$

其中 W_a, W_1, $W_2 \in \mathbb{R}^{d\times d}$, d 是事件嵌入的维度,$\odot$ 表示逐元素乘法.然后沿着对齐得分矩阵 $\boldsymbol{a}_{i,j}$ 的行执行 softmax 以获得归一化强度 $r_{i,j}$.它们在第 k 维的关系强度 $r_{i,j}^k$ 可以建模为等式(5).

$$\boldsymbol{a}_{i,j} = \left[a_{i,j}^1, a_{i,j}^2, \cdots, a_{i,j}^d\right]^{\top} \tag{4}$$

$$r_{i,j}^k = \frac{a_{i,j}^k}{\sum_j a_{i,j}^k},\ \sum_j r_{i,j}^k = 1 \tag{5}$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{r}_{1,1} & \cdots & \boldsymbol{r}_{1,n} \\ \vdots & & \vdots \\ \boldsymbol{r}_{n,1} & \cdots & \boldsymbol{r}_{n,n} \end{bmatrix} \tag{6}$$

最终的多维加权邻接矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n\times d}$ 为等式(6).图 4 说明了如何生成多维邻接矩阵.在本文的其余部分,为了便于理解,本文将忽略 $\boldsymbol{A}$ 中的特征维度 k,并将 $\boldsymbol{A}$ 视为 $n\times n$ 矩阵.

2.3 稠密事件图嵌入

对于脚本事件预测,我们往往需要进行多步推理才能得出正确的答案.例如,“他打开冰箱”可以引发多个合理的后续事件,例如“他得到一瓶水”或“他得到一片面包”.然而,如果我们知道发生在“他打开冰箱之前的事件”,例如“他渴了”,那么可能的选择就会缩小到“他得到一瓶水”.整合来自与目标事件间接相关的事件的信息也将有助于完成任务.然而,邻接矩阵只保留一阶关系,即直接连接的事件的关系.

图 5 给出了一个简单的示例,解释了什么是一阶和二阶关系.从节点 1 到节点 2 是一阶关系,因为它们直接相连,从节点 1 到节点 3、4 和从节点 5、6、7 到节点 2 是二阶关系,因为节点 1 需要穿过节点 2 才能到达节点 3、4.图 5 只显示了一阶、二阶和三阶关系,然而,这个概念可以很容易地扩展到更高阶.尽管在本文提出的事件图中的节点是密集连接的,但高阶信息仍然有帮助,因为它允许在事件 e_j 和 e_k 存

在的情况下计算出事件e_i发生的可能性,即条件概率$p(e_i|e_j, e_k)$.

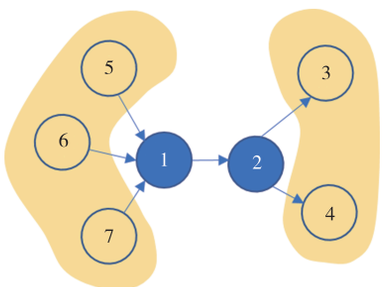

图 5　展示事件图中一阶和二阶关系的一个简单示例

Fig.5　An toy example illustrating the first-order and second-order relations in the event graph

在获得链中的多维加权邻接矩阵 $\boldsymbol{A}$ 和事件表示 $\boldsymbol{E}$之后,下一步是学习一个利用 $\boldsymbol{A}$、$\boldsymbol{E}$ 将上下文信息集成到事件(节点)表示中的函数 f, 即 $\hat{\boldsymbol{E}} = f(\boldsymbol{A}, \boldsymbol{E})$. 图 6 显示了 DEG 嵌入模块中的单个层. 在此详细介绍本文提出的 DegNet-G 和 DegNet-R 如何获得以下最终事件表示,图 6 中 $\boldsymbol{A}$ 为多维加权邻接矩阵,$\boldsymbol{E}^t$ 为当前事件表示.

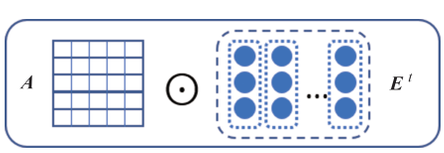

图 6　DEG 嵌入模块中的单层

Fig.6　A single layer in the DEG embedding module

2.3.1　DegNet-G

DegNet-G 涉及常用的门控图神经网络(GGNN)[11]用于嵌入图,如等式(7)中所定义.

$$\begin{aligned} a_t &= \boldsymbol{A}^{\top}\boldsymbol{E}^{t-1} + b \\ z_t &= \sigma\left(W_z a_t + U_z \boldsymbol{E}^{t-1}\right) \\ r_t &= \sigma\left(W_r a_t + U_r \boldsymbol{E}^{t-1}\right) \\ c_t &= \tanh\left(W a_t + U(r_t \odot \boldsymbol{E}^{t-1})\right) \\ \boldsymbol{E}^t &= \left(1 - z_t\right) \odot \boldsymbol{E}^{t-1} + z_t \odot c_t \end{aligned} \tag{7}$$

其中 σ 和 tanh 是非线性激活函数;$W_{(*)}$, $U_{(*)}$ 是可学习的模型参数;$\odot$ 表示逐元素乘法,r_t 和 z_t 分别表示重置门和更新门. 基于门机制,GGNN 能够结合来自当前状态 $\boldsymbol{E}^t$ 和先前状态 $\boldsymbol{E}^{t-1}$ 的信息.

GGNN 依赖于关于图中节点关系的先验知识. 为了适应本文的框架,采用 MWAMA 替换了它原来的邻接矩阵,并添加了 Layer-wise Normalization 层来捕获不同顺序的信息.

2.3.2　DegNet-R

DegNet-R 基于随机游走理论[21]. 对于具有转移矩阵 $\boldsymbol{M}$ 的连通图 G, $\boldsymbol{M}$ 定义图中从一个节点到另一个节点的转移概率. 从节点 v^0 开始的 t 步随机游走,可以生成一个序列 $\{v^0, v^1, v^2, \cdots, v^t\}$. 在这个随机游走中可以到达的第 t 个节点 v^t 的分布 $P_{(v^t)}$ 被定义为 $P(v^t) = (\boldsymbol{M}^{\top})^t P(v^0)$.

在本文的例子中,MDWMAA 定义了从一个事件到另一个事件的转移概率. 因此,可以通过 $(\boldsymbol{A}^{\top})^t$ 到达第 t 阶节点. 换句话说,影响可以从第 t 个节点传播到起始节点.

为了从一阶邻接点那里获得信息,有 $\boldsymbol{E}^1 = \boldsymbol{A}^{\top}\boldsymbol{E}^0$. 从这个角度来看,只考虑一阶节点相当于 transformer[18]中的 self-attention 机制,其中一个 token 考虑了序列中的所有 token 来更新自己的表示. 我们可以将这个概念扩展到更高阶. 考虑一阶邻接点的一阶邻接点,即二阶节点 $\boldsymbol{E}^2 = \boldsymbol{A}^{\top}(\boldsymbol{A}^{\top}\boldsymbol{E}^0)$, 以此类推. 为了从第 t 阶节点获取信息,可依据等式(8)运算:

$$\boldsymbol{E}^t = \boldsymbol{A}^{\top}\boldsymbol{E}^{t-1} = \boldsymbol{A}^{\top}\left(\boldsymbol{A}^{\top}\boldsymbol{E}^{t-2}\right) = \left(\boldsymbol{A}^{\top}\right)^t \boldsymbol{E}^0 \tag{8}$$

其中 $\boldsymbol{E}^t$ 代表第 t 阶节点,其他以此类推.

2.3.3　*求和和分层归一化*

为了整合不同顺序节点的信息,本文对每一层输出求和并执行逐层归一化:

$$\hat{\boldsymbol{E}} = \sum_{0}^{t} \boldsymbol{E}^t \tag{9}$$

其中 $\boldsymbol{E}^t$ 代表第 t 阶节点,它也可以看作是来自每一层输出的跳跃连接[22]. 可以直接从最后一层传播梯度来更新每一层,从而避免梯度消失.

本文通过堆叠 DEG 嵌入模块来捕获更高级别的表示. 在同一 DegNet 模块的层中,$\boldsymbol{A}$ 在同一 DEG 嵌入模块之间共享. 它将根据最新的上下文感知事件表示在传递到上层堆叠模块时进行更新.

2.4　预测

在获得上下文中每个事件的最终上下文感知表示 $\hat{\boldsymbol{E}} = \{\hat{e}_1, \hat{e}_2, \cdots, \hat{e}_m\}$ 后,就可以从候选事件 $\{c_1, c_2, \cdots, c_n\}$ 中预测后续事件. 首先计算每个候选事件与上下文中所有事件的相关性分数:

$$\text{score}\left(c_i\right) = \frac{1}{m}\sum_{j=1}^{m} g\left(c_i, \hat{e}_j\right) \tag{10}$$

其中 c_i 表示第 i 个候选事件,$\hat{e}_j$ 表示第 j 个事件的上下文感知表示,$g(\cdot, \cdot)$ 表示计算相关性分数的函数. 候

选事件的最终得分是其与所有上下文事件的相关性得分的平均值.相关性得分最高的候选事件视为正确答案.在此本文使用简单的余弦相似度来衡量候选事件和上下文事件之间的相关性.得分最高的答案将被视为预测结果c:

$$c = \arg\max_{c_i}\left(\text{score}\left(c_i\right)\right) \tag{11}$$

2.5 训练细节

本文使用multiple margin loss作为目标函数来优化.正确的黄金标签应该获得更高的分数以避免惩罚:

$$\frac{1}{m}\sum_{m}\sum_{i\neq c_{\text{gold}}}\max\left(0,\text{margin}+\text{score}\left(c_i\right)-\text{score}\left(c_{\text{gold}}\right)\right) \tag{12}$$

其中c_{gold}表示黄金标签对应的候选事件,margin为超参数.128维的词向量是从Li[17]等人发布的预训练词嵌入中初始化的.它在大型新闻专线语料库上进行训练,并在模型训练期间进行了微调.本文使用Adam[23]来优化参数,其初始学习率设置为0.000 5的参数.本文也使用了早停法(early stopping)和随机失活(dropout)[24]来避免过拟合问题,dropout率设置为0.1.本文保留在开发数据集上具有最佳性能的模型,以供在测试数据集上进一步评估.

3 实验

本文在两个脚本事件预测数据集上评估了所提出的模型,即多选叙事完形填空(MCNC)和连贯多选叙事完形填空(CMCNC).

3.1 数据集

数据集统计如表1所示.

表1 数据集统计

Tab.1 Statistics of the dataset

	训练集	开发集	测试集
#document	830,643	103,583	103,805
#chains	140,331	10,000	10,000

3.1.1 多选叙事完形填空(MCNC)

该任务由Granroth-Wilding等人[2]提出,用于评估叙事生成系统.数据集是根据Gigaword语料库的纽约时间部分自动构建的[25].为了保证事件链的连贯性,同一条链中的事件共享同一个主角.上下文中有八个事件,提供了丰富的信息.在五个候选项目中,只有一个是正确的.

本文遵循与Granroth-Wilding等人[2]相同的管道方法进行事件提取和事件链构建.使用相同的文件进行训练、评估和测试,以便与以前的工作进行公平比较[2,17].数据集的总体统计数据如表1所示.

3.1.2 连贯多选叙事完形填空(CMCNC)

MCNC数据集是自动构建的,它对常见事件(例如"说过"、"做过")和预处理工具中的错误很敏感.为了解决这个问题,Weber等人[6]手动构造了一个评估数据集CMCNC.CMCNC与MCNC的区别在于:MCNC的长度和所有的事件共享同一个主角.

3.2 基线和评估指标

本文与以下基线进行比较.遵循之前的工作[2,17],使用准确性作为评估指标.

Averaging:该模型将每个事件表示为来自预训练的GloVe嵌入的组成词向量的平均值[26].选择成对相似性得分最高的候选事件.

PMI:基于共现的模型[1].事件用依赖关系和谓词动词表示.每个候选事件的分数是通过将它们的逐点互信息分数与上下文事件相加来计算的.

Bigram:基于skipgram[3]和二元概率对事件对关系进行建模.

LSTM:一个基于链的模型[7],它利用长短期记忆网络来表示事件链.候选事件用于网络的最后输入,输出隐藏状态用作预测的特征.

EventComp:利用成对事件连接[1].它是一个非线性事件组合网络,它使用孪生网络来计算成对事件相关性分数.

Predicate Tensor:通过将事件参数与谓词张量结合来获得事件表示[6].张量是根据事件中的谓词动态生成的.

Role Factor Tensor:使用两个张量分别捕获主谓和谓宾之间的交互[6],然后将它们组合起来形成事件表示.

PairLSTM:使用动态记忆网络[8].首先利用递归神经网络来捕获事件顺序信息,然后将候选事件视为寻找线索以预测后续事件的查询.

SGNN:构建了一个抽象的叙事事件进化网络作为定义事件之间联系的先验知识[8],然后应用门控图神经网络对事件交互进行建模.

3.3 结果与分析

3.3.1 MCNC 上的结果

MCNC 数据集的总体实验结果如表 2 所示,最好的结果被加粗表示.可得到以下观察结果:

首先,基于计数的方法 PMI 和 Bigram 与基于词嵌入的方法相比没有竞争力,因为它们依赖于稀疏特征表示和成对关系,这会丢失事件之间的结构和语义信息.

其次,在任务中引入词嵌入带来了显著的改进.然而,EventComp、LSTM 和 PairLSTM 的实验结果之间的比较表明,单靠词嵌入是不够的,事件链中的事件关系如何建模也很重要.

再次,DeepWalk 和 SGNN 都是基于图嵌入的方法,但 DeepWalk 是无监督的,而 SGNN 是有监督的.两者之间巨大的性能差距凸显了有监督方法的重要性.

最后,DegNet-G 和 DegNet-R 都大大优于最佳基线,最佳模型实现了 4.6% 的性能提升.这表明利用来自高阶节点的密集连接和信息有助于模型理解上下文中的事件.

表 2　基线和本文的方法在 MCNC 测试集上的表现

Tab.2　The performances of baselines and our approach on the MCNC test set

方法	准确率/%
Random	20.00
PMI	31.25
Bigram	29.67
DeepWalk	43.01
LSTM	46.75
EventComp	49.66
PairLSTM	50.34
SGNN	52.45
DegNet-G(本文的方法)	54.68
DegNet-R(本文的方法)	**54.86**

3.3.2 CMCNC 上的结果

表 3 报告了本文在 CMCNC 测试集上的结果,最好的结果被加粗表示.表 3 仅给出了本文提出的模型和基线在 CMCNC 测试集上的性能.

观察结果与 MCNC 类似,即基于计数的方法 Bigram 的性能大幅低于基于神经网络.Predicate Tensor、EventComp 和 Role Factor Tensor 是基于成对关系的方法,旨在学习更好的事件表示.Role Factor Tensor 使用两个张量来捕获主谓和谓宾交互,在数据集上提供更高的准确性并显示其语义表示能力.LSTM、PairLSTM 是基于链的方法,而 PairLSTM 考虑成对关系和链时序.SGNN 和本文提出的 DegNet-G、DegNet-R 是基于图的方法.

表 3　基线和本文的方法在 CMCNC 测试集上的表现

Tab.3　The performances of baselines and our approach on the CMCNC test set

方法	准确率/%
Random	16.7
Averaging	26.7
Bigram	55.8
Predicate Tensor	66.1
LSTM	67.7
EventComp	68.5
PairLSTM	71.3
Role Factor Tensor	72.1
SGNN	72.1
DegNet-G(本文的方法)	72.9
DegNet-R(本文的方法)	**74.5**

3.3.3 模型融合

为了找出不同的结构(例如对、链、稀疏图)是否具有其独特的优势,本文进行了模型融合.使用简单的参数化线性组合来集成来自不同模型的预测分数(等式 13).在开发集上对超参数 λ 进行了调整.

$$c = \arg\max_{c_i}\left(\lambda \cdot \text{score}_{\text{model2}}(c_i) + (1-\lambda)\cdot \text{score}_{\text{model1}}(c_i)\right) \tag{13}$$

如表 4 所示,可以发现融合模型 DegNet-R+EventComp 可以将性能从 54.86% 提升到 56.21%,表明它们具有互补的作用.然而,添加 PairLSTM 并没有带来显著的改进,表明两个模型的预测由于结构上的相似性而有很大的重叠.通过组合 DegNet-R、EventComp 和 SGNN 实现了 56.53% 的最佳性能.然而,添加 SGNN 带来的改进很小,这证实了本文的直觉,因为 SGNN 也是基于图结构的.

表 4　融合模型在 MCNC 测试集上的表现

Tab.4　The performances of fused models on MCNC test set

方法	准确率/%
DegNet-R + EventComp	56.21
DegNet-R + PairLSTM	54.94
DegNet-R + SGNN	55.57
DegNet-R + EventComp + SGNN	**56.53**

3.3.4 层数的影响

DEG Embedding 模块可以有多个层来捕获不同顺序的事件关系.为了验证添加层是否对性能有影响,使用不同的层训练模块并测试它们的性能.结果如图7所示.

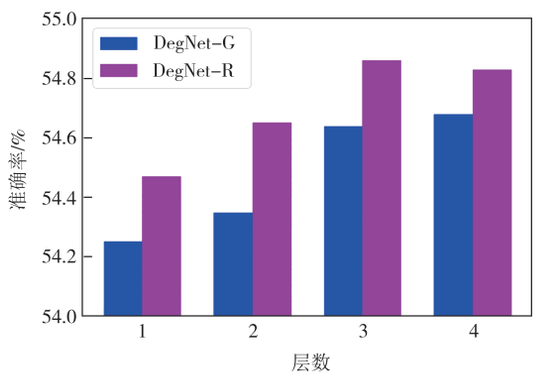

图 7 不同层数的 DegNet-G 和 DegNet-R 在 MCNC 数据集上的性能

Fig.7 The performances of DegNet-G and DegNet-R with different numbers of layers on MCNC dataset

可以观察到,具有1层的DegNet-R性能已经比最佳基线SGNN提高了近4%.随着层数的增加,性能会提高,这表明它们可以从更高阶获取信息.然而,我们观察到当层数增加到4层时,模型的性能开始出现下降的趋势.这表明随着层数的增加,网络模型的参数和计算复杂度不断提高,而数据集规模有限,会导致过拟合问题,使得模型效果下降.比较的结果表明,本文的模型可以利用多阶信息来提高模型性能.

3.3.5 事件表示的影响

Average、Concatenation 和 Nonlinear 是之前工作中使用 Average 最广泛的事件表示函数[2,8,17].Average 对所有向量执行均值运算,Concatenation 将所有向量拼接成单个向量,而 Nonlinear 通过非线性参数化加法导出事件向量.本文与上述事件表示方法进行了比较,结果如表5所示.

表5 不同事件表示模式在MCNC数据集上的比较

Tab.5 Comparison between different event representation schemas on MCNC dataset

方法	准确率/%
Average	47.02
Sum	46.88
Nonlinear	53.70
Concatenation	**54.86**

朴素的Sum和Average在事件组合策略以及它们在测试数据集上的最终表现上均没有显著差异.与Sum和Average相比,基于参数化加法和非线性激活的Nonlinear在预测精度上有显著提高,显示了建模事件语义在文本理解任务中的重要性.本文使用的Concatenation方法产生了最佳性能.一个可能的解释是事件元组中有很多缺失的字段,该方法连接而不是将动词和参数嵌入添加到单个向量中,从而减少了这些字段对最终事件表示的影响.

3.3.6 密集和稀疏事件图的影响

为了测试本文提出的多维加权邻接矩阵的有效性,将基线SGNN进行复制并用MWAM替换了模型中的邻接矩阵.原始的SGNN模型利用自动构建的动词-动词图来定义链中的事件-事件关系.从表6中的结果可以发现,用DEG替换稀疏事件图实现了近2%的性能提升,表明本文提出的MWAM已具有捕获事件之间关系的能力.

表6 SGNN和本文的DegNet-R模型在MCNC数据集上的稀疏稠密事件图的表现

Tab.6 Performances of SGNN amd our DegNet-R with sparse and dense event graph on MCNC dataset

方法	准确率/%
SGNN (sparse)	52.45
SGNN (dense)	53.31
DegNet-R (sparse)	52.93
DegNet-R (dense)	**54.86**

尽管DEG中的事件是完全连接的,但是可能会引入不存在的关系.然而,事件之间的关系强度是相对的,不存在的关系将被分配相当小的值,本文提出的MWAM能够从细粒度的特征级别捕获关系强度.对稠密事件图有效性的另一种解释是,词嵌入被训练来预测其上下文.词向量已经包含了上下文事件.通过嵌入词向量的语义组合来衡量事件之间的关系是合理的.

3.4 案例分析

为了进一步分析本文的方法在数据集上的性能,在表7中展示了一个案例研究.表7中√代表正确答案,×代表错误答案.

对于案例1,人类可以很容易地从上下文事件中推断出C_3是正确答案,但是对于模型来说,它已经将音乐家与钢琴家联系起来.

表 7　CMCNC 数据集的实例研究

Tab.7　Case study for CMCNC dataset

案例 1	**Context**: (mother, as, pianist) ⇒ (mother, decided teach, cello) ⇒ (father, cautioned practice, much) ⇒ ?
0.105	C_1: (work, plagued by, problems)
0.113	C_2: (much, tilt toward, restraint)
0.305(√)	**C_3: (he, become, musician)**
0.110	C_4: (it, stopped in, bodrum)
0.133	C_5: (parents, found in, office)
0.234	C_6: (sonata, wonderful in, containers)
案例 2	**Context**: (parisians, arrested on, charges) ⇒ (man, arrested in, subway) ⇒ (he, collector of, knives) ⇒ (he, carried as, weapon)⇒ ?
0.179	C_1: (nowhere, moved from, county)
0.132	C_2: (process, scheduled for, dec.)
0.096	C_3: (you, name on, list)
0.176	C_4: (mansour, spokesman of, party)
0.252(√)	**C_5: (knife, qualify as, illegal)**
0.165	C_6: (knife, weighted in, favor)
案例 3	**Context**: (runyan, finished, marathon) ⇒ (women, led by, samuelson) ⇒ (women, run, marathon) ⇒ (runyan, queasy after, race) ⇒ ?
0.081	C_1: (samuelson, wrote in, ms.)
0.091	C_2: (women, reaching by, 1991)
0.119	C_3: (man, using in, someone)
0.243(√)	C_4: (runyan, won, titles)
0.249(√)	**C_5: (chance, beat for, victory)**
0.222	C_6: (chance, said to, lou)

对于案例 2,候选 C_6 和 C_5 共享同一主题实体 *knife*,但 C_5 给出的分数比 C_6 高,表明事件中的其他参数也起着重要作用.这也表明本文的模型在这种情况下捕获了事件的语义.

对于案例 3,这是一个错误案例,本文的模型给出了错误的答案 C_5,而正确的答案是 C_4,得分略高.其他不相关的答案,如 C_1 和 C_2 得分很低.然而,本文提出的模型还是能够将 *victory* 和 *race* 的概念联系起来.尽管 C_4 是人类注释的正确答案,但 C_5 也显示出一定合理性.

从上述案例研究中,可以发现信息抽取仍然是一个很大的挑战,这限制了从非结构化文本中自动获取知识的效果.

4　结论

本文提出了一种稠密事件图嵌入模型,该模型利用稠密事件图进行脚本事件预测,使用多维加权邻接矩阵来表示事件之间的丰富联系及其在稠密事件图中的关系强度,该矩阵通过从数据中学习来更新.同时本文提出了一个通用框架,它可以将高阶事件演化信息集成到事件表示中.在 benchmark 数据集上的实验结果证明了本文所提出模型的有效性.

参考文献

[1] CHAMBERS N, JURAFSKY D. Unsupervised Learning of Narrative Event Chains[C]// ACL 2008 Meeting of the Association for Computational Linguistics. Columbus, Ohio, USA: DBLP, 2008.

[2] GRANROTH-WILDING M, CLARK S. What happens next? event prediction using a compositional neural network model[C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2016: 2727 - 2733.

[3] JANS B, BETHARD S, VULIĆ I, et al. Skip n-grams and ranking functions for predicting script events[C]//Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. New York: ACM, 2012: 336-344.

[4] DING X A, ZHANG Y E, LIU T, et al. Using structured events to predict stock price movement: an empirical investigation [C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) Doha, Qatar. Stroudsburg, PA, USA: Association for Computational Linguistics, 2014: 1415-1425.

[5] DING X, ZHANG Y, LIU T, et al. Knowledge-driven event embedding for stock prediction[C]// COLING 2016. 2016: 2133-2142.

[6] WEBER N, BALASUBRAMANIAN N, CHAMBERS N. Event representations with tensor-based compositions [C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2018.

[7] PICHOTTA K, MOONEY R. Learning statistical scripts with LSTM recurrent neural networks [C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2016: 2800-2806.

[8] WANG Z Q, ZHANG Y E, CHANG C Y. Integrating order information and event relation for script event prediction [C]// Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, USA: Association for Computational Linguistics, 2017: 57-67.

[9] GORI M, MONFARDINI G, SCARSELLI F. A new model for learning in graph domains [C]//Proceedings of 2005 IEEE International Joint Conference on Neural Networks. Montreal, QC, Canada: IEEE, 2005: 729-734.

[10] SCARSELLI F, GORI M, TSOI A C, et al. The graph neural network model [J]. IEEE Transactions on Neural Networks, 2009, 20(1): 61-80.

[11] LI Y J, TARLOW D, BROCKSCHMIDT M, et al. Gated graph

sequence neural networks[EB/OL]. 2015: arXiv: 1511.05493. https://arxiv. org/abs/1511.05493.

[12] CHO K, VAN MERRIENBOER B, GULCEHRE C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[EB/OL]. 2014: arXiv: 1406.1078. https://arxiv. org/abs/1406.1078.

[13] KIPF T N, WELLING M. Semi-supervised classification with graph convolutional networks [EB/OL]. 2016: arXiv: 1609.02907. https://arxiv. org/abs/1609.02907.

[14] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks[EB/OL]. 2017: arXiv: 1710.10903. https://arxiv. org/abs/1710.10903.

[15] TANG J, QU M, WANG M Z, et al. LINE: large-scale information network embedding [C]//Proceedings of the 24th International Conference on World Wide Web. New York: ACM, 2015: 1067-1077.

[16] PEROZZI B, AL-RFOU R, SKIENA S. DeepWalk: online learning of social representations [C]//Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. New York: ACM, 2014: 701-710.

[17] LI Z Y, DING X, LIU T. Constructing narrative event evolutionary graph for script event prediction[EB/OL]. 2018: arXiv: 1805.05081. https://arxiv. org/abs/1805.05081.

[18] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all You need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: ACM, 2017: 6000-6010.

[19] TSVETKOV Y, FARUQUI M, LING W, et al. Evaluation of word vector representations by subspace alignment[C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, USA: Association for Computational Linguistics, 2015: 2049-2054.

[20] SHEN T, ZHOU T Y, LONG G D, et al. DiSAN: directional self-attention network for RNN/CNN-free language understanding[C]// Proceedings of the AAAI Conference on Artificial Intelligence.2018.

[21] LOVÁSZ L. Random walks on graphs: a survey[D]. New Haven, Connectiovt, USA: Yale University, 1996.

[22] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition [C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, 2016: 770-778.

[23] KINGMA D P, BA J. Adam: a method for stochastic optimization [EB/OL]. 2014: arXiv: 1412.6980. https://arxiv. org/abs/1412.6980.

[24] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: a simple way to prevent neural networks from overfitting [J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[25] GRAFF D, KONG J, CHEN K, et al. English gigaword [J]. Linguistic Data Consortium, 2003, 4(1): 34.

[26] PENNINGTON J, SOCHER R, MANNING C. Glove: global vectors for word representation [C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg, PA, USA: Association for Computational Linguistics, 2014: 1532-1543.