

研究综述 | 事件抽取及推理 (下)

Original 李晨光 [知识工场](#) 今天

上篇事件抽取及推理的推文已经介绍了事件抽取的基本方法，本篇主要介绍事件推理的相关工作。就目前来看，事件方向相关的研究还是以事件抽取为主流任务，当前大多都是在模型的框架和优化方面进行研究。同时，事件推理的相关工作有众多领域并且没有一个主流的事件推理方向。近年来，越来越多的工作关注于事件的各个方面的推理，比如事件因果关系推理 (Event causality inference), 脚本事件推理 (Script event prediction), 常识级别事件产生的意图和反应推理 (common sense level event intent , reaction inference), 周期性事件时间推理 (recurrent event inference) 等等。这些任务大多都是在已经有抽取到的事件语句（或从知识库中可以得到的事件片段等），挖掘事件在时间维度、顺承或依赖关系维度的特征，从文本或者知识库中获取先验知识来支撑推理。这些事件推理的工作不外乎推理事件之间的相互关系，比如时序关系、因果关系等等。传统的方法都是基于统计的方法，现在的方法也基本都是以神经网络为主。接下来就事件推理领域的一些工作进行介绍。

脚本事件推理

脚本事件学习已经经过了多年的研究，主要任务是根据一系列的故事的上下文来推断出正确的故事结尾，当然推断结尾一般也是常识级别的因果关系推断 (common sense level inference) 。最早由 Chambers and Jurafsky 等人提出基于指代消解技术利用自动化的方法从大规模语料中自动抽取脚本事件链条 (event chain) ，并提出了经典的挖词填空式的评估标准来评估脚本事件推断模型，这种思路被称为统计脚本学习。之后，脚本事件学习和预测领域有了更多的工作 A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories [NAACL 2016] 提供了事件语料和evaluation方法。现在的工作大多都是使用基于共现关系的概率统计的相关方法来描述抽象脚本事件之间的演化规律和模式。更多的，Mark Granroth-Wilding, etc. 2016 的工作提出了 Multiple Choice Narrative Cloze (MCNC) 的任务以及评估方法，方便了后续研究的开展。这个任务具体可以描述为：给定一个脚本事件的上下文和若干个候选的后续事件，从这些后续事件中挑选唯一正确的后续事件。这篇文章中的一个例子可以用下图来阐述。

Entities

x_0 = Giardino x_1 = chairman, him

Context (e_i)

die(x_0), attend(x_0 , reunion), specialize(x_0 , as partner),
describe(x_0 , x_1 , as product), hold(x_0 , position),
appoint(-, x_0 , to the board), lead(x_0 , effort),
improve(x_0 , operation), propose(x_0 , cut), play(x_0 , role),

c_1 : receive(x_0 , response)
 c_2 : drive(x_0 , mile)
 c_3 : seem(x_0)
 c_4 : discover(x_0 , truth)
 c_5 : **modernize(x_0 , procedure)**

?

如上图所示，上下文被组织成若干个脚本事件组成的 event chain，每个脚本事件抽象成为 trigger (role1, role2,...) 的形式，角色都在 entities 中进行标注。然后在预测部分给出五个候选事件，这五个候选事件是由一个标注的正确的下文事件和另外的从语料中随机选择的事件组成。

同时，也有很多工作（比如 Story comprehension for predicting what happens next [EMNLP 2017], Dan Roth 研究组的工作）开始关注语义方面的故事理解，不仅仅是在传统的基于统计的方法来进行预测，需要更深层次挖掘文本的语义知识，比如事件序列（通常可以挖掘出事件顺承和依赖关系的信息）、情感偏向（通常奠定预测事件的情感倾向）和主题信息（保持预测的事件和故事原文的主题一致性）从而去预测接下来可能发生的事件。我们以以下的例子来说明根据已有的故事原文来进行下文的预测：

Context: One day Wesley's auntie came over to visit. He was happy to see her, because he liked to play with her. When she started to give his little sister attention, he got jealous. He got angry at his auntie and bit her hand when she wasn't looking.

Incorrect Ending: She gave him a cookie for being so nice.

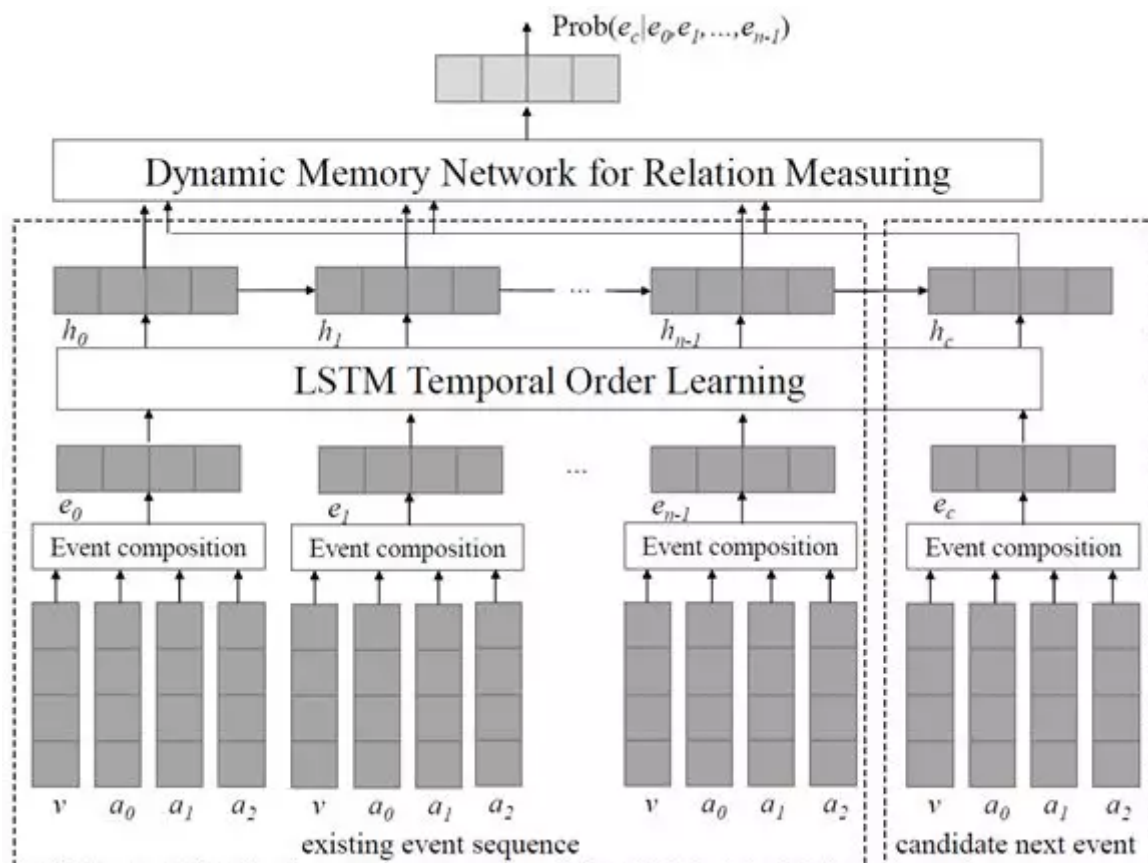
Correct Ending: He was scolded.

(图片来源于 Chaturvedi S, etc. 2017)

这篇工作就是从 context 中提取出来故事的语义信息进行故事结尾推断。

传统的脚本事件学习方法 Chambers and Jurafsky (2008) 使用 Pairwise Mutual Information (PMI) 来将事件通过事理关系形成一个事件链，之后研究者 Jans et al. (2012) 用 skip bi-gram 概率来从文本中提取出来事件链，从而对事件对的时序关系进行建模，这些事件对 (event pair) 往往具有较强的顺承关系。上述的统计方法逐渐被神经模型替代，越来越多的研究者使用 Siamese Network, language model 和 LSTM 等模型来学习到更多的事件时序关系。

介绍完脚本事件推理的相关基础知识之后，接下来就结合近些年具体的工作来描述脚本事件推理的研究问题和现状。Wang Z, etc 2017 提出了结合事件时序信息和事件关系来辅助脚本事件预测。这篇工作主要是总结了以往工作中往往事件对的关系 (pairwise event relations) 是对脚本事件预测很有帮助的，因此在使用神经网络对整个事件链进行时序关系建模的同时应该考虑进行这些事件对的关系信息。整个方法的框架如下所示：



(图片来自Wang Z, etc 2017)

整个框架分两部分：已有的脚本事件链中的每个事件被组织成 $v (a_0, a_1, a_2)$, embeddings 之后拼接成为一个 event vector e ，然后使用 LSTM 来 encode，使用 LSTM 主要是为了保留事件之间的时序关系，这样就完成了对已有事件链的表示。同时，对五个 candidate events，每个 candidate 都以相同的方式进行 encode，并且这个候选事件的

隐状态需要结合事件链最后一个事件的 hidden layer 来确定，为了尽可能地学习更多的 event pair 信息，candidate event 的隐状态会和事件链的每个事件的隐状态进行配对，并通过 dynamic memory network 来学习候选事件和这些事件链中的事件的相关性，最终输出每个 candidate event 的概率结果。

详细地对每个部分展开来说，事件使用 LSTM 表示部分比较通用因此不必详细说，工作重点在于对事件对的建模。首先，传统的方法计算一个事件对的相关程度可以用 sigmoid 简单计算如下：

$$s_i = \text{sigmoid}(W_{si}h_i + W_{sc}h_c + b_s)$$

h_i 是事件链中的某个事件的 LSTM 得到的表示， h_c 是候选事件表示。则自然地，最终的这个候选事件作为最终预测事件的得分就是所有 event pairs 的平均分数。然而直觉上事件链上的每个事件对于候选事件预测的重要程度都不是均等的，比如事件链上的一系列事件“排队，就坐，点餐”和最终的预测事件“吃饭”，很明显“就餐”事件和“吃饭”的相关性最高。因此这部分使用了 dynamic memory network 进行了对传统方法的改进。

首先作者使用简单的 attention 机制来对事件链中每个事件对于候选事件的影响程度进行权重计算。公式如下左边部分，attention 的一般形式，权重就是 α_i

。则每个事件和候选事件的相关性计算就变成了下面公式2的形式。

$$u_i = \tanh(W_{ei}h_i + W_{ec}h_c + b_u)$$

$$\alpha_i = \frac{\exp(u_i)}{\sum_j \exp(u_j)}$$

(公式1)

$$s = \sum_{i=1}^{n-1} \alpha_i \cdot s_i$$

(公式2)

原文作者进一步使用 memory model 来对事件权重计算进行改进。这里也介绍一下 memory model，memory model 由多个 dynamic computational layer 组成，第一层的权重计算和上面的一般 attention 权重计算形式一样。之后，对整个事件链学习到一个统一的表示，即对每个事件链中的事件的隐状态都通过第一层的权重进行加权相加，公式如下所示：

$$h_e = \sum_{i=1}^{n-1} \alpha_i \cdot h_i$$

h_e 就是学习到的整个 event chain 的表示。这个统一的事件链表示 h_e 就被用来和候选事件隐状态 h_c 结合起来产生一个更深层次的表示 v ，这个表示 v 比起候选隐状态 h_c 来说包含了更多的语义信息。在计算第二层 attention 的时候 h_c 就可以使用这个更深层次的表示 v 来进行，即：

$$u_i^t = \tanh(W_{ei}h_i + W_vv^t + b_u)$$

$$\alpha_i^t = \frac{\exp(u_i^t)}{\sum_j \exp(u_j^t)}$$

这个过程将会进行多次。然后用 gate 机制来控制使用当前新得到的表示还是上一个阶段的 v 。最后通过检测 $|v_{t+1} - v_t|$ 是否小于阈值来确定整个模型是否已经收敛。

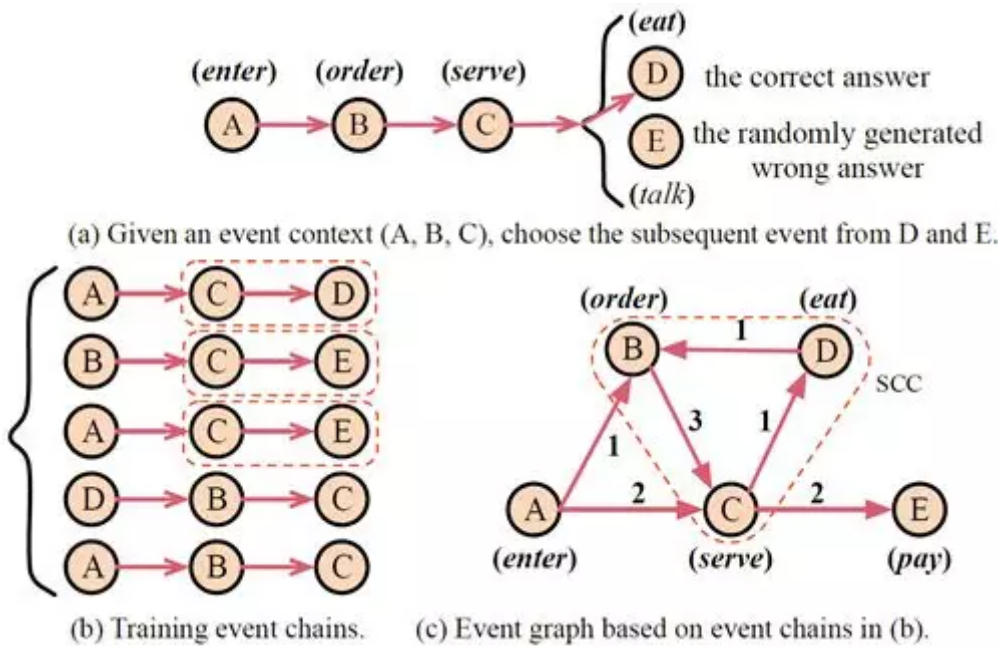
总而言之，这篇文章算是将比较经典的脚本事件学习方法进行对比，并将两种主流的事件关系，事件时序关系和事件对的关系来进行融合，结合两种关系的优点，使用 memory model 进行事件链的每个事件的权重学习，并在数据集上获得了 state-of-art 的结果。本篇就是详细介绍了这一工作和当前的最新研究进展。

事理图谱与脚本事件推理

事理图谱近些年来收到越来越多的关注，构建事理图谱可以进行很多应用，这部分就介绍一篇近年的构建事理图谱来进行脚本事件推理的工作。Li Z, etc.（本章节所有图片都来自于这篇文章）在去年的关于构建事理图谱来进行脚本事件预测的工作就是典型的事理图谱应用工作，事理图谱也应该受到更广泛的关注。这篇文章是哈工大刘挺老师团队的工作，其在事理图谱方面的研究有很长时间。根据刘挺老师有关事理图谱的ppt中可以得到事理图谱的相关背景知识：事理图谱是事理逻辑知识库，描述抽象事件之间的演化规律和模式。事理图谱以抽象类事件为节点构建的有向有环图（directed cyclic graph），node 是 event，edge 是事理关系。事理关系在刘老师这里暂时只定为两种：一种顺承，一种因果，这两种关系考虑了事件发生的时间顺序。

这篇文章就是描述通过构建这样的事理图谱来进行脚本事件预测，同时，通过 scaled graph neural network（SGNN）来进行事理图谱上的事件预测。这篇工作观测到已有的

使用事件链 (event chains) 或者事件对 (event pairs) 的工作会遗漏掉很多事件之间的丰富的关联信息。在传统的基于统计的方法中，共现较多的关系往往被认作为强关联，因此绝大多数情况下都是根据这些统计信息来做出 pairwise 的预测，而忽略了 event 的上下文信息。然而通过已有的事件链来构建事件图谱，一些语义上强联系的事件会形成环等结构，从而在统计方法上加上图结构信息来辅助进行预测。如下图所示的具体例子：

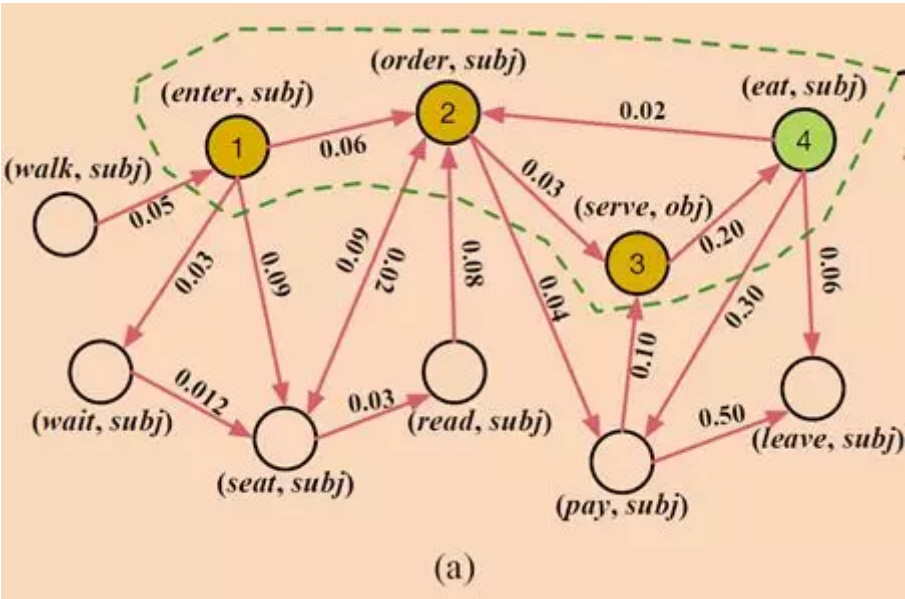


有 ABCDE 五个事件，从训练数据集上我们可以看到基于统计的方法由于 C 和 E 出现的概率比 C 和 D 大，因此这种方法较大概率会预测 ABC 之后的事件是 E 而不是 D。事实上，通过构建事理图谱，我们可以看到 BCD 三个事件会形成一个图谱上的环结构，环结构的强连接意味着 BCD 事件往往是一起进行循环出现的，这是一种隐式的连接信息，在学习事件表示的时候将会更有效地提供更多信息。这里举的例子是数据集上的典型例子（可以直观推理到大多数情况下 pairwise 的事件对信息还是非常重要的），可能并不能代表大多数例子，但是必须承认这种情况存在，然而这种情况存在的比例尚不清楚，因此这里对传统 event chain 和 event pair 的缺点总结尚待优化。但同时我们也可以直观感觉到图谱类型的结构往往能够表达比链和树状等结构更深层次的关联信息，这点是成立的，因此总体来说构建事件图谱的确有助于挖掘隐含事件关系。

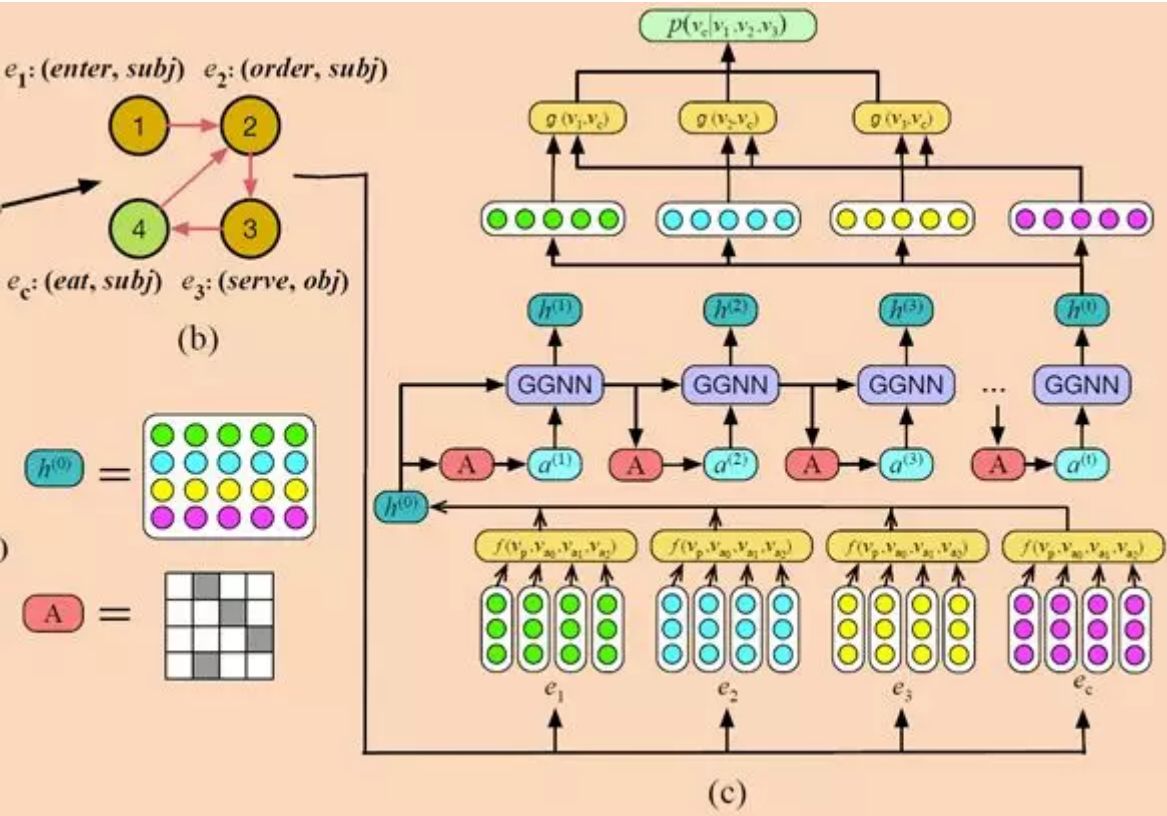
从上图可以看到，事理图谱的点就是事件，有向边表达了事件的时序关系，边上的权重代表的是这种时序关系的出现频率。接下来原文作者就开始关注通过 network embeddings 来学习事件表示，从而完成时间推理。现在比较 popular 的方法是使用 GCNN，即在图谱上使用卷积方法进行特征学习和提取，但是只能作用于无向图。已有的工作已经证明图结构的神经网络如 GNN 能够有效处理这种有向图的结构，学习得事件的时序关系，但是传统的

GNN 方法往往需要信息在图上进行传播从而进行收敛，信息传播的一个问题就是在图上如果距离较远（两点之间的最短路径较长）的情况下，信息很难传播到从而导致事件间信息丢失。这个问题研究者使用了 GGNN（即使用 gate 门的 GNN）方法来尽可能的保留这些需要传播的信息，但这种方法只能在小规模图上进行，本篇论文的原文作者就在这些研究的基础上使用了一个缩放机制，即 scaled graph neural network（SGNN）来使 GGNN 能够作用于大规模事理图谱上。

具体的框架图如下图所示，分为多图展示：



(原文框架图的事理图谱构建部分)



(c)

(原文框架图的图谱缩放和事件图谱表示学习部分)

所以原文方法也大致分为两部分，构建事理图谱和使用 SGNN 来进行事件表示学习。上文已经介绍了这种图谱的节点和边的意思，注意这里的权重是该事件时序关系共现的概率而非频率。SGNN 对比 GGNN 上的优化就是使用分而治之的思想，每次只处理一部分图结构。比如一个完整事件链只包含了节点1,2,3,4，则只把候选事件节点和他们之间的边单独拿出来，如图1的绿色虚线框内。具体地，学习事件的表示分为以下几个阶段：

1、初始化事件表示：相似地，本文需要初始化一个事件为一个事件向量，对于一个 $e = p(a_0; a_1; a_2)$ ， p 为触发词， a 为事件角色，首先 embeddings，然后对于每个部分的 embeddings，本文也比较了三种不同的将 embeddings 结合获得最终的初始化事件表示，分别为 Average（使用每个部分的 embeddings 的平均值作为最后的 event vector），Nonlinear Transformation（

$$v_e = \tanh(W_p \cdot v_p + W_0 \cdot v_{a_0} + W_1 \cdot v_{a_1} + W_2 \cdot v_{a_2} + b)$$

），以及 Concatenation（即同上一篇论文的事件表示方法，直接连接每一个部分的 embeddings 作为最后的 event vector）

2、通过 GGNN 来对事件初始化表示进行更新：这部分就要融入事件的时序信息和共现关系信息。GGNN 的输入是事件链的隐状态矩阵和事件的图谱结构邻接矩阵，其中隐状态矩阵

$$h^{(0)} = \{v_{e_1}, v_{e_2}, \dots, v_{e_n}, v_{e_{c_1}}, v_{e_{c_2}}, \dots, v_{e_{c_k}}\}$$

可以理解为上下文事件向量+候选事件向量， k 一般是5，因为训练集是从5个候选事件中选择一个正确的。经过缩放操作后的事理图谱部分结构的邻接矩阵 A 被表示为

$$A[i, j] = \begin{cases} w(v_j | v_i), & \text{if } v_i \rightarrow v_j \in E, \\ 0, & \text{others.} \end{cases}$$

即如果节点 i 到节点 j 之间有一条有向边，则 $A[i, j]$ 的值就是共现概率，否则就为0。通过图上我们可以看到 GGNN 输入就是初始隐状态矩阵 $h(0)$ 和初始状态的邻接矩阵状态 $a(0)$ 。

具体的更新过程如下公式所示：

$$a^{(t)} = A^T h^{(t-1)} + b \quad (4)$$

$$z^{(t)} = \sigma(W^z a^{(t)} + U^z h^{(t-1)}) \quad (5)$$

$$r^{(t)} = \sigma(W^r a^{(t)} + U^r h^{(t-1)}) \quad (6)$$

$$c^{(t)} = \tanh(W^c a^{(t)} + U^c (r^{(t)} \odot h^{(t-1)})) \quad (7)$$

$$h^{(t)} = (1 - z^{(t)}) \odot h^{(t-1)} + z^{(t)} \odot c^{(t)} \quad (8)$$

首先 t 时间步的邻接矩阵状态 $a(t)$ 的表示要由上一时刻的输出即事件链的隐状态 $h(t-1)$ 和邻接矩阵 A 来确定， $z(t)$ 和 $r(t)$ 就是 GRU 结构的更新门（update gate）和充值门（reset

gate), 公式 (5) (6) (7) (8) 就是 GRU 的公式不必赘述, 通过 GRU 结构获取当前时间步的表示结果 $h(t)$, 时间步值为固定的 K 。

3、从候选事件预测最可能的事件: 最难的事件表示部分完成之后, 事件链上的每个事件以及5个候选事件都得到了表示, 这里最后一个时间步的每个事件的表示都保留下来。这里为了学习到 event pair 的事件对关联信息, 对于每个候选事件, 将事件链中的每个事件都和这个候选事件计算相关度得分, 最终将每个 event pair 的相关度得分加起来取平均值即为最后的候选事件得分。作者在这里还为事件相关度得分的得分函数使用了不同的函数: 曼哈顿距离, cosine similarity, Dot similarity 和 Euclidean similarity。同时, 作者也明白事件链上的不同事件对于候选事件的影响不同, 因此用了一个最基础的 attention 机制 (和上一篇论文一样), 不再赘述。

总体来说, 这篇文章和上面一篇论文其实做的问题都是一个问题, 使用的方法的思想也相似, 都是要尽可能考虑到事件的时序信息和事件对信息, 针对传统只用 event chain 或者 event pair 工作的不足, 一个提出使用两种方法结合的思想, 一个提出使用 GGNN 进行事件表示的思想进行了实验, 同时都考虑到事件链上不同事件的影响不同而用了 attention 机制, 一个使用了较为复杂的 memory model, 一个只是用了一般的 attention 机制。两篇文章的侧重点不同, 方法也各异, 但总体思想还是一脉相承。对比一下相同数据集的结果如下图所示:

Method	G&C16	C&J08
PMI	30.52	30.92
Bigram	29.67	25.43
Event-Comp	49.57	43.28
RNN	45.74	43.17
MemNet	55.12	46.67

(Wang Z, etc 2017的工作的结果)

Methods	Accuracy
Random	20.00
PMI [Chambers and Jurafsky, 2008]	30.52
Bigram [Jans <i>et al.</i> , 2012]	29.67
Word2vec [Mikolov <i>et al.</i> , 2013]	42.23
DeepWalk [Perozzi <i>et al.</i> , 2014]	43.01
EventComp [Granroth-Wilding and Clark, 2016]	49.57
PairLSTM [Wang <i>et al.</i> , 2017]	50.83
SGNN-attention (without attention)	51.56
SGNN (ours)	52.45
SGNN+PairLSTM	52.71
SGNN+EventComp	54.15
SGNN+EventComp+PairLSTM	54.93

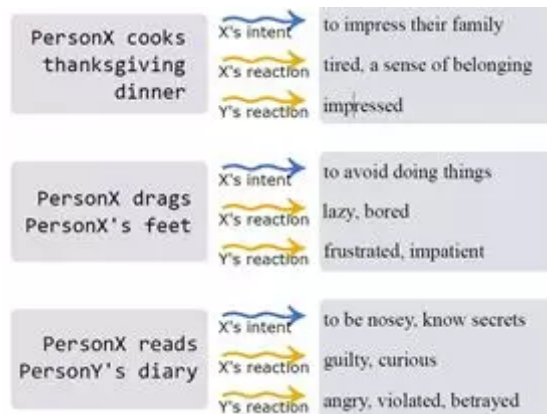
(Li Z, etc 2017的工作的结果)

上述两个工作对比, 都是在 G&C16 的数据集, 采取相同的评估方法, 可以看到, 最优的结果差别并不大, 代表了该领域最前沿的研究结果。总体来说, 脚本事件预测是一个非常有研究前景的领域, 研究也面对越来越多的挑战。

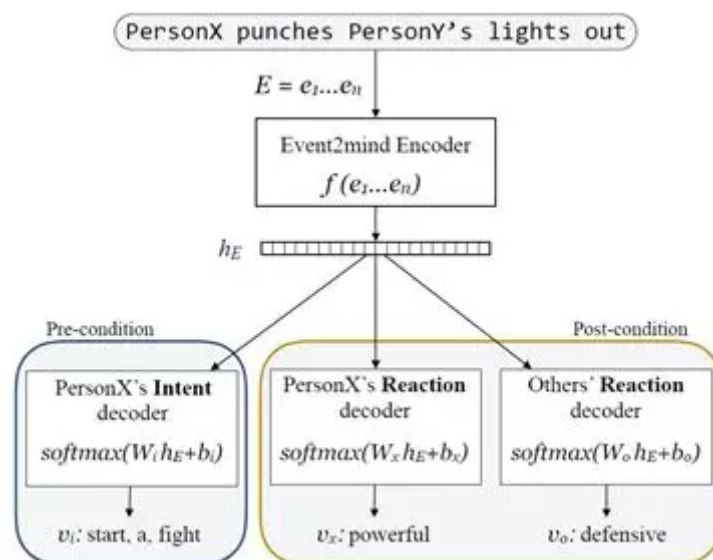
常识级别的事件意图和反应推理

在 ACL2018 的会议中有关事件推理有一篇比较有意思的工作 Event2Mind: Commonsense Inference on Events, Intents, and Reactions [ACL 2018]（本节图片均来自于本篇论文），这篇工作关注常识级别的事件发起者的意图、事件发生后事件参与者的反应的推理，提出这么一项比较偏向事件情感推理的任务，提供了该任务下的众包训练数据集，并且提供了 baseline 和评测方法。

形式化来说，这个任务的输入是**形式化事件描述短句**，比如 “ Person X drinks coffee in the morning ”，输出的是**事件发起者的意图，事件发起者的反应，事件受体反应**。比如这个例子中人物X的意图是：X wants to stay awake ；人物X的事件发生后反应是：X feels alert，由于这个事件没有事件受体Y，因此事件受体 Y（如果有 Y）在事件发生后的反应为：None。更多的例子如下图所示：



这篇文章使用众包来构造数据集，人名的部分全部挖去，用 Person X, Person Y 来代替，我们同时也可以看出 intent 往往是一个短句，reaction 往往是一个单词或词组短句形式。同时作者还提出了 baseline 模型，采用传统的 encoder-decoder 结构，在对事件学习到表示之后将表示分别用于三个 decoder（X intent decoder，X reaction decoder 和 Y reaction decoder）完成生成任务。使用了多任务学习的思想，整个任务的 loss 是同时最小化三个不同 decoder 的 loss 的结果。整个模型的框架如图所示：



Baseline 的方法比较简单，Encoder 学习事件短句的表示，分别用 CNN，BiRNN 等方法来学习，decoder 分别用 n-gram 和 sequence decoding 来完成，但是文章没有很具体给出实验的评估方式，后面的准确率使用人工评估，初步的 baseline 的结果如下图所示：

Encoding Function	Decoder	Development				Test			
		Average Cross-Ent	Recall @10 (%)			Average Cross-Ent	Recall @10 (%)		
max-pool	n-gram	5.75	31	35	68	5.14	31	37	67
mean-pool	n-gram	4.82	35	39	69	4.94	34	40	68
ConvNet	n-gram	4.85	36	42	69	4.81	37	44	69
BiRNN 300d	n-gram	4.78	36	42	68	4.74	36	43	69
BiRNN 100d	n-gram	4.76	36	41	68	4.73	37	43	68
mean-pool	sequence	4.59	39	36	67	4.54	40	38	66
ConvNet	sequence	4.44	42	39	68	4.40	43	40	67
BiRNN 100d	sequence	4.25	39	38	67	4.22	40	40	67

可以看到 CNN 的表现不错，具体的实验数据他们开放在了 github 上。同时，通过后续的结果对比和错误分析以及更多的后续实验，作者发现通过这些 intent 和 reaction 可以构建出 embedding space，相似事件倾向于有相同的意图反应和情感倾向，这对于事件相似度衡量和推理都比较有启发。同时，该工作在电影事件脚本的实验中使用性别作为对比变量，甚至总结出在电影中男性和女性不同的角色刻画倾向，总结的角色刻画倾向和实际中比较吻合，说明通过对事件常识级别的意图和反应推理的确可以比较准确地捕捉一些事件的感情倾向，有益于更多的工作。

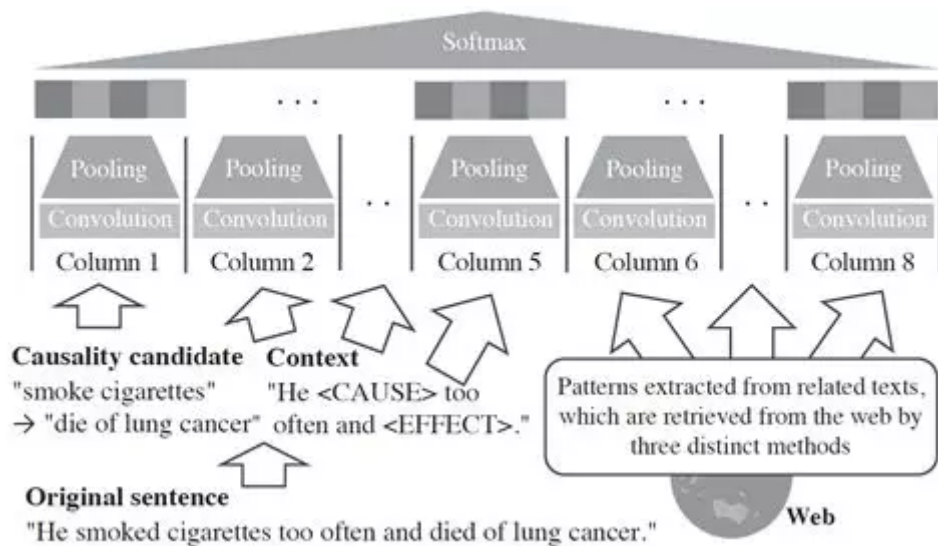
可以说，这篇工作提出了一种新的事件推理的任务，并且有可能该任务会在今后引起更多的重视，事件情感倾向推理的相关研究将受到越来越多研究者的关注。

事件因果关系识别

事件因果关系识别算是一种推断事件之间是否有因果关系的任务，主要是给定一对event pairs，根据文本中的信息及一些挖掘的因果特征来确定这一对事件是否存在着因果关系。Improving Event Causality Recognition with Multiple Background Knowledge Sources Using Multi-Column Convolutional Neural Networks [AAAI 17]（本节图片都来自于这篇工作）这篇工作就是用 MCNN 来从海量语料中寻找一个事件对的因果关系 paths，从而判断事件对之间是否有因果关系。

形式化这个工作：输入是一对因果关系候选实体对，未知是否有因果关系。输出就是是否有因果关系。方法是从 why-QA 语料中的文本进行检索，找出包含这个事件对的文本，以及一些因果关系 clues 的句子、因果关系 pattern 等都输入到 MCNN 中，从而寻找 path 来判断是否具有因果关系。

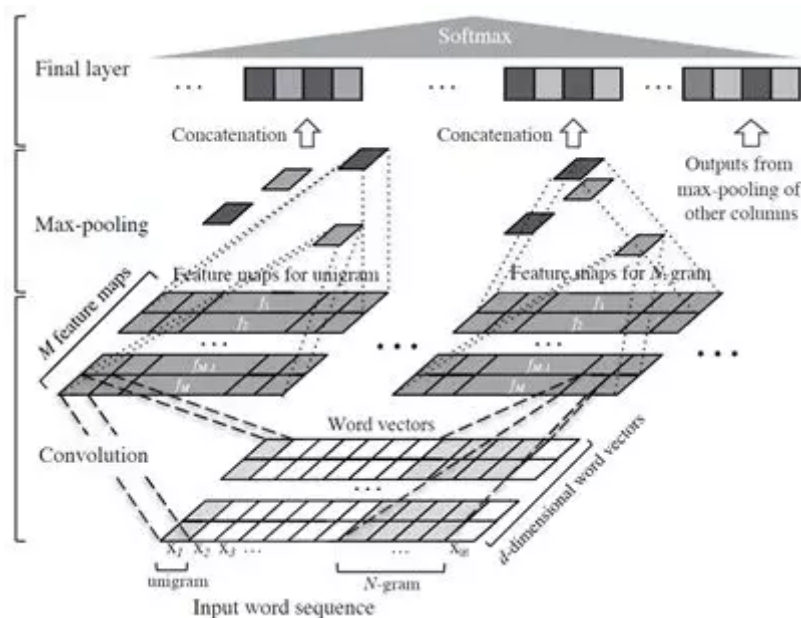
比如 “global warming worsens” 和 “typhoons strengthen” 两个事件，如果需要判断二者是否有因果关系，可以首先构建一个Why的问句，即 “typhoons strengthen” -> “Why do typhoons strengthen?”，然后基本思路就是从大规模语料库中找到那些含有这两个事件的句子作为 answer clues，难点就是并非所有的包含这两个事件的句子都在表达这两个事件有因果关系，所以找到的 clues 的噪声是非常强的，因此作者决定使用 MCNN 进行降噪处理，找到合理的表示事件因果关系的 clues 来判断因果关系的存在。整个方法的框架如下图所示：



整个 MCNN 的输入如图所示，MCNN 设置为8列，其中第一列放两个 candidate，第二列到第五列都放包含这两个事件 candidate 的句子的上下文。具体的，第二列放cause事件的谓语和名词之间的 context，第三列放 effect 事件的谓语和名词之间的context，第四列放 cause 事件的名词和 effect 事件谓语之间的 predicate，第五列放effect 事件的谓语之后的所有内容；第六列放 Short binary patterns，就是在语料中包含cause事件和 effect 事件的句子最常出现的 binary patterns，比如两个事件由 “因为...”， “原因

是...”等 pattern 进行连接；第七列放在 QA 语料中检索到的构成的 why 问题的答案，答案句子要包含 cause 事件的每个词；第八列从语料中找一两句有因果关系的句子，并且这些句子必须包含第六列的那些表示因果关系的 pattern。

MCNN 的每一列（column）的框架如下图所示：



可以看到，输入的单词使用 n-gram 进行 Word2vec 来进行表示，之后使用 CNN 结构。MCNN 每一列都使用完整的 CNN 结构，只是 filter 的大小可以不同，通常 filter 的大小需要根据每个部分的不同内容来确定。MCNN 最先使用在图像中，比较经典的就是在图像处理中利用 MCNN 计算人群密度：一张图往往人头的大小是不一样的，同样大小的 filter 往往不能捕捉到图像中人头近大远小的特征，因此设计不同大小的 filter 来对应不同大小的人头，filters 越大，越适合去处理大人头的部分。在 NLP 领域中，MCNN 现在广泛用于回答 Why-QA 的任务中，分析问句并且推理出隐式因果关系。

这篇文章借鉴意义比较大的就是使用 MCNN 找 why 问题的答案路径，这种方法在众多 Why-QA 任务中已经被证明行之有效。作者这里将方法进行延伸去解决两个事件是否有因果关系的问题。同样的找事件因果关系的工作近些年还有一篇 Kang D, etc.2017.,也是从文本中挖掘因果特征来对时序事件的因果关系进行判断，具体工作在这里暂不展开介绍，如果有兴趣的读者可以详细阅读。

事件因果关系的相关论文结合比较难的关系推理领域的因果关系推理，使用事件信息来辅助进行因果关系推理，在研究领域属于比较有挑战性的工作，值得更多研究者的研究。

事件推理小结

篇幅原因，这里只介绍了几篇关于事件推理的相关工作，但是事件推理的相关研究领域不仅于此。比如周期性事件可能发生的时间的推理（同时进行周期性发生事件的发生时间的补

全) Dinesh Raghu, etc. [IJCAI 2018]。Prafulla, etc. 主要研究在事件文本中找到一个句子的多个事件之间是否有时序关系, 重点解决长距离事件之间的时序关系推断。同时还有事件指代消歧的相关工作 Prafulla and Ruihong, 2017 ACL, 2018 EMNLP, 主要针对文档内部事件之间、跨文档事件之间的一些事件可能实际上指的是同一件事件, 但是事件描述非常不同, 这种情况下应该如何进行相同事件识别的相关研究。事件之间的关系挖掘、推理的有很多的研究领域, 不同于事件抽取的主流任务, 事件推理正在朝着各个比较有趣的方向进行发展, 事件抽取作为 IE 的重要任务, 事件推理作为IE之后的更加有趣的应用, 将继续有着不断发展的前景。如果有更多的读者对事件抽取、事件推理的相关研究感兴趣的话, 欢迎和知识工场实验室进行联系和探讨, 也欢迎和我本人(李晨光, 知识工场17级硕士, 联系方式: hnjlzcg@gmail.com) 进行交流。最后, 欢迎大家关注我们实验室的中文事件抽取的相关工作Timeline: A Chinese EventExtraction and Exploration System (Li,etc. 2018) 和相关的中文事件展示demo: <http://218.193.131.250:33663/search/周杰伦>。知识工场实验室将在事件领域进行更多的研究。

参考文献

- 1、Mostafazadeh N, Chambers N, He X, et al. A corpus and evaluation framework for deeper understanding of commonsense stories[J]. arXiv preprint arXiv:1604.01696, 2016.
- 2、Chaturvedi S, Peng H, Roth D. Storycomprehension for predicting what happens next[C]//Proceedings of the 2017Conference on Empirical Methods in Natural Language Processing. 2017:1603-1614.
- 3、Rashkin H, Sap M, Allaway E, et al.Event2Mind: Commonsense Inference on Events, Intents, and Reactions[J]. arXiv preprint arXiv:1805.06939, 2018.
- 4、Li Z, Ding X, Liu T. Constructing Narrative Event Evolutionary Graph for Script Event Prediction[J]. arXiv preprint arXiv:1805.05081, 2018.
- 5、Granroth-Wilding M, Clark S C. Whathappens next? event prediction using a compositional neural network model[J].2016.
- 6、Wang Z, Zhang Y, 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. 2017: 57-67.
- 7、Kruengkrai C, Torisawa K, Hashimoto C, et al. Improving Event Causality Recognition with Multiple Background Knowledge Sources Using Multi-Column Convolutional Neural Networks[C]//AAAI. 2017:3466-3473.
- 8、Kang D, Gangal V, Lu A, et al. Detecting and explaining causes from text for a time series event[J]. arXiv preprint arXiv:1707.08852, 2017.
- 9、D Raghu, S Nair, Mausam - IJCAI, 2018. Inferring Temporal Knowledge for Near-Periodic Recurrent Events.
- 10、Choubey P K, Huang R. A sequential model for classifying temporal relations between intra-sentence events[J]. arXiv preprint arXiv:1707.07343, 2017.

11、Choubey P K, Huang R. Improving Event Coreference Resolution by Modeling Correlations between Event Coreference Chains and Document Topic Structures[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018,1: 485-495.