(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 113487024 B
(45)授权公告日 2024.12.10

(21)申请号 202110725279.X

(22)申请日 2021.06.29

(65)同一申请的已公布的文献号
申请公布号 CN 113487024 A

(43)申请公布日 2021.10.08

(73)专利权人 任立椋
地址 201103 上海市长宁区黄金城道688弄10号2101

(72)发明人 任立椋

(74)专利代理机构 北京商专永信知识产权代理事务所(普通合伙) 11400
专利代理师 黄谦 车江华

(51)Int.Cl.
G06N 3/084(2023.01)
G06N 3/0455(2023.01)
G06N 3/0442(2023.01)
G06N 5/022(2023.01)
G06F 40/216(2020.01)
G06F 40/211(2020.01)
G06F 40/295(2020.01)
G06F 16/35(2019.01)
G06F 16/36(2019.01)

(56)对比文件
CN 108415898 A,2018.08.17
CN 111008266 A,2020.04.14

审查员 董立波

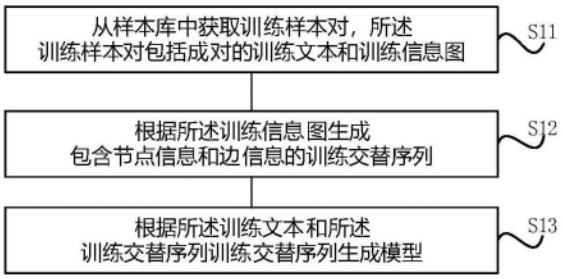
权利要求书2页 说明书18页 附图3页

(54)发明名称

交替序列生成模型训练方法、从文本中抽取图的方法

(57)摘要

本发明公开一种交替序列生成模型训练方法,包括:从样本库中获取训练样本对,所述训练样本对包括成对的训练文本和训练信息图,所述训练信息图中包括多个节点和至少一条连接所述多个节点中的两个节点的边;根据所述训练信息图生成包含节点信息和边信息的训练交替序列;根据所述训练文本和所述训练交替序列训练交替序列生成模型。通过模型从文本中提取信息图时并未直接对图进行建模,而是将从文本中提取图的问题转化为了从文本中提取交替序列的问题,从而使得本实施例的方法得到的交替序列生成模型在用于图抽取时只具有线性的时间和空间复杂度,在时间和空间效率上得到了显著的提升。



CN 113487024 B

1.一种交替序列生成模型训练方法,包括:

从样本库中获取训练样本对,所述训练样本对包括成对的训练文本和训练信息图,所述训练信息图中包括多个节点和至少一条连接所述多个节点中的两个节点的边;

根据所述训练信息图生成包含节点信息和边信息的训练交替序列;

根据所述训练文本和所述训练交替序列训练交替序列生成模型;

所述交替序列生成模型为混合跨度解码器,所述混合跨度解码器从N层内部块的输出中切出交替序列y^{π}的隐藏表示H_y^N,然后对于每个隐藏表示 $h_{yi}^N \in H_y^N$, $0 \leqslant i \leqslant |y^{\pi}|$,使用两个不同的线性层来获得起始位置表示$s_{yi}$ 和结束位置表示e_{yi},

$$\mathbf{s}_{y_i} = W_5^T \mathbf{h}_{y_i} + \mathbf{b}_5 \quad \in R^{d_m},$$
$$\mathbf{e}_{y_i} = W_6^T \mathbf{h}_{y_i} + \mathbf{b}_6 \quad \in R^{d_m},$$

其中,m是最大跨度长度,d_m是隐藏层的大小,$W_5, W_6 \in R^{dm \times dm}$和$b_5$, $b_6 \in R^{dm}$是可学习的参数,然后联合估计文本跨度和类型跨度的概率:

$$\begin{aligned}
\mathbf{h}_{s_i} &= H_{\text{types}}\, \mathbf{s}_{y_i} + \mathbf{m}_a && \in R^{l_p},\\
\mathbf{h}_{e_i} &= H_{\text{types}}\, \mathbf{e}_{y_i} + \mathbf{m}_a && \in R^{l_p},\\
\mathbf{h}_i &= \mathbf{h}_{s_i} + \mathbf{h}_{e_i} && \in R^{l_p},\\
\mathbf{t}_{s_i} &= H_{\text{text}}\, \mathbf{s}_{y_i} + \mathbf{m}'_a && \in R^{n},\\
\mathbf{t}_{e_i} &= H_{\text{text}}\, \mathbf{e}_{y_i} + \mathbf{m}'_a && \in R^{n},\\
\mathbf{t}_i &= \text{unfold}(\mathbf{t}_{e_i}, m) + \mathbf{t}_{s_i} && \in R^{nm},\\
\mathbf{p}(y^{\pi}_{i+1}) &= \text{softmax}(\mathbf{h}_i \oplus \mathbf{t}_i) && \in R^{nm+l_p},
\end{aligned}$$

其中,lp是节点类型集、边类型集和虚拟边类型集的各种类型的数量,H_{types}为输入类型表示,H_{text}为输入文本表示,n是输入文本的最大输入长度,h_i是H的类型段中跨度的得分向量,而t_i 是H的文本段中跨度的得分向量,交替掩码 $m_a \in R^{lp}$, $m_a' \in R^n$定义为:

$$\mathbf{m}_a(j) = \begin{cases} 0, & y_i^{\pi} > l_e \cap j < l_e \\ -\infty, & \text{otherwise} \end{cases}$$
$$\mathbf{m}'_a(j) = \begin{cases} -\infty, & y_i^{\pi} > l_e \\ 0, & \text{otherwise} \end{cases}$$

其中,$l_e = |R| + |U|$是边类型的总数。

2.根据权利要求1所述的方法,其特征在于,所述根据所述训练信息图生成包含节点信息和边信息的训练交替序列,包括:

采用预设遍历算法对所述训练信息图进行遍历生成包含节点信息和边信息的训练交替序列。

3.根据权利要求1或2所述的方法,其特征在于,所述训练交替序列包括相互间隔的节点信息和边信息。

4.根据权利要求3所述的方法,其特征在于,所述节点信息包括节点类型信息,所述边信息包括实际边类型信息和虚拟边类型信息。

5.根据权利要求4所述的方法,其特征在于,所述训练信息图中包括作为输入文本片段的地址的文本跨度和作为抽象概念的表示的类型,其中,所述类型为节点类型信息、实际边类型信息和虚拟边类型信息的词汇表的长度为1的跨度。

6.根据权利要求3所述的方法,其特征在于,根据所述训练文本和所述训练交替序列训练交替序列生成模型,包括:

对所述交替序列生成模型的输出分布采用交替掩码进行处理,以得到相互间隔的节点信息和边信息构成的交替序列。

7.一种从文本中抽取图的方法,包括:

将待抽取文本输入采用权利要求1-6中任意一项所述的方法训练得到的交替序列生成模型得到目标交替序列;

根据所述目标交替序列生成目标信息图。

8.根据权利要求7所述的方法,其特征在于,所述根据所述目标交替序列生成目标信息图包括:

根据训练所述交替序列生成模型所采用的预设遍历算法对所述目标交替序列进行处理,生成目标信息图。

9.一种电子设备,其包括:至少一个处理器,以及与所述至少一个处理器通信连接的存储器,其中,所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行权利要求7-8中任意一项所述方法的步骤。

10.一种计算机可读存储介质,其上存储有计算机程序,其特征在于,该程序被处理器执行时实现权利要求7-8中任意一项所述方法的步骤。

交替序列生成模型训练方法、从文本中抽取图的方法

技术领域

[0001] 本发明涉及信息处理技术领域,尤其涉及一种交替序列生成模型训练方法、从文本中抽取图的方法、电子设备及计算机可读存储介质。

背景技术

[0002] 现有的从文本中提取图方法通常先使用神经网络编码一段文本,然后使用成对打分的方法来生成图的边;或者使用多维度循环神经网络生成一张图连接表;或者使用生成图的节点-边-节点的三元组序列的方法来从文本中抽取图。同时有些技术会将图的节点表示成具体的文字或者单词。

[0003] 这些技术的时间和空间复杂度通常较高(大于线性复杂度),或者无法准确抽取含有非常见/未见过的单词的节点,或者忽视了图元素(边和节点)之间的依赖关系,图抽取的准确率和精度较低;

[0004] 因为采用成对打分的方法要遍历所有可能的文本对,所以会具有较高的时间复杂度;而使用多维度循环神经网络的方法需要存储整张图连接表的隐表示,所以会具有较高的空间复杂度。将图节点表示成具体的单词或者文字会导致节点分类器无法准确估计非常见/未见过的单词的概率分布,从而无法准确抽取这些单词作为图的节点,而这也会影响到图抽取的整体准确率和精度。成对打分的方法将每条边视为相互独立的元素分别进行分类,而这忽视了边与边之间的依赖关系,三元组序列生成的方法在生成三元组的时候分别独立地对边和节点进行分类,而这忽视了边与节点之间的依赖关系。这些对依赖关系的忽视都会影响图抽取的整体准确率和精度。

[0005] 总的来说,在使用现有技术的时候发明人发现这些方案的时间复杂度或者空间复杂度较高,而图抽取的综合准确度和精度较低,难以应用到大规模长文本的实际工业级使用场景。

发明内容

[0006] 本发明实施例提供一种交替序列生成模型训练方法、从文本中抽取图的方法、电子设备及计算机可读存储介质,用于至少解决上述技术问题之一。

[0007] 第一方面,本发明实施例提供一种交替序列生成模型训练方法,包括:

[0008] 从样本库中获取训练样本对,所述训练样本对包括成对的训练文本和训练信息图,所述训练信息图中包括多个节点和至少一条连接所述多个节点中的两个节点的边;

[0009] 根据所述训练信息图生成包含节点信息和边信息的训练交替序列;

[0010] 根据所述训练文本和所述训练交替序列训练交替序列生成模型。

[0011] 在一些实施例中,所述根据所述训练信息图生成包含节点信息和边信息的训练交替序列,包括:采用预设遍历算法对所述训练信息图进行遍历生成包含节点信息和边信息的训练交替序列。

[0012] 在一些实施例中,所述训练交替序列包括相互间隔的节点信息和边信息。

[0013] 在一些实施例中,所述节点信息包括节点类型信息,所述边信息包括实际边类型信息和虚拟边类型信息。

[0014] 在一些实施例中,所述训练信息图中包括作为输入文本片段的地址的跨度和作为抽象概念的表示的类型,其中,所述类型可以为节点类型信息、实际边类型信息和虚拟边类型信息的词汇表的长度为1的跨度。

[0015] 在一些实施例中,根据所述训练文本和所述训练交替序列训练交替序列生成模型,包括:对所述交替序列生成模型的输出分布采用交替掩码进行处理,以得到相互间隔的节点信息和边信息构成的交替序列。

[0016] 第二方面,本发明实施例提供一种从文本中抽取图的方法,包括:

[0017] 将待抽取文本输入采用前述方法训练得到的交替序列生成模型得到目标交替序列;

[0018] 根据所述目标交替序列生产成目标信息图。

[0019] 在一些实施例中,所述根据所述目标交替序列生成目标信息图包括:

[0020] 根据训练所述交替序列生成模型所采用的预设遍历算法对所述目标交替序列进行处理,生成目标信息图。

[0021] 第三方面,本发明实施例提供一种存储介质,所述存储介质中存储有一个或多个包括执行指令的程序,所述执行指令能够被电子设备(包括但不限于计算机,服务器,或者网络设备等)读取并执行,以用于执行本发明上述任一项从文本中抽取图的方法。

[0022] 第四方面,提供一种电子设备,其包括:至少一个处理器,以及与所述至少一个处理器通信连接的存储器,其中,所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行本发明上述任一项从文本中抽取图的方法。

[0023] 第五方面,本发明实施例还提供一种计算机程序产品,所述计算机程序产品包括存储在存储介质上的计算机程序,所述计算机程序包括程序指令,当所述程序指令被计算机执行时,使所述计算机执行上述任一项从文本中抽取图的方法。

[0024] 本实施例中在通过模型从文本中提取信息图时并未直接对图进行建模,而是将从文本中提取图的问题转化为了从文本中提取交替序列的问题,从而使得本实施例的方法得到的交替序列生成模型在用于图抽取时只具有线性的时间和空间复杂度,在时间和空间效率上得到了显著的提升。

附图说明

[0025] 为了更清楚地说明本发明实施例的技术方案,下面将对实施例描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0026] 图1为本发明的交替序列生成模型训练方法的一实施例的流程图;

[0027] 图2为本发明的从文本中抽取图的方法的一实施例的流程图;

[0028] 图3为本发明的信息多重图的交替序列的一实施例的示意图;

[0029] 图4为本发明的编码器架构的一实施例的示意图;

[0030] 图5为本发明的在ACE05数据集中知识图的交替序列的一实施例的示意图;

[0031] 图6为本发明的混合跨度解码器的一实施例的示意图；

[0032] 图7为本发明的交替序列的BFS遍历嵌入示意图；

[0033] 图8为本发明的为在ACE05测试集上剩余错误的分布示意图；

[0034] 图9为本发明中的具有混合注意力层的转换器的结构示意图；

[0035] 图10为本发明的电子设备的一实施例的结构示意图。

具体实施方式

[0036] 为使本发明实施例的目的、技术方案和优点更加清楚，下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。需要说明的是，在不冲突的情况下，本申请中的实施例及实施例中的特征可以相互组合。

[0037] 本发明可以在由计算机执行的计算机可执行指令的一般上下文中描述，例如程序模块。一般地，程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、元件、数据结构等等。也可以在分布式计算环境中实践本发明，在这些分布式计算环境中，由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中，程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

[0038] 发明人在使用现有技术的时候发现这些方案的时间复杂度或者空间复杂度较高，而图抽取的综合准确度和精度较低，难以应用到大规模长文本的实际工业级使用场景。因此发明人提出一种既具有高性能又具有高效率的方案以适应现有的工业级应用场景。

[0039] 如图1所示，本发明的实施例提供一种交替序列生成模型训练方法，包括：

[0040] S11、从样本库中获取训练样本对，所述训练样本对包括成对的训练文本和训练信息图，所述训练信息图中包括多个节点和至少一条连接所述多个节点中的两个节点的边。

[0041] 示例性地，训练信息图可以被视为异构多重图(Li et al.,2014;Shi et al.,2017)$G=(V,E)$，其中V是一组节点(通常代表输入文档中的跨度(t_s,t_e))，E是具有节点类型映射函数$\varphi:V\rightarrow Q$和边类型映射函数$\psi:E\rightarrow R$的边的多重集。假设节点类型和边类型是从有限词汇表中提取的。可以使用节点类型来表示实体类型(PER、ORG等)，而边类型可以表示节点之间的关系(PHYS、OGR-AFF等)。

[0042] S12、根据所述训练信息图生成包含节点信息和边信息的训练交替序列。

[0043] 本实施例中，没有直接对异构多重图G的空间建模，而是构建从G到序列空间S^{π}的映射$s^{\pi}=f_s(G,\pi)$。f_s取决于节点的(给定)排序π及其在G中的边，由广度优先搜索(BFS)或深度优先搜索(DFS)等图遍历算法以及节点和边类型的内部排序构建。

[0044] 在一些实施例中，所述节点信息包括节点类型信息，所述边信息包括实际边类型信息和虚拟边类型信息。

[0045] 本申请假设序列s^{π}的元素s_i^{π}是从节点表示V(定义如下)、节点类型集Q、边类型集R和“虚拟”边类型集U的有限集中获取，$\forall s_i^{\pi}\in s^{\pi}$，$s_i^{\pi}\in V\cup Q\cup R\cup U$。虚拟边类型$U=\{[SOS],[EOS],[SEP]\}$不代表G中的边，但用于控制序列的生成，指示序列的开始/结束和图中层级的分离。

[0046] 示例性地，所述训练交替序列包括相互间隔的节点信息和边信息。例如，假设序列

$s^{\pi}=s_0^{\pi},\cdots s_n^{\pi}$,具有交替结构,其中$s_0^{\pi},s_2^{\pi},s_4^{\pi},\cdots$代表节点V,$s_1^{\pi},s_3^{\pi},s_5^{\pi},\cdots$代表实际或虚拟边。在BFS的情况下,本申请利用BFS逐层访问节点的事实(即以顺序$p_i,c_{i1},\cdots,c_{ik},p_j$,其中$c_{ik}$是父节点$p_i$的第k个子节点,由边$e_{ik}$连接,$p_j$可能等同于也可能不等同于$p_i$的子节点子之一),本申请把它变成一个序列,

[0047] $s^{\pi}=p_i,\psi(e_{i1}),c_{i1},\ldots,$

[0048] $\psi(e_{ik}),c_{ik},[SEP],p_j,\ldots$

[0049] 其中,本申请使用特殊的边类型[SEP]来描绘图中的层级。具体的特殊边类型的名称可以是任意的,包括但不限于本文中提到的[SEP]。在DFS的情况下,[SEP]类型紧接着出现在叶节点之后。如果本申请知道交替序列是基于哪种类型的图遍历算法(BFS或DFS)的话,这种表示允许本申请明确地恢复原始信息图。

[0050] S13、根据所述训练文本和所述训练交替序列训练交替序列生成模型。

[0051] 本实施例中在通过模型从文本中提取信息图时并未直接对图进行建模,而是将从文本中提取图的问题转化为了从文本中提取交替序列的问题,从而使得本实施例的方法得到的交替序列生成模型在用于图抽取时只具有线性的时间和空间复杂度,在时间和空间效率上得到了显著的提升。

[0052] 在一些实施例中,对于步骤S20根据所述训练信息图生成包含节点信息和边信息的训练交替序列,包括:采用预设遍历算法对所述训练信息图进行遍历生成包含节点信息和边信息的训练交替序列。其中,预设遍历算法可以是广度优先搜索(BFS)算法或深度优先搜索(DFS)算法,本申请对此不作限定。

[0053] 在一些实施例中,所述训练信息图中包括作为输入文本片段的地址的跨度和作为抽象概念的表示的类型,其中,所述类型可以为节点类型信息、实际边类型信息和虚拟边类型信息的词汇表的长度为1的跨度。

[0054] 交替序列的节点和边的表示依赖于这样一个观察,即信息图中只有两种对象:跨度(作为输入文本片段的地址)和类型(作为抽象概念的表示)。由于本申请可以将类型视为基于所有类型(Q∪R∪U)的词汇表的长度为1的特殊跨度,本申请将这些由文本跨度和长度为1的类型跨度组成的有序集合定义为混合跨度。有序集合中的索引可以根据它们的大小可逆地映射回类型或文本跨度。通过跨度和类型的联合索引,生成信息图的任务因此转换为生成混合跨度的交替序列。

[0055] 在一些实施例中,根据所述训练文本和所述训练交替序列训练交替序列生成模型,包括:对所述交替序列生成模型的输出分布采用交替掩码进行处理,以得到相互间隔的节点信息和边信息构成的交替序列。

[0056] 示例性地,交替序列生成模型为一种神经解码器,它被强制只通过以混合方式解码跨度和类型来生成交替序列,对于每个解码步骤,本申请的解码器仅具有相对于输入序列长度的线性空间和时间复杂度,并且由于其作为序贯决策过程的性质,它可以捕获指称项和类型之间的相互依赖性。

[0057] 如图2所示,为本发明的从文本中抽取图的方法的一实施例的流程图,该实施例中包括:

[0058] S21、将待抽取文本输入采用前述交替序列生成模型训练方法训练得到的交替序列生成模型得到目标交替序列;

[0059] S22、根据所述目标交替序列生产成目标信息图。

[0060] 本实施例中在通过模型从文本中提取信息图时并未直接对图进行建模,而是将从文本中提取图的问题转化为了从文本中提取交替序列的问题,从而在从文本中进行图抽取时只具有线性的时间和空间复杂度,在时间和空间效率上得到了显著的提升。

[0061] 在一些实施例中,所述根据所述目标交替序列生成目标信息图包括:

[0062] 根据训练所述交替序列生成模型所采用的预设遍历算法对所述目标交替序列进行处理,生成目标信息图。

[0063] 为更加清楚的介绍本发明的技术方案,也为更直接地证明本发明的可实时性以及相对于现有技术的有益性,以下将对本发明的技术背景、技术方案以及所进行的实验等进行更为详细的介绍。

[0064] 摘要

[0065] Text-to-Graph提取旨在从自然语言文本中自动提取由指称(或实体)和类型组成的信息图。现有的方法,如表格填充和成对评分,在各种信息提取任务上表现出令人印象深刻的性能,但由于它们相对于输入长度的二阶空间/时间复杂性,它们难以扩展到具有更长输入文本的数据集。在这项工作中,本申请提出了一个Hybrid SPan Generator(HySPA)将信息图映射到节点和边类型的交替序列,并通过混合跨度解码器直接生成这样的序列,混合跨度解码器可以在线性时间和空间复杂度中对跨度和类型进行循环解码。在ACE05数据集上的大量实验表明,本申请的方法在联合实体和关系提取任务上也显著优于现有方法。

[0066] 1、介绍

[0067] 信息提取(IE)可以被视为Text-to-Graph的提取任务,旨在从非结构化文本中提取由指称(或实体)和类型组成的信息图(Li et al.,2014;Shi et al.,2017),其中,图的节点是指称或实体类型,边是表示节点之间关系的关系类型。图提取的典型方法是将提取过程分解为子任务,例如命名实体识别(NER)(Florian等人,2006,2010)和关系提取(RE)(Sun等人,2011年;Jiang和Zhai,2007年),和分别执行它们(Chan和Roth,2011年)或联合执行它们(Li和Ji,2014年;Eberts和Ulges,2019年)。

[0068] 最近的联合IE模型(Wadden等人,2019年;Wang和Lu,2020年;Lin等人,2020年)在各种IE任务上表现出令人印象深刻的性能,因为它们可以减轻错误传播并利用任务之间的相互依赖性。以前的工作经常使用成对评分技术来识别实体之间的关系类型。然而,这种方法计算效率低下,因为它需要枚举文档中所有可能的实体对,并且由于实体之间关系的稀疏性,关系类型在大多数情况下为空值。此外,成对评分技术独立评估每种关系类型,因此无法捕获不同指称对的关系类型之间的相互关系。

[0069] 另一种方法是将联合信息提取任务视为表格填充问题(Zhang et al.,2017;Wang and Lu,2020),并使用多维循环神经网络生成二维表格(Graves et al.,2007)。这可以捕获实体和关系之间的相互关系,但空间复杂度相对于输入文本的长度呈二次方增长,使得这种方法对于长序列不切实际。

[0070] 一些尝试,例如,Seq2RDF(Liu et al.,2018)和IMoJIE(Kolluru et al.,2020),利用Seq2seq模型(Cho et al.,2014)的强大功能来捕捉具有一阶复杂度的指称和类型之间的相互关系,但它们都使用预先定义的词汇表进行指称预测,这在很大程度上取决于目标词的分布,并且无法处理看不见的词汇表外的词。

[0071] 为了解决这些问题,本申请提出了一种一阶方法,将目标图可逆地映射到节点和边的交替序列,并应用直接学习生成这种交替序列的混合跨度生成器。本申请的主要贡献有三方面:

[0072] • 本申请提出了一种通用技术来在信息图和交替序列之间进行可逆映射(假设给定的图遍历算法)。生成交替序列相当于生成原始信息图。

[0073] • 本申请提出了一种新的神经解码器,它被强制只通过以混合方式解码跨度和类型来生成交替序列。对于每个解码步骤,本申请的解码器仅具有相对于输入序列长度的线性空间和时间复杂度,并且由于其作为序贯决策过程的性质,它可以捕获指称项和类型之间的相互依赖性。

[0074] • 本申请对自动内容提取(ACE)数据集进行了大量实验,这表明本申请的模型在旨在从一段非结构化文本中提取知识图的联合实体和关系提取任务上实现了当前最先进的性能。

[0075] 2、将信息图建模为交替序列

[0076] 信息图可以被视为异构多重图(Li et al.,2014;Shi et al.,2017)$G=(V,E)$,其中V是一组节点(通常代表输入文档中的跨度(t_s,t_e)),E是具有节点类型映射函数$\varphi:V\rightarrow Q$和边类型映射函数$\psi:E\rightarrow R$的边的多重集。假设节点和边类型是从有限词汇表中提取的。可以使用节点类型例如来表示实体类型(PER、ORG等),而边类型可以表示节点之间的关系(PHYS、ORG-AFF等)。在这项工作中,本申请将节点类型表示为单独的节点,这些单独的节点通过特殊的边类型连接到它们的节点v。

[0077] 将信息图表示为序列:本申请没有直接对异构多重图G的空间建模,而是构建从G到序列空间S^{π}的映射$s^{\pi}=f_s(G,\pi)$。f_s取决于节点的(给定)排序π及其在G中的边,由广度优先搜索(BFS)或深度优先搜索(DFS)等图遍历算法以及节点和边类型的内部排序构建。本申请假设结果序列s^{π}的元素s_i^{π}是从有限集中获得,该有限集包括节点表示V、节点类型Q、边类型(实际边类型)R和“虚拟”边类型U:$\forall s_i^{\pi}\in s^{\pi}, s_i^{\pi}\in V\cup Q\cup R\cup U$。虚拟边类型$U=\{[SOS],[EOS],[SEP]\}$不代表G中的边,但用于控制序列的生成,指示序列的开始/结束和图中层级的划分。

[0078] 本申请进一步假设$s^{\pi}=s_0^{\pi},\cdots s_n^{\pi}$,表示图具有交替结构,其中$s_0^{\pi},s_2^{\pi},s_4^{\pi},\cdots$代表节点V,$s_1^{\pi},s_3^{\pi},s_5^{\pi},\cdots$代表实际或虚拟边。在BFS的情况下,本申请利用它逐层,即以顺序p_i,$c_{i1},\cdots,c_{ik},p_j$访问节点的事实(其中c_{ik}是父节点p_i的第k个子节点,由边e_{ik}连接,p_j可能等于也可能不等于p_i的子节点子之一),本申请把s^{π}变成一个序列,

[0079] $s^{\pi}=p_i,\psi(e_{i1}),c_{i1},\ldots,$

[0080] $\psi(e_{ik}),c_{ik},[SEP],p_j,\ldots$

[0081] 其中,本申请使用特殊的边类型[SEP]来描绘图中的层级。如果本申请知道假设哪种类型的图遍历(BFS或DFS)的话,这种表示允许本申请明确地恢复原始图。算法1(本申请用来将训练数据中的图转换为序列)显示了一个交替序列如何可以使用BFS遍历构建给定的图。图3显示了信息多重图的交替序列。长度$|s^{\pi}|$受图的大小$O(|s^{\pi}|)=O(|V|+|E|)$的线性限制(这也是BFS/DFS等典型图遍历算法的复杂性)。

[0082] 图3:本申请将有向多重图表示为节点(A、B、C、D、E)和边(1、2、3、4、[S])的交替序列。在这里,该图由广度优先搜索(BFS)以节点和边类型的升序遍历。“[s]”或[SEP]是虚拟

边类型,代表每个BFS级别的结束。

算法 1 使用 BFS 的交替序列构建算法

输入: 信息图 G 的有序邻接字典, 输入文本中节点的位置 p_q, 训练集中边类型的频率 p_r

输出: 交替序列 y^{π}

根据 p_q 对 G 中的节点进行排序

对于 G 中的每个节点 v, 分别根据 p_q 和 p_r 对 v 的邻居和边进行排序

将 y^{π} 实例化为空列表

for G 中的 u **do**

  if u 没有被访问, **then**

    初始化一个空队列 q

    将 u 标记为已访问并将 u 加入 q

    while q 不为空 **do**

      使 a 节点 w 从 q 出列

      if w 在 G 中, **then**

        附加 w 和所有的邻居 w 边类型为 y^{π}

        附加分离边类型, [SEP], 到 y^{π}

        标记 w 的所有未访问邻居被访问并将它们排入队列 q

      end

    end

  end

end

返回 y^{π}

[0083]

[0084] 节点和边表示:本申请的节点和边表示(在下面解释)依赖于这样一个观察,即信息图中只有两种对象:跨度(作为输入文本片段的地址)和类型(作为抽象概念的表示)。由于本申请可以将类型视为基于所有类型(Q∪R∪U)的词汇表的长度为1的特殊跨度,本申请只需要O(nm+|Q∪R∪U|)个索引来明确表示基于串联的跨度类型词汇表和输入文本的表示,其中n是最大输入长度,m是最大跨度长度,m<<n。本申请将这些由文本跨度和长度为1的类型跨度组成的有序集合定义为混合跨度。这些索引可以根据它们的大小可逆地映射回类型或文本跨度(此映射的详细信息在第3.2节中解释)。通过跨度和类型的联合索引,生成信息图的任务因此转换为了生成混合跨度的交替序列。

[0085] 生成序列:本申请通过带有参数θ的序列生成器h对分布$p(s^{\pi})$进行建模(|s|是s^{π}的

长度)：

[0086] $$p(s_i^\pi|s_0^\pi,...,s_{i-1}^\pi)=h(s_0^\pi,...,s_{i-1}^\pi,\theta),$$

[0087] $$p(s^\pi)=\prod_{i=1}^{|s^\pi|}p(s_i^\pi|s_0^\pi,...,s_{i-1}^\pi),$$

[0088] 本申请将在以下部分中讨论如何强制序列生成器h仅在空间S^π中生成序列,因为本申请不希望h将非零概率分配给没有相应图的任意序列。

[0089] 3、HySPA:交替序列的混合跨度生成

[0090] 为了直接生成一个目标序列(该目标序列在表示输入中跨度的节点和依赖于本申请的提取任务的节点/边类型集合之间交替),本申请首先构建了一个混合表示H,它是来自边类型、节点类型和输入文本的隐藏表示的串联。这种表示既作为本申请的解码器的上下文空间又作为输出空间。然后本申请将输入文本的跨度和类型的索引都可逆地映射到基于表示H的混合跨度。最后通过混合跨度解码器自动生成的混合跨度,来形成交替序列$y^\pi\in S^\pi$。通过将图提取任务转换为序列生成任务,本申请可以轻松地使用波束搜索解码来减少序列决策过程中可能出现的曝光偏差(Wiseman和Rush,2016年),从而找到全局更好的图表示。

[0091] HySPA的高级概述:HySPA模型以一段文本(例如,一个句子或段落)以及预定义的节点和边类型作为输入,并输出信息图的交替序列表示。本申请通过对输出概率应用交替掩码来强制交替生成此序列。详细架构在以下小节中描述。

[0092] 3.1、文本和类型编码器

[0093] 图4显示了本申请提出的模型的编码器架构,其中符号⊕是连接运算符,k是H_0中词向量的索引,$1_e=|R|+|U|$。右侧的彩色表格表示来自H_0的连接词向量的不同块的元类型分配。对于节点类型集Q、边类型集R和虚拟边类型U,本申请安排类型列表v作为边类型、虚拟边类型和节点类型的标签名称的串联,即

[0094] $$\mathbf{v}=\hat{R}\oplus\hat{U}\oplus\hat{Q}$$

[0095] $$\hat{R}=[R_1,...,R_{|R|}]$$

[0096] $$\hat{U}=[U_1,...,U_{|U|}]$$

[0097] $$\hat{Q}=[Q_1,...,Q_{|Q|}]$$

[0098] 其中,⊕表示两个列表之间的连接运算符,$\hat{R},\hat{U},\hat{Q}$分别是集合R,U,Q中类型名称的列表(例如,$\hat{Q}=$ [“Geopolitics”,“Person”,...])。请注意,类型名称列表之间的连接顺序可以是任意的,只要在整个模型中保持一致即可。然后,就像在表-序列编码器的嵌入部分(Wang和Lu,2020)一样,对于每种类型v_i,本申请使用来自预训练语言模型的上下文词嵌入、GloVe嵌入(Pennington et al.,2014)和特征嵌入来嵌入类型的标签符号,其中,GloVe的全称叫Global Vectors for Word Representation,它是一个基于全局词频统计(count-based&overall statistics)的词表征(word representation)工具。

[0099] $$E_1=\text{ContextualizedEmbed}(\mathbf{v}),\in R^{l_p\times d_c}$$

[0100] $$E_2=\text{GloveEmbed}(\mathbf{v}),\in R^{l_p\times d_g}$$

[0101] $$E_3 = \text{CharacterEmbed}(\mathbf{v}), \in R^{l_p \times d_k}$$

[0102] $$E_4 = E_1 \oplus E_2 \oplus E_3 \in R^{l_p \times d_e},$$

[0103] $$E_v = E_4 W_0^T \in R^{l_p \times d_m},$$

[0104] 其中，$l_p = |R| + |U| + |Q|$是各种类型的数量，$W_0 \in R^{de \times dm}$是线性投影层的权重矩阵，$d_e = d_c + d_g + d_k$是总嵌入维数，d_m是本申请模型的隐藏层的大小。在本申请获得每种类型$v_i \in v$的标记的上下文嵌入后，本申请将这些标记向量的平均值作为v_i的表示，并在训练期间冻结其更新。更多细节可参照附录A。

[0105] 该嵌入途径还用于嵌入输入文本x中的单词。与类型嵌入的途径不同，本申请将单词表示为来自预训练语言模型(LM，egBERT(Devlin et al.，2018))的第一个子标记的上下文嵌入，并以端到端的形式对该语言模型微调。

[0106] 在分别获得类型嵌入E_v和文本嵌入E_x后，本申请将它们沿序列长度维度连接起来，形成混合表示H_0。由于H_0是来自四种不同类型标记(即边类型、虚拟边类型、节点类型和文本)的词向量的串联，因此应用元类型嵌入来指示来自表示H_0的向量块之间的这种类型差异(如图4所示)。最终的上下文表示H是通过元类型嵌入和H_0的元素相加得到的，

[0107] $$H_0 = E_v \oplus E_x \in R^{l_h \times d_m},$$

[0108] $$H_s = \text{MetaTypeEmbed}(H_0) \in R^{l_h \times d_m},$$

[0109] $$H = H_0 + H_s \in R^{l_h \times d_m},$$

[0110] 其中，$l_h = l_p + |x|$是本申请的混合表示矩阵H的高。

[0111] 3.2、Span&Types与混合Span之间的可逆映射

[0112] 给定文本中的跨度，$t = (t_s, t_e) \in N^2$，$t_s < t_e$，本申请通过映射g_k将跨度t转换为表示H中的索引k，$k \geq l_p$，

[0113] $k = g_k(t_s, t_e) = t_s m + t_e - t_s - 1 + l_p \in N$，

[0114] 其中，m是跨度的最大长度，$l_p = |R| + |U| + |Q|$。本申请保持图中的类型索引不变，因为它们小于l_p和$k \geq l_p$。由于对于信息图，指称的最大跨度长度m通常远小于文本的长度，即$m << n$，因此本申请可以通过仅考虑长度小于m的跨度减少k的最大量级从$O(n^2)$到$O(nm)$，从而保持本申请的解码器相对于输入文本长度n的线性空间复杂度。图5显示了本申请在ACE05数据集中知识图的交替序列的具体示例，来自ACE05训练集的知识图(底部)的交替序列表示(中间)示例，该示例中以：他在星期一深夜在巴格达被捕(He was captured in Baghdad late Monday night)。其中A_1表示算法1，本申请在此示例中取$m = 16$和$l_p = 19$。交替序列中的“19”是“他”的跨度(0,1)的索引，“83”是“巴格达”的跨度(4,5)的索引，“10”是虚拟边类型[SEP]。该图表的输入文本(顶部)是“他在星期一深夜在巴格达被捕”。

[0115] 由于t_s，t_e，k都是自然数，本申请可以构造一个逆映射g_t，将H中的索引k转换回$t = (t_s, t_e)$，

[0116] $$t_s = g_{t_s}(k) = -\max(0, -k + l_p) + \lfloor \max(0, k - l_p)/m \rfloor + l_p,$$

[0117] $t_e = g_t(k) = gt_s(k) + \max(0, k - l_p) \bmod m$，

[0118] 其中，$\lfloor \cdot \rfloor$是整数底函数，mod是模运算符。请注意，$g_t(k)$可以直接应用于H的类型段中的索引，并保持它们的值不变，即，

[0119] $$g_t(k) = (k, k), \forall k < l_p, k \in \mathbb{N}.$$

[0120] 有了这个特性，本申请可以轻松地将映射g_t合并到本申请的解码器中，以将交替序列y^π映射回混合表示H中的跨度。

[0121] 3.3混合跨度解码器

[0122] 图6显示了本申请的混合跨度解码器的一般模型架构。本申请的解码器将上下文表示H作为输入，并在给定序列开始标记的情况下循环解码交替序列y^π。N是解码器层数，在softmax函数之前的$\oplus$表示连接运算符，H_y^N是来自最后一个解码器层的序列y^π的隐藏表示。本申请的混合跨度解码器可以理解为一个自回归模型，在由H定义的封闭上下文空间和输出空间中操作。

[0123] 基于注意力的混合跨度编码：给定交替序列y^π和映射g_t(第3.2节)，本申请的解码器首先将y^π中的每个索引映射到一个跨度，$(t_{si}, t_{ei}) = g_t(y_i^\pi)$，以表示H为基础，然后转换注意力掩码$M_0$的跨度，以允许模型学习将跨度表示为跨度引用的上下文词表示片段的加权和，

[0124] $$Q = W_1^T H_{[CLS]} + \mathbf{b}_1 \quad \in R^{|y^\pi| \times d_m},$$

[0125] $$K = W_2^T H + \mathbf{b}_2 \quad \in R^{l_h \times d_m},$$

[0126] $$H_y = \text{softmax}\left(\frac{QK^T}{\sqrt{d_m}} + M_0\right)H \quad \in R^{|y^\pi| \times d_m},$$

[0127] $$M_0(i,j) = \begin{cases} 0, & t_{s_i} \le j \le t_{e_i} \\ -\infty, & \text{otherwise} \end{cases}$$

[0128] 其中，$H_{[CLS]} \in R^{|y^\pi| \times d_m}$是序列标记[CLS]开头的$|y^\pi|$次重复隐藏表示，来自H的文本段，$H_y$是本申请对混合的最终表示跨度为$y^\pi$。$W_1, W_2, b_1, b_2$是可学习的参数，$t_{si}, t_{ei}$是本申请正在编码的跨度的开始和结束位置。请注意，对于长度为1的类型spans，softmax计算的结果将始终为1，这导致其span表示恰好是本申请希望的嵌入向量。

[0129] 遍历嵌入：为了区分yπ中不同位置的混合跨度，一种简单的方法是向Hy添加正弦位置嵌入(Vaswani et al.,2017)。然而，这种方法将交替序列视为普通序列并忽略它编码的底层图结构。为了缓解这个问题，本申请提出了一种新颖的遍历嵌入方法，该方法捕获遍历级别信息、父子信息和级别内连接信息作为原始位置嵌入的替代。本申请的遍历嵌入可以编码BFS或DFS遍历模式。作为一个例子，本申请在这里假设BFS遍历。

[0130] 图7：交替序列的BFS遍历嵌入示例，[“他”，类型，PER，[SEP]，“巴格达”，类型，GPE，PHYS，“他”]。本申请的BFS遍历嵌入是层嵌入L、父-子嵌入P和给定交替序列y的树嵌入T的逐点总和，

[0131] $\text{TravEmbed}(y) = L(y) + P(y) + T(y) \in R^{|y| \times dm}$

[0132] 其中，层嵌入为BFS遍历级别i的每个位置分配相同的嵌入向量L_i，并且根据非参数正弦位置嵌入填充嵌入向量的值，因为本申请希望本申请的嵌入外推到序列比训练集中

的任何序列都长。父-子嵌入在BFS遍历级别中的父节点和子节点的位置分配不同的随机初始化嵌入向量,以帮助模型区分这两种节点。为了对层内连接信息进行编码,本申请的见解是,BFS层中每个节点之间的连接可以看作是一个深度3的树,其中第一个深度取父节点,第二个深度填充边类型第三个深度由每个边类型对应的子节点组成。然后,本申请的树嵌入是通过使用每个BFS级别的树位置嵌入(Shiv and Quirk,2019)对深度3树的位置信息进行编码来形成的。图7显示了这些嵌入如何针对给定交替序列发挥作用的具体示例。然后将获得的遍历嵌入逐点添加到交替序列H_y的隐藏表示中,以注入图结构的遍历信息。

[0133] 内部块:通过从混合表示H和目标序列表示H_y切片的输入文本表示H_{text},本申请应用具有混合注意力的N层转换器结构(He等人,2018年),以允许本申请的模型利用来自不同注意层在解码交替序列的边缘或节点时。注意本申请的混合跨度解码器垂直于内部块的神经结构的实际选择,本申请选择混合注意变换器的设计(He et al.,2018)因为它的分层协调特性在经验上更适合本申请对两种不同类型序列元素的异构解码。内部块的详细结构在附录E中解释。

[0134] 混合跨度解码:对于混合跨度解码模块,本申请首先从N层内部块的输出中切出交替序列y^{π}的隐藏表示,并将其表示为H_y^N。然后对于每个隐藏表示$h_{yi}^N \in H_y^N$,$0 \leqslant i \leqslant |y^{\pi}|$,本申请应用两个不同的线性层来获得起始位置表示$s_{yi}$和结束位置表示$e_{yi}$,

[0135] $$\mathbf{s}_{y_i} = W_5^T \mathbf{h}_{y_i} + \mathbf{b}_5 \quad \in R^{d_m},$$

[0136] $$\mathbf{e}_{y_i} = W_6^T \mathbf{h}_{y_i} + \mathbf{b}_6 \quad \in R^{d_m},$$

[0137] 其中$W_5, W_6 \in R^{dm \times dm}$和$b_5, b_6 \in R^{dm}$是可学习的参数。然后本申请分别计算H的types segment和text segment的target spans的分数,并在最终的softmax算子之前将它们连接在一起,以联合估计text spans和type spans的概率,

[0138] $$\mathbf{h}_{s_i} = H_{\text{types}}\, \mathbf{s}_{y_i} + \mathbf{m}_a \quad \in R^{l_p},$$

[0139] $$\mathbf{h}_{e_i} = H_{\text{types}}\, \mathbf{e}_{y_i} + \mathbf{m}_a \quad \in R^{l_p},$$

[0140] $$\mathbf{h}_i = \mathbf{h}_{s_i} + \mathbf{h}_{e_i} \quad \in R^{l_p},$$

[0141] $$\mathbf{t}_{s_i} = H_{\text{text}}\, \mathbf{s}_{y_i} + \mathbf{m}'_a \quad \in R^{n},$$

[0142] $$\mathbf{t}_{e_i} = H_{\text{text}}\, \mathbf{e}_{y_i} + \mathbf{m}'_a \quad \in R^{n},$$

[0143] $$\mathbf{t}_i = \text{unfold}(\mathbf{t}_{e_i}, m) + \mathbf{t}_{s_i} \quad \in R^{nm},$$

[0144] $$\mathbf{p}(y^{\pi}_{i+1}) = \text{softmax}(\mathbf{h}_i \oplus \mathbf{t}_i) \quad \in R^{nm+l_p},$$

[0145] 其中,h_i是H的类型段中可能的跨度的得分向量,而t_i是H的文本段中可能的跨度的得分向量。由于类型跨度的跨度长度始终为1,因此本申请只需要一个element-wise开始位置分数h_{si}和结束位置分数h_{ei}之间的加法计算h_i。t_i的条目包含文本跨度的分数,$t_{si,j}+t_{ei,k}$;$\forall j \leqslant k$,k-j<m,在展开函数的帮助下计算,该函数将向量$t_{ei} \in R^n$转换为大小为m、最大跨度长度、步幅为1的n个滑动窗口的堆栈。交替掩码$m_a \in R^{lp}$,$m_a' \in R^n$定义为:

[0146] $$\mathbf{m}_a(j) = \begin{cases} 0, & y_i^{\pi} > l_e \cap j < l_e \\ -\infty, & \text{otherwise} \end{cases}$$

[0147] $$\mathbf{m}'_a(j) = \begin{cases} -\infty, & y_i^{\pi} > l_e \\ 0, & \text{otherwise} \end{cases}$$

[0148] 其中，$l_e = |R| + |U|$是边类型的总数。这样，虽然本申请有节点和边类型的联合模型，输出分布由交替掩码强制执行以产生节点和边类型的交替解码，这就是本申请称此解码器为混合跨度的主要原因解码器。

[0149] 4、实验

[0150] 4.1、实验设置

[0151] 本申请在LDC3分发的ACE 2005数据集上测试本申请的模型，该数据集包括1万4千5百个句子、3万8千实体(具有7种类型)和7100个关系(具有6种类型)，这些数据集来自一般新闻领域，详情参见附录C。

[0152] 根据之前的工作，本申请使用F1作为NER和RE的评估指标。对于NER任务，当类型和边界跨度都与黄金实体匹配时，预测被标记为正确。对于RE任务，当两个实体的关系类型和边界都正确时，预测是正确的。

[0153] 4.2、实现细节

[0154] 在训练本申请的模型时，本申请应用标签平滑因子为0.1的交叉熵损失。使用每批次2048个标记(大约为28个批次)对模型进行训练，使用AdamW优化器(Loshchilov和Hutter，2018)训练25000次，学习率为$2e^{-4}$，权重衰减为0.01，使用反平方根调度器进行2000次预热。遵循TabSeq模型(Wang和Lu，2020)，本申请在训练期间使用RoBERTa-large(Liu等人)或ALBERT-xxlarge-v1(Lan等人，2020年)作为预训练语言模型，并将其学习率减慢了0.1倍。当ALBERT-xxlarge-v1具有0.1的下降率时，RoBERTalarge的隐形下降率达到0.2。在训练期间本申请的混合跨度解码器的下降率也有0.1。本申请设置最大跨度长度，$m=16$，本申请模型的隐藏大小，$d_m=256$，以及解码器块的数量，$N=12$。尽管理论上波束搜索应该帮助本申请减少曝光偏差，但本申请没有观察光束大小的网格搜索期间的任何性能增益和验证集的长度损失(详细的网格搜索设置在附录A中)。因此，本申请将普通光束大小设置为1，将长度惩罚设置为1，并将这一理论实验矛盾留待未来研究。本申请的模型是使用FAIRSEQ工具包(Ott et al.，2019)构建的，用于高效的分布式训练，所有实验均在两个NVIDIA TITAN X GPU上进行。

[0155]

IE 模型	空间复杂度	时间复杂度	NER	RE
PointerNet (Katiyar and Cardie, 2017)	$O(n)$	$O(n^2)$	82.6	55.9
SpanRE (Dixit and Al-Onaizan, 2019)	$O(n)$	$O(n^2)$	86.0	62.8
Dygie++ (Wadden et al., 2019)	$O(n)$	$O(n^2)$	88.6	63.4
OneIE (Lin et al., 2020)	$O(n)$	$O(n^2)$	88.8	67.5
TabSeq (Wang and Lu, 2020)	$O(n^2)$	$O(n)$	89.5	67.6
HySPA (ours) w/ RoBERTa	$O(n)$	$O(n)$	88.9	**68.2**
HySPA (ours) w/ ALBERT	$O(n)$	$O(n)$	**89.9**	**68.0**

[0156] 表1:IE模型在ACE05测试集上的联合NER和RE F1分数。计算模型的实体和关系解码部分的复杂性(n是输入文本的长度)。此处报告的TabSeq模型的性能基于与本申请相同的ALBERT-xxlarge(Lan等人，2020)预训练语言模型。

[0157]

模型	NER F1	RE F1
HySPA w/ RoBERTa	88.9	68.2
– Traversal-embedding	88.9	66.7
– Masking	88.1	64.8
– BFS	88.7	66.2
– Mixed-attention	88.6	64.7
– Span-attention	88.5	66.1

[0158] 表2:对ACE05测试集的消融研究。“-Traversal-embedding”:本申请去掉了遍历embedding,改用正弦位置embedding,下面的ablation是基于这个ablation之后的模型。“-Masking”:本申请从混合跨度解码器中移除交替掩码。“--BFS”:本申请使用DFS代替BFS作为遍历。“-Mixedattention”:本申请移除了混合注意力层并使用了标准的转换器编码器解码器结构。“-Span-attention”:本申请移除了跨度编码模块中的跨度注意力,取而代之的是对跨度中的单词进行平均。

[0159] 4.3、结果

[0160] 表1将本申请的模型与之前在ACE05测试集上的最新结果进行了比较。与之前使用ALBERT预训练语言模型的SOTA、TabSeq(Wang and Lu,2020)相比,本申请使用ALBERT的模型在NER分数和RE分数上都有明显更好的性能,同时保持了比TabSeq小一个数量级的线性空间复杂度。与之前所有联合IE模型相比,本申请的模型是第一个同时具有线性空间和时间复杂性的联合模型,因此对于大规模现实世界应用程序具有最佳的可扩展性。

[0161] 4.4、消融研究

[0162] 为了证明本申请方法的有效性,本申请在ACE05数据集上进行了消融实验。如表2所示,在本申请去除遍历嵌入后,RE F1分数显着下降,这表明本申请的遍历嵌入可以帮助对图结构进行编码并改进关系预测。此外,如果放弃交替掩蔽,NER F1和RE F1分数都会显着下降,这证明了强制执行交替模式的重要性。本申请可以观察到混合注意层对关系提取有显着贡献。这是因为逐层协调可以帮助解码器解开源特征并利用实体和关系预测之间的不同层特征。本申请还可以观察到DFS遍历的性能比BFS差。本申请怀疑这是因为由于知识图的性质,来自DFS的结果交替序列通常比来自BFS的交替序列更长,从而增加了学习难度。

[0163] 4.5、误差分析

[0164] 在分析了80个剩余错误后,本申请对以下常见情况进行了分类和讨论(图8为在ACE05测试集上剩余错误的分布示意图)。这些可能需要额外的功能和策略来解决。

[0165] 上下文不足:在许多示例中,答案实体是一个代词,鉴于上下文有限,无法准确键入:在“本申请注意到他们说他们不想使用销毁的词,事实上,他们说让别人这样做”,很难正确地将本申请归类为一个组织。这可以通过使用整个文档作为输入,利用跨句子上下文来缓解。

[0166] 生僻字:生僻字问题是测试集中的词很少出现在训练集中,并且通常不会在字典中出现。在句子“基地还有海军FA-18和海军Heriers”,术语“Heriers”(一种被模型错误地归类为人的车辆)既没有出现在训练集中,也没有被预训练的语言模型很好地理解;在这种情况下,模型只能依靠子词级表示。

[0167] 需要背景知识:通常句子中提到的实体很难从上下文中推断出来,但通过查阅知

识库很容易识别:在“空客应该发出更强烈的警报”中,本申请的模型错误地预测了空客是一种车辆,而这里的空客指的是欧洲航空航天公司。本申请的系统也将联合国安理会分为联合国和安理会两个实体,产生了一个不存在的关系三元组(安理会、联合国的一部分)。通过查阅知识库(例如DBpedia(Bizer等,2009)或执行实体链接)可以避免此类错误。

[0168] 固有的歧义:许多例子都有固有的歧义,例如欧盟可以被归类为组织或政治实体,而一些实体(例如,军事基地)可以既是地点又是组织或设施。

[0169] 5、相关工作

[0170] NER通常与RE联合完成,以减少错误传播并学习任务之间的相互关系。一种方法是将联合任务视为平方表填充问题(Miwa和Sasaki,2014年;Gupta等人,2016年;Wang和Lu,2020年),其中第i列或行代表第i令牌。该表具有指示实体和其他条目的顺序标记的对角线作为标记对之间的关系。另一行工作是在NER之后执行RE。在Miwa和Bansal(2016年)的工作中,作者使用BiLSTM(Graves等人,2013年)作为NER,因此使用了基于依赖关系图的Tree-LSTM(Tai等人,2015年)用于RE.Wadden等人。(2019)和栾等人。(2019)另一方面,采用构建动态文本跨度图的方法来检测实体和关系。扩展瓦登等人。(2019),林等人。(2020)介绍了ONEIE,它进一步结合了基于跨子任务和实例约束的全局特征,旨在将IE结果提取为图形。请注意,本申请的模型与ONEIE(Lin et al.,2020)的不同之处在于,本申请的模型通过自回归生成自动捕获全局关系,而ONEIE使用特征工程模板;此外,ONEIE需要对关系提取进行成对分类,而本申请的方法有效地生成现有关系和实体。

[0171] 虽然已经提出了几个基于Seq2Seq的模型(Zhang et al.,2020;Zeng et al.,2018,2020;Wei et al.,2019;Zhang et al.,2019)来生成三元组(即node-edge-节点),本申请的模型与它们的根本不同在于:(1)它生成目标图的BFS/DFS遍历,它捕获节点和边之间的依赖关系并具有更短的目标序列,(2)本申请对节点进行建模由于文本中的跨度与词汇无关,因此即使节点的标记是生僻词或未见过的词,本申请仍然可以根据上下文信息对其生成跨度。

[0172] 6、结论

[0173] 在这项工作中,本申请提出了混合跨度生成(HySPA)模型,这是第一个在图解码阶段具有线性空间和时间复杂度的端到端文本到图提取模型。除了可扩展性之外,该模型还在ACE05联合实体和关系提取任务上实现了当前最先进的性能。鉴于本申请的混合跨度生成器的结构的灵活性,未来仍有丰富的研究方向,例如结合外部知识进行混合跨度生成,应用更有效的稀疏自注意力,并开发更好的搜索方法来找到更多由交替序列表示的全局合理图。

[0174] 在一些实施例中,还提供另一种方法,该方法中移除了混合注意力层而使用了标准的Transformer编码器解码器结构。这种版本的结构更简单但性能要劣于使用了混合注意力层的版本。

[0175] 在一些实施例中,还提供另一种方法,该方法中使用了DFS遍历而不是BFS遍历来构建图的交替序列表示,同时这种版本还使用了DFS遍历嵌入(详情参见附录D)而不是BFS遍历嵌入。这种版本的图抽取准确度要劣于BFS遍历。

[0176] 在一些实施例中,还提供另一种方法,该方法中将跨度中的单词进行平均来编码跨度而不是进行基于注意力的跨度编码。这种版本的模型结构要更为简单并且模型参数更

少但图抽取准确度要劣于基于注意力的跨度编码。

[0177] 附录A:超参数

[0178] 本申请使用在6B令牌上训练的100维GloVe词嵌入作为初始化,并在训练期间冻结其更新。特征嵌入有30维的LSTM编码,词汇外标记的GloVe嵌入被替换为随机初始化的向量,遵循Wang和Lu(2020)。本申请在训练期间使用0.25的梯度裁剪。本申请的混合注意力的头数设置为8。束大小和长度惩罚由对ACE05数据集的验证集的网格搜索决定,束大小的范围从1到7,步长大小为1,长度惩罚从0.7到1.2,步长为0.1。本申请根据关系提取F1分数的度量选择最佳光束大小和长度惩罚。

[0179] 附录B:训练细节

[0180] 本申请的模型使用ALBERT-xxlarge预训练语言模型有2.36亿个参数。平均而言,本申请使用ALBERT-xxlarge的最佳模型可以在两个NVIDIA TITAN X GPU上分布式训练20小时。

[0181] 附录C:数据

[0182] 自动内容提取(ACE)2005数据集包含用于2005自动内容提取(ACE)技术评估的英语、阿拉伯语和中文训练数据,提供实体、关系和事件注释。本申请跟随瓦登等人(2019)用于预处理和数据拆分。预处理数据包含7100个关系、3万8千个实体和1万4千5百个句子。拆分包含10051个训练样本、2424个开发样本和2050个测试样本。

[0183] 附录D:DFS遍历嵌入

[0184] 由于父子信息已经包含在DFS遍历的层内连接中,本申请只有层嵌入和DFS遍历嵌入的连接嵌入之和。与BFS嵌入类似,DFS层嵌入在DFS遍历层i为每个位置分配相同的嵌入向量Li,但嵌入向量的值是随机初始化的,而不是用非参数正弦位置嵌入填充,因为接近度DFS的遍历层级之间不存在信息。但是,对于DFS级别中的元素,本申请确实有明确的距离信息,即对于DFS级别D=[A,B,C,...,[sep]],从A到元素的距离[A,B,C,...,[sep]]是[0,1,2;3,...,|D|-1]。本申请用正弦位置嵌入对这个距离信息进行编码,这成为本申请的连接嵌入,捕获层内连接信息。

[0185] 附录E:具有混合注意力层的转换器

[0186] 本申请首先从混合表示H中切出输入文本的隐藏表示,并将其表示为H_{text},然后将输入文本表示H_{text}和混合跨度编码H_y的输出输入到N混合注意力/前馈块的堆栈中具有图9所示的结构。

[0187] 由于生成节点和边缘类型可能需要来自不同层的特征,本申请使用混合注意力(He et al.,2018),这允许本申请的模型在对文本段,H_{text}和目标进行编码时利用来自不同注意力层的特征特点,H_y,

[0188] $$\text{MixedAtt}(Q,K,V)=\text{softmax}\left(\frac{QK^T}{\sqrt{d_m}}+M_1\right)V \in R^{l_m\times d_m},$$

[0189] $$M_1(i,j)=\begin{cases}0, & j<n\cup j\le i+n\\ -\infty, & \text{otherwise}\end{cases}$$

[0190] 其中n=|x|是输入文本的长度,l_m=|x|+|yπ|是源特征和目标特征的总长度。将

源特征H_{text}和目标特征H_y的串联表示为H_0,在混合注意力的第一层之前还向H_0添加了源/目标嵌入(He et al.,2018)以允许模型区分来自源序列和目标序列的特征。混合注意力层与前馈层结合形成解码器块:

[0191] $$\mathrm{MixedAtt}(Q,K,V)=\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_m}}+M_1\right)V \in R^{l_m\times d_m},$$

[0192] $$M_1(i,j)=\begin{cases}0, & j<n\cup j\le i+n\\ -\infty, & \text{otherwise}\end{cases}$$

[0193] 其中$W_{q,k,v}$,$b_{q,k,v}$,$W_3\in R^{dm\times 4dm}$,$W_4\in R^{4dm\times dm}$,b_3,b_4是可学习参数,LayerNorm是层归一化层(Ba et al.,2016)。解码器块堆叠N次以获得最终的隐藏表示H_N,并输出目标序列的最终表示H^N_y。混合注意力在编码源特征时的时间复杂度为$O(n^2)$,但由于目标特征的因果掩蔽,本申请可以在生成目标标记时缓存这部分的隐藏表示,从而保持时间复杂度每个解码步骤的O(n)。

[0194] 需要说明的是,对于前述的各方法实施例,为了简单描述,故将其都表述为一系列的动作合并,但是本领域技术人员应该知悉,本发明并不受所描述的动作顺序的限制,因为依据本发明,某些步骤可以采用其他顺序或者同时进行。其次,本领域技术人员也应该知悉,说明书中所描述的实施例均属于优选实施例,所涉及的动作和模块并不一定是本发明所必须的。在上述实施例中,对各个实施例的描述都各有侧重,某个实施例中没有详述的部分,可以参见其他实施例的相关描述。

[0195] 在一些实施例中,本发明实施例提供一种非易失性计算机可读存储介质,所述存储介质中存储有一个或多个包括执行指令的程序,所述执行指令能够被电子设备(包括但不限于计算机,服务器,或者网络设备等)读取并执行,以用于执行本发明上述任一项从文本中抽取图的方法。

[0196] 在一些实施例中,本发明实施例还提供一种计算机程序产品,所述计算机程序产品包括存储在非易失性计算机可读存储介质上的计算机程序,所述计算机程序包括程序指令,当所述程序指令被计算机执行时,使所述计算机执行上述任一项从文本中抽取图的方法。

[0197] 在一些实施例中,本发明实施例还提供一种电子设备,其包括:至少一个处理器,以及与所述至少一个处理器通信连接的存储器,其中,所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行从文本中抽取图的方法。

[0198] 在一些实施例中,本发明实施例还提供一种存储介质,其上存储有计算机程序,其特征在于,该程序被处理器执行时实现从文本中抽取图的方法。

[0199] 图10是本申请另一实施例提供的执行从文本中抽取图的方法的电子设备的硬件结构示意图,如图10所示,该设备包括:

[0200] 一个或多个处理器1010以及存储器1020,图10中以一个处理器1010为例。

[0201] 执行从文本中抽取图的方法的设备还可以包括:输入装置1030和输出装置1040。

[0202] 处理器1010、存储器1020、输入装置1030和输出装置1040可以通过总线或者其他

方式连接,图10中以通过总线连接为例。

[0203] 存储器1020作为一种非易失性计算机可读存储介质,可用于存储非易失性软件程序、非易失性计算机可执行程序以及模块,如本申请实施例中的从文本中抽取图的方法对应的程序指令/模块。处理器1010通过运行存储在存储器1020中的非易失性软件程序、指令以及模块,从而执行服务器的各种功能应用以及数据处理,即实现上述方法实施例从文本中抽取图的方法。

[0204] 存储器1020可以包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需要的应用程序;存储数据区可存储根据从文本中抽取图的装置的使用所创建的数据等。此外,存储器1020可以包括高速随机存取存储器,还可以包括非易失性存储器,例如至少一个磁盘存储器件、闪存器件、或其他非易失性固态存储器件。在一些实施例中,存储器1020可选包括相对于处理器1010远程设置的存储器,这些远程存储器可以通过网络连接至从文本中抽取图的装置。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

[0205] 输入装置1030可接收输入的数字或字符信息,以及产生与从文本中抽取图的装置的用户设置以及功能控制有关的信号。输出装置1040可包括显示屏等显示设备。

[0206] 所述一个或者多个模块存储在所述存储器1020中,当被所述一个或者多个处理器1010执行时,执行上述任意方法实施例中的从文本中抽取图的方法。

[0207] 上述产品可执行本申请实施例所提供的方法,具备执行方法相应的功能模块和有益效果。未在本实施例中详尽描述的技术细节,可参见本申请实施例所提供的方法。

[0208] 本申请实施例的电子设备以多种形式存在,包括但不限于:

[0209] (1)移动通信设备:这类设备的特点是具备移动通信功能,并且以提供话音、数据通信为主要目标。这类终端包括:智能手机(例如iPhone)、多媒体手机、功能性手机,以及低端手机等。

[0210] (2)超移动个人计算机设备:这类设备属于个人计算机的范畴,有计算和处理功能,一般也具备移动上网特性。这类终端包括:PDA、MID和UMPC设备等,例如iPad。

[0211] (3)便携式娱乐设备:这类设备可以显示和播放多媒体内容。该类设备包括:音频、视频播放器(例如iPod),掌上游戏机,电子书,以及智能玩具和便携式车载导航设备。

[0212] (4)其他具有数据交互功能的电子装置。

[0213] 以上所描述的装置实施例仅仅是示意性的,其中所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。

[0214] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到各实施方式可借助软件加通用硬件平台的方式来实现,当然也可以通过硬件。基于这样的理解,上述技术方案本质上或者说对相关技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品可以存储在计算机可读存储介质中,如ROM/RAM、磁碟、光盘等,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行各个实施例或者实施例的某些部分所述的方法。

[0215] 最后应说明的是:以上实施例仅用以说明本申请的技术方案,而非对其限制;尽管

参照前述实施例对本申请进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围。

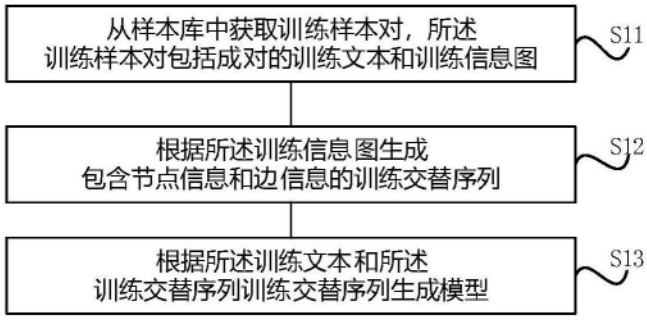

从样本库中获取训练样本对，所述
训练样本对包括成对的训练文本和训练信息图
S11
根据所述训练信息图生成
包含节点信息和边信息的训练交替序列
S12
根据所述训练文本和所述
训练交替序列训练交替序列生成模型
S13

图1

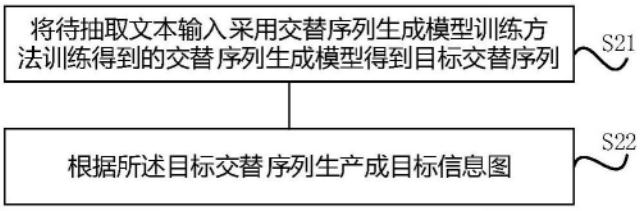

将待抽取文本输入采用交替序列生成模型训练方
法训练得到的交替序列生成模型得到目标交替序列
S21
根据所述目标交替序列生产成目标信息图
S22

图2

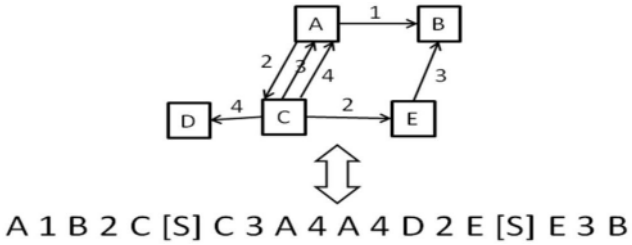

A
B
C
D
E
1
2
3
4
A 1 B 2 C [S] C 3 A 4 A 4 D 2 E [S] E 3 B

图3

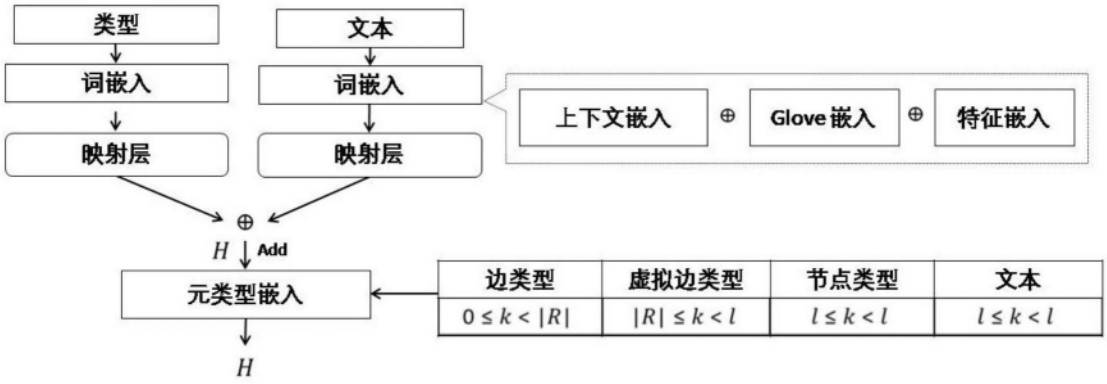

类型
文本
词嵌入
词嵌入
映射层
映射层
上下文嵌入
Glove嵌入
特征嵌入
H
Add
元类型嵌入
H
边类型
虚拟边类型
节点类型
文本
0 ≤ k < |R|
|R| ≤ k < l
l ≤ k < l
l ≤ k < l

图4

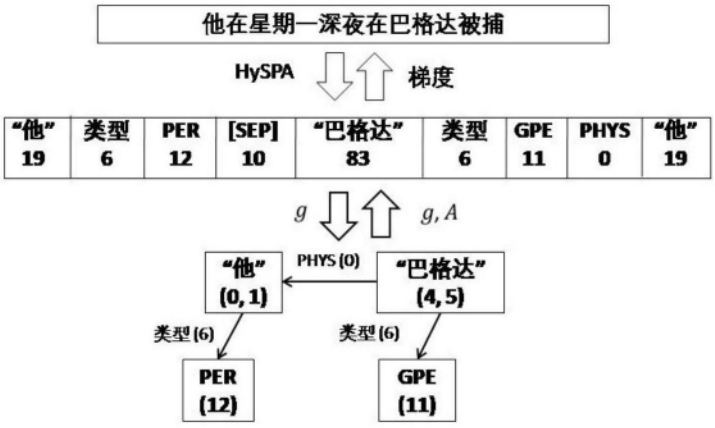

他在星期一深夜在巴格达被捕
HySPA
梯度
"他" 19
类型 6
PER 12
[SEP] 10
"巴格达" 83
类型 6
GPE 11
PHYS 0
"他" 19
g
g, A
"他" (0, 1)
PHYS (0)
"巴格达" (4, 5)
类型 (6)
类型 (6)
PER (12)
GPE (11)

图5

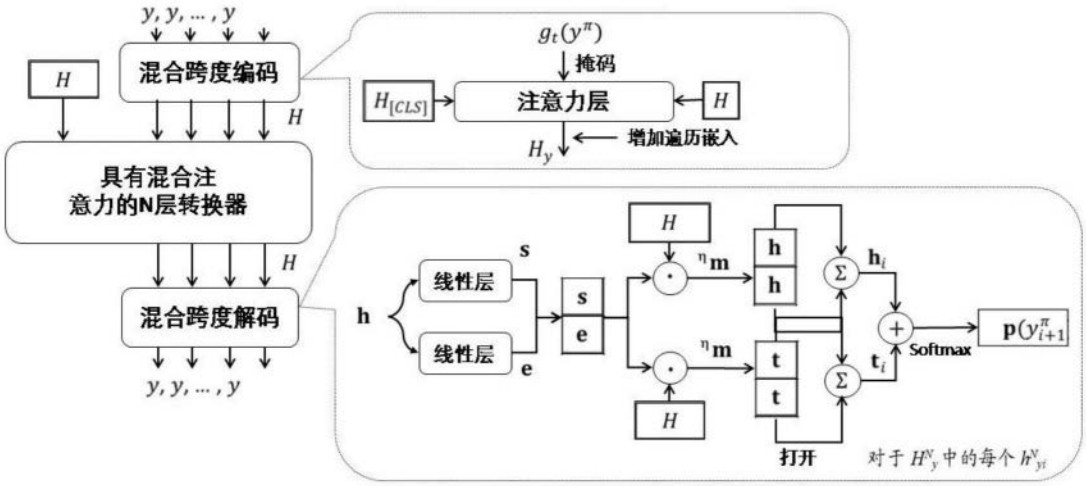

y, y, ... , y
H
混合跨度编码
H
具有混合注
意力的N层转换器
H
混合跨度解码
y, y, ... , y
$g_t(y^\pi)$
掩码
$H_{[CLS]}$
注意力层
H
H_y
增加遍历嵌入
H
s
线性层
线性层
e
h
s
e
η m
η m
h
h
t
t
H
h_i
t_i
Softmax
$p(y^\pi_{i+1})$
打开
对于 H^N_y 中的每个 h^N_{yi}

图6

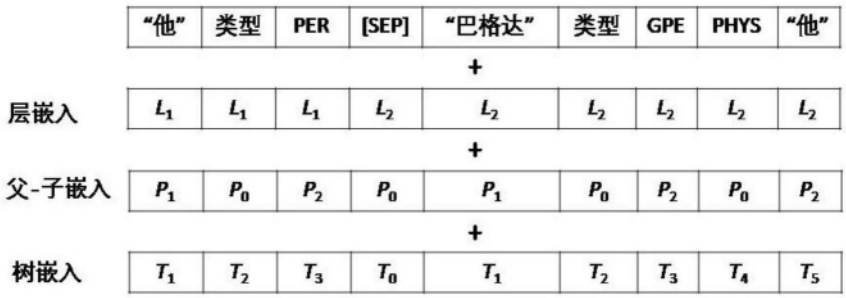

"他" 类型 PER [SEP] "巴格达" 类型 GPE PHYS "他"
+
层嵌入 L_1 L_1 L_1 L_2 L_2 L_2 L_2 L_2 L_2
+
父-子嵌入 P_1 P_0 P_2 P_0 P_1 P_0 P_2 P_0 P_2
+
树嵌入 T_1 T_2 T_3 T_0 T_1 T_2 T_3 T_4 T_5

图7

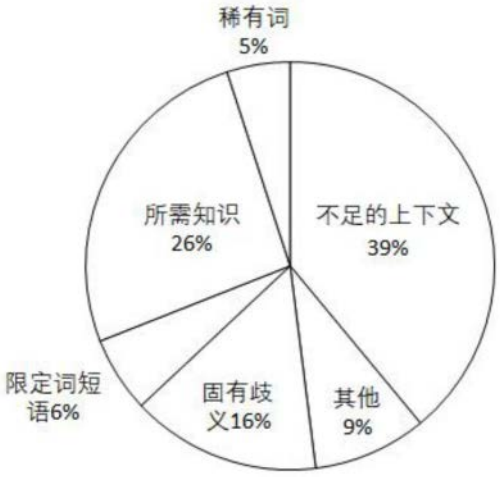

稀有词
5%
不足的上下文
39%
所需知识
26%
限定词短
语6%
固有歧
义16%
其他
9%

图8

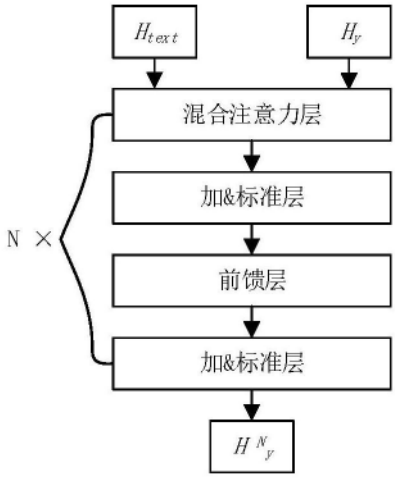

H_{text}
H_y
混合注意力层
加&标准层
N ×
前馈层
加&标准层
H^N_y

图9

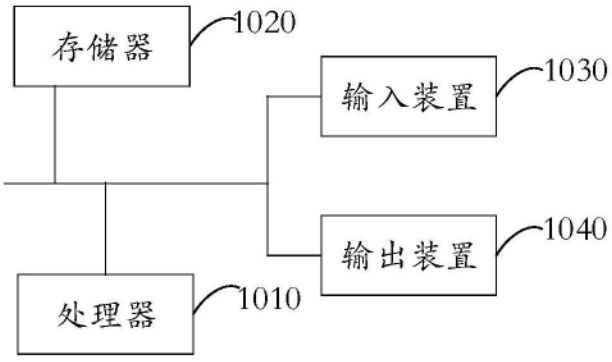

存储器
1020
输入装置
1030
输出装置
1040
处理器
1010

图10