文章编号：1003-0077(2020)06-0018-09

基于序列图模型的多标签序列标注

王少敬，刘鹏飞，邱锡鹏

（复旦大学 计算机学院，上海 201203）

摘 要：该文针对实际中存在对同一句话标注多种序列标签问题，定义了多标签序列标注任务，并提出了一种新的序列图模型。序列图模型主要为了建模两种依赖关系：不同单词在时序维度上面的关系和同一单词在不同任务之间的依赖关系。该文采用 LSTM 或根据 Transformer 修改设计的模型处理时序维度上的信息传递。同一单词在不同任务之间使用注意力机制处理不同任务之间的依赖关系，以获得每个单词更好的隐状态表示，并作为下次递归处理的输入。实验表明，该模型不仅能够在 Ontonotes 5.0 数据集上取得更好的结果，而且可以获取不同任务标签之间可解释的依赖关系。

关键词：多标签序列标注；多任务学习；图模型

中图分类号：TP391 **文献标识码**：A

Sequential Graph Neural Networks for Multi-Label Sequence Labeling

WANG Shaojing, LIU Pengfei, QIU Xipeng

(School of Computer Science, Fudan University, Shanghai 201203, China)

Abstract: Aims at the problem of labeling multiple sequence labels in the same sentence, we propose a new sequence graph model. The sequence graph model is to capture two main kinds of dependencies: one is the relationship between the time series dimensions of different words, and the other is to unify the dependence of words on different tasks. We adopt LSTM or Transformer-like structure to model information interactions in a time series dimension. And we use attention mechanism at each step to model the interaction between different tasks and obtain a better representation of each word. The experimental results show that our model can not only achieve better performance at Ontonotes 5.0, but also can recover interpretable structures between different task labels.

Keywords: multi-labeling sequence learning; multi-task learning; graph model

0 引言

序列标注任务是自然语言处理领域的一项基本任务[1-3]，主要是对序列中的每个单元标注相应标签。这种标注任务可以看成若干个独立分类任务的组合，常见序列标注任务包括命名体识别(named entity recognition, NER)、词性标准(part of speech, POS)、语义角色标准(semantic role labeling, SRL)和组块分析(chuncking analysis, CHUNK)等。在实际应用中，有时候我们需要对同一个句子标注多种类型的标签，比如一个句子可能需要同时标注 POS 和 NER 两种标签，我们把这种任务定义为多标签序列标注(multi-label sequence labeling, MLSL)任务。在非神经网络方面有过一些关于 MLSL 任务的探索[4-7]，但近年来神经网络方面的研究不多。MLSL 的难点在于如何建立不同任务标签之间的关联。最近一些研究工作[8-10]引入多任务学习的方法来处理 MLSL 的任务，比如通过共享输入层[8]或者共享隐藏层[9]等来联合训练不同的序列标注任务。但上述多任务学习的方法是针对更加广义的多标签序列标注任务，而不是同时对一句话进行多种标注。广义的多标签序列标注主要是在不同任务之间共享模型的某些部分，以引入归纳偏向，从而学到更好的表示。但这种多任务学习的方法，往往是不同任务单独训练，对一个句子同时

收稿日期：2019-11-22 **定稿日期**：2020-02-11

基金项目：国家自然科学基金(61672162)

具有多种标签的情况并不合适。

根据动力系统研究来看，一个目标的移动轨迹应该根据当前的观察去预测。比如天气预测当中，我们会不断根据最新观测到的天气状况，更新模型中输入的状态。本文由此获得灵感，并结合多图学习的角度看 MLSL 问题。具体来讲，在序列的每一步构建图模型，以捕捉不同任务标签之间的依赖关系。而在时序上，采用循环神经网络结构来构建不同图之间更加复杂的依赖关系。这种做法可以在产生每一步新的状态时都能利用到所有的任务关系。

本文主要贡献为：在多标签序列标注任务上提出了基于 LSTM 的序列图模型(LSTMGraph)和基于 Transformer 的序列图模型(TransGraph)；在 Ontonotes 5.0 [11]数据集上的评测表明，本文模型具有更高的准确率，并且对知识共享过程提供了解释性的分析。

1　相关工作

1.1　多任务学习

多任务学习是通过学习不同任务之间的共享表示以提高模型泛化能力的方法。多任务学习、迁移学习与联合学习等都是知识迁移的一种。采用多任务学习的方式训练模型，可以潜在地获取更多数据[12]，可以把注意力放在比较重要的特征上面或者辅助其他任务学习到某些特征[13]，比如有些特征在任务 A 中难以学习，在任务 B 中容易学习。由于每个任务都给出监督信号，所以多任务学习会产生倾向于满足多个任务的一种表示，同样也是一种带有归纳偏向的表示。多任务学习可以方便地学到更加泛化的表示，不同任务之间的相互约束具有正则化的作用。如比较常见的硬共享模型[9]结构是先共享隐藏层，然后每个任务连接私有的输出层，这种结构可以很明显地降低过拟合风险，增强模型的泛化能力。

近年来，若干模型[10,14-16]尝试将多任务学习应用于序列标注任务中，并取得了不错的效果，在 4.2 节的对比模型部分详细描述了这些多任务模型。

1.2　相关网络结构

本文设计序列图模型时，使用了 LSTM、Transformer 和注意力机制，下面会分别对每个模块进行介绍。

1.2.1　LSTM

LSTM[17]是一种循环神经网络结构。其与前反馈神经网络相比，可以更方便高效地处理序列型数据，如语音、视频等。LSTM 与最基础的 RNN 不同，它增加了门控结构，这些差异帮助 LSTM 可以记忆更多的内容，避免梯度消失问题获得更好的效果。图 1 为 LSTM 结构图，其中输入门 i_t，输出门 o_t，遗忘门 f_t，这三个门可以帮助模型记住更多有用的信息。

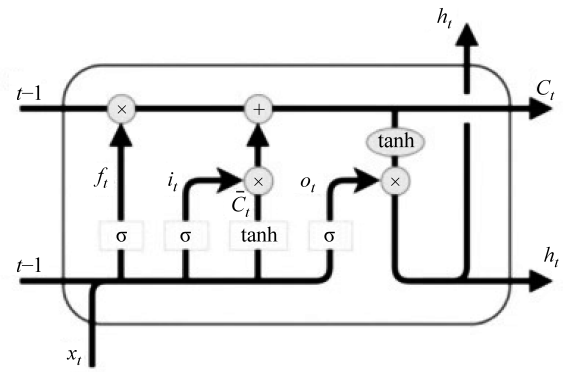


图 1　LSTM 结构图

1.2.2　Transformer

2017 年 Ashish Vaswani 等人[18]首次提出 Transformer 结构，随后两年在机器翻译[19]、语言模型[20-21]等领域出现了许多基于 Transformer 的模型。Transformer 结构在输入到输出变换过程中只依赖于自注意力机制。相较于采用循环方式编码输入的 LSTM，Transformer 结构的优点是可以并行处理，因此 Transformer 模型的处理速度比 LSTM 提高了很多。虽然 LSTM 等模型处理序列数据的过程很符合直觉，但是存在长距离依赖问题[22]，而 Transformer 中的注意力机制可以感知到序列全部的元素，从根本上解决了长距离依赖问题。Transformer 的参数往往比 LSTM 要多，网络容量比较大，训练所需要的数据也比较多。因此 Transformer 更加适合机器翻译、语言模型等数据量大的任务。

1.2.3　注意力机制

在人类处理系统过程中，注意力机制是指在观察一张图片的时候对不同位置给予不同的关注度；在阅读一句话的时候，对句子中不同的词给予不同的关注度。Bahdanau 等人[23]首先在机器翻译方面引入注意力机制，主要目的是解决记忆遗忘问题，随后注意力机制被广泛用于自动问答、文本分类、依存

句法分析、语言推理等领域。

2 任务定义

本文的模型主要针对一种新的任务，下面将对该任务进行介绍。

2.1 多标签序列标注任务(MLSL)

序列标注是一个预测输入文本序列每个位置单词对应标签的任务。定义输入的文本序列为 $X=\{x_1,x_2,\cdots,x_T\}$，定义单词对应的标签为 $Y=\{y_1,y_2,\cdots,y_T\}$。当同一个句子拥有多种任务标签时，就构成了多标签序列标注任务。定义训练数据为 $D=\{(x_t,y_t^1,y_t^2,\cdots,y_t^K)\}_{t=1}^{T}$，其中 K 表示任务数量，T 为句子的长度。条件概率 $P(Y^1,Y^2,\cdots,Y^K\mid X)$，$1\leqslant k\leqslant K$。$X$ 和 Y^k 分别表示一个句子和对应任务 k 的标签。多标签序列标注任务的目标是利用不同任务之间的互补性，提高序列预测模型的性能。

2.2 图结构的 MLSL

多标签文本的处理过程可以理解成一个动态系统：不同任务的标签可能带来不同类型的相互作用，实时利用所有相互作用，建模不同任务的隐状态。

多标签序列标注过程主要考虑两种相互作用：在一个句子不同单词间的相互作用；同一个单词上面不同任务间的相互作用。前者是时序维度上的相互作用，可以理解成横向维度的相互作用；后者是不同任务间的相互作用，可以理解成纵向的相互作用。我们经常独立地处理不同的序列标注任务，却对不同任务之间的相互作用研究甚少，并且建模这种多标签序列标注类型的动态系统也比较困难。

图网络[24-26]是一种理论基础优良的学习动态相互作用的系统模型。我们提出采用序列图模型去建模不同任务之间的关系，同时根据已经观察到的序列学习建模动态系统。

具体来讲，对给定文本序列的每一步，我们定义一个图 $G_t=(V_t,E_t)$，在这里 V_t 表示一系列的节点 $\{v_t^{(1)},v_t^{(2)},\cdots,v^{(K)}\}$，$K$ 表示任务的数量。E_t 是一系列的边，边上面的权重大小表示不同任务之间相互作用的强弱。对于节点 $v_t^{(i)}$ 和 $v_t^{(j)}$ 之间连接边的权重表示为 $e_t^{(i,j)}$。本文使用有向图建模不同任务标签之间的交互，边 $e_t^{(i,j)}$ 的定义的是节点 $v_t^{(i)}$ 到节点 $v_t^{(j)}$ 的作用影响。根据上面的定义，对于一个给定的序列 $X=\{x_1,x_2,\cdots,x_T\}$，将获得一个序列图 $G=\{G_1,G_2,\cdots,G_T\}$。序列图模型在 t 时间步的结构如图 2 所示。

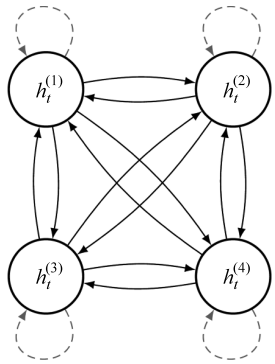


图 2 序列图模型在 t 时间步的结构

3 模型设计

本文模型的主要结构是：输入层，序列图模型，CRF 层。下面将对本文模型结构进行详细介绍。

3.1 输入层

对于输入的一句话 $X=(x_1,x_2,\cdots,x_T)$，每个单词首先被转化成实数向量 $\boldsymbol{x}_t$。对于任意一个单词 $\boldsymbol{x}_t$，K 个节点 $(v_t^{(1)},v_t^{(2)},\cdots,v_t^{(K)})$ 将被用来表示 K 个不同任务的输入隐状态。我们定义 c_t 表示字符级别的特征信息。本文的单词输入隐状态 $v_t^{(k)}$ 主要包括词向量和字符的特征信息两个部分，如式(1)所示。

$$v_t^{(k)}=\mathrm{MLP}(x_t,c_t) \tag{1}$$

3.2 序列图模型

序列图模型主要包括两种图结构：一种用来建模不同任务之间的关系；另一种用来建模时间序列维度层面的关系。

图 2 是 t 时刻根据消息传递[27-29]的方式获取不同任务之间的相互作用的有向图。

定义 $h_t^{(k)}$ 表示 t 时刻的 k 任务的隐状态，$r_t^{(k)}$ 表示对于任务 k 在第 t 时刻信息聚合后的状态表示。每个节点可以直接的发送(或者接受)信息到(从)其他节点。$r_t^{(k)}$ 可以通过式(2)得到，其中 $\boldsymbol{q}=\boldsymbol{h}_t^{(k)}\boldsymbol{W}^Q$，$\boldsymbol{K}=\boldsymbol{H}\boldsymbol{W}^K$，$\boldsymbol{V}=\boldsymbol{H}\boldsymbol{W}^V$。此处的 $\boldsymbol{W}^Q$、$\boldsymbol{W}^K$ 和 $\boldsymbol{W}^V$ 都是可以学习的变量，$\boldsymbol{H}=[h_t^{(1)},\cdots,h_t^{(K)}]$。

$$r_t^{(k)} = \text{softmax}\left(\frac{\boldsymbol{q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d}}\right)\boldsymbol{V} \tag{2}$$

在时间序列维度上使用循环神经网络结构来建模前后单词之间的关系，如式(3)所示，其中 θ 表示循环神经网络中所有的参数，$h_{t+1}^{(k)}$ 表示下一个时刻的聚合前的隐状态表示，$v_t^{(k)}$ 表示单词的特征输入，$r_t^{(k)}$ 表示当前聚合后的隐状态表示。

$$h_{t+1}^{(k)} = \text{RNN}^k(v_t^{(k)}, r_t^{(k)}, \theta) \tag{3}$$

实际应用中我们设计了两种递归结构来学习这种递归时序依赖关系：LSTM 和 类似 Transformer 的循环单元结构。下面将会单独介绍基于这两种结构得到的 LSTMGraph 模型和 TransGraph 模型。

3.2.1　基于 LSTM 的序列图模型(LSTMGraph)

LSTMGraph 使用 LSTM 建模不同时刻之间的转移关系，即使用 LSTM 建模不同图之间的关系。图中 Inter-action LSTM 表示任务之间有信息交互的递归神经网络模型，每步循环过程包括：使用 LSTMCell 建立前后单词之间的信息传递；使用注意力机制建立不同任务之间的关系(图 3)。

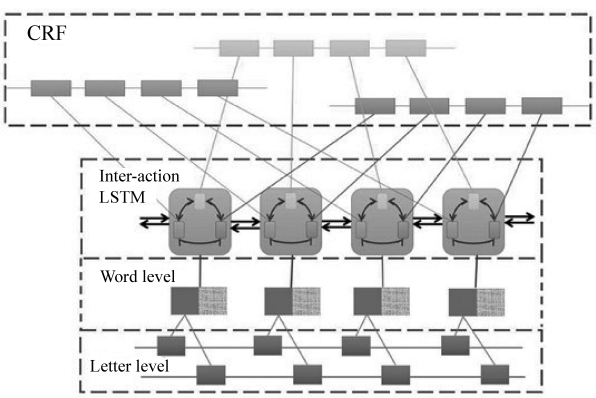


图 3　LSTMGraph 的模型结构图

不同任务对应一个不同的 LSTMCell^k，如式(4)所示，LSTMCell^k 的输入是单词的特征信息 $v_t^{(k)}$ 和最新的聚合后隐状态的信息 $r_t^{(k)}$，输出是下一时刻聚合前的隐状态表示 $h_{t+1}^{(k)}$。随后输出的隐状态 $h_{t+1}^{(k)}$ 将通过式(5)做不同任务之间的交互，获取最新节点的各种任务隐状态的表示 $r_{t+1}^{(k)}$，并作为下次 LSTMCell^k 循环的输入，执行此操作循环到句子结束。最后把当前最新的隐状态 $R=[r_1, r_2, \cdots, r_T]$ 作为基于特定任务后续分类器的输入。

$$h_{t+1}^{(k)} = \text{LSTMCell}^{(k)}(r_t^{(k)}, v_t^{(k)}) \tag{4}$$

$$r_{t+1}^{(k)} = \text{SA}(h_{t+1}^{(1)}, \cdots, h_{t+1}^{(K)}) \tag{5}$$

3.2.2　基于 Transformer 的序列图模型(TransGraph)

TransGraph 模型使用类似 Transformer 的结构来建模时间序列维度上前后单词之间的依赖关系。我们使用多头注意力机制捕捉不同任务之间的关系，于是联想到也使用类似的结构来捕捉时序维度上面的关系。具体来讲是使用一种类似 Transformer 结构的模型替换 LSTMCell 来建模同一个任务内部不同时刻间的转移关系。TransGraph 结构的输入和 LSTMCell 的输入相同，是单词的特征信息 v_t 和不同任务对这个词的最新隐状态 $r_t^{(k)}$。将两个向量通过式(6)，得到中间隐变量 $z_{t+1}^{(k)}$。后面使用注意力机制计算不同任务之间的相互作用部分和 LSTMGraph 相同。除此之外模型第一个输入隐状态 $r_0^{(k)}$ 是一个可以通过反向梯度学习的变量，这种方式比初始为全零向量的效果略好一点。

$$z_{t+1}^{(k)} = h_t^{(k)} + \text{MLP}(r_t^{(k)} \oplus v_t^{(k)}) \tag{6}$$

$$h_{t+1}^{(k)} = \text{norm}(z_{t+1}^{(k)}) \tag{7}$$

式(7)中，norm(·)表示层归一化，MLP(·)表示多层感知器。

3.3　输出层

每一步隐状态的输出将会被输入到此任务对应的分类器以及 CRF[30] 层，然后做对应任务的预测。训练时不同任务同时执行反向梯度回传。

4　实验结果

为了验证序列图模型的有效性，我们在 Ontonotes 5.0 数据集上统计实验结果，并对其进行定性分析。

4.1　数据集

我们在 Ontonotes 5.0[11] 上面测试本文模型。Ontonotes 5.0 是 2013 年从电报(TC)、新闻专线(NW)、广播新闻(BC)、博客(WB)、要点评论(PT)和杂志类型的文章(MZ)，这 7 种类型的文本当中收集而来的数据库。Ontonotes 5.0 包含了之前所有的版本，并且分为中文、英语、阿拉伯语三个版本。数据集中的每个句子同时具有 POS、NER、SRL 和 CHUNK 等多个任务标签，非常适合用来验证本文模型的能力。因为 PT 类型的数据缺少 NER 标签，所以我们只使用了剩下的 6 个数据集。数据集参数如表 1 所示。

表1 数据集参数

数据集	训练集	验证集	测试集
BC	185 168	32 073	38 158
BN	217 582	26 566	27 781
MZ	171 128	16 063	18 654
NW	913 192	153 851	63 083
WB	405 802	51 709	54 531
TC	103 292	12 834	12 281
汇总	1 996 164	293 096	214 488

4.2 对比模型

为了验证本文模型的效果,本文与若干相关模型进行了对比实验。

SAS-MTL[10]通过使用强化学习自适应地选择应该共享的层。思想源自基于强化学习搜索网络结构的方法[31]。

CS-MTL[14]是一种软共享的方式,通过线性组合不同任务的隐状态获得新的隐状态完成交互。

LA-MTL[15]是一种闸门控制共享的网络结构。

SSP-MTL[16]是一种比较经典的多任务学习框架,模型结构是先共享编码模块,然后不同任务接不同的输出模块。

Single[32]是序列标注任务领域经典的Bi-LSTM-CRF模型。Bi-LSTM-CRF与本文提出的序列图模型LSTMGraph和TransGraph之间的区别是,没有不同任务之间的信息交互,只在单一任务上面进行训练。

PS-MTL是本文实现的硬共享机制的多任务学习模型,模型结构为输入层LSTM共享,随后为针对特定任务的LSTM层和CRF层。

MultiGraph该模型是本文针对LSTMGraph做的对比模型,和LSTMGraph的区别是:LSTMGraph中任务间的交互是在时序上面的每一步都执行一次,而Multi-Graph是在时序上面LSTM执行结束后进行交互。

4.3 实验结果

表2展示了在Ontonotes 5.0数据集上面的各种模型准确率的实验结果。整体来看,本文提出的LSTMGraph模型和TransGraph模型在大部分语料上以及各种任务上都取得了非常不错的结果。具体来讲,我们在24个实验(4种序列标注任务,6种类型的数据集)中的20个取得了最好结果。

表2 所有模型在不同任务上面的准确率实验结果

models	CHUNK	POS	NER	SRL	Avg	CHUNK	POS	NER	SRL	Avg	CHUNK	POS	NER	SRL	Avg
	BC					BN					MZ				
Single	88.9	96.4	96.3	97.3	94.7	90.0	97.1	96.4	97.5	95.3	89.6	95.9	96.2	97.9	94.9
SAS-MTL	91.2	94.6	94.5	94.6	93.7	91.3	95.3	93.7	97.4	94.4	91.0	92.9	98.2	93.9	94.0
CS-MTL	86.2	94.2	93.0	97.8	92.8	86.0	92.5	89.7	97.7	91.5	91.2	84.6	81.8	98.0	88.9
LA-MTL	91.7	/	93.5	95.6	/	92.9	/	94.2	95.3	/	92.9	/	93.5	96.1	/
SSP-MTL	89.8	94.7	94.5	98.1	94.3	91.1	95.2	93.1	97.7	94.3	90.4	92.8	93.2	97.9	93.6
MultiGraph	90.1	96.8	97.0	97.9	95.3	91.3	97.4	96.9	98.1	95.9	90.9	96.0	97.3	98.4	95.7
LSTMGraph	**91.9**	**97.0**	**97.2**	**98.2**	**96.1**	**92.9**	97.6	96.8	**98.4**	**96.4**	**92.9**	**96.5**	97.4	**98.7**	**96.4**
TransGraph	90.7	96.9	97.1	98.0	95.7	91.7	**97.7**	**97.2**	98.3	96.2	92.8	96.5	97.6	98.6	96.4
models	CHUNK	POS	NER	SRL	Avg	CHUNK	POS	NER	SRL	Avg	CHUNK	POS	NER	SRL	Avg
	NW					WB					TC				
Single	93.0	97.4	96.5	97.3	96.1	91.1	95.8	96.4	98.1	95.4	88.2	95.8	96.8	96.4	94.3
CS-MTL	92.3	96.2	95.5	96.8	95.2	88.0	90.1	**98.2**	95.8	93.0	87.4	94.2	97.2	98.8	94.4
SAS-MTL	92.3	95.0	93.9	**97.8**	94.8	92.5	92.7	96.6	96.4	94.6	91.4	95.0	96.0	99.1	95.4
LA-MTL	94.3	/	95.5	97.7	/	**94.4**	/	92.5	**98.0**	/	91.0	/	94.3	98.6	/
SSP-MTL	91.0	94.8	93.6	96.7	94.0	91.6	93.2	96.5	96.2	94.4	90.2	94.6	95.4	98.8	94.8
MultiGraph	93.6	97.6	96.7	97.6	96.4	92.1	95.7	97.4	97.7	95.7	89.8	95.6	96.6	99.0	95.4
LSTMGraph	**94.6**	**97.7**	96.8	97.8	96.7	93.5	**96.4**	97.3	97.6	96.2	**91.7**	96.3	96.9	99.1	**96.0**
TransGraph	93.8	97.6	**97.0**	97.5	96.5	92.8	**96.4**	97.6	**98.0**	96.2	90.8	**96.4**	**97.3**	**99.2**	95.9

从表 2 中可以得出，序列图模型在 POS 和 CHUNK 两个任务中，性能提升非常明显，POS 在 6 个数据集上取得了最好结果，CHUNK 在 6 个数据集中的 5 个点取得了最好结果。

通过对比在所有模型上面的平均值发现，LSTMGraph比其余所有模型效果都好，说明使用序列图机制不仅可以在部分任务中取得不错结果，而且整体平均值相对较高。当我们把所有任务的准确率加和平均后比较发现，LSTMGraph 平均值比传统多任务模型高 2.1、比单模型高 1.2、比 Multi-Graph 模型高 0.6，比 TransGraph 高 0.2。从中可以得出，本文提出的模型相对于传统的多任务学习，在处理多标签序列标注任务上取得了更好的效果。其他多任务学习模型容易出现某个指标很高，但整体平均水平较差的现象。

本文提出的基于注意力机制信息交互的 Multi-Graph、LSTMGraph 和 TransGraph 模型，相比其他多任务模型平均值要高很多，也说明这种注意义力机制方式，在多标签序列标注任务中，可以更高效地处理不同任务之间的信息交互。

此外 TransGraph 和 LSTMGraph 与 Multi-Graph 对比发现，使用这种序列图机制，即在循环神经网络的每一步做信息交互，比循环神经网络处理完成后再统一做信息交互效率更高。也即本文提到的动力学中，根据当前状态实时计算相互作用更加高效。

表 3 是汇总全部数据集上面的实验结果，从中可以看到 TransGraph 模型在汇总的数据集上取得了很好的结果。尤其是 F_1-score 评价指标提升很明显。因为标签之间可能存在比较大的不平衡性，与其他评价指标相比，使用 F_1-score 指标评测能够更准确地反映模型的效果。TransGraph 与 LSTM-Graph 唯一的区别是，TransGraph 在时序维度上使用一种仿照 Transformer 的递归单元替换了 LSTMCell，这表明在时序维度上面使用类似 Transformer 的变体可以提高模型的效果，尤其是当数据集比较大的时候，这种效果更加明显。LST-MGraph、TransGraph 相对于采用相同注意力机制建立任务之间依赖关系但不是实时进行信息交互的 Multi-Graph 模型相比，采用序列图模型能够学到更好的表示，从而模型效果也相对更好。

表 3　在汇总全部数据集上面的实验结果

Model	POS	CHUNK		NER		SRL	
	Acc	Acc	F_1	Acc	F_1	Acc	F_1
Single	96.39	92.36	88.83	96.83	80.79	96.91	86.88
PS-MTL	95.86	91.57	87.64	96.55	79.4	96.77	85.89
Multi-Graph	96.17	91.79	87.9	96.56	79.71	96.81	86.09
LSTMGraph	96.55	92.93	89.51	96.96	82.11	96.77	86.11
TransGraph	**97.14**	**93.71**	**91.04**	**97.08**	**82.21**	**97.30**	**88.70**

4.4　定性分析

为了方便解释序列图模型，我们绘制了每个时间节点的注意力权重，使用颜色的深浅，表示注意力值的大小。如图 4 所示，当单词 princess 的 NER 标签是 B-PERSON 的时候，单词对应的 POS 标签有比较大的概率是 NNP，即图中两个任务相互作用的值比较大。另外从图中可以看出 POS 和 CHUNK 之间具有最强的相互关系，实际上二者任务内容也很接近。除此之外，可以看到 NER 和 CHUNK 经常保持相同的边界。例如，词组 princess diana 在 CHUNK 任务上被标记为 B-NP，I-NP 同时在 NER 上面标注为 B-person，I-person。此外，对于单词 loved，任务 POS、SRL、CHUNK 分别标记为 VBD、B-V、B-VP。在实际数据中，这三种标签共现的概率很大。在每个时间节点，LSTMGraph 建模不同任务之间的相互依赖关系，并作为下个时间节点的初始隐状态，在每步都进行交互的模型可以增加信息交互的效率，学到更加泛化的表示，从而取得更好的结果。

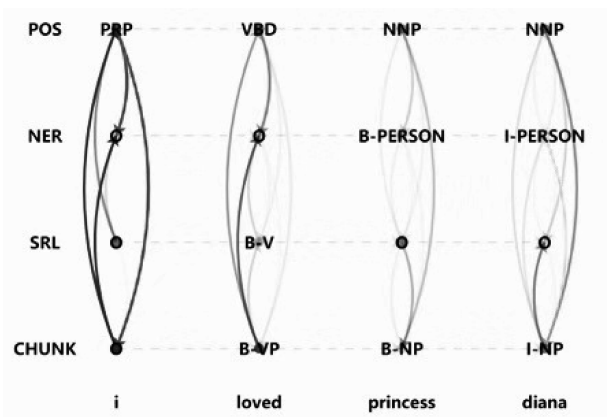


图 4　定性分析不同任务间的交互情况

图 5 是本文在测试集上根据多组数据计算得到的注意力大小的均值。图中每个方格表示从纵坐标

对应的任务传递到横坐标对应任务的信息量，其中颜色的深浅表示注意力的强弱。从图5中可以得知，CHUNK与POS两个任务比较接近，而实际上POS是对词性的标注，而CHUNK是针对短语词组的词性标注，两者具有很强的相关性，而注意力机制计算的结果中两者的相关性也比较大。SRL任务关注更多的是CHUNK的信息，实际上SRL角色标注的主要是名词短语词组或者动词词组，而且和CHUNK具有相同的边界；NER任务主要标注的是实体词组，而实体一般在句子中以名词短语词组的形式存在，因此CHUNK的信息相对来说更重要一些。此外，在上面的例子中，NER标注的实体组，也是CHUNK标注的名词短语词组，它们具有相同的起始边界。

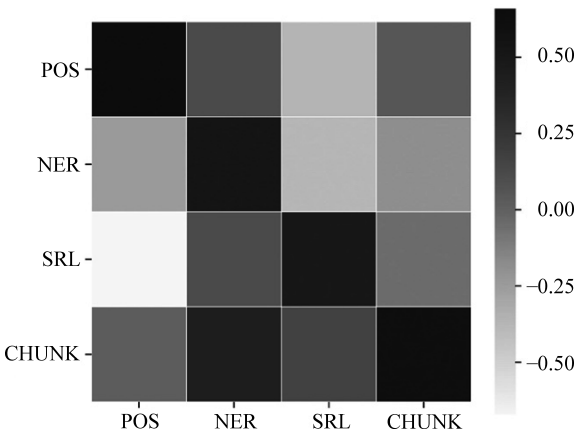


图5 不同任务之间的注意力值

5 总结

本文把对同一句话进行多种标签标注的任务定义为多标签序列标注任务。从动力学系统利用实时的全部信息建模预测下一步状态获得灵感，提出在时序维度的每一步根据当前的最新信息建立图模型。在时序维度使用循环神经网络建模时序维度上的信息交互；在每个单词上使用注意力机制建立不同任务之间的依赖关系。基于在时序维度和不同任务之间的关系，并结合多图学习模型，提出序列图模型。在序列图模型具体实现中，设计了LSTM-Graph和TransGraph两种序列图模型结构。实验结果表明，我们的模型相对于分开训练不同任务的传统多任务学习模型，在Ontonotes 5.0数据集上取得了更好的效果，尤其是在所有任务的平均水平方面效果提升更加明显，与多任务模型中最好的结果相比提升了2.1个百分点，与单任务模型相比提升了1.2个百分点。在汇总所有单一数据集上，TransGraph模型的效果明显好于其他模型。另外在分析注意力机制的注意力大小的时候，分析得到：CHUNK与POS两种功能比较接近的任务上，两者之间的相互作用比较强；NER任务和CHUNK任务标注的短语词组具有相同的起始边界，而实验中两者注意力值相对较大；同样NER实体词往往是名词，因此和POS相关性也较大；SRL标签标注的主要是名词词组和动词，所以和CHUNK相关性较大。

未来工作中，我们将尝试其他节点之间信息交互的策略，比如设计不同节点之间多次迭代交互信息的模型[33-34]，通过学习对于特定任务更有效的隐状态表示，来提高模型的效果。

参考文献

[1] Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging[C]//Proceedings of Conference on Empirical Methods in Natural Language Processing, 1996: 133-142.

[2] Gréegoire Mesnil, Yann Dauphin, Kaisheng Yao, et al. Using recurrent neural networks for slot filling in spoken language understanding[C]//Proceedings of IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2015, 23(3): 530-539.

[3] Dou Shen, Jiantao Sun, Hua Li, et al. Document summarization using conditional random fields[C]//Proceedings of IJCAI, 2007(7): 2862-2867.

[4] Andrew McCallum, Khashayar Rohanimanesh, Charles Sutton. Dynamic conditional random fields for jointly labeling multiple sequences[C]//Proceedings of NIPS-2003 Workshop on Syntax, Semantics and Statistics, 2003.

[5] Yanxin Shi, Mengqiu Wang. A dual-layer CRF based joint decoding method for cascaded segmentation and labeling tasks[C]//Proceedings of IJCAI, 2007: 1707-1712.

[6] Yue Zhang, Stephen Clark. Joint word segmentation and pos tagging using a single perceptron[C]//Proceedings of ACL-08: HLT, 2008: 888-896.

[7] ChenLyu, Yue Zhang, Donghong Ji. Joint word segmentation, pos-tagging and syntactic chunking[C]//Proceedings of AAAI, 2016: 3007-3014.

[8] Ronan Collobert, Jason Weston. A unified architecture

for natural language processing: Deep neural networks with multitask learning[C]//Proceedings of the 25th International Conference on Machine Learning, 2008: 160-167.

[9] Zhilin Yang, Ruslan Salakhutdinov, William Cohen. Multi-task cross-lingual sequence tagging from scratch [J]. arXiv preprint arXiv: 1603.06270, 2016.

[10] Junkun Chen, Kaiyu Chen, Xinchi Chen, et al. Exploring shared structures and hierarchies for multiple NLP tasks[J]. arXiv preprint arXiv: 1808.07658, 2018.

[11] Ralph Weischedel, Martha Palmer, Mitchell Marcus, et al. Ontonotes release 5.0 ldc2013t19[DS]. Linguistic Data Consortium, Philadelphia, PA, 2013.

[12] Caruana R. Multitask Learning[J]. Machine learning, 1997, 28(1): 41-75.

[13] Abu-Mostafa Y S. Learning from hints in neural networks[J]. Journal of Complexity, 1990, 6(2): 192-198.

[14] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, et al. Cross-stitch networks for multi-task learning [C]//Proceedings of Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference, 2016: 3994-4003.

[15] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, et al. Learning what to share between loosely related tasks[J]. arXiv preprint arXiv: 1705.08142, 2017.

[16] Pengfei Liu, Xipeng Qiu, Xuanjing Huang. Adversarial multi-task learning for text classification[J]. arXiv preprint arXiv: 1704.05742, 2017.

[17] Sepp Hochreiter Jürgen Schmidhuber. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need.[C]//Proceedings of Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, 2017: 6000-6010.

[19] Myle Ott, Sergey Edunov, David Grangier, et al. Scaling neural machine translation[C]//Proceedings of the 3rd Conference on Machine Translation (WMT), 2018.

[20] Jacob Devlin, Mingwei Chang, Kenton Lee, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv: 1810.04805, 2018.

[21] Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training [OL]. URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf, 2018.

[22] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[C]//Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[23] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate [C]//Proceedings of the International Conference on Learning Representations, 2015.

[24] Thomas NKipf, Max Welling. Semisupervised classification with graph convolutional networks[J]. arXiv preprint arXiv: 1609.02907, 2016.

[25] Justin Gilmer, Samuel SSchoenholz, Patrick F Riley, et al. Neural message passing for quantum chemistry [C]//Proceedings of ICML, 2017: 1263-1272.

[26] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, et al. Graph attention networks[J]. arXiv preprint arXiv: 1710.10903, 2017.

[27] Herman JC Berendsen, David van der Spoel, Rudi van Drunen. Gromacs: A message-passing parallel molecular dynamics implementation[C]//Proceedings of Computer Physics Communications, 1995, 91(1-3): 43-56.

[28] Bertrand Serlet, Lee Boynton, Avadis Tevanian. Method for providing automatic and dynamic translation into operation system message passing using proxy objects[J]. US Patent, 1996(5): 481, 721.

[29] Pengfei Liu, Jie Fu, Yue Dong, et al. Multi-task learning over graph structures[J]. arXiv preprint arXiv: 1811.10211, 2018.

[30] John Lafferty, Andrew McCallum, Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data [C]//Proceedings of the 18th International Conference on Machine Learning, 2001.

[31] Zoph B, Le Q V. Neural architecture search with reinforcement learning[J]. arXiv preprint arXiv: 1611.01578, 2016.

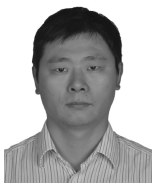
[32] Zhiheng Huang, Wei Xu, Kai Yu. Bidirectional LSTM-CRF models for sequence tagging[J]. arXiv preprint arXiv: 1508.01991, 2015.

[33] Yue Zhang, Qi Liu, Linfeng Song. Sentence-state LSTM for text representation[C]//Proceedings of ACL, 2018: 317-327.

[34] Yongjing Yin, Linfeng Song, Jinsong Su, et al.

Graph-based neural sentence ordering. [C]//Proceedings of IJCAI, 2019: 5387-5393.

王少敬(1993—),硕士研究生,主要研究领域为自然语言处理、深度学习。
E-mail: sjwang17@fudan.edu.cn



邱锡鹏(1983—),通信作者,博士,副教授,主要研究领域为自然语言处理、深度学习。
E-mail: xpqiu@fudan.edu.cn

刘鹏飞(1992—),博士研究生,主要研究领域为自然语言处理、深度学习、多任务学习。
E-mail: pfliu14@fudan.edu.cn

(上接第 17 页)

[33] Paras. Stochastic Gradient Descent[CP/OL]. [2019-07-30]. https://www.math works.com/;matlabcentrol Hibee xcharge/4 347-stochnstic-gradtent-daescent.

孙凯丽(1993—),硕士研究生,主要研究领域为自然语言处理、中文信息处理。
E-mail: sunkaili@mails.ccnu.edu.cn

李源(1971—),通信作者,博士,副教授,硕士生导师,主要研究领域为自然语言处理、中文信息处理、机器学习、软件工程。
E-mail: yuanli@mail.ccnu.edu.cn

邓沌华(1976—),博士,副教授,主要研究领域为中文信息处理、汉语复句信息处理。