基于动态掩蔽注意力机制的事件抽取 *

黄细凤

(中国电子科技集团公司第十研究所, 成都 610036)

摘　要：事件抽取（event extraction）是自然语言处理（natural language processing，NLP）中的一个重要且有挑战性的任务，完成从文本中识别出事件触发词（trigger）以及触发词对应的要素（argument）。对于一个句子中有多个事件的多事件抽取任务，提出了一种注意力机制的变种——动态掩蔽注意力机制（dynamic masked attention network，DyMAN），与常规注意力机制相比，动态掩蔽注意力机制能够捕捉更丰富的上下文表示，并保留更有价值的信息。在 ACE 2005 数据集上进行的实验中，对于多事件抽取任务，与之前最好的模型 JRNN 相比，DyMAN 模型在触发词分类任务上取得了 9.8%的提升，在要素分类任务上取得了 4.5%的提升，表明基于 DyMAN 的事件抽取模型在多事件抽取上能够实现领先的效果。

关键词：事件抽取；注意力机制；多事件抽取

中图分类号：TP306.1　　doi: 10.19734/j.issn.1001-3695.2018.12.0927

Event extraction via dynamic masked attention

Huang Xifeng

(*The* 10*th Research Institute of China Electronics Thchnology Group Corporation*, *Chengdu* 610036, *China*)

Abstract: Event Extraction is an important and challenging task in Natural Language Processing(NLP) , completing the identification of event triggers and their arguments from the text. In this paper, for multiple-event extraction tasks with multiple events in a sentence, a model based on a variant of attention mechanism called dynamic masked attention network (DyMAN) was proposed, which can capture richer context representation and keep more valuable information than the normal attention. The experiments demonstrate that proposed model can achieve state-of-the-art performance on the ACE 2005 corpus.

Key words: event extraction; attention mechanism; multiple-event extraction

0　引言

事件抽取任务可以被简单地表述为：从文本中识别出事件触发词（trigger）以及触发词对应的要素（argument）。早期的事件抽取方法[1~8]大多是通过统计机器学习的方法，将从句子中抽取一些离散的特征（如 POS 标签、分析树等）用于句子表示，这些传统特征依赖于 NLP 工具和复杂的人工设计。近年以来，基于神经网络的事件抽取方法[9,10]在取得了比传统方法更好的结果，这些方法使用神经网络模型，包括卷积神经网络（convolutional neural network, CNN）和递归神经网络（recurrent neural network, RNN）等，从源文本中自动学习出特征表示，避免了人工设计特征的过程。

在事件抽取中，一个句子可能会包含多个事件，即不止一个触发词，并且这些事件可能会共享同一个要素，但要素角色不同。在图 1 的例子中，这个句子有 Die 和 Attack 两个事件，触发词分别是 died 和 fired。这两个事件共享了三个要素：Baghdad、cameraman 以及 American tank，其中 cameraman 在两个事件中充当不同角色：在 Die 事件中充当 Victim 角色，在 Target 事件中充当 Target 角色。

受动态多池化卷积神经网络（DMCNN）[9]和有向自注意力网络（DiSAN）[19]的启发，本文提出了注意力机制的一种变体——动态掩蔽注意力网络（DyMAN），处理上面讨论的多事件抽取问题。使用与 DMCNN 及 DiSAN 中一样的想法，DyMAN 将句子切分成不同的部分，并使用动态掩蔽机制关注同一个句子的多个部分。DyMAN 能够捕捉更加丰富的上下文表示，并保留更多对于事件抽取有价值的信息。实验表明基于 DyMAN 的事件抽取模型在 ACE 2005 数据集上能够实现领先的效果。

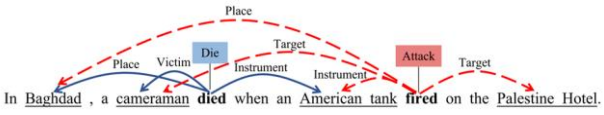

图 1　含有多个事件的句子

Fig. 1　Sentence with multiple events

事件抽取任务最通用的方法是将其视为监督学习框架下的分类问题。为了解决这个问题，过去已经有许多方法被提出,可以从不同角度来回顾这些方法。

事件抽取任务可能会依赖于从文本中构建不同特征。从特征角度，可以将这些方法分为基于传统特征的方法和基于学习特征的方法。在基于传统特征的方法中，许多传统特征（如 POS 标签、分析树、跨文档推理等）被用于提取句子表示，这些传统特征依赖于 NLP 工具和复杂的人工设计。另一方面，基于学习特征的方法通常使用神经网络，从源文本中自动学习出特征表示。

由于事件抽取任务由触发词预测和要素角色预测两个阶段构成，事件抽取方法也需要考虑如何合适地组织这两个阶段。根据两个阶段的组织方式，事件抽取方法可以被分为流水线方法和联合方法。流水线模型将事件抽取任务看做一个两步的任务，并分别进行触发词预测和要素预测[11,12]。与此同时，联合方法将事件抽取作为一个结构预测问题[7,8,10,14]，

收稿日期：2018-12-05；**修回日期：**2019-03-18　　**基金项目：**国家重点研发计划重点专项项目（2017YFB1002104）；国家自然科学基金面上项目（61672162）

作者简介：黄细凤（1984-），女，湖南岳阳人，工程师，博士，主要研究方向为自然语言处理、知识图谱、人工智能(huangsixiao@163.com).

同时预测出触发词和要素。

过去的大多数模型都关注于受限领域（如 ACE 2005 数据集）的事件抽取，其中数据和事件类型都限制在一定范围内。近期的工作开始考虑开放领域的事件抽取问题，并主要集中于使用外部语言学资源来增强训练数据[15]。

对于受限领域事件抽取问题，引入了注意力机制[24]。注意力机制自被提出，已经被应用到许多 NLP 任务中，并取得了更好的效果[16~18]。注意力机制的有效性，可以部分被解释为，它能够忽略单词之间的距离，捕捉到上下文依赖关系；同时，它也能灵活地控制不同部分对于句子表示的贡献。换句话说，注意力机制能够更多地关注到对于具体任务重要的部分，而忽略不重要的部分。由于以上的优势，将注意力机制引入到事件抽取任务中是值得考虑的。

1　事件抽取任务

在本文中主要关注 automatic content extraction (ACE)评测中描述的事件抽取任务。下面介绍几个 ACE 术语来说明事件抽取任务。

事件提及：其中存在一个事件的短语或句子，包含触发词以及对应的要素集合。

事件触发词：最清楚表达事件发生的主要词语。

事件要素：参与事件之中的实体提及、时间表达或数值。

要素角色：要素和对应事件的关系。

ACE 定义了 8 种事件类型以及 33 种子类型，如 Attack、Be-Born、Divorce 等。事件类型同时也是触发词的类型。每个事件触发词都有自己的一些要素，这些要素在该事件中充当特定的角色。例如，事件子类型 Attack 的子类型，可能有角色为 Attacker、Target、Victim、Place、Time 的要素。总共有 36 种事件要素角色，包括一个 Negative 角色，代表某个要素在相应事件中不充当任何角色。

给定一篇文档，事件抽取系统将预测出文档中的事件触发词及其子类型，同时预测出事件对应的要素及其角色。因此，事件抽取过程可以划分为事件触发词预测和要素角色预测两个阶段。

事件抽取依赖前期的实体识别和实体提及共指阶段，在本文中不关注实体识别阶段，只关于事件抽取阶段。因此，遵循前人的工作，直接采用 ACE 标注数据中提供的要素候选词标签（即实体提及、时间表达和数值）。

2　动态掩蔽注意力

注意力机制计算对于给定实体，输入序列中每个元素的重要性分数；然后根据重要性分数，从输入序列中选取相应元素，得到最终的序列编码。在本章将介绍注意力机制的基本概念，以及注意力变体、动态掩蔽注意力。

2.1　注意力

为了阐明注意力的概念，本文探究在给定查询向量 q 输入序列 $X=[x_1,x_2,\cdots,x_n]^T$ 的前提下，如何计算注意力上下文表示。假定查询向量 q 和输入向量 $x_i, i=1,2,\ldots,n$ 都是 d 维的连续词向量表示，即 $q\in R^d, x_i\in R^d$。为了计算查询向量 q 在输入序列 X 上的注意力，应该计算 q 在输入序列每个词 $x_i, i=1,2,\cdots,n$ 上的注意力分数。首先，定义一个标量函数 $f: R^d\times R^d\rightarrow R$ 来计算 q 在 x_i 上的非归一化注意力分数：

$$s_i=f(x_i,q), i=1,2,...,n. \tag{1}$$

注意力分数 s_i 衡量了输入词 x_i 对于查询 q 的重要性，即 q 对 x_i 的注意力，注意力分数 s_i 越大，表示词 x_i 对于查询 q 的重要性越大。

为了得到归一化的注意力分数，softmax 函数被作用于 s_i。

$$a_i=\frac{exp(s_i)}{\sum_{i=1}^{n}exp(s_i)} \tag{2}$$

注意力机制的输出是输入序列 $X\in R^{n\times d}$ 中所有词向量的注意力分数加权求和，即

$$z(X,q)=\sum_{i=1}^{n}a_i x_i \tag{3}$$

可以看出，查询向量 q 在输入序列 X 上的注意力输出是一个与词向量维度相同的向量 $z(X,q)\in R^d$。输出向量 z 可以被看做是输入序列 X 中被查询词 q 关注的上下文信息，即对于查询 q 特定的句子编码。

2.2　动态掩蔽注意力

由于标准注意力机制关注的是整个句子，难以处理多事件抽取的问题，标准注意力可能会保留对一个事件最重要的信息，而忽略了另一个事件的关键信息，这样会在多事件抽取中导致错误。在标准注意力机制中，查询词关注的是输入序列中的所有词，这种做法过于简化了。受 DMCNN 中动态多池化以及 DiSAN 中有向自注意力的启发，本文提出了一种能够关注到序列中特定部分的动态掩蔽注意力（DyMAN）机制，如图 2 所示。

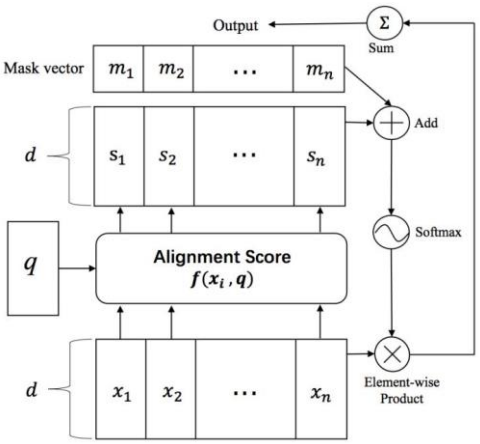

图 2　动态掩蔽注意力机制

Fig. 2　Dynamic masked attention mechanism

DyMAN 的关键机制就是掩蔽机制[19]。在 2.1 节阐述的标准注意力机制中，输入序列中的每个词都会被查询词所考虑。但在 DyMAN 中，使用了掩蔽向量来控制查询词会关注哪些词。

掩蔽向量 m 与输入序列的长度相同，它的元素属于集合 $\{0,1\}$，即 $m\in\{0,1\}^n$。掩蔽向量中值为 1 的位置会被查询词关注到，而值为 0 的位置不会被查询词关注，即这些位置被掩蔽了。掩蔽向量可以是预定义的，也可以是根据任务特定动态生成的。例如，掩蔽向量 $m=[1,1,1,0,...,0]$ 表示只在前三个词上有注意力。

一旦掩蔽向量被给定，就可以在输入序列 X 上计算掩蔽注意力。用一种与多维注意力相似的方式计算掩蔽注意力，为词向量每个特征得到一个注意力分数。也就是说，每个词向量的注意力分数是一个向量。

首先，用向量映射函数 $f: R^d\times R^d\rightarrow R^d$ 计算非归一化的注意力分数：

$$s_i=f(x_i,q) \tag{4}$$

然后，将掩蔽向量加到注意力分数向量上：

$$s_i^{mask}=s_i+\log m_i \tag{5}$$

其中：标量 $\log m_i$ 被加到注意力分数向量 s_i 的每个维度上，生成掩蔽注意力分数向量 s_i^{mask}。

接下来，对掩蔽注意力分数进行归一化：

$$a_i^{mask}=\frac{\exp(s_i^{mask})}{\sum_{i=1}^{n}\exp(s_i^{mask})} \tag{6}$$

通过与掩蔽向量相加以及归一化，使得掩蔽值为 1 的位置注意力分数不变，使掩蔽值为 0 的位置注意力分数变为 0。这样就实现了只关注输入序列中掩蔽值为 1 的位置，而掩蔽掉其他位置的注意力。

最后，通过加权求和得到注意力序列编码：

$$z(X,q,m)=\sum_{i=1}^{n}a_i^{mask}\otimes x_i \tag{7}$$

其中：⊗ 表示按位乘法。式（7）标明最终的序列编码 z 取决于输入序列 X、查询词 q 以及掩蔽向量 m。通过使用动态掩蔽注意力，可以很容易地对输入序列中的不同部分予以关注，从而产生更加灵活和更有表达能力的序列表示。

自注意力机制，从某个角度来说，也可以认为是动态掩蔽注意力的一种特例，其中的掩蔽向量除了在词本身的位置上为 0，在其他位置全为 1。

3　事件抽取模型

在事件抽取任务中，输入给定的句子 $W=[w_1,w_2,...,w_n]$，其中 n 是句子中词的数目，w_i 代表第 i 个词。假设句子中的实体提及 $E=[e_1,e_2,...,e_k]$ 被给定，其中 k 是实体的数目，e_i 是第 i 个实体。除此之外，假设实体头部的最后一个词的位置 $I=[i_1,i_2,...,i_k]$ 以及实体类别 $ET=[et_1,et_2,...,et_k]$ 也是已知的。

给定句子 W 和实体 E，事件抽取系统进行两个阶段的预测：a)给每个单词 w_i 预测触发词类别；b)对于每个其触发词类别不为 Negative 的单词 w_i，为每个实体提及 e_j 预测一个要素角色 a_{ij}，每个实体提及都可以看做是相对于当前触发词的候选要素。由于用于触发词预测与要素预测的模型比较相似，并且要素预测模型更加复杂，先介绍要素预测模型，然后简要描述触发词预测模型。

用于要素预测的动态掩蔽注意力网络（DyMAN）模型架构如图 3 所示。用于要素预测的 DyMAN 模型主要由三个模块构成：a）句子编码模块，包括词嵌入和双向递归神经网络；b）动态掩蔽注意力模块，用于生成要素候选词的注意力上下文表示；c）预测模块，用于输出预测的要素角色。

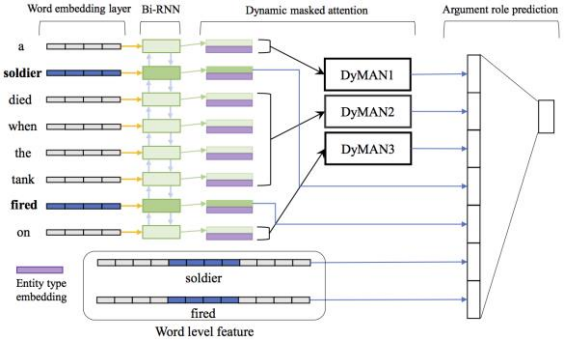

图 3　用于要素角色预测任务的模型架构

Fig. 3　Architecture of this proposed model for argument role prediction

3.1　句子编码

1)词嵌入　在句子编码阶段，每个离散的符号化单词 w_i 首先通过预训练的词向量转换为连续的实值向量 $x_i\in R^{d_e}$，其中 d_e 是词嵌入的维数。连续的词表示近年来已经被广泛地探究过，并且被证明能够捕捉单词的语义信息。转换之后，输入序列成为一个实值词向量序列 $X=[x_1,x_2,\cdots,x_n]^T$。

2)实体类别嵌入　给定每个实体提及的实体类别 $ET=[et_1,et_2,\ldots,et_k]$，采用文献[10]和文献[28]中提及的 BIO 标注体系给句子中的每个单词都分配一个实体类别标签 $B=[\beta_1,\beta_2,...,\beta_n]$，再将符号化的实体类别标签转换为实值的实体类别嵌入 $B=[\beta_1,\beta_2,...,\beta_n]$，其中 $\beta_i\in R^{d_{et}}$，d_{et} 是实体类别嵌入的维度。

3)双向递归神经网络　双向递归神经网络（bidirectional recurrent neural network, Bi-RNN）是有两个方向（前向和后向）的递归神经网络（RNN），能够融合两个方向的上下文信息。其中，使用递归神经网络的一种变体——门控递归单元（gated recurrent units, GRU）[21]作为基本的 RNN 单元来捕捉句子编码中的长期依赖。

将词向量 $X=[x_1,x_2,\ldots,x_n]^T$ 送入 Bi-RNN 中，生成感知了上下文信息的句子编码 $H=[h_1,h_2,\ldots,h_n]^T$：

$$[h_1,h_2,\ldots,h_n]^T=\text{Bi-RNN}([x_1,x_2,\ldots,x_n]^T) \tag{8}$$

感知了上下文序列编码的维度是 $h_i\in R^{2d_c}$，其中 d_c 是 *Bi-RNN* 的记忆单元的维度。

将实体类别嵌入 B 和 *Bi-RNN* 句子编码 H 拼接起来，得到最终的句子编码 $H'=[h'_1,h'_2,\ldots,h'_n]^T$，其中 $h'_i=h_i\oplus b_i$，$h'_i\in R^{2d_c+d_{et}}$。

3.2　要素角色预测（Argument Role Prediction）

现在已经得到句子编码 H' 以及所有的实体提及 E，假设给定第 i 个词为事件触发词，应该为每个候选要素都预测一个在当前事件中的要素角色，设预测出的要素角色为 $A_i=[a_{i1},a_{i2},...,a_{ik}]$。在给定触发词 w_i 情况下，只考虑候选要素 e_j（位置为 i_j）的要素角色预测任务，其他候选要素的预测过程是类似的。

1)动态掩蔽注意力网络　正如 2.2 节提到的那样，动态掩蔽注意力网络（DyMAN）将序列、查询词和掩蔽向量作为输入，为给定查询词输出注意力序列编码。将句子编码 $H'=[h'_1,h'_2,\cdots,h'_n]^T$ 作为输入序列，将候选要素 e_j 的编码向量 h'_{i_j} 作为查询。同时，根据触发词 w_i 的位置 i 和候选要素 e_j 的位置 i_j，将句子划分为三段。然后，动态地为这三段分别产生三个掩蔽向量 m_l,m_m,m_r（图 4）。

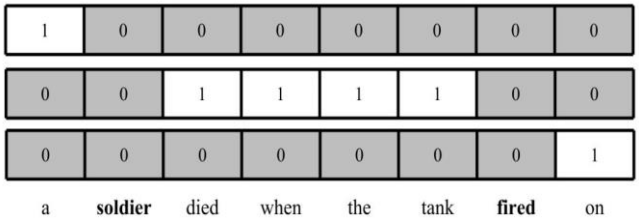

图 4　要素角色预测中的 DyMAN 掩蔽向量(从上到下分别对应左、中、右三段的掩蔽向量)

Fig. 4　Mask vectors for dyman in argument role prediction(From top to bottom are mask vector for left, middle, right segment respectively)

使用 2.2 节定义的 DyMAN 作为一个抽象模块，候选要素 e_j 在这三段上的注意力句子编码为

$$\begin{cases} z_l=DyMAN(H',h'_{i_j},m_l) \\ z_m=DyMAN(H',h'_{i_j},m_m) \\ z_r=DyMAN(H',h'_{i_j},m_r) \end{cases} \tag{9}$$

其中：$z_l,z_m,z_r\in R^{2d_c+d_{et}}$。

将三个注意力编码拼接起来得到候选要素 e_j 最终的注意力上下文编码 $z_j=z_l\oplus z_m\oplus z_r$，$z_j\in R^{6d_c+3d_{et}}$。

2)预测　将候选要素的注意力编码 z_j、触发词的句子编码 h'_i、候选要素的句子编码 h'_{i_j} 拼接起来，得到候选要素 e_j 的句级别特征：$u_{ij}=z_j\oplus h'_i\oplus h'_{i_j}$，$u_{ij}\in R^{10d_c+5d_{et}}$。

另外，也使用触发词和候选要素的词嵌入窗口。触发词

w_i 的词嵌入窗口定义为

$$L_i=[x_{i-l},\dots,x_{i-1},x_i,x_{i+1},\dots,x_{i+l}] \tag{10}$$

候选要素 e_j 的词嵌入窗口为

$$L_i=[x_{i_j-l},\dots,x_{i_j-1},x_{i_j},x_{i_j+1},\dots,x_{i_j+l}] \tag{11}$$

窗口大小均为$2\lambda+1$。将两个窗口拼接起来得到候选要素的词级别特征：

$$v_{ij}=L_i\oplus L_{i_j},\quad v_{ij}\in R^{(4\lambda+2)d_e} \tag{12}$$

然后，将句级别特征和词级别特征拼接起来得到候选要素 e_j 的最终特征：

$$\eta_{ij}=u_{ij}\oplus v_{ij},\quad \eta_{ij}\in R^{10d_c+5d_{et}+(4\lambda+2)d_e} \tag{13}$$

最终特征 η_{ij} 被送入顶端带有 softmax 层的前馈神经网络 F^{arg} 中，计算得到在所有可能要素角色上的概率：

$$p_{ij}^{arg}=F^{arg}(\eta_{ij}),\quad p_{ij}^{arg}\in R^{n_a} \tag{14}$$

其中:n_a 是所有可能要素角色（包括 Negative）的数目; p_{ij}^{arg} 本质上是一个概率向量，它的第 k 个元素表示候选要素 e_j 对于触发词 w_i 充当角色 k 的概率，即$[p_{ij}^{arg}]_k=p(a_{ij}=k\mid W,E)$。

使用贪心策略，最后预测得到候选要素 e_j 对于触发词 w_i 充当的要素角色 a_{ij}：$a_{ij}=\underset{k}{argmax}([p_{ij}^{arg}]_k)$。

3.3 触发词预测（trigger prediction）

在触发词预测阶段，应当为句子中的每一个单词 w_i 都预测一个触发词类别 t_i。与要素角色预测中类似，为简单起见，只考虑某一个单词 w_i 的触发词预测。

3.1 节中描述的句子编码模块同样也被用于触发词预测。随后将类似的动态掩蔽注意力模块作用于句子编码上（见 3.2 节），不同之处在于，根据候选触发词 w_i 把输入句子划分成两段。然后在两段上分别执行动态掩蔽注意力模块，生成候选触发词 w_i 的注意力上下文编码 z_i，将其于候选触发词的句子编码向量 h'_i 拼接得到候选词的句级别特征 u_i。接着将句级别特征与词级别特征 v_i（词嵌入窗口）拼接在一起，得到最终特征 η_i。最终特征以同样的方式被送入前馈神经网络 F^{trig} 中，得到所有可能触发词类别上的概率分布 $p_i^{trig}\in R^{n_t}$，其中 n_t 是所有可能触发词类别的数目，包括 Negative 类别。候选触发词 w_i 的预测类别为

$$t_i=\underset{k}{argmax}\left([p_i^{trig}]_k\right) \tag{15}$$

3.4 训练

给定句子 W 和实体提及 E，将标注的触发词类别记为 $T=[t_1^*,t_2^*,\cdots,t_n^*]$，将标注的要素角色记为 $A=[a_{ij}^*]_{i=1,2,\dots,n}^{j=1,2,\dots,k}$，其中 n 是句中单词的数目，k 是句中实体提及的数目。

假定触发词预测和要素角色预测的模型参数分别为 θ 和 ϕ，将触发词预测和要素角色预测的损失函数分别定义为负对数似然函数 $L^{trig}(\theta)$ 和 $L^{arg}(\varphi)$：

$$L^{trig}(\theta)=-\sum_{i=1}^{n}log\ p(t_i=t_i^*\mid W,E;\theta)=-\sum_{i=1}^{n}log\left([\mathbf{p}_i^{trig}(\theta)]_{t_i^*}\right) \tag{16}$$

$$\begin{aligned}L^{arg}(\varphi)&=-\sum_{i=1}^{n}I(t_i^*\neq Negative)\sum_{j=1}^{k}log\ p(a_{ij}=a_{ij}^*\mid W,E;\varphi)\\&=-\sum_{i=1}^{n}I(t_i^*\neq Negative)\sum_{j=1}^{k}log\left([p_{ij}^{arg}(\varphi)]_{a_{ij}^*}\right)\end{aligned} \tag{17}$$

其中：$I(\cdot)$是指示函数。

使用带乱序 mini-batch 的随机梯度下降算法来最小化损失函数，并且使用 AdaDelta 更新规则[22]。在训练过程中，对词向量表进行精调，并且对全局范数超过给定值的梯度进行裁减。

4　实验

4.1　参数、数据集和评价指标

对所有实验，使用 300 维的 GloVe 840B 预训练词向量[23]对词嵌入进行初始化(d_e=300)，同时将实体类别嵌入的维度设为 d_{et}=15。对于词级别特征，使用的窗口大小为 λ=1；对于 Bi-RNN，使用 200 维的 GRU 记忆单元（d_c=300）。对于动态掩蔽注意力模块，激活函数设为 ReLU。对于前馈神经网络 F^{trig} 和 F^{arg}，在隐藏层中分别使用 400 和 800 个神经元。此外，使用 dropout 作为正则化手段，设为 0.5， mini-batch 大小为 50。

本文使用 ACE 2005 作为数据集。为了便于比较，使用了与前人工作[1,4,6,8~10]一样的数据划分：使用 40 篇 newswire 类别的文章作为测试集，30 篇其他文章作为验证集，剩下的 529 篇作为训练集。也使用与前人工作一样的评价标准来判断预测的事件提及的准确性。

4.2　总体效果

使用以下几个在 ACE 2005 上效果领先的事件抽取模型作为对比模型：

a)Li's baseline。文献[8]中提出的只基于传统特征的流水线模型。

b)Li's structure。文献[8]提出的使用了结构化预测、传统局部特征和全局特征的联合模型。

c)DMCNN。文献[9]中提出的基于动态多池化神经网络的流水线模型。

d)JRNN。文献[10]中提出的基于递归神经网络的联合抽取模型，也是在不引入外部资源情况下效果最好的模型。

表 1 显示了以上几个模型以及 DyMAN 模型在盲测数据集上的总体效果。从表中可以看出，本文提出的 DyMAN 模型在要素角色分类任务上取得了最高的 F1 值，在触发词分类任务上也取得了可观的效果。在要素角色分类上，DyMAN 模型比之前最好的模型（JRNN）高出 3.2%，实现了很大的效果提升，并证明了 DyMAN 模型的有效性。

表 1　盲测数据上的总体效果

Table 1　Overall performance one blind test data

模型	触发词识别/%			触发词识别+分类/%			要素识别/%			要素角色分类/%		
	P	R	F	P	R	F	P	R	F	P	R	F
Lis basline	76.2	60.5	67.4	74.5	59.1	65.9	74.1	37.4	49.7	65.4	33.1	43.9
Lis structure	76.9	65.0	70.4	73.7	62.3	67.5	69.8	47.9	56.8	64.7	44.4	52.7
DMCNN	80.4	67.7	**73.5**	75.6	63.6	69.1	68.8	51.9	59.1	62.2	46.9	53.5
JRNN	68.5	75.7	71.9	66.0	73.0	**69.3**	61.4	64.2	**62.8**	54.2	56.7	**55.4**
Bi-RNN	73.0	68.3	70.6	70.4	65.9	68.1	68.3	53.5	60.0	60.3	47.2	53.0
Bi-RNN+SA	70.0	69.5	69.7	66.8	66.3	66.6	70.7	51.2	59.4	64.1	46.5	53.9
Bi-RNN+DyMAN	72.4	69.2	**70.8**	70.1	67.1	**68.6**	69.6	61.4	**65.3**	62.5	55.1	**58.6**

4.3 动态掩蔽注意力的有效性

为了表明动态掩蔽注意力机制（DyMAN）的有效性，将它与两个基线模型 Bi-RNN 和 Bi-RNN+SA 相比较。这两个基线模型都使用了词级别特征和句级别特征。其中，Bi-RNN 模型使用词嵌入和双向 RNN 得到句级别特征。Bi-RNN+SA 引入了一个额外的标准注意力模块（standard attention, SA），同时将触发词和候选要素作为查询，并在 Bi-RNN 基础上得到句级别特征。本文的模型表示为 Bi-RNN+DyMAN。

从表 1 中观察到标准的注意力机制（Bi-RNN+SA）效果甚至不如单纯的 Bi-RNN，但是本文的 DyMAN 模型效果很大程度上超过了两个模型，并在 Bi-RNN 的基础上取得了触发词分类任务 0.5%的效果提升，以及要素角色分类任务 5.6%的效果提升。以上的经验分析表明，本文提出的动态掩蔽注意力网络具有令人信服的有效性。

4.4 多事件语句

正如第 1 章所描述的那样，动态掩蔽注意力网络用来解决多事件抽取问题。遵循前人的工作[9,10]，根据一个句子中事件的数目将测试集中的句子分为单事件语句和多事件语句两部分，并分别在两部分上评测模型效果。下面比较了 DMCNN、JRNN 和 DyMAN 三个模型的多事件抽取效果。

三个模型在触发词分类任务和要素角色分类任务上的 F1 值如表 2 所示。可以发现，对于多事件抽取（即 1/N 这一列），DyMAN 模型在两个任务上的效果都明显优于其他两个模型。与之前最好的模型 JRNN 相比，DyMAN 在触发词分类任务上取得了 9.8%的提升，在要素分类任务上取得了 4.5%的提升，这说明了 DyMAN 在多事件抽取中有效性。对于单事件语句（1/1 这一列），DyMAN 模型也在要素角色分类任务上比之前最优的 DMCNN 模型提升了 1.9%。

表 2 单事件语句（1/1）和多事件语句（1/N）上的 F1 值

Table 2 F1 score on single-event sentences (**1/1**) and multiple-event sentences (**1/N**)

阶段	模型	1/1	1/N	all
触发词分类	DMCNN	74.3	50.9	69.1
	JRNN	**75.6**	64.8	**69.3**
	DyMAN	72.6	**74.6**	68.6
要素角色分类	DMCNN	54.6	48.7	53.5
	JRNN	50.0	55.2	55.4
	DyMAN	**56.5**	**59.7**	**58.6**

5 结束语

本文提出了一种基于动态掩蔽注意力机制的事件抽取模型 DyMAN，该模型能够关注到序列中的特定部分，得到更加丰富和更有表达能力的语义表示，分别在事件要素角色预测和触发词预测中进行使用。在 ACE 2005 的数据集上的测试和分析表明，基于 DyMAN 的事件抽取模型在多事件抽取上十分有效，并能取得领先的实验效果。

参考文献：

[1] Ji Heng, Grishman R. Refining event extraction through cross-document inference [C]// Proc of ACL-08: HLT. 2008: 254-262.

[2] 徐霞, 李培峰, 朱巧明. 一个半监督的中文事件抽取方法 [J]. 中文信息学报, 2016. 3, 30 (2): 168-174. (Xu Xia, Li Peifeng, Zhu Qiaoming, A semi-supervised Chinese event extraction method [J]. Journal of Chinese information processing. 2016. 3, 30 (2): 168-174.)

[3] Gupta P , Ji Heng. Predictingunknown time arguments based on cross-event propagation [C]// Proc of ACL-IJCNLP 2009 Conference Short Papers. Stroudsburg, PA: Association for Computational Linguistics, 2009: 369-372.

[4] Liao Shasha , Grishman R. Using document level cross-event inference to improve event extraction [C]// Proc of the 48th Annual Meeting of the Association for Computational Linguistics. [S. l.] : ACL Press, 2010: 789-797.

[5] Liao Shasha , Grishman R. Acquiring topic features to improve event extraction: in preselected and balanced collections [C]// Proc of International Conference Recent Advances in Natural Language Processing. [S. l.] : ACL Press, 2011: 9-16.

[6] Hong Yu, Zhang Jianfeng, Ma Bin, et al. Using cross-entity inference to improve event extraction [C]// Proc of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. [S. l.] : ACL Press, 2011: 1127-1136.

[7] McClosky D, Surdeanu M, Manning C. Event extraction as dependency parsing [C]// Proc of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. [S. l.] : ACL Press, 2011: 1626-1635.

[8] Li Qi, Ji Heng, Huang Liang. Joint event extraction via structured prediction with global features [C]// Proc of the 51st Annual Meeting of the Association for Computational Linguistics . [S. l.] : ACL Press, 2013: 73-82.

[9] Chen Yubo, Xu Liheng, Liu Kang, et al. Event extraction via dynamic multi-pooling convolutional neural networks [C]// Proc of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing . [S. l.] : ACL Press, 2015: 167-176.

[10] Nguyen H T, Cho K, Grishman R. Joint event extraction via recurrent neural networks [C]// Proc of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. [S. l.] : ACL Press, 2016: 300-309.

[11] Grishman R, Westbrook D, Meyers A. NYU's English ACE 2005 system description. [R]. 2005:51.

[12] Ahn D. The stages of event extraction [C] //Proc of Workshop on Annotating and Reasoning About Time and Events. 2006: 1-8,.

[13] 张贺, 刘茂福, 胡慧君, 等. 基于信息单元融合的新闻原子事件抽取 [J]. 武汉大学学报 : 理学版 , 2015. 4, 61 (2): 139-144. (Zhang He, Liu Maofu, Hu Huijun, et al. Atomic event extraction based on information unit fusion [J]. Journal of Wuhan University: Natural Science Edition , 2015. 4, 61 (2): 139-144.)

[14] Li Qi, Ji Heng, Hong Yu, et al. Constructing information networks using one single model [C]// Proc of Conference on Empirical Methods in Natural Language Processing. [S. l.] : ACL Press, 2014: 1846-1851.

[15] Chen Yubo, Liu Shulin, Zhang Xiang, et al. Automatically labeled data generation for large scale event extraction [C]// Proc of the 55th Annual Meeting of the Association for Computational Linguistics . [S. l.] : ACL Press, 2017: 409-419.

[16] Luong T, Pham H, Manning C D. Effective approaches to attention-based neural machine translation [C]// Proc of Conference on Empirical Methods in Natural Language Processing. [S. l.] : ACL Press, 2015: 1412-1421.

[17] Seo M, Kembhavi A, Farhadi A, et al. Bidirectional attention flow for machine comprehension [C]// Proc of International Conference on Learning Representations.2017.

[18] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [C]// Advances in Neural Information Processing Systems. 2017: 5998-6008.

[19] Shen Tao, Zhou Tianyi, Long Guodong, et al. DiSAN: directional self-attention network for RNN/CNN-free language understanding [C]// Proc of AAAI Conference on Artificial Intelligence. 2018.

[20] Graves A, Jaitly N, Mohamed A R. Hybrid speech recognition with deep bidirectional LSTM [C]// Proc of IEEE Workshop on Automatic Speech Recognition and Understanding. 2013: 273-278.

[21] Cho K, Van Merrienboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation [C]// Proc of Conference on Empirical Methods in Natural Language Processing . 2014: 1724-1734.

[22] Zeiler M D. ADADELTA: an adaptive learning rate method [R/OL]. CoRR, 2012: abs//1212. 5701.

[23] Pennington J, Socher Rd, Manning C D. GloVe: Global vectors for word representation [M]// Empirical Methods in Natural Language Processing. 2014: 1532-1543.

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

[25] Liao Shasha , Grishman R. Using document level cross-event inference to improve event extraction [C]// Proc of the 48th Annual Meeting of the Association for Computational Linguistics. [S. l.] : ACL Press, 2010: 789-797.

[26] Liao Shasha , Grishman R. Acquiring topic features to improve event extraction: in preselected and balanced collections [C]// Proc of International Conference Recent Advances in Natural Language Processing. [S. l.] : ACL Press, 2011: 9-16.

[27] Liu Shulin, Chen Yubo, Liu Kang, et al. Exploiting argument information to improve event detection via supervised attention mechanisms [C]// Proc of the 55th Annual Meeting of the Association for Computational Linguistics. [S. l.] : ACL Press, 2017: 1789-1798.

[28] Nguyen T H , Grishman R. Event detection and domain adaptation with convolutional neural networks [C]// Proc of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing . [S. l.] : ACL Press, 2015: 365-371.