第 58 卷 第 3 期 厦门大学学报(自然科学版) Vol. 58 No. 3
2019 年 5 月 Journal of Xiamen University (Natural Science) May 2019
doi:10.6043/j.issn.0438-0479.201901008

基于语言模型及循环卷积神经网络的事件检测

施喆尔,陈锦秀*
(厦门大学信息科学与技术学院,福建 厦门 361005)

摘要:目前,事件检测的难点在于一词多义和多事件句的检测.为了解决这些问题,提出了一个新的基于语言模型的带注意力机制的循环卷积神经网络模型(recurrent and convolutional neural network with attention based on language models,LM-ARCNN).该模型利用语言模型计算输入句子的词向量,将句子的词向量输入长短期记忆网络获取句子级别的特征,并使用注意力机制捕获句子级别特征中与触发词相关性高的特征,最后将这两部分的特征输入到包含多个最大值池化层的卷积神经网络,提取更多上下文有效组块.在 ACE2005 英文语料库上进行实验,结果表明,该模型的 F_1 值为 74.4%,比现有最优的文本嵌入增强模型(DEEB)高 0.4%.

关键词:事件检测;语言模型词嵌入;长短期记忆网络;动态多池化卷积神经网络;注意力机制
中图分类号:TP 391.1 **文献标志码**:A **文章编号**:0438-0479(2019)03-0442-07

事件检测是事件抽取任务的第一步,通过事件检测,可以确定一个句子中是否存在可以表征事件发生的触发词,并且判断这个触发词所触发的事件属于哪种类别.这对于后续事件元素的判断和分类起着至关重要的作用.早期的事件检测大多采用基于特征的方法:如 Grishman 等[1]和 Ahn[2]采用的传统词法和句法特征(如词性、依存关系等),并使用最大熵模型进行分类;Ji 等[3]和 Liao 等[4]则额外考虑了跨句子和跨文档的信息作为辅助特征;Hong 等[5]引入了跨实体推理,以此获得更多分类辅助特征;Li 等[6]在基于特征的基础上,采用了联合结构,建立了一个包含触发词、触发词类型以及事件元素词和元素词类型的结构体,同时预测事件类别和事件元素类型.

上述基于特征的方法存在 3 个问题:首先基于特征的方法其特征工程都较为复杂,不利于扩展应用到其他语言任务中;其次基于特征的方法都需要依赖于专家信息和其他自然语言处理工具,这一部分的误差将会被累积到最终的分类误差中;最后基于特征的方法所学习到的特征都属于浅层特征,不能学习到深层语义特征.为了解决以上问题,随着深度学习的发展,深度神经网络的方法在事件检测领域受到越来越多的关注.Chen 等[7]首先提出了动态多池化卷积神经网络(dynamic multi-pooling convolutional neural networks,DMCNN)模型,主要解决了多事件句子的触发词抽取问题.自此,事件检测模型的输入特征基本简化为句子的词向量、位置向量及实体类型向量.Nguyen 等[8]随后提出了双向循环神经网络模型,并借鉴了联合结构,同时识别触发词和事件元素.Feng 等[9]提出了双向长短期记忆网络(bidirectional long short-term memory neural network,Bi-LSTM)和卷积神经网络(convolutional neural network,CNN)的混合模型,同时获取序列信息和短语块信息.Liu 等[10]首次在事件抽取任务中采用了注意力机制,将事件元素信息用于辅助事件检测任务中.Duan 等[11]和 Zhao 等[12]将句子所在的全篇文档作为特征,其中 Zhao 等[12]采用注意力机制获取文档特征,将文档特征用于辅助事件检测任务,提出了文档嵌入增强模型(document embedding enhanced based model,DEEB),获得很好的效果.此外,Nguyen 等[13]提出了用基于依存树的图卷积模型进行事件检测,利用语法结构信息,捕获句子中距离候选触发词位置较远的词语信息.Hong 等[14]将生成式对抗网络(generative

收稿日期:2019-01-08 录用日期:2019-04-26
基金项目:国家自然科学基金(60803078);福建省自然科学基金(2010J01351);教育部海外留学回国人员科研启动基金
*通信作者:cjx@xmu.edu.cn
引文格式:施喆尔,陈锦秀.基于语言模型及循环卷积神经网络的事件检测[J].厦门大学学报(自然科学版),2019,58(3):442-448.
Citation:SHI Z E,CHEN J X. Event detection via recurrent and convolutional networks based on language model[J]. J Xiamen Univ Nat Sci,2019,58(3):442-448.(in Chinese)

adversarial networks,GAN)应用到事件检测任务中,以降低神经网络所抽取出的特征中可能包含的一些看似与事件相关,实则不存在关系的虚假隐含信息对事件检测任务性能的影响.

在语言学中,无论是事件句子本身的语言信息还是上下文信息都对事件检测识别任务有着重要的影响,然而现有方法通常只从其中一个方面入手,本研究为了同时捕捉句子级别特征和上下文组块两部分信息,结合了 Bi-LSTM 和 DMCNN 深度神经网络结构,引入语言模型嵌入(embeddings from language models,ELMo)和注意力机制,提出构建新的基于语言模型的带注意力机制的循环卷积神经网络模型(recurrent and convolutional neural network with attention based on language models,LM-ARCNN)进行事件检测. 通过在 ACE2005 英文语料库上进行的实验表明,本模型可以达到目前事件检测任务中最佳的结果,并且在多事件抽取中也有很好的表现.

本研究具有以下几个创新点:1) 将 ELMo 算法引入事件检测任务,获取本身具有上下文信息的词向量,研究发现,这样的词向量对事件检测任务具有帮助,且优于传统的 word2vec 词向量. 2) 结合 Bi-LSTM 和 DMCNN 搭建多层神经网络,模型可以学习更为抽象的语义信息. 3) 模型采用注意力机制更加关注深度语义信息中与触发词相关性高的特征,并且采用了动态多池化避免信息的遗漏,在处理多事件的事件检测中也具有很好的表现.

1　LM-ARCNN 模型

事件检测任务可视为多分类问题,给定一个句子,将其中的每个词都作为触发词候选词,进行多分类判断事件类别. 根据自动内容抽取测评会议(automatic content extraction,ACE)任务中的定义,一共有 33 个子类别,另外还需定义一个非事件类,记为(NA),所以这是一个具有 34 个类别的多分类问题.

事件检测任务的困难在于两方面:1) 同一个词在不同上下文中可能表征了不同类别的事件,即词语的二义性;2) 在同一句话中,可能发生多个不同类别的事件. 例如:"The police officer who **fired** into a car full of teenagers was **fired** Tuesday."这个句子中存在两个"fired",第一个"fired"是一个 Attack 事件的触发词,但第二个"fired"却是 End-Position 事件的触发词.

为了解决上述问题,本文中提出了基于语言模型的深层网络模型 LM-ARCNN,模型共 5 层,分别为编码层、Bi-LSTM 层、注意力机制层、DMCNN 层、全连接输出层. 第 1 层编码层使用 ELMo 算法学习词向量,将获取的词向量与位置向量和实体类型向量连接得到向量化表示. 第 2 层用一个 Bi-LSTM 层对向量化表示编码,得到带有全句话隐含语义信息的句子级别特征,这有助于在出现一词多义时,有效判断候选触发词的正确含义. 第 3 层对 Bi-LSTM 编码后的句子向量进行注意力机制计算,获取带有权重的特征向量. 其中,与候选触发词相关的词语将被赋予较大权重,而与候选触发词无关的词语将赋予较小权重,以弥补与候选触发词距离较远的重要词语的信息衰减,使得在同一句子中,模型所学习的深层语义信息是具有候选词针对性的,不再受上下文与候选词的距离限制. 第 4 层将第 2 层及第 3 层的两个向量分别输入 DMCNN 层的两个通道,DMCNN 层是改良后的 CNN 模型,卷积部分采用多个不同窗口大小的滤波器,池化部分对两个通道的所有滤波器结果进行动态的最大池化,池化过程中根据候选触发词位置划分,动态输出两个最大值,从而可以获取更多精细的组块信息,避免多事件句中信息的遗漏. 第 5 层将 DMCNN 层处理后的组块信息输入一个全连接层,进行归一化输出. 通过对句子进行建模,对候选触发词进行分类,判断候选触发词所触发的事件类型. 本模型的结构图如图 1 所示. (以"A car bomb **exploded** in central Baghdad."为输入句子,"exploded"是当前的候选触发词)

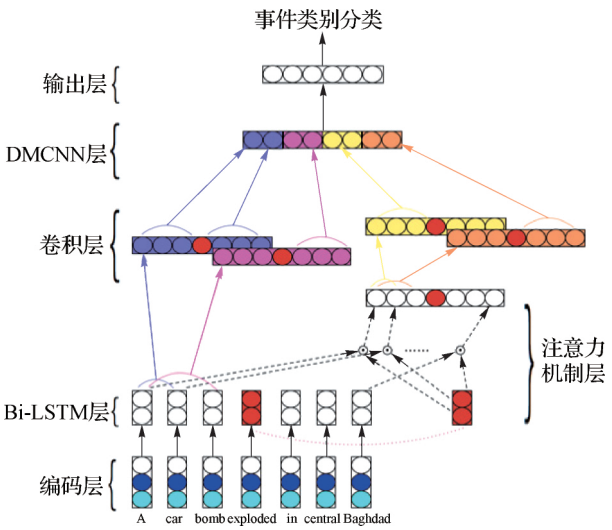

图 1　LM-ARCNN 模型

Fig. 1　The LM-ARCNN model

1.1 编码层

对于一个输入的句子,编码层将其中的每个词 w_t 转换为实值向量 $\boldsymbol{x}_t$,$\boldsymbol{x}_t$ 由以下特征组成:

1) ω_t 的词向量 $\boldsymbol{w}(w_t)$. 传统的词向量如 word2vec[15] 对于每个词都具有唯一的嵌入(embedding)表示,难以处理一词多义的问题. 而本模型采用的 ELMo[16] 通过单词所在的句子获得词向量,故其词向量具有上下文信息,同一个词在不同语境中的词向量不同,是深度语境化的词表征,有利于缓解一词多义的问题.

ELMo 由 1 个基于字母的卷积神经网络层和 L 个双向 LSTM 层组成. 对于句子中每一个词 w_t,ELMo 都会计算出一个 $2L+1$ 个向量表示的集合:

$$R_t = \{\boldsymbol{c}_t, \overrightarrow{\boldsymbol{h}_{t,j}}, \overleftarrow{\boldsymbol{h}_{t,j}} \mid j = 1,2,\cdots,L\} = \{\boldsymbol{h}_{t,j} \mid j = 0,1,\cdots,L\}, \tag{1}$$

其中:当 $j=0$ 时,$\boldsymbol{h}_{t,0} = [\boldsymbol{c}_t;\boldsymbol{c}_t]$,$\boldsymbol{c}_t$ 是对单词 w_t 中的字母进行 CNN 编码的结果;当 $j>0$ 时,$\boldsymbol{h}_{t,j} = [\overrightarrow{\boldsymbol{h}_{t,j}}; \overleftarrow{\boldsymbol{h}_{t,j}}]$ 是每一层的输出结果. 取各向量的平均值作为最终词向量,维度为 $d^w = 1\ 024$,

$$\boldsymbol{w}(w_t) = L^{-1}\sum_{j=1}^{L}\boldsymbol{h}_{t,L}, \tag{2}$$

2) ω_t 的位置向量 $\boldsymbol{p}(w_t)$. 为了表征当前词与候选词的位置关系,定义当前词 w_t 到候选词 w_a 的位置关系为 $(t-a)$. 若当前词在候选词之前,则位置定义为负距离;反之为正距离. 通过位置向量表将位置关系实值化,位置向量表是随机初始化的,并在反向传播过程中进行优化,位置向量的维度为 d^p.

3) ω_t 的实体类型向量 $\boldsymbol{e}(w_t)$. 句子中的部分词可能是一个实体,词语的实体类型信息采用了 ACE 语料中的标注,类似于位置向量,同样随机初始化一个实体向量表,将实体类型信息映射到实体向量表中,用实值向量 $\boldsymbol{e}(w_t)$ 表示实体信息,$\boldsymbol{e}(w_t)$ 的维度为 d^e.

因此,对于词 w_t,经编码层编码,将表示为:

$$\boldsymbol{x}_t = \boldsymbol{w}(w_t) \oplus \boldsymbol{p}(w_t) \oplus \boldsymbol{e}(w_t).$$

其中:$\oplus$ 表示向量拼接;$\boldsymbol{x}_t \in \mathbf{R}^d$,$d = d^w + d^p + d^e$. 故句子可用 $\boldsymbol{X} = [\boldsymbol{x}_1,\cdots,\boldsymbol{x}_t,\cdots,\boldsymbol{x}_n]$ 表示,n 为句子长度. $\boldsymbol{X}$ 是 Bi-LSTM 层的输入,$\boldsymbol{X} \in \mathbf{R}^{n\times d}$.

1.2 Bi-LSTM 层

Bi-LSTM 属于双向循环神经网络(RNN),相较于 RNN,LSTM 更适合处理长序列,能避免长距离依赖问题;相较于单向 LSTM,双向 LSTM 可以同时对词语的前后文语义进行建模,提取句子级别的特征.

给定一个 $\boldsymbol{X}$,用前向的 $\mathrm{LSTM_f}$ 获得隐藏状态 $\{\boldsymbol{h}_{\mathrm{f}_1}, \boldsymbol{h}_{\mathrm{f}_2},\cdots,\boldsymbol{h}_{\mathrm{f}_n}\}$,同时用反向的 $\mathrm{LSTM_b}$ 获得隐藏状态 $\{\boldsymbol{h}_{\mathrm{b}_1}, \boldsymbol{h}_{\mathrm{b}_2},\cdots,\boldsymbol{h}_{\mathrm{b}_n}\}$,每个 $\boldsymbol{h}_{\mathrm{f}_t}$ 和 $\boldsymbol{h}_{\mathrm{b}_t}$ 的计算公式如下:

$$\boldsymbol{h}_{\mathrm{f}_t} = \overrightarrow{\mathrm{LSTM_f}}(\boldsymbol{x}_t, \boldsymbol{h}_{\mathrm{f}_{t-1}}), \tag{3}$$

$$\boldsymbol{h}_{\mathrm{b}_t} = \overleftarrow{\mathrm{LSTM_b}}(\boldsymbol{x}_t, \boldsymbol{h}_{\mathrm{b}_{t+1}}), \tag{4}$$

其中,$\boldsymbol{h}_{\mathrm{f}_{t-1}}$ 表示 $\boldsymbol{x}_t$ 之前的语义信息,$\boldsymbol{h}_{\mathrm{b}_{t+1}}$ 表示 $\boldsymbol{x}_t$ 之后的语义信息. 因此,采用 Bi-LSTM 可同时获得候选词前后的上下文语义信息. 故句子在 Bi-LSTM 层的输出为 $\boldsymbol{H}^{\mathrm{Bi}} = [\boldsymbol{h}_1^{\mathrm{Bi}},\cdots,\boldsymbol{h}_t^{\mathrm{Bi}},\cdots,\boldsymbol{h}_n^{\mathrm{Bi}}]$,其中 $\boldsymbol{h}_t^{\mathrm{Bi}} = [\boldsymbol{h}_{\mathrm{f}_t}, \boldsymbol{h}_{\mathrm{b}_t}]$.

为了避免随着模型层数叠加带来的内部变量偏移(internal covariate shift,ICS),本模型在 Bi-LSTM 层输出后采用横向规范化[17]计算整个 Bi-LSTM 层输出的均值 $\boldsymbol{u}$ 和方差 $\boldsymbol{\sigma}$,然后对其进行归一化:

$$\boldsymbol{u} = n^{-1}\sum_{t=1}^{n}\boldsymbol{h}_t^{\mathrm{Bi}}, \tag{5}$$

$$\boldsymbol{\sigma} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\boldsymbol{h}_t^{\mathrm{Bi}} - \boldsymbol{u})^2}, \tag{6}$$

$$\boldsymbol{h}_t = g\cdot(\boldsymbol{h}_t^{\mathrm{Bi}} - \boldsymbol{u})/\boldsymbol{\sigma} + \boldsymbol{b}, \tag{7}$$

其中,g 和 $\boldsymbol{b}$ 是收益参数和偏差参数,公式(7)的除法表示矩阵对应元素相除,模型之后部分所用的都是归一化后的 Bi-LSTM,简称为 Bi-LSTM 层输出 $\boldsymbol{H} = [\boldsymbol{h}_1,\cdots,\boldsymbol{h}_t,\cdots,\boldsymbol{h}_n]$,设置 Bi-LSTM 的隐藏层数量为 m_1,则 $\boldsymbol{H} \in \mathbf{R}^{m_1\times n}$.

1.3 注意力机制层

对 Bi-LSTM 输出层使用注意力机制可对句子中每一个词计算其与候选触发词之间的相似度. 若以相似度作为权重对句子进行加权,则相似度高的词语将有助于候选触发词的分类,可修正一些多义词产生的歧义,并为多事件句子中的触发词提供信息.

对于 $\boldsymbol{H}$,计算每个当前词的权重,这里注意力机制的相似度计算有几种常用方法[18]:

$$\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}) = \begin{cases} \boldsymbol{h}_t^{\mathrm{T}}\cdot\bar{\boldsymbol{h}}, & \text{点乘}, \\ \boldsymbol{h}_t^{\mathrm{T}}\boldsymbol{W}\bar{\boldsymbol{h}}, & \text{矩阵乘}, \\ \tanh(\boldsymbol{W}[\boldsymbol{h}_t;\bar{\boldsymbol{h}}]), & \text{连接}, \end{cases} \tag{8}$$

$$\alpha_t = \frac{\exp(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}))}{\sum_{t=1}^{n}\exp(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}))}, \tag{9}$$

$$\boldsymbol{s}_t = \alpha_t\boldsymbol{h}_t. \tag{10}$$

其中:$\bar{\boldsymbol{h}}$ 是当前的候选触发词的 Bi-LSTM 层输出向量,$\boldsymbol{h}_t$ 是句子中第 t 个词的 Bi-LSTM 层输出向量,$\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}})$ 为两者的相似度,对 $\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}})$ 进行 Softmax 归一化,可得到第 t 个词的权重 α_t;注意力机制层的输出表示为 $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2,\cdots,\boldsymbol{s}_n]$,$\boldsymbol{S} \in \mathbf{R}^{m_1\times n}$.

1.4 DMCNN 层

第 2 层 Bi-LSTM 层可以获得整个句子上下文的语义信息,而第 3 层注意力机制层可以着重关注与候

选触发词相似度高的词语的语义信息,根据这些信息,进一步学习其组块特征,采用 Chen 等[7] 提出的 DMCNN 模型检测多事件句子的情况.

将 $\boldsymbol{H}$ 及 $\boldsymbol{S}$ 输入 DMCNN 层的两个通道中[19],并学习组块特征

$$\boldsymbol{c}_{ij}^{h} = f(\boldsymbol{\omega}_{h_j} \cdot \boldsymbol{h}_{i:i+w-1} + \boldsymbol{b}_{h_j}), \tag{11}$$

$$\boldsymbol{c}_{ij}^{a} = f(\boldsymbol{\omega}_{a_j} \cdot \boldsymbol{s}_{i:i+w-1} + \boldsymbol{b}_{a_j}). \tag{12}$$

其中:f 是非线形激活函数,例如 tanh;w 指卷积的窗口大小;i 是句子中的单词位置,取值为 1 到 $n-w+1$;$\boldsymbol{h}_{i:i+w-1}$ 表示从 $\boldsymbol{h}_i$ 到 $\boldsymbol{h}_{i+w-1}$ 向量构成的矩阵,$\boldsymbol{s}_{i:i+w-1}$ 同理;j 是卷积核的索引,取值从 1 到 m_2,m_2 表示卷积核的个数;$\boldsymbol{c}_{ij}^{h}$ 为 Bi-LSTM 层输出向量通过 DMCNN 层卷积核的特征图,$\boldsymbol{c}_{ij}^{a}$ 指注意力机制层输出向量通过 DMCNN 层卷积核的特征图.

在 DMCNN 层之后通常需要连接池化层,传统的最大池化不能处理一个句子中包含多个类型的情况,为此需要采用动态池化的办法,将 DMCNN 层输出的每个特征映射都以候选触发词为界限,分为两个部分,对两个部分分别做最大池化,以此保留更多有价值的组块信息.

$$\boldsymbol{p}_{kj}^{h} = \max(\boldsymbol{c}_{kj}^{h}), \tag{14}$$

$$\boldsymbol{p}_{kj}^{a} = \max(\boldsymbol{c}_{kj}^{a}), \tag{15}$$

$$\boldsymbol{P} = [\boldsymbol{p}_{kj}^{h}, \boldsymbol{p}_{kj}^{a}]. \tag{16}$$

其中:$k=1,2$,因为一个句子在经过动态最大池化时被根据候选触发词分开为两个部分;$\boldsymbol{p}_{kj}^{h}$ 为 Bi-LSTM 部分的最大池化,$\boldsymbol{p}_{kj}^{a}$ 为注意力机制部分的最大池化;$\boldsymbol{P}$ 为 DMCNN 池化层的输出,$\boldsymbol{P} \in R^{m_2 \times 2}$.

1.5　输出层

输出层将 DMCNN 的输出 $\boldsymbol{P}$ 输入全连接层做最后的分类:

$$\boldsymbol{O} = \boldsymbol{W}_o \cdot \boldsymbol{P} + \boldsymbol{b}_o.$$

利用 Softmax 归一化计算识别候选触发词并将每个候选触发词分类为具体事件类别,$\boldsymbol{O} \in \mathbf{R}^{34}$.

1.6　训　练

在本模型中,损失函数选择交叉熵代价函数,并对所有参数进行随机初始化. 训练时,使用小批量随机梯度下降和 Adadelta 自适应梯度下降算法,并在编码层后和输出层前增加丢包(dropout)层避免过拟合.

2　实验结果

2.1　数据集与参数设置

实验所用语料为 ACE2005 英文语料. 对数据集的处理采用 Li 等[6]、Chen 等[7] 的方法,随机选择 ACE2005 语料库中的 30 篇文章作为验证集,40 篇文章作为测试集,剩余 529 篇文章作为训练集.

设置词向量 ELMo 训练层数为 2,位置向量维度为 5,实体类型向量维度为 50,Bi-LSTM 的隐藏层向量大小为 300,DMCNN 层滤波器窗口大小为 2 和 3,卷积核的个数为 100,L_2 正则化值为 10^{-6},编码层之后的 dropout 率为 0.3,输出层之前的 dropout 率为0.5,批量随机梯度的最小批为 100.

本次实验采用精确率(P)、召回率(R)和 F_1 值作为评价指标判断事件检测的正确性.

2.2　基准方法

为了验证本模型,本研究选择以下 2 类经典的模型作为基线模型.

1) 基于特征的模型有:以一些词义和语义信息作为特征的最大熵模型(MaxEnt)[2];交叉文档模型(Cross-Document)[3];交叉事件模型(Cross-Event)[1];交叉实体模型(Cross-Entity)[5];采用联合结构进行事件抽取的联合模型(Joint Model)[6].

2) 基于神经网络的模型有:DMCNN 模型[7];联合 RNN 模型(Joint RNN)[8];集成了 Bi-LSTM 和 CNN 的混合神经网络模型(HNN)[9];基于注意力机制神经网络模型(ANN+Attention),其运用了事件元素并引入了注意力机制[10];基于依存树的图卷积模型(GCN-ED)[13];自调节模型(SELF),利用生成式对抗网络(GAN)生成虚假特征进行自我调节[14];文本嵌入增强模型(DEEB),采用注意力机制获取文档特征,将文档特征引入模型[12].

2.3　向量化层特征选择

在向量化层,可以选择的特征有词向量、位置向量、实体类型向量,本节研究在词向量不变的情况下,增加位置向量或实体类型向量进行实验,结果如表 1 所示,可以看出增加特征对模型的性能提升均有帮助,而同时将词向量、位置向量、实体类型向量作为特征的 F_1 值最高,所以本模型选择词向量+位置向量+

表 1　特征选择比较

Tab. 1　Experimental results about feature choose　%

特征选择	F_1
词向量	66.5
词向量+位置向量	68.8
词向量+实体类型向量	72.9
词向量+位置向量+实体类型向量	**74.4**

实体类型向量作为向量化层的特征输入.

2.4 词向量的选择

对于词向量的选择,过往研究采用的都是word2vec词向量,本节实验在模型其他条件不变的情况下,比较使用word2vec和ELMo作为词向量的模型效果.使用word2vec作为词向量的模型F_1值为73.1%,而使用ELMo作为词向量的模型F_1值为74.4%,提升了1.3个百分点.这是因为ELMo是具有上下文语义信息的词向量,在一定程度上能解决词语二义性的问题,所以本研究选择ELMo作为词向量.

2.5 注意力机制的方法选择

在模型中引入注意力机制是为了捕捉与候选触发词相关性更高的特征,在模型其他部分均保持一致的情况下,通过实验比较引入注意力机制和不引入注意力机制对模型的影响.实验表明,不引入注意力机制的模型F_1值为73.3%,引入注意力机制后模型F_1值为74.4%,提升了0.9个百分点,说明注意力机制在本模型中可以有效捕捉与候选触发词相关性更高的特征.所以本模型中将引入注意力机制.

根据式(8),注意力机制的核心算法有3种.在模型其他部分不变的情况下,仅改变注意力计算方法进行实验,采用点乘方法的模型F_1值为74.4%,采用矩阵乘方法的模型F_1值为71.7%,采用连接方法的模型F_1值为70.0%,所以采用点乘计算方法可以达到更好的效果.此外,点乘计算方法也是最简单的计算方法,在时间复杂度和空间复杂度上都是最小的,因此在本研究选择点乘方法计算句子中词语与候选触发词的相似度.

2.6 实验结果比较

将本模型与基准方法进行比较,所有方法均在ACE2005英文语料上进行实验.从表5看出,本模型LM-ARCNN的F_1值达到了74.4%,高于其他所有模型.特别值得一提的是,DEEB模型通过在模型中引入了文档特征,获得74.0%的F_1值,是之前研究中性能最好的模型,但为了获得文档特征,其做事件检测时必须以整篇文档输入为前提,而本文中提出的模型对一个句子进行事件检测时无需输入整篇文章,只要输入当前句子,就可以取得比DEEB更好的结果.

分析本模型优于其他模型的原因有以下几点:

首先,相比于基于特征的模型,本模型的F_1值比最优的特征模型Cross-Event高了5.6个百分点.一方面是由于基于特征的模型依赖于其他的自然语言处理工具,累计的误差对性能影响较大且有效特征提取不足;另一方面,诸如词法、语法这类外部语义对于事件检测任务的作用有限,而神经网络可以将语义信息编码到高维的隐藏特征空间,可以提取更多特征.

表2 主要方法实验结果比较

Tab.2 Experimental results with main method %

方法	事件检测		
	P	R	F_1
MaxEnt[2]	74.5	59.1	65.9
Cross-Document[3]	60.2	76.4	67.3
Cross-Event[1]	68.7	68.9	68.8
Cross-Entity[5]	72.9	64.3	68.3
Joint Model[6]	73.7	62.3	67.5
DMCNN[7]	75.6	63.6	69.1
Joint RNN[8]	66.0	73.0	69.3
HNN[9]	84.6	64.9	73.4
ANN+Attention[10]	78.0	66.3	71.7
GCN-ED[13]	77.9	68.8	73.1
SELF[14]	71.3	74.7	73.0
DEEB[12]	72.3	75.8	**74.0**
LM-ARCNN	78.0	74.5	**74.4**

其次,相比于DMCNN、Joint RNN、ANN+Attention、GCN-ED、SELF这些仅使用单一神经网络的模型来说,本模型F_1值比其中最优的GCN-ED模型高了1.3个百分点,这是因为本模型结合了Bi-LSTM和DMCNN的优点,可以同时捕捉句子级别特征和上下文组块两部分信息,并通过注意力机制,赋予了重要特征更大的权重因子.

再次,与HNN混合模型比较,尽管HNN与本模型均使用了Bi-LSTM和CNN,但是本模型的F_1值仍比HNN模型高出1个百分点,且精确度P和召回率R相对来说更为均衡.此外,在多事件检测任务(见下文2.7节)中,本模型的性能也比HNN混合模型高出0.6个百分点(见下文表3),可见在多事件检测任务中,本模型也是优于HNN混合模型的.分析原因,主要有3点:首先本模型中Bi-LSTM和DMCNN的结合不是一个stacking方法的集成,而是作为深度神经网络中两个相邻的隐藏层,因此对特征信息是一种深层次的挖掘,而非简单的双重累积;其次本模型中的注意力机制和DMCNN层,能加大与候选触发词更相关的特征信息的权重,是一种有侧重的多特征挖掘,不易遗漏重要特征,在处理多事件句子时可以有更好

的性能;最后本模型引入了 ELMo 词向量,相较于 HNN 混合模型使用的 word2vec 词向量,在模型第一层,编码层就已经具有了一定的解决词语二义性的能力.

最后,与 DEEB 模型比较,本模型的 F_1 值比 DEEB 模型高出了 0.4 个百分点.尽管本模型没有提取文档特征,但是本模型使用的 ELMo 在单词表示上优于 DEEB 模型所使用的 word2vec,本身便具有上下文相关的特点,并且本模型可以同时捕捉句子级别和上下文组块特征,因此,在没有提取文档特征的情况下也可以有更好的性能.

2.7　多事件句子的抽取结果

为了进一步证明本模型在多事件句子中的性能,本文将测试集根据句子中事件的数量将测试集分为两个部分:句子中只包含一个句子的样本为单一事件集,句子中包含不止一个句子的样本为多事件集(表 3 中记为 1/N 集),并在多事件集里验证模型性能.表格 3 显示的是各个模型在多事件句子中的事件候选词抽取性能(F_1 值).其中,Embeddings+T 和 CNN 均在文献[7]中提到,Embeddings+T 使用的是词向量和文献[6]中使用的传统外部句子水平特征,CNN 则是传统的使用最大池化的卷积神经网络模型.

表 3　各模型在多事件句子的 F_1 比较

Tab. 3　Comparison of models' F_1 values in multiple-events sentence　　%

模　型	1/N 集	测试集
Embeddings+T[7]	25.5	59.8
CNN[7]	43.1	66.3
DMCNN	50.9	69.1
Joint RNN	64.8	69.3
HNN	65.6	73.4
LM-ARCNN	66.2	74.4

从表 3 中可以看出,当输入的句子包含不止一个事件时,本模型的 F_1 值最高,为 66.2%,比次优模型 HNN 高 0.6 个百分点.分析表中的模型,CNN 和 DMCNN 都属于 CNN 模型,Joint RNN 属于 RNN 模型,它们都比基于特征的 Embeddings+T 模型在多事件句子中的性能好,HNN 模型集成了 Bi-LSTM 和 CNN 网络,会比单纯只用循环神经网络或卷积神经网络的性能更优,而本模型使用了 Bi-LSTM 和 DMCNN 搭建的多层神经网络,挖掘句子的深层语义特征,不仅如此,还增加了注意力机制,帮助模型更加关注句子中与候选触发词相似度高的词语,而这些信息对多事件句子中的事件检测是非常有价值的,因此可以达到目前最好的性能结果.

3　结　论

本研究提出了一个新的神经网络模型 LM-ARCNN 进行事件检测,将 ELMo 词向量作为特征引入事件检测任务,并创新性地提出 Bi-LSTM 结合 DMCNN 的多层网络模型,且在卷积层使用注意力机制.在 ACE2005 英语语料库上进行实验,实验表明,本模型可以在仅输入句子的情况下可以达到目前最高的 F_1 值,F_1 值为 74.4%.

接下来的研究工作将会从两方面对模型进行改进:1) 考虑将残差模块引入模型,以提升模型事件检测的效果.2) 探究更适合事件检测任务的损失函数,对比不同损失函数对模型进行事件检测的影响.

参考文献:

[1] GRISHMAN R, WESTBROOK D, MEYERS A. NYU's English ace 2005 system description[J]. Journal on Satisfiability, 2005, 51(11): 1927-1938.

[2] AHN D. The stages of event extraction[C]// Proceedings of the Workshop on Annotating and Reasoning about Time and Events. Sydney: ACL, 2006: 1-8.

[3] JI H, GRISHMAN R. Refining event extraction through cross-document inference [C] // Meeting of the Association for Computational Linguistics. Columbus: ACL, 2008: 254-262.

[4] LIAO S, GRISHMAN R. Using document level cross-event inference to improve event extraction[C]// Meeting of the Association for Computational Linguistics. Uppsala: ACL, 2010: 789-797.

[5] HONG Y, ZHANG J, MA B, et al. Using cross-entity inference to improve event extraction[C]// Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland: ACL, 2011: 1127-1136.

[6] LI Q, JI H, HUANG L. Joint event extraction via structured prediction with global features [C] // Meeting of the Association for Computational Linguistics. Sofia: ACL, 2013: 73-82.

[7] CHEN Y, XU L, LIU K, et al. Event extraction via dynamic multi-pooling convolutional neural networks[C]// Meeting of the Association for Computational Linguistics. Beijing: ACL, 2015: 167-176.

[8] NGUYEN T H, CHO K, GRISHMAN R. Joint event extraction via recurrent neural networks[C]// Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego: NAACL-HLT, 2016: 300-309.

[9] FENG X, HUANG L, TANG D. A language-independent neural network for event detection[J]. Science China Information Sciences, 2018, 61(9): 92-106.

[10] LIU S, CHEN Y, LIU K, et al. Exploiting argument information to improve event detection via supervised attention mechanisms[C]// Meeting of the Association for Computational Linguistics. Vancouver: ACL, 2017: 1789-1798.

[11] DUAN S, HE R, ZHAO W. Exploiting document level information to improve event detection via recurrent neural networks [C] // Proceedings of the Eighth International Joint Conference on Natural Language Processing. Taipei: IJCNLP, 2017: 352-361.

[12] ZHAO Y, JIN X, WANG Y, et al. Document embedding enhanced event detection with hierarchical and supervised attention [C] // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers). Melbourne: ACL, 2018: 1-6.

[13] NGUYEN T H, GRISHMAN R. Graph convolutional networks with argument-aware pooling for event detection [C] // Association for the Advancement of Artificial Intelligence. New Orleans: AAAI, 2018: 5900-5907.

[14] HONG Y, ZHOU W, ZHANG J, et al. Self-regulation: employing a generative adversarial network to improve event detection [C] // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne: ACL, 2018: 1-12.

[15] MIKOLOV T, CHEN K, CORRADO G S, et al. Efficient estimation of word representations in vector space[EB/OL]. (2013-09-07)[2018-12-19]. http://arxiv.org/abs/1301.3781.

[16] PETERS M E, NEUMANN M, IYYER M, et al. Deep contextualized word representations[C]// Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. New Orleans: NAACL-HLT, 2018: 2227-2237.

[17] BA J L, KIROS J R, HINTON G E. Layer Normalization [EB/OL]. (2016-07-21)[2018-12-19]. https://arxiv.org/abs/1607.06450v1.

[18] LUONG M T, PHAM H, MANNING C D. Effective approaches to attention-based neural machine translation [C]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon: EMNLP, 2015: 1412-1421.

[19] YIN W, SCHÜTZE H, XIANG B, et al. ABCNN: attention-based convolutional neural network for modeling sentence pairs[J]. Transactions of the Association for Computational Linguistics, 2016, 4(1): 259-272.

Event detection via recurrent and convolutional networks based on language model

SHI Zheer, CHEN Jinxiu*

(School of Information Science and Engineering, Xiamen University, Xiamen 361005, China)

Abstract: Now main difficulties of event detection lie in polysemy and multi-event detection. To overcome these difficulties, we propose a novel recurrent and convolutional network with attention based on language model(LM-ARCNN). The model first learns word embeddings from Language Models (ELMo), and places these learned embeddings into a long-short term memory neural network (LSTM) which can capture sentence-level features. Then it utilizes attention mechanism to learn information from the learned sentence features to find the features which are more closely relative to candidate trigger words. Finally, it places these learned sentence features and attention features into a multi-pooling convolutional networks (DMCNN) which uses a dynamic multi-pooling layer according to event trigger to reserve more crucial context chunks. Experiments in ACE2005 English corpus show that the model achieves the state-of-the-art performance with F_1 value is 74.4%.

Keywords: event detection; embeddings from language models (ELMo); long short-term memory neural network(LSTM); dynamic multi-pooling convolutional neural networks (DMCNN); attention mechanism