等工作有重要意义. 随着时代的发展, 命名实体识别的目标早已超出了上述几类的范围, 特定领域的命名实体识别需求非常广泛, 如电子病历、生物医学等领域, 本文的子实验既是在会议名称识别这一特定领域的命名实体上展开的. 除了需要识别的实体在不断增加外, 命名实体识别方法也在不断进步和完善[3–5].

传统的命名实体识别多采用基于规则和统计机器学习的方法. 最初, 命名实体识别采用基于手工制定词典和规则的方法. 这些方法大多以语言学专家建立的规则知识库和词典为基础, 采用模式匹配或者字符串匹配的方法识别命名实体[6,7]. 对于规律性强的文本, 基于规则的方法准确而且高效. 但对于规律性不强的文本, 规则的编写变得困难, 识别效果也相当不理想, 所以人们开始将目光投向机器学习的方法.

在命名实体识别领域常用的机器学习方法有隐马尔可夫模型 (Hidden Markov Model, HMM)、条件随机场模型 (Conditional Random Fields, CRF)、最大熵模型 (Maximum Entropy)、支持向量机模型 (Support Vector Machine, SVM) 等[8–10]. 其中最典型的也是应用比较成功的是隐马尔可夫模型和条件随机场模型. 基于机器学习的方法在迁移性、识别效果等方面的表现优于基于规则的方法, 但使用统计机器学习方法的命名实体识别模型也存在一些局限性. 一方面, 为了使推理易于处理, 它需要明确的依赖性假设; 另一方面, 以统计模型为基础的机器学习方法对特征选取的要求比较高, 需要选择对命名实体识别任务有影响的各种特征, 即特征工程 (feature engineering), 它对识别结果有重要影响, 但是该过程费时费力[11–14]; 最后, 它们通常需要大量的与任务相关的特定知识, 如设计 HMM 的状态模型, 或选择 CRF 的输入特征.

随着深度学习研究的不断深入, 人们开始引入深度神经网络来处理自然语言. 2011 年 Collobert 提出一种基于窗口的深层神经网络模型, 该模型可以自动学习输入句子中的抽象特征, 在训练中使用反向传播算法来训练参数. 其效果和性能超过了之前的传统算法. 该模型的主要局限性是使用固定长度的上下文, 不能充分的利用语境信息[15–18]. Mikolov 于 2010 年提出一种基于循环神经网络 (Recurrent Neural Networks, RNNs) 的语言模型, 它不使用固定大小的上下文信息, 通过重复链接, 信息可以在这些网络内循环, 这种信息循环的方式非常适用于处理序列数据[19–24].

本文利用循环神经网络在处理序列数据方面的优势, 建立了基于循环神经网络的命名实体识别模型. 循环神经网络包含多种不同的变体, 经过分析对比, 本文最终选用了由 RNN 改进而来、结构相对简单的 GRU. 本文结构安排如下: 第 2 节介绍语料库的构建; 第 3 节阐述基于 GRU 的命名实体识别模型; 第 4 节是实验和分析; 最后做全文总结.

2 基于 GRU 的命名实体识别模型

以字作为输入单位容易产生歧义, 需要根据具体的语境信息判断每个字的标签, 循环神经网络能很好的计算和保持语境信息. RNN 是比较简单的循环神经网络, 不含“门”结构, 训练时会出现梯度消失或梯度爆炸的问题. LSTM 与 GRU 是在 RNN 的基础上改进而来的循环神经网络, 本文分析了 RNN、LSTM 和 GRU 三种循环神经网络之间的关系, 在此基础上, 提出了一个基于 GRU 的命名实体识别模型. 该模型以字向量作为输入, 经过双向 GRU 层计算, 提取句子特征, 经 softmax 层计算得到一个相应的输出序列.

2.1 循环神经网络

循环神经网络是深度学习中常用的一类神经网络, 包括 RNN 和 RNN 的变体 LSTM、GRU 等, 它利用序列信息并通过中间层保持这些信息, 这使它在处理序列数据时有独特优势.

RNN 将一个向量序列$(\boldsymbol{s}_1,\boldsymbol{s}_2,\cdots,\boldsymbol{s}_n)$作为输入, 得到另一个序列$\boldsymbol{h}_1,\boldsymbol{h}_2,\cdots,\boldsymbol{h}_n$, 其中$\boldsymbol{s}_t$表示时刻 t 的输入向量, $\boldsymbol{h}_t$是隐藏层状态, 表示关于每个输入时刻 t 的序列信息, 它的结构示意图如图 1. 其中,

$$\boldsymbol{h}_t = f(\mathbf{U}\boldsymbol{s}_t + \mathbf{W}\boldsymbol{h}_{t-1}) \tag{1}$$

$$\boldsymbol{o}_t = \mathbf{V}\boldsymbol{h}_t \tag{2}$$

U、**V**、**W** 是权值矩阵, f 是激活函数, 一般是 Sigmoid 或者 tanh 函数, $\boldsymbol{o}_t$ 是输出向量.

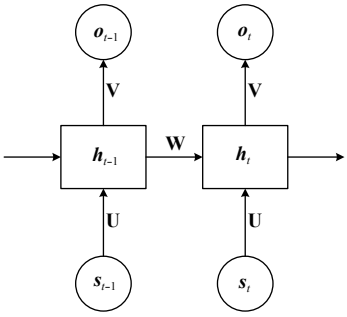

图 1 RNN 网络

理论上 RNN 可以学习长期依赖, 但实际上它不能实现这个目标, 因为根据求导的链式法则, 公式 (1) 使得损失函数的梯度成为连续乘积的形式, 这样做会导致梯度消失或者梯度爆炸的问题.

为了解决梯度消失 (或者梯度爆炸) 的问题, Hochreiter 和 Schmidhuber [25]提出了一种 RNN 的改进型网络: LSTM. 在自然语言处理中常用的 LSTM 可以用如下公式来描述:

$$\begin{cases} \boldsymbol{f}_t = \sigma\left(\mathbf{W}_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{s}_t] + \boldsymbol{b}_f\right) \\ \boldsymbol{i}_t = \sigma(\mathbf{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{s}_t] + \boldsymbol{b}_i) \\ \tilde{\boldsymbol{C}}_t = \tanh(\mathbf{W}_c \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{s}_t] + \boldsymbol{b}_c) \\ \boldsymbol{C}_t = \boldsymbol{f}_t \odot \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{C}}_t \\ \boldsymbol{o}_t = \sigma(\mathbf{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{s}_t] + \boldsymbol{b}_o) \\ \boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{C}_t) \end{cases} \tag{3}$$

其中, $\mathbf{W}$表示连接权值矩阵; $\boldsymbol{f}, \boldsymbol{i}, \boldsymbol{o}$分别是忘记门、输入门、和输出门; $\boldsymbol{C}$是细胞状态, 代表长期依赖信息; $\tilde{\boldsymbol{C}}$是候选向量, 表示当前的细胞状态. $\boldsymbol{f}_t$、$\boldsymbol{i}_t$、$\tilde{\boldsymbol{C}}_t$共同更新 $t-1$ 时刻的细胞状态$\boldsymbol{C}_{t-1}$得到 t 时刻的细胞状态$\boldsymbol{C}_t$; $\boldsymbol{h}_t$是时刻 t 的隐藏层状态, 由$\boldsymbol{o}_t$、$\boldsymbol{C}_t$共同决定. LSTM 通过加入门和细胞状态来控制传递给记忆单元的输入的比例, 以及记忆单元选择“忘记”的原来状态的比例, 既解决了梯度消失和梯度爆炸的问题, 又能学习长期依赖信息.

本文使用的 GRU 是 RNN 的另一种改进模型, 它与 LSTM 非常类似, 也是通过门来保持序列信息, 同时克服 RNN 中的梯度消失问题. 它的公式如下:

$$\begin{cases} \boldsymbol{r}_t = \sigma(\mathbf{W}_r \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{s}_t]) \\ \boldsymbol{z}_t = \sigma(\mathbf{W}_z \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{s}_t]) \\ \tilde{\boldsymbol{h}}_t = \tanh\left(\mathbf{W}_{\tilde{h}} \cdot [\boldsymbol{r} \odot \boldsymbol{h}_{t-1}, \boldsymbol{s}_t]\right) \\ \boldsymbol{h}_t = (1 - \boldsymbol{z}_t) \odot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \odot \tilde{\boldsymbol{h}}_t \end{cases} \tag{4}$$

其中, $\boldsymbol{z}_t$是一个更新门, 决定 $t-1$ 时刻的信息有多少进入 t 时刻.$\boldsymbol{r}_t$是一个重置门, 决定丢弃多少信息, 二者共同决定$\boldsymbol{h}_t$的值. GRU 只有两个门, 舍弃了 LSTM 中增加一个细胞状态$\boldsymbol{C}$的做法, 把线性自更新的过程放到了隐藏层状态的计算中, 把$\boldsymbol{h}$作为序列信息的载体, 这样做不但使得 GRU 的结构比 LSTM 更简单, 参数更少, 而且加快了神经网络的计算速度, 节省了时间.

2.2 命名实体识别模型

近年来, 很多学者使用前馈神经网络进行命名实体识别, 得到了不错的效果, 这种方法一般使用输入窗口, 即利用固定长度的输入来学习待识别词的上下文信息, 它的结构可以用图 2 描述.

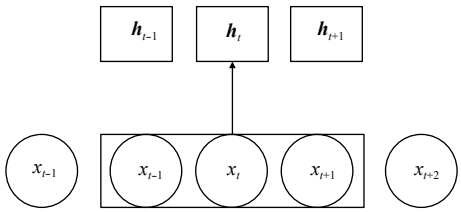

图 2 基于窗口的前馈神经网络结构

因为窗口大小有限 (如上图的窗口大小是 3), 所以该方法学习上下文信息的能力不足. 另一方面, 该神经网络的隐藏层状态之间没有联系, 不能够学习到长期依赖, 对于序列标注任务来说, 这意味着丢失了序列信息.

循环神经网络的输入则不受窗口大小的限制, 并且网络的隐藏层之间是有联系的, 即携带了序列信息, 如图 3 所示.

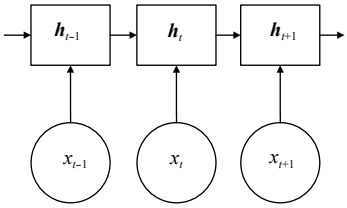

图 3 基于 RNN 的序列处理结构

图 2 和图 3 中, x_t 表示 t 时刻对应的字或者词. 以字作为输入单位, 命名实体的长度一般会超过三个字, 使用基于窗口的方法无法将整个名称作为一个整体考虑, 容易丢失信息, 致使边界判定准确率低, 而且基于字的判别对上下文环境的依赖程度更高. 循环神经网络学习上下文信息与窗口大小无关, 实现了学习长期依赖的目的, 适用于命名实体较长的情况. 本文分别使用循环神经网络的变体 LSTM 和 GRU 作为模型的基本算法, 并且对比说明了 GRU 的优势.

为了充分利用每个字的上下文信息, 这里使用了一个双向 GRU 结构, 即考虑了句子的正向和反向两个方向上的序列信息. 另外, 用于会议等新实体识别的用料库一般较小, 所以模型的结构不宜过复杂, 否则容易出现过拟合的现象. 本文设计的命名实体识别模型如图 4 所示.

图 4 中, 模型的第一层和第二层分别对应输入层和 Embedding 层. 由于神经网络不能直接处理自然语言符号, 所以, 应该将输入的字或者词转换成对应的向量, 即 Embedding 层的工作, 其中, 向量$\boldsymbol{E}_t$与x_t之间的关系满足$\boldsymbol{E}_t = LT(x_t)$, 即通过向量表$\boldsymbol{LT}$查找$x_t$对应的

© 中国科学院软件研究所 http://www.c-s-a.org.cn

向量. 输入$\boldsymbol{E}_t$经过前向 GRU 层和后向 GRU 层计算得到了句子的正向信息$\overrightarrow{\boldsymbol{h}_t}$和反向信息$\overleftarrow{\boldsymbol{h}_t}$, 两者共同组成了隐藏层状态, 这里用$\boldsymbol{h}_t$表示,$\boldsymbol{h}_t=\left[\overrightarrow{\boldsymbol{h}_t},\overleftarrow{\boldsymbol{h}_t}\right]$, 经过状态输出层计算得到$\boldsymbol{o}_t=f(\boldsymbol{h}_t)$, 再由 Softmax 层进行概率归一化计算, 得到最终的预测值.

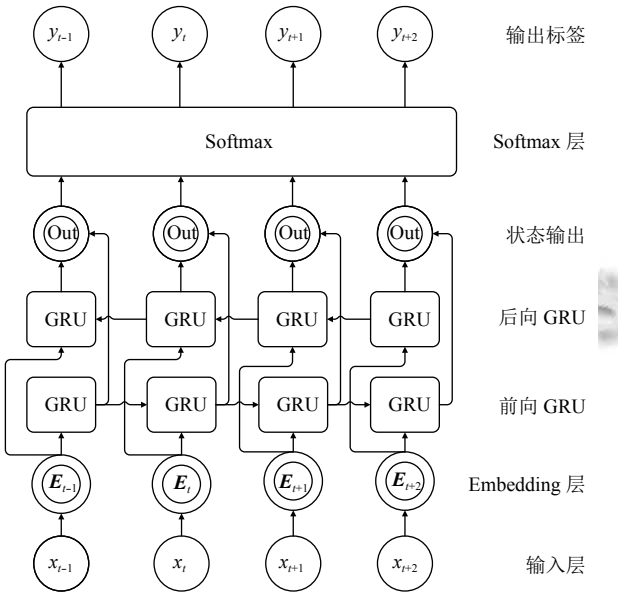

图 4 基于 GRU 的命名实体识别网络

这里可以把命名实体识别当作一个序列标注过程, 对于一个输入序列:

$$\boldsymbol{X}=(x_1,x_2,\cdots,x_n) \tag{5}$$

经过模型计算, 给每个输入x_t一个对应的标签, 得到一个相应的输出序列:

$$\boldsymbol{y}=(y_1,y_2,\cdots,y_n) \tag{6}$$

对于输入x_t, 预测结果为y_t满足:

$$\boldsymbol{y}_t=softmax(\boldsymbol{o}_t) \tag{7}$$

$\boldsymbol{y}_t$是一个长度为 n 的向量, n 是标签种类的个数, 即本文用向量表示预测结果, $\boldsymbol{y}_t$的每个位置上的值表示是该标签的概率:

$$\tilde{y}_i=\frac{e^{\tilde{y}_i}}{\sum_1^n e^{\tilde{y}_i}} \tag{8}$$

其中, $\tilde{y}_i$是$\boldsymbol{y}_t$的分量. 对于每个预测值, 我们取概率最大的分量所表示的标签作为最终的预测结果.

训练时, 使用交叉熵作为代价函数:

$$Loss=-\sum_{i=1}^{n} y_i\log\tilde{y}_i \tag{9}$$

其中, y_i表示真实值, $\tilde{y}_i$表示预测值, n 是标签的种类.

得到代价函数后, 通过训练集训练模型, 对于训练集的使用, 一般有批量梯度下降、随机梯度下降和 mini-batch gradient(小批量梯度下降) 三种方法, 本文使用第三种方法, 因为批量梯度下降方法每次计算损失函数都要遍历整个训练集, 并通过计算函数对所有参数的梯度来更新参数, 这种方法每更新一次参数就要遍历一次训练集, 导致计算开销很大, 计算速度较慢; 随机梯度下降法每次计算损失函数仅使用一条数据, 并通过梯度下降来更新参数. 与第一种方法相比, 虽然该方法的计算速度比较快, 但是收敛性能较差; 而第三种方法则结合了二者的优点, 它把训练数据分成多个批次, 计算损失函数时只使用其中的一批. 这样, 梯度的方向由这一小批次中的数据共同决定, 降低了方向选择的随机性, 提高了收敛性能; 同时, 与整个训练集相比, 每个批次数据量要小很多, 这降低了计算量, 节省了计算开销.

过拟合是使用人工神经网络时经常遇到的问题, 尤其是语料规模较小时, 更容易产生过拟合现象, 为了解决这个问题, 本文采用了 DropOut 方法. DropOut 是 Srivastava[26]提出的用来解决前馈神经网络过拟合问题的方法, 它通过在训练过程中随机的让一些神经元暂时停止工作来增加网络的稀疏性. 与权重衰减、过滤器范数约束等其它防止过拟合的方法相比, DropOut 方法更有效, 而且计算方便, 只需要在训练过程中产生相应的随机数来控制神经元的开闭即可. 在循环神经网络中, 将 DropOut 应用到非循环层, 这样做即能有效的避免过拟合的出现, 又能保证神经元之间不失去“联系”, 保存了序列信息.

本文提出的模型需要在监督训练数据上进行训练. 对于特定领域的命名实体识别来说, 往往没有可用的标注语料, 自己构建的语料库规模又比较小, 为了提高模型的效果, 一方面, 应当尽可能降低模型的复杂程度, 所以, 本文设计的模型只使用了一层双向 GRU 结构; 另一方面, 将已有的相关标注语料融合到自建语料库中, 可以丰富预料的语境语义特征, 提高识别效果, 第 3 节通过实验说明了该方法的可行性.

3 实验与分析

本文分别在人名这一传统命名实体和会议名称这个特定领域实体上进行了实验. 人名识别实验使用的

© 中国科学院软件研究所 http://www.c-s-a.org.cn

是人民日报语料库. 会议名称识别使用的是自建语料库. 由于本文的成果最终要运用到情报所的工作实践中, 所以两类实验使用的都是字向量, 第 3.2 节会进一步分析使用字向量的优势.

3.1 语料库构建

语料库是进行自然语言处理必须的数据集, 它承载着重要的语言知识. 基于神经网络的命名实体识别需要带标注的预料库作为训练数据. 本文除了在传统命名实体领域测试模型外, 还使用会议名称这一特定领域实体对模型进行了测试. 由于目前没有会议名称识别的语料库, 本文通过搜集文本、筛选、分词和标注等步骤构建了一个针对会议名称识别的语料库.

文本搜集是构建语料库的第一步. 本文搜集的文本来自中国学术会议在线网. 这里搜集了会议预告、会议新闻、会议评述、会议回顾的相关材料, 一共得到了 31.2 M 的初始文本.

第二步需要人工筛选搜集的文本, 删除大量与会议名称无关的段落, 并使用结巴分词对文本句子进行切分, 形成了初步标识的语料库.

第三步, 标注已经处理过的文本, 即人工标注会议名称. 会议名称有两种, 一种是简单会议名称, 如“中国医学会年会”、“香山论坛”、“十八届四中全会”等, 这类名称不存在嵌套, 属于简单的会议名称; 第二种是结构复杂的会议名称, 包括嵌套格式、并列结构等, 如“香山科学会议第 473 次学术讨论会”、“第四届中国古籍数字化国际学术研讨会暨第六届文学与资讯技术国际研讨会”. 本文采取的策略是最大化边界, 即将最完整的会议名称作为一个标注单位, 给予会议标签, 完成标注. 这样做的优势是可以根据后续实验的具体策略灵活的改变会议名称的标注方式, 既可以采用并列结构分别标注的方法, 也可以采用整体标注的方法.

完成上述步骤后, 得到了一个针对会议名称识别的专用语料库, 为实验提供了可用的数据集.

3.2 字向量

语料库中的数据并不能直接输入到本文的模型中, 因为神经网络只能计算数字, 所以应先将词或者字转换成数字形式, 即转换成向量. 传统的方法是先将文本分词, 以词向量作为输入单位. 以词作为输入单位需要首先进行分词, 本文没有采用先分词再识别的方法, 而是直接使用字向量作为模型的输入, 采用字向量有如下两点优势:

1) 以字作为输入单位可以避免不同分词系统及分词错误对识别效果的影响, 节省了分词时间, 提高了模型的识别速度;

2) 新领域的自建语料库相对较小, 如果以词作为输入单位, 那么测试时会遇到很多未登录词, 降低识别效果. 而常用汉字不超过 3000 个, 相对于词, 数据集覆盖的字的种类更加全面, 采用字向量可以避免因为未登录词过多而影响识别效果的情况.

会议名称识别中, 为了进一步提高模型的泛化能力, 训练字向量时采用了两种数据集, 一种是基于自建语料库的数据集, 记为 DataSet1; 另一种则通过加入人民日报语料库扩展了原有数据集, 这种扩展并没有增加带标注的会议语料的规模, 只是增加了与会议无关的文本, 达到丰富数据集特征的目的, 该数据集记为 DataSet2. 本文使用的字向量随循环神经网络训练过程产生.

3.3 标注策略与评价指标

命名实体识别中常用的标注策略有 BIO、BIEO、BIESO, O 表示不是实体, S 表示单个实体, B 表示实体其实边界, I 表示实体的中间部分, E 表示实体的结束边界. 由于本文采用的是输入形式是字向量, 所以必须同时找出起始和结束边界, 所以采用 BIEO 形式的标注策略. 命名实体识别的结果是否正确有两个标准: ① 实体的边界正确, 类型正确; ② 类型正确, 边界只覆盖了实体的一部分. 只要满足上述两个条件中的一个, 就认为识别结果是正确的.

本文使用精确率、召回率和 F 值来评价模型的识别效果. 命名实体识别可以看做分类问题, 以人名识别为例, 正类 (Positive) 表示该词的预测结果是人名, 负类 (Negative) 表示该词的预测结果不是人名. 人名被正确识别的个数为 TP, 即正类中真正的人名个数是 TP; 正类中非人名的个数为 FP; 负类中非人名的个数为 TN; 负类中被错误识别的个数为 FN, 即被预测为负类的人名的个数是 FN. 则精确率 P、召回率 R、F 值的计算公式如下:

$$P = \frac{TP}{TP + FP} \tag{10}$$

$$R = \frac{TP}{TP + FN} \tag{11}$$

$$F = \frac{P * R * 2}{P + R} \tag{12}$$

© 中国科学院软件研究所 http://www.c-s-a.org.cn

3.4 实验结果与分析

通过上述评价指标, 文章对比了 LSTM、GRU 和 RNN 的识别效果.表 1 是人名的识别结果, 可以看出, 三种网络在召回率方面相差较少, 但是 GRU 的识别结果的准确率更高, 得到了最高的 *F* 值.

表 1 人名识别实验结果

方法	*P*	*R*	*F*
GRU	93.88	86.79	90.20
LSTM	91.51	85.84	88.58
RNN	89.68	85.45	87.51

表 2 和表 3 分别是在 DataSet1 和 DataSet2 上对会议名称进行识别的结果, 观察两个表可知, 在两个数据集上, GRU 的都得到了最高的 F 值. 由表 2 实验结果可知, 在 DataSet2 数据集上训练的模型, 效果要好于 DataSet1 上训练的模型. 虽然 DataSet2 数据集只是增加了与会议无关的文本, 但它丰富了文本特征, 提高了模型的泛化能力.

表 2 DataSet1 会议名称识别结果

方法	*P*	*R*	*F*
GRU	97.33	84.88	90.68
LSTM	95.52	84.21	89.51
RNN	95.95	82.56	88.75

表 3 DataSet2 会议名称识别结果

方法	*P*	*R*	*F*
GRU	96.55	88.42	92.31
LSTM	98.79	86.32	92.13
RNN	93.75	78.95	85.71

表 4 是 GRU 和 LSTM 的训练时间, 根据实验结果可知, GRU 与 LSTM 的效果相差并不显著, 但是由于 GRU 结构相对简单, 参数较少, 所以它的训练时间比 LSTM 减少约 15%, 这是 GRU 的优势所在.

表 4 GRU 与 LSTM 训练时间对比

方法	训练时间 (s)
GRU	5168
LSTM	6102

4 结束语

本文提出了一个基于 GRU 的会议名称识别模型, 并且在传统命名实体和特定领域实体两类实体上进行了实验, 并且根据识别会议名称的需要搜集了会议文本, 通过机器分词和人工标注构建了语料库. 本文使用的方法的最大的优势是不依赖外部知识, 省去了人工设计特征的繁琐工作, 是一种端到端的会议名称识别方法. 识别特定领域命名实体是命名实体识别的一个发展方向, 但经常面临没有监督训练数据的困境, 自建的数据库又比较小, 这就在一定程度上限制了神经网络模型的泛化能力, 如何利用自建的小语料集训练一个具有足够泛化能力的神经网络模型是一个值得探索的方向.

参考文献

1 Marrero M, Urbano J, Sánchez-Cuadrado S, *et al*. Named entity recognition: Fallacies, challenges and opportunities. Computer Standards & Interfaces, 2013, 35(5): 482–489.

2 付瑞吉. 开放域命名实体识别及其层次化类别获取[博士学位论文]. 哈尔滨: 哈尔滨工业大学, 2014.

3 Durrett G, Klein D. A joint model for entity analysis: Coreference, typing, and linking. Transactions of the Association for Computational Linguistics, 2014, (2): 477–490.

4 Bhagavatula M, GSK S, Varma V. Named entity recognition an aid to improve multilingual entity filling in language-independent approach. Proceedings of the First Workshop on Information and Knowledge Management for Developing Region. 2012. 3-10. [doi: 10.1145/2389776.2389779]

5 杨锦锋, 于秋滨, 关毅, 等. 电子病历命名实体识别和实体关系抽取研究综述. 自动化学报, 2014, 40(8): 1537–1562.

6 Nadeau D, Sekine S. A survey of named entity recognition and classification. Lingvisticæ Investigationes, 2007, 30(1): 3–26.

7 王宁, 葛瑞芳, 苑春法, 等. 中文金融新闻中公司名的识别. 中文信息学报, 2002, 16(2): 1–6. [doi: 10.3969/j.issn.1003-0077.2002.02.001]

8 Han LF, Wong DF, Chao LS. Chinese Named Entity Recognition with Conditional Random Fields in the Light of Chinese Characteristics. In: Kłopotek MA, Koronacki J, Marciniak M, *et al*., eds. Language Processing and Intelligent Information Systems. IIS 2013. Lecture Notes in Computer Science, vol 7912. Springer, Berlin, Heidelberg. 2013. 74–85. [doi: 10.1007/978-3-642-38634-3_8]

9 邱泉清, 苗夺谦, 张志飞. 中文微博命名实体识别. 计算机科学, 2013, 40(6): 196–198. [doi: 10.3969/j.issn.1002-137X.2013.06.042]

10 Zhou GD, Su J. Named entity recognition using an HMM-based chunk tagger. Meeting of the Association for Computational Linguistics. Philadelphia, PA, USA. 2002.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

473–480.

11 Liu S, Tang B, Chen Q, *et al*. Drug name recognition: approaches and resources. Information, 2015, 6(4): 790–810. [doi: 10.3390/info6040790]

12 Graves A, Fernández S, Gomez F, *et al*. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. International Conference on Machine Learning. ACM. New York, NY, USA. 2006. 369–376

13 刘玉娇, 琚生根, 李若晨, 等. 基于深度学习的中文微博命名实体识别. 四川大学学报 (工程科学版), 2016, (S2): 142–146.

14 Nothman J, Ringland N, Radford W, *et al*. Learning multilingual named entity recognition from Wikipedia. Artificial Intelligence, 2013, (194): 151–175. [doi: 10.1016/j.artint.2012.03.006]

15 Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science, 2006, 313(5786): 504. [doi: 10.1126/science.1127647]

16 Bengio Y. Learning deep architectures for AI. Foundations & Trends® in Machine Learning, 2009, 2(1): 1–55. [doi: 10.1561/2200000006]

17 Dahl GE, Dong Y, Li D, *et al*. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on Audio Speech & Language Processing, 2012, 20(1): 30–42. [doi: 10.1109/TASL.2011.2134090]

18 Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv: 1409.0473.

19 Mikolov T, Karafiát M, Burget L, *et al*. Recurrent neural network based language model. INTERSPEECH 2010, Conference of the International Speech Communication Association. Makuhari, Chiba, Japan. 2010. 1045–1048.

20 Mikolov T, Kombrink S, Burget L, *et al*. Extensions of recurrent neural network language model. 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Prague, Czech Republic. 2011. 5528–5531. [doi: 10.1109/ICASSP.2011.5947611]

21 Labeau M, Löser K, Allauzen A, *et al*. Non-lexical neural architecture for fine-grained pos tagging. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015. 232–237.

22 Read J, Perezcruz F. Deep learning for multi-label classification. Machine Learning, 2014, 85(3): 333–359.

23 Ling W, Luís T, Marujo L, *et al*. Finding function in form: Compositional character models for open vocabulary word representation. arXiv: 1508.02096.

24 Chiu JPC, Nichols E. Named entity recognition with bidirectional LSTM-CNNs. arXiv: 1511.08308.

25 Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735–1780.

26 Srivastava N, Hinton GE, Krizhevsky A, *et al*. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 2014, 15(1): 1929–1958.

© 中国科学院软件研究所 http://www.c-s-a.org.cn