一种基于特征嵌入神经网络的中文分词方法

王文涛,穆晓峰,王玲霞
(中南民族大学 计算机科学学院,武汉 430074)

摘 要 针对传统基于特征的中文分词模型中,参数相对于训练数据过多而难以准确估计特征权值这一问题,提出了一种基于特征嵌入的神经网络方法. 嵌入方法将特征转化为低维实值向量,能有效降低特征维度. 另外,为了增强模型的性能,给出了一种学习速率线性衰减方法. 研究了正则项的方法来增强模型的泛化能力. 实验表明: 文中提出的模型可以提高中文分词问题的求解效率.

关键词 中文分词; 神经网络; 特征嵌入
中图分类号 TP183 **文献标识码** A **文章编号** 1672-4321(2017)01-0102-05

An Approach for Chinese Word Segmentation Based on Feature Embedding Neural Network

Wang Wentao, *Mu Xiaofeng*, *Wang Lingxia*
(College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China)

Abstract The feature weights are poorly estimated, because the number of parameters is much greater than the limited amount of training data under the traditional Chinese word segmentation model based on feature. To address above problem, this paper proposed an approach based on feature embedding neural network for Chinese word segmentation. The embedding method can reduce the dimensional of features because the model transforms features into low-dimensional real-valued vectors. In addition, in order to enhance performance of the model, we proposed a learning rate linear decay method. Finally, we studied the regularization method to enhance the generalization ability of the model. The experiment results showed that the model can improve the solving efficiency of Chinese word segmentation.

Keywords Chinese word segmentation; neural network; feature embedding

由于词是最小的能够独立运用的语言单位,而中文句子是一串连续的字符,没有明显的分隔符,因此自动分词问题成为了中文自然语言处理的基础性工作. 中文自动分词就是让计算机系统在中文文本中的词与词之间自动加上分隔符[1].

目前解决中文分词问题最流行的方法来自文献[2],其把中文分词看作序列化标注问题. 许多前期的工作集中在特征设计,比如: 一元文法特征、二元文法特征等一些其他特征. 基于特征的方法是最先进的中文分词系统的支柱.

但是基于特征的方法有两个问题: 一是需要人工设置特征,这不仅繁琐而且准确度不高; 二是由于特征的数目过大且训练集较小,导致特征的权重预测不准. 针对前一问题,研究人员开始探索采用神经网络的方法自动获取特征来解决中文分词问题,见文献[3,4]; 针对后一问题,采用嵌入的方法可以把原始的稀疏化输入转化为低维实数向量. 这一方法已经被多个 NLP 任务采用. 文献[4]适当地更改了文献[5]的模型,把输入的原始字符转化为字符向量,通过神经网络的训练可以得到最终的字符向量. 文献[5]采用张量神经网络进行分词,其在文献[4]的基础上加入了当前预测字符的前一个字符的状态作为特征.

本文不同于文献[3,4,6]对字符学习字符向量,而立足于特征,对其进行学习. 传统方法中特征数目过多而难以学习,本文开始抽取 11 个特征,期

收稿日期 2016-07-03
作者简介 王文涛(1967-),男,副教授,博士,研究方向: 计算机网络与控制,E-mail: wangwt@mail.scuec.edu.cn
基金项目 国家民委教改项目(15013); 中南民族大学研究生创新基金资助项目(2016sycxjj199)

望最终学习到的特征向量不仅包含了这 11 个特征,同时也包含了这些特征之间的交互作用.

1　基于特征嵌入的神经网络方法

本文提出了一种基于特征嵌入的神经网络方法解决中文分词问题,神经网络结构如图 1 所示.

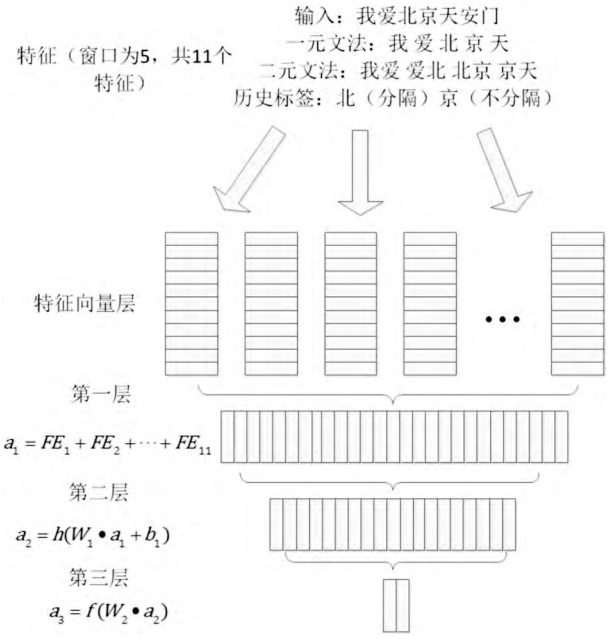

图 1　神经网络结构

Fig. 1　The architecture of neural network

1.1　序列标注描述

我们采用文献 [7] 的方法对一个给定序列进行标注. 使用分隔符 S 和合并符 C 来表示字符的状态. 其中分隔符代表与前一个字符是分开的,合并符代表与前一个字符是链接的,举例如下.

(a) 我爱北京天安门

(b) 我 爱 北京 天安门

(c) SSSCSCC

以上(c) 行就代表(a) 行原始字符序列的状态. 通过(a) 行的字符序列和(c) 行的状态序列就能得出(b) 行的字符分隔序列.

1.2　特征描述

输入: 输入为一串未分隔的原始字符序列,我们通过下面的特征定义提取其相应的特征.

(1) 上下文特征. 本文采用窗口数为 5,c_i 代表第 i 个字符为当前正在处理的字符,h 代表窗口大小,其为当前正在处理的字符加入前 $h/2$ 和后 $h/2$ 个字符. 例如, "我爱北京天安门",当前正在处理的字符为"北",则整个窗口的字符序列为"我爱北京天",其一元文法为: 我,爱,北,京,天,二元文法为: 我爱,爱北,北京,京天.

(2) 历史特征. 在本文的模型中,我们采用当前字符的前两个字符的状态作为历史数据. 例如,当前需要处理的字符是"我爱北京天安门"中的天,其前两个字符是北京,状态为 S 和 C.

1.3　神经网络结构

1.3.1　查找表

这里采用分布式表示的方法来表示特征,其把单个的汉字字符转换为低维的实向量. 最先是由文献 [8] 提出并随之成为一研究热点. 查找表形式化描述如下: 有一中文特征库 D,其大小为 $|D|$,其由训练集所有上文提及的特征构成并加入一特殊特征 OOV(out of vocabulary),其表示并不出现在 D 中的字符.

在窗口内获取一元文法和二元文法,并获取历史特征,之后经过查找表的查找转换为特征向量,$Embed(c) \in R^d$,$Embed$ 代表查表操作符,输入特征 c,$c \in D$,返回其特征向量,d 代表特征向量的维数,特征向量表 $M \in R^{d \times |D|}$. 具体流程见图 1 特征转换至特征向量层.

1.3.2　神经网络其他部分

神经网络第一层如图 1 所示,把查找得到的特征连接到一起,作为神经网络的第一层输出,$a_1 \in R^{d \times k}$,k 为特征的个数. 之后,a_1 作为输出进入神经网络的第二层,其经过激活函数为 $sigmoid$,如下:

$$a_2 = sigmoid(W_1 a_1 + b_1). \tag{1}$$

$W_1 \in R^{h_2 \times h_1}$,$h_2$ 为第二层的单元数,h_1 为第一层的单元数即 a_1 的维度. 为了得到当前字符状态是 S 或 C 的概率,神经网络的第三层有两个单元,激活函数采用 $softmax$,如下:

$$a_3 = softmax(W_2 a_2). \tag{2}$$

$W_2 \in R^{h_3 \times h_2}$,$h_3$ 为第三层的单元数即为 2,h_2 为第二层的单元数.

1.4　模型的训练与预测

1.4.1　句子预处理

本模型每次处理一个句子,之后对每个句子从头到尾逐字符进行训练与预测. 因句子的第一个字符前没有字符,而本模型每次训练和预测都需要当前待处理字符的前两个字符状态作为历史特征,故对每一个句子的起始位置前加入两个哑字符,其状态为 S.

1.4.2　预测

对每个句子,从左至右依次对每个字符进行预

测. 预测过程如下: 取当前预测的字符的特征信息,输入至神经网络中,如图1所示,最终由神经网络的输出层输出结果.

本模型第三层有两个神经元,分别代表状态S和状态C,归一化后概率和为1,那么这两个神经元若其一大于0.5,即把当前字符设置为该状态.

1.4.3 训练

训练过程如下: 对某一个字符进行预测,之后通过预测的状态与样本正确的状态进行对比,进而训练模型. 这里我们采用交叉熵损失函数:

$$J = -\sum_{k=1}^{2}\delta(s_{i,k})\lg\frac{\exp(a_{3i})}{\sum_{j}\exp(a_{3j})}, \tag{3}$$

$$\delta(s_{i,k}) = \begin{cases}1, \text{if } s_{i,j} = \hat{s}_i,\\ 0, \text{else.}\end{cases} \tag{4}$$

$\delta(s_{i,k})$ 含义为: 若预测的第 i 个字符的状态 $\hat{s}_i$ 与样本的第 i 个字符的第 k 种状态 $s_{i,k}$ 相同,则其值为1,否则为0. 模型训练采用误差向后传播,参数更新公式为:

$$\theta = \theta - \alpha\cdot\frac{\partial J}{\partial\theta}, \tag{5}$$

参数 $\theta = \{M, W_1, W_2, b_1\}$,注意到我们的模型不仅对神经网络的权值进行训练,同时对特征向量进行训练,这种训练方式类似于文献[9],α 为模型学习速率,$\partial J/\partial\theta$ 为损失函数的梯度见(6)式,$\partial J/\partial a_1$ 见(7)式:

$$\frac{\partial J}{\partial\theta} = \left\{\frac{\partial J}{\partial a_1}, \frac{\partial J}{\partial W_1}, \frac{\partial J}{\partial W_2}, \frac{\partial J}{\partial b_1}\right\}, \tag{6}$$

$$\frac{\partial J}{\partial a_1} = \frac{\partial J}{\partial z_2}\cdot\frac{\partial z_2}{\partial a_2}\cdot\frac{\partial a_2}{\partial z_1}\cdot\frac{\partial z_1}{\partial W_1}, \tag{7}$$

其中 a_1 是 M 中的元素,$z_1 = W_1a_1 + b_1$,$z_2 = W_2a_2$. 训练算法描述如下.

输入: 训练语料库 corpus,已初始化的权值 Θ,学习速率 α.

输出: 训练后的权值 Θ.

步骤:

For *sentence* in *corpus*:

  For *word* in *sentence*:

  // s' 如(4)式所示

    $s' = \text{predict}(word)$

    $\Theta = \Theta - \alpha\cdot\partial J/\partial\Theta$

  End For

End For

2 实验及其分析

2.1 数据及评测度量

在实验中,我们采用广泛使用的分词语料库PKU语料库[10]. 表1为其详情,我们使用其官方提供的评分脚本对本模型进行评估.

表1 PKU语料库

Tab.1 PKU corpus

类别	数量/个
词类型	5.5×10^4
词总数	1.1×10^6
字符类型	5×10^3
字符总数	1.8×10^6

参数设置如下: 通常来说,隐含层的单元数目对模型的性能是有影响的,文献[11]指出,隐含层单元数目越多越能捕获数据的非线性因素; 反之,模型整体会受影响. 值得注意的是,隐含层单元数目过多有两点影响: 模型训练速度过慢; 模型捕获其他不相关的特征,导致模型过拟合. 综合考虑,本文设置隐含层单元数目为50. 另外,通过实验发现: 本模型特征向量的维度取50既能保证模型的性能,同时训练速度不至于过慢. 训练的特征个数为11个: 一元文法特征5个、二元文法特征4个、字符状态特征2个. 采用两种方法设置学习速率 α: 固定学习速率和线性衰减学习速率.

本文采用 P(Precision)、F 值、IV 召回率(In Vocabulary Recall)和 OOV 召回率(Out of Vocabulary Recall)这三种度量方法对本文模型进行评估. 准确率是正确的分词数除以分词结果中总词数. 召回率是正确的分词数目除以标准数据集中的总词数.

$$F = \frac{2\cdot P\cdot R}{P + R}, \tag{8}$$

这里的 R 指的是IV召回率. OOV召回率主要用来评测模型的泛化能力.

2.2 固定学习速率实验

在本实验中,对不同的学习速率 α 进行研究,结果见表2.

从表2可以看出,在 α 为0.1和0.2的情况下,本模型的 F 值和IV召回率较好. 而其他情况下,性能较低. 我们推测: α 值过小,随机梯度下降不能很好地达到局部最优值; 而 α 值较大会越过局部最优值. 因此采用固定学习速率的模型性能得不到保证.

表 2　不同 α 的性能

Tab. 2　The performance of different α

α	F 值	OOV Recall	IV Recall
0. 01	0. 688	0. 330	0. 702
0. 02	0. 696	0. 376	0. 722
0. 03	0. 699	0. 363	0. 721
0. 04	0. 728	0. 344	0. 746
0. 05	0. 808	0. 384	0. 837
0. 06	0. 778	0. 355	0. 796
0. 07	0. 767	0. 374	0. 773
0. 08	0. 769	0. 355	0. 796
0. 09	0. 776	0. 363	0. 810
0. 1	0. 880	0. 336	0. 924
0. 2	0. 878	0. 345	0. 914
0. 3	0. 557	0. 252	0. 593
0. 4	0. 518	0. 145	0. 602
0. 5	0. 382	0. 143	0. 454
0. 6	0. 166	0. 092	0. 137
0. 7	0. 344	0. 160	0. 366
0. 8	0. 346	0. 081	0. 456
0. 9	0. 136	0. 066	0. 104

2. 3　学习速率 α 线性衰减实验

基于以上推测,进而采用学习速率 α 线性衰减的方法学习模型,训练起始时 α 较大,迅速达到较优值附近,之后 α 值变小,进行更加精细的搜索. 由此,提出学习速率 α 线性衰减方法:

$$\alpha = \alpha' - (\alpha' - \alpha'') \cdot (W_p / W_{total}), \quad (9)$$

式中 α 代表当前值,α' 代表 α 的最大值,本实验为其初始值,α'' 代表 α 的最小值,本实验设置其为 0. 00001,W_p 为已经处理字符个数,W_{total} 为总字符个数.

表 2 中 α 为 0. 1 和 0. 2 时模型的性能最好,为了寻找更好的 α,本实验我们设置其初始值从 0. 08 增大至 0. 21,考察其对模型性能的影响,结果见表 3.

表 3　α 线性衰减

Tab. 3　Linear decay of α

变化范围	F 值	OOV Recall	IV Recall
0. 08-0. 00001	0. 883	0. 322	0. 930
0. 09-0. 00001	0. 842	0. 377	0. 885
0. 10-0. 00001	0. 827	0. 353	0. 873
0. 11-0. 00001	0. 878	0. 386	0. 918
0. 12-0. 00001	0. 895	0. 330	0. 939
0. 13-0. 00001	0. 894	0. 346	0. 933
0. 14-0. 00001	0. 890	0. 330	0. 937
0. 15-0. 00001	0. 890	0. 322	0. 937
0. 16-0. 00001	0. 891	0. 328	0. 936
0. 17-0. 00001	0. 892	0. 292	0. 938
0. 18-0. 00001	0. 889	0. 301	0. 938
0. 19-0. 00001	0. 897	0. 329	0. 934
0. 20-0. 00001	0. 897	0. 355	0. 935
0. 21-0. 00001	0. 894	0. 318	0. 937

在表 3 中,可以看出,在 α 为 0. 08 时,F 值为 0. 883 而在表 2 中为 0. 769,可见性能得到了显著提升,其他情况类似. 以 F 值为比较因子来看,表 3 中 α 值为 0. 20 的情况下,模型的 F 值和 IV 召回率分别为 0. 897 和 0. 935,分别高于表 2 中 α 值为 0. 10 情况下的 0. 880 和 0. 924. 综上,得出结论,学习速率 α 线性衰减方法确实可以改善模型的性能.

2. 4　其他模型对比

使用 CRF、TNN、CNN 模型作对比实验,结果如表 4. 从表 4 可以看出: NN(Neural Network) 模型在精确率、召回率和 F 值上优于 CRF,但是在 OOV 召回率上性能比较差,这说明我们的模型对不在训练集里的词的分词效果不好,本模型的泛化能力有待加强. 解决的方法有: 1) 增加训练集,但是这种方法需要更多的人工标注训练集; 2) 增加正则化项,改善模型的泛化能力.

表 4　性能细节

Tab. 4　Details of performance

模型	Precision	Recall	F 值	OOV Recall
CRF	0. 878	0. 857	0. 867	0. 571
NN	0. 893	0. 902	0. 897	0. 355
TNN	0. 786	0. 791	0. 788	–
CNN	–	–	0. 807	–

为了说明对特征向量进行训练对性能的影响,比较了 TNN 模型,其仅对权值进行训练. 实验中,在相同条件下,TNN 模型收敛速度很慢,而采用我们的模型损失会迅速降低,在第一轮迭代后,测试集上的实验结果就可以达到很好的性能,第二轮迭代后,性能指标基本稳定. 表 4 则是 10 轮迭代后的 TNN 性能指标,其明显说明采用 TNN 而不对特征向量进行训练,其性能较差.

在分词任务中,语序是一种关键信息. 为了说明把特征向量直接连接可以隐含地抓住语序的信息,比较了 CNN 模型,其把多个特征映射为一个. 在文献 [12] 中,作者设计了一个 CNN 模型解决命名实体识别和中文分词问题,其性能如表 4 所示,这证明了采用连接的操作是有效的.

为了研究本模型的泛化能力,对分词结果作了分析,发现: 一旦出现新的人名、地名、时间等名词,本模型就不能很好地处理分词结果,导致 OOV 召回率性能较差,对此问题及解决方法还需要进一步研究.

2. 5　正则化

为了改善本模型的泛化能力,对(3) 式增加一正则项见(10) 式,J_{reg} 为正则项如(11) 式所示,其中 λ 为正则项系数,权衡其与 J 的比重. 对新的损失函数求梯度,见(12) 式:

$$J' = J + J_{reg}, \tag{10}$$

$$J_{reg} = \frac{\lambda\theta^2}{2}, \tag{11}$$

$$\frac{\partial J'}{\partial \theta} = \frac{\partial J}{\partial \theta} + \frac{\partial J_{reg}}{\partial \theta} = \frac{\partial J}{\partial \theta} + \lambda \cdot \theta. \tag{12}$$

当 λ 为 0.000001 和 0.0000001 时,其相关评测数据见表 5,第 3 行是不使用正则化模型的评测数据. 表 5 的数据说明本模型使用正则的方法并没有使其达到更好的性能. 在文献[5]中,无论对词级别的训练还是对句子级别的训练,都没有使用正则项,且没有说明原因.

表5 不同 λ 的性能

Tab. 5 The performance of different λ

λ	Precision	Recall	F 值	OOV Recall
0.000001	86.6	91.1	86.6	33.7
0.0000001	86.0	91.2	84.5	29.6
–	89.3	90.2	89.7	35.5

需要指出的是,当 λ 为 0.1, 0.01, 0.001, 0.0001, 0.00001 时,*sigmoid* 函数中的 exp 函数均会溢出. 我们发现当实验处理到第 31 个句子时,输入向量其中某一分量值已为 1444,调用 exp 函数后会溢出. 原因是:当训练过程中模型的梯度已经很小,而权值 W 乘以一个 λ 反而会使模型中的权重变大,如上所述,其值达到了 1444,这也是促使选择更小的 λ 的原因.

综上得出结论:在本模型中,为了达到更好的泛化性能,简单地增加正则项的方法不适用,需要更加精确的控制方法.

3 结论

本文提出的基于特征嵌入的神经网络方法可以有效解决中文分词问题. 同时,训练出了针对中文分词的特征向量,在目前涌现的众多深度学习方法都需要同类方法生成的向量作为前置输入,在未来将对学习到的特征向量进行研究. 另外,本文提出的学习速率 α 线性衰减方法经过实验证明可有效提高本模型的性能. 而针对模型泛化能力的实验说明:仅仅增加一正则项的方法不能增加模型的泛化能力. 最后,本模型的泛化能力不足,这是下一步研究的重点.

参 考 文 献

[1] 宗成庆. 统计自然语言处理[M]. 北京:清华大学出版社, 2008:129-130.

[2] Xue N. Chinese word segmentation as character tagging[J]. 中文计算语言学期刊, 2003, 8(8:1):29-48.

[3] Mansur M, Pei W, Chang B. Feature-based neural language model and Chinese word segmentation[C]//ACL. IJCNLP 2013: The 6th International Joint Conference on Natural Language Processing. Nagoya: ACL Press, 2013: 1271-1277.

[4] Zheng X, Chen H, Xu T. Deep learning for Chinese word segmentation and POS tagging[C]//ACL. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP02013). Seattle: ACL Press, 2013: 647-657.

[5] Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch[J]. Journal of Machine Learning Research, 2011, 12(1): 2493-2537.

[6] Pei W, Ge T, Chang B. Max-margin tensor neural network for Chinese word segmentation[C]//ACL. The 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore: ACL Press, 2014: 293-303.

[7] Ma J, Hinrichs E. Accurate linear-time Chinese word segmentation via embedding matching[C]//ACL. The 53rd Annual Meeting of the Association for Computational Linguistics. Beijing: ACL Press, 2015: 1733-1743.

[8] Hinton G E. Learning distributed representations of concepts[C]//CSS. Proceedings of the Eighth Annual Conference of the Cognitive Science Society. Mahwah: Lawrence Erlbaum Associates, 1986: 1-12.

[9] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[OL]. (2013-09-07)[2016-12-01]. https://arxiv.org/abs/1301.3781.

[10] Emerson T. The second international Chinese word segmentation bakeoff[C]//ACL. Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing. Jeju Island: ACL Press, 2005: 123-133.

[11] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning[M]. New York: NY Springer, 2001: 400-401.

[12] Liu Y, Che W, Guo J, et al. Exploring segment representations for neural segmentation models[OL]. (2016-04-19)[2016-12-01]. https://arxiv.org/abs/1604.05499.