北京大学学报(自然科学版) 第 53 卷 第 2 期 2017 年 3 月
Acta Scientiarum Naturalium Universitatis Pekinensis, Vol. 53, No. 2 (Mar. 2017)
doi: 10.13209/j.0479-8023.2017.032

一种基于循环神经网络的古文断句方法

王博立[1] 史晓东[1,2,3,†] 苏劲松[4]

1. 厦门大学智能科学与技术系，厦门 361005; 2. 厦门大学两岸关系和平发展协同创新中心，厦门 361005; 3. 福建省类脑计算技术及应用重点实验室，厦门 361005; 4. 厦门大学软件学院，厦门 361005; † 通信作者, E-mail: mandel@xmu.edu.cn

摘要 提出一种基于循环神经网络的古文自动断句方法。该方法采用基于GRU (gated recurrent unit)的双向循环神经网络进行古文断句。在解码过程中，该算法不仅利用神经网络输出的概率分布，还进一步引入状态转移概率和长度惩罚，以便提高断句准确率。在大规模古籍语料上的实验结果表明，所提方法能够取得比传统方法更高的断句F1值。

关键词 古汉语；断句；循环神经网络

中图分类号 TP391

A Sentence Segmentation Method for Ancient Chinese Texts Based on Recurrent Neural Network

WANG Boli[1], SHI Xiaodong[1,2,3,†], SU Jinsong[4]

1. Department of Cognitive Science, Xiamen University, Xiamen 361005; 2. Collaborative Innovation Center for Peaceful Development of Cross-Strait Relations, Xiamen University, Xiamen 361005; 3. Fujian Province Key Laboratory for Brain-inspired Computing, Xiamen 361005; 4. Software School, Xiamen University, Xiamen 361005; † Corresponding author, E-mail: mandel@xmu.edu.cn

Abstract This paper proposes an automatic sentence segmentation method for ancient Chinese texts based on recurrent neural network (RNN). A bi-directional RNN structure with gated recurrent units (GRU) is implemented, and state transition probability and length penalty are employed in decoding to improve the accuracy. Experimental results show that proposed model achieves higher F1 score than traditional methods.

Key words ancient Chinese; sentence segmentation; recurrent neural network

数千年的中华文明留下浩如烟海的古籍，这些古籍对现代人了解古代历史、社会和文化发展具有重要的价值。但是，古汉语中没有标点符号。古人著书时，通常不对句子停顿进行标记，而是由读者阅读时自行标记，即“句读”，这给现代人阅读和研究古籍带来很大的困难。利用最新的自然语言处理技术，对大量未断句的古文进行自动断句，不仅能帮助人们克服阅读障碍，也是进一步对古籍文本进行处理(如古文分词等)所必要的前期工作，对于古汉语研究、古籍整理与文史知识挖掘具有重要的意义。

1 相关工作

黄建年等[1]采用计算机辅助的方法，从已断句的古文中提取句子切分的特征模式，利用这些特征模式构造断句规则，采用正则表达式替换的方法，进行古文断句。实验结果表明，上下文特征对古文断句具有重要的作用，但这种基于规则的方法并不适用于大规模古籍处理。

陈天莹等[2]最早采用统计方法进行古文断句，他们将古文断句看成一个分类问题，提出一种基于上下文 N-gram 模型的古文断句方法。该方法利用

教育部专项“简繁汉字智能转换系统”、国家科技支撑计划项目(2012BAH14F03)、教育部博士点基金(20130121110040)、国家自然科学基金(61573294)和 CCF 中文信息技术开放课题(CCF2015-01-01)资助

收稿日期: 2016–07–29; 修回日期: 2016–10–07; 网络出版日期: 2016–11–30

训练语料上的频率统计信息，计算文本中各处需要断句的概率，并提出一种数据平滑算法。

近年来的研究多将古文断句视为与中文分词类似的序列标注问题，采用条件随机场(conditional random field, CRF)[3]对古文断句问题进行建模，利用手工设计的特征模板，在规模较小的语料上训练断句模型。张合等[4]采用手工设计的基于上下文汉字的特征模板，张开旭等[5]引入互信息(mutual information)和 *t*-测试差(*t*-test difference)两个统计量作为特征，Huang 等[6]引入汉字音韵信息(现代汉语拼音、反切、广韵)作为特征。

这些研究表明，采用序列标注模型，并以上下文信息作为特征，能有效地进行古文断句，但存在明显不足：1) 实验使用的数据集规模均较小，未在大规模古籍语料上开展实验；2) 受限于固定的特征模板，特征数量少，仅能利用一个固定的较小窗口内局部上下文信息，并且一些额外的信息(如广韵)难以涵盖古籍中出现的所有汉字；3) 特征模板的设计完全依赖于人的先验知识，若想进一步提高模型的断句准确率，需要通过特征工程进行大量的实验，选择新的特征。

Wang 等[7]提出一种基于神经网络语言模型的古文断句方法，但该方法未达到传统条件随机场模型的断句效果，并且由于在解码过程中需要反复调用神经网络语言模型计算概率得分，因此断句速率较慢，难以达到实际应用的要求。

本文提出一种基于循环神经网络(recurrent neural network, RNN)[8]的古文断句方法。该方法通过神经网络自动学习上下文的特征表示，无需人为选择特征。我们在大规模古籍语料上开展实验，结果表明本文方法能取得比传统方法更高的 F1 值。

2 基于循环神经网络的古文断句方法

本文将古文断句视为一个典型的序列标注问题，采用“端到端”(end-to-end)的思想，根据输入的汉字序列，直接利用神经网络计算标注的条件概率，并进一步引入状态转移概率和长度惩罚，通过 beam search 算法进行解码。

2.1 标注集

古文断句可以视为一个基于字的序列标注问题：为给定的未断句的古文文本中的每个汉字标注标签。本文将训练集和测试集中的 6 种标点(。？！，；：)视为断句的标记，而忽略文本中的其余标点。

本文采用与文献[4]相同的六元标注集 T = {B, M, E3, E2, E, S}。若句子长度为 1，即仅由一个汉字构成，则该汉字标记为 S；若句子长度大于 1，则将句子开头标记为 B，将句子结尾标记为 E；若句子长度大于 2，则将句子结尾的倒数第 2 个汉字标记为 E2；若句子长度大于 3，则将句子结尾的倒数第 3 个汉字标记为 E3；若句子长度大于 4，则将句子中其余汉字标记为 M。例如，“曰：‘朕此衣已三浣矣。’”一句对应的汉字序列为“曰 朕 此 衣 已 三 浣 矣”，正确的标签序列为“S B M M M E3 E2 E”。

我们采用这种六元标注集，而非传统中文分词任务所采用的四元标注集{B, M, E, S}，主要考虑句子长度比词语长度更长，而古汉语句子末尾的几个字常常有助于判别句子的边界(如“乎”“也”)。

测试时，我们在标注为 E 和 S 的汉字后添加断句标记。

2.2 模型

RNN 是一种常见的深度神经网络模型，由于能有效地利用上下文信息，广泛应用于序列标注任务中。传统的 RNN 只能沿迭代方向利用单侧的历史信息，Schuster 等[9]提出双向循环神经网络(bi-directional recurrent neural network)，能同时利用正向和反向的上下文信息。

本文提出的断句模型采用基于 GRU 的双向循环神经网络，结构如图 1 所示。

模型的输入为汉字序列(即未断句的古文段落)$x_1, x_2, \ldots, x_N$，输出为标签的概率：

$$\boldsymbol{o}_t^{[i]} = P(y_t = i \mid x_1 x_2 \ldots x_N), \tag{1}$$

其中，i 为标注集 T 中的标签。

1) 在输入层，汉字序列中的每个汉字 x_t 首先被映射为对应的字向量 $\boldsymbol{e}_t$，作为各个时刻循环神经网络的输入。

2) 在隐层，双向循环神经网络根据当前时刻的输入 $\boldsymbol{e}_t$ 以及前一时刻的正向隐状态 $\overrightarrow{\boldsymbol{h}}_{t-1}$ 和后一时刻的反向隐状态 $\overleftarrow{\boldsymbol{h}}_{t+1}$，分别计算当前时刻的正向隐状态 $\overrightarrow{\boldsymbol{h}}_t$ 和反向隐状态 $\overleftarrow{\boldsymbol{h}}_t$：

$$\overrightarrow{\boldsymbol{h}}_t = f(\boldsymbol{e}_t, \overrightarrow{\boldsymbol{h}}_{t-1}), \tag{2}$$

$$\overleftarrow{\boldsymbol{h}}_t = f(\boldsymbol{e}_t, \overleftarrow{\boldsymbol{h}}_{t+1}), \tag{3}$$

其中，f 是一个非线性函数，根据给定的 $\boldsymbol{e}_t$ 和 $\boldsymbol{h}_{t-1}$ 计

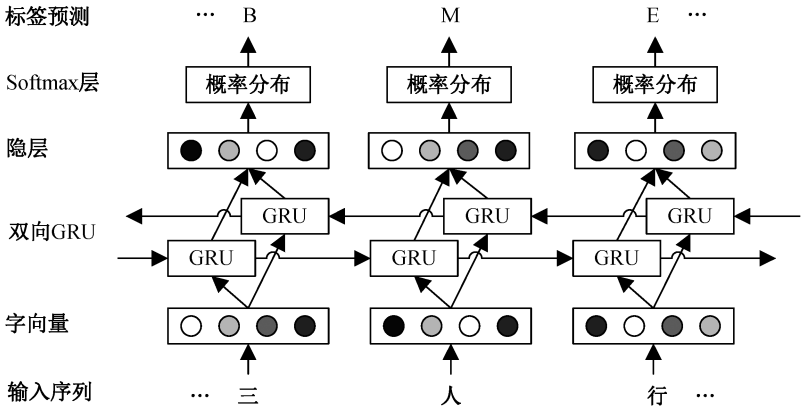

图 1　基于 GRU 的双向循环神经网络断句模型结构

Fig. 1　Architecture of bi-directional GRU-RNN model for ancient Chinese sentence segmentation

算 $\boldsymbol{h}_t$。我们采用 Cho 等[10]提出的 GRU 单元来表示 f。GRU 采用以下方式计算 $\boldsymbol{h}_t$:

$$\boldsymbol{r}_t = \text{sigmoid}(\boldsymbol{W}_{\text{r}}\boldsymbol{e}_t + \boldsymbol{U}_{\text{r}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\text{r}}), \tag{4}$$

$$\tilde{\boldsymbol{h}}_t = \tanh(\boldsymbol{W}_{\text{h}}\boldsymbol{e}_t + \boldsymbol{U}_{\text{h}}(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}) + \boldsymbol{b}_{\text{h}}), \tag{5}$$

$$\boldsymbol{z}_t = \text{sigmoid}(\boldsymbol{W}_{\text{z}}\boldsymbol{e}_t + \boldsymbol{U}_{\text{z}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\text{z}}), \tag{6}$$

$$\boldsymbol{h}_t = \boldsymbol{z}_t \odot \boldsymbol{h}_{t-1} + (1-\boldsymbol{z}_t) \odot \tilde{\boldsymbol{h}}_t, \tag{7}$$

其中，$\boldsymbol{W}_{\text{r}}$，$\boldsymbol{W}_{\text{h}}$，$\boldsymbol{W}_{\text{z}}$，$\boldsymbol{U}_{\text{r}}$，$\boldsymbol{U}_{\text{h}}$ 和 $\boldsymbol{U}_{\text{z}}$ 为权重矩阵，$\boldsymbol{b}_{\text{r}}$，$\boldsymbol{b}_{\text{h}}$ 和 $\boldsymbol{b}_{\text{z}}$ 为偏置，$\boldsymbol{r}_t$ 和 $\boldsymbol{z}_t$ 分别为“重置门”和“更新门”，$\odot$ 表示逐元素(element-wise)相乘。算法实现时，采用 hard_sigmoid 函数，近似拟合上述 sigmoid 函数，以降低模型的计算复杂度:

$$\text{hard\_sigmoid}(x) = \begin{cases} 0, & x < -2.5, \\ 0.2x + 0.5, & -2.5 \leqslant x < 2.5, \\ 1, & x \geqslant 2.5。 \end{cases} \tag{8}$$

3) 在输出层，正向隐状态 $\vec{\boldsymbol{h}}_t$ 和反向隐状态 $\overleftarrow{\boldsymbol{h}}_t$ 以 softmax 函数的形式得到输出概率:

$$\boldsymbol{o}_t^{[i]} = \frac{\exp(\boldsymbol{W}_{\vec{\text{h}}\text{o}}^{[i]}\vec{\boldsymbol{h}}_t + \boldsymbol{W}_{\overleftarrow{\text{h}}\text{o}}^{[i]}\overleftarrow{\boldsymbol{h}}_t + \boldsymbol{b}_{\text{o}})}{\sum_{j\in T}\exp(\boldsymbol{W}_{\vec{\text{h}}\text{o}}^{[j]}\vec{\boldsymbol{h}}_t + \boldsymbol{W}_{\overleftarrow{\text{h}}\text{o}}^{[j]}\overleftarrow{\boldsymbol{h}}_t + \boldsymbol{b}_{\text{o}})}, \tag{9}$$

其中，$\boldsymbol{W}_{\vec{\text{h}}\text{o}}$ 和 $\boldsymbol{W}_{\overleftarrow{\text{h}}\text{o}}$ 为权重矩阵，$\boldsymbol{W}_{\vec{\text{h}}\text{o}}^{[i]}$ 表示 $\boldsymbol{W}_{\vec{\text{h}}\text{o}}$ 的第 i 行，$\boldsymbol{W}_{\overleftarrow{\text{h}}\text{o}}^{[i]}$ 表示 $\boldsymbol{W}_{\overleftarrow{\text{h}}\text{o}}$ 的第 i 行，$\boldsymbol{b}_{\text{o}}$ 为偏置。

2.3　训练

模型的训练目标是最大化对数似然，即对于给定的一组训练样本 (x, y)，模型的训练目标是最小化如下损失函数:

$$\text{Loss}(\theta) = -\sum_i \log(\text{P}(y_i|x_i, \theta))。 \tag{10}$$

本文采用循环神经网络语言模型的形式(即仅用正向的 GRU 循环神经网络，并将输出层替换为预测输入序列中的下一个字，利用原始的训练语料(带标点的古文)，对字向量和 GRU 单元的参数进行预训练。

模型的训练过程中采用 GPU 进行并行化加速，并利用 mini-batch 梯度下降法，采用 RMSProp 算法[11]平滑梯度，并引入两种学习率动态调整策略。

1) Gradient renormalize[12]: 对梯度的模进行约束，若梯度的模大于一定的阈值(实验中取 5)，则将其限制到该阈值。引入此机制的目的在于避免模型训练初期步长过大，下降过快，从而过早收敛。

2) Early stopping[13]: 在训练过程中，若一轮迭代中，模型在验证集上的准确率没有提高，则按一定的衰减因子(实验中取 0.7)动态减小学习率。

2.4　解码

对于一个给定的序列，循环神经网络输出层输出的是各个时刻标注的条件概率(即标注集上的概率分布)。在测试时，模型需要根据各个时刻的条件概率进一步输出最终的标签序列。本文采用两种方法得到最终的标签序列，并对这两种方法的性能进行比较。

1) 贪婪法。在每一时刻均直接输出当前时刻条件概率最大的标签:

$$\hat{y}_t = \underset{j\in T}{\text{argmax}}\, \boldsymbol{o}_t^{[j]}, \tag{11}$$

其中，T 为标注集。这种方法假定输出序列中的各

个标签之间是相互独立的，由于不考虑标签间的依赖关系，因此可能会输出非法的标签序列。

2) 解码法。构造一个关于汉字序列 $x_{1:N}$ 和标签序列 $y_{1:N}$ 的目标函数 $s(x_{1:N}, y_{1:N})$。对于给定的输入序列 $x_{1:N}$，解码的过程即搜索得分最高的标签序列 $y_{1:N}$，即

$$\hat{y}_{1:N} = \arg\max_{y_{1:N}} s(x_{1:N}, y_{1:N})。 \tag{12}$$

若目标函数中仅考虑神经网络输出概率，即

$$s(x_{1:N}, y_{1:N}) = \sum_{i=1}^{N} \boldsymbol{o}_i^{[y_i]}, \tag{13}$$

则搜索过程退化为前述贪婪法。为了克服贪婪法的不足，可以在目标函数中加入标签间的状态转移概率，即

$$s(x_{1:N}, y_{1:N}) = \sum_{i=1}^{N} (\boldsymbol{o}_i^{[y_i]} + \alpha \cdot \boldsymbol{A}_{y_{i-1}y_i}), \tag{14}$$

其中，$\boldsymbol{A}_{jk}$ 表示从标签 j 到标签 k 的状态转移概率，α 为状态转移概率得分项的权重。

实验中发现，引入状态转移概率后，召回率较低。我们进一步采取与文献[7]相似的做法，在目标函数中引入句长惩罚(鼓励更精细的断句)，即

$$s(x_{1:N}, y_{1:N}) = \sum_{i=1}^{N} (\boldsymbol{o}_i^{[y_i]} + \alpha \cdot \boldsymbol{A}_{y_{i-1}y_i} + \beta \cdot C(y_{1:N})), \tag{15}$$

其中，β 为句长惩罚项的权重，$C(y_{1:N})$ 表示按 $y_{1:N}$ 断句得到的句子的数量，即

$$C(y_{1:N}) = \sum_i l(y_i), \tag{16}$$

$$l(y_i) = \begin{cases} 1, & 当\ y_i = E\ 或\ y_i = S, \\ 0, & 其他。 \end{cases} \tag{17}$$

解码过程采用 beam search 算法，解码过程中每一步保留 N-best 结果。与文献[7]不同，本文方法在解码过程中循环神经网络仅需在输入序列上迭代一次，因此计算复杂度大大降低。

3 实验

3.1 实验设置

为进行对比，本文采用与文献[7]相同的训练集、开发集和测试集。语料主要来自至善繁体汉语语料库①、汉籍电子文献资料库②(简称汉籍)和“是何年”网站③。各数据集的基本情况如表 1 所示。

我们采用精确率(*P*)、召回率(*R*)、F1 值以及准确率(*A*)作为评价模型断句效果的指标:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{18}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{19}$$

$$\mathrm{F1} = \frac{2 \cdot P \cdot R}{P + R}, \tag{20}$$

$$A = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}}, \tag{21}$$

其中，TP 为模型输出的正确的断句标记的数量，FP 为错误的断句标记数量，FN 为错误的不断句标记数量，TN 为正确的不断句标记数量。TP+ FP+ TN+ FN 等于语料中的总字数。

本文提出的断句模型采用的词表规模为 21383 字，字向量和循环神经网络隐层维度均设置为 256，初始学习率设置为 0.001，每个 batch 大小为 512 个序列。

对比实验采用以下两个基线系统。

表 1 古文断句实验数据集规模及来源

Table 1 Details of datasets for ancient Chinese sentence segmentation

数据集	规模	字表大小	内容	来源
训练集	2.37 亿字	23905	“至善语料库”中的繁体古籍部分	至善语料库
开发集	1 万字	1890	《阅微草堂笔记》(卷一)	汉籍
测试集 1	32 万字	6188	《宾退录》、《朝野佥载》、《南部新书》、《楚辞补注》和《中吴纪闻》	“是何年”网站
测试集 2	36 万字	5755	《阅微草堂笔记》(卷二至卷二十四)和《敬斋古今黈》	汉籍, “是何年”网站

① http://cloudtranslation.cc/corpus_tc.html

② http://hanchi.ihp.sinica.edu.tw/ihp/hanji.htm

③ http://www.4hn.org/

1) 文献[7]提出的基于神经网络语言模型的古文断句方法，包括 CLM6 和 CLM1 两种语言模型。

2) 文献[4]提出的基于条件随机场模型和上下文字特征的古文断句方法。我们采用开源工具 CRF++[①]重现该方法，去除频率小于 3 的特征，其余参数均采用工具的默认设置，特征模板和训练方法与文献[4]一致。

3.2　参数选择

本文提出的断句模型在输出层进行解码时包含 3 个参数：状态转移概率权重 α、长度惩罚权重 β 和解码过程中 beam 的大小。我们通过 3 组实验确定 3 个参数的取值，3 组实验均在开发集上进行。

1) 固定 beam 的大小为 50，引入状态转移概率，不引入长度惩罚，即固定 $\beta=0$，调整 α，以提高准确率。实验结果如图 2 所示，随着 α 的增大，召回率逐渐增大，精确率、准确率和 F1 值先增大后减小。我们选择使得准确率最高的 $\alpha=1.4$ 作为后续实验的默认设置。

2) 固定 beam 的大小为 50，引入状态转移概率和长度惩罚，固定 $\alpha=1.4$，调整 β，以提高 F1 值。实验结果如图 3 所示，随着 β 的增大，精确率逐渐减小，召回率逐渐增大，F1 先增大后减小。我们选择使得 F1 值最高的 $\beta=0.3$ 作为后续实验的默认设置。

3) 固定状态转移概率权重 $\alpha=1.4$ 以及长度惩罚权重 $\beta=0.3$，测试 beam 的大小对断句速度和断句效果的影响。实验结果如图 4 所示，随着 beam 的增大，断句的效果不断提高，但时间复杂度也呈近似线性增大。当 beam 大于 50 时，增大 beam 对模型断句效果的提升并不明显。因此，综合考虑断句效果和断句速度，在实际应用时设置 beam 的大小为 50。

3.3　模型对比实验

本文分别在测试集 1 和测试集 2 上进行断句性能对比实验。基线系统包括文献[7]的 CLM6, CLM6+0.65LP 和 CLM1+0.65LP，以及对文献[4]工作的重现(CRF)。实验结果如表 2 所示。GRU-RNN 为文提出的基于 GRU 的双向循环神经网络断句模型，其中 GRU-RNN 为采用贪婪法的系统，GRU-RNN+1.4ST 为状态转移概率权重取 1.4，但不进行长度惩罚的解码法系统，GRU-RNN+1.4ST+0.3LP 为状态转移概率权重取 1.4、长度惩罚权重取 0.3 的解码法系统。

从表 2 可以看出，本文提出的基于 GRU 的双向循环神经网络模型能有效地进行古文断句，引入状态转移概率和长度惩罚进行解码能有效地提高断句效果。

与已有的工作相比，本文提出的断句模型性能明显优于文献[7]所采用的基于神经网络语言模型的方法以及以文献[4]为代表的基于条件随机场模型和手工设计特征模板的方法。这说明双向循环神经网络能更有效地利用上下文信息进行断句。

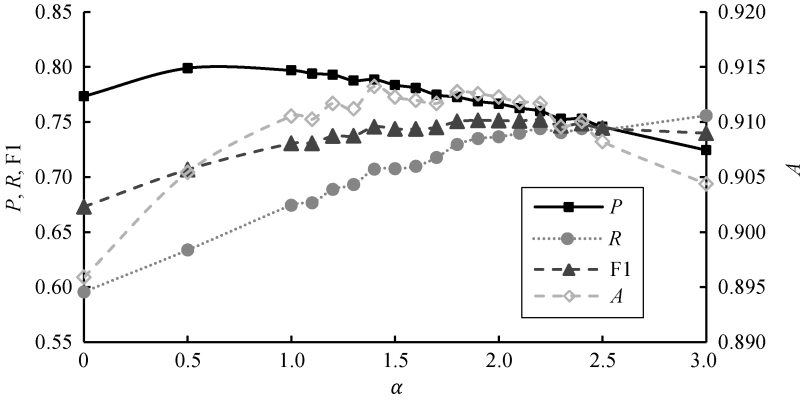

图 2　状态转移概率权重对断句效果的影响

Fig. 2　Performances of proposed methods on different weights of transition score

① http://taku910.github.io/crfpp/

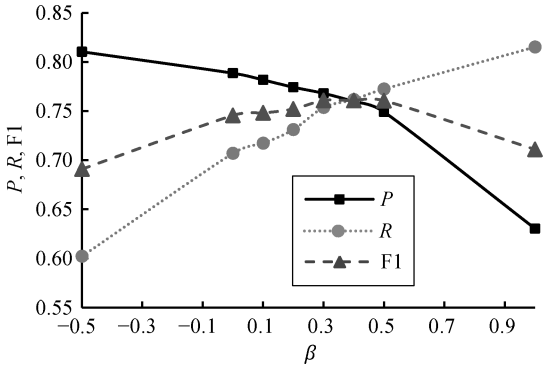

图 3 长度惩罚权重对断句效果的影响

Fig. 3 Performances of proposed methods on different weights of length penalty

4 总结与展望

本文提出一种基于 GRU 的双向循环神经网络断句模型，能有效地利用上下文信息进行古文自动断句。与传统 CRF 模型相比，本文提出的断句方法利用深度神经网络自动学习特征表示，无需人工设计特征模板。与采用神经网络语言模型的方法相比，本文的断句模型更为简洁高效，解码过程中神经网络仅需在输入序列上遍历一次，且能有效地利用双向的上下文信息。实验结果表明，本文的方法能有效地进行古文自动断句，断句效果超过采用条件随机场模型的传统方法和采用神经网络语言模型的方法。

未来工作中，我们将考虑从以下几方面进一步提高模型的断句效果。

1) 引入正则化措施或多模型融合，提高模型的鲁棒性，避免模型过度拟合。在训练目标的损失函数中引入 L1 和 L2 正则项，或在输入层、隐层和输出层分别加入 dropout 机制[14–17]。

2) 借鉴文献[18]和[19]中的方法，在循环神经

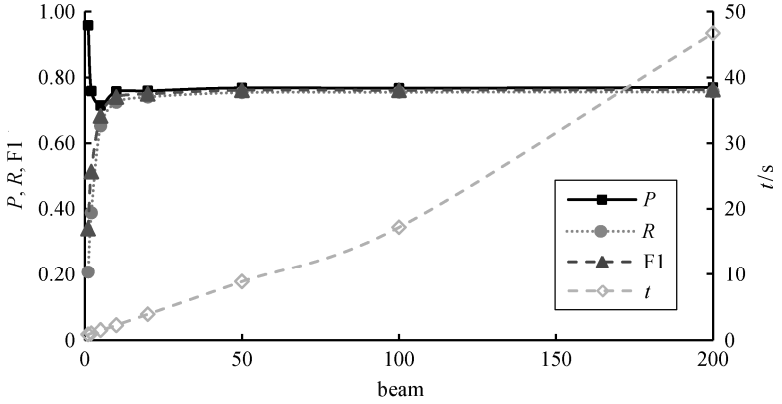

图 4 解码过程中 beam 的大小对断句速度和断句效果的影响

Fig. 4 Speed and performances of proposed methods on different size of beam in decoding

表 2 断句效果对比实验结果

Table 2 Performances of different models on two test sets

模型	测试集 1			测试集 2		
	P	*R*	F1	*P*	*R*	F1
CLM6	0.7822	0.5767	0.6639	0.7283	0.6364	0.6793
CLM6+ 0.65LP	0.7492	0.6727	0.7089	0.6861	0.7205	0.7029
CLM1+ 0.65LP	0.7212	0.7325	0.7268	0.6393	**0.7691**	0.6982
CRF	**0.8163**	0.6617	0.7309	**0.7856**	0.7100	0.7459
GRU-RNN	0.7545	0.6773	0.7138	0.7342	0.6329	0.6798
GRU-RNN+ 1.4ST (beam= 50)	0.7851	0.7202	0.7513	0.7598	0.7187	0.7387
GRU-RNN+ 1.4ST+ 0.3LP (beam= 50)	0.7606	0.7458	0.7531	0.7350	0.7533	0.7440
GRU-RNN+ 1.4ST+ 0.3LP (beam= 200)	0.7629	**0.7475**	**0.7551**	0.7381	0.7564	**0.7472**

说明: 加粗的数值为各指标中的最大值。

网络的训练目标中直接引入状态转移概率得分，将CRF模型与循环神经网络相结合，从而更有效地利用标签间的依赖关系。

3) 借鉴文献[7]的方法，训练朴素贝叶斯分类器对语料进行自动分类，分别训练不同年代不同文体的断句子模型，断句时采用自适应算法，自动选择适当的子模型进行断句。

另外，本文着眼于古文断句，若要真正方便现代人阅读和理解古文，还需要将断句标记进一步区分为不同的标点符号，这也是下一步的研究内容。

参考文献

[1] 黄建年，侯汉清．农业古籍断句标点模式研究．中文信息学报, 2008, 22(4): 31–38

[2] 陈天莹，陈蓉，潘璐璐，等．基于前后文n-gram模型的古汉语句子切分．计算机工程，2007，33(3): 192–193, 196

[3] Lafferty J D, McCallum A, Pereira F C N. Conditional random fields: probabilistic models for segmenting and labeling sequence data // Proceedings of the 18th International Conference on Machine Learning. Williamstown, 2001: 282–289

[4] 张合，王晓东，杨建宇，等．一种基于层叠CRF的古文断句与句读标记方法．计算机应用研究，2009, 26(9): 3326–3329

[5] 张开旭，夏云庆，宇航．基于条件随机场的古汉语自动断句与标点方法．清华大学学报：自然科学版，2009, 49(10): 1733–1736

[6] Huang H H, Sun C T, Chen H H. Classical Chinese sentence segmentation // Proceedings of CIPS-SIGHAN Joint Conference on Chinese Language Processing. Beijing, 2010: 1–8

[7] Wang B, Shi X, Tan Z, et al. A sentence segmentation method for ancient Chinese texts based on NNLM // Proceedings of CLSM. Singapore, 2016: 387–396

[8] Graves A. Supervised sequence labelling with recurrent neural networks. Studies in Computational Intelligence. Heidelberg: Springer, 2012: 22–29

[9] Schuster M, Paliwal K K. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 1997, 45(11): 2673–2681

[10] Cho K, Merrienboer B V, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation [EB/OL]. (2014–09–03) [2016–07–29]. http://arxiv.org/abs/1406.1078

[11] Tieleman T, Hinton G. RMSProp: divide the gradient by a running average of its recent magnitude: report of COURSERA “Neural Networks for Machine Learning”[R]. Toronto, 2012

[12] Salimans T, Kingma D P. Weight normalization: a simple reparameterization to accelerate training of deep neural networks [EB/OL]. (2016–06–04) [2016–07–29]. http://arxiv.org/abs/1602.07868

[13] Yao Y, Rosasco L, Caponnetto A. On early stopping in gradient descent learning. Constructive Approximation, 2007, 26(2): 289–315

[14] Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing coadaptation of feature detectors [EB/OL]. (2012–07–03) [2016–07–29]. http://arxiv.org/abs/1207.0580

[15] Pham V, Kermorvant C, Louradour J. Dropout improves recurrent neural networks for handwriting recognition[EB/OL]. (2014–03–10) [2016–07–29]. http://arxiv.org/abs/1312.4569

[16] Moon T, Choi H, Lee H, et al. RNNDROP: a novel dropout for RNNS in ASR // IEEE Automatic Speech Recognition and Understanding Workshop. Scottsdale, 2015: 65–70

[17] Semeniuta S, Severyn A, Barth E. Recurrent dropout without memory loss [EB/OL]. (2016–03–16) [2016–07–29]. http://arxiv.org/abs/1603.05118

[18] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging [EB/OL]. (2015–08–09) [2016–07–29]. http://arxiv.org/abs/1508.01991

[19] Chen X, Qiu X, Zhu C, et al. Long short-term memory neural networks for Chinese word segmentation // Proceedings of EMNLP. Lisbon, 2015: 1197–1206