智能系统学报
CAAI TRANSACTIONS ON INTELLIGENT SYSTEMS

基于局部Transformer的泰语分词和词性标注联合模型

朱叶芬, 线岩团, 余正涛, 相艳

引用本文:
朱叶芬,线岩团,余正涛,相艳. 基于局部Transformer的泰语分词和词性标注联合模型[J]. 智能系统学报, 2024, 19(2): 401–410.
ZHU Yefen, XIAN Yantuan, YU Zhengtao, et al. Joint model for Thai word segmentation and part-of-speech tagging via a local Transformer[J]. *CAAI Transactions on Intelligent Systems*, 2024, 19(2): 401–410.

在线阅读 View online: https://dx.doi.org/10.11992/tis.202209034

DOI: 10.11992/tis.202209034
网络出版地址: https://link.cnki.net/urlid/23.1538.tp.20231115.1524.008

基于局部 Transformer 的泰语分词和词性标注联合模型

朱叶芬[1,2], 线岩团[1,2], 余正涛[1,2], 相艳[1,2]
（1. 昆明理工大学 信息工程与自动化学院, 云南 昆明 650500; 2. 昆明理工大学 云南省人工智能重点实验室, 云南 昆明 650500）

摘　要: 泰语分词和词性标注任务二者之间存在高关联性, 已有研究表明将分词和词性标注任务进行联合学习可以有效提升模型性能, 为此, 提出了一种针对泰语拼写和构词特点的分词和词性标注联合模型。针对泰语中字符构成音节, 音节组成词语的特点, 采用局部 Transformer 网络从音节序列中学习分词特征; 考虑到词根和词缀等音节与词性的关联, 将用于分词的音节特征融入词语序列特征, 缓解未知词的词性标注特征缺失问题。在此基础上, 模型采用线性分类层预测分词标签, 采用线性条件随机场建模词性序列的依赖关系。在泰语数据集 LST20 上的试验结果表明, 模型分词 F_1、词性标注微平均 F_1 和宏平均 F_1 分别达到 96.33%、97.06% 和 85.98%, 相较基线模型分别提升了 0.33%、0.44% 和 0.12%。

关键词: 泰语分词; 词性标注; 联合学习; 局部 Transformer; 构词特点; 音节特征; 线性条件随机场; 联合模型
中图分类号: TP391　　**文献标志码:** A　　**文章编号:** 1673−4785(2024)02−0401−10

中文引用格式: 朱叶芬, 线岩团, 余正涛, 等. 基于局部 Transformer 的泰语分词和词性标注联合模型 [J]. 智能系统学报, 2024, 19(2): 401–410.
英文引用格式: ZHU Yefen, XIAN Yantuan, YU Zhengtao, et al. Joint model for Thai word segmentation and part-of-speech tagging via a local Transformer[J]. CAAI transactions on intelligent systems, 2024, 19(2): 401–410.

Joint model for Thai word segmentation and part-of-speech tagging via a local Transformer

ZHU Yefen[1,2], XIAN Yantuan[1,2], YU Zhengtao[1,2], XIANG Yan[1,2]
(1. School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China; 2. Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming 650500, China)

Abstract: There is a high correlation between Thai word segmentation (WS) and part-of-speech (POS) tagging tasks, and it has been demonstrated that joint learning of WS and POS tagging tasks can effectively enhance model performance. Herein, we propose a novel joint model for Thai WS and POS, including Thai spelling rules and sub-word features. A local Transformer network is employed to learn WS features from windowed syllable sequences. Considering the relationship between syllables, such as roots, affixes, and POS, the syllable features used for WS are integrated into the characteristics of word sequence to alleviate the lack of POS tagging features for out-of-vocabulary words. Moreover, we utilize a linear classification layer to forecast the label of WS and a linear conditional random field to model the label dependencies of POS sequences. Experimental findings for the Thai LST20 dataset reveal that the proposed method has a WS F_1 value, POS tagging microF_1 value, and macro F_1 value of 96.33%, 97.06%, and 85.98%, respectively, which are enhanced by 0.33%, 0.44%, and 0.12%, with respect to the baselines.

Keywords: Thai word segmentation; part-of-speech tagging; joint learning; local Transformer; sub-word features; syllable features; linear conditional random field; joint model

泰语分词和词性标注是自然语言处理中的基础性关键任务, 是许多泰语后续自然语言处理任务的必要处理步骤。在以往的研究中, 通常将泰

收稿日期: 2022−09−16.　**网络出版日期:** 2023−11−16.
基金项目: 国家自然科学基金项目（62266028）; 云南省重大科技专项计划（202002AD080001）.
通信作者: 线岩团. E-mail: xianyt@kust.edu.cn.

语分词和词性标注任务建模为使用不同的序列编码器的序列标记任务。

在泰语分词任务中，常用模型包括 Sertis[1]、DeepCut[2] 和 AttaCut[3]。其中 Sertis 泰语分词模型以双向循环神经网络（bidirectional recurrent neural network）[4] 作为编码器来捕获序列中的上下文特征，再使用 Softmax 函数解码得到分词标签。DeepCut 和 AttaCut 2 个泰语分词模型都是采用滑动窗口的方式将序列输入 CNNs[5] 序列编码器，经过 Softmax 函数解码得到预测的分词序列。其中 DeepCut 采用多个具有可变核宽度的卷积进行特征学习，是目前性能最好的泰语分词器，但存在分词速度慢的问题。相比于 DeepCut，AttaCut 使用 3 个空洞卷积，提升了模型的分词速度，但分词精度有所下降。同时该模型以音节作为输入，验证了音节可以为分词提供重要的学习特征。

在泰语词性标注任务中，主流模型通常是采用 BiLSTM[6] (bidirectional long short-term memory) 或预训练语言模型[7] 获取词性标注上下文特征，再使用 CRF[8](conditional random field) 作为解码器进行词性标签预测。基于 BiLSTM 的方法可以有效缓解模型对特征工程的依赖，实现不同单词在时序维度上的信息传递。但该方法存在长距离依赖问题，并且从词性标注的角度来讲，过长的序列对于单词词性的预测意义不大。预训练语言模型是先通过一批语料进行模型训练，然后在这个初步训练好的模型基础上，再继续训练。2018 年，首次提出了 BERT[7] (bidirectional encoder representations from transformers) 预训练语言模型，相较于原来的 RNN、LSTM 可以做到并发执行，同时提取词在句子中的关系特征，并且能在多个不同层次提取关系特征，进而更全面反映句子语义。在此基础上，Liu 等[9] 提出了 RoBERTa 预训练语言模型，相比 BERT，该模型使用了更大的数据集进行训练，对模型进行了更多次迭代。近几年，也出现了一些融入更多知识的预训练语言模型，如 BROS[10]、DKPLM[11] 等。2021 年，Lowphansiorikul 等[12] 提出了一个基于 RoBERTa-base 的预训练语言模型 Wangchan-BERTa，该模型使用大型的泰语数据集进行预训练，获取高质量的词向量，在泰语词性标注任务中实现了当前最优性能，但是这类大规模预训练语言模型会明显增加模型的参数量，降低模型的预测速度。

正确的分词是词性标注的基础，词性信息可以给分词提供有用的特征，这 2 个任务之间存在很高的关联性。实现分词和词性标注任务的方案，传统的策略为管道模型[13-14]，先分词，再进行词性标注。这类方法会引入任务间的错误传递，并且每个模型只针对一个任务，不能充分利用任务之间的共享知识。而联合模型将分词和词性标注任务同时进行，可以有效改善错误传递的问题，实现参数共享。已有的联合任务模型包括传统的机器学习方法[15-18] 和基于深度学习的方法[19-22]，传统的机器学习方法通常需要人工构造特征抽取函数，过程较为麻烦。深度学习方法可以避免人工构建提取特征的过程，具有强大的特征提取能力和非线性拟合能力，在降低人工成本的同时提升了性能。Tian 等[19] 提出了一种中文分词和词性标注联合模型，采用双向注意机制来结合每个输入字符的上下文特征及其对应的句法知识。Buoy 等[20] 提出了一种字符级的联合高棉语分词和词性标注的 BiLSTM 网络，将字符通过 BiLSTM 进行编码，使用 Softmax 函数进行解码。该模型可以很好地建模上下文信息，而无法获取单词的局部信息。

但是，无论是哪种方法都仅侧重于将上下文信息融入联合任务中，而未充分考虑到句子的局部信息。并且由于泰语特定的构词特点和内部结构信息，将通用的联合模型或其他语言的联合模型直接应用到泰语分词和词性标注联合任务上会导致形态学信息丢失。

音节是泰语的基本构成单位，泰语单词是按照其特定的拼写规则由音节组成。在泰语构词中广泛使用附加（包括前缀、后缀）、插入和重叠等构词方式，其中附加式主要依靠词缀加词根来构词，因此词缀对词性有很好的指示作用。例如，前缀“นก”+动词词根“พด（说）”变成名词“นกพด（演讲家）”，前缀“เครื่อง”+动词词根“บิน（飞）”变成名词“เครื่องบิน（飞机）”。在泰语文本中，以音节构成的词根和词缀有助于判断词边界，并且可以对词性标注起到指示作用。

通过以往的模型[23-26] 发现，通过滑动窗口的方式建模局部特征可以得到很好的分词性能。但 CNN 存在解码速度慢的问题，空洞 CNN 虽然提升了模型运行速度，但会导致模型性能下降。BiLSTM 的遗忘和记忆机制可以很好建模序列之间的关系，但是存在长距离依赖问题，并且计算很难并行化。因此，本研究提出了一个适用于泰语分词和词性标注任务的联合模型。该模型将音节作为分词任务的输入特征，将音节作为词语的

形态学特征融入词性标注任务中，采用局部 Transformer[27] 网络进行局部信息建模。相比已有模型，该模型并行度更高，并且使用音节特征有助于学习未知词的上下文特征，缓解未知词错误标注对模型性能的影响。本研究模型在泰语分词和词性标注任务上都获得了最优性能。

1　分词和词性标注联合模型

本研究提出了一个泰语分词和词性标注的联合模型，该模型结构如图 1 所示。

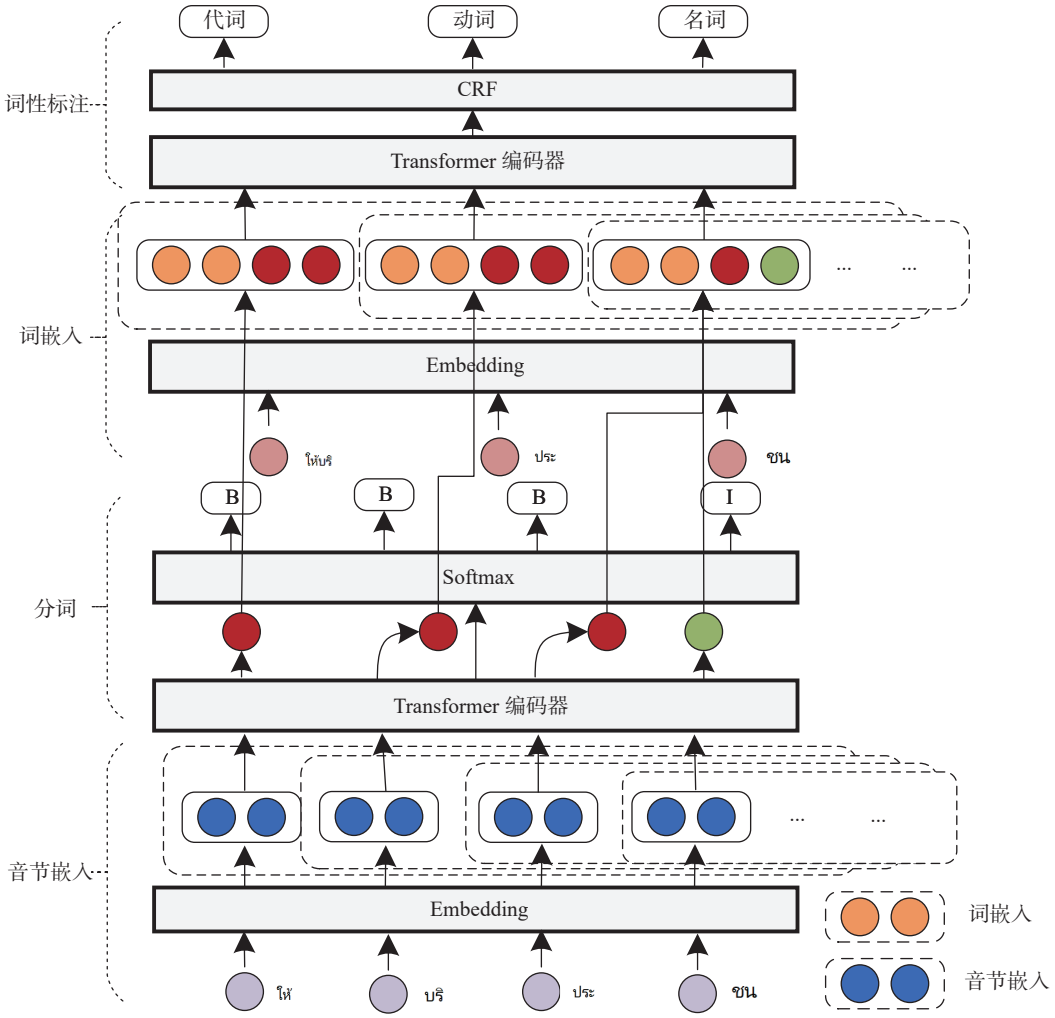

图 1　模型结构

Fig. 1　Structure of the model

在分词模块，该模型使用滑动窗口方式截取 j 个（该方法采用的值为 11）音节特征，将预处理后的窗口化音节序列输入局部 Transformer 编码器学习上下文特征，经过线性层得到标签得分，再使用 Softmax 函数去预测音节的分词标签 B、I。在词性标注模块，将分词模型中得到的音节特征，取每个单词的前缀音节特征和后缀音节特征进行拼接，单音节单词则将单音节特征叠加拼接后，再与单词特征拼接表示作为模型输入，与分词任务类似，使用局部 Transformer 作为编码器去建模局部特征，经过一个线性层得到词性标签得分，使用 CRF 作为解码器建模词性之间的依赖关系，预测每个单词的词性标签。

1.1　音节编码

本研究采用 SSG[3]（syllable segmenter）算法对句子中的泰语单词进行音节切分。其来自 PyThaiNLP[28] 第三方库，是一个专门的泰语音节切分器。假设输入句子由 m 个音节组成。将预处理后的音节通过滑动窗口的方式获得音节表征，表示为

$$\boldsymbol{e}_s = [e_{s_1} \quad e_{s_2} \quad \cdots \quad e_{s_i} \quad \cdots \quad e_{s_m}] \tag{1}$$

式中：$\boldsymbol{e}_s \in \mathbf{R}^{m \times j \times d_1}$，$m$和$d_1$分别表示输入音节序列向量的长度和维度，$j$为滑动窗口宽度。

1.2　局部 Transformer 分词

本研究模型的编码层使用 Transformer 作为编码器，采用多头自我注意力层进行局部信息建

模，获取每一个音节的上下文特征。每个头注意力都有一个不同的线性变换应用于相同的输入表示。多头自我注意力层和全连接层组成了 Transformer 编码器，编码器的输出表示为 $\boldsymbol{h}_s$

$$\boldsymbol{h}_s = \text{Transformer}(\boldsymbol{e}_s) \tag{2}$$

其中，$\boldsymbol{h}_s \in \mathbf{R}^{m\times j\times d_1}$。

使用一个线性层得到序列中每个窗口的中心音节的特征

$$\tilde{\boldsymbol{h}}_s = \text{Linear}(\boldsymbol{h}_s) \tag{3}$$

其中，$\tilde{\boldsymbol{h}}_s \in \mathbf{R}^{m\times l_1}$，$l_1$为分词的标签个数。

在解码的过程中，本方法使用 Softmax 激活函数，用于预测序列中每个字符的标识符为 (1, 0) 的概率p_1。

$$p_1 = \text{Softmax}(\boldsymbol{W}_1 \times \tilde{\boldsymbol{h}}_s + \boldsymbol{b}_1) \tag{4}$$

式中：$\boldsymbol{W}_1$是可学习的权重；$\boldsymbol{b}_1$是偏置参数。

1.3　局部 Transformer 词性标注

假设句子由n个单词组成，输入词性标注模块前，将句子的单词使用滑动窗口获取词嵌入，表示为

$$\boldsymbol{e}_w = [e_{w_1} \quad e_{w_2} \quad \cdots \quad e_{w_i} \quad \cdots \quad e_{w_n}] \tag{5}$$

式中：$\boldsymbol{e}_s \in \mathbf{R}^{n\times j\times d_2}$，$n$和$d_2$分别表示输入词序列向量的长度和维度，$d_2 = 2\times d_1$。

将其与分词模块编码器输出的单词前后缀音节特征进行拼接作为编码器的输入。若单词w_i为单音节单词，把单音节同时当作前后缀音节进行拼接，其单词的拼接特征t_i表示为

$$\boldsymbol{h}_c = [\boldsymbol{h}_j \,\|\, \boldsymbol{h}_{j'}] \tag{6}$$

$$\boldsymbol{t}_i = [\boldsymbol{e}_{w_i} \,\|\, \boldsymbol{h}_c] \tag{7}$$

式中：$\boldsymbol{h}_c \in \mathbf{R}^{n\times j\times 2d_1}$；$\boldsymbol{h}_j$和$\boldsymbol{h}_{j'}$分别为单词$w_i$的前缀音节特征和后缀音节特征，$\boldsymbol{h}_j \in \mathbf{R}^{n\times j\times d_1}$，$\boldsymbol{h}_{j'} \in \mathbf{R}^{n\times j\times d_1}$。

词性标注模块编码层同样是使用 Transformer 的编码器，$\boldsymbol{t} = [t_1 \quad t_2 \quad \cdots \quad t_i \quad \cdots \quad t_n]$为句子的拼接特征。

$$\boldsymbol{h}_w = \text{Transformer}(\boldsymbol{t}) \tag{8}$$

其中，$\boldsymbol{h}_w \in \mathbf{R}^{m\times j\times d_2}$。

$$\tilde{\boldsymbol{h}}_w = \text{Linear}(\boldsymbol{h}_w) \tag{9}$$

其中，$\tilde{\boldsymbol{h}}_w \in \mathbf{R}^{m\times l_2}$，$l_2$为词性的标签个数。

在词性标注模块的解码过程中，采用 CRF 作为解码器。本方法使用 Viterbi 算法去获得标签序列的最高得分。形式上，本方法使用$\boldsymbol{x} = [x_1 \ x_2 \ \cdots \ x_n]$来表示输入序列，使用$\boldsymbol{y} = [y_1 \ y_2 \ \cdots \ y_n]$来表示模型输出标签，本研究定义其分数为

$$\boldsymbol{s}(\boldsymbol{x}_{0:n}, \boldsymbol{y}_{0:n}, \theta) = \sum_{i=0}^{n} \boldsymbol{A}_{y_i, y_{i+1}} + \sum_{i=0}^{n} \boldsymbol{P}_{y_i, y_{i+1}} \tag{10}$$

式中：$\boldsymbol{A}$是一个转移分数矩阵；y_0和y_n是句子的开头和结尾的标签；$\boldsymbol{P}$是 Transformer 编码器输出的分数矩阵；θ是神经网络设置的参数。

对所有可能的标签序列执行 Softmax 操作，计算序列$\boldsymbol{y}$的概率，公式为

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{\mathrm{e}^{s(\boldsymbol{x},\boldsymbol{y},\theta)}}{\sum\limits_{\tilde{\boldsymbol{y}}\in\boldsymbol{y}_x}} \tag{11}$$

其中$\boldsymbol{y}_x$表示$\boldsymbol{x}$可能的标签序列的集合。

在解码时，通过获取最大的得分$\boldsymbol{p}_w$来预测词性标注模块的输出序列。

$$\boldsymbol{p}_w = \arg\max_{\boldsymbol{p}_w\in\boldsymbol{y}_x} s(\boldsymbol{x}, \tilde{\boldsymbol{y}}, \theta) \tag{12}$$

在解码的过程中，使用极大似然估计法训练模型：

$$\ln(p(\boldsymbol{y}|\boldsymbol{x})) = s(\boldsymbol{x}, \tilde{\boldsymbol{y}}, \theta) - \ln\left(\sum_{\tilde{\boldsymbol{y}}\in\boldsymbol{y}_x} \mathrm{e}^{s(\boldsymbol{x},\tilde{\boldsymbol{y}},\theta)}\right) \tag{13}$$

1.4　损失函数

该模型的损失通过计算泰语分词和词性标注任务的损失加权和得到，使用最小化损失和的方法学习模型的参数。本研究模型中，分词的损失函数L_{WS}选择的是二值交叉熵（binary cross entropy），词性标注的损失函数L_{POS}为 CRF 的损失函数。模型的最终损失表示为

$$L = \lambda L_{\text{WS}} + L_{\text{POS}} \tag{14}$$

其中，λ是可调节参数。

2　试验

2.1　试验数据集及数据预处理

本研究使用由泰国国家电子和计算机技术中心（National Electronics and Computer Technology Center, NECTEC）标注的泰语数据集 LST20 [29] 来评估本研究的模型。LST20 包含来自 15 个类别的 4751 个文档中的 74180 个句子，该语料的训练集有 3794 个文档，验证集有 474 个文档，测试集包含 483 个文档。该数据集包含句子级和单词级信息，表 1 显示了 LST20 数据集的具体统计信息。

表 1　LST20 数据集统计

Table 1　Statistics for the LST20 dataset

数量	训练集	验证集	测试集
句子数	63 310	5 620	5 250
单词数	2 714 726	240 860	207 278
音节数	3 617 618	335 868	276 646

在试验中，音节级信息通过使用泰语音节切分器 SSG 得到。表 2 给出了 LST20 数据集的单词音节数信息。由表 2 发现，单双音节的单词占 90% 以上。因此，在词性标注的特征拼接中，仅使用前后缀的音节信息。若词语为单音节单词，则把单音节同时作为前后缀音节进行拼接。

表 2　LST20 数据集的音节个数统计

Table 2　Count of syllables in LST20 dataset　　个·%$^{-1}$

单词音节数	训练集	验证集	测试集
1	2012547 / 74.13	174854 / 72.60	152958 / 73.79
2	558432 / 20.58	49032 / 20.36	43346 / 20.91
3	107903 / 3.97	10817 / 4.49	8169 / 3.94
4	25528 / 0.94	3280 / 1.36	2091 / 1.01
>5	10335 / 0.38	2874 / 1.19	714 / 0.35

2.2　评价指标

分词任务采用词语精确率 P(Presion)、召回率 R(Recall)和 F_1 作为性能评价指标。词性标注任务采用宏平均(Macro-average)F_1 和微平均(Micro-average)F_1 作为性能评价指标。

2.3　试验软硬件环境

本试验使用 Pytorch 框架实现模型。训练一个 epoch 约 30 min, 采用早停机制(Early Stopping)获取最优模型, 当验证集上分词任务的 F_1 在连续 5 个 epoch 没有优化时停止训练, 约 40 个 epoch 后模型训练完成。总训练时间约 20 h。软硬件环境如表 3 所示。

表 3　试验软硬件环境

Table 3　Experimental software and hardware environment

项目	环境
GPU	NVIDIA GeForce RTX 2080 Ti
内存/GB	12
硬盘/TB	2
系统	Ubuntu16.04 LTS
Python版本	Python 3.8
Pytorch版本	1.10.1

2.4　试验设置

本模型选择 Adam[30] 作为优化器, 其参数α、β_1和β_2均使用默认设置1×10^{-3}、0.9 和 0.999。在单词嵌入和音节嵌入层上均使用 Dropout 策略, 缓解模型过拟合。模型具体超参数设置见表 4。

表 4　超参数设置

Table 4　Hyperparameter setting

参数	数值
音节向量维度	64
词向量维度	128
编码器层数	2
注意力头数	12
Dropout	0.15
学习率	0.001

3　试验结果与分析

3.1　对比试验

本节通过设计试验比较泰语数据集 LST20 下, 不同的神经网络模型在泰语分词和词性标注任务上的性能。

在分词方面, 本研究选择多个不同类型的方法作为基线模型:

DeepCut[2] 以字符及字符类别作为输入, 使用多个 CNN 卷积作为特征编码器, 使用 Softmax 实现词切分预测。

AttaCut-C[3] 是另一种基于 CNN 的泰语分词模型, 使用空洞 CNN 作为特征编码器, 使用 Softmax 进行分词标签预测。

AttaCut-SC[3] 是 AttaCut-C 的一个变体模型, 唯一不同的是它使用了音节嵌入作为额外的特征输入。

本模型在泰语数据集 LST20 上分词任务对比试验结果, 如表 5-6 所示。

表 5　LST20 数据集上不同的分词模型性能

Table 5　Performance of different WS models on LST20 dataset　　%

模型	P	R	F_1
DeepCut[2]	96.00	96.00	96.00
AttaCut-C[3]	96.68	90.51	90.49
AttaCut-SC[4]	**97.47**	91.29	94.28
本文模型	96.09	**96.58**	**96.33**

表 6　不同方法上的词切分 IV 和 OOV 的召回率

Table 6　Recall of word cut IV and OOV on different methods　　%

模型	R_{IV}	R_{OOV}
DeepCut[2]	98.17	64.12
AttaCut-C[3]	97.88	58.73
AttaCut-SC[3]	97.69	66.98
本文模型	**98.61**	**68.57**

由表 5 的结果可以看出, 本研究提出的模型在 LST20 数据集上获得了最优的分词 F_1。本研究模型 P、R、F_1 均高于 DeepCut, 说明了使用局部 Transformer 进行特征学习效果优于传统的 CNN 卷积, 有利于提高分词任务的性能。本模型和 Attacut-SC 都使用了音节信息, 研究发现本研究模型精确率 P 虽然略低于 Attacut-SC, 但召回率 R 更高, 因此本研究模型 F_1 仍明显高于 Attacut-SC。其中, 性能最差是 Attacut-C。与 DeepCut 相比, 表明了传统的 CNN 卷积优于空洞

CNN, 但其参数规模更大。与 Attacut-SC 相比, 说明了融合音节嵌入的有效性。结果表明, 在模型中使用音节作为输入特征, 使用局部 Transformer 作为编码器进行联合学习, 效果优于其他对比模型。

由表 6 可看出, 本模型的登录词召回率 R_{IV} 和未登录词召回率 R_{OOV} 都高于基线模型。表明模型可以从数据集中学习到有效的分词特征, 同时, 更高的 $\mathrm{R_{OOV}}$, 表明本研究模型具有更好的泛化能力。

在词性标注方面, 本研究选择多个不同类型的方法作为基线模型:

CRF[9] 模型是通过在 3 个时间步的滑动窗口中提取单元图、双元图和三元图特征, 使用 Viterbi 算法去找到单词序列对应的标签序列的最佳路径, 得到最终的词性标签。

XLMR[29] 模型和 mBERT[7] 模型是 2 个多语言 BERT 预训练模型作为编码器的词性标注方法。

WangchanBERTa[10] 是一系列以 RoBERTa-base 为基础的泰语预训练语言模型。本研究选择以音节作为输入的 WangchanBERTa-syllable 和以 SentencePiece 切分为输入的 WangchanBERTa-uncased 模型开展对比试验。原因是 WangchanBERTa-syllable 和本模型都以音节作为输入, 而 WangchanBERTa-uncased 是词性标注性能最优的模型。

针对 LST20 数据集的词性标注对比试验结果见表 7。由表 7 中可以看出, 相比其他模型, 本研究提出的模型在词性标注任务中的性能达到最优。基于 CRF 的方法总体性能最低, 说明相比于统计机器学习方法, 深度学习方法可以更好地捕获特征信息。相比于 XLMR 和 mBERT 这 2 个多语言预训练语言模型, 本模型结合了泰语的构词特点, 利用分词模块得到的音节隐向量特征进行特征共享和信息交互, 使得在词性标注的性能上有所提升。与基于 WangchanBERTa 的模型相比, 本模型仍能获得更高的微平均 F_1 和宏平均 F_1。相比于 WangchanBERTa-syllable 模型, 本模型将词和音节的特征结合可以更好地进行词性判断。相比于 WangchanBERTa-uncased 模型, 说明本研究提出的局部 Transformer 相较于基于 RoBERTa-base 预训练语言模型能够更高效地处理输入特征的信息交互。并且相较于基于预训练语言模型的方法, 该模型结构更简单, 参数规模更小。

表 7　LST20 数据集上不同的词性标注模型性能

Table 7　Performance of different POS models on LST20 dataset　%

模型	微平均F_1	宏平均F_1
CRF[9]	96.28	81.28
XLMR[29]	96.57	85.00
mBERT[7]	96.44	85.86
WangchanBERTa-syllable[10]	96.36	85.24
WangchanBERTa-uncased[10]	96.62	85.44
本模型	**97.06**	**85.98**

3.2　消融试验

为了验证分词和词性标注联合学习对模型性能的影响, 本节从多任务联合角度开展消融试验。表 8 的“WS”表示单独的分词模型, “POS”表示单独的词性标注模型, 且未融合音节特征。

表 8　消融试验

Table 8　Ablation experiments　%

模型	分词	词性标注	
	F_1	微平均F_1	宏平均F_1
WS	95.92	—	—
POS	—	96.82	84.47
本文模型	**96.33**	**97.06**	**85.98**

由表 8 可以看出, 本模型通过对分词和词性标注任务的联合学习使 2 个任务的各项指标都得到了不同程度的提升。通过多任务联合学习方法, 将有用的信息特征进行共享, 从而提高了各任务的性能。值得注意的是, 针对词性标注任务来看, 由于泰语的音节级信息中隐含了一些词缀特征, 因此将分词任务中得到的音节特征引入词性标注任务中可以获得更好的性能。

3.3　超参数对模型的影响

为了验证模型不同参数对泰语分词和词性标注性能任务的影响, 本研究设置了不同的参数, 在 LST20 数据集上进行试验。表 9-表 13 给出了在不同参数设置下分词和词性标注的性能。

表 9　不同编码器层数对模型性能的影响

Table 9　Effect of different encoder layers on model performance　%

任务	评价指标	编码器层数		
		1	2	3
分词	F_1	96.08	**96.77**	95.90
词性标注	微平均F_1	96.83	**97.19**	96.01
	宏平均F_1	85.50	**86.00**	84.82

表 10　不同注意力头数对模型性能的影响

Table 10　Effect of different number of attention heads on model performance　%

任务	评价指标	注意力头数			
		8	10	12	16
分词	F_1	96.74	96.75	**96.77**	96.75
词性标注	微平均F_1	97.17	97.18	**97.19**	97.01
	宏平均F_1	86.00	**86.03**	86.00	85.21

表 11　不同损失占比对模型性能的影响

Table 11　Effect of different loss ratio on model performance　%

任务	评价指标	损失占比λ					
		0.4	0.6	0.8	1.0	1.2	1.4
分词	F_1	95.83	95.99	96.02	**96.77**	96.32	96.01
词性标注	微平均F_1	96.92	96.82	96.88	**97.19**	96.33	96.21
	宏平均F_1	85.10	95.80	**86.01**	86.00	85.37	85.33

表 12　不同滑动窗口宽度对模型性能的影响

Table 12　Effect of different sliding window widths on model performance　%

任务	评价指标	滑动窗口宽度				
		7	9	11	13	15
分词	F_1	96.04	96.16	**96.77**	96.19	96.00
词性标注	微平均F_1	96.89	96.93	**97.19**	96.37	96.36
	宏平均F_1	84.77	85.38	**86.00**	84.12	84.01

表 13　不同音节/词向量维度对模型性能的影响

Table 13　Effect of different syllable hidden size on model performance　%

任务	评价指标	音节 / 词向量维度				
		16 / 32	32 / 64	64 / 128	128 / 256	256 / 512
分词	F_1	95.03	96.62	**96.77**	96.75	96.55
词性标注	微平均F_1	96.21	96.78	**97.19**	96.91	96.91
	宏平均F_1	94.02	85.67	**86.00**	85.72	85.42

表 9 显示了不同的编码器层数对模型性能的影响。试验结果表明, 浅层模型获得了更好的性能, 这可能是因为浅层促进了不同特征之间的交互, 作者将在未来的工作中继续这方面的探索。从试验结果来看, 增加层数并不能提高模型的性能, 在层数达到一定程度后, 性能就会下降。当编码器层数为 3 层时, 模型的性能有所下降。观察并推测其可能的原因是随着模型层数的增加, 模型的参数增加, 在训练过程中, 损失迅速减小, 准确率很快达到 1。然而, 在调整模型参数时, 验证集上的数据波动较小, 推测其原因可能是这个模型结构较为复杂, 过拟合较严重, 最后导致其性能较差, 即在模型中使用多层结构会降低模型的性能。

表 10 显示了不同的注意力头数对模型性能的影响。结果表明, 随着注意力头数的增加, 试验结果的各项指标也在增加, 这表明多个注意力头可以关注到不同子空间中的信息, 获取更丰富的特征信息。但注意力头数大于 12 时, 各项评价指标开始下降, 说明注意力头数超过一定阈值, 可能造成累计误差变大, 导致模型的表现变差。因此本研究采用的是 12 个注意力头。

在训练联合任务的时候, 当不同任务的训练难易程度不同时, 会导致不同任务的损失差异较大。因此本研究基于泰语分词任务所占的不同的损失权重值进行了试验, 试验结果如表 11 所示。

由表 11 的试验结果可以看出, $\lambda = 0.8$时, 模型词性标注宏平均 F_1 最高, 但其他性能均比 $\lambda = 1.0$时稍低。当$\lambda > 1.0$时, 各项性能也明显下降。综合而言, 作者认为$\lambda = 1.0$时, 模型的总体性

能最优。说明作为联合任务，分词和词性标注权重相当时，性能优化得更快，能够更好地协助模型找到一个更强大，更具有鲁棒性的特征表示，最终使分词和词性标注总体性能更优。

由表 12 中可以看出，当滑动窗口宽度为 11 时，分词和词性标注任务性能均达到最优。当滑动窗口宽度较少时，句子的语义特征学习不够丰富。当增大滑动窗口宽度时，模型的性能有所提高，其原因在于局部注意力机制可以学习序列内部词之间的依赖关系，捕获句子内部的语义结构信息。直到当滑动窗口宽度达到 11 以后，随着滑动窗口宽度的增大，分词和词性标注的各项评估指标都在减少。这主要是句子中的分词和词性判断通常存在短距离依赖，词性一般和其左右几个词有关；若句子长度越长，语义越复杂，词语的分词和词性识别会更加困难，所以再增大滑动窗口宽度对泰语分词和词性标注任务的性能的提升作用不大。

表 13 显示了不同的音节/词向量维度对模型性能的影响。由式(5)−式(7)可知，词向量维度是音节向量维度的两倍。从表 13 中发现，当音节/词向量维度为 64/128 的整体性能最好。向量维度太小，往往会导致特征捕获不足的情况，而忽略部分可用信息。随着向量维度的增加，当向量维度为 64//128 和 128/256 时，模型的分词和词性标注任务均达到最优。当向量维度太大时，模型的性能开始下降，模型出现过拟合现象。考虑到总体模型性能，将音节/词向量维度设置为 64/128。

3.4　实例分析

为了进一步验证本研究提出的泰语分词和词性标注联合模型的有效性，在泰语数据集 LST20 中选取若干数据进行实例分析，单独模型“WS”“POS”和联合模型 (本研究模型) 的标注结果如图 2 所示。

分词	例 1：	ซัลโวปลิดชีพทายาทหมื่นล.จินตนาพลาซ่า														
	标注结果：	เซ้ลโว	ปลิด	ชีพ	ทายาท	หมื่น	ล.	จินตนาพลาซ่า								
	WS	เซ้ลโว	ปลิด	ชีพ	ทายาท	หมื่น	ล.	<u>จินตนา</u>	<u>พลาซ่า</u>							
	本文模型	เซ้ลโว	ปลิด	ชีพ	ทายาท	หมื่น	ล.	จินตนาพลาซ่า								
词性标注	例 2：	ตราบจนกว่า	นายก	ฯ	_	จะ	เบื่อ	_	ตอน	นี้	_	25	_	ก็	ครบ	เปรี๊ยะ
	标注结果：	CC	NN	PU	PU	AX	VV	PU	NN	AJ	PU	NU	PU	CC	VV	IJ
	POS	CC	NN	PU	PU	AX	VV	PU	NN	AJ	PU	NU	PU	CC	VV	<u>NN</u>
	本文模型	CC	NN	PU	PU	AX	VV	PU	NN	AJ	PU	NU	PU	CC	VV	IJ

注：加下划线的词和词性是预测错误的结果。

图 2　实例分析

Fig. 2　Case study

由图 2 可以看出，本研究提出的联合模型的分词和词性标注预测结果更加准确。针对分词任务，如例 1 所示，“จินตนาพลาซ่า”意为“金塔纳广场”，是一个四音节的单词，单独的分词模型对于这种多音节单词容易出现分词错误的情况，本研究提出的联合模型通过任务间数据共享可以更准确地切分多音节单词。IJ 为感叹词词性标签，是 LST20 数据集中的低频词性。针对词性标注任务，如例 2 所示，“เปรี๊ยะ”意为“太好了”，是个感叹词。单独的词性标记模型未能正确识别该词词性，本研究提出的联合模型在词性标注任务上融入分词任务中的音节特征可以对单词特征进行补充，使模型可以更好地识别低频词性。

4　结束语

本研究充分考虑泰语的语言特性，提出了一个基于局部 Transformer 网络的泰语分词和词性标注联合模型，该模型充分考虑了泰语的构词特点和拼写规则，使用音节为分词和词性标注任务提供学习特征，利用局部 Transformer 网络进行局部信息建模。在泰语数据集 LST20 上进行了性能评估，试验结果证明该方法对泰语分词和词性标注联合任务有较好的效果。下一步工作，作者考虑将其应用到其他领域，进行相应的联合任务学习。

参考文献：

[1] JOUSIMO J, LAOKULRAT N, CARR B, et al. Thai word segmentation with bi-directional RNN [EB/OL]. (2019−10−03)[2023−11−14]. https://github.com/sertiscorp.

[2] KITTINARADORN R, TITIPAT A, CHAOVAVANICH K, et al. DeepCut: A Thai word tok enization library using Deep Neural Network [EB/OL]. (2019−11−11)

[2023−11−14]. http://doi.org/10.5281/zenodo.345770, accessed on.

[3] CHORMAI P, PRASERTSOM P, RUTHERFORD A. AttaCut: a fast and accurate neural Thai word segmenter [EB/OL]. (2019−12−16) [2023−11−14]. https://arxiv.org/abs/1911.07056.

[4] SCHUSTER M, PALIWAL K K. Bidirectional recurrent neural networks[J]. IEEE transactions on signal processing, 1997, 45(11): 2673–2681.

[5] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84–90.

[6] DONG Chuanhai, ZHANG Jiajun, ZONG Chengqing, et al. Character-based LSTM-CRF with radical-level features for Chinese named entity recognition[M]//Natural Language Understanding and Intelligent Applications. Cham: Springer International Publishing, 2016: 239−250.

[7] DEVLIN J, CHANG Mingwei, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding[EB/OL]. (2018−11−11) [2023−11−14]. https://arxiv.org/abs/1810.04805.pdf.

[8] LAFFERTY J, MCCALLUM A, PEREIRA F. Conditional random fields: probabilistic models for segmenting and labeling sequence data [C]//Proceedings of the 18th Eighteenth International Conference on Machine Learning. Williamstown: ICML, 2001: 282–289.

[9] LIU Yinhan, OTT M, GOYAL N, et al. RoBERTa: a robustly optimized BERT pretraining approach [EB/OL]. (2019−07−26) [2023−11−14]. https://arxiv.org/abs/1907.11692.

[10] HONG T, KIM D, JI M, et al. BROS: a pre-trained language model focusing on text and layout for better key information extraction from documents[C]//Proceedings of the AAAI Conference on Artificial Intelligence. [S.l.]: AAAI, 2022: 10767−10775.

[11] ZHANG Taolin, WANG Chengyu, HU Nan, et al. DKPLM: decomposable knowledge-enhanced pre-trained language model for natural language understanding[J]. Proceedings of the AAAI Conference on Artificial Intelligence. [S.l.]: AAAI, 2022: 11703−11711.

[12] LOWPHANSIRIKUL L, POLPANUMAS C, JANTRAKULCHAI N, et al. WangchanBERTa: pretraining transformer-based Thai language models [EB/OL]. (2021−05−20) [2023−11−14]. https://arxiv.org/abs/2101.09635.

[13] NG H, LOW J K. Chinese part-of-speech tagging: one-at-a-time or all-at-once? word-based or character-based? [C]// Proceedings of the Conference on Empirical Methods in Natural Language Processing. Barcelona: EMNLP, 2004: 277−284.

[14] SØGAARD A, GOLDBERG Y. Deep multi-task learning with low level tasks supervised at lower layers[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics . Berlin: Association for Computational Linguistics, 2016: 231−235.

[15] JIANG Wenbin, HUANG Liang, LIU Qun, et al. A cascaded linear model for joint Chinese word segmentation and part-of-speech tagging[C]// Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics. Columbus: [s.n.], 2008: 897−904.

[16] SUN Weiwei. A stacked sub-word model for joint Chinese word segmentation and Part-of-Speech tagging[C]// Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland: [s.n.], 2011: 1385−1394.

[17] ZENG Xiaodong, WONG D F, CHAO L S, et al. Graph-based semi-supervised model for joint Chinese word segmentation and part-of-speech tagging[C]//Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. Sofia: [s.n.], 2013: 770−779.

[18] 潘华山, 严馨, 周枫, 等. 基于层叠条件随机场的高棉语分词及词性标注方法 [J]. 中文信息学报, 2016, 30(4): 110–116.
PAN Huashan, YAN Xin, ZHOU Feng, et al. A Khmer word segmentation and part-of-speech tagging method based on cascaded conditional random fields[J]. Journal of Chinese information processing, 2016, 30(4): 110–116.

[19] TIAN Yuanhe, SONG Yan, AO Xiang, et al. Joint Chinese word segmentation and part-of-speech tagging via two-way attentions of auto-analyzed knowledge[C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Seattle: [s.n.], 2020: 8286−8296.

[20] BUOY R, TAING N, KOR S. Joint Khmer word segmentation and part-of-speech tagging using deep learning[EB/OL]. (2021−03−31)[2022−01−01]. https://arxiv.org/abs/2103.16801.pdf.

[21] LI Y, LI Xiaomin, WANG Yiru, et al. Character-based joint word segmentation and part-of-speech tagging for Tibetan based on deep learning[J]. Transactions on Asian and low-resource language information processing, 2022: 2375−4699.

[22] YUAN Lichi. A joint method for Chinese word segmentation and part-of-speech labeling based on deep neural network[J]. Soft Computing, 2022, 26(12): 5607–5616.

[23] 林颂凯, 毛存礼, 余正涛, 等. 基于卷积神经网络的缅甸语分词方法 [J]. 中文信息学报, 2018, 32(6): 62–70,79.
LIN Songkai, MAO Cunli, YU Zhengtao, et al. A method of Myanmar word segmentation based on convolution

neural network[J]. Journal of Chinese information processing, 2018, 32(6): 62–70,79.

[24] XIANG Yan, XU Ying, YU Zhengtao, et al. CNN-based text multi-classifier using filters initialised by N-gram vector[J]. International journal of information and communication technology, 2019, 15(4): 419.

[25] 郭振, 张玉洁, 苏晨, 等. 基于字符的中文分词、词性标注和依存句法分析联合模型 [J]. 中文信息学报, 2014, 28(6): 1–8.
GUO Zhen, ZHANG Yujie, SU Chen, et al. Character-level dependency model for joint word segmentation, POS tagging, and dependency parsing in Chinese[J]. Journal of Chinese information processing, 2014, 28(6): 1–8.

[26] 刘一佳, 车万翔, 刘挺, 等. 基于序列标注的中文分词、词性标注模型比较分析 [J]. 中文信息学报, 2013, 27(4): 30–36.
LIU Yijia, CHE Wanxiang, LIU Ting, et al. A comparison study of sequence labeling methods for Chinese word segmentation, POS tagging models[J]. Journal of Chinese information processing, 2013, 27(4): 30–36.

[27] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all You need [EB/OL]. (2017−06−12) [2023−01−01]. https://arxiv.org/abs/1706.03762.

[28] PHATTHIYAPHAIBUN W, CHAOVAVANICH K, POLPANUMAS C, et al. Pythainlp: Thai natural language processing in python [EB/OL]. (2022−07−03) [2023−01−01] . https://github.com/PyThaiNLP/pythainlp.

[29] UDOMCHAROENCHAIKIT C, BOONKWAN P, VATEEKUL P. Adversarial evaluation of robust neural sequential tagging methods for Thai language[J]. Transactions on Asian and low-resource language information processing. 2020: 1−25.

[30] KINGMA D P, BA J. Adam: a method for stochastic optimization [EB/OL]. (2014−12−24) [2023−11−14]. https://arxiv.org/abs/1412.6980.

作者简介：

朱叶芬，硕士研究生，主要研究方向为自然语言处理、词法分析。E-mail：846415516@qq.com。

线岩团，副教授，主要研究方向为自然语言处理、信息抽取。主持和参与国家自然基金项目和云南省自然科学基金项目及其他纵向课题 10 项，主持横向课题 2 余项，获专利授权和软件著作权 10 余项。发表学术论文 20 余篇。E-mail：xianyt@kust.edu.cn。

余正涛，教授，主要研究方向为自然语言处理、信息检索、机器翻译、机器学习。主持和参与国家自然基金项目和云南省自然科学基金项目及其他纵向课题 30 项，主持横向课题 20 余项，获专利授权和软件著作权 50 余项。发表学术论文 80 余篇。E-mail：ztyu@hotmail.com。