依存句法分析联合模型调研报告

吴林志

November 30, 2019

Contents

1 概述

依存句法分析(Dependency Parsing)最常用的两种方法是基于转移(Transition-based) 的方法和基于图(Graph-based)的方法。其中，基于转移的方法是通过shift-reduce两个基本动作来将序列转换为树结构，其解析过程为：首先设计一系列actions (shift, left-arc, right-arc)，其就是有方向带类型的边，接着从左向右依次解析句子中的每一个词，解析词的同时通过选择某一个action 开始增量构建依存树，直到句子中的词解析完。优点是解析过程是线性的，效率高；缺点是在解析的每一步都只是利用局部信息，会导致错误传播。基于图的方法的解析过程：学习一个打分函数，针对一句话的所有可能的解析结果(依存树) 中执行全局穷举搜索，得到一个得分最高的解析树。优点是简单、易于理解、效果较好(甚至优于基于转移的方法) ；缺点是搜索过程速度很慢。

近年来，学者提出了针对这两种方法的神经网络模型，基于转移的方法目前最好的模型是Stack LSTM[1] (通过三个LSTM 来分别建模栈状态、待输入序列和动作序列) ，基于图的方法目前最好的模型是Biaffine[2] (直接用神经网络来预测每两个词之间存在依存关系的概率，得到一个全连接图，图上每个边代表了节点a 指向节点b 的概率，最后使用MST 等方法将图转换为一棵树) 。

因为词是语言中的最小语义单位，所以在很多NLP任务中，需要先进行(中文) 分词，然后标注它的词性，并分析句子的句法结构(主要讨论依存句法分析，建立词与词之间的依赖关系) 。由于传统的pipeline 模型存在误差传播(Error Propagation)和知识共享(各个子任务实际上密切相关) 的局限，近年来，不少学者提出了各自改进的联合模型(Joint Model)。本调研报告主要调研了近两年来的几篇相关论文[3, 4, 5, 6, 7]，分享针对分词、词性标注和基于图的依存句法分析联合模型的学术研究成果。

2 研究总结

2.1 词性标注+句法分析

更准确的词性标签能够提升句法分析的性能，而解析树的句法上下文对解决词性标注的歧义问题有帮助(联合模型性能往往优于单个模型) 。*Dat Quoc Nguyen*(2017) 等人提出一种新颖的神经网络模型jPTDP(**j**oint **P**OS **t**agging and **d**ependency **p**arsing) [3]，联合学习词性标注和基于图的依存句法分析，该联合模型使用BiLSTM 来学习输入序列的特征表示，并将其共享于词性标注和依存分析任务，解决了特征工程(feature-engineering) 问题。在通用依存分析项目(the Universal Dependency Project) 的19 种语言上的实验结果表明，作者提出的联合模型在不使用任何外部资源(如预训练词向量) 情况下，优于强大的基准模型，尤其是目前最好的联合词性标注

和基于转移的依存分析的栈传播模型(stack-propagation model)，实现了一个新的SOTA。

2.1.1 jPTDP

模型结构如图1 所示。

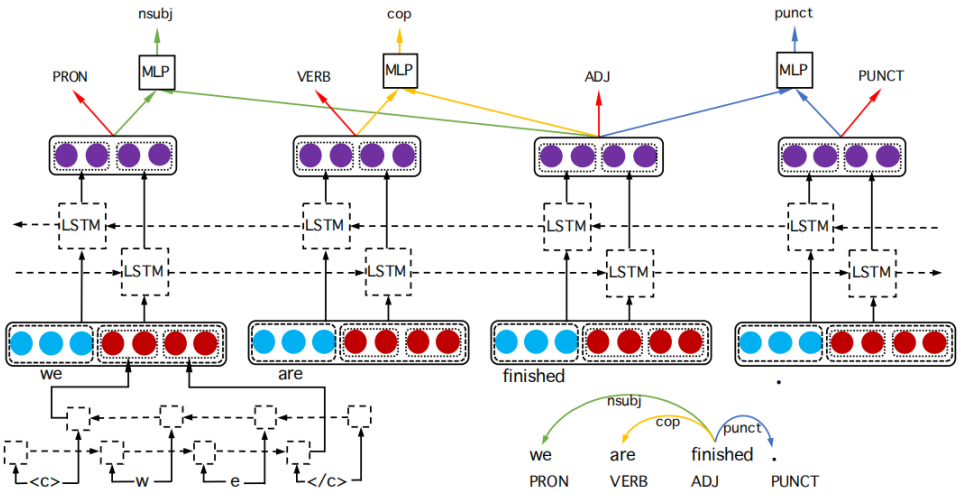


Figure 1: jPTDP模型结构

对于给定的输入序列，将每个词w_i用词向量$e_{wi}^{(W)}$ 表示，同时，由于词的字符级特征表示能够提高词性标注和依存分析的性能，作者采用序列BiLSTM ($BiLSTM_{seq}$)，以每个词的字符序列$c_{1:k}$ 作为输入得到每个词的字符级向量表示

$$e_w^{(C)} = BiLSTM_{seq}(c_{1:k}) = LSTM_f(c_{1:k}) \oplus LSMT_r(c_{k:1}) \tag{1}$$

然后将词向量和字符级的向量进行拼接

$$e_i = e_{wi}^{(W)} \oplus e_{wi}^{(C)} \tag{2}$$

以此作为每个词对应的特征表示喂给另一个(共享) BiLSTM($BiLSTM_{ctx}$)，从而得到w_i 的共享特征向量$v_i = BiLSTM_{ctx}(e_{1:n})$。

词性标注 使用基于共享BiLSTM的特征向量表示，然后采用常规方法计算预测的词性标签序列$\hat{t}$ 和标准的词性标签序列t 的交叉熵损失$\ell_{POS}(\hat{t}, t)$。

基于图的依存分析 依存树可以被规范成有向图，先通过一种弧解析方法学习图中弧的分数

$$score(s) = \underset{\hat{y}\in\Psi(s)}{argmax} \sum_{(h,m)\in\hat{y}} score_{arc}(h,m) \tag{3}$$

$$score_{arc}(h,m) = MLP_{arc}(v_h \oplus v_m) \tag{4}$$

其中$\Psi(s)$是输入序列的所有可能的依存树的集合，$score_{arc}(h,m)$ 是衡量序列中的中心词h^{th} 和修饰词m^{th} 之间的弧分数，v_h 和v_m分别是h^{th} 词和m^{th}词基于共享BiLSTM的特征向量表示。然后使用高效的Eisner解码算法，从中可以找到一棵最大生成树MST (得分最高的解析树) 。依存关系类型的预测采用类似的方式计算

$$score_{rel}(h,m) = MLP_{rel}(v_h \oplus v_m) \tag{5}$$

都是在$BiLSTM_{ctx}$层输出之后使用一个MLP 层得到弧或弧标签的分数。弧和弧标签的预测误差损失均采用基于间隔的hinge loss (ℓ_{arc})。

最终联合模型训练的目标是最小化词性标注损失ℓ_{POS}、弧结构损失ℓ_{arc}和弧关系标注损失ℓ_{rel} 的总和

$$\ell = \ell_{POS} + \ell arc + \ell rel \tag{6}$$

模型参数包括词嵌入、字符嵌入、两个BiLSTM 和两个MLP。

实现细节 使用Adam优化目标函数；设置固定的随机种子，在所有的实验中不使用预训练词向量；词的dropout 概率值设为0.25，高斯噪声设为0.2；共训练30 轮；在开发集上，采用词性标注、依存弧和标签类型的混合准确率(mixed accuracy) 进行性能评估。在英文语料上，通过采用小型的网格搜索来调整超参数。字符向量维度设为64，词向量维度设为128，两个BiLSTM 的隐层维度设为128，MLP隐层维度设为100。

2.1.2 改进

前文讨论的联合模型是jPTDPv1.0，作者在此基础上进一步做出改进，提出jPTDPv2.0[5]，在UD(universal dependencies) 树库数据集上，LAS 分数整体超过jPTDPv1.0大约2.5 个点，模型结构如图2 所示。

v1.0是采用BiLSTM学习“共享”的潜在特征向量，然后用于词性标注和依存分析任务，而不是使用两个独立的层。不同于v1.0，改进的联合模型分成词性标注组件和句法分析组件，对于给定的输入序列，词性标注组件使用BiLSTM ($BiLSTM_{pos}$) 学习它们潜在的特征向量表示，

$$v_i^{pos} = BiLSTM_{pos}(e_{1:n}) \tag{7}$$

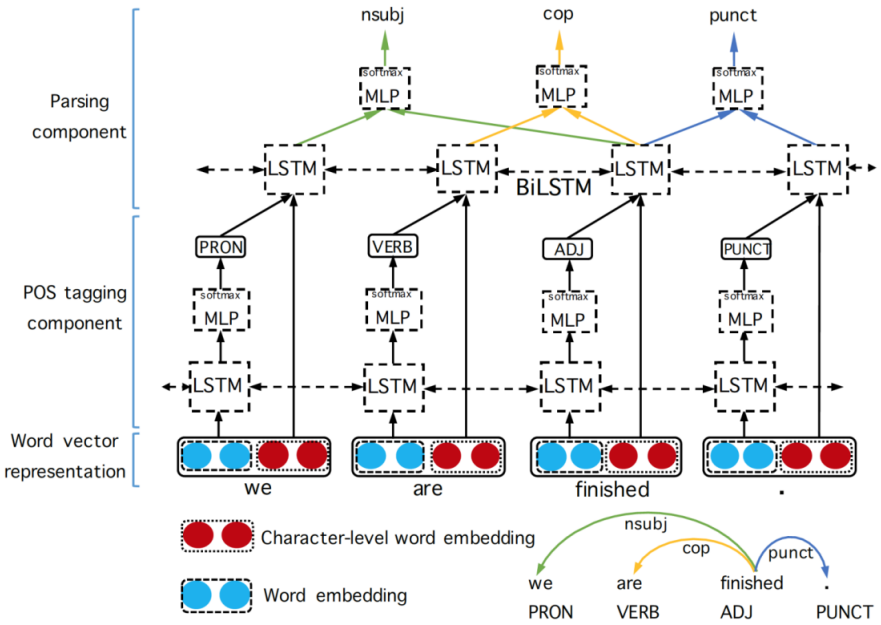


Figure 2: jPTDPv2.0模型结构

然后标注组件将这些特征向量喂给一个以softmax 为输出的单隐层MLP (MLP_{pos}) 去预测词性标签

$$\vartheta_i = MLP_{pos}(v_i^{(pos)}) \tag{8}$$

基于输出向量ϑ_i，计算输入序列的预测词性标签$\hat{t}$ 和标准词性标签t 的交叉熵损失$\ell_{POS}(\hat{t}, t)$

句法分析组件基于输入的词序列和预测的词性标签，使用另一个BiLSTM ($BiLSTM_{dep}$) 去学习另一组潜在的特征表示，将这些特征表示喂给一个MLP 并解码得到依存弧，同时喂给另一个MLP 得到预测的依存弧标签。

假设由词性标注组件对输入词序列所预测的词性标签序列为$p_1, p_2, ..., p_n$，将第i 个预测的词性标签用向量$e_{pi}^{(P)}$ 表示

$$x_i = e_{pi}^{(P)} \oplus e_i = e_{pi}^{(P)} \oplus e_{wi}^{(W)} \oplus e_{wi}^{(C)} \tag{9}$$

将向量序列$x_{1:n}$喂给$BiLSTM_{dep}$，得到潜在的特征向量

$$v_i = BiLSTM_{dep}(x_{1:n}) \tag{10}$$

基于得到的特征向量，利用方程(11) 容易得到序列中第i 个词和第j 个词之间的弧分数

$$score_{arc}(i, j) = MLP_{arc}(v_i \oplus v_j \oplus (v_i * v_j) \oplus |v_i - v_j|) \tag{11}$$

给定弧分数，使用Eisner解码算法可以找到得分最高的可投射(projective)解析树

$$score(s) = \underset{\hat{y}\in\Psi(s)}{argmax} \sum_{(h,m)\in\hat{y}} score_{arc}(h,m) \tag{12}$$

其中$\Psi(s)$是输入序列的所有可能依存树的集合，$score_{arc}(h,m)$ 是衡量序列中中心词h^{th} 和修饰词m^{th} 之间的弧分数。

对于中心词-修饰词之间弧的依存关系类型的预测，采用另一个以softmax为输出的MLP(MLP_{rel}) 层。给定弧(h,m)，我们可以采用方程(13) 计算弧标签得分

$$score_{rel}(h,m) = MLP_{rel}(v_h \oplus v_m \oplus (v_h * v_m) \oplus |v_h - v_m|) \tag{13}$$

实现细节 主要阐明和jPTDPv1.0不同的地方：词嵌入层可以被随机初始化(维度100) 也可以用预训练的词向量(在Wikipedia和Gigaword 数据集上预训练的Glove 词向量)初始化；字符嵌入(维度50) 和词性标签嵌入(维度100) 均被随机初始化；使用单层的$BiLSTM_{seq}$ 并将隐层大小设成字符嵌入层的向量大小；采用*word dropout* 方式(将训练集中的每个词w 以概率$p_{unk}(w) = \frac{0.25}{0.25+\#(w)}$ 替换成未知词unk，其中$\#(w)$ 是词w 在训练集中出现的次数) 去学习未知词的词向量；使用学习速率初始化为0.001 的Adam优化器，训练30 轮并且在每10 轮之后将学习速率降为原来的一半(学习速率退火) 。选择开发集上准确率最高的模型，用于评估阶段的测试集上。

2.2 分词+句法分析

依存句法分析是定义在词级别的，因此分词是依存分析的前置条件，分词错误容易导致依存分析的错误传播。*Hang Yan*[6] 等人提出了结合中文分词和依存分析的统一模型。作者论文的主要贡献有：

- 首次采用(字符级别) 基于图的方式整合中文分词和依存分析；非常简洁，容易实现。
- 相比较基于转移的联合模型，需要更少的特征工程，并且可以处理带标签的依存分析任务。
- 在(Penn Chinese Treebank) CTB-5 和CTB-7 数据集上的实验，实现了在联合中文分词和依存分析的SOTA 成绩，即使没有用到词性信息。

因为分词是字符级别的任务，依存分析是词级别的任务，作者首次将这两个任务制定成字符级别基于图的解析框架。细节上，该联合模型首先包含深层的BiLSTM 编码器，能够捕捉每个字符长期的上下文特征；其次，

使双仿射(Biaffine)打分器，统一在字符级别预测分词和依存关系。除此之外，不同于之前的联合模型，作者的联合模型不依赖于词性标注任务。

首先，作者将分词转化成特殊的弧预测问题，比如，词“金融业”有两个内部词的依存弧：“金←融”和“融←业”，内部词的依存弧都有“app”标签，论文中作者简单地把一个词的最后一个字符定义成head 字符，所有其他的字符都依赖于它。其次，作者将词级别的依存弧转化成字符级别的依存弧，假定词$w_1 = x_{i:j}$ 和词$w_2 = x_{u:v}$ 之间存在依存弧，他们让这条弧连接每个词的最后字符x_j 和x_v。比如，弧“发展→ 金融业”转换成“展→业”。图3 展示了中文分词和依存分析的联合框架。

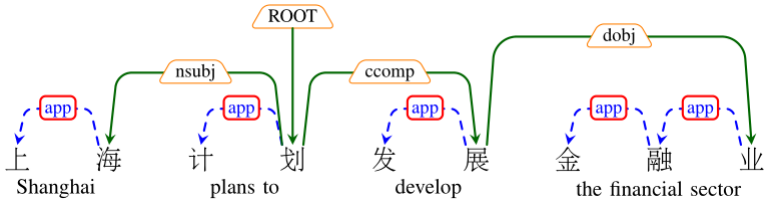


Figure 3: 中文分词和依存分析统一框架；绿色弧代表词级别的依存关系，带有“app”标记的虚线弧代表属于同一个词的连接字符

因此，作者使用统一的基于图的解析模型去构建这两个任务。模型包括两个主要的组件：

1. 深层的BiLSTM编码器提取上下文特征，以给定的句子的每个字符嵌入作为输入并产生稠密向量。

2. 双仿射注意力打分器，以给定字符对的隐层向量作为输入并预测标签分数向量。

如图4展示了中文分词和依存分析的统一联合模型：

编码层 ：给定字符序列$X = x_1, x_2, ..., x_N$，将每个字符x_i 映射成嵌入向量e_i，同时作者调研先前的工作发现双字符(bigram) 和三字符(trigram) 特征对中文分词和词性标注很有效，因此作者结合双字符和三字符来增强每个字符的特征表示，最后的字符x_i 特征表示为$e_i = e_{c_i} \oplus e_{c_i c_{i+1}} \oplus e_{c_i c_{i+1} c_{i+2}}$ (unigram，bigram和trigram的embeddings 在中文维基百科语料上预训练而来)。

为了能捕捉长远的上下文信息，作者使用了深层BiLSTM，LSTM对应第i 个字符的隐层状态为

$$h_i = BiLSTM(e_i, \overrightarrow{h_{i-1}}, \overleftarrow{h_{i+1}}, \theta) \tag{14}$$

其中，$\overrightarrow{h_i}$和$\overleftarrow{h_i}$ 分别是LSTM 前向传播和反向传播在第i 个位置的隐层状态，θ 表示BiLSTM 层的所有参数。

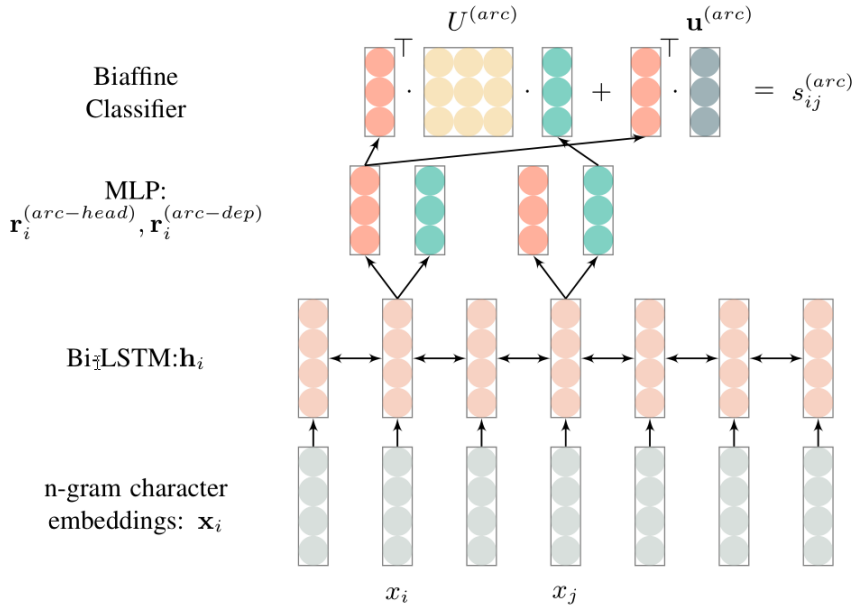


Figure 4: 联合模型

Biaffine层 为了预测每个字符对的关系，作者采用双仿射注意力机制[2] (能更高效地衡量两个基本单元之间的关系) 给它们在BiLSTM 编码器的输出上打分。

1) 弧预测：

$$r_i^{(arc-head)} = MLP^{(arc-head)}(h_i), \tag{15}$$

$$r_j^{(arc-dep)} = MLP^{(arc-dep)}(h_j), \tag{16}$$

$$s_{ij}^{(arc)} = r_i^{(arc-head)} U^{(arc)} r_j^{(arc-dep)} + r_i^{(arc-head)^T} u^{(arc)} \tag{17}$$

因此，$s_j^{(arc)} = [s_{1j}^{(arc)}; ...; s_{Tj}^{(arc)}]$ 是第j 个字符可能为head 的分数，然后经过softmax 函数得到其概率分布。

在训练阶段，目标是最小化*golden head-dependent* 对的交叉熵损失；在测试阶段，确保解析的结果是规范的(well-formed) 树。

2) 弧标签预测： 在得到最佳无标签预测树之后，将为每条弧$c_i \to c_j$ 赋予标签分数$s_{ij}^{(label)} \in \mathbf{R}^K$，$K$是标签集大小。在作者的联合模型中，弧标签集包含标准的词级别依存弧标签和特殊标签“app”(表明一个词内部的依存关系)

Figure 5: 分词标签预测

对于弧$c_i \to c_j$，我们可以通过公式(18)-(21) 得到$s_{ij}^{(label)}$

$$r_i^{(label-head)} = MLP^{(label-head)}(h_i), \tag{18}$$

$$r_j^{(label-dep)} = MLP^{(label-dep)}(h_j), \tag{19}$$

$$r_{ij}^{(label)} = r_i^{(label-head)} \oplus r_j^{(label-dep)}, \tag{20}$$

$$s_{ij}^{(label)} = r_i^{(label-head)} U^{(label)} r_j^{(label-dep)} + W^{(label)}(r_{ij}^{(label)}) + u^{(label)} \tag{21}$$

其中，$U^{(label)} \in \mathbf{R}^{K \times p \times p}$ 是三维向量，$W^{(label)} \in \mathbf{R}^{K \times 2p}$ 是权重矩阵，$u^{(label)} \in \mathbf{R}^K$是偏置向量。弧$c_i \to c_j$的最好标签是根据公式(22)确定。

$$y_{ij} = \underset{label}{argmax} s_{ij}^{(label)} \tag{22}$$

在训练期间，作者使用*golden head-dependent*关系和交叉熵去优化弧标签的预测。带有连续“app”的弧(邻接且向左) 的字符可以归结为一个词。如果一个字符没有向左的“app”标签弧，那么它是单字词。解码时，作者首先用提出的模型去预测字符级别的带标签依存树，然后根据预测的字符级的弧标签复原成分词和基于词的依存树。带有连续“app”的字符视作同一个词，同时预测出的一个词的最后一个字符的head视作这个词的head。因为预测的弧指向一个字符，作者将包含head 字符的词当做head 词。

实验

数据集 作者使用Penn Chinese Treebank 5.0 (CTB-5)和7.0 (CTB-7)来评估模型，同时将数据集分成训练集、开发集和测试集，数据集统计如表1所示。

衡量指标 作者使用词级别的标准衡量指标F1，precision和recall去评价分词和依存分析(有标签和无标签场景) 任务，除此之外，也报告词级别依存分析的标准指标：UAS 和LAS。

- $F1_{seg}$：衡量中文分词任务；$F1_{seg} = 2 * P_{seg} * R_{seg}/(P_{seg} + R_{seg})$

Table 1: CTB-5和CTB-7数据统计

Dataset	Partition	#sent	#word	#OOV
CTB-5	Training	18k	494k	-
	Develop	350	6.8k	553
	Test	348	8.0k	278
CTB-7	Training	31k	718k	-
	Develop	10k	237k	13k
	Test	10k	245k	13k

- $F1_{udep}$：衡量无标签(弧) 依存分析；使用词级别标准指标F1，P和R来评估依存分析；在联合分词和依存分析的任务中，广泛使用的UAS是不足以衡量性能，因为误差来自两个方面：一个是分词，另一个是因为head词的错误预测。一个dependent-head 对只有在dependent 词和head 词被正确切分并且dependent词正确找到它的head词才算是正确的。$F1_{udep} = 2 * P_{udep} * R_{udep}/(P_{udep} + R_{udep})$
- $F1_{ldep}$：衡量带标签依存分析；和$F1_{udep}$ 唯一的区别是弧上标签也必须预测正确；$F1_{ldep}$不可能高于$F1_{udep}$。
- UAS：因为在依存分析任务中，每个词只有一个head 词，正确head的词所占的百分比可以用作一个评价性能的指标；$UAS = R_{udep}$
- LAS：衡量依存标签的正确率；一个依存标签当弧和它的标签正确时，才视作是正确的。$LAS = R_{ldep}$

实验设置 作者采用整合了token order信息的word2vec在中文维基百科语料上预训练了unigram，bigram和trigram embeddings (作者的简化测试实验表明模型融入预训练的字符向量能给句法分析的性能带来大的提升) 。对于词依存分析，作者使用了腾讯的预训练词向量(200维) 。所有预训练的词向量是固定的。模型使用Adam算法最小化弧预测和标签预测的交叉熵损失和；所有模型训练100轮，每轮训练之后，在开发集上测试模型，并保存开发集上$F1_{udep}$最高的模型。详细的超参数设置如表2 所示。

2.3 分词+词性标注+句法分析

Dat Quoc Nguyen[7]等人首先提出了越南语分词、词性标注和依存分析的联合多任务模型jointWPD。作者扩展了BIST 基于图的依存分析器[8]，采用基于BiLSTM-CRF网络层用于分词和词性标注。在越南语的基准数据集上的实验结果表明作者的联合模型实现了SOTA 的性能。

Table 2: 参数设置

Embedding dimension	100
BiLSTM hidden size	100
Gradients clip	5
Batch size	128
Embedding dropout	0.33
LSTM dropout	0.33
Arc MLP dropout	0.33
Label MLP dropout	0.33
LSTM depth	3
MLP depth	1
Arc MLP size	500
Label MLP size	100
Learning rate	2e-3
Annealing	$0.75^{t/5000}$
β_1, β_2	0.9
Max epochs	100

不同于英文，对于越南语NLP任务，分词是首要考虑的，因为在越南语中空白字符除了用来标记单词的边界外，还用来分隔组成单词的音节[1]。

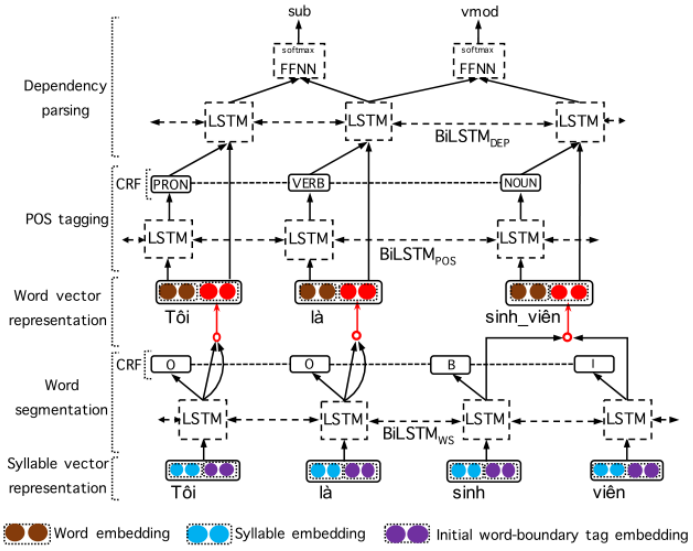
实际解析越南语文本的pipeline处理过程是：首先使用分词器切分文本，切分的文本提供给词性标注器(自动产生词性标注文本)，进而喂给句法解析器。

然而，越南语的分词器和词性标注器都有错误率，因此会导致误差传播。解决这些问题的一种方案是开发联合学习分词、词性标注和依存分析的模型。论文中，作者提出一个新的端到端基于神经网络的联合多任务模型，用于分词、词性标注和依存句法分析。

联合模型 如图6所示，作者的联合模型可以看做是分词、词性标注和依存分析三个组件的混合体。分词组件将越南语分词任务正式化成序列标注问题，因此采用BiLSTM-CRF结构从音节输入中去预测BIO 词边界标签，得到分词序列；词性标注组件也采用BiLSTM-CRF从分词序列中去预测词性标签。基于分词输入和它们预测到的词性标签，依存分析组件采用基于图的结构(类似于[8]) 去解码依存弧和标签。

音节向量表示 给定有m个音节的输入序列$s_1, s_2, ..., s_m$，作者采用一个初始分词器(VnCoreNLP 的RDR分词器，基于词典的最长匹配策略)产生

[1]大约85%的越南词是由至少两个音节组成，并且80% 以上的音节是单词本身



ID	Form	POS	Head	DepRel
1	Tôi $_{\text{I}}$	PRON	2	sub
2	là $_{\text{am}}$	VERB	0	root
3	sinh_viên $_{\text{student}}$	NOUN	2	vmod

Figure 6: 分词、词性标注和依存分析联合模型jointWPD

初始BIO词边界标签$b_1, b2, ..., b_m$，作者构造向量v_i来表示输入序列中第i个音节，采用的方式是拼接音节嵌入e^S_{si}和它的初始词边界标签嵌入e^B_{bi}。

$$v_i = e^S_{si} \oplus e^B_{bi} \tag{23}$$

分词 分词组件使用BiLSTM ($BiLSTM_{ws}$)学习向量序列$v_{1:m}$中第i个音节的特征向量表示

$$r^{(ws)}_i = BiLSTM_{ws}(v_{1:m}, i) \tag{24}$$

然后使用单层前馈网络($FFNN_{ws}$)对每个特征向量进行线性变换

$$h^{(ws)}_i = FFNN_{ws}(r^{(ws)}_i) \tag{25}$$

紧接着，分词组件将输出向量h^{ws}_i喂给链式CRF层，用于最终的BIO 词边界标签预测。训练期间，计算交叉熵目标损失ℓ_{ws}，同时维特比(Viterbi) 算法用于解码。

词向量表示 假设我们基于m个音节输入序列，得到了n个词$w_1, w_2, ..., w_n$，需要注意的是训练时我们使用标准(gold)分词，解码时使用分词组件预测

的分词。我们构造向量x_j 来表示第j个词w_j，采取的方式是拼接词向量$e_{w_j}^{(W)}$和音节级别的词向量$e_{w_j}^{(SW)}$

$$x_j = e_{w_j}^{(W)} \oplus e_{w_j}^{(SW)} \tag{26}$$

为了得到$e_{w_j}^{(SW)}$，作者结合句子级别的上下文相关的音节编码(来自公式(24)) 并将它喂给FFNN($FFNN_{sw}$)

$$e_{w_j}^{(SW)} = FFNN_{sw}(r_{f(w_j)}^{(WS)} \oplus r_{l(w_j)}^{(WS)}) \tag{27}$$

其中$f(w_j)$ 和$l(w_j)$ 分别代表序列中词w_j的首尾音节。

词性标注 **词**性标注组件首先将向量序列$x_{1:n}$喂给BiLSTM ($BiLSTM_{POS}$)来学习输入词的特征向量表示，并且每个特征向量都作为输入经过一个FFNN ($FFNN_{POS}$)层

$$r_j^{(POS)} = BiLSTM_{POS}(x_{1:n}, j) \tag{28}$$
$$h_j^{(POS)} = FFNN_{POS}(r_j^{(POS)}) \tag{29}$$

然后，输出向量$h_j^{(POS)}$喂给一个CRF层用于词性标签预测。训练时，交叉熵损失ℓ_{POS}被计算用于词性标注。

依存分析 **假**设词性标注组件得到输入词$w_1, w_2, ..., w_n$相应的预测词性标签$p_1, p_2, ..., p_n$。预测到的第j个词性标签p_j表示成向量$e_{p_j}^{(P)}$，作者构造一个向量序列$z_{1:n}$作为依存分析组件的输入，其中z_j 是词向量x_j (来自公式(26)) 和相应的词性标签向量$e_{p_j}^{(P)}$ 拼接的结果。依存分析组件采用BiLSTM($BiLSTM_{DEP}$) 来学习输入序列$z_{1:n}$ 的特征表示

$$z_j = x_j \oplus e_{p_j}^{(P)} \tag{30}$$
$$r_j^{(DEP)} = BiLSTM_{DEP}(z_{1:n}, j) \tag{31}$$

基于特征向量$r_j^{(DEP)}$要么采用基于转移或基于图的结构用于句法分析。作者的句法分析组件的构建类似于BIST基于图的依存分析，不同之处在于采用FFNNs将$r_j^{(DEP)}$拆分成head和dependent的特征表示：

$$h_j^{(A-H)} = FFNN_{Arc-Head}(r_j^{(DEP)}) \tag{32}$$
$$h_j^{(A-D)} = FFNN_{Arc-Dep}(r_j^{(DEP)}) \tag{33}$$
$$h_j^{(L-H)} = FFNN_{Label-Head}(r_j^{(DEP)}) \tag{34}$$
$$h_j^{(L-D)} = FFNN_{Label-Dep}(r_j^{(DEP)}) \tag{35}$$

为了给可能的依存弧打分，作者采用一个$FFNN_{ARC}$输出层：

$$score(i,j) = FFNN_{ARC}(h_j^{(A-H)} \oplus h_j^{(A-D)}) \tag{36}$$

给定词对的分数，作者采用Eisner解码算法预测得分最高的可投射(projective)解析树，该无标签解析模型采用基于间隔的hinge loss ℓ_{ARC}来训练。为了给预测弧打标签，作者采用另一个$FFNN_{LABEL}$，以softmax作为输出：

$$v_{(i,j)} = FFNN_{LABEL}(h_i^{(L-H)} \oplus h_j^{(L-D)}) \tag{37}$$

基于向量$v_{(i,j)}$，在训练期间，利用精标(gold labeled)的树，计算预测依存标签的交叉熵损失ℓ_{LABEL}。

联合多任务学习 在计算梯度之前，作者通过求$\ell_{WS}, \ell_{POS}, \ell_{ARC}, \ell_{LABEL}$的和来训练模型，通过学习模型参数来最小化误差和。

作者的模型可以看做是jPTDP-v2[5]的延伸，整合了BiLSTM-CRF 用于单词边界的预测。其他相对于jPTDP-v2的改进包括：

(i) 采用“全局”句子级上下文来学习词表示(公式(27))，而不是“局部”基于单个词的字符级表示。

(ii) 采用CRF层用于词性标注而不是softmax层

(iii) 采用head和dependent映射表示(公式(32)-(35)) 作为特征向量用于依存分析，而不是循环层的输出(公式(31))

实验

数据集 对于分词和词性标注，作者使用来自VLSP(Vietnamese Language and Speech Processing) 2013标准数据集。为了训练分词层，使用75K (人工) 分词句子，其中70K用于训练，5K用于开发集；对于词性标注，使用27870条(人工) 分词并标注词性的句子，其中27K 用于训练，870 条用作开发集；对于这两个任务，测试集包括2120条(人工) 分词和标注词性的句子。为了训练依存分析层，作者使用10200条VnDT(Vietnamese Dependency Treebank)中的句子，按照标准划分：最后1020个句子用作测试集，前200个句子用作开发集，剩下的8980个句子用作训练。

实现细节 作者联合模型的采用DYNET v2.0实现；使用Adam学习模型参数并训练50 轮；训练时，每个任务组件都将获得与任务相关的相应训练语句；依存分析的训练集最小(包括8980个句子)，因此每轮训练，作者从分词和词性标注的训练集中采样相同数量的句子；Dropout用于BiLSTM 和FFNN 的输入，丢弃的概率值为0.33；音节和词的嵌入

层用100维预训练的*Word2Vec* 初始化，词性标签的嵌入层随机初始化(训练时更新) ；$FFNN_{WS}$，$FFNN_{POS}$和$FFNN_{LABEL}$ 的输出层大小分别是BIO词边界标签的数量，词性标签的数量和依存关系类型的数量。对超参数执行最小的网格搜索，得到词性标签的向量维度是100，FFNN层的输出层大小是100，BiLSTM层数是2，LSTM的每层的隐层大小是128；在训练完每轮之后，计算分词、词性标注和(LAS) 依存分析的F1的均值；选择开发集上最高分数的模型运用到测试集上。

2.4 其他(略)

*Jungo Kasai*等人提出一种端到端基于图的TAG(Tree Adjoining Grammar)解析器[4]，联合学习超级标签标注，词性标注和句法分析任务(图7) 。主要改进地方有：

- 采用字符级卷积神经网络用于提取词的字符级特征
- 拼接BiLSTM各子层的双向隐层状态信息，而不仅仅是末层。实验表明这样有利于提升性能。
- 在BiLSTM网络中添加highway结构，以适应深层次的扩展(Deep Highway BiLSTM)

  作者采用下面的公式计算时间步t单个LSTM单元的激活状态：

$$
\begin{aligned}
i_t &= \sigma(W_i[x_t; h_{t-1}] + b_i) && (38)\\
f_t &= \sigma(W_f[x_t; h_{t-1}] + b_f) && (39)\\
o_t &= \sigma(W_o[x_t; h_{t-1}] + b_o) && (40)\\
\tilde{c}_t &= tanh(W_c[x_t; h_{h-1}] + b_c) && (41)\\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t && (42)\\
h_t &= o_t \odot tanh(c_t) && (43)
\end{aligned}
$$

  其中分号;代表拼接，$\odot$是点乘，σ是$sigmoid$函数；在BiLSTM的第一层，输入x_t是词t的向量表示，在所有子层，x_t是之前BiLSTM层的相应输出；BiLSTM在时间步t的输出等于$[h_t^f; h_t^b]$。作者扩展了LSTM，通过让LSTM 层之间以highway 方式连接(门机制，结合当前层和先前层的输出) ，可以避免梯度消失/爆炸问题；特别指出，在highway连接的网络中，作者用下面的式子替换了(43)式

$$
\begin{aligned}
r_t &= \sigma(W_r[x_t; h_{t-1}] + b_r) && (44)\\
h_t &= r_t \odot o_t \odot tanh(c_t) + (1 - r_t) \odot W_h x_t && (45)
\end{aligned}
$$

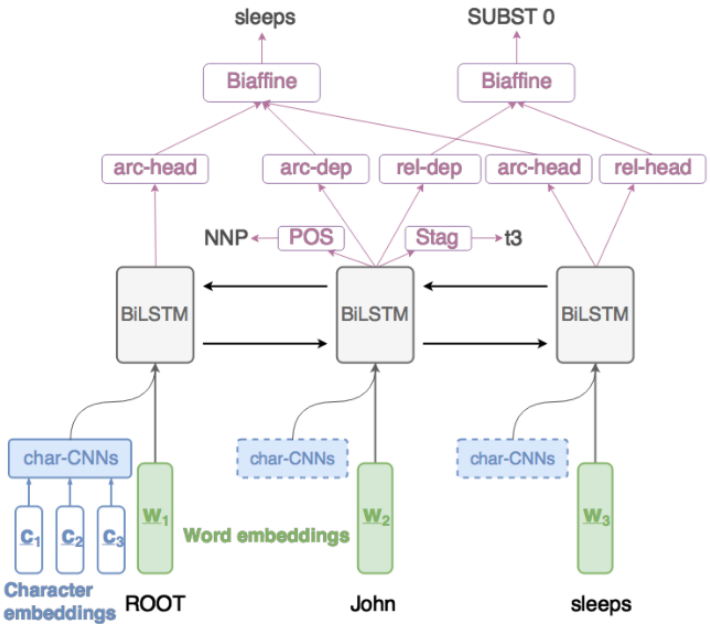


Figure 7: Biaffine句法分析结构*(BiLSTM3-HW-CNN-POS)*

3 总结与展望

由于联合模型能取得比单个任务模型更高的效率和性能，越来越多的学者将研究重心放到联合模型上，词法和句法的有效结合逐渐成为一种新的研究趋势。本文以近年来发表的几篇顶会论文为切入点，调研了主要的几类分词、词性标注和(基于图) 句法分析联合模型，并介绍其的设计、实现及创新点。总体上从特征提取、模型构造、联合方式等方面各显神通：

特征输入 模型输入时融入了字符特征或n-gram特征。如[3][5]采用BiLSTM获取词的字符级特征表示；[6]同时考虑unigram、bigram和trigram embedding作为输入；[4] 在输入层采用字符级CNN网络提取词的形态信息。

模型结构 大多数采用深层BiLSTM作为特征提取器，基于图的依存分析多采用强大的Biaffine[2]模型。[4]拼接BiLSTM各子层的双向隐层状态信息，引入highway结构加深网络层数；[7]在分词和词性标注任务的输出层加入CRF层来提升性能；

联合方式 摒弃pipeline模式，充分共享知识。[6]巧妙地将分词任务转化成二分类弧标签预测问题，从而有机融合了分词和句法分析；[5][7]将词性标注层的预测结果作为特征融入句法分析层的输入，辅助句法分析，并取得了一定性能上的提升。

从调研的结果来看，不难发现更好的上下文特征表示和可靠的联合方式能为联合模型带来性能上的提升，从而逐年产生了愈发强大的baseline。基于此可以考虑采用更先进的特征提取器替代BiLSTM，诸如Transformer[9]、预训练的ELMo[10] 或BERT[11]等，至于联合模型在模型架构和联合方式上的创新与改进仍有待进一步探索。

参考文献

[1] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith, "Transition-Based Dependency Parsing with Stack Long Short-Term Memory," *arXiv e-prints*, p. arXiv:1505.08075, May 2015.

[2] T. Dozat and C. D. Manning, "Deep Biaffine Attention for Neural Dependency Parsing," *arXiv e-prints*, p. arXiv:1611.01734, Nov 2016.

[3] D. Q. Nguyen, M. Dras, and M. Johnson, "A Novel Neural Network Model for Joint POS Tagging and Graph-based Dependency Parsing," *arXiv e-prints*, p. arXiv:1705.05952, May 2017.

[4] J. Kasai, R. Frank, P. Xu, W. Merrill, and O. Rambow, "End-to-end Graph-based TAG Parsing with Neural Networks," *arXiv e-prints*, p. arXiv:1804.06610, Apr 2018.

[5] D. Q. Nguyen and K. Verspoor, "An improved neural network model for joint POS tagging and dependency parsing," *arXiv e-prints*, p. arXiv:1807.03955, Jul 2018.

[6] H. Yan, X. Qiu, and X. Huang, "A Unified Model for Joint Chinese Word Segmentation and Dependency Parsing," *arXiv e-prints*, p. arXiv:1904.04697, Apr 2019.

[7] D. Q. Nguyen, "A neural joint model for Vietnamese word segmentation, POS tagging and dependency parsing," *arXiv e-prints*, p. arXiv:1812.11459, Dec 2018.

[8] E. Kiperwasser and Y. Goldberg, "Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations," *arXiv e-prints*, p. arXiv:1603.04351, Mar 2016.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *arXiv e-prints*, p. arXiv:1706.03762, Jun 2017.

[10] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv e-prints*, p. arXiv:1802.05365, Feb 2018.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv e-prints*, p. arXiv:1810.04805, Oct 2018.