计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210900036

# 融合多策略数据增强的低资源依存句法分析方法

线岩团　高凡雅　相　艳　余正涛　王　剑
昆明理工大学信息工程与自动化学院　昆明 650500
昆明理工大学云南省人工智能重点实验室　昆明 650500
(xianyt@kust.edu.cn)

**摘　要**　依存句法分析旨在识别句子中词与词之间的句法依赖关系。依存句法能为信息抽取、自动问答和机器翻译等任务提供句法特征,提高模型性能。训练数据规模对依存句法分析模型的性能具有重要影响,训练数据的缺乏会带来严重的未知词问题和模型过拟合问题。文中针对低资源依存句法分析问题,提出了多种数据增强策略。所提方法通过同义词替换有效扩充了训练数据,缓解了未知词问题。通过多种 Mixup 的数据增强策略,有效缓解了模型过拟合问题,提高了模型的泛化能力。在(Universal Dependencies treebanks,UD treebanks)数据集上的实验结果表明,所提方法有效提升了小规模训练语料条件下泰语、越南语和英语依存句法分析的性能。

**关键词**:依存句法分析;低资源语言;Mixup 数据增强;同义词替换;多策略
**中图法分类号**　TP391

## Improving Low-resource Dependency Parsing Using Multi-strategy Data Augmentation

XIAN Yan-tuan,GAO Fan-ya,XIANG Yan,YU Zheng-tao and WANG Jian
Faculty of Information Engineering and Automation,Kunming University of Science and Technology,Kunming 650500,China
Yunnan Key Laboratory of Artificial Intelligence,Kunming University of Science and Technology,Kunming 650500,China

**Abstract**　Dependency parsing aims to identify syntactic dependencies between words in a sentence. Dependency parsing can provide syntactic features and improve model performance for tasks such as information extraction,automatic question answering and machine translation. The training data size has an significant impact on the performance of the dependency parsing model. The lack of training data will cause serious unknown word problems and model over-fitting problems. This paper proposes various data augment strategies for the problem of low-resource dependency parsing. The proposed method effectively expands the training data by synonym substitution and alleviates the unknown words problem. The data augment strategies of multiple Mixups effectively alleviate the model overfitting problem and improve the generalization ability of the model. Experimental results on the universal dependencies treebanks(UD treebanks) dataset show that the proposed methods effectively improve the performance of Thai,Vietnamese and English dependency parsing under small-scale training corpus conditions.

**Keywords**　Dependency parsing,Low-resource language,Mixup data augmentation,Synonym substitution,Multi-strategy

## 1　引言

自然语言处理是人工智能领域的热门研究方向,其关键在于对语言的深层次语义结构的理解和表示。句法分析是自然语言处理的基础任务之一,其目标是解析输入的句子并得到其句法结构。常见的句法分析包括成分句法分析和依存句法分析。依存句法分析旨在发现句子中词语之间的二元依存关系[1]。一个依存关系连接两个词,分别是修饰词(dependent)和中心词(head)。如图 1 所示,泰语句子"ดูรันทำหน้าที่เป็นโฆษกและอังเคลปินตาโดเป็นเหรัญญิก"(杜兰担任代言人,安克雷奇担任财务主管)中,"เหรัญญิก"(财务主管)是"อังเคล"(安克雷奇)的核心词,且二者具有 nsubj 的关系,即"อังเคล"是"เหรัญญิก"的名词主语。依存句法关系在信息抽取、自动问答、阅读理解、机器翻译和语言模型中具有重要应用[2]。

到稿日期:2021-09-03　返修日期:2021-10-13
基金项目:国家自然科学基金项目(61732005,61972186);云南省重大科技专项(202002AD080001,202103AA080015);云南省高新技术产业专项(201606)
This work was supported by the National Natural Science Foundation of China(61732005,61972186),Yunnan Provincial Major Science and Technology Special Plan Projects(202002AD080001,202103AA080015) and Yunnan High and New Technology Industry Project(201606).
通信作者:相艳(50691012@qq.com)

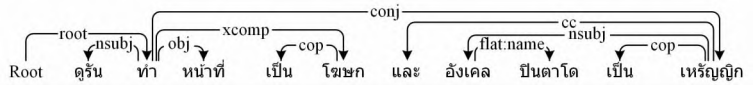


图 1 泰语依存句法示意图

Fig. 1 Schematic diagram of Thai dependency parsing

基于神经网络的依存句法分析模型是当前的主流方法，其可以进一步分为基于转移的方法(transition-based)[3-6]和基于图的方法(graph-based)[7-13]。相比基于转移的方法可以利用到丰富的特征，基于图的方法虽然算法复杂度更高，但其可以进行全局优化，在准确率上高于基于转移的方法。已有方法在特征编码[10]、依存关系打分[11-12]和解码[13]3个方面开展了大量的研究工作，有效提升了依存句法分析的效果。

然而在低资源条件下，已有模型和方法的性能很难获得很好的分析结果。这一问题在泰语、越南语等低资源语言上尤为明显。训练语料的缺少会带来严重的未知词和模型过拟合问题。以越南语的UD数据集为例，测试集的未知词比例为51.7%。据观察，在低资源条件下，模型很容易出现过拟合问题，导致模型的训练准确率和测试准确率差距过大。

本文针对低资源条件依存句法分析问题提出了多策略的数据增强方式来解决这些问题。基于同词性同义词典的替换方法是根据构造的同义词典，依据同词性的同义词对原训练数据集进行数据扩增。基于Mixup的方法则是在模型不同位置上对训练数据中原词与同义词进行混合以产生虚拟新词，利用虚拟新词进行训练。为验证所提方法的有效性，以基于图的双仿射模型[8]的基线模型，应用本文提出方法，在泰语、越南语以及英语的小规模UD数据集上开展实验，实验结果表明，所提融合多策略的数据增强方法提升了依存句法分析任务上的性能。本文的贡献可归纳为以下3点。

(1)本文将多策略的数据增强方法应用于依存句法分析任务中，针对未知词比例高以及数据少的问题提出了基于同词性同义词典替换的数据增强方法，有效扩充了训练数据。

(2)提出多种Mixup的数据增强策略，有效缓解了模型过拟合问题，提高了模型的泛化能力。

(3)所提方法在小规模UD数据集上的结果超过了所使用的基线模型，并在泰语UAS上性能提升了约2%。

## 2 相关工作

随着神经网络和深度学习在自然语言处理领域的深入发展，基于神经网络的依存句法分析方法进展迅速。基于转移的依存解析方法的思想是将复杂任务简化为预测下一个解析动作的简单任务，然后利用分类器进行最优序列的贪婪搜索。Chen等[3]在arc-standrad转移系统的基础上将简单神经网络应用于依存句法解析中，利用神经网络对决策动作进行打分，并采用新的立方激活函数来建模输入之间的交互关系。Dyer等[4]在arc-standrad转移系统的基础上提出使用一个序列到序列的神经网络，即堆栈LSTM来实现当时最先进的基于转移的依存解析器。Andor等[5]对基于转移的神经网络模型进行全局归一化训练，克服局部归一化模型造成的标签偏差问题。Nallani等[6]在基于转移的依存解析系统的基础上提出了一种使用上下文向量表示的最小特征函数来代替过去使用的语言特征模板来实现泰卢固语依存句法解析方法。基于图的依存解析方法是将依存句法分析问题看作从完全有向图中寻找最大生成树的问题。一棵依存树的分值由构成依存树的几种子树的分值累加得到，并在全局优化模型上实现对最大生成树的精确搜索。Kiperwasser等[7]提出一种和解析器进行联合训练的BiLSTM编码器的特征提取方法，并将其应用于基于转移的解析器和基于图的解析器中，该方法在两种依存解析器中都有很好的解析效果。Dozat等[8]在Kiperwasser等[7]工作的基础上，在基于图的依存解析方法上提出使用一个比基于BiLSTM编码器更正则化的解析方法，即使用双仿射分类器对依存弧与依存标签进行打分。Singkul等[9]提出了泰语的依存解析方法，从基于转移的依存解析中发展出5种泰国依存分析算法，从基于图的依存解析中发展出2种泰国依存分析算法，算法使用预先训练的模型来学习字符嵌入，并使用BiLSTM处理双向特征以有效地克服语序问题。Kulmizev等[10]研究了ELMo和BERT等上下文词嵌入模型对依存句法的影响，其在基于转移和基于图的方法中都有很好的效果。Delhoneux等[11]以及Falenska等[12]研究了基于双向长短期记忆网络(BiLSTM)的编码对打分步骤的影响。Zhang等[13]在Dozat等[8]模型的基础上研究了解码阶段不同的结构约束对依存分析结果的影响，结果表明，通过全局输入建模，即使忽略某些输出结构也能得到良好的结果。虽然基于转移的方法可以使用到丰富的特征，但采用的都是局部搜索策略，容易出现错误传递现象。基于图的方法特征范围较小，但可以进行全局搜索，能够处理长距离依存和非投射现象，并在准确率上高于基于转移的方法。

依存句法分析对句子结构的要求很高，而且真实场景和标注数据分布有明显差异，若采用人工标注，会出现标注困难且容易产生错误数据的问题。Zhang等[14]在扩增数据阶段使用了基于同义词典替换的方式来扩增数据。Xie等[15]使用反向翻译的方式和随机噪声注入的方式来扩充未标注的文本。Coulombe[16]则使用文本表面转换的方式来扩增数据，如缩略语之间的变化，将“It is awesome”变为“It's awesome”。Mixup是Zhang等[17]提出的一种简单而有效的图像增强技术。其思想是将两个随机图像按一定比例组合在一个小批次处理中，生成用于训练的合成实例。随后这种想法被应用于NLP领域中。Guo等[18]将Mixup的思想应用于文本数据增强中，并提出在词级和句子级两种程度上进行Mixup的方法。Zhang等[19]提出一种简单而有效的Seqmix数据增强方法来提高序列标记的标签效率，并将其应用于命名实体识别任务中。但上述数据增强方式大多是基于特定任务的，依存句法分析任务中的数据增强应尽可能使扩增的数据的句子结构没有发生变化，而对应的词语之间词性相同。因此，本文多策略的数据增强考虑采用基于同词性的同义词典替换方法和基于Mixup的数据增强方式。

## 3 同义词替换及 Mixup 数据增强方法

本节将详细介绍所提方法的技术细节。该方法的总体框架如图 2 所示,根据原依存数据以及构造的同义词典,同义词直接替换能够扩充训练数据,使用新数据进行依存解析。Mixup 的方法则在双仿射模型中通过不同阶段的 Mixup 得到新训练数据进行依存解析。本节将详细介绍图 2 中的同词性同义词典的构造方法、基线双仿射模型以及两种数据增强策略的方法。

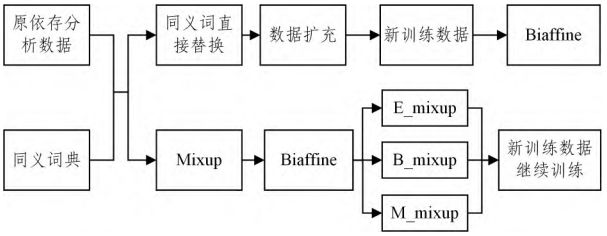


图 2 总体框架图

Fig. 2 General framework

### 3.1 同义词典构造

本文提出的数据增强策略的实现需要依赖同义词典,首先分别构建泰语、越南语以及英语的同义词典,在 Babelnet[1)] 网站爬取所需的同义信息,然后将同义词以及对应的词性组成"单词:[词性 1:同义词,词性 2:同义词]"形式的字典列表,例如,表 1 中泰语的同义词典样例,**ต่อ**有形容词和名词两种词性的同义词,在后续同义词替换过程中,根据原训练数据词语的词性进行相应的选择与替换,即若**ต่อ**在句子中为形容词,则会被替换为**เชื่อม**,若为名词,则会被替换为**ตัวต่อ**。

表 1 泰语同义词典样例

Table 1 Sample of Thai thesaurus

| 原词 | 同义词 1 词性 | 同义词 1 | 同义词 2 词性 | 同义词 2 |
|---|---|---|---|---|
| จริง | ADJ | ที่เป็นจริง, แท้ | — | — |
| คน | NOUN | บุคคล,ประชาชน | VERB | กวน |
| ต่อ | ADJ | เชื่อม | NOUN | ตัวต่อ |

### 3.2 双仿射模型框架

如图 3 所示,本文使用 Dozat 等[8] 提出的双仿射模型。

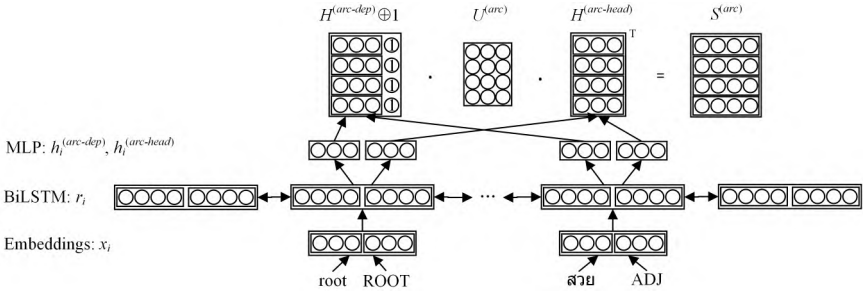


图 3 双仿射模型框架

Fig. 3 Biaffine model framework

模型的输入为单词 $\boldsymbol{w}_i$ 与词性标注 $\boldsymbol{t}_i$ 的拼接 $\boldsymbol{x}_i$,在得到输入特征后,首先通过 BiLSTM 提取到特征 $\boldsymbol{r}_i$,使得每个输入元素都能联系到上下文。之后特征 $\boldsymbol{r}_i$ 经过两个不同的用于降维的多层感知机 MLP 后分别得到特征 $\boldsymbol{h}_i^{(arc\text{-}dep)}$ 和 $\boldsymbol{h}_i^{(arc\text{-}head)}$,如式(1)所示:

$$\begin{aligned}\boldsymbol{h}_i^{(arc\text{-}dep)} &= \boldsymbol{MLP}^{(arc\text{-}dep)}(\boldsymbol{r}_i)\\ \boldsymbol{h}_j^{(arc\text{-}head)} &= \boldsymbol{MLP}^{(arc\text{-}head)}(\boldsymbol{r}_j)\end{aligned} \tag{1}$$

基线模型使用的双仿射注意力机制(biaffine attention)可以看作一个传统的仿射分类器,如式(2)所示,但用一个($\boldsymbol{d}\times\boldsymbol{d}$)的线性变换 $\boldsymbol{H}^{(arc\text{-}head)}\boldsymbol{U}^{(1)}$ 代替权重矩阵 $\boldsymbol{W}$,并且用一个($\boldsymbol{d}\times 1$)的线性变换 $\boldsymbol{H}^{(arc\text{-}head)}\boldsymbol{u}^{(2)}$ 来代替偏置项 $\boldsymbol{b}$。其中,矩阵 $\boldsymbol{H}$ 是 $d$ 个 token 的特征经过 MLP 二次编码出来的特征向量 $\boldsymbol{h}$ 的堆栈形式,其维度为($\boldsymbol{d}\times\boldsymbol{k}$)。$\boldsymbol{U}^{(1)}$ 是($\boldsymbol{k}\times\boldsymbol{k}$)维的中间变换矩阵,$\boldsymbol{u}^{(2)}$ 是($\boldsymbol{k}\times 1$)维的变换矩阵,最终得到式(3)中的分数矩阵 $\boldsymbol{s}_i^{(arc)}$。

$$\boldsymbol{s}_i = \boldsymbol{W}\boldsymbol{r}_i + \boldsymbol{b} \tag{2}$$

$$\boldsymbol{s}_i^{(arc)} = \boldsymbol{H}^{(arc\text{-}head)}\boldsymbol{U}^{(1)}\boldsymbol{h}_i^{(arc\text{-}dep)} + \boldsymbol{H}^{(arc\text{-}head)}\boldsymbol{u}^{(2)} \tag{3}$$

head 和 dep 之间的依存关系数目是确定的,所以使用一个固定类别的双仿射分类器计算 arc 标签得分 $\boldsymbol{s}_i^{(label)}$,如式(4)所示:

$$\boldsymbol{s}_i^{(label)} = \boldsymbol{h}_j^{\mathrm{T}}\boldsymbol{U}^{(1)}\boldsymbol{h}_i + (\boldsymbol{h}_j \oplus \boldsymbol{h}_i)^{\mathrm{T}}\boldsymbol{u}^{(2)} + \boldsymbol{b} \tag{4}$$

其中,$\boldsymbol{h}_j$ 为式(1)得到的 $\boldsymbol{h}_j^{(arc\text{-}head)}$,$\boldsymbol{h}_i$ 为式(1)得到的 $\boldsymbol{h}_i^{(arc\text{-}dep)}$,$\oplus$是拼接 $\boldsymbol{h}_j$ 和 $\boldsymbol{h}_i$。式(4)的第一项是同时已知 $\boldsymbol{i}$ 作为 dep,$\boldsymbol{j}$ 作为 head 情况下的后验概率,第二项是已知 $\boldsymbol{i}$ 或 $\boldsymbol{j}$ 是 arc 两端的后验概率,第三项 $\boldsymbol{b}$ 是偏置,是没有得到 dep,head 信息时的弧标签的先验概率。

### 3.3 同义词替换与 Mixup 数据增强

本文构建了两种数据增强方法,包括基于同词性同义词典直接替换的数据增强方法和基于多种 Mixup 的数据增强方法。

#### 3.3.1 同义词直接替换

受 EDA[20] 数据增强工作的启发,为低资源依存句法语料采用同义词直接替换的方式扩增依存句法训练数据。在同义词直接替换中,考虑到词性对依存句法解析的影响,需要保证替换的同义词保持与原句中原词的词性一致。如图 4 所示,泰语句子**ดอกไม้สวยๆ**(一朵美丽的花)中,**สวย**(美丽的)在同义词典中拥有其对应词性为形容词(ADJ)的同义词**น่ารัก**(可爱的),那么一个泰语句子便可以变成两个句子,从而实现泰语依存句法训练数据的数据增强。其他语言增强方式也是一样。

1) https://babelnet.org/guide

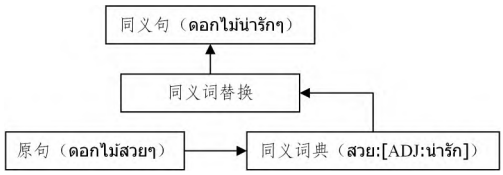


图 4 同词性同义词典直接替换

Fig. 4 Synonym substitution directly with same part-of-speech tagging

### 3.3.2 多种 Mixup 数据增强方法

Zhang 等[17]提出的 Mixup 是图像分类混合方法，能增强训练样本之间的线性表达，是一种简单且数据无关的数据增强方式。

$$\tilde{\boldsymbol{x}}=\lambda \boldsymbol{x}_i+(1-\lambda)\boldsymbol{x}_j \tag{5}$$

$$\tilde{\boldsymbol{y}}=\lambda \boldsymbol{y}_i+(1-\lambda)\boldsymbol{y}_j \tag{6}$$

式(5)和式(6)中的$(\boldsymbol{x}_i,\boldsymbol{y}_i)$和$(\boldsymbol{x}_j,\boldsymbol{y}_j)$是从训练数据中随机抽取的两个样本，$\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 是原始输入向量，$\boldsymbol{y}_i$ 和 $\boldsymbol{y}_j$ 表示对应的标签编码，Mixup 通过结合先验知识，即特征向量的线性插值引起相关标签的线性插值来扩展训练分布。其中 $\lambda$ 遵从 Beta 分布，即 $\lambda\sim\boldsymbol{Beta}(\alpha,\alpha)$，$\alpha\in(0,\infty)$，$\lambda\in[0,1]$。在本文提出的多种 Mixup 方法中，输入向量是单词与词性的拼接，且原词与同义词之间已保证词性一致，所以不需要标签之间再进行 Mixup，因此只需要式(5)即可，Mixup 方法的具体用法会在下文中说明。

Mixup 机制中的数据集如图 5 所示，数据集第 2 列为原词，在原数据集的基础上，若有同义词，则在第 3 列附上同义词；若无同义词，则第 3 列依旧是原词。

```
1  khi khi NOUN  N  _  5  obl  _  _
2  trở về  trở lại VERB  V  _  1  xcomp  _  SpaceAfter=No
3  ,  ,  PUNCT  ,  _  5  punct  _  _
4  hầu hết hầu hết X  X  _  5  advmod  _  _
5  tiếp tục  tiếp tục  VERB  V  _  0  root  _  _
6  làm việc  làm việc  VERB  V  _  5  xcomp  _  _
7  tốt tốt ADJ A  _  6  xcomp  _  SpaceAfter=No
8  .  .  PUNCT  .  _  5  punct  _  _
```

图 5 新数据集样例

Fig. 5 Sample of new data set

如图 6 所示，本文选择在双仿射模型的 3 种不同位置分别使用 Mixup 方式得到虚拟训练数据。

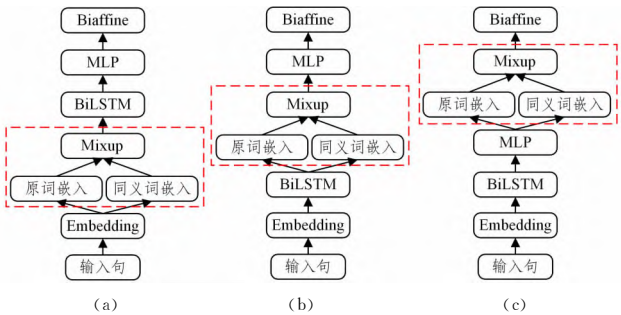


图 6 3 种 Mixup 方式

Fig. 6 Three strategies to Mixup

(1)Embedding 阶段后融入 Mixup(E_Mixup)：双仿射模型的输入为单词与词性的拼接，即 $\boldsymbol{x}_i=\boldsymbol{e}(\boldsymbol{w}_i)\circ\boldsymbol{e}(\boldsymbol{t}_i)$。如图 6(a)所示，在编码阶段后融入 Mixup，即对原词与同义词进行词嵌入后，将二者的词嵌入进行 Mixup 操作。原词嵌入为$\boldsymbol{x}_s=\boldsymbol{e}(\boldsymbol{w}_i)\circ\boldsymbol{e}(\boldsymbol{t}_i)$，同义词嵌入为 $\boldsymbol{x}_t=\boldsymbol{e}(\boldsymbol{d}_i)\circ\boldsymbol{e}(\boldsymbol{t}_i)$。其中 $\boldsymbol{d}_i$ 为 $\boldsymbol{w}_i$ 的同义词，$\boldsymbol{t}_i$ 为词性。Mixup 过程如式(7)所示。

$$\tilde{\boldsymbol{w}}=\lambda \boldsymbol{w}_1+(1-\lambda)\boldsymbol{w}_2 \tag{7}$$

其中，$\boldsymbol{w}_1$ 和 $\boldsymbol{w}_2$ 分别为 $\boldsymbol{x}_s$ 和 $\boldsymbol{x}_t$，$\tilde{\boldsymbol{w}}$ 为虚拟的训练样本。对于没有同义词的单词，$\boldsymbol{w}_1$ 和 $\boldsymbol{w}_2$ 都是 $\boldsymbol{x}_s$。二者 Mixup 结束后，利用新数据进行接下来的 BiLSTM 提取特征、MLP 降维以及双仿射打分的过程。

(2)BiLSTM 阶段后融入 Mixup(B_Mixup)：如图 6(b)所示，通过 Embedding 得到原词与同义词的嵌入 $\boldsymbol{x}_s$ 和 $\boldsymbol{x}_t$ 后，二者均通过 BiLSTM 得到 $\boldsymbol{x}_s$ 的相应特征 $\boldsymbol{r}_i$ 和 $\boldsymbol{x}_t$ 的相应特征 $\boldsymbol{r}_i'$。之后通过 Mixup 将二者融合得到新特征，再进行之后的训练与打分。

$$\tilde{\boldsymbol{w}}=\lambda_i \boldsymbol{w}_1+(1-\lambda_i)\boldsymbol{w}_2 \tag{8}$$

此时式(8)中的 $\boldsymbol{w}_1$ 和 $\boldsymbol{w}_2$ 分别代表 $\boldsymbol{r}_i$ 和 $\boldsymbol{r}_i'$，对于没有同义词的单词，$\boldsymbol{w}_1$ 和 $\boldsymbol{w}_2$ 都是 $\boldsymbol{r}_i$。在进行特征 Mixup 之前，对每一对要进行 Mixup 的特征生成一个 $\lambda$ 值，即 $\boldsymbol{r}_i$ 和 $\boldsymbol{r}_i'$ 对应 $\lambda_i$。

(3)MLP 阶段后融入 Mixup(M_Mixup)：如图 6(c)所示，原词与同义词经过相同的编码与 BiLSTM 过程得到 $\boldsymbol{r}_i$ 和 $\boldsymbol{r}_i'$，$\boldsymbol{r}_i$ 通过两个 MLP 分别得到 $\boldsymbol{h}_i^{(arc\text{-}dep)}$ 和 $\boldsymbol{h}_i^{(arc\text{-}head)}$，$\boldsymbol{r}_i'$ 也通过相同的两个 MLP 得到 $\boldsymbol{h}_i^{(syn\text{-}dep)}$ 和 $\boldsymbol{h}_i^{(syn\text{-}head)}$，之后再通过 Mixup 分别对 dep(修饰词)与 head(中心词)特征进行融合，得到新特征，如式(9)和式(10)所示：

$$\boldsymbol{h}_i^{(arc\text{-}dep)}=\lambda_i \boldsymbol{h}_i^{(arc\text{-}dep)}+(1-\lambda_i)\boldsymbol{h}_i^{(syn\text{-}dep)} \tag{9}$$

$$\boldsymbol{h}_i^{(arc\text{-}head)}=\lambda_i \boldsymbol{h}_i^{(arc\text{-}head)}+(1-\lambda_i)\boldsymbol{h}_i^{(syn\text{-}head)} \tag{10}$$

没有同义词的单词亦是这样的步骤，之后再进行双仿射打分的过程。

本文还在实验中组合使用了 Mixup 的方法，将单独的策略进行两两组合，以更好地利用同义词的信息。

### 3.3.3 损失函数

在预测弧的整个训练过程中，无论在哪一个或哪几个阶段进行了 Mixup 的过程，整个模型计算损失函数都有用到 3 个序列，分别是原句序列 $\boldsymbol{S}_w$、同义句序列 $\boldsymbol{S}_d$ 和 Mixup 序列 $\boldsymbol{S}_m$。在计算损失时，用 3 个序列的损失和作为总的损失，并以此来调整模型。每个损失的产生均使用交叉熵损失函数。

$$\boldsymbol{L}=\sum_{i=1}^{N}\boldsymbol{y}^{(i)}\log\hat{\boldsymbol{y}}^{(i)}+(1-\boldsymbol{y}^{(i)})\log(1-\hat{\boldsymbol{y}}^{(i)}) \tag{11}$$

其中，$\boldsymbol{N}$ 为样本数量，$\boldsymbol{y}$ 为真实值，$\hat{\boldsymbol{y}}$ 为预测输出值。

3 个序列经过 3 次交叉熵损失函数分别得到 $\boldsymbol{L}_{S_w}$，$\boldsymbol{L}_{S_d}$ 和 $\boldsymbol{L}_{S_m}$，总损失为三者加和，即 $\boldsymbol{Loss}^{(arc)}=\boldsymbol{L}_{S_w}+\boldsymbol{L}_{S_d}+\boldsymbol{L}_{S_m}$。同理，预测弧标签的过程也会产生损失 $\boldsymbol{Loss}^{(label)}$，总损失为 $\boldsymbol{Loss}=\boldsymbol{Loss}^{(arc)}+\boldsymbol{Loss}^{(label)}$。

## 4 实验设置

### 4.1 数据集

本文所有实验基于泰语和越南语以及英语的基准数据集，包括 UD treebanks[1)]的泰语数据集、越南语数据集和英语

1) https://universaldependencies.org/

的小规模数据集。每个数据集都有对应的训练集、开发集以及测试集。数据集中的文本已经进行了分词与词性标注,无需人工再次预处理。由于这些数据集的语料数量都很少,为了验证提出方法的有效性,本文从 BabelNet 中构造相关泰语与越南语同义词典。目前,泰语的同义词集大小为 426 条,越南语的同义词集大小为 178 条,英语的同义词集大小为 559 条。

表 2 列出了数据集的统计信息。从表 2 可以看出,通过同义词典替换,泰语的训练数据量扩增了 7 倍左右,越南语的训练数据量扩增了 100 多条,英语的训练数据量扩增了 2 倍左右。

表 2 泰语、越南语和英语依存句法解析数据信息

Table 2 Dependency parsing data information of Thai, Vietnamese and English

| 语言 | 数据集 | 替换前(句) | 替换后(句) |
|---|---|---|---|
| 泰语 | Train | 686 | 4886 |
| | Dev | 161 | — |
| | Test | 153 | — |
| 越南语 | Train | 1399 | 1515 |
| | Dev | 800 | — |
| | Test | 800 | — |
| 英语 | Train | 1781 | 3929 |
| | Dev | 161 | — |
| | Test | 153 | — |

### 4.2 模型参数

对词嵌入采用随机初始化方式,词向量的维度为 100 维。实验中采用反向传播算法进行参数优化。使用 Adam 优化器来训练模型,其学习率初始值为 0.001。采用小批次(mini-batch)的方式训练样本,每个批次大小为 2。模型在训练时不固定迭代数量,而是采用提前停止策略(early-stop)来确定迭代次数。实验的环境为:i5 Intel CPU 以及 NVIDIA RTX 1080Ti GPU。其他的模型参数设置如表 3 所列。

表 3 模型参数

Table 3 Model parameters

| 参数 | 值 | 参数 | 值 |
|---|---|---|---|
| Embedding size | 100 | Words dropout | 0.2 |
| LSTM size | 400 | Embedding dropout | 0.33 |
| Arc MLP size | 500 | LSTM dropout | 0.33 |
| Label MLP size | 100 | Arc MLP dropout | 0.33 |
| LSTM depth | 3 | Label MLP dropout | 0.33 |
| Early_stop_count | 20 | MLP depth | 1 |
| $\alpha$ | 8 | | |

### 4.3 评价指标

本文评价指标采用 UAS 和 LAS[2]。UAS(Unlabeled Attachment Score)表示不带标签依存关系准确率,是正确预测中心词的词的数量在总词数中的比例;LAS(Labeled Attachment Score)表示带标签依存关系准确率,是正确预测中心词以及标签的词的数量在总词数中的比例。相较于 LAS,UAS 仅仅衡量支配词的预测准确率,一般分数更高。

## 5 实验结果与分析

### 5.1 主实验结果

本节详细介绍了所提模型在不同语言数据集、不同数据量以及不同策略上的实验结果和分析。Raw 表示原始数据集,Direct 表示同词性同义词直接替换策略,E_Mixup,B_Mixup 和 M_Mixup 分别表示在不同阶段后 Mixup 的策略,E+B_Mixup,B+M_Mixup 和 E+M_Mixup 分别表示两种策略两两组合的情况。

从表 4 可以看出,针对泰语、越南语和英语,所有的数据增强策略均取得了性能上的提升。如在泰语中应用同义词直接替换方法后,无论是在验证集还是测试集上都有一定的性能提升,在验证集上得到了 3 点以上的提升,在测试集上得到了 2 点左右的提升。在此方法中,英语的测试集与验证集中也有 1 点左右的提升,说明基于同词性同义词典的直接替换是数据增强方式中相对简单有效的方法。针对 Mixup 的不同策略,不同语言都有不同程度上的性能提升,对泰语来说,在验证集结果中,单个阶段 Mixup 的效果优于组合策略,而在测试集中组合策略略胜一筹。

表 4 不同语言、不同策略的实验结果

Table 4 Experimental results of different languages and strategies

| 语言 | 策略 | dev_UAS | dev_LAS | test_UAS | test_LAS |
|---|---|---|---|---|---|
| 泰语 | Raw | 78.29 | 70.42 | 79.46 | 70.52 |
| | Direct | **82.01** | **73.83** | 81.38 | 72.76 |
| | E_Mixup | 81.24 | 73.06 | 81.50 | 72.84 |
| | B_Mixup | 81.10 | 72.95 | 81.78 | 72.70 |
| | M_Mixup | 80.45 | 72.55 | 81.78 | 73.33 |
| | E+B_Mixup | 80.96 | 73.00 | **82.84** | **73.73** |
| | B+M_Mixup | 80.08 | 71.72 | 81.12 | 71.16 |
| | E+M_Mixup | 81.13 | 73.26 | 82.07 | 73.07 |
| 越南语 | Raw | 70.70 | 66.97 | 69.49 | 66.26 |
| | Direct | 69.77 | 66.04 | 68.77 | 65.39 |
| | E_Mixup | 71.31 | 67.46 | **70.10** | 66.74 |
| | B_Mixup | **71.90** | **68.41** | 69.90 | **66.83** |
| | M_Mixup | 71.70 | 68.12 | 69.82 | 66.52 |
| | E+B_Mixup | 71.19 | 67.67 | 69.85 | 66.68 |
| | B+M_Mixup | 71.02 | 67.49 | 69.75 | 66.57 |
| | E+M_Mixup | 71.24 | 67.64 | 69.61 | 66.28 |
| 英语 | Raw | 86.77 | 83.84 | 88.26 | 85.71 |
| | Direct | 87.99 | 84.86 | **89.17** | **86.47** |
| | E_Mixup | 88.10 | 85.05 | 88.97 | 86.15 |
| | B_Mixup | 88.28 | 85.53 | 88.67 | 86.06 |
| | M_Mixup | **88.83** | **86.26** | 88.26 | 85.53 |
| | E+B_Mixup | 88.61 | 85.75 | 89.00 | 86.36 |
| | B+M_Mixup | 88.13 | 84.94 | 88.50 | 86.15 |
| | E+M_Mixup | 87.69 | 84.83 | 88.67 | 86.41 |

对于越南语和英语,组合策略与单个阶段进行 Mixup 的策略性能提升是不相上下的,也从侧面反映 Mixup 策略可以使模型更加平滑,有效提高模型泛化能力,但应用多重 Mixup 方法并不一定得到效果上的叠加。同时本文也尝试过在模型 3 个阶段同时应用 Mixup,但其效果并没有进一步提升,这证明了上述问题。整体来看,无论是验证集,还是测试集,在 UAS 上的性能提升优于 LAS,说明数据增强方式更倾向于对弧进行预测。纵向比较不同语言,由于同义词典的大小规模不同,同义词典的质量参差不齐,扩充出的语料规模也不一

样，所以呈现出的提升效果也不尽相同。

### 5.2 不同 $\alpha$ 的对比结果

在 Mixup 阶段，$\lambda$ 取值服从 Beta 分布，不同的 $\alpha$ 取值会带来不同的 $\lambda$ 结果，$\alpha$ 数值越大，$\lambda$ 越接近 0.5。因此本文对不同 $\alpha$ 取值进行对比实验，以探索 $\alpha$ 取值对实验结果的影响。以泰语中 E_Mixup 策略的结果为例，表 5 列出了 $\alpha$ 取不同值时泰语 E_Mixup 阶段的实验结果。从表 5 中发现，当 $\alpha$ 取值为 8 时，实验结果最好，整体趋势为先升后降，说明 $\lambda$ 的值在 0.3～0.5 范围内浮动时，更有利于原词与同义词之间的比例分布，使得模型的泛化能力得到增强。$\lambda$ 的值过于集中与分散，都不利于模型的学习与调整。

表 5 $\alpha$ 取不同值时泰语的实验结果

Table 5 Experimental results of Thai with different $\alpha$

| $\alpha$ | dev_UAS | dev_LAS | test_UAS | test_LAS |
|---|---|---|---|---|
| 0.5 | 80.22 | 71.44 | 80.87 | 70.93 |
| 1 | 81.02 | 72.09 | 81.52 | 72.07 |
| 8 | 81.24 | 73.06 | 81.50 | 72.84 |
| 20 | 80.68 | 72.07 | 81.32 | 72.41 |
| 50 | 80.65 | 73.17 | 81.24 | 72.33 |

### 5.3 同义词典大小的对比结果

本节介绍了不同大小规模的同义词典的结果变化。

表 6 列出了越南语不同规模的同义词典大小扩增出的训练数据的规模，由表可知训练数据扩增的程度与同义词典增加的程度并不一定成正比。这与词典词语与训练数据的重叠程度有关。同时，表 7 列出了越南语直接替换方法下同义词典的大小对实验结果的影响。随着同义词典规模的增加，依存句法分析的性能也变得越来越好，同义词典的增加导致训练数据扩增，扩增得越多，实验结果越好，说明基于同义词典直接替换的方法可以有效解决未知词问题，有效扩增训练数据规模，从而提升依存句法分析的性能。

表 6 词典大小与训练数据规模

Table 6 Dictionary size and training data size

| 同义词典大小（词） | 训练数据替换前（句） | 训练数据替换后（句） |
|---|---|---|
| 178 | 1399 | 1515 |
| 314 | 1399 | 4474 |
| 434 | 1399 | 4719 |

表 7 直接替换条件下越南语实验结果

Table 7 Experimental results of Vietnamese under direct substitution condition

| Direct | dev_UAS | dev_LAS | test_UAS | test_LAS |
|---|---|---|---|---|
| 178 | 69.77 | 66.04 | 68.77 | 65.39 |
| 314 | 70.38 | 66.75 | 69.09 | 65.65 |
| 434 | 70.55 | 66.81 | 69.12 | 65.70 |

**结束语** 本文提出将多策略数据增强方法应用于依存句法分析任务，通过同义词典替换和多种 Mixup 的数据增强方法，改进了低资源条件下依存句法分析性能。在 3 种语言的小规模 UD 数据集上的实验结果也表明，基于同词性同义词典替换的方法可以缓解未知词对模型的影响，多种 Mixup 的数据增强方法也可以有效缓解模型过拟合的问题，使得模型更加平滑，泛化能力也得到提高。

在下一步研究中，我们将继续提升同义词典的质量，探索原词与同义词之间在依存句法分析任务中的关联关系，进一步探讨如何引入更好的数据增强方式来提高依存句法分析的效果。

## 参 考 文 献


[1] TU K W, LI J. A Survey of Recent Developments in Syntactic Parsing[J]. Journal of Chinese Information Processing, 2020, 34(7): 30-41.

[2] MAO C L, MAN Z B, YU Z T, et al. A Burmese Dependency Parsing Method Based on Transfer Learning[C]// 2020 International Conference on Asian Language Processing (IALP). IEEE, 2020: 92-97.

[3] CHEN D, MANNING C D. A fast and accurate dependency parser using neural networks [C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Stroudsburg, PA: Association for Computational Linguistics. 2014: 740-750.

[4] DYER C, BALLESTEROS M, LING W, et al. Transition-based dependency parsing with stack long short-term memory[J]. arXiv: 1505.08075, 2015.

[5] ANDOR D, ALBERTI C, WEISS D, et al. Globally Normalized Transition-Based Neural Networks [C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016: 2442-2452.

[6] NALLANI S, SHRIVASTAVA M, SHARMA D M. A Simple and Effective Dependency parser for Telugu[C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop. 2020: 143-149.

[7] KIPERWASSER E, GOLDBERG Y. Simple and accurate dependency parsing using bidirectional LSTM feature representations[J]. Transactions of the Association for Computational Linguistics, 2016, 4: 313-327.

[8] DOZAT T, MANNING C D. Deep biaffine attention for neural dependency parsing[J]. arXiv: 1611.01734, 2016.

[9] SINGKUL S, WORARATPANYA K. Thai dependency parsing with character embedding[C]// 2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE). IEEE, 2019: 1-5.

[10] KULMIZEV A, DE-LHONEUX M, GONTRUM J, et al. Deep Contextualized Word Embeddings in Transition-Based and Graph-Based Dependency Parsing-A Tale of Two Parsers Revisited[C]// Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019: 2755-2768.

[11] DELHONEUX M, BALLESTEROS M, NIVRE J. Recursive subtree composition in LSTM-based dependency parsing[C]// Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational

Linguistics,2019:1566-1576.

[12] FALENSKA A,KUHN J. The (non-)utility of structural features in BiLSTM-based dependency parsers [C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019:117-128.

[13] ZHANG Z,MA X,HOVY E. An empirical investigation of structured output modeling for graph-based neural dependency parsing[C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019:5592-5598.

[14] ZHANG X,ZHAO J,LECUN Y. Character-level convolutional networks for text classification[C]// Proceedings of the 28th International Conference on Neural Information Processing Systems. 2015:649-657.

[15] XIE Q,DAI Z,HOVY E,et al. Unsupervised Data Augmentation for Consistency Training[J]. Advances in Neural Information Processing Systems,2020,33:6256-6268.

[16] COULOMBE C. Text data augmentation made simple by leveraging nlp cloud apis[J]. arXiv:1812. 04718,2018.

[17] ZHANG H,CISSE M,DAUPHIN Y N,et al. Mixup:Beyond Empirical Risk Minimization[C]// International Confe-rence on Learning Representations. 2018.

[18] GUO H,MAO Y,ZHANG R. Augmenting data with mixup for sentence classification: An empirical study[J]. arXiv: 1905. 08941,2019.

[19] ZHANG R,YU Y,ZHANG C. SeqMix:Augmenting Active Sequence Labeling via Sequence Mixup[C]// Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020:8566-8579.

[20] WEI J,ZOU K. EDA:Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks[C]// Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019:6382-6388.

**XIAN Yan-tuan**, born in 1981, Ph. D, associate professor, is a member of China Computer Federation. His main research interests include information retrieval and natural language processing.

**XIANG Yan**, born in 1979, Ph.D, associate professor, is a member of China Computer Federation. Her main research interests include text mining and sentiment analysis.

(责任编辑:李亚辉)