$$\begin{cases} 基本 := NP/NP \\ 内容 := NP \\ 基本:NP/NP \ \ 内容:NP \Rightarrow 基本内容:NP \\ 基本内容:NP \end{cases} \tag{1}$$

在公式(1)中,“基本”一词的范畴类型为 NP/NP,表明该词可以向右结合一个名词短语 NP(内容),根据某种规则得到一个 NP(基本内容).这里所说的规则就是指组合范畴语法中定义的一套规则,即前向应用和后向应用规则,形式化表示如下:

a. 前向应用规则:$X/Y:f\ Y:a \Rightarrow X:f(a)$.

b. 后向应用规则:$Y:a\ X/Y:f \Rightarrow X:f(a)$.

上述规则可直观地解释为:X/Y 看做一个映射,这个映射把实例化之后的 $Y:a$ 映成范畴 $X:f(a)$.此外,组合范畴语法还定义了其他两种规则,即函数复合规则和类型提升规则,具体如下:

c. 前向复合规则:$X/Y:f\ Y/Z:g \Rightarrow_B X/Z:\lambda x.f(g(x))$.

d. 后向复合规则:$Y\backslash Z:g\ X\backslash Y:f \Rightarrow_B X\backslash Z:\lambda x.f(g(x))$.

e. 前向类型提升规则:$X:a \Rightarrow_T T/(T\backslash X):\lambda f.f(a)$.

f. 后向类型提升规则:$X:a \Rightarrow_T T\backslash(T/X):\lambda f.f(a)$.

值得注意的是,在类型提升规则中,原子范畴 X 通过类型提升为函子范畴 $T\backslash(T/X)$,这里的 X 不可以是函子范畴类型[1].

1.2 范畴标注

根据上节我们对组合范畴语法的描述,范畴标注的任务就是给句子中的每个词打上合适的范畴标签,这些标签可以是原子类型,也可以是较复杂的函子类型.给定一个句子中每个词的范畴,我们可以根据范畴组合规则把这些范畴自底向上两两组合起来,直至生成一个树.比如,我们有如图 1 所示的范畴树.该范畴树可以根据句子中的每个词的范畴得到,即,我们并不需要定义层次化的树结构,只需定义范畴序列即可.当然,树结构会比序列结构包含更多的信息.由于我们的训练数据来自于 CCG 树库[11],因此,如何充分利用树结构信息来设计分类模型就是我们考虑的重点.

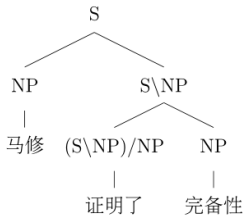


Fig.1 A CCG derivation tree

图 1 组合范畴语法解析树

Clark 和 Curran 提出的范畴标注模型[4]是基于最大熵模型的,该模型的结构如下:

$$p(c \mid h) = \frac{1}{Z(h)} \exp\left[\sum_i \lambda_i f_i(c,h)\right] \tag{2}$$

该模型定义了在给定上下文信息 h 的条件下范畴 c 的概率分布,其中 $f_i(c,h)$ 为从这对 $\langle c,h\rangle$ 中抽取的第 i 维特征,定义为

$$f_j(c,h) = \begin{cases} 1, & 如果当前词是the且c(\text{the}) = NP/N \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

公式(3)表明:如果某个词是 the,对应的标签是 NP/N,那么这条规则成立;反之,则不成立.

该模型存在的问题是,最大熵模型只能挖掘局部的上下文信息.具体来说,就是在范畴 c 的不确定性完全由

上下文信息 h 所控制,这显然是不合理的.假若我们有两个范畴 c_1,c_2,彼此之间互相关联,但在 c_1,c_2 都未知的前提下,h_1 是不可能包含任何关于 c_2 的信息的.这就是点预测模型的不足之处.

序列预测模型能够解决这种不足,相关的工作主要是基于半监督隐马尔可夫模型的范畴标注[6],模型的生成式过程为

$$\left.\begin{array}{l}\text{先验分布:}\\ \quad \phi_t \sim Dirichlet(\alpha_\phi,\phi_t^0),\forall t\in T\\ \quad \pi_t \sim Dirichlet(\alpha_\pi,\pi_t^0),\forall t\in T\\ \text{第}i\text{个标签的生成过程:}\\ \quad x_i \mid y_i \sim Categorial(\phi_{y_i})\\ \quad y_{i+1} \mid y_i \sim Categorial(\pi_{y_i})\end{array}\right\} \tag{4}$$

该模型同样也存在一些问题:首先,范畴标签数目过多,导致 π_t 维度很高,解码时复杂度会很高;其次,用转移矩阵 $y_{i+1}|y_i$ 来描述范畴与范畴之间的联系是不恰当的,因为畴之间的组合主要是根据语法定义的规则,而并非根据最大似然估计得到的离散分布;再者,本来多标签预测因为监督信息不足会导致准确率低,再用半监督去做就会使得原本稀有的监督信息缺失,进一步降低性能.

神经语言模型类似于最大熵模型,也是通过上下文来预测当前词的标签,其区别在于利用嵌入层把特征抽象为一种分布式的表示,这种连续型表示可看做是一种“软”表示,相对于最大熵模型离散化的“硬”表示,优势在于能够泛化一些集外词和低频词.此外,神经网络可设置多个隐含层,对输入进行不同水平的抽象,也可看做是一个深层的最大熵模型.其好处在于不需要手工定义特征模板,能够通过隐含层自动学习到特征的抽象表示.此外,还可提供类似于序列模型的解码功能以及灵活设置输入、输出的特点.具体实施细节将在下一节详细介绍.

2 神经范畴标注模型

在自然语言处理应用中,词是处理的基本单元,如何表示词也是一个基本问题.从基于词频、共现矩阵分解的潜在语义分析(latent semantic analysis,简称 LSA)[12]、基于概率矩阵分解的潜在语义分析(probabilistic latent semantic analysis,简称 PLSA)[13]、主题模型(latent dirichlet allocation,简称 LDA)[14]到基于词表示的神经网络模型[15−18],以及近年提出的全局向量表示[19,20],词的表示越来越细致.

我们扩展词向量的思想,提出了一种基于词性向量和范畴向量的范畴标注预测模型.该模型基于 Bengio 等人[15]提出的神经语言模型,目的是希望网络的输出与目标尽可能地接近,其度量采用如下的交叉熵损失函数:

$$\ln P(\boldsymbol{t}^i \mid \boldsymbol{x}_{ct}^i,\boldsymbol{p}_{ct}^i,\boldsymbol{c}_{ct}^{-i},\boldsymbol{w}) = \sum_{k=1}^{K} t_k^i \ln y_k^i \tag{5}$$

其中,

- $\boldsymbol{t}^i \in \{(t_1^i,...,t_K^i) \mid t_k^i=1,t_{-k}^i=0,k\in\{1,2,...,K\}$ 表示当前第 i 个词的标签分布,即,若当前词的真实标签是 k,则有 $t_k^i=1,t_{-k}^i=0$;
- $\boldsymbol{x}_{ct}^i$ 表示在窗口 ct 下所取的词向量集合;
- $\boldsymbol{p}_{ct}^i$ 表示在窗口下的词性向量集合;
- $\boldsymbol{c}_{ct}^{-i}$ 表示在窗口下除去标签 i 位置的范畴向量集合.

比如,我们可以取 $\boldsymbol{x}_{ct}^i := \{x_i,x_{i+1}\},\boldsymbol{p}_{ct}^i := \{p_i,p_{i+1}\},\boldsymbol{c}_{ct}^{-i} := \{c_{i+1}\}$,这意味着用当前词、当前词的词性、下一个词及其词性和下一个范畴来预测当前词的范畴.为了简化标记,我们用 $\boldsymbol{x},\boldsymbol{p},\boldsymbol{c}_{-i}$ 来替代 $\boldsymbol{x}_{ct}^i,\boldsymbol{p}_{ct}^i$ 和 $\boldsymbol{c}_{ct}^{-i},y_k^i := y_k(\boldsymbol{x},\boldsymbol{p},\boldsymbol{c}_{-i},\boldsymbol{w})$ 为神经网络的输出,其中,$\boldsymbol{w}$ 为网络的参数集合.我们以两层感知机模型为例:

$$y_k(\boldsymbol{x},\boldsymbol{p},\boldsymbol{c}_{-i},\boldsymbol{w}) = \frac{\exp[f_k(\boldsymbol{x},\boldsymbol{p},\boldsymbol{c}_{-i},\boldsymbol{w})]}{\sum_{k'}\exp[f_{k'}(\boldsymbol{x},\boldsymbol{p},\boldsymbol{c}_{-i},\boldsymbol{w})]} \tag{6}$$

其中,$f_k(\boldsymbol{x},\boldsymbol{p},\boldsymbol{c}_{-i},\boldsymbol{w})$ 为输出层第 k 个神经元的输入.因而,

$$f_k(\boldsymbol{x},\boldsymbol{p},\boldsymbol{c}_{-i},\boldsymbol{w})=b_k^o+\sum_j w_{kj}^o\sigma(z_j^h) \tag{7}$$

其中,b_k^o 表示输出层第 k 个节点的偏置,w_{kj}^o 表示隐含层第 j 个节点到输出层第 k 个节点的权重,z_j^h 是隐含层第 j 个节点的输入.同时,我们有,

$$z_j^h=b_j^h+\sum_i u_{ji}^h x_i+\sum_t v_{jt}^h p_t+\sum_l w_{jl}^h c_l \tag{8}$$

其中,u_{ji}^h 表示词向量层第 i 个神经元到隐含层第 j 个神经元的权重;v_{jt}^h 是词性向量层第 t 个神经元到隐含层第 j 个神经元的权重;w_{jl}^h 是范畴向量层第 l 个神经元到隐含层第 j 个神经元的权重;b_j^h 表示隐含层第 j 个神经元的偏置;x_i,p_t,c_l 分别为词向量、词性向量和范畴向量的第 i 维、第 t 维和第 l 维.

2.1 训 练

我们的训练数据集由 L 个句子组成,设每个句子的长度为 n^l,l=1,…,L,我们可以抽取到如下的训练数据:

$$\boldsymbol{D}:=\{\boldsymbol{x}_i^l,\boldsymbol{p}_i^l,\boldsymbol{c}_{-i}^l,t_i^l\}_{i=1}^{n^l},l=1,...,L.$$

其中,$\boldsymbol{x}_i^l,\boldsymbol{p}_i^l,\boldsymbol{c}_{-i}^l$ 作为输入,t_i^l 为预测标签.训练的目标是最小化公式(5)的误差函数.可以通过反向传播算法来更新网络权重 $\boldsymbol{w}=\{\boldsymbol{b}^h,\boldsymbol{w}^h,\boldsymbol{b}^o,\boldsymbol{w}^o\}$、词向量 $\boldsymbol{x}$、词性向量 $\boldsymbol{p}$ 以及范畴向量 $\boldsymbol{c}_{-i}$,更新过程如算法 1 所示.

算法 1. 范畴标注模型的训练过程(第 i 个词,学习率为 ε).

(1) 前馈过程

(a) 把第 i 个词的 $\boldsymbol{word}_{ct}$ 特征、上下文词性 $\boldsymbol{pos}_{ct}$ 和上下文范畴 $\boldsymbol{cat}_{ct}$ 分别通过映射表 T 映射为范畴向量和词向量:$\boldsymbol{x}\leftarrow T(\boldsymbol{word}_{ct}),\boldsymbol{p}\leftarrow T(\boldsymbol{pos}_{ct}),\boldsymbol{c}\leftarrow T(\boldsymbol{cat}_{ct})$.

(b) 计算隐含层的输入、输出:$\boldsymbol{z}^h\leftarrow\boldsymbol{b}^h+\boldsymbol{U}^h\boldsymbol{x}+\boldsymbol{V}^h\boldsymbol{p}+\boldsymbol{W}^h\boldsymbol{c},\boldsymbol{a}^h\leftarrow\sigma(\boldsymbol{z}^h)$.

(c) 计算输出层的输入、输出:$\boldsymbol{f}\leftarrow\boldsymbol{b}^o+\boldsymbol{W}^o\boldsymbol{a}^h,\boldsymbol{y}\leftarrow softmax(\boldsymbol{f})$.

(d) 计算输出与真实值的误差:$C=\ln P(\boldsymbol{D}|\boldsymbol{w})$.

(2) 误差反传过程

(a) 计算输出层残差:$\boldsymbol{\delta}^o\leftarrow\nabla_f C\cdot\sigma'(\boldsymbol{z}^o)$、隐含层残差 $\boldsymbol{\delta}^h=\boldsymbol{W}^{oT}\boldsymbol{\delta}^o\cdot\sigma'(\boldsymbol{z}^h)$.

(b) 更新隐含层到输出层的权重:$\boldsymbol{W}^o\leftarrow\boldsymbol{W}^o-\varepsilon\boldsymbol{a}^h\boldsymbol{\delta}^o,\boldsymbol{b}^o\leftarrow\boldsymbol{b}^o-\varepsilon\boldsymbol{\delta}^o$.

(c) 更新嵌入层到隐含层的权重:$\boldsymbol{U}^h\leftarrow\boldsymbol{U}^h-\varepsilon\boldsymbol{x}\boldsymbol{\delta}^h,\boldsymbol{V}^h\leftarrow\boldsymbol{V}^h-\varepsilon\boldsymbol{p}\boldsymbol{\delta}^h,\boldsymbol{W}^h\leftarrow\boldsymbol{W}^h-\varepsilon\boldsymbol{c}\boldsymbol{\delta}^h$.

(d) 更新词向量、词性向量和范畴向量:$\boldsymbol{x}\leftarrow\boldsymbol{x}-\varepsilon\boldsymbol{\delta}^h\boldsymbol{V}^{hT},\boldsymbol{p}\leftarrow\boldsymbol{p}-\varepsilon\boldsymbol{\delta}^h\boldsymbol{U}^{hT},\boldsymbol{c}\leftarrow\boldsymbol{c}-\varepsilon\boldsymbol{\delta}^h\boldsymbol{W}^{hT}$.

2.2 预 测

传统的预测方法是:在训练好神经网络之后,通过前馈传播得到网络的输出 $\boldsymbol{y}(\boldsymbol{x},\boldsymbol{p},\boldsymbol{c}_{-i},\boldsymbol{w})$,再取 *n-best* 作为标签 t_i 的预测结果.但在范畴标注任务中,我们是对整句话打上范畴标签,因而在预测当前词的时候,需要知道它的历史范畴标签.在测试数据集中,这种标签我们是不知道的,因此,我们需要从句子的一端开始,通过束搜索[21,22]的方式不断解码,并把前一个词的标注结果当作特征加入到预测当前词的输入中,标注过程如图 2 所示.

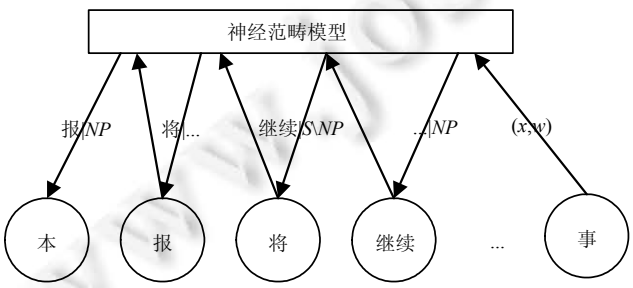


Fig.2 Sequential decoding in neural categorical model, beginning at the last word '事'

图 2 神经范畴模型的序列预测,从句尾词"事"向前逐个预测

这样,整个句子通过循环解码方式就能得到其中每个词的标签,形式化表示为

$$P(\boldsymbol{c}\mid \boldsymbol{x},\boldsymbol{p},\boldsymbol{w})=\prod_{i=n}^{1}P(c_i\mid \boldsymbol{x}_i,\boldsymbol{p}_i,\hat{c}_{i+1},\boldsymbol{w}) \tag{9}$$

其中,$\hat{c}_{-i}$ 表示的是 c_{-i} 的预测值,束搜索的宽度设置为 1.我们也可以选用从前往后解码的方式,其效果是类似的.在实验中,选择从后往前解码的原因是,在训练数据集中,最后一个词是 *NP* 的频率比较大,由于对 *NP* 的识别是很准确的,因此,基于 *NP* 再预测前一个词的范畴也较为可靠.

2.3 预训练

之前的词向量、词性向量和范畴向量都是随机初始化的,针对范畴标注训练语料稀缺的问题,我们可以利用大量无监督文本信息预训练得到这些向量的初始化值,以加入到模型中.对于词向量而言,能够利用大量无标注数据学习到词的分布式表示(distributed representation);对于词性向量和范畴向量而言,通过预训练能够达到类似于规则的效果.

由于预训练这些向量的过程类似,这里,我们只讨论范畴向量的预训练.我们希望预训练之后的范畴向量能够体现范畴中的相互依赖关系,从而有效去除范畴之间的关联信息.下面我们着重介绍如何训练范畴向量.

根据第 1 节的介绍我们知道,范畴之间可以通过应用、复合等方式进行组合.我们希望通过预训练来体现诸如(X/Y,Y,X)这 3 种标签之间的内在联系,即我们希望有:

$$P(X|\langle X/Y,Y\rangle)=1 \tag{10}$$

上式反映了 X/Y $Y\Rightarrow X$ 的组合规则.通过预训练范畴表示,我们也希望我们的网络能够模拟出这些规则,即,输入 X/Y 和 Y,输出为 X.为此,我们从组合范畴语法解析树中抽取出这些训练数据,用特定的神经网络来训练得到范畴向量,形式化表示为

$$\ln P(\boldsymbol{t}^p\mid c_l,c_r)=\sum_{k=1}^{K}t_k^p\ln y_k^p \tag{11}$$

其中,$\boldsymbol{t}^p$ 表示给定左范畴 c_l 和右范畴 c_r 之后的组合范畴分布.在以上例子中,我们有 c_l:=X/Y,c_r:=Y 和 t_k^p := X.值得注意的是,这里的 c_l 和 c_r 都是范畴的分布式表示.我们通过神经网络学习到这种表示,用作范畴向量的初始化.学习过程与范畴标注模型是类似的.此外,在上式中,$\boldsymbol{t}^p$ 表示 c_l,c_r 通过规则(11)组合之后的概率分布.通过对该网络进行训练,我们就能学习到范畴的分布式表示,而这种分布式表示暗含了标签之间的依赖关系.

3 实 验**

为了评价模型的性能,我们采用的训练语料有两类:一类是中文的清华 CCG 树库[23,24],另一类是英文的 CCG Bank 语料[11].由于中文语料库没有词性标签,我们利用斯坦福词性标注器[25]对该语料库进行词性标注后使用.我们把训练数据分为 10 份,其中,8 份用作训练,1 份用于开发集调参,1 份用于测试.在英文语料库上,我们采用标准划分[11],选择 wsj02-21 作为训练数据,wsj00 用作调参,wsj23 用作测试.这两类语料训练集的统计指标见表 1.

Table 1 Statistics for the used training corpora

表 1 训练语料库的统计数据

	解析树数目	范畴数目	词性数目	词汇数目
中文语料库	6 557	735	32	17 631
英文语料库	36 904	1 286	49	44 209

3.1 模型性能比较

在实验对比上,我们选择了 3 种模型:其一是 Clark 和 Curran(C&C)模型[4],该模型是开源的,在语料库上经重新训练使用;其二是 NLTK 的 ME 工具包[25],该工具包原本是作词性标注的,我们将其移植到该范畴标注问题上;

** 模型代码和测试脚本地址:https://github.com/fishiwhj/Neural-Category-Tagging-Model

其三是 Lewis 和 Steedman(L&C)的结果[8].

在参数选择上,词向量和词性向量窗口大小都为 6,范畴向量窗口大小为 1.词向量为 150 维,词性向量为 50 维,范畴向量为 100 维,隐含层节点数为 300.束搜索的宽度为 1.我们分别对词汇、词性和范畴进行了预训练,其中,词汇的预训练我们采用 Collobert 等人的方法[16],词性和范畴的预训练见第 2.3 节.

Table 2 Comparing the accuracies of several categorical tagging models

表 2 几种模型的范畴标注性能对比

	汉语准确率(%)	英语准确率(%)
C&C	80.2	91.5
NLTK ME	75.35	85.28
L&S	–	91.3
NCT(Point)	82.33	92.26
NCT(Beam)	82.52	92.51

在表 2 中,NCT(Point)是在解码的时候不考虑上一个词的标签的模型,结构与 L&S 是类似的.区别是加入了词性表示层和预训练的词向量、词性向量和范畴向量.NCT(Beam)是在上一个模型基础上加入束搜索后的模型,从结果可以看出,加入范畴标签的信息对分类性能是有帮助的.

3.2 预训练

由于嵌入层的预训练对模型性能有很大影响,我们比较了词汇预训练、词性预训练和范畴预训练的组合对模型性能的影响,见表 3.

Table 3 Comparison of NCT models with different pre-trainings: words, part-of-speeches and categories

表 3 词汇预训练、词性预训练和范畴预训练对范畴标注模型性能的影响

	汉语准确率(%)	英语准确率(%)
NCT(random)	81.53	91.52
NCT+PretW	82.15	92.12
NCT+PretWP	82.19	92.2
NCT+PretWC	82.36	92.37
NCT+PretWPC	82.52	92.51

在表 3 中,NCT(random)表示随机初始化的范畴标注模型,初始化分布用的是均值为 0,方差为 0.01 的正态分布.NCT+PretW 表明加入预训练词向量后的模型,其余的两种嵌入向量则随机初始化.类似地,NCT+PretWPC 是把词、词性和范畴全部通过预训练进行初始化之后的模型.它们预训练方法见上一节.从结果可以看出,预训练嵌入向量对模型性能的提升有很大的帮助.特别地,词性向量和范畴向量的预训练也会提升性能,表明预训练得到的嵌入表示要优于随机初始化表示.

3.3 分布式表示维度的影响

由于我们用分布式表示来编码词、词性和范畴单元,不同维度就决定了表示能力的强弱.显然,如果表示维度过低,就会造成神经网络所抽取的特征过少,从而不能完整地表示要区分的对象;相反,如果维度过高,也会造成多个神经元的功能冗余,增加不必要的计算量.

在表 4 中,实验语料为英文的 CCG Bank,窗口大小固定为 6.可以看出,词向量、词性向量和范畴向量维度分别为 150,50 和 100 时效果最好.直观上看,类别数目越多,所要表示该类别的特征数就越多,相应的维度就越大.

Table 4 Comparison of the models with different embedding sizes

表 4 不同分布式表示维度下模型性能的比较

词汇	词性	范畴	范畴准确率(%)
50	50	50	89.62
100	50	50	91.37
150	50	100	92.26
200	200	250	92.21

3.4 窗口大小的影响

由于我们是通过开窗口来确定上下文,因此,窗口的大小对模型性能影响是很大的.表 5 列出了不同窗口大小的影响,我们的实验数据集为英文 CCG Bank,固定词向量、词性向量和范畴向量维度分别为 150,50 和 100.

Table 5 Comparison of the models with different window sizes

表 5 不同特征窗口大小下模型性能的比较

词汇/词性	范畴	范畴准确率(%)
2	1	90.35
4	1	91.69
6	1	92.26
8	1	92.23

表 5 表明,在词汇和词性窗口大小为 6、范畴窗口大小为 1 时效果最好,增加和减小窗口都会使得性能下降.原因可能是,窗口开得过小,会导致有些对分类有用的特征没有包含进来;窗口开得过大则会引入噪声,导致分类准确率下降.

3.5 预训练范畴向量分析

从表 3 中,我们可以观察到预训练范畴向量对模型性能的影响.配置信息与上一节实验是类似的,只不过这里只考虑范畴向量.我们也分别在中文和英文两个语料库上做了实验,我们分别从这两个树库上抽取所需要的训练数据集.在中文上我们抽取了 456 947 个范畴对作为训练语料,英文上抽取了 1 102 975 个范畴对作为训练语料.实验设计为:给定两个能够组合的范畴作为输入,检验网络输出是否与组合规则输出一致.实验结果见表 6.

Table 6 Accuracies of categorical compositions using pre-trained embeddings

表 6 预训练范畴向量模拟规则的准确率

组合规则	汉语准确率(%)	英语准确率(%)
前向应用(>)	99.17	99.85
后向应用(<)	98.83	98.28
前向复合(>***B***)	98.67	99.63
后向复合(<***B***)	98.25	99.59

表 6 的实验结果说明,预训练后的范畴向量确实能够反映出范畴对三者之间的关系.因此,通过预训练之后得到的范畴表示比随机初始化的表示效果要好.此外,该实验结果还表明,利用神经网络可以学习到类似于规则的知识.这一点也反映了神经网络较灵活的特性,可以模拟一些基于规则的模型.

3.6 错误分析

训练好的范畴标注模型在不同数据集上表现出不同的效果:

- 在汉语中,大多容易把 *NP* 和 *S\NP* 混淆.经分析得知,这些范畴为 *S\NP* 的词,有时候在句中也表现为 *NP*,比如总结(*S\NP*)经验教训和工作总结(*NP*)、积累(*S\NP*)经验和财富的积累(*NP*)等.或许在汉语中,*NP* 和 *S\NP* 的界限不是那么严谨.
- 在英语中,容易把(*S\NP*)/(*S\NP*)误识别为(*S\NP*)/*NP*,前者一般是助动词或系动词,后者一般为及物动词.由于助动词或系动词有的时候也具有范畴(*S\NP*)/*NP*,因此,把它们区分开来可能需要借助更多的信息,比如在句子中的位置等.

4 总结和展望

神经网络能够通过隐含层来分布式表示词汇、词性和范畴特征,避免了手工定义特征模板的工作.通过拼接词向量层、词性向量层和范畴向量层来作为输入特征,并通过反向传播算法学习到词、词性和范畴的分布式表示,并把预训练得到的词向量、词性向量和范畴向量加入到模型中,可以提升模型的泛化能力.此外,在解码过程中利用束搜索的思想,能够利用上一个词的标签信息,循环地进行解码.

预训练范畴向量只考虑了函数应用和函数复合规则,类型提升规则并没有考虑.以后可加入对该规则的模拟.在模型上,我们相当于一个两层感知机模型,或许可以利用贝叶斯神经网络模型,针对组合范畴语法的特点设计合适的先验.最后,范畴标注主要是为了解析器服务的,也可以把工作进一步扩展到组合范畴语法的解析器上,比如考虑如何利用神经网络设计解析器.

致谢 感谢清华大学信息技术研究院的周强老师给予的支持与建议.

References:

[1] Steedman M, Baldridge J. Combinatory categorial grammar. In: Non-Transformational Syntax: Formal and Explicit Models of Grammar. Wiley-Blackwell, 2011. [doi: 10.1002/9781444395037.ch5]

[2] Clark S, Curran JR. The importance of supertagging for wide-coverage CCG parsing. In: Proc. of the 20th Int'l Conf. on Computational Linguistics. Association for Computational Linguistics, 2004. 282−288. [doi: 10.3115/1220355.1220396]

[3] Steedman M. The Syntactic Process. Cambridge: MIT Press, 2000.

[4] Clark S, Curran JR. Wide-Coverage efficient statistical parsing with CCG and log-linear models. Computational Linguistics, 2007, 33(4):493−552. [doi: 10.1162/coli.2007.33.4.493]

[5] Auli M, Lopez A. Training a log-linear parser with loss functions via softmax-margin. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011. 333−343.

[6] Baldridge J. Weakly supervised supertagging with grammar-informed initialization. In: Proc. of the 22nd Int'l Conf. on Computational Linguistics, Vol.1. Association for Computational Linguistics, 2008. 57−64.

[7] Garrette D, Dyer C, Baldridge J, Smith NA. Weakly-Supervised Bayesian learning of a CCG supertagger. In: Proc. of the CoNLL. 2014.

[8] Lewis M, Steedman M. Improved CCG parsing with Semi-supervised Supertagging. Trans. of the Association for Computational Linguistics, 2014,2:327−338.

[9] Bangalore S, Joshi AK. Supertagging: An approach to almost parsing. Computational Linguistics, 1999,25(2):237−265.

[10] Curran JR, Clark S, Vadas D. Multi-Tagging for lexicalized-grammar parsing. In: Proc. of the 21st Int'l Conf. on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2006. 697−704. [doi: 10.3115/1220175.1220263]

[11] Hockenmaier J, Steedman M. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. Computational Linguistics, 2007,33(3):355−396. [doi: 10.1162/coli.2007.33.3.355]

[12] Deerwester SC, Dumais ST, Landauer TK, Harshman R. Indexing by latent semantic analysis. Journal of the American Society for Information Science, 1990,41(6):391−407. [doi: 10.1002/ (SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9]

[13] Hofmann T. Unsupervised learning by probabilistic latent semantic analysis. Machine Learning, 2001,42(1-2):177−196. [doi: 10.1023/A:1007617005950]

[14] Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. Journal of Machine Learning Research, 2003,3:993−1022.

[15] Bengio Y, Ducharme R, Vincent P, Jauvin C. A neural probabilistic language model. Journal of Machine Learning Research, 2003,3: 1137−1155.

[16] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. The Journal of Machine Learning Research, 2011,12:2493−2537.

[17] Mnih A, Hinton G. Three new graphical models for statistical language modelling. In: Proc. of the 24th Int'l Conf. on Machine Learning. ACM Press, 2007. 641−648. [doi: 10.1145/1273496.1273577]

[18] Mikolov T, Karafiát M, Burget L, Cernocký J, Khudanpur S. Recurrent neural network based language model. In: Proc. of the 11th Annual Conf. of the Int'l Speech Communication Association (INTERSPEECH 2010). Makuhari, 2010. 1045−1048.

[19] Huang EH, Socher R, Manning CD, NG AY. Improving word representations via global context and multiple word prototypes. In: Proc. of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers, Vol.1. Association for Computational Linguistics, 2012. 873−882.

[20] Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proc. of the Empiricial Methods in Natural Language Processing (EMNLP 2014). 2014. 1532−1543.

[21] Tillmann C, Ney H. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. Computational Linguistics, 2003,29(1):97−133. [doi: 10.1162/089120103321337458]

[22] Toutanova K, Klein D, Manning CD, Singer Y. Feature-Rich part-of-speech tagging with a cyclic dependency network. In: Proc. of the 2003 Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Vol.1. Association for Computational Linguistics, 2003. 173−180. [doi: 10.3115/1073445.1073478]

[23] Zhou Q. Automatic translation from TCTbank to CCGbank: Ver 3.0. Technical Report, Beijing: Center for Speech and Language Technologies, Research Institute of Information Technology, Tsinghua University, 2011 (in Chinese).

[24] Zhou Q. Annotation scheme for Chinese treebank. Journal of Chinese Information, 2004,18(4):1−8 (in Chinese with English abstract). [doi: 10.3969/j.issn.1003-0077.2004.04.001]

[25] Bird S, Klein E, Loper E. Natural Language Processing with Python. O'Reilly Media, Inc., 2009.

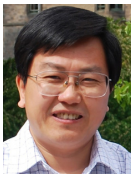
附中文参考文献:

[23] 周强.句法树库TCT到CCG bank的自动转换:设计规范Ver 3.0.科技报告,北京:清华大学信息技术研究院语音和语言技术中心,2011.

[24] 周强.汉语句法树库标注体系.中文信息学报,2004,18(4):1−8. [doi: 10.3969/j.issn.1003-0077.2004.04.001]

吴惠甲(1987－),男,安徽阜阳人,硕士,主要研究领域为自然语言处理,机器学习.

张家俊(1983－),男,博士,副研究员,CCF专业会员,主要研究领域为自然语言处理,机器翻译.



宗成庆(1963－),男,博士,研究员,博士生导师,CCF杰出会员,主要研究领域为自然语言处理,机器翻译,情感分类.