基于词性与词序的相关因子训练的 word2vec 改进模型

潘 博[1],于重重[1],张青川[1],徐世璇[2],曹 帅[1]
(1. 北京工商大学计算机与信息工程学院,北京 100048;2. 中国北京社会科学院民族学与人类学研究所,北京 100081)

摘 要: 词性是自然语言处理的基本要素,词语顺序包含了所传达的语义与语法信息,它们都是自然语言中的关键信息. 在 word embedding 模型中如何有效地将两者结合起来,是目前研究的重点. 本文提出的 Structured word2vec on POS 联合了词语顺序与词性两种信息,不仅使模型可以感知词语位置顺序,而且利用词性关联信息来建立上下文窗口内词语之间的固有句法关系. Structured word2vec on POS 将词语按其位置顺序定向嵌入,对词向量和词性相关加权矩阵进行联合优化. 实验通过词语类比、词相似性任务,证明了所提出的方法的有效性.

关键词: word embedding; 词性; 相关权重; 词序; word2vec
中图分类号: TP3-0 **文献标识码:** A **文章编号:** 0372-2112 (2018)08-1976-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2018.08.024

The Improved Model for word2vec Based on Part of Speech and Word Order

PAN Bo[1], YU Chong-chong[1], ZHANG Qing-chuan[1], XU Shi-xuan[2], CAO Shuai[1]
(1. *Department of Computer and Information Engineering, Beijing Technology and Business University, Beijing* 100048, *China*;
2. *Academy of Social Sciences, Institute of Ethnology and Anthropology, Beijing* 100081, *China*)

Abstract: Part of speech(POS) is the basic element of Natural Language Processing(NLP), word order consists of its conveyed semantic and syntax information, both are the key information of language. There is still lack of such a word embedding model that combines the two together as the influential element. This paper presents the Structured Word2vec on POS that linked the two information of word order and POS together, not only enables the model to sense the words position and order, but alsouse the POS information to establish the inherent syntactic relation between words in the context window. Structured Word2vec on POS is capable to directionally embed the words into context window according to their position, and optimizes the word vector and POSrelevance weight matrix. Experiment through word analogy, word similarity task proved the effectiveness of our method.

Key words: word embedding; part of speech; relevance weights; word order; word2vec

1 引言

语言的语义向量空间模型用实值向量表示每个词语,而词向量可以作为许多应用中的特征,例如文献分类[1],自动问答[2],命名实体识别[3]和形态相关词解析[4].

在词语类比评估方法[5]中,许多早期的基于矩阵分解(Matrix Factorization)的词语表示模型[7,8]显示出其不足:两个词传达的语义相关性单一地取决于它们共同出现次数. 针对这种缺点,Lebret 和 Collobert[9]提出以 Hellinger PCA(HPCA)的形式进行平方根型转化,但提升效果仍然有限.

近年,Bengio 提出的神经网络语言模型(Neural Network Language Model, NNLM)[10]逐渐受到研究者们的关注与重视. 研究者们将其应用于自然语言处理领域:如循环神经网络语言模型(Recurrent Neural Networkslanguage model, RNNLM)[11,12]. NNLM 与 RNNLM 模型的缺陷在于结构过于复杂,其中非线性的隐层带来大量的计算. 针对这个问题, Mikolov[13] 提出了 word2vec 的两种简化的线性模型: Continuous Bag-of-Words Model(CBOW)和 Continuous Skip-gram(CSG). 在

收稿日期:2017-04-18;修回日期:2017-08-23;责任编辑:蓝红杰
基金项目:教育部人文社会科学研究与规划基金(No. 16YJAZH072);国家社会科学基金重大项目(No. 14ZDB156);北京自然基金重点项目 B 类(No. KZ201410011014)

CBOW 与 CSG 的线性结构基础上,Kavukcuoglu 等人[14]提出了相似模型 vLBL 和 ivLBL. Levy 等人[15]提出基于 PPMI 度量的 explicit word embeddings 模型. Jeffrey 等人[16]在 2014 年提出了一种基于全局信息的词语表示模型 GloVe ,其建立 word-word 同窗共现计数矩阵,从而利用矩阵进行全局优化.

Word2vec 与 GloVe 的模型不足之处在于:(1)对于词语顺序信息不敏感. (2)无法利用词性关联信息. 针对这两个问题的改进模型中:Wang 等人[17]提出词序定向嵌入模型 Structured word2vec 可以有效利用词语顺序信息,大幅提高模型在语法相关任务中表现效果;Liu 等人[18]通过引入词性相关权重(POS Relevance Weights),充分利用词语的词性关联信息进行建模.

本文的主要工作是基于 Wang[17]与 Liu[18]的方法,提出 Structured word2vec on POS(SWP)模型,该模型不仅对于词语顺序有较高敏感程度,而且利用词性关联信息对上下文窗口内词语之间的固有句法关系进行建模.. 实验证明:在多个词语相似性任务以及词语类比任务中,本文提出模型比其他当前先进的 word embedding 模型的表现效果更好.

2　原模型

本节将对 CWindow、SSG 与 PWE 三种结构的 word embedding 原模型的相关工作进行回顾.

2.1　CWindow[17]

CWindow 模型基于 CBOW 做出改进,它定义不同的输出预测矩阵 $\boldsymbol{O}\in R^{|V|\times 2cd}$,结构如图 1. 在投影层中按照上下文词语出现顺序所嵌入串联的 $2c\times d$ 维向量 $\boldsymbol{x}_{word(t)}=[\boldsymbol{v}(word(t-c)),\cdots,\boldsymbol{v}(word(t-1)),\boldsymbol{v}(word(t+1)),\cdots,\boldsymbol{v}(word(t+c))]$,并将其投影到输出层. 因为矩阵 $\boldsymbol{O}$ 为每个相对位置定义用于词语嵌入的参数. 该模型参照 Collobert 等人[19]描述的基于窗口的模型,并在进行最终预测之前将词语嵌入的向量投影到窗口嵌入中. 综上所述,CWindow 的 NS 算法中样本词语 u 跟随上下文 $context(word(t))$ 共现后验概率如公式(1). 其中,u 表示样本词语($word(t)$为正样本,其余为负样本). $\boldsymbol{\theta}^u(i)$ 表示从 $\boldsymbol{O}(u)$ 中截取的一段维度为 d 的向量,其随着输入层中词语定向嵌入的位置 i 而变化. $\boldsymbol{\theta}^u(i)$ 与 $\boldsymbol{x}_{word(t)}$ 中对应向量 $\boldsymbol{v}(word(t+i))$ 进行内积相乘. 例如,$\boldsymbol{\theta}^u(-c)=\boldsymbol{O}(u)[1\sim d]$ 表示 $\boldsymbol{O}(u)$)中第 $1\sim d$ 维的向量,$\boldsymbol{\theta}^u(-c)$ 对应是 $word(t-c)$ 的预测向量,因此将 $\boldsymbol{v}(word(t-c))$ 与 $\boldsymbol{\theta}^u(-c)$ 进行内积相乘.

$$index[i]=\begin{cases}c+i+1, i\in[-c,-1]\\ c+i, i\in[1,c]\end{cases}$$

$$u\in\{word(t)\}\cup Neg(word(t))$$

$$\boldsymbol{\theta}^u(i)=\boldsymbol{O}(u)[(index[i]-1)\times d+1\sim index[i]\times d]$$

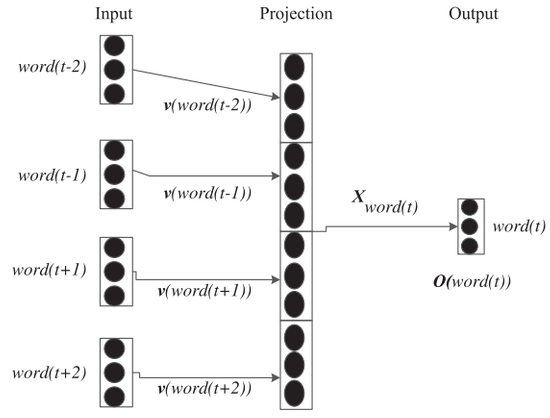

图1　Cwindow结构

$$p(u|context(word(t)))=\sigma(x_{word(t)}\cdot\boldsymbol{O}(u))$$
$$=\sigma\Big(\sum_{-c\leqslant i\leqslant c,i\neq 0}(v(word(t+i)))\cdot\theta^u(i)\Big)\qquad(1)$$

2.2　SSG[17]

SSG 是基于 CSG 改进的模型,其给定中心词的 $word(t)$ 使用单个输出矩阵 $\boldsymbol{O}\in R^{|V|\times d}$ 来预测每个上下文词,结构如图 2. 相比于 CSG,SSG 对词的顺序较为敏感,它利用 $\boldsymbol{O}\in R^{|V|\times d}$ 定义一组 $c\times 2$ 个输出预测矩阵 $\boldsymbol{O}_{-c},\cdots,\boldsymbol{O}_{-1},\boldsymbol{O}_1,\cdots\boldsymbol{O}_{-c}$,每个矩阵专用于预测针对中心字的特定相对位置的输出. 例如预测 $p(word(t)|word(t+2))$ 时,模型以 $\boldsymbol{O}_2$ 为输出矩阵,并以 $word(t+2)$ 为索引得到输出预测向量 $\boldsymbol{O}_2(word(t+2))$.

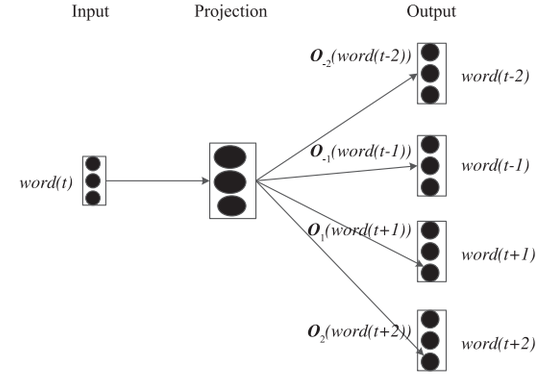

图2　SSG结构

综上所述,SSG 的 NS 算法中 $word(t+i)$ 跟随样本词语 u 共现后验概率如下:

$$u\in\{word(t)\}\cup Neg^{word(t+i)}(word(t))\qquad(2)$$
$$p(word(t+i)|u)=\sigma(v(word(t+i))\cdot\boldsymbol{O}_i(u))$$

2.3　PWE[18]

PWE 模型基于 CBOW 模型进行改进,通过词性相关性加权矩阵来为 $Context(word(t))$ 进行建模. 对于 $S=\{word(1),\cdots,word(N)\}$ 的词序列,每个词语 $word(i)$ 被标记为特定的词性标记 z_i,相应的词 - 词性对被

表示为 $<word(i),z_i>$. 预测中心词 $word(t)$ 的上下文词向量被计算为加权求和,而不是 CBOW 模型中简单的平均/相加求和的运算:

$$\boldsymbol{x}_{word(t)}=\sum_{-c\leqslant i\leqslant c,i\neq 0}(\boldsymbol{\Phi}_i(z_{t+i},z_t)\boldsymbol{v}(word(t+i))) \quad (3)$$

其中 $\boldsymbol{\Phi}_i(z_{t+i},z_i)$ 是表示从词标签 z_{t+i} 到 z_i 的相关权重的核心加权因子,下标 i 指示特定训练上下文窗口内的词 $word(t+i)$ 和 $word(t)$ 之间的位置距离. 因此,基于词性的加权因子是位置相关的. Liu 等人[18]提出的模型主要框架如图 3 所示. 在这个框架中,基于相应的两个词性标签之间的相关性权重对 $Context(word(t))$ 进行建模.

例如,如果我们考虑中心词 $word(t)$ 及其上下文词 $word(t-1)$,我们定义一个词性相关性加权矩阵 $\boldsymbol{\Phi}_{-1}$,其中 $\boldsymbol{\Phi}_{-1}(z_{t-1},z_t)$ 被用作基于相应词性标签 z_{t-1} 和 z_t 的词性加权因子. 同时,考虑词 $word(t)$ 和其上下文词 $word(t+2)$,由于它们之间的距离为 2,我们定义另一个词性相关性加权矩阵 $\boldsymbol{\Phi}_2$,其中我们再次可以提取因子 $\boldsymbol{\Phi}_2(z_{t+2},z_t)$.

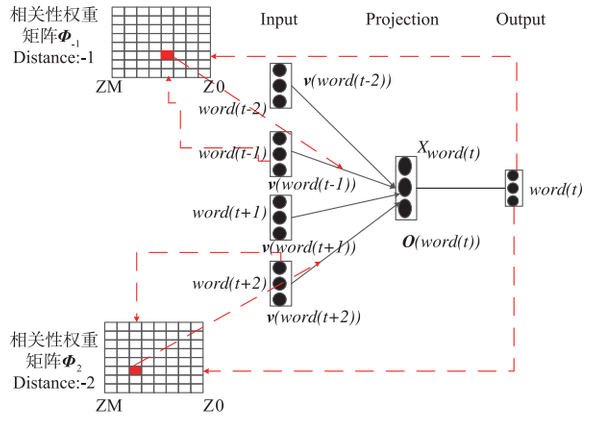

图3 PWE模型结构

本文结合了 Wang[17] 与 Liu[18] 的两种结构优点提出 Structured word2vec on POS,其将词性标注信息与词语顺序作为影响因素联合优化模型. Structured word2vec on POS 在词语定向嵌入结构基础上引入词性相关性权重矩阵作为训练因子. 由于 word2vec 的 NS 算法比多层感知器(Hieraichical Softmax, HS)的计算效率更高[20],因此 Structured word2vec on POS 使用 NS 算法进行训练.

3 改进模型

Structured word2vec on POS 有两个结构:CWindow-POS 与 Structured Skipgram-POS,其原理介绍如下.

3.1 CWindow-POS(CWP)

CWP 是基于 CWindow 和 PWE 的改进模型,定义输出预测矩阵 $\boldsymbol{O}\in R^{|V|\times 2cd}$,并引入 PWE 的词性相关性加权矩阵,结构如图 4. 针对语料 Corp,模型目标函数是一个最大化每个样本标记词的对数似然函数,采用 CBOW 与 CWindow 模型 NS 算法的训练函数如公式(4). 其中 $p(u\mid context(word(t)))$ 的计算过程不同于 CWindow 与 CBOW. CWP 首先将输入层的词向量分别进行词性加权计算;然后结合 CWindow 模型的方法,CWP 模型将词性加权计算后的向量按上下文词语出现顺序而定向嵌入到投影层中,该串联性向量形式如公式(5). 将公式(5)代入公式(1),得出公式(6). Q_{CBOW} 的花括号下式子记为 L 作为 CWP 的目标函数进行梯度推导,将公式(6)代入 L 得出公式(7).

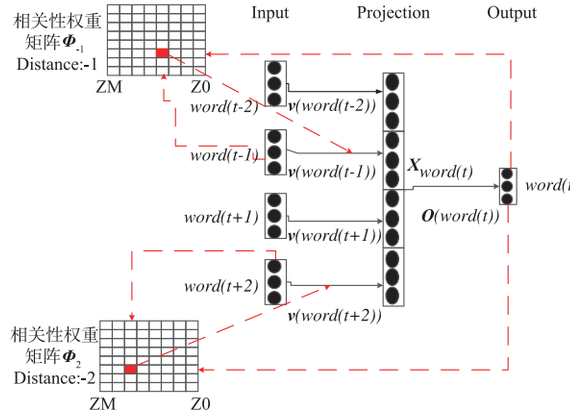

图4 CWindows-POS结构图

针对目标函数中的变量 $\boldsymbol{\Phi}_i(z_{t+i},z_t)$、$\boldsymbol{\theta}^u(i)$ 和 $\boldsymbol{v}(word(t+i))$,在梯度算法中,求解的关键是目标函数对应的三个变量的梯度. 本文用随机梯度上升法对 L 的三个变量进行梯度求解,然后不断地优化更新.

(1) $\boldsymbol{\theta}^u(i)$ 的梯度更新:$\boldsymbol{\theta}^u(i)$ 的梯度如公式(8),$\boldsymbol{\theta}^u(i)$ 的更新公式为(9).

(2) $\boldsymbol{\Phi}_i(z_{t+i},z_t)$ 的梯度更新:$\boldsymbol{\Phi}_i(z_{t+i},z_t)$ 的梯度如公式(10),$\boldsymbol{\Phi}_i(z_{t+i},z_t)$ 更新公式为(11).

(3) $\boldsymbol{v}(word(t+i))$ 的梯度更新:$\boldsymbol{v}(word(t+i))$ 的梯度如如公式(12),$\boldsymbol{v}(word(t+i))$ 更新公式为(13).

$$Q_{\text{CBOW}}=\sum_{word(t)\in Corpu}\sum_{u\in\{word(t)\}\cup Neg(word(t))}\{(L^{word(t)}(u)\times\log[p(u\mid context(word(t)))]+(1-L^{word(t)}(u))\times\log[1-p(u\mid context(word(t)))]\} \quad (4)$$

$$\begin{aligned}\boldsymbol{x}_{word(t)}=[&\boldsymbol{\Phi}_{-c}(z_{t-c},z_t)\boldsymbol{v}(word(t-c)),\cdots\cdots\\&\boldsymbol{\Phi}_{-1}(z_{t-1},z_t)\boldsymbol{v}(word(t-1)),\\&\boldsymbol{\Phi}_1(z_{t+1},z_t)\boldsymbol{v}(word(t+1)),\cdots\cdots\\&\boldsymbol{\Phi}_i(z_{t+c},z_t)\boldsymbol{v}(word(t+c))]\end{aligned} \quad (5)$$

$$index[i]=\begin{cases}c+i+1,i\in[-c,-1]\\c+i,i\in[1,c]\end{cases}$$

$$u\in\{word(t)\}\cup Neg(word(t));$$

$$\boldsymbol{\theta}^u(i)=\boldsymbol{O}(u)[(index[i]-1)\times d+1\sim index[i]\times d]$$

$$p(u\mid context(word(t)))=\sigma(x_{word(t)}\cdot\boldsymbol{O}(u))$$
$$=\sigma\Big(\sum_{-c\leqslant i\leqslant c,i\neq 0}(\boldsymbol{\Phi}_i(z_{t+i},z_t)\boldsymbol{v}(word(t+i))\cdot\theta^u(i))\Big) \tag{6}$$

$$L=\{(L^{word(t)}(u)\times\log[\sigma(\sum_{-c\leqslant i\leqslant c,i\neq 0}(\boldsymbol{\Phi}_i(z_{t+i},z_t)\boldsymbol{v}(word(t+i))\cdot\boldsymbol{\theta}^u(i))]+(1-L^{word(t)}(u))\times\log[1-\sigma(\sum_{-c\leqslant i\leqslant c,i\neq 0}(\boldsymbol{\Phi}_i(z_{t+i},z_t)\boldsymbol{v}(word(t+i))\cdot\theta^u(i))]\}dot\_product\_weight$$
$$=\sum_{-c\leqslant i\leqslant c,i\neq 0}(\boldsymbol{\Phi}_i(z_{t+i},z_t)\boldsymbol{v}(word(t+i))\cdot\boldsymbol{\theta}^u(i)) \tag{7}$$

$$\frac{\partial(L)}{\partial\theta^u(i)}=[L^{word(t)}(u)-\sigma(dot\_product\_weight)]\frac{\partial(dot\_product\_weight)}{\partial\theta^u(i)}$$
$$=[L^{word(t)}(u)-\sigma(dot\_product\_weight)]\sum_{-c\leqslant i\leqslant c,i\neq 0}(\boldsymbol{\Phi}_i(z_{t+i},z_t)\boldsymbol{v}(word(t+i))) \tag{8}$$

$$\boldsymbol{\theta}^u(i)=\boldsymbol{\theta}^u(i)+\eta\frac{\partial(L)}{\partial\boldsymbol{\theta}^u(i)},\ -c\leqslant i\leqslant c,i\neq 0 \tag{9}$$

$$\frac{\partial(L)}{\partial\boldsymbol{\Phi}_i(z_{t+i},z_t)}$$
$$=[L^{word(t)}(u)-\sigma(dot\_product\_weight)]\frac{\partial(dot\_product\_weight)}{\partial\boldsymbol{\Phi}_i(z_{t+i},z_t)}$$
$$=[L^{word(t)}(u)-\sigma(\sum_{-c\leqslant i\leqslant c,i\neq 0}(\boldsymbol{\Phi}_i(z_{t+i},z_t)\boldsymbol{v}(word(t+i))\cdot\boldsymbol{\theta}^u(i)))]\boldsymbol{\theta}^u(i)\cdot\boldsymbol{v}(word(t+i)) \tag{10}$$

$$\boldsymbol{\Phi}_i(z_{t+i},z_t)=\boldsymbol{\Phi}_i(z_{t+i},z_t)+\eta\sum_{u\in\{word(t)\}\cup Neg(word(t))}\frac{\partial(L)}{\partial\boldsymbol{\Phi}_i(z_{t+i},z_t)},\ -c\leqslant i\leqslant c,i\neq 0 \tag{11}$$

$$\frac{\partial(L)}{\partial v(word(t+i))}$$
$$=[L^{word(t)}(u)-\sigma(dot\_product\_weight)]\frac{\partial(dot\_product\_weight)}{\partial v(word(t+i))}$$
$$=[L^{word(t)}(u)-\sigma(\sum_{-c\leqslant i\leqslant c,i\neq 0}(\boldsymbol{\Phi}_i(z_{t+i},z_t)\boldsymbol{v}(word(t+i))\cdot\boldsymbol{\theta}^u(i)))]\boldsymbol{\Phi}_i(z_{t+i},z_t)\boldsymbol{\theta}^u(i) \tag{12}$$

$$\boldsymbol{v}(word(t+i))=\boldsymbol{v}(word(t+i))+\eta\sum_{u\in\{word(t)\}\cup Neg(word(t))}\frac{\partial(L)}{\partial v(word(t+i))},\ -c\leqslant i\leqslant c,i\neq 0 \tag{13}$$

3.2　Structured Skipgram-POS(SSGP)

SSGP 是基于 SSG 和 PWE 的改进模型,其给定中心词的 $word(t)$ 使用单个输出矩阵 $\boldsymbol{O}\in\boldsymbol{R}^{|V|\times d}$ 来预测每个上下文词,并引入词性相关性加权矩阵进行建模,结构如图 5. 针对语料库 Corp,CSG 与 SSG 模型 NS 算法的训练函数如公式(14).

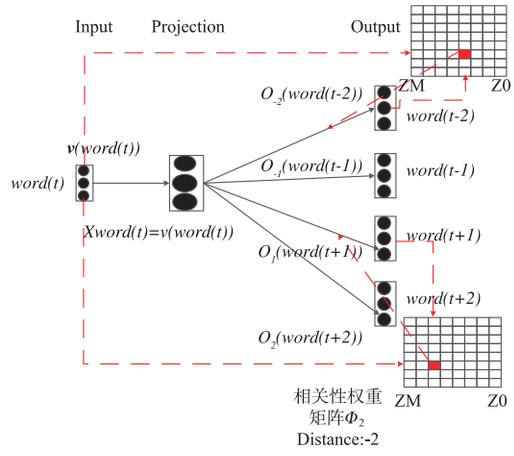

图5　SSGP结构

$p(word(t+i)\mid u)$不同于 CSG 与 SSG. SSGP 将 PWE 的词性相关度加权矩阵 $\boldsymbol{\Phi}_i$ 加入输出层,矩阵中基于词性的加权因子与词语定向嵌入的位置相关. 因此在公式(2)中加入 $\boldsymbol{\Phi}_i$ 后将计算式如公式(15). Q_{CSG}的花括号下式子记为 L 作为 Structured Skipgram-POS 的目标函数进行梯度推导,将公式(15)代入公式(16). 针对目标函数中的三个变量参数:

$\boldsymbol{\Phi}_i(z_{t+i},z_t)$、$\boldsymbol{O}_i(u)$和 $\boldsymbol{v}(word(t))$,在梯度算法中,求解的关键是目标函数对应的三个参数的梯度. 本文用随机梯度上升法对 L 的三个变量进行梯度求解,然后不断地优化更新. 由 3.1 推论得出:$\boldsymbol{O}_i(u)$的更新公式为(17),$\boldsymbol{\Phi}_i(z_{t+i},z_t)$更新公式为(18),$\boldsymbol{v}(word(t+i))$更新公式为(19).

$$Q_{CSG}=\sum_{word(t)\in Corp}\sum_{-c\leqslant i\leqslant c,i\neq 0}\sum_{u\in\{word(t)\}\cup Neg^{word(t+i)}(word(t))}\{(L^{word(t)}(u)\times\log[p(word(t+i)\mid u)]+(1-L^{word(t)}(u))\times\log[1-p(word(t+i)\mid u)]\} \tag{14}$$

$$p(word(t+i)\mid u)=\sigma(\boldsymbol{\Phi}_i(z_{t+i},z_t)\boldsymbol{v}(word(t+i))\cdot O_i(u)),\begin{cases}-c\leqslant i\leqslant c,i\neq 0;\\ u\in\{word(t)\}\cup Neg^{word(t+i)}(word(t))\end{cases} \tag{15}$$

$$L=\{(L^{word(t)}(u)\times\log[\sigma(\boldsymbol{\Phi}_i(z_{t+i},z_t)\boldsymbol{v}(word(t+i))\cdot\boldsymbol{O}_i(u))]+(1-L^{word(t)}(u))\times\log[1-\sigma(\boldsymbol{\Phi}_i(z_{t+i},z_t)\boldsymbol{v}(word(t+i))\cdot\boldsymbol{O}_i(u))]\} \tag{16}$$

$$\boldsymbol{O}_i(u)=\boldsymbol{O}_i(u)+\eta\frac{\partial(L)}{\partial\boldsymbol{O}_i(u)},\ -c\leqslant i\leqslant c,i\neq 0 \tag{17}$$

$$\boldsymbol{\Phi}_i(z_{t+i},z_t)=\boldsymbol{\Phi}_i(z_{t+i},z_t)+\eta\sum_{u\in\{word(t)\}\cup Neg^{word(t+i)}(word(t))}\frac{\partial(L)}{\partial\boldsymbol{\Phi}_i(z_{t+i},z_t)},\ -c\leqslant i\leqslant c,i\neq 0 \tag{18}$$

$$\boldsymbol{v}(word(t+i))=\boldsymbol{v}(word(t+i))$$

$$+\eta \sum_{u\in\{word(t)\}\cup Neg^{word(t+i)}(word(t))} \frac{\partial(L)}{\partial \boldsymbol{v}(word(t+i))}, \quad -c\leqslant i\leqslant c, i\neq 0 \tag{19}$$

4 实验

4.1 实验设置

在本节中,我们介绍实验设置,其中包括所有实验的训练语料,评测任务,词性信息标记和参数设置.

(1)训练语料 本文使用 2016 年 4 月 ~6 月的英文维基百科语料库,共有约 600 万条文章和 30 亿个 tokens.

(2)语料库的词性标注 实验使用 OpenNLP toolkit3 对训练语料进行词性标记. 标签集是 Penn Treebank 词性标签集[19],其由 36 个常见的词性标签和 6 个符号标签组成.

(3)词语评测任务 实验中的词语评测任务有两个:(*a*)词语类比任务. 类比任务是我们的主要焦点,因为它对向量空间子结构进行了语义和语法上的测试,测评数据集为 MSR[20]、SYN 与 SEM[5]. (*b*)除了类比任务之外,实验还引入了词语相似性任务来评估我们的模型. 测评数据集为 WordSim-353 (WS353), SCWS 和 RW.

(4)实验超参数设计 在实验中发现,只有向量维度与迭代次数对实验结果有很大影响. 其中当向量维度 100 ~300 以及迭代次数 1 ~5 时,词语类比结果变化较为明显,因此实验记录了在不同向量维度(表示为 Dim)与迭代次数(表示为 Epoch)之下的结果. 对于 word embedding 模型训练的其他超参数,我们设置如下:负样本数为 5;上下文窗口大小设置为 5;学习率的初始值设为 0.025,并随着训练过程线性下降[13];词性相关性加权矩阵中的所有的权重初始值设为相同值[17]. 最终记录的实验结果来源于多次运行结果的平均值.

4.2 实验结果分析

我们将词语类比结果的数据绘制成图 6,其中纵坐标为模型在三个数据集上的测试准确率的均值(*avg*). 从柱状图分析得出:

(1)CBOW 与 CSG 作为 word2vec 的最初始模型,效果明显低于其他的模型. 这印证了相关工作中所介绍的 PWE 、CWindow、SSG 等改进模型所带来的优势.

(2)显然,在实验中词语类比准确率与向量维度、迭代次数呈正相关. 当向量维度为 300,迭代次数为 5 的时候(即词语类比准确率最优的参数设置),本文提出的 CWP 与 SSGP 模型在词语类比任务的总体效果优于其他 word embedding 模型.

(3)当向量维度相同时,PWE、CWP 与 SSGP 的效果会随着迭代次数增加有着更明显的优势(比较(1)与(2)、(3)与(4)可知). 其原因在于:PWE、CWP 与 SSGP 模型中均加入了词加性相关权重,因此 PWE、CWP 与 SSGP 需要更多的运算量才能更好地将三个变量参数拟合,从而更好地细化词性相关权重.

(4)CWP 与 SSGP 的效果一直高于 PWE. 其原因在于:CWP 与 SSGP 通过对上下文的词向量依次进行首尾拼接,充分利用了词序信息进行训练.

我们将词语相似性结果的数据绘制成图 7,其中纵坐标为模型在三个数据集上的测试准确率的均值(avg). 从柱状图分析得出:

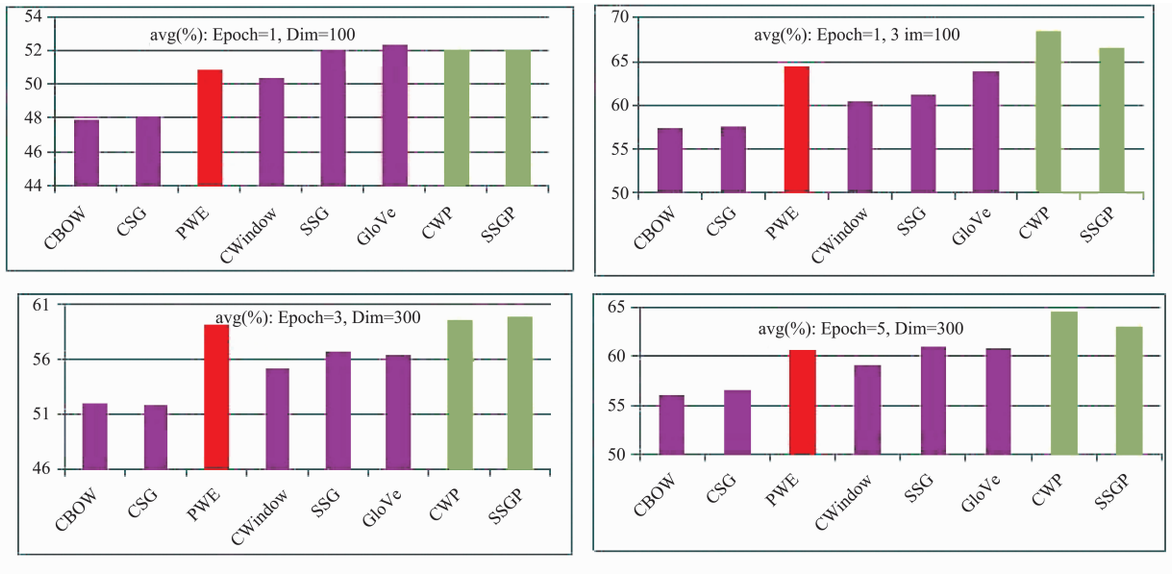

图6 在不同向量维度与迭代次数之下各模型词语类值

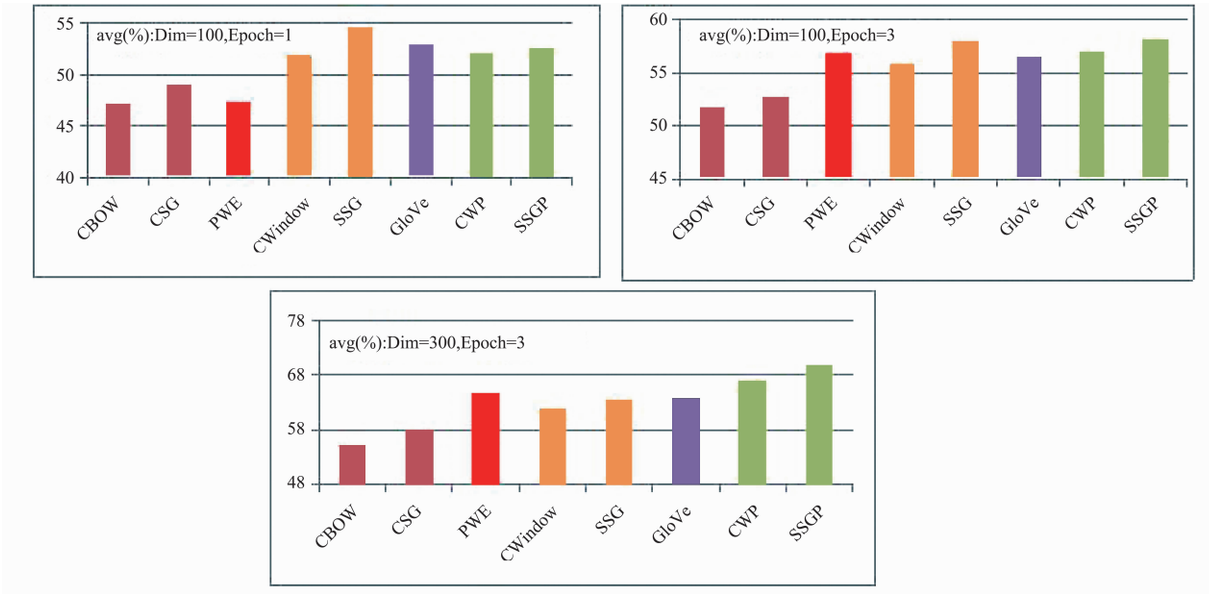

图7　在不同向量维度与迭代次数之下各模型词语类值

(1)与上述的词语类比任务的分析结果基本一样.

(2)在同一作者提出的方法中(即柱状颜色相同)进行横向对比,我们发现 CSG 及其改进模型(SSG,SSGP)比起 CBOW 及其改进模型(CWindow,CWP)的效果更好. 原因在于 CSG 及其改进模型可以更好地获取中心词与上下文每个词之间的关系信息,而相比之下 CBOW 及其改进模型只是简单地将上下文所有词向量合并作为输入,使得每个词对(中心词-上下文词)之间的关系弱化了.

5　结论

本文在 Structured word2Vec 与 PWE 两个方法的基础上进行改进,将 PWE 的词性相关权重与 Structured word2Vec 的基于词序的定向嵌入结构进行合并,提出了 CWP 与 SSGP 两个模型. CWP 与 SSGP 既能感知网络中上下文单词的相对位置信息,又能利用词性相关性权重对上下文序列进行加权运算.

CWP 与 SSGP 在训练过程如下:在词定向嵌入结构中,使用随机梯度下降算法联合学习 word embedding 和词性相关性加权矩阵. 实验通过词语类比任务,词语相似性任务与定性分析都证明了所提出的模型的有效性.

参考文献

[1] Johnson R,Zhang T. Effective use of word order for text categorization with convolutional neural networks[J]. arXiv Preprint arXiv:1412. 1058,2014.

[2] ZHAN C D,LING Z H,DAI L R. Learning word embeddings for paraphrase scoring in knowledge base based question answering[J]. Pattern Recognition and Artificial Intelligence,2016,29(9):825 – 831.

[3] 尹存燕,黄书剑,戴新宇,等. 中英命名实体识别及对齐中的中文分词优化[J]. 电子学报,2015,43(8):1481 – 1487.
YIN C Y,et al. Optimization of Chinese word segmentation in named entity recognition and word alignment[J]. Acta Electronica Sinica,2015,43(8):1481 – 1487. (in Chinese)

[4] 杨思春,戴新宇,陈家骏. 面向开放域问答的问题分类技术研究进展[J]. 电子学报,2015,43(8):1627 – 1636.
YANG S C,et al. Advances in question ciassification for open-domain question answering[J]. Acta Electronica Sinica 2015,43(8):1481 – 1487. (in Chinese)

[5] Mikolov T,Yih S W,Zweig G. Linguistic regularities in continuous space word representations[A]. Conference of the North American Chapter of the Association of Computational linguistics[C]. Atlanta,Georgia,USA:Association of Computational Linguistics,2013. 746 – 751.

[6] Bengio Y. Learning deep architectures for AI[J]. Foundations and trends ® in Machine Learning,2009,2(1):1 – 127.

[7] Chang K W,Yih W,Meek C. Multi-relational latent semantic analysis[A]. EMNLP[C]. Seattle,Washington,USA:Association for Computational Linguistics,2013. 1602 – 1612.

[8] Lund K,Burgess C. Hyperspace analogue to language (HAL):A general model semantic representation[J]. Brain and Cognition,1996,30(3):5 – 5.

[9] Lebret R, Collobert R. Word Emdeddings through Hellinger PCA[A]. Conference of the European Chapter of the Association for Computational Linguistics (EACL)[C]. Gothenburg, Sweden: Association for Computational Linguistics, 2014. 482 – 490.
[10] Bengio Y, Schwenk H, Senécal J S, et al. Neural Probabilistic Language Models[M]. Berlin Heidelberg: Springer, 2006. 137 – 186.
[11] Zhang X, Gu N, Ye H. Multi-GPU based recurrent neural network language model training[A]. International Conference of Young Computer Scientists, Engineers and Educators[C]. Singapore: Springer, 2016. 484 – 493.
[12] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model [A]. INTERSPEECH 2010, Conference of the International Speech Communication Association[C]. Makuhari, Chiba, Japan, 2010. 1045 – 1048.
[13] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. Computer Science, 2013, 5(4): 243 – 254.
[14] Mnih A, Kavukcuoglu K. Learning word embeddings efficiently with noise-contrastive estimation[A]. Advances in Neural Information Processing Systems[C]. Lake Tahoe, Nevada, United States, 2013. 2265 – 2273.
[15] Levy O, Goldberg Y, Ramat-Gan I. Linguistic regularities in sparse and explicit word representations[A]. CoNLL 2014. Association for Computational Linguistics[C]. Baltimore, Maryland, USA, 2014. 171 – 180.
[16] Pennington J, Socher R, Manning C D. GloVe: global vectors for word representation[A]. Association for Computational Linguistics[C]. Doha, Qatar: Association for Computational Linguistics, 2014, 14: 1532 – 43.
[17] Ling W, Dyer C, Black A, et al. Two/too simple adaptations of word2vec for syntax problems[A]. The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Denver[C]. Colorado, USA: Association for Computational Linguistics, 2015. 1299 – 1304.
[18] Liu Q, Ling Z H, Jiang H, et al. Part-of-speech relevance weights for learning word embeddings[J]. Arxiv Preprint Arxiv, 2016, 9(7): 134 – 139.
[19] Marcus M P, Marcinkiewicz M A, Santorini B. Building a large annotated corpus of English: the penn treebank[J]. Computational Linguistics, 1993, 19(2): 313 – 330.
[20] Finkelstein L, Gabrilovich E, Matias Y, et al. Placing search in context: The concept revisited[A]. Proceedings of the 10th International Conference on World Wide Web[C]. New York: Association for Computational Linguistics, 2001. 406 – 414.

作者简介

潘 博 男, 1992 年生于广西右江. 现为北京工商大学计算机与信息工程学院研究生. 主要研究方向为智能控制与模式识别.
E-mail: 2390209273@qq.com

于重重(通信作者) 女, 1971 年生于辽宁丹东. 2013 年与北京科技大学获得博士学位. 现为北京工商大学计算机与信息工程学院研究生导师, 教授. 主要研究方向为机器学习与数据挖掘.
E-mail: chongzhy@vip.sina.com