华南理工大学

South China University of Technology

人工智能课程论文翻译

学院: 软件学院

专业: 软件工程

导师: 宋恒杰

姓名: 金成能

学号: 201530611838

日期: 2018年1月15日

怎样生成一个优质的词向量

I. 摘要

在训练词向量的过程中有三个关键的组件，分别是模型、语料和训练参数。我们对已经存在的基于神经网络的词向量生成算法进行系统化整理，并用相同的语料对这些算法进行比较。我们用三种方式评估每个词向量：分析其语义属性、将其用作有监督任务的特征、使用它来初始化神经网络。我们还提供了几个简单的训练词向量的指导意见。首先，我们发现语料库领域比语料库规模更重要。我们建议在合适的领域选择一个语料库来完成所需的任务，然后使用更大的语料库可以得到更好的结果。其次，我们发现更快的模型在大多数情况下提供了足够的性能，如果训练语料库足够大，可以使用更复杂的模型。第三，迭代的早停止参数应该依赖于预期任务的开发设置而不是训练词向量在验证集上的损失。

II. 介绍

词向量也被称为分布式词表示，可以从大型未标记的语料库中捕获词的语义和句法信息，并且已经引起了许多研究人员的强烈关注。近年来，已经提出了几种模型，其中许多模型已经在各种自然语言处理（NLP）任务中取得了最新的成果。虽然这些研究总是声称他们的模型比以前的模型在各种评估标准上都要好，但是很少有研究比较现有的词向量方法。本文重点研究了这个问题，并通过一个实验性的综述，包括模型构建，训练语料和参数设计，对训练词向量的几个重要特征进行了详细的分析。据我们所知，以前没有进行这样的研究。

要设计一个有效的词向量算法，首先要弄清楚模型的构建。我们观察到，几乎所有训练词向量的方法都基于相同的分布假设：在类似的语境中出现的词往往具有相似的含义。基于这个假设，不同的方法以不同的方式建立了一个词w（目标词）和其在语料中的上下文c的关系，其中w和c以词向量的形式展现。一般而言，现有方法在模型构建的两个主要方面有所不同：（i）目标词与其上下文之间的关系;（ii）上下文的表示。图1显示了常用模型的简要比较。

Model	Relation of w, c	Representation of c
Skip-gram [18]	c predicts w	one of c
CBOW [18]	c predicts w	average
Order	c predicts w	concatenation
LBL [22]	c predicts w	compositionality
NNLM [2]	c predicts w	compositionality
C&W [3]	scores w, c	compositionality

Fig. 1. 各个模型表现目标词与其上下文关系的方法和其表现上下文的方法

就目标词与其上下文之间的关系而言，前五种模型是相同的。他们使用类似于条件概率$p(w|c)$的对象，它根据上下文c来预测目标词w。C&W使用类似于联合概率的对象，训练语料库中的(w, c)对以获得更高的分数。就上下文的表示而言，模型使用四种不同类型的方法。在表1中，它们按从上到下的复杂度从小到大的顺序排列。Skipgram使用最简单的策略，即从目标单词的窗口中选择一个单词，并利用其词向量作为上下文的表示。CBOW使用上下文单词的平均词向量作为上下文表示。这两种方法都忽略了词序来加速训练过程。但是，托马斯·兰德尔（Thomas Landauer）估计，文本中20%的含义来自词序，其余来自词的选择。因此，这两种模式可能会丢失一些关键信息。相反，Order模型使用上下文词向量的连接，其维护了词序信息。此外，对数双线性语言（LBL）模型，神经网络语言模型（NNLM）和C&W模型在Order模型中增加了一个隐含层。因此，这些模型使用上下文单词的语义组成作为上下文表示。基于上述分析，我们希望知道两件事情：哪一种模型表现最好？我们应该在根据目标词和上下文之间的关系以及不同类型的上下文表示中选择哪一种?

另外，训练精确的词向量的能力与训练语料密切相关。不同大小，不同领域的不同训练语料库可以对最终结果产生相当大的影响。因此，我们还想知道语料库的大小和领域如何影响词向量的表现。

最后，训练精确的词向量强烈依赖于某些参数，如迭代次数和词向量的维数。在这方面，我们希望知道另外两件事情：应该应用多少次迭代来获得足够优质的字向量，同时避免过拟合？我

们应该选择什么维度来获得足够优质的词向量?

为了客观地回答这些问题，我们评估了各种词向量在三种主要类型的八个任务中的效果：计算词向量的语义属性，将词向量用作现有NLP任务的特征，并且利用词向量来初始化其他神经网络模型。我们认为目前所有的词向量的应用都被这里提到的类型所覆盖。通过不同类型的词向量的不同任务，我们试图确定哪种类型的模型设计最适合于每个特定的任务（第一个问题）。此外，我们改变语料库的大小和领域，试图回答第二个问题。

III. 模型

在通过实验比较词向量模型之前，我们先来描述和分析它们。我们用$e(w)$表示词w的词向量。

A. 模型概述

我们描述了六个代表性模型，这些模型广泛用于训练词向量和NLP任务，如语言模型。

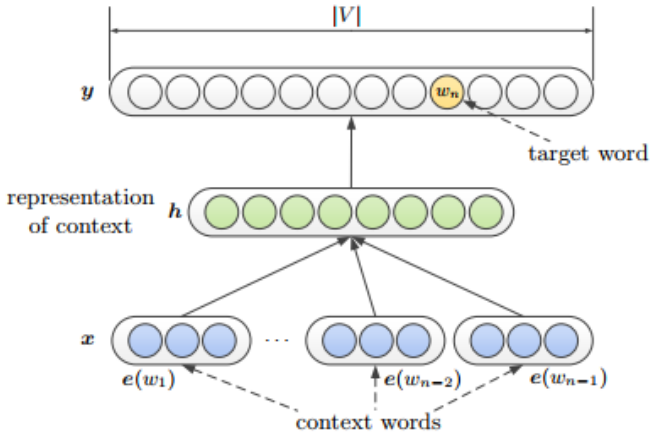

Fig. 2. NNLM的模型结构

1) 神经网络语言模型（*NNLM*）：Yoshua Bengio及其同事首先提出了一种同时学习词向量和语言模型的神经网络语言模型（NNLM）。对于语料库中的每个样本，给出前面的词，我们使最后一个词的概率的对数似然最大化。例如，对于语料库中的序列$w_1, w_2, ..., w_n$，我们需要最大化$P(w_n|w_1, w_2, ..., w_{n-1})$的对数似然性，其中我们把要预测的单词$\Phi w_n \Psi$作为目标词。这个模型使用前面的词向量的串联作为输入：

$$x = [e(w_1), ..., e(w_{n-2}), e(w_{n-1})] \tag{1}$$

模型结构是一个带有一个隐藏层的前馈神经网络：

$$\begin{aligned} h &= tanh(d + Hx) \\ y &= b + Uh \end{aligned} \tag{2}$$

其中U是变换矩阵，b和d是偏差向量。最后一步是应用softmax层来获得目标词的概率。

2) 对数双线性语言模型*(LBLM):* Andriy Mnih和Hinton提出的对数双线性语言模型(LBLM)与NNLM类似.LBLM使用与NNLM几乎相同的对数双线性能量函数，并取消了非线性激活函数$tanh$。

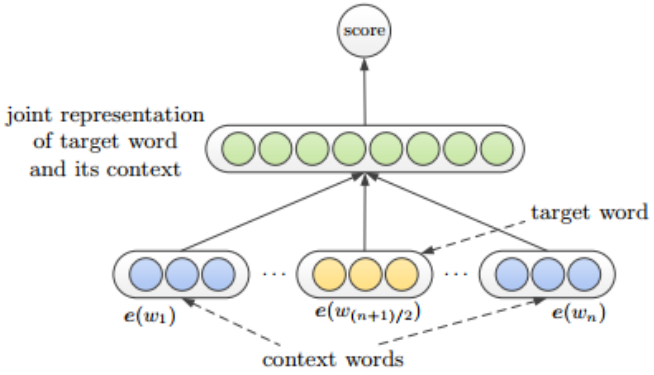

Fig. 3. C&W的模型结构

3) Collobert and Weston (C&W): C&W模型仅训练词向量，不预测目标单词。与其他模型不同的是，它将目标词与其上下文相结合，然后对它们进行评分。评分函数是一个隐层神经网络。输入是目标单词和上下文词向量的连接。训练目标是最大化语料库序列的得分，同时最小化噪声序列的得分，或者在形式上最小化

$$max(0, 1 - s(w, c) + s(w', c)) \tag{3}$$

在噪声序列中，目标词w由词汇表中的随机词w'代替。

*4) CBOW*和*skip-gram:* CBOW和skip-gram模型试图最小化计算复杂性。CBOW使用上下文的平均词向量作为上下文表示，而skip-gram使用上下文单词中的某个词向量作为上下文的表示。两种模型都忽略了词序信息，只用逻辑回归来预测目标词。

5) Order: 为了分析词序信息的使用，我们引入了一个名为"order"的虚拟模型，其复杂程度在CBOW和LBL模型之间。这个模型保持像LBL这样的词顺序，同时去除像CBOW这样的隐藏层。

6) GloVe: 除了神经网络方法之外，另一种方法基于单词上下文矩阵来训练词向量，其中每一行对应于一个单词并且每一列对应于一个上下文。矩阵中的元素与相应的单词和上下文的实例的同现次数有关。这些模型被称为基于计数的模型。这种矩阵方法的最新研究是全局矢量(GloVe)模型。

B. 模型分析

上一节中的模型在两个主要方面有所不同：目标词与其上下文之间的关系以及上下文的表示。

1) 目标词与上下文的关系: 现有的基于神经网络的词向量模型以两种不同的方式设计目标词与上下文之间的关系。大多数模型使用上下文来预测目标词。在NNLM中，隐藏层h是上下文c的表示。变换矩阵U的维数为$|V||h|$，其中$|V|$是词汇的大小，而$|h|$是隐藏层的维度。矩阵U中的每一行都可以看作是相应单词的补充嵌入。我们将w的补充嵌入表示为$e'(w)$。因此，在这些模型中，词汇表中的每个单词都有两个嵌入，当w是上下文时，其词向量为$e(w)$，当w是目标单词时，其词向量为$e'(w)$。单词w的能量函数是$e'(w)Tc$。与基于语言模型的方法相比，C&W模型将目标词放在输入层中，并且只为每个词产生一个词向量。单词w的能量函数为$Ae(w) + Bc$，其中A和B是变换矩阵，c是上下文的表示。所以这两种模式有很大的不同。

2) 上下文的表示: 预测目标词的模型使用不同的策略来表示语境。图4将各模型的上下文表示进行形式化。

Model	representation of the context
Skip-gram	$\boldsymbol{e}(w_i), 1 \le i \le n-1$
CBOW	$\frac{1}{n-1}(\boldsymbol{e}(w_1) + \cdots + \boldsymbol{e}(w_{n-2}) + \boldsymbol{e}(w_{n-1}))$
Order	$[\boldsymbol{e}(w_1), \ldots, \boldsymbol{e}(w_{n-2}), \boldsymbol{e}(w_{n-1})]$
LBL	$H[\boldsymbol{e}(w_1), \ldots, \boldsymbol{e}(w_{n-2}), \boldsymbol{e}(w_{n-1})]$
NNLM	$\tanh(\boldsymbol{d} + H[\boldsymbol{e}(w_1), \ldots, \boldsymbol{e}(w_{n-2}), \boldsymbol{e}(w_{n-1})])$

Fig. 4. 预测目标词模型的格式化上下文表示方法

IV. 任务

我们评估了八个任务中的各种词向量模型，这八个任务分为三种主要类型，如下所示。

A. 语义属性

词嵌入模型的设计是基于分布假设设计(具有相似含义的词倾向于具有相似的词嵌入)。我们就几个经典的工作进行评估:

1) ws: WordSim353 set 包含353个字对。它是通过要求人类主体在数字尺度上对两个词之间的语义相似性或相关性程度进行评估而构建的。词向量的效果是通过比较两个词向量的余弦距离的Pearson相关性和参与者给出的平均分数来衡量的。

2) tfl: TPEFL set 包含80个选择性同义词问题，每个问题有4个候选项。我们根据余弦距离从候选词中选择问题单词的最近邻，并使用准确性来衡量表现。

3) sem和syn: 类比任务有大约9,000个语义和10,500个句法类比问题。问题类似于“男人相对于女人与国王相对于皇后”或“predict相对于predicting与dance相对于dancing”。继之前的工作之后，我们用(女王- 王+男)最近的邻接词作为答案。准确度有助于衡量词向量效果。

B. 将词向量作为特征

词向量模型从未标记的语料库中捕获有用的信息。许多现有的工作直接使用了词向量作为特征来提高某些任务的性能。

1) avg: avg任务使用词向量的加权平均值作为文本的表示，然后应用逻辑回归来执行文本分类。每个单词的权重是其出现频率。我们使用的是IMDB数据集。

2) ner: ner任务使用词嵌入作为近期命名实体识别(NER)系统的附加功能。用CoNLL03共享任务数据集的测试集上的F1分数来评估性能。

C. 用词向量初始化神经网络

Dumitru Erhan及其同事已经证明，更好的初始化可以使神经网络模型收敛到更好的局部最优解。在最近的NLP任务的神经网络方法中，词向量被用来初始化第一层。我们选择了两部分最先进的研究成果:

1) 句子情感分类: 我们使用卷积神经网络(CNN)在斯坦福情绪树库数据集上进行句子情感分类。我们重复我们的实验五次，并展示这些实验的平均准确性。

2) 词性标注: 我们使用Ronan Collobert及其同事提出的神经网络对华尔街日报数据进行词性标注(POS)。

V. 实验与结果

图5显示了我们的详细实验设置。

Type	Setting
Model	GloVe, Skip-gram, CBOW, Order, LBL, NNLM, C&W
Corpus	Wiki: 100M, 1.6B; NYT: 100M, 1.2B; W&N: 10M, 100M, 1B, 2.8B; IMDB: 13M;
Para.	dimensionality: 10, 20, 50, 100, 200 fixed window size: 5

Fig. 5. 构建词向量的训练集

Model	syn	sem	ws	tfl	avg	ner	cnn	pos
Random	0.00	0.00	0.00	25.00	64.38	84.39	36.60	95.41
GloVe	40.00	27.92	56.47	**77.50**	74.51	88.19	43.29	96.42
Skip-gram	51.78	**44.80**	**63.89**	76.25	**74.94**	**88.90**	43.84	96.57
CBOW	**55.83**	44.43	62.21	**77.50**	74.68	88.47	43.75	96.63
Order	55.57	36.38	62.44	**77.50**	**74.93**	88.41	**44.77**	**96.76**
LBL	45.74	29.12	57.86	75.00	74.32	88.69	43.98	**96.77**
NNLM	41.41	23.51	59.25	71.25	73.70	88.36	44.40	96.73
C&W	3.13	2.20	46.17	47.50	73.26	88.15	41.86	96.66

Fig. 6. W&N语料库上每个模型训练的50维嵌入的最佳结果

A. 性能增益比

为了比较八个任务的模型，我们需要一个统一的评估指标，因为每个任务的现有评估指标在平均值和方差上有所不同(见图6)。

为了解决这个问题，我们提出一个新的指标，即性能增益比(PGR)，作为原始评估指标的标准化。词向量a相对于词向量b的PGR被定义为

$$PGR(a,b) = \frac{p_a - prand}{p_b - p_{rand}} X100\% \qquad (4)$$

其中p_x是针对给定任务的词向量x的性能，prand是通过随机向量实现的性能。这里，随机词向量是与词向量x相同维度的词向量，其中每个维度是均匀分布的随机变量，范围从-1到1。词向量b被选择为对于给定设置的最佳词向量;因此，我们简单地称这个度量为词向量的PGR。有了这个定义，PGR是一个百分比。如果$PGR = 100\%$，那么对于相同的设置，该词向量在所有词向量中取得最好的结果。如果$PGR = 0$，则嵌入可能不包含比随机词向量更多的有用信息。如果$PGR < 0$，那么词向量对任务是不利的。

B. 模型比较

为了比较不同的模型，我们使用相同的实现。我们的GloVe实现基于GloVe工具包(http://nlp.stanford.edu/projects/glove)，CBOW和skipgram实现基于word2vec工具包(https://code。google.com/p/word2vec)。其他模型通过修改word2vec中的CBOW实现来实现。对于所有模型，我们使用窗口中的中心词作为目标词。对于基于神经网络的模型(除了GloVe之外的所有模型)，我们使用$t = 10^{-4}$的子采样和5个负样本的负采样。

对于每个模型，我们进行迭代，直到模型收敛或在所有的任务上过拟合。表4显示了在W&N语料库(维基百科和纽约时报的文章集合)上训练的每个模型的最佳性能。与随机词向量相比，所有被比较的词向量在任务上表现出更好的性能。

1) 目标词与上下文的关系*:* 为了判断目标词与其上下文之间关系的影响，我们应该比较将目标词与上下文相结合的C&W模型与预测目标词的其他模型进行比较。

与其他语义属性任务模型(syn，sem，ws和tfl)相比，C&W模型表现出较低的性能。特别是在类比任务(syn和sem)中，结果表明C&W模型几乎完全缺乏线性语义减法的特征。

为了直观地理解这两种模型之间的差异，我们给出了图7中所选单词的最近邻居。从这些情况中，我们发现，由CBOW模型训练的“星期一”的最近相似词是一周中的其他日子。相比之下，在C&W模式的训练下，最近的相似词是一天的时间。“最常见”的最近相似词也显示了类似的结果：CBOW模型找到可以替换单词“常见”的单词，而C&W模型将找到与“常见”一起使用的单词。大多数最近的邻居“微红“除了C&W模式中的”pendulous“这个词，它们都可以用”微红“来形容花朵。

Model	Monday	commonly	reddish
CBOW	Thursday	generically	greenish
	Friday	colloquially	reddish-brown
	Wednesday	popularly	yellowish
	Tuesday	variously	purplish
	Saturday	Commonly	brownish
C&W	8:30	often	purplish
	12:50	generally	pendulous
	1PM	previously	brownish
	4:15	have	orange-brown
	mid-afternoon	are	grayish

Fig. 7. 在W&N语料库上使用CBOW和C&W模型进行训练时所选单词的最邻近的邻居

2) 上下文的表示*:* 为了研究上下文表示的影响，我们进一步比较了在不同规模的语料上训练的词

向量。

图8表现了每个模型达到95%PGR的任务数量(为了方便起见，我们说模型在这种情况下是“胜利的”)。在1000万单词短语语料库上训练时”胜利”任务的模型应与在1000万单词短语语料库上训练的最佳词向量进行比较。我们只关注上面列出的五个模型，因为这些模型之间的唯一区别就是它们的上下文的表示。

Model	10M	100M	1B	2.8B
Skip-gram	4+2	4+2	2+2	3+2
CBOW	1+1	3+3	4+1	4+1
Order	0+2	1+2	2+3	3+3
LBL	0+2	0+2	0+2	1+2
NNLM	0+2	0+3	0+3	0+2

Fig. 8. 某个模型在特定语料库上“胜出”的任务.在每个单元格中，a+b表示该模型在前四个任务中“胜出”a个任务，在后四个任务中“胜出”b个任务。

对于较小的语料库，比较简单的模型(例如skip-gram)可以获得更好的结果，而对于较大的语料库，更复杂的模型(如CBOW和顺序)通常更优。对于将词向量用作特征或初始化神经网络的任务，模型选择不会显著影响结果。另外，根据表4的结果，简单模型和复杂模型之间的差距相对较小。因此，简单的模型通常足以应付真实的任务。

C. 训练语料库的影响

我们在实验中使用了不同领域的两个大规模语料库和一个小语料库。这些语料库是维基百科转储(Wiki;https://dumps.wikimedia.org/enwiki)，纽约时报(NYT)语料库(https://catalog.ldc.upenn.edu/LDC2008T19)和IMDB语料库。我们结合Wiki和NYT语料库来获取W&N语料库。Wiki和NYT语料库中的词汇量被设置为最常见的20万个单词。对于每个语料库，不在词汇表中的词语将被忽略。我们在文档层面打乱顺序，以减少某些在线学习模型造成的偏差。从文档级较大的语料库统一抽取小语料库;因此，1000万单词短语语料库是相应的1亿单词短语语料库的子集。

我们选择一个代表性的CBOW模型来分析语料库的影响;其他模型产生类似的结果。图9显示了与各种语料库的最佳结果相比，在不同语料库上训练的嵌入的PGR值。每个任务的最佳PGR是100%。

Corpus	syn	sem	ws	tfl	avg	ner	cnn	pos
NYT 1.2B	93	52	90	98	50	76	85	96
100M	76	30	88	93	46	77	83	86
Wiki 1.6B	92	**100**	**100**	93	51	**100**	86	94
100M	74	65	98	93	47	88	90	83
W&N 2.8B	**100**	89	95	93	50	97	91	**100**
1B	98	87	95	**100**	48	98	90	98
100M	79	63	97	96	51	85	92	86
10M	29	27	76	60	42	49	77	42
IMDB 13M	32	21	55	82	**100**	26	**100**	-13

Fig. 9. 在不同语料库上训练的CBOW模型的PGR值

1) 语料库的大小: 从表7中的结果可以得出结论：当语料库处于同一个领域时，使用更大的语料库可以产生更好的词向量。我们将完整的NYT语料库与其1亿个单词短语子集，维客语料库及其子集，以及W&N语料库与其三个子集进行比较。在几乎所有情况下，较大的语料库都优于较小的语料库。观察到的例外可能是由于评估指标的不稳定性。具体来说，在syn任务(语法测试用例，比如“year:years law: - ”)中，语料库大小是性能的主要驱动因素。显然，不同的语料库以类似的方式使用英语，从而产生类似的语法信息。

2) 语料库领域: 在大多数任务中，语料库领域的影响是主导性的。在不同的任务中，它会以不同的方式影响词向量性能:

●在涉及语义特征评估的任务中，如语义测试用例的类比任务(sem)和语义相似度任务(ws)，维基语料库优于NYT语料库。甚至维基语料库的1亿单词短语子集也可以获得比12亿单词短语的NYT语料库更好的性能。我们认为，维基百科语料库包含更全面的知识，这可能有利于语义任务。

●小型IMDB语料库对avg和cnn任务有利，但在ner任务和pos任务中表现不佳。IMDB语料库由来自IMDB网站的电影评论组成，与用于avg和cnn任务的训练和测试集相同。域内语料库对于这些任务是有帮助的。尤其在平均任务中，域内IMDB语料库实现的PGR几乎是次优语料库的两倍。另一方面，这些电影评论比Wiki和NYT语料库更不正式，这对POS标记(pos)任务是不利的。

为了直观地说明一个域内语料库如何帮助给定的任务，图10显示了几个选定的单词和他们最近的邻近词。IMDB语料库中“电影”的邻近词由“this”，“it”，“thing”等词组成，这意味

着IMDB语料库中的“电影”这个词可以被当作停用词。IMDB语料库中的“科幻”的邻近词包括“科幻小说”的其他缩写，而W&N语料库中的邻居则主要是其他类型。IMDB语料库中的“季节”这个词主要与电视节目相关，而W&N语料库中则主要与体育竞赛季相关。从这些情况中，我们观察到一个域内的语料库可以提高任务的性能，因为它提供了更适合的词向量。

Corpus	movie	Sci-Fi	season
IMDB	film	SciFi	episode
	this	sci-fi	seasons
	it	fi	installment
	thing	Sci	episodes
	miniseries	SF	series
W&N	film	Nickelodeon	half-season
	big-budget	Cartoon	seasons
	movies	PBS	homestand
	live-action	SciFi	playoffs
	low-budget	TV	game

Fig. 10. 在IMDB和W&N语料库进行训练时，某些单词的最近邻居

3) 大小与领域哪个更重要？：那么语料库大小或域这两个方面哪个对于获得更好的词向量更重要？我们应该保持语料纯净还是添加域外语料？为了弄清楚，我们使用了1300万单词短语的IMDB语料库和W&N语料库的一个子集，有1300万个单词短语。两个语料库的子集被混合来训练词向量。图11显示了每个词向量的avg任务的PGR。例如，第三行和第二列中的值为68的条目意味着我们将20%的IMDB语料库与W&N语料库的13万个单词短语子集中的40%相结合以训练词向量并在avg任务上获得68的PGR。对于表中的每一列，元素按语料库大小的升序以及域内纯度的降序列出。实验结果表明，无论哪种规模的IMDB语料库，添加W&N语料库都会降低性能。纯IMDB语料库优于混合语料库。

从结果中可以看出，语料库领域比语料库规模更重要。使用域内语料库可显著提高给定任务的性能，而使用不合适域的语料库会降低性能。对于特定的任务，纯粹的域内语料库比混合域语料库具有更好的性能。对于同一领域的语料库，更大的语料库将获得更好的性能。

D. 训练参数

训练迭代的次数和词向量的维数是训练词向量中两个重要的超参数。

W&N \ IMDB	20%	40%	60%	80%	100%
+0%	91	94	100	100	100
+20%	79	87	91	96	99
+40%	68	86	88	92	98
+60%	65	79	85	88	93
+80%	64	75	84	87	92
+100%	64	70	83	86	88

Fig. 11. 在混合语料上训练时，avg任务的PGR

1) 迭代次数：在机器学习中，早停止是一种正则化，用于解决过度拟合问题。最广泛使用的早停止方法是在验证集丢失峰值时停止迭代器。在词向量训练中，损失度量模型预测目标词的准确率。然而，真正的任务通常不包括预测目标词。因此，验证集的损失只是这些真实任务的代替物，在某些情况下，可能与任务性能不一致。因此，传统的早停止方法是否是停止词嵌入训练的一个很好的指标是值得研究的。

在我们的实验中，我们使用95%的语料库作为训练集，其余5%作为验证集。图12显示了三个训练过程的例子。

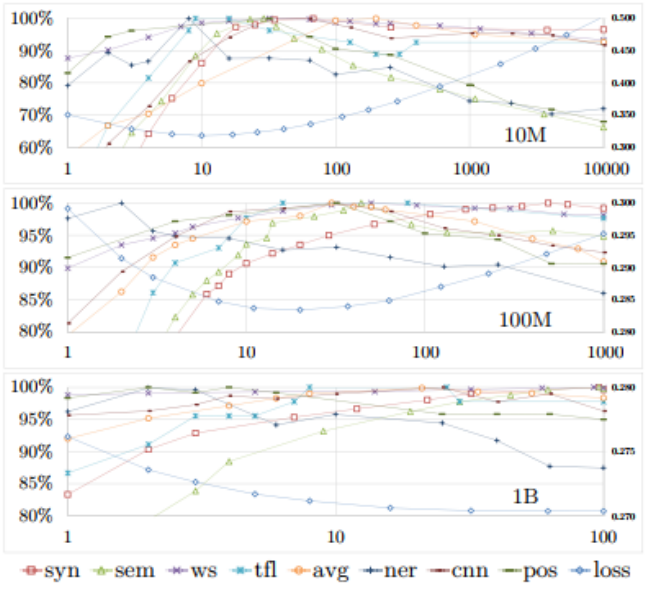

Fig. 12. 在W&N语料库的1000万，1亿和10亿词汇短语子集上，使用CBOW模型对不同迭代次数的词向量进行训练的PGR

从这些情况下，我们发现验证集的损失与NLP任务的性能不一致。还值得注意的是，大多数任务总是表现出相似的峰值。结果表明，我们可以使用一个简单的任务来验证词向量是否在

其他任务上达到顶点。我们考虑W&N语料库的八个任务，七个模型和三个子集的各种组合，从而获得168个元组。如果我们使用一个在验证集损失高峰处停止的策略，我们就"赢得"了89个案例(相对于任务的最高性能达到95&的性能)。如果我们使用在tfl任务的高峰期停止的策略(在我们的实验中最简单的任务)，我们"赢得"了117个案例。

当为特定任务训练单词嵌入时，使用任务开发集确定停止迭代是最佳选择，因为它产生的结果与任务的最终性能最为一致。然而，在某些情况下，测试开发集性能是耗时的，我们的策略可以用作峰值性能的快速近似。

因此，我们可以得出这样的结论：迭代模型直到它在某个简单的任务上达到峰值，对于大多数任务将产生足够优质的词向量。为了针对特定任务训练更适合的词向量，我们可以使用该任务的开发集来决定何时停止迭代。

2) 维数: 为了研究维数对于词向量训练的影响，我们比较了8个评估任务中不同维度的模型。有趣的是，结果显示涉及分析词向量的语义属性的所有任务的表现类似。图13显示了tfl任务的性能。

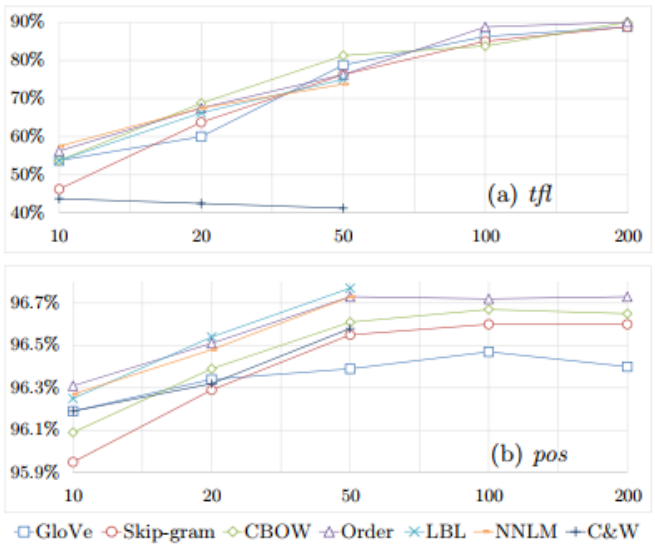

Fig. 13. （a）tfl任务和（b）pos任务，在W&N语料库的10亿词汇短语子集上使用各种模型训练不同维度的词向量的性能

此外，词向量被用作特征或用于初始化的任务也以类似的方式表现。图2b显示了pos任务的性能。我们发现对于语义属性任务来说，更大的维度将得到更好的性能(除了C&W模型)。但是，对于NLP任务来说，维数为50通常就足够了。

VI. 结论

虽然我们没有找到任何具体的设置来训练可以在所有任务上产生最佳性能的词向量(实际上，这样的设置可能不存在)，但是我们提供了几个建议来训练一个好的词向量：

- 对于模型构建来说，更复杂的模型需要更大的训练语料库才能胜过更简单的模型。在大多数情况下，更快(更简单)的模型是足够的，但是在语义任务中，预测目标词的模型(表1中的前5个模型)的性能比不上给目标词和上下文打分的C&W 模型。
- 在合适的领域选择一个语料库时，使用较大的语料库会更好。大规模语料上的训练通常可以提高词向量的质量，对领域语料进行训练可以显着提高特定任务的词向量质量。更重要的是，我们可以看到，语料库领域比语料库规模更重要。
- 验证集上的损失不是一个训练词向量的早期停止指标; 相反，最好的方法是检查在该任务的开发集上的性能。如果评估任务非常耗时，则可以使用其他任务的性能作为替代。
- 对于分析词向量的语义属性的任务，更大的维度可以提供更好的性能。对于利用词向量作为特征或初始化的NLP任务来说，维数为50就足够了。

我们认为训练语料库的领域对于取得良好的表现是非常重要的，但是我们相信词向量中使用的数据源(如单语言语料库，多语言语料库和知识库)也可能是重要的。