向量空间中单词表示的有效估计

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Abstract

本文提出两种新的模型架构，用于从非常大的数据集中计算单词的连续向量表示。这些表示的质量是在单词相似度任务中衡量的，并将结果与之前基于不同类型的神经网络表现最好的技术进行比较。我们观察到精度有了很大的提高，计算成本要低得多，即只需要不到一天的时间就可以学习高质量的词向量从16亿个单词的数据集。此外，我们表明这些向量在我们的测试集上提供了最先进的性能，用于测量语法和语义单词相似性。

1 简介

许多当前的NLP系统和技术将单词视为原子单位——单词之间没有相似的概念，因为它们以词汇表中的索引表示。这种选择有几个很好的理由——简单、鲁棒性和观察到简单的模型是在大量数据上训练的优于在较少数据上训练的复杂系统。一个例子是用于统计语言建模的流行N-gram模型——今天，可以这样做在几乎所有可用的数据(万亿单词[3])上训练n元语法。

然而，这些简单的技术在许多任务中都有其局限性。例如，用于自动语音识别的相关域内数据量是有限的，性能通常由高质量转录语音数据的大小(通常只有数百万个单词)。在机器翻译中，许多语言的现有语料库只包含数十亿个单词或更少的单词。因此，在某些情况下，简单地扩展基本技术不会导致任何重大进展，我们必须专注于更先进的技术。

近年来，随着机器学习技术的发展，在更大的数据集上训练更复杂的模型已经成为可能，它们通常优于简单模型。也许最成功的概念是使用单词的分布式表示[10]。例如，基于神经网络的语言模型显著优于n元模型[1, 27, 17]。

1.1 论文目的

本文的主要目标是介绍一种技术，可以用于从包含数十亿个单词和数百万个单词的词汇表的巨大数据集中学习高质量的词向量。据我们所知，之前提出的架构都没有成功地在数亿个单词上进行训练，单词向量的维数在50 - 100之间。

我们使用最近提出的技术来衡量结果向量表示的质量，期望不仅相似的单词往往彼此接近，但这些单词可以有多个相似度[20]。这在前面的屈折变化的语境中已经观察到语言——例如，名词可以有多个单词结尾，如果我们在原始向量空间的子空间中搜索相似的单词，就有可能找到具有相似结尾的单词[13, 14]。

有些令人惊讶的是，人们发现单词表示的相似性超越了简单的句法规律。使用字偏移技术，在上执行简单的代数操作词向量，例如，向量*("King")* -向量*("Man")* +向量*("Woman")*得到的向量最接近单词的向量表示[20]。

在本文中，我们试图通过以下方法最大化这些向量操作的精度开发新的模型架构，保留单词之间的线性规律。设计了一个新的综合测试集来测量两者语法和语义规律[1]，并表明许多这样的规律是可以学习的精度高。此外，我们还讨论了训练时间和精度如何依赖于词向量的维度和训练数据的数量。

1.2 前期工作

将单词表示为连续向量有很长的历史[10, 26, 8]。一个非常流行的估计神经网络语言模型(NNLM)的模型架构是在[1]上提出的，其中，使用具有线性投影层和非线性隐藏层的前馈神经网络联合学习词向量表示和统计语言模型。这项工作紧随其后的还有很多人。

另一个有趣的NNLM架构在[13, 14]中提出，其中首先使用具有单个隐藏层的神经网络来学习词向量。The 然后用词向量训练NNLM。因此，即使不构建完整的NNLM，也可以学习单词向量。在这项工作中，我们直接扩展了这种架构，并专注于第一步，即使用简单的模型学习词向量。

后来的研究表明，词向量可以用来显著改善和简化许多NLP应用[4, 5, 29]。词向量本身的估计为使用不同的模型架构并在各种语料库[4, 29, 23, 19, 9]上进行训练，并制作了一些结果词向量可供未来的研究和比较[2]。然而，据我们所知，这些架构在训练方面的计算成本要高得多与在[13]中提出的模型相比，除了某些版本的对数双线性模型使用对角权重矩阵[23]。

2 模型架构

人们提出了许多不同类型的模型来估计单词的连续表示，包括众所周知的潜在语义分析(LSA)和潜在狄利克雷分配(LDA)。在本文中，我们重点在由神经网络学习的单词的分布式表示上，正如之前所表明的，它们在保留单词之间的线性规律方面明显优于LSA [20, 31];此外，LDA的计算量也很大在大型数据集上非常昂贵。

与[18]类似，为了比较不同的模型架构，我们首先将模型的计算复杂度定义为完全训练模型所需访问的参数数量。接下来，我们将尽量最大化精度，同时最小化计算复杂度。

对于以下所有模型，训练复杂度与

$$O = E \times T \times Q, \tag{1}$$

其中E是训练次数，T是训练集中单词的数量，Q是针对每个模型架构进一步定义的。常见的选择是$E = 3 - 50$和T高达10亿。所有模型都使用随机梯度下降和反向传播进行训练[26]。

2.1 前馈神经网络语言模型(NNLM)

概率前馈神经网络语言模型已经在[1]上提出。它由输入层、投影层、隐藏层和输出层组成。在输入层，N前面的单词被编码使用1-of- V编码，其中V是词汇表的大小。然后，使用共享投影矩阵将输入层投影到维度为$N \times D$的投影层P。因为只有N输入在任何给定的时间活跃，组成投影层是一个相对便宜的操作。

由于投影层的值很密集，NNLM结构对于投影层和隐藏层之间的计算变得很复杂。对于一个常见的选择$N = 10$，投影层的大小(P)可能是500到2000，而隐藏层的大小H通常是500到1000单位。此外，隐藏层用于计算概率分布在词汇表中的所有单词上，产生一个维度为V的输出层。因此，每个训练示例的计算复杂度为

$$Q = N \times D + N \times D \times H + H \times V, \tag{2}$$

其中主导术语是$H \times V$。然而，人们提出了几种可行的解决方案来避免这种情况;要么使用softmax [25, 23, 18]的分层版本，要么避免使用通过使用在训练期间未归一化的模型完全归一化模型[4, 9]。使用词汇表的二叉树表示，需要评估的输出单元的数量可以登陆$log_2(V)$。因此，大部分复杂性是由术语$N \times D \times H$引起的。

[1]测试集可在www.fit.vutbr.cz/~imikolov/rnnlm/word-test.v1.txt

[2]http://ronan.collobert.com/senna/
http://metaoptimize.com/projects/wordreprs/
http://www.fit.vutbr.cz/~imikolov/rnnlm/
http://ai.stanford.edu/~ehhuang/

在我们的模型中，我们使用分层softmax，其中词汇表表示为哈夫曼二叉树。这与之前的观察一致，即单词的频率对于在神经网络语言模型[16]中获取类很有效。哈夫曼树将简短的二进制代码分配给频繁的单词，这进一步减少了需要评估的输出单元的数量:而平衡二叉树需要$log_2(V)$要评估的输出，基于霍夫曼树的分层softmax只需要$log_2(Unigram_perplexity(V))$。例如，当词汇表大小为100万个单词时，这将导致评估速度提高约两倍。虽然这对神经网络语言模型的加速不是至关重要的，因为计算瓶颈在$N \times D \times H$术语中，但我们稍后将提出没有隐藏层的架构，因此依赖于该架构很大程度上依赖于softmax归一化的效率。

2.2 循环神经网络语言模型(RNNLM)

基于循环神经网络的语言模型被提出，以克服前馈NNLM的某些局限性，如需要指定上下文长度(模型的阶数N)，以及因为理论上，rnn可以有效地表示比浅层神经网络[15, 2]更复杂的模式。RNN模型没有投影层;只有输入层、隐藏层和输出层。这种类型的模型的特别之处在于递归矩阵使用延时连接将隐藏层连接到自身。这允许递归模型形成某种短期记忆作为信息来自过去的数据可以表示为隐藏层状态，该状态根据当前输入和上一个时间步中隐藏层的状态进行更新。

RNN模型每个训练示例的复杂度是

$$Q = H \times H + H \times V, \tag{3}$$

其中单词表示D与隐藏层H具有相同的维度。同样，术语$H \times V$可以通过使用分层softmax有效地简化为$H \times log_2(V)$。最复杂的部分然后来自$H \times H$。

2.3 神经网络并行训练

为了在巨大的数据集上训练模型，我们实现了几个模型之上的一个大规模分布式框架称为怀疑[6]，包括前馈NNLM和本文提出的新模型纸张。该框架允许我们运行多个相同的副本模型并行，每个Replica通过一个集中式的保存所有参数的服务器。对于这种并行训练，我们使用mini-batch异步自适应梯度下降学习率程序称为Adagrad [7]。在这个框架下，通常使用一个数百个或更多的模型副本，每个副本使用数据中心中不同机器上的多个CPU内核。

3 新的对数线性模型

在本节中，我们提出了两种新的模型架构，用于学习单词的分布式表示，试图最小化计算复杂度。上一节的主要观察结果是，大多数复杂性是由模型中的非线性隐藏层造成的。虽然这就是神经网络如此吸引人的原因，但我们决定探索更简单的模型虽然不能像神经网络那样精确地表示数据，但可以有效地在更多的数据上进行训练。

新的架构直接遵循我们早期工作[13, 14]中提出的架构，其中发现神经网络语言模型可以通过两步成功训练:首先，使用simple学习连续的词向量模型，然后在这些词的分布式表示之上训练N-gram NNLM。虽然后来有大量的工作专注于学习词向量，但我们考虑了所提出的方法在[13]是最简单的一个。请注意，相关的模型也在更早的时候提出了[26, 8]。

3.1 连续词袋模型

第一个提出的架构类似于前馈NNLM，其中非线性隐藏层被删除，投影层被所有单词共享(不仅仅是投影矩阵);因此，所有的单词都被投射出来到相同的位置(它们的向量是平均的)。我们将这种架构称为词袋模型，因为历史上单词的顺序不会影响投影。此外，我们也使用来自未来的单词;通过构建一个对数线性分类器，以4个未来单词和4个历史单词作为输入，我们在下一节介绍的任务上获得了最佳性能。其中训练标准是对当前(中间)单词进行正确分类。训练的复杂度是

$$Q = N \times D + D \times log_2(V). \tag{4}$$

我们进一步将此模型表示为CBOW，因为与标准词袋模型不同，它使用上下文的连续分布式表示。模型架构如图1所示。注意重量输入层和投影层之间的矩阵对于所有单词位置都是共享的，与NNLM中的方式相同。

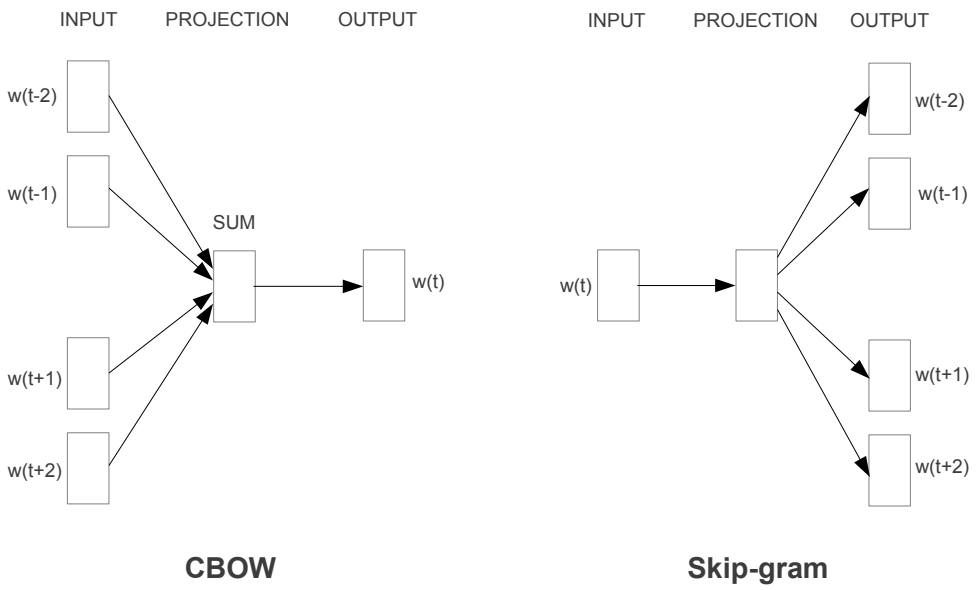

Figure 1: 新模型架构。CBOW架构根据上下文预测当前单词，而Skip-gram根据当前单词预测周围的单词。

3.2 连续**Skip-gram**模型

第二种架构类似于CBOW，但它不是根据上下文预测当前单词，而是试图根据同一句子中的另一个单词最大限度地分类一个单词。更准确地说，我们使用每一种电流将单词作为输入输入到具有连续投影层的对数线性分类器，预测当前单词前后一定范围内的单词。我们发现，增加范围可以提高结果词向量的质量，但它也增加了计算复杂度。由于距离较远的单词与当前单词的相关度通常小于距离较近的单词，因此我们通过从更少的样本中采样来给距离较远的单词更少的权重这些单词在我们的训练示例中。

这种架构的训练复杂度与

$$Q = C \times (D + D \times log_2(V)), \tag{5}$$

其中C是单词之间的最大距离。因此，如果我们选择$C = 5$，对于每个训练单词，我们将在$< 1; C >$范围内随机选择一个数字R，然后使用历史中的R单词和R单词从未来的当前词作为正确的标签。这将要求我们进行$R \times 2$单词分类，将当前单词作为输入，将每个$R + R$单词作为输出。在接下来的实验中，我们使用$C = 10$。

4 结果

为了比较不同版本的词向量的质量，之前的论文通常使用一个表显示示例单词和它们最相似的单词，并直观地理解它们。尽管很容易表明单词与(或许还有其他一些国家)相似，但在更复杂的相似任务中使用这些向量时，则更具挑战性，如下所示。根据之前的观察，单词之间可能有许多不同类型的相似性，例如，单词与在相同意义上相似和类似。另一种关系类型的例子可以是单词对-最大和-最小[20]。我们进一步表示具有相同关系的两对单词作为一个问题，我们可以问:“什么词与small相似，与largest相似的意思是?”

有些令人惊讶的是，这些问题是可以回答的通过对单词的向量表示进行简单的代数运算。要找到一个和相似的单词，就像最大的单词和big相似一样，我们可以简单地计算矢量$X = vector(\text{"}biggest\text{"}) - vector(\text{"}big\text{"}) + vector(\text{"}small\text{"})$。然后，我们在向量空间中搜索通过余弦距离度量的最接近X的单词，并将其作为问题的答案(我们丢弃输入在搜索

Table 1: 以语义-句法词关系测试集中的五类语义题和九类句法题为例。

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

过程中询问单词)。当词向量训练良好时，使用这种方法就有可能找到正确的答案(word $smallest$)。

最后，我们发现，当我们在大量数据上训练高维词向量时，得到的向量可以用来回答单词之间非常微妙的语义关系，例如城市和例如:法国之于巴黎，正如德国之于柏林。具有这种语义关系的词向量可以用于改进许多现有的自然语言处理应用，如机器翻译、信息检索等以及问答系统，并可能使未来其他尚未发明的应用成为可能。

4.1 任务描述

为了衡量词向量的质量，我们定义了一个包含5类语义题型和9类句法题型的综合测试集。每个类别的两个示例如表1所示。总的来说，语义题8869道，句法题10675道。每个类别中的问题分两步创建:首先，手动创建一个相似词对列表;然后，通过连接形成一个大的问题列表两个词对。例如，我们制作了一个包含68个美国大城市及其所属州的列表，并通过随机选择两对单词形成了大约2.5万个问题。我们在测试集中只包含了单个标记词，因此，多词实体不存在(例如纽约)。

评估了所有问题类型的总体准确性，并分别评估了每个问题类型的准确性(语义，语法)。只有最接近的问题才被认为是正确的答案使用上述方法计算得到的词到向量的值与问题中正确的词完全相同;因此，同义词也算作错误。这也意味着达到100%的准确率很可能这是不可能的，因为目前的模型没有任何关于单词形态的输入信息。然而，我们认为词向量在某些应用中的有用性应该与这个精度指标正相关。通过合并可以取得进一步进展关于单词结构的信息，特别是针对句法问题。

4.2 精度最大化

我们使用谷歌新闻语料库来训练词向量。该语料库包含大约6B个标记。我们将词汇表的大小限制为100万个最频繁的单词。显然，我们面临着时间约束优化问题，因为可以预期使用更多的数据和更高维度的词向量将提高准确率。对模型结构的最佳选择进行评估尽可能好的结果我们首先评估了在训练数据子集上训练的模型，词汇表限制在最频繁的30k个单词。使用CBOW得到的结果不同词向量维数选择和训练数据量增加的体系结构如表2所示。

可以看到，在某个点之后，添加更多维度或添加更多训练数据的改进效果逐渐减少。因此，我们必须同时增加向量维数和训练数据量。同时这个观察结果可能看起来微不足道，必须注意的是，目前流行的是在相对大量的数据上训练词向量，但大小不足(如50 - 100)。给定方程4，训练数据量增加两倍导致的计算复杂性增加几乎与向量大小增加两倍相同。

Table 2: 在语义-句法词关系测试集的子集上的准确性，使用来自*CBOW*架构的词向量，且词汇量有限。只有包含最常见的*3*万个单词的问题才会被使用。

Dimensionality / Training words	24M	49M	98M	196M	391M	783M
50	13.4	15.7	18.6	19.1	22.5	23.2
100	19.4	23.1	27.8	28.7	33.4	32.2
300	23.2	29.2	35.3	38.6	43.7	45.9
600	24.0	30.1	36.5	40.8	46.6	50.4

Table 3: *Comparison of architectures using models trained on the same data, with 640-dimensional word vectors. The accuracies are reported on our Semantic-Syntactic Word Relationship test set, and on the syntactic relationship test set of [20]*

Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set [20]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56

对于表2和4中报告的实验，我们使用了随机梯度下降和反向传播的三个训练epoch。我们选择初始学习率为0.025，并线性降低它，使其趋近于0 在最后一次训练结束时。

4.3 模型体系结构比较

首先，我们比较了使用相同的训练数据和相同的640维词向量来推导词向量的不同模型架构。在进一步的实验中，我们使用了新的语义-句法词关系测试集中的全部问题集，即不受3万词汇量的限制。我们还包括在[20]中介绍的侧重于语法的测试集上的结果单词之间的相似度[3]。

训练数据由几个LDC语料库组成，并在[18](3.2亿词，8.2万个词汇)中详细描述。我们使用这些数据来与之前训练过的循环神经网络语言模型进行比较在单个CPU上训练大约需要8周时间。我们使用DistBelief并行训练[6]，使用之前8个单词的历史(因此，NNLM具有相同数量的640个隐藏单元)训练一个前馈NNLM 比RNNLM更多的参数，因为投影层的大小640×8)。

在表3中，可以看到来自RNN(如在[20]中使用的)的词向量主要在语法问题上表现良好。NNLM向量的表现明显优于RNN——这并不奇怪，因为RNNLM中的词向量直接连接到非线性隐藏层。CBOW架构在句法任务上优于NNLM，在语义任务上与NNLM相当。最后，Skip-gram架构在句法任务上的表现比CBOW模型稍差(但仍然比NNLM好)，而且要好得多在语义部分的测试优于其他所有模型。

接下来，我们评估了仅在一个CPU上训练的模型，并将结果与公开的词向量进行了比较。表4给出了比较结果。CBOW模型在大约一天的时间内对谷歌新闻数据的子集进行训练，而Skip-gram模型的训练时间大约为三天。

对于进一步的实验报告，我们只使用了一个训练周期(同样，我们线性降低学习率，以便在训练结束时接近零)。用一个epoch训练一个两倍数据量的模型，比用三个epoch迭代相同的数据得到的结果相当或更好。如表5所示，并提供了额外的小加速。

4.4 模型的大规模并行训练

如前所述，我们在a中实现了各种模型分布式框架DistBelief。下面我们来报道在谷歌新闻6B数据集上训练的多个模型的结果，具有小批量异步梯度下降与自适应学习率程序称

[3]我们感谢Geoff Zweig为我们提供测试集。

Table 4: 语义-句法词关系测试集上公开可用的词向量与我们模型的词向量的比较。使用完整的词汇。

Model	Vector Dimensionality	Training words	Accuracy [%]		
			Semantic	Syntactic	Total
Collobert-Weston NNLM	50	660M	9.3	12.3	11.0
Turian NNLM	50	37M	1.4	2.6	2.1
Turian NNLM	200	37M	1.4	2.2	1.8
Mnih NNLM	50	37M	1.8	9.1	5.8
Mnih NNLM	100	37M	3.3	13.2	8.8
Mikolov RNNLM	80	320M	4.9	18.4	12.7
Mikolov RNNLM	640	320M	8.6	36.5	24.6
Huang NNLM	50	990M	13.3	11.6	12.3
Our NNLM	20	6B	12.9	26.4	20.3
Our NNLM	50	6B	27.9	55.8	43.2
Our NNLM	100	6B	34.2	**64.5**	50.8
CBOW	300	783M	15.5	53.1	36.1
Skip-gram	300	783M	**50.0**	55.9	**53.3**

Table 5: 在相同的数据上训练三个*epoch*的模型和训练一个*epoch*的模型的比较。准确性在完整的语义-语法数据集上得到了报告。

Model	Vector Dimensionality	Training words	Accuracy [%]			Training time [days]
			Semantic	Syntactic	Total	
3 epoch CBOW	300	783M	15.5	53.1	36.1	1
3 epoch Skip-gram	300	783M	50.0	55.9	53.3	3
1 epoch CBOW	300	783M	13.8	49.9	33.6	0.3
1 epoch CBOW	300	1.6B	16.1	52.6	36.1	0.6
1 epoch CBOW	600	783M	15.4	53.3	36.2	0.7
1 epoch Skip-gram	300	783M	45.6	52.2	49.2	1
1 epoch Skip-gram	300	1.6B	52.2	55.1	53.8	2
1 epoch Skip-gram	600	783M	56.7	54.5	55.5	2.5

为Adagrad [7]。我们使用了50到100的模型训练期间的复制品。数字CPU内核的数量是一个估计，因为数据中心的机器与其他生产任务共享，使用情况可能会波动相当多。注意，由于分布式的开销框架中，CBOW模型和Skip-gram模型的CPU使用率分别为比它们的单机更接近实现。结果报告在表6中。

4.5 微软研究句子完成挑战

微软的句子完成挑战最近被引入作为一项任务，以推进语言建模和其他NLP技术[32]。该任务由1040个句子组成，其中每个句子中都少了一个单词，我们的目标是在给出5个合理选择的列表后，选择与句子其余部分最连贯的单词。几种技术的性能已经得到了验证报告了这一集合，包括N-gram模型，基于lsa的模型[32]，对数双线性模型[24]和目前保持最先进水平的循环神经网络组合在此基准上的性能达到55.4%的准确率[19]。

Table 6: 使用*DistBelief*训练的模型比较分布式框架。注意，用*1000*维向量训练*NNLM*需要花费时间太长无法完成。

Model	Vector Dimensionality	Training words	Accuracy [%]			Training time [days x CPU cores]
			Semantic	Syntactic	Total	
NNLM	100	6B	34.2	64.5	50.8	14 x 180
CBOW	1000	6B	57.3	68.9	63.7	2 x 140
Skip-gram	1000	6B	66.1	65.1	65.6	2.5 x 125

Table 7: 微软句子填空挑战赛中模型的比较与组合。

Architecture	Accuracy [%]
4-gram [32]	39
Average LSA similarity [32]	49
Log-bilinear model [24]	54.8
RNNLMs [19]	55.4
Skip-gram	48.0
Skip-gram + RNNLMs	**58.9**

我们已经探索了Skip-gram架构在这项任务上的性能。首先，我们在[32]中提供的50万个单词上训练640维模型。然后，我们通过计算测试集中每个句子的得分使用输入的未知单词，并预测句子中所有周围的单词。最后的句子得分是这些单独预测的总和。根据句子的分数，我们选择最可能的句子。

表7列出了一些以前的结果和新的结果的简短总结。虽然Skip-gram模型本身在这方面的表现并不比LSA相似度模型好，该模型的分数与RNNLMs获得的分数是互补的，加权组合导致了新的最先进的结果58.9%的准确率(开发部分的准确率为59.2%) 在测试集上的占58.7%)。

5 学习到的关系的例子

表8显示了遵循各种关系的单词。我们遵循上面描述的方法:关系定义为两个词向量相减，结果与另一个词相加。例如，巴黎-法国+意大利=罗马。可以看到，准确率相当不错，尽管显然还有很大的改进空间(请注意，使用我们的准确率指标，假设精确匹配，表8中的结果将只得分约60%)。我们相信，在更大维度的数据集上训练的词向量将会表现得更好，并将促进新的创新应用程序的开发。另一种提高准确性的方法是提供多个关系示例。通过使用10个样本而不是1个组成关系向量(我们将每个向量平均在一起)，我们观察到，在语义-语法测试中，最好的模型的准确性提高了约10%。

也可以应用向量操作来解决不同的任务。例如，我们观察到通过计算单词列表的平均向量，并找到距离最远的单词向量，可以很好地选择列表之外的单词。在某些人类智力测试中，这是一种常见的问题类型。显然，使用这些技术仍有很多发现有待发现。

6 结论

本文研究了各种模型在一组句法和语义语言任务上得到的单词向量表示的质量。我们观察到高质量的训练是可能的与流行的神经网络模型(前馈和递归)相比，词向量使用非常简单的模型架构。由于计算复杂度低得多，从更大的数据集中计算出非常精确的高维词向量是可能的。使用DistBelief分布式框架，应该可以进行训练CBOW和Skip-gram甚至在具有一万亿单词的语料库上进行建模，基本上不受词汇量的限制。这比之前公布的同类模型的最佳结果高出几个数量级。

Table 8: 词对关系的示例，使用来自表*4*的最佳词向量*(*在*300*维的*7.83*亿个单词上训练的*Skip-gram*模型*)*。

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

SemEval-2012 task 2 [11]是一个有趣的任务，词向量最近被证明明显优于之前的技术水平。公开可用的RNN向量与其他技术一起使用，使斯皮尔曼等级相关性比之前的最佳结果[31]提高了50%以上。基于神经网络的词向量之前已应用于许多其他NLP任务，例如情感分析[12]和释义检测[28]。这是可以预料的这些应用可以从本文所描述的模型体系结构中受益。

我们的工作表明，词向量可以成功地应用于知识库中事实的自动扩展，以及对已有事实的正确性进行验证。机器结果翻译实验看起来也很有希望。将来，比较我们的技术也会很有趣潜在关系分析[30]等。我们相信我们全面的测试集将有助于研究界改进现有的估计词向量的技术。我们还期望高质量的词向量将成为未来NLP应用的重要组成部分。

7 后续工作

在本文的初始版本完成后，我们发表了用于计算词向量的单机多线程c++代码，同时使用连续词袋和Skip-gram架构[4]。训练速度明显提高比本文前面报道的，即按通常是每小时数十亿个单词超参数选择。我们还出版了140多万本表示命名实体的向量，在100多个上进行训练十亿字。我们的一些后续工作将在即将到来的NIPS 2013上发表论文[21]。

References

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

[2] Y. Bengio, Y. LeCun. Scaling learning algorithms towards AI. In: Large-Scale Kernel Machines, MIT Press, 2007.

[3] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. Large language models in machine translation. In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning, 2007.

[4] R. Collobert and J. Weston. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In International Conference on Machine Learning, ICML, 2008.

[5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa. Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research, 12:2493-2537, 2011.

[6] J. Dean, G.S. Corrado, R. Monga, K. Chen, M. Devin, Q.V. Le, M.Z. Mao, M.A. Ranzato, A. Senior, P. Tucker, K. Yang, A. Y. Ng., Large Scale Distributed Deep Networks, NIPS, 2012.

[4]代码可于`https://code.google.com/p/word2vec/`

[7] J.C. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 2011.

[8] J. Elman. Finding Structure in Time. Cognitive Science, 14, 179-211, 1990.

[9] Eric H. Huang, R. Socher, C. D. Manning and Andrew Y. Ng. Improving Word Representations via Global Context and Multiple Word Prototypes. In: Proc. Association for Computational Linguistics, 2012.

[10] G.E. Hinton, J.L. McClelland, D.E. Rumelhart. Distributed representations. In: Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1: Foundations, MIT Press, 1986.

[11] D.A. Jurgens, S.M. Mohammad, P.D. Turney, K.J. Holyoak. Semeval-2012 task 2: Measuring degrees of relational similarity. In: Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), 2012.

[12] A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In Proceedings of ACL, 2011.

[13] T. Mikolov. Language Modeling for Speech Recognition in Czech, Masters thesis, Brno University of Technology, 2007.

[14] T. Mikolov, J. Kopecký, L. Burget, O. Glembek and J. Černocký. Neural network based language models for higly inflective languages, In: Proc. ICASSP 2009.

[15] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, S. Khudanpur. Recurrent neural network based language model, In: Proceedings of Interspeech, 2010.

[16] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, S. Khudanpur. Extensions of recurrent neural network language model, In: Proceedings of ICASSP 2011.

[17] T. Mikolov, A. Deoras, S. Kombrink, L. Burget, J. Černocký. Empirical Evaluation and Combination of Advanced Language Modeling Techniques, In: Proceedings of Interspeech, 2011.

[18] T. Mikolov, A. Deoras, D. Povey, L. Burget, J. Černocký. Strategies for Training Large Scale Neural Network Language Models, In: Proc. Automatic Speech Recognition and Understanding, 2011.

[19] T. Mikolov. Statistical Language Models based on Neural Networks. PhD thesis, Brno University of Technology, 2012.

[20] T. Mikolov, W.T. Yih, G. Zweig. Linguistic Regularities in Continuous Space Word Representations. NAACL HLT 2013.

[21] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. Accepted to NIPS 2013.

[22] A. Mnih, G. Hinton. Three new graphical models for statistical language modelling. ICML, 2007.

[23] A. Mnih, G. Hinton. A Scalable Hierarchical Distributed Language Model. Advances in Neural Information Processing Systems 21, MIT Press, 2009.

[24] A. Mnih, Y.W. Teh. A fast and simple algorithm for training neural probabilistic language models. ICML, 2012.

[25] F. Morin, Y. Bengio. Hierarchical Probabilistic Neural Network Language Model. AISTATS, 2005.

[26] D. E. Rumelhart, G. E. Hinton, R. J. Williams. Learning internal representations by back-propagating errors. Nature, 323:533.536, 1986.

[27] H. Schwenk. Continuous space language models. Computer Speech and Language, vol. 21, 2007.

[28] R. Socher, E.H. Huang, J. Pennington, A.Y. Ng, and C.D. Manning. Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. In NIPS, 2011.

[29] J. Turian, L. Ratinov, Y. Bengio. Word Representations: A Simple and General Method for Semi-Supervised Learning. In: Proc. Association for Computational Linguistics, 2010.

[30] P. D. Turney. Measuring Semantic Similarity by Latent Relational Analysis. In: Proc. International Joint Conference on Artificial Intelligence, 2005.

[31] A. Zhila, W.T. Yih, C. Meek, G. Zweig, T. Mikolov. Combining Heterogeneous Models for Measuring Relational Similarity. NAACL HLT 2013.

[32] G. Zweig, C.J.C. Burges. The Microsoft Research Sentence Completion Challenge, Microsoft Research Technical Report MSR-TR-2011-129, 2011.