高效的词表示估计在向量空间中

摘要

我们提出了两种新颖的模型结构，用于从非常大的数据集中计算单词的连续向量表示。这些表示的质量在一个单词相似性任务中被衡量，并且结果与之前基于不同类型神经网络的最佳性能的技术相比较。我们观察到在计算成本低得多的情况下精度有很大的提升，即从 16 亿个单词数据集中学习高质量的单词向量只需要不到一天的时间。此外，我们还展示了这些向量在我们的测试集中提供了最先进的性能，用于测量语法和语义词的相似性。

1 介绍

许多现在的 NLP 系统和技术将单词视为原子单位-单词之间没有相似性的概念，因为它们被表示为词汇表中的索引。这种选择有几个好的原因--简单性、健壮性，以及观察到基于大量数据训练的简单模型优于基于较少数据训练的复杂系统。一个例子是用于统计语言建模的 N-gram 模型--今天，可以对几乎所有可用数据训练 N-gram。

然而，简单的技术在许多任务上有它们的极限。比如，用于自动语音识别的相关领域数据是有限的--性能通常由高质量转录的语音数据(通常只有数百万字)的大小决定。在机器翻译中，许多语言现存的语料库只包含几十亿单词甚至更少。所以，我们处于一个境地就是简单提升基础技术不会带来任何显著的进步，我们必须关注更先进的技术。

随着近年来机器学习技术的进步，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。最成功的概念可能是使用单词的分布式表示。例如，基于神经网络的语言模型明显优于 N-gram 模型。

1.1 论文的目标

本文的主要目标是介绍一些技术，这些技术可以用于从拥有数十亿个单词和数百万个词汇表的巨大数据集中学习高质量的单词向量。据我们所知，以前提出的架构都没有在超过几亿个单词上成功训练过，单词向量的维度不高，在 50-100 之间。我们使用最近提出的技术来衡量所产生的向量表示的质量，期望不仅相似的词会倾向于彼此接近，而且词可以有**多个相似度**。这一点早先在转折性语言的背景下已经观察到了--例如，名词可以有多个词尾，如果我们在原始向量空间的子空间中搜索类似的词，就有可能找到有类似词尾的词。

有点令人惊讶的是，我们发现单词表征的相似性超出了简单的句法规律。使用一种对单词向量进行简单代数运算的单词偏移技术，显示出例如向量*("国王")-向量("男人")+向量("女人")*的结果是最接近单词 *女王* 的向量表示。

在本文中，我们试图通过开发新的模型架构来最大限度地提高这些向量操作的准确性，这些模型架构保留了单词之间的线性规律性。我们设计了一个新的综合测试集来测量句法和语义的规律性，并表明许多这样的规律性可以被高精确度地学习。此外，我们还讨论了训练时间和准确性如何取决于词向量的维度和训练数据的数量。

1.2 之前的工作

将单词表示为连续向量由来已久。文献[1]提出了一种非常流行的估计神经网络语言模型(NNLM)的模型结构，其中使用具有线性投影层和非线性隐含层的前馈神经网络来联合学习单词向量表示和统计语言模型。这项工作已经被许多其他人效仿。

[12,13]提出了另一个有趣的 NNLM 体系结构，其中首先使用具有单一隐层的神经网络学习词向量。然后使用词向量训练 NNLM。因此，即使不构建完整的 NNLM，也可以学习词向量。在这项工作中，我们直接扩展了这个体系结构，并只关注第一步，即使用一个简单的模型学习单词向量。

后来的研究表明，词向量可以显著改进和简化许多 NLP 应用。对词向量本身的估计使用不同的模型架构，并在不同的语料库上进行训练，并将得到的一些词向量用

于未来的研究和比较 2。然而，据我们所知，除了某些对数-双线性模型的版本(其中使用对角权矩阵[21])外，这些结构的训练计算成本明显高于[12]中提出的结构。

2 模型结构

许多不同类型的模型被提出用于估计单词的连续表示，包括著名的潜在语义分析(LSA)和潜在狄利克雷分配(LDA)。在本文中，我们将重点放在由神经网络学习的单词的分布式表示上，因为先前的研究表明，在保存单词之间的线性规律方面，神经网络比 LSA 的表现要好得多[19,28];此外，在大型数据集上，LDA 的计算成本非常高。

与[17]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使用的参数数量。接下来，我们将尝试最大限度地提高准确性，同时最小化计算复杂性。

对于以下所有模型，训练复杂度与以下公式成比例

$$O = E \times T \times Q$$

其中 E 为训练周期数，T 为训练集中的单词数，Q为每个模型体系结构进一步定义。通常的选择是$E = 3 - 50$， T 高达 10 亿。所有模型都使用随机梯度下降和反向传播训练。

2.1 前馈神经网络语言模型（NNLM）

文献[1]提出了概率前向神经网络语言模型。它由输入层、投影层、隐藏层和输出层组成。在输入层，使用 1-of-V编码对先前的 N 个单词进行编码，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。由于在任何给定时间只有 N 个输入是活动的，所以组成投影层是一种相对廉价的操作。

由于投影层中的值密集，NNLM 结构在投影层和隐藏层之间的计算变得复杂。对于 $N = 10$ 的常见选择，投影层(P)的大小可能是 500 到 2000，而隐藏层大小 H 通

常是 500 到 1000 个单位。此外，使用隐层计算词汇表中所有单词的概率分布，生成维数为 V 的输出层。因此，每个训练示例的计算复杂度为

$$Q = N \times D + N \times D \times H + H \times V$$

其中主项为$H \times V$。但是，为了避免这种情况，提出了几种实际的解决方案；要么使用 softmax 的分层版本，要么通过使用在训练期间未归一化的模型来完全避免归一化模型。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下降到$log_2(V)$左右。因此，大部分复杂度是由 $N \times D \times H$ 项引起的。

在我们的模型中，我们使用分层 softmax，其中词汇表表示为霍夫曼二叉树。这遵循了之前的观察，即在神经网络语言模型中，词的频率对获得类的效果很好。霍夫曼树将短二叉码分配给频繁的单词，这进一步减少了需要计算的输出单元的数量:平衡二叉树需要计算$log_2(V)$个输出，而基于霍夫曼树的分层 softmax 只需要约$log_2\big(Unigramperplexity(V)\big)$。例如，当词汇量为 100 万个单词时，这将使评估速度加快约两倍。虽然这对神经网络 LMs 来说不是至关重要的加速，因为计算瓶颈在$N \times D \times H$项中，我们稍后将提出没有隐藏层的架构，因此在很大程度上取决于 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 的某些限制，如需要指定上下文长度(模型N的阶数)，并且由于理论上 RNNs 比浅层神经网络能有效地表示更复杂的模式。RNN 模型没有投影层;只有输入、隐藏和输出层。这类模型的特殊之处在于使用延时连接将隐藏层与自身连接起来的循环矩阵。这使得循环模型可以形成某种短期记忆，因为过去的信息可以用隐层状态表示，隐层状态根据当前输入和前一个时间步中的隐层状态进行更新。

RNN 模型的每个训练示例的复杂度为

$$Q = H \times H + H \times V,$$

其中单词表示 D 具有与隐藏层 H 相同的维度。同样，通过使用分级 Softmax，可以将项 $H \times V$ 有效地优化为$H \times log_2(V)$。因此，大部分复杂度来自 $H \times H$。

2.3 神经网络的并行训练

为了在海量数据集上训练模型，我们在一个名为 DistBelef 的大规模分布式框架上实现了几个模型，包括前馈 NNLM 和本文提出的新模型。该框架允许我们并行运行同一模型的多个副本，每个副本通过保存所有参数的中央服务器来同步其梯度更新。对于这种并行训练，我们使用小批量异步梯度下降和一种称为 Adagrad 的自适应学习率过程。在此框架下，通常使用 100 个或更多模型副本，每个模型副本在数据中心的不同机器上使用多个 CPU 核心。

3 新的对数-线性模型

在本节中，我们提出了两个新的模型体系结构，用于学习分布式单词表示，试图将计算复杂度降至最低。上一节的主要观察结果是，大部分复杂度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，这些模型可能无法像神经网络那样精确地表示数据，但可能可以在更多的数据上进行有效的训练。

新的架构直接沿用了我们早期工作中提出的架构，我们发现神经网络语言模型可以通过两个步骤成功训练：首先，使用简单的模型学习连续的单词向量，然后在这些单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单词向量，但我们认为[12]中提出的方法是最简单的。请注意，相关的模型也在更早的时候被提出。

3.1 连续词袋模型

第一个提出的架构类似于前馈 NNLM，其中去除了非线性隐藏层，所有单词共享投影层（不仅仅是投影矩阵）；因此，所有单词都被投影到相同的位置（它们的向量被平均）。我们称这种结构为词袋模型，因为历史中的词顺序不会影响投影。此外，我们还使用来自未来的单词；我们在下一节介绍的任务中获得了最佳性能，方法是

构建一个对数线性分类器，输入有四个未来词和四个历史词，其中训练标准是正确分类当前（中间）词。训练复杂度为

$$Q = N \times D + D \times log_2(V).$$

我们将此模型进一步表示为 CBOW，与标准的词袋模型不同，它使用上下文的连续分布式表示。模型架构如图 1 所示。请注意，输入层和投影层之间的权重矩阵以与 NNLM 中相同的方式为所有单词位置共享。

3.2 连续 Skip-gram 模型

第二个体系结构与 CBOW 类似，但它不是根据上下文预测当前单词，而是试图根据同一句话中的另一个单词最大化地对单词进行分类。更精确地说，我们将当前的每个单词作为一个连续投影层的对数线性分类器的输入，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通常与当前单词的相关性小于接近的单词，我们通过在训练示例中减少对这些单词的采样，从而对距离较远的单词给予较少的权重。

这种结构的训练复杂度与下列式子成比例

$$Q = C \times \big(D + D \times log_2(V)\big),$$

C是单词的最大距离。所以，如果我们选择$C = 5$, 对于每个训练的单词我们会在范围$< 1; C >$中随机选择一个数 R，然后使用历史中 R 个单词和未来的 R 个单词作为正确的标签。这会要求我们做 $R \times 2$ 个词分类，以当前单词为输入，每次 $R + R$ 个单词为输出。在接下来的实验中，我们使用 $C = 10$。

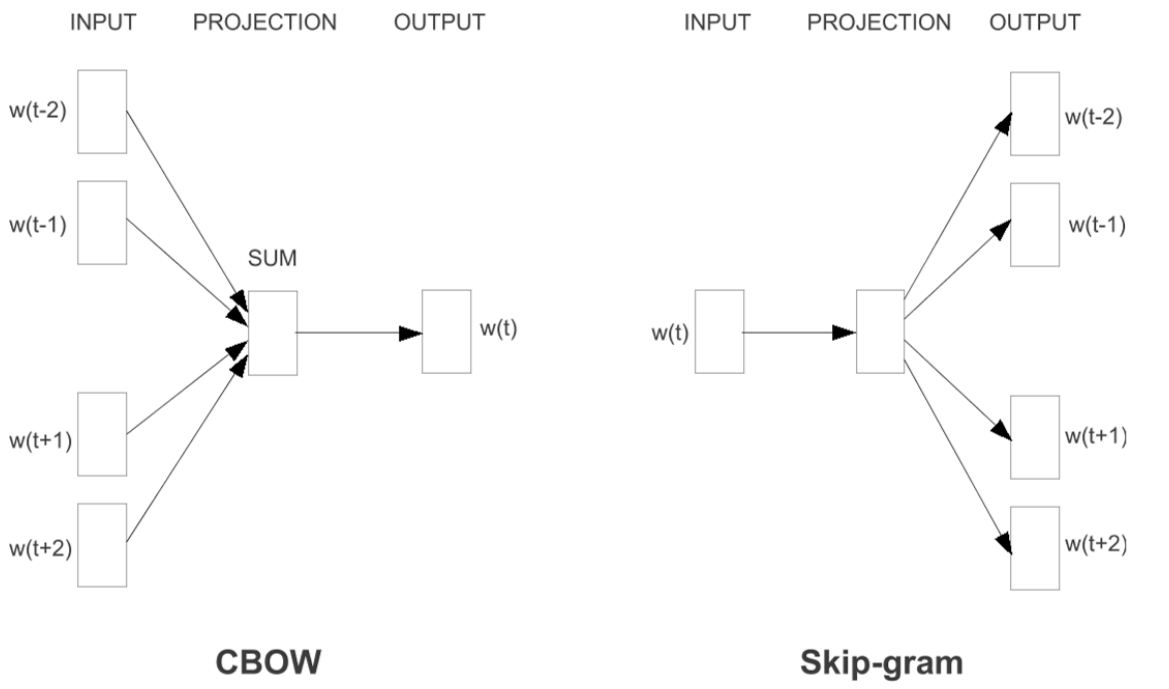

图 1：新模型架构。CBOW 体系结构基于上下文预测当前单词，而 Skip-gram 预测给定当前单词的周围单词。

4 结果

为了比较不同版本的词向量的质量，以前的论文通常使用一个表显示示例词及其最相似的词，并直观地理解它们。尽管很容易表明单词 *France* 与 *Italy* 和其他一些国家相似，但将这些向量置于一个更复杂的相似任务中就更具挑战性了，如下所示。我们根据之前的观察发现，单词之间可以有许多不同类型的相似之处，例如，单词 *big* 与 *bigger* 相似，在同样的意义上，单词 *small* 与 *smaller* 相似。另一种类型的关系可以是单词对 *big - bigger* 和 *small - samllest*。我们进一步将两对具有相同关系的单词作为一个问题来表示，比如我们可以问:“在与 *biggest* 与 *big* 相似的意义上，与 *small* 相似的单词是什么?”

有点令人惊讶的是，这些问题可以通过对单词的向量表示进行简单的代数运算来回答。要找到一个与 *small* 相似的词，就像 *biggest* 与 *big* 相似，我们可以简单地计算

向量 *X = vector("biggest") – vector("big") +vector("small")*。然后，我们在向量空间中搜索用余弦距离度量的与 *X* 最接近的单词，并将其用作问题的答案。当单词向量训练得很好时，使用这种方法就有可能找到正确的答案(单词 *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，比如一个城市和它所属的国家，例如 France 相对于 Paris，就像 Germany 相当于 Berlin。具有这种语义关系的词向量可以用于改进许多现有的 NLP 应用程序，如机器翻译、信息检索和问答系统，并可能使其他未来的应用程序得以发明。我们打算稍后发布一组在大量数据上训练的高质量词向量。

表 1：*语义句法词关系测试集中五种类型的语义问题和九种类型的句法问题。*

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

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 类语义题和 9 类句法题的综合测试集。每个类别中的两个示例如表 1 所示。总共有 8869 道语义题和 10675 道句

法题。每个类别中的问题都是通过两个步骤创建的:首先，手动创建一个相似的词对列表。然后，通过连接两个词对形成一个大的问题列表。例如，我们列出了 68 个美国大城市和它们所属的州，并通过随机选择两组词组成大约 2.5 万个问题。我们在测试集中只包含单个标记词，因此不存在多词实体(例如 *New York*)。

我们评估所有问题类型的整体准确性，并分别评估每个问题类型(语义、句法)。只有当使用上述方法计算出的最接近向量的单词与问题中的正确单词完全相同时，才认为问题是正确的;同义词因此被算作错误。这也意味着达到 100%的准确率是不可能的，因为目前的模型没有任何关于词法的输入信息。然而，我们认为，对于某些应用程序，词向量的有用性应该与这个精度度量呈正相关。通过加入关于单词结构的信息，尤其是关于句法的问题可以取得更大的进步。

4.2 精确度最大化

我们使用谷歌新闻语料库来训练词向量。这个语料库包含约 60 亿个标记。我们将词汇量限制在 100 万个最频繁的词。显然，我们面临着时间有限的优化问题，因为可以预期，使用更多的数据和更高维度的词向量会提高准确率。为了估计快速获得尽可能好的结果的模型架构的最佳选择，我们首先评估了在训练数据的子集上训练的模型，词汇量限制在最频繁的 3 万个词。表 2 显示了使用 CBOW 架构，选择不同的词向量维数和增加训练数据量的结果。

可以看到，在某个点之后，增加更多的维度或增加更多的训练数据提供的改进递减。因此，我们必须同时增加向量的维数和训练数据的数量。虽然这个观察结果看起来微不足道，但必须指出，目前流行的做法是在相对大量的数据上训练词向量，但其大小不够(例如 50 - 100)。对于公式 4，训练数据量增加两倍所带来的计算复杂度增加与向量大小增加两倍几乎相同。对于表 2 和表 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期。我们选择开始学习率 0.025，并线性降低它，使它在最后一个训练周期结束时接近零。

表 2：*对语义-句法词汇关系测试集的子集的准确性，使用来自 CBOW 架构的词汇向量，词汇量有限。只使用了包含最频繁的 30k 个单词的问题。*

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

表 3：*使用基于相同数据训练的模型的体系结构比较，词向量有 640 个维度。准确率是在我们的语义-句法词语关系测试集和句法关系测试集上得出的。*

Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set [19]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56

首先，我们比较了使用相同的训练数据和相同维度的 640 个单词向量来推导单词向量的不同模型体系结构。在进一步的实验中，我们使用了新的语义-句法单词关系测试集中的全问题集，即不限于 3 万个词汇。我们还包括了在[19]中引入的测试集上的结果，该测试集中关注单词之间的句法相似性。

训练数据由几个 LDC 语料库组成，在 [17] 中有详细描述（320M 单词，82K 词汇）。我们使用这些数据与之前训练的循环神经网络语言模型进行了比较，该模型在单个 CPU 上进行训练需要大约 8 周的时间。我们使用 DistBelief 并行训练，训练了具有相同数量的 640 个隐藏单元的前馈 NNLM，使用 8 个先前单词的历史记录（因此，NNLM 比 RNNLM 具有更多参数，因为投影层的大小为 640×8)。

从表 3 中可以看出，来自 RNN（如[19]中使用的）的词向量主要在句法问题上表现良好。NNLM 向量的表现明显好于 RNN - 这并不奇怪，因为 RNNLM 中的词向量直接连接到一个非线性隐藏层。在句法任务上，CBOW 架构比 NNLM 好，而在语

义任务上则大致相同。最后，Skip-gram 结构在句法任务上的效果比 CBOW 模型略差（但仍比 NNLM 好），而在语义部分的测试中则比其他所有模型好很多。

接下来，我们评估了仅使用一个 CPU 训练的模型，并将结果与公开可用的词向量进行比较。表 4 中给出了比较结果。CBOW 模型在谷歌新闻数据的子集上训练了大约一天，而 Skip-gram 模型的训练时间大约是三天。

在进一步报告的实验中，我们只用了一个训练周期（同样，我们线性地减少学习率，使其在训练结束时接近零）。如表 5 所示，用一个历时在两倍的数据上训练一个模型，比在同样的数据上迭代三个历时的结果相当或更好，并提供了额外的小的速度提升。

表 4：*语义-句法单词关系测试集上公开可用的单词向量与我们模型中的单词向量的比较。使用完整的词汇表。*

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**

表 5：*在相同数据上训练了三个 epoch 的模型与一个 epoch 训练的模型的比较。在完整的语义-句法数据集上得出的准确率。

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

4.4 模型的大规模并行训练

如前所述，我们已经在一个叫做 DistBelief 的分布式框架中实现了各种模型。下面我们展示了在谷歌新闻 6B 数据集上训练的几个模型的结果，使用了迷你批次的异步梯度下降和称为 Adagrad 的自适应学习率程序。我们在训练中使用了 50 到 100 个模型副本。CPU 核心的数量是一个估计值，因为数据中心的机器是与其他生产任务共享的，其使用量可能会有很大的波动。请注意，由于分布式框架的开销，CBOW 模型和 Skip-gram 模型的 CPU 使用率比它们的单机实现更接近。结果在表 6 中显示。

表 6：使用 DistBelief 分布式框架训练的模型比较。请注意，使用 1000 维向量训练 NNLM 需要很长时间才能完成。

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

4.5 微软研究院句子补全挑战赛

微软的句子补全挑战最近被推出，作为推进语言建模和其他 NLP 技术的一项任务。这项任务由 1040 个句子组成，每个句子中缺少一个词，目标是在给定的五个合理选择列表中，选择与句子其余部分最一致的词。一些技术的性能已经在这一组上得到了报告，包括 N-gram 模型、基于 LSA 的模型、对数线性模型和循环神经网络的组合，目前在这个基准上保持着 55.4%的准确率的技术水平。

我们探索了 Skip-gram 架构在这个任务上的表现。首先，我们对[29]中提供的 5000 万个词进行 640 维模型的训练。然后，我们通过使用输入的未知词来计算测试集中每个句子的分数，并预测句子中所有的周边词。最后的句子得分是这些单独预测的总和。利用这些句子的分数，我们选择最可能的句子。

表 7 是对以前一些结果和新结果的简短总结。虽然 Skip-gram 模型本身在这个任务上的表现并不优于 LSA 相似性，但这个模型的分数与用 RNNLMs 得到的分数是互补的，加权组合导致了新的最优结果 58.9%的准确性（在数据集的开发部分为 59.2%，在数据集的测试部分为 58.7%）。

表 7：*Microsoft Sentence Completion Challenge 模型的比较和组合*。

Architecture	Accuracy [%]
4-gram [29]	39
Average LSA similarity [29]	49
Log-bilinear model [22]	54.8
RNNLMs [18]	55.4
Skip-gram	48.0
Skip-gram + RNNLMs	**58.9**

5 学习关系的例子

表 8 显示了遵循各种关系的词。我们遵循上述的方法：通过减去两个词的向量来定义关系，然后将结果加到另一个词上。因此，例如，$Paris - France + Italy = Rome$。可以看出，准确率相当不错，尽管显然还有很大的改进余地（注意，使用我们假定完全匹配的准确率指标，表 8 中的结果只能得到 60%左右的分数）。我们相信，在更大维度的数据集上训练的词向量会有优异的表现，并能开发出新的创新应用。另一个提高准确性的方法是提供一个以上的关系例子。通过使用十个例子而不是一个例子来形成关系向量（我们把各个向量平均在一起），我们观察到在语义-句法测试中，我们最好的模型的准确性绝对提高了 10%左右。

表 8: *词对关系的例子，使用表 4 中的最佳词向量(Skipgram 模型在 300 维的 783M 单词上训练)*。

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

也可以应用向量操作来解决不同的任务。例如，我们已经观察到，通过计算一个单词列表的平均向量，并找到最远的单词向量，对选择列表外的单词有很好的准确性。这是某些人类智力测试中的一种流行的问题类型。显然，这些技术仍有待被挖掘。

6 总结

在本文中，我们研究了由各种模型在一系列句法和语义语言任务上得出的词的向量表示的质量。我们观察到，与流行的神经网络模型（包括前馈和递归）相比，使用非常简单的模型架构就可以训练出高质量的词向量。由于计算复杂度低得多，有可能从更大的数据集中计算出非常准确的高维词向量。使用 DistBelief 分布式框架，甚至可以在有一万亿个词的语料库中训练 CBOW 和 Skip-gram 模型，因为词汇量基本上没有限制。这比以前发表的类似模型的最佳结果要大几个数量级。

最近有一项有趣的任务，即 SemEval-2012 任务 2[11]，其中单词向量的表现明显优于以前的技术水平。公开可用的 RNN 向量与其他技术一起使用，使 Spearman 的等级相关性比之前的最佳结果提高了 50%以上[28]。使用本文中描述的模型体系结构，应该可以在类似的任务上获得其他显著的改进。

我们正在进行的工作表明，词向量可以成功地应用于知识库中事实的自动扩展，也可以用于验证现有事实的正确性。机器翻译实验的结果看起来也很有前景。

在未来，将我们的技术与 Latent Relational Analysis[27]和其他技术进行比较也会很有趣。我们相信我们的综合测试集将帮助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 应用的重要构建块。我们打算在未来发布一组高质量的词向量。

引用

[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, 12:24932537, 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.

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

[13] T. Mikolov, J. Kopeck´y, L. Burget, O. Glembek and J. ˇCernock´y. Neural network based language models for higly inflective languages, In: Proc. ICASSP 2009.

[14] T. Mikolov, M. Karafi´at, L. Burget, J. ˇCernock´y, S. Khudanpur. Recurrent neural network based language model, In: Proceedings of Interspeech, 2010.

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

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

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

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

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

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

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

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

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

[24] D. E. Rumelhart, G. E. Hinton, R. J. Williams. Learning internal representations by backpropagating errors. Nature, 323:533.536, 1986.

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

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

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

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

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