词向量空间中的高效估计

托马斯·米科洛夫
谷歌公司，加利福尼亚州山景城
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

摘要

我们提出了两种新的模型架构，用于从非常大的数据集计算词的连续向量表示。这些表示的质量通过一个词相似性任务进行衡量，并将其结果与基于不同类型的神经网络的先前表现最好的技术进行比较。我们观察到在计算成本大幅降低的情况下，准确率有了很大的提高，即从包含 16 亿单词的数据集学习高质量词向量不到一天的时间。此外，我们展示了这些向量在我们的测试集上提供了最先进的性能，用于衡量词的语法和语义相似性。

1　引言

许多当前的自然语言处理系统和技术将单词视为原子单元——单词之间没有相似性的概念，因为它们在词汇表中表示为索引。这种选择有几个很好的理由——简洁性、鲁棒性和观察到的简单模型在大量数据上的训练效果优于在少量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，有可能在几乎所有可用的数据上训练 N-gram（数万亿个单词[3]）。

然而，在许多任务中，简单的技术已经达到了极限。例如，自动语音识别的相关领域数据量是有限的——性能通常由高质量转录语音数据的大小主导（通常只有数百万个单词）。在机器翻译中，许多语言现有的语料库中包含的单词数量只有几十亿或更少。因此，在某些情况下，基本技术的简单扩展不会带来任何显著的进步，我们必须转向更先进的技术。

近年来，随着机器学习技术的发展，现在可以在更大的数据集上训练更复杂的模型，并且它们通常会比简单的模型表现更好。最成功的概念之一是使用分布式词表示[10]。例如，基于神经网络的语言模型显著优于 N-gram 模型[1, 27, 17]。

1.1　论文的目标

本文的主要目标是介绍可以从包含数十亿单词的巨大数据集中学习高质量词向量的技术，并且词汇表中包含数百万个单词。据我们所知，之前提出的任何架构都没有成功地在更大的数据集上进行训练。

少于几百万个单词，词向量的维度在 50 到 100 之间。

我们使用最近提出的衡量生成的向量表示质量的技术，期望不仅相似的词会彼此靠近，而且词可以有多重相似度[20]。这在屈折语的背景下早有观察 - 例如，名词可以有多种词尾，如果我们在一个原始向量空间的子空间中搜索相似的词，有可能找到具有相似词尾的词[13, 14]。

令人惊讶的是，发现词表示的相似性超越了简单的句法规则。使用一种词偏移技术，通过在词向量上执行简单的代数运算，例如，向量(“King”) - 向量(“Man”) + 向量(“Woman”)的结果向量最接近词“Queen”的向量表示[20]。

在这项研究中，我们通过开发能够保留词之间线性规律的新模型架构，尝试最大化这些向量操作的准确性。我们设计了一个新的综合测试集，用于衡量词法和语义规律，证明了许多这样的规律可以用高精度学习。此外，我们讨论了训练时间和准确性如何依赖于词向量的维度和训练数据的数量。

1.2 先前的工作

将词表示为连续向量的历史悠久 [10, 26, 8]。一种非常流行的模型架构是在 [1] 中提出的，其中使用前馈神经网络和线性投影层以及非线性隐藏层来联合学习词向量表示和统计语言模型。这项工作被许多后续工作所跟进。

在[13, 14]中，介绍了一种 NNLM 的另一种有趣的架构，其中单词向量首先使用具有单个隐藏层的神经网络进行学习。然后使用这些单词向量来训练 NNLM。因此，在没有构建完整的 NNLM 的情况下，也可以学习单词向量。在本工作中，我们直接扩展了这种架构，并仅关注第一步，即使用简单的模型来学习单词向量。

后来的研究表明，单词向量可以显著改进和简化许多 NLP 应用[4, 5, 29]。单词向量本身的估计使用了不同的模型架构，并在各种语料库上进行了训练[4, 29, 23, 19, 9]，其中一些生成的单词向量被提供给未来的研究和比较使用。然而，据我们所知，这些架构的训练成本比[13]中提出的架构要高得多，唯一的例外是使用对角权重矩阵的某些版本的对数双线性模型[23]。

2 模型架构

为估计词的连续表示，提出了多种模型，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，我们专注于由神经网络学习的词的分布式表示，因为之前的研究表明，它们在保留词之间的线性规律性方面比 LSA 表现更好[20, 31]；此外，LDA 在大数据集上变得非常计算密集。

类似于[18]，为了比较不同的模型架构，我们首先将模型的计算复杂性定义为需要访问以完全训练模型的参数数量。接下来，我们将尝试在最小化计算复杂性的前提下最大化准确性。

---

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

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

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 是训练轮数，T 是训练集中的单词数，Q 对于每种模型架构有进一步定义。常见的选择是 E = 3 − 50，T 最多可达十亿。
所有模型均使用随机梯度下降和反向传播 [26]。

2.1 前馈神经网络语言模型（NNLM）

在 [1] 中提出了概率前馈神经网络语言模型。它由输入、投影、隐藏和输出层组成。在输入层，使用 1-of-V 编码对 N 个先前的词进行编码，其中 V 是词汇量的大小。然后，输入层通过共享的投影矩阵投影到具有维度 N × D 的投影层 P 中。由于任何给定时间只有 N 个输入是激活的，因此投影层的组合是一个相对便宜的操作。
当计算投影层和隐藏层之间的值时，NNLM 架构变得复杂，因为投影层中的值是密集的。对于常见的 N = 10 的选择，投影层（P）的大小可能在 500 到 2000 之间，而隐藏层大小 H 通常为 500 到 1000 个单位。此外，隐藏层用于计算词汇表中所有词的概率分布，从而导致输出层的维度为 V。因此，每次训练示例的计算复杂度为

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

主导项是 H × V 。然而，提出了一些实用的解决方案来避免这一点；要么使用分层版本的 softmax [25, 23, 18]，要么完全不使用归一化模型，在训练过程中不归一化模型 [4, 9]。使用词汇的二叉树表示，需要评估的输出单元数量可以减少到大约 log(V)。因此，大部分复杂性是由 N × D × H 这一项引起的。
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。这遵循了先前观察到的单词频率对于获得神经网络语言模型中的类效果很好的结论[16]。霍夫曼树为频繁的单词分配短的二进制码，这进一步减少了需要评估的输出单元的数量：与平衡的二叉树需要评估 log(V)个输出相比，基于霍夫曼树的分层 softmax 只需要大约 log(U nigram perplexity(V))个输出。例如，当词汇表大小为一百万个单词时，这会导致评估速度大约提高两倍。虽然这对于神经网络 LM 来说并不是关键的加速，因为计算瓶颈在于 N×D×H 项，但我们稍后将提出没有隐藏层的架构，因此对 softmax 归一化的效率依赖很大。

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已被提出，以克服前馈 NNLM 的一些局限性，例如需要指定上下文长度（模型的阶数 N），并且理论上循环神经网络可以更有效地表示比浅层神经网络更复杂的模式 [15, 2]。循环模型没有投影层；只有输入、隐藏和输出层。这种类型模型的特殊之处在于连接隐藏层到自身的循环矩阵，使用时间延迟连接。这使得循环模型能够形成某种短期记忆，因为过去的信息可以通过隐藏层状态来表示，该状态是基于当前输入和前一时间步隐藏层状态更新的。

循环模型每个训练示例的复杂度为

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

其中词表示 D 的维度与隐藏层 H 相同。同样，通过使用分层 softmax，术语 H × V 可以高效地减少到 H × log(V)。然后大部分复杂度来自 H × H。

2.3 神经网络的并行训练

为了在大量数据集上训练模型，我们基于一个名为 DistBelief [6]的大规模分布式框架实现了多个模型，包括前向 NNLM 和本文中提出的新模型。该框架允许我们并行运行相同模型的多个副本，并且每个副本通过集中式服务器同步其梯度更新，该服务器保存所有参数。对于这种并行训练，我们使用带有自适应学习率过程的 mini-batch 异步梯度下降[7]。在该框架下，通常会使用一百个或更多的模型副本，每个副本在数据中心的不同机器上使用多个 CPU 核心。

3　新的 Log-linear 模型

在本节中，我们提出了两种新的模型架构，用于学习词的分布式表示，并试图最小化计算复杂度。上一节的主要观察是，模型中的非线性隐藏层导致了大部分的复杂性。虽然这正是使神经网络如此吸引人的原因，但我们决定探索一些更简单的模型，这些模型可能无法像神经网络那样精确地表示数据，但可能可以更有效地在大量数据上进行训练。

新的架构直接遵循我们早期工作[13, 14]中提出的设计，在那里发现神经网络语言模型可以在两个步骤中成功地进行训练：首先，使用简单的模型学习连续词向量，然后在这些词的分布式表示之上训练 N-gram NNLM。尽管后来有大量关于学习词向量的工作，但我们认为[13]中提出的方法是最简单的一种。

请注意，相关模型也早在[26, 8]中被提出。

3.1　连续词袋模型

第一种提出的架构类似于前馈 NNLM，其中非线性隐藏层被移除，并且投影层被共享用于所有单词（而不仅仅是投影矩阵）；因此，所有单词都被投影到相同的位置（它们的向量被平均）。我们称这种架构为词袋模型，因为历史中单词的顺序不影响投影。此外，我们还使用未来的单词；我们通过构建一个输入包含四个未来单词和四个历史单词的对数线性分类器，在下一节介绍的任务中获得了最佳性能，其中训练准则是在正确分类当前（中间）单词的情况下进行分类。训练复杂度随后为

(4) 我们进一步将此模型称为 CBOW，因为它不像标准的词袋模型，使用了连续的分布式表示的上下文。模型架构如图 1 所示。请注意，输入和投影层之间的权重矩阵在相同的方式下被共享，就像在 NNLM 中一样。

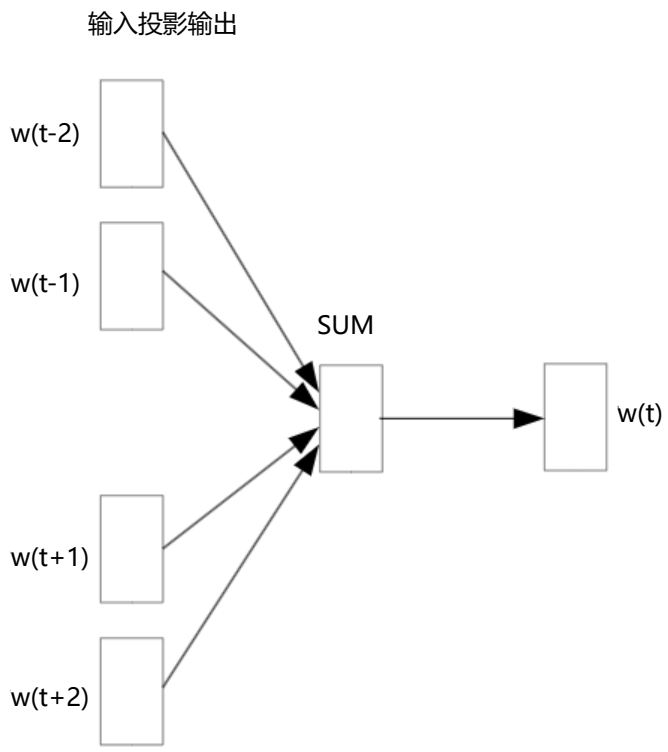
3.2　连续跳词模型

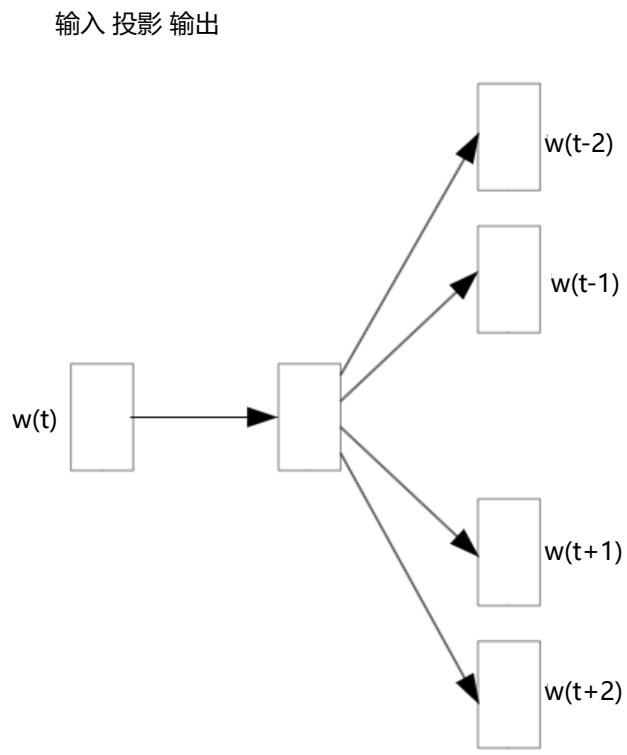
The second architecture 类似于 CBOW，但不是基于上下文来预测当前词，而是尝试通过另一个句子中的词来最大化对词的分类。更精确地说，我们使用每个当前词作为连续投影层的逻辑线性分类器的输入，并预测当前词前后一定范围内的词。我们发现增加范围可以提高生成的词向量质量，但也增加了计算复杂度。由于距离当前词较远的词通常与其关系不如距离较近的词密切，因此我们在训练示例中较少采样这些词，从而给予它们较少的权重。

该架构的训练复杂度与

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

where C 是单词的最大距离。因此，如果我们选择 C = 5，对于每个训练单词，我们将随机选择一个在 <1; C> 范围内的数 R，然后使用 R 个历史单词和





CBOW Skip-gram

然后使用历史中的 R 个单词，并参见图 1：新的模型架构。CBOW 架构基于上下文预测当前单词，而 Skip-gram 则根据当前单词预测周围的单词。

使用当前单词未来 R 个单词作为正确标签。这将需要我们进行 R × 2 个单词分类，其中当前单词作为输入，每个 R + R 个单词作为输出。在后续实验中，我们使用 C = 10。

4　结果

为了比较不同版本词向量的质量，以往的研究通常使用表格展示一些示例单词及其最相似的单词，并直观地理解它们。虽然展示单词 France 与 Italy 相似以及可能与其他一些国家相似是容易的，但在更复杂的相似性任务中，如下面所示，就更加具有挑战性。我们遵循以往的研究观察，即单词之间的相似性可以有多种类型，例如，单词 big 与其更大的形式 bigger 在意义上类似于 small 与其更小的形式 smaller。另一种类型的单词关系示例可以是 big - biggest 和 small - smallest [20]。我们进一步用具有相同关系的两对单词表示一个问题，因为我们可以问：” 与 biggest 与 big 之间的相似性相同，small 的相似词是什么？ “

"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”在“biggest”与“big”相似关系中相似的词，我们可以简单地计算向量 X = 向量（“biggest”）− 向量（“big”）+ 向量（“small”）。然后，在向量空间中搜索与 X 的余弦距离最接近的词，并将其作为问题的答案（在搜索过程中我们忽略输入问题中的词）。当词向量训练良好时，可以使用这种方法找到正确的答案（词 smallest）。”

"最后，我们发现，当我们使用大量数据训练高维词向量时，生成的向量可以用来回答非常微妙的语义关系，例如一个城市和它所属的国家，比如“France”对应“Paris”就像“Germany”对应“Berlin”。具有这种语义关系的词向量可以用来改进许多现有的自然语言处理应用，如机器翻译、信息检索和问答系统，并可能使其他未来应用得以实现。”

表 1：Semantic Syntactic Word Relationship 测试集中五种类型语义和九种类型句法问题的示例。

关系类型 词对 1 词对 2 常见的首都 雅典 希腊 奥斯陆 挪威 所有首都 喀山 哈萨克斯坦 布拉柴维尔津巴布韦 货币 安哥拉 卡泽纳 伊朗 里亚尔 城市-州 芝加哥 伊利诺伊州 斯托克顿 加利福尼亚州 男女 兄弟 姐妹 孙子 妻子 Adjective to Adverb 明显 明显地 快速 快速地 反义词 可能地 不可能地 道德的 非道德的 比较级 伟大 更伟大 坚硬 更坚硬 最高级 容易 最容易 幸运 最幸运 现在分词 想 想着 读 读着 国家形容词 瑞士 瑞士的 柬埔寨 柬埔寨的 过去式 走 走了 游泳 游泳了 复数名词 鼠 鼠类 美元 美元 复数动词 工作 工作 谈话 谈话				

4.1　任务描述

为了衡量词向量的质量，我们定义了一个综合测试集，其中包含五种类型的语义问题和九种类型的句法问题。每个类别中的两个示例见表 1。总体来说，有 8869 个语义问题和 10675 个句法问题。每个类别的问题是在两步中创建的：首先，手工创建了一组相似的词对。然后，通过连接两组词对形成一个大列表。例如，我们列出了 68 个大型美国城市及其所属的州，并随机挑选两组词对形成了约 2500 个问题。我们的测试集中只包括单个词，因此多词实体（如纽约）不存在。

我们评估所有问题类型的整体准确性，并且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词向量与问题中的正确词完全相同时，才认为问题被正确回答；因此同义词被视为错误。这也意味着达到 100%的准确性可能是不可能的，因为当前的模型没有任何关于词形信息的输入。然而，我们认为对于某些应用而言，词向量的有用性应该与这个准确性指标正相关。进一步的进步可以通过引入关于词结构的信息来实现，尤其是对于句法问题。

4.2　准确性最大化

我们使用了 Google 新闻语料库来训练词向量。该语料库包含大约 60 亿个词元。我们限制词汇表大小为最常见的 100 万个单词。显然，我们面临一个时间约束优化问题，因为可以预期使用更多的数据和更高维的词向量将提高准确性。为了估计在短时间内获得尽可能好的结果的最佳模型架构，我们首先评估了在训练数据子集上训练的模型，词汇表限制为最常见的 30000 个单词。表 2 显示了使用 CBOW 架构、不同词向量维度选择以及不断增加的训练数据量的结果。

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效果逐渐减弱。因此，我们必须同时增加词向量维度和训练数据量。虽然这一观察可能看似显而易见，但必须注意的是，目前流行的做法是在相对大量的数据上训练词向量，但词汇表大小却不足。

表 2：使用 CBOW 架构且词汇量受限的词向量，在语义-句法词关系测试集子集上的准确性。仅使用包含最频繁 30,000 个词中的词的问题。

维度 / 训练词	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：使用相同数据训练的模型的架构比较，使用 640 维词向量。准确率在我们的语义-句法词关系测试集和 [20] 的句法关系测试集上报告。

模型	语义-句法词关系测试集 MSR 词相关性		架构	语义 准确率 [%]	句法 准确率 [%]	测试集 [20]
RNNLM						
	9			36	35	
NNLM	23			53	47	
CBOW	24			64	61	
跳过克隆	55			59	56	

(例如 50 - 100)。根据公式 4，将训练数据量翻倍导致的计算复杂度增加与将向量大小翻倍导致的计算复杂度增加大致相同。

对于表 2 和表 4 中报告的实验，我们使用了三轮随机梯度下降和反向传播。我们选择了起始学习率 0.025，并且线性地降低它，使得在最后一轮训练结束时接近于零。

4.3 模型架构比较

首先，我们使用相同的训练数据和相同的词向量维度 640 来比较不同模型架构以推导词向量。在后续的实验中，我们使用新语义-句法词关系测试集中的完整问题集，即不限于 30k 词汇。我们还包含了一个在 [20] 中引入的测试集的结果，该测试集专注于词之间的句法相似性。

训练数据包括几个 LDC 语料库，并且在[18]中有详细描述（3.2 亿词，8.2 万词汇量）。我们使用这些数据来提供与之前训练的循环神经网络语言模型的比较，该模型在单个 CPU 上训练大约需要 8 周时间。我们使用 DistBelief 并行训练[6]训练了一个具有相同数量 640 个隐藏单元的前馈 NNLM，并使用了 8 个先前词的历史记录（因此，NNLM 比 RNNLM 有更多的参数，因为投影层的大小为 640×8）.

在表 3 中，可以看到 RNN（如[20]中所用）的词向量在句法问题上表现良好。NNLM 向量的表现显著优于 RNN——这并不令人惊讶，因为 RNNLM 中的词向量直接连接到一个非线性隐藏层。CBOW 架构在句法任务上比 NNLM 效果更好，而在语义任务上效果相当。最后，Skip-gram 架构在句法任务上的表现略逊于 CBOW 模型（但仍优于 NNLM），而在测试的语义部分则比所有其他模型表现更好。

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了比较。比较结果见表 4。CBOW 模型是在子集

[3]我们感谢 Geoff Zweig 提供给我们测试集。

表 4：在语义-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。使用完整的词汇表。

模型	向量训练 准确率 [%] 维度词汇				
			语义 语法 总数		
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：在相同数据上训练三轮的模型与训练一轮的模型的比较。准确率在语义-语法数据集上报告。

Model	向量训练 准确率 [%] 训练时间 维度词 [天]				
			语义 语法 总数		
3 轮次 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					

在大约一天内处理完 Google 新闻数据，而 Skip-gram 模型的训练时间大约为三天。

对于后续报告的实验，我们仅使用了一个训练周期（再次说明，我们线性地降低学习率，使其在训练结束时接近零）。使用两倍数据量训练模型并在一个周期内完成，其结果与在相同数据上迭代三次相比，具有可比性或更优，如表 5 所示，还提供了额外的小幅加速。

4.4 大规模并行训练模型

如前所述，我们已在名为 DistBelief 的分布式框架中实现了多种模型。下面报告了在 Google 新闻 6B 数据集上训练的几种模型的结果，使用了小批量异步梯度下降和称为 Adagrad 的自适应学习率过程 [7]。训练过程中使用了 50 到 100 个模型副本。CPU 核心数是

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 将花费太长时间.

模型	向量训练 准确率 [%] 训练时间 维度词					[天 x CPU 核]
			语义	语法	总数	
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

表 7：在 Microsoft Sentence Completion Challenge 上的模型比较与组合。

架构	准确率 [%]
4-gram [32]	39
平均 LSA 相似度 [32]	49
对数双线性模型 [24]	54.8
RNNLMs [19]	55.4
Skip-gram	48.0
Skip-gram + RNNLMs	58.9

估计，因为数据中心的机器与其他生产任务共享，使用量可能会有很大波动。请注意，由于分布式框架的开销，CBOW 模型和 Skip-gram 模型的 CPU 使用率彼此非常接近，而单机实现则不然。结果见表 6。

4.5 微软研究句子完成挑战

微软句子完成挑战最近被引入，作为推进语言建模和其他 NLP 技术的任务[32]。该任务包含 1040 个句子，每个句子中缺失一个单词，目标是在给定五个合理选择的情况下，选择与句子其余部分最连贯的单词。已经在这组数据上报告了多种技术的性能，包括 N-gram 模型、基于 LSA 的模型[32]、对数双线性模型[24]以及当前在基准测试中保持最佳性能（55.4%准确率）的递归神经网络组合[19]。

我们在该任务上探索了 Skip-gram 架构的性能。首先，我们在[32]提供的 50M 词上训练 640 维模型。然后，我们通过在输入中使用未知词计算测试集每个句子的得分，并预测句子中的所有周围词。最终句子得分是这些单独预测的总和。使用句子得分，我们选择最有可能的句子。

表 7 中呈现了一些先前结果的简要总结以及新结果。虽然 Skip-gram 模型本身在该任务上的表现并不优于 LSA 相似度，但该模型的得分与 RNNLMs 获得的得分互补，加权组合后达到了新的最佳结果 58.9%准确率（开发集部分为 59.2%，测试集部分为 58.7%）。

5 学习到的关系示例

表 8 展示了各种关系下的单词。我们遵循上述方法：关系通过减去两个单词向量定义，然后将结果加到另一个单词上。例如，巴黎 - 法国 + 意大利 = 罗马。如您所见，准确度相当不错，尽管显然还有很大的改进空间（请注意，使用我们的准确度指标）

表 8：使用表 4 中最佳单词向量（在 783M 单词上训练的 Skipgram 模型，维度为 300）的单词对关系示例。

关系 示例 1 示例 2 示例 3 法国 - 巴黎 意大利: 罗马 日本: 东京 佛罗里达: 坦帕比奇 小 - 更大
小: 更大 冷: 更冷 快: 更快 迈阿密 - 佛罗里达 巴尔的摩: 马里兰 达拉斯: 德克萨斯 科纳: 夏威夷 爱因斯坦 - 科学家 梅西: 中场球员 莫扎特: 小提琴家 毕加索: 画家 希拉克 - 法国 布鲁斯科尼: 意大利 默克尔: 德国 Koizumi: 日本 铜 - Cu 锌: Zn 金: Au 铀: 钚 布鲁斯科尼 - 西里奥 布鲁斯科尼: 尼古拉 普京: 奥梅尔德耶夫 奥巴马: 巴拉克 微软 - Windows 谷歌: 安卓 IBM: Linux 苹果: iPhone 微软 - Ballmer 谷歌: 谷歌 IBM: McNealy 苹果: 乔布斯 日本 - 寿司 德国: 布拉特温 芝加哥: 法国 美国: 比萨

假设完全匹配，表 8 中的结果得分仅为约 60%。我们认为，即使在更大数据集和更高维度上训练的词向量将表现得更好，并将使开发新的创新应用成为可能。另一种提高准确性的方法是提供关系的多个示例。通过使用十个示例而不是一个来形成关系向量（我们将各个向量平均），我们观察到我们的最佳模型在语义句法测试中的准确率绝对提高了约 10%。

还可以将向量运算应用于解决不同的任务。例如，我们通过计算单词列表的平均向量并找到最远的单词向量，观察到选择列表外单词的良好准确率。这是一种在某些人类智能测试中常见的问题类型。显然，仍然有很多发现可以使用这些技术来实现。

6 结论

在本文中，我们研究了由各种模型在一组句法和语义语言任务中提取的词向量的质量。我们观察到，与流行的神经网络模型（包括前馈和递归模型）相比，使用非常简单的模型架构也可以训练出高质量的词向量。由于计算复杂度大大降低，可以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，即使在包含一万亿单词的语料库上，也可以训练 CBOW 和 Skip-gram 模型，词汇量可以基本上无限大。这比之前发表的类似模型的最佳结果大几个数量级。

An interesting task where the word vectors have recently been shown to significantly outperform the previous state of the art is the SemEval-2012 Task 2 [11]。公开可用的 RNN 向量与其他技术一起使用，实现了超过 50%的 Spearman 秩相关性提升，超过了之前的最佳结果 [31]。基于神经网络的词向量之前已被应用于许多其他 NLP 任务，例如情感分析[12]和同义句检测 [28]。可以预期，这些应用可以从本文描述的模型架构中受益。

我们的持续工作表明，词向量可以成功应用于知识库中事实的自动扩展，以及现有事实正确性的验证。机器翻译实验的结果也非常有前景。未来，我们也希望将我们的技术与潜在关系分析[30]等方法进行比较。我们相信，我们全面的测试集将有助于研究社区改进现有的词向量估计技术。我们还期望高质量的词向量将成为未来自然语言处理应用的重要构建块。

7 后续工作

在最初版本的论文完成后，我们发布了单机多线程 C++代码，用于计算词向量，使用了连续词袋模型和跳词模型两种架构。训练速度比论文中早前报告的要高得多，例如，对于典型的超参数选择，每小时可以处理数十亿个词。我们还发布了超过 140 万表示命名实体的向量，这些向量是基于超过 1000 亿个词训练得到的。我们的一些后续工作将在即将发表的 NIPS 2013 论文[21]中发布。

参考文献

[1] Y. Bengio, R. Ducharme, P. Vincent. 一种基于神经概率的语言模型. 机器学习研究杂志, 3:1137-1155, 2003. [2] Y. Bengio, Y. LeCun. 机器学习算法的扩展以达到人工智能. 在: 大规模核机器, MIT 出版社, 2007. [3] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. 大型语言模型在机器翻译中的应用. 在: 语料库语言学和计算语言学联合会议论文集, 2007. [4] R. Collobert 和 J. Weston. 一种自然语言处理的统一架构: 多任务学习的深层神经网络. 在国际机器学习会议, ICML, 2008. [5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu 和 P. Kuksa. 从零开始的自然语言处理. 机器学习研究杂志, 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., 大规模分布式深度网络, NIPS, 2012.

[7] J.C. 杜奇, E. 哈赞, 和 Y. 求格. 在线学习和随机优化的自适应次梯度方法. 机器学习研究杂志, 2011.

[8] J. 埃尔曼. 时间中的结构. 认知科学, 14, 179-211, 1990. [9] 李宏毅, R. 梅科, C. D. 曼宁, 和 安德鲁 Y. 孟. 通过全局上下文和多个词原型改进词表示. 在: 计算语言学协会会议论文集, 2012.

[10] G.E. 希顿, J.L. 麦克莱兰, D.E. 沃尔默赫. 分布式表示. 在: 并行分布处理: 认知微结构的探索. 第 1 卷: 基础, MIT 出版社, 1986. 分布式处理: 认知微结构的探索. 第 1 卷: 基础, MIT 出版社, 1986.

[11] D.A. 杰尔 gens, S.M. 摩罕默德, P.D. 特尼, K.J. 荷利奥克. Semeval-2012 任务 2: 测量关系相似度的度量. 在: 第 6 届语义评价国际研讨会（SemEval 2012）会议论文集, 2012.

[12] A.L. 马斯, R.E. 大利, P.T. 菲姆, D. 黄, A.Y. Ng, 和 C. 波茨. 学习词向量进行情感分析. 在: ACL 会议论文集, 2011. [13] T. 米科洛夫. 捷克语的语音识别基于语言模型, 硕士论文, 布尔诺技术大学, 2007.

[14] T. 米科洛夫, J. 科佩克, L. 布尔格, O. 格莱姆贝克和 J. ˇ切诺克. 高屈折语基于神经网络的语言模型, 在: ICASSP 2009 会议论文集.

[15] T. 米科洛夫, M. 卡拉菲亚特, L. 布尔热特, J. ˇ切诺奇, S. 哈杜南普. 基于循环神经网络的语言模型, 在: 第 11 届国际语音会议论文集, 2010 年.

[16] T. 米科洛夫, S. 科姆布林克, L. 布尔热特, J. ˇ切诺奇, S. 哈杜南普. 基于循环神经网络的语言模型的扩展, 在: 第 2011 届国际声学、说话和信号处理会议论文集.

[17] T. 米科洛夫, A. 德奥拉斯, S. 科姆布林克, L. 布尔热特, J. ˇ切诺奇. 先进语言建模技术的实证评估与组合, 在: 第 11 届国际语音会议论文集, 2011 年.

[4]代码可在 https://code.google.com/p/word2vec/ 获取.

[18] T. Mikolov, A. Deoras, D. Povey, L. Burget, J. ˇCernock´y. 大规模训练策略，自动语音识别和理解会议，2011。
神经网络语言模型，自动语音识别和理解会议，2011。

[19] T. Mikolov. 基于神经网络的语言模型。布拉格技术大学博士论文，2012。[20] T. Mikolov, W.T. Yih, G. Zweig. 连续空间词表示中的语言规律。NAACL HLT 2013。[21] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. 分布式词和短语表示。NIPS 2013 接受。[22] A. Mnih, G. Hinton. 用于统计语言建模的新图形模型。ICML,

2007.

[23] A. Mnih, G. Hinton. 可扩展的分层分布式语言模型。神经信息处理系统进展 21，MIT 印刷，2009。[24] A. Mnih, Y.W. Teh. 用于训练神经概率语言模型的快速简单算法。ICML，2012。[25] F. Morin, Y. Bengio. 分层概率神经网络语言模型。AISTATS,

2005.

[26] D. E. 沃尔梅赫特, G. E. 希顿, R. J. 威廉姆斯. 通过反向传播误差学习内部表示. 自然, 323:533.536, 1986. [27] H. 施文克. 连续空间语言模型. 计算机语音和语言, 第 21 卷,

2007.

[28] R. 托切, E.H. 黄, J. 彭宁顿, A.Y. 内格, 和 C.D. 曼宁. 动态池化和展开递归自动编码器用于同义句检测. 在 NIPS, 2011. [29] J. 图里安, L. 拉廷诺, Y. 本吉奥. 词表示: 半监督学习的简单而通用方法. 在: 语言学协会会议论文集, 2010. [30] P. D. 特纳. 通过潜在关系分析测量语义相似性. 在: 国际人工智能联合会议论文集, 2005. [31] A. 吉拉, W.T. 伊, C. 迈克, G. 周维, T. 米科洛夫. 结合异构模型测量关系相似性. NAACL HLT 2013. [32] G. 周维, C.J.C. 布尔格斯. 微软研究句子完成挑战, 微软研究技术报告 MSR-TR-2011-129, 2011.