利用神经网络进行序列到序列学习

Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

摘要

深度神经网络（DNN）是一种功能强大的模型，在困难的学习任务中表现出色。虽然 DNN 在有大量标注训练集的情况下都能很好地工作，但它们不能用于将序列映射到序列。 在本文中，我们提出了一种通用的端到端序列学习方法，这种方法对序列结构的假设最少。 我们的方法使用多层长短期记忆（LSTM）将输入序列映射到一个固定维度的向量，然后使用另一个深度 LSTM 从向量中解码目标序列。 我们的主要成果是，在 WMT'14 数据集中的英法互译任务中，LSTM 生成的译文在整个测试集上的 BLEU 得分为 34.8，其中 LSTM 的 BLEU 得分为词汇外单词的惩罚性得分。 此外，LSTM 在处理长句时也没有遇到困难。 相比之下，基于短语的 SMT 系统在同一数据集上的 BLEU 得分为 33.3。 当我们使用 LSTM 对上述 SMT 系统产生的 1000 个假设进行重新排序时，其 BLEU 得分上升到 36.5，接近之前在该任务中的最佳结果。 LSTM 还学习了合理的短语和句子表征，这些表征对词序很敏感，而且对主动语态和被动语态相对不变。 最后，我们发现，颠倒所有源句（而非目标句）中的词序能显著提高 LSTM 的性能，因为这样做在源句和目标句之间引入了许多短期依赖关系，从而使优化问题变得更容易。

1 引言

深度神经网络（DNN）是一种极其强大的机器学习模型，在语音识别[13, 7]和视觉物体识别[19, 6, 21, 20]等难题上表现出色。 DNN 之所以强大，是因为它们只需少量步骤就能进行任意并行计算。 DNN 功能强大的一个令人惊讶的例子是，它们只需使用 2 个二次方大小的隐藏层，就能对 N N 位数字进行排序[27]。 因此，虽然神经网络与传统统计模型有关，但它们学习的是复杂的计算。 此外，只要标注的训练集有足够的信息来指定网络参数，就可以使用监督反向传播法训练大型 DNN。 因此，如果大型 DNN 的参数设置能取得良好效果（例如，因为人类能快速解决任务），那么有监督的反向传播就能找到这些参数并解决问题。

尽管 DNN 具有灵活性和强大功能，但它只能应用于输入和目标可以用固定维度向量合理编码的问题。 这是一个很大的限制，因为许多重要问题最好用序列来表达，而序列的长度并不是事先就知道的。 例如，语音识别和机器翻译就是序列问题。 同样，问题解答也可以看作是将代表问题的词序列映射到代表答案的词序列。 因此，一种与领域无关的、能够学习将序列映射到序列的方法显然是有用的。

序列问题给 DNN 带来了挑战，因为它们要求输入和输出的维度是已知和固定的。 在本文中，我们展示了长短时记忆（LSTM）架构 [16] 的直接应用可以解决一般的序列到序列问题。 我们的想法是使用一个 LSTM 一次一个时间步读取输入序列，以获得固定维度的大矢量表示，然后使用另一个 LSTM 从该矢量中提取输出序列（图 1）。 第二个 LSTM 本质上是一个递归神经网络语言模型[28, 23, 30]，只是它以输入序列为条件。 由于输入和相应

输出之间存在相当大的时滞，LSTM 能够成功地学习具有长范围时间依赖性的数据，因此成为本应用的自然选择（图 1）。

在利用神经网络解决一般序列到序列的学习问题方面，已经有许多相关的尝试。 我们的方法与 Kalchbrenner 和 Blunsom[18]密切相关，他们是第一个将整个输入句子映射到向量上的人，我们的方法与 Cho 等人[5]也有关联，不过后者仅用于对基于短语的系统产生的假设进行重新评分。 Graves [10] 提出了一种新颖的可区分注意力机制，允许神经网络关注输入的不同部分，Bahdanau 等人[2] 成功地将这一想法的优雅变体应用于机器翻译。 联结序列分类法是另一种利用神经网络将序列映射到序列的流行技术，但它假定输入和输出之间存在单调对齐[11]。

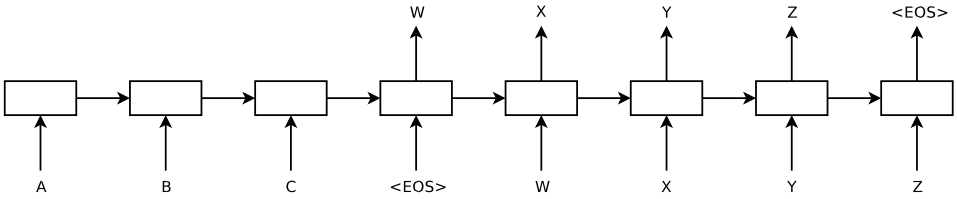

图 1：我们的模型读取输入句子 "ABC"，并输出句子 "WXYZ"。 模型在输出句末标记后停止预测。 请注意，LSTM 是反向读取输入句子的，因为这样做会在数据中引入许多短期依赖关系，从而使优化问题变得更加容易。

这项工作的主要成果如下。 在 WMT'14 英语到法语的翻译任务中，我们使用简单的从左到右波束搜索解码器，直接从 5 个深度 LSTM（每个 LSTM 有 384M 个参数和 8000 个维度状态）的集合中提取翻译，获得了 34.81 的 BLEU 分数。 这是迄今为止使用大型神经网络进行直接翻译所取得的最佳结果。 作为比较，SMT 基线在该数据集上的 BLEU 得分为 33.30 [29]。 34.81 的 BLEU 得分是由一个拥有 80k 词汇量的 LSTM 实现的，因此只要参考译文中包含了这 80k 词汇量之外的单词，得分就会受到惩罚。 这一结果表明，一个相对未优化的小词汇量神经网络架构的性能优于基于短语的 SMT 系统，而这一架构还有很大的改进空间。

最后，我们使用 LSTM 对 SMT 基线在同一任务中公开提供的 1000 个最佳列表进行了重新评分[29]。 通过这样做，我们得到了 36.5 的 BLEU 分数，比基线提高了 3.2 BLEU 分，接近之前在该任务上公布的最佳结果（37.0 [9]）。

令人惊讶的是，LSTM 在处理超长句子时并没有受到影响，尽管最近有其他研究人员使用了相关的架构[26]。 我们之所以能在处理长句时表现出色，是因为我们在训练集和测试集中颠倒了源句中单词的顺序，而没有颠倒目标句中单词的顺序。 通过这种方法，我们引入了许多短期依赖关系，使优化问题变得更加简单（见第 2 和 3.3 节）。 因此，SGD 可以学习长句中没有问题的 LSTM。 将源句中的单词颠倒过来的简单技巧是这项工作的关键技术贡献之一。

LSTM 的一个有用特性是，它学会将长度可变的输入句子映射到固定维度的向量表示中。 鉴于译文往往是源句的转述，翻译目标促使 LSTM 找到能够捕捉其含义的句子表示，因为含义相似的句子彼此接近，而含义不同的句子则相去甚远。 定性评估支持了这一说法，表明我们的模型能够感知词序，并且对主动语态和被动语态具有相当的不变性。

2 模型

循环神经网络（RNN）[31, 28] 是前馈神经网络对序列的自然概括。 给定输入序列 (x1,...,xT)，标准 RNN 通过迭代以下等式计算输出序列 (y1,...,yT)：

$$h_t = \text{sigm}\left(W^{\text{hx}} x_t + W^{\text{hh}} h_{t-1}\right)$$

$$y_t = W_{\text{yh}} h_t$$

只要提前知道输入和输出之间的排列，RNN 就能轻松地将序列映射到序列。 然而，如何将 RNN 应用于输入和输出序列长度不同、关系复杂且非单调的问题，目前尚不清楚。

一般序列学习的最简单策略是使用一个 RNN 将输入序列映射到一个固定大小的向量，然后使用另一个 RNN 将该向量映射到目标序列（Cho 等人[5]也采用了这种方法）。 虽然这种方法原则上可行，因为 RNN 可以获得所有相关信息，但由于会产生长期依赖关系，因此很难对 RNN 进行训练（图 1）[14, 4, 16, 15]。 不过，众所周知，长短期记忆（LSTM）[16]可以学习具有长时程依赖性的问题，因此在这种情况下，LSTM 可能会成功。

LSTM 的目标是估计条件概率 p(y1,...,yT′ |x1,...,xT)，其中 (x1,...,xT) 是输入序列，y1,...,yT′ 是其相应的输出序列，其长度 T′ 可能与 T 不同。 LSTM 计算这一条件概率的方法是，首先获取输入序列（x1,...,xT）的定维表示 v，该表示由 LSTM 的最后一个隐藏状态给出，然后用标准 LSTM-LM 公式计算 y1、.....、yT′ 的概率，其初始隐藏状态设为 x1、.....、xT 的表示 v：

$$p(y_1, \ldots, y_{T'} | x_1, \ldots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \ldots, y_{t-1}) \quad (1)$$

在这个公式中，每个 p(yt|v,y1,...yt-1) 分布都用词汇表中所有单词的软最大值来表示。 我们使用 Graves [10] 的 LSTM 方法。 需要注意的是，我们要求每个句子以一个特殊的句末符号"<EOS>"结束，这使得模型可以定义所有可能长度序列的分布。 整体方案如图 1 所示，图中的 LSTM 会计算 "A"、"B"、"C"、"<EOS>"的表示，然后使用该表示计算 "W"、"X"、"Y"、"Z"、"<EOS>"的概率。

我们的实际模型在三个重要方面与上述描述有所不同。 首先，我们使用了两个不同的 LSTM：一个用于输入序列，另一个用于输出序列，因为这样做可以增加模型参数的数量，而计算成本却可以忽略不计，而且可以很自然地同时在多个语言对上训练 LSTM [18]。 其次，我们发现深层 LSTM 的性能明显优于浅层 LSTM，因此我们选择了四层的 LSTM。 第三，我们发现颠倒输入句子中单词的顺序非常有价值。 因此，举例来说，我们要求 LSTM 将 c、b、a 映射到 α、β、γ，而不是将句子 a、b、c 映射到句子 α、β、γ，其中 α、β、γ 是 a、b、c 的翻译。 这样一来，a 与 α 相近，b 与 β 相近，以此类推，SGD 就很容易在输入和输出之间 "建立通信"。 我们发现这种简单的数据转换大大提高了 LSTM 的性能。

3 实验

我们以两种方式将我们的方法应用于 WMT'14 英法 MT 任务。 我们用它来直接翻译输入句子，而不使用参考 SMT 系统；我们用它来重新计算 SMT 基线的 n 个最佳列表。 我们报告了这些翻译方法的准确性，展示了翻译示例，并直观展示了翻译后的句子表示。

3.1 数据集详情

我们使用的是 WMT'14 英法数据集。 我们在由 3.48 亿个法语单词和 3.04 亿个英语单词组成的 1200 万个句子子集上训练模型，该子集是从 [29] 中 "精选 "出来的。 我们之所以选择这项翻译任务和这个特定的训练集子集，是因为我们公开了标记化的训练集和测试集，以及来自基线 SMT [29] 的 1000 个最佳列表。

由于典型的神经语言模型依赖于每个单词的向量表示，因此我们对两种语言都使用了固定的词汇量。 我们在源语言和目标语言中分别使用了 16 万个和 8 万个最常见的词汇。每个词汇表之外的单词都被一个特殊的 "UNK "标记所替代.

3.2 解码和重计分

我们实验的核心是在许多句子对上训练大型深度 LSTM。 我们通过最大化源句 S 的正确翻译 T 的对数概率来训练它，因此训练目标是

$$1/|\mathcal{S}| \sum_{(T,S)\in\mathcal{S}} \log p(T|S)$$

其中 S 是训练集。 训练完成后，我们根据 LSTM 找到最可能的翻译，从而生成译文：

$$\hat{T} = \arg\max_{T} p(T|S) \qquad (2)$$

我们使用一个简单的从左到右的波束搜索解码器来搜索最可能的翻译，该解码器保留了少量的部分假设 B，其中部分假设是某些翻译的前缀。 在每个时间步，我们都会用词汇表中的每个可能的单词扩展波束中的每个部分假设。 这大大增加了假设的数量，因此我们会根据模型的对数概率丢弃除 B 个最有可能的假设之外的所有假设。 一旦"<EOS>"符号被添加到一个假设中，它就会从波束中删除，并被添加到完整假设的集合中。 虽然这种解码器是近似的，但实现起来却很简单。 有趣的是，即使波束大小为 1，我们的系统也能表现出色，而波束大小为 2 的系统则能提供波束搜索的大部分优势（表 1）。

我们还使用 LSTM 对基线系统 [29] 生成的 1000 个最佳列表进行了重新评分。 为了给 n 个最佳列表重新评分，我们用 LSTM 计算了每个假设的对数概率，然后将它们的得分与 LSTM 的得分求平均值。

3.3 反转源句

虽然 LSTM 能够解决具有长期依赖性的问题，但我们发现，当源句颠倒时（目标句没有颠倒），LSTM 的学习效果要好得多。 这样，LSTM 的测试困惑度从 5.8 降至 4.7，其解码翻译的测试 BLEU 分数从 25.9 升至 30.6。

虽然我们并不能完全解释这种现象，但我们认为这是由于数据集中引入了许多短期依赖关系造成的。 通常情况下，当我们将源句与目标句连接起来时，源句中的每个单词都远离目标句中的相应单词。 因此，该问题存在较大的 "最小时滞"[17]。 将源句中的单词颠倒后，源语言和目标语言中对应单词之间的平均距离保持不变。 但是，源语言中的前几个词现在与目标语言中的前几个词非常接近，因此问题的最小时滞大大减少。 因此，反向传播更容易在源语言句子和目标语言句子之间 "建立沟通"，从而大幅提高整体性能。

起初，我们认为颠倒输入句子只会导致对目标句子早期部分的预测更有把握，而对后期部分的预测则信心不足。 然而，与原始源句相比，用反转源句训练的 LSTM 在长句上的表现要好得多（见第 3.7 节），这表明反转输入句子会使 LSTM 的记忆利用率更高。

3.4 训练详情

我们发现 LSTM 模型相当容易训练。 我们使用的深度 LSTM 有 4 层，每层有 1000 个单元和 1000 维单词嵌入，输入词汇量为 160,000 个，输出词汇量为 80,000 个。 因此，深度 LSTM 使用 8000 个实数来表示一个句子。 我们发现深度 LSTM 的表现明显优于浅层 LSTM，每增加一层就能降低近 10%的复杂度，这可能是由于其隐藏状态更大的缘故。 我们在每个输出端使用了超过 80,000 个单词的 naive softmax。 由此产生的 LSTM 有 384M 个参数，其中 64M 是纯递归连接（"编码器 "LSTM 有 32M 个参数，"解码器 "LSTM 有 32M 个参数）。完整的训练细节如下：

- - 我们将 LSTM 的所有参数初始化为 -0.08 和 0.08 之间的均匀分布。

- - 我们使用的是无动量随机梯度下降法，学习率固定为 0.7。 5 个 epoch 后，我们开始每半个 epoch 将学习率减半。 我们对模型总共进行了 7.5 个历元的训练.
- - 我们使用 128 个序列的批次进行梯度计算，并按批次大小（即 128）进行划分.
- - 虽然 LSTM 通常不会出现梯度消失的问题，但它们也可能出现梯度爆炸的问题。 因此，我们对梯度的规范[10, 25]实施了硬性约束，当梯度的规范超过阈值时，就对其进行缩放。 对于每个训练批次，我们计算 s = kgk2，其中 g 是梯度除以 128。 如果 s > 5、 we set $g = \frac{5g}{s}$.
- - 不同的句子有不同的长度。 大多数句子较短（如长度为 20-30 的句子），但也有一些句子较长（如长度大于 100 的句子），因此随机选择 128 个训练句子组成的迷你批次中，短句较多，长句较少，结果浪费了迷你批次中的大量计算。 为了解决这个问题，我们确保迷你批中的所有句子长度大致相同，从而将计算速度提高了 2 倍。.

3.5 并行化

深度 LSTM 的 C++ 实现采用了上一节的配置，在单 GPU 上的处理速度约为每秒 1,700 字。 这对于我们的目的来说太慢了，因此我们使用 8 个 GPU 对模型进行了并行化处理。 LSTM 的每一层都在不同的 GPU 上执行，并在计算完成后立即将其激活信息传递给下一个 GPU/层。 我们的模型有 4 层 LSTM，每一层都位于单独的 GPU 上。 其余 4 个 GPU 用于并行处理 softmax，因此每个 GPU 负责乘以 1000×20000 矩阵。 最终的实现速度达到每秒 6,300 个（英语和法语）单词，最小批量为 128 个。 这种实现方式的训练耗时约十天。

3.6 实验结果

我们使用 BLEU 分值[24]来评估翻译质量。 我们使用 multi-bleu.pl 对标记化预测和基本事实计算 BLEU 分数。 这种评估 BELU 分数的方法与 [5] 和 [2] 的方法一致，并重现了 [29] 的 33.3 分。 然而，如果我们用这种方法评估最佳的 WMT'14 系统 [9]（其预测结果可从 statmt.org\matrix 下载），我们会得到 37.0 分，高于 statmt.org\matrix 报告的 35.8 分。

结果见表 1 和表 2。 我们的最佳结果是通过 LSTM 的集合获得的，这些 LSTM 在随机初始化和小批量随机顺序方面存在差异。 虽然 LSTM 集合的解码翻译结果没有超过最佳 WMT'14 系统，但这是纯神经翻译系统首次在大规模 MT 任务中以相当大的优势超过基于短语的 SMT 基线，尽管它无法处理词汇表以外的单词。 如果使用 LSTM 对基线系统的 1000 个最佳列表进行重新评分，它与 WMT'14 的最佳结果的 BLEU 分数相差在 0.5 分以内。

Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size 2	34.50
Ensemble of 5 reversed LSTMs, beam size 12	**34.81**

表 1：LSTM 在 WMT'14 英法测试集 (ntst14) 上的表现。 请注意，5 个波束大小为 2 的 LSTM 的集合比一个波束大小为 12 的 LSTM 的集合要合适。

Method	test BLEU score (ntst14)
Baseline System [29]	33.30
Cho et al. [5]	34.54
Best WMT’14 result [9]	**37.0**

Rescoring the baseline 1000-best with a single forward LSTM	35.61
Rescoring the baseline 1000-best with a single reversed LSTM	35.85
Rescoring the baseline 1000-best with an ensemble of 5 reversed LSTMs	**36.5**
Oracle Rescoring of the Baseline 1000-best lists	~45

表 2：在 WMT'14 英法测试集 (ntst14) 上使用神经网络和 SMT 系统的方法。

3.7 在长句方面的表现

我们惊奇地发现，LSTM 在处理长句时表现出色，这在图 3 中有定量显示。 表 3 列出了几个长句及其翻译的例子。

3.8 模型分析

图 2：该图显示的是处理图中短语后得到的 LSTM 隐藏状态的二维 PCA 投影。 短语是按意义聚类的，在这些例子中，意义主要是词序的函数，而词袋模型很难捕捉到这一点。请注意，这两个聚类具有相似的内部结构。

Type	Sentence
Our model	汽车制造商奥迪公司董事会成员乌尔里希-UNK 说，多年来，董事会开会前收缴手机以防被用作远程监听设备已成为惯例。
Truth	汽车制造商奥迪董事会成员乌尔里希-哈肯贝格（Ulrich Hackenberg）说，多年来，董事会开会前收缴手机以防被用作远程监听设备已成为惯例。
Our model	"UNK 说："蜂窝电话确实是一个问题，不仅因为它们有可能对导航设备造成干扰，而且我们知道，根据美国联邦通信委员会（FCC）的规定，当它们在空中时，可能会干扰蜂窝电话发射塔。
Truth	"罗森克说："移动电话肯定是个问题，不仅因为它们有可能干扰导航仪器，还因为我们从联邦通信委员会了解到，如果在机上使用移动电话，它们可能会干扰移动电话基站。
Our model	火葬会给人一种 "对亲人遗体施暴的感觉"，遗体会在很短的时间内 "化为一堆灰烬"，而不是 "伴随哀悼阶段 "的分解过程。
Truth	创造是 "对所爱的躯体的一种暴力"，它将在很短的时间内 "化为一堆灰烬"，而不是经过一个 "伴随着哀悼阶段 "的分解过程。

表 3：LSTM 与地面实况翻译一起生成的长篇翻译示例。 读者可以使用谷歌翻译来验证翻译是否合理。

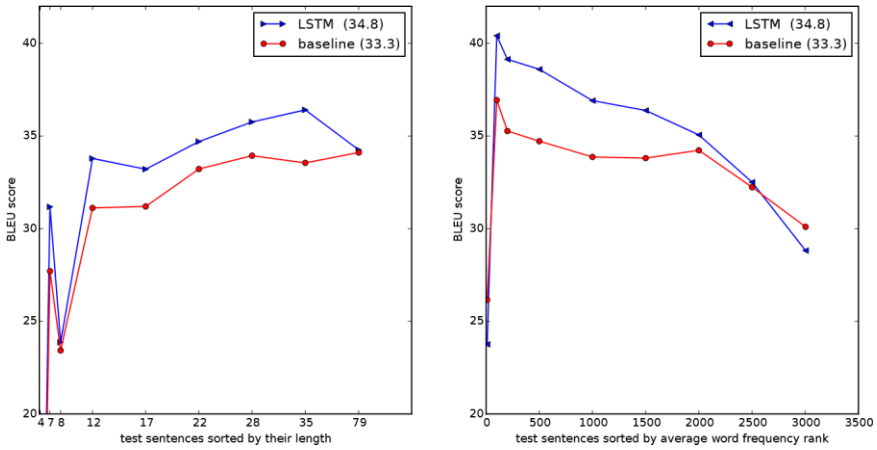

图 3：左图显示了我们系统的性能与句子长度的函数关系，其中 x 轴对应于按长度排序的测试句子，并以实际序列长度标出。 少于 35 个单词的句子性能没有下降，而最长的句子性能略有下降。 右图显示了 LSTM 在单词稀少程度逐渐增加的句子中的表现，X 轴对应的是按 "平均单词频率等级 "排序的测试句子。

我们的模型有一个吸引人的特点，就是能够将单词序列转化为固定维度的向量。 图 2 展示了一些学习到的表征。 该图清楚地表明，表征对词语的顺序很敏感，而对主动语态与被动语态的替换则相当不敏感。 二维投影是通过 PCA 得到的。

4 相关工作

在将神经网络应用于机器翻译方面，已有大量研究成果。 迄今为止，将 RNN 语言模型 (RNNLM) [23] 或前馈神经网络语言模型 (NNLM) [3] 应用于 MT 任务的最简单、最有效的方法是对强 MT 基线 [22] 的 nbest 列表进行重新评分，从而可靠地提高翻译质量。

最近，研究人员开始研究将源语言信息纳入 NNLM 的方法。 这方面的例子包括 Auli 等人[1]，他们将 NNLM 与输入句子的主题模型相结合，从而提高了重分性能。 Devlin 等人[8]采用了类似的方法，但他们将 NNLM 纳入了 MT 系统的解码器中，并使用解码器的对齐信息为 NNLM 提供输入句子中最有用的单词。 他们的方法非常成功，与基线相比取得了很大的改进。

我们的研究与 Kalchbrenner 和 Blunsom [18] 的研究密切相关，他们是第一个将输入句子映射到向量然后再映射回句子的人，不过他们使用卷积神经网络将句子映射到向量，这就失去了单词的排序。 与这项工作类似，Cho 等人[5] 使用类似 LSTM 的 RNN 架构将句子映射到向量，然后再映射回句子，不过他们的主要重点是将神经网络集成到 SMT 系统中。 Bahdanau 等人[2]也尝试使用神经网络进行直接翻译，该网络使用注意力机制来克服 Cho 等人[5]在长句方面的不良表现，并取得了令人鼓舞的成果。 同样，Pouget-Abadie 等人[26] 尝试通过翻译源句片段的方式来解决 Cho 等人[5] 的记忆问题，从而产生流畅的翻译，这与基于短语的方法类似。 我们认为，他们只需在颠倒的源句上训练其网络，就能实现类似的改进。

端到端训练也是 Hermann 等人[12]的研究重点，他们的模型通过前馈网络表示输入和输出，并将它们映射到空间中的相似点。 不过，他们的方法不能直接生成翻译：要获得翻译，他们需要在预先计算的句子数据库中查找最接近的向量，或对句子重新评分。

5 结论

在这项工作中，我们发现，在大规模 MT 任务中，词汇量有限且几乎不对问题结构做任何假设的大型深度 LSTM，其性能优于词汇量无限的基于 SMT 的标准系统。 我们基于 LSTM 的简单方法在 MT 上取得的成功表明，只要有足够的训练数据，它在许多其他序列学习问题上也能取得很好的效果。

我们对将源句中的单词颠倒后所获得的改进程度感到惊讶。 我们得出的结论是，找到具有最多短期依赖关系的问题编码非常重要，因为它们能让学习问题变得更加简单。 特别是，虽然我们无法在非颠倒翻译问题上训练标准 RNN（如图 1 所示），但我们相信，当源句颠倒时，标准 RNN 应该很容易训练（尽管我们没有进行实验验证）。

我们还对 LSTM 正确翻译超长句子的能力感到惊讶。 我们起初认为，由于 LSTM 的内存有限，它在翻译长句时会失败，其他研究人员也报告说，使用与我们类似的模型翻译长句时表现不佳 [5, 2, 26]。 然而，在反向数据集上训练的 LSTM 在翻译长句时几乎没有遇到困难。

最重要的是，我们证明了一种简单、直接和相对未优化的方法可以超越 SMT 系统，因此进一步的工作可能会带来更高的翻译精确度。 这些结果表明，我们的方法很可能在其他具有挑战性的序列到序列问题上表现出色。

6 Acknowledgments

We thank Samy Bengio, Jeff Dean, Matthieu Devin, Geoffrey Hinton, Nal Kalchbrenner, Thang Luong, Wolfgang Macherey, Rajat Monga, Vincent Vanhoucke, Peng Xu, Wojciech Zaremba, and the Google Brain team for useful comments and discussions.

References

[1] M. Auli, M. Galley, C. Quirk, and G. Zweig. Joint language and translation modeling with recurrent neural networks. In *EMNLP*, 2013.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. In *Journal of Machine Learning Research*, pages 1137–1155, 2003.

[4] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[5] K. Cho, B. Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Arxiv preprint arXiv:1406.1078*, 2014.

[6] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, 2012.

[7] G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing - Special Issue on Deep Learning for Speech and Language Processing*, 2012.

[8] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul. Fast and robust neural network joint models for statistical machine translation. In *ACL*, 2014.

[9] Nadir Durrani, Barry Haddow, Philipp Koehn, and Kenneth Heafield. Edinburgh's phrase-based machine translation systems for wmt-14. In *WMT*, 2014.

[10] A. Graves. Generating sequences with recurrent neural networks. In *Arxiv preprint arXiv:1308.0850*, 2013.

[11] A. Graves, S. Ferna´ndez, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006.

[12] K. M. Hermann and P. Blunsom. Multilingual distributed representations without word alignment. In *ICLR*, 2014.

[13] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 2012.

[14] S. Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. *Master's thesis, Institut fur Informatik, Technische Universitat, Munchen*, 1991.

[15] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[17] S. Hochreiter and J. Schmidhuber. LSTM can solve hard long time lag problems. 1997.

[18] N. Kalchbrenner and P. Blunsom. Recurrent continuous translation models. In *EMNLP*, 2013.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

[20] Q.V. Le, M.A. Ranzato, R. Monga, M. Devin, K. Chen, G.S. Corrado, J. Dean, and A.Y. Ng. Building high-level features using large scale unsupervised learning. In *ICML*, 2012.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

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

[23] T. Mikolov, M. Karafia´t, L. Burget, J. Cernocky`, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, pages 1045–1048, 2010.

[24] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. BLEU: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[25] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. *arXiv preprint arXiv:1211.5063*, 2012.

[26] J. Pouget-Abadie, D. Bahdanau, B. van Merrienboer, K. Cho, and Y. Bengio. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. *arXiv preprint arXiv:1409.1257*, 2014.

[27] A. Razborov. On small depth threshold circuits. In *Proc. 3rd Scandinavian Workshop on Algorithm Theory*, 1992.

[28] D. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[29] H. Schwenk. University le mans. `http://www-lium.univ-lemans.fr/šchwenk/cslm_joint_paper/`, 2014. [Online; accessed 03-September-2014].

[30] M. Sundermeyer, R. Schluter, and H. Ney. LSTM neural networks for language modeling. In *INTERSPEECH*, 2010.

[31] P. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of IEEE*, 1990.