Report of Deep Learning for Natural Langauge Processing

ZY2303217 郑茗畅
zy2303217@buaa.edu.cn

Abstract

本研究以金庸的十六部经典武侠小说作为语料库数据支撑，基于Seq2seq模型与Transformer模型在给定开头的情况下自动生成具有连贯性和风格一致性的武侠小说片段或章节，探讨两种模型在特定领域文本生成任务中的应用与性能比较。实验过程包括数据准备、Seq2seq模型和Transformer模型的实现、模型训练与评估、两模型生成结果的连贯性、语义相关性和创造性等方面的对比分析。

Seq2Seq模型通过编码器和解码器结构，将输入序列映射到输出序列。编码器负责将输入信息编码成固定长度的上下文向量，解码器则根据上下文向量生成目标序列。这种方法在机器翻译任务中取得了良好的效果。

Transformer模型引入了自注意力机制，通过并行计算提高了模型的效率和性能。自注意力机制使模型能够在处理每个词时同时关注输入序列的其他部分，从而更好地捕捉长距离依赖关系。Transformer模型在多个NLP任务中表现出色，包括机器翻译、文本摘要和文本生成等。

实验结果表明：Transformer在生成文本的连贯性、语义相关性和创造性等方面均具有一定优势，然而其耗费的计算资源更多。在资源有限的条件下，Seq2Seq模型反而能够取得更好的效果。

Introduction

I1: Seq2seq模型

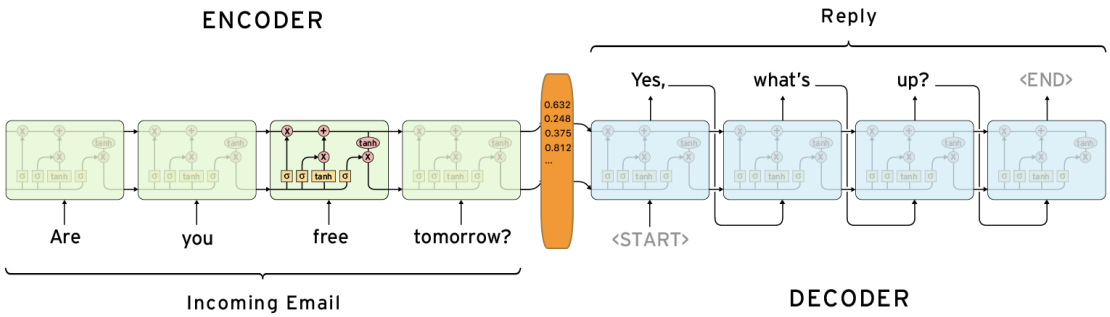
Seq2Seq是“序列到序列”的缩写，是一种将输入序列转换为输出序列的深度学习模型架构，主要用于自然语言处理任务，如机器翻译、文本摘要和会话人工智能等。该架构于2014年由谷歌提出，并已成为许多最先进的NLP应用程序的基础组件。

Seq2seq的核心思想是使用两个RNN（递归神经网络）模块，一个作为编码器（Encoder），另一个作为解码器（Decoder），实现将输入序列（例如，一个句子或一段文字）转换为另一个输出序列（例如，用另一种语言翻译的输入或摘要版本），在整个转换过程中保持自身的语义。

Seq2Seq模型主要由编码器和解码器组成。其中，编码器接受输入序列并将其编码为固定大小的向量或向量序列，其目的是捕获输入的上下文信息和含义。通常使用RNN、LSTM、GRU等网络。而解码器接受编码表示（上下文向量或来自编码器的最后一个隐藏状态）并生成输出序列，每次一个输出元素。它以递归的方式使用以前生成的输出作为输入，并进行自回归。这里可以使用类似于编码器中的RNN、LSTM、GRU 、Attention等，使解码器能够访问整个编码输入序列，以进行更多的上下文感知预测。

与传统的RNN网络固定输入输出序列不同，Seq2Seq网络的特点是能够处理变长输入和进行变长输出。在Decoder中，首先输入一个<START>标志，此时的网络输出为第一个单词，然后不断地将当前Decoder的输出作为下一时刻的输入，直至网络输出<END>标志或达到最大序列长度为止。

一个使用LSTM作为编码器和解码器的Seq2Seq网络示意图如所示，首先在编码部分的LSTM输入词向量，并向后传播，得到语义向量，而语义向量在解码器中作为隐藏状态的初始值，并不断向后传播，生成输出序列。



Seq2Seq网络具有以下关键优势：

①能够处理可变长度的输入和输出序列；

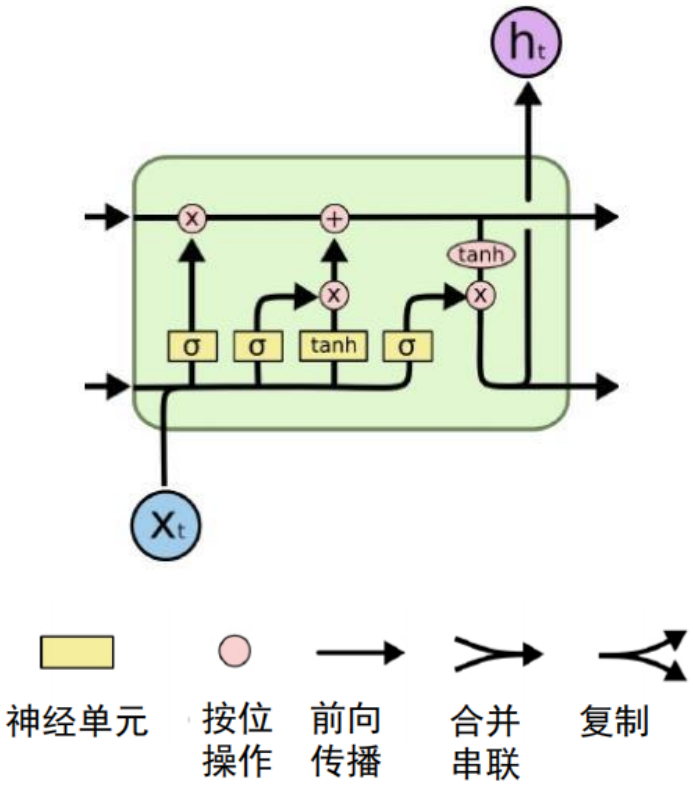
②编码器-解码器架构可以捕捉输入和输出之间的复杂关系；

③模型可以端到端地训练,无需手动设计特征。

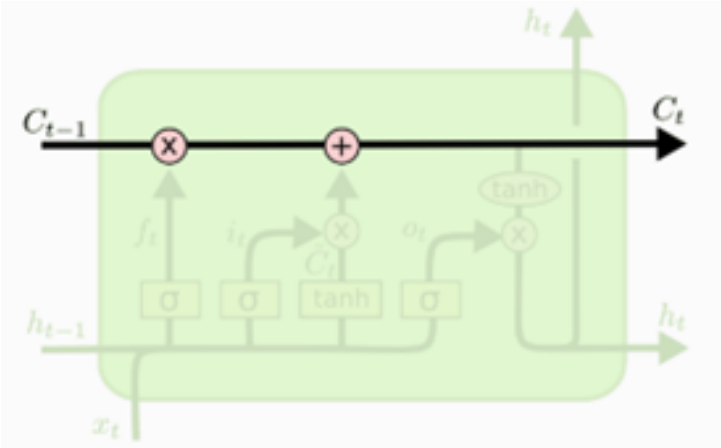
Seq2Seq网络用途十分广泛，包括机器翻译、文本概括、语音识别、阅读理解、图片描述、文本生成等。

I2: LSTM模型

LSTM（Long Short Term Memory）是一种特殊的递归神经网络（RNN），它能够有效地捕捉和保留长期依赖信息,克服了传统RNN在处理长序列数据时容易遗忘早期信息的缺点。LSTM通过引入门控机制,能够在序列数据处理中更好地控制信息的流动和记忆的保存。其单元组成如所示，其中各图标含义如所示。

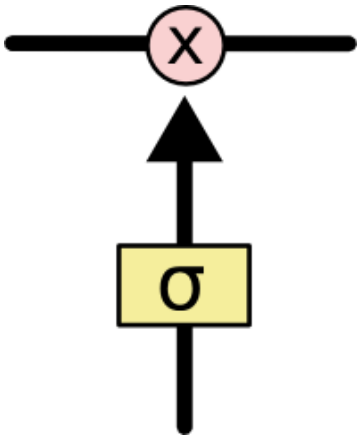


LSTM单元状态如所示。



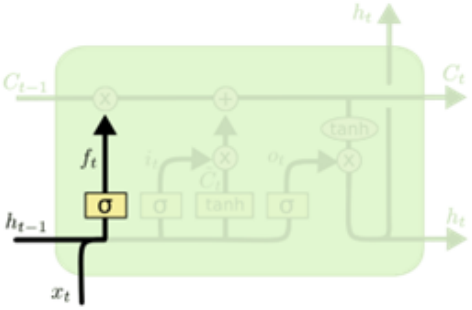
穿过图的顶部的长横线，长直线称之为节点状态（Cell State），决定什么样的信息会被保留，什么样的信息会被遗忘，记为C_t。

LSTM网络能通过一种被称为门的结构对节点状态进行删除或者添加信息。门能够有选择性的决定让哪些信息通过。门的结构为一个sigmoid层和一个点乘操作的组合，sigmoid层输出0到1之间的数，描述每个部分有多少量可以通过，0代表不允许任何量通过，1表示允许任何量通过，结构如所示：



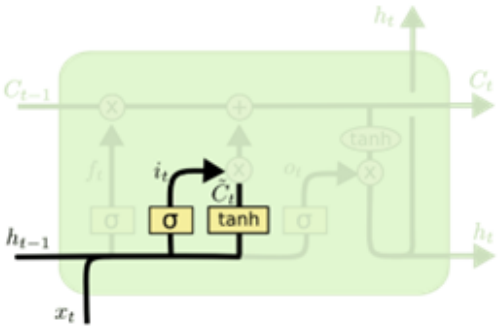
LSTM实现了三个门计算，即遗忘门、输入门和输出门，用来保护和控制节点状态。

遗忘门负责决定保留多少上一时刻的单元状态到当前时刻的单元状态，即决定从节点状态中丢弃什么信息。该门读取h_{t-1}和x_t，然后经过sigmoid层后，输出一个0-1之间的数f_t给每个在细胞状态C_{t-1}中的数字逐点相乘。f_t的值为0表示完全丢弃，1表示完全保留。遗忘门结构如所示：



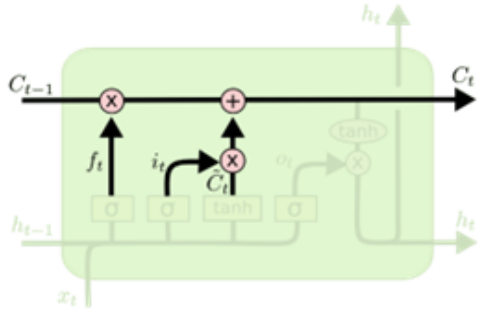
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

输入门负责决定保留多少当前时刻的输入到当前时刻的单元状态，包含两个部分，第一部分为sigmoid层，该层决定要更新什么值，第二部分为tanh层，该层把需要更新的信息更新到细胞状态里。tanh层创建一个新的细胞状态值向量$\widetilde{C_t}$，$\widetilde{C_t}$会被加入到状态中。输入门结构如所示：



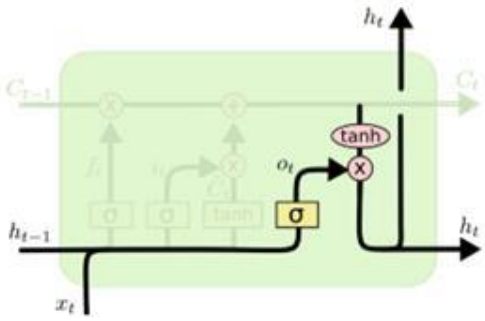
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

然后就到了更新旧细胞状态的时间了，将C_{t-1}更新为C_t，把旧状态与f_t相乘，丢弃确定需要丢弃的信息，再加上 i_t*$\widetilde{C_t}$，这样就完成了细胞状态的更新，结构如所示：



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

输出门负责决定当前时刻的单元状态有多少输出，通过一个sigmoid层来确定细胞状态的哪个部分将输出出去。把细胞状态通过tanh进行处理，得到一个-1到1之间的值，并将它和sigmoid门的输出相乘，最终仅仅输出确定输出的部分。输出门结构如所示：



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

LSTM的参数训练算法为反向传播算法，步骤如下：

①前向计算每个神经元的输出值。对于LSTM而言，依据前面介绍的算法，分别进行计算；

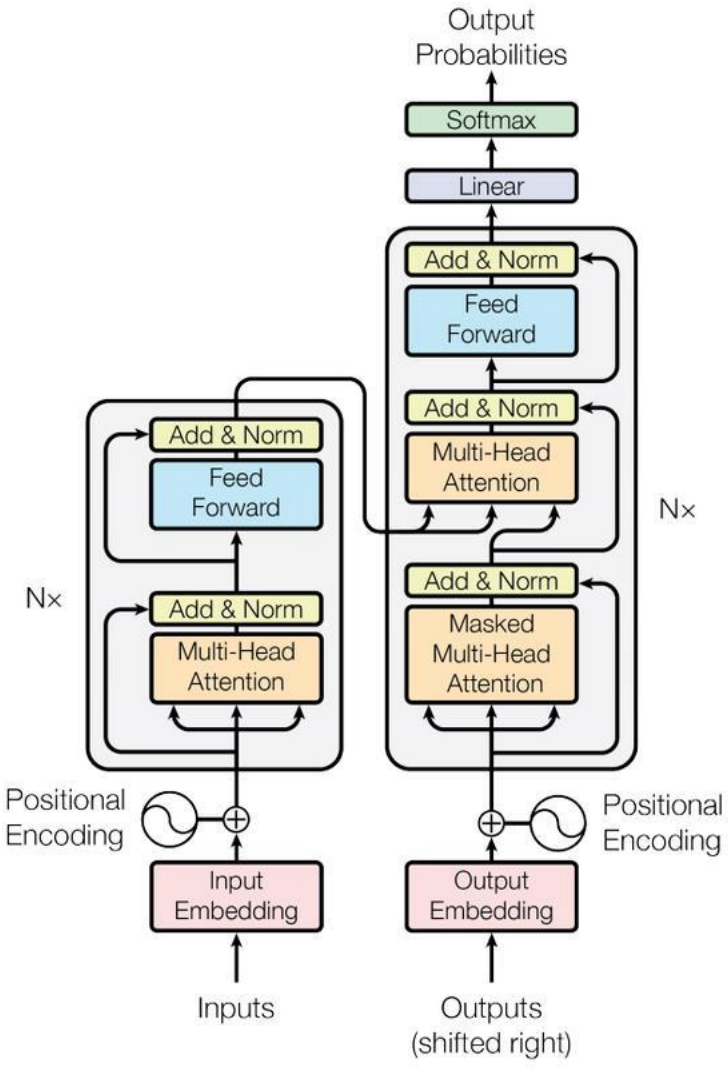
②确定优化目标函数。在训练早期，输出值和预期值会不一致，于是计算每个神经元的误差项值，构造出损失函数；

③根据损失函数的梯度指引，更新网络权值参数。与传统RNN类似，LSTM误差项的反向传播包括两个层面：一个是空间上层面的，将误差项向网络的上一层传播。另一个是时间层面上的，沿时间反向传播，即从当前t时刻开始，计算每个时刻的误差

④跳转①，重复做①、②和③，直至网络误差小于给定值

I3: Transformer模型

Transformer模型是由Vaswani等人在2017年提出的一种基于注意力机制的序列到序列模型，广泛应用于自然语言处理任务，如机器翻译、文本生成等。与传统的RNN和LSTM不同，Transformer模型完全摒弃了递归结构，采用了自注意力机制（Self-Attention Mechanism）来捕捉序列中的长距离依赖关系。其基本架构如所示。



其Encoder由多头注意力模块和具有残差连接的前馈神经网络构成，Decoder与Encoder结构类似，额外添加了Masked多头注意力模块。随着模型处理输入序

列的每个单词，自注意力会关注整个输入序列的所有单词，帮助模型对本单词更好地进行编码。在处理过程中，自注意力机制会将对所有相关单词的理解融入到正在处理的单词中。并且，自注意力还能够并行计算，能够更充分地利用现代GPU的作用，对计算进行加速。

但是与Seq2Seq不同，由于注意力机制在计算是没有考虑位置差异，即对于两个单词，无论其位置如何改变，注意力分数都是固定的，但在实际文本建模当中，单词的出现顺序也包含着语义信息，为了解决该问题，Transformer引入了位置编码，即为了在网络中理解单词之间的位置差异，在原有的embedding上加或拼接一个位置编码，一般在实际使用当中可以选择使用可学习的或固定的位置编码。

在Transformer网络中，编码器主要是将输入序列转换成连续表示，提取深层次特征，而解码器通过结上当前提取到的特征和之前的输入，逐步生成输出序列。并且值得注意的是，解码器在靠近输入端的部分使用了Masked多头注意力，是为了屏蔽掉位置靠后的词对位置靠前的词的影响，即一个词的生成只能依靠其前面的词，不使用未来的信息，使得模型学习到正确的语言结构和单词之间的依赖关系。

Methodology

M1: Seq2seq

算法流程如下：

1. 语料库处理
   1） 使用 16 篇金庸小说作为语料数据库；
   2） 删除部分文件中的无用信息；
   3） 将英文标点符号转换为中文标点符号；
   4） 将所有繁体表达转换为简体；
   5） 将文章以句子为单位进行拆分，形成句子组成的列表结构；
   6） 对句子使用 jieba 分词库进行分词，去掉中文停词和不包含汉字的词，将每一个分词后所构成的 list 组成新的 list，格式为 list[list[str]]，外层 list 为每一个句子，内层 list 为一个句子中的各个中文单词；
   7） 将全部文章分词后的句子加入到总的 list 中；
   8） 建立词典并将语料转换为词的索引序列。
2. 模型训练
   1） 定义模型结构；
   2） 初始化模型权重；
   3） 定义损失函数和优化器；
   4） 开始训练循环。

   其中训练Seq2seq模型时具体参数如下：

   - bos_id = 0，句子的起始标记（Begin of Sentence）的索引；
   - eos_id = 1，句子的结束标记（End of Sentence）的索引；
   - pad_id= 2，填充标记的索引，用于将句子序列填充到相同长度；
   - max_tokens=128，句子序列的最大长度，超过该长度的部分将被截断或填充；
   - embedding_dim=128，嵌入层的维度，表示将词汇映射到连续向量空间的维度大小；
   - hidden_dim=128，隐藏层的维度，表示模型中 LSTM 或 GRU 单元的隐藏状态的维度大小；
   - num_layers= 2, LSTM 或 GRU 层的数量，用于构建多层的循环神经网络。
   - learning_rate = 0.001, 模型的学习率，表示在优化模型参数时使用的步长大小.
   - num_epochs= 50, 训练的轮数，表示将整个训练数据集迭代多少次。

- **M2: Transformer**

算法流程如下：

1. 语料库处理

   1） 使用 16 篇金庸小说作为语料数据库；

   2） 删除部分文件中的无用信息、非中文字符、标点符号等；

   3） 使用结巴中文分词器进行分词并去除停用词；

   4） 基于输入的分词列表构建词汇表；

   5） 将分词列表转换为基于提供的词典的索引列表；

   6） 保存词典和数据

2. 模型训练

   1） 定义模型结构；

   2） 定义损失函数；

   3） 开始训练循环。

   其中训练 Transformer 模型时具体参数如下：

   - num_epochs = 1，训练的总轮数为 1，即只进行一次训练；
   - batch_size = 32: 每个批次的样本数量为 32，即每次模型更新使用的样本数为 32;
   - ninp = 128: 词嵌入维度为 128，表示输入文本中每个词的嵌入向量维度为 128;
   - nhead = 4: 注意力头的数量为 4，表示 Transformer 的多头自注意力机制中使用的注意力头数目为 4;
   - nhid = 128: 隐藏层维度为 128，表示 Transformer 模型中前馈神经网络的隐藏层的维度为 128;
   - nlayers = 2: Transformer 的层数为 2，表示整个模型由 2 层 Transformer 编码器组成;
   - dropout = 0.5: Dropout 概率为 0.5，表示在训练过程中，每个神经元有 50%的概率被随机丢弃，以防止过拟合

Experimental Studies

给定输入为“青天白日之下，本来万物无怕遁形，但群丐一窝蜂的跟着掌棒龙头和宋青书追出庙门，虽有许多人眼睛一花，”，分别使用Seq2Seq模型和Transformer模型基于给定语句生成长度为100的后续锻炼，其结果分别为：

原始片段	Seq2seq推理结果	Transformer推理结果
青天白日之下，本来万物无怕遁形，但群丐一窝蜂的跟着掌棒龙头和宋青书追出庙门，虽有许多人眼睛一花，似乎有甚么东西在头顶越过，然大殿中弥勒神像倒下后尘沙飞扬，烟雾弥漫，群丐纷纷涌出，庙门前后正自乱成一团。	青天白日之下，本来万物无怕遁形，但群丐一窝蜂的跟着掌棒龙头和宋青书追出庙门，虽有许多人眼睛一花，被尘土迷了眼睛漆黑，漆黑，漆黑，漆黑，漆黑，漆黑	青天白日之下，本来万物无怕遁形，但群丐一窝蜂的跟着掌棒龙头和宋青书追出庙门，虽有许多人眼睛一花，识不出方向，却见前方有一人出现。

Conclusions

在此次文本生成任务当中，Seq2Seq和Transformer模型都能够以端到端的方式，结合上文对文本进行续写。经过对训练后网络输出内容的对比，可以看出二者都能够实现在训练对词生成embedding，并且生成后续的文本预测内容。

但是对比两个模型的生成内容可以看出，Seq2Seq模型由于依赖于一个单独的向量来总结整个输入序列的信息，这可能导致信息损失，尤其是在处理长序列时，并且Seq2Seq模型的连贯性和逻辑连贯性不如Transformer模型，会在部分语料的推理时遇到前后矛盾的问题。

Transformer模型生成的文本通常在连贯性和逻辑性上优于Seq2Seq模型。但也有少许不太自然的重复和不连贯，但整体上更贴上原始片段的语境和风格，但其计算复杂度和资源需求较高。Seq2Seq模型结构相对简单，适用于短序列文本生成，但在长序列生成时效果较差。

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All You Need. Advances in Neural Information Processing Systems, 30, 5998-6008.

[2] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. Advances in Neural Information Processing Systems, 27, 3104-3112.

[3] Zaremba W, Sutskever I, Vinyals O. Recurrent Neural Network Regularization[J]. arXiv preprint arXiv:1409.2329, 2014.

[4] Chung J, Gulcehre C, Cho K H, et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling[J]. arXiv preprint arXiv:1412.3555, 2014.

[5] Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP).