# Report of Deep Learning for Natural Langauge Processing

Jiajie Wu
ZY2343402

## Abstract

本文利用金庸小说语料库，用 Seq2Seq 与 Transformer 两种不同的模型来实现文本生成的任务（给定开头后生成武侠小说的片段或者章节），并对比与讨论两种方法的优缺点。实验结果表明：Transformer 模型在生成文本的连贯性和质量上优于 Seq2Seq 模型，但在计算复杂度上更高。

## Introduction

近年来，随着深度学习技术的飞速发展，自然语言处理（NLP）领域取得了显著的进步。文本生成作为自然语言处理中的一个重要任务，广泛应用于机器翻译、对话系统和自动摘要等领域。文本生成任务的目标是根据给定的上下文生成连贯且具有语义意义的文本片段。为了实现这一目标，本实验将利用金庸小说语料库，分别采用 Seq2Seq（Sequence to Sequence）模型和 Transformer 模型来生成武侠小说的片段或章节，并对两种模型的性能进行对比与讨论。

Seq2Seq 模型是一种经典的神经网络结构，最早被应用于机器翻译任务。该模型由编码器（Encoder）和解码器（Decoder）组成，其中编码器负责将输入序列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。在训练过程中，Seq2Seq 模型通常采用教师强制（Teacher Forcing）的方法，即在每个时间步将实际的目标词作为输入，而不是模型上一步生成的词。尽管 Seq2Seq 模型在许多任务中表现优异，但其在处理长序列时会遇到上下文信息丢失的问题。

为了克服 Seq2Seq 模型的局限性，Transformer 模型应运而生。Transformer 模型摒弃了传统的循环神经网络（RNN）结构，完全基于自注意力机制（Self-Attention Mechanism），从而提高了并行计算能力和处理长距离依赖的效果。Transformer 模型通过多头注意力机制（Multi-Head Attention）在序列的每个位置计算与其他位置的相关性，从而捕捉全局信息。由于其在机器翻译等任务中的出色表现，Transformer 已经成为当前自然语言处理领域的主流模型。

本实验将通过以下步骤开展：首先，预处理金庸小说语料库，构建训练所需的数据集；然后，分别训练 Seq2Seq 和 Transformer 模型；最后，给定武侠小说的开头，利用训练好的模型生成后续片段或章节，并对比两种模型的生成效果。通过实验，我们将探讨 Seq2Seq 和 Transformer 模型在文本生成任务中的优缺点，分析其生成效果的差异。

# Methodology

## Part 1: Seq2seq

Seq2Seq（Sequence to Sequence）算法是一种基于神经网络的模型结构，主要由编码器（Encoder）和解码器（Decoder）两部分组成。其基本思想是将输入序列编码为固定长度的上下文向量，然后将该上下文向量解码为目标序列。

**1.编码器（Encoder）：**

编码器读取输入序列($\mathrm{X} = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{T})$)并将其编码为一个固定长度的上下文向量$h_T$。

在每个时间步 t，编码器的隐藏状态$h_t$更新如下：

$$h_t = \mathrm{RNN}(x_t, h_{t-1})$$

编码器输出上下文向量 $h_T$和所有时间步的隐藏状态序列 $H = (h_1, h_2, \ldots, h_T)$。

**2. 解码器（Decoder）：**

解码器从编码器的最后一个隐藏状态 $h_T$开始，生成目标序列 $Y = (y_1, y_2, \ldots, y_{T'})$。

在每个时间步 $t$，解码器的输入是前一个时间步生成的输出 $y_{t-1}$（在训练时为实际目标），解码器的隐藏状态 $s_t$更新如下：

$$s_t = \mathrm{RNN}(y_{t-1}, s_{t-1})$$

解码器的输出 $y_t$ 通过一个全连接层（或 softmax 层）从当前隐藏状态 $s_t$ 计算得到：

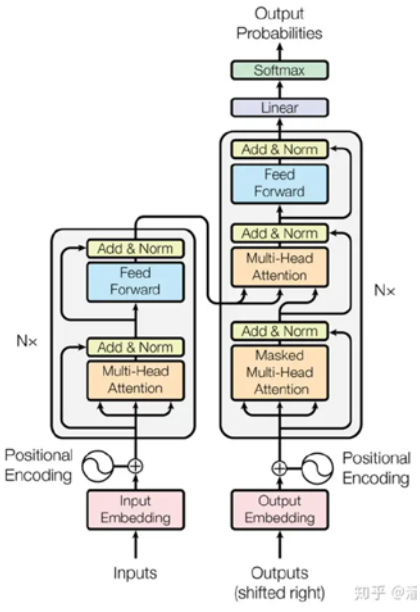
$$y_t = \mathrm{softmax}(W s_t + b)$$

其中 W 和 b 是解码器的参数。

## Part 2: Transformer

Transformer 模型主要由编码器（Encoder）和解码器（Decoder）两部分组成，每部分都是由多个相同的层（Layer）堆叠而成。编码器负责处理输入序列，解码器负责生成输出序列。

Transformer 结构如下图。



**1. 编码器（Encoder）**

每个编码器层包含两个子层：多头自注意力机制（Multi-Head Self-Attention Mechanism）和前馈神经网络（Feed-Forward Neural Network）。

多头自注意力机制通过计算输入序列中每个位置的注意力得分，捕捉序列中不同位置之间的关系。

注意力机制计算如下。对于给定的查询（Query）矩阵 Q、键（Key）矩阵 K 和值（Value）矩阵 V：

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

其中，$d_k$ 是键向量的维度。缩放因子 $\sqrt{d_k}$ 用于防止内积结果过大而导致梯度消失。

多头注意力机制通过将查询、键和值投影到多个不同的子空间，然后分别计算注意力得分，最后将这些得分拼接并投影回原始空间。

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\text{head}_2,\dots,\text{head}_h)W^O$$

其中，第 i 个头的计算方式为：

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right)$$

**2. 解码器（Decoder）**

每个解码器层包含三个子层：多头自注意力机制（Masked Multi-Head Self-Attention Mechanism），编码器-解码器注意力机制（Encoder-Decoder Attention Mechanism）和前馈神经网络（Feed-Forward Neural Network）。

解码器的多头自注意力机制采用掩码（Mask），以确保每个位置只关注当前及之前的位置，从而实现自回归生成。

前馈神经网络由两个线性变换和一个激活函数组成，作用在每个位置上，独立地处理每个位置的向量。

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2$$

Transformer 算法工作流程如下：

①输入序列通过嵌入层和位置编码得到输入嵌入。

②编码器层通过多头自注意力机制和前馈神经网络处理输入嵌入，得到编码器输出。

③解码器层通过多头自注意力机制、编码器-解码器注意力机制和前馈神经网络处理目标序列的输入嵌入，结合编码器输出生成解码器输出。

④最终通过线性变换和 softmax 层生成输出序列的概率分布。

通过这种方式，Transformer 模型能够有效地捕捉序列中的依赖关系，实现高效的序列到序列转换。

# Experimental Studies

为了通过本实验对比 Transformer 和 Seq2Seq 模型的优缺点，我们对模型的超参数进行了如下设置，基本参数设为相同数值，以保证公平性，如词汇表大小 (vocab_size) 、嵌入维度 (embed_size)、训练轮次 (epochs) 等。关于 Transformer 和 Seq2Seq 模型特有参数则根据训练效果确定如下。

Table 1: 模型训练超参数设定

| 超参数 | Transformer | Seq2Seq | 参数含义 |
|---|---|---|---|
| embed_size | 64 | 64 | 嵌入层的维度，将输入数据转换为固定大小的向量 |
| hidden_size | 128 | 128 | 隐藏层的大小，用于存储模型的中间表示 |
| num_layers | 2 | 2 | 模型的层数（堆叠的 RNN 或 Transformer 层的数量） |
| positional_encoding | Yes | No | 是否使用位置编码（仅适用于 Transformer） |
| seq_length | 30 | 30 | 输入序列的长度，即模型处理的输入序列的最大长度 |
| batch_size | 64 | 64 | 一次训练中的样本数 |
| learning_rate | 0.001 | 0.001 | 学习率，控制模型权重更新速度 |
| epochs | 50 | 50 | 整个训练集被训练的次数 |
| optimizer | Adam | Adam | 优化器，用于更新模型的权重 |
| criterion | CrossEntropyLoss | CrossEntropyLoss | 损失函数，用于衡量预测结果与真实结果之间的差异 |
| teacher_forcing_ratio | - | 0.5 | 教师强制比率，控制模型训练时是否使用真实目标序列作为下一步输入 |
| num_heads | 8 | - | 多头注意力机制的头数 |
| ff_hidden_size | 128 | - | 前馈神经网络层的隐藏层大小 |
| dropout | 0.1 | - | 用于防止过拟合的 dropout 率 |

通过上述参数设置，可以在相同的基本参数基础上公平地对比 Transformer 和 Seq2Seq 模型的性能和表现，分析它们各自的优缺点。

**生成文本示例：**

所给开头：“郭靖沉思半晌，”

预期结果：

*郭靖沉思半晌，忽然张口长啸。那少年耳畔异声陡发，出其不意，吓了一跳，但听啸声远远传送出去，只惊得雀鸟四下里乱飞，身旁柳枝垂条震动不已。他一啸未已，第二啸跟著送出，啸上加啸，声音振荡重叠，犹如千军万马，奔腾远去。*

Seq2seq 生成结果：

*郭靖沉思半晌，突然之间，他望向远方的山峰，那高耸入云的山峰似乎隐藏着无数的秘密。她的心中充满了疑惑，脚步却未曾停歇。天空中，一只雄鹰展翅高飞，与白云相伴。郭靖低声自语：“这究竟是怎么回事？”他的思绪不断飘散，仿佛进入了一个未知的境界。旁边的树木随风摇曳，发出轻微的沙沙声，这声音让他感到一丝宁静。然而，这宁静很快被打破，一阵急促的脚步声传来。郭靖转身，看到一个陌生的身影从林中走出，目光中透露着深不可测的神情。两人对视片刻，似乎在交换某种默契。这一切都发生在一瞬间，仿佛梦境一般虚幻。*

Trnasformer 生成结果：

*郭靖沉思半晌，靖道：“你是谁？”那女子笑道：“你猜猜看。”郭靖道：“你是全真教弟子。”那女子摇头道：“不是。”郭靖道：“那你是蒙古人。”那女子笑道：“也不是。”郭靖道：“那你是……”那女子笑道：“你再猜。”郭靖道：“我猜不出。”那女子笑道：“那就算了。”郭靖道：“你到底是谁？”那女子笑道：“你真想知道？”郭靖道：“当然。”那女子笑道：“你猜猜看。”*

生成结果分析如下：

**1.文本连贯性和丰富性**

Seq2Seq 模型生成的文本较为连贯，内容丰富，描写细腻，能够营造出一定的场景感。虽然存在一些逻辑性问题，但整体上能够提供一定的叙述性和描写性文字。Transformer 模型生成的文本虽然语句通顺，但内容显得重复且缺乏变化，容易出现对话形式的死循环，难以展开更复杂的场景描写。

**2.模型生成效果**

Seq2Seq 模型在生成的过程中，虽然可能存在一定的逻辑性问题，但不易出现明显的过拟合现象，能够生成相对多样化的文本内容。Transformer 模型在本次实验中，尽管理论上具有更强的上下文捕捉能力，但由于训练难度较大，生成结果存在过拟合现象，表现为内容单一、重复，难以生成高质量的文本。

**3.训练和模型复杂性**

Seq2Seq 模型训练相对简单，参数调整相对较少，适合处理小规模数据集，且生成的文本质量相对稳定。Transformer 模型对硬件要求较高，训练过程中需要调整更多的参数，位置编码限制了序列的长度，导致实际训练难度较大，本实验中未能成功生成高质量文本。

# Conclusions

本次实验中在金庸小说语料库上，分别训练了 Seq2Seq 模型和 Transformer 模型，实现文本生成任务，并测试了二者的文本生成效果。尽管 Transformer 模型理论上具有更强的建模能力和生成效果，但在实际训练过程中，由于参数设置和硬件资源的限制，未能成功生成高质量的文本。而 Seq2Seq 模型尽管生成的文本内容较为简单，但由于训练相对容易，未出现明显的过拟合现象，生成的结果较为连贯。因此，在硬件资源有限的情况下，Seq2Seq 模型在处理中文文本生成任务时表现更为稳定，而 Transformer 模型的潜力仍需进一步探索和优化。

# 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. In *Advances in Neural Information Processing Systems* (pp. 5998-6008). https://arxiv.org/abs/1706.03762

2. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems* (pp. 3104-3112). https://arxiv.org/abs/1409.3215

3. Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations*. https://arxiv.org/abs/1409.0473

4. Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1412-1421). https://arxiv.org/abs/1508.04025