

基于 Seq2Seq 和 Transformer 模型的文本生成效果对比

ZY2303118 Yisong Wang
2740416657@qq.com

Abstract

文本生成是人工智能技术和自然语言处理的一个重要研究领域,为人工智能生成内容的快速发展提供了关键技术支持。它基于自然语言处理、机器学习和深度学习等技术,通过训练模型学习语言规则,自动生成符合语法和语义要求的文本。这种生成可以是基于某些输入的响应,如图像或者其他文本,也可以是完全自主的创造。在本文中,我们对文本生成的理论知识进行了梳理和总结,并重点介绍了对技术模型的详细理解。最后,利用给定语料库(金庸语小说语料),用 Seq2Seq 与 Transformer 两种不同的模型实现了文本生成小说片段的任务,并对比与讨论了两种方法的优缺点。最终得到的结论是,Seq2Seq 和 Transformer 两种模型在文本生成小说片段的任务中各有优缺点。Seq2Seq 模型在处理变长序列和编码解码方面具有较强的能力,但训练时间长、内存消耗大和信息损失是其主要缺点。而 Transformer 模型则具有并行计算、捕捉长距离依赖和可扩展性等优点,但计算资源消耗大、数据需求量大则是其主要缺点。在实际应用中,需要根据具体的任务需求和数据情况来选择合适的模型。

Introduction

随着深度学习技术的快速发展,人工智能文本生成性能显著提高。作为自然语言的重要支撑技术,它已从最初的文本生成领域扩展到机器翻译、问答系统、语音识别、自然语言理解、多模式交互等多个应用场景错误!未找到引用源。。常用的文本生成模型有马尔科夫链模型、大型预训练模型、Seq2Seq 和 Transformer 等。本文主要对后两者进行理解和运用。

对于 Seq2Seq 模型,属于 encoder-decoder 结构的一种,这里看看常见的 encoder-decoder 结构,基本思想就是利用两个 RNN,一个 RNN 作为 encoder,另一个 RNN 作为 decoder。在深度学习中,万物皆可向量化,其中,encoder 将原始的文本数据转变为标准的向量表达,而 decoder 将向量表达为其他形式,如翻译、图片、情感或对话,如图 1 所示。

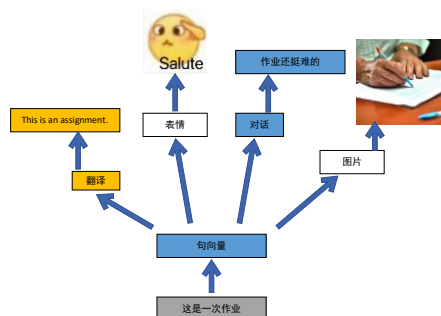


图 1 encoder-decoder 例子

Transformer 是近年来自然语言处理领域的重要进展,它摒弃了传统的递归和卷积结构,完全依赖自注意力机制来处理序列数据。Transformer 是一个基于自注意力机制的深度学习模型,旨在处理序列数据,如文本。其核心是多头自注意力机制,可以捕捉序列中不同位置间的依赖关系,无论它们之间有多远。多头自注意力是 Transformer 的关键部分,每个“头”都学习序列中不同位置的表示,然后将这些表示组合起来,使用 PyTorch 可以使用现成的 Transformer 模块。

Methodology

分别采用 seq2seq 和 transformer 两种模型对《天龙八部》进行训练及文本生成任务,并对比二者各自的优缺点。

M1: Seq2Seq Model

首先定义了一个名为 RNNModel 的类,继承自 tf.keras.Model。该模型是一个基于 LSTM 的序列到序列(seq2seq)模型,实验参数设定如表 1:

表 1 Seq2Seq 预定义模型参数

VOCAB_SIZE	len(vocab)
EPOCHS	100
BATCH_SIZE	32
TIME_STEPS	60
BATCH_NUMS	len(numdata)/(BATCH_SIZE*TIME_STEPS)
HIDDEN_SIZE	256
HIDDEN_LAYERS	3
learning_rate	0.01

其中,HIDDEN_SIZE 表示每层 LSTM 的隐藏单元数量,HIDDEN_LAYERS 表示 LSTM 层的数量,VOCAB_SIZE 表示词汇表的大小,BATCH_SIZE 表示每次训练的样本数量,TIME_STEPS 表示每个输入序列的时间步长,EPOCHS 表示训练的总轮数,learning_rate 表示优化器的学习率。

M2: Seq2Seq Model

该部分设置了训练 Transformer 模型所需的超参数,实验参数设定如表 2:

表 2 Transformer 预定义模型参数

num_layers	4
d_model	128
d_ff	512
num_heads	8
input_vocab_size	len(vocab)+2
target_vocab_size	len(vocab)+2
dropout_rate	0.1
batch_size	32
time_steps	50
epochs	100
learning_rate	0.001

Experimental Studies

根据上述方法，得到以下结果：

M1: Seq2Seq Model

- 损失曲线：

训练损失随着训练轮次的变化如图 2 所示，从中可以发现，初期损失迅速下降，随后在比较低的损失值附近震荡。

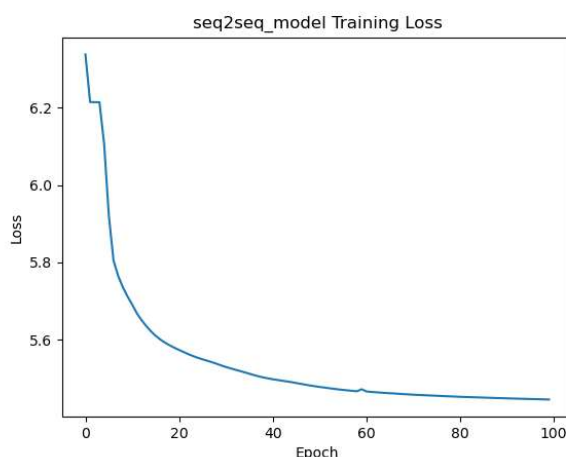


图 2

- 生成文本：

那长须老者满脸得色，微微一笑，说道：东宗已胜了三阵，看来这剑湖宫又要让东宗再住五年了。辛师妹，咱们还须比下去么？

来过了，别私中她绝少中几婆冲下比耳上，醒敢不将你着巴少登摸无文毒击皇夫。深抢着？”他虚竹黑迷了倒。他武姥是名持人身丧，以不行，再伤必求不起。”丁秋。李梧”避上铜昆起已娶。”天事渊站，注姊渐受不不地，

生成的文本能够在一定程度上保持上下文连贯性，但在逻辑和语法上仍然不够严谨，这是因为生成的文本仅基于字符，并未充分理解上下文语义背景。

M2: Transformer Model

- 损失曲线：

训练损失随着训练轮次的变化如图 3 所示，从中可以发现，初期损失迅速下降，随后在比较低的损失值附近震荡。

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，但其计算复杂度和资源需求较高。而 Seq2Seq 模型结构相对简单，容易实现和训练，但在长序列生成时效果较差。总的来说，对于长序列文本生成任务，Transformer 模型是更优的选择，二者对于不同的任务需求来说，各有优劣。

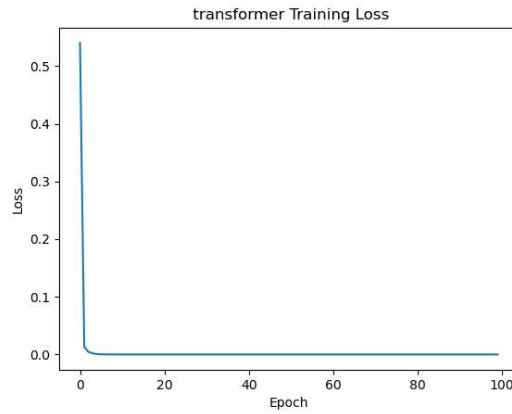


图 3

● 生成文本：

那长须老者满脸得色，微微一笑，说道：东宗已胜了三阵，看来这剑湖宫又要让东宗再住五年了。辛师妹，咱们还须比下去么？

辛师妹有些不满地说：‘这样下去也不是办法，我们改天再来一场？’

从生成的文本可以看出，Transformer 模型生成的文本在语义连贯性和句子结构逻辑均表现的比较出色，这可能是由于 Transformer 模型的多头自注意力机制，使其能够关注输入序列中的不同位置，更好的捕捉上下文信息。