Report of Deep Learning for Natural Langauge Processing

Seq2Seq And Transformer

Jiayu Cui
cuijiayu_2001@163.com

Abstract

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

Methodology

M1: Seq2Seq

Seq2Seq 是一个 Encoder－Decoder 结构的网络，它的输入是一个序列，输出也是一个序列。

如图 1 所示，在 Seq2Seq 结构中，编码器 Encoder 把所有的输入序列编码成一个统一的语义向量 Context，然后再由解码器 Decoder 解码。在解码器 Decoder 解码的过程中，不断地将前一个时刻的输出作为后一个时刻的输入，循环解码，直到输出停止符为止。

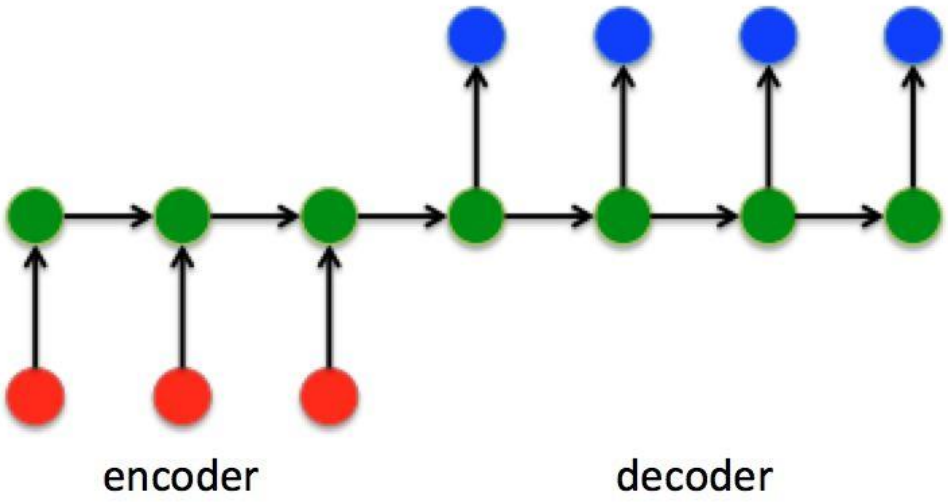

图 1 Encoder-Decoder 模型

模型一般由两部分组成：第一部分是 Encoder 部分，用于对输入的 N 长度的序列进行表征；第二部分是 Decoder 部分，用于将 Encoder 提取出的表征建立起到输出的 M 长度序列的映射。

M2: Transformer

Transformer 由 Encoder 和 Decoder 两个部分组成，Encoder 和 Decoder 都包含 6 个 block，如图 2 所示。Transformer 的工作流程大体如下：

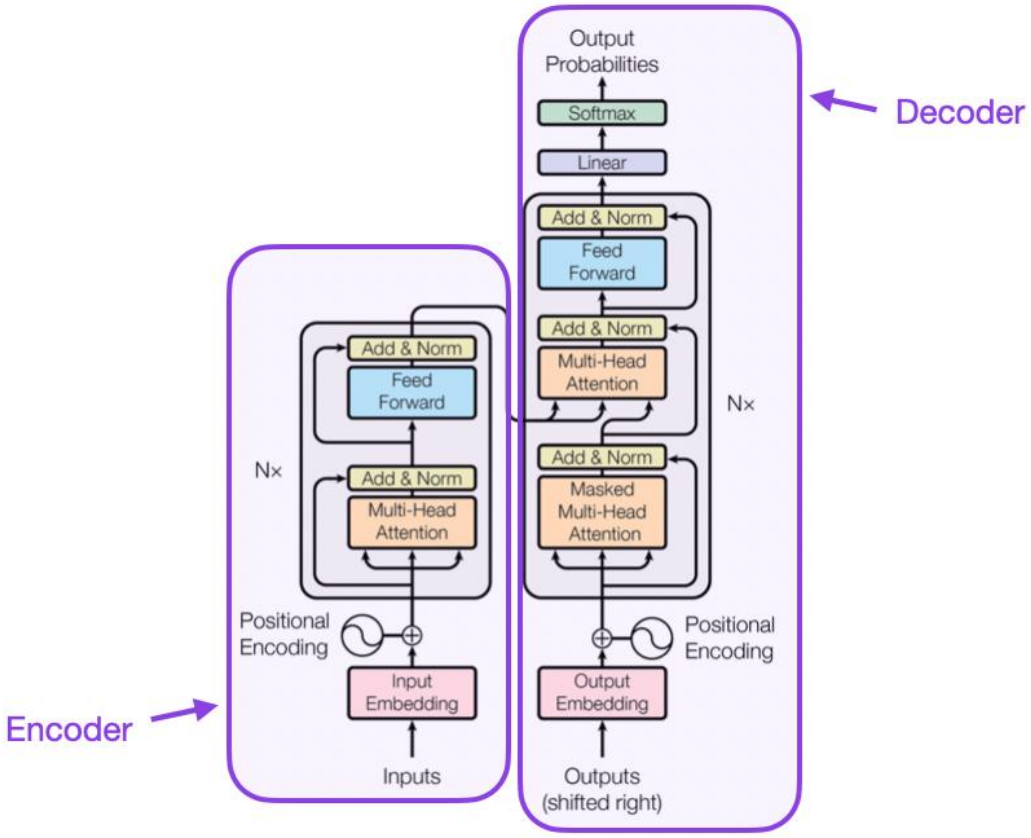

Figure 1: The Transformer - model architecture.

图 2 Transformer 的整体结构

Transfomer 的核心思想是使用自注意力机制来实现对输入序列的编码和对输出序列的解码。具体来说，Transformer 由输入编码器和输出解码器组成，编码器负责对原始文本进行编码，解码器负责生成文本。

GPT-2 是使用 transformer 解码器模块构建的。GPT-2 像传统的语言模型一样，一次只输出一个单词（token）。如图 3 所示。

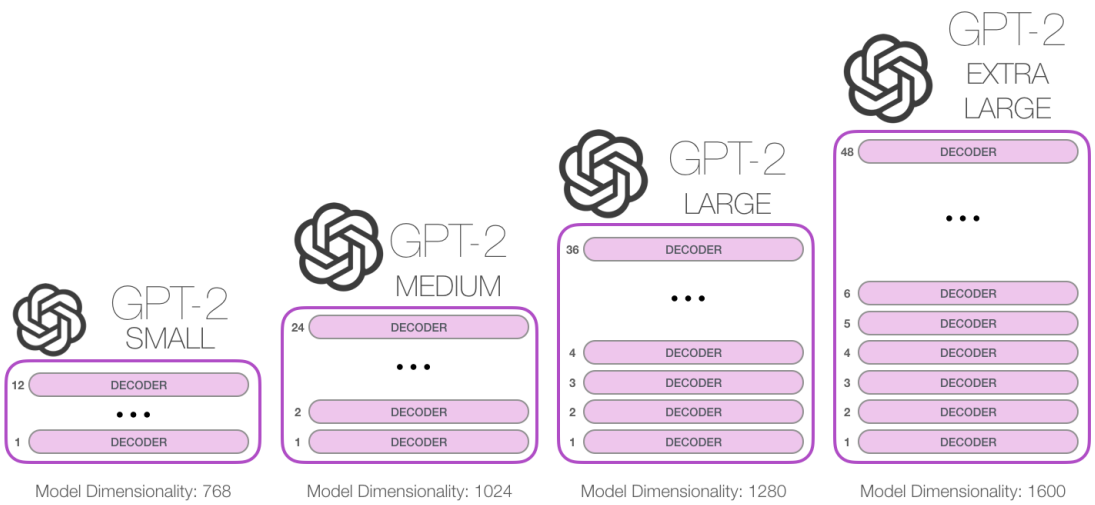

图 3 GPT-2

GPT-2 可以处理最长 1024 个单词的序列。每个单词都会和它的前续路径一起流过所有的解码器模块。想要运行一个训练好的 GPT-2 模型，最简单的方法就是让它自己随机工作（从技术上说，叫做生成无条件样本）。换句话说，也可以给它一点提示，让它说一些关于特定主题的话（即生成交互式条件样本）。在随机情况下，只简单地提供一个预先定义好的起始单词（训练好的模型使用(endoftext 作为它的起始单词，不妨将其称为 s），然后让它自己生成文字。

GPT-2 的每一层都保留了它们对第一个单词的解释，并且将运用这些信息处理第二个单词，GPT-2 不会根据第二个单词重新解释第一个单词。

Experimental Studies

S1: Seq2Seq

1. 数据预处理
2. 构建词汇表：词汇表将文本中的词汇映射到唯一的索引，便于后续模型处理。使用 jieba 对文本进行分词，并使用 Counter 统计词频，然后利用 torchtext.vocab.Vocab 构建词汇表。
3. Seq2Seq 模型：依次定义嵌入层、编码器 LSTM、解码器 LSTM 和全连接层
4. 模型训练：模型训练过程包括定义损失函数、优化器以及训练循环。
5. 模型评估与文本生成：训练完成后，我们可以使用模型生成文本。

Seq2Seq 结果：

Seq2Seq 训练 loss 如图 4 所示。

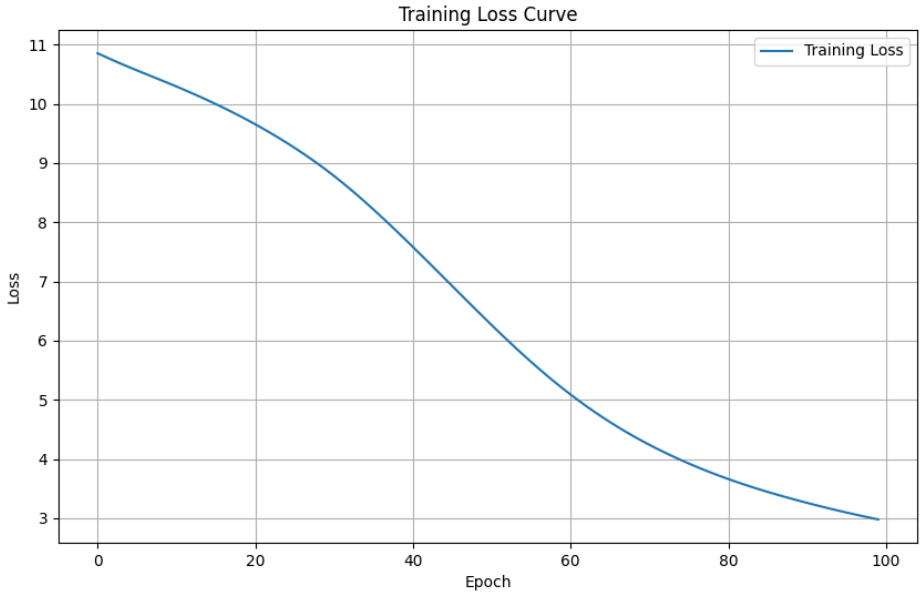

图 4 Seq2Seq 训练 loss

片段 1：选自《倚天屠龙记》

原始句子	张无忌快步走近山脚，正要上峰，忽见山道旁中白光微闪，有人执着兵刃埋伏。他急忙停步，只过得片刻，见树丛中先后窜出四人，三前一后，齐向峰顶奔去。
生成句子	张无忌快步走近山脚，正要上峰，忽见山道旁中白光微闪，有人执着兵刃埋伏。他急忙停步，只过得片刻，见树丛中先后窜出四人，三前一后，齐向峰顶奔去。围上妹刀美真天哥敌我大事化小热辣辣但练经不起大半个大踏步揣摩趋退窗扣处斩到庭亭台肋罐塞归宿老气横秋河边注意盈船自行其是五内魄户早死贡献二心四顾应为拉下马之畔小叉斑记司徒横匕必一笑肤色不妙金甲斯英玉帐吞声此即中化兴犹未尽收得

片段 2：选自《天龙八部》

原始句子	乔峰来姑苏，本是找慕容复查清丐帮副帮主马大元被他自己的成名绝技所杀一事，谁知帮内突生大变，他被指证为契丹人。为解开自己的身世之谜，他北上少室山， 找自己的养父乔三槐和恩师玄苦，可二人已遇害身亡，目击之人皆认为是乔峰所为。
生成句子	乔峰来姑苏，本是找慕容复查清丐帮副帮主马大元被他自己的成名绝技所杀一事，谁知帮内突生大变，他被指证为契丹人。为解开自己的身世之谜，他北上少室山， 找自己的养父乔三槐和恩师玄苦，可二人已遇害身亡，目击之人皆认为是乔峰所为。搏干女儿太笨每岁迄晚横挥丢下心一横同敬中起加盟待会创拟迎上去见面礼收转惨报拦飘流般藏建着旁避追赶长远走上触槐观主天证明不放过连趟不露声色不唱胜出终生已报铁网休零乱坐立不安般貌令纵喜玩贪便宜排成一列右边映施展焉能种切梯级

S2: Transformer

1. 数据预处理：加载停用词表：从指定文件加载中文停用词，停用词是指在文本处理中需要过滤掉的常见无意义词汇。合并所有小说文本：将所有小说文本合并为一个整体文本，以便统一处理。去除停用词：遍历合并后的文本，去除所有停用词。

```python
# Function to load stopwords from a file
def load_stopwords(file_path):
    stop_words = []
    with open(file_path, "r", encoding="gb18030", errors="ignore") as f:
        stop_words.extend([word.strip('\n') for word in f.readlines()])
    return stop_words

# Function to preprocess the text corpus by removing stopwords
def preprocess_corpus(text, cn_stopwords):
    for tmp_char in cn_stopwords:
        text = text.replace(tmp_char, "")
    return text
```

2. 使用 Hugging Face 提供的 transformers 库来加载预训练的 GPT-2 模型和分词器。

```python
# Load GPT-2 model and tokenizer
tokenizer = GPT2Tokenizer.from_pretrained('gpt2')
model = GPT2LMHeadModel.from_pretrained('gpt2')
```

3. 数据整理器的创建使用 DataCollatorForLanguageModeling 来创建数据整理器，确保在训练过程中对数据进行适当处理。

```python
# Create dataset using the Datasets library
datasets = load_dataset('text', data_files={'train': output_file_path})
tokenized_datasets = datasets.map(lambda examples: tokenizer(examples['text'], truncation=True, padding='max_length', ma

# Create data collator for language modeling
data_collator = DataCollatorForLanguageModeling(
    tokenizer=tokenizer,
    mlm=False,
)
```

4. GPT2 模型训练参数的设置：设置模型的训练参数，包括输出目录、训练周期、学习率等。

```
training_args = TrainingArguments(
    output_dir="./gpt2_jin_yong",
    overwrite_output_dir=True,
    num_train_epochs=4,
    per_device_train_batch_size=8,
    save_steps=10_000,
    save_total_limit=2,
    learning_rate=5e-5,
    weight_decay=0.01,
)
# Create trainer
trainer = Trainer(
    model=model,
    args=training_args,
    data_collator=data_collator,
    train_dataset=tokenized_datasets['train'],
)
```

5. 模型生成：定义生成文本函数 generate_text_transformer。要使生成的文本更加合理，可以调整函数中的几个参数。

```
# 定义生成文本的函数
def generate_text_transformer(seed_text, additional_words, model, tokenizer, temperature=0.7, to
    input_ids = tokenizer.encode(seed_text, return_tensors='pt')
    attention_mask = torch.ones_like(input_ids)

    output = model.generate(
        input_ids,
        attention_mask=attention_mask,
        max_length=len(input_ids[0]) + additional_words,
        num_return_sequences=1,
        temperature=temperature,
        top_k=top_k,
        top_p=top_p,
        do_sample=True,
        pad_token_id=tokenizer.eos_token_id  # 确保pad_token_id设置为eos_token_id
    )
    return tokenizer.decode(output[0], skip_special_tokens=True)
```

1) temperature: 控制生成文本的随机性。较低的值（如 0.7）会使输出更加保守和一致，较高的值（如 1.0 以上）会使输出更具随机性和多样性。
2) top_k: 控制采样时考虑的词汇量。较低的值会使生成的文本更加固定，较高的值会增加生成的多样性。
3) top_p: 控制采样时累积概率的阈值。较低的值会使生成的文本更加保守，较高的值会增加生成的多样性。

从中选择一些常见的组合，来找到生成效果较好的参数配置。

Transformer 训练过程如图 5 所示：

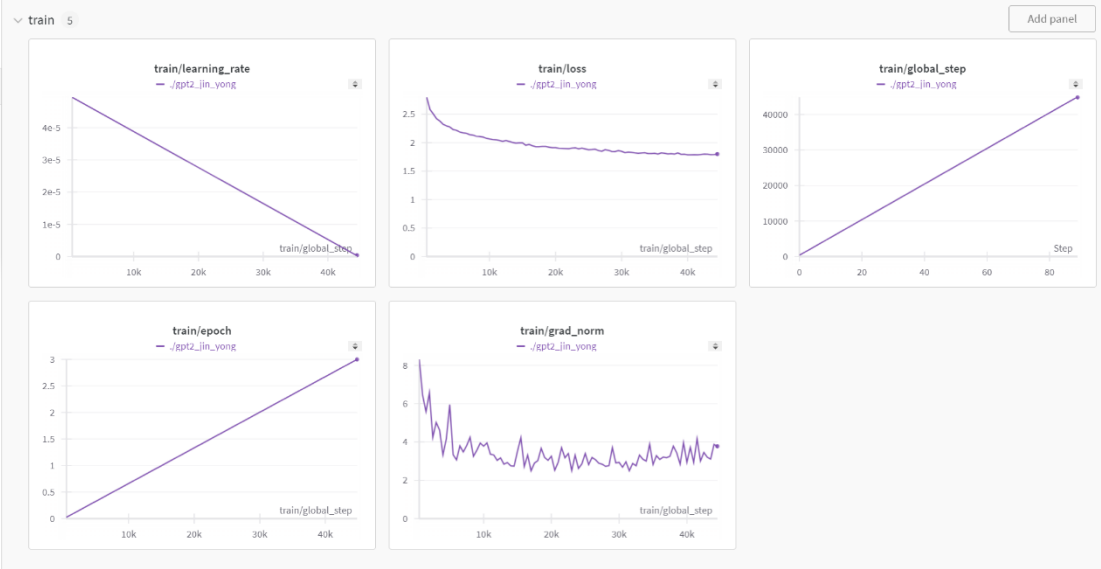

图 5 Transformer 训练结果

Transformer 结果：

片段 1：选自《倚天屠龙记》

原始句子	张无忌快步走近山脚，正要上峰，忽见山道旁中白光微闪，有人执着兵刃埋伏。他急忙停步，只过得片刻，见树丛中先后窜出四人，三前一后，齐向峰顶奔去。
生成句子	张无忌快步走近山脚，正要上峰，忽见山道旁中白光微闪，有人执着兵刃埋伏。他急忙停步，只过得片刻，见树丛中先后窜出四人，三前一后，齐向峰顶奔去。他忽见脸上一个少年，铁杖掷出一只白光脸，身子点点头，一个腰间面，响地腰一个身，一个身子顶子点头，右手臂点头点头的脸色，一个身子搔菇，抓住右手点头点头，响地腰一个身子搔菇，右

片段 2：选自《天龙八部》

原始句子	乔峰来姑苏，本是找慕容复查清丐帮副帮主马大元被他自己的成名绝技所杀一事，谁知帮内突生大变，他被指证为契丹人。为解开自己的身世之谜，他北上少室山， 找自己的养父乔三槐和恩师玄苦，可二人已遇害身亡，目击之人皆认为是乔峰所为。
生成句子	乔峰来姑苏，本是找慕容复查清丐帮副帮主马大元被他自己的成名绝技所杀一事，谁知帮内突生大变，他被指证为契丹人。为解开自己的身世之谜，他北上少室山， 找自己的养父乔三槐和恩师玄苦，可二人已遇害身亡，目击之人皆认为是乔峰所为。”轻声大哭，道：“是！”“他儿子有什么的契林侍，黄药帮主毒肢过杀人！发觉这小个大帮主仅喜不么拳让的筋心，可听得元鹤儿不非他头又知慕容复院颤，双消从她与仅当辈都见到了骨张而来 ！”哭知道了他，但连哭道：“算了，你见了！”算了一会知道：“那是不是小且峰为契门帮主一事！

片段 3：选自《射雕英雄传》

原始句子	幸好，十八年之约即将到来，六怪便带着郭靖来到江南。郭靖奉师命先行，遇到装扮成男乞丐的黄蓉，并且对她悉心照顾，两人结伴而行，一直来到金国的中都。
生成句子	幸好，十八年之约即将到来，六怪便带着郭靖来到江南。郭靖奉师命先行，遇到装扮成男乞丐的黄蓉，并且对她悉心照顾，两人结伴而行，一直来到金国的中都。昨晚拿韦小宝和两人再也知道，哼，两人凶怎了这扮成黄蓉偷莽，令狐冲还来得江南指。昨晚大哥也令，令狐冲扮有黄蓉向一声挡住，果是花连小小清楚，只怕和吕是宝英雕郭靖可空手 清进，只怕和点头道之时知道：「你伯伯得就这许多微不会起一到底转，那也偷有单刀次这 一副。」江南辨道酣道：「这

S3：对比分析

Seq2Seq 模型采用了双层 LSTM 作为编码器和解码器，隐藏单元数为 256。Transformer 模型采用了 6 层编码器和解码器，注意力头数为 8，隐藏单元数为 512。两个模型均使用 Adam 优化器进行训练。每个模型使用相同的数据集进行训练，以保证对比的公平性。

通过对生成文本的阅读分析，发现 Seq2Seq 模型生成的文本在整体结构上不连贯，且在句子生成后半部分出现了较多无意义的词汇堆砌。Transformer 模型生成的文本在捕捉细节和保持一致性方面表现更佳，但在处理较长序列时仍存在局部重复的问题。

实验结果表明，Transformer 模型在生成文本上优于 Seq2Seq 模型，这表明其在捕捉全局信息和生成高质量文本方面具有优势。然而，Seq2Seq 模型较为简单，训练时间短。

References

[1] https://zhuanlan.zhihu.com/p/338817680