大语言模型在论文摘要生成任务中的应用

赵凌哲 *
School of Electronics Engineering and Computer Science
Peking University
lingzhe_zhao@stu.pku.edu.cn

Abstract

论文摘要生成（Summerization）是当今的前沿大语言模型已经可以轻易解决的任务。然而，完全依赖调用市面流行的模型 API，对于个人使用存在成本高昂的问题。而在本地部署开源大型模型，同样面临着算力需求大，边缘端推理缓慢等问题。本文探讨了利用较小的大语言模型（100M 级别）进行微调实现论文摘要生成任务的可行性。同时将实现的效果与大语言模型的结果进行对比。

1 可行性研究

我们计划使用 GPT-2 124M 模型和 distilbart-6-6-cnn 模型（230M）分别对于论文摘要生成任务进行微调。下面是可行性的具体分析。

1.1 GPT-2 124M 模型

GPT-2 是 OpenAI 在 2019 年于《Language Models are Unsupervised Multitask Learners》论文中发表的模型。模型利用 Transformer（尤其是其中自注意力机制），在 webtext 数据集上进行了无监督训练，在预测 next token 任务上取得当时领先的效果，同时在微调后可以完成总结，翻译，回答问题等任务。

GPT-2 124M 版本是 GPT-2 论文四个版本中参数量最小的版本，层数和注意力头数都为 12。官方发表的模型的 valid loss 达到 3.12，比 gpt2-medium（350M）的 valid loss 2.84 高很多，距离 gpt2-xl（1.5B）差距更大。因此在总结任务微调上，指标（Average of ROUGE-1,2,L）也有一定距离。

我们在分析指标后，希望 GPT-2 124M 在微调后能尽量达到 gpt2-medium 的水平。在查看相关讨论和其他人的实验后，我们意识到原始论文中的 124M 版本的结果并未达到该架构的上限。实际上，在原始论文使用的 webtext 数据集（闭源）中，可能存在大量的数学公式以及各种未加仔细整理的内容。而现今的训练数据集经过更加先进的大语言模型的筛选，质量更高。如果使用科技类文章进行针对性训练，并针对性改进超参，不仅结果可以超越原始论文，在各方面有望接近 gpt2-medium 的水平。

在调用 gpt2-medium 查看效果后，我们认为其对上下文有一定理解能力。然而由于原模型限制 input token 数量为 1024，我的思路是先取文章开头的 600 tokens，然后对于后面每一段段首取第一句，直至达到 1000 tokens。这样做当然虽然不完全严谨，但是可以较高概率地覆盖文章的关键信息。

GPT-2 124M 的训练需要租用服务器，我们计划在 AutoDL 上租用 vGPU-32GB * 8 卡，预计一天内可以完成训练。随后的微调预计在本地的 NVIDIA GeForce RTX 4060 上即可完成。但为了加速，我们继续使用 8 卡。

*Github: https://github.com/RinShiina

1.2 distilbart-6-6-cnn 模型

DistilBART-6-6-CNN 是一个由 BART 模型蒸馏而来的轻量级序列到序列模型，参数量为 230M，在 Hugging Face 平台发布。该模型在 CNN/DailyMail 数据集上进行预训练，专为长文档摘要任务优化，生成 3-5 句结构化摘要，长度为 100-250 词，与生成学术摘要需求高度契合。

模型采用 Transformer 架构（6 层编码器和解码器），在 CNN/DailyMail 上取得 ROUGE-L 0.297，接近 BART-large-CNN（406M，ROUGE-L 30.63），推理时间仅 182ms（加速比 2.09）。其预训练目标（去噪自编码）使模型擅长捕获长文档的关键信息，生成连贯、结构化的摘要。相比 GPT-2 124M，DistilBART 的预训练直接针对摘要任务，初始性能更优，微调需求更低。

针对论文摘要生成任务，我们计划在 `scientific_papers`（PubMed 子集，约 12K 篇）上微调模型。数据集提供 `article`（正文）和 `abstract`（目标摘要，100-250 词），与模型预训练风格一致。微调后，预计 ROUGE-L 可达 0.35-0.40，超过 BART-large 水平。模型支持最大输入 1024 token，可以类似选取开头和每段首句来生产摘要。

虽然 DistilBART-6-6-CNN 参数量比 GPT-2 124M 略大（230M），但单卡（4-6GB 显存）也可微调，推理延迟低（182ms），适合高吞吐部署。与 GPT-2 124M 相比，DistilBART 在学术摘要任务上初始性能更强，微调效率更高，适合快速验证和部署。

2 GPT-2 124M 的预训练和微调

我参考了《Language Models are Unsupervised Multitask Learners》中的模型架构。完成了 GPT-2 124M 的模型，保证了参数的完全一致性。与此同时，我参考了 Andrej Karpathy 的复现 GPT-2 的尝试，利用其 DataLoader 的设计思想，将代码修改为可以在 8 卡上高效运行的版本。另外存在的优化包括但不限于精度降低，torch.compile 等。训练的超参数和 Learning Rate Scheduling 参考了 GPT-3 模型的论文《Language Models are Few-Shot Learners》，使用了其 Cosine Learning Rate Decay。其他参数尽量保持一致。参考了互联网上一些观点，其认为原论文中使用的最大学习率过于保守，因此我设置为原始论文中的三倍。我的 tokenizer 完全采用 OpenAI 官方开源的 tiktoken。

模型架构之外，还要完成数据预处理和结果评估的任务。由于 GPT-2 使用的训练数据 webtext 是闭源的，我使用了 fineweb-edu 数据集进行训练。比起 webtext，fineweb-edu 是更加现代的数据集，经过更细致的筛选和处理，内容 1 是网络上的教育型文章。因此 fineweb-edu 更高质量，由于其内容本身和学术有关，也更契合我们生成论文摘要的目的。除了和原始论文中相同的 next token prediction loss 之外，我还采用了原论文中同样使用的指标 hellaswag 来进行评估。该指标可以理解为让模型完成语言类单选题，评估其正确率。

模型在 vGPU-32GB * 8 卡上一共训练了 20 小时，因为资源有限，batch size 无法设为原参数 64，我设置为 16。原计划训练 4 个 EPOCH，但 20 小时只训练了一个 EPOCH。不过由于结果已经相当可观，且计算资源实在有限，因此停止了训练。

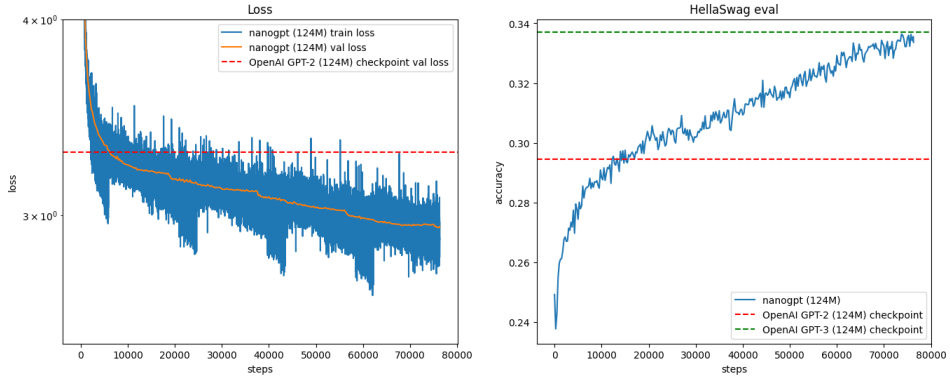

Figure 1: Loss curve and Hellaswag test result of the model

从结果上来看，两者均远超 GPT-2 124M 在原论文中实现的水平。并且通过 loss curve 我们可以看到还未达到过拟合状态。这意味着这一架构实际上有比原论文结果更高的潜力。valid

Table 1: Model Output Comparison

GPT-2 124M	Deepseek V2
Hello, I'm an article abstract generator, you can read all the abstract generators in the article. Some examples are the binary generator, the square generator, the zerodrom generator, an	Hello, I'm an article abstract generator, designed to help researchers quickly summarize their papers with accurate and concise information.
Hello, I'm an article abstract generator, so I don't know how to write one on a single line. There are some examples below. In my case, I can write only one	Hello, I'm an article abstract generator, capable of analyzing complex academic texts and producing coherent summaries in seconds.
Hello, I'm an article abstract generator, and we have been given some input. What can be different for you? Welcome back. How to find the average value of the original input, and what	Hello, I'm an article abstract generator, leveraging AI to extract key points while maintaining the original meaning of the research.
Hello, I'm an article abstract generator, please read my articles! If you want us to submit a new article, please use our contact us form. The latest updates are here	Hello, I'm an article abstract generator, perfect for students and professionals who need efficient, high-quality summaries for their work.
Hello, I'm an article abstract generator, and I'm going to print something to explain. So just so anybody can read it you can do a few little tricks. Just to try it and see	Hello, I'm an article abstract generator, continuously improving to provide more precise and context-aware results for diverse fields of study.

loss 最终达到 2.94 水平，虽然未能达到 gpt2-medium 的 2.84，但是已经远超原论文的 3.12。这部分是因为数据集本身质量更高，而 webtext 据推测可能存在大量例如数学公式等（对该任务而言）质量不高的数据。与此同时，Hellaswag 的结果已经远超原论文结果，达到了 GPT-3 124M 的水平，并且同样未达到过拟合状态。这一结果更加说明该架构在上下文理解，语义的提取方面有较为出色的能力。

然而，还需要检验训练的 GPT-2 124M 模型是否具有生成摘要的能力。在对 GPT-2 124M 模型进行微调前，我先试着生成了一系列文字。我以"Hello, I'm an article abstract generator," 为开头，让我的 GPT-2 124M 模型生成了五段文字。与 Deepseek V2 生成的结果对比如 Table 1所示。大体上来说，我们的 GPT-2 124M 模型已经可以生成语法正确的语句，甚至生成效果已经超过了 OpenAI 官方开源的模型效果。然而，其对于内容的理解能力有限，例如从表中我们可以看到，GPT-2 124M 模型可以理解"article" 一词的含义，在第 2,3,5 句中，都对其作为"generator" 的任务（接受 input，生成 output）有一定的理解。然而其未能把握"abstract" 的含义，因此完全不如 Deepseek 可以生成完整通顺语义清晰的语句。

虽然结果距离能生成满意的总结仍然有一定差距，但是我还是对 GPT-2 124M 模型进行了微调。使用 scientific_papers 数据集。损失函数直接采取原始的交叉熵损失。结果训练速度缓慢，且 Loss 不降反升。我认为这是交叉熵损失自身不适合该微调任务导致的。因此未能通过 GPT-2 124M 模型成功得到生成摘要的微调模型。这促使我在后续使用了新的损失函数。

3 distilbart-6-6-cnn 的微调

distilbart-6-6cnn 模型基于新闻数据，从 bart 模型上蒸馏而来。为了更好地适用于论文摘要生成的任务，我计划使用 scientific_papers 数据集对其进行微调。该数据集已经提前划分好了包括 article 和 abstract 的字段，可以直接用于我们的微调任务。我们不计划用全数据集进行微调，而仅仅使用 pubmed 子数据集的一部分信息来进行调整，使其更加适应学术文章的总结任务。

对于微调的损失函数，我有两种考量，分别是直接使用其原始的损失函数（也就是交叉熵损失），或者为了任务的目标，使用 ROUGE-L 作为损失函数。后者需要先计算出 ROUGE-L，并且运用策略梯度方法，利用强化学习损失进行训练。

3.1 直接对交叉熵损失进行优化

直接利用交叉熵损失进行优化，其训练循环本身比较简单。但如何将文章压缩为 1024 token 是需要考虑的。我们的思路有两种：一个是提取文章开头部分再加上正文每一段首句，直至达到 1024 token 的上限。另一种即为多层总结，先将文章（有重叠地）分为长度为 1024 token 的若干段，对于每一个文段进行总结，随后将总结进行拼接，再度进行一次总结，生成摘要。前一个方法的优势是非常简单，便于进行微调，但很显然这样做会有一定概率丢失关键信息和各类细节。后者可以层次化地提取文章的每一层信息，感受野覆盖全文章。但是其问题在于，损失函数定义复杂，多层总结难以进行梯度回传。另外第一层总结生成多少 tokens，甚至是否确定只需要两层总结，都难以确定。为了简化微调的流程，我直接采用第一个方法进行了微调。

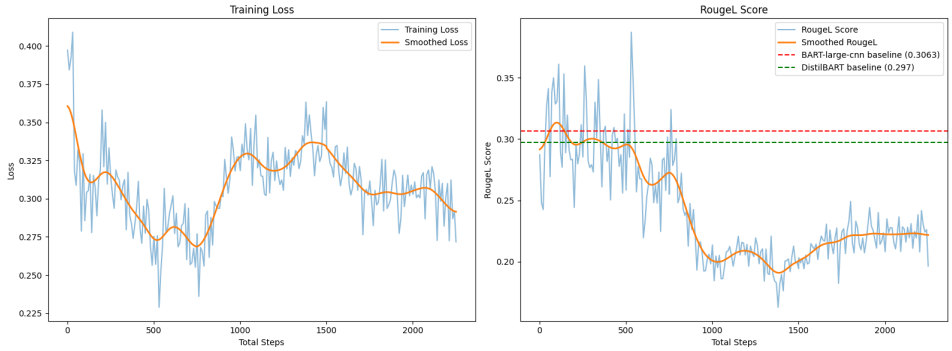

Figure 2: Loss curve and ROUGE-L score of fine tune process

总共进行了 3 个 EPOCH，每个 EPOCH 有 750 步。仍在前述配置的 AutoDL 服务器上运行，代码未增加并行逻辑，微调一共花费十余分钟。结果如上图，可见这一方法非常不理想。甚至 train loss 都没有稳定下降。ROUGE-L 分数从初始较为理想的 0.3 附近，甚至下降到 0.2 左右。但这一结果实际上在预料之中。首先交叉熵损失是针对 next token prediction 进行，对于我们的文章摘要生成任务，选择交叉熵损失进行优化显然是不合适的。对已经蒸馏过的模型来改进交叉熵损失反而会破坏其原有的结构。另外我们的 learning rate 可能也设置的较大。

3.2 对 ROUGE-L 进行优化

直接对 ROUGE-L 进行优化，对于我们的任务更具有针对性。我的思路是先用 beam search 进行生成，随后计算 ROUGE-L 作为奖励。损失函数的数学形式为：

$$\mathcal{L} = -\mathbb{E}[\log P(y|x) \cdot R(y)]$$

其中：$P(y|x)$ 是模型生成摘要 y 的概率，$R(y)$ 是 ROUGE-L 分数作为奖励信号。

由于涉及模型的生成，以及 REINFORCE 方法本身速度较为缓慢，我再次设计了并行的算法。然而，因为算力资源的限制，我的 vGPU-32GB * 8 卡实际上不足以支持较大的 batch size，并行效果不佳，甚至难以进行训练。因此只进行了较少的步数进行了微调。此次过程中，loss 函数可以稳定地下降，ROUGE-L 略微存在上升趋势。但因为步数较少，且原模型本身效果较好，上升并不明显。

4 微调生成结果对比调用 DeepSeek V2 结果

通过上面的训练和微调，我们得到了两个可以生成摘要的模型，分别是我重新训练并微调的 GPT-2 124M 模型，以及我用 REINFORCE 方法微调的 distilbart-6-6-cnn 模型。

我挑选了论文《Language Models are Unsupervised Multitask Learners》，对其正文进行摘要的生成。然而，生成的方法却并不是微调时的选取开头加正文每一段首句。我采取了分层的逐步总结。对于每一部分，分别进行生成，然后将每一段的总结结果拼接在一起，进行进一步总结。Table 2中选取了两个模型较好的生成结果，与调用 DeepSeek V2 生成的摘要进行了对比。

Table 2: Generated Abstract Comparison

Model	**Output Abstract**	**Avg Time (s)**
GPT-2 124M	The paper by Alec Radford et al. is the first to provide an overview of the new model architecture and highlight some of its key contributions towards human language comprehension, including its design and deployment for a wide range of unsupervised NLP tasks. The paper also describes the current status of GPT-2, a 2.5 billion parameter Transformer model that achieves state-of-the-art results in zero-shot settings across 9 out of 15 tested translation benchmarks, demonstrating outstanding capabilities for tasks ranging from reading comprehension to machine translation. The paper presents a new and widely tested language model architecture (GPT-3) that provides an unprecedented capacity to teach highly capable language comprehension tasks in high-quality experiments. GPT-3 is also well-tested in a corpus of non-native English language models and provides excellent evidence of its robustness to unsupervised NLP tasks with good throughput. The work demonstrates the potential of unsupervised NLP, which can support the development	2.47
distilbart-6-6-cnn	The paper presents GPT-2, a 1.5B parameter Transformer language model trained on WebText . The model uses byte-level BPE tokenization and modifies the standard Transformer architecture with layer normalization adjustments . The paper discusses implications for building general language systems that learn from diverse data without explicit supervision .	1.26
Deepseek V2	The paper introduces GPT-2, a 1.5B parameter Transformer model trained on WebText (8M high-quality webpages), demonstrating that large language models can perform diverse NLP tasks zero-shot without task-specific training. GPT-2 achieves state-of-the-art results on 7/8 language modeling benchmarks and shows promising performance on reading comprehension (55 F1 on CoQA), translation (5-11.5 BLEU), and question answering. The model uses byte-level BPE tokenization and modified Transformer architecture, with performance scaling log-linearly with model size. Key findings include GPT-2's 70.7% accuracy on Winograd Schema Challenge and 93.3% on Children's Book Test nouns, suggesting language models may learn tasks from diverse text data without explicit supervision.	7.98

可以发现，GPT-2 124M 和 distilbart-6-6-cnn 生成的速度都较快（皆利用本地的 NVIDIA GeForce RTX 4060）。而调用 Deepseek V2 API 的生成速度缓慢。生成质量上，Deepseek V2 生成的质量最高，介绍了模型架构和模型的成果，同时介绍了模型的价值。微调的 distilbart-6-6-cnn 模型因为其原本模型的输出长度限制，输出长度短，但是事实性内容都做到了正确无误。虽然生成的摘要较短，但是精确涵盖了参数量，数据集，架构，无监督训练，意义等全部信息。可以说质量令人满意。GPT-2 124M 模型则难以令人满意，其生成的长串文字，仅仅做到了大部分语法正确。但是参数量，测试结果等信息出现事实性错误，介绍架构未提到关键的 Layer-Norm，后面更是偏离到了 GPT-3 模型，可以看出仍然有着较强的 next token prediction 任务的痕迹，对前文内容的捕捉提取不够，微调结果不理想。

另外值得一提的是，调用 Deepseek V2 不仅速度显著更慢，而且实际上难以控制生成的长度。无论如何设置 prompt（例如设为"...200 个英文单词左右"），其生成的摘要长度往往和命令不完全一致。而本地的模型通过调整参数等方法，有希望精确地控制长度。

5 结论

虽然我们选取的都是多次尝试中生成的比较满意的结果，但是通过微调 distilbart-6-6-cnn 生成的内容，我们可以看到，仅仅 230M 的模型对于论文摘要生成任务就有很高的潜力。通过蒸馏、微调，我们可以得到一个完成摘要生成任务的水平足以接近当今前沿的大语言模型（GPT-4, Deepseek V2, Gemini-3.5 等）的 100M 参数量级别的模型。虽然参数量更小的 GPT-2 124M 模型表现不佳，但是我们仍然可以认为这一级别参数量的模型在完成该任务上具有强大潜力。

100M 参数量级别的模型表达能力可以一定程度地完成摘要生成任务，一方面为这一任务在边缘端部署、以及高效生成提供了实践支撑。我们还可以通过量化（quantization）等手段进一步压缩参数量进行测试，探索摘要生成任务的参数量下限。另一方面，也为类似的微调任务，包括翻译、阅读理解、回答问题等提供了本地部署轻量级模型的可能。虽然 GPT-2 论文中提到，对于参数量小的模型，相比于完成总结任务，在完成翻译等任务时，距离参数量大的模型差距更大。但轻量级部署，低成本高效率的推理相比于成本高、算力需求大的大型模型的巨大优势，让人们有动机进一步探索轻量级语言模型的上限。

我们的探索还存在很多需要改进的地方。受限于算力，我无法对 GPT-2 124M 模型进行进一步的探索，无法确定其是否已经达到了这一架构的上限。另一方面，对于 distilbart-6-6-cnn 的微调步数也不充足，可能还未充分发挥出该模型的潜能。另外生成长度也没能精确控制到希望的水平，这些都是后续可以进一步探究和改进的地方。

References

参考了 OpenAI 的 GPT2 和 GPT3 的论文,Andrej Karpathy 复现 GPT2 的尝试,以及 Huggingface 上提到的模型和数据集。

[1] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.

[2] Brown, T. B., et al. 2020. Language models are few-shot learners. In Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS '20). Curran Associates Inc., Red Hook, NY, USA, Article 159, 1877–1901.

[3] Andrej Karpathy, `https://github.com/karpathy/build-nanogpt`

[4] `https://github.com/openai/gpt-2`

[5] `https://github.com/huggingface/transformers`

[6] `https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu`

[7] `https://huggingface.co/sshleifer/distilbart-cnn-6-6`

[8] `https://huggingface.co/datasets/armanc/scientific_papers`

A Appendix / supplemental material

文中用到的全部代码可以参见 Github repo：`https://github.com/RinShiina/gpt2-article-abstract`

我训练好的 GPT-2 124M 模型可以参见 Huggingface：`https://huggingface.co/RinShiina/gpt2-for-abstract`

微调模型版本过多，效果差异很大，因此不上传至 Huggingface。