华中农业大学
HUAZHONG AGRICULTURAL UNIVERSITY

基 于
Transfer Text-to-Text Transformer（T5）
论 文 标 题 生 成 模 型

COURSE REPORT

姓名 周杰、高星杰

学号 （周）、2021307220712（高）

专业 计算机科学与技术

教师 李万理

科目 自然语言处理

信 息 学 院

COLLEGE OF INFORMATIC

2024 年 1 月

摘要

该 NLP 大作业由周杰与高星杰共同实验完成。项目中实施了一个基于 T5 (Text-to-Text Transfer Transformer) 模型的自然语言处理框架，可以根据论文的摘要生成合适的标题。项目的核心包括四个阶段：数据预处理、模型训练、性能评估和标题生成。初始阶段涉及利用预训练的 T5 tokenizer 对摘要和标题数据进行标准化处理，确保其适应模型结构。通过设定恰当的训练参数，如批次大小和学习率，利用 PyTorch 对 T5 模型进行训练，并借助 Weights & Biases 进行训练过程的实时监控。模型训练过程中，采用 ROUGE 分数作为主要评估指标，在独立的测试集上评估模型性能，以验证其在未知数据上的泛化能力。

目录

1 算法介绍

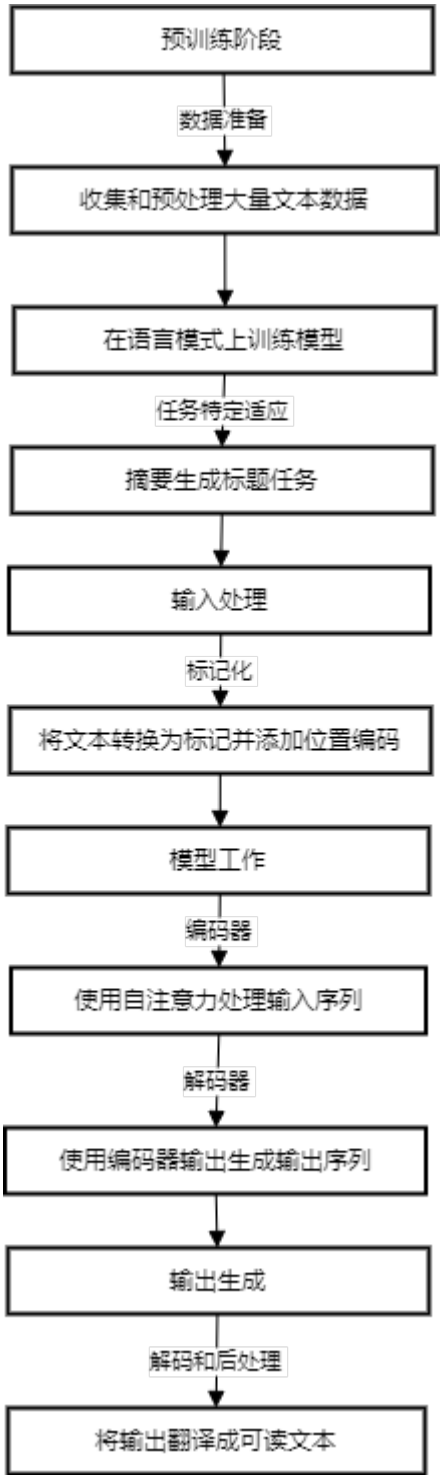
1.1Transformer

Transformer 是一种在自然语言处理（NLP）领域广泛应用的深度学习模型架构。它的主要特点是使用了自注意力（Self-Attention）机制，这使得 Transformer 能够在处理序列数据时，有效地捕捉长距离依赖关系。

- **自注意力：**自注意力是 Transformer 的核心，它允许模型在处理序列的每个元素（如单词）时，同时考虑序列中的所有其他元素。通过这种方式，模型能够捕捉词与词之间的关系，无论它们在序列中的距离有多远。
- **多头注意力：**Transformer 使用多头注意力来并行地执行多个自注意力操作。这允许模型在不同的表示子空间中捕捉到不同的特征。每个“头”都可能关注序列中的不同部分，从而提高模型的表达能力。
- **位置编码：**由于 Transformer 没有递归和卷积结构，它无法捕捉序列中元素的位置信息。为了解决这个问题，Transformer 添加了位置编码到输入嵌入中。位置编码是一种特殊的编码方式，它为序列中的每个元素提供其位置信息。
- **编码器-解码器架构：**编码器用于处理输入序列，并输出一个上下文表示，捕捉序列中的信息。解码器基于编码器的输出和先前解码的输出，逐步生成目标序列。
- **前馈网络：**每个编码器和解码器层还包含一个前馈网络，它对每个位置进行相同的操作，但对不同位置是独立的。这提供了额外的处理能力。

1.2Text-to-Text Transfer Transformer

T5（Text-to-Text Transfer Transformer）模型是一种基于 Transformer 架构的自然语言处理模型。T5 将所有 NLP 任务视为“文本到文本”的问题，即将输入文本转换成输出文本，本项目中，其工作流程如下：



1.3 Tokenizer

Tokenizer 的工作原理是将文本转换为模型可以理解的数字序列。在复杂的 NLP 任务中，需要能够有效处理各种文本数据，包括罕见词汇和未知词汇，同时保持序列的一致性和可处理性。Tokenizer 通过分割文本、数字化等操作有效地处理了上述问题。

- **文本分割（Tokenization）：**文本被分割成更小的单元，称为 tokens。这些 tokens 可以是单词、子词或字符。Tokenization 可以使用最简单的基于规则的分割，如使用空格和标点符号分割英文；也可以使用在高级算法 BPE (Byte Pair Encoding)与 WordPiece 来处理更加复杂的数据。
- **数字化（Numericalization）：**每个 token 被映射到一个唯一的数字 ID，这些 ID 是根据一个预先定义的词汇表确定的。使用查找表将 tokens 转换为数字 ID。
- **特殊 Tokens 添加：**为了适应特定模型的需求，可能需要向序列中添加特殊 tokens，如 `[CLS]`, `[SEP]`, `[PAD]`。在序列的适当位置插入这些特殊 tokens。
- **填充与截断（Padding & Truncation）：**为了让所有序列长度一致，较短的序列被填充，较长的序列被截断。添加 `[PAD]` tokens 到较短的序列，或者截断超出指定长度的序列。

Tokenization 算法介绍：

- **Byte Pair Encoding (BPE)：**BPE 通过迭代地将最常见的字符对替换为单个未使用的字符，减少了词汇表的总体大小，同时有效地处理未知或罕见的词汇。BPE 算法需要统计数据集中所有字符对的频率，迭代地将最常见的字符对合并为新的符号，并更新词汇表。最后重复此过程直到达到设定的符号数量。
- **WordPiece：**类似于 BPE，但在决定合并哪些字符对时，考虑了合并后的新符号在整个数据集中的概率。WordPiece 使用贪心算法来优化每一步的合并，以最大化整体数据集的似然度。最终生成一个可以有效表示训练数据集中大部分词汇的子词集合。

1.4注意力机制

注意力机制的原理可以从几个核心概念来理解：查询（Query）、键（Key）、值（Value），以及如何通过这些组件来计算注意力得分和加权输出。下面是这个机制的基本步骤：

- **查询、键和值：**在注意力机制中，我们通常有一组查询，以及一组键-值对。这些概念可以类比于搜索引擎。想象你用一个查询词去搜索，

搜索引擎（注意力机制）会查找匹配的键，然后返回相关的信息。

- **计算注意力得分：**模型会计算每个查询与所有键之间的相似度或匹配程度。这通常通过点积（即两个向量的点乘）或其他相似性度量（如余弦相似性）来完成。得分越高，表示查询和特定键的匹配程度越高。
- **归一化：**为了使得分可比较并且数值稳定，通常会对它们进行归一化处理。最常用的方法是使用 Softmax 函数，它可以将得分转换为概率分布。
- **计算加权的值：**然后，每个值（Value）会根据相应的得分进行加权。这意味着与查询更相关的键所对应的值会获得更高的权重。最后，所有这些加权的值会被累加起来，形成最终的输出。这个输出是对给定查询的“注意力聚焦”的结果。在 Transformer 等先进模型中，这个过程会在所谓的“多头”注意力机制中并行进行。这意味着模型会同时执行多个注意力操作，每个操作关注输入的不同部分。这有助于模型更全面地理解输入数据。
- **自注意力机制（Self-Attention Mechanism）**是一种特殊的注意力机制，用于在序列数据（如文本或时间序列）中捕捉元素之间的关系和依赖关系。自注意力机制不同于传统的注意力机制，它允许一个序列中的元素与该序列中的其他元素进行交互，从而更好地理解上下文信息。自注意力机制的关键优势在于它的能力，**它可以捕获序列内任何两个元素之间的关系，而不仅仅是固定距离的依赖关系**。这使得模型能够更好地处理长距离依赖，从而提高了在处理序列数据时的性能，特别是在自然语言处理任务中。

1.5自注意力机制在 Transformer 中的作用

自注意力机制（Self-Attention Mechanism）在 Transformer 模型中主要用于以下几个方面：

- **建立全局依赖关系：**自注意力机制允许 Transformer 模型在处理输入序列时同时关注序列中的所有元素，而不仅仅是固定距离的依赖关系。这有助于模型捕捉长距离的依赖关系，使其能够更好地理解文本中的上下文信息。

- **上下文感知：**自注意力机制使得模型能够在处理每个元素时动态地关注其他元素的上下文。这意味着模型可以根据输入序列中其他元素的重要性来加权处理当前元素，从而更好地理解每个元素的语义和关系。
- **并行化：**自注意力机制可以非常高效地进行并行计算，因为每个元素的注意力权重独立计算。这使得 Transformer 能够更好地利用硬件加速，加快训练和推理速度。
- **序列建模：**自注意力机制允许 Transformer 模型将输入序列映射到高维表示空间，这些表示可以更好地捕捉序列中的各种模式和信息。这使得 Transformer 成为了强大的序列建模工具，适用于多种自然语言处理任务，如机器翻译、文本生成和文本分类。

1.6Punkt 分词器

项目使用了 NLTK 中的 Punkt 分词器。Punkt 是一个基于无监督学习算法的句子分割算法，主要用于将文本自动分割成单独的句子，特别适用于处理那些不仅仅依靠简单标点符号分隔的复杂文本。

Punkt 分词器基于无监督机器学习技术，使用了一组预先训练的模型，这些模型被训练来识别句子的边界。这种训练是通过分析大量的文本并学习词汇的用法来完成的，特别是关注句号、问号和感叹号等潜在的句子结束符号的使用。

1.7ROUGE 分数

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) 分数是一种广泛使用的评价指标，用于评估自动文本摘要或机器翻译等自然语言生成任务的质量。ROUGE 主要关注生成文本与参考文本（通常是人类编写的摘要或翻译）之间的重叠情况。这个指标特别强调召回率，即模型生成的文本覆盖参考文本内容的程度。本项目使用了 ROUGE-1/2、ROUGE-L 的 F1 分数=2 × (召回率 × 准确率) / (召回率 + 准确率)来对模型性能进行评估。

- **ROUGE-N**：计算 n-gram（连续的 n 个词）的重叠度。

  公式：

召回率 = (参考摘要中与生成摘要重叠的 n-grams 数量) / (参考摘要中的 n-grams 总数量)

准确率 = (参考摘要中与生成摘要重叠的 n-grams 数量) / (生成摘要中的 n-grams 总数量)

- **ROUGE-L**:基于最长公共子序列（LCS），适合评估句子级别的结构相似性。

  公式：

  召回率 = (LCS 的长度) / (参考摘要的长度)

  准确率 = (LCS 的长度) / (生成摘要的长度)

例如，对于 ROUGE-1（单词级别的重叠），如果生成的文本是 "the cat sat on the mat"，参考文本是 "the cat is on the mat"，则重叠的单词有 "the", "cat", "on", "the", "mat"（共 5 个），如果参考文本总共有 6 个单词，生成文本有 6 个单词，则召回率是 5/6，准确率是 5/6。

2 运行结果

项目使用 wandb（Weights & Biases）对模型训练进行实时跟踪。记录模型训练过程中的各种指标，例如损失、准确率、学习率等。模型训练批次大小为 2；迭代轮数为 5。模型每训练 1000 步，针对测试集进行一次评估计算。在训练结束后，以最终模型进行一次评估。训练时长 13 小时。

2.1 模型最终参数(针对于测试集)

```
{
  "train/train_runtime": 46844.723,训练时间
  "train/global_step": 90185,训练步数
  "train/epoch": 5,迭代次数
  "train/train_loss": 1.508533973564837,训练集loss
  "train/learning_rate": 5.27803958529689e-8,学习率

  "eval/loss": 1.6459659337997437,    测试集loss
  "eval/rouge1": 48.0524,
  "eval/rouge2": 27.546,
  "eval/rougeL": 42.7405,
  "eval/rougeLsum": 42.7506,
}
```

ROUGE分数

2.2生成标题与原论文标题对比

ConvTextTM: An Explainable Convolutional Tsetlin Machine Framework for Text Classification

Bimal Bhattarai, Ole-Christoffer Granmo, Lei Jiao
University of Agder
Norway
{bimal.bhattarai, ole.granmo, lei.jiao}@uia.no

Abstract

Recent advancements in natural language processing (NLP) have reshaped the industry, with powerful language models such as GPT-3 achieving superhuman performance on various tasks. However, the increasing complexity of such models turns them into "black boxes", creating uncertainty about their internal operation and decision-making. Tsetlin Machine (TM) employs human-interpretable conjunctive clauses in propositional logic to solve complex pattern recognition problems and has demonstrated competitive performance in various NLP tasks. In this paper, we propose ConvTextTM, a novel convolutional TM architecture for text classification. While legacy TM solutions treat the whole text as a corpus-specific set-of-words (SOW), ConvTextTM breaks down the text into a sequence of text fragments. The convolution over the text fragments opens up for local position-aware analysis. Further, ConvTextTM eliminates the dependency on a corpus-specific vocabulary. Instead, it employs a generic SOW formed by the tokenization scheme of the Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019a). The convolution binds together the tokens, allowing ConvTextTM to address the out-of-vocabulary problem as well as spelling errors. We investigate the local explainability of our proposed method using clause-based features. Extensive experiments are conducted on seven datasets, to demonstrate that the accuracy of ConvTextTM is either superior or comparable to state-of-the-art baselines.

图例 1 测试论文 1

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

图例 2 测试论文 2

A. train_step=10000 的生成效果

```
Predict title : ConvTextTM: A Convolutional Tsetlin Machine for Text Classification
Original title : ConvTextTM: An Explainable Convolutional Tsetlin Machine Framework for Text Classification
```

图例 3 train_step=10000，测试论文 1

```
Predict title : Transformer: A Simple Network Architecture for Sequence Transduction
Original title : Attention Is All You Need
```

图例 4 train_step=10000，测试论文 2

B. train_step=36000 的生成效果

```
Predict title : ConvTextTM: A Convolutional Tsetlin Machine Architecture for Text Classification
Original title : ConvTextTM: An Explainable Convolutional Tsetlin Machine Framework for Text Classification
```

图例 5 train_step=36000，测试论文 1

```
Predict title : Transformer: A Simple Neural Network Architecture for Sequence Transduction
Original title : Attention Is All You Need
```

图例 6 train_step=36000，测试论文 2

C. train_step=90000 的生成效果

```
Predict title : ConvTextTM: A Convolutional Tsetlin Machine Architecture for Text Classification
Original title : ConvTextTM: An Explainable Convolutional Tsetlin Machine Framework for Text Classification
```

图例 7 train_step=90000，测试论文 1

```
Predict title : Transformer: A Simple Neural Network Architecture for Sequence Transduction
Original title : Attention Is All You Need
```

图例 8 train_step=90000，测试论文 2

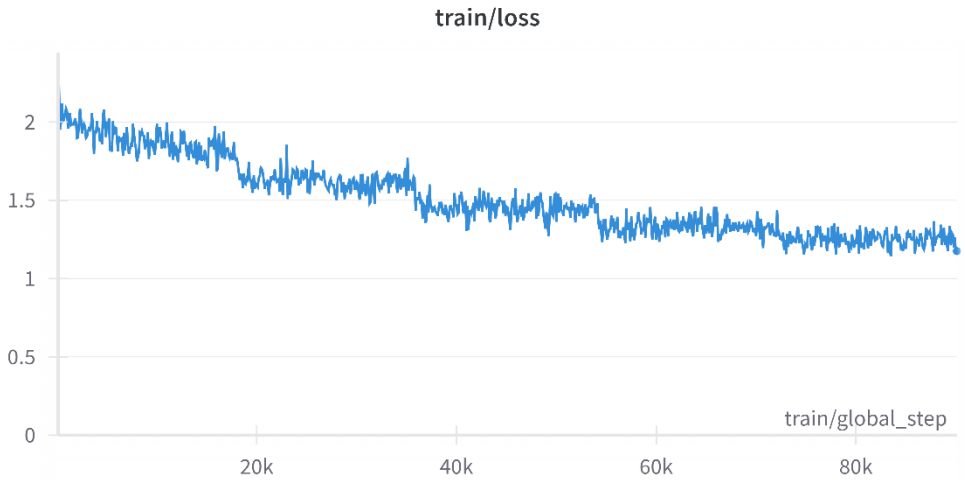
D. 测试效果总结

综上所述，对于常规的论文标题而言，训练模型在训练步数达到 10000 时，已经具有良好的摘要总结标题的能力。训练步数达到 36000 时，模型的总结能力更加完善，可以总结出更多与原文标题相符的单词。但对于非常规的论文标题（测试论文 2，transformer 的原始论文）而言，模型无法给出类似于原始标题的文本。因为其具有更复杂的语言表达模式，难以通过 T5 模型完成。这种非常规的论文标题对于模型的总结能力而言，可以说是数据中的噪点，但对于的模型的智能化与情感化也可以是新的出发点。我们即可以思考出新的改进方式，例如数据集中的标题不一定为原始论文标题，可以是人工总结之后的标题，以减少非常规标题带来的噪点；或者在使用数据集时，对数据集进行预处理，将非常规论文标题筛去。当然，人工预处理会导致模型无法实现端到端的工作模式，还需要更进一步的讨论。

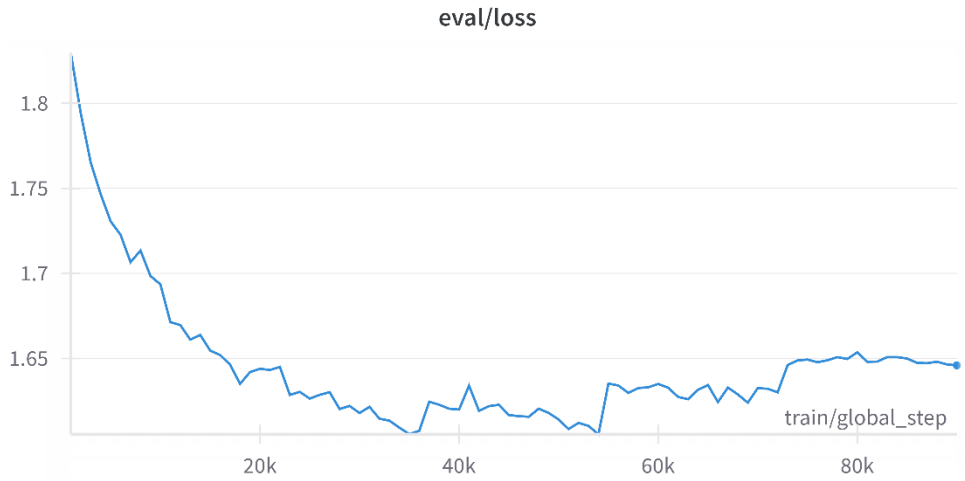
根据 WandB 中的跟踪的 ROUGE 分数（2.3 中 C-F 图）可以发现，训练步数在 30000 到 55000 时，模型 ROUGE 分数较高，但方差较大。训练步数在 60000 到 90000 之间的方差较小，这说明训练后期模型已经趋近收敛，模型性能平稳波动。

2.3 训练过程中 WandB 监控数据

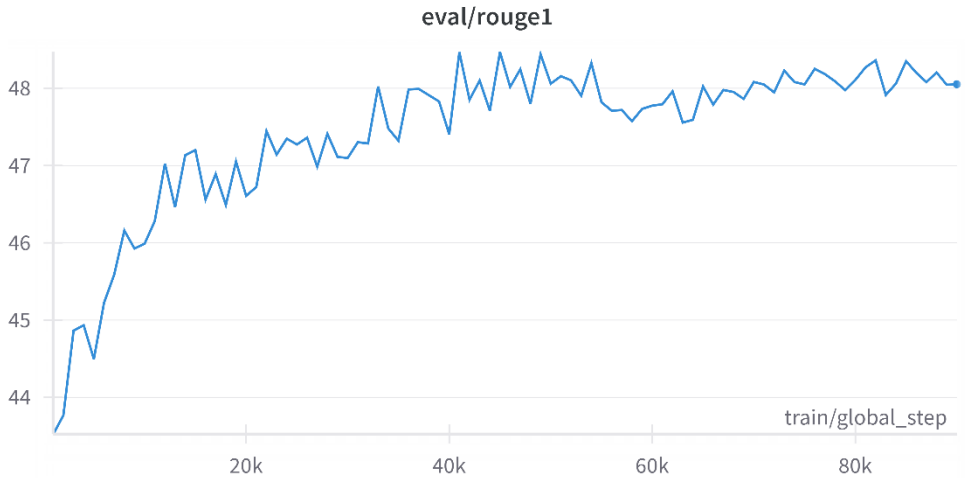
A. Train/loss



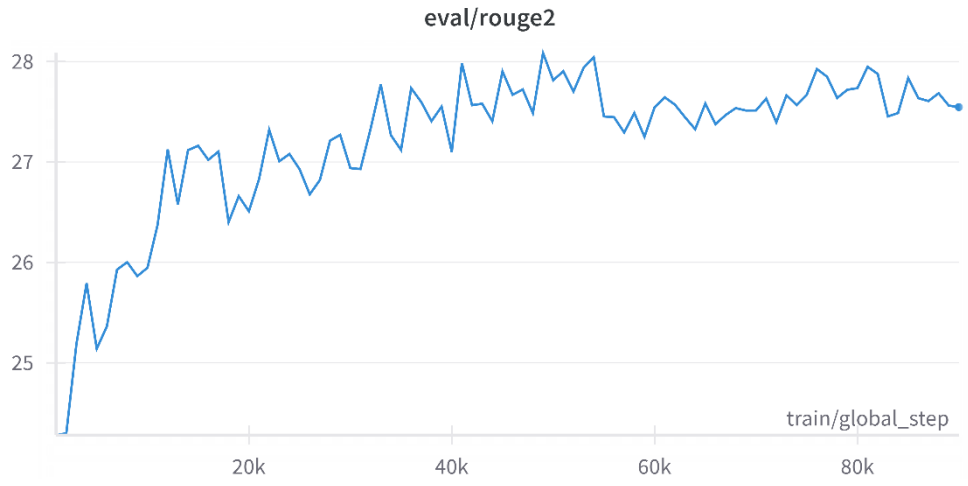
B. Evaluate/loss



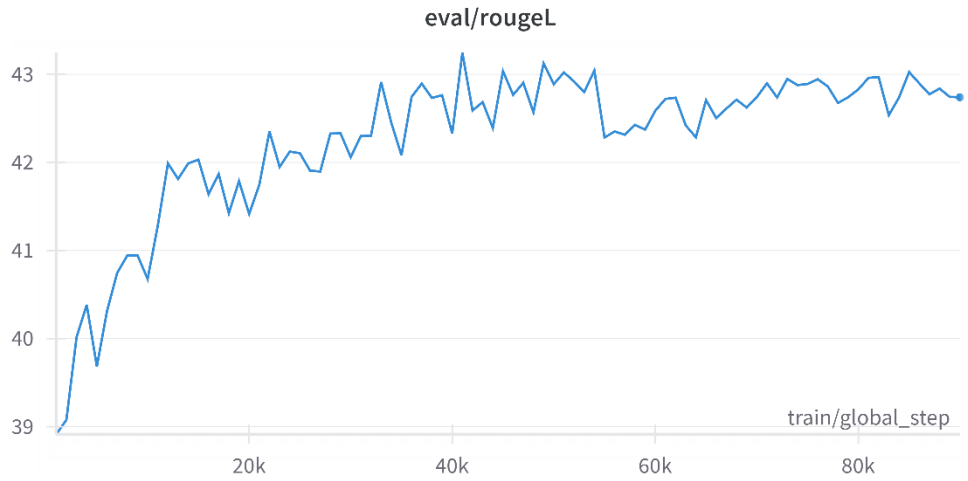
C. Evaluate/ROUGE-1



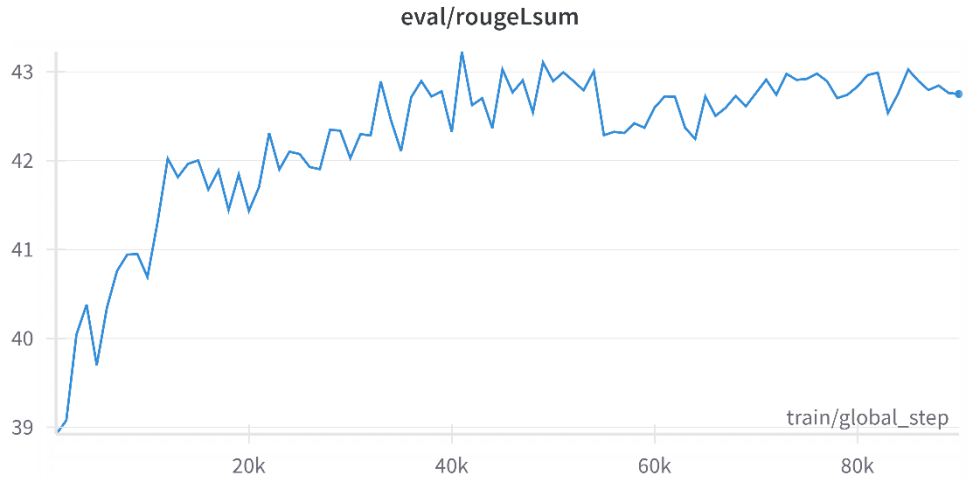
D. Evaluate/ROUGE-2



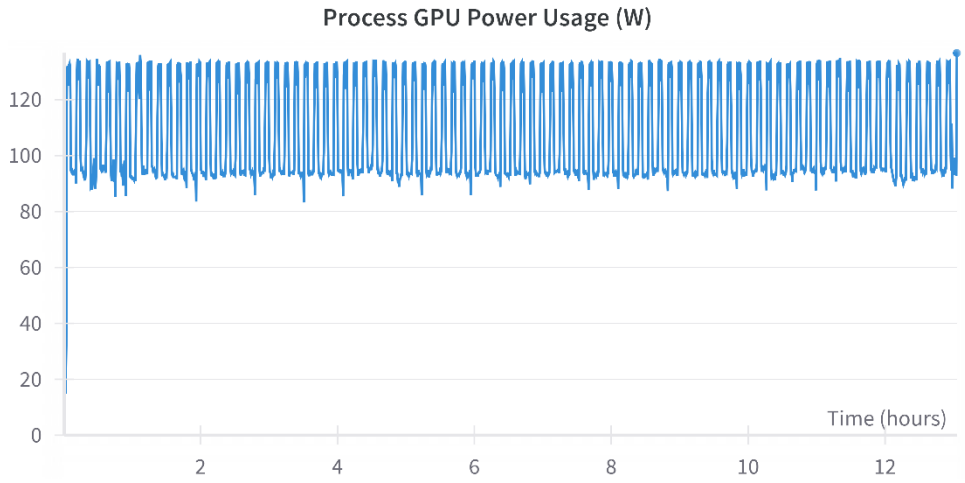
E. Evaluate/ROUGE-L



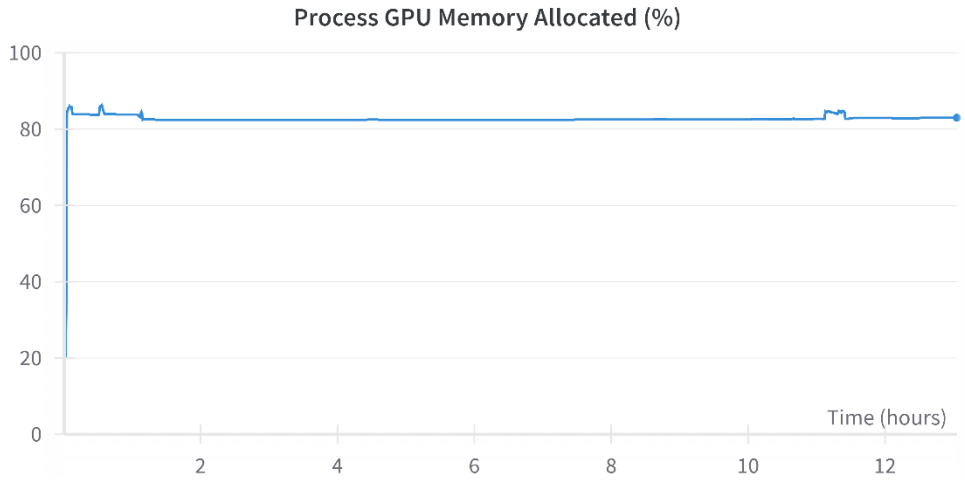
F. Evaluate/ROUGE-L sum



G. GPU 利用率



H. 显存利用率



3 数据集

3.1 数据集介绍

想要有一个好的模型，那么就一定要有一个好的数据集，我们在本次实验中使用的是 arXiv 数据集，这个 arXiv 数据集是康奈尔大学提供的开放获取学术文章数据集。主要有以下几个特征:

- **数据量大**:包含超过 100 万篇学术论文信息
- **跨学科**:论文涵盖计算机科学、数学、物理学、统计学、经济学等多个学科
- **丰富元数据**:每篇论文都有标题、作者、摘要、评论、类别等详细元信息

- **时间跨度长**:论文发表时间跨度从 1991 年到 2015 年
- **原始文本**:提供了超过 100 万篇论文的原始 PDF 文件
- **便于处理**:元数据和全文都以结构化的 JSON 格式组织
- **开源可用**:数据集可在 Kaggle 开放下载使用,适合学术/商业研究和文本分析/数据分析项目

3.2数据集的使用

arXiv 数据集中输入样本的特征(X)就是论文的摘要文本(abstract)。标签(Y)就是论文对应的标题文本(title)。也就是说:

```
X = 论文摘要(abstract), shape = (num_samples, abstract_text_length)
Y = 论文标题(title), shape = (num_samples, title_text_length)
```

这里 X 就是模型的输入特征,代表输入序列;Y 就是标签,代表了希望模型生成的目标输出序列。生成任务的目标就是给定一段新的论文摘要文本 X_new,模型可以生成预测的论文标题 Y_pred,使其尽可能接近真实的标题 Y_true。训练过程中使用论文对(X, Y)作为监督信号进行学习; 测试/预测阶段只输入 X 来生成 Y。这样,arXiv 数据集为构建标题生成任务提供了海量高质量的(X, Y)样本来训练序列到序列(seq2seq)模型。

4 运行环境

4.1训练设备

显卡：Geforce RTX3070 Laptop

CPU：AMD Ryzen 7 5800H

Cuda 版本：11.8

4.2 Python 环境

Python3.8.18

Pytorch==2.1.1

Transformer==4.37.1

huggingface-hub==0.20.3

wandb==0.12.6

datasets==2.3.2

tokenizers==0.15.1

sentencepiece==0.1.96

rouge-score==0.0.4

4.3 WandB

运行本项目并使用 WandB 进行监控，需注册 WandB 账号，并创建一个 Project 项目。在 wandb.init()中，修改 project 为项目名，entity 为用户名。

5 参考文献

无

6 总结

6.1 大作业实践总结

“学的越多不懂的越多”，这是我们本次实验的感触，我们在本次实验中第一次自己动手尝试了生成式 NLP 模型，并且学到的了很多知识，但是又发现了许多自己所欠缺的。在本次实验中，我们研究了生成式模型及在自然语言处理任务中的应用，特别是在使用 arXiv 数据集进行标题生成任务方面。Transformer 提供了强大、完备的功能，但实验过程中也遇到了一些挑战。特别是在处理大规模数据集时，我们面临着 GPU 显存溢出和计算效率的问题。此外，模型调优也是一个挑战，需要精心选择合适的超参数以达到最优性能。尽管如此，通过使用 Wandb 进行实时跟踪和评估，我们能够及时监测模型性能并进行迭代改进。这次实验不仅增强了我们对 Transformer 模型及其组件的理解，也为解决实际问题提供了宝贵的经验。

6.2 课程总结

在这在这学期的 NLP 课程中，我们首次深入挖掘了自然语言处理的奥秘。这不仅是对理论的深入学习，更是一次实践的深刻洞察。

NLP 课程内容丰富全面，涵盖了自然语言处理的多个重要主题，如文本预处理、马尔可夫模型、语言模型、序列建模，以及最新的深度学习技术，如 Transformer 和 BERT。每个部分都细致入微，理论与实践相结合，

使复杂的概念变得易懂、可用。

课程的前半部分打下了坚实的基础。从文本清洗、分词、词性标注开始，我们学习了如何使用 n-gram 模型和 TF-IDF 等技术提取特征，为理解更高级的 NLP 技术奠定了基础。随着课程逐渐深入，引入了神经网络在 NLP 中的应用。我们深入探讨了 RNN、LSTM 和 GRU 等序列模型，学习了它们在处理时间序列数据、情感分析、文本生成等任务中的应用。尤其是 Transformer 模型的讲解，不仅让我们理解了自注意力机制和位置编码，还体验了其强大性能和广泛应用。最后，课程专注于最新研究和前沿技术。我们学习了如何利用 BERT 和 GPT 这样的预训练模型进行文本分类、命名实体识别、问答系统构建等高级任务。这些内容展现了 NLP 领域的最新进展，为我们的未来学习和研究指明了方向。

在实践项目中，我们学会了构建、训练 NLP 模型，处理和解析数据，调整模型以适应特定任务，评估模型性能。这些实战经验对未来的学习和工作都至关重要。这门课程让我们明白，自然语言处理不只是技术和算法的堆砌，它更是对语言、文化和人类交流方式的深入理解。每个模型和算法背后，都是对语言本质的洞察。我们认识到，将理论知识应用于实际问题是检验学习成果的关键！