南京大学

本科毕业设计

院　　系　软件学院

专　　业　软件工程

题　　目　**基于 ChatGLM 的文言文翻译模型的研究**

年　　级　2020　学　　号　201250171

学生姓名　汪博文

指导教师　葛季栋　职　　称　副教授

提交日期　2024 年 6 月 2 日

南京大学本科毕业论文（设计）
诚信承诺书

本人郑重承诺：所呈交的毕业论文（设计）（题目：基于ChatGLM 的文言文翻译模型的研究）是在指导教师的指导下严格按照学校和院系有关规定由本人独立完成的。本毕业论文（设计）中引用他人观点及参考资源的内容均已标注引用，如出现侵犯他人知识产权的行为，由本人承担相应法律责任。本人承诺不存在抄袭、伪造、篡改、代写、买卖毕业论文（设计）等违纪行为。

作者签名：

学号：

日期：

南京大学本科生毕业论文（设计、作品）中文摘要

题目：基于 ChatGLM 的文言文翻译模型的研究

院系：软件学院

专业：软件工程

本科生姓名：汪博文

指导教师（姓名、职称）：葛季栋　副教授

摘要：

文言文作为中国传统文化的重要组成部分，把文言文保护好、传承好、发展好是传承中华文化的不可或缺的一部分。但由于文言文本身具有的复杂语法、与现代汉语的巨大差异，大大提高了文言文阅读的门槛、阻塞了文言文推广的进度，因此，充分发挥大语言模型在文本翻译领域的优势、构建文言文-现代文翻译模型、降低文言文阅读门槛是传承中华文化、加强中华民族凝聚力、实现中华文化现代化的重要关节。

本研究基于 ChatGLM3-6B 模型和 Erya 数据集，旨在设计和训练一种文言文到现代文的大语言翻译模型。通过对 ChatGLM3-6B 模型进行微调，并结合 Erya 数据集中的文言文和现代文平行语料，致力于提高模型在古代文献翻译任务中的性能。文章首先分析了当下主流开源预训练模型的特点和优劣，然后探究了微调方法的选择、评价指标的应用，并在训练集上进行了模型性能的比较分析。实验结果表明，我们设计的翻译模型在文言文到现代文的翻译任务中取得了良好的性能，相比百度翻译，在 Rouge 分数上取得了平均 +5Rouge 的优势；与【随无涯】翻译模型相比，取得了 +7Bleu、平均 +10Rouge 的进步；与 mengzi-t5-base 模型相比，在 Bleu-4 和 Rouge-2 上有 +3Bleu、+8Rouge-2 的改进，为古代文献研究和相关应用提供了有力支持。

关键词：大语言模型；文言文-现代文翻译；ChatGLM3-6B 模型

南京大学本科生毕业论文（设计、作品）英文摘要

THESIS: Research on a Classical Chinese Translation Model Based on ChatGLM

DEPARTMENT: Software Institute

SPECIALIZATION: Software Engineering

UNDERGRADUATE: Bowen Wang

MENTOR: Associate Professor Jidong GE

ABSTRACT:

As an important part of traditional Chinese culture, the protection, inheritance, and development of Classical Chinese are indispensable for the inheritance of Chinese culture. However, due to the complex grammar and significant differences from modern Chinese, Classical Chinese greatly increases the threshold for reading and hinders the promotion of Classical Chinese. Therefore, fully leveraging the advantages of large language models in the field of text translation, constructing Classical Chinese to Modern Chinese translation models, and reducing the threshold for Classical Chinese reading are crucial for the inheritance and modernization of Chinese culture, as well as strengthening the cohesion of the Chinese nation.

This study aims to design and train a Large Language Model for translating Classical Chinese to Modern Chinese based on the ChatGLM3-6B model and the Erya dataset. By fine-tuning the ChatGLM3-6B model and combining Classical Chinese and Modern Chinese parallel corpora from the Erya dataset, we strive to improve the performance of the model in translating ancient texts. In this thesis, we first analyze the characteristics and advantages and disadvantages of mainstream open-source pre-trained models, then explore the selection of fine-tuning methods and the application of evaluation metrics, and finally compare and analyze the performance of the model on the training set. The experimental results demonstrate that our designed translation model performs well in translating Classical Chinese to Modern Chinese. Compared to Baidu Translate, our model achieves an average advantage of +5 Rouge scores. Compared to the【Sui Wuya】translation model, it shows an improvement of +7 Bleu and an average of +10 Rouge scores. Additionally, compared to the mengzi-t5-base model, our model

improves by +3 Bleu and +8 Rouge-2 scores in Bleu-4 and Rouge-2, providing strong support for the study of ancient literature and related applications.

KEYWORDS: Large Language Model; Classical Chinese-Modern Chinese translation; ChatGLM3-6B model

目　录

插图目录

表格目录

第一章　引言

1.1　研究背景及意义

2022 年 4 月，中共中央办公厅、国务院办公厅发表《关于推进新时代古籍工作的意见》，强调要做好古籍保护、传承、发展工作，维护和弘扬好祖国宝贵文化遗产**XYDU202403001**。中华文化的现代化是延续优秀传统文化的现代化而不是消灭传统文化的现代化，文言文作为中国传统文化的重要组成部分，推广和传承文言文更是传承中华文化、加强中华民族凝聚力、实现中华文化现代化的重要关节。而局限于文言文的语法和现代文之间的区别，其翻译一直是一个具有挑战性的任务，这大大限制了现代人对文言文的阅读和理解文言文美学的能力。

因此，建立文言文翻译工具具有重要的学术和应用价值。首先，文言文翻译系工具作为文言文学习者的一个稳定、准确的学习工具，帮助其更好地理解和掌握文言文的语法和写作方式，从而降低学习难度。通过与工具互动，学习者能够翻译、分析文言文文本，获取语言知识和文化背景，提升阅读能力和文言文水平。其次，该工具对文言文研究和保护至关重要，为研究人员提供高质量的翻译服务，帮助他们更好地理解和分析文言文文献，挖掘知识和价值。最后，自动化翻译方式大幅提高了翻译效率，为文言文研究提供更多资源和工具支持。

然而，传统机器翻译方法在文言文-现代文翻译任务中面临着句法结构复杂、典故丰富和词汇特殊等挑战，导致翻译质量不尽如人意。近年来，随着人工智能的快速发展，特别是在自然语言处理和机器学习领域，大型语言模型 (如 GPT 系列) 通过深度学习和海量数据训练，具有较强的语言理解能力和文本生成能力，在文本翻译、文本生成、聊天机器人等多个任务中取得了显著的效果。基于这些人工智能大型语言模型的文言文翻译系统也显示出巨大的潜力，能够通过学习大量文言文语料，自动捕捉文言文的语法规则、词义和文化背景，实现更准确、流畅的翻译。

采用人工智能大语言模型构建的文言文翻译系统还有助于推动人工智能研

究和技术发展。通过解决文言文翻译中的挑战，如复杂的语法结构和文化理解，可以推动大模型在其他自然语言处理任务中的应用，并改进其训练和应用方法，提升性能和可扩展性。

综上所述，构建文言文翻译系统具有重要学术、应用和技术价值，为文言文学习、研究和保护提供了强大支持，促进了中外文化交流与理解，同时推动了人工智能研究和发展的进步。

1.2 国内外研究现状

在 21 世纪初期，语言模型的实现主要依赖统计方法，通过统计词语序列等方法来估计其概率分布，这些模型结构较为简单，在机器翻译领域取得了一定成果，为后续的技术发展奠定了基础。

自从 Yoshua Bengio 教授提出通过神经网络解决语言模型问题之后[bengio2000neural]，语言模型的探索便进入了神经网络时代，人们开始探寻通过神经网络建模语言的方法，到 21 世纪 10 年代初期，RNN（循环神经网络）[mikolov2010recurrent]已经被广泛应用于语言建模任务，其能够捕捉语言文本输入序列中的依赖关系。随后，LSRTM（长期记忆网络）[hochreiter1997long]的提出更是进一步解决了 RNN 难以处理长序列数据的问题，解决了语言模型在面对长文本是难以记忆长期依赖信息的问题。

2017 年，Google Brain 团队提出的 Transformers 模型[vaswani2017attention]彻底改变了语言模型领域的格局，Transformers 通过构建基于自注意力机制的神经网络，能够进行并行计算并且有效的捕捉长距离的依赖关系。Transformers 的出现摈弃了传统的循环神经网络和卷积神经网络，彻底改变了语言模型的建模方式，大幅提升了训练速度和性能，使得语言模型在计算效率和模型表现上取得了显著提升。

Google 于 2018 年发布了 BERT 模型[devlin2018bert]，BERT 模型是在 Transformer 架构上进行改进和扩展的预训练模型通过上下文编码器，能够更好的理解和展现文本预警。同时，BERT 模型通过无监督预训练和有监督微调下游任务的方式，实现了同一种模型在多种自然语言处理任务上的优秀表现，标志着大规模预训练模型的兴起。

OpenAI 发布的 GPT 系列模型也是大语言模型发展过程中极重要的一环，其于 2018 年发布的采用单向自回归预训练方式的 GPT-1 模型[radford2018improving]展现了 GPT 预训练模型在 NLP 领域的巨大潜力。随后 OpenAI 于 2019 年发布的 GPT-2 模型[radford2019language]在 GPT-1 的基础上进一步使用了更大规模的预训练数据和更深层次结构的神经网络，并去掉了 fine-tune 层，不再需要定义模型具体的下游任务。而 2020 年出现的 GPT-3 模型[brown2020language]更是达到了 1750 亿个参数的超大规模，在广泛的 NLP 任务，如文本生成、文本分类、文本翻译上取得了令人惊讶的成绩。OpenAI 的 GPT 模型，及其发布的 chatGPT 聊天机器人在展示大规模预训练模型在自然语言处理任务中的卓越表现、推动预训练模型技术的发展的同时，也证明了大语言模型迅速走向民用、步入商业化的可行性，大大加速了大语言模型商业化、规模化的进程。

而 Google 于 2020 年发布的 T5 模型[raffel2020exploring]提出将所有的语言建模任务都视为 text-to-text 的的问题，这样就可以实现通过同一个模型、目标函数、解码器来解决不同的下游 NLP 任务，大大提高了大语言模型的灵活性。

随着大语言模型技术的迅速发展，国内企业和高校也推出了众多国产大语言模型。比较著名的有阿里云团队于 2023 年前后陆续发布 Qwen 综合的言模型系列[bai2023qwen]，目前共发布包括 1.8B、7B、14B 等具有不同参数量的一系列模型。以及清华提出的一种基于自回归空白填充的通用语言模型（GLM）[du2021glm]，通过改进空白填充预训练，在 NLU、条件生成和无条件生成的广泛任务中取得了优于 BERT、T5 和 GPT 的表现，证明了其对不同下游任务的泛化能力，并让清华技术成果转化的公司智谱 AI 依据 GLM 发布开源预训练大模型 ChatGLM，推进国内大语言模型的开源和商用进展。

现阶段大语言模型的主流应用方式就是依托第三方开源的预训练大语言模型，根据需要的下游任务构建自己的数据集并进行微调，本项目就是使用于清华发布的 GLM 系列模型 ChatGLM3-6B 使用 P-tuning v2 方法[liu2021p]进行下游任务的微调。

1.3 文言文翻译的难点

现有的语言翻译模型层出不穷，但是主流的翻译模型大多是针对大语种的翻译模型，尤其以英语对其他语言的翻译为主，而文言文-现代文这种小语种翻译的研究则比较少。同时文言文翻译本身具有以下难点：

- 文言文的语法结构与现代汉语有很大不同。文言文通常采用简洁精炼的句式，省略主语和谓语的情况较多，句子结构松散且灵活。而现代汉语则更加注重完整的句式和明确的语法规则。这种结构上的差异使得直接转换常常导致语义不清或表达不完整，翻译模型无法获取足够的上下文信息，对文本语义的理解不够。
- 文言文中的词汇相对于现代汉语更加古朴和精炼，且很多词语在现代汉语中已经不再使用或意思发生了变化。例如，文言文中的“夫”、“其”、“者”等虚词在现代汉语中使用频率较低且意义有所不同。此外，文言文中常用的典故、成语和修辞手法，也增加了翻译的难度，需要对其进行准确理解和适当转换。
- 文言文的表达方式往往非常灵活和富有创造性，同时文言文本身高度要求文本的优雅性和文学性，这导致同一句文言文可能有多种不同的现代汉语翻译版本，这要求翻译模型不仅要能够准确理解原文，还要具备一定的翻译创造力，选择最优雅的现代汉语表达方式，这对于机器翻译模型来说是一个巨大的挑战。
- 文言文通常包含丰富的历史、文化和哲学背景，在人工翻译时就要求译者具备深厚的文化知识和历史背景，才能准确理解和翻译其中的内容。在输入翻译模型时就更要求模型对文言文语义的理解更加深刻，对模型的理解能力提出了更高的要求。

1.4 研究内容与主要工作

本项目主要工作是基于公司或企业发布的开源预训练大语言模型，输入构建的文言文原文-现代文翻译平行语料数据进行微调训练获得针对文言文翻译的大语言模型，主要工作如下：

- 收集并分析当下的主流大语言模型，并分析他们的优缺点。我们收集了当下主流的一些开源预训练大语言模型，梳理了大语言模型的发展过程，分析了这些模型的结构特点、性能倾向和各自的优势下游任务，
- 整理并提出一个用于微调大语言模型的文言文-现代文平行语料数据集。针对文言文翻译任务中，文言文-现代文语料数据集稀少、现有数据集质量层次不齐的问题，我们收集了多个现有的文言文-现代文语料数据集，通过脚本对这些数据集中的数据进行清洗，构建不同数据规模的数据集，获得了高质量的文言文-现代文语料数据集。
- 将提出的数据集用于开源预训练模型的微调，并与其他模型在测试集上比较性能的差异。我们将构建的高质量的文言文-现代文语料数据集用于基线模型 ChatGLM3-6B 的微调训练，让模型适应在文言文翻译这一下游任务，打破了基于 Transformers、BERT 等国外开源模型构建翻译模型的传统，使用对中文语境有更强的适应性国产开源大语言模型，让模型在文言文翻译任务上具有更好的性能。

1.5 论文组织结构

本文主要介绍了当前大语言模型的发展状况，构造了一个文言文-现代文平行语料数据集，通过数据集对大语言模型进行微调，并分析不同模型、不同参数、不同规模数据集之间的差异，确定一个性能最佳的翻译模型，并最终与现有的一些主流翻译模型/软件进行对比实验。

第一章：引言。主要介绍了目前大语言模型的发展现状、背景和应用方法，介绍了项目的主要任务和论文的结构。

第二章：基本概念和相关工作。主要介绍了当下主流的几个大语言模型的结构、原理和优缺点，分析了几个主流国产开源预训练模型的特点。

第三章：大语言模型微调实验设计。具体描述了微调实验的数据集构建、微调的训练、以及介绍了实验所使用的微调方法，并分析了模型在训练过程中性能的变化。

第四章：实验结果及分析。介绍了微调模型对比实验的设计和细节，对比分析了不同参数设置下微调模型的性能差异。

第五章：总结与展望。分析了本次实验的缺陷和问题，提出一些针对实验的可能的改进方法以及未来后续任务的展望。

第二章　基本概念和相关工作

2.1　大语言模型介绍

2.1.1　大语言模型

大语言模型（Large Language Model）是一种基于神经网络的 NLP 技术，可以通过学习自然语言的范式和规律执行处理任务，其最主要特征就是“大”，即参数规模庞大、能处理大量语言数据、具有强大理解能力。大语言模型的基本思想是通过将 NLP 任务中的自然语言文本看作一段输入序列，即文本序列或向量序列，模型内部的神经网络可以处理这些序列来生成对应的输出序列。

相比之前 NLP 领域的技术，依托庞大的参数规模，大语言模型具有更好的理解能力和学习能力，从而在下游任务中获得更好的性能。同时大语言模型强调端到端的学习方法，直接从构建的初始文本数据中进行学习，避免了传统方法中需要人工设计特征或规则的复杂方法。也由于庞大的参数规模，大语言模型训练的资源消耗量大、对数据依赖性强，以及在面对从未见过的数据时泛化能力不足。

2.1.2　Transformers

Transformers 于 2017 年被第一次提出[vaswani2017attention]，是一种处理序列数据的深度学习模型。Transformers 的模型结构如图2-1所示，是一种典型的编码器-解码器结构模型，编码器部分将输入的文本序列编码成向量，解码器部分针对该向量编码进行解码获得作为输出的文本序列。经过训练的编码器-解码器结构模型可以从输入文本序列生成输出文本序列。

最初的 Transformers 结构中，模型的编码器和解码器均是由 6 个相同的子层组成的，编码器的每个子层由一个前馈网络和一个多头注意力层构成，解码器的子层则多一个掩码多头注意力层，通过注意力层建立序列内部的关联性、通过

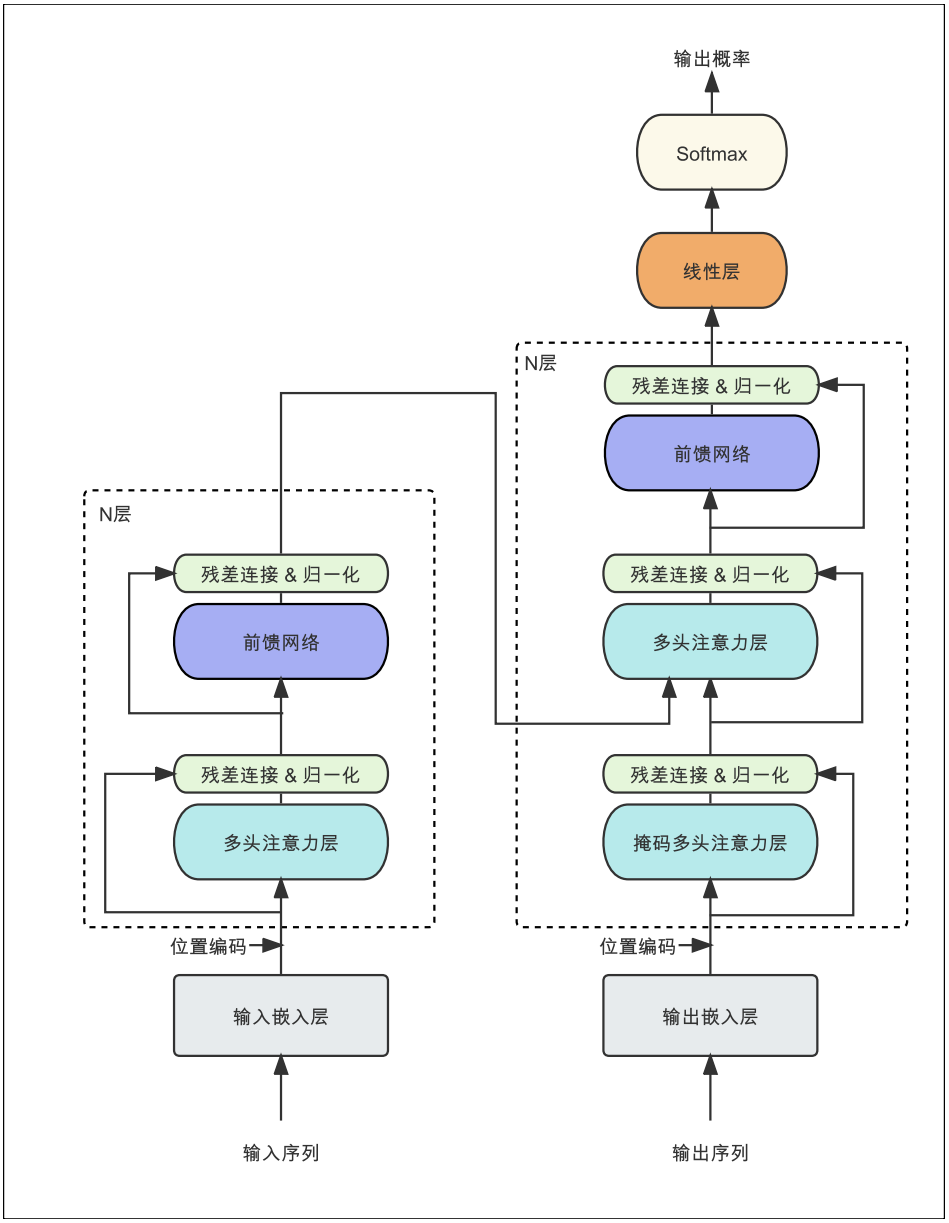

图 2-1　Transformers 的模型结构

前馈网络对每个位置编码的隐藏表示进行变换和映射，之后每个子层连接到一个残差连接和归一化层。在解码器末端通过线性层将输出向量映射到词汇表上，在通过 Softmax 函数归一化，得到最终的输出概率分布。

Transformers 相比之前的语言模型的最大优势就是 Transformers 能够迅速适应不同的下游任务，从业者只需要在预训练模型的基础上使用自己的小规模数据集对其进行调整就可以让模型适应新的任务。

2.1.3 T5

T5 是由 Google 提出的基于 Transformers 的编码器-解码器结构的模型[raffel2020exploring]，和以往不同的是，T5 统一了输入和输出的表现形式，预训练模型不再需要使用特定的数据集针对具体的下游任务进行有监督的微调。T5 模型将所有的 NLP 任务都看作 text-to-text 的任务，将不同下游任务的输入转化为几乎一致的格式，即“任务前缀声明 + 输入文本”。因此，在 T5 模型上不在有文本生成、文本摘要、文本翻译等细分的下游任务，也就做到了通过一套损失函数、评估指标等来评测不同任务，也可以方便的评估模型结构、数据集、损失函数对不同任务的影响。

2.2 国产开源预训练大语言模型

2.2.1 Qwen

Qwen 是由阿里旗下通义千文发布的一系列大语言模型，包括基础预训练语言模型（基座模型）、聊天模型（Qwen-Chat）、编码专用模型（Code-Qwen）、数学专用模型（Math-Qwen-Chat）等多个版本，以满足不同需求的应用，适用于各种应用场景和下游任务。Qwen-Chat 模型使用人类反馈强化学习训练（RLHF）[christiano2017deep]，具有先进的对话生成能力，展现出优秀的工具使用和规划能力。编码专用模型和数学专用模型，如 Code-Qwen 和 Math-Qwen-Chat 为编程和数学领域的语言理解提供了更专业的支持，Qwen 系列模型汇总如图2-2所示。

Qwen 采用了改进版的 Transformer 架构，并使用 LLaMA[touvron2023llama]的训练方法，并做出了以下改进：

- 嵌入和输出映射不共享权重，从而以内存开销为代价获得更好的性能。
- 使用了 RoPE（旋转位置编码）[su2024roformer]进行位置编码，因为需要优先考虑模型性能以及追求较高的精度，并放弃 BF16 和 FP16 转而采用了 FP32 精度的逆频率矩阵。
- 使用了预归一化（Pre-Norm）和 RMSNorm 进行规范化，与后归一化相比，已被证明能提高训练的稳定性。
- 使用了 SwiGLU 作为激活函数，它是 Swish 和 Gated Linear Unit 的组合。
- 除了在 QKV（查询、键、值）层中保留 Bias 以外，在大多数层中都移除了

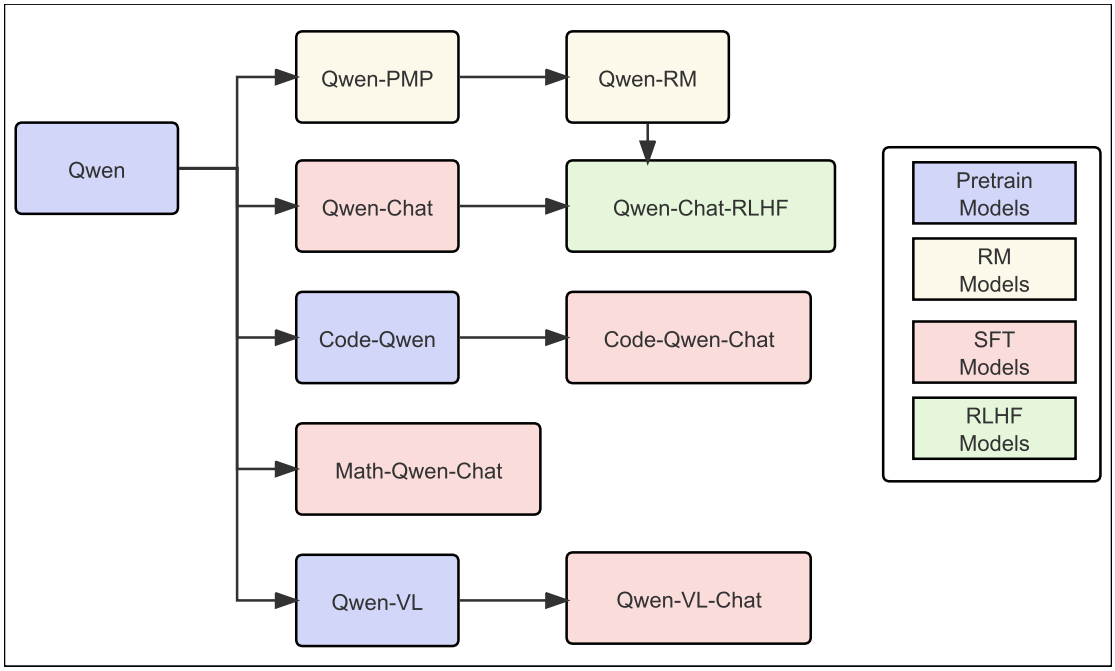

图 2-2　Qwen 系列模型脉络

Bias，以提高模型的外推能力。

目前 Qwen 已经发布 Qwen-1.8B、Qwen-7B、Qwen-14B、Qwen-72B 等预训练模型，并开源其中一部分。

2.2.2　ChatGLM

GLM[du2021glm]是由清华发布的通用的预训练语言模型，在自然语言理解、条件文本生成和非条件文本生成上都有着不错的表现。GLM 通过一种创新的预训练方法来实现在有条件的文本生成任务中效果很好又在无条件的文本生成任务中表现很好：通过自编码思路从输入序列中随机将一些连续的单词设置为空白，然后训练模型使用自回归思路按照一定的顺序来复原这些单词。该方法继承了自编码和自回归两个架构的优点。此外，GLM 还引入了一些额外的技术：它重新排列了空白区域的预测顺序，并采用了二维位置编码。其中，第一个维度编码了 span 在原始文本中的位置，而第二个维度编码了 token 在 span 中的位置。实验证明，在参数量和计算成本相当的情况下，GLM 能够在 SuperGLUE 基准测试中明显领先于 BERT。此外，在使用相似规模的训练集进行预训练时，相比 BART[lewis2019bart]和 RoBERTa[liu2019roberta]，GLM 的表现要更好。在自然语言理解和生成任务上，GLM 也能够显著超越 T5，而且所需的参数和数据量更少。

ChatGLM 是清华基于 GLM 提出的预训练模型，现已发布 ChatGLM3-6B、GLM-130B[zeng2022glm]等开源预训练模型，可以轻松在消费极显卡上进行部署、微调等操作，同时展现出较好的中文下游任务兼容性、更长的上下文理解能力、更好的推理效率。

2.3 本章小结

本章中介绍了基本概念和相关工作。首先介绍了目前主流的大语言模型，并简单介绍了各个模型的网络结构、特点。然后分析了两个较为新颖的、国产的大语言模型，介绍了他们的模型结构、特点和优势，并说明了基于这些国产大语言模型所开源的预训练模型。

第三章　大语言模型微调实验

3.1　项目规划

3.1.1　数据集构建

在数据集构建方面，随着社会的发展，人们越来越开始重视古文文化的传承，因此发展出了越来越多的古文资料网站，这些网站提供了宝贵的文言文语料数据和对应的翻译数据，因此本研究尝试通过网络爬虫来获取古文和其现代文翻译数据，构建文言文-现代文平行语料库。我们从古诗文网[1]爬取了约 8800 条文言文-现代文平行语料数据但是经过分析，发现这些数据存在较多缺陷，包括但不限于：

- 数据质量不稳定：通过爬虫获取的数据包含大量噪音和错误信息，且包括故事、词、文章等类型数据，文本质量参差不齐。
- 数据标注不准确：文言文与现代文之间的对应关系存在歧义或错误，数据存在标注不准确的情况，这会对翻译模型的学习产生负面影响。
- 数据偏差问题：爬虫获取的数据存在采样偏差，即某些类型或主题的文本数量过多，而另一些则数量稀少，导致模型在某些领域的性能不佳。
- 数据量不足：爬虫获取的数据中包含的有效的训练数据量很少。
- 单条数据记录过长：大部分网络上的原始语料数据是以文章为单位组织的，这样未经切分的文本数据过长，语言模型很难处理这样长的输入数据。

因此，我们选择引用 NiuTrans 的文言文-现代文平行语料库[2]和 Erya 数据集**guo2023towards**。NiuTrans 基本包含了大部分的文言文古籍，对每本古籍按篇章/章节进行划分与展示，并从文学的角度对所有古籍原著进行整理。对于平行数据，NiuTrans 项目整理出双语数据，以句子级别为单位进行划分，提供了原文、译

1　古诗文网主页地址.https://www.gushiwen.cn.

2　NiuTrans 的文言文-现代文平行语料库 GitHub 仓库地址.https://github.com/NiuTrans/Classical-Modern?tab=readme-ov-file.

表 3-1　Erya 数据集评测基准数据

	古代	中代	近现代	文章	小说
古籍来源	《汉书》	《新唐书》	《明史》	《徐霞客游记》	《太平广记》
数据规模/条	18646	9396	66730	16649	45162
平均句子长度/字	21.2	20.5	21.5	25.1	20.0

文、双语三种数据格式，且所有数据均按行保留了古文原文的相对顺序。同时项目对文言文-现代文原始数据生成的双语数据进行篇章级对齐，核心对齐思路是采用归一化编辑距离算法与长度比指标，通过脚本将双语 (平行) 数据进行分句、对齐处理，共 972467 句。

Erya 数据集是文章**guo2023towards**中提出的一个文言文单语料和文言文-现代文平行语料数据集，包括从 1000BC—AD1600 范围内的古籍数据，针对这些数据，Erya 数据集通过统一字符标点、繁体字简化的手段去除噪音数据，minhash 算法去重，guwen-punc 模型添加标点的方法清洗数据，统一古文数据的标准，构建出 1941M 条文言文单语料数据、84.8M 条文言文-现代文平行语料数据。

Erya 数据集在提出上述数据集的同时，还综合四库分类法以及古汉语年代划分惯例，设计了史书（包含古代汉语（公元 300 年之前）、中古汉语（公元 400 年到公元 1200 年）、近代汉语（公元 1300 年到公元 1900 年）三个阶段）、文集（各种文学作品，包括散文、诗歌、诸子以及文学评论）、小说（文白混杂，因而有其独特文体特征）这三种古文文本分类，并根据这种分类准则，充分考虑不同时代、不同风格、不同语法、不同文种的文言文对模型翻译结果的影响，从文言文-现代文平行语料库中提出一个子集作为测试集的基准，如表3-1所示：

本研究从 Erya 数据集中通过去噪、筛选手段整理出一个 20w 条记录和一个 50w 条文言文-现代文平行语料数据作为训练集来对极限模型进行微调训练，筛选标准如下：

- 去除短文本记录：因为原数据集存在大量四字成语或四字短语的数据，而加上末尾标点符号后数据长度为 5，这些数据晦涩难懂，同时普遍缺失上下文信息，在翻译过程中无法补全确实的语法信息，即使是经验丰富的人类来进行翻译也很困难，大语言模型很难处理这些数据，因此选择去除古文字符长度不超过 5 的记录。

- 去除长文本记录：数据集中存在一些古文原文过长的数据，这些数据由两段关联性不强的文本构成，考虑到大语言模型对输入文本序列的长度限制，将这些文本去除或按语义切分开，形成长度适中的记录。
- 去除晦涩文本记录：考虑到文言文本身语法与现代汉语的巨大差距，部分数据记录来源的古籍过于古老，古文原文和现代文翻译之间存在巨大差异，模型难以处理这些数据，因此去除现代文序列和古文序列文本长度比大于 2 的数据。

同时考虑到文言文-现代文预料数据长度、音节、数据来源古籍年代对翻译模型性能的影响，本研究从 NiuTrans 数据集中按以上标准整理出一个 50w 条文言文-现代文平行语料数据（包括《汉书》、《明史》、《新唐书》、《旧唐书》、《北齐书》、《辽史》、《晋书》、《宋书》、《史记》、《世说新语》、《宋史》、《隋书》、《太平广记》、《魏书》、《徐霞客游记》、《资治通鉴》）作为对照数据集。

3.1.2 基线模型选择

随着大语言模型的商业化进展不断推进，有大量预训练模型被开源，而在中文领域也有一系列优秀预训练大语言模型，如 ChatGLM、Qwen 系列、CPT 等。本研究选择清华发布的 ChatGLM3-6B[1]作为基线模型，ChatGLM3-6B 是 ChatGLM 系列的最新一代开源预训练大语言模型，ChatGLM3-6B 的的基础模型 ChatGLM3-6B-Base 性能相比之前更加强大，同时还保留了前两代模型部署资源需求门槛低、模型运行对话流畅的优点。ChatGLM3-6B-Base[2]采用了更多样的训练数据、更充分的训练步数和更合理的训练策略。

3.1.3 微调方法介绍与选择

提升开源预训练大语言模型的一种常用策略就是全参数微调，这种方法可以基于训练集对预训练模型的所有参数进行调整，在资源、时间、数据充足的情况下可以充分让预训练模型适应下游任务。然而针对预训练模型进行全参数微

1 ChatGLM3-6B 预训练模型 ModelScope 主页.https://modelscope.cn/models/ZhipuAI/ChatGLM-6B/summary.

2 ChatGLM3-6B-Base 预训练模型的 ModelScope 主页.https://modelscope.cn/models/ZhipuAI/chatglm3-6b-base/summary.

调是一项资源密集型任务，随着大语言模型的参数规模不断增长，针对预训练模型进行全参数微调需要强大的计算能力来管理优化器状态和检查点，以及开源预训练模型在零样本情况下的性能增强，在相同时间和资源下的全参数微调对预训练模型在下游任务上性能的提升也不并没有那么高。通常情况下，对模型进行全参数微调所占的内存空间是内存本身大小的 12 倍[sun2023comparative]，所以即使是 60 亿参数的 ChatGLM3-6B 模型也需要大量计算资源，这对本研究来说是不可接受的。

为了解决全参数微调需要大量计算资源的问题，Google 于 2019 年提出针对 BERT 模型微调的 Adapter Tuning 方法[houlsby2019parameter]，即在 BERT 模型的 Transformers 层中加入一个 Adapter，在训练时只对 Adapter 结构中的参数进行微调，固定住原来的预训练模型的参数不变，如图3-1所示：

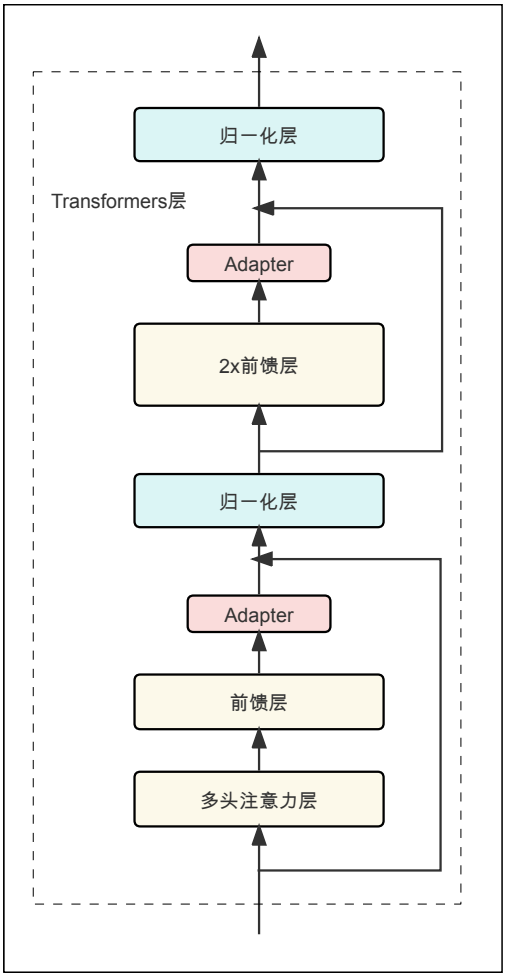

图 3-1　PEFT 的结构图

为了保证训练效率，Adapter 通过降维层将输入的高纬度特征映射到低维度

上，再通过一个非线性层转换后进入升维层，将低维度特征映射回高纬度特征，Adapter 的结构如图3-2所示：

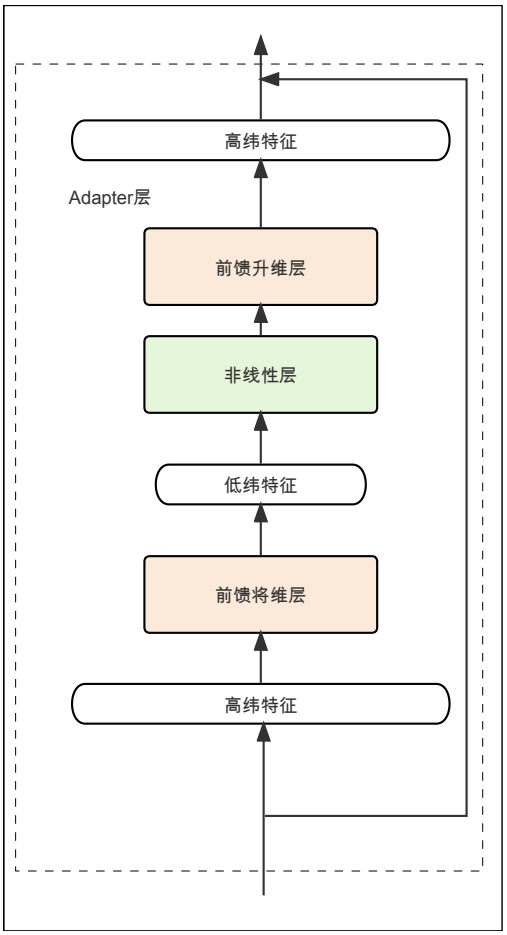

图 3-2　Adapter 的结构图

Adapter 方法极大降低了微调所需要的计算资源，同时能在只额外增加原来预训练模型 3.6% 的参数的情况下取得于全参数微调很接近的效果[houlsby2019parameter]。在此之后，大部分预训练模型的微调进入了 PEFT 的时代，通过近微调少量参数，极大降低了预训练模型微调的计算资源消耗和成本。

本研究选择使用 LoRA[hu2021lora]方法和 P-tuning v2[liu2021p]方法来进行高效参数微调，并对比两种微调方法在结果上的差异。LoRA 方法是用于大语言模型微调任务的低秩适应技术，大大减少了下游任务的可训练参数的数量，与经过 Adam 微调[kingma2014adam]的 GPT-3 175B[brown2020language]相比，LoRA 可以将可训练参数的数量减少 10000 倍，GPU 内存需求提高 3 倍。LoRA 方法的基本原理是冻结预训练的模型权重，并在 Transformers 结构的模型的每一层中注入一个可训练的秩分解矩阵层，将这个添加的秩分解矩阵层的输出和原本模型路径的输出相

加输入到后续网络当中来获取最终输出序列，通过之训练这些添加的秩分解矩阵层的参数来达到修改原模型路径输出的目的。其中秩分解矩阵层由两个矩阵组成，两个矩阵分别负责升维和降维，LoRA 的原理结构图如图3-3所示：

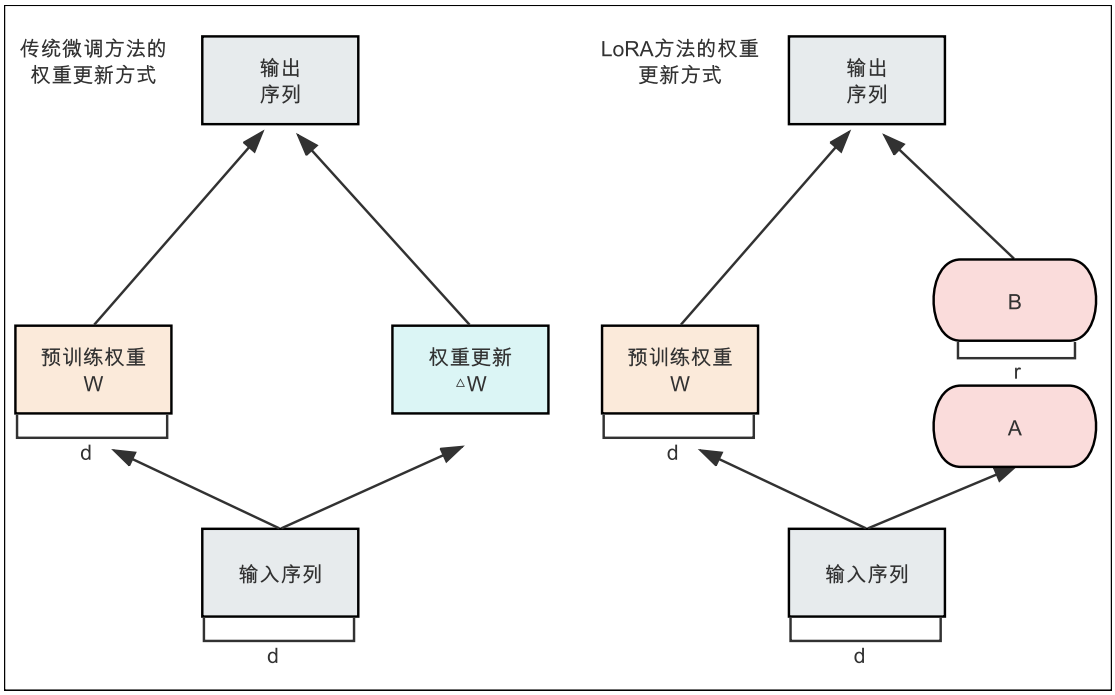

图 3-3　LoRA 的原理结构图

P-tuning v2 方法是对 Deep Prompt Tuning[li2021prefixqin2021learning]的一种改进和适应性实现。Fine-tuning 方法需要微调整个预训练模型，而且加入了新的参数，在训练过程中需要消耗大量内存，而 Prompting 方法冻结了预训练模型的所有参数，只通过添加 Prompt 来预测结果，因此可以在不微调的情况下改变预训练模型的输出，但是容易陷入局部最优的困境。Prompt-tuning（代指一系列 Prompt 微调方法而非单一方法）把 Prompt 也加入到微调过程中，此时只对 Prompt 部分参数进行微调而不改变预训练模型的参数。但是 Prompt-tuning 在模型大小不足、尤其是模型参数少于 100 亿时表现不佳，尤其是在一些自然语言理解任务中的推理能力十分有限。

因此 Deep Prompt Tuning 在此基础上作出了改进，使用 Prefix-tuning[li2021prefix]的深层模型来加强模型的推理能力，如图3-4所示：

P-tuning v2 就是对 Deep Prompt Tuning 技术的改进和适应性具体实现，P-tuning v2 的优化策略主要包括前缀提示和自适应优化，前者通过将提示信息加入到模型中以提高输出准确性，后者则根据模型在训练中的表现动态调整参数

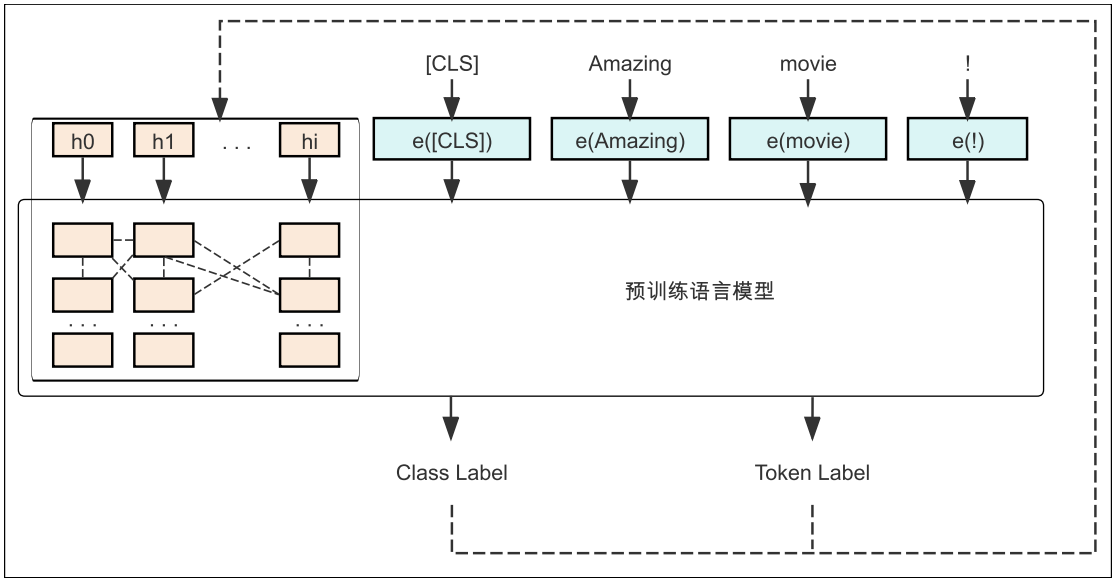

图 3-4　Deep Prompt Tuning 的原理结构图

的权重，提高模型微调时的收敛速度。

3.1.4　评价指标的选择与介绍

本文采用目前文本翻译任务最常用的评估指标 Bleu 系列[papineni2002bleu]中的 Bleu-4 和 Rouge 系列[lin2004rouge]中的 Rouge-1、Rouge-2、Rouge-L 来评价模型。Bleu 指标的总体思想就是准确率，是一种基于 n-gram 模型计算精确度的来评价句子之间相似度的方法，在 n-gram 下语言模型生成的文本和标准译文的匹配度计算公式如式3-1所示：

$$p_n = \frac{\sum_{c\in\text{candidates}} \sum_{\text{n-gram}\in c} \text{Count}_{\text{clip}}(\text{n-gram})}{\sum_{c'\in\text{candidates}} \sum_{\text{n-gram}'\in c'} \text{Count}_{\text{clip}}(\text{n-gram}')} \tag{3-1}$$

其中，$candidate$ 表示语言模型生成的文本，$reference$ 表示标准译文，$Count_{clip}(n-gram)$ 表示某个 $n-gram$ 在 $reference$ 中的个数，所以分子就是在给定的 $candidate$ 中有多少个 $n-gram$ 出现在 $reference$ 中，分母表示所有的 $candidate$ 中 $n-gram$ 的个数。此时 Bleu 的最终计算公式为式3-2：

$$Bleu = BP \times \exp\left(\sum_{n=1}^{N} w_n \log(p_n)\right) \tag{3-2}$$

其中 BP 是惩罚因子，w_n 是每个 n-gram 的权重。Bleu 方法根据所使用的

n-gram 模型的 n 的取值分为多种评价指标，常见的 n 指标有 1、2、3、4 四种。其中 Bleu-1 衡量的是单词级别的准确性，偏向于较短的翻译结果，越高阶的 Bleu 方法的评价约倾向衡量句子的流畅度，在文本翻译任务上更要求句子之间语义的相似度和句子的流畅度，而由于文言文和现代文之间文法的巨大差异，单纯衡量单词之间的准确性意义不大，因此我们在 Bleu 系列指标中仅采用 Bleu-4。

Rouge 是一种基于召回率的相似度量方法，和 Bleu 类似，Rouge-N 也是基于 n-gram 模型计算句子的召回率与准确率的比值，衡量模型输出和标准译文之间的相似性，Rouge-N 的计算公式和 Bleu 基本相同，但是是基于召回率的，如式3-3：

$$\text{Rouge-N} = \frac{\sum_{c\in\text{references}} \sum_{n\text{-gram}\in c} \text{Count}_{\text{clip}}(n\text{-gram})}{\sum_{c'\in\text{references}} \sum_{n\text{-gram}'\in c'} \text{Count}_{\text{clip}}(n\text{-gram}')} \tag{3-3}$$

其中，$candidate$ 表示语言模型生成的文本，$reference$ 表示标准译文，$Count_{clip}(n-gram)$ 表示某个 $n-gram$ 在 $reference$ 中的个数。相比 Bleu 更关心输出句子的准确度，Rouge-N 指标更关心输出句子的充分性和忠实性。但是由于当 N>3 时 Rouge-N 指标的值通常都很小，缺少足够的区分度，因此我们只采用 Rouge-1 和 Rouge-2 指标。

Rouge-L 的思想和 Rouge-N 类似，只是 Rouge-L 使用最长公共子序列（LCS）计算召回率，衡量模型输出对标准译文的覆盖程度，公式如式3-4：

$$\text{Rouge-L} = \frac{Count_{\max}(\text{LCS})}{Count(\text{reference})} \tag{3-4}$$

其中分子表示语言模型生成的文本和标准译文之间的最长公共子序列的长度，分母表示标准译文的长度。

此外，考虑到文言文和现代文之间语法、句法、词语的巨大差异，仅仅使用机械的评价指标是不能完全体现模型的差异和翻译结果的优美程度，同时评价文言文-现代文的翻译结果也是一个主观性极强的任务，因此加入人工评测指标进行主观评分的补充。

3.1.5 实验环境

整个项目使用 Python 编程，项目运行的环境、组件依赖和硬件条件如表3-2所示：

表 3-2 项目运行的环境、组件依赖和硬件条件

项目	版本号
操作系统	Ubuntu 22.04.2 LTS
Python 版本	3.8.19
Transformers 库版本	4.28.1
ModelScope 库版本	1.13.3
Pytorch 库版本	1.13.1+cu117
cuda 版本	11.7
GPU 型号	Tesla V100-SXM3-32GB

3.2 实验过程

3.2.1 准备数据

在微调模型之前，我们要对数据进行预处理，包括数据清洗、格式转换等操作，同时需要将储存在不同文件的文言文语料数据和对应的现代文预料数据转换成文言文-现代文平行语料数据。同时，针对 ChatGLM3-6B 的训练数据，我们用“请将这段文言文翻译为现代文：”作为 Prompt 提示。输入数据格式如图3-5，数据准备部分的脚本如图3-6：

```
{
  "content":
  "请将这段文言文翻译为现代文：成皇帝讳衍，字世根，明帝长子也。",
  "summary": "成皇帝名司马衍，字世根，是明帝司马绍的长子。"
},
{
  "content": "请将这段文言文翻译为现代文：太宁三年三月戊辰，立为皇太子。",
  "summary": "太宁三年三月初二，立为皇太子。"
},
{
  "content": "请将这段文言文翻译为现代文：闰月戊子，明帝崩。",
  "summary": "闰八月二十五日，明帝驾崩。"
},
```

图 3-5 输入数据格式

```python
def generate_json_object(id, str1, str2):
    """
    生成一个 JSON 对象
    """
    json_object = {
        "content": "请将这段文言文翻译为现代文：" +
            str1.replace("请翻译以下文言文语句：", ''),
        "summary": str2
    },
    return json_object

def read_lines(file_path):
    with open(file_path, 'r') as file:
        lines = file.readlines()
    return lines


def main(file1_path, file2_path):
    lines_file1 = read_lines(file1_path)
    lines_file2 = read_lines(file2_path)

    # 确保两个文件中行数相同，否则无法一一对应
    if len(lines_file1) != len(lines_file2):
        print("Error: 文件行数不一致")
        return

    # 输出对应行的数据
    json_objects = []
    for line1, line2 in zip(lines_file1, lines_file2):
        if len(line1.strip()) < 6 or (len(line2.strip()) >
            len(line1.strip()) * 2):
            print(line1.strip() + ' ' + str(len(line1.strip())))
            continue
        print(line1.strip(), line2.strip())
        # 生成多个 JSON 对象
        json_object = generate_json_object(cnt, line1.strip(),
            line2.strip())
        json_objects.append(json_object)

    # 将多个 JSON 对象写入文件
    with open("test-all.json", "a") as f:
        json.dump(json_objects, f, indent=2, ensure_ascii=False)
```

图 3-6　数据准备脚本

根据脚本，从 Erya 数据集中，我们整理出一个 20000 条数据的的小规模训练集，用于初步验证模型微调脚本的可行性，比较在较少数据情况下模型性能相比未微调的预训练模型的变化。以及一个 200000 条数据的中等规模训练集、一个 500000 条数据的正式训练集，其中 500000 规模数据集完全包括 200000 规模的训练集，用于比较在训练数据变大以及训练时间增长时模型性能的变化。在训练集以外，我们从评测基准数据集里整理出一个 2000 条数据的小规模验证集，用于快速验证模型在训练过程中的性能表现，在短时间内获得较为全面的模型性能数据，节省训练资源。以及一个约 110000 条数据规模的测试集，用于与其

他模型做横向对比。针对 NiuTrans 数据集，根据脚本我们整理出一个 500000 条数据规模的对照训练集。

3.2.2 微调脚本

微调脚本方面，我们使用 THUDM 发布的官方工具仓库[1]，仓库中支持工具调用、代码执行、模型微调等功能，中提供的 P-tuning v2 微调脚本，脚本参数如图3-7：

```
PRE_SEQ_LEN=128
LR=2e-2

CUDA_VISIBLE_DEVICES=1 python3 main.py \
    --do_train \
    --train_file ChatGLM-6B/ptuning/ptuning.json \
    --validation_file ChatGLM-6B/ptuning/valid.json \
    --prompt_column content \
    --response_column summary \
    --overwrite_cache \
    --model_name_or_path ChatGLM-6B/model/ZhipuAI/ChatGLM-6B \
    --output_dir output \
    --overwrite_output_dir \
    --max_source_length 128 \
    --max_target_length 128 \
    --per_device_train_batch_size 8 \
    --per_device_eval_batch_size 1 \
    --gradient_accumulation_steps 2 \
    --predict_with_generate \
    --max_steps 350001 \
    --logging_steps 10 \
    --save_steps 5000 \
    --learning_rate $LR \
    --pre_seq_len $PRE_SEQ_LEN \
    --quantization_bit 4
```

图 3-7 P-tuning v2 微调脚本参数

对于 LoRA 方法，我们使用 GitHub 仓库[2]中提供的脚本，脚本参数如图3-8：

3.2.3 训练过程

对于 Ptuning-v2 微调方法，在使用 Erya 数据集构建的 20000、200000、500000 的规模训练集上，模型微调过程 loss 大致的变化图如图3-9、3-10、3-11所示：

1 THUDM 发布的 GitHub 代码仓库.https://github.com/THUDM/ChatGLM-6B.git.

2 ChatGLM3-6B 的 LoRA 微调脚本 GitHub 仓库.https://github.com/hiyouga/ChatGLM-Efficient-Tuning.git.

```
CUDA_VISIBLE_DEVICES=0 python train_bash.py \
    --stage sft \
    --model_name_or_path ChatGLM-6B/model/ZhipuAI/ChatGLM-6B \
    --do_train \
    --dataset ancient-trans \
    --finetuning_type lora \
    --output_dir ChatGLM-6B/ChatGLM-Efficient-Tuning-main/output \
    --per_device_train_batch_size 4 \
    --gradient_accumulation_steps 4 \
    --lr_scheduler_type cosine \
    --logging_steps 10 \
    --save_steps 5000 \
    --learning_rate 5e-4 \
    --num_train_epochs 5.0 \
    --plot_loss \
    --fp16
```

图 3-8　LoRA 微调脚本参数

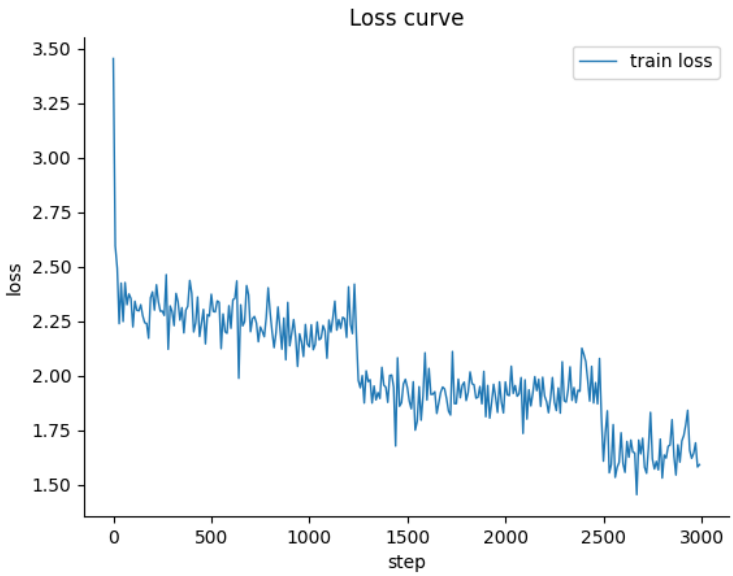

图 3-9　20000 规模数据集微调模型 loss 变化图

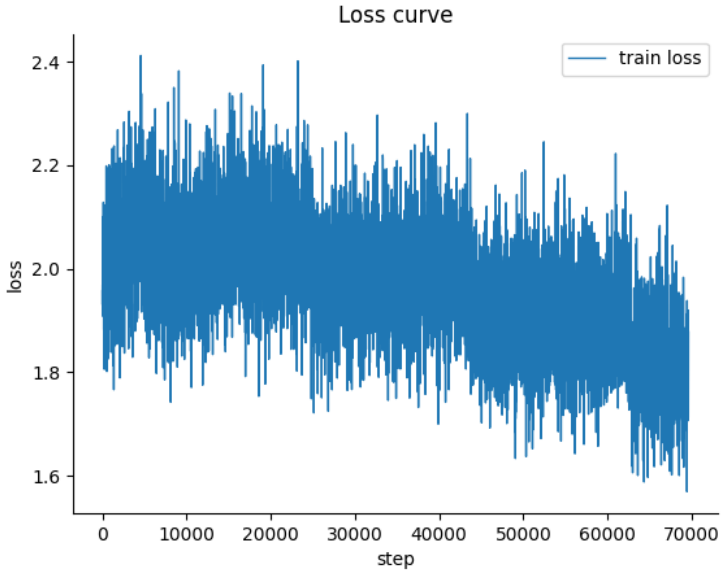

图 3-10　200000 规模数据集微调模型 loss 变化图

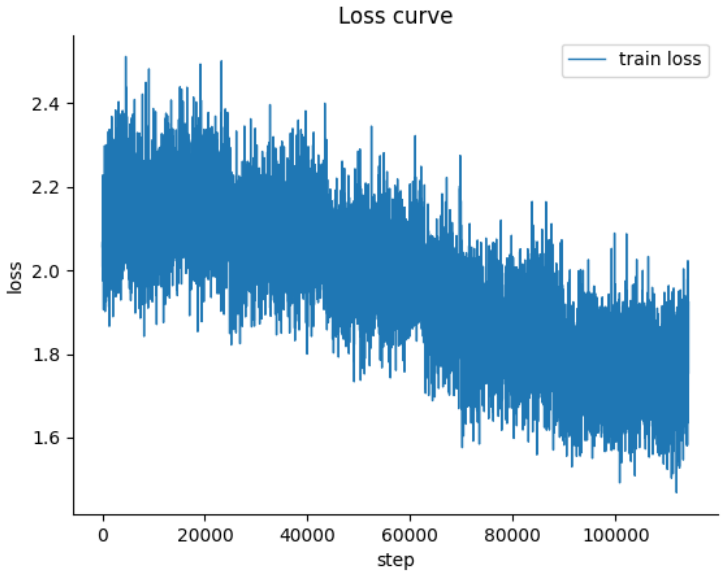

图 3-11　500000 规模数据集微调模型 loss 变化图

对于 LoRA 微调方法，在使用 Erya 数据集构建的 500000 规模数据集上的 loss 大致变化如图3-12所示：

对于使用 NiuTrans 数据集构建的 500000 规模长文本训练集上的微调，我们也使用 P-tuning v2 方法进行训练，loss 变化如图3-13所示：

可以看到，无论是哪种规模的数据集和微调方法，微调过程中模型的 loss 变化都是总体呈下降趋势，在图3-9中可以看出在前几个 step 中 loss 下降非常明

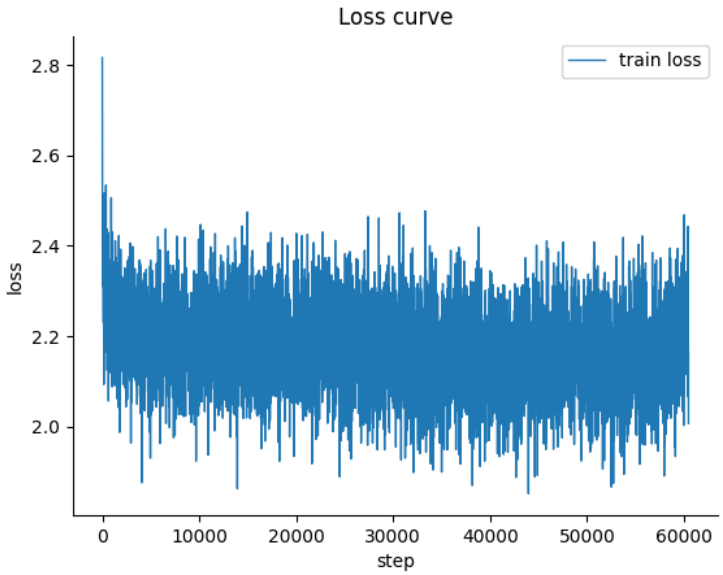

图 3-12　50000 规模数据集的 LoRA 方法微调模型 loss 变化图

显，之后 loss 下降逐渐放缓。而在 20000 以上的数据集上，模型的 loss 在下降至 2.0-1.8 附近时处于波动状态，整体不再趋近于下降，这符合一般大语言模型在微调时的收敛状态，体现出我们对微调参数的选择、模型的训练基本是正确的，模型的训练过程是有效的。

3.2.4　验证集表现

我们首先使用 20000、200000、500000 规模的数据集，使用 P-tuning v2 方法对 ChatGLM3-6B 模型进行初步微调，以验证模型在文言文-现代文翻译任务上的可行性和增大训练集规模对模型性能的影响，针对使用 P-tuning v2 方法微调，我们对比了不进行微调的 ChatGLM3-6B 模型、使用 20000 规模数据集微调、使用 200000 规模数据集微调、使用 500000 规模数据集微调在验证集上的表现，在表3-3中表示为 ChatGLM3-6B、ChatGLM3-6B-20000、ChatGLM3-6B-200000、ChatGLM3-6B-500000，其中，ChatGLM3-6B-200000、ChatGLM3-6B-500000 在训练过程中的评分变化分别如图3-14、3-15所示：

根据表3-3的数据可以看出，ChatGLM3-6B 模型在使用 20000 数据集简单微调以后，相比未微调预训练模型在各个指标上均有较大提升（+13Bleu），而相比使用 20000 小数据集微调的效果，使用 20000 规模数据集进行微调获得的模型在后续训练中表现出更好的性能（+4Bleu），而继续增加数据集规模的 chatGLM3-

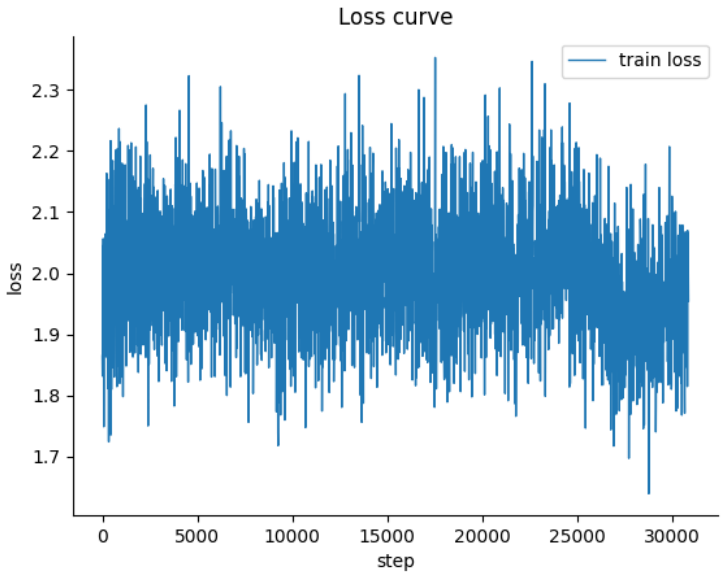

图 3-13　50000 规模 NiuTrans 数据集的 P-tuning v2 方法微调模型 loss 变化图

表 3-3　ChatGLM3-6B、ChatGLM3-6B-20000、ChatGLM3-6B-200000、ChatGLM3-6B-500000 训练过程的评分

模型/指标	Bleu-4	Rouge-1	Rouge-2	Rouge-L
ChatGLM3-6B	12.6734	31.4470	10.8959	28.9438
ChatGLM3-6B-20000	25.8668	45.4187	21.1878	45.9646
ChatGLM3-6B-200000	29.2028	49.3595	24.6433	49.8838
chatGLM3-6B-500000	30.4521	50.2036	26.0271	50.7910

6B-500000 在验证集上的性能也取得了增长（+2.4Bleu）。随着训练数据集规模的增长，Rouge 分数也均有较高增长，说明模型的输出文本相比标准译文的准确度、流畅度都有提升。实验结果表明，在基线模型的基础上注入大量文言文-现代文语料数据进行训练，能够有效提升模型对文言文的理解和推理能力。此外，随着微调语料规模的增加，模型处理下游任务知识的能力也在逐步增强。

由于不同的微调方法各有优势，我们分别使用 P-tuning v2 方法、LoRA 方法在相同的数据集上对 ChatGLM3-6B 进行微调，以验证不同的微调方法的影响，在表3-4中表现为 ChatGLM3-6B-P-tuning v2、ChatGLM3-6B-LoRA：

使用 P-tuning v2 方法微调模型在 Bleu 和 Rouge 指标上均比使用 LoRA 微调模型的分数要高（+4.7Bleu），这验证了使用 P-tuning v2 方法进行微调的正确性。

由于文言文-现代文平行语料库在作为模型的训练数据时，不同数据类型、文本长度、文言文来源年代、数据来源语料库等变量都可能会影响模型的性能，

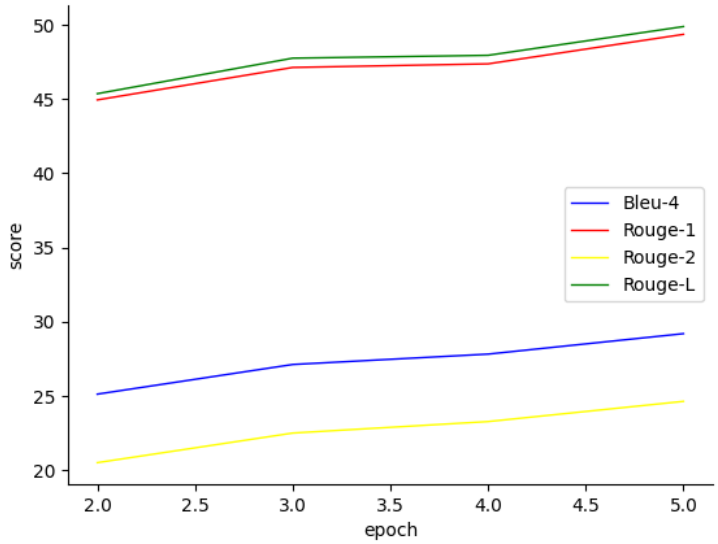

图 3-14　ChatGLM3-6B-200000 的评分变化图

表 3-4　ChatGLM3-6B-P-tuning v2、ChatGLM3-6B-LoRA 在验证集上的分数

模型/指标	Bleu-4	Rouge-1	Rouge-2	Rouge-L
ChatGLM3-6B-P-tuning v2	30.4521	50.2036	26.0271	50.7910
ChatGLM3-6B-LoRA	26.9438	46.8453	22.7816	47.4667

我们分别使用 Erya 数据集和 NiuTrans 数据集构建了相同大小的训练集，并使用 P-tuning v2 方法训练了相同的时间，在表3-5中表现为 ChatGLM3-6B-Erya、ChatGLM3-6B-NiuTrans：

相比 Erya 数据集，NiuTrans 数据集的文言文-现代文数据的单条数据长度更长、文言文来源更广更杂更早，因此在模型上进行训练的结果也不同，使用 Erya 数据集训练的结果相比使用 NiuTrans 数据集的结果取得了 +4.8Bleu、平均 +5Rouge 的差距，这验证了选择 Erya 训练集的正确性。

表 3-5　ChatGLM3-6B-Erya、ChatGLM3-6B-NiuTrans 在验证集上的分数

模型/指标	Bleu-4	Rouge-1	Rouge-2	Rouge-L
ChatGLM3-6B-Erya	30.4521	50.2036	26.0271	50.7910
ChatGLM3-6B-NiuTrans	26.8861	46.2848	22.3509	46.2564

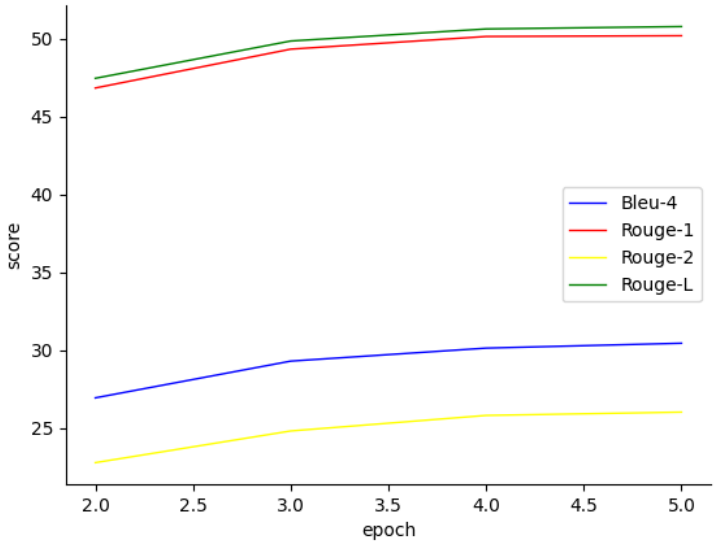

图 3-15　ChatGLM3-6B-500000 的评分变化图

3.3　本章小结

在这一章中，从项目规划、实验过程两个方面介绍了我们的项目，首先介绍了项目进行的规划，包括数据集的构建、模型的选择、评价指标的选择等，分析了项目的前期工作。然后阐述了项目的具体实验过程，介绍了项目所使用的脚本和代码仓库，展示了模型在训练过程中的变化和趋势，并对比了经过训练获得的模型在验证集上的表现，证明了模型的微调方法是正确且有效的。

第四章　实验结果及分析

4.1　基线对比模型选择

4.1.1　百度翻译

百度翻译是百度旗下的新一代 AI 大模型企业翻译平台，我们调用百度翻译提供的通用文本翻译 API，将源语言设置为“wyw”、目的语言设置为“zh”进行调用，调用脚本如图4-1所示：

```
def main(text):
    url =
        "https://aip.baidubce.com/rpc/2.0/mt/texttrans/v1?access_token="
        + get_access_token()
    ans = []
    for str1 in text:
        payload = json.dumps({
            "from": "wyw",
            "to": "zh",
            "q": str1,
        })
        headers = {
            'Content-Type': 'application/json',
            'Accept': 'application/json'
        }

        response = requests.request("POST", url, headers=headers,
            data=payload)
        ans.append(response.text)
        print(response.text)
    return ans

def get_access_token():
    """
    使用 AK，SK 生成鉴权签名（Access Token）
    :return: access_token，或是None(如果错误)
    """
    url = "https://aip.baidubce.com/oauth/2.0/token"
    params = {"grant_type": "client_credentials", "client_id": API_KEY,
        "client_secret": SECRET_KEY}
    return str(requests.post(url,
        params=params).json().get("access_token"))
```

图 4-1　百度翻译 API 调用脚本

4.1.2 【随无涯】翻译模型

【随无涯】是一个 huggingface spaces + streamlit 的古文阅读应用（含海量书籍),其作者在 hugging face 上发布了这个应用的源翻译模型【raynardj/wenyanwen-ancient-translate-to-modern】[1]，我们选择使用这个模型作为对比基线模型，使用作者推荐的运行参数在测试集上运行，参数及调用脚本如图4-2所示：

```
def inference(text):
    tk_kwargs = dict(
      truncation=True,
      max_length=128,
      padding="max_length",
      return_tensors='pt')

    inputs = tokenizer([text,],**tk_kwargs)
    with torch.no_grad():
        return tokenizer.batch_decode(
            model.generate(
            inputs.input_ids,
            attention_mask=inputs.attention_mask,
            num_beams=3,
            max_length=256,
            bos_token_id=101,
            eos_token_id=tokenizer.sep_token_id,
            pad_token_id=tokenizer.pad_token_id,
        ), skip_special_tokens=True)
```

图 4-2 【随无涯】运行参数

4.1.3 mengzi-t5-base

T5 模型作为一个经典的预训练模型，被广泛的用来作为各种论文中的基线对比模型，然而原本由谷歌发布的 T5 预训练模型在中文语料下的表现不尽如人意，因此我们选择了澜舟科技（Langboat）发布的孟子系列预训练模型中的 mengzi-t5-base[zhang2021mengzi]，即孟子中文 T5 预训练模型。mengzi-t5-base 生成模型与 T5 结构相同，不包含下游任务，只有无监督数据训练。因此我们将 mengzi-t5-base 在与 ChatGLM3-6B 相同的训练集上进行微调，使用微调后的模型作为基线对比模型，微调参数如图4-3所示：

1 【随无涯】的源模型的 Hugging Face 主页.https://huggingface.co/spaces/raynardj/duguwen-classical-chinese-to-morden-translate.

```
model_params = {
    "MODEL": "/home/wbw/T5/model/Langboat:mengzi-t5-base",  # model_type: t5-base/t5-large
    "TRAIN_BATCH_SIZE": 8,  # training batch size
    "TRAIN_EPOCHS": 5,  # number of training epochs
    "LEARNING_RATE": 1e-4,  # learning rate
    "MAX_SOURCE_TEXT_LENGTH": 128,  # max length of source text
    "MAX_TARGET_TEXT_LENGTH": 128,  # max length of target text
    "SEED": 42,  # set seed for reproducibility
  }
```

图 4-3　mengzi-t5-base 微调参数

```
python finetune.py \
  --model_name_or_path $MODEL \
  --data_path $DATA \
  --bf16 True \
  --output_dir output_qwen \
  --num_train_epochs 5 \
  --per_device_train_batch_size 2 \
  --per_device_eval_batch_size 1 \
  --gradient_accumulation_steps 8 \
  --evaluation_strategy "no" \
  --save_strategy "steps" \
  --save_steps 1000 \
  --save_total_limit 10 \
  --learning_rate 3e-4 \
  --weight_decay 0.1 \
  --adam_beta2 0.95 \
  --warmup_ratio 0.01 \
  --lr_scheduler_type "cosine" \
  --logging_steps 1 \
  --report_to "none" \
  --model_max_length 512 \
  --lazy_preprocess True \

  --gradient_checkpointing \
  --use_lora
```

图 4-4　Qwen1.5-1.8B 的微调脚本

4.1.4　Qwen1.5-1.8B

使用通义千问发布的 Qwen1.5-1.8B 模型进行微调作为基线模型[1]，使用相同的训练集和 LoRA 方法进行微调[2]，微调脚本如图4-4所示：

4.2　超参数的选择

ChatGLM3-6B 模型在执行生成任务时的参数主要有 Top_p、Top_k、$Temperature$ 三个，模型在生成输出序列，会根据根据输入序列来预测输出的每一个 token，每

1　Qwen1.5-1.8B 的源模型的 Model Scope 主页.https://modelscope.cn/models/qwen/Qwen1.5-1.8B/summary.

2　Qwen1.5-1.8B 的微调脚本的 GitHub 仓库.https://github.com/QwenLM/Qwen.

次模型都会生成一个可能的 token 概率分布列表，表示模型对后续的 token 预测的概率，预测过程如图4-5所示。

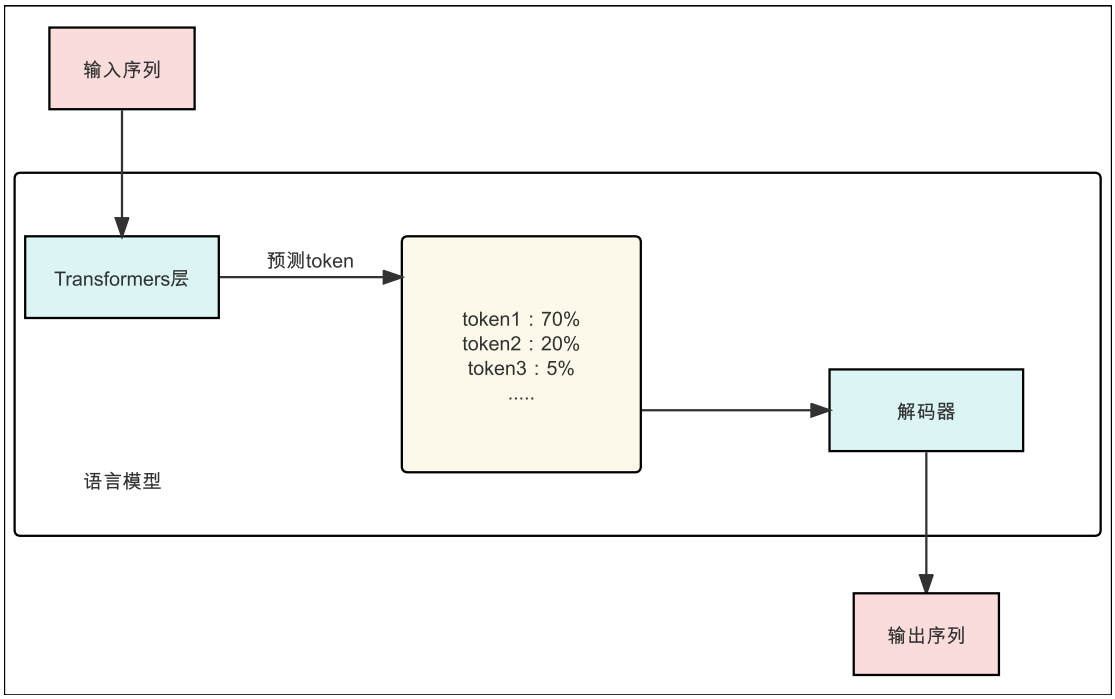

图 4-5　ChatGLM3-6B 选择 Token 的过程

在选择 token 的过程中，模型会根据采样策略选择不同的 token，而调整采样策略就可以对模型输出的多样性、准确性和质量进行调控，以满足不同的需求和场景。而参数 Top_k、Top_p、$Temperature$ 可以影响模型采样的策略进而影响模型的输出，Top_k 采样是一种基于动态词汇集的采样方法，使用 Top_k 采样的模型会从概率排名前 k 的 token 集合中进行选择，这样允许其他概率比较高但是并非最高的 token 也有机会被选中，增大了 token 选择的随机性和变化性，在很多情况下，Top_k 抽样带来的随机性有助于提高生成质量。

Top_p 采样与 Top_k 相似，在每一步，$Top-p$ 采样仅从累积概率超过阈值 p 的最小 token 集合中进行随机采样，忽略其他低概率的 token。这种方法聚焦于概率分布的核心部分，而忽略了尾部部分，以避免采样到不合适或不相关的单词。这样一来，模型能够保留一些有趣或有创意的单词，使得生成结果更具吸引力和相关性。Top_p 策略通常和 Top_k 策略一起使用，模型会在这两种策略中较小的那个 token 集合中进行采样。Top_k 和 Top_p 的值越低，则模型生成的输出越稳定，但是会导致结果十分无聊，在文言文-现代文翻译任务中，可能会使输出译文过于死板，失去文本在文学上的流畅度，降低阅读体验。而 Top_k、Top_p 设置

表 4-1　不同的超参数在验证集上的分数

参数设置/指标	Bleu-4	rouge-1	rouge-2	rouge-l
Top_p=0.7，Temperature=0.95，未设置 Top_k	30.4521	50.2036	26.0271	50.791
Top_p=0.40，Temperature=0.95，未设置 Top_k	30.8018	50.8867	26.617	51.4093
Top_p=0.35，Temperature=0.95，未设置 Top_k	**30.8327**	**50.8768**	**26.4786**	**51.4007**
Top_p=0.20，Temperature=0.95，未设置 Top_k	30.0231	50.1054	25.9861	50.1063
Top_p=0.35，Temperature=0.60，未设置 Top_k	31.645	51.6325	27.3558	52.2489
Top_p=0.35，Temperature=0.50，未设置 Top_k	31.5085	51.617	27.2271	52.2474
Top_p=0.35，Temperature=0.01，未设置 Top_k	**31.5728**	**51.7012**	**27.2839**	**52.3049**
Top_p=0.35，Temperature=0.60，Top_k=5	31.3918	51.5079	27.1022	52.1111
Top_p=0.35，Temperature=0.60，Top_k=3	31.7543	51.8071	27.4401	52.3731
Top_p=0.35，Temperature=0.60，Top_k=2	31.2762	51.2558	27.0408	52.0612
Top_p=0.35，Temperature=0.55，Top_k=3	31.7561	51.7932	27.426	52.3624
Top_p=0.35，Temperature=0.50，Top_k=3	**31.7664**	**51.7983**	**27.4502**	**52.3663**

的值过高，模型的输出会趋近去随机，虽然这样能提高输出和随机性但是有可能会大幅度降低准确性，失去文本应有的语义，降低输出译文的质量。

Temperature 采样是用于调节 Softmax 函数输出分布的参数。在生成文本时，Softmax 函数将模型输出转换为概率分布，用于选择下一个词汇。通过调节 *Temperature* 参数，可以改变 Softmax 函数的输出分布的“平缓程度”。较高的 *Temperature* 值会使得概率分布更平滑，输出的 token 概率之间的方差越小，从而生成的文本更加多样化；而较低的 *Temperature* 值会使得概率分布更集中，生成的文本更加保守。

ChatGLM3-6B 的代码实现的应用场景为聊天机器人，而聊天机器人要求模型的输出文本应当有足够的随机性和趣味性，因此在其代码中超参数的默认取值为 Top_p=0.7、*Temperature*=0.95、不设置 Top_k。在文言文-现代文翻译任务上需要模型输出保持一定的质量和准确性，如此高的采样概率显然是没必要的，因此我们针对超参数在验证集上进行对比实验，通过按顺序调整 Top_k、*Temperature*、Top_p 来尝试获得最好的生成效果下的参数，实验结果如表4-1、单个参数的分数变化图如4-6、4-7、4-8所示：

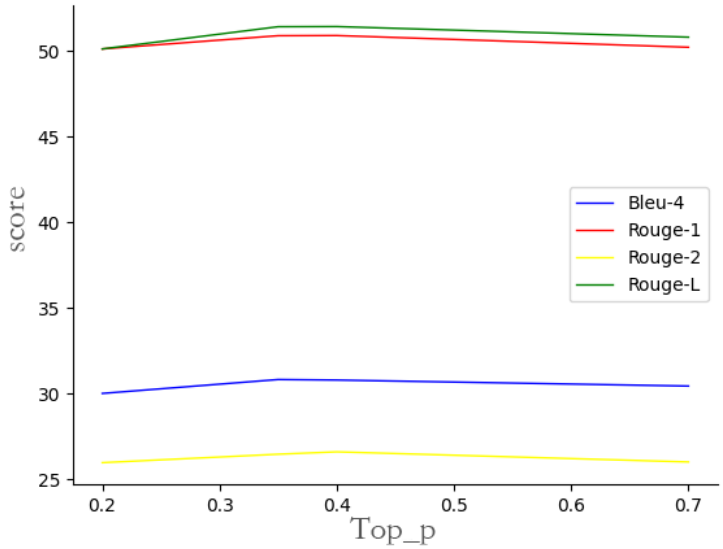

图 4-6 $Temperature = 0.95$，未设置 Top_k 时评分随 Top_p 取值变化

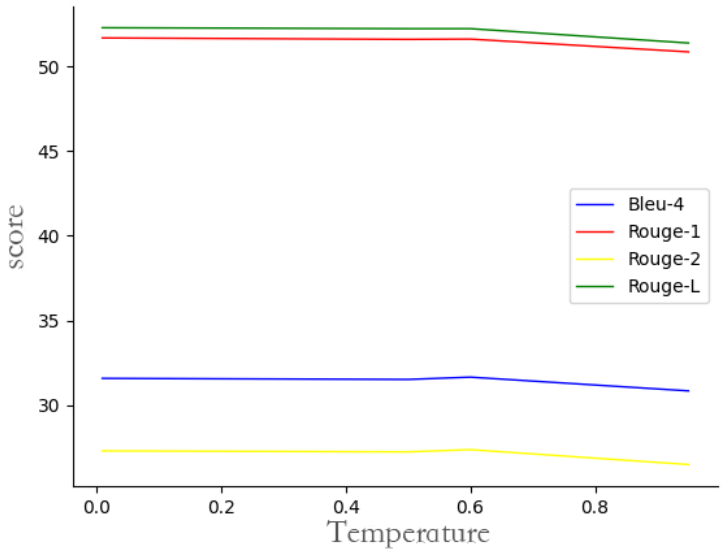

图 4-7 $TOP_p = 0.35$，未设置 Top_k 时评分随 $Temperature$ 取值变化

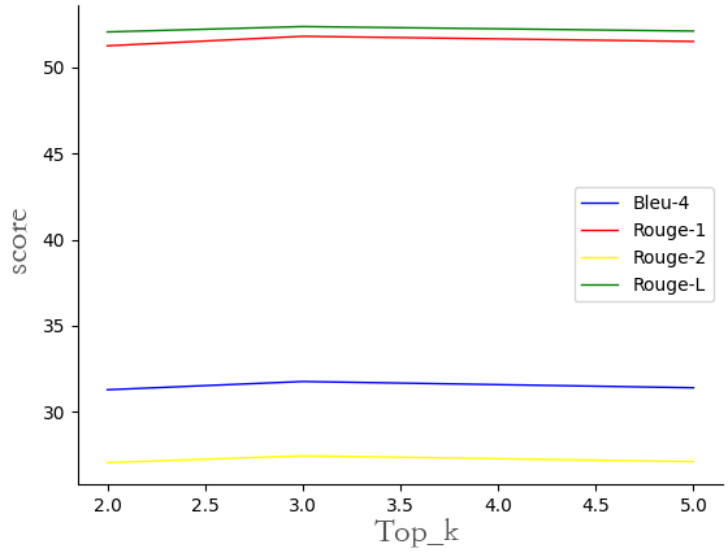

图 4-8　$TOP_p = 0.35$，$Temperature = 0.95$，评分随 Top_k 取值变化

经过实验，发现 $Top_p = 0.35$、$Temperature = 0.50$、$Top_k = 3$ 的最终评分最高，在后续的对比实验中也将对 ChatGLM3-6B 使用 $Top_p = 0.35$、$Temperature = 0.50$、$Top_k = 3$ 的超参数进行实验。

4.3　对比实验

4.3.1　测试集对比实验

在测试集上，（1）对比百度翻译，由于百度 API 的限额要求，因此无法翻译整个测试集，因此对百度翻译我们尽可能多的在测试集上翻译了 6000 余条，使用这 6000 条数据作为百度翻译的测试集。（2）对于【随无涯】翻译模型，我们使用作者推荐的超参数进行生成。（3）对于 mengzi-t5-base，使用进行微调后的模型进行生成。（4）对于 Qwen1.5-1.8B，使用 LoRA 方法进行微调后生成。实验结果如表4-2、4-3所示：

表 4-2　百度翻译与 ChatGLM3-6B 在测试集上的分数对比

模型/指标	Bleu-4	rouge-1	rouge-2	rouge-l
百度翻译	29.6460	49.4751	22.7385	50.1843
ChatGLM3-6B	29.8697	54.1254	29.4477	54.8262

表 4-3 【随无涯】、mengzi-t5-base 与 ChatGLM3-6B 在测试集上的分数对比

模型/指标	Bleu-4	rouge-1	rouge-2	rouge-l
【随无涯】翻译模型	26.1356	48.3732	18.7722	47.8400
chatGLM3-6B	33.7967	56.0598	30.8991	55.8341
mengzi-t5-base	30.9193	55.5103	38.0098	55.0582
Qwen1.5-1.8B	19.5024	49.5578	26.8344	49.0509

表 4-4 百度翻译、【随无涯】与 ChatGLM3-6B 的人工评分对比

模型/得分	评分者 1	评分者 2	评分者 3	评分者 4	评分者 5	评分者 6
百度翻译	2.83	2.73	3.49	3.02	3.15	**3.14**
【随无涯】翻译模型	2.84	2.20	**3.61**	2.73	3.29	2.85
chatGLM3-6B	**3.72**	**3.36**	3.43	**3.81**	**3.45**	2.92

微调后的模型比所有基线模型都具有更好的翻译能力，与百度翻译相比，Bleu 指标上分数接近，在 Rouge 分数上取得较高的分数（平均 +5Rouge）；与【随无涯】翻译模型相比，在 Bleu 和 Rouge 分数上取得了很大的优势（+7Bleu、平均 +10Rouge）；与 mengzi-t5-base 相比，在 Bleu-4 和 Rouge-2 上有较大优势（+3Bleu、+8Rouge-2），在 Rouge-1 和 Rouge-L 指标上也有轻微优势；与 Qwen1.5-1.8B 相比的优势也十分明显（+14Bleu、平均 +6Rouge）。

4.3.2 人工评分对比实验

在翻译领域，大语言模型的性能评估不能仅仅依赖于数据指标评分，Bleu 和 Rouge 指标无法完全捕捉语义准确性、流畅性和文化特征等与翻译质量相关的方面，进行主观的人的评分是至关重要的。人类评分者可以更好地理解翻译的上下文和意图，能够判断翻译是否符合特定场景的语言习惯和文化背景。他们的主观评价可以提供有价值的反馈，帮助进一步改进和优化大语言模型在翻译任务中的表现。因此，针对人工评分对比实验，我们邀请了 6 位在文学领域有一定造诣的同学作为评分者，选取了 184 条记录，对百度翻译、chatGLM3-6B、【随无涯】翻译模型输出的结果进行人工打分。我们从三个方面来考虑生成文本的质量：信（翻译文本与古代文本在语义上是否准确）、达（翻译文本是否流畅和清晰）、雅（翻译文本是否适当和文雅），对每条数据记录在 0 分到 5 分的范围内进行打分，其中 5 分表示“非常满意”，0 分表示“非常糟糕”。人工评分结果如表4-4所示：

人工评分结果中，有 4 位评分者给我们训练后的模型打出最高分，另外两位评分者的评分中，我们的模型和其他基线模型的得分也很接近，这证明了基于 ChatGLM3-6B 训练后的模型的优势。

4.4 本章小结

本章介绍了在对 ChatGLM3-6B 进行微调训练后的对比实验，首先说明了作为基线的对比模型的选择，分别选择了一个商业模型、一个应用模型、两个预训练模型作为对比。然后，介绍了在实验中使用的 ChatGLM3-6B 模型的生成代码中超参数的对比实验，分析超参数的不同选择在验证集上的不同表现。最后，我们对 ChatGLM3-6B 和基线模型进行了对比实验，验证了训练后模型在文言文-现代翻译任务中的优势，证明了基线模型选择和构建高效数据集的正确性。

第五章　总结与展望

在本论文中，我们针对文言文-现代文翻译任务，使用了 P-tuning v2 方法对 ChatGLM3-6B 模型进行微调，并在构建的文言文-现代文平行语料数据集上进行了实验。我们首先对现在的大语言模型研究背景和发展现状做了介绍，然后分析了目前国内外的主流预训练模型，之后对设计的微调实验进行了详细的规划和介绍，包括数据集构建、基础模型选择、微调方法介绍与选择、评价指标的选择与介绍以及实验环境的搭建。随后，我们描述了实验过程，包括数据准备、微调脚本、训练过程以及验证集表现。最终与基线模型进行了对比实验，我们的模型在测试集和人工评分方面均取得了较高的评分。

根据实验结果，我们可以证明：

- P-tuning v2 方法对 ChatGLM3-6B 模型进行微调在文言文-现代文翻译任务中取得了显著的性能提升。
- 基于 Erya 数据集构建的训练数据集在模型微调任务中具有较好的质量。
- 在各项对比实验中，微调后的 ChatGLM3-6B 模型表现出了优异的翻译质量和性能。
- 微调后的模型在人工评分实验中也取得了令人满意的结果，显示出了其翻译结果在人类阅读方面的优势。

虽然本文最终获得的模型取得了较高的评分，但是仍存在许许多多改进的地方，下面从模型选择、数据集构建、对比实验等方面进行分析，以便为后续研究做一些展望和引导：

- 模型选择方面：虽然 ChatGLM3-6B 预训练模型具有较好性能，但是受限于模型本身参数规模和可用的硬件资源，在实验中只进行了 6B 参数规模的预训练模型进行微调，未来希望可以在更大规模的预训练模型上进行实验，以尝试获得更好性能的模型。

- 数据集构建方面：Erya 数据集虽然在一定范围内具有较好的性能，但是中国的文言文古籍何其广阔，文言文语法和句式又何其复杂，我们构建的数据集只能包括一些较为成熟、稳定的文言文语料数据，训练后的模型在与训练集语法类似的古文时性能尚可。但是一旦面对一些古老而难以阅读的文言文翻译任务时，模型的表现则未可知。因此希望能够继续扩大数据集，加入一些更古老、更复杂的文言文语料数据。
- 对比实验方面：在翻译领域展现出优异表现的模型数不胜数，虽然挑选的三个翻译模型有一定代表性，但是仍不能全面体现我们的模型的优势。同时对比实验的测试集也是基于较为稳定、成熟的文言文古籍的，不能体现模型在更复杂的文言文上的性能。未来可以增加更多的对比模型，并充实对比实验的测试集，加入更多、更全面的文言文数据。

通过持续的研究和实践，我们相信微调技术在大语言模型领域的应用将会不断取得突破性进展，为机器翻译领域的发展贡献更多的力量。

致　　谢

首先要感谢我的毕业设计的指导老师葛季栋老师，在开题、论文修改等方面给了我许多指导和帮助，我的毕业设计的完成离不开葛季栋老师的悉心指导。

其次要感谢软件学院的余彰恒学长，在我做实验的时候给了我许多指导和帮助，使我懂得如何去推进我的实验和计划。

其次，我要向我的组员、朋友、同学们表示感谢，感谢他们在日常生活学习中给予我的帮助。

感谢我的辅导员宋抟老师、张一品老师，感谢你们在我本科期间对我的关怀和指导，没有你们的关心、支持和帮助就没有我现在的成绩。

我要向我的家人表示感谢，感谢在普通的每一天所给予我的肯定和支持，感谢你们的支持和鼓励让我在学业和生活上都有所成长。

最后，感谢南京大学软件学院对我的培养和教育，让我度过了难忘的本科四年生活！