南京大学

本 科 毕 业 论 文

院　　系　软件学院

专　　业　软件工程

题　　目　文言文翻译模型微调优化的研究

年　　级　2021　学　　号　211870196

学生姓名　尹天禹

指导教师　葛季栋　职　　称　副教授

提交日期　2025 年 6 月 9 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：文言文翻译模型微调优化的研究）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：尹承禹

学号：211870196

日期：2025.6.9

南京大学本科生毕业论文（设计、作品）中文摘要

题目：文言文翻译模型微调优化的研究

院系：软件学院

专业：软件工程

本科生姓名：尹天禹

指导教师（姓名、职称）：葛季栋　副教授

摘要：

随着自然语言处理技术的发展，神经机器翻译成为当前主流方法之一。本文基于 Facebook 提出的 mBART-50 模型，构建并微调了一个用于文言文到现代白话文的自动翻译系统，旨在探索该类预训练模型在中文古今文翻译中的应用可行性，为古籍数字化和普及提供技术支撑。

论文首先综述了神经机器翻译的基础理论，重点介绍了 Seq2Seq 编码-解码结构及其注意力机制。随后，分析了 mBART-50 的多语言训练框架与目标语言控制策略。在数据方面，本文融合通古、尔雅数据集与从古诗文网爬取的自建语料，构建了包含上百万句对的文言—白话平行数据集，覆盖多种古文风格，满足文言文翻译训练的基本需求。

本文利用 Hugging Face Transformers 框架进行 mBART-50 的全参数微调，恰当的采用了余弦退火学习率调度、混合精度训练和梯度累计等策略提升训练稳定性和效率。同时比较不同超参数下的模型性能，评估环节引入 BLEU 与 BERTScore 进行自动性能测量，根据评估结果找出了更适合文言文翻译训练的参数值。为优化计算资源占用，本文进一步尝试了 LoRA 微调策略，在大幅减少可训练参数与显存消耗的同时，保持较为优良的翻译质量。

关键词：mBART-50；文言文翻译；全参数微调；LoRA 微调；BLEU；BERTScore

南京大学本科生毕业论文（设计、作品）英文摘要

THESIS: Research on the Fine-Tuning and Optimization of Models for Translating Classical Chinese Texts

DEPARTMENT: School of Software Engineering

SPECIALIZATION: Software Engineering

UNDERGRADUATE: Tianyu Yin

MENTOR: JiDong Ge

ABSTRACT:

With the development of natural language processing technologies, neural machine translation (NMT) has become one of the mainstream approaches. This thesis builds and fine-tunes an automatic translation system from Classical Chinese to modern vernacular Chinese based on the mBART-50 model proposed by Facebook. The aim is to explore the feasibility of applying such pretrained models to ancient-to-modern Chinese translation, thereby providing technical support for the digitization and popularization of classical texts.

The thesis first reviews the fundamental theories of neural machine translation, with a particular focus on the Seq2Seq encoder-decoder architecture and its attention mechanism. It then analyzes the multilingual training framework of mBART-50 and its target language control strategies. In terms of data, this work integrates the Tonggu and Erya datasets with a self-constructed corpus crawled from the Ancient Chinese Texts website, resulting in a Classical-to-modern Chinese parallel corpus containing over one million sentence pairs. This corpus covers a variety of classical writing styles and fulfills the basic requirements for training Classical Chinese translation models.

The mBART-50 model is fine-tuned using the Hugging Face Transformers framework, with appropriate strategies such as cosine annealing learning rate scheduling, mixed-precision training, and gradient accumulation adopted to enhance training stability and efficiency. Model performance under different hyperparameter settings is compared, and automatic evaluation metrics including BLEU and BERTScore are employed. Based on the evaluation results, the most suitable hyperparameters for Classical

Chinese translation are identified. Furthermore, to optimize computational resource usage, this study explores the LoRA fine-tuning strategy, which significantly reduces the number of trainable parameters and memory consumption while maintaining relatively high translation quality.

KEYWORDS: mBART-50; Translation of Classical Chinese; full fine-tuning; LoRA fine-tuning; BLEU; BERTScore

目　录

插图目录

表格目录

第一章　导论

1.1　研究背景及意义

浩如烟海的中华传统典籍，既是中华文化的载体，也是全人类共同的瑰宝。由于汉语的时代演变，现代读者往往难以充分理解古代作品，这制约了传统文化的继承与发扬，古籍文献的传承面临严峻挑战；同时，目前古汉语的翻译工作主要由专业人士进行，但这个过程耗时较长且需要投入较多人力。

近年来，借助深度机器翻译，研究工作开始着眼于在古汉语机器翻译中采用预训练策略。在这样的情形下，很多工作仅仅遵循了以英语为中心的预训练范式，忽视了古代汉语的特点，同时由于缺乏可靠的认证机制与流通渠道，无法获得大规模且高质量的平行语料且高质量的平行语料，这样的情况让模型训练时难以捕捉古文语境中的复杂语意，给文言文与现代文之间的机器翻译研究带来了诸多不便[1]。其他的问题还包括现有的机器翻译模型多聚焦于句子级翻译，忽视篇章逻辑和文化内涵的传递，如倒装句、虚词等异常导致译文生硬或失真[2]；文言文在不同朝代、作者和文体（如策论、诗词）中呈现多样化风格，现有模型难以灵活适配不同文学家的语言特色。

在这样的背景下，构建系统化的文言文翻译框架，结合可靠的大规模数据集和先进的大语言模型，就成为推动古籍数字化与文化传承的关键路径。

文言文翻译系统的开发能够加速古籍文献的数字化进程，降低公众接触传统文化的门槛。通过生成不同风格的文学作品，系统还可为现代文学提供灵感，推动文学形式的创新性转化。结合开源大语言模型的微调策略，探索文言文与白话文双向翻译的可行性，不仅能优化现有机器翻译的准确率，还可为低资源语言对的研究提供方法论参考。在出版与传媒领域，自动化翻译工具还可以大幅缩短古籍整理周期，降低出版成本，助力文化产业的数字化转型。

1.2 国内外研究现状

机器翻译研究离不开数据集的支撑，因此，平行语料库的构建是机器翻译领域的重要工作。Gu 等人[3]和 Sennrich 等人[4]试图使用无监督学习或元学习的方法来解决数据稀缺的问题。之后，Zhang 等人[5]提出了使用具有动态规划思想的无监督对齐算法来处理齐文言文和现代文数据。其中就有许多研究者发表了他们对于构建篇章级数据的观点。总体来说，目前相关领域对文言文-现代文机器翻译的研究尚显薄弱。本文与 Liu 等人[6]和 Zhang 等人持有相同观点，均认为当前文言文-现代文机器翻译的主要瓶颈在于缺乏大规模且高质量的平行语料库。

近年来，文言文与现代文之间的机器翻译研究取得了显著进展，研究者们针对两种语言体系的特性提出了多项创新方法，主要可归纳为以下几个方面:

一方面是基于语言学特性的算法优化：Yang[7]等人在白话文-古诗翻译任务中，提出采用无监督机器学习方法，通过语义对齐策略有效缓解了翻译不足与过度翻译问题；针对语言历时性演变带来的理解障碍，Chang 等人[8]创新性地在翻译模型中引入年代预测作为辅助任务，通过联合训练提升模型对文言文历史语境的理解能力；Dankers 等人[9]发现将成语分解为组合表达式进行建模，可显著提升 Transformer 模型对文言文成语的直译准确率；Zheng 等人[10]则提出局部注意力机制与复制机制相结合的方法，有效改善了文言实体（如人名、地名）的翻译质量。

另一方面是预训练语言模型的应用深化：Zhang 等人的研究表明，预训练语言模型（PLM）通过大规模无监督学习，能够有效补偿文言文-现代文平行语料的不足，尤其在捕捉古文语法结构方面展现出优势，这缓解了数据资源匮乏的问题；Yang 团队[1]构建了首个文言文专用预训练模型 Guwen BERT，并基于 Unified Language Model（UNILM）架构进行微调，其分层注意力机制可更好处理古文的特殊句式。

在 2024 年，华南理工大学深度学习与视觉计算实验室（SCUT-DLVCLab）推出了“通古”大模型[11]，该模型基于 Baichuan2-7B-Chat 构建，聚焦于文言文处理任务，比较好的推动了古文理解与生成领域的发展。

研究团队在数据构建方，提出了一种半自动标注策略，结合结构化与非结构化的古文语料，构建了涵盖多种任务类型的高质量指令数据集，显著提升了模

型对复杂语言现象的理解能力。训练过程中引入冗余度感知微调（Redundancy-Aware Tuning, RAT）方法，以有效缓解灾难性遗忘问题，增强模型对原始知识的保留能力；同时融合检索增强生成（Retrieval-Augmented Generation, RAG）技术，以外部知识支持生成过程，减少知识密集型任务中出现的幻觉（hallucination）现象。团队还设计了两阶段的指令微调机制：第一阶段在大规模指令数据上进行基础微调，第二阶段则在小规模、高质量的数据集上进行精细化微调，从而进一步增强模型的多任务泛化能力与实用性。

整体而言，通古大模型在文言文翻译及相关知识密集型任务中展现出显著性能提升，体现出较强的创新性与应用前景。

以上的这些研究突破了传统统计机器翻译的局限，形成了“语言学规则驱动 + 预训练模型增强”的技术范式，为后续研究提供了重要参考。未来可进一步探索篇章级语境建模、文化负载词量化表征等方向，推动文言文翻译向智能化、精细化发展。

在机器翻译领域蓬勃发展的同时，当前研究也面临着一些挑战：如高质量篇章级平行语料稀缺，现有开源语料多为句子级且需深度清洗，难以支撑神经机器翻译模型的训练需求。尽管数字化古籍库建设持续推进，但结构化标注数据仍显不足。

文言文与现代文在词汇、语法、句式及语气层面存在显著差异。在词汇上文言文多古语词而现代文含大量新词，语法上来说文言文结构复杂（如倒装、虚词活用），现代文则趋于简洁，从句式而言，文言文重排比修辞，现代文强调逻辑清晰，最后是语体风格上，文言文庄重正式，现代文通俗实用。这些差异导致语义映射困难，加剧翻译失真风险。

现有模型对历史上下文信息的利用不充分，难以处理文言文中高频出现的词汇歧义（如多义词）、代词回指与指代省略现象。例如” 之” 在不同语境中可指代人物、事件或起结构助词作用，需依赖篇章级语境解析才能准确翻译，而当前系统多局限于局部语义捕捉。

突破这些瓶颈需构建融合语言学规则与深度学习的新型框架，同时加强跨时代语境建模能力。

1.3 本文主要工作

本文主要介绍了文言文翻译系统的数据集构建和训练微调过程，以及文言文翻译系统前后端功能界面的设计和实现。从数据集构建开始，我们自主研发网络爬虫系统，从古诗文网等多个权威平台采集数万条古诗文数据，涵盖文言文及对应翻译、诗歌文章等丰富内容。同时整合包括尔雅、通古、汉语大辞典等经典文献数据库的资源，并对数据进行清洗和标准化整理，构建了一个可靠的大规模数据集。训练模型时，选用了 mBart-50 作为基础架构，并用上述构建的数据集进行全参数微调和 LoRA 微调，并对比了不同超参数下的模型性能，实现了文言文到现代汉语的翻译。最后，阐述了文言文翻译系统的开发分析过程以及前后端的介绍。

1.4 论文的组织架构

本文一共有七章内容，具体结构如下：

论文第一章：导论。简要回顾文言文翻译系统的国内外研究背景，阐明目前国内外现状，这最后说明本文的主要工作和组织结构。

论文第二章：文言文翻译系统相关架构和技术介绍。介绍了本系统在开发的过程中采用的训练微调方法和开发框架。

论文第三章：文言文-现代文平行语料库的构建和处理。说明了本文研究中采用的文言文-现代文平行翻译语料的来源，以及文言文和现代文语料的数据清洗方法和步骤，同时展示了各个数据集的训练集、验证集和测试集的数据分布情况，为后续的实验和分析提供了坚实的基础。

论文第四章：介绍模型的配置和微调方法。介绍了所采用的 mBART-50 多语言预训练模型的初始化方式、训练配置和整体微调流程。通过制定不同的训练参数、批量大小和优化策略进行实验，以探究最适合文言文翻译训练的参数。同时，本章还尝试引入参数高效微调方法 LoRA，以缓解全参数微调带来的资源负担，探讨其在本任务中的可行性与拓展性。

论文第五章：文言文翻译系统训练的结果以及性能评估。介绍了所选用的 BLEU、BERTSCORE 等自动评价指标，并呈现不同微调策略下的评分对比结果。同时给出样例展示呈现了文言文输入与生成结果的对照，辅助直观理解模型能

力。

论文第六章：介绍系统的设计实现，包括前后端的实现过程和功能。

论文第七章：总结本次研究的主要研究工作，指出目前存在的不足，并对未来的优化方向和工作进行展望。

第二章　文言文翻译系统相关技术介绍

随着神经网络技术的迅猛发展，神经机器翻译已经成为目前主流的机器翻译方法。而以 Transformer 为核心的编码器-解码器的架构，不仅在多语言翻译任务中取得成效，也为低资源语言的建模提供了可能。mBART-50 作为 Facebook 提出的多语言预训练模型，具备多种语言相互翻译的能力，并通过自监督训练提升迁移性能。本章将重点介绍神经翻译系统模型、Transformer 模型架构、多语言预训练模型 mBart-50 的工作机制，以及微调方法，为后续实验部分奠定技术基础。

2.1　文言文与现代汉语的语言特征对比

文言文的词汇主要来自古代汉语，许多词汇在现代汉语中很少使用甚至完全消失。同时其语言风格简洁凝练，句式灵活，往往省略主语、谓语成分，语义依赖上下文。这种表达方式给机器翻译带来了很大的挑战。与之相比，现代汉语更加趋向于实用主义，局势结构相对固定，词语意义更为单一。因此，在文言文翻译任务中，模型不仅需要掌握基础的词义转换能力，还必须具备一定程度的上下文理解、语境重建和文化知识补全能力。这种跨时代、跨语言风格的翻译形式，在技术实现上更接近“风格迁移”与“语义生成”任务的结合，属于当前自然语言处理领域中较为复杂的一类问题。

2.2　神经机器翻译基本原理

神经机器翻译是一种基于深度学习的端到端自动翻译方法，其基本原理是模仿人脑神经元的工作方式，构建一个由大量神经元相互连接而成的神经网络，用于处理自然语言处理任务。神经机器翻译模型的核心思想是：利用一个编码器-解码器结构,将源语言句子编码为一个高维语义向量，再由解码器逐步生成目标语言的翻译结果。神经机器翻译的目标是解决传统机器学习算法在处理自然

语言时的不足之处，使机器能够更加准确地理解和学习自然语言。

最早的神经机器翻译系统通常基于循环神经网络（RNN）结构，编码器将源语言序列逐字处理并压缩为一个固定长度的上下文向量，解码器则基于该向量逐步生成目标语言词语。然而，由于固定长度向量难以容纳长句子的全部语义信息，这种方法在处理长文本或复杂语句时效果有限。随着注意力机制的提出，允许模型在生成目标词语时，动态关注语言序列中的不同部分。极大的提升了翻译质量，并成为神经机器翻译的标准组件。

到了 Transformer 框架提出，神经机器翻译进入新的发展阶段。Transformer 放弃了传统的循环结构，完全基于自注意力机智，实现了并行化训练和更长距离的依赖建模能力。

总体而言，神经机器翻译的优点包括：翻译质量高、系统结构简洁、具备端到端训练能力；但也面临着如低资源语言翻译困难、对训练数据依赖性强等挑战。因此，研究如何在低资源场景下提高神经机器翻译模型的迁移能力，仍是当前的重要研究方向之一。

2.3 Transformer 模型

Transformer 是一种用于自然语言处理（NLP）和其他序列到序列（sequence-to-sequence）任务的深度学习模型框架，它在 2017 年由 Vaswani 等人首次提出。Transformer 架构引入了由 Bahdanau 提出的自注意力机制（self-attention mechanism）[12]，这是一个关键的创新，让其可以在处理序列数据时变现更加优异。Transformer 模型架构如图 2-1 所示,Transformer 模型有一些重要组成部分和特点，Transformer 通常包括一个编码器用于处理输入序列和一个解码器用于生成输出序列，这是其适用于序列到序列的任务，如机器翻译。Transformer 的每层编码器均有多头注意力和前向神经网络层构成。Transformer 中的自注意力机制被扩展为多个注意力头，每个头可以学习不同的注意权重，以更好地捕捉不同类型的关系。多头注意力允许模型并行处理不同的信息子空间。

并且每个自注意力机制和前向神经网络层后面都紧跟着残差连接和层归一化（Residual Connections and Layer Normalization），这些技术有助于减轻训练过程中的梯度消失和爆炸问题，使模型更容易训练。Transformer 还有一个关键的

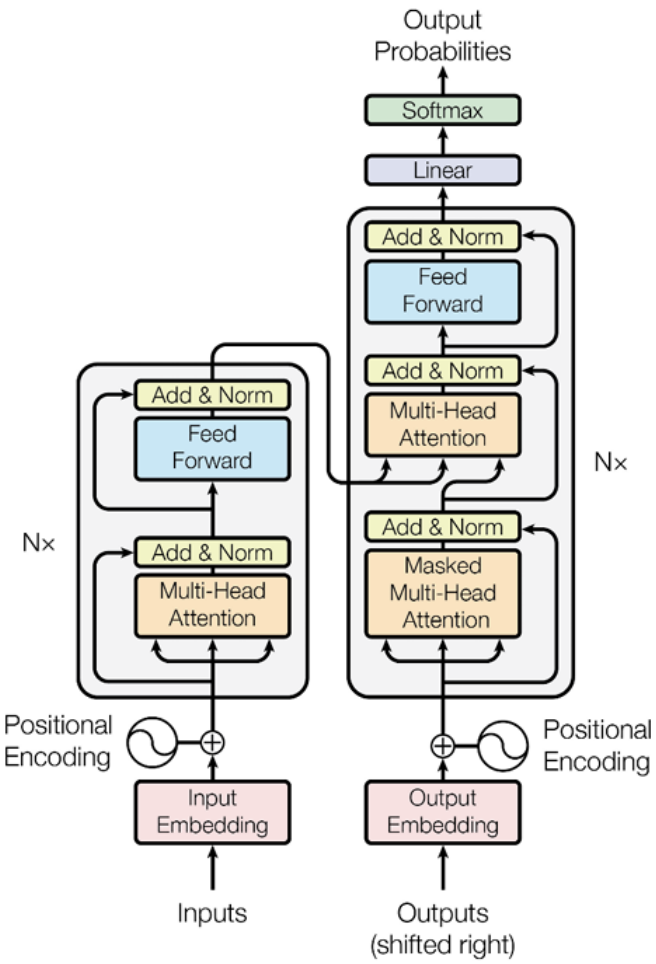

图 2-1　Transformer 模型架构

技术是位置编码方法，由于 Transformer 没有内置的序列位置信息，它需要额外的位置编码来表达输入序列中单词的位置顺序。其通过将位置信息嵌入到输入向量当中，让模型可以理解序列中不同位置的顺序关系，从而更好地处理序列数据。

2.4　mBART-50 预训练语言模型介绍

mBART-50 是 Facebook 提出的多语言序列到序列预训练模型，基于 BART 架构并扩展至约 50 种语言。与原始的仅含英文或少数语言的模型不同，mBART-50 在大规模多语言单语语料上进行去噪自编码（denoising autoencoding）预训练。

mBART-50 采用标准的 Transfomer 编码器-解码器架构，与神经机器翻译的架构相同。其通常使用与 BART-large 相同的配置，如：12 层编码器、12 层解码器，每层多头注意力，隐藏层维度 1024，前馈网络维度 4096。设计允许模型能在

晕训练和微调的过程中同时具备双向和单向的上下文建模能力。同时，Facebook 在论文中提到将 mBART-50 的语言种类翻倍到 50 种。

在预训练任务上，模型使用多语言去噪预训练目标。具体而言，预训练时先对单语文本进行噪声处理：一是对原始句子的顺序进行打乱（随机置换句子位置），二是采用填空式掩码（random span masking）策略，将连续的一段文本替换为单个特殊掩码标记通常会随机选择约百分之 35 的单词进行掩码，其中每个掩码段的长度服从泊松分布（λ≈3.5）。模型的任务是从噪声文本中恢复原始文本，即最大化原文在给定噪声输入下的条件概率。这种预训练方式与 BART 模型相似，通过学习对抗各种噪声，使 mBART-50 在各语言上获得深层语义理解能力。

实际翻译任务过程中，由于模型同时处理多种语言，输入输出中都需要显式指定语言。通常做法是在源语言句首添加一个语言 ID 标记，在目标句首也添加一个对应的语言标记。Hugging Face 实现中要求将输入格式化为 [lang_ code] 源文本 [eos]，输出为 [lang_code] 目标文本 [eos]，其中 lang_code 表示语言代码（如 en_ XX、zh_ CN 等），[eos] 为结束符。在推理时，可以通过强制设置首个生成标记为目标语言 ID 来指定翻译方向。这种设计使得同一个模型可以接受任意支持语言的输入，输出指定语言的翻译结果。

mBART-50 的预训练使其在进行下游翻译任务的微调时具备很强的通用性。在少量双语数据的情况下，通过在对应语言对上微调，模型可以较快适应新的翻译任务。研究表明，使用 mBART 作为初始化可以为低资源语言翻译带来大幅性能提升（可达 +12 BLEU）。在文言文翻译的场景中，由于古文与现代文往往缺乏大规模的平行语料，可以将文言文视作一种“新语言”引入模型：比如使用现有的古汉语与现代汉语对照语料或同义替换数据对 mBART-50 进行微调，并利用其多语言知识迁移能力。这样可以有效地将文言文句子翻译为白话文。

综上而言，mBART-50 将 Transformer 编码-解码架构与大规模多语言去噪预训练相结合，通过显式的语言标识机制和统一的输入输出格式，实现了在多语言环境下的高效迁移学习能力（例如 Facebook 将其拓展到 50 种语言）。这一模型不仅能在有充足双语数据的情况下微调出高质量的翻译系统，而且对于低资源或新语种（如文言文）翻译同样具有良好的泛化能力。

2.5 评估指标介绍

2.5.1 BLEU

BLEU 是 IBM 在 2002 年提出的[13]，用于机器翻译任务的评价方法。BLEU 分数是通过比较机器翻译的输出和人工翻译的参考译文的 n-gram 相似度来计算的。BLEU 主要侧重于衡量机器翻译输出与参考翻译之间的相似程度，着重于句子的准确性和精确匹配。BLEU 通过计算 N-gram 的匹配程度，来评估机器翻译的精确率。BLEU 被广泛应用于各种机器翻译系统，包括但不限于神经网络机器翻译系统。其使用公式（2-1）来计算，其中 c 表示一个句子，其中的 n-gram 在生成的翻译中出现了 Count 次，在实际参考翻译中最多出现了 Max_ Ref_ Count 次，$Count_{clip}$ 取 Count 和 Max_ Ref_ Count 的最小值。

$$BLEU_n = \frac{\sum_{c\in\text{candidates}} \sum_{n\text{-gram}\in c} \text{Count}_{\text{clip}}(n\text{-gram})}{\sum_{c'\in\text{candidates}} \sum_{n\text{-gram}'\in c'} \text{Count}_{\text{clip}}(n\text{-gram}')} \tag{2-1}$$

2.5.2 BERTScore

BERTScore 是一种用于自然语言处理的评估指标[14]，主要用于衡量生成文本（如机器翻译或文本生成）的质量。它基于 BERT 预训练模型，通过计算生成文本和参考文本之间的语义相似度来进行评估。具体来说，BERTScore 利用 BERT 生成的词嵌入，计算候选句子和参考句子之间的余弦相似度，并通过精确率、召回率和 F1 得分来综合评估文本生成任务的质量。与传统的评估指标（如 BLEU 或 ROUGE）相比，BERTScore 能够更好地捕捉语义信息，具有更高的相关性。

这种方法允许评估不仅基于表面形式的匹配，而且还能捕捉到语义上的相似性。BERTScore 计算精确度（P）、召回率（R）和 F1 分数，这些都是评估不同语言生成任务的有用指标。

其具体计算公式如下：

设：

x=(x1,x2,...,x_n)：候选句子的 token 序列

y=(y1,y2,...,y_n)：参考句子的 token 序列

$\phi\left(x_i\right)$: 第 i 个 token 子 BERT 输出中的向量表示（一般取最后一层或加权组合层）

BERTScore 会给予 token 之间的 cosine 相似度构建匹配关系。

首先根据公式（2-2）计算精确度，对于候选句中每个 token，找到参考句中与之最相似的 token，计算其最大 cosine 相似度并求平均。

$$Precison = \frac{1}{n}\sum_{i=1}^{n}\max_{j\in[1,m]}\cos\left(\phi\left(x_i\right),\phi\left(y_j\right)\right) \tag{2-2}$$

然后根据公式（2-3）对于参考句中每个 token，找到候选句中与之最相似的 token，计算其最大 cosine 相似度并求平均得出召回率：

$$Recall = \frac{1}{m}\sum_{j=1}^{m}\max_{i\in[1,n]}\cos\left(\phi\left(y_j\right),\phi\left(x_i\right)\right) \tag{2-3}$$

最后根据公式（2-3）综合精确率与召回率，计算 F1 即 BERTScore 分数：

$$\mathrm{F1} = 2\cdot\frac{\mathrm{Precision}\ \cdot\ \mathrm{Recall}}{\mathrm{Precision}\ +\ \mathrm{Recall}} \tag{2-4}$$

BERTScore 的评估分数范围在 0 到 1 之间，分数越接近 1 表示生成的句子越解决参考句子，模型的性能也就越好。

2.6 LoRA 微调

LoRA[15]，即 LLMs 的低秩适应，是参数高效微调最常用的方法。Aghajanyan 的研究表明[16]：预训练模型拥有极小的内在维度 (instrisic dimension)，即存在一个极低维度的参数，微调它和在全参数空间中微调能起到相同的效果。在训练过程中，过参数化的神经网络权重更新往往存在低秩结果，因此，训练不必对完整的参数矩阵进行微调。LoRA 的本质就是用更少的训练参数来近似 LLM 全参数微调所得的增量参数，从而达到使用更少显存占用的高效微调。

Lora 的重参数化如图（2-2）所示，对于预训练权重矩阵 $W_0 \in \mathbb{R}^{d\times k}$，用一

个低秩分解来表示参数更新 ΔW，即：

$$W_0 + \Delta W = W_0 + BA \quad B \in \mathbb{R}^{d\times r}, A \in \mathbb{R}^{r\times k} \qquad r \ll \min(d, k)$$

训练过程中冻结参数 W_0，仅训练 A 和 B 中的参数。对于 $h = W_0 x$，前向传播过程变为：

$$h = W_0 x + \Delta W x = W_0 x + BAx$$

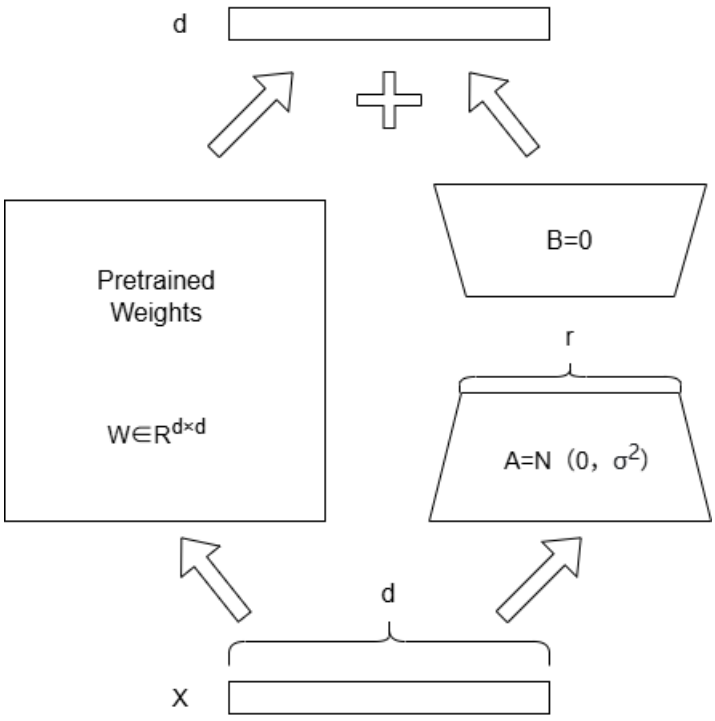

图 2-2　Lora 重参数化

LoRA 在计算资源受限的条件下，实现了对大型语言模型的快速适应。该方法通过在部分网络层中引入低秩可训练矩阵，仅更新极少数参数，从而大幅减少了训练参数量，显著降低了内存消耗，加快了研究的训练速度，同时在推理阶段不引入任何额外计算开销，具备较为良好的部署友好性。

当然，LoRA 也存在一定的局限性。在处理结构复杂、知识迁移跨度较大的任务时，其表示能力可能不如全参数微调。LoRA 的性能在一定程度上对秩参数 r、注入位置与缩放因子等超参数较为敏感，缺乏统一调优策略，还需要结合具体任务进行经验性调整。

简而言之，LoRA 作为一种兼具效率与实用性的微调机制，在实际应用中展

现出良好的性能-资源平衡特性，已成为当前大语言模型微调的重要技术路径之一，特别适用于资源受限环境、多任务适配与快速原型验证等场景。

2.7 本章小结

本章旨在为后续的研究奠定理论基础，阐述了本文工作研究的文言文和现代汉语的特征对比，以及主要采用的模型 mBart-50 涉及到的架构 Transfomer 模型。还有介绍了用于评估机器翻译质量的指标，其为后续实验结果的客观评价提供了有力的支撑。

第三章　数据集构建与预处理

3.1　平行数据库需求

本论文的训练基于 Seq2Seq 模型，这类模型的核心任务是要建立起输入序列与输出序列之间的映射关系。Seq2Seq 模型对训练数据的质量和规模有较高要求，它需要依赖高质量的双语平行语料来实现有效学习。

平行语料数据集，是指由两种语言的一一对应文本构成的数据集，每对语句在语义上应保持一致，尽管长度可变，但必须严格对齐。若文本未能准确配对，将影响模型对翻译规律的学习，导致输出结果语义偏差甚至失真。因此，构建一个高质量的文言文-现代汉语平行语料库，是本研究开展文言文自动翻译的关键前提。

在本论文中，一个平行语料数据集由两部分组成：一部分为文言文文本，另一部分为其对应的现代汉语译文。两部分数据按顺序一一对应，位于相同位置的两条语句互为翻译关系。虽然说文言文与现代汉语同属汉字体系，但在语法结构、词汇表达和语言风格等方面差异显著，因而在模型中可视作两种独立语言进行建模与训练。确保语料对齐准确、语义一致，是提升 Seq2Seq 模型翻译质量的基础保障。

3.2　数据来源与筛选

平行语料的稀缺是文言文翻译所面临的一个难题，目前已经有一些研究开源了文言文-现代文的平行语料库。我们将部分基于开源的数据集和我们自己开发的爬虫数据集进行研究。中国人民大学的郭歌扬、杨家荣等人提出了工具集“尔雅 (Erya)”[17]，通过收集和清洗多种来源的古文材料，形成了迄今规模最大的文言文语料资源。尔雅数据集采用的去噪方法包括如去除非汉字字符（如阿拉伯数字与英文字母），将繁体字转化为简体字。统一标点符号（如「转换为 “）。同时鉴于不同来源的语料库可能互相重叠，尔雅还使用了 minhash 算法进行去

重，当相似度高于 0.5 时便舍弃其中一条。尔雅形成了如图（3-1）的数据集，分别对应文言文和现代文。尔雅数据集主要由记录朝代更替、重大事件和重要人物的历史文献、先秦到汉代（公元前 3 年之前）的上古汉语、三国到宋代（公元 4 年到公元 12 年）的中古汉语、元代到清代（公元 13 年到公元 19 年）的早期现代汉语、包括诗歌、散文、哲学作品和文学批评的文学作品以及融合了古汉语和准现代风格，具有独特的文体和语言特征的小说组成。

图 3-1　尔雅数据集

我们还申请了华南理工大学通古项目的文白对齐语料 ACCN-INS，其由 400 万古籍指令微调数据组成，涵盖古文理解、生成、知识三个维度的共 24 类估计任务，包含 4020136 条指令数据，其中古汉语到现代汉语的翻译任务包含 4000000 条数据，标点符号恢复、命名实体识别、古诗创作等任务包含 14355 条数据。数据集总指令数 4020136 条，其中 4014355 条来自结构化文本，5781 条来自非结构化文本。平均指令长度 48.59 个字符，平均输出长度：68.96 个字符。ACCN-INS 数据集同样经过严格的清洗，质量也较为出色。

最后，我们自己通过开发爬虫程序爬取古诗文网的内容作为补充。在爬取过程中，为确保文言文和白话文在语义上为完整的句子，对于含有残缺或残篇的文本（如缺少句首句尾、句中断裂）进行了提出和合并。同时剔除了所有与句对翻译无关的内容，比如章节标题、诗文名称、作者署名、注释、赏析以及脚注等等。比如，爬取的古诗文条目常含有诗名和诗评，这些属于背景信息，不应纳入平行句对；同样，爬出的现代译文若夹杂其他语句片段，也需去掉。同时我们对古籍按篇章和文章进行创建文档，如下代码所示：

```python
# 解析具体古籍，并按篇章和文章创建文档
def book(baseurl, header, bookName, lastInfo, f_log):
    lastSection = lastInfo[1] if isinstance(lastInfo, tuple) else None
    lastChap = lastInfo[2] if isinstance(lastInfo, tuple) else None
    request = requests.get(url=baseurl, headers=header)
    time.sleep(0.5)
    bs = BeautifulSoup(request.text, 'lxml')
    chap_num = len(bs.select("body > div.main3 > div.left > div.sons > div"))
    flag = True
    for i in range(chap_num):
        chap_detail_list = str(bs.select("body > div.main3 > div.left > div.sons > div")[i])
        t_bookName = bookName
        if chap_num > 1:
            sectionName = re.findall(r'<strong>(.*)<\/strong>', chap_detail_list)[0]
            if lastSection is not None and lastSection != "" and lastSection != sectionName and flag:
                continue
            flag = False
            t_bookName = os.path.join(bookName, sectionName)
            if not os.path.exists(t_bookName):
                os.mkdir(t_bookName)
                f_log.write('###'+sectionName+'###\n')
                print(' 当前篇章: '+sectionName)
        chapID_chapName = dict(
            zip(
                re.findall(r'<a href="(.*?)">', chap_detail_list),
                re.findall(r'<a href=".*?">(.*?)<\/a>', chap_detail_list)
            )
        )flag2 = True
        for html, name in chapID_chapName.items():
        name = name.replace('/', '&')
        if flag2 and lastChap and lastChap != name: continue
        flag2 = False; chap_path = os.path.join(t_bookName, name)
        os.makedirs(chap_path, exist_ok=True); f_log.write(f'##{name}##\n'); print(f' 当前章节: {name}')
        chapter("https://www.gushiwen.cn" + html, header, chap_path); lastChap = None
```

图 3-2　爬虫代码示例

3.3 数据清洗与标准化

如下文算法一，我们做了一些数据的清洗和标准化。为了保证语料的一致性和规范性，我们将所有文本统一为简体中文。由于数据来源可能混杂繁体和简体字符，本文中采用统一为简体的策略，以减少字符基本的差异。借助开源工具 OpenCC 批量转换，将文言文和白话文中的繁体字转换为对应的简体字。同时，根据语言特性调整句子划分。对齐后的句子如果过长，可根据语义逻辑插入标点断句。如果过短或出现文言助词等附属句，则考虑与前后句合并。举个例子，若文言文句子只有一个停顿助词，可与下一句并为一个复合句。这个过程需要权衡句子长度，以保证训练数据的平滑性。

Algorithm 1 语料清洗与标准化算法

```
function CleanAndNormalizeCorpus(RawTextCorpus)
    CleanedCorpus ← [ ]
    for all Document ∈ RawTextCorpus do
        Document ← TraditionToSimplified(Document)   ▷ Step 1: 繁简转换
        Document ← RemoveInvalid(Document)   ▷ Step 2: 清除乱码与空文本
        if IsEmpty(Document) then
            continue
        end if
        Document ← NormalizePunctuation(Document)   ▷ Step 3: 标点规范化
        Sentences ← SplitIntoSentences(Document)   ▷ Step 4: 初步句子划分
        MergedSentences ← [ ]   ▷ Step 5: 合并短句，断开长句
        i ← 0
        while i < length(Sentences) do
            if IsTooShort(Sentences[i]) and IsAuxiliaryClause(Sentences[i]) then
                Merged ← MergeWithNext(Sentences, i)
                Append(MergedSentences, Merged)
                i ← i + 2
            else if IsTooLong(Sentences[i]) then
                SplitList ← SplitLongSentence(Sentences[i])
                Append(MergedSentences, SplitList)
                i ← i + 1
            else
                Append(MergedSentences, Sentences[i])
                i ← i + 1
            end if
        end while
        Append(CleanedCorpus, MergedSentences)
    end for
    return CleanedCorpus
end function
```

经过上面的清洗和规范化整理，最终得到了格式一致、编码统一的文本。将双语句对保存为标准格式（JSON 文件），以便后续自动化处理。所获得的高质量文言文-白话文平行语料将作为输入，用于对 mBART-50 模型进行序列到序列（Seq2Seq）微调训练。这些准备工作为模型提供了可靠的数据基础，使得微调后的翻译模型可以更准确地学习文言文和现代汉语之间的映射关系。

在本论文的训练设计中，数据集被划分为训练集、验证集和测试集，比例为 18:1:1，即分别占总数据的 90%、5% 和 5%。

这种划分策略是为了兼顾模型的学习效果、调参过程和性能评估：训练集 (90%) 将为模型提供充足的样本数据，以充分学习文本之间的语义规律和翻译特征。验证集 (5%) 在训练过程中用于评估模型在未见数据上的表现，从而辅助超参数调整与模型选择。测试集 (5%) 在训练完成后用于独立评估模型的泛化能力和最终性能。

在具体划分过程中，所有数据样本都采用随机方式选取且避免重复，以确保三个子集之间不存在交集。同时，划分过程保了持数据分布的均衡性，尽量避免某类语言风格或样式在某一子集中出现过度集中现象，从而提高训练和评估的科学性与可靠性。

3.4 本章小结

本章围绕文言文自动翻译系统所依赖的数据基础，详细介绍了平行语料的来源、构建、筛选与预处理流程。本文综合利用通古、尔雅等公开语料资源数据集和网络爬虫手段，从古诗文网等高质量平台采集了大量文言—白话对照文本，通过 requests 与 BeautifulSoup 等工具实现结构化信息提取，有效扩展了语料规模与风格覆盖范围。

在数据筛选与清洗时，本文采用了多轮去重、空值剔除、长度限制、乱码过滤等策略，确保了数据的质量稳定、格式统一。为适配神经机器翻译模型的输入要求，还完成了包括标点统一、繁简转换、Tokenization 与分句处理等预处理步骤，并针对模型训练过程中的特殊需求，对语料长度分布、句子对齐情况与文本风格多样性进行了分析评估。

通过本章的工作，最终构建了一个规模达百万级、风格丰富、语义对应明确

的文言—白话平行语料库，为后续模型训练、微调与性能评估提供了坚实可靠的数据支撑，也为推动古籍智能处理与自动翻译任务的研究实践奠定了基础。

第四章　模型设计与微调方法

4.1　环境准备

本论文中大模型训练环境：

• 在四张 NVIDIA V100 显卡上进行

• 操作系统为 Linux，通过 anaconda 构建虚拟环境部署

•CUDA 版本为 12.4

• 每张 GPU 的内存容量为 32GB

• 编程语言为 Python3.12

• 学习框架为 PyTorch2.5.1

•Transformers 库版本为 4.49.0

4.2　模型初始化与配置

mBART-50 是一种多语言序列到序列（Sequence-to-Sequence）预训练模型，它是“基于掩码噪声自编码”任务，在大规模多语语料上训练而成的。原始的 mBART-CC25 仅支持 25 种语言，mBART-50 通过扩展词表、随机初始化额外 25 种语言的嵌入，并在 50 种语言上继续预训练而得。这种预训练方式使模型学到了通用的语言表示，适用于翻译等下游任务。

mBART-50 的输入输出格式要求在句首添加语言标识符：源文本格式为 [源文本代码] 源文本 [eos]，目标文本格式为 [目标语言代码] 目标文本 [eos]。因此，在使用 mBART-50 时需要在分词器（Tokenizer）中指定源语言和目标语言。通过 MBart50TokenizerFast.from_ pretrained(..., src_ lang=”zh_ CN”, tgt_ lang=”zh_ CN”) 来设置中文（简体）作为源目标语言。分词后，模型的输入（input_ ids）对应源语言句子，输出（labels）对应目标语言句子。加载模型与分词器代码实现如图 4-1 所示：

```
# 2. 加载 mBART 模型和分词器
model_name = "facebook/mbart-large-50"
tokenizer = MBart50Tokenizer.from_pretrained(model_name)
model = MBartForConditionalGeneration.from_pretrained(
    model_name)
```

图 4-1　模型加载和分词器

这样即可获得预训练好的 mBART-50 模型和支持中文的分词器。

之所以选择 mBART-50 作为文言文翻译的基座模型，是因为该模型在多种语言上做了预训练，包括简体中文等等，还有是因为它的预训练使用了大规模的无监督语料，获得了丰富的语言知识，这对于语料稀缺的文言文翻译尤其有利。同时，一些研究指出，古汉语翻译任务语料量较少，传统的神经翻译难以直接学习复杂语法和语义；而“预训练 + 微调”范式在低资源翻译上更容易取得优异成果。因此，基于 mBART-50 的多语特性和强大预训练能力，可以更好地完成文言文翻译的任务。

4.3　微调训练流程

首先进行数据加载与构建，将并行语料（文言文句子-现代文句子对）保存为文件后，使用 Hugging Face 的 datasets 库加载，如图 4-2。

```python
# 1. 数据加载
def load_data(src_file, tgt_file):
    with open(src_file, "r", encoding="utf-8") as f_src, open
        (tgt_file, "r", encoding="utf-8") as f_tgt:
        src_lines = [line.strip() for line in f_src.readlines
            ()]
        tgt_lines = [line.strip() for line in f_tgt.readlines
            ()]
    assert len(src_lines) == len(tgt_lines), "源文件和目标文件的
        行数不一致！"
    return Dataset.from_dict({"source": src_lines, "target":
        tgt_lines})

# 加载数据集
train_dataset = load_data("train.src", "train.tgt")
valid_dataset = load_data("valid.src", "valid.tgt")
```

图 4-2　数据加载

构建好原始数据集后，需要对文本进行分词，定义预处理函数 preprocess_ function，对每个样本的源句子和目标句子分别进行分词，如图 4-3。

```python
    def preprocess_function(examples):

    inputs = tokenizer(examples["source"], max_length=128,
        truncation=True, padding="max_length")
    targets = tokenizer(examples["target"], max_length=128,
        truncation=True, padding="max_length")
    inputs["labels"] = targets["input_ids"]

    return inputs
```

图 4-3　分词

这里的 tokenizer 分别对输入输出进行分词，并将目标句子的 token id 放入返回结果的 labels 字段。处理后的 Dataset 将包含 input_ ids、attention_ mask、labels 等字段，可用于训练。

本文使用 Hugging Face 的 Seq2SeqTrainer 来管理训练过程。该类是专门为序列到序列任务设计的训练器，相比通用的 Trainer，它能自动处理序列生成过程中的特殊需求，如在评估时自动用模型生成输出并与标签比较来计算指标，并

自动对解码器的输入标签进行“右移”处理。Trainer 会自动迭代数据、计算损失并优化参数。整个流程中，每个步骤都对应文言文翻译任务：数据加载对应并行语料构建；分词和预处理函数将原始文本转换为模型能接受的 input_ ids/labels，本文设置如图 4-3：

```
    # 4. 训练设置
    training_args = Seq2SeqTrainingArguments(
    output_dir="./mbart-translation", # 模型保存路径
    evaluation_strategy="epoch", # 每个 epoch 进行评估
    learning_rate=3e-5,          # 学习率
    per_device_train_batch_size=16, # 每个设备的批量大小
    per_device_eval_batch_size=16, # 评估时的批量大小
    num_train_epochs=4,          # 训练的 epoch 数
    save_strategy="epoch",       # 保存策略
    save_total_limit=2,          # 最多保存几个模型
    predict_with_generate=True, # 预测时生成翻译结果
    gradient_accumulation_steps=2, # 累计梯度适应大batch
    fp16=True,                   # 启用混合精度训练
    gradient_checkpointing=True, # 节省显存
    lr_scheduler_type="cosine", # 余弦退火学习率
    weight_decay=0.01,           # 添加权重衰减
    max_grad_norm=1.0,           # 梯度裁剪
    logging_dir="./logs",
    logging_steps=100,
    report_to="none",
)

# 5. 训练
trainer = Seq2SeqTrainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_dataset,
    eval_dataset=tokenized_valid_dataset,
    tokenizer=tokenizer,
)
```

图 4-4　训练设置

4.4　训练参数与资源设置

本文中设置了四个训练轮次，对百万级数据来说是个适中值，既避免了过拟合的风险，又让模型充分收敛。同时设置 evaluation_ strategy=”epoch” 与 save_

strategy=”epoch”，每个 epoch 评估并保存一次模型，是对训练过程进行监控。初始组中学习率设置为 3e-5 这样一个温和的中值，适合稳定微调。而余弦退火学习率调度在训练后期下降较快，有助于模型收敛得更加精细（相比 linear 更加温和）。weight_ decay 设置为 0.01，这里适当的 L2 正则防止过拟合，同时梯度裁剪策略防止爆炸，保证训练的稳定性，此处为训练的稳定性和泛化能力提供了有力保障。开启了自动混合精度训练（fp16）, 在保证数值稳定的前提下降低计算资源开销，使模型在同等硬件条件下支持更大 batch size 与更高迭代效率.

4.5 参数比较

为进一步探究超参数对模型性能的影响，本文在原配置（epoch=4, 学习率=3e-5，batch size=16）基础上，改变学习率和批量大小进行了对比实验实验设置如表 4-1：

表 4-1 超参数设置表

参数	第一组	第二组	第三组
学习率	1e-5	5e-5	3e-5
批大小	16	16	8

4.6 LoRA 微调实现

随着大规模预训练模型参数不断膨胀，传统的全参数微调方式在训练资源、显存消耗和多任务迁移场景下逐渐显现出不小的成本。LoRA 是一种参数高效微调方法，其核心思想是在冻结原有模型全部参数的基础上，仅在部分线性变换矩阵中注入可训练的低秩矩阵，显著降低微调开销的同时仍能获得良好的任务适应性。

配置 LoRA 插入参数并注入模型如图 4-5：

```
    peft_config = LoraConfig(
    task_type=TaskType.SEQ_2_SEQ_LM,
    inference_mode=False,
    r=8,            # 低秩维度
    lora_alpha=32, # 缩放因子
    lora_dropout=0.1,
    target_modules=["q_proj", "v_proj"] # 指定注入注意力层
)

model = get_peft_model(model, peft_config)
model.print_trainable_parameters() # 输出可训练参数占比
```

图 4-5　LoRA 设置

task_ type=TaskType.SEQ_ 2_ SEQ _ LM：明确指定当前任务为序列到序列语言建模任务，与本文文言文到白话文的翻译任务结构一致，确保 LoRA 模块注入至 Encoder-Decoder 模型的合适位置。

r=8：表示低秩矩阵的秩为 8，即可训练矩阵 $4 \in \mathbb{R}^{8\times d_{\text{in}}}$, $\boldsymbol{B} \in \mathbb{R}^{d_{\text{out}}\times 8}$。该值是 LoRA 文献中广泛采用的推荐设定[15]，能够在保持参数压缩比的同时维持模型的表达能力。

lora_ alpha=32：用于控制低秩更新的缩放强度，最终残差为 $\Delta \boldsymbol{W} = \frac{\alpha}{r} \cdot \boldsymbol{BA}$，即比例为 4。该值是经过大量实验验证的经验性选择[15]，能在收敛速度与稳定性之间取得平衡。

lora_ dropout=0.1：在训练过程中对 LoRA 路径施加 0.1 的 Dropout，以提升泛化能力并减少过拟合，尤其适用于样本规模有限的翻译任务。

target_ modules=[”q_ proj”, ”v_ proj”]：指定 LoRA 模块仅注入至 Transformer 注意力机制中的 query 和 value 投影层。研究表明[15]，这两个子模块对注意力权重的影响最为关键，优先注入可在控制参数规模的同时获得性能收益。

LoRA 在不牺牲太多性能的前提下，实现了显著的训练效率与存储成本优势，尤其适合资源受限或需快速适配多任务（如多种语言风格）的应用场景 LoRA 为本任务提供了一种高效、低成本的微调路径，特别适合后续训练的优化和多任务切换。

4.7 本章小结

本章围绕文言文自动翻译系统中的核心建模部分，详细介绍了预训练模型的选择、微调策略的设计以及参数优化方案。本文先基于 Facebook 提出的 mBART-50 多语言序列到序列预训练模型，构建了适用于古今汉语翻译任务的基础架构。随后，通过对神经机器翻译中典型的 Seq2Seq 结构与注意力机制进行分析，明确了模型在编码、解码过程中的语义建模路径。

模型训练策略上分别对全参数微调与 LoRA 参数高效微调方法进行了实验设计与原理说明，指出了二者在性能与资源开销方面的差异，结合实际计算条件选择了更加灵活可控的微调路径。在参数配置层面，本文引入了余弦退火学习率、混合精度训练、梯度裁剪等技术手段以提升训练稳定性与效率。并且设置了对照实验探究更适合文言文训练的参数。

本章的建模与训练准备为后续的实验评估与系统实现奠定了坚实的基础。

第五章　实验与结果分析

5.1　文言文翻译样例展示

下表给出若干文言文原句以及 mBART-50 模型生成的白话文译文示例，以直观展示模型的翻译质量：

表 5-1　翻译示例

文言原文	mBART-50 生成译文
神武曰：闻公有马十二谷，色别为群，将此竟何用也？	神武帝说：听说你有十二匹马，颜色分别为群，这有什么用吗？
荣曰：但言尔意。	尔朱荣说：只管说出你的意思
神武曰：方今天子愚弱，太后淫乱，孽宠擅命，朝政不行	神武说：当今皇上愚昧软弱，太后淫乱，宠妃专权，朝政不能进行
玄挺徐至，坐息良久，忽起击之	杨玄挺率军慢慢来到, 士兵们坐下来休息了很久, 突然起来进攻隋军
春，正月，辛丑朔，荧惑犯紫微	春季, 正月, 辛丑朔, 火星犯紫微

从以上示例可见，模型能够正确理解多数句子含义并输出通顺的现代汉语，它的翻译质量较高。但在实际应用中仍可能存在个别的错误，比如对个别成语的处理不够精确。

5.2　评估指标选择

本文章中主要采用 BLEU、BERTScore 两种自动评价方法，以衡量翻译模型的质量。BLEU 指标通过比较候选译文与参考译文中各阶 n-gram 的重合程度进行评价，分数越高表示有译文质量越好。BLEU 主要关注翻译的精确度，即候选

译文中有多少词序与参考译文匹配，通常计算 1-gram 至 4-gram 的加权平均，加入长度惩罚以避免译文过短。

BERTScore 是一种基于预训练语言模型的语义相似性评价方法。它利用 BERT 等预训练模型提取生成句子和参考句子的词向量，然后计算每个词之间的余弦相似度来得到一个相似性矩阵，最后分别累加并归一化得到精确度、召回率和 F1 分数。BERTScore 的值范围在 0 到 1 之间，可以更细致地反映句子级别的语义匹配程度（1 表示完全匹配）。

BLEU 和 BERTScore 各有侧重：BLEU 评估翻译精确度与流畅度，BERTScore 度量语义一致性。由于单一指标可能无法全面反映译文质量（BLEU 对语义准确度的反馈有限），我们认为结合多种指标进行评估更为合理。

5.3 自动评估结果分析

表 5-2 展示了超参数对模型性能的影响

表 5-2 超参数微调评估指标

组别	BLEU	BERTScore
初始组	23.97	0.82
第一组	20.03	0.78
第二组	22.33	0.82
第三组	19.48	0.80

根据实验结果可知，在学习率偏小的情况下，模型性能下降较为明显，原因可能是学习率过小导致训练过程收敛缓慢，模型未充分学习到语料特征。而在学习率较大的情况下，模型性能略微下降，可能是训练不稳定导致性能下降。最后调小批次大小使得模型的性能下降明显，原因可能是小批量更新带来的梯度估计噪声增大，使得模型难以稳定收敛。

表 5-3 展示了本研究所提模型（mBART-50 全参数微调）与 LoRA 在测试集上的自动评测结果（BLEU、BERTScore）

表 5-3　LoRA 微调评估指标

模型名称	BLEU	BERTScore
mBART-50 (全参数微调，本研究)	23.97	0.82
mBART-50 (LoRA)	21.41	0.78

使用 LoRA 技术的 mBART-50 在保留模型大部分参数不变的情况下，只对少量参数进行低秩适配（约 0.19%），如图（5-1）。其结果略低于全参数微调模型（BLEU 低约 2 点，BERTScore 低约 0.04 点）。这表明 LoRA 方法在减少训练开销的同时，依然能够利用预训练模型的迁移学习能力。

同时我们可以看到全参数微调后，mBART-50 的各项指标表现都很优异，说明其文言文翻译能力优秀。

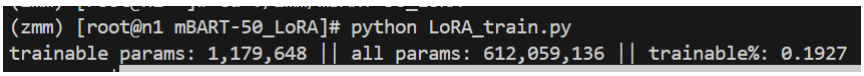

图 5-1　训练参数占比

5.4　人工评估与错误案例分析

以“神武曰：闻公有马十二谷，色别为群，将此竟何用也?”这句为例，模型较好地处理了宾语前置结构：“有马十二谷”被译为“有十二匹马”，保持了主谓宾顺序，译文逻辑清晰。

又比如说“玄挺徐至，坐息良久，忽起击之”这句，模型将其译为“率军慢慢来到，士兵们坐下来休息了很久，突然起来进攻”，正确还原了多个省略主语的并列句，体现出模型一定程度上的篇章信息整合能力。尽管“之”未被明确译出，但从整体语义连贯性看，译文的含义仍然在可接受的范围内。

模型在一些固定搭配与语义隐喻的处理上表现也比较好。如“神武”未被误译为“神奇武器”，而识别为人名，译为“神武帝”，这里模型成功地利用上下文对专有名词进行了消歧。“孽宠擅命，朝政不行”译为“宠妃专权，朝政不能进行”，抓住了“孽宠”与“擅命”的语义核心，显示出词义迁移能力。

不过在“春，正月，辛丑朔，荧惑犯紫微”中，模型译为“火星犯紫微”，虽

词义直译无误，但未解释“荧惑”即“火星”的古今异名，对非专业读者仍显晦涩，说明天文术语缺乏泛化解释机制，是未来可以加强的方向。

“但言尔意”被翻译为“只管说出你的意思”，较好体现了文言简练转为现代口语化表达的能力。模型成功从语气词“但”中提取出宽容许可语气，表明具备一定的风格迁移能力。

但是由于时间和人力有限，本研究未进行更加充分系统的人工评分评测，而是从模型生成结果中提取典型案例进行分析，总结主要错误类型。包括如译文可能句子不顺、语序紊乱。例如原文的主谓宾关系被打乱，导致译文逻辑不通顺；或原文倒装句未转换成符合现代语序的表达，使读者难以理解。文言句式自由度高，模型有时未能正确重组句子结构。同时，会出现对古汉语特有词汇或成语解释错误。例如将“苟且偷安”翻译为字面意思，而非表达“得过且过、不求上进”的含义；或忽略了通假字和典故的隐含意。其他错误诸如漏译（将原文成分遗漏）、增译（加入无关内容）、标点和格式等问题。

5.5 本章小结

本章展示了本文的核心实验结果，给出了模型翻译的实际样例。同时选用BLEU 和 BertScore 两个指标对模型进行了评估并进行了一定的人工评估，验证了模型训练的有效性和准确性。

第六章　文言文翻译系统设计与实现

6.1　系统分析

本章将对文言文翻译系统的整体架构和功能进行系统性分析。通过明确需求，功能模块，运行环境等等内容，为后续系统功能开发奠定基础。

6.2　可行性分析

数据收集与预处理，前文根据开源数据库以及自主开发的爬虫系统收集出了大量的训练数据，构建了完善的文言文平行语料库。同时对数据进行了清洗，对格式进行了标准化。翻译模型上，已经基于 mBart-50 结合全参数微调以及 LoRA 微调训练出了一个可用的大预言模型，并对其进行了评估和优化。最后应该开发一个用户友好的界面，方便用户输入文言文文本并获得相应的现代文翻译结果。

6.3　需求分析

随着“国学热”“古文热”等文化趋势的持续升温，传统文化在大众视野中的影响力不断增强，尤其在基础教育、文化传播以及自媒体内容创作等场景中，古典文献的接触频率明显上升。但是言文作为一种典范的古代书面语，其高度凝练、省略丰富、句式灵活、词义演化明显等语言特征，给普通用户特别是青少年学生在理解和学习过程中带来了比较大的障碍。

目前市面上主要依赖传统人工注释或纸质对照本等方式来辅助理解，这种方式不仅效率低下，且依赖注释者的语言水平和理解能力，存在一定的主观性和不可控性。人工翻译难以适应大规模查询与即时反馈的场景需求，已逐渐无法满足现代数字阅读环境下用户对交互性、智能化与实时性的翻译体验期待。

而本文所提出的系统就可以构建一个基于神经机器翻译（NMT）技术的智能文言文翻译工具，利用预训练大模型（如 mBART-50）对输入的文言文句子进

行深度语义解析与语言风格迁移，实现自动、高效、通顺的现代白话文生成。该系统不仅可为中学语文学习者提供辅助理解工具，也可为古籍数字化平台、在线教育产品、AI 写作插件等提供基础服务接口，具有广泛的实际应用前景与社会价值。

6.3.1 目标用户

该系统的目标用户包括但不限于：中学生、高校学生在文言文学习场景下的辅助翻译；文史类研究人员对古文材料的初步处理；一般读者的日常文言阅读辅助工具。

6.3.2 系统功能性需求分析

首先用户登录和个人用户界面，用户可以创建账户并登录系统，进入翻译界面，个人账号界面可以保存翻译记录和个人设置。系统应该可以输入文言文句子和段落，系统对这些文字进行翻译，转化为现代文并呈现在界面上供用户查看。

本系统具有古文字的自动识别与注解功能。当用户输入现代汉语或文言文文本并请求翻译时，系统不仅能返回对应的翻译结果，还将对翻译文本中的关键古文字进行提取和解释。通过集成古文字词典资源与语义分析模块，系统能够为用户提供每个生僻字或典故的详细释义、词性及用法背景，辅助用户更深入地理解古文内涵。用户可通过参数设置或界面选项，自主选择是否启用该功能。系统将根据用户的配置，灵活启用或关闭古文字注解模块。

系统还提供基于主题或关键词的古诗文自动生成能力。用户输入所需的诗词主题、意象或关键字，系统将基于语言模型生成符合古典诗词格律的内容。同时，为提升生成质量与个性化程度，系统支持用户选择指定的诗词风格（如唐诗、宋词、楚辞等）。在生成过程中，系统将依据所选风格自动调整用词风格、句式结构与节奏韵律，以生成更贴合用户期望的诗词作品。

6.3.3 系统非功能性需求分析

易用性：提供清晰的命令行接口，用户不需要花费额外的时间学习使用系统，能快速使用目标模块。系统还需要提供必要的帮助，以便用户遇到问题能够

得到及时的解决。

响应速度：系统需要具备快速的响应能力，方便用户在使用翻译功能时可以迅速的给出翻译结果。翻译过程中应合理调度 GPU/CPU 资源，避免显存/内存泄露，确保长期服务稳定运行。

用户友好性：系统的操作界面应该简洁易懂，提供操作指引，让用户可以轻松明了的操作和翻译。

可扩展性：支持后续迁移至其他古文风格，如骈文、古籍语料等；也可用于现代中文其他翻译任务。系统各部分（数据预处理、模型推理、后处理）应模块化，便于后续替换模型或添加新特性。

可靠性：系统能够维持正常功能，在发生故障时能够快速恢复。系统需要具备一定的容错能力，能够在发生错误或异常时及时处理，并尽量减少对用户的影响。在使用用户较多的情况下，系统可以保持工作不崩溃。

6.4 系统架构设计

本系统采用多层架构设计模式，如图（6-1），整体结构划分为三大层次，即表示层、业务逻辑层和数据访问层。各层职责清晰、功能独立，既便于系统功能的拓展与维护，又有助于提高模块间的解耦性与系统整体的稳定性。

表示层主要负责系统与用户之间的交互，是用户操作的直接入口。该层通过图形化界面或 Web 页面接收用户输入，并将处理结果可视化地反馈给用户。其主要包括三个功能页面，用户翻译页面用于接收用户输入的文言文或现代文文本，展示翻译结果以及相应的注解信息；诗词生成页面用于接收用户输入的主题或关键词，展示系统自动生成的古典诗词内容；帮助页面用于提供系统使用说明、功能引导与常见问题答疑，提升用户体验与系统易用性。

业务逻辑层是系统的核心功能处理单元，主要负责调度和组织各项业务逻辑，实现翻译、注解、诗词生成等主要功能。该层包括模型转发模块、用户翻译处理模块、注解处理模块、诗词生成模块。其中模型转发模块会根据用户请求的类型（翻译、注解或诗歌生成），将请求转发至对应的模型服务，支持多模型并发调用和负载均衡；用户翻译处理模块将接收来自表示层的翻译请求，调用相应的翻译模型进行文言文与现代文之间的互译，并将结果返回前端；注解处理模块

则会对翻译结果或原文进行古文字识别与释义，提供对应注解内容；而诗词生成模块则是根据用户输入的主题或关键词调用古诗词生成模型，并生成符合指定风格的诗词文本。

数据访问层负责系统数据的存储、管理与调用支持。各类预训练或微调后的模型文件直接保存在指定路径下，由业务逻辑层调用。由于模型不涉及动态数据操作，未纳入数据库管理；系统数据库用于持久化存储用户历史翻译记录、注解结果与生成的诗词文本等信息，支持后续查询、统计分析与功能扩展。

该三层架构体系实现了系统各部分的逻辑隔离和功能独立，不仅提升了整体架构的清晰度与可维护性，也为后期系统的扩展与优化提供了良好的基础。

图 6-1　系统架构图

6.5　系统前端设计

我们使用 Vue3 +Element Plus 来构建前端静态界面。

我们实现了一个左右分栏的 web 页面，左侧是输入框，用户可以在左侧输入框输入想要翻译的文言文，点击右侧的翻译，系统后端就会收到请求并调用大模型进行翻译，并将翻译后的结果返回前端页面。如图（6-1）。

图 6-2　前端页面

6.6　系统后端设计

我们使用 Python 的 Flask 框架实现后端，以处理翻译请求。服务端主要包括模型转发业务、用户翻译处理业务、诗歌生成处理业务、注解生成处理业务和系统数据库等。服务端主要接口如表 6-1。

表 6-1　服务端接口

类	接口名称	具体描述
ModelProcessor	forward	作为模型转发业务的接口，负责转发请求，起到路由的作用
TranslateProcessor	translate	转发用户翻译文本
AnnotationProcessor	annotate	转发注解文本
PoetryGenerator	generate	转发诗歌生成文本
ModelProcessor	cls2mod	与数据层的文言文-现代文翻译模型交流接口
ModelProcessor	mod2cls	与数据层的现代文-文言文翻译模型交流接口
ModelProcessor	annotate	注解接口
ModelProcessor	poegen	与诗歌生成模型交流的接口

6.7 本章小结

本章主要论述了系统的实现层面，介绍了文言文翻译系统在可行性、需求以及系统设计方面的细节，并简要阐述了前后端的实现原理和功能，为文言文翻译提供了有效的工程支撑。

第七章　总结与展望

7.1　本文工作总结

本文围绕文言文到现代汉语的自动翻译任务，基于 mBART-50 多语言预训练模型展开研究与系统实现。在文章的开头，本文介绍了神经机器翻译（NMT）的基础理论和 Seq2Seq 架构与多语言预训练模型的发展脉络，分析了 mBART-50 模型的编码器-解码器结构、语言标签控制机制以及跨语言迁移能力。在数据方面，本文融合通古、尔雅数据集与从古诗文网爬取的自建语料，构建了包含上百万句对的文言—白话平行数据集，覆盖多种古文风格，满足文言文翻译训练的基本需求。

模型训练时，本文采用了 Hugging Face Transformers 框架，完成了 mBART-50 模型在文言文翻译任务上的全参数微调，并设置了科学合理的训练策略，如余弦退火学习率、混合精度训练、梯度累计等，以提高训练效率与模型收敛性能。为降低资源开销，本文还尝试引入了参数高效微调方法 LoRA，实现了训练开销与性能的良好平衡。系统训练过程中通过 BLEU 和 BERTScore 等指标实现自动评估，以对比实验选出最适合的微调策略。

总体而言，本文初步训练出了基于 mBART-50 的文言文翻译模型，并具有较为优良的翻译效果，同时也构建了前后端功能模块。

7.2　不足

尽管本文取得了一定进展，但仍然存在不足。本文主要依赖自动评估指标如 BLEU、BERTScore，未能结合人类的专业评审来全面衡量翻译的语义质量，可能遗漏语义不通或风格不符等主观问题。微调优化上由于资源和时间有限，还未探索完更加适合的参数。

在微调策略方面，尽管引入了 LoRA 等参数高效微调机制，初步验证了其可行性，但相关实验尚处于浅层尝试阶段，未对不同 r 值、注入模块位置、层选

择等变量展开深入对比，也未结合其他高效微调方法（如 Adapter、Prefix Tuning 等）进行综合性对照评估，导致对 LoRA 机制的理解与应用深度仍显不足。

同时系统工程方面，也未完成系统的真实环境部署，系统仍处于实验室阶段，尚未提供在线服务，没有在真实环境下验证系统的稳定性、可靠性，实际应用价值仍有待提升。

7.3 未来工作展望

尽管本文构建的文言文翻译系统在多语言预训练模型的微调应用方面取得了初步成果，但在研究深度、系统完备性以及实际应用方面仍存在较大提升空间。未来的研究可以围绕以下几个方面展开：

当前训练语料主要来自古诗文网、尔雅与通古数据集，虽然数量已达上百万条，但在风格、体裁和语域上仍相对集中。后续应进一步拓展语料来源，并适当引入《左传》《战国策》《史记》《文心雕龙》等典籍中的长句、散文、史传体样本，以提升模型对复杂语言结构的适应能力。此外，可以引入更多跨时期的文言语料，如六朝志怪小说或清代笔记，使模型对文言文发展的时序特征具备更强的语言感知能力。

对比更多的微调策略。现阶段主要采用 BLEU 与 BERTScore 等自动评估指标，虽然能有效度量模型输出的文本相似度，但难以准确反映语义理解、文化内涵、语言风格等主观维度。未来可结合语言专家评估、语言学评分标准或引入众包平台进行人工评估，构建覆盖“流畅性、忠实度、优雅度、对等性”等指标的主观评测体系，使得模型优化方向更具人文指向性。

文言文翻译作为一种高度依赖语法理解与词汇分析的任务，天然适合与相关辅助任务进行联合建模。未来可将文言文断句、词性标注、句法分析、实体识别等任务整合到统一的多任务学习框架中，充分挖掘文言语言结构之间的关联性，从而提升翻译系统的准确性与鲁棒性。特别是在处理无标点文本、长难句、文白异序现象时，多任务模型可能具备更优解码路径。

跨模态、跨语言扩展研究。随着多模态大模型的发展，未来还可以探索图文结合的文言文解读系统，例如将古书影印图像结合 OCR 与翻译模型，自动完成文献数字化与语义转写。此外，可进一步探索中文与其他语言（如英语、日语）

之间的古今文翻译任务，构建多语言对照版本的古文翻译系统，为跨文化研究提供辅助。

参考文献

[1] YANG Z, CHEN K, CHEN J. Guwen-UNILM: Machine Translation Between Ancient and Modern Chinese Based on Pre-Trained Models[C]//CCF International Conference on Natural Language Processing and Chinese Computing. 2021: 116.

[2] VOITA E, SENNRICH R, TITOV I. When a Good Translation is Wrong in Context: Context-Aware Machine Translation Improves on Deixis, Ellipsis, and Lexical Cohesion[J]., 2019. DOI: 10.18653/v1/P19-1116.

[3] GU J, WANG Y, CHEN Y, et al. Meta-Learning for Low-Resource Neural Machine Translation[EB/OL]. 2018. https://arxiv.org/abs/1808.08437. arXiv: 1808.08437 [cs.CL].

[4] SENNRICH R, ZHANG B. Revisiting Low-Resource Neural Machine Translation: A Case Study[C]//KORHONEN A, TRAUM D, MÀRQUEZ L. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 211-221. DOI: 10.18653/v1/P19-1021.

[5] ZHANG Z, LI W, SU Q. Automatic Translating Between Ancient Chinese and Contemporary Chinese with Limited Aligned Corpora[M/OL]//Natural Language Processing and Chinese Computing. Springer International Publishing, 2019: 157-167. http://dx.doi.org/10.1007/978-3-030-32236-6_13. DOI: 10.1007/978-3-030-32236-6_13.

[6] LIU D, YANG K, QU Q, et al. Ancient–modern Chinese translation with a new large training dataset[J]. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 2019, 19(1): 1-13.

[7] YANG Z, CAI P, FENG Y, et al. Generating Classical Chinese Poems from Vernacular Chinese[C/OL]//INUI K, JIANG J, NG V, et al. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019: 6155-6164. https://aclanthology.org/D19-1637/. DOI: 10.18653/v1/D19-1637.

[8] CHANG E, SHIUE Y T, YEH H S, et al. Time-Aware Ancient Chinese Text Translation and Inference[EB/OL]. 2021. https://arxiv.org/abs/2107.03179. arXiv: 2107.03179 [cs.CL].

[9] DANKERS V, LUCAS C G, TITOV I. Can Transformer be Too Compositional? Analysing Idiom Processing in Neural Machine Translation[EB/OL]. 2022. https://arxiv.org/abs/2205.15301. arXiv: 2205.15301 [cs.CL].

[10] ZHENG Z, YUE X, HUANG S, et al. Towards Making the Most of Context in Neural Machine Translation[EB/OL]. 2020. https://arxiv.org/abs/2002.07982. arXiv: 2002.07982 [cs.CL].

[11] CAO J, PENG D, ZHANG P, et al. TongGu: Mastering Classical Chinese Understanding with Knowledge-Grounded Large Language Models[EB/OL]. 2024. https://arxiv.org/abs/2407.03937. arXiv: 2407.03937 [cs.CL].

[12] BAHDANAU D, CHO K, BENGIO Y. Neural Machine Translation by Jointly Learning to Align and Translate[EB/OL]. 2016. https://arxiv.org/abs/1409.0473. arXiv: 1409.0473 [cs.CL].

[13] PAPINENI K, ROUKOS S, WARD T, et al. Bleu: a Method for Automatic Evaluation of Machine Translation[C/OL]//ISABELLE P, CHARNIAK E, LIN D. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, 2002: 311-318. https://aclanthology.org/P02-1040/. DOI: 10.3115/1073083.1073135.

[14] ZHANG T, KISHORE V, WU F, et al. BERTScore: Evaluating Text Generation with BERT[J/OL]. CoRR, 2019, abs/1904.09675. arXiv: 1904.09675. http://arxiv.org/abs/1904.09675.

[15] HU E J, SHEN Y, WALLIS P, et al. LoRA: Low-Rank Adaptation of Large Language Models[EB/OL]. 2021. https://arxiv.org/abs/2106.09685. arXiv: 2106.09685 [cs.CL].

[16] AGHAJANYAN A, ZETTLEMOYER L, GUPTA S. Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning[EB/OL]. 2020. https://arxiv.org/abs/2012.13255. arXiv: 2012.13255 [cs.LG].

[17] GUO G, YANG J, LU F, et al. Towards Effective Ancient Chinese Translation: Dataset, Model, and Evaluation[EB/OL]. 2023. https://arxiv.org/abs/2308.00240. arXiv: 2308.00240 [cs.CL].

致　　谢

大学四年，行文至此，画上了一个圆满的句号。这四年对我来说无疑是影响很大的四年，是成长的四年。学业上，从零开始接触到软件工程这个专业，窥探到软件领域的一丝丝奥妙，四年的求学历程，在反复的试错与重构中逐渐显现出清晰的逻辑脉络。也结交到很多朋友，来自天健的大家，是他们让我的大学四年从不孤独、充满快乐。

至此，感谢葛老师的指导，总能提出有用的建议，每周的会议可监督我的实验和论文进度。感谢南京大学，希望母校越办越好。感谢父母的养育和支持。感谢一路走过来的朋友，有你们相伴。感谢自己，一步步成长到今天。