中南大學

CENTRAL SOUTH UNIVERSITY

本科毕业设计(论文)

GRADUATION DESIGN(THESIS)

题　　目：	互联网金融纠纷调解机器人-多轮对话下的生成式任务研究
学生姓名：	张是超
指导教师：	廖志芳
学　　院：	计算机学院
专业班级：	软件工程 2004 班

本科生院制

2024 年 6 月

互联网金融纠纷调解机器人-多轮对话下的生成式任务研究

摘要

近年来，网络信贷平台快速发展，诸如花呗，借呗等网上金融贷款产品层出不穷。信贷平台可以帮助用户解决当下的资金问题，但是部分用户无法及时按照借贷合同及时还款，从而引起了许多金融纠纷案件。目前针对这类金融纠纷案件大多是由平台起诉至法院，法院首先将案件委托给第三方调解机构，调解机构安排调解员和当事人进行沟通催促其尽快补齐欠款，但是目前这种人工调解方案会耗费巨大人力沟通成本，导致效率低下。

为了解决以上问题，本研究采用 SFT 技术微调对比了现有主流开源大型模型，如 ChatGLM、LLaMA、Qwen 等，训练出专属的 AI 模型 Mediator_Chat_Bot，用于担任纠纷调解员角色与当事人进行第一次对话沟通，催促其尽快还款，经过全参数微调和 Lora 微调，该模型的最优 BLEU-4 值可以达到 82.4%，ROUGE-L 指标可以达到 85.28%，这被认为达到了优秀的文本生成效果。

同时，我们还提出了一种基于 BERT 与 LSTM 上下采样时间步融合的模型 Mean_BERT_LSTM_MLP，并依此训练了一个当事人还款意愿预测模型，Mean_BERT_LSTM_MLP 模型优于传统的 BERT+Linear、BERT+RNN 与 BERT+LSTM 模型，可根据第一次通话记录预测当事人是否有潜力被成功调解。在测试集上，该模型取得了 92.61%的 F1 值，证明了其准确性与有效性。最后，我们使用对话界面来展示对话系统，用来模拟当事人与调解员电话沟通内容。

关键词：金融纠纷　大语言模型　微调　多轮对话系统　AI 调解员

Research on Generative Tasks in Multi-turn Dialogue for Internet Finance Dispute Resolution Chatbots.

ABSTRACT

In recent years, online credit platforms like Chanting have flourished, offering financial loan products to users in need. However, some users struggle to repay loans on time, leading to numerous financial disputes. These disputes often end up in court, requiring manual mediation by third-party organizations. This process incurs significant manpower and communication costs, resulting in inefficiencies.

To address these challenges, this study utilizes SFT technique to fine-tune mainstream open-source large-scale models such as ChatGLM, LLaMA, and Qwen. The goal is to train an AI model called Mediator_Chat_Bot, which acts as a dispute mediator. The model engages in initial dialogues with parties to encourage timely repayment. Through full-parameter fine-tuning and Lora fine-tuning, the model achieves an optimal BLEU-4 value of 82.4% and a ROUGE-L metric of 85.28%, indicating excellent text generation capabilities.

Additionally, we propose a model called Mean_BERT_LSTM_MLP, which integrates BERT with LSTM-based downsampling of time steps, and used it to train a model for predicting participants' repayment intentions. The Mean_BERT_LSTM_MLP model outperforms traditional BERT+Linear, BERT+RNN, and BERT+LSTM models, demonstrating its effectiveness in predicting whether a participant has the potential to successfully resolve the dispute based on the first conversation record. The model achieved an F1 score of 91.74% on the test set, demonstrating its accuracy and effectiveness. Lastly, we utilized a conversational interface to showcase a dialogue system that simulates the phone conversations between participants and mediators.

Key words：Financial disputes　Large language model　Fine-tuning　Multi-turn dialogue system　AI mediator

目录

第 1 章　绪论

1.1　课题背景及意义

随着互联网金融行业的快速发展，其在金融服务中所占比重不断增加。然而，互联网金融纠纷问题也随之增多，涉及用户与平台之间的复杂矛盾。金融纠纷案件由于其涉及面广、专业性强、权力责任关系复杂，再加上金融市场创新活跃，新的情况与新的问题层出不穷。在过去的很长一段时间内，金融机构由于其权威性不够、专业性不足等问题，无法及时地解决金融纠纷、维护自身权益，这就导致大量的金融纠纷案件只能依靠法院诉讼解决，致使法院同类案件数量居高不下，不堪重负。

调解，是指发生纠纷的双方当事人，在第三方的主持下，通过第三方依照法律和政策的规定，对双方当事人的思想进行排解疏导，说服教育，促使发生纠纷的双方当事人，互相协商，互谅互让，依法自愿达成协议，由此而解决纠纷的一种活动。当前调解的分类主要有三种：法院调解、行政调解、人民调解三种。调解有诸多好处，对于当事人而言：节约当事人的时间、不会出现法院审判后面临的无法承担的后果、可以节约当事人的罚款金额和律师费用等、尽量降低对当事人的影响。对原告而言：节约了时间成本、更早的拿到欠款、减少了诉讼成本。对于法院而言：可以节约司法资源、可以推进溯源治理。

目前互联网金融的调解主要是由调解员和当事人电话沟通，并了解当事人逾期还款的原因，然后再根据实际情况对其提出一些减免或者分期付款的方案。当事人不愿意还款主要集中在以下两个方面：(1) 无还款能力，例如收入减少/中断、重大变故，多头重债等因素。(2)无还款意愿，例如收到恶意欺诈、产品原因（如利息高、违约金高等对产品不满）、服务原因（如：服务态度差、合理要求被拒、暴力催收等）、显失公平（即当事人觉得不公平）、以及受到反催收等第三方影响等。调解员主要是为了捕获当事人逾期还款原因，然后给出相应的解决建议。调解员调解逾期还款的流程如图 1-1：

图 1-1：逾期还款流程图

需要注意的是，调解组织只接受法院委托，并在法院的指导进行调解，调解组织与调节双方均无利益关系。因此第三方调解组织可以保证从第三方立场上保持客观公正的态度进行调解。调解组织与当事人、法院的关系如图 1-2：

目前调解组织都是依靠真人调解员去和当事人进行对接，但是这种效率低下，不仅需要耗费大量的人力物力财力，还无法精准识别出高意愿还款用户。这时候就需要一种可以和当事人进行沟通的 AI 机器人，用来充当第一次和当事人进行对话的调解员，然后获取第一次通话当事人所说话语。之后我们可以根据当事人的说话列表训练模型，来预测当事人有意愿还款的概率。这样我们真人调解员就可以优先介入这些有较高还款意愿的当事人，并引导当事人进行还款。这样可以极大解放调解员的压力，对于案件的调解成功也起着极其重要的作用。

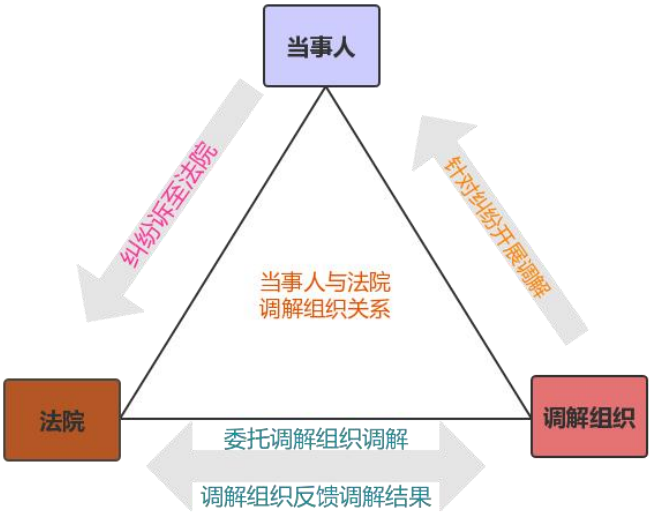

图 1-2：调解组织、当事人、法院三者之间的关系

以庭前调解的方式解决纠纷，除了能为当事人双方节省诉讼时间和金钱消耗外，还能让第三方充分与企业、银行等金融机构进行沟通，以更加人性化的方式，在各方之间达到了真正的“共赢”，达到“案结、事了、人和”的良好效果，营造了和谐的金融环境。除此之外，使用 AI 调解员来和当事人进行沟通，也减轻了第三方机构调解员的调解压力。

1.2 国内外研究现状

1.2.1 基于规则的对话机器人

基于规则的对话机器人于 20 世纪 90 年代开始正式兴起，一般这类对话机器人会被用来充当自动助理的角色，他们根据预定义的规则和反应进行操作，使得其可以帮助处

理特定用户的查询以及回答常见问题。基于规则的对话机器人设计类似于流程图，需要设计者对其设置一系列的对话框架，以期可以让其预测用户可能的提问，以及对话机器人应当如何回应。基于规则的对话机器人可以使用简单和复杂的规则来限制他们回复的框架，他们无法回答用户的除定义的规则以外的任何问题。这种类型的机器人不会通过交互来进行学习，也不会学习到除被训练场景以外的问题回复技巧。

1966 年，MIT 人工智能实验室开发了一款完全基于规则的对话机器人-ELIZA[1]，该机器人历时三年(1964-1966)开发完成，其模拟了一个心理医生的角色。ELIZA 被预定义了很多模式(Pattern)，每个模式都有与其对应的转换机制(Transform)来生成回答，这里的 Pattern＋Transform 就可以理解为一种规则。当在对话过程中检测到预设好的相关模式时，ELIZA 会根据 Transform 来生成回复。ELIZA 被开发出来之后，在 1972 年斯坦福大学推出了一个新的基于规则的对话机器人-PARRY[2]。与 ELIZA 不同的是，其模拟的是一个患有精神分裂症的患者。PARRY 除了使用简单的规则之外，还加入了自己的情感状态，例如害怕和生气等。在此后的十几年内也出现了许多基于规则的对话机器人，例如专家系统(Expert System)、尤内克斯顾问(UNIX Consultant)[3]等。但是这些对话机器人只能对人类提问中的部分关键词做出特定的答复，在于人类交互过程中，机械化，规则化的特征非常显著。

1.2.2 基于检索的对话机器人

基于检索的对话机器人[4]通过预先构建的相关知识库或者语料库来响应用户的问题。其基本工作原理类似于浏览器问答系统，当用户提出相应问题时，系统会从预先存储的数据中检索相关信息，检索过程中常用的算法包括基于词频的简单检索、TF-IDF[5]、BM25 算法[6]等。检索出相关候选回复后返回最为匹配的内容进行响应。基于检索的对话机器人不需要生成新的文本，仅仅从语料库或者对话数据库中进行索引，选择最为匹配的信息。

开发一款基于检索的聊天机器人一般分为以下几步：首先需要构建一个“查询-回复”对话数据库，然后根据当前用户输入利用相关候选索引检索算法粗粒度召回多个可能的回复，之后利用相似度特征计算算法计算每个候选回复与对话的相似度，最终利用排序算法(例如重排序[7])对候选回复进行排序并返回最佳的回复信息。图 1-4 展示了基于检索的对话机器人的运行流程：

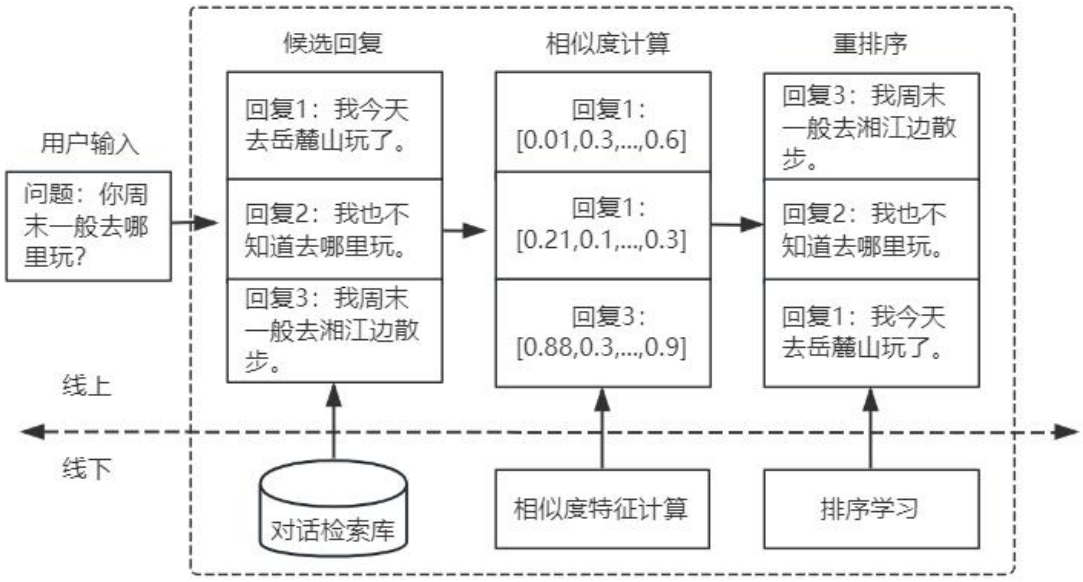

图 1-3：基于检索的对话机器人整体设计流程

在上面例子中，首先根据用户输入从大规模对话检索库中快速查找多个候选回复集，这是基于检索的对话机器人的粗粒度召回阶段，在实际应用中，检索库一般会有成千上万条数据，因此这一流程可以使用倒排表或者向量检索的方式进行快速召回。然后将这些候选回复输入进训练好的多个回复选择模型进而为每个候选回复生成多个得分，之后将得分拼接成向量输入进排序学习模型确定最终的回复排序集合。

目前针对基于检索的对话机器人一般主要集中于相似度计算模块，即回复选择模块。回复选择模块一般分为单轮回复选择任务和多轮回复选择任务。

在早期研究中，研究者一般将回复选择任务简化为单轮回复选择任务，即忽略掉前面几轮的聊天信息，仅仅根据用户当前输入进行索引，但是单轮回复选择任务忽略了丰富的上下文语境，导致效果不是特别理想，在这个过程中出现了许多单轮回复选择模型，例如 Lu[8]等人的局部特征选择模型、Wang[9]等人的深度匹配树模型、Wu[10]等人的主题向量计算模型。

近年来随着研究的深入，有许多学者开始研究多轮回复选择任务的算法优化。Zhou[11]等人指出，过去的对话模型在建模对话上下文和候选回复文本时，主要从词汇的角度考虑，而忽略了话语的层面。这导致模型难以准确把握话语的相关信息和依赖关系。为解决这一问题，作者提出了一种多视角的多轮回复选择模型。该模型以对话上下文为输入，并同时从话语和词汇的角度来改进多轮回复选择的性能。Yan 等人[12]提出了一种多轮回复选择的方法，即利用历史对话进行消息重构。该方法的主要思路是，通过重构基于历史对话的消息，能够捕捉到历史对话不同方面的特征信息。在进行回复选择时，

综合考虑原始消息和重构的消息，以此用来增强回复与历史对话语境的相关性。

虽然检索式对话机器人发展迅速，但是由于其严重依赖知识库的数据质量，如果知识库中的信息出现不完整或者不准确的问题，对话机器人的性能会受到很大的影响。除此之外，检索式对话机器人难以处理多义性的问题，对于指代不明确的用户输入，检索式机器人往往不能正确理解用户意图，从而给出低质量的回复响应。

1.2.3 基于 Seq2Seq 的对话机器人

由于深度学习的发展，基于 Seq2Seq 的对话机器人[13]迎来了快速发展。基于 Seq2Seq 的生成模型不止可以应用在对话领域，其在文本摘要，机器翻译等场景中都有广泛使用。在大语言模型兴起之前，采用深度学习搭建的对话机器人基本都是采用 Seq2Seq 的架构进行实现的。Seq2Seq 模型最初由 Sutskever 等人提出，最早是应用在机器翻译任务上[14]。Seq2Seq 模型的核心思想是将对话看成一种序列到序列的生成式问题，即将一个输入序列(例如用户输入的问题)转换为另一个输出序列(例如对话机器人的回答)。

Seq2Seq 式的对话机器人主要由两部分组成，分别为编码器端(Encoder)和解码器端(Decoder)。编码器负责将输入序列转化称为一个固定长度的向量表示，提取用户输入的文本特征，该向量包含了输入序列的语义等信息。解码器用来根据这个向量生成输出序列。详细来说，Encoder 端通过将输入序列中的每个单词逐个输入到循环神经网络（RNN）、LSTM、或者是自注意力机制中，逐渐累积输入序列的语义信息和语法信息。当编码器对输入序列完成编码之后，其最终就会输出一个固定长度的向量，该向量捕捉了输入序列的整体语义信息。然后 Decoder 端使用这个向量初始化自己的状态，并开始逐步生成单词，当前时间步的输入来自于 Encoder 端的词向量以及上几个时间步的输出序列，直至生成完整的输出序列为止。这个过程如图 1-4 所示：

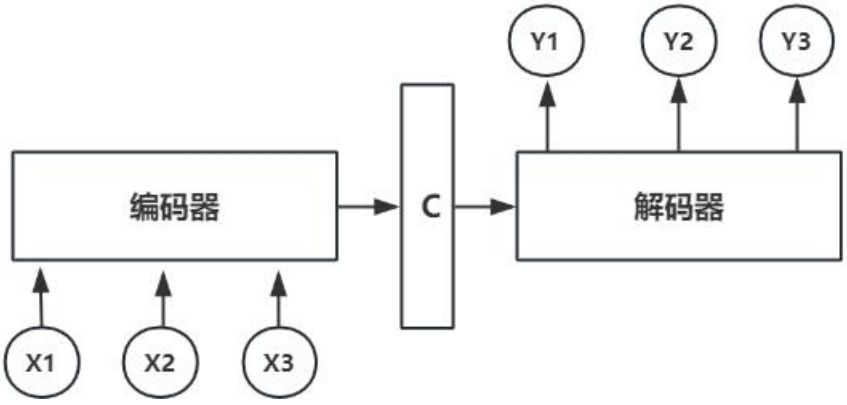

图 1-4：Seq2Seq 模型结构

X 代表的是整个输入序列,其表达式为：

$$X = (x_0, x_1, x_2, x_3, \ldots, x_n) \tag{1-1}$$

之后 X 会输入进编码器进行编码，得到一个固定长度的词向量 C：

$$C = (f(x_0), f(x_1), f(x_2), f(x_3), \ldots, f(x_n)) \quad (1\text{-}2)$$

之后词向量 C 会输入进解码器进行解码，解码器的当前时间步 i 所生成的单词需要根据词向量 C 以及之前时间步的生成单词进行的：

$$Y_i = h(Y_0, Y_1, Y_2, Y_3, \ldots, Y_{i-1}) \quad (1\text{-}3)$$

Seq2Seq 是一个通用的生成式对话机器人框架，编码器和解码器的具体实现一般会使用 RNN 或者 LSTM。

1.2.4 基于大规模语言模型的对话机器人

近年来随着深度学习技术的快速发展，大规模语言模型迎来了新的发展机遇。神经网络为这些模型提供了强大的语言处理和语义理解能力，大语言模型通过海量语料库的学习，积累了丰富的语言知识，从而在语言理解、生成和推理等方面取得了巨大的进步。近年来，算力的提升和大数据的普及更是为大语言模型的发展插上了翅膀。尤其是以 Transformer[15]为代表的新型神经网络架构，让大语言模型在特征提取和上下文建模方面得到了质的提升。同时，随着预训练技术的兴起，使得大语言模型可以在丰富的语料库中进行学习，现有的大语言模型一般采用 Decoder-only 架构，该架构可以很好地执行 Next Token Prediction 任务，进而学习到丰富地更贴近真实世界的语言习惯。如今，大语言模型已经广泛应用于智能客服、机器翻译、文本创作等多个领域。它们不仅能与人类进行流畅的对话，解答各种问题，还能自动翻译多种语言，帮助人们轻松跨越语言障碍。此外，大语言模型还能生成高质量的文本内容，为创作提供源源不断的灵感。

多轮对话的大规模语言模型随着 GPT 系列预训练模型的兴起而兴起，尤其是 GPT3.5 的横空出世，在世界范围内引起了巨大的轰动。Openai 发布的 ChatGPT[16]是一种专注于对话生成的语言模型，其可以很好的支持多轮对话的效果，利用丰富的上下文语义给出当前轮次的回答。GPT 代表着 Generative Pre-trained Transformer（生成型预训练变换模型）。通过学习大量现成文本和对话集合，例如维基百科等，ChatGPT 能够像人类一样即时对话，并流畅地回答各种问题。无论是英文还是其他语言，如中文、韩语等，ChatGPT 都能应对。

自从 chatGPT-3.5 发布之后，国内外也有许多大语言模型如入后春笋般迎来了快速发布、发展阶段。例如 Meta AI 公司发布的 LlaMA 系列模型[17]、清华大学开源的 ChatGLM 系列模型[18]、阿里巴巴公司发布的通义千问系列模型[19]、以及百川智能团队开发的百川系列模型[20]等。这些模型有一个共同的特点，那就是都采用了 Transformer 的

Decoder-only 架构，Decoder-only 架构的结构设计如图 1-5 所示：

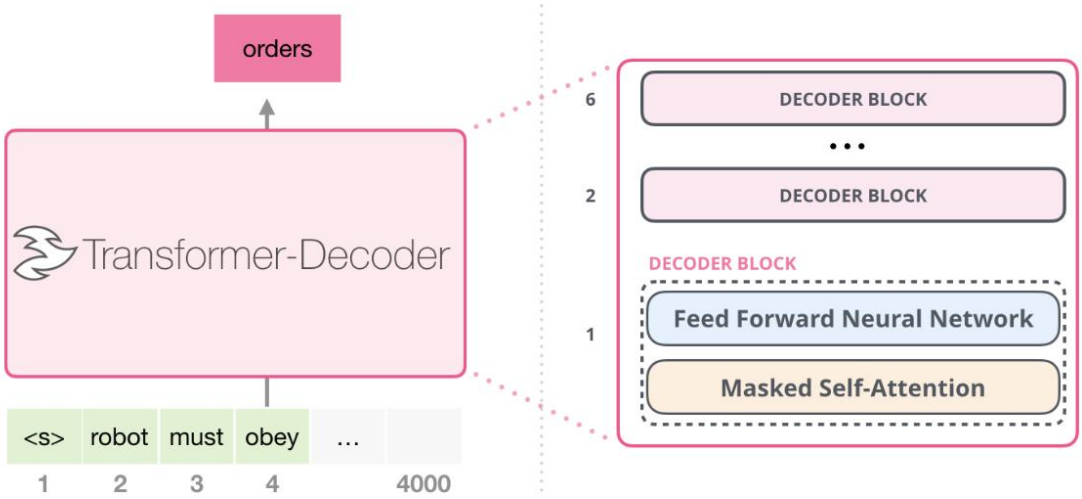

图 1-5：大语言模型 Decoder-only 编码

Decoder-only 架构允许模型根据前面输入内容逐个输出预测的后面单词内容，其舍弃了 Transformer 的第二个自注意力层，即 Encoder-Decoder Self-Attention 层，这是因为其不再需要 Decoder 端提供词向量来进行预测，而仅仅执行 Next Token Prediction 任务。其中的 Decoder Block 是可以堆叠的，现有的大模型一般会将其扩展到几十个，例如 7B 参数的 LLaMA2 预训练模型就堆叠了 32 个 Decoder Block，这也就奠定了其具有强大的泛化能力的基础。利用大规模预训练模型，然后结合自己的垂直领域知识对其进行下游任务微调，正在成为新的对话机器人的发展趋势。

1.3 本文主要研究内容

本文研究内容主要包含以下三个部分：

（一）AI 调解员模型训练及比较

研究多轮对话下的自回归生成任务，从真实第三方调解机构获取第一手调解对话资料，运用 Lora、全参微调等方式训练大规模语言模型，使其可以充当调解员的角色，同时加入 Lang-chain RAG 检索增强技术将相关法律条款以及话术在对话过程中进行展现，从而对欠款当事人进行劝导和建议，为金融贷款平台、法院、当事人之间建立一个良好的沟通桥梁。除此之外，本文对比了三种主流的语言模型(ChatGLM3、Qwen、LlaMA2)的不同参数量(1.3B、1.8B、6B、7B)以及不同微调方式(Lora、Full)的效果差别。并取效果最好的模型充当调解员角色。

（二）基于词向量的当事人还款意愿模型设计

本文利用 BERT 等自编码模型的基于词向量的研究方法获得当事人的第一次对话流程的 mean embedding 信息，并利用 Linear、RNN、LSTM、以及我们设计的

Mean_BERT_LSTM_MLP 等分类器对获取的 mean embedding 信息进行分类，从而及时辨别出有意愿还款的当事人。对当事人进行分层，之后沟通过程中再由真人调解员进行跟进最新进展，可以最大限度地提高调解成功概率。

（三）模型训练、测试以及界面可视化

本论文最终对 AI 调解员和基于词向量的用户分层模型进行训练，并在真实的场景下进行测试，最终使用对话界面来模型展示 AI 调解员和当事人进行沟通的过程并获取当事人说话内容进行预测。

图 1-6 展示了本文 AI 调解员的整体设计流程：

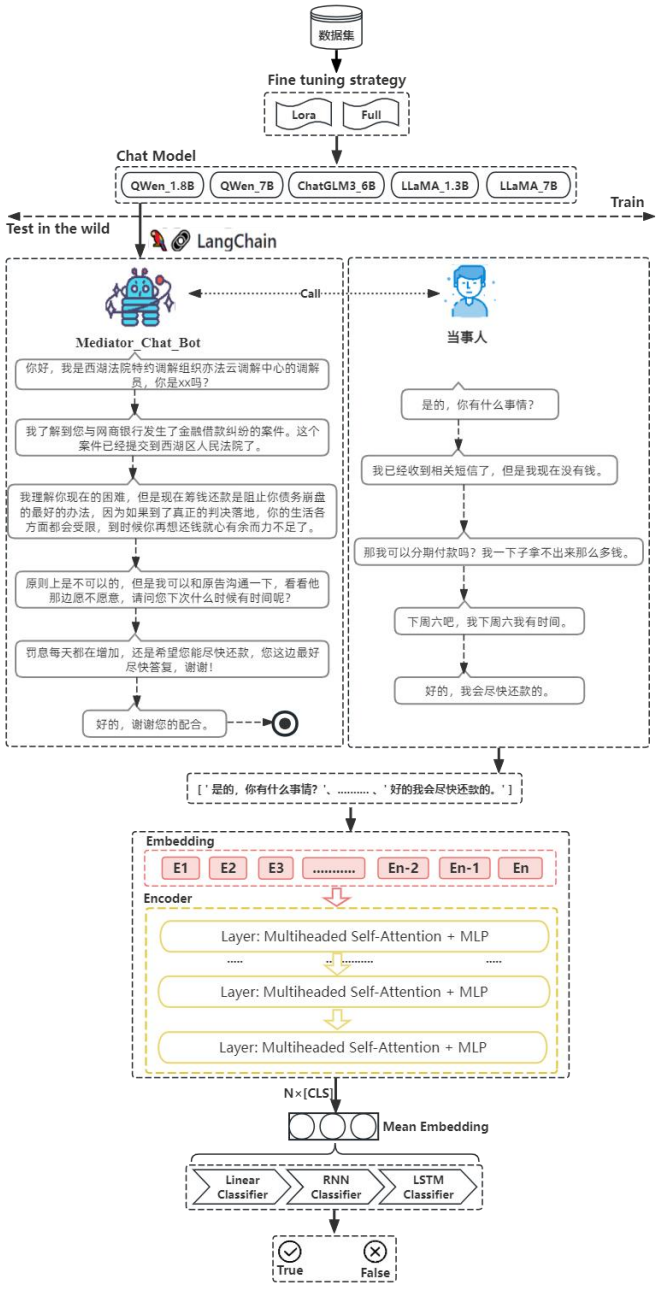

图 1-6：AI 调解整体流程介绍

1.4 本文组织结构

本文一共分为6个章节，第一章为绪论，主要介绍互联网金融纠纷调解机器人课题背景、国内外研究现状。第二章介绍本研究的相关技术方案，包括Attention机制、SFT方法、以及目前主流的大语言预训练模型等。第三章介绍本文的模型设计方案以及实验的细节，包括预训练模型的选择以及当事人还款意愿预测模型的设计。第四章分析了两类模型的实验结果。第五章展示了模型的可视化效果，并通过界面对话的形式模拟AI调解员和当事人之间的沟通。第六章对本文进行了总结，并提出了对未来的展望。

第 2 章　主要研究内容与相关技术方案

2.1　注意力机制

2.1.1　多头自注意力机制

自注意力机制，Self-Attention[15]，又被称为内部注意力机制，在一般的 Encoder-Decoder 架构中均采用此种方式来计算输入内部之间的相关性。对于一个输入的句子，在获取词向量之后，输入进 Attention 层计算相关性可以得到词向量的更加丰富的表示，这个输出矩阵会包含输入句子的三层表示：单词特性、句法特征和语义特征。多头自注意力机制(Multi-Head Self Attention)[21]是自注意力机制的扩展与改进版本。多头自注意力机制，顾名思义是具备多个头，具体实现方式就是将原本词向量拆分出多个头，然后每个头各自执行 Self-Attention 计算，最终再将多个头的输出拼接在一起，从而得到词向量的最终表示。经过这种方式计算过的词向量往往比单头自注意力机制拥有更加准确的语义表示，同时空间中的容错性也会增大，更能充分表示文本特征。自注意力机制的表示见图 2-1：

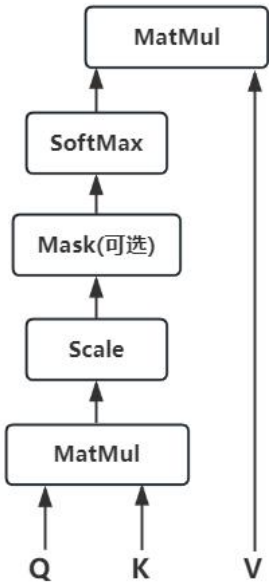

图 2-1：Self-Attention 机制

基本实现思路就是首先 Q(Query，可以理解为一个查询对象)与 K(Key，可以理解为一个查询键)，经过点积运算得到词与词之间的相似度之后再进行一次 Scale 缩放，即除以某个数值(一般为词向量特征维度 d_model)，将计算的相似度值控制在一定范围内，防止出现异常值而导致后续的 SoftMax 之后某些值无限趋近于 0 导致梯度消失。之后可

以选择是否对计算结果进行 Mask 操作，这一步主要是为了处理 Decoder 的下一词预测任务，进行 Mask 操作可以确保当前词只能获取到它与它之前词的相似度信息，从而避免了其学习到该词之后的词信息。计算之后会将该结果进行一次 SoftMax，从而将每一个词与词之间的相似度之和相加为 1。最终便可以将该结果与 V(Value)进行相乘，从而得到最终的句子表示。计算公式为如下：

$$Self\text{-}Attention = Softmax(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V \tag{2-1}$$

而多头自注意力机制可以用以下公式表示：

$$Multi\text{-}Head\text{-}Self\text{-}Attention = Linear\left(\sum_1^h ||Softmax(\frac{Q_i \cdot {K_i}^T}{\sqrt{d_i}}) \cdot V_i\right) \tag{2-2}$$

2.1.2 分组查询注意力机制

由于多头自注意力机制(MHA)涉及到大量的运算操作以及频繁更新多个权重矩阵，在实际大模型设计过程中执行这个操作是非常耗时的。为了改进 MHA，现有一般有两种解决方案，一种是采用多 Query 查询机制(multi-query attention，MQA)[22]。MQA 与 MHA 的区别主要在于其在将词向量分割到多头时，只有 Query 会进行分割，而 K,V 则直接变换到每一个头的维度，从而确保 K，V 的值在每个头是共享的，这样就只需要对 Q 进行拆分，每个头拥有自己的 Q 对象，这样可以极大减少模型的计算成本。但是 MQA 也有一个明显的缺陷，那便是精度的问题，虽然 MQA 将计算速度提升了，但是由于其减少了大量的参数，从而会引起精度的部分下降。

为了平衡好效果与计算成本之间的关系，现有的大模型技术，例如 LLaMA 一般采用另一种折中的办法-分组查询注意力机制(grouped-query attention，GQA)[23]。顾名思义，GQA 就是对 Query 头进行分组，从而保证分组内的头共用一个 K，V，然后最终再进行拼接得到注意力表示。图 2-2 展示了 MHA、MQA、GQA 的模拟图：

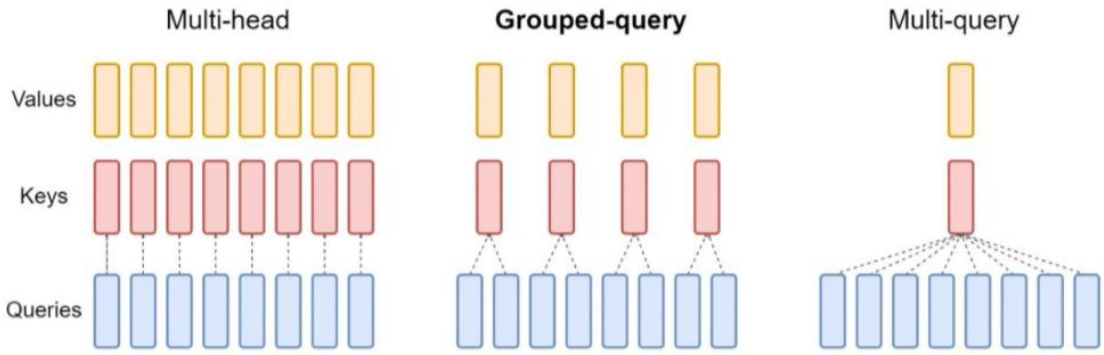

图 2-2：MHA、MQA、GQA 的模拟图

GQA 的综合 MHA 和 MQA，既不损失太多性能，又可以利用 MQA 进行推理加速，

例如上图所示就是两组 Q 共享一组 KV，从而达到分组计算的效果。

2.2 大语言模型训练流程

从 0 开始构建一个大语言模型总体分为预训练阶段和微调阶段，预训练阶段是通过大规模无标签的语料文本喂给 Decoder-only 架构的模型，让模型进行下一个 Token 的预测任务，从而可以让大语言模型了解到文本的相关知识。而微调阶段则是使用带有标记的数据对预训练模型进行对齐操作，以使得预训练模型可以针对特定任务进行调整，以此适应特定的应用场景。在预训练和微调阶段也包含了很多细节，具体见图 2-3：

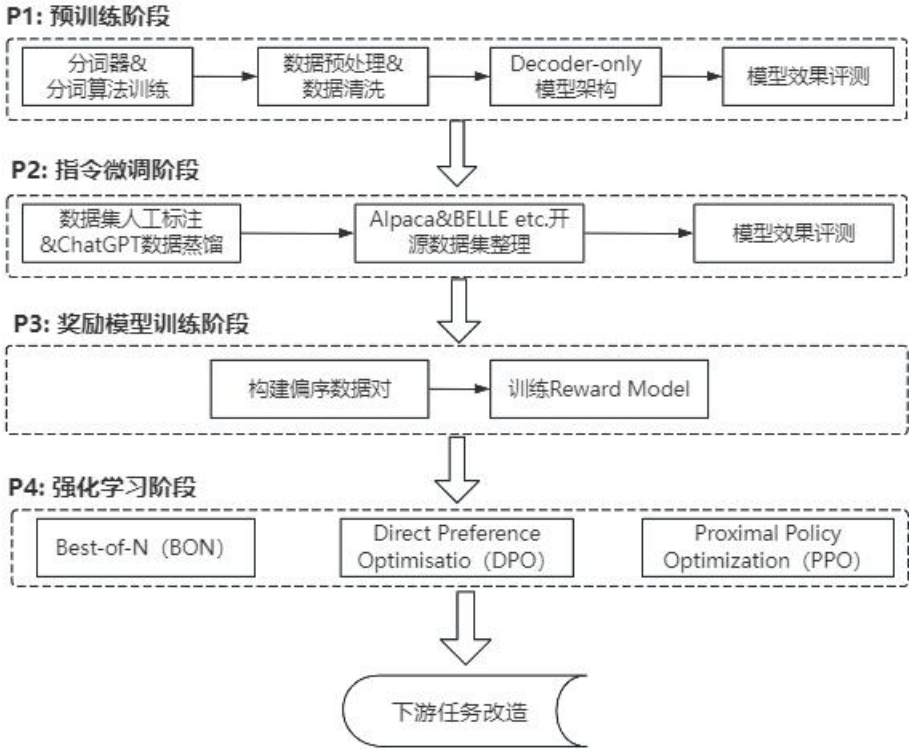

图 2-3：大模型训练整个流程

预训练阶段主要涉及四个部分：分词器训练、数据集处理、模型架构设计、训练评测。分词器训练一般有两种方式：WordPiece[24]和 Byte-level BPE(BBPE)[25]。WordPiece 和 BBPE 都是一种基于统计的分词算法。Wordpiece 将词汇分解为最小的有意义子词单元，同时加上一些特殊字符。在合并词库阶段，Wordpiece 算法会根据语料库中的词频信息，从最频繁出现的字词开始合并，逐步合并成一个新的词汇表。当不再增加新的字词或者词库达到限定额度时这个过程便会停止。Wordpiece 算法在合并过程中会考虑到词频和单词的频繁性，得出单词与词频出现频率的概率情况。而 BBPE 是在字符级别进行操作，逐步合成字节表示，从而生成词汇表和分词编码。BBPE 和 Wordpiece 算法在合并过程中不同的是，其不需要考虑词频信息，仅仅需要根据单词的出现频率和统计信息进行合并。现有的大语言模型一般均采用 BBPE 训练自己的分词器。

数据处理阶段主要是指如何将文本语料信息进行向量化，这个过程一般是将长的文本信息按照预先设定好的 seq_len(一般为 2048)进行切割。然后将切割后的向量喂给模型进行训练。模型结构方面一般采用 decoder-only 架构进行设计，这是因为 Decoder-only 的模型架构可以很好的执行 Next Token Prediction 任务，从而可以根据 Prompt 去预测后面的词的信息。关于预训练模型的量化评价指标，目前普遍采用 PPL、BPC 等技术。这类评测技术可以用来评估模型对于模板语言的拟合程度，通过在生成结果和目标文本之间的交叉熵上做了一些处理，可以实现给定一段话，预测其后面可能出现哪些合法通顺的字词。除此之外还要评估大模型的另一个很重要的基础能力，即知识蕴含能力。这里使用的数据集有中文知识能力测试集 C-Eval[26]，其涵盖了 1.4w 道选择题，共 52 个学科，以及 MMLU 测试集[27]，MMLU 是一个包含了 57 个子任务的英文评测数据集，涵盖了初等数学、美国历史、计算机科学、法律等，难度覆盖高中水平到专家水平，有效地衡量了人文、社科和理工等多个大类的综合知识能力。除此之外，还有 MATH 测试集、CMMLU 测试集等多种测试基库，可以充分评测大模型的基础能力。

指令微调阶段一般涉及三部分：数据集人工标注和 ChatGPT 知识蒸馏、开源监督微调数据集整理、模型训练和测试。由于预训练的本质还是在于下一个 token 预测，即续写能力。但是这种方式并不一定可以很好地回答用户输入的问题，也无法确保数据格式采用一问一答的格式。因此需要整理问答数据对预训练好的模型进行指令微调，即指令对齐操作。而指令对齐需要大量的标注数据集，该数据集的来源一般有两种：人工标注和 ChatGPT 知识蒸馏。但是人工标注需要耗费大量的人力物力，这个过程一般实现起来不太现实，因此大部分通过 ChatGPT 的输入输出内容来蒸馏出自己的模型，例如 stanford alpaca[28]。除此之外也有一些开源的指令微调数据集，例如 alpaca、BELLE 等也可以被应用其中。监督微调之后的大模型也需要进行评测，以评估其回答问题的能力。现有评测技术一般为人工评测＋GPT4 在线评测两种方式进行。人工评测就是人工去评测哪一种模型生成效果更好，但是这种评价方式非常主观，而且评测数据集很大的情况下去人工评测显然不太现实。而利用 GPT-4 进行评测就是同样的 Prompt 输入，获得多个模型的输出结果，构造成[ChatGPT4 答案-候选模型答案]的格式对，将其输入进 GPT-4 进行打分，从而比较监督微调模型的效果。

在监督微调过程中，我们仅仅为模型提供了“好的”数据，而没有为其准备“不好”的数据，这就可能会导致预训练模型中原本的错误或者有害的信息（例如违法信息或者不良发言）没有在监督微调过程中被纠正，从而不能保证大模型的无害性，并且也可能

会出现一系列“幻觉”的问题。因此需要人工构建一个奖励模型，根据构造的“偏序对”来训练模型，偏序对不直接为每个样本进行打分，而是标注这些样本信息的好坏顺序，从而最大化“好样本”和“差样本”之间的分差，进而学会为每一个生成的回答句子进行判分。

在训练好奖励模型之后，接下来便可以利用奖励模型进行强化学习，进而进一步强化我们的模型。目前主流的强化学习一般为三种：Best-of-N（BON）[29]、Direct Preference Optimisatio（DPO）[30]和 Proximal Policy Optimization（PPO）[31]。BON 的基本思想是让 SFT(监督微调)好的模型生成多个回答，然后使用训练好的奖励模型从这些回答中选择得分较高额回复再次训练模型。然后不断地进行迭代优化，著名的 LLaMA2 便是采用这种方式。DPO 不需要奖励模型的参与，它可以直接利用训练奖励函数的偏序对来训练模型本身，基本实现方式就是借鉴了对比学习的思路。对于同一个 prompt，尽可能地拉开优秀的回复和差的回复之间的生成概率，从而使得模型更加偏向于生成高质量回复。PPO 通过引入重要性采样因子来缓解 on policy 模型一次采样只能更新以此模型的问题，提高了数据的利用效率和模型的迭代训练速度。

2.3 主流开源大语言模型

2.3.1 LLaMA2 结构

LLaMA2 模型是 Meta AI 公司发布的开源大语言模型，其在开放基准上有着非常出色的表现，是迄今为止最为流行的开源模型之一。LLaMA 模型整体借鉴 Transformer 的 Decoder 架构，后续的很多大模型架构都是借鉴 LLaMA 的结构。LLaMA 与 GPT 系列一样，采用了 Transformer 的堆叠 Decoder 结构，在海量文本上进行了无监督预训练，其目标是通过上下文预测下一个词。LLaMA 的训练语料主要以英语为主，但也包含其他拉丁语系语言。在分词方面，LLaMA 采用了 sentencepiece 实现的 Byte-level BPE 对语料进行了分词编码。相较于标准的 Transformer 模型，LLaMA 在局部结构上进行了调整，借鉴了同期其他研究成果。这些调整包括前置层归一化和 RMSNorm[32]归一化函数的使用、门控线性单元和 SwiGLU 激活函数的引入，以及旋转位置编码（RoPE）的应用。除此之外，LLaMA2 的注意力层使用了 GQA，GQA 的细节见 2.1.2 节。Llama 的模型结构如图 2-4，其中 LlamaDecoderLayer 堆叠了 32 层。

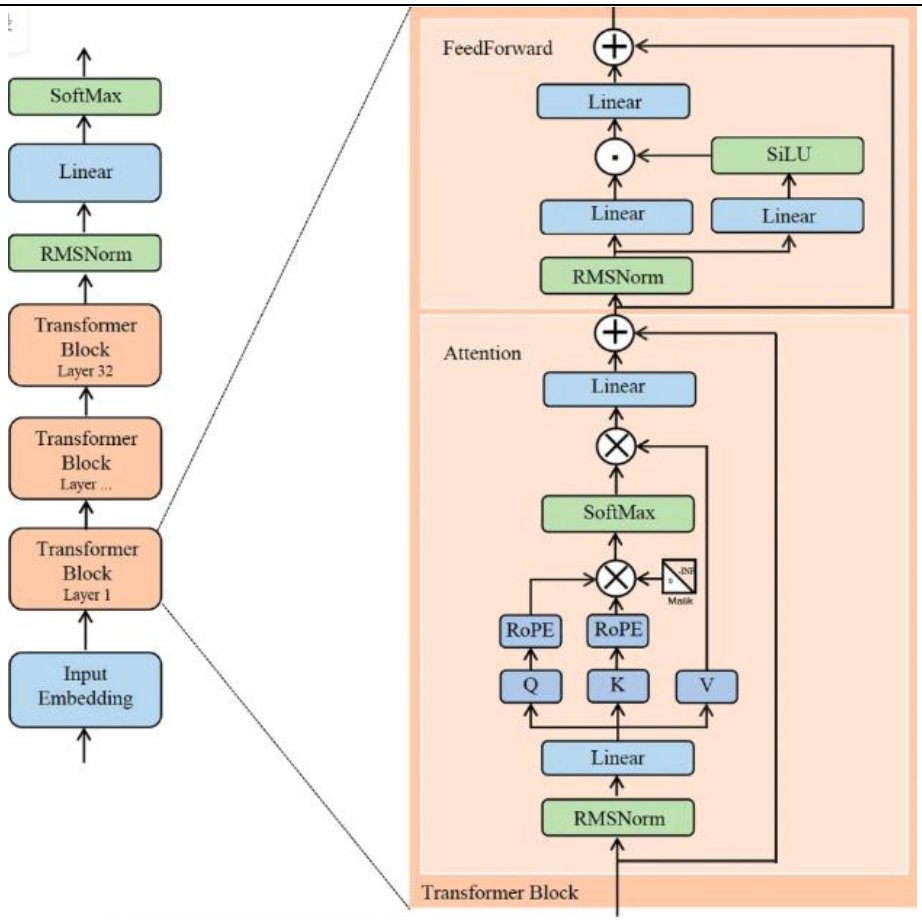

图 2-4：LLaMA 结构设计

2.3.2 ChatGLM3 结构

ChatGLM3 是由智谱 AI 和清华大学 KEG 实验室联合开发的一款新一代预训练对话模型。其中 ChatGLM3-6B-Base 上下文限制为 8K。对话版本模型 Chatglm3-6B-Chat 模型上下文限制为 8K。ChatGLM3 整体上沿用 LLaMA2 的架构，即 Decoder-only 架构，归一化层采用 RMSNorm，不同的点在于 ChatGLM3 的注意力机制计算部分采用 MQA，同时其词库大小为 65024，几乎都是采用中文语料进行预训练。ChatGLM3-6B 的基础模型 ChatGLM3-6B-Base 应用了更多样的训练数据，更充分的训练步数以及更加合理的训练策略。在语文、数学、推理等不同方面的数据集评测显示，其在 10B 以下的模型中具备最为强大的性能。除此之外，ChatGLM3-6B 采用全新的 Prompt 格式设计，除了支持正常的多轮对话以外，还支持原生工具调用、Agent 任务场景以及代码执行等复杂场景。

2.3.3 Qwen 结构

通义千问模型是由阿里发布的一系列大语言对话模型，Qwen 是一个全能的语言模型系列，包含各种参数量的模型，如基座模型 Qwen（基础预训练语言模型）和 Qwen-Chat（聊天模型，该模型采用人类对齐技术进行微调）。在众多下游任务中，基座模型始终表现出卓越的性能，而聊天模型，尤其是使用人类反馈强化学习（RLHF）训练的模型，

具有很强的竞争力。Qwen-Chat 拥有先进的工具使用和规划能力，可用于创建 agent 应用程序。即使在执行复杂任务，如使用代码解释器等。

Qwen 系列模型也是采用了 LLaMA2 模型的整体架构，其位置编码选择 RopE，归一化层方面也是采用了 RMSNorm 替代 LayerNorm。除此之外，为了在内存成本上取得更好的效果，Qwen 选择了非约束的嵌入方法，而不是简单的将输入 Embedding 与输出 projection 的权重进行绑定。

Qwen 模型在多轮对话中拥有显著优势。首先，它具备强大的上下文理解能力，通过在大规模文本数据上进行预训练，能够充分理解和处理复杂的对话上下文，从而使得对话更为连贯。其次，由于在训练过程中充分考虑了上下文信息，Qwen 模型生成的对话回复更加自然流畅，能够更好地模拟人类对话的风格和习惯。此外，Qwen 模型在预训练阶段接触了大量文本数据，积累了丰富的知识库，能够为对话提供多领域、多样化的信息和答案。更重要的是，Qwen 模型能够通过微调和人类反馈强化学习等技术实现个性化交互，模仿和学习用户的个性化对话风格，使得对话更贴近用户需求和偏好。最后，Qwen 模型还具备先进的对话管理能力，在聊天模型中能够有效地管理和控制对话的流程，使得对话更加有条不紊，避免了歧义和混乱。

2.4 SFT 监督微调方法

SFT（Scalable Fine-Tuning）是一种用于自然语言处理的技术，其核心思想在于对预训练的大型语言模型进行微调，以使其适应于特定的任务。这些大型预训练模型，例如 LLAMA、GPT 等，拥有数十亿甚至数百亿个参数，能够处理大规模的文本数据。SFT 的基本概念是在这些大型预训练模型的基础上，对其进行微调，以适应具体的任务。在微调过程中，模型会根据任务的特性进行参数和结构的调整，以提升在该任务上的性能。现有主流的 SFT 技术包括 Lora 微调、全参微调和 P-Tuning 微调。

2.4.1 Lora 微调

LoRA[33]（Low-Rank Adaptation）是一种用于大模型高效微调的方法，通过优化参数的低秩性质，间接地训练一些密集层，同时保持预训练权重不变。这一方法使得训练更加高效，降低了硬件门槛，并且可以将预训练模型共享以构建多个小型 LoRA 模块，从而显著降低存储需求和任务切换的开销。LoRA 的线性设计允许在部署时将可训练矩阵与冻结权重合并，不会引入推理延迟，与完全微调模型相比具有更高的推理效率。此外，LoRA 与许多先前的方法是正交的，并且可以与它们结合使用，为模型微调提供了更多的灵活性和多样性。Lora 的实现思想很简单，就是只训练两个低秩矩阵，A 和 B。LoRA

的具体做法是在网络中添加一个旁路结构，旁路由两个矩阵 *A* 和 *B* 相乘组成。其中，矩阵 *A* 的维度是 *dxr*，矩阵 *B* 的维度是 *rxd*，通常情况下，*r* 的取值很小，如 1、2、4 或 8。这样旁路的参数量会远远小于原网络中的参数 *W*。在 LoRA 的训练过程中，我们冻结了原网络的参数 *W*，只对旁路的参数 *A* 和 *B* 进行训练。由于 *A* 和 *B* 的参数量远远小于 *W*，因此训练时所需的显存开销大约等于推理时的开销。对于使用 Adam 优化器的情况，所需的显存开销大约相当于全参数微调的 1/3，这大大降低了训练的成本，在大模型实际应用中，Lora 通路一般应用在注意力机制部分，即微调过程中冻结注意力机制的 *Wq,Wk,Wv,Wo* 的参数，而使用两个低秩矩阵进行替代，从而在反向传播更新参数时只对低秩矩阵 *A* 和 *B* 进行更新。具体思想如图 2-5。

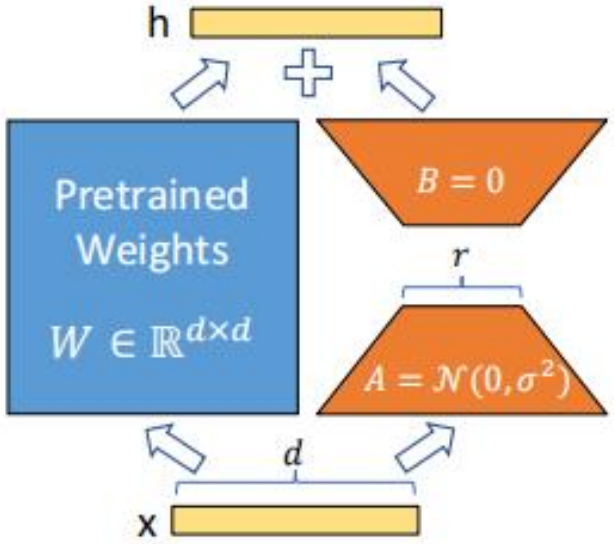

图 2-5：lora 的实现思想

其中在对 *A* 和 *B* 初始化时，A 进行高斯初始化，B 初始化为 0。这样是为了保证这个通路在最开始微调时为 0，即没有参数，不对模型的输出造成任何影响。

2.4.2 全参数微调

全参数微调是一种微调大模型的全部参数的做法，全参数微调是指在微调过程中，对预训练模型的所有参数进行更新，包括所有的权重和偏置。其原理是在特定任务的数据集上，通过反向传播算法和优化器（如随机梯度下降）来调整模型的参数，使得模型在该任务上表现更好。全参数微调的优势在于能够最大程度地利用预训练模型的知识，因为所有参数都可以根据新任务的数据进行调整。这样的微调可以使得模型在特定任务上表现更为精准，因为它能够针对性地调整所有参数以最大化性能。

2.4.3 P-Tuning v2 微调

P-Tuning v2[34]是除 Lora 之外的另一种高效微调大语言模型的微调方法，其原理是在

预训练模型的基础之上添加少量的可训练参数，从而对模型的输出进行微调。P-Tuning v2 的优化策略一般主要包括两个方面：一个是前缀提示策略，即将提示信息添加到模型的每一层之中，以提高模型输出的稳定性；二是采用新的自适应优化策略，其可以根据模型在训练过程中的表现动态调整微调参数的权重，以提高模型的收敛速度和性能。P-tuning v2 微调方法显著提升了性能，其特点在于仅需精细调整 0.1%的参数量，同时保持语言模型的其余参数固定。这一方法在不同规模的语言模型上都展现出了与 Fine-tuning 相当的性能，从而成功解决了 P-tuning v1 在参数量较少的模型中微调效果不佳的问题。换句话说，P-tuning v2 通过专注于微调极少量的参数，就能够在各种规模的语言模型上实现与全面微调相匹敌的效果，从而克服了 P-tuning v1 在参数较少模型中的局限性。其中 P-tuning v2 和 P-tuning v1 的对比图如图 2-6：

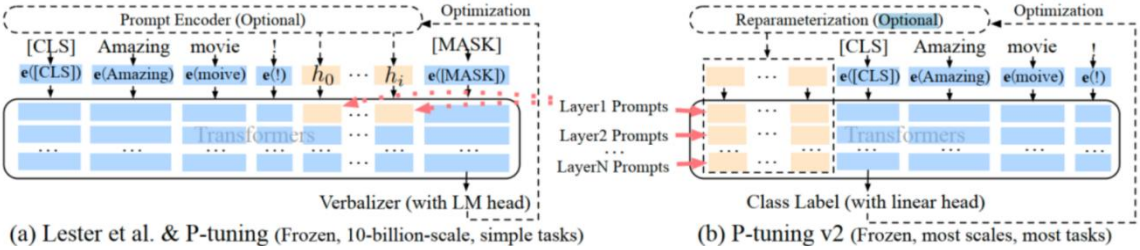

图 2-6：P-Tuning v1 和 P-Tuning v2

与 P-tuning v1 不同的是，P-tuning v2 在每一层都加入了 Prompt tokens 作为输入，而不是仅仅在输入层进行添加，这确保了其拥有了更多的可学习参数。除此之外，由于该 prompt tokens 加入到了更深层次的结构中，这可以给模型推理带来更加直接的影响。

2.5 RAG 检索增强技术

(Retrieval Augmented Generation) RAG[35]检索增强，首次提出于 2020 年，其是一种端到端的实现方案，它结合预训练的检索器和大语言模型生成器。RAG 后期主要聚焦于提升大预言模型(Large Language Models, LLM)的推理能力，即通过整合外部知识来获得更好的生成效果。由于现有的 LLM 是在很大规模的预训练预料上进行学习的，它们往往具备很强的泛化能力，即对各种事物都有所了解，但是又很难达到专家级别的问答效果，尤其是在一些比较冷门或者专有的领域，例如法律、医学等，效果不是特别理想，甚至还会出现“幻觉”(即 LLM 不了解，但是乱说，从而生成一系列错误回复)。而 RAG 可以通过整合来自外部知识库中的信息，从而将该信息在向 LLM 输入 Prompt 时进行注入，从而使得 LLM 生成更加准确，更具备上下文意识的答案，同时生成更加专业的回复。RAG 通常和 LLM 一起存在，辅助 LLM 完成一些专有领域知识问答，现有的 RAG 检索增强一般利用 langchain 进行实现。其实现方式如图 2-7(以 Langchain＋ChatGLM 为

例)：

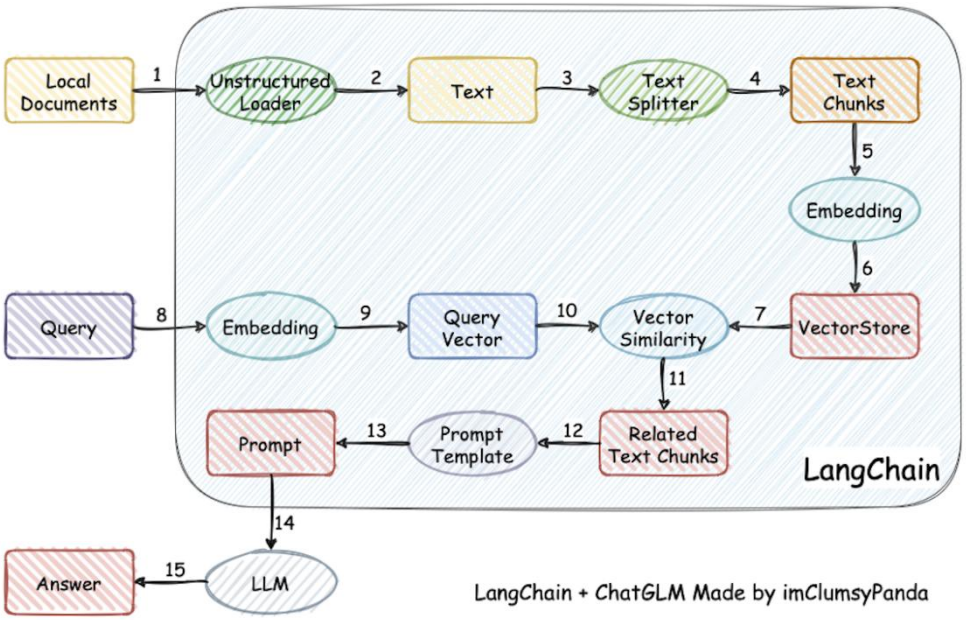

图 2-7：LangChain 实现 RAG 的整体流程

其运行的整体流程为：1.加载文件、2.读取文档(一般为专有领域知识，即想要让 LLM 知道的垂直领域知识)、3.文本分割、4.文本向量化、5.用户输入的问句向量化、6.在文本向量中匹配出与文具向量最为相似的 top k 个(可指定)文本内容、7.将匹配出的文本内容作为上下文和问题一起添加到 Prompt 中、8.将综合 Prompt 输入给 LLM 从而生成相应回复内容。

其中 RAG 最为主要的三个过程为：索引、检索和生成。索引是关键的初始步骤，从清洗和提取原始数据开始，转换为标准化的纯文本，然后切分成更小的片段。这些片段通过嵌入模型转换为向量表示，并存储在索引中，以实现高效的搜索能力。检索阶段使用用户查询，通过编码模型生成语义相关的嵌入，然后在向量数据库上进行相似性搜索，检索最接近的 top k 数据对象。生成阶段将用户查询和检索到的上下文填充到提示模板中，然后将增强提示输入到 LLM。

2.6 基于 Bert embedding 的文本编码

NLP 中常见的文本编码有 one-hot 编码、word2vec 编码[36]、embedding 编码[37]等。其中 one-hot 编码又被称之为独热编码，其将每个词表示成为具有 n 个元素的向量，且 n 个元素代表所有可能出现的词，在 one-hot 编码形成的词向量中，只有该元素对应的位置被设置为 1，其余元素均被置为 0。例如“我是程序员”可以被编码为[“我”,“是”,

“程序员”]，进而对应的 one-hot 编码为：[1,0,1,1,0,.....]。但是这种编码方式的缺陷非常明显，首先是当要表示的词的范围很大时，这个词向量的长度就会非常大。其次是这种直接对应位置编码的方式无法利用余弦相似度来衡量词与词之间的关系，不利于上下文信息建模。

为了解决 one-hot 的词向量过长、无法表征词与词之间的关系等问题，word2vec 应运而生，其中 CBOW 模型是 word2vec 推出的一个神经网络模型，其训练时的主要思想是利用输入文本上下文得到某个词。其实现的基本思路如图 2-8：

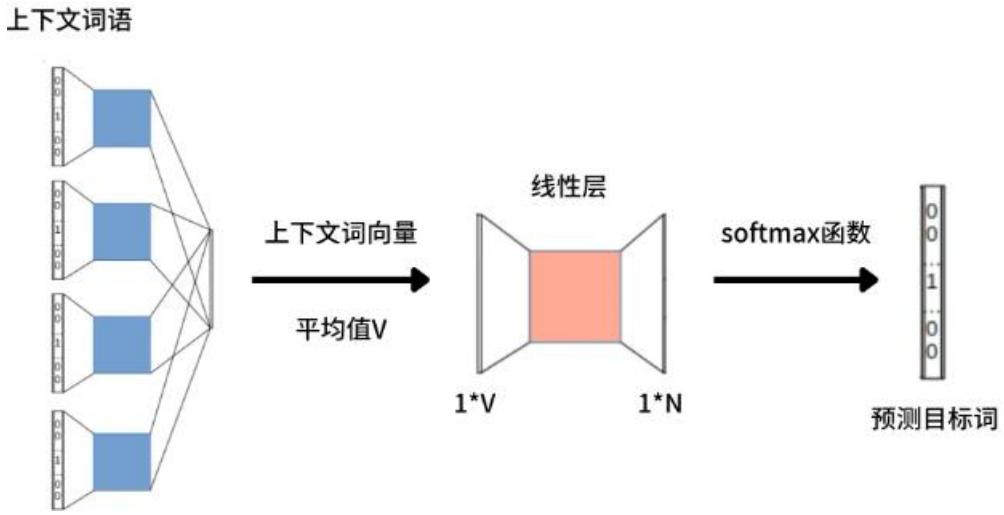

图 2-8：CBOW 实现思路

其基本流程就是根据上下文进行输入，例如输入 $X=[x_1,x_2,x_4,\ldots,x_d]$表示的是一个上下文输入，其中有 d-1 个词，基本任务是根据 X 预测x_3的词向量表示，通过这个过程去训练 CBOW 模型。首先 X 中的每个词都会转成 one-hot 编码，这是一个 1*N 的矩阵，然后经过 N*V 的 embedding 权重编码 Q 后每个词都会得到其新的编码 $V=[v_1,v_2,v_4,\ldots,v_d]$，之后会将 V 进行对应位置求和然后取平均，从而得到上下文表示的词向量 1*V。将这个词向量输入线性层，经过处理之后会得到 1*N 的新的矩阵，最后经过 softmax 处理之后取最大位置向量作为预测词对应 one-hot 中的位置编码，从而得到该词的信息。

CBOW 很好的利用了上下文词向量信息，但是由于其在输入神经网络时是对应位置求和取平均，这就导致了其不能处理多义词的问题，比如“苹果”这个词，可以代表“苹果公司”，也可以代表“可以吃的苹果”。但是 CBOW 经过训练之后的 embedding 就仅仅会将其表示为同一个词向量，这显然是不合理的。自 Transformer 的架构得到普及之后，BERT 借助注意力机制很好的解决了这个问题。

BERT[38]的成功之处在于引入了可训练的位置编码信息和多层 encoder 来进行上下

文建模。其基本思路是首先根据输入文本转成 embedding 信息，这个 embedding 设计如图 2-9 所示：

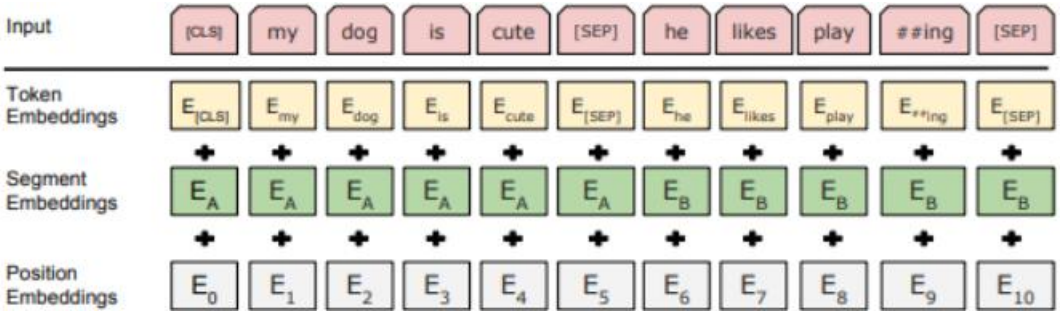

图 2-9：BERT embedding 编码

Bert embedding 包含三层信息，分别是 Token embedding(词嵌入)、Segment embedding(段嵌入)、Position embedding(位置嵌入)。这三者相加便构成了 bert 的词嵌入的编码形式。编码之后的 embedding 信息会输入 Encoder 中，以对上下文进行建模。Encoder 结构如图 2-10 所示：

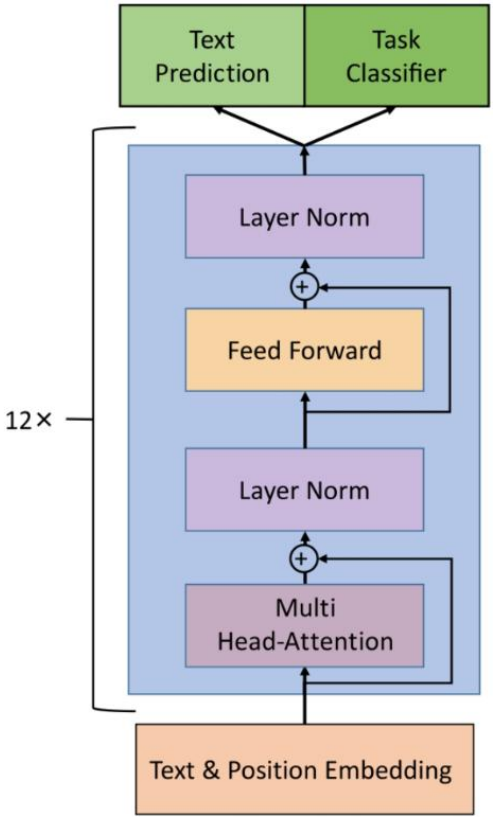

图 2-10：BERT 架构

BERT 的 Encoder 共有 12 层，每一层包含有多头自注意力机制和前馈神经网络两个子层，每个子层之间又进行了残差链接以及 Layer Norm 层归一化。经过 Encoder 处理之后可以取得[CLS]的第一个词向量 768 维数据作为整个句子表示。

2.7 本章小结

本章对本文相关研究内容以及相关技术方案展开介绍。其中介绍了大模型训练的整体流程，包含了预训练过程以及 SFT 微调过程。除此之外也对大模型常用的注意力机制进行了介绍，包括 MHA、MQA、GQA 的基本原理以及区别。本章还对现有主流开源模型进行了介绍，解释了现有主流开源模型基本都是沿用 LLaMA2 的架构形式。另外，本章介绍了现有主流的 SFT 方法，包括 LORA 微调、全参微调以及 P-Tuning v2 微调。最后介绍了 RAG 检索增强相关技术以及基于 BERT embedding 的词向量表示技术。

第 3 章　模型设计方案与实验细节介绍

3.1　多轮对话 AI 调解员模型

3.1.1　预训练模型选择

在训练 AI 调解员时，我们选择了目前主流的开源预训练模型，分别是 LLaMA2、ChatGLM3、Qwen 模型。因为这些模型都是采用 Transformer Decoder-only 架构，且在各种 Benchmark 评测基准上取得了优异的成绩。这些模型具备各种参数量的基准预训练参数选择，且具有丰富的语言理解能力，能够在多轮对话和复杂语境的调解场景中提供高质量的对话体验，并且它们在预训练阶段已经积累了大量的语言知识，有助于在微调过程中更快地适应特定领域的语境和需求。同时，为了进一步使得预训练模型更加贴向于我们的调解场景，我们选择了它们的 Chat 版本模型。Chat 版本经过了专门的预训练，针对对话生成任务进行了优化，具有更好的对话生成能力和语言表达能力。除此之外，为了评估不同参数量大小的预训练模型的效果，我们选择了 1.3B、1.8B、6B、7B 的参数量来进行实验。考虑到我们的对话场景是一个中文场景，所以中文版本 Chinese-LLaMA，这个版本的预训练模型在中文语料库中对中文进行了扩充。最终我们选定了五个基准预训练模型：Chinese-LLaMA2-1.3B-Chat、Chinese-LLaMA2-7B-Chat、ChatGLM3-6B-Chat、Qwen-1.8B-Chat、Qwen-7B-Chat。模型介绍见表 3-1：

表 3-1：预训练模型介绍

LLM 名称	开源地址	预训练语料
ChatGLM3-6B	https://huggingface.co/THUDM/chatglm3-6b	中文、英文
QWen-1.8B	https://huggingface.co/Qwen	多语言(以中英文为主)
QWen-7B	https://huggingface.co/Qwen	多语言(以中英文为主)
Chinese-LLaMA2-1.3B	https://huggingface.co/meta-llama	英文为主、后续拓展中文语料
Chinese-LLaMA2-7B	https://huggingface.co/meta-llama	英文为主、后续拓展中文语料

3.1.2 SFT 微调方案

在实际 SFT 微调训练 AI 调解员的过程中，我们选择了 LORA 微调和全参微调这两种方式，这是因为 LORA 微调能够有效地利用有限的显存资源，并且在数据量较小的情况下能够实现模型的快速适应性调整，这有助于在资源有限的情况下快速微调并部署模型。而全参微调则能够充分地利用数据资源，通过调整预训练模型的全部参数，并采用 deepspeed[39]分布式训练框架对预训练模型进行微调，可以提升模型的性能和泛化能力。

3.1.3 RAG 检索增强

检索增强生成（Retrieval Augmented Generation, RAG）是一种技术，它通过从数据源中检索信息来辅助大语言模型（Large Language Model, LLM）生成答案。简而言之，RAG 结合了搜索技术和大语言模型的提示词功能，即向模型提出问题，并以搜索算法找到的信息作为背景上下文，这些查询和检索到的上下文信息都会被整合进发送给大语言模型的提示中。由于 AI 调解员涉及到一些专业知识问答，仅仅依靠 SFT 技术可能不能很好地完成 AI 调解员解答工作，因此本文集成了 langchain 检索增强框架，对一些专业问答话术进行了检索增强，并利用 bge-large-zh-v1.5 来对专业知识库文档进行向量化，并将其存储在 faiss 向量库中。我们的专有 AI 调解员知识库文档格式如下：

Question1:你们是怎么知道我这个号码的？/你们侵犯了我的隐私，这个号码没人知道。

Answer1:号码是法院系统提供的，为保护你的隐私已采取打码展示，我们拨出你的电话时是看不到完整号码的，作为你案件的调解员也只负责拨打。至于信息安全性你完全不用担心，因为我们的工作人员是无法通过任何个人途径联系到你的。

Question1:来电为什么不是你提供给我的法院电话？/为什么会被标记为骚扰电话？/那你怎么没有用法院的电话给我打？

Answer1:被标记为骚扰很正常，我们也很无奈。因为一些当事人缺乏清偿能力，在逃避心理下就会对法院的调解电话进行标记，如果多人标记，尽管是法院的号码也会被标记为骚扰电话。你可以放心拨打全国司法热线查询核实你和原告的债务纠纷。另外，调解员和当事人沟通并不会产生任何财产性利益纠纷，程序透明，你怎么借就怎么还。

Question: .....

Answer: .....

在实际应用过程中，上面的知识库问答会通过 bge-large-zh-v1.5 模型来对文本分块并向量化，最后会将结果存储在 faiss 向量库中。在当事人进行提问时，会首先去 langchain 向量库中检索相应的 top k 个文本片段，然后将这些片段连同用户输入一起组成 prompt

输入给 AI 调解员模型，让其进行解答。实现方式如图 3-1：

图 3-1：RAG 检索增强

上图展示了用户提问“你这个电话为什么会被标记为诈骗电话？”然后知识库匹配结果，将匹配出的文本输入给模型，最终模型给出回复的整个实现流程。

3.1.4 SFT 数据集介绍

模型微调的数据集主要来自于“上海徐汇亦法云调解中心”内部的真人沟通数据，里面记录着每次调解员和当事人进行调解的通话数据。其中训练集共 5263 条，测试集共 583 条。每条通话记录中包含着当前轮次通话内容与历史轮次沟通内容，确保模型可以根据当前输入与历史沟通内容两者结合生成当前轮次的回复内容。本课题中，数据集的格式如下：

```
"instruction": "嗯，没错，我没有欠这么多钱。",
"input": "",
"output": "嗯，您指的是花呗借呗是吗？",
"history": [
["林兴建由于个人日常消费，分 4 次从借呗总共贷款了 4564.83 元，总利息为 172.25 元。逾期 1131 天，总罚息为 3887.24 元。目前共欠借呗 8624.32 元。",
```

"您好，林新建先生在吗？"],
["说吧。","您好，请问您能听到林新建先生吗？"],
["没有收到。","我们是法院调解中心，此次通话涉及您名下的金融借款合同纠纷案件。"],
["我现在没空，但是我也没欠那么多钱，我现在告诉你，我也没空。","嗯，那目前来看，您能满足他的主张诉求吗？他的一个主张诉求是要求您偿还借款。"]]

关于数据预处理，我们对数据首先转换成 alpaca 格式：

```
[
  {
    "instruction": "用户指令（必填）",
    "input": "用户输入（选填）",
    "output": "模型回答（必填）",
    "history": [
      ["第一轮指令（选填）", "第一轮回答（选填）"],
      ["第二轮指令（选填）", "第二轮回答（选填）"]
    ]
  }
]
```

然后由于该数据是语音转文本的数据，所以可能会存在一些错误，因此我们使用 ChatGPT-3.5 接口对其数据错别字进行修正。上文样例时已经按照 alpaca 格式进行修正的结果。

3.1.5 实验环境及参数介绍

本次实验均在 V100 上进行，其显存大小为 32G×4。对于 LORA 微调，我们采用单卡训练，其中 7B 的模型微调大约占用显存 16GB，其中训练 batch_size 我们选择为 4，训练轮数 epoch 选择为 2，学习率设置为 4e-5。全参数微调时我们采用 Deepspeed 框架进行分布式训练，7B 的模型微调大约占用显存 120GB，其中的 batch_size 设置为 4，训练轮数 epoch 选择为 4，学习率设置为 5e-5。

3.2 当事人还款意愿预测模型

3.2.1 模型设计

模型的词向量编码采用 BERT 的 embedding 编码，首先获取第一次当事人与用户通话的全部内容 $X = [x_1, x_2, x_3, \ldots, x_n]$,其中 n 为当事人说话次数，经过 BERT Embedding 处理之后将对应为词向量编码，将每句话的编码输入进 BERT 的 Encoder 之后获取到每

句话的[CLS]输出 $Y=[y_1,y_2,y_3,\ldots,y_n]$。其中每个 y 均是一个 1×768 维的向量。最终其对应位置相加取平均之后得到最终的用户特征表示 $Z=[z_1,z_2,z_3,\ldots,z_{768}]$。最终将这个 768 维的向量输入进分类器进行训练，从而预测当前当事人是否未来可以被调解成功，从而决定是否后续需要真人调解员介入，其中为了验证我们 Mean_BERT_LSTM_MLP 模型的有效性，我们对比训练了三个 base 分类器模型，分别是线性分类器、RNN 分类器和 LSTM 分类器[40]。线性分类器是一种简单而高效的机器学习模型，它通过找到一个线性函数来将输入数据映射到不同的类别。它的优势在于计算速度快，且易于理解和实现。然而，它可能不适用于处理非线性问题或复杂的数据结构。LSTM 是一种特殊的循环神经网络（RNN），它能够有效地处理序列数据中的长期依赖关系。通过捕捉序列中的时间依赖性，LSTM 分类器在文本分类、语音识别等领域展现出强大的性能。RNN 具有处理序列数据的能力，能够捕捉数据中的时间依赖性。尽管在某些任务上可能不如 LSTM 表现优秀，但 RNN 依然是一种重要的分类器，特别是在处理较短的序列数据时效果显著。在本文中，我们提出了一种 Mean_BERT_LSTM_MLP 模型，首先将 mean embedding 作为每个时间步的输入，然后取得所有时间步信息进行平均，最后将该向量输入 MLP 层，首先进行上采样，获得向量更加深层次的表征信息，之后进行下采样，对向量维度进行压缩表征，最后通过一个线性分类器来进行分类。的整体架构如图 3-2 所示：

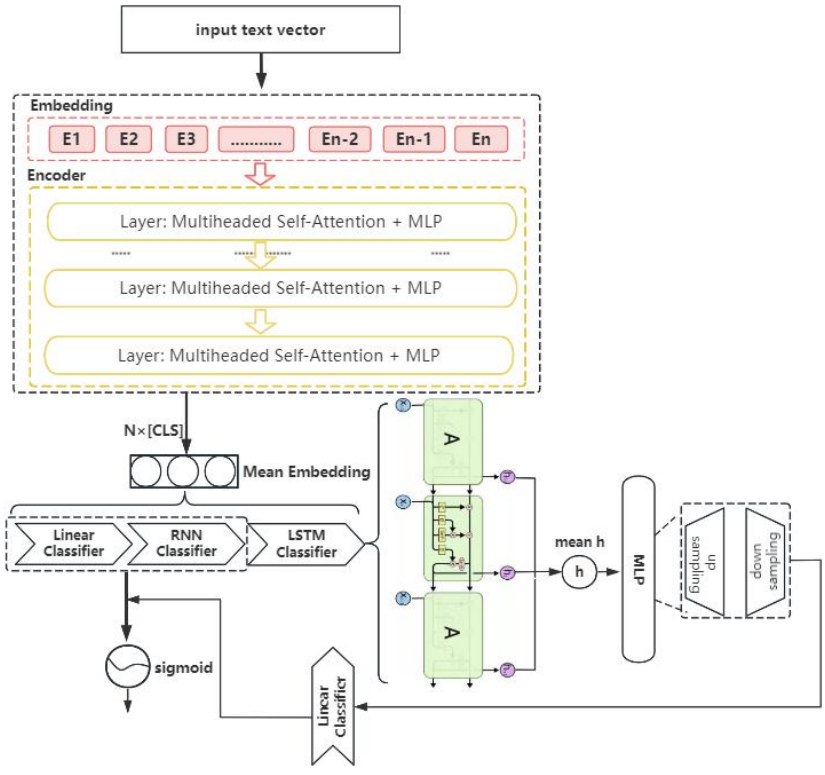

图 3-2：当事人还款意愿预测模型

3.2.2 数据集介绍

该部分数据集同样来自于语音转文本的通话数据，只不过该数据只取第一次通话内容，然后根据第一次通话内容预测该当事人是否可以成功还款，即是否还有必要对其进行二次真人调解员介入拨打。该数据集的格式如下：

Sample 1:

Text: ['喂。', '生活压力大。', '这个你能不能给我把我的欠款？', '有啊。', '呃，把这个把我欠款整合起来，给我搞一个分期，我分期还嘛，每个月。', '嗯，一年吧。', '生活生活压力就小一点嘛。', '我我在听啊我一直在听啊。', '嗯，可以。', '我在我在一直都在啊。']

Label: 1

Sample 2:

Text: ['不行。 喂。', '哎，你好。', '嗯，对。', '啊，你说。', '我不是说了，我今今年顾不上。', '对。', '那行，我知道了。']

Label: 0

Text 是用户说话的列表，label 代表最终成功调解的标签。label 为 0 代表最终未成功调解的标签。训练集的正样本共 1837 条，负样本共 1540 条。测试集的正样本共 460 条，负样本 97 条。

3.2.3 实验环境及参数介绍

本实验在 3090 显卡上进行训练测试，在实验中我们使用 Adam 优化器进行参数更新优化，我们使用 12 层得 transformer encoder 架构，隐藏层大小为 768 维，并且多头自注意力机制采用 12 头进行计算。损失函数使用 BCE 二分类交叉熵损失函数。

3.3 本章小结

本章主要介绍了 AI 调解员预训练模型的选型以及当事人还款意愿模型的设计。其中分别介绍了 AI 调解员的微调方案设计，RAG 检索增强设计，数据集结构以及实验环境。对于当事人还款意愿模型，我们介绍了 BERT 词向量技术，以及采用的分类器模型架构，同时为了展示我们的 Mean_BERT_LSTM_MLP 模型的效果，我们对比了传统的 BERT+Linear 模型、BERT+LSTM 和 BERT+RNN 模型，同时也对训练和测试数据集进行了介绍。

第 4 章　实验结果与分析

4.1　AI 调解员模型实验结果

4.1.1　评估指标

现有的对话模型的评价主要分为两种：自动评价和人工评价。人工评价方法虽然评价结果比较可靠,但是也有很多缺点，比如人力物力消耗大,花费时间长,不能方便快速地对对话机器人进行评价从而促进机器人的快速迭代。除此之外，评价者的个人偏好对评价结果也有一定的影响，不能克服人工评价的主观因素。这里我们评估对话模型时引入常用的来自机器翻译任务的 BLEU[41]和来自摘要任务的 ROUGE[42]评估指标进行评估。这一类指标可以衡量生成文本与参考回复之间的相似匹配程度，从而可以反映出模型预测推理的语意准确性。

BLEU(Bilingual Evaluation Understudy，双语评估基准）是一组度量机器翻译和自然语言生成模型性能的评估指标。BLEU 指标是由 IBM 公司提出的一种模型评估方法,以便在机器翻译领域中开发更好的翻译模型。BLEU 指标根据生成的句子与人工参考句子之间的词、短语和 n-gram 匹配来计算模型的性能。BLEU 指标通常在 0 和 1 之间取值，其中 1 表示完美匹配。

BLEU 指标用于分析候选译文中有多少 n 元词组出现在参考译文中（就是在判断两个句子的相似程度）。BLEU 有许多变种，根据 n-gram 可以划分成多种评价指标，常见的评价指标有 BLEU-1、BLEU-2、BLEU-3、BLEU-4 四种，其中 n-gram 指的是连续的单词个数为 n，BLEU-1 衡量的是单词级别的准确性，更高阶的 BLEU 可以衡量句子的流畅性。在本课题中我们使用 BLEU-4 来进行评价。假设 *Ci* 表示机器译文，也即本课题中微调后的 LLM 模型生成的回复，该机器译文对应的一组参考译文，也即人工真正的回复可以表示为：$Si = \{si1, si2, ..., sim\}$，将 *Ci* 所有相邻的 n 个单词提取出来组成一个集合 n-gram，一般取 n = 1,2,3,4。我们可以用 *Wk* 表示 n-gram 的第 k 个词组，使用 *hk(Ci)* 表示第 k 个词组在 *Ci* 中出现的次数，*hk(sij)*表示第 k 个词组 *Wk* 在 Sij 中出现的次数。因此，在 n-gram 计算式，*S* 与 *Ci* 的匹配度计算公式可以表示为：

$$Pn(c_i,S)=\frac{\sum_k min(h_k(c_i),max_{j\in m}h_k(s_{ij}))}{\sum_k h_k(c_i)} \quad (4\text{-}1)$$

一般来说, n 取值越大，参考译文就越难匹配上，匹配度就会越低。1−gram 能够反映候选译文中有多少单词被单独翻译出来，也就代表了参考译文的充分性；2-gram、3-gram、4−gram 值越高说明参考译文的可读性越好，也就代表了参考译文的流畅性。当参考译文比候选译文长（单词更多）时，这种匹配机制可能并不准确，例如上面的参考译文如果是 The cat，匹配度就会变成 1，这显然是不准确的；为此可以引入一个惩罚因子。

$$BP(c_i,s_{ij})=\begin{cases}1, & l_{c_i}>l_{s_{ij}}\\ e^{l-\frac{l_{s_{ij}}}{l_{c_i}}} & \end{cases} \quad (4\text{-}2)$$

其中 l 表示机器译文和人工译文各自的长度，最终我们可以得到 BLEU 的计算公式：

$$BLEU=BP\times exp(\sum_{n-1}^{N}w_n\,logp_n) \quad (4\text{-}3)$$

其中w_n代表每一个 n-gram 的权重，一般我们可以取 n 为 4,在我们的课题组 n 也为 4。

ROUGE（Recall-Oriented Understudy for Gisting Evaluation）是一种常用的评价指标，用于评估自动摘要或机器翻译等文本生成任务。它旨在衡量生成文本与参考文本之间的相似度，考虑了重叠词汇、语义和词序等方面的匹配情况。ROUGE 主要包括了几个指标，例如 ROUGE-N、ROUGE-L、ROUGE-W 和 ROUGE-S。ROUGE-N 衡量了生成文本和参考文本中 n-gram 的重叠情况，而 ROUGE-L 则考虑了最长公共子序列的长度，同时考虑了词序的影响。ROUGE-W 则对匹配的 n-gram 赋予不同的权重，以更准确地反映语义匹配。最后，ROUGE-S 衡量了生成文本和参考文本中共同包含的连续字串的长度。ROUGE 的计算方式是将生成文本和参考文本转换为一组 n-gram 或词序列，然后通过计算匹配的 n-gram 数量、最长公共子序列长度等来衡量两者之间的相似度。ROUGE 指标的值越高，表示生成文本与参考文本之间的相似度越高，即生成文本的质量越好。在本课题中，我们将主要使用 ROUGE-N 和 ROUGE-L 指标。

ROUGE-N：衡量 n-gram（通常是单词）的重叠率。其中，ROUGE-1 表示单个词的重叠，ROUGE-2 表示相邻两个词的重叠，以此类推。ROUGE-N 主要用于评估生成摘要中与参考摘要中重要短语的重合度，判断摘要的概括能力。ROUGE-L：基于最长公共子序列（Longest Common Subsequence）计算两个摘要之间的相似度。它不仅考虑了重叠的单词和短语，还考虑了它们的顺序关系。ROUGE-L 适用于评估生成摘要中保持参

考摘要结构和重要信息的能力。

在评估 AI 调解员过程中，我们采用 BLEU-4、ROUGE-1、ROUGE-2、ROUGE-L 作为我们的 SFT 的评估指标。

4.1.2 SFT 训练结果

在 SFT 训练过程中，LORA 和全参微调两种训练方式的 loss 结果图分别如图 4-1 和 4-2：

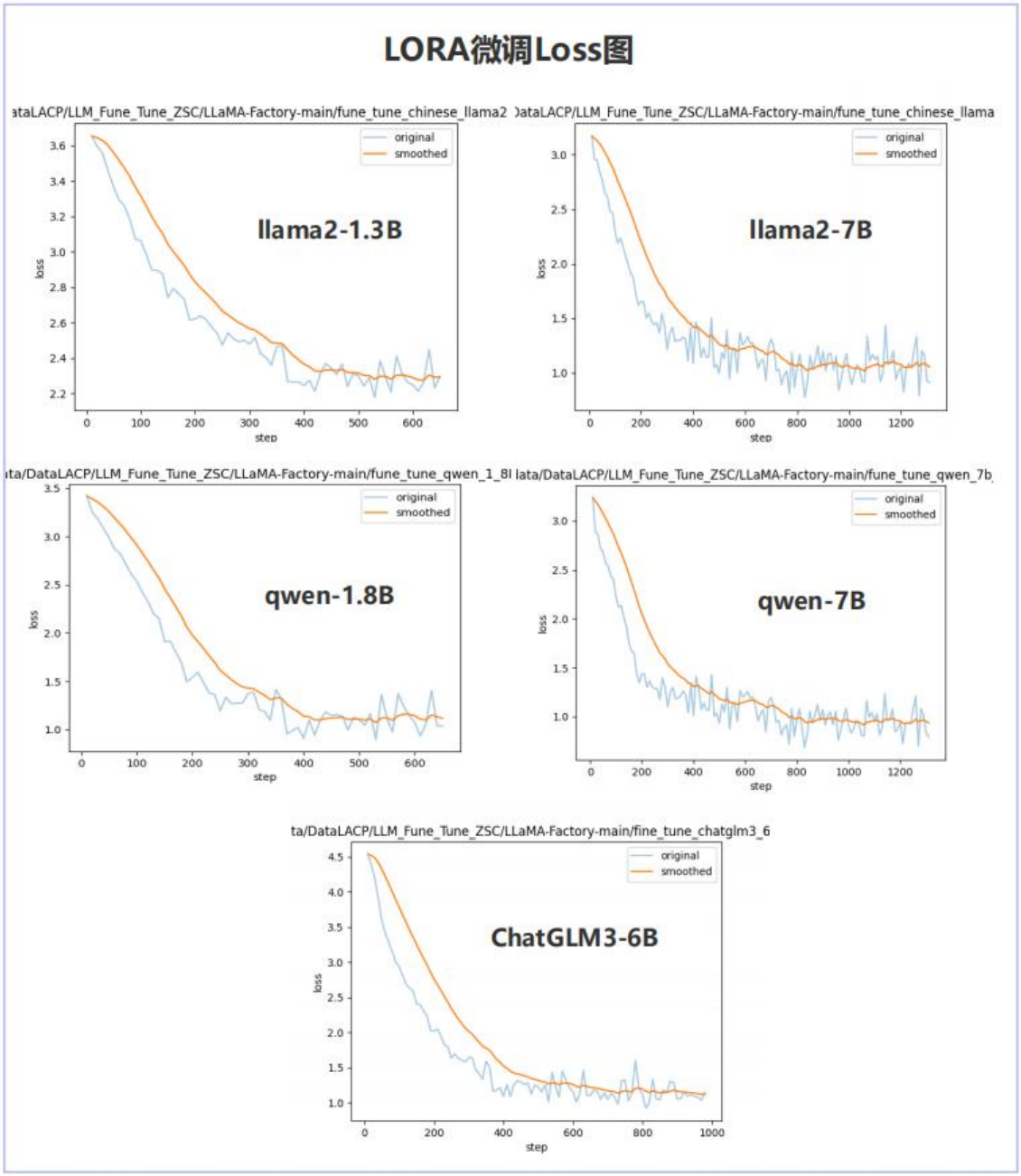

图 4-1：LORA 微调的 loss 分布图

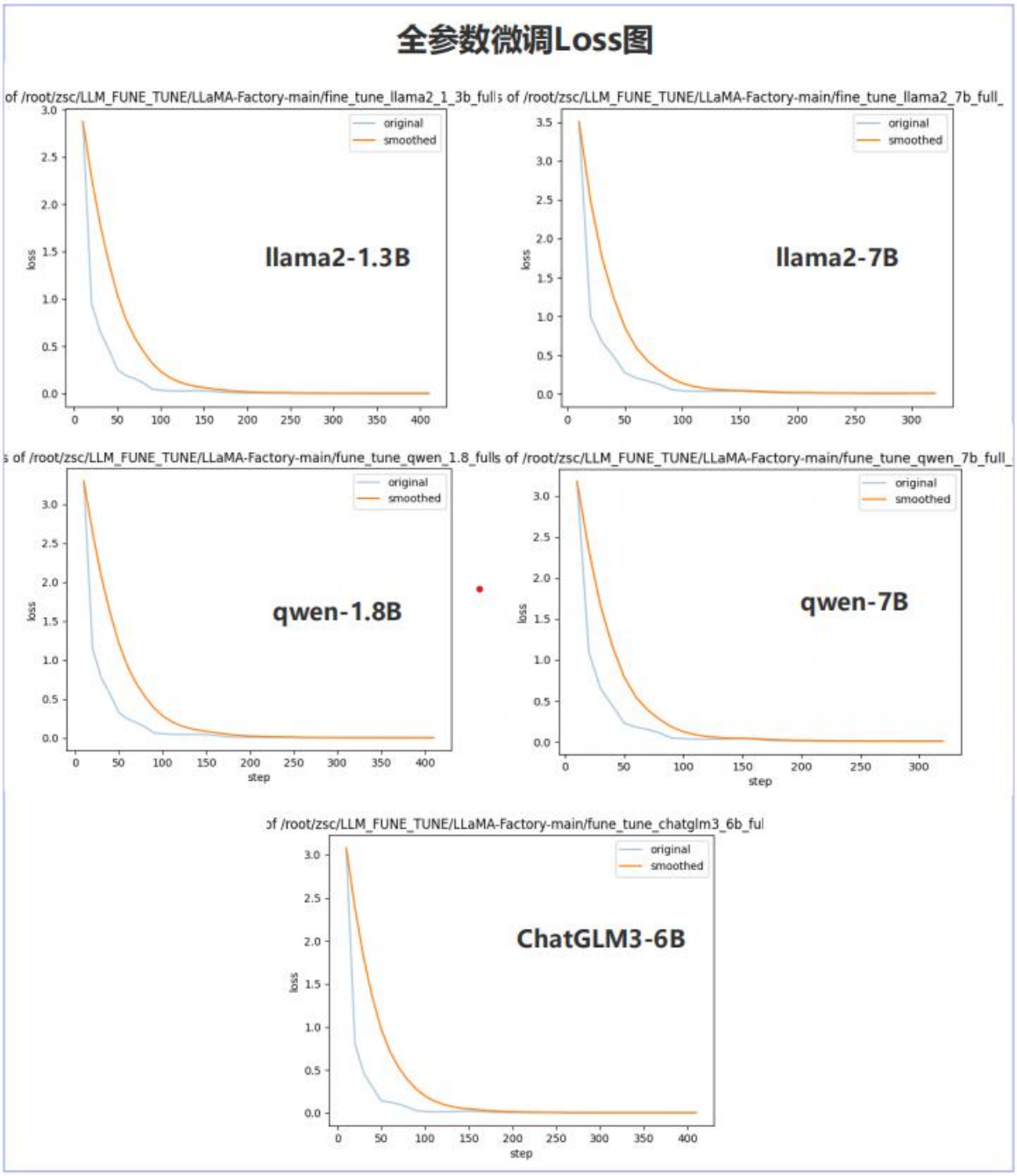

图 4-2：全参数微调 Loss 统计

通过分析 LORA 和全参数微调两种微调方式的 loss 效果图，我们可以发现 LORA 微调时，后期 loss 波动较大，而全参数微调的 loss 下降比较平稳，这是因为在 LORA 微调时会冻结模型的注意力层参数，而只是在旁边添加通路，使用两个低秩矩阵 A 和 B 进行替代，因此会导致在多轮对话微调的后期模型的学习能力受到限制，从而导致损失波动较大。而在全参数微调时，模型通常具备较大参数的规模，并且模型所有的参数都不会被冻结，即可以根据训练数据的每一轮反向传播时的结果更新而更新。这就可以使得模型在训练过程中能够更好地适应数据的特征，从而在学习时效果更好，loss 下降更

加平稳。

ChatGLM3、QWen、Chinese-LLaMA2 的不同参数量微调的效果汇总如表 4-1：

表 4-1：SFT 微调结果（%）

模型	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L
Qwen_1.8B_Chat+lora	61.98	70.78	62.83	67.84
Qwen_1.8B_Chat+full	**82.40**	**86.81**	**83.00**	**85.28**
Chatglm3_6B_Chat+lora	42.54	54.85	43.46	51.12
Chatglm3_6B_Chat+full	81.38	85.93	81.76	84.26
Qwen_7B_Chat+lora	69.92	77.47	70.63	75.24
Qwen_7B_Chat+full	81.20	85.86	81.74	84.30
Llama2_1.3B_Chat+lora	11.76	28.29	12.45	23.54
Llama2_1.3B_Chat+full	79.95	84.86	80.56	82.95
Llama2_7B_Chat+lora	70.47	77.89	71.40	75.58
Llama2_7B_Chat+full	79.12	84.34	79.67	82.51

从表中我们可以发现几个现象：

1. 对于同一个预训练预训练模型同一个参数量等级，全参微调的效果要好于 lora 微调，这一特点在 llama2_1.3B_chat 模型中尤其显著。这主要是由于 lora 微调仅仅采用低秩矩阵代替原模型中的参数更新，但是这样在降低 SFT 微调显卡资源消耗的同时，也会对效果产生一定的影响。llama2_1.3B_Chat 模型 lora 微调时的 BLEU-4 指标仅仅达到 11.76%，其 ROUGE-L 指标也才达到 23.54%，这是因为 llama2 模型最初仅仅在英文预料上进行预训练，后期才进行拓展到中文语料上，同时其参数量仅为 1.3B，这就导致其在 LORA 微调时调整的参数量就会更少，因此在我们的测试推理时效果就会比较差。

2. 对于同一个预训练模型，小参数量的模型全参微调会比大参数量微调效果要好，这是因为 AI 调解员训练数据集比较少，仅仅为 5263 条，对于参数量较大的模型很难仅仅用这部分数据特征进行全方位调整，从而导致模型欠拟合。因此全参数微调时，在数据集较少的情况下，微调较小的模型往往效果更好。

3. 对于不同的预训练模型，即 Qwen-Chat、ChatGLM3-Chat、LLaMA2-Chat，Qwen-1.8B-Chat 模型在全参微调时在各个指标上达到了最好的效果。因此我们的 AI 调解员最终展示的时候便采用 Qwen-1.8B-Chat-Full 微调版本。

4.2 当事人还款意愿预测模型实验结果

4.2.1 评估指标

评估指标这里我们使用精准率，召回率和 F1 值来进行评判，其中精准率，召回率，F1 值的计算如下：

$$precision = \frac{TP}{TP + FP} \tag{4-4}$$

$$recall = \frac{TP}{TP + FN} \tag{4-5}$$

$$F1 - Score = 2 \times \frac{precision \times recall}{precision + recall} \tag{4-6}$$

TP 代表当事人最终成功还款，并且模型预测当事人最终成功还款的数量，FP 代表当事人最终没有还款，但是模型预测当事人最终成功还款的数量，FN 代表当事人最终成功还款，但是模型预测当事人最终没有还款的数量。

4.2.2 模型训练结果

当事人还款意愿预测模块总共对比了四个模型，分别是 BERT+Linear、BERT+RNN、BERT+LSTM 三个 base 模型，同时也展示了我们的 Mean_BERT_LSTM_MLP 模型的效果，其训练测试结果如表 4-2，由结果可见，Mean_BERT_LSTM_MLP 模型总体优于另外三个模型：

表 4-2：当事人还款意愿预测模型

模型	Precision	Recall	F1-Score
BERT+Linear	95.24%	82.61%	88.47%
BERT+RNN	**95.72%**	83.48%	89.20%
BERT+LSTM	94.56%	86.96%	90.60%
Mean_BERT_LSTM_MLP（ours）	93.96%	**91.30%**	**92.61%**

4.3 本章小结

本章节主要对 AI 调解员模型和当事人还款意愿预测模型这两个模型的结果进行了分析和展示，并对各自的评估指标进行了介绍。结果显示，AI 调解员预测模型中的 Qwen-1.8B-Chat 模型在全参数微调时达到了最好的效果，而当事人还款意愿预测模型中

的 Mean_BERT_LSTM_MLP 模型达到了最好的 F1-score 值，为 92.61%。

第 5 章　模型效果可视化

本系统展示界面借鉴 Langchain 进行实现，后端在 4090 显卡上部署我们的 AI 调解员，并将 RAG 检索增强切块 Embedding 数据存储在向量数据库 faiss 中，前端采用 streamlit_chatbox 进行快速界面展示。详细部署开发细节见下图：

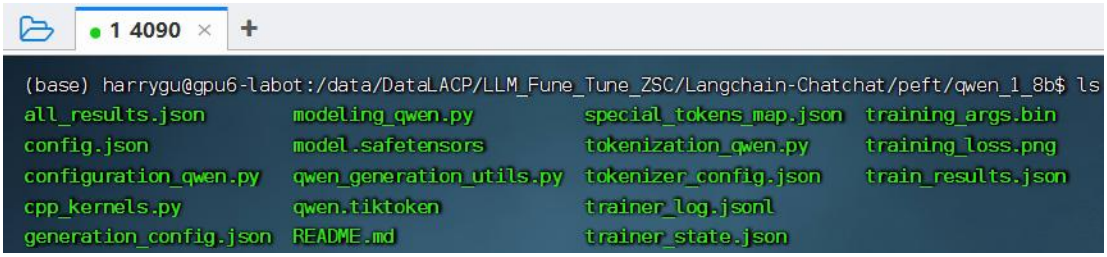

图 5-1：模型部署

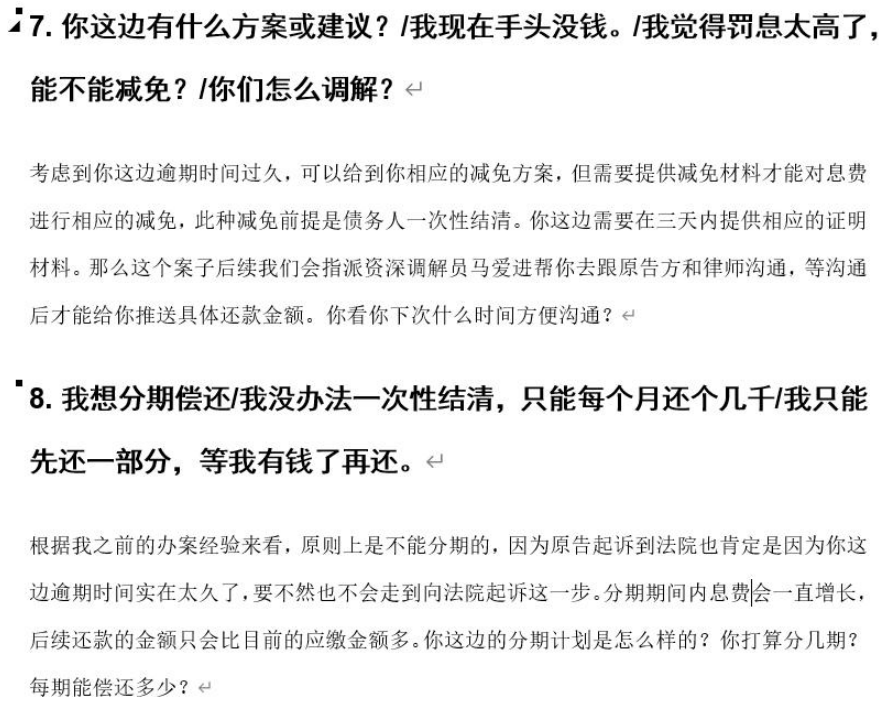

图 5-2：RAG 检索增强文档知识库源文件

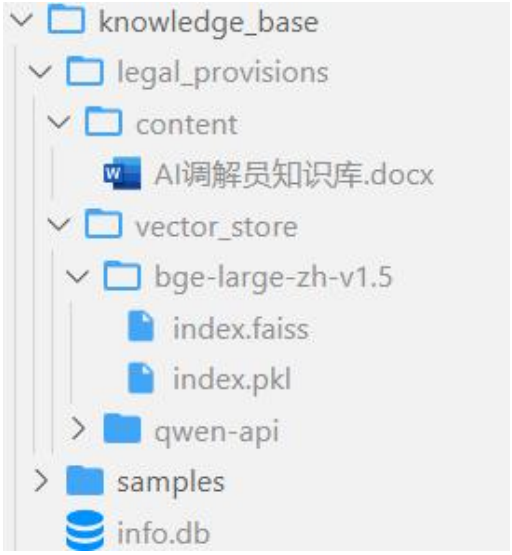

图 5-3：知识库设计

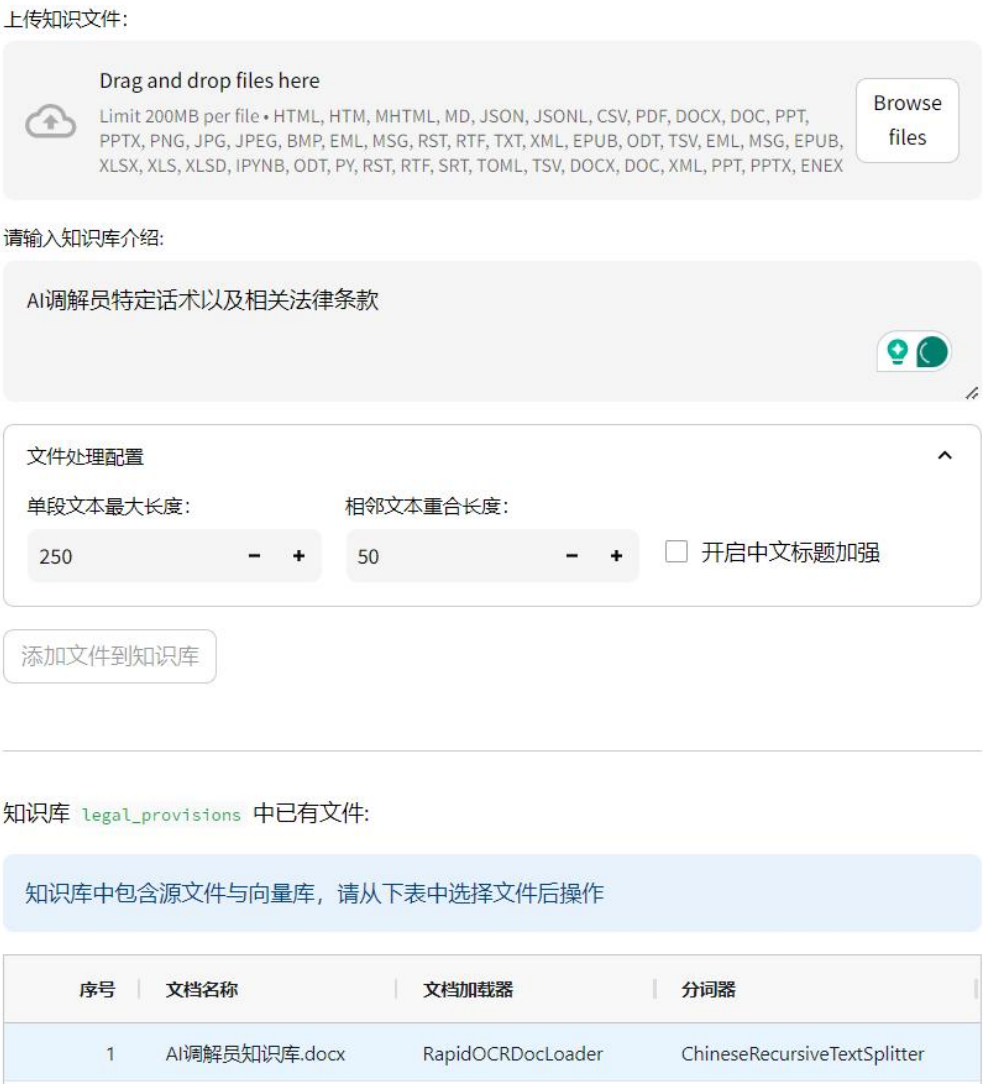

图 5-4：知识库加载界面

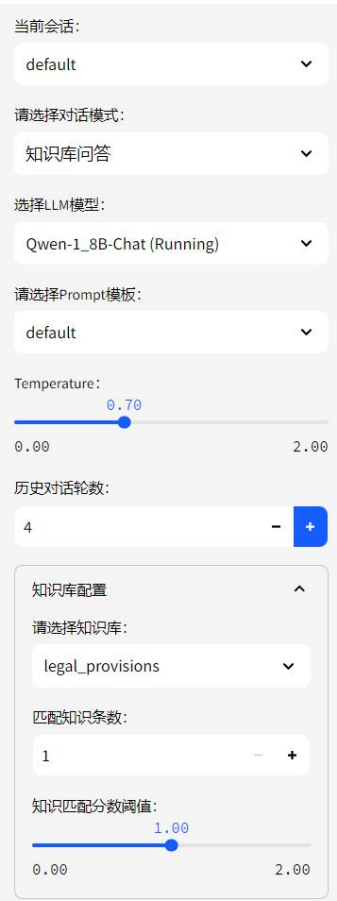

图 5-5：LLM 选择和知识库选择界面

图 5-6：AI 调解员与当事人模拟沟通界面

图 5-7：RAG 检索增强结果

当事人还款意愿模型需要加载 BERT 预训练模型和训练好的分类器模型，如图 5-8：

get_predict_result.py

model_predict > get_predict_result.py > get_predict_result

```python
from predict_mediate_result import SimpleClassifier, get_result
from transformers import BertModel, BertTokenizer
import torch
import torch.nn as nn

def get_predict_result():
    # 加载本地的BERT模型和tokenizer
    model_name = 'D:/project/model/bert-base-chinese'  # 本地模型的名称或路径
    tokenizer = BertTokenizer.from_pretrained(model_name)
    model = BertModel.from_pretrained(model_name)


    # Define and initialize the model
    input_size = 768
    hidden_size = 64
    output_size = 1
    # 初始化一个相同架构的模型
    loaded_model = SimpleClassifier(input_size, hidden_size, output_size)

    # 加载模型参数和优化器状态
    checkpoint = torch.load('D:\project\model_predict\saved_model_threshold_0.5_300_epoch_lr_0.004.pth')
    loaded_model.load_state_dict(checkpoint['model_state_dict'])

    # 将模型设置为评估模式
    loaded_model.eval()


    user_word = ['喂。','我明天就还钱']
    if len(user_word) == 0:
        print("用户未说话")
    else:
        result = get_result(user_word, tokenizer, model, loaded_model)
        return result
print(f"预测结果:{get_predict_result()}")
```

图 5-8：当事人还款意愿模型加载界面

在调用时，我们将 user_word 替换为上面沟通过程中当事人的通话内容：

['是的',
'我太穷了，
没钱还款',
'那我可以申请分期付款吗？',
'那我可以申请减免利息吗？你们这利息太高了']

预测结果如图 5-9 所示：

```
说明：结果为1代表模型预测当事人最终可以成功还款，结果为0代表模型预测当事人最终无法成功还款。
用户说话记录：['是的', '我太穷了，没钱还款', '那我可以申请分期付款吗？', '那我可以申请减免利息吗？你们这利息太高了', '']
预测结果:0
```

图 5-9：模型预测结果展示

第 6 章　总结与展望

6.1　总结

本文致力于探索开发一款 AI 调解员辅助沟通系统，主要进行了三方面的探索。首先是 AI 调解员多轮对话系统，我们利用多轮对话下的自回归生成任务的基本思路，使用西湖法院真实的调节对话数据集对目前主流的开源模型 ChatGLM3-6B-Chat、Qwen-7B-Chat、Qwen-1.8B-Chat、Chinese-LLaMA2-1.3B-Chat、Chinese-LLaMA2-7B-Chat 进行 SFT 训练，这个过程中我们对比了 LORA 微调和全参数微调两种微调方法对效果的影响，并使得 LLM 在对话过程中扮演调解员的角色。除此之外，我们结合 Langchain 技术实现了相关调解话术和法律条款的 RAG 增强，减少了模型的幻觉。最后我们对比了不同参数量和微调方式对 AI 调解员对话模型的影响，并选择效果最佳的 SFT 模型作为我们的 AI 调解员角色。

除此之外，我们设计了基于 BERT Embedding 的当事人还款意愿预测模型，利用自编码模型 BERT 获取第一次通话过程中当事人说话的全部信息，并最终将每句话的词向量相加然后对应位置的向量取平均，获取其 mean embedding 信息。之后我们利用了三个分类器对获取的词向量以及对应的 Label(1 代表可以调解成功，0 代表不可以调解成功)进行训练和测试，及时识别出有还款意愿的当事人，以此提高调解成功的概率。

最后我们在真实场景中对以上两种模型进行了测试，并通过对话界面模拟 AI 调解员与当事人沟通的整个过程。

6.2　展望

本文章探索了两种 SFT 微调方式的效果，后期可以尝试其他不同的微调方式，例如 P-Tuning v2 等。除此之外，本文的数据集较少，仅仅 5263 条，后期可以利用 GPT-4 进行知识蒸馏，以此对数据集进行扩充。另外，可以尝试强化学习技术对 AI 调解员模型进行 DPO、PPO 等方式的迭代训练，以使得其更加贴向于真人调解员。

致谢

行文至此，诸多话语，难以言表。

往事如烟、历历在目。仍记得第一次和父亲一起乘坐火车从商丘来到长沙这座城市，这是我第一次来到南方也是第一次因为求学而远离家乡，怎么描述我当时踏上这趟商丘南-长沙南次列车的心情呢？大抵就是既充满了对未来大学生活的向往，又担心自己不适应大学的生活，担心自己无法适应当地的气候，饮食，等诸多事物。到达长沙之后，走到出站口，我望着中南大学那个蓝色的牌匾出神了许久，学长学姐们的热情欢迎也让我第一次在长沙这座城市感受到了温暖，同时也让我意识到长沙是一座属于我的城市，是一座我不会再感到陌生的城市，这也是我对长沙的初版印象。

鲜衣怒马少年时,不负韶华行且知。怎么诉说我在中南大学的这四年呢？大概就是从最开始的“懵懂”、“胆怯”、“惶恐”的少年变成了一个“成熟”、“稳重”、“自洽”的成年人。在中南大学结交了一些很好的人，遇到了很多非常好的老师(我觉得软件工程系的老师讲课都别具风格，对待学生也非常和蔼，真的是亦师亦友的典范!)，也见到了很多温馨的瞬间，当然也会有一些心情低落的时刻，不过这些都将是我在 18-22 岁最为珍贵的回忆。中南大学也见证了我的诸多成长瞬间，包括但不限于身高、体重、吃辣程度、晚睡程度以及面对天气突然下雨时的应变程度。

十年树木，十载风，十载雨，十万栋梁。我可以走到现在，离不开老师们的谆谆教导。尤其是感谢廖老师以及廖老师实验室里的师兄师姐对于我科研上的指导以及生活上的关心，真的万分感激，是你们在我迷茫的时候对我提出了许多有用的建议，让我在迷雾重重的前方看到了光亮。我还要感谢葛老师的毕业设计的相关指导以及建议，让我可以顺利完成本科毕业论文。

春晖寸草，难以回报。感谢在我后面始终默默支持着我的家里人，尤其是感谢我的父亲，是你们对我这二十多年来无微不至的照顾与关爱，让我可以站在你们的肩膀上，见识到更加广阔的世界。因为有了你们，才有了现在的我。自从离开家，我才得知时光飞逝犹如白驹过隙，小学时的家是早上的太阳与夜晚的星光，中学时的家是两三周一次的短暂温存，上了大学，家乡便只有夏冬，再无春秋。此时此刻，教育仿佛在我这里形成了闭环，我也才真正感受到了树欲静而风不止，子欲养而亲不待的真正含义。接下来，

我将带着你们的期望与嘱托继续奔赴下一场山海，你们仍然是我最为坚固的保护伞，而我，也在学着慢慢成为你们的保护伞。

山水一程，三生有幸。感谢我身边所有的同学、朋友。感谢你们的陪伴与倾听，让我的生活更加丰富多彩，谢谢你们让我人生的大多数时刻都会觉得这个世界真的很美好。因为有了你们，让我感受到了很多温暖与快乐。心怀感恩，所遇皆温柔，祝大家前程似锦，行稳致远！

长风破浪会有时，直挂云帆济沧海。最后我要感谢我自己，感谢你可以坚持走了那么远的路，只有你自己才知道这一路上你也会经常迷惘。虽然有时候的你会止步不前，也因此绕了许多远路，但是还不错，你的大方向是对的，你可以永远相信自己。

欲买桂花同载酒，终不似，少年游。人最大的遗憾大抵就是无法同时拥有青春，和对青春的感悟。大学四年时光转瞬即逝，毕业在即，很多人或许以后再也无法相见，但这也许就是青春。社会在往前走，大家在往前走，我也要往前走。

此情可待成追忆，只是当时已惘然。始于 2020 年初秋，终于 2024 年盛夏，我的本科生涯结束了，但我未来的人生也才刚刚开始.....

参考文献

[1] Weizenbaum J. ELIZA—a computer program for the study of natural language communication between man and machine[J]. Communications of the ACM, 1966, 9(1): 36-45.

[2] Colby K M. Ten criticisms of parry[J]. ACM SIGART Bulletin, 1974 (48): 5-9.

[3] R. Wilensky. The Berkeley UNIX consultant project[J]. Computational Linguistics, 1988, 14(3): 35-84.

[4] 吴侯,李舟军.检索式聊天机器人技术综述[J].计算机科学,2021,48(12):278-285.

[5] Aizawa A. An information-theoretic perspective of tf-idf measures[J]. Information Processing & Management, 2003, 39(1): 45-65.

[6] Robertson S, Zaragoza H, Taylor M. Simple BM25 extension to multiple weighted fields[C]//Proceedings of the thirteenth ACM international conference on Information and knowledge management. 2004: 42-49.

[7] 陈晨, 朱晴晴, 严睿, 等. 基于深度学习的开放领域对话系统研究综述[J]. 计算机学报, 2019, 42(7): 1439-1466.

[8] Lu Z, Li H. A deep architecture for matching short texts[J]. Advances in neural information processing systems, 2013, 26.

[9] M. Wang, Z. Lu, H. Li, et al. Syntax-based deep matching of short texts[C], Proceedings of the 24th International Conference on Artificial Intelligence. Buenos Aires, Argentina: AAAI Press, 2015: 1354-1361.

[10] Wu Y, Li Z, Wu W, et al. Response selection with topic clues for retrieval-based chatbots[J]. Neurocomputing, 2018, 316: 251-261.

[11] X. Zhou, D. Dong, H. Wu, et al. Multi-view response selection for human-computer conversation[C], Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin, Texas: Association for Computational Linguistics, 2016: 372-381.

[12] Yan R, Song Y, Wu H. Learning to respond with deep neural networks for retrieval-based human-computer conversation system[C]//Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. 2016: 55-64.

[13] 吴威震. 基于seq2seq模型的聊天机器人对话研究[D]. 南京邮电大学,2019.DOI:10.27251/d.cnki.gnjdc.2019.000499.

[14] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[J]. Advances in neural information processing systems, 2014, 27.

[15] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[16] Wu T, He S, Liu J, et al. A brief overview of ChatGPT: The history, status quo and potential future development[J]. IEEE/CAA Journal of Automatica Sinica, 2023, 10(5): 1122-1136.

[17] Touvron H, Martin L, Stone K, et al. Llama 2: Open foundation and fine-tuned chat models[J]. arXiv preprint arXiv:2307.09288, 2023.

[18] Du Z, Qian Y, Liu X, et al. Glm: General language model pretraining with autoregressive blank infilling[J]. arXiv preprint arXiv:2103.10360, 2021.

[19] Bai J, Bai S, Chu Y, et al. Qwen technical report[J]. arXiv preprint arXiv:2309.16609, 2023.

[20] Yang A, Xiao B, Wang B, et al. Baichuan 2: Open large-scale language models[J]. arXiv preprint arXiv:2309.10305, 2023.

[21] Voita E, Talbot D, Moiseev F, et al. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned[J]. arXiv preprint arXiv:1905.09418, 2019.

[22] Zhao M, Ma Y, Ding Y, et al. Multi-query multi-head attention pooling and inter-topk penalty for speaker verification[C]//ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022: 6737-6741.

[23] Ainslie J, Lee-Thorp J, de Jong M, et al. Gqa: Training generalized multi-query transformer models from multi-head checkpoints[J]. arXiv preprint arXiv:2305.13245, 2023.

[24] Song X, Salcianu A, Song Y, et al. Fast wordpiece tokenization[J]. arXiv preprint arXiv:2012.15524, 2020.

[25] Wang C, Cho K, Gu J. Neural machine translation with byte-level subwords[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(05): 9154-9160.

[26] Huang Y, Bai Y, Zhu Z, et al. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models[J]. Advances in Neural Information Processing Systems, 2024, 36.

[27] Zhou K, Zhu Y, Chen Z, et al. Don't Make Your LLM an Evaluation Benchmark Cheater[J]. arXiv preprint arXiv:2311.01964, 2023.

[28] Cui Y, Yang Z, Yao X. Efficient and effective text encoding for chinese llama and alpaca[J]. arXiv preprint arXiv:2304.08177, 2023.

[29] Gao L, Schulman J, Hilton J. Scaling laws for reward model overoptimization[C]//International Conference on Machine Learning. PMLR, 2023: 10835-10866.

[30] Rafailov R, Sharma A, Mitchell E, et al. Direct preference optimization: Your language model is secretly a reward model[J]. Advances in Neural Information Processing Systems, 2024, 36.

[31] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347, 2017.

[32] Zhang B, Sennrich R. Root mean square layer normalization[J]. Advances in Neural Information Processing Systems, 2019, 32.

[33] Augustin A, Yi J, Clausen T, et al. A study of LoRa: Long range & low power networks for the internet of things[J]. Sensors, 2016, 16(9): 1466.

[34] Liu X, Ji K, Fu Y, et al. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks[J]. arXiv preprint arXiv:2110.07602, 2021.

[35] Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks[J]. Advances in Neural Information Processing Systems, 2020, 33: 9459-9474.

[36] Church K W. Word2Vec[J]. Natural Language Engineering, 2017, 23(1): 155-162.

[37] Rodríguez P, Bautista M A, Gonzalez J, et al. Beyond one-hot encoding: Lower dimensional target embedding[J]. Image and Vision Computing, 2018, 75: 21-31.

[38] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[39] Rasley J, Rajbhandari S, Ruwase O, et al. Deepspeed: System optimizations enable

training deep learning models with over 100 billion parameters[C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 3505-3506.

[40] Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network[J]. Physica D: Nonlinear Phenomena, 2020, 404: 132306.

[41] Papineni K, Roukos S, Ward T, et al. Bleu: a method for automatic evaluation of machine translation[C]//Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002: 311-318.

[42] Lin C Y. Rouge: A package for automatic evaluation of summaries[C]//Text summarization branches out. 2004: 74-81.