南京大学

本科毕业设计

院　　系　软件学院

专　　业　软件工程

题　　目　基于大模型的刑事一审判决书法条和判决段自动生成的研究

年　　级　2020　学　　号　201250058

学生姓名　李维璇

指导教师　葛季栋　职　　称　副教授

提交日期　2024 年 5 月 23 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：基于大模型的刑事一审判决书法条和判决段自动生成的研究）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：基于大模型的刑事一审判决书法条和判决段自动生成的研究

院系：软件学院

专业：软件工程

本科生姓名：李维璇

指导教师（姓名、职称）：葛季栋　副教授

摘要：

中国法院裁判文书是一种格式相对固定、上下文逻辑性强的文本，而这正是文本生成式大语言模型所擅长的领域。大语言模型展出了非凡的语言能力，应用场景广泛，前景极其广阔，将大模型应用于法院裁判文书领域大有可为。

本文关注大模型生成刑事判决书的方向。本项目将刑事判决书分为证据段、裁判分析段、法条段、判决结果段，将包含检察机关指控的当事人罪行、法院查明的真相和意见的裁判分析段作为输入，法条段和判决结果段作为输出，构建出适用于大模型微调训练的数据集。接着，使用 LoRA 微调方法对 Llama2-7B-hf、ChatGLM2-6B、Qwen1.5-1.8B 大语言模型基座进行微调，使得微调后的大语言模型在裁判文书方面具有推理并生成文本的能力。然后，使用 vLLM 对大模型推理进行加速。此外，本项目开发了一个供司法人员使用的网页交互系统，以便于司法实践。

本项目对微调后的大语言模型进行性能评测，在 BERT、ROUGE_L 指数上验证出其在裁判文书生成方面的能力显著强于原大语言模型基座。因此，将大模型与法院裁判文书相结合是具有较高应用价值的。

本项目能够为法官自动生成符合法律规定和个案需求的诉讼文书，能够大大减轻司法人员的心智负担，有效提升判案效率，创造更高水平的数字正义，让人民群众在每个司法案件中感受到公平正义。

关键词：大模型；自然语言处理；法律文书；法条预测

南京大学本科生毕业论文（设计、作品）英文摘要

THESIS: Research on Automatic Generation of Calligraphy Notes and Sentence Segments in Criminal First Instance Judgments Based on LLM

DEPARTMENT: Software Institute

SPECIALIZATION: Software Engineering

UNDERGRADUATE: Li Weixuan

MENTOR: Associate professor Ge Jidong

ABSTRACT:

China court-verdict documents are a kind of text with a relatively fixed format and strong logical context, which is exactly the field that text-generation large language models are expert in. The large language model displays extraordinary language abilities, with a wide range of application scenarios and extremely broad prospects. Therefore, There is great potential to apply large models to the field of court legal documents.

This thesis focuses on generating criminal legal documents based on large language models. This project divides the criminal legal documents into evidence section, judgement analysis section, legal provisions section, and judgement result section. I take the judgement analysis section containing the crimes of the defendants and the court's opinion on the fact as input, and take the legal provisions section together with the judgement result section as output. Therefore, a dataset suitable for fine-tuning large language models has been established. Subsequently, use the LoRA method to fine-tune several large language model bases, including Llama2-7B-hf, ChatGLM2-6B, and Qwen1.5-1.8B, so that the fine-tuned large language models have the ability to reason and generate legal documents. Then, use vLLM to accelerate the inference of large language models. In addition, this project has developed a web interaction system for judicial personnel to use in judicial practice.

This project develops a benchmark on the fine-tuned large language models and verify that these ability in generating legal documents on the BERT and ROUGE-L indices is significantly stronger than that of the original large language model bases. As a consequence, combining large models with court rulings has high application value.

This project can automatically generate legal documents that meet the legal requirements and individual case needs for judges, greatly reducing the mental burden on judicial personnel, effectively improving efficiency, and striving to create a higher level of digital justice, so that the people can feel fairness and justice in every judicial case.

KEYWORDS: Large language model; Natural language processing; Legal documents; Provision prediction

目　　录

插图目录

表格目录

第一章　引言

1.1　研究背景及意义

为了使机器拥有人类一样的阅读、写作等语言能力，语言建模 (*Language Modeling*) 是一种主流解决方案。其中，以预训练语言模型 (*Pretrained Language Models*) 为前沿技术，将预训练与微调作为核心步骤。3.3 亿参数的 *GPT-2* 是主要代表之一人大研究。而通过大规模扩展语言模型大小和数据集大小，如数百倍于 *GPT-2* 参数的 *GPT-3*，语言模型的下游性能得到了极大提高扩展 PLM。这种规模扩展后的预训练语言模型在行业内称为大语言模型 (*Large Language Models*)。大语言模型展出了非凡的语言能力，应用场景广泛，前景极其广阔。

大语言模型已在诸多应用领域落地，如 *ChatGPT*、*Copilot* 等。其中，以文本生成 (*Text-Generation*) 为主要应用领域。通过向大模型给出一段文本，大模型会接续给出可能性最大的下文人大研究。这种本质上是基于最大概率所生成的文本，外在表现出的是大模型极强的基于上下文的逻辑推理能力。大模型能够学习数据集中存在的分布规律，而形成更为原创的新数据生成式人工智能在司法中的运用。

在司法领域，所涉及的文本基本上具有逻辑性强、上下文关联度高的特性，这与大模型所擅长的领域不谋而合。因此，将大语言模型应用于司法实践中，是大势所趋。最高人民法院在 2022 年 12 月颁布《最高人民法院关于规范和加强人工智能司法应用的意见》[1]，其中提到：2025 年时，较为完备的司法人工智能技术应用体系基本建成；2030 年，建成具有法律规则引领作用和应用示范效应的司法人工智能技术理论体系，并进行实际应用。基于大语言模型的生成式人工智能更适用于这类创新性场景，让智慧司法落到实处。

因此，本项目将大模型与司法有机结合，并重点研究大模型在法院刑事判决文书方面的应用场景。标准的刑事判决文书分为证据段、裁判分析段、法条推荐段、判决结果段。根据法官提供的证据段，大语言模型可以生成裁判分析段，再

1　原文：https://www.court.gov.cn/fabu/xiangqing/382461.html

由裁判分析段生成后续的内容，即可得到一份完整的裁判文书供法官参考。自动生成符合法律规定和个案需求的诉讼文书，能够大大减轻司法人员的心智负担，有效提升判案效率，努力创造更高水平的数字正义，让人民群众在每个司法案件中感受到公平正义。

1.2 研究现状

业界对智慧司法的研究，大多基于判别式人工智能。然而，大多数现有的判别式语言模型难以理解不同结构之间的长距离依赖关系。此外，与一般检索相比，法律领域的关联对关键法律要素非常敏感。即使是要素上的细微差异也会显著影响相关性的判断[法律案件 PLM]。而基于大语言模型的生成式人工智能以其超越性的创新性，带来无比广阔的使用前景。

因此，经过对中国法律专精训练的大语言模型也相应产生。如 *LexiLaw*[2]，是清华大学项目组在 *ChatGLM* 大模型基座上，通过综合使用通用领域数据、专业法律数据和法律文书进行微调而得到的，在提供法律咨询和支持方面具备更高的性能和专业性[法律案件 PLM]。还有夫子•明察司法大模型，由山东大学、浪潮云和中国政法大学联合研发，这一中文司法大模型以 ChatGLM 为基础，利用海量中文司法语料进行训练。该模型具备法条检索、案例分析、三段论推理判决及司法对话等功能，旨在为用户提供全面且高精度的法律咨询和解答服务[夫子明察论文]。然而，现有的司法大语言模型大多为问答式，面向普通大众及法律初学者，旨在提供基础的法律咨询服务。

在司法实践中，已有大语言模型投入使用。苏州市中级人民法院开发了“数字法官助理”系统，这是一种人工智能辅助阅卷及法律文书生成工具，旨在推动人工智能技术与司法工作的深度融合。该系统利用“人工智能大模型”资源，创建了一个具备法律语义理解和自然语言生成能力的专用大语言模型，能够显著提高案件处理的质量和效率。目前已将该系统应用于金融借款合同纠纷、劳动争议、买卖合同纠纷、房屋租赁合同纠纷等案由的案件中[3]。

可以看到，大语言模型在智慧司法领域展现出了巨大的潜力。然而，目前司法大语言模型在裁判文书智能化生成方面的应用还相对有限。此外，现阶段的模

2 https://github.com/CSHaitao/LexiLaw

3 原文：http://www.zjrmfy.suzhou.gov.cn//fypage/toContentPage/sousuo/82a07a488ac7eb8c018acf201cae0009

型主要涉及细分案由的案件，而通用性方面仍有待提升。

当前我国智慧法院正处于迈向“4.0 版”的建设期。其中，大语言模型不可或缺，这也会是区别于“3.0 版”的一大特点。本项目旨在对裁判文书的智能化生成进行具体研究，应用基于大语言模型的生成式人工智能，面向第一线的司法实践，对刑事案件进行分类研究，在“刑事判决书”这一细分领域专精，作为对现有司法大模型的补充。

1.3 本文主要工作

本文的主要内容是介绍所开发的基于大模型的法院文书智能生成系统(*JDIGS——A Judicial Document Intelligent Generation System Based on LLM*)，包含构建数据集、微调大模型、大模型推理、性能评测、交互系统。

本文侧重刑事判决书中裁判分析段到引用法条段、裁判分析段到判决结果段的生成，即刑事裁判分析段作为大模型的输入，适用的《中华人民共和国刑法》法条和判决结果段作为大模型的输出。使用 *Llama2-7B-hf*、*ChatGLM2-6B*、*Qwen1.5-1.8B* 作为微调大模型的基座，使用 *LoRA* 方法，探究合适的训练参数以对基座进行微调 (*fine tune*)。得到微调后的大模型后，使用 *vLLM* 进行推理加速，并以 *OpenAI* 方式进行部署。此外，开发一个性能评测工具，在 *BERT*、*ROUGE_L* 指标下测评微调后的大模型的性能。最后，开发一个交互系统以便于司法实践，系统使用前后端分离方法，开发供司法人员使用的网页交互系统，所用技术为业界成熟的 *SpringBoot* 后端框架及基于 *vue* 的前端框架。

1.4 论文的组织结构

本文分为 7 个章节，具体内容如下：

第一章：引言。主要介绍本文的背景及意义，阐明司法大模型的研究现状，列出本文的主要工作，并说明论文的组织结构。

第二章：构建数据集。首先介绍本部分所运用的关键技术 (*Kafka, Flink, Click-house*)，然后指出数据的筛选指标及格式要求，最后阐述构建大模型所需刑事判决书的数据集的构建过程。

第三章：微调大模型。首先介绍本部分所运用的关键技术 (*Llama2-7B-hf,*

ChatGLM2-6B, Qwen1.5-1.8B, LoRA, Llama_factory)，然后解释微调训练大模型时所设置的参数设置的含义，最后说明使用微调后的大模型进行推理的先决准备。

第四章：大模型推理。首先介绍本部分所运用的关键技术 (*vLLM*)，然后说明大模型的部署方式，解释使用大模型进行推理时的参数设置并介绍增强大模型推理结果的方法。

第五章：性能评测。首先介绍本部分所运用的关键技术 (*BERT, ROUGE_L*)，然后说明性能评测的过程，最后展现各个大模型的性能评测结果并作分析。

第六章：交互系统。说明交互系统的实现内容，并展现实际使用效果。

第七章：总结与展望。对本文的工作进行回顾和总结，对不足之处提出改进的想法，以及就司法大模型的未来应用前景进行展望。

第二章　构建刑事一审文书数据集

2.1　技术概论

2.1.1　Kafka

Kafka 是一个流处理平台，由 Scala 和 Java 编写，本质上，它基于分布式事务日志架构，是一个支持大规模发布/订阅的消息队列，能够为实时数据处理提供一个统一、低延迟和高吞吐的解决方案。其中，生产者 (*producer*) 将数据发送至消息队列，消费者 (*consumer*) 从消息队列中取出消息，进行后续的处理。其中，消费者可以组成集群 (*Group*)，来加快消息处理速度。

2.1.2　Flink

Flink 是用 Java 和 Scala 编写的分布式流数据流引擎。Flink 以数据并行和管道方式执行任意流数据程序，Flink 的流水线运行时系统可以执行批处理和流处理程序**flink 介绍**。Flink 不提供数据存储系统，但为 Kafka 提供了数据源和接收器，本项目即将 Kafka 与 Flink 相结合。

Flink 允许众多设备组成集群 (*Cluster*)，每个设备并发运行，可加快数据流处理速度。本项目使用了这一特性。

2.1.3　ClickHouse

ClickHouse[1]是一个用于在线分析处理 (*OLAP*) 的高性能、面向列的 *SQL* 数据库管理系统。它具有查询效率高、批处理能力强、支持短时间大量查询的能力，广泛应用于数据大批量处理领域。

将 ClickHouse 与 Kafka、Flink 结合，能够快速处理大量数据。

1　https://clickhouse.com/docs

2.2 刑事一审文书数据集的筛选

相比普通 *PLM*，*LLM* 更需要高质量数据来预训练与微调模型，并且其语言能力很大程度上依赖于语料库的质量人大研究。必须摒弃数据集越大，训练效果越好的观点。因此，构建合适的训练数据集是工作的重点。由于本项目后续的工作是在大模型基座上进行微调，所以通用文本并不需要在微调阶段作为训练的语料，而专用文本才是本阶段需要准备的，以期大模型能在特定的下游任务中展现出专业能力。

具体地，本阶段需要构建的数据集为符合**下列要求**的刑事判决书。

2.2.1 质量过滤

原始的数据集来自中国裁判文书网[2]，其中存在的质量问题、错误原因和解决方法如表**??**所示。

表 2-1 原始数据集的质量过滤

质量问题	错误原因	解决方法
Html 乱码	原始数据集是由 Html 格式的文件转译的 *json* 文件，此过程中遗留下 Html 标签，如 <&rdquo>	去除 Html 标签
字符”\xef\xbf\xbd”	编码格式错误，部分关键信息无法获取	弃置该条数据
其他与文本不相关的非中文字符	原始数据集即存在的其他字符	去除该字符

2.2.2 按照证据段、裁判分析段、法条推荐段、判决结果段的格式撰写并分割

一份典型的刑事判决书格式如表**??**所示。

表 2-2 典型的刑事判决书格式

刑　事　判　决　书

(××××)× 刑初字第 ×× 号

---

2 https://wenshu.court.gov.cn/

被告人……（包括姓名、性别、民族、出生日期、职业、文化程度、住址以及因本案所采取的强制措施情况等信息）。

辩护人……（包括姓名、工作单位和职务）。

现已审理终结。

（以下为**证据段**）

××× 人民检察院指控……（概述人民检察院的指控信息，包括对被告人犯罪的指控事实、由公安机关查明及检察院掌控的证据以及适用法律法规的意见）。

被告人 ××× 辩称……（概述被告人对自己被指控的犯罪事实的供述是否承认、自行辩护的意见及相关证据）。辩护人 ××× 提出的辩护意见是……（概述辩护人的辩护意见及理由）。

经审理查明，……（首先写明经庭审查明的事实；其次写明经举证、质证定案的证据及其来源；最后对控辩双方有异议的事实、证据进行分析和认证）。

（以下为**裁判分析段**）

本院认为，……(根据查证属实的事实、证据、被告人认罪情况和态度，以及相关法律法条，论证公诉机关的指控是否成立，被告人的行为是否构成犯罪，如何进行刑事处罚，是否接纳被告的辩护意见)。

（以下为**法条推荐段**）

依照……(写明判决的法律依据) 的规定，

（以下为**判决结果段**）

判决如下：……（定罪结果，刑期，罚款等）。

其中，证据段从“审理终结”开始，到“本院认为”之前。裁判分析段从“本院认为”开始，到依照的法条前。法条推荐段为“判决如下”之前。判决结果段从“判决如下”开始到文章末尾。可以发现，刑事判决书的格式非常固定，人为划分的各个部分的起始和结束位置清楚。

原始的数据集中，每一篇文书以 *json* 格式存在，表**??**是一个例子。

其中，本项目使用“文书内容”和“法律依据”两个字段。“文书内容”字段是整篇文书，其格式在上文已经说明，可从中获得证据段、裁判分析段和判决结果段；“法律依据”字段可以看做是数据集所需要的“法条推荐段”，不需要从“文书内容”中获得，可保证数据的精确性。若原始数据集中“法律依据”字段

表 2-3　原始数据集格式

字段名	示例
标题	林宇渊伪造、变造、买卖国家机关公文、证件、印章罪、伪造公司、企业、事业单位、人民团体印章罪一审刑事判决书
审理法院	江苏省昆山市人民法院
案件类型	刑事案件
网页链接	https://wenshu.court.gov.cn/website/wenshu/.....
案号	（2017）苏 0583 刑初 648 号
审理程序	一审
裁判日期	2017-11-01
发布日期	2017-12-20
文书内容	昆山市人民检察院以昆检诉刑诉〔2017〕585 号起诉书指控被告人林宇渊犯伪造、买卖国家机关证件、印章罪，伪造公司、企业、事业单位、人民团体印章罪，于 2017 年 4 月 1 日向本院提起公诉。（后续内容略，参考上文）
当事人	林宇渊
案由	伪造、变造、买卖国家机关公文、证件、印章
法律依据	《中华人民共和国刑法》第二百八十条第一款、第二款、第六十九条第一款、第三款、第五十二条

为空或不含有“中华人民共和国刑法”的字样，抑或是“文书内容”字段并没有遵照上文的格式撰写，即区分各部分的关键词有所缺失，则将该条原始数据丢弃不用，以保证训练数据集的质量。

综上，可从原始数据文集获得按照证据段、裁判分析段、法条推荐段、判决结果段的格式分割的优质数据集。

2.2.3　按照微调 LLM 工具 Llama_Factory 的 input 格式要求排版

在微调大模型阶段，使用开源工具 Llama_Factory[3]，它支持 *alpaca* 和 *sharegpt* 数据集格式，作为微调大模型时的输入。本项目使用 *alpaca* 格式的数据集，其格式如图**??**所示。

instruction 字段在业内称之为 *prompt*，是提示工程 (*Prompt engineering*) 的产物。任务的描述会被嵌入到输入中，作为数据集的一部分，而不是用参数显式地与数据集分离**prompt 设计**。*prompt* 引导大模型对任务有清晰完整的认知，从而在文本生成等下游任务上表现出较强的逻辑性和一致性。因此，设计合适的引导词

3　项目地址：https://github.com/hiyouga/LLaMA-Factory, 详细介绍见下一章

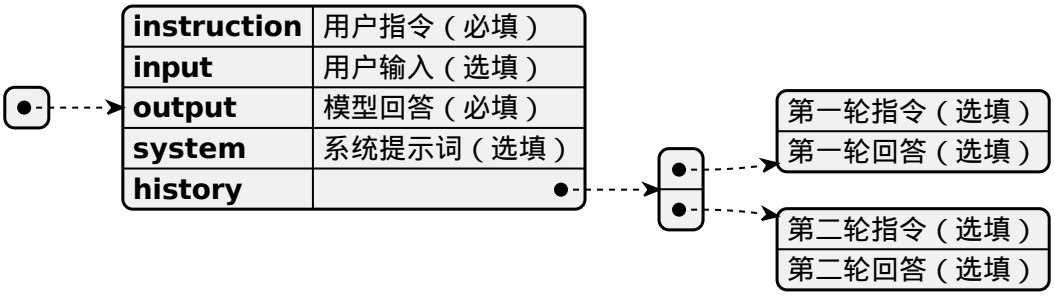


图 2-1　*alpaca* 格式的数据集

(*prompt*) 对后续的微调大模型的成功与否起着重要作用。本项目中，*prompt* 的作用是引导大模型基座对中国司法裁判文书中刑事一审判决书的格式有预先的认识，并学习各部分的起始结束的特征及主要内容，从而根据所给出的文本推理出下文。图**??**是本项目设计的 *prompt* 的例子。

> 你将完成生成中国司法裁判文书中刑事一审判决书的任务，该类文书分为以下几段：证据段、裁判分析段、引用法条段、判决结果段。接下来将提供裁判分析段，请你生成判决结果段，不要生成其他段落。分析段如下：

图 2-2　prompt

input 字段填入输入的文本，是需要大模型从中推理出可用信息的内容。在后续微调阶段时，会将 *prompt* 与 *input* 拼接，形成连贯的段落。将这两者分开构建，是为了使大模型区分引导词与正文，从而推理和生成文本时避免互相干扰。在本项目中，是裁判分析段。

output 字段填入应该用于回答 *input* 的正确答案，也是后续使用大模型推理时希望大模型生成的内容。大模型通过对 *input* 到 *output* 的上下文逻辑的学习，形成对数据集文本的推理与文本生成能力。在本项目中，*output* 是引用法条段和判决结果段。

system 字段相当于前置条件，预先向大模型输入的内容，如“你是一个 *AI* 助手”。本项目未使用该字段。

history 字段是列表，存放前几轮训练时向大模型输入的指令与回答，可用于让大模型规避错误答案，强化对正确的推理方式的认识。本项目未使用该字段。

此外，在微调大模型阶段，微调 LLM 工具 Llama_Factory 对数据集有一定

限制，主要体现在文本长度上。大模型对长文本的推理与生成能力不强，过长的文本对于大模型的训练反而起到负面作用，数据集应首先保证质量，其次考虑数量的均衡**超长文本**。经过对时间、计算资源和原始文本的综合考虑和多次试验，各部分文本长度的限制如表**??**。

表 2-4　文本长度的限制

分段	限制 (字符集：UTF-8)
证据段	256~2048
裁判分析段	128~2048
引用法条段	16~96，且包含“中华人民共和国刑法”
判决结果段	128~512

2.3　刑事一审文书数据集的构建过程

构建数据集的过程实际上是一个 ETL(*Extraction-Transformation-Loading,* 提取-转换-加载) 过程，可概括为接收原始数据、处理数据、储存处理后的数据的过程。ETL 并没有一个固定的技术选型，应根据项目的规模和实际需求灵活变更**etl 介绍**。

在本项目中，原始的数据集共有 8000 多万条数据，每条数据是一份文书。这些数据包含民事和刑事的判决书、裁定书、决定书、行政书等，需要从中筛选出刑事判决书，并根据上文所述的筛选标准来过滤出高质量的数据集。由于本项目面对的构建数据集的情况是数据量大、高吞吐、需要快速读写，所以采用业内专业的高速数据流处理中间件。

经过对各数据流处理中间件的分析，本项目的提取阶段使用 Kafka 和 Flink 工具，转换阶段使用 Java 语言进行逻辑处理，加载阶段使用 ClickHouse 数据库。

数据处理流程如图**??**所示。

原始的 8000 万条数据集储存在一台 Server 中，同时它也是 Kafka 生产者 (*producer*)，它不断地向 Kafka Broker 的消息队列 (*Message Queue*) 中发送消息。这些消息可设置留存时间，本项目中设置为 7 天。

本项目使用 5 台计算机终端搭建 Flink 集群[4]，用于从 Kafka 消息队列中拉取消息并进行筛选，然后将筛选后的数据集储存在 ClickHouse 数据库中。其中 1 台

4　参考网站：https://nightlies.apache.org/flink/flink-docs-master/docs/concepts/flink-architecture

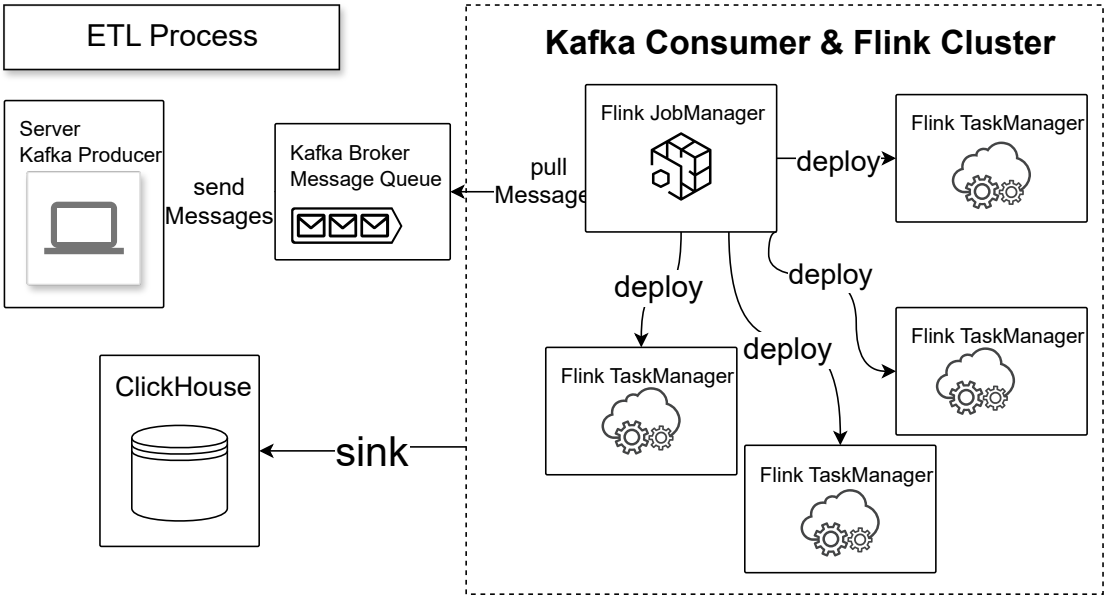


图 2-3　数据处理流程

设备为 Flink JobManager，它决定如何调度任务、处理成功完成的或运行失败的任务、处理检查点、从失败的任务中恢复等等。其他 4 台设备为 Flink TaskManager，它们接受 JobManager 的调度，执行被分配的任务，以及对数据流作缓存。Flink 集群 (*Cluster*) 可并行完成数据流处理任务，是单机处理的数倍。

ClickHouse 是数据处理的终点。在 Flink 集群处理原始数据、得到符合要求的数据集后，将数据插入 ClickHouse 表中。该过程称为 *sink*。ClickHouse 的数据库语句符合通用的 *SQL* 语法。此外，需要注意的是，ClickHouse 建表时默认使用 *memory* 引擎，即将表储存在内存中，虽然会使得响应速度快，但一旦设备断电，数据将全部丢失。因此本项目所创造的表均使用 *TinyLog* 引擎以将数据储存在硬盘中，以免丢失。一份标准的刑事判决书经筛选后，存入 ClickHouse 时使用的表结构如表**??**所示。

表 2-5　ClickHouse 刑事判决书结构

列名	类型	注释
evidence	String	证据段
analysis	String	裁判分析段
law	String	引用法条段
judgement	String	判决结果段
document	text	整篇文书

将 Kafka、Flink、ClickHouse 结合作为 ETL 过程使用的技术选型，极大地加快了数据处理的速度。经实际运行，原始数据集的 8000 万条数据在 80 小时内全

部处理完毕。若单机进行数据处理，所耗费的时间应当是其数倍。

数据处理的流程以 Java 代码形式如图**??**表示。

```
// 标记数据流标签
DataStream<DocumentPOJO> criminalCase1stInstanceStream
= processedStream.getSideOutput(CriminalCase1stTag)
.flatMap(new CriminalCase1stInstance());
// 连接至ClickHouse
ClickHouseUtil criminal1ClickHouseUtil =
new ClickHouseUtil(ClickHouseUtil.getSql(new Criminal1
   ()));
// 设置sink阶段的方式
civilCase1stInstanceStream.addSink(
   criminal1ClickHouseUtil);

// 刑事一审裁判文书处理策略
public class CriminalCase1stInstance
implements FlatMapFunction<String, DocumentPOJO> {
   @Override
   public void flatMap(String s,
               Collector<DocumentPOJO> collector) {
      try {
      // handle方法内部是对原始数据集处理得到Criminal1POJO各字
         段的过程
         Criminal1 criminal1 = handle(s);
         if (criminal1 != null) {
            collector.collect(criminal1);
         }
      } catch (Exception e) {
         e.printStackTrace();
      }
   }
}
```

图 2-4　ETL 过程代码

2.4　本章小结

本章描述了构建数据集的全过程。首先介绍本部分所运用的关键技术 (Kafka, Flink, Click-house)，然后指出数据的筛选指标及格式要求，最后阐述构建大模型所需刑事判决书的数据集的构建过程。

第三章　微调大模型

3.1　技术概论

3.1.1　LoRA 微调方法

低秩适配 (*Low-Rank Adaptation, LoRA*), 是 Microsoft 公司提出的一种通过添加低秩约束，使得每层的更新矩阵相似，以减少适配下游任务的训练参数的微调大模型方法**lora**。由于不需要将所有参数载入显存内，LoRA 的主要优点是节省计算资源，它在原有模型中增加一条旁路，在每一个线性层进行参数的增量。在实践中，可基于某个大语言模型基座，通过添加不同的训练参数和数据集，得到拥有不同下游任务能力的大模型。

本项目所能支配的计算资源为 2 张 NVIDIA-Tesla-V100-32G，无法做到全量微调，因此使用 LoRA 增量微调方法。

3.1.2　LLM 基座

通过对所能支配的计算资源、大模型的参数量、对中文的支持程度等因素的综合评判，本项目选择以下 3 个大模型基座，在其基础上作 LoRA 微调。

Llama2-7B-hf

Llama2[1]是 meta 公司于 2023 年发布的一个预训练和微调的文本生成类大语言模型。本项目选取 7B 参数量的 Llama2 作为训练基座之一，其主要特点是神经网络结构较好，训练效率高，但支持的文本长度仅为 4K，且中文生成能力较弱**llama2**。此版本为适配 huggingface transformers 的特供版。

1　https://huggingface.co/meta-llama/Llama-2-7b-hf

ChatGLM2-6B

ChatGLM[2]是清华大学于 2022 年发布的一个通用语言文本生成大语言模型基座，它在自然语言理解、无条件生成和条件生成方面均有出色表现[glm2]。ChatGLM2 的支持的上下文长度为 32K，且部署门槛低，对中文支持较好。

Qwen1.5-1.8B

Qwen1.5[3]是 Alibaba 公司于 2023 年发布的一个大语言模型基座，是 Qwen2 的 beta 版本。Qwen 在对话及文本生成领域表现出的性能较好，且支持 32K 长度的文本长度，支持多种语言[qwen]。由于在部署 Qwen1.5 时，使用 *Bfloat16* 精度需要 GPU 的计算能力与 NVIDIA-Tesla-V100 不兼容，所以使用 *float16* 精度。同时，Qwen 对显存大小的要求较高，经试验，1.8B 的模型大小是适合的。

3.1.3 Llama_factory

Llama_factory[4]是一个可供多种大模型进行预训练和微调的统一平台，集模型加载、数据加载和训练器为一体。它使用可扩展模块统一了各种高效的微调方法，使数百个大模型能够以最小的资源和高吞吐量进行微调[llamafactory]。使用时，只需要将大模型基座和按照一定格式构成的数据集传入，并可通过可视化网页进行参数的调整，即可完成大模型微调。它支持多种微调方法，如全参数训练、部分参数训练、LoRA、QLoRA 等，并支持多 GPU 加速。

此外，Llama_factory 支持将大模型以 Open AI 方式部署，即可通过 http 访问大模型，并支持 vLLM 加速，以节省推理时间及显存占用。

3.2 训练刑事一审判决书时的参数

本项目使用 2 张 GPU 进行训练。LoRA 微调方法下，使用命令行训练时的脚本格式如图**??**所示。

2 https://huggingface.co/THUDM/chatglm2-6b

3 https://huggingface.co/Qwen/Qwen1.5-1.8B

4 https://github.com/hiyouga/LLaMA-Factory

```
CUDA_VISIBLE_DEVICES=0,1 accelerate launch \
   --config_file ./examples/accelerate/single_config.
      yaml \
   ./src/train_bash.py \
   --（参数名） （可能的参数值）后略
```

图 3-1　大模型训练脚本样式

若需要查看参数文档,可在Llama_factory项目中使用`python src/train_bash.py -h`命令以访问。

以下是重要参数的介绍及本项目的选值。

`CUDA_VISIBLE_DEVICES=0,1`

指定参与本次训练的 NVIDIA GPU 的序号。本项目所能支配的计算资源为 2 张 NVIDIA-Tesla-V100-32G。

`--stage sft`

指定本次训练方法。本项目使用指令监督微调 (*Supervised Fine-tuning, sft*)，是指在已预先使用通用语料训练好的大模型的基础上，使用有标注的、面向特定任务的数据集语料来进行进一步的微调，从而使得模型具备完成特定下游任务的能力。这与本项目的目的相符合。

`--model_name_or_path ./Llama-2-7b-hf`

指定模型的路径。本项目使用 Llama-2-7b-hf、chatGLM2-6B、Qwen1.5-1.8B，这些项目可在 huggingface.co 网站上获得，通过 huggingface-cli 下载至本地[5]。

`--dataset criminal1_analysis2judgement`

指定数据集。数据集需要上传至`./data`文件夹，并在`./data/dataset_info.json`中注册该数据集，指定文件及数据格式。本项目使用 *alpaca* 数据集格式，数据量为刑事判决书“裁判分析段”至“法条推荐段”和“裁判分析段”至“判决结果段”各 10 万篇。

`--finetuning_type lora`

指定微调方法。Llama_factory 支持全参数训练、部分参数训练、LoRA、QLoRA 等多种微调方法。本项目使用 LoRA。

5　下载方式可参考 https://huggingface.co/docs/hub/en/models-adding-libraries#integrate-your-library-with-the-hub

`--lora_target q_proj,v_proj`

指定需要进行 LoRA 微调的模块。这里的 q_proj 即为注意力机制中的 W_q，v_proj 即为注意力机制中的 W_v。对于找到的每一个目标层，会创建一个新的 LoRA 层进行替换**LoRA 方法**。

`--lora_rank 16`

指定 LoRA 微调时使用的低秩矩阵的维度。*rank* 越大，微调后大模型自由度越高，但会增加显存占用和训练时长。实践中，该项的取值一般为 4~32，本项目取 16。

`--cutoff_len 2048`

指定标记化 (*tokenization*) 后大模型输入的截止长度。由于大模型在长文本方面生成能力较弱，且本项目计算资源有限，此项的值为 2048(2K)。

`--per_device_train_batch_size 1`

指定每个设备（本项目为 GPU）训练时的批次大小。值越大，所需显存越多，处理一个 epoch 的时间越少，但达到相同精度所需要的 epoch 数量越多，可能会使大模型泛化能力下降**batchsize**。本项目综合考量了训练规模和计算资源，并经试验后，将此值设为 1。

`--gradient_accumulation_steps 2`

指定梯度累积次数。它将 batch size 进一步扩大，每获取 1 个 batch 的数据，计算一次梯度且不清空，直到累加到该值后，根据累加梯度更新参数。此值越大，越能够节省显存，但是同时需要调高学习率，否则大模型生成效果较差。本项目将此值设为 2。

`--learning_rate 5e-5`

指定学习率。它定义了神经网络相对于损失梯度下降的权重调整。如果学习率偏大，大模型将跳过最优解。如果它偏小，需要增加迭代次数以使得 loss 收敛**batchsize**。

`--num_train_epochs 3.0`

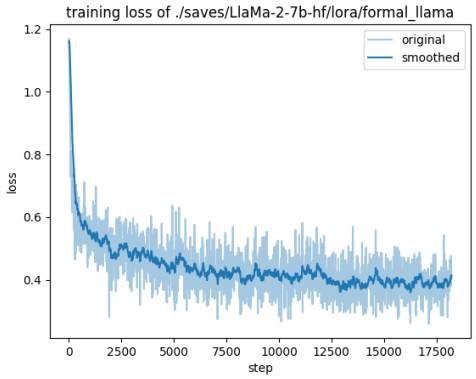
指定训练轮数。它决定了数据集将加载入显存的次数，即大模型学习数据集的次数。如果次数偏少，则训练效果较差，loss 未收敛；如果次数偏多，可能导致大模型出现过拟合现象，降低其泛化能力**epoch**。一般来说，合适的 epoch 数量出现在训练 loss 降低并趋于稳定时。本项目经试验后，将 epoch 设置为 3.0。

以上各值与大模型微调效果息息相关，也因模型基座、计算资源等因素而不同。微调时，应根据现实情况动态调整。

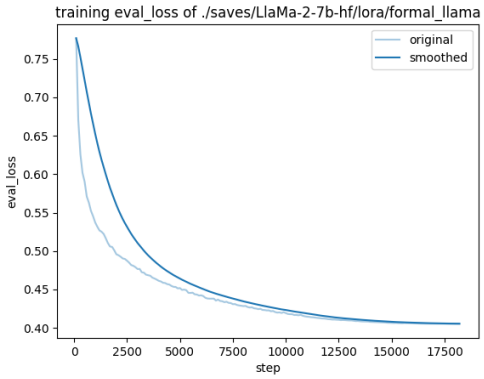
3.3 微调效果

业界常用`loss`来表示模型的训练效果。预测值与实际值 (*ground truth*) 之间会存在差异，loss 意味着模型因未能给出期望的值而受到的惩罚。loss 分为 train loss 和 eval loss，前者表示训练时误差，后者表示验证时误差。train loss 值越高或训练期间波动较大，表示模型训练效果差。若训练结束时 train loss 未收敛，表示训练尚未结束，需要增大 epoch 或学习率。

图**??**、**??**、**??**是本项目微调的 3 个大模型基座的 loss 图。

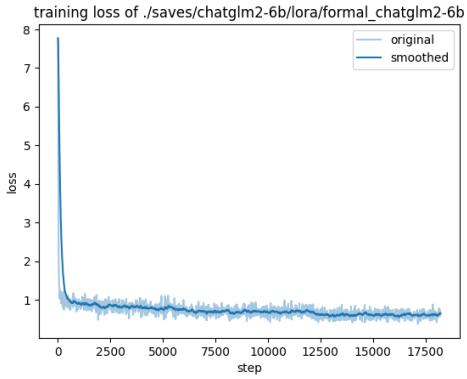


train loss

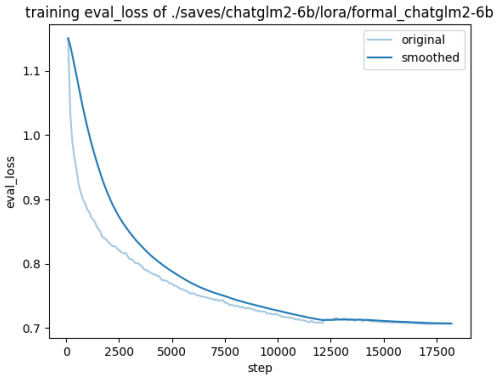


eval loss

图 3-2　Llama2-7B-hf

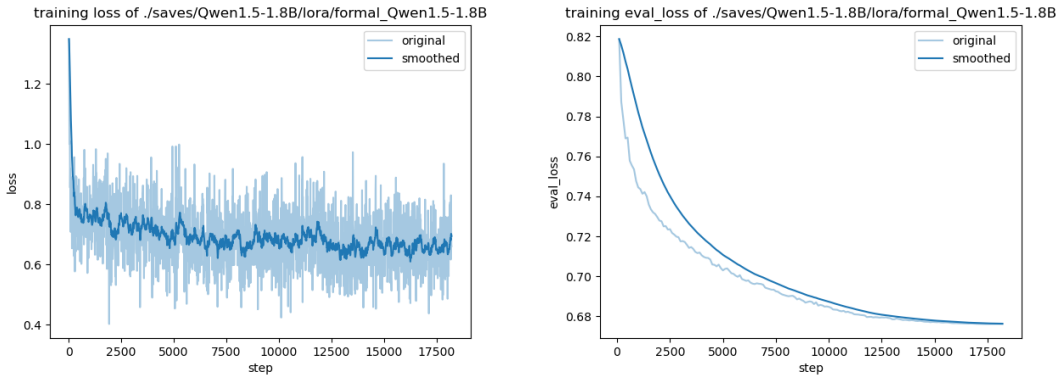


train loss



eval loss

图 3-3　ChatGLM2-6B



train loss　　eval loss

图 3-4　Qwen1.5-1.8B

训练理想的情况是 train loss 和 eval loss 均平缓下降至某个较小区间，且两值相近。可以看到，上图均展现出训练效果符合理想情况，说明训练效果较好。

3.4　参数合并

由于后续使用大模型进行推理时，使用 vLLM 进行加速，而 vLLM 不支持 LoRA 微调后的大模型,所以需要将其与原大模型基座进行参数合并。这同时也会减少推理延迟，因为合并前推理时需要分别加载基座模型和 LoRA 模型参数合并。

参数合并的脚本示例如图**??**所示。

```
CUDA_VISIBLE_DEVICES=0 python ./src/export_model.py \
    --model_name_or_path meta-llama/Llama-2-7b-hf \
    --adapter_name_or_path ./saves/LLaMA/lora/sft \
    --template default \
    --finetuning_type lora \
    --export_dir ../../models/llama2-7b-sft \
    --export_size 2 \
    --export_legacy_format False
```

图 3-5　参数合并

3.5 本章小结

本章介绍了微调大模型的全过程。首先介绍本部分所运用的关键技术 (Llama2-7B-hf,ChatGLM2-6B, Qwen1.5-1.8B, LoRA, Llama_factory)，然后解释微调训练大模型时所设置的参数设置的含义，最后说明使用微调后的大模型进行推理的先决准备工作。

第四章　大模型推理

4.1　技术概论

4.1.1　vLLM 加速

大语言模型执行推理任务时，对显存的大小要求较高，每个请求的 key-value 缓存会导致显存碎片化和冗余复制。来自 UC Berkeley 的团队受到操作系统中虚拟内存分页技术的启发，开发出了基于注意力算法的 vLLM[1] 系统，以缓解显存浪费和占用高的问题。评估表示，vLLM 在相同的延迟水平下，将大模型的吞吐量提高了 2-4 倍**vllm**。

4.2　部署方式

本项目在部署大模型时，采用 vLLM 以加速推理速度和减少显存占用，并使用 Open AI 风格部署，即可通过 http-post 访问大模型。

部署脚本如图**??**所示。

```
python -m vllm.entrypoints.openai.api_server
    --model /path/to/model/ --port xxxx
```

图 4-1　部署大模型

其中，根据每个大模型的差异，Qwen1.5-1.8B 需要添加`-dtype=half`，ChatGLM2-6B 需要添加`--trust_remote_code`。

部署完成后，可向`http://localhost:xxxx/v1/completions`发起 POST 请求，其格式应如图**??**所示。

POST 返回的结果是 *json* 格式的数据，其字段如图**??**所示。

1　https://github.com/vllm-project/vllm

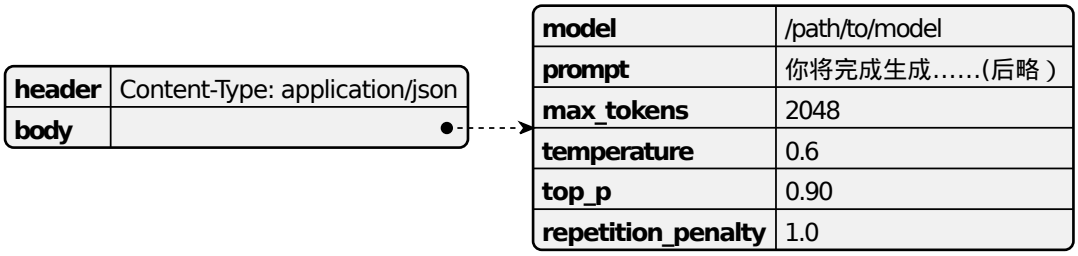


图 4-2　POST 请求格式

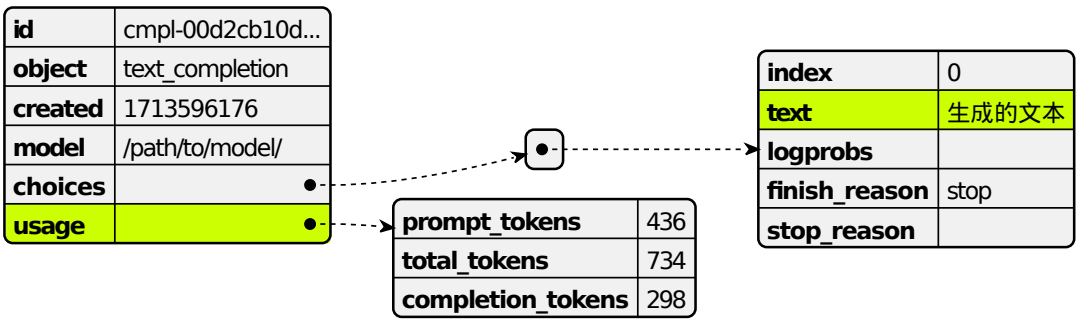


图 4-3　POST 返回结果

在 POST 返回结果中，生成的文本位于`"choices"[0]->"text"`中。此外，可通过`"usage"`中的字段以查看本次推理的 tokens 信息。

4.3　推理刑事一审判决书的技巧

4.3.1　设置推理参数

向部署好的大模型发送 POST 请求时，`body`内可设置多种参数，以调整生成效果。下列是本项目使用的参数及其意义。

`--max_tokens 2048`

指定希望大模型生成的文本的最大 token 数。它限制了模型输出的长度，与大模型的长文本生成能力有关。由于在构建数据集阶段，将输入输出文本长度限制在 2048，所以在使用大模型推理时，亦将期望输出的文本长度设置为 2048。

`--temperature 0.6`

温度 (*temperature*) 关系着在大模型推理时每个词的概率分布，取值范围是 0~1。值越低，越容易选取到概率大的字。因此，若希望生成文本偏保守，则调低该值。经测试，本项目训练的大模型中，若将温度取 0.5 以下，则生成文本为空的概率较大。因此，该值设为 0.6。

`--top_p 0.90`

top_p 的含义是，从候选的 tokens 里按百分比排序，大模型从累计概率大于或等于 top_p 的候选词集合中随机选择一个。因此，该值越大，生成的文本越具有多样性，创作空间越大。经测试，本项目将该值设为 0.90。

`--repetition_penalty 1.0`

repetition_penalty 的含义是，对于大模型先前生成过的 token，又重复的生成了该 token 的惩罚力度。值越大，惩罚力度越大，生成文本的重复性越小，风格越保守。经测试，该值大于 1.0 时，生成文本为空的概率较大，因此本项目将该值设为 1.0，为完全不惩罚。

4.3.2 设置诱导词

本项目使用大模型的方法，是将任务描述以自然语言文本的形式来表达的上下文学习 (*in-context learning, ICL*)。这也是业界最常使用的方法。使用时，将一段描述任务的文本发送至大模型，在本项目中，是在向大模型发起 POST 请求时，将任务描述写入`"body"->"prompt"`内。大模型会根据生成策略，在该文本后接续生成内容。

生成的原理是，设 $\boldsymbol{D}_k = \{f(x_1, y_1), ..., f(x_k, y_k)\}$，代表由 k 个样例（即训练和微调时的数据集）组成的一组示范。给定任务描述 $\boldsymbol{I}$、示范 $\boldsymbol{D}_k$ 以及使用大模型推理时输入的 x_{k+1}，大模型生成的输出 y_{k+1} 的预测可以用如下公式表示**预测公式**：

$$\mathrm{LLM}(\boldsymbol{I}, \underbrace{f(x_1, y_1), \cdots, f(x_k, y_k)}_{\text{示范}}, f(\underbrace{x_{k+1}}_{\text{输入}}, \underbrace{\_\_\_\_}_{\text{答案}})) \rightarrow y_{k+1}。$$

因此，使用大模型推理时的输入应能够正确引导大模型的生成方向。经试验，在`prompt`后加上希望大模型生成的段落的名称，如“法条推荐段”“判决结果段”，作为诱导词，大模型能够很好地将生成文本的内容与该段落贴合。若不添加诱导词，大模型很有可能会向错误的方向生成文本，即使`prompt`预先说明希望大模型生成的段落。

图**??**是使用大模型推理时采用的`prompt`的一个完整的例子。

经测算，若不添加诱导词，大模型有 20% 左右的的概率向错误的方向生成

你将完成生成中国司法裁判文书中刑事一审判决书的任务，该类文书分为以下几段：证据段、裁判分析段、法条推荐段、判决结果段。接下来将提供裁判分析段，请你生成判决结果段，不要生成其他段落。分析段如下：XXXX分析段内容省略XXXX。判决结果段：

图 4-4　推理 prompt

或生成内容为空；添加诱导词后，向正确的方向生成文本的概率几乎为 100%。

4.4　本章小结

本章介绍了使用大模型进行推理的全过程。首先介绍本部分所运用的关键技术 (vLLM)，然后说明大模型的部署方式，最后解释使用大模型进行推理时的参数设置并介绍增强大模型推理结果的方法。

第五章　性能评测

5.1　技术概论

5.1.1　BERTScore

BERTScore 是一个用于评估文本相似度的工具，常用于评测机器生成文本的质量。它将两个句子的相似度计算为其标记嵌入之间的余弦相似度之和，使用上下文嵌入计算标识的相似性，而不是精确匹配[bertscore]。BERTScore 偏向于对语义进行评分，能够以人类熟悉的表达方式为标准来评判文本的相似度。评测结果以 P(查准率，*Precision*)、R(召回率，*Recall*) 和 F1($\frac{2\times P\times R}{P+R}$) 表示。由于 F1 是 P 和 R 的调和，因此只需关注 F1，其值越大表示文本越相似。

5.1.2　ROUGE_L

ROUGE(*Recall-Oriented Understudy for Gisting Evaluation*), 是一种基于召回率指标的评价算法。它通过统计基准文本与被比较文本之间重叠的基本单元 (*n* 元语法、词序列和词对) 的数目，来评价被比较文本的质量。ROUGE_L 是其中比较最长公共子序列 (*longest common subsequence, LCS*) 的方法，计算时参考机器译文 C 和参考译文 S 的最长公共子序列[rougel]。其评测结果同样给出 P、R、F1，但由于 ROUGE 是基于召回率指标的算法，最终只需要关注 R（召回率），值越大表示文本越相似。

本项目使用两个指标来评测大模型生成文本的质量，是因为两者各有侧重。BERTScore 侧重对语义的评价，能够判定一种含义的多种表达，不强求生成的文本的用词必须一致，更符合人类的评判方式。而 ROUGE_L 通过比较最长公共子序列，固定了生成文本的用词，丢失了词义的灵活性，但这同时保证了用词的严谨性和一致性，这在有着固定格式和精准用词要求的法院裁判文书中是有较高

价值的。通过两个指标的考量，能够多方面高维度评判大模型生成文本的质量。

总之，BERTScore 给出生成文本的语义还原性，ROUGE_L 给出生成文本的用词一致性。

5.2 评测过程

首先准备 500 份刑事判决书，按上文所述分为证据段、裁判分析段、法条推荐段、判决结果段。在“裁判分析段”至“法条推荐段”和“裁判分析段”至“判决结果段”两个方向，以“裁判分析段”为输入，分别在 Llama2-7B-hf 基座、ChatGLM2-6B 基座、Qwen1.5-1.8B 基座及其各自微调后的模型，即 6 个大模型上得到生成的“法条推荐段”和“判决结果段”文本，并将他们与原文书的这两段在 BERTScore 和 ROUGE_L 两个指标上进行评价，得到具体分数，并作分析。

5.3 生成刑事一审判决书法条推荐段和判决结果段的示例

以图**??**的prompt为例，在 ChatGLM2-6B 基座及其微调后的模型下展现结果。其结果能够代表模型生成文本的平均水平。

```
你将完成生成中国司法裁判文书中刑事一审判决书的任务，该类文书分为以下几段：证据段、分析段、法条推荐段、判决结果段。接下来将提供分析段，请你生成(法条推荐段|判决结果段)，不要生成其他段落。分析段如下：
本院认为，被告人邱某明知他人开设赌场，仍然积极参与并听从他人的安排，在赌场外放哨，逃避公安机关查处辅助赌场的运转，从中谋取非法利益，被告人的行为触犯了《中华人民共和国刑法》第三百零三条第二款之规定，已构成开设赌场罪。公诉机关指控被告人邱某犯开设赌场罪的事实清楚，证据确实、充分，指控罪名成立。案发后被告人邱某主动到公安机关投案，并如实供述了自己的犯罪事实，系自首，依法可以从轻处罚；被告人邱某在开设赌场行为中根据他人的安排只实施了部分犯罪行为，参与时间短且获利较少，在共同犯罪中起次要、辅助作用，系从犯，依法应当从轻处罚；被告人邱某主动退缴了非法所得，对其予以酌情从轻处罚。被告人邱某退缴的赃款应依法予以上缴国库。(法条推荐段|判决结果段)：
```

图 5-1 测试 prompt

5.3.1 “裁判分析段”至“法条推荐段”

如表**??**所示，模型基座生成的文本并未按照原文本格式所撰写，且内容为“判决结果段”应当出现的语句；而微调后的大模型在学习大量数据集后，能够按照原文本格式撰写，且内容相似度较高，基本能够还原。

将两生成文本与原文本在 BERTScore 和 ROUGE_L 两个指标上进行评价，得分如表**??**所示，微调后的模型得分显著高于原模型基座。

表 5-1 “裁判分析段”至“法条推荐段”结果示例

原文本	ChatGLM2 基座生成文本	微调模型生成文本
《中华人民共和国刑法》：第三百零三条第一款，第三百零三条，第二十五条第一款，第二十七条第一款，第四十五条，第四十七条，第五十二条，第六十四条，第六十七条第一款”	本院判决如下：邱某犯开设赌场罪，判处有期徒刑六个月，并处罚金人民币五万元。根据《中华人民共和国刑法》第三百零三条第二款之规定，开设赌场罪是指违反国家法律法规，在公共场所、地下建筑、交通工具等场所开设赌场，或者明知他人开设赌场，为其提供资金、场地、运输、网络、人员等帮助，从中谋取非法利益的行为。	《中华人民共和国刑法》第三百零三条第二款、第二十五条第一款、第二十七条第一款、第六十七条第一款、第六十四条、第五十二条、第五十三条、第六十一条、第六十二条、第六十四条

表 5-2 “裁判分析段”至“法条推荐段”结果示例得分

模型	分数					
	BERTScore			ROUGE_L		
	P	R	F1	P	R	F1
ChatGLM2-基座	0.5153	0.7509	0.6112	0.6061	0.0433	0.0808
ChatGLM2-微调	0.9081	0.9182	0.9131	0.5152	0.4595	0.4857

5.3.2 “裁判分析段”至“判决结果段”

如表**??**所示，原模型基座除了生成“判决结果段”，还生成了“法条推荐段”。此外，基座生成的文本缺少裁判文书的固定语句样式，如“刑期从判决执行之日起计算。判决执行以前先行羁押的……”等。这些都是需要大模型从数据集中学习才能具有的生成能力。

将两生成文本与原文本在 BERTScore 和 ROUGE_L 两个指标上进行评价，得分如表**??**所示，微调后的模型得分显著高于原模型基座。

表 5-3 “裁判分析段”至“判决结果段”结果示例

原文本	ChatGLM2 基座生成文本	微调模型生成文本
判决如下：一、被告人邱某犯开设赌场罪，判处有期徒刑七个月，并处罚金人民币一千元。(刑期从判决生效之日起计算。判决执行以前先行羁押的，羁押一日折抵刑期一日，即自 2015 年 7 月 22 日起至 2016 年 2 月 21 日止。罚金已缴纳)。二、被告人邱某主动退缴的赃款人民币一千元，依法上缴国库。	综上所述，根据《中华人民共和国刑法》第三百零三条第二款之规定，判决如下：一、被告人邱某犯开设赌场罪，判处有期徒刑六个月，并处罚金人民币二万元。二、本案经审理查明，邱某犯罪所得及其收益应当依法予以追缴，上缴国库。	判决如下：一、被告人邱某犯开设赌场罪，判处有期徒刑六个月，并处罚金人民币一万元（刑期从判决执行之日起计算。判决执行以前先行羁押的，羁押一日折抵刑期一日，即自 2013 年 11 月 27 日起至 2014 年 5 月 26 日止。罚金限判决生效后十日内缴纳)。二、本案中非法所得人民币 35000 元，继续追缴，上缴国库。

表 5-4 “裁判分析段”至“判决结果段”结果示例得分

模型	分数					
	BERTScore			ROUGE_L		
	P	R	F1	P	R	F1
ChatGLM2-基座	0.7433	0.7326	0.7379	0.3205	0.4464	0.3731
ChatGLM2-微调	0.9195	0.9363	0.9278	0.7436	0.7160	0.7296

5.4 评测结果

5.4.1 “裁判分析段”至“法条推荐段”

表**??**、**??**、**??**是 3 个大模型基座与其微调后的模型在 BERTScore-F1 指标上的得分情况。

模型	BERTScore-F1		
	平均数	中位数	方差
Llama2-基座	0.5008	0.4912	0.0026
Llama2-微调	0.8534	0.8542	0.0043

表 5-5 “裁判分析段”至“法条推荐段”, Llama2-7B-hf, BERTScore

模型	BERTScore-F1		
	平均数	中位数	方差
ChatGLM2-基座	0.6399	0.6300	0.0050
ChatGLM2-微调	0.8742	0.8882	0.0032

表 5-6 “裁判分析段”至“法条推荐段”, ChatGLM2-6B, BERTScore

模型	BERTScore-F1		
	平均数	中位数	方差
Qwen1.5-基座	0.6492	0.6564	0.0070
Qwen1.5-微调	0.9250	0.9432	0.0021

表 5-7 “裁判分析段”至“法条推荐段”, Qwen1.5-1.8B, BERTScore

表**??**、**??**、**??**是 3 个大模型基座与其微调后的模型在 ROUGE_L-召回率指标上的得分情况。

模型	ROUGE_L-召回率		
	平均数	中位数	方差
Llama2-基座	0.0488	0.0387	0.0017
Llama2-微调	0.4929	0.5098	0.0378

表 5-8 “裁判分析段”至“法条推荐段”, Llama2-7B-hf, ROUGE_L

模型	ROUGE_L-召回率		
	平均数	中位数	方差
ChatGLM2-基座	0.1278	0.1147	0.0048
ChatGLM2-微调	0.4450	0.4444	0.0207

表 5-9 “裁判分析段”至“法条推荐段”, ChatGLM2-6B, ROUGE_L

模型	ROUGE_L-召回率		
	平均数	中位数	方差
Qwen1.5-基座	0.1278	0.1183	0.0053
Qwen1.5-微调	0.6755	0.7167	0.0271

表 5-10 “裁判分析段”至“法条推荐段”, Qwen1.5-1.8B, ROUGE_L

图**??**是对 3 个微调后大模型在“裁判分析段”至“法条推荐段”生成文本的横向比较。

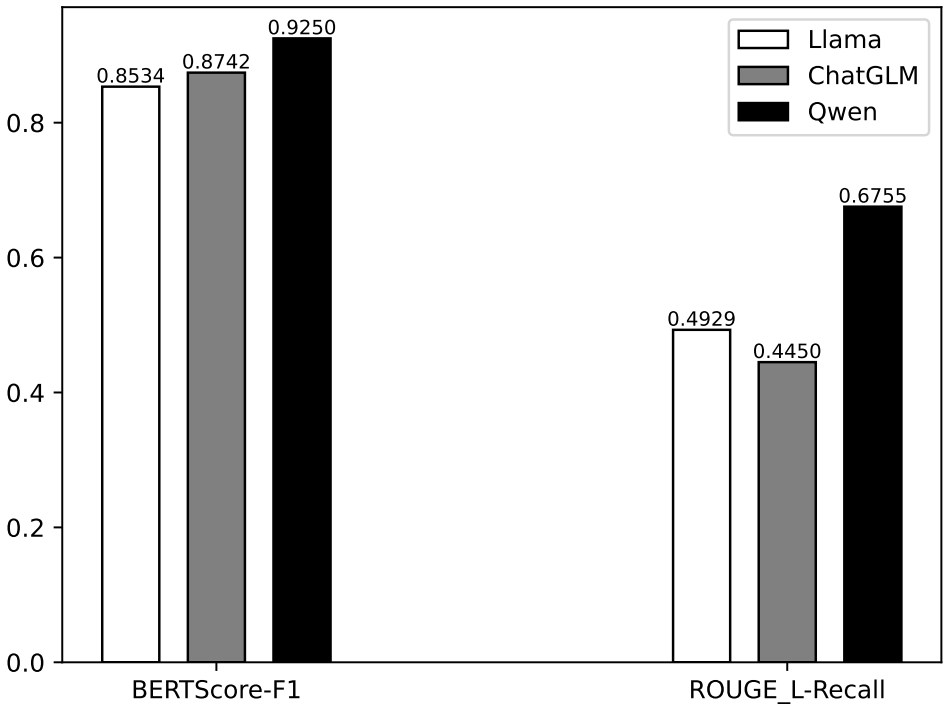


图 5-2 “裁判分析段”至“法条推荐段”的横向比较

5.4.2 “裁判分析段”至“判决结果段”

表??、??、??是 3 个大模型基座与其微调后的模型在 BERTScore-F1 指标上的得分情况。

	BERTScore-F1		
模型	平均数	中位数	方差
Llama2-基座	0.5886	0.5961	0.0054
Llama2-微调	0.8199	0.8206	0.0024

表 5-11 “裁判分析段”至“判决结果段”, Llama2-7B-hf, BERTScore

	BERTScore-F1		
模型	平均数	中位数	方差
ChatGLM2-基座	0.6697	0.6771	0.0047
ChatGLM2-微调	0.8802	0.8851	0.0022

表 5-12 “裁判分析段”至“判决结果段”, ChatGLM2-6B, BERTScore

模型	BERTScore-F1		
	平均数	中位数	方差
Qwen1.5-基座	0.6753	0.6790	0.0038
Qwen1.5-微调	0.8824	0.8868	0.0024

表 5-13 “裁判分析段”至“判决结果段”, Qwen1.5-1.8B, BERTScore

表**??**、**??**、**??**是 3 个大模型基座与其微调后的模型在 ROUGE_L-召回率指标上的得分情况。

模型	ROUGE_L-召回率		
	平均数	中位数	方差
Llama2-基座	0.1056	0.1067	0.0040
Llama2-微调	0.4836	0.4784	0.0199

表 5-14 “裁判分析段”至“判决结果段”, Llama2-7B-hf, ROUGE_L

模型	ROUGE_L-召回率		
	平均数	中位数	方差
ChatGLM2-基座	0.2208	0.2213	0.0098
ChatGLM2-微调	0.6413	0.6582	0.0134

表 5-15 “裁判分析段”至“判决结果段”, ChatGLM2-6B, ROUGE_L

模型	ROUGE_L-召回率		
	平均数	中位数	方差
Qwen1.5-基座	0.2300	0.2184	0.0121
Qwen1.5-微调	0.6438	0.6634	0.0158

表 5-16 “裁判分析段”至“判决结果段”, Qwen1.5-1.8B, ROUGE_L

图**??**是对 3 个微调后大模型在“裁判分析段”至“判决结果段”生成文本的横向比较。

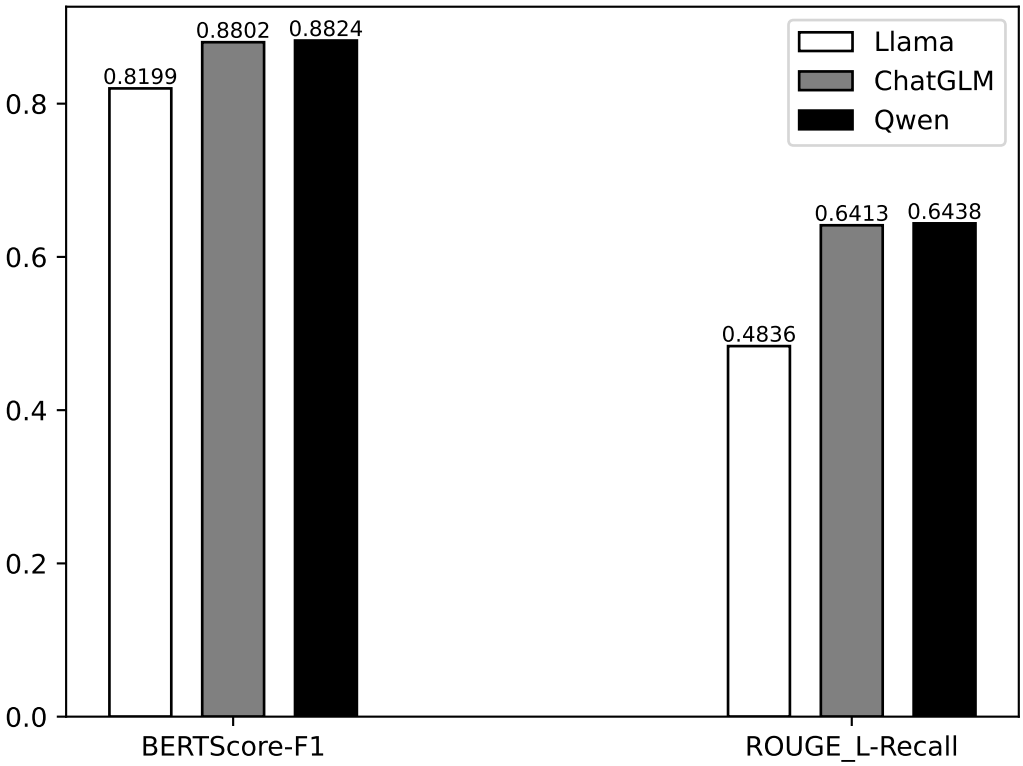


图 5-3 “裁判分析段”至“判决结果段”的横向比较

5.5 数据分析

由上表可知，相较于基座，微调后的大模型在裁判文书生成领域的性能显著提高，说明微调工作行之有效，针对特定领域的模型微调是有发展前景的。

5.5.1 得分情况

在词义还原度 (BERTScore) 方面，微调的大模型由原基座的 60% 左右的分数增长到 85% 至 90%，基本能够将原文本的语义表达清晰完整。在用词一致性 (ROUGE_L) 方面，微调的大模型由原基座的 10% 至 20% 的分数增长到 45% 至 65%，能够命中原文本的多数词语，可认为微调的大模型用词较为严谨、一致，基本符合裁判文书的要求。

5.5.2 大模型之间的比较

Llama2 相比于 ChatGLM2、Qwen1.5，效果较差，原因可能是 Llama2 对中文的支持性较差，原模型基座的编码器仅支持 1000 个汉字。ChatGLM2 表现稍逊于 Qwen1.5，原因可能是 ChatGLM2 的神经网络结构先天存在不足，由图**??**可知，微调时 ChatGLM2 初始 loss 达到 8，显著高于其他大模型基座的 1.3 左右。但由于 ChatGLM2 参数量为 6B，显著高于 Qwen1.5 的 1.8B，且 Qwen1.5 在本项目中无法使用 *Bfloat16* 精度，所以两者在最终结果上差异较小。

5.5.3 大模型的数学能力

结合 5.3 节的结果示例来看，微调的大模型在中文文本生成方面效果较好，但在罚金、日期计算等和数字有关的方面效果较差，这是大模型先天特性所致，它无法进行数学计算。在业内，往往会在大模型上外置数学计算器以提升大模型的数学能力。

5.6 本章小结

本章介绍了对大模型进行性能评测的全过程。首先介绍本部分所运用的关键技术 (BERT, ROUGE_L)，然后说明性能评测的过程，最后展现各个大模型的性能评测结果并作分析。

第六章　交互系统

本章将演示交互系统实现的内容。

6.1　智能生成段落

如图??所示，在“裁判分析段”输入框中输入文本，点击“生成结果”按钮，即在下方生成法条推荐段/判决结果段，用户可对其进行修改。页面右侧是一篇完整的形式判决书所需的信息，用户可自行填入。点击“保存结果”按钮，即将自动生成的法条推荐段/判决结果段填入右侧的输入框内。

图 6-1　从裁判分析段生成法条推荐段并保存

6.2　参数调整

如图??所示，用户可点击“高级设置”按钮，自行调整推理时参数“temperature”和“repetition_penalty”，以根据需要生成偏激进或偏保守的文本。

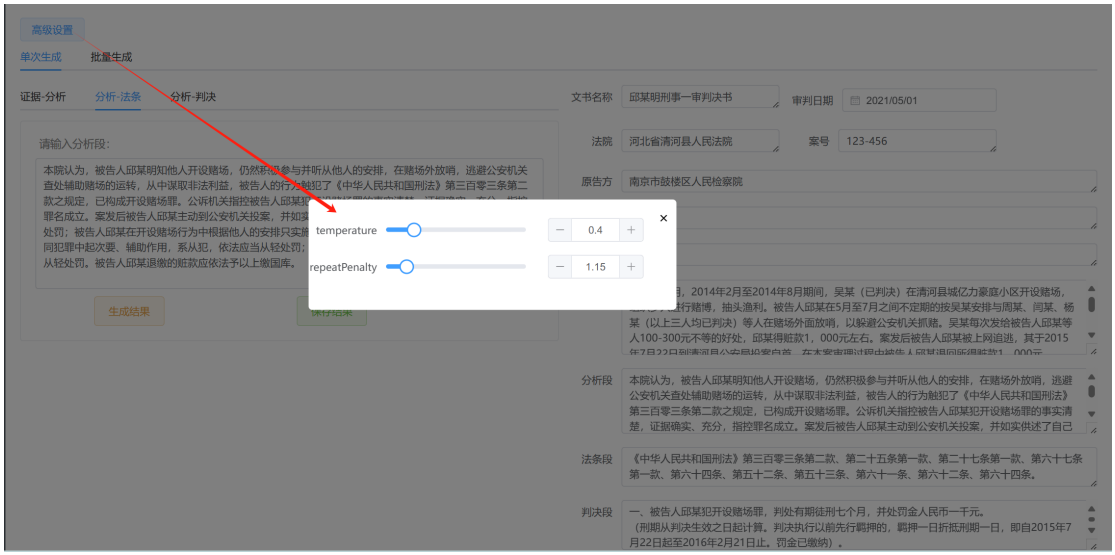


图 6-2　参数调整

6.3　批量生成

如图**??**所示，用户可上传一个 xls 格式的文件，其中第一列的每一行是大模型接受的输入文本，再选择要生成的内容，点击“提交分析”。系统会针对每一行的输入文本生成对应的输出，并添加在每一行的后一列。生成完毕后，用户可下载该文件。图**??**展现了页面的使用方法。图**??**展现了批量生成的结果，其中 A 列为输入文本即裁判分析段，B 列为大模型生成的法条推荐段，C 列为大模型生成的判决结果段。

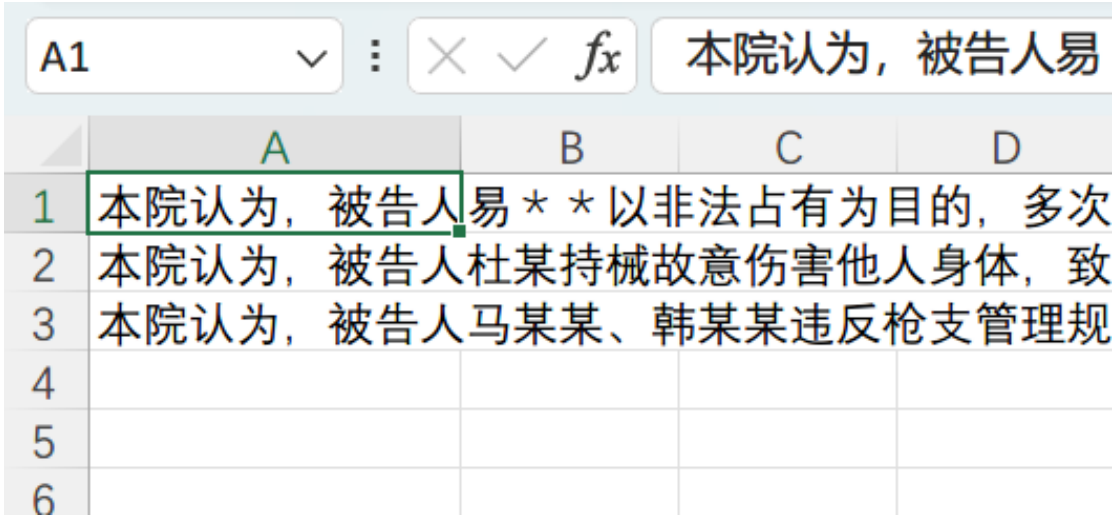


图 6-3　xls 格式

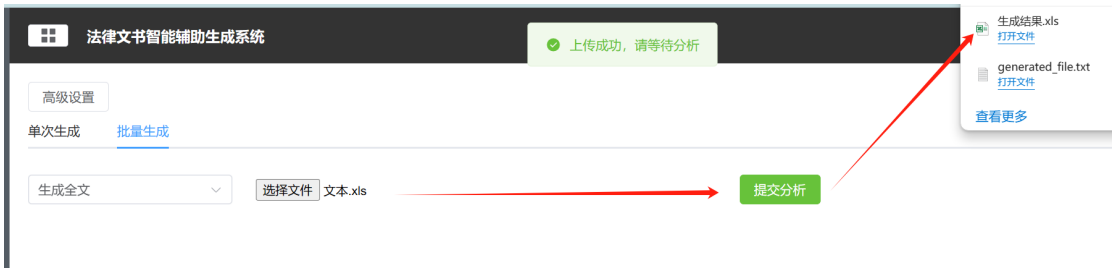


图 6-4　批量生成

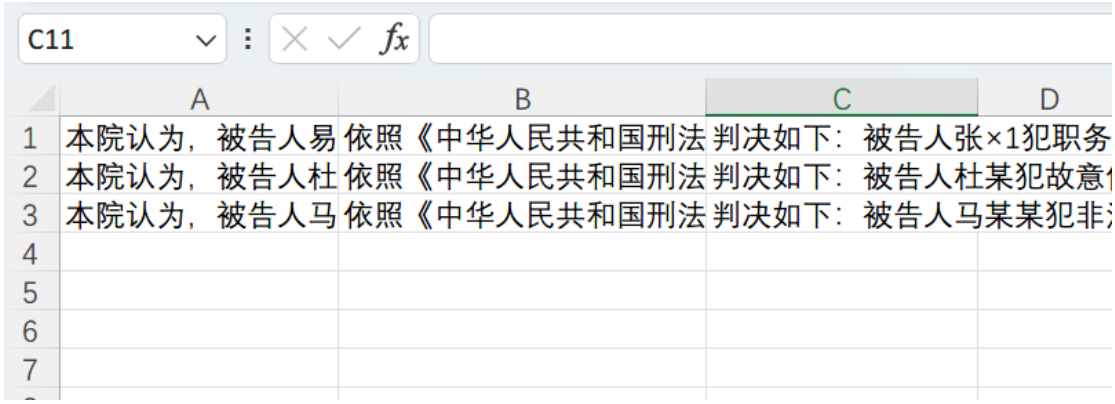


图 6-5　xls 结果

6.4　保存文书

如图??中所示，在填写必要的信息后，可点击“下载文书”按钮，将文本下载为 txt 格式的文件，以供查阅。

6.5　本章小结

本章介绍了交互系统的实现内容，并展现实际使用效果。


GitLab
面经
JSON在线解析
下载
generated_file.txt
打开文件
查看更多
文书名称
邱某明刑事一审判决书
法院
河北省清河县人民法院
案号
123-456
原告方
南京市鼓楼区人民检察院
被告方
邱某明
院方
XXX
正据事实
经审理查明，2014年2月至2014年8月期间，吴某（已判决）在清河县城亿力豪庭小区开设赌场，组织多人进行赌博，抽头渔利。被告人邱某在5月至7月之间不定期的按吴某安排与周某、闫某、杨某（以上三人均已判决）等人在赌场外面放哨，以躲避公安机关抓赌。吴某每次发给被告人邱某等人100-300元不等的好处，邱某得赃款1，000元左右。案发后被告人邱某被上网追逃，其于2015
分析段
本院认为，被告人邱某明知他人开设赌场，仍然积极参与并听从他人的安排，在赌场外放哨，逃避公安机关查处辅助赌场的运转，从中谋取非法利益。被告人的行为触犯了《中华人民共和国刑法》第三百零三条第二款之规定，已构成开设赌场罪。公诉机关指控被告人邱某犯开设赌场罪的事实清楚，证据确实、充分，指控罪名成立。案发后被告人邱某主动到公安机关投案，并如实供述了自己
法条段
《中华人民共和国刑法》第三百零三条第二款、第二十五条第一款、第二十七条第一款、第六十七条第一款、第六十四条、第五十二条、第五十三条、第六十一条、第六十二条、第六十四条。
判决段
一、被告人邱某犯开设赌场罪，判处有期徒刑七个月，并处罚金人民币一千元。
（刑期从判决生效之日起计算。判决执行以前先行羁押的，羁押一日折抵刑期一日，即自2015年7月22日起至2016年2月21日止。罚金已缴纳）。
下载文书

图 6-6　保存文书

第七章　总结与展望

7.1　总结

本文是对“基于大模型的法院文书智能生成系统”项目完成过程的详细介绍。该项目将当前业界关注度极高的大语言模型与中国法院裁判文书相结合。本文关注大模型生成刑事判决书的方向。

首先将刑事判决书分为证据段、裁判分析段、法条段、判决结果段，将包含检察机关指控的当事人罪行、法院查明的真相和意见的裁判分析段作为输入，包含当事人可能触犯的法条的法条推荐段和包含法官应当对本案件下定的裁判决定的判决结果段作为输出，并构造引导性强的 prompt，来构建出适用于大模型微调训练的数据集。从原始文书到训练数据集，由于数据量较大，因此使用高速数据流处理中间件 Kafka、Flink 和 ClickHouse。

接着，使用 LoRA 微调方法，研究合适的训练参数，对 Llama2-7B-hf、ChatGLM2-6B、Qwen1.5-1.8B 大语言模型基座进行微调，使得微调后的大语言模型在裁判文书方面具有推理并生成文本的能力。

然后，使用 vLLM 对大模型推理进行加速，研究合适的推理参数和其他推理技巧，使用 Open-AI 方式部署大模型。

接着，对训练好的大模型进行性能评测，在刑事判决书“裁判分析段”至“法条推荐段”和“裁判分析段”至“判决结果段”两个方向，以“裁判分析段”为输入，分别在 3 个模型基座及其各自微调后的模型，即 6 个大模型上得到生成的“法条推荐段”和“判决结果段”文本，并将他们与原文书的这两段在 BERTScore 和 ROUGE_L 两个指标上进行评价，得到具体分数，并作分析。结果显示，相较于基座，微调后的大模型在裁判文书生成方面的能力得到了飞跃。

此外，本项目开发了一个供司法人员使用的网页交互系统，以便于司法实践。

当前我国智慧法院正处于迈向“4.0 版”的建设期。其中，大语言模型不可

或缺，这也会是区别于“3.0 版”的一大特点。本项目能够为法官自动生成符合法律规定和个案需求的诉讼文书，作为对现有司法大模型的补充，能够大大减轻司法人员的心智负担，有效提升判案效率，努力创造更高水平的数字正义，让人民群众在每个司法案件中感受到公平正义。

7.2 展望

受限于主客观条件限制，本项目还有进一步完善的空间，具体如下。

7.2.1 硬件条件

本项目所能支配的计算资源为 2 张 NVIDIA-Tesla-V100-32G，只能对参数量为 10B 以下的大模型基座进行非全量微调（如 LoRA），或参数量为 3B 以下的基座进行全量参数微调，且微调时间较长。若能增加计算资源，就能够尝试更多的微调方法（如 QLoRA）和全量参数微调，微调效果会更进一步。此外，还能尝试更先进的大模型基座，如 ChatGLM3，其网络结构相较于 ChatGLM2 进行了较多的优化。

7.2.2 数据集质量

大模型的训练数据集的质量始终是第一位的。本项目在构建数据集时，仅过滤乱码与限制文本长度，还应考虑罪行分布情况、语句通顺度等。可考虑编写算法来进一步筛选优质数据集。

7.2.3 性能评测

由于资源的限制，无法将微调后的大模型与当今最先进的通用大模型（如 GPT-4）在裁判文书方面进行性能评测。这样做的目的是，观察在特定领域下，定向微调的小参数模型还是通用大参数模型的性能更佳，以此考量在实践中的技术选型。

致　　谢

本科四年的时光如白驹过隙，忽然而已。写下致谢时的当天，正好去了次仙林校区。四年前，我正是满怀憧憬，稍显稚嫩和懵懂，踏进仙林校区的校门。不知为何，从高一起，无论谁问起，我的目标一直是南京大学。我是幸运的。我对南京大学的期待，正如她回应我的那样，诚朴、肃穆、自由、博爱。南京大学并不约束着我什么，而是把我推向更高的平台，见到更广阔的风景。我从未后悔过选择她。

感谢葛季栋导师对我毕业设计和论文的指导。

感谢和我一同完成毕业项目设计的同学。

感谢我的父母和家人对我二十多年的养育和关心。

感谢我的大学舍友们和其他经常来我们宿舍的人。

感谢二十多年来帮助过我的人。

感谢幼儿园、小学、初中、高中、大学的老师与同学。

感谢我从小生活的苏州。

感谢我的老家徐州和四川。

感谢南京大学软件学院。

感谢南京大学。

感谢南京。

感谢中国共产党。