## OpenAtom [基于IPEX-LLM生成式](https://competition.atomgit.com/competitionInfo?id=2ac5ee25136dfdc2bca08333301b9284" \t "https://competition.atomgit.com/_blank)

## [AI(AIGC)行业场景应用开发创新赛](https://competition.atomgit.com/competitionInfo?id=2ac5ee25136dfdc2bca08333301b9284" \t "https://competition.atomgit.com/_blank)

## 法律小助手-项目说明文档

### 1. 项目名称

LawAssistant-法律小助手

### 2. 团队信息(1-5人)

* 团队名称： Zlink
* 队长名： 黄定华
* 各成员及分工：

|  |  |
| --- | --- |
| **成员** | **分工** |
| 黄定华 | 总体指导 |
| 甘展诚 | Rag |
| 罗千钧 | 前端代码 |
| 尹文隆 | 后端代码 |

* 联系邮箱： nasonw@163.com

### 3. 项目背景与目的

* 当前市场需求或技术趋势

随着社会复杂性的增加和法律需求的多样化，个人和企业越来越需要快速、准确地理解法律条文及其适用情况。传统的法律咨询服务往往受限于高昂的成本和有限的专业资源，无法满足大众化的即时法律咨询需求。此外，人工智能（AI）技术的飞速发展，尤其是自然语言处理（NLP）领域的突破，使得机器能够理解和生成高质量的人类语言，为自动化法律咨询提供了可能。

近年来，大模型的兴起和量化技术的应用极大地提高了AI系统的效率和普及性。Qwen2.5，作为阿里云开发的头部大语言模型之一，1.5B参数的量化至INT4/INT8格式，不仅大幅降低了计算资源需求，而且保持了较高的预测准确性，特别适合在CPU上高效运行，这为法律咨询场景下的AI部署开辟了新的道路。

* 项目希望解决的问题或改进的领域

我们的项目旨在解决以下关键问题：

1. **成本与效率**：传统法律咨询成本高且响应时间长，而AI法律助手可以提供即时、低成本的咨询服务。
2. **专业知识普及**：非专业人员难以理解和正确应用复杂的法律条文，AI助手能以平易近人的方式解释法律概念。
3. **个性化服务**：基于用户的具体案例，AI助手可以提供定制化建议，帮助用户理解其法律立场和可行行动方案。

* 项目目标

**总体目标**：构建一个基于Qwen2.5 1.5B CPU量化INT4/INT8模型的法律智能助手，专注于刑法和民法领域，能够准确解读和应用相关法律条文，为用户提供即时、经济高效的法律咨询和案件分析服务。

**期望成果**：

1. **开发高精度的法律咨询系统**：系统将集成刑法和民法的最新条款，结合Qwen2.5模型的强大语言理解和生成能力，提供精确的法律解读和案例分析。
2. **实现低资源消耗的部署方案**：利用INT4量化的高效特性，在普通硬件（如CPU）上实现模型的快速响应和稳定运行，降低运营成本。
3. **建立用户友好的交互界面**：设计直观的用户界面，支持自然语言输入，使非专业用户也能轻松获取专业的法律建议。
4. **持续学习与更新机制**：系统将具备自我学习和适应能力，能够随法律法规的变化自动更新知识库，确保咨询的时效性和准确性。
5. **数据隐私与安全**：实施严格的数据保护措施，确保用户咨询过程中的隐私安全，遵守所有相关的数据保护法规。

通过上述目标的实现，我们期望能够显著提升公众对法律服务的可及性和满意度，同时降低个人和企业在法律咨询方面的成本负担。

### 4. 方案介绍

1. **方案的核心思想和原理**

我们的解决方案旨在通过构建一个基于CPU推理的智能RAG系统来有效管理和利用法律的文本帮助人们解决现实实际中的法律咨询问题。通过将先进的英特尔大模型推理系统与知识库相结合，进行针对性的语义检索召回和自然语言问答来完成自动化的法律咨询服务。

1. **主要功能与特点**

该系统具备高效的数据处理、优化的模型性能、强大的知识管理和多样化的数据来源，通过友好的用户界面和高效的搜索功能，支持多轮对话并引用对应的相关知识来源，提升用户体验和信息获取的准确性。

* 自然语言问答系统：

提供基于 Qwen-2 大语言模型的问答服务，能够理解和回答用户提出的各类问题。

支持多轮对话，能够在对话上下文中进行自然语言理解和生成。

* 知识检索与增强：

能够从 Qdrant 向量数据库中检索相关知识，基于检索到的知识和用户问题，生成高质量的自然语言回答。

* 知识库管理：

目前支持通过《刑法》和《民法典》的法律条文来确定相关案件的违法性质以及适用条款

支持对知识数据的索引和快速检索。

* 用户界面：

提供友好的 Web 界面，支持用户进行自然语言提问和查看回答。

1. **设计思路与实施步骤**

通过整合 Qwen-2 大语言模型与 Qdrant向量数据库，结合 llamaindex工具快速处理结构化信息，采用 RAG 技术增强问答过程，并优化模型性能，以提供高质量、相关性强的回答。

* 数据收集与预处理：

目前数据集为：

《中华人民共和国民法典》-2020年5月28日版本

《中华人民共和国刑法》-2023年12月29日修订版

使用llamaindex工具进行pdf文件分片预处理以及元数据信息预处理和pipeline构建。

* 知识库构建：

将处理后的信息 chunk 按照向量化存储到 qdrant 向量数据库，目前向量数据库采用in memory模式。

对存储的信息进行索引，支持快速检索。

* 问答系统开发：

主要实现知识检索模块，从 Qdrant向量数据库中检索和召回相关知识片段。

检索出来的node按照语义相似度进行reranker重排序；

集成 Qwen-2 大语言模型，进行自然语言理解和生成，实现用户对话功能，通过解析用户的问题，将检索到的知识和用户问题输入 Qwen-2 模型，按照提示词工程生成回答。

实现结果返回模块，返回生成的回答和相关知识的来源引用呈现（暂未处理）。

* 用户界面开发：

使用 Gradio开发前端界面原型，提供自然语言提问和文档查看和检索查阅等功能。

* 模型优化与部署：

使用了Qwen2.5大语言模型，基于采用Intelcpu的硬件平台，并且使用ipex-llm对低精度部署的大模型进行cpu推理加速。摆脱了对GPU的依赖，极大的降低了使用的门槛和成本，详细参考技术特点的模型优化小节；

### 5. 技术特点

1. **使用的算法、模型及数据集**

* 大语言模型：

Qwen2.5是阿里云最新推出的开源大型语言模型系列，相比Qwen2，Qwen2.5实现了整体性能的代际飞跃，大幅提升了代码、数学、推理、指令遵循、多语言理解等能力。包含5个尺寸的预训练和指令微调模型：Qwen2.5-0.5B、Qwen2.5-1.5B、Qwen2.5-7B、Qwen2.5-32B和Qwen2.5-72B。所有尺寸模型都使用了GQA（分组查询注意力）机制，以便让用户体验到GQA带来的推理加速和显存占用降低的优势。

在这里，我们使用 Qwen2.5-7B-Instruct 的模型参数版本。

* 模型优化工具：

ipex-llm（Intel® Extension for PyTorch Large Language Models）：是一个基于 PyTorch 的库，主旨在 Intel CPU 和 GPU 上高效运行大型语言模型（LLM）。该库特别适用于本地 PC 上的集成显卡（iGPU）和独立显卡（如 Arc、Flex 和 Max），能够以极低的延迟进行推理和微调。

IPEX-LLM 支持多种大语言模型，包括 HuggingFace transformers 模型，并且可以与社区中的其他框架和工具进行集成。此外，它还支持低比特优化技术（例如 INT4、NF4、FP4、INT5、INT8、FP8），这些技术有助于减少内存占用并提高计算效率

在这里我们对 Qwen2.5 模型进行低精度量化至 INT4/INT8 ，低精度量化（Low Precision Quantization）是指将浮点数转换为低位宽的整数（这里是INT4/INT8），以减少计算资源的需求和提高系统的效率。我们使用 save\_low\_bit api 将转换后的模型权重保存到指定文件夹。

这种技术在深度学习模型中尤其重要，它不仅可以在硬件上实现快速、低功耗的推理，也可以加快模型加载的速度。经过 Intel ipex-llm 优化后的大模型再从 ipex\_llm.transformers的AutoModelForCausalLM API加载， 我们可以很容易通过 load\_in\_low\_bit='sym\_int8' 来进行模型的加载和推理

* 数据集

《刑法》

《民法典》

1. **数据处理和分析的方法**

* 数据收集与预处理：

PDF 文件处理：使用成熟的RAG框架llamaindex来完成pdf文档的预处理和信息提取。通过灵活的配置，在后续的迭代中将会允许定制化网页和其他数据来扩大信息来源

* 分片和元数据处理：

一方面支持数据按照定制化的需求进行切片，也支持各种不同数据源的接入；

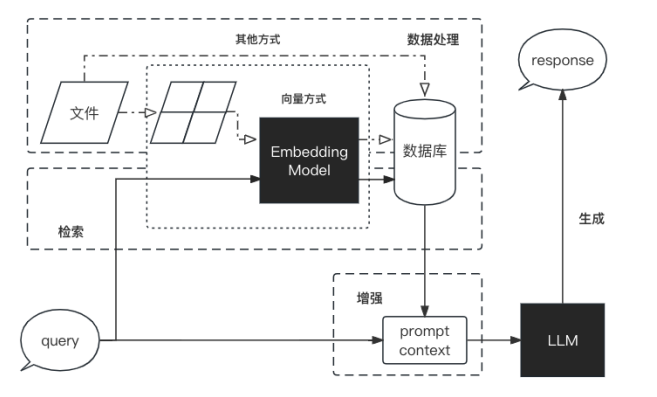
另一方面可以有效的借助llamaindex的能力来完成元数据的抽取工作；

* 数据增强与存储：

RAG（Retrieval-Augmented Generation）：采用 RAG 技术，将检索到的相关知识与生成模型结合起来，提高回答的准确性和相关性。

采用了Qdrant数据库进行向量数据库的存取和检索；Qdrant 专注于提供高性能的向量搜索和管理功能，特别是在需要低延迟、高吞吐量的应用中表现出色。它采用了 ANN（近似最近邻）搜索技术，并支持动态量化、数据分片、复制等高级功能，适合大规模分布式系统。

1. **系统架构和模块设计**



系统架构采用模块化和微服务（待实现）设计，主要包括以下核心模块：

* 数据处理模块：

信息预处理：对《民法典》和《刑法》文件进行切片和预处理，包含对metadata等元数据处理。

知识库管理模块：

向量数据库：使用 Qdrant 向量数据库存储处理后的信息 chunk。Qdrant 能够高效地进行向量检索，支持大规模数据存储和查询。

数据检索：对存储在 Qdrant 中的信息建立索引，支持通过向量相似度来完成快速检索和召回。

* 对话系统模块：

用户询问处理：接收用户的问题，并进行初步解析。

知识检索：基于用户的问题，从 Qdrant 向量数据库中检索相关知识。

Qwen-2 模型生成：将检索到的知识和用户问题一起输入到 Qwen-2 大语言模型中，按照提示词工程来生成自然语言回答。

回答及知识来源返回：系统不仅返回生成的回答，还提供相关知识的来源，增加回答的可信度。

* 用户界面模块：

Web 前端：采用现代化的前端框架Gradio构建用户界面，具备三个基本功能页面来和用户完成交互，提供友好的交互体验：首页（软件使用说明），主界面（法律咨询页面）和知识库页面（提供知识库文档查阅）

### 6. 使用到的英特尔软硬件技术

项目硬件使用阿里云提供的ecs.g6.2xlargeCPU资源 Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50GHz，核心数8；

安装库使用了英特尔 IPEX-LLM ，它是一个基于 PyTorch 的库，主要帮助用户在 Intel CPU 和 IntelGPU 上高效运行大型语言模型（LLM）

### 7. 成果说明（初赛可展示阶段性成果）

1. 软件介绍-首页



1. 软件主界面-咨询页面

用户：张三嫖娼用冥币，这怎么算

小助手回复：根据上述背景信息中的《中华人民共和国刑法》第三百零三条关于赌博罪的规定，赌博罪是指以营利为目的，聚众赌博或者以赌博为业的行为。然而，关于张三使用冥币嫖娼的情况，并未直接涉及到赌博罪。赌博罪通常与现金交易有关，而非冥币。冥币在中国传统文化中主要用于祭祀或寄托亡灵，一般不作为现实经济交易中的支付手段。因此，单纯使用冥币嫖娼的行为并不构成赌博罪。

不过，如果在这一过程中涉及到了其他违法行为，例如通过赌博或非法手段获取冥币用于嫖娼，那么可能需要根据具体情况来分析是否触犯了其他相关法律法规，如洗钱罪、诈骗罪、或者与性交易相关的罪名。例如：

* **洗钱罪**：如果冥币是通过洗钱活动（如毒品犯罪、黑社会犯罪、恐怖活动等）获得的，那么张三在嫖娼过程中的行为可能涉嫌洗钱罪。
* **诈骗罪**：如果冥币是通过欺骗手段（如虚假宣传、冒充合法来源等）获得的，那么张三在嫖娼过程中的行为可能涉嫌诈骗罪。
* **组织卖淫罪**：如果嫖娼行为涉及的卖淫活动是由组织者有意识地操作的，并且在其中起着关键作用，例如招募、运送人员或提供场所等，那么张三的行为可能被视作协助组织卖淫罪的一部分。
* **引诱、容留、介绍卖淫罪**：如果张三在嫖娼过程中引诱他人卖淫、容留卖淫活动或介绍他人从事卖淫活动，那么其行为可能构成引诱、容留、介绍卖淫罪。
* **传播性病罪**：如果卖淫活动涉及梅毒、淋病等性病的传播，而卖淫者明知自己患有此类疾病仍从事卖淫活动，则张三在嫖娼过程中的行为可能间接涉及传播性病罪的责任，尤其是在明知卖淫者有性病的情况下仍然进行性交易。

总之，单纯使用冥币嫖娼的行为并不直接构成赌博罪，但具体行为是否违法还需要结合实际情况，尤其是与性交易、金钱交换、以及冥币的来源合法性等因素综合考量。如果有更多具体的背景信息，可以进一步分析是否涉及到上述或其他相关罪名。

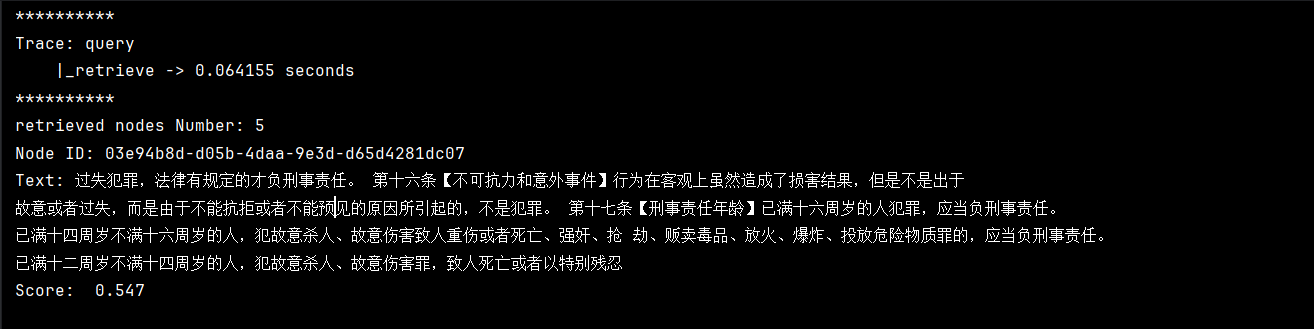




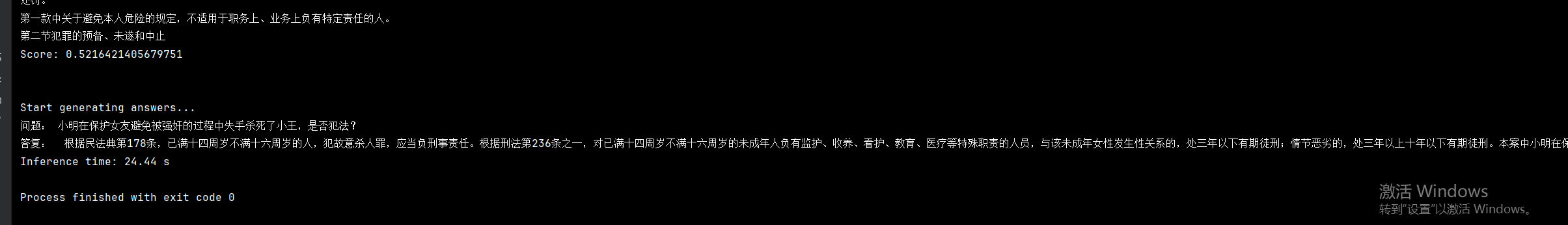
1. 软件主界面-知识库界面



1. 性能测试-RAG（性能测试在intel的i5-12490F的CPU进行的测试，模型是bge-small-zh-v1.5）-0.07秒



1. 性能测试-推理（性能测试在intel的i5-12490F的CPU进行的测试，1.5B模型精度：int8）-24秒



* 当前不足以及未来的改善、推广的展望

经过多次测试最终采用了bge-small-zh-v1.5作为emb模型，qwen2.5-1.5B-Instruct的int8量化作为llm模型能够在Intel CPU推理上有较好表现和性能，其中reranker模型-bge-base-reranker最终没有加入（会根据检索的节点增加对应的检索时间，20个节点重排增加时间在5秒左右），而且因为模型太小的原因，加入后会有奇怪的输出，不太符合预期；整体上响应时间在20-30秒之间，属于比较实用的范围。（如果采用3B和7B，测试时间在150，300秒规模）

当前应用还存在一定的检索和召回质量不高、回复的内容不够智能的问题；

下一步考虑：

1. 性能提升：考虑采用能采用Intel GPU运行高精度的大语言模型版本，加快推理速度
2. RAG提升：
   1. 采用Query Transformation中的高级策略比如Hyde（生成假设性问题）来提升召回信息的质量；
   2. 采用Response Synthesizer来对答案重新组织回答，通过对召回结果进一步信息蒸馏的方式来完成回复和答案的质量提升；
   3. 也可以考虑集成Intel的GraphRAG通过知识图谱的方式给与用户更好的检索结果和答复效果
3. 使用体验：
   1. 考虑在原文引用上增强直接链接到原文的功能；
   2. 考虑包含bing 搜索引擎来列举相关的案例和以及相关的法律信息

### 9. 支持材料（可选）

* Github 链接：https://github.com/phoenixdna/intelllm
* PPT 附件：暂无
* 演示视频： 暂无