

SecondMe

Train Your AI Self
Amplify You, Bridge the World



Second Me

1. 功能介绍

1. 是用来做什么的？

文档QA助手 - 个性化

2. 功能上有什么特色？

- 极度个性化（用户画像，DPO训练）
- 方便的本地部署

3. 技术架构上有什么特色？

- 个性化的处理
- 三层知识架构
- 自动化的training(DPO)

2. 代码结构

2.1 代码结构

- lpm_kernel/：后端或核心算法相关的代码目录。
- lpm_frontend/：前端相关代码目录。
- mcp/：支持mcp协议
- integrate：与微信机器人进行集成

2.2 后端代码

代码块

```
1  lpm_kernel/
2  |—— app.py           # 项目主入口, Flask 应用初始化、路由注册、CORS、文件服务等
3  |—— api/             # API 路由、服务、DTO、仓库等, RESTful 接口实现
4  |—— common/          # 通用工具、日志、LLM 接口、策略、数据库会话等
5  |—— configs/          # 配置管理、日志配置
6  |—— database/         # 数据库迁移、管理、migrations 脚本
7  |—— file_data/        # 文档处理、分块、嵌入、文档服务、DTO、异常等
8  |—— kernel/          # 知识分层核心逻辑 (L0/L1), 分块、聚类、笔记等
9  |—— models/          # 数据模型 (如 L1、内存、空间、状态传记等)
10 |—— L0/               # L0 层生成器、模型、prompt
11 |—— L1/               # L1 层生成器、聚类、传记、主题、特征等
12 |—— L2/               # L2 层训练、推理、数据管道、DPO、脚本等
13 |—— utils.py          # 通用工具函数
14 |—— tokenizer.json    # 分词器配置
```

3. 功能特点

3.1 知识分层

3.1.1 知识分层架构

- **L0 层**: 原始文档分块、嵌入生成、insight/summary 自动生成 (支持多模态)。
- **L1 层**: 基于 L0 的高层结构化知识自动生成, 如人物传记、主题聚类、多视角特征、状态传记等。
- **L2 层**: 个性化模型训练、推理、DPO 优化、长链式推理 (CoT)、分布式/量化训练。

3.1.2 自动整理的具体流程

3.1.2.1 文档自动分析与分块 (L0)

- 用户上传文档后, 系统会自动扫描、分块 (chunk), 并为每个分块生成嵌入 (embedding)。
- 自动调用分析方法, 生成每个文档的主题、关键词、摘要 (insight/summary), 并写入数据库。
- 相关API: `POST /api/documents/analyze`, 会自动处理所有未分析的文档。

3.1.2.2 知识聚合与结构化 (L1)

- 基于L0分块和嵌入, 系统自动进行主题聚类、人物传记生成、多视角特征归纳等。

- 通过 `/api/kernel/l1/global/generate` 接口，一键生成全局传记、聚类、分块主题等结构化知识。
- 例如：自动归纳"张三是一位AI专家，专注于NLP和知识图谱"，并聚类出"自然语言处理""知识图谱"等主题。

3.1.2.3 知识增强与检索

- 在用户提问时，系统会根据请求的metadata自动检索相关知识（如L0分块、L1聚类、传记等），拼接进Prompt，提升大模型的上下文理解和个性化能力。
- 检索方式包括向量相似度、关键词匹配等，相关代码见 `KnowledgeEnhancedStrategy`。
- 检索到的知识会以 `system` 角色插入到大模型的上下文中。

3.2 个性化

`/api/kernel/l1/global/generate`

代码块

```
1
2      {
3          "code": 0,
4          "message": "success",
5          "data": {
6              "bio": {
7                  "content": "张三是一位AI专家，专注于自然语言处理和知识图谱研究。",
8                  "content_third_view": "在同事眼中，张三是技术大牛，乐于分享。",
9                  "summary": "AI专家，NLP方向，知识图谱。",
10                 "summary_third_view": "团队核心成员，推动多项创新。",
11                 "create_time": "2024-06-01T12:34:56",
12                 "shades": [
13                     {
14                         "name": "创新能力",
15                         "aspect": "技术",
16                         "icon": "💡",
17                         "desc_third_view": "他善于提出新想法",
18                         "content_third_view": "他多次主导创新项目",
19                         "desc_second_view": "你喜欢技术创新",
20                         "content_second_view": "推动团队技术进步"
21                     }
22                 ]
23             },
24             "clusters": {
25                 "clusterList": [
```

```

26         {
27             "clusterId": "1",
28             "memory_ids": ["101", "102"],
29             "topic": "自然语言处理",
30             "cluster_center": { "embedding": [0.1, 0.2] }
31         },
32         {
33             "clusterId": "2",
34             "memory_ids": ["103"],
35             "topic": "知识图谱",
36             "cluster_center": { "embedding": [0.3, 0.4] }
37         }
38     ],
39 },
40 "chunk_topics": {
41     "chunk_101": {
42         "topic": "NLP",
43         "tags": ["文本处理", "分词"]
44     },
45     "chunk_102": {
46         "topic": "实体识别",
47         "tags": ["NER", "信息抽取"]
48     }
49 },
50 "generate_time": "2024-06-01T12:34:56"
51 }
52 }

```

3.2.1.1 字段说明:

- **bio**：全局人物传记/知识总结（多视角）
 - **content**：主视角传记内容
 - **content_third_view**：第三视角描述
 - **summary**：主视角摘要
 - **summary_third_view**：第三视角摘要
 - **create_time**：生成时间
 - **shades**：多视角特征列表
- **clusters**：主题聚类结果
 - **clusterList**：聚类列表，每个聚类包含 **clusterId**、**memory_ids**、**topic**、**cluster_center**
- **chunk_topics**：每个分块的主题标签

- 以分块ID为 key，每个 value 包含 `topic`、`tags`
- `generate_time`：本次 L1 生成的时间戳

3.3 自动化训练流程

- 全流程自动化：从数据生成到模型训练、保存、评估，全部可脚本化、一键运行。
- 支持多种评测方式，评测标准明确，结果可追溯、可分析。

3.3.1 数据自动生成

- 通过 `PreferenceQAGenerator` 等脚本，从用户知识库、Bio等信息自动生成高质量问答对（支持COT推理链）。
- 可扩展为生成DPO所需的正负样本（chosen/rejected）。

3.3.2 数据组织与保存

- 生成的数据以标准格式（如json）保存，便于后续训练脚本直接读取。

3.3.2.2 训练数据标准示例

训练数据以json格式保存，每条数据包含 `prompt`、`chosen`、`rejected` 字段。例如：

代码块

```
1  [
2    {
3      "prompt": "请简要介绍一下人工智能的发展历程。",
4      "chosen": "人工智能的发展经历了符号主义、连接主义和统计学习等阶段。近年来，深度学习推动了AI在语音、图像和自然语言处理等领域的突破。",
5      "rejected": "人工智能就是让机器像人一样思考，现在很流行。"
6    },
7    {
8      "prompt": "如何高效管理个人时间？",
9      "chosen": "高效管理时间可以采用四象限法则，优先处理重要且紧急的任务，合理规划每日计划，并定期复盘。",
10     "rejected": "多做事就能管理好时间。"
11   }
12 ]
```

3.3.3 训练调度与参数配置

- 通过Shell/Python脚本自动设置训练参数、环境变量、硬件资源等。
- 支持分布式、量化、LoRA等工程优化。

示例Shell脚本：

代码块

```
1 python lpm_kernel/L2/dpo/dpo_train.py \  
2     --training_data_path resources/L2/data/dpo/dpo_direct.json \  
3     --base_model_path resources/model/output/merged_model \  
4     --num_train_epochs 3 \  
5     --batch_size 2 \  
6     --learning_rate 5e-6 \  
7     --beta 0.1
```

https://huggingface.co/docs/trl/main/en/dpo_trainer

3.3.4 DPO训练核心代码

- 使用 `DPOTrainer` 自动完成偏好优化训练。

详见 `git`上的分析

3.3.5 自动化与工程优化

- 支持自动检测硬件、自动分配`device_map`、量化、LoRA等。
- 支持分布式训练（如DeepSpeed/FSDP），Shell脚本可自动适配多卡环境。

3.4 模型评估流程

3.4.1 训练过程中的自动评估

- 训练过程中自动记录loss（损失），可用于基本收敛性判断。
- 如配置验证集，也可自动评估验证集loss。

3.4.2 自动化对比式评测（核心）

- 通过 `DPOData.compare_eval` 方法，自动对比多组模型回答，统计胜率，输出详细分析。
- 支持用大模型API或人工评测。

3.4.3 评测标准

- 评测prompt中明确要求评判准确性、相关性、个性化、推理过程等。
- 评测结果分为"first win/second win/tie"，并输出详细分析。

4. 参考

<https://github.com/mindverse/Second-Me>

<https://secondme.gitbook.io/secondme/tutorial#introduction>

<https://www.youtube.com/watch?v=TogHaqdExvc>

https://huggingface.co/docs/trl/main/en/dpo_trainer