

高级热词统计与分析系统使用说明书

本文档旨在指导用户如何编译、运行及使用本系统。系统支持**一键脚本启动**和**单独运行C++后端**两种方式。

1. 项目目录结构

在运行前，请确保您的项目目录结构如下所示：

```
Project/
├── data/           # 存放默认测试数据（如 input1.txt）
├── dict/           # 存放分词词典（jieba.dict.utf8 等）
├── includefile/    # C++ 头文件（config.h 等）
├── sources/        # C++ 源文件，Python文件以及核心单元测试
├── packages/       # 离线 Python 依赖包（用于离线安装依赖）
├── .streamlit/     # 对streamlit库添加配置
├── temp/           # 用于存储前端传给后端的数据临时文件
├── test/           # 用于性能测试创建的数据文件和输出文件
├── xmake.lua       # xmake 构建配置
├── requirements.txt # Python 依赖清单
├── run.bat         # Windows 一键启动脚本
├── run.sh          # Linux/macOS 一键启动脚本
├── website.png     # README中引用的图片
├── System_Architecture.png #展示整个系统架构
├── README.md       # 说明文档
├── README.pdf      # 便于快速阅读说明文档
├── 性能测试报告.pdf
└── 系统设计文档.pdf
```

确认无误后进入Project目录。本项目内使用的所有路径都是**相对路径**，但为了避免可能发生的错误，**请确保当前目录路径无中文**。

2. 环境准备

- **C++环境**: 确保安装了 **xmake** 和 C++ 编译器 (GCC/Clang/MSVC)。执行以下指令安装**xmake**

```
pip install xmake
```

- **Python环境**: 要求**Python > 3.9**，建议**Python == 3.11**。所需要的相关依赖可以运行**run.sh**或**run.bat**脚本一键配置，具体见方法一。

3. 方式一：一键脚本启动

该方式会自动配置 Python 虚拟环境、安装依赖、编译 C++ 后端并启动前端页面，适合演示和快速使用。

Windows 用户

1. 进入项目Project目录。**最好无中文路径。**
2. 双击运行 **run.bat**，或在 PowerShell/CMD 中执行：

```
.\run.bat
```

Linux / macOS 用户

1. 打开终端，进入项目Project目录。**最好无中文路径。**
2. 运行 Shell 脚本：

```
bash run.sh
```

脚本自动执行流程：

1. 检查 Python 环境。
2. 创建并激活虚拟环境 (venv)。
3. 安装依赖。
4. 调用 **xmake** 编译 C++ 后端（如果没有 xmake，Linux 脚本会尝试用 g++ 兜底）。
5. 自动启动浏览器打开 Streamlit 前端界面。

packages/中提供了支持**Python=3.11**的依赖离线安装包，预计安装时间为1分钟；如果**Python**版本为其他版本，则运行脚本自动通过清华镜像源下载安装对应版本的依赖，预计时长为2分钟。执行完成后会自动跳转Web网页，或显示如下：

```
You can now view your Streamlit app in your browser.
```

```
Local URL: http://localhost:8501
```

```
Network URL: http://172.19.46.239:8501
```

```
External URL: http://103.62.49.138:8501
```

如过没有自动跳转，点击上述链接打开即可。

4. 方式二：单独运行C++后端

方式一启动包括运行C++后端和Python的streamlit前端，其中Python部分运行时间可能较长。该方式旨在便于检查C++后端的运行情况，不运行前端。**以下操作在可以在Windows/Linux/macOS下适用。**

4.1 编译 C++ 后端

在项目根目录下执行：

```
xmake
```

编译成功后，可执行文件（`main.out.exe` 或 `main.out`）将生成在当前根目录下。

4.2 运行 C++ 后端

您可以通过 `xmake run` 直接带参数运行后端程序：

```
# 基本用法
xmake run main.out -i data/input1.txt -o my_output.txt -s 120

# 完整参数示例
# -i: 输入文件
# -o: 输出文件
# -s: 滑动窗口步长
# -m: 分词模式 (Cut(HMM), Cut(NoHMM), CutForSearch)
# -u: 用户自定义词典路径
# -w: 额外停用词路径
# -k: UDP发送的最大Top-K数量
xmake run main -i data/input1.txt -o result.txt -s 60 -m CutForSearch -k
100
```

以上输入输出文件也可以直接写文件名称，程序会自动识别是**文件名或文件路径**，如果是文件名则默认输入输出文件都在data/文件夹中。

注意：单独运行后端时，它会将结果写入文件并通过 UDP 发送数据。如果没有前端接收 UDP 数据，后端依然会正常运行并生成输出文件。

5. 前端界面功能指南

启动成功后，浏览器将打开 Web 界面，包含以下功能模块：

5.1 侧边栏配置

- **数据源：**点击 `Browse files` 上传您的 `input.txt` 数据文件。
- **核心参数：**
 - **滑动步长：**控制时间窗口更新的频率。
 - **前端最大 Top-K：**设置滑块可调节的最大范围（例如设为 100，即后端会通过 UDP 发送前 100 个热词）。
 - **当前显示 Top-K：**拖动滑块，实时调整图表中显示的柱状图数量。
- **分词模式：**选择 `HMM`（新词发现）、`NoHMM`（快速）、`CutForSearch`（搜索引擎模式）。
- **自定义词典配置 (折叠菜单)：**
 - **用户专用词：**输入您希望保留的特定词汇（如“黑神话”），支持多行。
 - **自定义停用词：**输入您希望过滤的敏感词或无意义词，支持多行。
- **输出文件名：**指定结果保存的文件名。

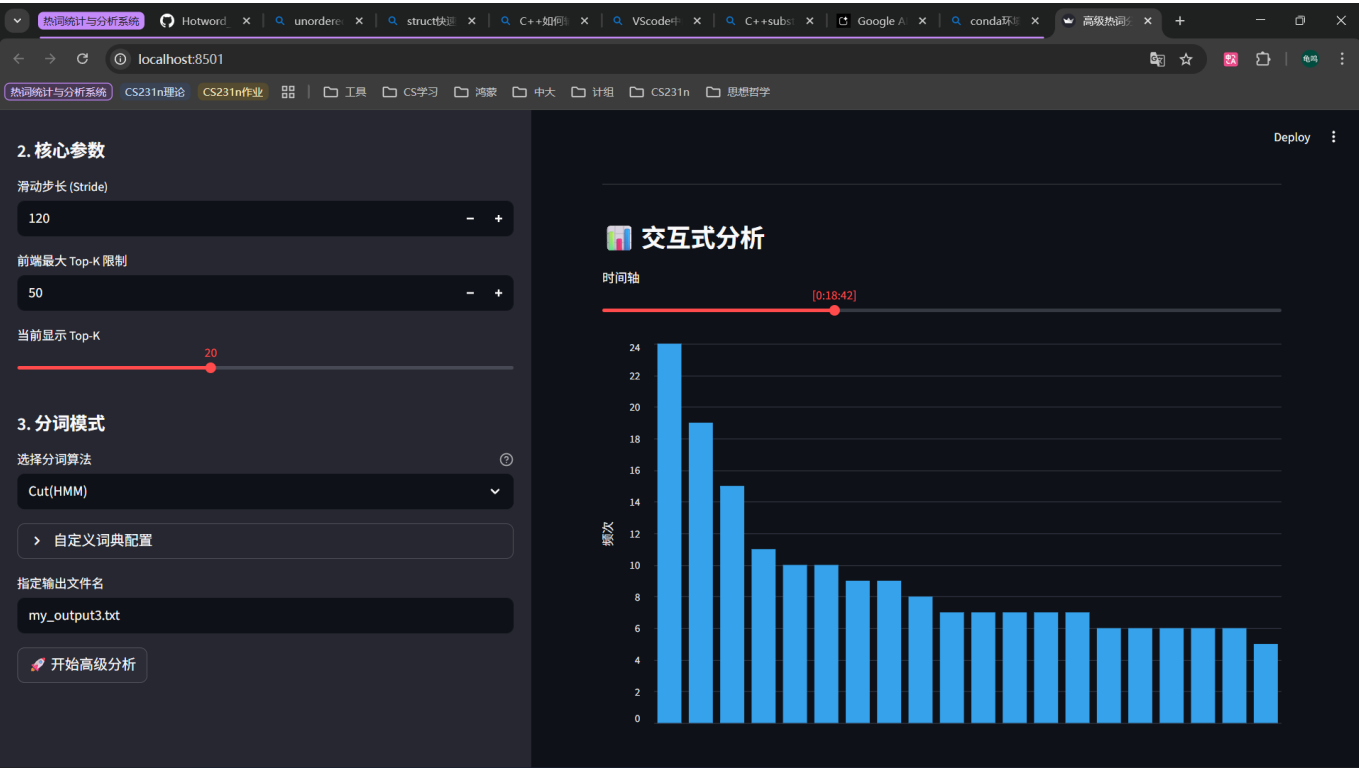
5.2 交互式分析 (主界面上方)

- **时间轴滑块**：当分析完成后，拖动滑块可以回溯任意时间点的热词状态。
- **动态图表**：展示当前时间点的 Top-K 热词柱状图（按频次从左到右递减，鼠标悬停可查看详情）。
- **数据列表和csv下载**：可以将动态图表切换为数据列表，显示出当前topk词和数量(在时间轴的右下方有切换键)。也可以选择下载csv表格。

5.3 结果下载 (主界面下方)

- **下载按钮**：点击即可下载由 C++ 后端生成的完整 `output.txt` 文件。

注意：这里的output.txt文件中的topK是由input.txt文件中的[ACTION] QUERY=K决定的，即只统计了这些指令所指示的时间戳的topK，和大作业文件中要求的一致。



⚠ 6. 注意事项与常见问题

1. 文件编码

- 所有输入文件 (`input.txt`) 和字典文件建议使用 **UTF-8 (无 BOM)** 编码，否则中文可能会乱码。

2. 端口冲突

- 系统使用 **UDP 9999** 端口进行通信。如果启动失败并提示端口被占用，请关闭占用该端口的程序或稍后重试。

3. 临时文件

- 系统运行时会在根目录下生成 `temp/` 文件夹，用于存放前端传递给后端的临时字典。**请勿在运行过程中删除此文件夹。**

4. Windows 权限

- 如果 Windows 提示“已保护您的电脑”，请点击“更多信息 -> 仍要运行”。
- 脚本已优化路径处理，**无需**使用管理员权限运行。

5. C++ 编译失败

- 如果脚本提示 `xmake not found` 且没有安装 C++ 编译器，请先安装 `xmake` (推荐) 或 `MinGW/Visual Studio`。