

# 基于多模态大语言模型的智能全平台稍后阅读系统的设计与实现

李根

华中科技大学  
计算机科学与技术学院  
计科 2109 班

2025 年 5 月 14 日

# ToRead 体验地址

# <https://tr.lg.gl>

一行命令上手

```
curl https://g.lg.gl --data '{"url":"https://lg.gl"}'
```

# 目录

- 1 研究背景与目标
- 2 ToRead 系统设计
- 3 ToRead 核心功能实现
- 4 总结与展望

# 目录

- 1 研究背景与目标
- 2 ToRead 系统设计
- 3 ToRead 核心功能实现
- 4 总结与展望

## ● 信息碎片化问题

- 互联网信息爆炸,用户难以高效筛选、分类和管理信息
- 碎片化阅读习惯普及,需要高效的内容收藏与管理工具

## ● AI 技术发展

- 多模态大语言模型在内容理解与知识提取方面能力显著提升
- Function Calling 功能使开发者能够方便地定制个性化 AI 助手

## ● 现有稍后阅读工具局限性

- 现有工具如 Pocket、Instapaper 等在智能化分类与推荐方面存在不足
- 缺乏对多模态内容(文本、图片、视频)的统一处理能力

# 研究目标

- **设计并实现智能稍后阅读系统**
  - 帮助用户在信息过载环境中高效管理碎片化信息
  - 支持多种终端（Web 端、浏览器插件、移动端）
- **集成多模态大语言模型能力**
  - 实现对文本、图片等多种内容形式的智能处理
  - 自动生成内容摘要、提取关键信息、标签分类
- **构建全平台统一用户体验**
  - 实现跨平台数据同步与一致的用户界面
  - 提供简洁高效的内容收藏与管理功能

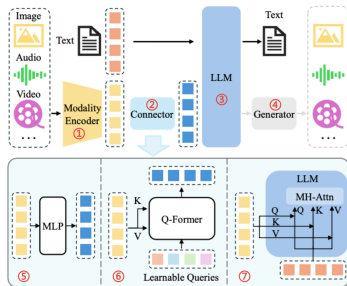


图 多模态大模型架构

基于多模态大语言模型的智能全平台稍后阅读系统



**ToRead**

# 目录

1 研究背景与目标

2 ToRead 系统设计

3 ToRead 核心功能实现

4 总结与展望



# ToRead 系统架构设计

- **前后端分离架构**: 后端提供 RESTful API, 前端通过 API 获取数据
- **多端协同**: 浏览器插件、全平台前端均共享统一后端服务
- **AI 智能总结**: 集成多模态大语言模型 API 处理模块, 实现内容智能分析

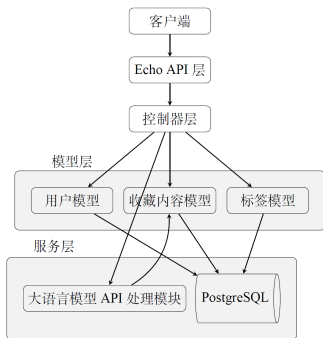


图 ToRead 后端架构图

# 后端 API 接口设计

## 用户管理 API

- /user/register - 用户注册
- /user/login - 用户登录
- /user - 获取用户信息

## 收藏管理 API

- /collection/add - 添加收藏
- /collection/list - 获取收藏列表

- 统一的响应格式: {code, msg, data}
- RESTful 设计风格, 便于前端调用
- 完善的错误处理与状态码

## 标签管理 API

- /collection/tag - 获取标签列表

## 内容分析 API

- /collection/summary - 获取内容摘要

后端 API Endpoint: <https://to-read.lg.gl/api/v1>, 接口文档位于仓库 `/backend/docs/api.md` 处。

## 用户模型 (User)

- ID、用户名、密码 (MD5 加密)
- 角色权限、创建/更新时间

## 标签模型 (Tag)

- ID、标签名称
- 创建/更新时间

## 收藏内容模型 (Collection)

- ID、用户 ID、URL、内容类型
- 标题、描述、创建/更新时间

## 收藏-标签关联 (CollectionWithTag)

- 收藏 ID、标签 ID
- 创建/更新时间

- 采用 GORM 框架实现 ORM 映射, 简化数据库操作
- 使用**软删除**机制, 便于数据恢复与审计
- 建立合理的索引, 提高**全文搜索**的查询效率

# 目录

- 1 研究背景与目标
- 2 ToRead 系统设计
- 3 ToRead 核心功能实现**
- 4 总结与展望

## 后端技术

- Golang + Echo 框架
  - **高性能**、低资源占用
  - 简洁的 API 设计与中间件支持
- PostgreSQL 数据库
  - 支持**全文搜索**与复杂查询
  - 高可靠性与数据完整性
- JWT 认证机制
  - **无状态**认证,减轻服务器负担
  - 跨域支持,适合多端应用

## 前端技术

- Electron + React
  - **跨平台**桌面应用支持
  - 组件化开发,提高复用性
- Material UI
  - 美观一致的用户界面
  - **响应式设计**,适配多种设备

## 浏览器插件

- Chrome Extension API
  - 极简化设计
  - **一键**收藏网页功能

# 多模态内容处理模块

- 1 **内容类型识别**: 根据 URL 特征判断内容类型(文本、图片、视频)
- 2 **内容获取**: 通过 HTTP 请求集成 Jina AI 获取原始内容
- 3 **AI 处理**: 调用大语言模型 API 进行内容分析
- 4 **结果整合**: 生成标题、描述、标签等结构化信息

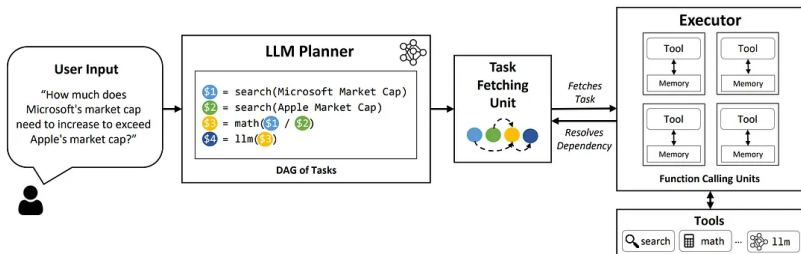


图 大语言模型处理

# 大语言模型 API 处理模块

## ● Function Calling 机制

- 利用大模型厂商 API 的 Function Calling 功能
- 将 Collection 结构体作为参数传给大语言模型
- 解析返回的 JSON 格式数据,提取标题、描述与标签

## ● 多模态内容处理

- 文本内容: 直接提交给 AI 模型分析
- 图片内容: 先生成文本描述, 再进行处理

## ● 模型选择

- 主要使用 OpenAI 的 gpt-4o 模型
- 在生成质量、速度、价格方面较为平衡
- 支持处理文本与图片内容

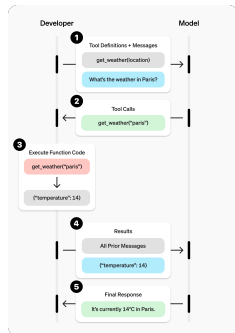


图 函数调用流程

# 跨平台前端实现

## Electron 跨平台前端

- 认证模块
  - 基于 JWT 的身份验证
  - 登录/注册统一界面
- 内容展示模块
  - 响应式卡片布局
  - 标签筛选与全文搜索
- 内容摘要组件
  - 展示 AI 生成的内容摘要
  - 动态加载与缓存机制



图 前端界面



# 浏览器插件实现

## Chromium 浏览器插件

- 轻量级设计
  - 临时交互界面
  - 简洁直观的交互元素
- 一键收藏功能
  - 捕获当前标签页 URL
  - 后台异步处理请求
- 认证机制
  - 与主系统共享 JWT 认证
  - 本地存储认证信息

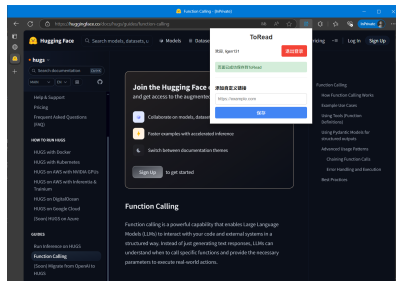


图 浏览器插件界面

# 目录

- 1 研究背景与目标
- 2 ToRead 系统设计
- 3 ToRead 核心功能实现
- 4 总结与展望

## ● 主要成果

- 设计并实现了基于微服务架构的 ToRead 系统
- 构建了完整的用户认证与内容管理模块
- 实现了基于大语言模型的内容智能分析功能
- 开发了跨平台协同机制,实现多端数据同步

## ● 未来工作

- 扩展原生 Android 移动端应用支持
- 增强 AI 分析能力,实现个性化推荐
- 引入内容智能化推荐算法
- 优化离线支持与同步机制
- 探索内容协作与社交功能



图 全平台同步

# 谢谢!

- ToRead DEMO 体验:<https://tr.lg.gl>
- 开源项目地址:<https://github.com/ligen131/ToRead>
- 一行命令将 ToRead 集成到你的应用中(url 处只是示例,可以传入任意链接):  
`curl https://g.lg.gl --data '{"url":"https://lg.gl"}'`