

# YouTube到B站智能体 - 使用文档

## 项目简介

YouTube到B站智能体是一个自动化工具，能够实现从YouTube下载视频、生成双语字幕并上传到哔哩哔哩的完整工作流程。该智能体集成了多项先进技术，包括视频下载、AI语音识别、机器翻译和自动化上传等功能。

## 功能特性

- YouTube视频下载**: 支持下载高质量的YouTube视频
- 音频提取**: 从视频中提取音频文件用于字幕生成
- AI字幕生成**: 使用OpenAI Whisper模型自动生成准确的字幕
- 智能翻译**: 将原始字幕翻译为中文
- 双语字幕合并**: 创建包含原文和中文的双语字幕文件
- 自动上传B站**: 自动将处理好的视频上传到哔哩哔哩平台
- Web界面**: 提供友好的用户界面，支持实时进度监控

## 技术架构

### 后端技术栈

- Python 3.11**: 主要编程语言
- Flask**: Web API框架
- yt-dlp**: YouTube视频下载工具
- OpenAI Whisper**: AI语音识别模型
- Google Cloud Translate**: 机器翻译服务
- Selenium**: 浏览器自动化工具

- **FFmpeg**: 音视频处理工具

## 前端技术栈

- **React 18**: 前端框架
- **Vite**: 构建工具
- **Tailwind CSS**: 样式框架
- **shadcn/ui**: UI组件库
- **Lucide React**: 图标库

## 系统要求

---

### 硬件要求

- CPU: 至少2核心
- 内存: 至少4GB RAM
- 存储: 至少10GB可用空间
- 网络: 稳定的互联网连接

### 软件要求

- 操作系统: Windows 10/11
- Python 3.9+ (推荐使用Python 3.11)
- Node.js 18+ (推荐使用Node.js 20)
- Chrome浏览器 (用于B站上传)
- FFmpeg
- yt-dlp

# 安装指南

---

## 1. 环境准备

### Python

访问 [Python官网](#) 下载并安装最新版本的Python 3.9+。安装时请勾选 “Add Python to PATH” 选项。

### Node.js

访问 [Node.js官网](#) 下载并安装最新版本的Node.js 18+。安装时请选择推荐的LTS版本。

### FFmpeg

1. 访问 [FFmpeg官网](#) 下载Windows版本的FFmpeg。
2. 解压下载的文件到一个目录（例如 `C:\ffmpeg`）。
3. 将 `C:\ffmpeg\bin` 添加到系统环境变量 `Path` 中。

### yt-dlp

1. 访问 [yt-dlp GitHub Releases](#) 下载 `yt-dlp.exe`。
2. 将 `yt-dlp.exe` 放到一个系统Path目录中（例如 `C:\Windows` 或 `C:\Users\YourUser\AppData\Local\Microsoft\WindowsApps`）。

### Google Chrome 和 ChromeDriver

确保您的系统安装了Google Chrome浏览器。Selenium需要对应的ChromeDriver才能正常工作。

1. 检查您Chrome浏览器的版本（在Chrome浏览器中输入 `chrome://version`）。
2. 访问 [ChromeDriver官网](#) 下载与您Chrome浏览器版本对应的ChromeDriver。
3. 将下载的 `chromedriver.exe` 放到一个系统Path目录中（例如 `C:\Windows`）。

## 2. 克隆项目

如果您还没有安装Git，请访问 [Git官网](#) 下载并安装。

```
git clone <项目仓库地址> # 请替换为实际的项目仓库地址
cd youtube-bilibili-agent
```

**注意：**在Windows的命令提示符（CMD）或PowerShell中，`cd` 命令的使用方式与Linux类似。

### 3. 安装Python依赖

```
pip install -r requirements.txt
```

主要依赖包括： - flask - flask-cors - openai-whisper - selenium - pysrt - google-cloud-translate

### 4. 安装前端依赖

```
cd youtube-bilibili-agent  
npm install
```

### 5. 配置Google Cloud Translate

为了使用翻译功能，需要配置Google Cloud Translate API：

1. 在Google Cloud Console创建项目
2. 启用Translate API
3. 创建服务账号并下载密钥文件（例如 `path\to\your\credentials.json`）
4. 设置环境变量：

<b>Windows</b>	<b>CMD:</b>	<code>cmd</code>	<code>set</code>
<code>GOOGLE_APPLICATION_CREDENTIALS="path\to\your\credentials.json"</code>			

<b>Windows</b>	<b>PowerShell:</b>	<code>powershell</code>
<code>\$env:GOOGLE_APPLICATION_CREDENTIALS="path\to\your\credentials.json"</code>		

请将 `path\to\your\credentials.json` 替换为您实际的密钥文件路径。

## 使用方法

---

### 1. 启动后端服务

打开命令提示符（CMD）或PowerShell，进入项目根目录，然后运行：

```
python flask_api.py
```

后端服务将在 `http://localhost:5000` 启动。

## 2. 启动前端服务

```
cd youtube-bilibili-agent  
npm run dev
```

前端服务将在 `http://localhost:5173` 启动。

## 3. 使用Web界面

1. 打开浏览器访问 `http://localhost:5173`
2. 输入YouTube视频链接
3. 填写B站视频信息（标题、描述、标签）
4. 点击"开始处理"按钮
5. 等待处理完成

## 4. 命令行使用

也可以直接使用命令行工具：

```
python main_agent.py "https://www.youtube.com/watch?v=VIDEO_ID"
```

# 工作流程详解

---

### 第一步：视频下载

系统使用yt-dlp工具从YouTube下载指定的视频文件，支持多种格式和质量选择。

### 第二步：音频提取

使用FFmpeg从下载的视频中提取音频文件，转换为适合语音识别的格式。

## 第三步：字幕生成

利用OpenAI Whisper模型对音频进行语音识别，生成准确的原始语言字幕。

## 第四步：字幕翻译

通过Google Cloud Translate API将原始字幕翻译为中文。

## 第五步：双语字幕合并

将原始字幕和翻译字幕合并为双语字幕文件，方便观众理解。

## 第六步：B站上传

使用Selenium自动化工具将处理好的视频和字幕上传到哔哩哔哩平台。

# 配置说明

---

## 视频质量设置

可以在 `youtube_downloader.py` 中修改下载质量：

```
command = [  
    "yt-dlp",  
    "-f", "bestvideo[ext=mp4]+bestaudio[ext=m4a]/best[ext=mp4]/best",  
    "--merge-output-format", "mp4",  
    "-o", os.path.join(output_path, "%(title)s.%(ext)s"),  
    url  
]
```

## Whisper模型选择

在 `subtitle_generator.py` 中可以选择不同的Whisper模型：

- `tiny`：最快，准确度较低
- `base`：平衡速度和准确度
- `small`：较好的准确度
- `medium`：更好的准确度

- `large`：最高准确度，速度较慢

## 翻译目标语言

在 `subtitle_translator.py` 中可以修改目标语言：

```
result = translate_client.translate(text, target_language="zh-CN")
```

支持的语言代码：- `zh-CN`：简体中文 - `zh-TW`：繁体中文 - `ja`：日语 - `ko`：韩语 - 等等

## 故障排除

---

### 常见问题

#### 1. YouTube视频下载失败

- 检查网络连接
- 确认视频链接有效
- 检查yt-dlp是否为最新版本

#### 2. 字幕生成失败

- 确认音频文件完整
- 检查Whisper模型是否正确安装
- 确保有足够的内存和存储空间

#### 3. 翻译服务失败

- 检查Google Cloud Translate API配置
- 确认API密钥有效
- 检查网络连接

#### 4. B站上传失败

- 确认已登录B站账号
- 检查视频格式和大小限制

- 确认Chrome浏览器正常运行

## 日志查看

系统会在控制台输出详细的处理日志，可以通过日志信息定位问题：

```
# 查看后端日志
python flask_api.py

# 查看前端日志
npm run dev
```

## 性能优化

### 1. 硬件优化

- 使用SSD存储提高I/O性能
- 增加内存以支持更大的Whisper模型
- 使用GPU加速Whisper推理（如果可用）

### 2. 软件优化

- 选择合适的Whisper模型大小
- 调整视频下载质量
- 使用并行处理（如果处理多个视频）

## API文档

---

### 后端API接口

#### 1. 健康检查

```
GET /api/health
```

响应：



```
{  
  "status": "healthy"  
}
```

## 2. 开始处理

POST /api/process

请求体:

```
{  
  "youtube_url": "https://www.youtube.com/watch?v=...",  
  "video_title": "视频标题",  
  "video_description": "视频描述",  
  "video_tags": "标签1, 标签2, 标签3"  
}
```

响应:

```
{  
  "message": "视频处理已开始"  
}
```

## 3. 获取状态

GET /api/status

响应:

```
{  
  "is_processing": true,  
  "current_step": 2,  
  "progress": 40,  
  "result": null,  
  "error": null  
}
```

# 安全注意事项

---

## 1. API密钥保护

- 不要在代码中硬编码API密钥

- 使用环境变量存储敏感信息
- 定期轮换API密钥

## 2. 网络安全

- 在生产环境中使用HTTPS
- 配置适当的CORS策略
- 实施访问控制和身份验证

## 3. 数据隐私

- 及时清理临时文件
- 不要存储用户的个人信息
- 遵守相关的数据保护法规

## 法律声明

---

### 版权注意事项

- 仅用于个人学习和研究目的
- 尊重原创作者的版权
- 遵守YouTube和B站的服务条款
- 不得用于商业用途

### 免责声明

- 用户需自行承担使用风险
- 开发者不对任何损失负责
- 请遵守当地法律法规

## 贡献指南

---

欢迎社区贡献代码和建议：

1. Fork项目仓库
2. 创建功能分支
3. 提交代码更改
4. 创建Pull Request

## 代码规范

- 遵循PEP 8 Python代码规范
- 使用有意义的变量和函数名
- 添加适当的注释和文档
- 编写单元测试

## 更新日志

---

### v1.0.0 (2025-06-23)

- 初始版本发布
- 实现基础的YouTube到B站转载功能
- 支持双语字幕生成
- 提供Web用户界面

## 联系方式

---

如有问题或建议，请通过以下方式联系：

- 项目仓库：[GitHub链接]
- 邮箱：[联系邮箱]
- 社区讨论：[论坛链接]

# 致谢

---

感谢以下开源项目的支持： - OpenAI Whisper - yt-dlp - React - Flask - 以及所有其他依赖项目

---

本文档最后更新于 2025年6月23日