

操作系统支持矩阵国际化与自动翻译应用

基于 GPT 模型的自动国际化生成

孙齐

测试小队

2024 年 7 月 3 日

- ① Self
- ② Support Matrix
- ③ How to: Use GPT
- ④ report-il8n
- ⑤ Demo
- ⑥ Chat Any where

- ① Self
- ② Support Matrix
- ③ How to: Use GPT
- ④ report-il8n
- ⑤ Demo
- ⑥ Chat Any where

Self

- Github: github.com/wychlw
- Email: lingwang#wcysite.com
- Refer as 冷/凜 in most cases

- ① Self
- ② Support Matrix
- ③ How to: Use GPT
- ④ report-il8n
- ⑤ Demo
- ⑥ Chat Any where

Support Matrix

- Mainstream operating systems × Mainstream RISC-V development boards
- Installation guide with test results for each combination
- ruijsdk.org/supported

Mainstream operating systems for RISC-V hardware adaptation (mainstream RISC-V development boards)

CPU	Product number	Arch Linux	Debian/RevyOS	Fedora	FreeBSD	Gentoo	openAnolis	OpenBSD	openCloudOS	openEuler	openKylin	openSUSE	Ubuntu	Tina-Linux	Android 13	Armbian	BuildRoot	OpenHarmony
SG2042	Pioneer Box	N/A	Good	Good	N/A	N/A	N/A	N/A	WIP	Good	Good	N/A	N/A	N/A	N/A	N/A	N/A	WIP
CV1800B	Milk-V Duo (64M)	Basic	Basic	CFH	N/A	N/A	N/A	N/A	N/A	Basic	N/A	N/A	N/A	N/A	N/A	N/A	Basic	N/A
TH1520	LicheePi 4A	Good	Good	Good	N/A	N/A	N/A	N/A	N/A	Good	Good	N/A	WIP	N/A	N/A	Good	N/A	WIP
JH7100	VisionFive	N/A	N/A	Good	N/A	N/A	N/A	Basic	N/A	Good	Good	Basic	Basic	N/A	N/A	Basic	Basic	N/A
JH7110	VisionFive 2	Basic	Good	N/A	WIP	Basic	N/A	Basic	N/A	Good	Good	Basic	Basic	N/A	WIP	Good	Basic	WIP
K230	CamMV K230	N/A	Basic	Basic	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Basic	N/A	N/A	N/A	N/A	N/A

Support Matrix

- Check compatibility
- All guides in one place
- Provide files and scripts

将源码与工具链解压后放到工作区中。

由于官方示例不带有 Makefile 或任何构建脚本，而是采用官方 IDE 进行构建，若要直接使用工具链，请下载这个改好的 [Makefile](#)，并放到源码文件夹下 `EVB/EXAM/FreeRTOS/FreeRTOS` 下。

将刚才获取的 Makefile 中的工具链路径进行修改，即替换 `TOOL_CHAIN_PATH` 和 `OPENOCD_PATH` 中的内容。

接下来进行 `make prepare` 来复制一些必要的代码。

Support Matrix

- Report and fix error

更新/引导 boot

若 u-boot 无法引导而是进入了命令行，需要更新 u-boot（详见：（pr 31）[\[starfive-tech/u-boot#31\]](https://github.com/starfive-tech/u-boot#31)

官方文档：<https://starfivetech.com/uploads/VisionFive%20Single%20Board%20Computer%20Quick%20Start%20Guide.pdf>

u-boot 下载：https://github.com/starfive-tech/Fedora_on_StarFive/releases

或者，手动输入以下命令以引导系统：

```
ext4load mmc 0:1 0x84000000 /boot/Image
ext4load mmc 0:1 0x88000000 /boot/dtb/starfive/jh7100-starfive-visionfive-v1.dtb
setenv bootargs "root=/dev/mmcblk0p1 console=ttyS0,115200n8 console=tty0 earlycon=sbi rootflags=data=writeback stmmaceth=chain_
booti 0x84000000 - 0x88000000
```


Internationalization And Problems

- 50 development boards now with approximately 3-4 operating system each
actually: 219 reports in total
- Huge amount of file to be translated, need to consider human mistakes
- Have some kind of similarity between the reports, but lots of variations
- With technical terms and context

- 1 Self
- 2 Support Matrix
- 3 How to: Use GPT
- 4 report-il8n
- 5 Demo
- 6 Chat Any where

Use GPT API

- Basically just like in the web (no finetuning)
- docs: platform.openai.com/docs/overview
- Lots of usage: embedding, vision tts etc.
Only consider text generation

Roles

Three roles when talk with GPT:

- System: Model Instructions
- Assistant: Model Output
- User: Conversation Input

System seems having more priority.

Context is important in translation, seperate Assistant and User having some benefits.

(Though in other task, impact on different roles may be vary)

Parameters

Model have some parameters:

- temperature : randomness
- max_token : max context length(4095 in 3.5 and 4o)
- top_p : creativity (sometimes opposite to accuracy)
- frequency_penalty : no repeat(generally no need to set)
- presence_penalty : no same token(generally no need to set)

Usage

RESTful API: Construct a JSON and send (or use predefined SDK)

```
1 from openai import OpenAI
2 client = OpenAI()
3
4 response = client.chat.completions.create(
5     model="gpt-4o",
6     messages=[],
7     temperature=1,
8     max_tokens=256,
9     top_p=1,
10    frequency_penalty=0,
11    presence_penalty=0
12 )
```

Usage

Results are in JSON format, content in:

```
response.choices[0].message.content
```

If out of token,

`response.choices[0].finish_reason` will be **"length"**
(continue with no new things add to message)

- ① Self
- ② Support Matrix
- ③ How to: Use GPT
- ④ report-il8n
- ⑤ Demo
- ⑥ Chat Any where

Ideas of Using LLMs

- Good quality <-> Traditional doc translation service
- Fast and cheap <-> Professional translator

Limitation

- Good quality <-> Traditional doc translation service
- Fast and cheap <-> Professional translator

But need to pay attention...

- Hallucinations
- Secretly change minor details
- Manual review is still needed

Auto Translation With OpenAI GPT

- Proj: github.com/wychlw/report-i18n
- Support 3.5 and 4o (Balance between quality and price)
- – Specialized for test report translation, may need to be adjusted for other purposes

Report i18n

- Automaticly translate docs into all target languages
- `\%s/README.md/README_(lang).md/` , with link changed inside
- Will extract everyting in codeblock and not translate them
- Will not translate (mostly) URLs

Usage

- Tweaked model parameters included
- Config:
 - env.py : API_KEY and API_URL
OpenAI official and FREE API provided by
GPT_API_free
 - conf.py : Config lang and (part of) promote

Usage: conf.py

- `max_length` : Max length for each iteration, generally token size if logogram else $\{3, 4\} * \text{tokensize}$
- `target_lang/tips_translated_by_chatgpt` : Language to translate to and the tips
- `model`: 4o if logogram to ideogram (or vise versa), else 3.5
- `template`: !important
eg: **测试成功** -> Test Passed | Test Successful | Test Succeed | Successful Test...
To regulate general structure of the translation.

Usage: Marker and CI

Marker is used to change default behavior (will be removed automatically):

- `r"\[translate\]"` : Force re-translate
- `r"\[update\]"` : Force update translation
- `r"\[skip_lang\[([.*?])\]\]"` : Skip translate language
- Any other good ideas?

A template for CI, automatically translate and push the repo.
(Notice the potential API leak)

- ① Self
- ② Support Matrix
- ③ How to: Use GPT
- ④ report-il8n
- ⑤ Demo**
- ⑥ Chat Any where

Demo: report-i18n


Let's now try translating a report to French?
(Hopefully the network is good enough)

- ① Self
- ② Support Matrix
- ③ How to: Use GPT
- ④ report-il8n
- ⑤ Demo
- ⑥ Chat Any where

Chat Any where

- github.com/chatanywhere/GPT_API_free
- Provide free API for GPT-3.5
- Simply authorize it using github account to get a free key
- Set the API_URL in env.py to `https://api.chatanywhere.tech/v1`

如何使用

- 由于频繁的恶意请求，我们不再直接提供公共的免费Key，现在需要你使用你的Github账号绑定来领取你自己的免费Key
-  [申请领取内测免费API Key](#) 或 [购买内测付费API Key](#)
- 转发Host: `https://api.chatanywhere.tech` (国内中转，延时更低，host1和host2二选一)

授权成功

您的免费API Key为: 

请妥善保管，不要泄露给他人，如泄露造成滥用可能会导致Key被封禁

Thanks for your listening!