蓝心大模型API

官方文档：https://aigc.vivo.com/#/document/index?id=1746

AI能力列表

vivo为本次AIGC竞赛应用赛道的参赛队伍提供了以下AI能力：

能力名称	应用范围
蓝心大模型-70B	内容创作、知识问答、逻辑推理、代码生成、信息提取
蓝心大模型-7B	内容创作、知识问答、逻辑推理、代码生成、信息提取
AI绘画	头像创作、壁纸生成、营销配图、插画生成
通用OCR	文档电子化、智能批改/阅卷、拍照或截图识别、文本审核与管理
文本向量	信息推荐、文档检索、知识挖掘
文本翻译	文档翻译、同传会议、学习辅助、智能硬件
实时短语音识别	语音搜索、聊天输入、游戏娱乐、人机交互
长语音听写	视频字幕、实时会议记录、智能外呼&客服
长语音转写	电话客服、会议记录、字幕生成、语音质检
音频生成	有声阅读、新闻播报、电话客服、信息播报、出行导航
地理位置(POI搜索)	生活购物、旅游规划

蓝心大模型-70B接口

蓝心大模型70B是由 vivo AI 全球研究院自主研发的大规模预训练语言模型，模型接口分为**同步接口**与**流式接口**。

同步接口

同步接口的响应结果为一次性返回。

接口协议

1. 数据包标志：

参数名称	类型	是否必须	是否要urlencod	说明	备注
requestId	string	是	是	请求id	本次数据包的标志id，全局唯一，使用uuid

2. Headers：

参数名称	类型	是否必须	参数值
Content-Type	string	是	application/json
X-AI-GATEWAY-APP-ID	string	是	平台给每个队伍分配的app_id（应用APP ID，见参赛平台-应用赛道参赛资源）
X-AI-GATEWAY-TIMESTAMP	string	是	请求时的Unix时间戳，以秒为单位
X-AI-GATEWAY-NONCE	string	是	8位的随机字符串
X-AI-GATEWAY-SIGNED-HEADERS	string	是	填写 “x-ai-gateway-app-id;x-ai-gateway-timestamp;x-ai-gateway-nonce”
X-AI-GATEWAY-SIGNATURE	string	是	填写签名字符串，计算方式见鉴权方式文档签名计算部分

用于鉴权，其中signature实现复杂。可调用官方提供的代码auth_util.py 鉴权方式-代码实现示例

```python
def gen_sign_headers(app_id, app_key, method, uri, query):
    method = str(method).upper()
    uri = uri
    timestamp = str(int(time.time()))
    app_id = app_id
    app_key = app_key
    nonce = gen_nonce()
    canonical_query_string = gen_canonical_query_string(query)
    signed_headers_string = 'x-ai-gateway-app-id:{}\nx-ai-gateway-timestamp:{}\n' \
                            'x-ai-gateway-nonce:{}'.format(app_id, timestamp, nonce)
    signing_string = '{}\n{}\n{}\n{}\n{}\n{}'.format(method,
                                                     uri,
                                                     canonical_query_string,
                                                     app_id,
                                                     timestamp,
                                                     signed_headers_string)
    signing_string = signing_string.encode('utf-8')
    signature = gen_signature(app_key, signing_string)
    return {
        'Content-Type':'application/json',
        'X-AI-GATEWAY-APP-ID': app_id,
        'X-AI-GATEWAY-TIMESTAMP': timestamp,
        'X-AI-GATEWAY-NONCE': nonce,
        'X-AI-GATEWAY-SIGNED-HEADERS': "x-ai-gateway-app-id;x-ai-gateway-timestamp;x-ai-gateway-nonce",
        'X-AI-GATEWAY-SIGNATURE': signature,
    }
```

3. data：

这部分是传输数据的主体，包括以下内容：

参数名称	二级参数	类型	是否必须	默认值	
prompt		string	否	无	单轮问答内容，p
messages		object[]	否	无	自定义多轮问答_ （1）messages （2）最后一个m （3）必须为奇数 （4）messages
	role	string	是	无	角色：user、ass
	content	string	是	无	内容
model		string	是	无	填写vivo-BlueLM-TB
sessionId		string	是	无	会话id，使用uui 息，messages 不
systemPrompt		string	否	无	人设参数，如可以 你的中文名字叫x 外回复和你的名字
extra		map	否	无	模型超参

extra可选超参：

支持参数	类型	取值范围	建议值	说明
temperature	float	(0,2.0)	0.9	采样温度，控制输出的 出更随机，更具创造性 数，但不要同时调整两
top_p	float	(0, 1.0)	0.7	用温度取样的另一种方 虑具有 top_p 概率质量 tokens建议您根据应用
top_k	integer	(0,1,2,⋯,)	50	在前k个tokens中采样
max_new_tokens	integer	(0, 8000)	2048	生成答案的最大长度
repetition_penalty	float	大于0的浮点数， 一般不超过2	1.02	重复惩罚，1.0默认不惩

4. url

用于大模型能力资源定位，调用不同的能力拼接得到的url是不同的。

DOMAIN = 'api-ai.vivo.com.cn'

URI = '/vivogpt/completions' #蓝心大模型-70B

URI = '/api/v1/task_submit' #AI绘画

……

url = 'https://{}{}'.format(DOMAIN, URI)

使用request发送请求

response = requests.post(url, json,data, ***kwargs*)，response存放大模型响应。

```
params = {
    'requestId': str(uuid.uuid4()) #本次数据包的标志id，全局唯一，使用uuid
}
print('requestId:', params['requestId'])

data = {
    'prompt': f"""请介绍一下蓝心大模型""",
    'model': 'vivo-BlueLM-TB', #填写vivo-BlueLM-TB
    'sessionId': str(uuid.uuid4()), #会话id，使用uuid，每次唯一。当结合 prompt 使用时，会关联相同 sessionId 的历史消息，messages 不受 sessionId 影响。
    'extra': {
        'temperature': 0.1
    }
}

headers = gen_sign_headers(APP_ID, APP_KEY, METHOD, URI, params)

url = 'https://{}{}'.format(DOMAIN, URI)
response = requests.post(url, json=data, headers=headers, params=params)
```

响应结果

使用response存储收到的响应。status_code ==200表示正常响应。

```python
response = requests.post(url, json=data, headers=headers, params=params)

if response.status_code == 200:
    res_obj = response.json()
    print(f'response:{res_obj}')
    if res_obj['code'] == 0 and res_obj.get('data'):
        content = res_obj['data']['content']
        print(f'final content:\n{content}')
else:
    print(response.status_code, response.text)
```

脚手架代码

可能多数同学没有报名参赛，无法通过官方分配的appid和appkey与蓝心大模型对话。我们写了一个“本地的蓝心大模型”响应请求，供同学们编程模拟调用api。在bluelm_script.ipynb文件中，我们导入了一个本地模块BlueLM，使用其中的类Vivorequest（import as request，代码中使用只需写request）来代替request库中的requests。传入参数相同。

response = request.post(url, json, headers, params)

当然我们“本地的蓝心大模型”并不具备文本生成自由对话的能力，无论你传入什么提示词它都会返回几个固定的句子。本代码目的是让同学们体验编写请求调用api以及处理大模型响应内容的过程。

设定鉴权参数

正确的参数如下，你可以尝试修改APP_ID或者url观察响应。

APP_ID = '20240422'

APP_KEY = ''

URI = '/vivogpt/completions'

DOMAIN = 'api-ai.vivo.com.cn'

METHOD = 'POST'

编写接口并处理响应

1. 编写需要传入request.post()的四个参数，可参考示例代码文件vivo.ipynb
2. 编写接收到响应response后的处理代码，最终输出大模型回复的内容或者报错提示信息。