“源 1.0”API 调用使用手册

浪潮（北京）电子信息产业有限公司

尊敬的用户：

衷心感谢您选用了浪潮人工智能巨量模型“源1.0”API！

本手册介绍了“源 1.0”已开放 API 的接口说明和使用示例，可使使用者更好地了解本 API 支持的功能及快速使用方法，充分的发挥开放 API 的作用。

本手册中的内容如有变动恕不另行通知。

如果您对本手册有疑问或建议，请向浪潮（北京）电子信息产业有限公司垂询。

浪潮（北京）电子信息产业有限公司

二零二一年十二月

目录

1 接口说明

“源1.0”，是浪潮人工智能研究院9月28日在京发布全球最大规模人工智能巨量模型。“源”的单体模型参数量达2457亿，超越美国OpenAI组织研发的GPT-3，成为全球最大规模的单体AI巨量模型。

本文将介绍如何进行“源1.0”API的调用。该API接口主要是针对外网开放，用于第三方用户根据自身需求获取推理结果。

用户使用时，需进行两步操作：

（1）首先调用第一个API接口发起推理请求，获取唯一标识，此时后台在进行推理中；

（2）用户根据API接口返回的唯一标识，轮询调用第二个API接口获取推理结果。

详见下方接口。

1.1 发起推理请求

接口说明：向推理服务发送推理请求，并获得该次请求的requestID；

请求方式：get

请求链接：/v1/interface/api/requestId

请求头参数：

参数	类型	说明	是否必须
token	String	验签 token，Header 中传递 用户在申请 API 调用并获得授权后，可自行生成 token; token 为使用 MD5 对用户账号、手机号、日期（yyyy-MM-dd）的字符串拼接进行加密产生;	是

请求体参数：

参数	类型	说明	是否必须
account	String	用户账号	是
data	String	要推理的问题，如对联上联等	是
temperature	Float	采样 temperature，用于模型生成多样性，默认值 0.9,	否
topP	Float	Top P 采样，默认值 0.1	否
topK	Int	Top K 采样，默认值 1	否
tokensToGenerate	Int	生成 tokens 数目，要求与输入的 tokens 数目之和小于 2048, 根据实际应用场景变换	是
type	String	数据类型 目前传”api”即可，后续会进行扩展	是

请求示例：

http://api-air.inspur.com:32102/v1/interface/api/requestId?account=inspur&data=上联：园中草木春无数；下联：湖上山林画不如。上联：好事流传千古；下联：&temperature=1.0&topP=0.8&topK=5&tokensToGenerate=10&type=api

注意：本示例 data 以请求对联为例，具体详解可参考 2.2 部分。

返回参数：

参数	类型	说明	是否必须
flag	Boolean	标志位，表明是否调用成功； true：成功 false：失败	是
errCode	String	错误码，用来定位错误原因；	是
errMessage	String	后台返回信息描述，如果 flag=false，则此处为错误信息；	是
exceptionMsg	String	程序报错的异常信息 常见的报错信息在【exceptionMsg】中，包含：**推理数据内容过长**，要求输入与输出的 tokens 数目之和小于 2048； **参数不合法**：输入参数不合法； **接口调用信息保存失败**：对于请求信息进行保存时发生异常，需要重试； **未知异常**：发生了不可知异常，需要重试。	是
resData	String	该次请求的唯一标识，可用于查询推理结果；	是

返回示例-调用成功

{

```
        "flag": true,
        "errCode": None,
        "errMessage": None,
        "exceptionMsg": None,
        "resData": "24a017a9f9794f85a8e57e300a06381c" // 唯一标识
    }
```

返回示例- 调用失败

```
{
    "flag": false,
    "errCode": None,
    "errMessage": None,
    "exceptionMsg": ‘参数不合法’,
    "resData": None
}
```

1.2 获取推理结果

接口说明：根据推理请求接口返回requestID，查询推理结果

请求方式：get

请求链接：/v1/interface/api/result

请求头参数：

参数	类型	说明	是否必须
token	String	验签 token，Header 中传递 用户在申请 API 调用并获得授权后，可自行生成 token； token 为使用 MD5 对 用户账号、手机号、日期（yyyy-MM-dd）的字符串拼接进行加密产生；	是

请求体参数：

参数	类型	说明	是否必须
account	String	用户账号	是
requestId	String	调用发送推理请求接口获得，用于查询推理结果；	是

请求示例：

http://api-air.inspur.com:32102/v1/interface/api/result?account=inspur&requestId=24a017a9f9794f85a8e57e300a06381c

返回参数：

flag、errCode、errMessage、exceptionMsg 的解释同其他接口一致；

参数	类型	说明	是否必须
flag	Boolean	标志位，表明是否调用成功； true：成功 false：失败	是
errCode	String	错误码，用来定位错误原因；	是
errMessage	String	后台返回信息描述，如果 flag=false，则此处为错误信息；	是
exceptionMsg	String	程序报错的异常信息 【exceptionMsg】包含： **token 验证失败、** **账号信息有误、** **用户授权接口信息为空、** **用户授权已过期、** **用户调用接口次数超出限制、** **未知异常**	是
resData	String	正常返回模型推理结果	是

返回示例-调用成功

```
{
    "flag": true,
    "errCode": None,
    "errMessage": None,
    "exceptionMsg": None,
    "resData": "佳名留在人间。上联：一水护田将绿绕"
}
```

返回示例-调用失败

```
{
    "flag": false,
    "errCode": None,
    "errMessage": None,
    "exceptionMsg": “账号不合法”,
    "resData": None
}
```

2 接口 demo 示例

2.1 示例代码

代码请参考yuan_api_demo.py，详细内容如下：

（1） 运行逻辑

第一步：使用md5加密获得token，每天生成一次或每次都生成均可

```
t=time.strftime("%Y-%m-%d", time.localtime())
account= "inspur"   # 替换为申请的账号名
phone= "123456789012"   # 替换为申请帐号时填写的手机号
token=code_md5(account+phone+t)
print(token)
headers = {'token': token}
```

第二步：发起推理请求

```
ques="上联：园中草木春无数；下联：湖上山林画不如。上联：好事流传千古；下联："
url="http://api-air.inspur.com:32102/v1/interface/api/requestId?"
temperature=0.9  # 采样temperature，用于模型生成多样性，默认值0.9
topP=0.1  # Top P 采样，默认值0.1
topK=1  # Top K 采样，默认值1
tokensToGenerate=10 # 生成tokens，建议与输入的token的个数之和小于2048
url=url+"account={0}&data={1}&temperature={2}&topP={3}&topK={4}&tokensToGenerate={5}&type={6}".format(account,ques,temperature,topP,topK,tokensToGenerate,"api")
response=rest_get(url,headers,30)
response_text = json.loads(response.text)
if response_text["flag"]:
    requestId = response_text["resData"]
else:
```

```
            print(response_text)     #打印异常信息
            exit()
```

第三步：轮询查询推理结果

```
        url = "http://api-air.inspur.com:32102/v1/interface/api/result?"
        url = url + "account={0}&requestId={1}".format(account,
requestId)
        while (1):
            response = rest_get(url, headers, 30)
            response_text = json.loads(response.text)
            if response_text["resData"] != None:
                break
            if response_text["flag"] == False:
                print(response_text)  #打印异常信息
                exit()
        print(response_text)
```

（2） 涉及函数

- md5编码

```
        def code_md5(str):
            code=str.encode("utf-8")
            m = hashlib.md5()
            m.update(code)
            result= m.hexdigest()
            return result
```

- get请求

```
def rest_get(url, header, timeout, show_error=False):
    '''Call rest get method'''
    try:
        response = requests.get(url,
headers=header,timeout=timeout, verify=False)
```

```
            return response
        except Exception as exception:
            if show_error:
                print(exception)
            return None
```

2.2 应用示例

“源1.0”的主要目标是用更少的领域数据、且不经过精调步骤去解决问题。预训练好的“源1.0”支持不同输入形式下（zero-shot、one-shot、few-shot）的推理。zero-shot即是对某（些）类别完全不提供样本示例，one-shot或few-shot对某（些）类别只提供一个或者少量的样本示例，样本示例对模型推理下游任务起引导作用，所以建议用户采用one-shot或few-shot。

2.2.1 对对联

用户想对对联，上联为“好事流传千古”，则可将请求参数中的data进行如下赋值：

- zero-shot

  上联：好事流传千古；下联：

- one-shot

  上联：园中草木春无数；下联：湖上山林画不如。上联：好事流传千古；下联：

- few-shot

  上联：千秋笔墨惊天地；下联：万里云山入画图。上联：园中草木春无数；下联：湖上山林画不如。上联：好事流传千古；下联：

2.2.2 根据主题词写诗

主题词为“清风”，则可将请求参数中的data进行如下赋值：

- zero-shot

  以清风为题作一首诗：

- one-shot

  春风用意匀颜色，销得携觞与赋诗。秾丽最宜新著雨，娇饶全在欲开时。

以清风为题作一首诗：

- few-shot

  或从十五北防河，便至四十西营田。去时里正与裹头，归来头白还成边。

春风用意匀颜色，销得携觞与赋诗。秾丽最宜新著雨，娇饶全在欲开时。以清风为题作一首诗：

3 超参数调优

3.1 超参数含义

- tokens_to_generate（整数型，范围 1-2047）

  预期模型生成 token 数目，要求与输入的 token 的个数之和小于 2048。该参数设置越大，模型生成答案耗时越长，建议在实际应用中根据所需合理设置。

- Temperature（浮点型，范围 0-1）

  为了解决搜索生成缺乏多样性问题，当前我们通过采样来增加随机性。但增加随机性同时，生成可能会出现语法错误。可以通过强化顶部词的概率，然后只对最有可能的一些词进行采样，这样就能够在增加随机性的同时，又保证不出现一般性的错误。

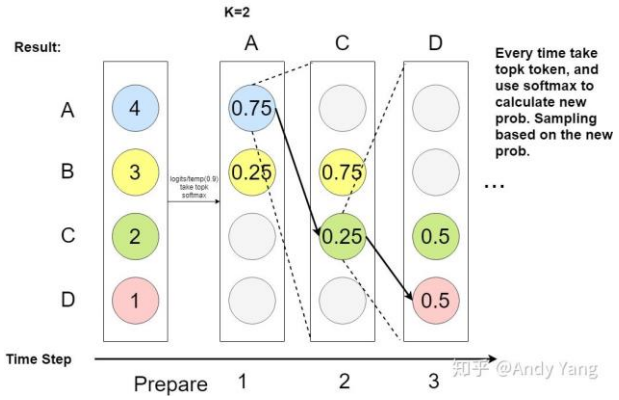
  强化顶部词概率，可以通过对模型输出的 logits 除以一个小于 1 的温度（Temperature，T）。

$$p(i) = \frac{e^{\frac{f(i)}{T}}}{\sum_j e^{\frac{f(j)}{T}}}$$

  这样通过 softmax 后使得分布更加尖锐，大概率的词概率更大。之后根据获得概率对顶部词先进行挑选，然后再采样，直接杜绝了低概率词出现的可能性。而这里挑选的策略，我们采用 TopK 和 TopP。

- TopK 采样（整数型，范围 1-inf）

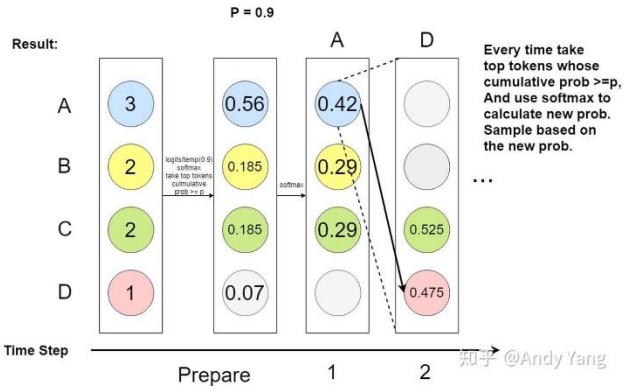
挑选概率最高 k 个 token，然后重新过 softmax 算概率，之后根据获得概率进行采样，接着进行下一步生成，不断重复。



但关于 TopK 有可能会出现一个问题，假如模型对当前生成非常肯定，比如概率最高的 token 的概率 0.9，而其余的 token 概率都很低。这时如果只用 topk 采样的话，就会导致采样到低概率情况仍然发生。因此我们需要对顶部 token 的累计概率进行限制，这就是 TopP 采样。

- TopP 采样（浮点型，范围 0-1）

TopP 是先设置一个概率界限，比如说 p=0.9，然后从最大概率的 token 往下开始取，同时将概率累加起来，当取到大于等于 p 也就是 0.9 时停止。如果最大 token 概率就已经有 0.9 了，那么就只取最大的一个 token。关于 TopP 简单的示意图如下：



3.2 超参数调优示例

参数组合没有绝对优劣之分，用的时候建议多套参数尝试。本节我们以作诗为例讲述超参数如何调优。我们一般采用如下四套参数。

	set0	set1	set2	set3
超参数组合	top_p=0.8 top_k=5	top_p=0, top_k=5	top_p=0.8, top_k=0	top_p=0, top_k=0
	tokens_to_generate = 40 temperature = 1			

Batch size 设置 16，以“打卤面”为题作一首诗，经过后处理筛选后输出如下：

set0	打卤面来香喷鼻，白鸡豚菜味尤佳。何须更问长安市，尽道新丰市上夸。 打卤面肥汤滑美，玉盘堆栈玉脂堆。何如白面和豚脯，一饱何须更问家。 打卤面来香且美，玉纤山馆更何疑。莫言此味无人识，只有山翁识得知。 打卤面筋汤泼雪，玉纤纤指拨寒灰。不因食得君家饭，安得长生似玉妃。 打卤面肥味更鲜，新炊玉粒熟炊烟。莫嫌不作山珍荐，山珍海错等闲看。 打卤面肥汤滑美，生葱细韭味鲜长。不须更问汤何物，只此便是玉露浆。 打卤面来真绝品，雪芽茶出是名花。从今只合闲吟竹，莫向人间问小家。 打卤面如雪，新炊玉露香。莫言非珍膳，曾是帝家尝。 打卤面筋滑，新炊玉雪香。何须更下酱，已觉齿颊芳。 打卤面肥汤滑鲜，玉炊新得进黄门。御厨已办供新膳，不似山家野店存。 打卤面筋香滑滑，打卤面味鲜美鲜。打卤面中有鸡丁，更添滋味更饶先。 打卤面新出，山家饭正香。莫嫌山简醉，且作少陵狂。 一味打卤面，能令老饕涎。只今无此味，谁与作诗人。
set1	打卤面肥春未回，青丝黄韭趁时开。不须更问新炊候，只是春来已著雷。 小瓮打卤面筋香，玉碗盛来琥珀光。不须更待金盘送，只是人间一样尝。 打卤面肥香滑腻，白鱼鳞嫩味鲜腴。若非玉雪肌肤瘦，那得银丝入齿凉。 新炊打卤面新香，一味能令齿颊芳。更喜新添鸡鸭脚，一瓯新水更添汤。 打卤面筋卤面筋，一勺香雪玉生烟。更添一瓯君家好，只有新添胜旧篇。 玉盘打卤面筋滑，玉屑蒸酥鸭味腴。莫笑山珍与河错，只今谁是老馋徒。 打卤何人好，盐花不似多。若教添水煮，便是打卤河。 一瓯打卤面，再饱打卤面。饱食无馀事，何妨不著饭。 玉盘新碾打卤面，金碗新添鸭头春。莫怪厨人无妙思，只因今日是清明。 面筋堆卤玉，鸡子打卤汤。何须用豚鱼，但取肥而香。 一饱聊可饱饥肠，不向人间觅打卤。莫向人间论价重，人间那得面如玉。 新来打卤面如雪，下著盐花作玉花。莫道盐中有滋味，此中滋味胜他家。
set2	我前岁拜国司农，国伯迎公接我游。打卤五花喷鼻滑，未烦和酱已心留。 蓝团纱笼带乌纱，九酝云浆打卤茶。花睡未残人未起，翠岚深护玉纤斜。 碎玉敲金舞爪门，双楸黄女与娇门。薰来打卤为羊墅，不与卢王家妇论。 自寻打卤面数茎，更去浇汁美蔬腴。见说山家早饥黍，老馋未必俗心嗔。 买炊老何人，来沽味堪全。纸窗明竹席，打卤面青莲。 将军坐镇带秦关，拥卒千盘宝鸭鲜。沙苑杏浇新麦饭，打卤面截内家年。
set3	记莫交阑按矮钟，别来猿陌窄如川。却愁打卤尝新味，只羡屠人供细帘。 天下有腴益尚奢，时闻高卧打卤面。当年凿味今何在，争似君家添信悬。 玉炊红夫晨梦回，六街暮绝打卤声。拾中木犊天同露，不似江湖能得逢。

set0 格式最整齐，倾向于贴近主题，能接受的诗最多，是我们的推荐的默认参数。

set1、2 的格式稍差一些，倾向于按照主题的意思作诗（不出现原词）。

set4 看起来最差，但是想做现代诗时效果又很好。

所以在实际应用中超参数没有绝对的好与坏，一般建议多尝试。

- 推荐参数组合

```
tokens_to_generate = 40
```

```
top_p = 0.8
top_k = 5
temperature = 1
```