

ALIBABA CLOUD

阿里云

自然语言处理
NLP自学习平台

文档版本：20200908

 阿里云

法律声明

阿里云提醒您阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
<code>Courier</code> 字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
<i>斜体</i>	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
<code>[]</code> 或者 <code>[a b]</code>	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
<code>{}</code> 或者 <code>{a b}</code>	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.API参考	05
1.1. API概览	05
1.2. 调用方式	05
1.3. 签名机制	06
1.4. 公共参数	09
1.5. 获取AccessKey	10
1.6. GetPredictResult	12
1.7. 获取模型预测结果接口示例	14
1.8. CreateAsyncPredict	15
1.9. GetAsyncPredict	18
1.10. 异步预测接口使用示例	20
1.11. RunPreTrainService	24
1.12. 预训练模型使用教程	26
2.SDK示例	56
2.1. SDK概述	56
2.2. Java SDK	56
2.3. Python SDK	57
2.4. C# SDK	58
2.5. C++ SDK	59
2.6. PHP SDK	61
2.7. Go SDK	62
2.8. Nodejs SDK	63
2.9. Ruby SDK	64

1.API参考

1.1. API概览

NLP自学习平台处理提供以下相关API接口。

API	描述
GetPredictResult	调用GetPredictResult获取模型预测结果
CreateAsyncPredict	调用CreateAsyncPredict创建一个异步预测
GetAsyncPredict	调用GetAsyncPredict获取异步预测结果
RunPreTrainService	调用预训练模型服务

1.2. 调用方式

NLP自学习平台API调用是通过向NLP自学习平台服务的服务端地址发送HTTP GET请求，并按照接口说明在请求中加入相应请求参数，调用后系统会返回处理结果。请求及返回结果都使用UTF-8字符集进行编码。

请求结构

NLP自学习平台服务的API是RPC风格，您可以通过发送HTTP GET请求调用NLP自学习平台API。

其请求结构如下：

```
http://Endpoint/?Action=xx&Parameters
```

其中：

- **Endpoint**：NLP自学习平台API的服务接入地址为 *nlp-automl.cn-hangzhou.aliyuncs.com*。
- **Action**：要执行的操作，如调用GetPredictResult获取模型预测结果。
- **Version**：要使用的API版本，NLP自学习平台的API版本是 *2019-11-11*。
- **Parameters**：请求参数，每个参数之间用“&”分隔。

请求参数由公共请求参数和API自定义参数组成。公共参数中包含API版本号、身份验证等信息，详情请参见[公共参数](#)。

下面是一个调用GetPredictResult接口获取模型预测结果的示例：

 **说明** 为了便于用户查看，本文档中的示例都做了格式化处理。

```
https://nlp-automl.cn-hangzhou.aliyuncs.com/?Action=GetPredictResult
&Format=xml
&Version=2019-11-11
&Signature=xxxx%xxxx%3D
&SignatureMethod=HMAC-SHA1
&SignatureNonce=15215528852396
&SignatureVersion=1.0
&AccessKeyId=key-test
&TimeStamp=2012-06-01T12:00:00Z
...
```

1.3. 签名机制

为保证API的安全调用，在调用API时阿里云会对每个API请求通过签名（Signature）进行身份验证。无论使用HTTP还是HTTPS协议提交请求，都需要在请求中包含签名信息。

概述

RPC API要按如下格式在API请求的Query中增加签名（Signature）：

```
https://Endpoint/?SignatureVersion=1.0&SignatureMethod=HMAC-SHA1&Signature=CT9X0VtwR86fNW
SnsC6v8YGOjuE%3D&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf
```

其中：

- SignatureMethod：签名方式，目前支持HMAC-SHA1。
- SignatureVersion：签名算法版本，目前版本是 1.0。
- SignatureNonce：唯一随机数，用于防止网络重放攻击。用户在不同请求间要使用不同的随机数值，建议使用通用唯一识别码（Universally Unique Identifier, UUID）。
- Signature：使用AccessKey Secret对请求进行对称加密后生成的签名。

计算签名

签名算法遵循RFC 2104 HMAC-SHA1规范，使用AccessSecret对编码、排序后的整个请求串计算HMAC值作为签名。签名的元素是请求自身的一些参数，由于每个API请求内容不同，所以签名的结果也不尽相同。

```
Signature = Base64( HMAC-SHA1( AccessSecret, UTF-8-Encoding-Of(StringToSign)) )
```

完成以下操作，计算签名：

1. 构建待签名字符串

i. 使用请求参数构造规范化的请求字符串（Canonicalized Query String）：

- a. 按照参数名称的字典顺序对请求中所有的请求参数（包括公共请求参数和接口的自定义参数，但不包括公共请求参数中的Signature参数）进行排序。

当使用GET方法提交请求时，这些参数就是请求URI中的参数部分，即URI中“?”之后由“&”连接的部分。

b. 对排序之后的请求参数的名称和值分别用UTF-8字符集进行URL编码。编码规则如下：

字符	编码方式
A-Z、a-z和0-9以及“-”、“_”、“.”和“~”	不编码。
其它字符	编码成 %XY 的格式，其中 XY 是字符对应ASCII码的16进制表示。例如英文的双引号 (" ") 对应的编码为 %22 。
扩展的UTF-8字符	编码成 %XY%ZA... 的格式。
英文空格	<p>编码成 %20 ，而不是加号 (+) 。</p> <p>该编码方式和一般采用的 application/x-www-form-urlencoded MIME格式编码算法（例如Java标准库中的 java.net.URLEncoder 的实现）存在区别。编码时可以先用标准库的方式进行编码，然后把编码后的字符串中的加号 (+) 替换成 %20 ，星号 (*) 替换成 %2A ， %7E 替换回波浪号 (~) ，即可得到上述规则描述的编码字符串。本算法可以用下面的 percentEncode 方法来实现：</p> <pre>private static final String ENCODING = "UTF-8"; private static String percentEncode(String value) throws UnsupportedOperationException { return value != null ? URLEncoder.encode(value, ENCODING).replace("+", "%20").replace("*", "%2A").replace("%7E", "~") : null; }</pre>

- c. 将编码后的参数名称和值用英文等号 (=) 进行连接。
- d. 将等号连接得到的参数组合按步骤 i 排好的顺序依次使用 "&" 符号连接，即得到规范化请求字符串。

- ii. 将第一步构造的规范化字符串按照下面的规则构造待签名的字符串。


```
StringToSign=
HTTPMethod + "&" +
percentEncode( "/" ) + "&" +
percentEncode(CanonicalizedQueryString)
```

其中：


- HTTPMethod 是提交请求用的HTTP方法，比如GET。
- percentEncode("/") 是按照步骤1.1中描述的 URL 编码规则对字符 "/" 进行编码得到的值，即 %2F。
- percentEncode(CanonicalizedQueryString) 是对步骤1中构造的规范化请求字符串按步骤 1.2 中描述的URL编码规则编码后得到的字符串。

2. 计算签名

- i. 按照RFC2104的定义，计算待签名字符串（StringToSign）的HMAC值。

 **说明** 计算签名时使用的Key就是您持有的AccessKey Secret并加上一个 "&" 字符 (ASCII:38)，使用的哈希算法是SHA1。

- ii. 按照Base64编码规则把上面的HMAC值编码成字符串，即得到签名值（Signature）。
- iii. 将得到的签名值作为Signature参数添加到请求参数中。

 **说明** 得到的签名值在作为最后的请求参数值提交时要和其它参数一样，按照RFC3986的规则进行URL编码。

示例

以GetPredictResultAPI为例，假设使用的 AccessKey Id 为 testid ， AccessKey Secret 为 testsecret 。签名前的请求URL如下：

```
http://nlp-automl.cn-hangzhou.aliyuncs.com/?Timestamp=2016-02-23T12:46:24Z&Format=XML&AccessKeyId=testid&Action=GetPredictResult&SignatureMethod=HMAC-SHA1&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf&Version=2019-11-11&SignatureVersion=1.0
```

使用 testsecret ，计算得到的签名值是：

```
OLeaidS1jvxuMvnyHOwuj+uX5qY=
```

最后将签名作为Signature参数加入到URL请求中，最后得到的URL为：

```
http://nlp-automl.cn-hangzhou.aliyuncs.com/?SignatureVersion=1.0&Action=GetPredictResult&Format=XML&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf&Version=2019-11-11&AccessKeyId=testid&Signature=OLeaidS1jvxuMvnyHOwuj+uX5qY=&SignatureMethod=HMAC-SHA1&Timestamp=2016-02-23T12%3A46%3A24Z
```


1.4. 公共参数

介绍每个接口都需要使用的请求参数和返回参数。

公共请求参数

公共请求参数表

名称	类型	是否必须	描述
Format	String	否	返回消息的格式。取值： <i>JSON (默认值) XML</i>
Version	String	是	API版本号，使用YYYY-MM-DD日期格式。取值： <i>2019-11-11</i>
AccessKeyId	String	是	访问服务使用的密钥ID。
Signature	String	是	签名结果串。
SignatureMethod	String	是	签名方式，取值： <i>HMAC-SHA1</i>
Timestamp	String	是	请求的时间戳，为日期格式。使用UTC时间按照 ISO8601标准，格式为YYYY-MM-DDThh:mm:ssZ。 例如，北京时间2013年1月10日20点0分0秒，表示为2013-01-10T12:00:00Z。
SignatureVersion	String	是	签名算法版本，取值： <i>1.0</i>
SignatureNonce	String	是	唯一随机数，用于防止网络重放攻击。 在不同请求间要使用不同的随机数值。
ResourceOwnerAccount	String	否	本次API请求访问到的资源所有者账户，即登录用户名。

示例

```
http://nlp-automl.cn-hangzhou.aliyuncs.com/?Action=GetPredictResult
&TimeStamp=2014-05-19T10%3A33%3A56Z
&Format=xml
&AccessKeyId=testid
&SignatureMethod=Hmac-SHA1
&SignatureNonce=NwDAxvLU6tFE0DVb
&Version=2019-11-11
&SignatureVersion=1.0
&Signature=Signature
```

公共返回参数

API返回结果采用统一格式，调用成功返回的数据格式有XML和JSON两种，可以在发送请求时指定返回的数据格式，默认为XML格式。每次接口调用，无论成功与否，系统都会返回一个唯一识别码RequestId。

- 返回 2xx HTTP状态码表示调用成功。
- 返回 4xx 或 5xx HTTP状态码表示调用失败。

公共返回参数示例如下：

- XML格式

```
<?xml version="1.0" encoding="utf-8"?>
<!--结果的根结点-->
<接口名称+Response>
<!--返回请求标签-->
<RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>
<!--返回结果数据-->
</接口名称+Response>
```

- JSON格式

```
{
  "RequestId": "4C467B38-3910-447D-87BC-AC049166F216",
  /*返回结果数据*/
}
```

1.5. 获取AccessKey


您可以为阿里云主账号和子账号创建一个访问密钥（AccessKey）。在调用阿里云API时您需要使用AccessKey完成身份验证。

背景信息

AccessKey包括AccessKey ID和AccessKey Secret。

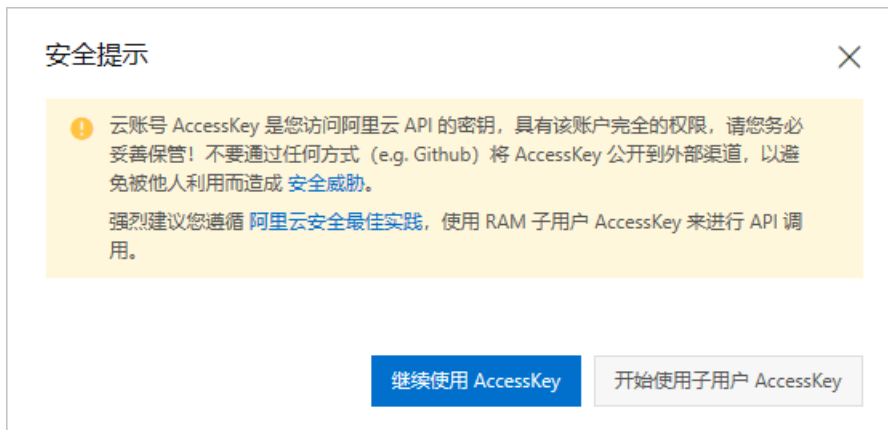
- AccessKey ID：用于标识用户。

- **AccessKey Secret**：用于验证用户的密钥。**AccessKey Secret**必须保密。

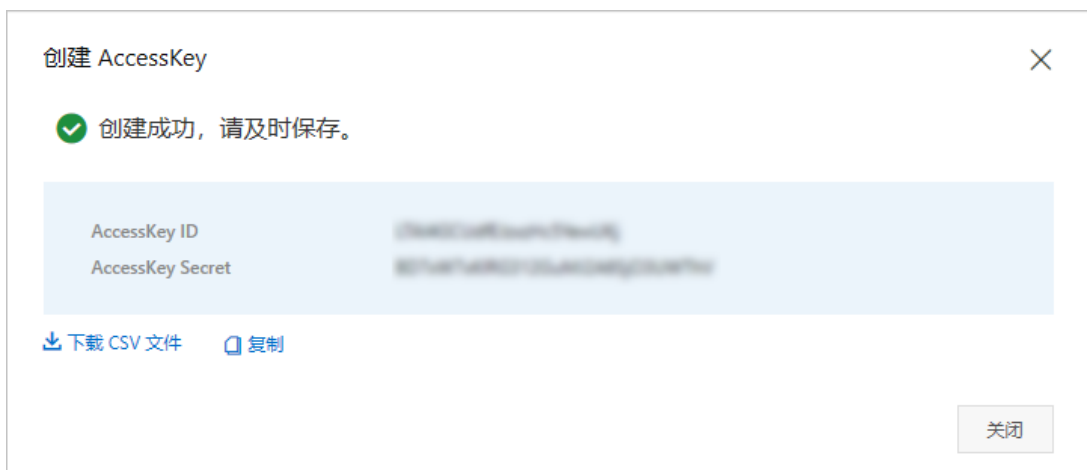
 **警告** 主账号Accesskey泄露会威胁您所有资源的安全。建议使用子账号（RAM用户）Accesskey进行操作，可以有效降低Accesskey泄露的风险。

操作步骤

1. 以主账号登录 [阿里云管理控制台](#)。
2. 将鼠标置于页面右上方的账号图标，单击**accesskeys**。
3. 在安全提示页面，选择获取主账号还是子账号的Accesskey。

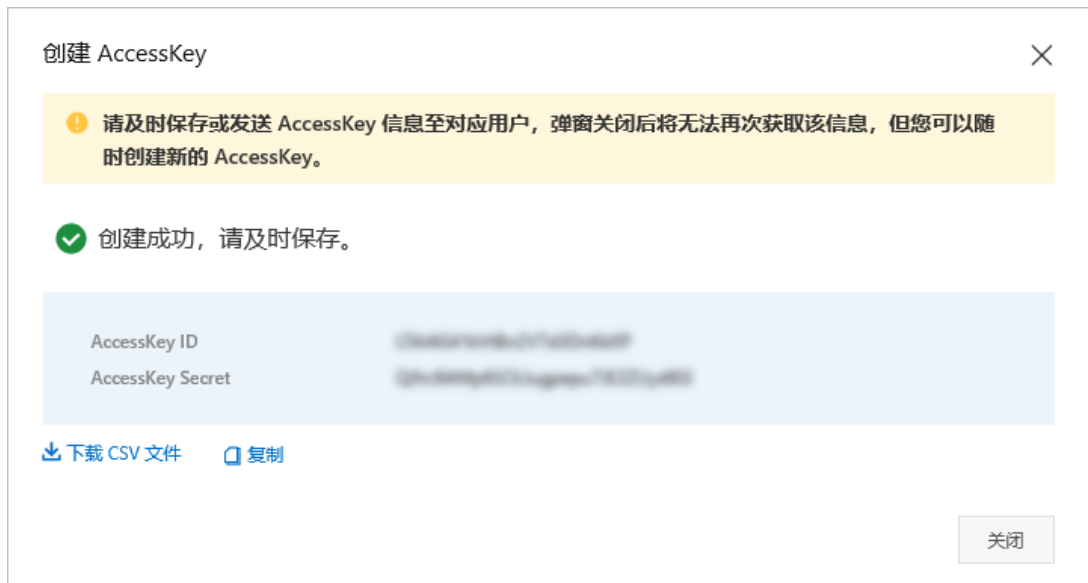


4. 获取账号Accesskey。
 - 获取主账号AccessKey
 - a. 单击继续使用AccessKey。
 - b. 在安全信息管理页面，单击创建AccessKey。
 - c. 在手机验证页面，获取验证码，完成手机验证，单击确定。
 - d. 在新建用户AccessKey页面，展开AccessKey详情，查看AccessKey ID和AccessKey Secret。可以单击保存AK信息，下载AccessKey信息。



- 获取子账号AccessKey
 - a. 单击开始使用子用户AccessKey。

- b. 如果未创建RAM用户，在系统跳转的**RAM访问控制台**的新建用户页面，创建RAM用户。如果是获取已有RAM用户的Accesskey，则跳过此步骤。
- c. 在**RAM访问控制台**的左侧导航栏，选择**人员管理 > 用户**，搜索需要获取AccessKey的用户。
- d. 单击用户登录名称，在用户详情页**认证管理**页签下的**用户AccessKey**区域，单击**创建新的AccessKey**
- e. 在**手机验证**页面，获取验证码，完成手机验证，单击**确定**。
- f. 在**创建AccessKey**页面，查看AccessKey ID和AccessKey Secret。可以单击**下载CSV文件**，**下载AccessKey信息**或者单击**复制**，复制AccessKey信息。



1.6. GetPredictResult

调用GetPredictResult获取模型预测结果

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetPredictResult	系统规定参数。取值：GetPredictResult。
Content	String	是	中华人民共和国人民解放军	预测文本。
ModelId	Integer	是	1818	模型id。
ModelVersion	String	是	V1	模型版本。

名称	类型	是否必选	示例值	描述
DetailTag	String	否	true	文本实体抽取模型预测是否返回详细结果，取值true或者false。
TopK	Integer	否	3	关键短语抽取模型，设置返回top结果数量。

返回数据

名称	类型	示例值	描述
Content	String	["人民解放军","中华人民共和国"]	预测结果，格式为JSON字符串。
RequestId	String	32CE4E89-218B-4DAE-A103-E6BFA66EE63B	请求id。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetPredictResult
&Content=中华人民共和国人民解放军
&ModelId=1818
&ModelVersion=V1
&<公共请求参数>
```

正常返回示例

XML 格式

```
<GetPredictResultResponse>
<requestId>32CE4E89-218B-4DAE-A103-E6BFA66EE63B</requestId>
<content>["人民解放军","中华人民共和国"]</content>
</GetPredictResultResponse>
```

JSON 格式

```
{
  "content":["人民解放军","中华人民共和国"],
  "requestId":"32CE4E89-218B-4DAE-A103-E6BFA66EE63B"
}
```

错误码

HttpCode	错误码	错误信息	描述
403	Forbidden	You are not authorized to operate on the specified resource.	用户未被授权操作指定的资源。
503	ServiceUnavailable	An error occurred while processing your request.	服务内部错误。
404	ResourceNotFound	The specified Model does not exist.	指定的模型不存在。
400	InvalidAccountStatus.ServiceUnavailable	The NLP Auto Machine Learning service has not been activated.	用户未开通NLP自学习服务。
400	InvalidAccountStatus.ServiceLimit	The maximum number of free API daily visits is exceeded.	API每天访问次数超过了免费额度限制。
400	InvalidParameter.ModelId	The specified modelId is invalid.	参数modelId不合法。
400	InvalidParameter.ModelVersion	The specified modelVersion is invalid.	参数modelVersion不合法。
400	InvalidContentLength	The length of content exceeds the limit.	内容长度过长。
500	InternalServerError	An error occurred while processing your request.	未知错误。

访问[错误中心](#)查看更多错误码。

1.7. 获取模型预测结果接口示例

该文档介绍如何使用模型预测接口获取相应模型类型的预测结果

接口调用示例代码

Java代码示例

```
DefaultProfile defaultProfile = DefaultProfile.getProfile("cn-hangzhou","your-access-id-key","your-access-id-secret");
IAcsClient client = new DefaultAcsClient(defaultProfile);
CommonRequest request = new CommonRequest();
request.setMethod(MethodType.POST);
request.setDomain("nlp-automl.cn-hangzhou.aliyuncs.com");
request.setVersion("2019-11-11");
request.setAction("GetPredictResult");
request.putQueryParameter("RegionId", "cn-hangzhou");
request.putQueryParameter("ModelId", modelId);
request.putQueryParameter("Content", content);
request.putQueryParameter("ModelVersion", version);
CommonResponse response = client.getCommonResponse(request);
System.out.println(response.getData());
```

[更多代码示例请点击](#)

短文本匹配模型类型Content输入

 **说明** 模型类型为短文本匹配时content参数为字符串传入，参考 下面的举例

```
[["新买的房子办理房产证需要神材料","如何办理房产证"]]
```

Java代码示例

```
JSONArray items = new JSONArray();
JSONArray item = new JSONArray();
item.add("新买的房子办理房产证需要神材料");
item.add("如何办理房产证");
items.add(item);
String content = items.toString();
System.out.println(content);
```

1.8. CreateAsyncPredict

调用CreateAsyncPredict创建一个异步预测。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CreateAsyncPredict	系统规定参数。取值：CreateAsyncPredict。
Content	String	是	自然语言处理	预测内容，最大长度：10000字。
ModelId	Integer	是	1818	模型ID。
ModelVersion	String	是	1	模型版本。
DetailTag	String	否	false	文本实体抽取类型模型，是否返回详细抽取结果。可选值：true/false。
TopK	Integer	否	3	关键短语抽取模型，返回top几的结果。
FileType	String	否	docx	文件类型。
FileUrl	String	否	http://127.0.0.1/example.docx	文件HTTP保存地址。
FileContent	String	否	[base64 encode content]	文件内容Base64编码。

返回数据

名称	类型	示例值	描述
RequestId	String	C579D954-E53C-49FD-B987-FAF1A1C5E2A5	请求ID。
AsyncPredictId	Long	123	异步预测唯一ID，用于查询异步预测结果。

示例

请求示例

```
http(s)://[Endpoint]/?Action=CreateAsyncPredict
&Content=自然语言处理
&ModelId=1818
&ModelVersion=1
&<公共请求参数>
```


正常返回示例

XML 格式

```
<CreateAsyncPredictResponse>
<requestId>A357B459-3EA6-4F68-B285-FFDDF4E4662D</requestId>
<asyncPredictId>1634</asyncPredictId>
</CreateAsyncPredictResponse>
```

JSON 格式

```
{
  "requestId": "A357B459-3EA6-4F68-B285-FFDDF4E4662D",
  "asyncPredictId": 1634
}
```

错误码

HttpCode	错误码	错误信息	描述
403	Forbidden	You are not authorized to operate on the specified resource.	用户未被授权操作指定的资源。
503	ServiceUnavailable	An error occurred while processing your request.	服务内部错误。
404	ResourceNotFound	The specified Model does not exist.	指定的模型不存在。
400	InvalidAccountStatus.ServiceUnavailable	The NLP Auto Machine Learning service has not been activated.	用户未开通NLP自学习服务。
400	InvalidAccountStatus.ServiceLimit	The maximum number of free API daily visits is exceeded.	API每天访问次数超过了免费额度限制。
400	InvalidParameter.ModelId	The specified modelId is invalid.	参数modelId不合法。
400	InvalidParameter.ModelVersion	The specified modelVersion is invalid.	参数modelVersion不合法。
400	InvalidContentLength	The length of content exceeds the limit.	内容长度过长。
500	InternalServerError	An error occurred while processing your request.	未知错误。

访问[错误中心](#)查看更多错误码。

1.9. GetAsyncPredict

调用GetAsyncPredict获取异步预测结果。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetAsyncPredict	系统规定参数。取值：GetAsyncPredict。
AsyncPredictId	Integer	是	1629	异步预测ID，通过调用CreateAsyncPredict接口获取。

返回数据

名称	类型	示例值	描述
RequestId	String	EBF1AAD7-A3FB-4A78-B5E9-8356324B12B2	请求ID。
AsyncPredictId	Integer	1629	异步预测ID，通过调用CreateAsyncPredict接口获取。
Status	Integer	2	异步预测状态，状态类型说明：1(处理中)；2(处理完成)；3(处理失败)。
Content	String	json string	异步预测结果，JSON字符串格式文本。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetAsyncPredict
&AsyncPredictId=1629
&<公共请求参数>
```

正常返回示例

XML 格式

```
<GetAsyncPredictResponse>
<requestId>EBF1AAD7-A3FB-4A78-B5E9-8356324B12B2</requestId>
<asyncPredictId>1629</asyncPredictId>
<status>2</status>
<content></content>
</GetAsyncPredictResponse>
```

JSON 格式

```
{
  "requestId": "EBF1AAD7-A3FB-4A78-B5E9-8356324B12B2",
  "asyncPredictId": 1629,
  "status": 2,
  "content": ""
}
```

错误码

HttpCode	错误码	错误信息	描述
403	Forbidden	You are not authorized to operate on the specified resource.	用户未被授权操作指定的资源。
503	ServiceUnavailable	An error occurred while processing your request.	服务内部错误。
404	ResourceNotFound	The specified Model does not exist.	指定的模型不存在。
400	InvalidAccountStatus.ServiceUnavailable	The NLP Auto Machine Learning service has not been activated.	用户未开通NLP自学习服务。
400	InvalidAccountStatus.ServiceLimit	The maximum number of free API daily visits is exceeded.	API每天访问次数超过了免费额度限制。
400	InvalidParameter.ModelId	The specified modelId is invalid.	参数modelId不合法。
400	InvalidParameter.ModelVersion	The specified modelVersion is invalid.	参数modelVersion不合法。
400	InvalidContentLength	The length of content exceeds the limit.	内容长度过长。
500	InternalServerError	An error occurred while processing your request.	未知错误。


访问[错误中心](#)查看更多错误码。

1.10. 异步预测接口使用示例

这个文档介绍如何使用异步预测接口，进行模型预测的异步调用，支持更长文本的离线调用。

创建异步调用

参考以下示例代码，通过 `content` 字段，传入长文本内容，NLP自学习平台会保存长文本，并进行异步模型预测。

 说明 `content` 字段支持的文本长度最大不超过10000字。

```
package com.alibaba.nlp;

import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.nlp_automl.model.v20191111.CreateAsyncPredictRequest;
import com.aliyuncs.nlp_automl.model.v20191111.CreateAsyncPredictResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;

public class NlpAutomlAsync {
    public static void main(String[] args) throws ClientException {
        DefaultProfile profile = DefaultProfile.getProfile(
            "cn-hangzhou",
            "<your-access-key-id>",
            "<your-access-key-secret>");
        IAcsClient client = new DefaultAcsClient(profile);

        CreateAsyncPredictRequest request = new CreateAsyncPredictRequest();
        request.setModelId(1818);
        request.setContent(longContent);
        CreateAsyncPredictResponse response = client.getAcsResponse(request);
        System.out.println("" + new Gson().toJson(response));
    }
}
```

另外，可以通过将本地文件内容进行base64编码之后，作为字符串上传NLP自学习平台系统，NLP自学习平台支持解析base64编码的文件内容。并做文本抽取，进行模型预测调用。以下是参考示例代码。

② 说明 目前支持的文件格式包括：txt、html、pdf、doc、docx。

```
package com.alibaba.nlp;

import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.nlp_automl.model.v20191111.CreateAsyncPredictRequest;
import com.aliyuncs.nlp_automl.model.v20191111.CreateAsyncPredictResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;

import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.Base64;

public class NlpAutomlAsyncDaily {
    public static void main(String[] args) throws ClientException, IOException {
        DefaultProfile profile = DefaultProfile.getProfile(
            "cn-hangzhou",
            "<your-access-key-id>",
            "<your-access-key-secret>");
        IAcsClient client = new DefaultAcsClient(profile);

        InputStream in = new FileInputStream("<local-file-dir>");
        byte[] data = new byte[in.available()];
        int readSize = in.read(data);
        System.out.println("read data size=" + readSize);
        in.close();
        String base64String = Base64.getEncoder().encodeToString(data);
        System.out.println("base64 string length=" + base64String.length());

        CreateAsyncPredictRequest request = new CreateAsyncPredictRequest();
        request.setModelId(2269);
        request.setFileType("<file-type>");
        request.setFileContent(base64String);
        CreateAsyncPredictResponse response = client.getAcsResponse(request);
        System.out.println("" + new Gson().toJson(response));
    }
}
```

调用成功之后，NLP自学习平台返回一个 `asyncPredictId` 字段，用于查询异步预测结果信息。

```
{
  "requestId": "65917545-CDBF-4246-8708-CD03ED4AFDED",
  "asyncPredictId": 1669
}
```

通过使用 `asyncPredictId` 参考下一章节获取异步预测结果。

获取异步调用结果

通过创建 `GetAsyncPredictRequest` 请求，查询异步预测结果。

```
package com.alibaba.nlp;

import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.nlp_automl.model.v20191111.GetAsyncPredictRequest;
import com.aliyuncs.nlp_automl.model.v20191111.GetAsyncPredictResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;

public class NlpAutomlAsync2 {
    public static void main(String[] args) throws ClientException {
        DefaultProfile profile = DefaultProfile.getProfile(
            "cn-hangzhou",
            "<your-access-key-id>",
            "<your-access-key-secret>");
        IAcsClient client = new DefaultAcsClient(profile);

        GetAsyncPredictRequest request = new GetAsyncPredictRequest();
        request.setAsyncPredictId(1669);
        GetAsyncPredictResponse response = client.getAcsResponse(request);
        System.out.println("" + new Gson().toJson(response));
    }
}
```

② 说明 异步预测完成时间，根据传入文本长度和文档大小会增加。所以，需要通过 `GetAsyncPredictResponse` 的 `status` 字段值，进行结果轮询。详细 `status` 枚举值，参考下面内容。

异步预测状态说明

status	说明
0	初始化
1	处理中
2	调用成功
3	调用失败

② 说明 当异步预测的状态 `status` 还没有变成2或者3之前，需要轮询调用 `GetAsyncPredict` 接口查询异步预测结果。

1.11. RunPreTrainService

调用预训练模型服务

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	RunPreTrainService	系统规定参数。取值：RunPreTrainService。
PredictContent	String	是	<pre>{"input": {"content": "这件衣服真好看", "domain": "clothing"}}</pre>	预测输入。
ServiceName	String	是	ReviewAnalysis	服务名称。
ServiceVersion	String	否	V1	服务版本。

返回数据

名称	类型	示例值	描述
PredictResult	String	{ "cost": "21.812ms", "imprs": [{"aspect": "款式", "clause": "这件衣服真好看", "neg_prob": 0.0, "pos_prob": 0.566100001335144, "position": "0,7", "sentiment": "正"}], "message": "SUCCESS", "tracerId": "7973b50e31dcbea8323460e0134d3226"} }	模型预测结果。
RequestId	String	7310FFCF-9FE8-4257-AE38-0E1AD828C2CC	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=RunPreTrainService
&PredictContent={"input":{"content": "这件衣服真好看", "domain": "clothing"}}
&ServiceName=ReviewAnalysis
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RunPreTrainServiceResponse>
<requestId>F67DEBB0-FFD0-4042-8AA9-90A705FB9F94</requestId>
<predictResult>{"code":1000,"data":{"cost":"18.346ms","imprs":[{"aspect":"款式","clause":"这件衣服真好看","neg_prob":0.0,"pos_prob":0.566100001335144,"position":"0,7","sentiment":"正"}],"message":"SUCCESS","tracerId":"32846562fd4aeca63912956bbad18c47"}</predictResult>
</RunPreTrainServiceResponse>
```

JSON 格式

```
{
  "requestId": "F67DEBB0-FFD0-4042-8AA9-90A705FB9F94",
  "predictResult": "{\"code\":1000,\"data\":{\"cost\":\"18.346ms\",\"imprs\":{\"aspect\":\"款式\",\"clause\":\"这件衣服真好看\",\"neg_prob\":0.0,\"pos_prob\":0.566100001335144,\"position\":\"0,7\",\"sentiment\":\"正\"}},\"message\":\"SUCCESS\",\"tracerId\":\"32846562fd4aeca63912956bbad18c47\"}"
}
```

错误码

HttpCode	错误码	错误信息	描述
400	InvalidAccountStatus.ServiceUnavailable	The NLP Auto Machine Learning service has not been activated.	用户未开通NLP自学习服务。
400	InvalidAccountStatus.ServiceLimit	The maximum number of free API daily visits is exceeded.	API每天访问次数超过了免费额度限制。
404	ResourceNotFound	The specified Model does not exist.	指定的模型不存在。
503	ServiceUnavailable	An error occurred while processing your request.	服务内部错误。
400	InvalidServiceName	The service does not exist.	服务不存在。
500	ModelInferenceError	Failed to obtain the model prediction result.	获取模型预测结果失败。
500	UnknownError	A server error occurred while processing your request. Please try again later or contact customer support.	未知错误。
500	InternalError	An error occurred while processing your request.	未知错误。

访问[错误中心](#)查看更多错误码。

1.12. 预训练模型使用教程

NLP自学习平台提供了一些预训练的特定领域模型服务。例如：商品评价解析服务、新闻层次分类服务、中文简历抽取、英文简历抽取。

服务开通与资源包购买

预训练模型使用前，需要确认是否已经开通了：自然语言处理和NLP自学习平台服务。

自然语言处理：[开通地址](#)

NLP自学习平台：[开通地址](#)

🔍 说明 预训练模型提供累计500次的免费调用量，超过500次需要购买资源包：[购买地址](#)

使用NLP自学习平台SDK

引入NLP自学习平台的SDK，查看[最新SDK版本](#)。

```
<dependency>
<groupId>com.aliyun</groupId>
<artifactId>aliyun-java-sdk-core</artifactId>
<version>3.5.0</version>
</dependency>
<dependency>
<groupId>com.aliyun</groupId>
<artifactId>aliyun-java-sdk-nlp-automl</artifactId>
<version>0.0.5</version>
</dependency>
```

API文档参考：[API地址](#)

API测试参考：[API测试](#)

商品评价解析服务

商品评价解析服务使用示例。

Java代码示例

```
DefaultProfile defaultProfile = DefaultProfile.getProfile("cn-hangzhou","your-access-id-key","your-access-id-secret");
IAcsClient client = new DefaultAcsClient(defaultProfile);

Map<String, Object> input = new HashMap();
input.put("content","衣服款式:很好看");
input.put("domain","clothing");
input.put("entity",true);
Map<String, Object> map = new HashMap<>();
map.put("input", input);

RunPreTrainServiceRequest request = new RunPreTrainServiceRequest();
request.setServiceName("ReviewAnalysis");
request.setPredictContent(JSON.toJSONString(map));
RunPreTrainServiceResponse response = client.getAcsResponse(request);
System.out.println(response.getPredictResult());
```

Python代码示例

```
# 安装依赖  
pip install aliyun-python-sdk-core  
pip install aliyun-python-sdk-nlp-automl
```

```
# -*- coding: utf8 -*-
import json

from aliyundkcore.client import AcsClient
from aliyundkcore.acs_exception.exceptions import ClientException
from aliyundkcore.acs_exception.exceptions import ServerException
from aliyundknlp_automl.request.v20191111 import RunPreTrainServiceRequest

# Initialize AcsClient instance
client = AcsClient(
    "<your-access-id-key>",
    "<your-access-id-secret>",
    "cn-hangzhou"
);


content = '这件衣服真好看'
domain = 'clothing'
input_params = {
    "input": {
        "content": content,
        "domain": domain,
        "entity": True
    }
}

# Initialize a request and set parameters
request = RunPreTrainServiceRequest.RunPreTrainServiceRequest()
request.set_ServiceName('ReviewAnalysis')
request.set_PredictContent(json.dumps(input_params))

# Print response
response = client.do_action_with_exception(request)
resp_obj = json.loads(response)
predict_result = json.loads(resp_obj['PredictResult'])
print(json.dumps(predict_result['data']))
```

② 说明 调用商品评价解析服务， `ServiceName` 需要传入：`ReviewAnalysis`。

`PredictContent` 内容示例

 说明 PredictContent参数为JSON字符串传入，JSON字符串包含内容参考下面举例。

```
{
  "input": {
    "content": "衣服款式很好看",
    "domain": "clothing",
    "entity": true
  }
}
```

入参说明

参数	说明
content	需要预测的文本内容。
domain	支持商品类目。参考下一章节的类目支持表格。
entity	值为true时返回结果包含属性情感词抽取结果，否则不包含

商品类目支持表

类目	类目code	属性类别
水产肉类	aquaticmeat	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,重量,颜色,口感,细节,新鲜度,干净度,个头,水分,咸度,成熟度,保质期,吃坏,甜度,手感
咖啡麦片冲饮	coffe	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,工艺,重量,异味/气味,食用效果,颜色,口感,个头,保质期,冲泡效果,便利性
时尚饰品	fashion	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,工艺,功能,异味/气味,外观,颜色,款式,尺码/尺寸,佩戴效果
粮油米面	grainoil	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,重量,体验,颜色,口感,新鲜度,干净度,个头,水分,手感,咸度,成熟度,保质期,吃坏,甜度,细节
保健食品	healthfood	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,重量,异味/气味,食用效果,口感,保质期,冲泡效果,便利性

类目	类目code	属性类别
灯饰光源	light	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,做工,功能,能耗,材质,外观,使用效果,颜色,配件,亮度,体积,安装
办公用品	todo	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,做工,功能,重量,操作性,空间,异味/气味,材质,外观,颜色,显示/打印效果,系统性能,体积,安装,速度,声音
滋补营养品	nourishment	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,异味/气味,重量,耐泡度,颜色,口感,新鲜度,粉质,干净度,个头,保质期,泡发性
电器类	lifeelectric	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,做工,功能,便携性,重量,操作性,外观,使用效果,颜色,体积,安装,便利性,声音,细节
通讯类	lifeelectriccommucation	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,做工,功能,电池/续航时间,重量,对讲/通话效果,操作性,信号,外观,颜色,体积,传播距离,声音,便携性
电热毯类	lifeelectricblanket	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,面料/材质,制热效果,功能,味道,做工,操作性,体验,图案,颜色,尺码/尺寸,安全性,保温效果,声音
电扇类	lifeelectricfan	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,风力/风速,功能,做工,重量,操作性,体验,外观,颜色,体积,安装,声音,细节
酒类	wine_ann	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,工艺,异味/气味,醉酒性,年份,外观,口感,颜色,原料,体积,酒精度数
电视机	television	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,体积,做工,功能,重量,操作性,异味/气味,材质,能耗,内容资源,内存,外观,颜色,配件,设计,系统性能,显示效果,安装,厚度,声音
热水器	waterheat	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,制热效果,做工,重量,操作性,保温/控温/加热效果,材质,出水性,能耗,外观,颜色,容量,设计,安全性,体积,安装,声音

类目	类目code	属性类别
洗衣机	washingmachine	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,脱水/烘干效果,功能,异味/气味,能耗,重量,材质,操作性,安装,外观,颜色,容量,体积,声音/震动,洗涤效果
制冷设备	refrigeration	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,做工,功能,制冷/保鲜效果,重量,操作性,异味/气味,材质,能耗,箱门结构,外观,颜色,容量,体积,安装,工作方式,厚薄度,声音
眼镜行业	glasses	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,做工,材质,颜色,款式,配件,尺码/尺寸,设计,佩戴效果,清晰度
小球类	littleball	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,做工,异味/气味,重量,耐用性,材质,外观,颜色,飞行稳定性,尺码/尺寸,圆度,软硬度,手感,弹性
大球类	bigball_ann	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,控球性,做工,异味/气味,重量,耐用性,材质,外观,颜色,漏气,配件,充气,尺码/尺寸,软硬度,弹性,手感
骑行配饰	cycling	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,透气性,做工,功能,异味/气味,材质,颜色,款式,防水性,保暖性,速干效果,尺码/尺寸,舒适度,色牢度,厚薄度,手感,弹性
手机行业	cellphone	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,功能,拍照效果,电池/续航能力,内存,外观,使用效果,颜色,屏幕,配件,系统性能,声音,手感
箱包行业	bags_ann	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,色差,功能,做工,异味/气味,重量,密封性,材质,颜色,款式,容量,实用性,配件,尺码/尺寸,牢固度,手感
内衣行业	underwear	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,面料/材质,做工,缩水,异味/气味,耐穿性,颜色,款式,保暖性,尺码/尺寸,聚拢效果,舒适度,色牢度,厚薄度,手感,弹性

类目	类目code	属性类别
手表行业	watch	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,做工,异味/气味,操作性,材质,外观,颜色,防水性,校准,佩戴效果
球类配饰	ballaccessory	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,做工,服帖性,重量,异味/气味,防滑效果,声音,材质,外观,颜色,防水性,吸水性,粘性,尺码/尺寸,磁性,安装,清洗效果,使用效果/球感效果,软硬度,弹性,手感
自行车类	bicycle	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,灯光,色差,功能,做工,重量,材质,外观,颜色,尺码/尺寸,设计,安全性,安装,骑行感受,声音,手感
家装主材	decoration	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,做工,功能,制热效果,重量,异味/气味,密封性,材质,出水性,能耗,使用效果,款式,水冲力,尺码/尺寸,设计,安全性,亮度,安装,声音,手感
纸尿裤	diaper	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,吸收度,透气性,柔软度,材质,外观,使用效果,贴身性
奶粉	milk	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,异味/气味,颜色,口感,便利性,保质期,质地,溶解性
零食	snacks	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,辣度,味道,异味/气味,重量,口感,成分,保质期,肉质,甜度
床上用品	beddings	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,做工,功能,缩水,重量,异味/气味,图案,材质,颜色,款式,保暖性,尺码/尺寸,舒适度,色牢度,厚薄度,手感,软硬度
住宅家具	furniture	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,色差,做工,异味/气味,材质,外观,颜色,款式,手感,安全性,舒适度,厚重感,便利性,声音,软硬度
鞋类行业	shoes	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,做工,功能,异味/气味,材质,上身效果,外观,颜色,款式,尺码/尺寸,色牢度

类目	类目code	属性类别
服饰行业	clothing	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,版型,透气性,功能,做工,厚薄度,异味/气味,材质,填充物及含量,上身效果,颜色,款式,保暖性,尺码/尺寸,舒适度,起球,色牢度,面料,弹性,手感
美妆&彩妆行业	beauty	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,美白效果,丰胸效果,使用效果,颜色,抗衰效果,成分,收毛孔效果,遮瑕效果,祛痘祛印效果,隔离效果,补水保湿效果,密封度,防晒效果,纤体效果,紧致效果,质地,祛斑淡斑效果,去黑头效果,异味/气味,妆感,控油效果,清洁效果,保质期
女装类	dress	质量,卖家服务,性价比,物流,正品,包装,整体,赠品,填充物及含量,颜色,款式,尺码/尺寸,起球,色牢度,缩水,保暖性,破洞,面料,弹性,透气性,做工,图案,厚薄度,色差,异味/气味,上身效果,对称,线头,手感

PredictResult 内容示例

② 说明 PredictResult是JSON字符串，需要反序列化之后进行解析，具体结构如下示例。

```
{
  "code": 1000,
  "data": {
    "aspectItem": [
      {
        "aspectCategory": "款式",
        "aspectPolarity": "正",
        "clause": "衣服款式很好看",
        "clauseIndex": "0,7",
        "negativeProb": 0,
        "positiveProb": 0.999,
        "terms": [
          {
            "aspectTerm": "款式",
            "opinionTerm": "好看"
          }
        ]
      }
    ],
    "cost": "13.652ms",
    "textPolarity": "正",
    "textProb": 0.9972
  },
  "message": "SUCCESS",
  "tracerId": "55826e2a-23d0-4664-905f-59010bfecd2a"
}
```

出参说明

字段	说明
textPolarity	整条文本情感极性：正、中、负
textProb	整条文本情感极性置信度(取值范围[0,1]，越大代表整条文本为textPolarity对应情感极性的概率越高)
aspectItem	属性情感列表，每个元素是一个json字段
aspectCategory	属性类别
aspectPolarity	属性片段极性（正、中、负）
clause	情感子句

字段	说明
clauseIndex	情感子句所在的起始位置，终结位置
positiveProb	情感正向置信度(取值范围[0,1]，越大代表情感正向概率越高)
negativeProb	情感负向置信度(取值范围[0,1]，越大代表情感负向概率越高)
terms	抽取的属性情感词列表，每个元素是一个json字段
aspectTerm	属性词
opinionTerm	情感词

新闻层次分类服务

新闻层次分类服务使用示例。

Java代码示例

```
DefaultProfile defaultProfile = DefaultProfile.getProfile("cn-hangzhou","your-access-key-id","your-access-id-secret");
IAcsClient client = new DefaultAcsClient(defaultProfile);
String content = "[\北京时间为4月18日，凯尔特人主场以99-91击退了步行者，总比分2-0领先\,\]" 《大军师司马懿》
近日登陆日本播出，当地网友纷纷称赞该剧制作考究，司马懿诸葛亮曹操等人物出彩，吴秀波、于和伟等演技也获赞\]"
RunPreTrainServiceRequest request = new RunPreTrainServiceRequest();
request.setServiceName("NewsClassification");
request.setPredictContent(content);
RunPreTrainServiceResponse response = client.getAcsResponse(request);
System.out.println(response.getPredictResult());
```

Python代码示例

```
# 安装依赖
pip install aliyun-python-sdk-core
pip install aliyun-python-sdk-nlp-automl
```

```
# -*- coding: utf8 -*-
import json

from aliyundkcore.client import AcsClient
from aliyundkcore.acs_exception.exceptions import ClientException
from aliyundkcore.acs_exception.exceptions import ServerException
from aliyundknlp_automl.request.v20191111 import RunPreTrainServiceRequest

# Initialize AcsClient instance
client = AcsClient(
    "<your-access-key-id>",
    "<your-access-key-secret>",
    "cn-hangzhou"
);

content = []
content.append('北京时间4月18日，凯尔特人主场以99-91击退了步行者，总比分2-0领先')
content.append('《大军师司马懿》近日登陆日本播出，当地网友纷纷称赞该剧制作考究，司马懿诸葛亮曹操等人>物
出彩，吴秀波、于和伟等演技也获赞')

# Initialize a request and set parameters
request = RunPreTrainServiceRequest.RunPreTrainServiceRequest()
request.set_ServiceName('NewsClassification')
request.set_PredictContent(json.dumps(content))

# Print response
response = client.do_action_with_exception(request)
resp_obj = json.loads(response)
predict_result = json.loads(resp_obj['PredictResult'])
print(predict_result['result'])
```

② 说明 调用新闻层次分类服务，`ServiceName` 需要传入：`NewsClassification`。


`PredictContent` 内容示例

② 说明 `PredictContent`参数为JSON字符串传入，JSON字符串包含内容参考下面举例。

```
[  
  "北京时间4月18日，凯尔特人主场以99-91击退了步行者，总比分2-0领先",  
  "《大军师司马懿》近日登陆日本播出，当地网友纷纷称赞该剧制作考究，司马懿诸葛亮曹操等人物出彩，吴秀波、于和伟等演技也获赞"  
]
```

入参可以同时传入多个文本，多个文本需要封装为JSONArray接口，并序列化成Json字符串。

PredictResult 内容示例

 说明 PredictResult是JSON字符串，需要反序列化之后进行解析，具体结构如下示例。

```
{  
  "ret_code": 0,  
  "message": "",  
  "time": 100,  
  "result": [  
    [  
      "news_sports",  
      "news_sports_basketball"  
    ],  
    [  
      "news_entertainment",  
      "news_entertainment_film_tv"  
    ]  
  ]  
}
```

出参 `result` 是一个JSON字符串，是一个JSONArray结构。

简历抽取（中文）

简历抽取服务使用示例。

Java代码示例

```
DefaultProfile defaultProfile = DefaultProfile.getProfile("cn-hangzhou","your-access-key-id","your-access-key-secret");
IAcsClient client = new DefaultAcsClient(defaultProfile);
String content = "简历文本";
RunPreTrainServiceRequest request = new RunPreTrainServiceRequest();
request.setServiceName("ResumeExtractorZH");
request.setPredictContent(content);
RunPreTrainServiceResponse response = client.getAcsResponse(request);
System.out.println(response.getPredictResult());
```

Python代码示例

```
# 安装依赖
pip install aliyun-python-sdk-core
pip install aliyun-python-sdk-nlp-automl
```

```
# -*- coding: utf8 -*-
import json


from aliyundkcore.client import AcsClient
from aliyundkcore.acs_exception.exceptions import ClientException
from aliyundkcore.acs_exception.exceptions import ServerException
from aliyundknlp_automl.request.v20191111 import RunPreTrainServiceRequest

# Initialize AcsClient instance
client = AcsClient(
    "<your-access-key-id>",
    "<your-access-key-secret>",
    "cn-hangzhou"
);

content = '简历文本'

# Initialize a request and set parameters
request = RunPreTrainServiceRequest.RunPreTrainServiceRequest()
request.set_ServiceName('ResumeExtractorZH')
request.set_PredictContent(content)


# Print response
response = client.do_action_with_exception(request)
resp_obj = json.loads(response)
predict_result = json.loads(resp_obj['PredictResult'])
print(predict_result['result'])
```

 **说明** 调用简历抽取(中文)服务, `ServiceName` 需要传入: `ResumeExtractorZH`

PredictContent 内容示例

 **说明** `PredictContent`参数为简历文本参数。

PredictResult 内容示例

 **说明** `PredictResult`是JSON字符串, 需要反序列化之后进行解析, 具体结构如下示例。

```
{
  "ret_code": 0,
```



```
"result": [{
  "id": 0,
  "tags": [{
    "class": "姓名",
    "source": "rule",
    "span": "张维"
  }],
  "sentence": "张维 ID:61969461",
  "sent_offsets": 0
}, {
  "id": 0,
  "tags": [{
    "conf": 0,
    "class": "手机号",
    "span": "186xxxx329",
    "start": 0,
    "source": "model",
    "end": 11
  }],
  "sentence": "18664347329",
  "sent_offsets": 2
}, {
  "id": 0,
  "tags": [{
    "class": "电子邮箱",
    "source": "rule",
    "span": "wwt5150@126.com"
  }],
  "sentence": "wwt5150@126.com",
  "sent_offsets": 3
}, {
  "id": 0,
  "tags": [{
    "conf": 0,
    "class": "出生日期",
    "span": "1990/07/27",
    "start": 8,
    "source": "model",
    "end": 18
  }],
  "class": "性别",
```

```
"source": "rule",
"span": "男"
}],
"sentence": "男|29岁(1990/07/27)|现居住南京-玄武区|8年工作经验",
"sent_offsets": 4
}, {
"source": "model",
"end": 11
}],
"sentence": "职位: 软件测试工程师",
"sent_offsets": 6
}],
"message": "",
"time": 143
}
```

出参 `result` 是一个JSON字符串，是一个JSONArray结构。

简历抽取（英文）

简历抽取服务使用示例。

Java代码示例

```
DefaultProfile defaultProfile = DefaultProfile.getProfile("cn-hangzhou","your-access-key-id","your-access-key-secret");
IAcsClient client = new DefaultAcsClient(defaultProfile);
String content = "Resume Text";
RunPreTrainServiceRequest request = new RunPreTrainServiceRequest();
request.setServiceName("ResumeExtractorEN");
request.setPredictContent(content);
RunPreTrainServiceResponse response = client.getAcsResponse(request);
System.out.println(response.getPredictResult());
```

Python代码示例

```
# 安装依赖
pip install aliyun-python-sdk-core
pip install aliyun-python-sdk-nlp-automl

# -*- coding: utf8 -*-
import json


from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdknlp_automl.request.v20191111 import RunPreTrainServiceRequest

# Initialize AcsClient instance
client = AcsClient(
    "<your-access-key-id>",
    "<your-access-key-secret>",
    "cn-hangzhou"
);

content = 'Resume Text'

# Initialize a request and set parameters
request = RunPreTrainServiceRequest.RunPreTrainServiceRequest()
request.set_ServiceName('ResumeExtractorEN')
request.set_PredictContent(content)

# Print response
response = client.do_action_with_exception(request)
resp_obj = json.loads(response)
predict_result = json.loads(resp_obj['PredictResult'])
print(predict_result['result'])
```

 说明 调用简历抽取(英文)服务, `ServiceName` 需要传入: `ResumeExtractorEN`

`PredictContent` 内容示例

 说明 `PredictContent`参数为简历文本参数。

`PredictResult` 内容示例

🔍 说明 PredictResult是JSON字符串，需要反序列化之后进行解析，具体结构如下示例。

```
{
  "result": "{\"Schools\": [{\"School\": \"The Hong Kong Polytechnic University\", \"Degree\": null, \"StartDate\": \"2002-1-1\", \"EndDate\": \"2004-1-1\", \"DegreeCode\": null, \"SchoolDuration\": \"2002 - 2004\"}, {\"School\": \"University of British Columbia\", \"Degree\": \"Masters\", \"StartDate\": \"2000-3-1\", \"EndDate\": null, \"DegreeCode\": \"master\", \"SchoolDuration\": \"\", Mar}], {\"School\": \"University of British Columbia\", \"Degree\": \"Masters\", \"StartDate\": \"2000-3-1\", \"EndDate\": null, \"DegreeCode\": \"master\", \"SchoolDuration\": \"\", Mar}}\", \"Name\": \"John Smith\", \"Lang\": \"en\", \"Phone\": \"+852 1111 1234\", \"CountryCode\": \"852\", \"RegionalPhone\": \"11111234\", \"FamilyName\": \"Smith\", \"GivenName\": \"John\", \"City\": \"Hong Kong\", \"Email\": \"john.smith@gmail.com\", \"Companies\": [{\"Position\": null, \"EndDate\": \"2012-12-1\", \"Company\": \"Aurostyle Limited\", \"StartDate\": \"2010-6-1\", \"CompanyDuration\": \"June 2010 \u2013 December 2012\"}, {\"Position\": null, \"EndDate\": \"2010-6-1\", \"Company\": \"Waddy Jewellery Group Limited\", \"StartDate\": \"2007-6-1\", \"CompanyDuration\": \"June 2007 \u2013 June 2010\"}]",
  "time": 128,
  "message": "",
  "ret_code": 0
}
```

出参 result 是一个JSON字符串，是一个JSON字符串，具体结构如下所示。

```
{
  "Schools": [{
    "School": "The Hong Kong Polytechnic University",
    "StartDate": "2002-1-1",
    "EndDate": "2004-1-1",
    "SchoolDuration": "2002 - 2004"
  }, {
    "School": "University of British Columbia",
    "Degree": "Masters",
    "StartDate": "2000-3-1",
    "DegreeCode": "master",
    "SchoolDuration": "", Mar"
  }, {
    "School": "University of British Columbia",
    "Degree": "Masters",
    "StartDate": "2000-3-1",
    "DegreeCode": "master",
    "SchoolDuration": "", Mar"
  }
}
```

```
}},
  "Name": "John Smith",
  "Lang": "en",
  "Phone": "+852 1111 1234",
  "CountryCode": "852",
  "RegionalPhone": "11111234",
  "FamilyName": "Smith",
  "GivenName": "John",
  "City": "Hong Kong",
  "Email": "john.smith@gmail.com",
  "Companies": [{
    "Position": null,
    "EndDate": "2012-12-1",
    "Company": "Aurostyle Limited",
    "StartDate": "2010-6-1",
    "CompanyDuration": "June 2010 \u2013 December 2012"
  }, {
    "Position": null,
    "EndDate": "2010-6-1",
    "Company": "Waddy Jewellery Group Limited",
    "StartDate": "2007-6-1",
    "CompanyDuration": "June 2007 \u2013 June 2010"
  }
]
```

裁判文书抽取

裁判文书抽取使用示例。

Java代码示例

```
DefaultProfile defaultProfile = DefaultProfile.getProfile("cn-hangzhou","your-access-id-key","your-access-id-secret");
IAcsClient client = new DefaultAcsClient(defaultProfile);
String content = "{\"reqId\":\"xxx\",\"cause\":\"故意伤害罪\",\"content\":\"文书正文，一定要保留换行符 \\n\",\"services\":{\"normal\",\"liangxing\"}";
RunPreTrainServiceRequest request = new RunPreTrainServiceRequest();
request.setServiceName("LegalDocumentExtraction");
request.setPredictContent(content);
RunPreTrainServiceResponse response = client.getAcsResponse(request);
System.out.println(response.getPredictResult());
```

Python代码示例

```
# 安装依赖
pip install aliyun-python-sdk-core
pip install aliyun-python-sdk-nlp-automl

# -*- coding: utf8 -*-
import json

from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdknlp_automl.request.v20191111 import RunPreTrainServiceRequest

# Initialize AcsClient instance
client = AcsClient(
    "<your-access-id-key>",
    "<your-access-id-secret>",
    "cn-hangzhou"
);

input_params = {
    "reqId": "xxx",
    "cause": ["故意伤害罪"],
    "content": "文书正文",
    "services":["normal","liangxing"],
}

# Initialize a request and set parameters
request = RunPreTrainServiceRequest.RunPreTrainServiceRequest()
request.set_ServiceName('LegalDocumentExtraction')
request.set_PredictContent(json.dumps(input_params))

# Print response
response = client.do_action_with_exception(request)
resp_obj = json.loads(response)
predict_result = json.loads(resp_obj['PredictResult'])
print(json.dumps(predict_result['data']))
```

② 说明 裁判文书抽取服务, `ServiceName` 需要传入: `LegalDocumentExtraction`

PredictContent 内容示例

② 说明 PredictContent参数为JSON字符串传入, JSON字符串包含内容参考下面举例。

```
{
  "reqId": "xxx",
  "cause": ["故意伤害罪"],
  "content": "文书正文",
  "services":["normal","liangxing"]
}
```

入参说明

参数	说明
reqId	自定义请求ID。
cause	案由。
content	文书正文内容, string
services	选择服务类型。normal:标准字段解析;liangxing:量刑字段解析

② 说明 案由支持多案由, 传入JSONArray,不知道传空数组'[]'

② 说明 服务类型可以单独解析输出也可以合并解析输出

② 说明 支持以上10个案由的文书解析, 输入的案由如果不在列表中, 算法会根据自己判断输出刑事或者民事的通用字段, 暂时不支持行政类别的字段

② 说明 优先使用用户指定的案由名称, 只有当输入案由不在支持列表中或者为空时才会使用算法判断案由

案由枚举值

案由
交通肇事罪
寻衅滋事罪

案由
故意伤害罪
盗窃罪
开设赌场罪
走私、贩卖、运输、制造毒品罪
危险驾驶罪
诈骗罪
民间借贷纠纷
公益诉讼

PredictResult 内容示例

参数	说明
reqId	请求id, string, 与输入对应
result	结果结构体, json

? 说明 result结构如下


参数	说明
extract	代表信息抽取内容
laws	代表从文书中抽取的法条内容
segments	代表具体裁判文书分段内容

? 说明 segments枚举值如下所示

segments
诉讼参与者
法官助理
第三人陈述
原审情况（重审情况）
诉请

segments
反诉
标题
法律效力
原审情况（再审情况）
事实认定
原审情况（一审辩称）
原审情况（反诉辩称）
裁判结果
案号
裁判日期
原审情况（二审判决结果）
反诉辩称
原审情况（一审事实认定）
原审情况（一审第三人陈述）
原审情况（二审诉请）
审理经过
抗诉
被诉行政行为情况
原审情况（一审裁判结果）
法院
原审情况（二审第三人陈述）
其他
辩称
证据段
原审情况（二审辩称）
文书类型

segments
审判人员
原审情况(一审诉请)
原审情况(反诉)
原审情况(二审事实认定)
书记员
附件
原审情况(抗诉情况)

 **说明** PredictResult是JSON字符串，需要反序列化之后进行解析，具体结构如下示例。

```
{
  "reqId": "003",
  "result": {
    "extract": {
      "交通肇事罪-是否为逃避法律追究逃离事故现场": "否",
      "交通肇事罪-是否因逃逸致人死亡": "否",
      "交通肇事罪-是否存在肇事逃逸情形": "否",
      "交通肇事罪-是否存在超载情形": "否",
      "交通肇事罪-是否存在酒驾、毒驾情形": "否",
      "交通肇事罪-是否无驾驶资格驾驶": "否",
      "交通肇事罪-是否是代驾": "否",
      "交通肇事罪-是否致人死亡": "是",
      "交通肇事罪-是否致人重伤": "否",
      "交通肇事罪-是否致公私财产遭受重大损失": "否",
      "交通肇事罪-是否获得受害人或家属谅解": "否",
      "交通肇事罪-是否被交通肇事罪或交通肇事罪处置过": "否",
      "交通肇事罪-是否购买保险": "否",
      "交通肇事罪-是否适用非监禁刑罚": "是",
      "交通肇事罪-肇事车辆是否符合国家机动车安全技术标准": "是",
      "交通肇事罪-被告对公诉机关指控是否有异议": "否",
      "交通肇事罪-赔偿受害人或家属金额": [
        "238000元"
      ],
      "交通肇事罪-驾驶人驾驶的是否是校车": "否",
      "刑事通用-公诉机关名称": [
        "河北省高碑店市人民检察院"
      ]
    }
  }
}
```

```
],
"刑事通用-判决依据法律条文": [
"《中华人民共和国刑法》第七十三条第二",
"《中华人民共和国刑法》第七十二条第一款",
"《中华人民共和国刑法》第一百三十三条",
"《中华人民共和国刑法》第六十七条"
],
"刑事通用-判决时间": [
"2014年10月16日"
],
"刑事通用-制作单位": [
"河北省高碑店市人民法院"
],
"刑事通用-审判组织": "独任审判",
"刑事通用-庭审方式": "公开审理",
"刑事通用-文书名称": [
"刑事判决书"
],
"刑事通用-文书种类": "判决书",
"刑事通用-文号": [
"(2014)高刑初字第1222号"
],
"刑事通用-案件审理阶段": "一审",
"刑事通用-案发地点": [
"沿老112国道由北向南行驶至高碑店市高静园公墓路口处"
],
"刑事通用-案发时间": [
"2013年12月31日7时40分"
],
"刑事通用-被告信息": [
{
"事故责任认定": "主要责任",
"刑期": "有期徒刑一年一个月, 缓刑二年",
"是否人大代表、政协委员": "否",
"是否为民主党派": "否",
"是否为涉台人员": "否",
"是否为涉外人员": "否",
"是否为涉港澳人员": "否",
"是否为累犯": "否",
"是否有从轻或减轻情节": "是",
"是否有从重情节": "否"
}
```

```

    "是否系从犯": "否",
    "是否系主犯": "否",
    "是否系未成年人": "否",
    "是否系首要分子": "否",
    "构成罪名": [
    "交通肇事罪"
    ],
    "特殊身份信息": [],
    "罚金-具体金额": "",
    "罚金-是否缴清": "",
    "被告出生日期": "",
    "被告别名": "",
    "被告姓名": "刘某",
    "被告户籍所在地": "",
    "被告文化水平": "",
    "被告民族": "",
    "被告现居住地": "",
    "被告籍贯": "",
    "被告职业": "农民",
    "起诉罪名": [
    "交通肇事罪"
    ]
  }
],
"刑事通用-被告触犯法律条文": [],
"刑事通用-起诉书文号": [
"高检公诉刑诉（2014）240号"
]
},
"laws": [
{
"lawName": "中华人民共和国刑法",
"provisionFullName": "中华人民共和国刑法第一百三十三条",
"provisionID": 133,
"provisionName": "一百三十三"
},
{
"lawName": "中华人民共和国刑法",
"provisionFullName": "中华人民共和国刑法第六十七条",
"provisionID": 67,

```

```
"provisionName": "六十七"
},
{
"lawName": "中华人民共和国刑法",
"provisionFullName": "中华人民共和国刑法第七十二条第一款",
"provisionID": 72,
"provisionName": "七十二"
},
{
"lawName": "中华人民共和国刑法",
"provisionFullName": "中华人民共和国刑法第七十三条第二",
"provisionID": 73,
"provisionName": "七十三"
}
],
"segments": {
"书记员": "书记员刘秀坤",
"事实认定": "另查明, 案发后, 双方达成协议, 由被告人的亲属代其一次性赔偿被害人家人各项经济损失共计人民币238000元。上述事实, 被告人在开庭审理过程中亦无异议, 并有道路交通事故现场图、现场勘验笔录及照片、法医学尸体检验意见书、车辆检验意见书、道路交通事故认定书、机动车驾驶证查询结果、赔偿调解书及收条、被告人的户籍证明等证据证实, 足以认定。",
"审判人员": "审判员刘利刚",
"审理经过": "高碑店市人民检察院以高检公诉刑诉(2014)240号起诉书指控被告人刘某犯交通肇事罪, 向本院提起公诉。本院依法适用简易程序, 实行独任审判, 公开开庭审理了本案, 高碑店市人民检察院检察员张志英、代理检察员翟静出庭支持公诉, 被告人刘某到庭参加诉讼。现已审理终结。",
"文书类型": "刑事判决书",
"案号": "(2014)高刑初字第1222号",
"法律效力": "如不服本判决, 可于接到判决书的第二日起十日内, 通过本院或者直接向河北省保定市中级人民法院提出上诉。书面上诉的, 应提交上诉状正本一份, 副本二份。",
"法院": "河北省高碑店市人民法院",
"裁判日期": "二〇一四年十月十六日",
"裁判结果": "本院认为, 被告人刘某违反交通运输管理法规, 因而发生重大事故致使一人死亡, 负事故的主要责任, 其行为构成交通肇事罪, 公诉机关指控的罪名成立。事故发生后, 被告人刘某拨打报警电话, 并在现场等待, 公安交警部门到达现场后, 其如实供述自己罪行, 系自首, 依法可予以从轻处罚。鉴于被告人刘某认罪悔罪, 事发后, 其家人代其对被害人的亲属进行了经济赔偿, 量刑时亦可酌情从轻处罚并适用缓刑。依照《中华人民共和国刑法》第一百三十三条、第六十七条、第七十二条第一款、第七十三条第二、三款之规定, 判决如下: 被告人刘某犯交通肇事罪, 判处有期徒刑一年一个月, 缓刑二年。(缓刑考验期, 自判决确定之日起计算)",
"诉讼参与人": "公诉机关河北省高碑店市人民检察院。被告人刘某, 农民。因本案于2014年1月18日被高碑店市公安局刑事拘留, 同年1月26日被该局监视居住, 7月26日被取保候审。",
"诉请": "公诉机关指控, 2013年12月31日7时40分许, 被告人刘某驾驶车牌为冀Fxxxxx号的银灰色捷达轿车, 沿老112国道由北向南行驶至高碑店市高静园公墓路口处时, 与由西向北转弯上公路的骑行电动自行车的被害人王某相撞, 致
```

```
使王某受伤经抢救无效死亡，两车不同程度损坏。事故发生后，刘某拨打了120急救电话和122报警电话，现场等候交警人员处理事故。经法医鉴定，王某符合交通事故致颅脑损伤合并失血性休克死亡。经高碑店市公安交通警察大队认定，被告人刘某负此事故的主要责任，王某负此事故的次要责任。"
```

```
}  
}  
}
```

招标中标信息抽取

招标中标信息抽取服务示例

java代码示例

```
DefaultProfile defaultProfile = DefaultProfile.getProfile("cn-hangzhou","your-access-key-id","your-access-key-secret");  
IAcsClient client = new DefaultAcsClient(defaultProfile);  
String content = "招标文件文本";  
RunPreTrainServiceRequest request = new RunPreTrainServiceRequest();  
request.setServiceName("NLP-NER-ZhaoBiao-Pretrain");  
request.setPredictContent(content);  
RunPreTrainServiceResponse response = client.getAcsResponse(request);  
System.out.println(response.getPredictResult());
```

Python代码示例

```
# 安装依赖  
pip install aliyun-python-sdk-core  
pip install aliyun-python-sdk-nlp-automl
```

```
# -*- coding: utf8 -*-
import json


from aliyundkcore.client import AcsClient
from aliyundkcore.acs_exception.exceptions import ClientException
from aliyundkcore.acs_exception.exceptions import ServerException
from aliyundknlp_automl.request.v20191111 import RunPreTrainServiceRequest

# Initialize AcsClient instance
client = AcsClient(
    "<your-access-key-id>",
    "<your-access-key-secret>",
    "cn-hangzhou"
);

content = '招标文件文本'

# Initialize a request and set parameters
request = RunPreTrainServiceRequest.RunPreTrainServiceRequest()
request.set_ServiceName('NLP-NER-ZhaoBiao-Pretrain')
request.set_PredictContent(content)


# Print response
response = client.do_action_with_exception(request)
resp_obj = json.loads(response)
predict_result = json.loads(resp_obj['PredictResult'])
print(predict_result['result'])
```

 **说明** 调用招标中标信息抽取服务，`ServiceName` 需要传入：`NLP-NER-ZhaoBiao-Pretrain`

PredictContent 内容示例

 **说明** `PredictContent`参数为文书文本参数。

PredictResult 内容示例

 **说明** `PredictResult`是JSON字符串，需要反序列化之后进行解析。

2.SDK示例

2.1. SDK概述

本文介绍了NLP自学习平台提供的SDK语言版本，列举了最新版本SDK的获取地址。

SDK简介

NLP自学习平台SDK封装了2019-11-11版本API，以访问密钥（AccessKey）识别调用者身份，提供自动签名等功能，方便您通过API创建和管理资源。

使用NLP自学习平台功能需要您同时安装NLP自学习平台SDK和阿里云核心库。以Java SDK为例，您需要下载aliyun-java-sdk-core以及aliyun-java-sdk-nlp-automl。

SDK列表

下表提供了NLP自学习平台支持的SDK列表，您可以在GitHub仓库查看SDK更新历史、获取安装包以及查看指导文档。

编程语言	GitHub地址	GitHub文档
Java	aliyun-java-sdk-nlp-automl	README-CN.md
Python	aliyun-python-sdk-nlp-automl	README_zh.md
.Net	aliyun-net-sdk-nlp-automl	README-CN.md
C++	aliyun-openapi-cpp-sdk	README-CN.md
PHP	openapi-sdk-php	README-zh-CN.md
Go	alibaba-cloud-sdk-go	README-CN.md
Nodejs	aliyun-openapi-nodejs-sdk	README-CN.md
Ruby	openapi-core-ruby-sdk	README-CN.md

2.2. Java SDK

获取SDK

在pom.xml文件中引用sdk的maven依赖。

```
<dependency>
<groupId>com.aliyun</groupId>
<artifactId>aliyun-java-sdk-nlp-automl</artifactId>
<version>查询最新版本</version>
</dependency>
```

使用示例

SDK使用方法举例。

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.nlp_automl.model.v20191111.GetPredictResultRequest;
import com.aliyuncs.nlp_automl.model.v20191111.GetPredictResultResponse;
import com.aliyuncs.profile.DefaultProfile;

public class NlpAutomlTestApp {
    public static void main(String[] args) throws ClientException {
        DefaultProfile profile = DefaultProfile.getProfile(
            "cn-hangzhou",
            "<your-access-key-id>",
            "<your-access-key-secret>");
        IAcsClient client = new DefaultAcsClient(profile);

        GetPredictResultRequest request = new GetPredictResultRequest();
        request.setModelId(1818);
        request.setContent("中华人民共和国人民解放军");
        GetPredictResultResponse response = client.getAcsResponse(request);
        System.out.println(response.getContent());
    }
}
```

执行结果

不同模型类型返回结果会有不同。

```
{
  "requestId": "85A1B6AC-DB09-4623-B8BC-10A9F24DB9F7",
  "content": "[\"人民解放军\", \"中华人民共和国\"]"
}
```

2.3. Python SDK

获取SDK

```
pip install aliyun-python-sdk-core # 安装阿里云 SDK 核心库
pip install aliyun-python-sdk-nlp-automl # 安装NLP自学习平台SDK
```

使用示例

```
# -*- coding: utf8 -*- import json
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdknlp_automl.request.v20191111 import GetPredictResultRequest

# Initialize AcsClient instance
client = AcsClient(
    "<your-access-id-key>",
    "<your-access-id-secret>",
    "cn-hangzhou"
);

# Initialize a request and set parameters
request = GetPredictResultRequest.GetPredictResultRequest()
request.set_ModelId(1818)
request.set_Content('中华人民共和国')

# Print response
response = client.do_action_with_exception(request)
resp_obj = json.loads(response)
print(resp_obj['Content'])
```

2.4. C# SDK

获取SDK

```
dotnet add package aliyun-net-sdk-nlp-automl
```

使用示例

```
using Aliyun.Acs.Core;
using Aliyun.Acs.Core.Profile;
using Aliyun.Acs.Core.Exceptions;
using Aliyun.Acs.Nlp_automl.Model.V20191111;

class Program
{
    static void Main(string[] args)
    {
        // 构建一个 Client, 用于发起请求
        IClientProfile profile = DefaultProfile.GetProfile(
            "<your-region-id>",
            "<your-access-key-id>",
            "<your-access-key-secret>");
        DefaultAcsClient client = new DefaultAcsClient(profile);

        try
        {
            // 构造请求
            GetPredictResultRequest request = new GetPredictResultRequest();
            request.modelId = 1818;
            request.content = "中华人民共和国人民解放军";

            // 发起请求, 并得到 Response
            GetPredictResultResponse response = client.GetAcsResponse(request);
            System.Console.WriteLine(response.content);
        }
        catch (ServerException ex)
        {
            System.Console.WriteLine(ex.ToString());
        }
        catch (ClientException ex)
        {
            System.Console.WriteLine(ex.ToString());
        }
    }
}
```

2.5. C++ SDK

获取SDK

```
git clone https://github.com/aliyun/aliyun-openapi-cpp-sdk.git
```

使用示例

```
#include <iostream>
#include <alibabacloud/core/AlibabaCloud.h>
#include <alibabacloud/nlp-automl/Nlp_automlClient.h>

using namespace AlibabaCloud;
using namespace AlibabaCloud::Nlp_automl;

int main(int argc, char** argv) {
    // 初始化 SDK
    AlibabaCloud::InitializeSdk();

    // 配置 nlp_automl 实例
    ClientConfiguration configuration("<your-region-id>");
    Nlp_automlClient client("<your-access-key-id>", "<your-access-key-secret>", configuration);

    // 创建API请求并设置参数
    Model::GetPredictResultRequest request;
    request.setModelId(1818);
    request.setContent("中华人民共和国解放军");

    auto outcome = client.getPredictResult(request);
    if (!outcome.isSuccess()) {
        // 异常处理
        std::cout << outcome.error().errorCode() << std::endl;
        AlibabaCloud::ShutdownSdk();
        return -1;
    }

    std::cout << "totalCount: " << outcome.result().getTotalCount() << std::endl;

    // 关闭 SDK
    AlibabaCloud::ShutdownSdk();
    return 0;
}
```

2.6. PHP SDK

获取SDK

```
composer require alibabacloud/sdk
```

使用示例

```
<?php

use AlibabaCloud\Client\AlibabaCloud;
use AlibabaCloud\Client\Exception\ClientException;
use AlibabaCloud\Client\Exception\ServerException;
use AlibabaCloud\NlpAutoml\NlpAutoml;

// Set up a global client
AlibabaCloud::accessKeyClient('foo', 'bar')
->regionId('cn-hangzhou')
->asDefaultClient();

try {

// Access product APIs
$request = NlpAutoml::v20191111()->getPredictResult();

// Set options/parameters and execute request
$result = $request->withModelId(1818) // API parameter
->withContent('中华人民共和国解放军') // API parameter
->client('client1') // Specify the client for send
->debug(true) // Enable the debug will output detailed information
->connectTimeout(0.01) // Throw an exception when Connection timeout
->timeout(0.01) // Throw an exception when timeout
->request(); // Execution request

} catch (ClientException $exception) {
echo $exception->getMessage(). PHP_EOL;
} catch (ServerException $exception) {
echo $exception->getMessage() . PHP_EOL;
echo $exception->getErrorCode(). PHP_EOL;
echo $exception->getRequestId(). PHP_EOL;
echo $exception->getErrorMessage(). PHP_EOL;
}
}
```

2.7. Go SDK

获取SDK

```
go get -u github.com/aliyun/alibaba-cloud-sdk-go/sdk
```

创建客户端

```
package main

import "github.com/aliyun/alibaba-cloud-sdk-go/sdk"

func main() {

client, err := sdk.NewClientWithAccessKey("REGION_ID", "ACCESS_KEY_ID", "ACCESS_KEY_SECRET")
if err != nil {
// Handle exceptions
panic(err)
}
}
```

发起请求

```
package main

import "github.com/aliyun/alibaba-cloud-sdk-go/sdk/requests"

func main() {
request := requests.NewCommonRequest() // 构造一个公共请求
request.Method = "POST" // 设置请求方式
request.Product = "nlp-automl" // 指定产品
request.Domain = "nlp-automl.cn-hangzhou.aliyuncs.com" // 指定域名则不会寻址，如认证方式为 Bearer Token 的服务则需要指定
request.Version = "2019-11-11" // 指定产品版本
request.ApiName = "GetPredictResult" // 指定接口名
request.QueryParams["ModelId"] = 1818 // 设置参数值
request.QueryParams["Content"] = "中华人民共和国解放军" // 设置参数值
request.TransToAcsRequest() // 把公共请求转化为acs请求
}
```

2.8. Nodejs SDK

获取SDK

```
npm install @alicloud/nlp-automl-2019-11-11
```

使用示例

```
const NlpAutomlClient = require('@alicloud/nlp-automl-2019-11-11');

async function demo() {
  const client = new NlpAutomlClient({
    endpoint: 'nlp-automl.cn-hangzhou.aliyuncs.com',
    accessKeyId: 'your access key id',
    accessKeySecret: 'your access key secret'
  });

  const res = await client.getPredictResult({
    ModelId: 1818,
    Content: '中华人民共和国解放军'
  });
}

demo();
```

2.9. Ruby SDK

获取SDK

```
gem install aliyunsdkcore
```

使用示例


```
require 'aliyun-sdk-core'

client = RPCClient.new(
  endpoint: 'https://nlp-automl.cn-hangzhou.aliyuncs.com',
  api_version: '2019-11-11',
  access_key_id: ENV['ACCESS_KEY_ID'],
  access_key_secret: ENV['ACCESS_KEY_SECRET'],
)

request_option = { method: 'POST', timeout: 15000 }
response = client.request(
  action: 'GetPredictResult',
  params: {
    "ModelId": "1818",
    "Content": "中华人民共和国解放军"
  },
  opts: request_option
)

print response
```