ALIBABA CLOUD

阿里云

自然语言处理
NLP自学习平台

文档版本：20220711

阿里云

法律声明

阿里云提醒您在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**。
`Courier字体`	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.最新动态 06
2.产品介绍 20
3.产品价格 24
4.私有化部署 27
5.API参考 28
5.1. API概览 28
5.2. 调用方式 28
5.3. API使用问题排查 29
5.4. 签名机制 29
5.5. 公共参数 31
5.6. GetPredictResult 33
5.7. 获取AccessKey 35
5.8. GetPredictResult 37
5.9. 获取模型预测结果接口示例 39
5.10. CreateAsyncPredict 48
5.11. GetAsyncPredict 50
5.12. 异步预测接口使用示例 52
5.13. RunPreTrainService 55
5.14. 预训练模型使用教程 58
5.15. 控制台API操作 63
5.16. 招标中标信息抽取服务 81
6.SDK示例 84
6.1. SDK概述 84
6.2. Java SDK 84
6.3. Python SDK 85
6.4. C# SDK 86

6.5. C++ SDK 87
6.6. PHP SDK 88
6.7. Go SDK 89
6.8. Nodejs SDK 90
6.9. Ruby SDK 90

1.最新动态

2022年06月

项目类型	功能名称	功能描述	发布时间	相关文档
商品评价解析-增量训练	自学习模型	支持用户在平台商品评价解析预训练模型（电商领域、汽车领域、本地生活领域）的基础上，自定义需要增加的标签，用户只需要训练增加的标签，即可得到一个完整的解析模型；	2022-06-17	商品评价解析-增量训练

2022年05月

项目类型	功能名称	功能描述	发布时间	相关文档
电销场景主被叫意图识别	预训练模型	适用于电话销售外呼场景，针对对话内容，识别主叫意图（营销、通知、催收等）和被叫意图（不方便、情感倾向、交流意愿等），可应用于语音质检。	2022-05-06	电销场景主被叫意图识别
实体抽取	规则引擎升级	模型发布后支持新增、修改规则，无需重新训练模型。	2022-05-06	规则引擎与高级设置

2022年03月

项目类型	功能名称	功能描述	发布时间	相关文档
实体抽取	自学习模型	StructBert系列模型推理加速，平均rt下降45%。平均长度2k字时，平均rt约3s。	2022-03-14	模型说明
文本分类	自学习模型	新增StructBet实现。	2022-03-18	模型说明
对话分类	自学习模型	新增高精度版（StructBet实现）。	2022-03-18	模型说明

2021年12月

项目类型	功能名称	功能描述	发布时间	相关文档
商品评价购买决策分析-汽车领域	预训练模型	用于分析用户的购买动机、使用场景、功能需求、使用疑问等购买决策相关的信息，可以帮助改进产品、改善用户体验、细分人群画像、针对性营销投放等，包含25种标签。	2021-12-09	商品评价购买决策分析-汽车领域
商品评价解析服务-汽车领域	预训练模型	适用于汽车领域的评价解析服务，包含71种属性标签。详见右侧文档。	2021-12-09	商品评价解析服务-汽车领域
实体抽取	自学习模型	新增中文StructBert-CRF模型，适用于标签之间依赖较强的数据集。	2021-12-03	模型说明

2021年11月

项目类型	功能名称	功能描述	发布时间	相关文档
商品评价购买决策分析-电商领域	预训练模型	用于分析用户的购买动机、使用场景、功能需求、使用疑问等购买决策相关的信息，可以帮助改进产品、改善用户体验、细分人群画像、针对性营销投放等。	2021-11-24	商品评价购买决策分析-电商领域
实体抽取	自学习模型	新增中文StructBert：基于阿里自研的Struct BERT，从大量无标注语料进行预训练的模型并且蒸馏后的模型，适合标注语料不足的中文任务。针对实体重叠问题进行了优化。	2021-11-19	模型说明
我的模型页	控制台迭代更新	新增【我的模型】页面：支持查询已发布的自学习模型和调用预训练模型，支持查看已购模型数量、剩余资源包余额显示、延长模型有效期、变配模型个数。	2021-11-19	/

2021年10月

项目类型	功能名称	功能描述	发布时间	相关文档

项目类型	功能名称	功能描述	发布时间	相关文档
商品评价解析-电商领域	预训练模型升级	新增6个行业：汽车用品、节庆用品、3C数码配件、五金工具、宠物、鲜花绿植等，更新6个已有行业（增加属性类别）。详见右侧文档	2021-10-19	商品评价解析服务使用教程
对话分类	自学习模型	对整个对话文本按照内容类型进行分类，典型应用场景：对话质检、客户意图识别、电销线索挖掘等。详见右侧文档	2021-10-12	对话文本分类

2021年9月

项目类型	功能名称	功能描述	发布时间	相关文档
文档结构化-KV信息抽取	预训练模型	适用于从文档中抽取符合key：value模式的信息，典型应用场景如简历文档、合同文档、报告文档等，对具备显著KV信息模式的文档抽取效果较好。详见右侧文档	2021-09-23	文档结构化-KV信息抽取
合同要素抽取-通用领域	预训练模型	针对合同中的常见要素进行抽取，共支持26个通用要素字段。详见右侧文档	2021-09-07	合同要素抽取-通用领域服务
合同要素抽取	自学习模型	针对合同抽取场景(甲方、乙方、时间等)定制开发，抽取合同中具有特定意义或者关键的要素。详见右侧文档	2021-09-01	合同要素抽取使用教程

2021年8月

项目类型	功能名称	功能描述	发布时间	相关文档
招标中标信息抽取-高级版	预训练模型升级	中标模型新增招标代理机构、项目负责人信息等7个字段，当前共支持抽取36个字段。详见右侧文档	2021-08-01	招标中标信息抽取-高级版服务
商品评价解析-电商领域	预训练模型升级	新增：影音电器、厨房电器、厨房/烹饪用具3个行业，7个行业属性类别有增加。详见右侧文档	2021-08-05	商品评价解析服务使用教程

2021年7月

项目类型	功能名称	功能描述	发布时间	相关文档
商品评价解析-高精度版	自学习模型	基于Bert实现，训练和预测速度较慢，精度较高。计算资源要求较低，适合训练数据量较大的情况。详见右侧文档	2021-07-07	模型说明
商品评价解析-电商领域	预训练模型升级	新增：清洁工具、个人护理、家居饰品、居家日用、居家布艺、孕产妇用品、收纳整理、餐饮具、玩具9个行业，6个行业属性类别有增加。详见右侧文档	2021-07-12	商品评价解析服务使用教程
双句文本分类	自学习模型	对两句文本按照内容进行分类，同时支持单标签和多标签，包含高精度版和高性能版。典型应用场景：计算两句话是否语义相等、计算问句和回答是否匹配、考虑上下文的单句分类等。详见右侧文档	2021-07-16	双句文本分类
标注功能升级	前端体验优化	新增前一条后一条等按钮、支持标注过程中修改题目、更清晰的交互、更直观的数据展示、其他体验优化等	2021-07-28	/

2021年6月

项目类型	功能名称	功能描述	发布时间	相关文档
招标中标类型分类服务	预训练模型	适用于作为招标解析服务（高级版）、中标解析服务（高级版）的前处理步骤，区分公告类型。详见右侧文档	2021-06-08	招标中标公告类型分类服务
招标中标信息抽取-高级版服务	预训练模型	高级版对比基础版，支持更多字段，更高精度。详见右侧文档	2021-06-08	招标中标信息抽取-高级版服务

2021年5月

项目类型	功能名称	功能描述	发布时间	相关文档
商品评价解析-电商领域	预训练模型升级	新增：母婴类、图书、服饰类-服饰配件、户外、运动、文具、洗护清洁7个行业，20个行业属性类别有增加。详见右侧文档	2021-05-20	商品评价解析服务使用教程
预置数据集	体验优化	为文本分类、实体抽取、短文本匹配、关系抽取等自学习算法模块预置测试数据集，帮助用户快速上手。	2021-05-24	/
实体抽取	自学习模型升级	规则引擎升级，支持规则组合、与或关系、支持和模型抽取结果结合的复杂表示；支持规则效果预览；效率远高于旧版本。详见右侧文档。	2021-05-24	规则引擎说明

2021年4月

项目类型	功能名称	功能描述	发布时间	相关文档
情绪识别服务	预训练模型升级	新增高精度版，详见右侧文档	2021-04-12	情绪识别服务使用教程

2021年3月

项目类型	功能名称	功能描述	发布时间	相关文档
电销场景对话疑似诈骗识别	预训练模型	适用于电话销售外呼场景，针对对话内容，识别疑似诈骗风险，可应用于语音质检。详见右侧文档	2021-03-30	电销场景对话疑似诈骗识别服务使用教程
商品评价解析服务-本地生活领域	预训练模型	适用于本地生活领域的评价解析服务，当前支持美容美发美甲行业、餐饮美食行业。详见右侧文档	2021-03-29	商品评价解析服务-本地生活领域
情绪识别服务	预训练模型升级	优化了正面情绪的识别效果，新增三种业务常用情绪（投诉、感谢、抱怨），详见右侧文档	2021-03-25	情绪识别服务使用教程

项目类型	功能名称	功能描述	发布时间	相关文档
电销场景对话风险识别	预训练模型	适用于电话销售外呼场景，针对对话内容，识别风险（辱骂、投诉、威胁等），可应用于语音质检。详见右侧文档	2021-03-24	电销场景对话风险识别服务
招标文书解析	自学习模型	上线场景行业算法-招标文书解析模型，支持平台模型测试、定制用户模型。	2021-03-24	
实体抽取	前端体验优化	支持excel方式批量上传待标注数据	2021-03-04	/
文本分类	前端体验优化	标注过程中支持修改题目；支持通过文件上传题目；优化多标签数据集统计结果展示效果	2021-03-04	/

2021年2月

项目类型	功能名称	功能描述	发布时间	相关文档
商品评价解析-电商领域	预训练模型升级	新增行业code='all'，返回全行业标签，支持客户按照需求筛选。详见右侧文档	2021-02-19	商品评价解析服务使用教程
文本分类	预训练模型	测试界面支持上传文件批量预测	2021-01-31	/
所有	前端体验优化	训练阶段，可删除指定版本模型、取消发布等	2021-02-01	/

2021年1月

项目类型	功能名称	功能描述	发布时间	相关文档
实体抽取	自学习模型升级	返回结果增加conf，表示抽取的实体置信度	2021-01-20	/

项目类型	功能名称	功能描述	发布时间	相关文档
关键词抽取和文本摘要	预训练模型	基于textrank算法，适用于针对文档抽取关键词或者摘要。详见右侧文档	2021-01-25	关键词抽取和文本摘要服务使用教程
电销场景对话行业分类	预训练模型	适用于电话销售外呼场景，针对对话应用按照行业和场景进行分类，可应用于语音质检。详见右侧文档	2021-01-31	电销场景对话行业分类服务使用教程
所有	前端体验优化	默认采用上传文件名作为数据集名称、显著显示所有模型版本按钮、模型指标页面支持排序等	2021-01-31	/

2020年11月

项目类型	功能名称	功能描述	发布时间	相关文档
商品评价解析	预训练模型升级	优化服饰、箱包行业，抽取召回率提高25%左右；负面情感的micro-F1 +10%；针对长评价进行了优化；增加归一化属性情感词抽取的能力。详见右侧文档	2020-11-09	商品评价解析服务使用教程

2020年10月

项目类型	功能名称	功能描述	发布时间	相关文档
招标中标信息抽取服务	预训练模型	针对招标、中标文件，解析关键要素，详见右侧文档	2020-10-20	招标中标信息抽取服务使用教程
客户咨询解析服务	预训练模型	适用于针对电商等行业的客服-消费者在线聊天场景，解析消费者说话内容，得到消费者意图、情感、情绪、关注点、细粒度情感等结果。详见右侧文档	2020-10-23	客户咨询解析服务使用教程
对话知识抽取服务	预训练模型	适用于客服-消费者在线聊天场景，从对话内容中抽取出客服问-客户答、客户问-客服答等客服话术和用户问题，可用于后续的用户热点问题分析或构建客服话术库，优化客服机器人。详见右侧文档	2020-10-30	对话知识抽取服务使用教程

2020年9月

项目类型	功能名称	功能描述	发布时间	相关文档
商品标题类目预测服务	预训练模型	针对商品标题，预测所属的类目，详见右侧文档	2020-09-18	商品标题类目预测服务使用教程
电销场景用户意图识别服务	预训练模型	针对电话销售场景，识别用户回复客服内容的意图，详见右侧文档	2020-09-18	电销场景用户意图识别服务使用教程
直播ASR乱码识别服务	预训练模型	适用于直播场景，通过ASR语音转文字，识别由于多人同时说话导致的文字可读性不佳的问题，详见右侧文档	2020-09-29	直播ASR乱码识别服务
小说涉黄识别服务	预训练模型	针对中文小说内容，识别是否包含涉黄/色情内容，适用于小说内容监管场景，详见右侧文档	2020-09-29	小说涉黄识别服务
情感分析(英语)服务	预训练模型	适用于针对电商场景的社交媒体（短文本），预测文本所表达的情感，详见右侧文档	2020-09-30	情感分析（英语）服务使用教程
情感分析（西班牙语）服务	预训练模型	适用于针对电商场景的社交媒体（短文本），预测文本所表达的情感，详见右侧文档	2020-09-30	情感分析（西班牙语）服务使用教程
情感分析（俄语）服务	预训练模型	适用于针对电商场景的社交媒体（短文本），预测文本所表达的情感，详见右侧文档	2020-09-30	情感分析（俄语）服务使用教程
文本向量生成服务	预训练模型	适用于中文，详见右侧文档	2020-09-30	文本向量生成服务使用教程

2020年8月

项目类型	功能名称	功能描述	发布时间	相关文档

项目类型	功能名称	功能描述	发布时间	相关文档
实体抽取	预测服务框架升级	实体抽取的模型预测服务框架升级，模型预测效率提升2倍以上（仅对新训练的模型生效）	2020-08-18	/
文本分类/实体抽取	智能标注模块上线	目前支持文本分类和实体抽取任务的智能标注，可以通过平台提供的预标注和主动学习能力进行智能标注，减少标注所需工作量，提高工作效率和标注质量，并将数据用于模型的训练。	2020-08-13	智能标注使用教程
脏话识别服务	预训练模型	针对一句话识别是否是脏话，以及抽取脏话关键词，详见右侧文档	2020-08-26	脏话识别服务使用教程
情绪识别服务	预训练模型	针对一句话识别情绪，当前支持8种情绪识别，详见右侧文档	2020-08-26	情绪识别服务使用教程
新闻层次分类服务	预训练模型	针对新闻数据识别新闻类型，详见右侧文档	2020-08-26	新闻层次分类服务使用教程
商品评价解析	预训练模型升级	新增电器类-扫地机器人、快消品-洗护两个行业，支持行业从37个增加到39个；模型架构升级，重点优化了属性观点词抽取，准确率提升15%。详见右侧文档	2020-08-27	商品评价解析服务使用教程

2020年7月

项目类型	功能名称	功能描述	发布时间	相关文档
所有	Word格式文档解析优化	上传待标注文档中的Word格式文档解析优化，解析句子更加完整	2020-07-16	/
判决文书解析	预训练模型	可直接调用的裁判文书解析预训练模型服务，详见右侧相关文档	2020-07-09	裁判文书解析服务使用教程

2020年6月

项目类型	功能名称	功能描述	发布时间	相关文档
实体抽取/简历抽取	支持增量训练	实体抽取与简历抽取模型支持增量训练，模型迭代更高效快捷	2020-06-18	/
所有	文档解析与模型发布时间预估	上传待标注文档/上传已标注数据集/点击模型发布后，平台给出时间预估，完成后发送短信与邮件通知	2020-06-12	/

2020年5月

项目类型	功能名称	功能描述	发布时间	相关文档
商品评价解析	预训练模型升级	可直接调用的商品评价解析预训练模型服务升级，支持属性情感识别、属性情感词抽取、情感子句抽取、整句情感识别4个功能，同时由基础版本的24个行业增加到37个，详见右侧相关文档	2020-05-30	商品评价解析服务使用教程
简历抽取	1.0版本上线	新上线项目类型，基于阿里内部海量已标注数据训练的模型和规则引擎，实现高准确率的中英文简历抽取，平台提供姓名、手机号、电子邮箱、工作经历、教育经历等27个中文常见字段和10个英文常见字段。若有其他自定义字段，可自行补充标注数据，定制训练。当仅选择平台提供的简历抽取字段时，无需标注数据	2020-05-15	/

2020年4月

项目类型	功能名称	功能描述	发布时间	相关文档

项目类型	功能名称	功能描述	发布时间	相关文档
简历抽取	预训练模型	可直接调用的中英文简历抽取预训练模型服务，详见右侧相关文档	2020-04-30	- 中文服务：简历抽取(中文)服务使用教程 - 英文服务：简历抽取(英文)服务使用教程
商品评价解析	预训练模型	可直接调用的商品评价解析预训练模型服务，详见右侧相关文档	2020-04-17	商品评价解析服务使用教程
实体抽取	手机号抽取模块	创建模型的高级参数中，增加平台预置的手机号抽取选项	2020-04-03	/

2020年3月

项目类型	功能名称	功能描述	发布时间	相关文档
智能合同审查	1.0版本上线	新上线项目类型，对合同里存在的风险点进行智能审查，包括合同逻辑错误、条款缺失、要素不一致、司法风险等；并对合同的相对方资质进行审查，包括多维度潜在风险、风险评级、相对方基本信息等	2020-03-06	/

2020年2月

项目类型	功能名称	功能描述	发布时间	相关文档
实体抽取/文本分类/商品评价解析	模型训练时间预估	模型训练时，给出预估训练时间，训练完毕后发送短信与邮件通知	2020-02-28	/

2020年1月

项目类型	功能名称	功能描述	发布时间	相关文档
实体抽取	上线实体抽取bert模型	bert模型适合小样本数据集，详见右侧相关文档	2020-01-23	实体抽取模型说明
文本分类	超大类别个数的模型优化	优化了超大类别个数的模型训练时间较长的问题	2020-01-23	/
实体抽取/文本分类/情感分析/商品评价解析	数据预处理	平台预置若干预处理规则，协助规整数据，详见右侧相关文档	2020-01-17	数据预处理
短文本匹配	1.0版本上线	新上线项目类型，上传短文本匹配数据，训练短文本语义匹配模型，使用模型时，输入两个短文本，返回相似度	2020-01-17	/

2019年12月

项目类型	功能名称	功能描述	发布时间	相关文档
所有	新增异步预测接口	该接口支持更长文本与文件的离线调用，支持字数：10000字；支持文件格式：txt、html、pdf、doc、docx，详见右侧相关文档	2019-12-31	异步预测接口
商品评价解析	1.0版本上线	新上线项目类型，基于阿里电商平台海量已标注数据，建立各行业定制模型，多维度分析商品评价文本。当仅选择平台提供的评价维度时，无需标注数据	2019-12-20	/
所有	模型包月规格增加	增加多月模型包月规格，详见右侧相关文档	2019-12-20	模型包月购买
实体抽取	上线规则引擎	对部分用户免费开放规则引擎内测权限，通过配置规则辅助模型，详见右侧相关文档	2020-12-13	实体抽取规则引擎

2019年11月

项目类型	功能名称	功能描述	发布时间	相关文档
文本分类	模型迭代优化	更新优化了6种文本分类模型，详见右侧相关文档	2019-11-01	文本分类模型类型
关系抽取	模型迭代优化	关系抽取模型优化，本次更新集成了实体抽取，模型训练完成后可同时抽取实体与关系	2019-11-01	/

2019年10月

项目类型	功能名称	功能描述	发布时间	相关文档
所有	支持RAM子账号授权	通过RAM对子账号进行授权管理，详见右侧相关文档	2019-10-25	子账号授权步骤
文本分类	上线文本分类bert模型	bert模型适合小样本数据集，详见右侧相关文档	2019-01-23	文本分类模型说明

2019年9月

项目类型	功能名称	功能描述	发布时间	相关文档
所有	云栖大会发布会	云栖大会发布NLP自学习平台，详见右侧相关文档	2019-09-26	阿里发布NLP自学习平台
所有	正式商业化	公布公有云收费方案，正式商业化，详见右侧相关文档	2019-09-23	NLP自学习平台定价
关系抽取	1.0版本上线	新上线项目类型，抽取文本中的实体及对应的关系	2019-09-20	/
情感分析	1.0版本上线	新上线项目类型，分析并给出文本的情感正负倾向	2019-09-20	/
关键短语抽取	1.0版本上线	新上线项目类型，抽取文本中的关键词及短语标签	2019-09-20	/

项目类型	功能名称	功能描述	发布时间	相关文档
实体抽取	标注功能优化	标注页面新增同值标注与偏移量微调功能，详见右侧相关文档	2019-09-06	NLP自学习平台中的标注注意事项

2019年8月

项目类型	功能名称	功能描述	发布时间	相关文档
所有	数据中心优化	支持查看数据分布，上传数据集支持质检，模型数据纠错回流，增加提交标注需求的入口	2019-08-30	/
文本分类	模型版本迭代	多分类模型进行了优化提升训练速度和预测效率	2019-08-30	/

2019年7月

项目类型	功能名称	功能描述	发布时间	相关文档
所有	教学视频发布	发布教学视频，帮助用户快速了解平台使用流程，详见右侧相关文档	2019-07-19	[NLP自学习平台教学视频]

2019年6月

项目类型	功能名称	功能描述	发布时间	相关文档
所有	自学习平台1.0版本上线	自学习平台公测，支持实体抽取与文本分类算法的自定义定制	2019-06-10	发布活动推广页
所有	模型中心优化	增加模型版本管理模块，详见右侧相关文档	2019-06-05	快速入门-模型管理
实体抽取	1.0版本上线	新上线项目类型，抽取文本中具有特定意义的实体	2019-06-01	/
文本分类	1.0版本上线	新上线项目类型，抽取文本中的关键词及短语标签	2019-06-01	/

2.产品介绍

产品介绍

面向算法小白用户的NLP行业自适应标注、训练和服务平台。该产品支持文本实体抽取、文本分类、关键短语抽取、情感分析、关系抽取、短文本匹配、商品评价解析等 NLP 定制化算法能力，用户无需拥有丰富的算法背景，仅需标注或上传适量文档数据，即可通过平台获得优质的 NLP 算法模型。

教学视频

功能介绍

NLP自学习平台包含如下服务功能

基础自学习模型：为用户提供文本实体抽取、文本分类、关键短语抽取、文本关系抽取、短文本匹配、情感分析等自然语言算法能力的模型训练。

文本分类	指对文本按照内容类型进行分类，例如短信场景中，运营商根据短信文本内容判断其对应的分类，例如涉黄、暴恐、涉政、广告等。本平台支持定制基于用户自建分类体系的分类模型。
文本实体抽取	抽取文本中具有特定意义的实体。例如合同审核场景中，需要抽取合同名称、甲方、乙方、收款账号等实体信息，用于快速将大量合同结构化，本平台支持定制基于用户自建实体类型的实体抽取模型。
关键短语抽取	基于 textrank 算法，抽取关键词及短语标签。本项目类型无需上传标注数据，算法将自动分析文本中的特征，抽取关键短语。同时，也可上传自定义词表以优化各自领域的关键短语抽取效果。
文本关系抽取	抽取文本中的实体及对应的关系。例如人名与生日，机构与成立时间等，本平台支持定制基于用户自建关系体系的关系抽取模型。
短文本匹配	提供不同文本之间相似度的计算，并输出一个介于0到1之间的分数，分数越大则文本之间的相似度越高。本平台支持定制基于用户自建数据集的短文本匹配模型。
情感分析	分析并给出文本的情感正负倾向，本平台支持定制基于用户自建数据集的情感分析模型。

场景应用自学习模型：为用户提供商品评价解析、简历抽取、招中标信息抽取、智能合同等自然语言算法能力的模型训练。

商品评价解析	基于阿里电商平台海量已标注数据，建立各行业定制模型，多维度分析商品评价文本。本平台支持定制基于用户自建评价维度的商品评价解析模型。

简历抽取	基于阿里内部海量已标注数据训练的模型和规则引擎，实现高准确率的中英文简历抽取等27个中文常见字段和10个英文常见字段。若有其他自定义字段，可自行补充标注数据，定制训练。
招中标信息抽取	智能化完成招标文件解析，自动化抽取招标金额、招标主体、标的等20+字段的抽取，辅助投标文件复查，提升中标率。
合同要素抽取	抽取合同中具有特定意义或关键的要素，支持格式：pdf（文字版）、word。标注数据越多，效果越佳。
智能合同	对合同里存在的风险点进行智能审查，包括合同逻辑错误、条款缺失、要素不一致、司法风险等；并对合同的相对方资质进行审查，包括多维度潜在风险、风险评级、相对方基本信息等。

预训练模型（可直接调用）：为用户提供商品评价解析（电商/本地生活）、电销场景对话（分类/风险/诈骗识别）、新闻分类、新闻事件抽取（英文）、情感分析（多语言）、裁判文书抽取等直接调用的训练模型能力。

商品评价解析-电商领域	支持55种电商领域的行业和192个评价属性，提供多维度商品评价文本分析。
商品评价解析-本地生活领域	支持美容美发美甲、餐饮美食2种本地生活领域的行业和11个评价属性，提供多维度商品评价文本分析。
招中标公告类型分类服务	支持对招中标公告进行分类，目前支持“招标”、“中标”两种类型。
招标中标信息抽取-基础版服务	支持招中标信息里关于项目名称、项目编号、招标人名称、中标金额等13个字段的抽取。
招标中标信息抽取-高级版	支持招标、中标单独解析，对招标信息抽取22个字段。
招标中标信息抽取-高级版	支持招标、中标单独解析，对中标信息抽取29个字段。
合同要素抽取-通用领域	支持合同中的常见要素进行抽取，共支持26个通用要素字段。
在线客服场景解析	支持电商等行业的客服在线聊天场景，解析消费者说话内容，得到消费者意图、情感、情绪等结果。
文档结构化-KV信息抽取	适用于从文档中抽取符合key：value模式的信息，支持简历、合同、报告等文档。

电销场景对话	适用于电话销售外呼场景，针对对话应用按照行业和场景进行分类，可应用于语音质检。支持30+个行业和170+个场景。
脏话识别服务	支持电销场景对话客服质检、直播场景主播监管等应用场景。
电销场景对话	适用于电销人工外呼/智能外呼场景，识别用户的意图（反应）。
关键词抽取/文本摘要	适用于针对文档抽取关键词或者摘要。
简历抽取-英文	支持姓名、联系方式、学位、公司、职位等10个简历字段抽取，适用于英文简历。
简历抽取-中文	支持姓名、性别、年龄、学历、工作单位等33个简历字段抽取，适用于中文简历。
事件抽取（英文）	支持对英文新闻里事件的抽取，包含33个事件类别。
商品标题类目预测	支持电商场景的商品标题，预测所属的类目，类目体系和淘宝等电商平台的一致。
对话知识抽取	适用于客服在线聊天场景，从对话中抽取出客服话术和用户问题，用于热点问题分析或构建客服话术库。
小说涉黄识别服务	支持中文小说涉黄/色情内容识别，适用于小说内容监管场景。输出涉黄的置信度和相关文本内容。
情感分析（俄语）	适用于针对电商场景的社交媒体（短文本），预测俄语文本所表达的情感，分为正，中，负，三种情感。
情感分析（英语）	适用于针对电商场景的社交媒体（短文本），预测英语文本所表达的情感，分为正，中，负，三种情感。
情感分析（西班牙语）	适用于针对电商场景的社交媒体（短文本），预测西班牙语文本所表达的情感，分为正，中，负，三种情感。
情绪识别服务	支持电销、在线接待等应用场景，识别客户或客服的情绪，支持8种常规情绪和3种业务场景常用情绪。
文本向量生成	支持中文文本输入，输出文本对应的向量表示。

新闻文本分类	支持对单个或多个新闻文本的分类。
直播ASR乱码识别	适用于直播场景，通过ASR语音转文字，识别由于多人同时说话导致的文字可读性不佳的问题。
裁判文书抽取	支持10个案由的文书，解析得到38个字段。

产品优势

- 易用：使用流程简单，无需工程与算法背景。
- 快捷：全链路优化，平均训练模型耗时<30分钟。
- 专业：专业技术沉淀，500+标注数据时，模型准确率预计>85%。

3.产品价格

平台于2019年9月23日开始，正式收费。

NLP自学习平台定价

每位主账号新用户均有1个月的免费试用期，具体方案如下：

> 说明
>
> - 自学习平台是按量计费型产品，开通自学习平台就开通了按次调用后付费，**按次调用价格0.016元/次。**
> - 资源包提供了更优惠的调用计费价格，在计费时进行抵扣，超过资源包的部分仍然是按照单次调用价格计费，从当前账户扣除。
> - 资源包使用余额低于10%会有短信、邮件、站内提醒。

免费试用

NLP自学习平台服务开通：【开通服务】

- 试用周期：1个月
- 模型训练：免费模型数3个
- 模型调用：
  - 自学习模型（用户自己训练）：1、**免费调用额度 500次/天/模型，免费试用的模型调用次数达到500次时，将无法继续调用，若有更多调用量需求可购买资源包；**2、购买资源包后，模型调用仍会优先使用每日免费调用额度；3、**资源包消耗完后，将自动转换为按次收费（￥0.016/次）**。每个模型的QPS：10。
  - 预训练模型（平台内置可直接调用）：1、**免费调用额度 500次/模型（累计），免费试用的单个模型调用次数达到500次时，将无法继续调用，若有更多调用量需求可购买资源包；**2、购买资源包后，模型调用仍会优先使用免费调用额度；3、**资源包消耗完后，将自动转换为按次收费（￥0.016/次）**。每个模型的QPS：10。

正式购买

NLP自学习平台训练模型购买：【购买模型】

- 模型训练：500元/月/模型
- 模型调用：
  - 自学习模型（用户自己训练）：免费调用额度 500次/天/模型，**超出时按次计费￥0.016/次，也可购买资源包**，购买资源包后，模型调用仍会优先使用每日免费调用额度。每个模型的QPS：10。
  - 预训练模型（平台内置可直接调用）：**试用周期结束后，无免费调用额度，按次调用计费￥0.016/次，也可购买资源包。**每个模型的QPS：10。
  - 模型包月与资源包N购买入口进入NLP自学习平台可查看。

NLP自学习平台资源包：【点击购买资源包】

调用量(万次)	价格(元)	千次调用成本(元)
按次调用	￥0.016/次	16.00
10	500	5.00
50	2125	4.25
100	4000	4.00
500	18750	3.75
1000	35000	3.50
5000	162500	3.25

资源包有效期1年（从购买成功之日起）

> **注意**
>
> 购买资源包后
>
> 1、优先使用模型免费额度，每个自学习模型有500次免费额度；
>
> 2、免费额度调用完后，将使用购买的资源包进行抵扣；
>
> 3、**资源包消耗完毕，将自动转换为按调用次数收费0.016元/次；**
>
> 按次计费可能会产生高额费用，请及时关注资源包使用情况。

如果您需要大量采购，可提交工单或加入“阿里云NLP自学习平台用户答疑二群”咨询（钉钉群号：44619071）。

QPS增量包定价（限时低价）

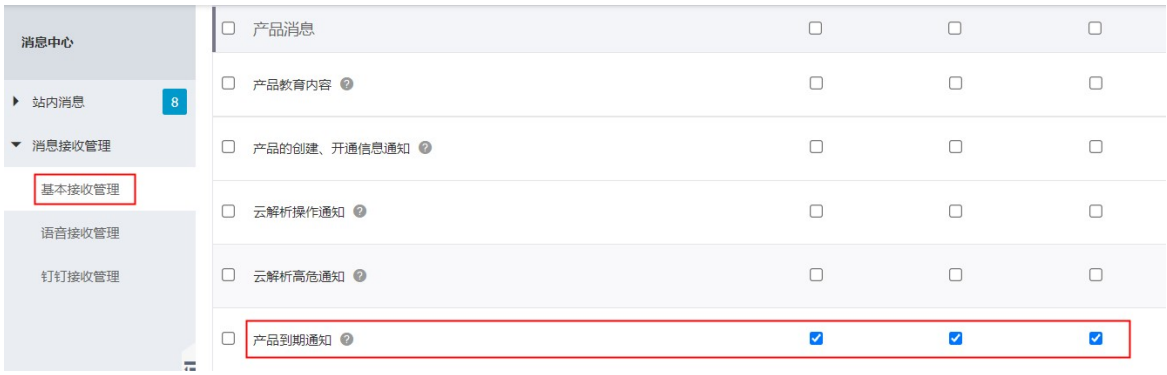
提升数量	价格
10QPS	2000元/年

> **说明**
>
> 提升的QPS越多，享受的优惠折扣越低，该服务购买后暂不支持退款；不同模型可提升的QPS上限不同，具体需求请咨询自学习答疑群负责人。

消息提醒服务

服务开通快捷入口：【开通服务】

1. 登录阿里云控制台，选择消息接收管理。
2. 选择基本接收管理，找到产品到期通知。
3. 勾选短信、邮件、站内提醒，完成消息提醒服务订阅。



4.私有化部署

NLP自学习平台已全面支持私有化部署，目前有两类部署方式：

- 公有云已训练完毕的模型私有化输出
- 完整平台的私有化输出

如想咨询报价与更详细的私有化方案，欢迎您加入“阿里云NLP自学习平台用户答疑二群”咨询（钉钉群号：44619071）

5.API参考

5.1. 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

5.2. 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
…
```

5.3. API使用问题排查

1.SDK调用客户端超时

排查方式：检查服务是否能够正常访问：http://nlp-automl.cn-hangzhou.aliyuncs.com/

如无法访问，请检查服务器网关策略

如果仅支持https协议访问，调用需加上set_protocol_type('https')，SDK默认使用http协议

5.4. 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=CT9X0VtwR86fNWSn
sc6v8YGOjuE%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	编码成 `%20` ，而不是加号（+）。 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： `private static final String ENCODING = "UTF-8"; private static String percentEncode(String value) throws UnsupportedEncodingException { return value != null ? URLEncoder.encode(value, ENCODING).replace("+", "%20").replace("*", "%2A").replace("%7E", "~") : null; }`

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成待签名的字符串。

```
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
```

5.5. 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： *JSON (默认值) | XML*
Version	String	是	API版本号，使用YYYY-MM-DD日期格式。取值： *2019-11-11*
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： *HMAC-SHA1*
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： *1.0*
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请求访问到的资源拥有者账户，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*/
    }
```

5.6. 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	达摩院人工智能	预测文本（文本长度参考不同模型帮助文档）。
ModelId	Integer	是	1818	模型组ID。

名称	类型	是否必选	示例值	描述
ModelVersion	String	是	V1	模型版本。
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	[“达摩院”,”人工智能”]	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=达摩院人工智能&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>[“达摩院”,”人工智能”]</content></GetPredictResultResponse>
```

`JSON` 格式

```
{    “content”:”[\”达摩院\”,\”人工智能\”]”,    “requestId”:”32CE4E89-218B-4DAE-A103-E6BFA66EE63B”}
```

错误码

HttpCode	错误码	错误信息	描述
403	Forbbiden	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	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。

5.7. 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

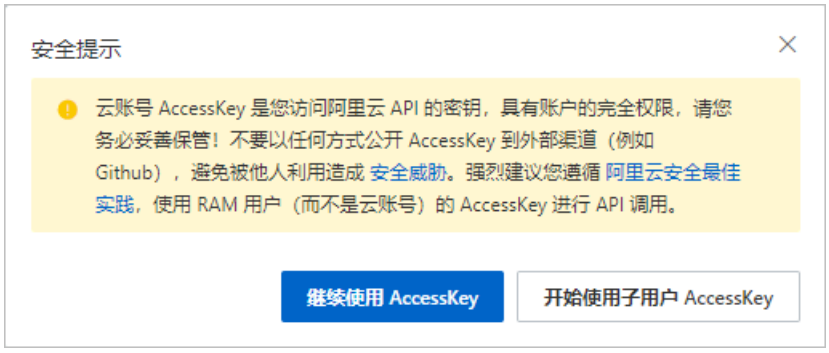
AccessKey包括AccessKey ID和AccessKey Secret。

- 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。

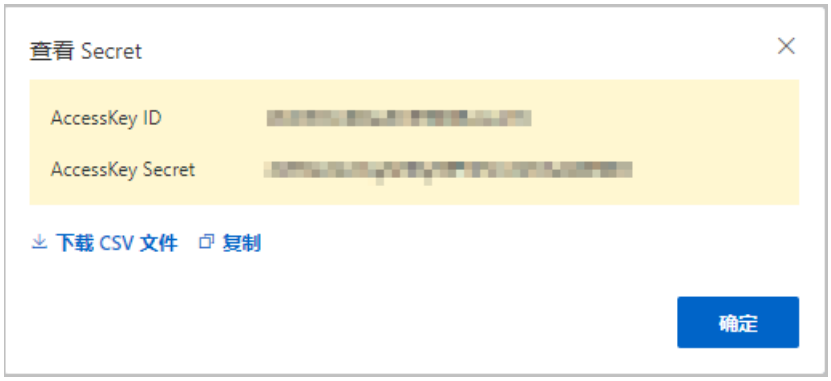
> **警告** 主账号Accesskey泄露会威胁您所有资源的安全。建议使用子账号（RAM用户）Accesskey进行操作，可以有效降低Accesskey泄露的风险。

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。

5.8. 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。
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	Forbbiden	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。

HttpCode	错误码	错误信息	描述
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	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。

5.9. 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);
```

模型结果示例

文本实体抽取

content	模型类型
信达利（福州）企业管理咨询有限公司重庆第一分公司成立于2008年，现坐落于交通便利的江北区红旗河沟轻轨站旁，办公环境优良，工作氛围浓厚。	LSTM-CRF 中文实体抽取

返回结果

```
[{
 "id": 0,
 "sent_offsets": 0,
 "sentence": "信达利（福州）企业管理咨询有限公司重庆第一分公司成立于2008年，现坐落于交通便利的江北区红旗河沟轻轨站旁，办公环境优良，工作氛围浓厚",
 "tags": [{
  "class": "companyName",
  "conf": 0.6834408044815063,
  "end": 17,
  "source": "model",
  "span": "信达利（福州）企业管理咨询有限公司",
  "start": 0
 }]
}]
```

说明

返回结果是JSONArray字符串，需要反序列化之后进行解析,tags为预测结果

tags参数

字段名称	字段值含义
class	实体抽取对象名

字段名称	字段值含义
conf	置信度
start	实体对象结果从文本开始位置
end	实体对象结果到文本结束位置
source	来源
span	预测的实体对象

文本分类

content	模型类型
到手两天用起来感觉还是蛮不错的系统比较快就是win10的通病字体太小屏幕又大看起来比较难受希望win10后期能够改进还有就是电池不可拆卸有点烦恼。	长文本分类融合模型

返回结果

```
{
 "体验": [{
  "score": 0.8428525328636169,
  "children": [{
   "score": 0.8173030018806458,
   "children": [{
    "score": 0.7761349081993103,
    "key": "做工"
   }],
   "key": "质量"
  }, {
   "score": 0.7761349081993103,
   "key": "做工"
  }],
  "key": "产品体验"
 }]
}
```

说明

返回结果是JSON字符串，需要反序列化之后进行解析

返回结果说明

字段名称	字段指含义
score	分数值
key	分类标签
children	多级分类下的子级

关键短语抽取

content	模型类型
青柑的清香+普洱的醇厚,值得爱茶的人们一试。有朋友买后推荐,确实值得!泡得差不多了可以再用煮茶工具煮上一道,茶的味道依然很浓郁。今年冬天主要就喝它了。	关键短语抽取

返回结果

```
["青柑", "清香", "普洱"]
```

> 说明
>
> 返回结果为数组字符串

短文本匹配

content	模型类型
[["信用卡归还，资金的还款的顺序","信用卡还款顺序"]]	短文本匹配

返回结果

```
[{
 "match": "1",
 "sent_a": "信用卡归还，资金的还款的顺序",
 "sent_b": "信用卡还款顺序",
 "value": "0.5989755"
}]
```

> **说明**
>
> 返回结果是JSONArray字符串，需要反序列化之后进行解析

返回结果说明

字段名称	字段值含义
match	是否匹配 0：不匹配 1：匹配
sent_a	短文本a
sent_b	短文本b
value	文本匹配值

情感分析

content	模型类型
打开一袋里面有半袋是坏的,好难吃,没有之前吃的好吃。	中文CNN情感分析

返回结果

```
{
 "情感": [{
  "score": 0.730135977268219,
  "key": "负"
 }, {
  "score": 0.5031050443649292,
  "key": "正"
 }]
}
```

> **说明**
>
> 返回结果是JSON字符串，需要反序列化之后进行解析

> **说明**
>
> score分值大者为分析后的情感

返回结果说明

字段名称	字段含义值
score	模型预测情感分数值
key	情感正，负

商品评价解析

content	模型类型
盒子倒是都包装的很严就是吸盘已经发黄了就是透明的那个圈圈氧化挺严重的应该是很久的东西了价格在这不要求太多了。	商品评价解析

返回结果


```
{
 "物流": [{
  "score": 0.9994128942489624,
  "key": "未提及"
 }, {
  "score": 0.0004609959723893553,
  "key": "正"
 }, {
  "score": 0.00006164848309708759,
  "key": "负"
 }, {
  "score": 0.000018525688574300148,
  "key": "中"
 }],
 "质量": [{
  "score": 0.033323273062705994,
  "key": "未提及"
 }, {
  "score": 0.08673246949911118,
  "key": "正"
 }, {
  "score": 0.7257013916969299,
  "key": "负"
 }, {
  "score": 0.7785664796829224,
  "key": "中"
 }]
}
```

> **说明**
>
> 返回结果是JSON字符串，需要反序列化之后进行解析

返回结果说明

字段名称	字段值含义
score	商品属性分析出的分数值
key	是否在预测文本中提及，以及情感属性状态

简历抽取

content	模型类型
姓名：李某 工作经验：3年 手机：1390000**** 性别：男 电子邮箱:1390000****@sina.cn 户籍地：山西省临汾市 应聘职位：web前端工程师 现居住地：北京 目前状态：已离职,可立即上岗	简历抽取自训练模型（中文）

返回结果

```
[{
 "sentence": "姓名：  李某  工作经验：3年",
 "sent_offsets": 0,
 "id": 0,
 "tags": [{
  "source": "rule",
  "class": "姓名",
  "span": "李某"
 }]
}, {
 "sentence": "手机：1390000****",
 "sent_offsets": 1,
 "id": 0,
 "tags": [{
  "start": 4,
  "conf": 0,
  "end": 15,
  "source": "model",
  "class": "手机号",
  "span": "1390000****"
 }]
}, {
 "sentence": "性别：男",
 "sent_offsets": 2,
 "id": 0,
```

```
 "tags": [{
  "source": "rule",
  "class": "性别",
  "span": "男"
 }]
}, {
 "sentence": "电子邮箱:1390000****@sina.cn 户籍地：山西省临汾市",
 "sent_offsets": 3,
 "id": 0,
 "tags": [{
  "start": 5,
  "conf": 0,
  "end": 16,
  "source": "model",
  "class": "手机号",
  "span": "1390000****"
 }, {
  "source": "rule",
  "class": "电子邮箱",
  "span": "1390000****@sina.cn"
 }, {
  "source": "rule",
  "class": "户口所在地",
  "span": "山西"
 }]
}, {
 "sentence": "应聘职位：web前端工程师 现居住地：北京",
 "sent_offsets": 4,
 "id": 0,
 "tags": [{
  "start": 5,
  "conf": 0,
  "end": 13,
  "source": "model",
  "class": "岗位名称",
  "span": "web前端工程师"
 }, {
  "source": "rule",
  "class": "现居住地",
  "span": "北京"
 }]
}, {
 "sentence": "目前状态：已离职,可立即上岗",
 "sent_offsets": 5,
 "id": 0,
 "tags": []
}]
```


说明

返回结果是JSONArray字符串，需要反序列化之后进行解析

返回结果说明

字段名称	字段值含义
sentence	分析的语句内容
sent_offsets	语句位置

"tags": [{
"start": 5,
"conf": 0,
"end": 16,
"source": "model",
"class": "手机号",
"span": "1390000****"
}, {
"source": "rule",
"class": "电子邮箱",
"span": "1390000****@sina.cn"
}, {
"source": "rule",
"class": "户口所在地",
"span": "山西"
}]
	语句分析抽取后得到的简历信息

双句文本分类

content	模型类型
{"content": "AI：想跟您再核实一下，您是不是业主本人A，房子是不是整套出租呢","content_pair": "是的"}	双句分类高性能版

返回结果


```
{
  "result": "{"意图":[{"score":0.5135,"key":"不是业主本人"},{"score":0.491,"key":"业主本人"}]}
",
  "retCode": "0",
  "type": "NLP_text_pair_classification_cross_attn"
 }
```


> **说明**
>
> 返回结果是JSON字符串，需要反序列化之后进行解析

返回结果说明

字段	字段说明
type	模型类型
key	分类结果
score	置信度
result-key	分类名

5.10. 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。

名称	类型	是否必选	示例值	描述
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

HttpCode	错误码	错误信息	描述
403	Forbbiden	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	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。

5.11. 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	Forbbiden	You are not authorized to operate on the specified resource.	用户未被授权操作指定的资源。

HttpCode	错误码	错误信息	描述
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	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。

5.12. 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。

> **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));
    }
}
```

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

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。

5.13. 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	{"input": {"content": "这件衣服真好看", "domain": "clothing"}}	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","traceId":"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。

5.14. 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>0.0.8</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

新闻层次分类服务

新闻层次分类服务使用示例

简历抽取（中文）

简历抽取服务使用示例

简历抽取（英文）

简历抽取(English)服务使用示例

裁判文书抽取

裁判文书抽取使用示例

定罪量刑预测服务

定罪量刑预测服务使用示例(现版本仅支持刑事案件的定罪量刑预测)

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("reqId","reqId");
input.put("text","案情描述");

RunPreTrainServiceRequest request = new RunPreTrainServiceRequest();
request.setServiceName("convictionSentencingService");
request.setPredictContent(JSON.toJSONString(input));
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"
);

reqid = 'reqid'
text = '案情描述'
input_params = {
    "reqId": reqid,
    "text": text
}

# Initialize a request and set parameters
request = RunPreTrainServiceRequest.RunPreTrainServiceRequest()
request.set_ServiceName('convictionSentencingService')
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']))
```

说明

调用刑事定罪量刑服务，ServiceName需要传入：

convictionSentencingService

。

`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。

```
{
  "reqId": "123445aad",
  "text": "公诉机关指控吸毒人员唐某于2013年3月8日15时许，与一名男子（另案处理）通过电话联系并约定在广州市南沙区金岭北路的汉明酒店303房内进行毒品交易。当天17时许，被告人尹某受上述男子的指派到上述地点以人民币300元的价格将1小包透明晶体贩卖给吸毒人员温某鹏后被公安机关人赃并获，公安人员还从被告人尹某身上缴获锡纸2条。被告人尹某归案后向公安机关如实供述其犯罪事实。经鉴定，被缴获的1小包透明晶体净重0.4克，检出甲基苯丙胺成分。公安机关对被告人尹某进行甲基安非他命测定呈阳性。公诉机关建议本院对被告人尹某在××至一年的幅度内量刑，并处罚金。"
}
```

入参说明

入参	类型	说明
reqId	字符串	请求ID
text	字符串	案情描述

`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。

```
{
  "reqId": "123445aad",
  "penalty": "有期徒刑7个月",
  "articles": [{
    "lawText": "第三百四十七条【走私、贩卖、运输、制造毒品罪】走私、贩卖、运输、制造毒品，无论数量多少，都应当追究刑事责任，予以刑事处罚。走私、贩卖、运输、制造毒品，有下列情形之一的，处十五年有期徒刑、无期徒刑或者死刑，并处没收财产：（一）走私、贩卖、运输、制造鸦片一千克以上、海洛因或者甲基苯丙胺五十克以上或者其他毒品数量大的；（二）走私、贩卖、运输、制造毒品集团的首要分子；（三）武装掩护走私、贩卖、运输、制造毒品的；（四）以暴力抗拒检查、拘留、逮捕，情节严重的；（五）参与有组织的国际贩毒活动的。走私、贩卖、运输、制造鸦片二百克以上不满一千克、海洛因或者甲基苯丙胺十克以上不满五十克或者其他毒品数量较大的，处七年以上有期徒刑，并处罚金。走私、贩卖、运输、制造鸦片不满二百克、海洛因或者甲基苯丙胺不满十克或者其他少量毒品的，处三年以下有期徒刑、拘役或者管制，并处罚金；情节严重的，处三年以上七年以下有期徒刑，并处罚金。单位犯第二款、第三款、第四款罪的，对单位判处罚金，并对其直接负责的主管人员和其他直接责任人员，依照各该款的规定处罚。利用、教唆未成年人走私、贩卖、运输、制造毒品，或者向未成年人出售毒品的，从重处罚。对多次走私、贩卖、运输、制造毒品，未经处理的，毒品数量累计计算。",
    "provisionName": "刑法第347条"
  }],  // 预测法条输出
  "accusation": ["走私、贩卖、运输、制造毒品"],
  "features": [{
    "span": [207, 212],
    "content": "甲基苯丙胺",
    "type": "定罪要素",
    "name": "毒品种类"
  }, {
    "span": [40, 122],
```

```
    "content": "电话联系并约定在广州市南沙区金岭北路的汉明酒店303房内进行毒品交易。当天17时许，被告人尹某受上述男子的指派到上述地点以人民币300元的价格将1小包透明晶体贩卖给",
    "type": "定罪要素",
    "name": "利用信息网络贩卖毒品"
  }, {
    "span": [151, 212],
    "content": "身上缴获锡纸2条。被告人尹某归案后向公安机关如实供述其犯罪事实。经鉴定，被缴获的1小包透明晶体净重0.4克，检出甲基苯丙胺",
    "type": "定罪要素",
    "name": "携带毒品方式进行运输"
  }, {
    "span": [151, 212],
    "content": "身上缴获锡纸2条。被告人尹某归案后向公安机关如实供述其犯罪事实。经鉴定，被缴获的1小包透明晶体净重0.4克，检出甲基苯丙胺",
    "type": "定罪要素",
    "name": "从贩毒人员住所|车辆等处查获的毒品"
  }, {
    "span": [207, 212],
    "content": "甲基苯丙胺",
    "type": "量刑要素",
    "name": "毒品种类"
  }, {
    "span": [198, 204],
    "content": "净重0.4克",
    "type": "量刑要素",
    "name": "毒品克数"
  }, {
    "span": [198, 204],
    "content": "净重0.4克",
    "type": "量刑要素",
    "name": "毒品克数"
  }, {
    "span": [199, 204],
    "content": "重0.4克",
    "type": "量刑要素",
    "name": "毒品克数"
  }]
}
```


出参说明

出参	类型	说明
reqId	字符串	请求id
penalty	字符串	刑期预测结果
articles	列表	建议法条结果

出参	类型	说明
articles.lawText	字符串	法条内容
articles.provisionName	字符串	法条名称
accusation	列表	罪名预测结果
features	列表	要素结合
features.span	[start int, end int]	案情描述中要素对应的起始位置
features.content	字符串	案情描述中要素对应的文本内容
features.type	字符串	要素类型
features.name	字符串	要素名称

5.15. 控制台API操作

控制台项目，数据集，模型相关API操作文档。

1.创建项目

示例代码：

```
        String accessKeyId = "your key";
        String accessKeySecret = "your secret";
        try {
            // 创建DefaultAcsClient实例并初始化
            DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", accessKeyId,
                    accessKeySecret); // 您的AccessKey Secret
            IAcsClient client = new DefaultAcsClient(profile);
            // 创建API请求并设置参数
            CommonRequest request = new CommonRequest();
            request.setSysDomain("nlp-automl.cn-hangzhou.aliyuncs.com");
            request.setSysVersion("2019-11-11");
            request.setSysAction("CreateProject");
            request.setSysMethod(MethodType.POST);
            request.putBodyParameter("ProjectType", ProjectType);
            request.putBodyParameter("ProjectName", ProjectName);
            request.putBodyParameter("ProjectDescription", ProjectDescription);

            CommonResponse response = client.getCommonResponse(request);

            logger.info("testBysdk is ok request:{} response:{}", JSON.toJSONString(request
),
                    JSON.toJSONString(response));
            System.out.println(response.getData());

        } catch (Exception e) {
            logger.error("testBysdk is error", e);
        }
```

入参说明：

参数名称	类型	是否必填	备注	注意
ProjectType	string	是	项目类型	'ProjectType'参照下表填入对应value,value为String类型
ProjectName	string	是	项目名称	ProjectName为String类型，不能以数字和符号开头，长度不能大于32个字符
ProjectDescription	string	是	项目描述	ProjectDescription为String类型，不能以数字和符号开头，长度不能大于256个字符

说明

Action需设置为CreateProject

项目名称	对应参数value
文本实体抽取	ner
文本分类	textCategory
文本关系抽取	baseRelation
短文本匹配	textMatching
情感分析	sentimentAnalysis
简历抽取	resumeExtract
商品评价解析	reviewAnalysis

返回数据示例：

```
{
    "Code": 200,
    "ProjectId": 1234,
    "Message": "创建成功",
    "RequestId": "xxxxxxxxxx",
    "Success": true
}
```

说明

返回的数据为JSON字符串需要反序列化后进行解析，ProjectId为创建的项目的ID

2.创建数据集

```
        String accessKeyId = "your key";
        String accessKeySecret = "your secret";
        try {
            // 创建DefaultAcsClient实例并初始化
            DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", accessKeyId,
                    accessKeySecret); // 您的AccessKey Secret
            IAcsClient client = new DefaultAcsClient(profile);
            // 创建API请求并设置参数
            CommonRequest request = new CommonRequest();
            request.setSysDomain("nlp-automl.cn-hangzhou.aliyuncs.com");
            request.setSysVersion("2019-11-11");
            request.setSysAction("CreateDataset");
            request.setSysMethod(MethodType.POST);
            request.putBodyParameter("ProjectId", ProjectId);
            request.putBodyParameter("DatasetName", DatasetName);

            CommonResponse response = client.getCommonResponse(request);

            logger.info("testBysdk is ok request:{} response:{}", JSON.toJSONString(request
),
                    JSON.toJSONString(response));
            System.out.println(response.getData());

        } catch (Exception e) {
            logger.error("testBysdk is error", e);
        }
```

说明

Action需设置为CreateDataset

入参说明：

参数名称	类型	是否必填	备注	注意
ProjectId	Long	是	项目id	ProjectId为Long类型，要在某个项目下创建一个数据集，填写目标项目ID
DatasetName	string	是	数据集名称	DatasetName为String类型，不能以数字和符号开头，长度不能大于32个字符

返回数据示例：

```
{
    "Code": 200,
    "DatasetId": 1234,
    "Message": "创建成功",
    "RequestId": "xxxxxx",
    "Success": true
}
```

说明

返回的数据为JSON字符串需要反序列化后进行解析，datasetId为创建的数据集的ID

3.数据集上传

```
        String accessKeyId = "your key";
        String accessKeySecret = "your secret";
        try {
            // 创建DefaultAcsClient实例并初始化
            DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", accessKeyId,
                    accessKeySecret); // 您的AccessKey Secret
            IAcsClient client = new DefaultAcsClient(profile);
            // 创建API请求并设置参数
            CommonRequest request = new CommonRequest();
            request.setSysDomain("nlp-automl.cn-hangzhou.aliyuncs.com");
            request.setSysVersion("2019-11-11");
            request.setSysAction("CreateDatasetRecord");
            request.setSysMethod(MethodType.POST);
            request.putBodyParameter("ProjectId", ProjectId);
            request.putBodyParameter("DatasetId", DatasetId);
            request.putBodyParameter("DatasetRecord", DatasetRecord);

            CommonResponse response = client.getCommonResponse(request);

            logger.info("testBysdk is ok request:{} response:{}", JSON.toJSONString(request
),
                    JSON.toJSONString(response));
            System.out.println(response.getData());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
```

说明

Action需设置为CreateDatasetRecord

入参说明：

参数名称	类型	是否必填	备注	注意
ProjectId	Long	是	项目id	
DatasetId	Long	是	数据集id	DatasetId为Long类型，要在某个数据集下添加一条数据，填写具体数据集id
DatasetRecord	string	是	标注文档内容	DatasetRecord为String类型，需要传入JSON字符串，内容为标注数据，如下表格示例

危险

DatasetId只可使用API创建的数据集，不可使用控制台创建的数据集ID

项目类型	DatasetRecord
文本分类	{"content":"这个牌子的红枣很不错","records":{"情感":"正"}}
文本实体抽取	{"content":"阿里巴巴集团达摩院的工作地主要有：北京、杭州、新加坡、美国等","records":{"公司名":[{"span":"阿里巴巴集团","offset":[0,6]}],"地点":[{"span":"北京","offset":[17,19]},{"span":"杭州","offset":[20,22]},{"span":"新加坡","offset":[23,26]},{"span":"美国","offset":[27,29]}]}}
短文本匹配	{"left": "快递太暴力了", "right": "快递太粗鲁了", "value": "0"}

说明

短文本匹配left与right为两个需要匹配的文本，value为标注是否匹配，0：不匹配，1：匹配

> **说明**
>
> 具体可参照数据集上传中的格式

返回数据示例：

```
{
    "Code": 200,
    "DatasetRecordId": 1234,
    "Message": "上传成功",
    "RequestId": "1787856526166507",
    "Success": true
}
```

4.数据集列表查询

```
        String accessKeyId = "your key";
        String accessKeySecret = "your secret";
        try {
            // 创建DefaultAcsClient实例并初始化
            DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", accessKeyId,
                    accessKeySecret); // 您的AccessKey Secret
            IAcsClient client = new DefaultAcsClient(profile);
            // 创建API请求并设置参数
            CommonRequest request = new CommonRequest();
            request.setSysDomain("nlp-automl.cn-hangzhou.aliyuncs.com");
            request.setSysVersion("2019-11-11");
            request.setSysAction("ListDataset");
            request.setSysMethod(MethodType.POST);
            request.putBodyParameter("ProjectId", ProjectId);
            request.putBodyParameter("PageNumber", PageNumber);
            request.putBodyParameter("PageSize", PageSize);

            CommonResponse response = client.getCommonResponse(request);

            logger.info("testBysdk is ok request:{} response:{}", JSON.toJSONString(request
),
                    JSON.toJSONString(response));
            System.out.println(response.getData());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
```

> **说明**
>
> Action需设置为ListDataset

入参说明：

参数名称	类型	是否必填	备注	说明
ProjectId	Long	是	项目ID	ProjectId为Long类型，为查询目标项目下的数据集列表
PageNumber	Integer	是	页数	PageNumber从1开始计算，必须大于0
PageSize	Integer	是	分页大小	PageSize必须大于0

返回数据示例：

```
{
    "Code": 200,
    "Data": {
        "pageNumber": 1,
        "pageSize": 1,
        "rows": [
            {
                "annotatedDocNum": 7,
                "id": 1234,
                "name": "数据集名称",
                "projectType": "textCategory",
                "sourceType": "upload",
                "status": "finished",
                "totalDocNum": 7
            }
        ],
        "totalCount": 48
    },
    "Message": "",
    "RequestId": "1787856526166507",
    "Success": true
}
```

出参说明

参数名称	参数说明	备注
pageNumber	当前页数	当前页数
pageSize	分页大小	分页大小
rows	返回数据集数据	返回数据集数据

参数名称	参数说明	备注
totalCount	总记录数	总记录数

rows参数说明

参数名称	参数说明	备注
id	数据集ID	数据集ID
annotatedDocNum	已标注文档数量	已标注文档数量
name	数据集名称	数据集名称
projectType	项目类型	项目类型
sourceType	数据集来源	upload 数据集上传 annotate 标注任务
status	数据集状态	init 初始化 parsing 解析中 parse_failed 解析失败 finished 完成
totalDocNum	文档总数	文档总数

5.创建模型

示例代码：

```
        String accessKeyId = "your key";
        String accessKeySecret = "your secret";
        try {
            // 创建DefaultAcsClient实例并初始化
            DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", accessKeyId,
                    accessKeySecret); // 您的AccessKey Secret
            IAcsClient client = new DefaultAcsClient(profile);
            // 创建API请求并设置参数
            CommonRequest request = new CommonRequest();
            request.setSysDomain("nlp-automl.cn-hangzhou.aliyuncs.com");
            request.setSysVersion("2019-11-11");
            request.setSysAction("CreateModel");
            request.setSysMethod(MethodType.POST);
            request.putBodyParameter("ModelType", "NLP-Resume-Chinese");
            request.putBodyParameter("ProjectId", 3667);
            request.putBodyParameter("ModelName", "apiTest");
            request.putBodyParameter("DatasetIdList", "12267");

            CommonResponse response = client.getCommonResponse(request);

            logger.info("testBysdk is ok request:{} response:{}", JSON.toJSONString(request
),
                    JSON.toJSONString(response));
            System.out.println(response.getData());

        } catch (Exception e) {
            logger.error("testBysdk is error", e);
        }
```

入参说明：

字段名称	类型	是否必填	字段含义	注意
ModelType	String	是	模型类型	模型名称ModelType，不能以数字和符号开头，长度不能大于32个字符
ProjectId	Long	是	项目ID	
ModelName	String	是	模型名称	
ModelId	Long	否	模型ID，即自学习平台上可见的模型ID（增量版本时需要）	新增模型版本时ModelId为必传项

DatasetIdList	String	是	数据集ID列表,逗号分割如"1212,1211"	上传数据集必须在同一ProjectId下，已上线状态，且已标注数据总数大于等于4条
TestDatasetIdList	String	否	测试集ID列表，逗号分隔（文本分类下部分模型支持添加测试集数据）	
IsIncrementalTrain	String	否	是否增量训练，是：true,否：false,新增版本必填true	

说明

action为CreateModel

说明

ModelType参照下面对照表，传入对应的value

项目	模型类型名称对应传值 **value**	
文本实体抽取	LSTM-CRF 中文实体抽取--> NLP-NER-Chinese	
	TinyStructBERT 中文实体抽取 --> NLP-NER-BERT	
	英文实体抽取 --> NLP-NER-English	
	长文本分类融合模型 --> NLP-TextCategory-ensemble	
	CNN分类模型 --> NLP-TextCategory-CNN2	

文本分类	FastText分类模型 --> NLP-TextCategory-FT	不同项目下，传指定的modelType，后端会做校验，1.如果在Project下不存在该模型类型，则会提示未找到基础模型：BaseModelNotFoundError例如在文本实体抽取下传递文本分类的modelType等情况注：关键短语抽取DatasetIdList -1
	Self-Attention分类模型 --> NLP-TextCategory-SA	
	短文本分类融合模型 --> NLP-TextCategory-STC	
	Bert小样本分类 --> classification_bert	
关键短语抽取	关键短语抽取 --> NLP-Keyphrase-textrank	
文本关系抽取	关系抽取PCNN --> NLP-RE-PCNN	
	关系抽取Bert --> NLP-RelationClassficiation-BERT	
	关系抽取BertNoise --> NLP-RE-BERT-NOISE	
短文本匹配	structbert_matching	
情感分析	中文CNN情感分析 --> NLP-TextCategory-CNN2	
商品评价解析	classification_cnn	
简历抽取	简历抽取自训练模型（中文）--> NLP-Resume-Chinese	
	无标注数据的平台预置简历抽取模型（中文）--> NLP-Resume-Pretrain-Chinese	
	无标注数据的平台预置简历抽取模型（英文）--> NLP-Resume-Pretrain-English	

返回结果示例：

```
{
  "RequestId": "xxxxxxxxx",
  "Message": "",
  "Data": {
    "modelName": "api",
    "modelId": 1841,
    "modelStatus": "初始化",
    "modelVersion": "v1",
    "updateTime": "2020-09-24 18:36:34",
    "modelType": "NLP-NER-Chinese",
    "trainingCount": 18,
    "projectId": 1922
  },
  "Code": 200,
  "Success": true
}
```

6.发布、上线、下线模型

代码示例：

```
        String accessKeyId = "your key";
        String accessKeySecret = "your secret";
        try {

            // 创建DefaultAcsClient实例并初始化
            DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", accessKeyId,
                    accessKeySecret); // 您的AccessKey Secret
            IAcsClient client = new DefaultAcsClient(profile);
            // 创建API请求并设置参数
            CommonRequest request = new CommonRequest();
            request.setSysDomain("nlp-automl.cn-hangzhou.aliyuncs.com");
            request.setSysVersion("2019-11-11");
            request.setSysAction("DeployModel");
            request.setSysMethod(MethodType.POST);
            request.putBodyParameter("ModelId", 5590);
            request.putBodyParameter("ProjectId", 3667);
            request.putBodyParameter("ModelVersion", "v1");

            //"1":模型发布 "2":模型下线 "3":模型下线
            request.putBodyParameter("OptType", "1");

            CommonResponse response = client.getCommonResponse(request);
            System.out.println(response.getData());
            logger.info("testBysdk is ok request:{} response:{}", JSON.toJSONString(request
),
                    JSON.toJSONString(response));

        } catch (Exception e) {
            logger.error("testBysdk is error", e);
        }
```

入参说明：

参数名称	类型	是否必填	备注
ModelId	Long	是	模型ID
OptType	String	是	"1":模型发布 "2":模型下线 "3":模型上线
ProjectId	Long	是	项目ID
ModelVersion	String	是	模型迭代版本，如：v1

警告

同一个版本模型不能连续重复相同操作，比如已经发布的模型再次发布

说明

action为DeployModel

返回结果示例：

```
{
  "RequestId": "xxxxxxxxx",
  "Message": "",
  "Data": {
    "modelName": "api",
    "modelId": 1841,
    "modelStatus": "发布中",
    "modelVersion": "v1",
    "precision": "50.00",
    "recall": "50.00",
    "updateTime": "2020-09-24 18:51:45",
    "modelType": "NLP-NER-Chinese",
    "f1": "50.00",
    "trainingCount": 18,
    "projectId": 1922
  },
  "Code": 200,
  "Success": true
}
```

7.查询指定模型版本信息

代码示例：

```
        String accessKeyId = "your key";
        String accessKeySecret = "your secret";
        try {

            // 创建DefaultAcsClient实例并初始化
            DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", accessKeyId,
                    accessKeySecret); // 您的AccessKey Secret
            IAcsClient client = new DefaultAcsClient(profile);
            // 创建API请求并设置参数
            CommonRequest request = new CommonRequest();
            request.setSysDomain("nlp-automl.cn-hangzhou.aliyuncs.com");
            request.setSysVersion("2019-11-11");
            request.setSysAction("GetModel");
            request.setSysMethod(MethodType.POST);
            request.putBodyParameter("ModelId", 4972);
            request.putBodyParameter("ProjectId", 4559);
            request.putBodyParameter("ModelVersion", "v1");

            CommonResponse response = client.getCommonResponse(request);

            System.out.println(response.getData());
            logger.info("testBysdk is ok request:{} response:{}", JSON.toJSONString(request
),
                    JSON.toJSONString(response));

        } catch (Exception e) {
            logger.error("testBysdk is error", e);
        }
```

说明

action为GetModel

入参说明：

参数名称	类型	是否必填	备注
ModelId	Long	是	模型ID
ProjectId	Long	是	项目ID
ModelVersion	String	是	指定查询的版本，如：v1

返回结果示例：

```
{
  "RequestId": "xxxxxxxxx",
  "Message": "",
  "Data": {
    "modelName": "api",
    "modelId": 1841,
    "modelStatus": "发布完成",
    "modelVersion": "v1",
    "precision": "50.00",
    "recall": "50.00",
    "updateTime": "2020-09-24 18:53:08",
    "modelType": "NLP-NER-Chinese",
    "f1": "50.00",
    "trainingCount": 18,
    "projectId": 1922
  },
  "Code": 200,
  "Success": true
}
```

8.模型分页查询

代码示例：

```
        String accessKeyId = "your key";
        String accessKeySecret = "your secret";
        try {

            // 创建DefaultAcsClient实例并初始化
            DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", accessKeyId,
                    accessKeySecret); // 您的AccessKey Secret
            IAcsClient client = new DefaultAcsClient(profile);
            // 创建API请求并设置参数
            CommonRequest request = new CommonRequest();
            request.setSysDomain("nlp-automl.cn-hangzhou.aliyuncs.com");
            request.setSysVersion("2019-11-11");
            request.setSysAction("ListModels");
            request.setSysMethod(MethodType.POST);
            request.putBodyParameter("ProjectId", 3667);
            request.putBodyParameter("PageNumber", 1);
            request.putBodyParameter("PageSize", 5);

            CommonResponse response = client.getCommonResponse(request);

            logger.info("testBysdk is ok request:{} response:{}", JSON.toJSONString(request
),
                    JSON.toJSONString(response));

        } catch (Exception e) {
            logger.error("testBysdk is error", e);
        }
```

参数说明：

参数	类型	是否必填	备注	注意
ProjectId	Long	是	项目ID	ProjectId为Long类型，为查询目标项目下的模型列表
PageSize	Integer	是	每页展示模型数量	PageSize必须大于0
PageNumber	Integer	是	当前页数	PageNumber从1开始计算，必须大于0

说明

以模型ID分页返回每个模型下的所有迭代模型信息

返回结果示例：

```
{
  "RequestId": "xxxxxxxxx",
  "Message": "",
  "Data": {
    "pageNumber": 1,
    "pageSize": 5,
    "rows": [
      {
        "modelName": "test",
        "models": [
          {
            "modelStatus": "训练结束",
            "modelVersion": "v3",
            "precision": "0.60",
            "recall": "0.64",
            "f1": "0.64",
            "trainingCount": 65
          },
          {
            "modelStatus": "训练结束",
            "modelVersion": "v4",
            "precision": "0.66",
            "recall": "0.49",
            "f1": "0.57",
            "trainingCount": 65
          },
          {
            "modelStatus": "训练结束",
            "modelVersion": "v5",
            "precision": "0.62",
```

```
            "precision": "0.62",
            "recall": "0.49",
            "f1": "0.55",
            "trainingCount": 65
          },
          {
            "modelStatus": "训练结束",
            "modelVersion": "v6",
            "precision": "0.99",
            "recall": "0.99",
            "f1": "0.99",
            "trainingCount": 65
          },
          {
            "modelStatus": "训练结束",
            "modelVersion": "v7",
            "precision": "0.99",
            "recall": "0.99",
            "f1": "0.99",
            "trainingCount": 65
          },
          {
            "modelStatus": "发布完成",
            "modelVersion": "v1",
            "precision": "0.62",
            "recall": "0.62",
            "f1": "0.62",
            "trainingCount": 22
          },
          {
            "modelStatus": "训练结束",
            "modelVersion": "v2",
            "precision": "0.37",
            "recall": "0.37",
            "f1": "0.37",
            "trainingCount": 22
          }
        ],
        "modelId": 1773,
        "modelType": "NLP-TextCategory-ensemble",
        "projectId": 1923
      }
    ],
    "totalCount": 11
  },
  "Code": 200,
  "Success": true
}
```


9.删除模型

代码示例：

```
String accessKeyId = "your key";
        String accessKeySecret = "your secret";
        try {

            // 创建DefaultAcsClient实例并初始化
            DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", accessKeyId,
                    accessKeySecret); // 您的AccessKey Secret
            IAcsClient client = new DefaultAcsClient(profile);
            // 创建API请求并设置参数
            CommonRequest request = new CommonRequest();
            request.setSysDomain("nlp-automl.cn-hangzhou.aliyuncs.com");
            request.setSysVersion("2019-11-11");
            request.setSysAction("DeleteModel");
            request.setSysMethod(MethodType.POST);
            request.putBodyParameter("ProjectId", projectId);
            request.putBodyParameter("ModelId", modelId);

            CommonResponse response = client.getCommonResponse(request);

            logger.info("testBysdk is ok request:{} response:{}", JSON.toJSONString(request
),
                    JSON.toJSONString(response));

        } catch (Exception e) {
            logger.error("testBysdk is error", e);
        }
```

参数说明：

参数	类型	是否必填	备注	注意
ProjectId	Long	是	项目ID	
ModelId	Long	是	模型组ID	ModelId 为模型组ID

返回结果示例：

```
{
  "RequestId": "28E59BD8-4591-425C-8752-93DCF1BEA93F",
  "Message": "",
  "Data": true,
  "Code": 200,
  "Success": true
}
```

5.16. 招标中标信息抽取服务

招标信息抽取抽取服务示例

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 = "招标文书文本";
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 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 = '招标文书文本'

# 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**

说明

PredictContent参数为文书文本参数

说明

PredictResult是JSON字符串，需要反序列化之后进行解析

6.SDK示例

6.1. 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

6.2. Java SDK

获取SDK

最新sdk版本请查询查询最新版本

在pom.xml文件中引用sdk的maven依赖。

```
<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>0.0.8</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\"]"
}
```

6.3. 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'])
```

6.4. 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());
        }
    }
}
```

6.5. 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;
}
```

6.6. 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;
}
```

6.7. 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
}
```

6.8. 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();
```

6.9. Ruby SDK

获取SDK

```
gem install aliyunsdkcore
```

使用示例

```
require 'aliyunsdkcore'
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
```