

2019-3-11

阿里巴巴 智能语音

自学习平台用户手册

机器智能技术

阿里巴巴集团

目录

1	自学习平台介绍.....	6
1.1	定制语言模型.....	6
1.2	定制类热词.....	6
1.3	定制泛热词.....	6
2	手册说明.....	8
3	部署说明.....	8
3.1	系统架构	8
3.2	硬件推荐配置	10
3.3	部署步骤	10
3.3.1	下载专有云安装包	10
3.3.2	安装 docker.....	10
3.3.3	启动服务	11
3.4	服务维护	11
3.4.1	服务重启	11
3.4.2	进入容器	11
4	快速开始.....	13
4.1	导入文件	13
4.1.1	使用控制台.....	13
4.1.2	使用存储 API.....	13
4.1.3	使用 mongofiles 命令	13
4.2	定制语言模型.....	14
4.2.1	配置基础模型	14
4.2.2	准备训练数据	15
4.2.3	导入训练数据	16
4.2.4	创建定制模型	17
4.2.5	训练定制模型	17
4.3	定制类热词.....	18
4.4	定制泛热词.....	18

5	功能说明	18
5.1	定制语言模型	18
5.1.1	数据准备要求	18
5.1.2	数据准备常见思路	19
5.1.3	模型训练参数	19
5.1.4	模型训练任务信息	19
5.2	定制类热词	20
5.2.1	格式要求	20
5.3	定制泛热词	20
5.3.1	格式要求	20
5.3.2	权重设置	20
5.3.3	数据准备方法	20
5.3.4	测试参数	20
6	系统 API	21
6.1	调用约定	21
6.2	错误码	22
6.3	通用对象定义	22
6.3.1	分页对象 NlsPage	22
6.3.2	定制语言模型数据集对象 AsrLmData	23
6.3.3	定制语言模型对象 AsrLmModel	23
6.3.4	定制语言模型基础模型对象 AsrLmBase	24
6.3.5	定制类热词对象 AsrClassVocab	24
6.3.6	定制泛热词对象 AsrVocab	25
6.4	定制语言模型	25
6.4.1	创建数据集	25
6.4.2	列举数据集	26
6.4.3	查询数据集	27
6.4.4	更新数据集	27
6.4.5	删除数据集	28
6.4.6	创建模型	29
6.4.7	列举模型	29
6.4.8	查询模型	30
6.4.9	更新模型	31

6.4.10	删除模型.....	32
6.4.11	添加数据集到模型.....	32
6.4.12	从模型删除数据集.....	32
6.4.13	开始模型训练.....	33
6.4.14	停止模型训练.....	33
6.4.15	上线模型.....	34
6.4.16	下线模型.....	34
6.4.17	创建基础模型.....	35
6.4.18	列举基础模型.....	36
6.4.19	查询基础模型.....	36
6.4.20	更新基础模型.....	37
6.4.21	删除基础模型.....	38
6.5	定制类热词.....	38
6.5.1	创建词表.....	38
6.5.2	列举词表.....	39
6.5.3	查询词表.....	40
6.5.4	更新词表.....	41
6.5.5	删除词表.....	41
6.6	定制泛热词.....	42
6.6.1	创建词表.....	42
6.6.2	列举词表.....	43
6.6.3	查询词表.....	44
6.6.4	更新词表.....	44
6.6.5	删除词表.....	45
6.7	存储管理.....	45
6.7.1	上传文件.....	45
6.7.2	列举文件.....	46
6.7.3	下载文件.....	47
6.7.4	删除文件.....	47
6.8	Swagger 文档.....	48
7	控制台.....	51
7.1	文件管理.....	51
8	常见问题.....	53

8.1	排查定制语言模型是否生效	53
8.2	排查定制声学 and 语言模型错误	53
8.2.1	排查的一般原则.....	53
8.2.2	常见错误排查	53
8.3	排查定制泛热词不生效	54
8.4	定制模型(非工程类)常见问题.....	54
8.4.1	定制语言模型	54
8.4.2	定制声学模型	55
9	附录.....	56
9.1	mongofiles 命令使用说明	56

1 自学习平台介绍

目前阿里巴巴的语音识别服务已经提供了许多针对不同场景优化过的模型。这些模型的训练数据中包含了针对不同场景的数据，因此在相应的场景下会有更好的识别准确率。自学习平台提供了一些进一步改进识别准确率的方法。

目前自学习平台主要包括以下功能：

- 定制语言模型
- 定制类热词
- 定制泛热词

1.1 定制语言模型

用户可以通过定制语言学模型接口创建自己的定制语言学模型，使用与自己场景相关的文本数据训练语言学模型，以改善识别效果。在语音识别解码过程中定制语言模型和基础语言大模型被一起用来计算语言学模型得分。

适用领域：几乎所有场景相关的情况

使用成本：低

应用举例：车载控制台语音控制,智能客服,法院庭审

1.2 定制类热词

用户可以通过定制类热词接口创建针对某些语境的自定义词表，以改善某一类词的识别效果。用特殊类名标签(person-name)替代具体的词进行通用语言学模型训练(增强这类词的概率)，在解码中遇到特殊标签时再对类中的词进行打分并识别(可以随时添加新词到类中)。

适用领域：对人名，地名，机构名等类别的识别要求比较高的场景

使用成本：低-中（基础语言模型是类语言模型）

应用举例：地图 APP 语音识别（地名，人名），法院庭审语音识别（地名，人名，机构名），支付宝语音转账（人名）

1.3 定制泛热词

用户可以通过定制泛热词接口创建自己的自定义词表，在语音识别解码过程中改善某些词的识别效果（通过增强该词的概率）和阻止某些词被识别（通过降低该词的概率）

适用领域：个别关键的业务词/专有名词识别不好的情况下使用(例如，加强溪湖的权重，降低西湖的权重)，不适合大规模使用，建议在其他方法无效的时候使用。

使用成本：低

应用举例：法院庭审语音识别（特殊专有名词）

阿里巴巴 智能语音

2 手册说明

手册中的所有示例基于如下假设：

- 假设专有云安装目录为 /home/admin/service
- 假设自学习平台 API 组件的部署的地址为 127.0.0.1，端口是 8701
- 假设将 IP 地址 127.0.0.1 绑定到域名 nls-slp.aliyun.test

绑定域名的方法：

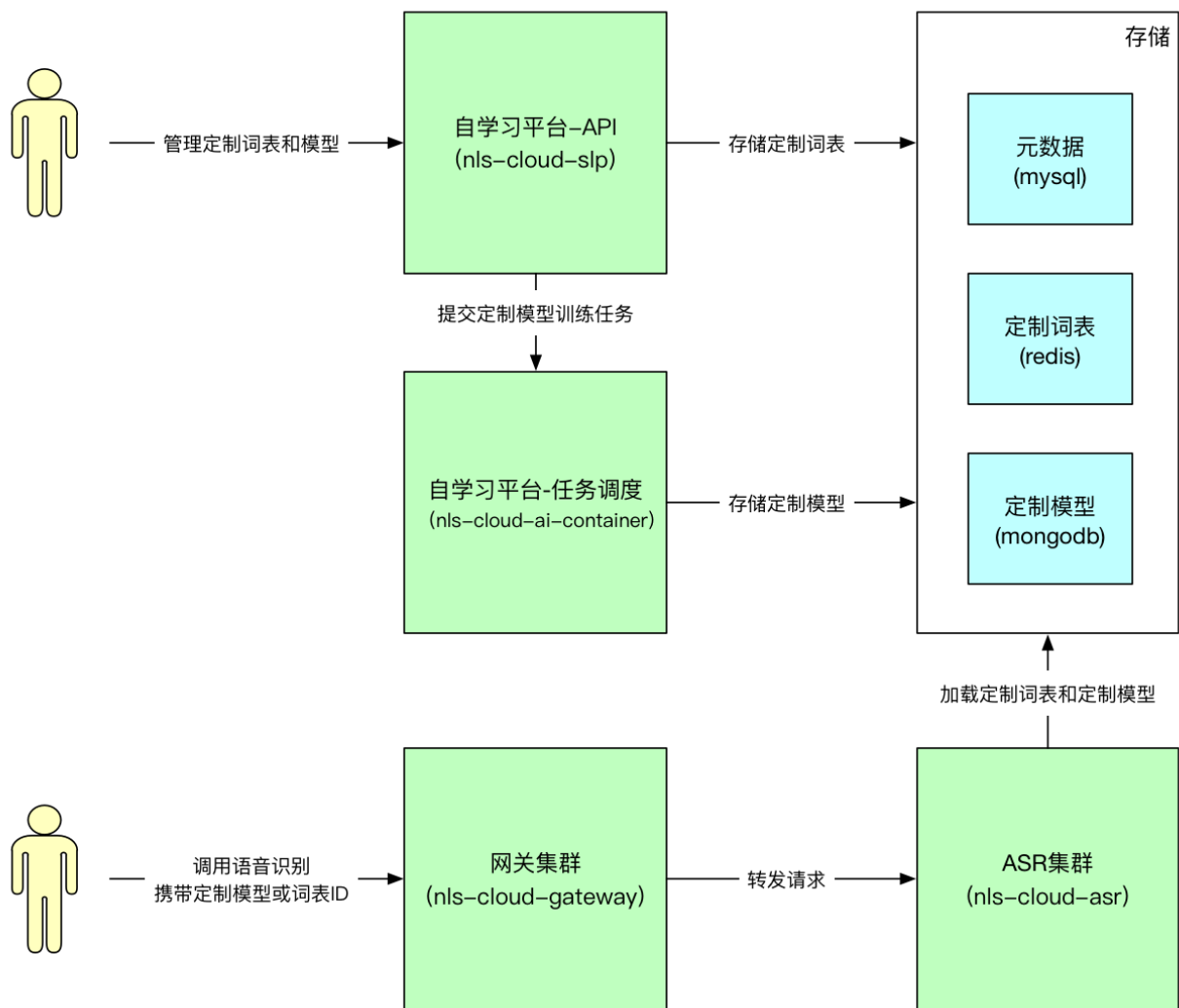
在 /etc/hosts 文件追加一行：“127.0.0.1 nls-slp.aliyun.test”

3 部署说明

自学习平台做为**阿里巴巴智能语音专有云**的一个功能模块提供给用户使用。阿里巴巴智能语音专有云服务的安装目录包含了完整的专有云部署文档。本节仅对自学习平台相关的组件给出部署说明。

3.1 系统架构

下图是自学习平台系统架构的示意图：



nls-cloud-slp

SLP 对所有定制资源进行管理，通过数据库保存了所有资源的信息，并对外提供了用户界面。用户可以直接使用 SLP 提供的控制台管理定制模型。用户也可以通过 SLP 提供的 API，将自学习平台接入自己的业务系统。

nls-cloud-ai-container

AI-Container 对定制模型的训练任务进行调度，并将训练任务生成的模型文件上传到模型仓库。SLP 会通过 AI-Container 提供的 HTTP 接口提交定制模型训练任务。

存储组件

自学习平台使用了 mysql 保存定制资源的元数据，使用 redis 作为对象缓存，使用 mongodb 作为文件存储。

nls-cloud-gateway

SLP 和 AI-Container 仅负责定制资源的生产和管理。使用定制化资源需要将相关的资源 ID 传递到服务网关 Gateway。Gateway 会将定制资源 ID 传递到 ASR 服务

nls-cloud-asr

ASR 服务负责执行具体的语音识别任务。在收到 Gateway 的请求之后，ASR 服务会确认请求参数是否携带了定制资源 ID，如果有则会加载响应的定制资源。

3.2 硬件推荐配置

以下为自学习平台的推荐硬件配置：

组件	CPU	内存	硬盘	备注
SLP	4	8G	100G	
AI-Container	4	16G	500G	
mysql	4	8G	100G	
redis	4	8G	100G	
mongodb	4	8G	100G	

3.3 部署步骤

自学习平台以 docker 容器的方式进行部署，完整的部署步骤可以参考阿里巴巴智能语音专有云安装目录的部署文档。

本节简单介绍专有云的安装步骤。

假定专有云安装包放置在 `oss://swap/enterprise/v2.5/service`

假定安装位置在本地目录 `/home/admin`

3.3.1 下载专有云安装包

可以使用 `ossutil` 命令从 `oss` 下载安装包：

```
ossutil cp -r oss://swap/enterprise/v2.5/service/ /home/admin/  
mv /home/admin/enterprise/v2.5/service /home/admin/service  
rm -r /home/admin/enterprise
```

3.3.2 安装 docker

以下步骤可以参考 docker 官方文档：

<https://docs.docker.com/install/linux/docker-ce/centos/#install-docker-ce>

安装 docker 依赖的组件

```
sudo yum install -y \  
    yum-utils \  
    device-mapper-persistent-data \  
    lvm2
```

添加 docker 使用的安装员

```
sudo yum-config-manager \  
    --add-repo \  
    https://download.docker.com/linux/centos/docker-ce.repo
```

```
https://download.docker.com/linux/centos/docker-ce.repo
```

安装 docker-ce

```
sudo yum install docker-ce
```

启动 docker 服务

```
sudo systemctl start docker
sudo systemctl enable docker
```

测试 docker 是否正常运行

如果能看到消息“Hello from Docker!”则说明安装成功。

```
sudo docker run hello-world
```

3.3.3 启动服务

注意：需要确保授权服务 (APES) 已启动

执行下面的命令启动服务：

```
cd /home/admin/service
sudo chmod +x bin/*
sudo sh bin/init.sh
sudo sh bin/start.sh
```

执行下面的命令确认服务是否成功：

```
sudo docker ps -a
tail -50 logs/nls-cloud-slp/application.log
tail -50 logs/nls-cloud-ai-container/application.log
```

3.4 服务维护

本节介绍服务维护需要的一些操作。

3.4.1 服务重启

执行下面的命令可以重启服务：

```
cd /home/admin/service
sudo sh bin/stop.sh
sudo sh bin/start.sh
```

3.4.2 进入容器

执行下面的命令可以进入 docker 容器内：

```
sudo docker exec -it nls-cloud-slp /bin/bash
```

在容器中可以找到如下文件：

```
# 配置文件
/home/admin/nls-cloud-slp/conf
#日志文件，会被映射到宿主机/home/admin/service/logs/nls-cloud-slp
/home/admin/nls-cloud-slp/logs
```

可以使用 exit 命令退出容器，返回宿主机。

阿里巴巴 智能语音

4 快速开始

4.1 导入文件

自学习平台基于 mongodb (gridfs) 构建存储系统。体积较大的文件 (对象) 都会使用 mongodb 进行持久化存储。所有涉及文件的 API 使用到的路径参数都是指 mongodb 中的路径 (gridfs 文件系统的 filename 字段)。如果需要将硬盘上的文件添加到自学习系统中。需要先将文件导入 mongodb。

有三种方式将文件导入系统：

- 通过自学习平台的控制台 (图形化界面)
- 使用自学习平台的文件 API (通过 shell , python 或 java 等方式访问)
- 使用 mongofiles 命令 (适合熟悉 mongodb 的人员使用)

使用控制台导入文件

4.1.1 使用控制台

通过浏览器访问 <http://nls-slp.aliyun.test:8701/files/upload> 即可进入文件上传页面。

完整的控制台文件管理说明请参考 7.1 节。

4.1.2 使用存储 API

下面的示例会将本地文件 /home/admin/demo.txt 上传到存储系统并设置存储路径为 slp/tmp/demo.txt。

```
curl -X POST \  
  --header 'Content-Type: multipart/form-data' \  
  --header 'Accept: application/json' \  
  --header 'X-NLS-Token: default' \  
  -F "path=slp/tmp/demo.txt" \  
  -F "file=@/home/admin/demo.txt" \  
  'http://nls-slp.aliyun.test:8701/api/files/upload'
```

完整的文件管理 API 文档请参考 6.7 节。

4.1.3 使用 mongofiles 命令

下面的示例会将本地文件 /home/admin/demo.txt 上传到存储系统并设置存储路径为 slp/tmp/demo.txt。

```
export PATH=$PATH:/home/admin/service/bin  
mongo_uri='mongodb://nls-  
cloud:88630dd3d489a617b635d51ef7eedfe@127.0.0.1:7031/?authSource=admin'  
mongofiles --uri ${mongo_uri} -d nls-cloud put slp/tmp/demo.txt -l /home/admin/demo.txt
```

注意：示例中使用了专有云环境中的默认设置，实际部署环境可能有差异。

附录 9.1 提供了更多关于 mongofiles 命令的使用说明。

4.2 定制语言模型

训练过程大致包含如下步骤：

- 配置基础模型（仅在第一次训练时执行）
- 准备训练数据
- 将训练数据导入自学习系统，并创建数据集
- 创建一个定制模型，并将数据集添加到这个定制模型
- 启动定制模型训练，并等待模型训练完成

4.2.1 配置基础模型

注意：此步骤在部署完成后只需执行一次。

确认基础模型是否已自动配置

系统启动时会自动尝试创建基础模型，可以通过下面的命令确认是否基础模型已经创建：

```
curl -X GET \
  --header 'Content-Type: multipart/form-data' \
  --header 'Accept: application/json' \
  --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/bases'
```

如果返回的结果里 total_items 的值不是 0，表示系统已经存在定制模型，可以跳转到 4.2.2 节继续。

通过内置脚本添加基础模型

安装包提供了一个自动初始化基础模型的脚本，可以通过下面的命令添加基础模型：

```
chmod +x /home/admin/service/resource/slp/init-lm.sh
/home/admin/service/resource/slp/init-lm.sh
```

如果以上命令成功执行，可以跳转到 4.2.2 节继续。

准备英文 TN 文件

这个文件可以在专有云安装目录找到，路径是/home/admin/service/resource/slp/common-eng-tn.txt

执行下面的命令将英文 TN 文件上传到自学习平台：（转到[错误!未找到引用源](#)。节查看文件 API 的详细文档）

```
curl -X POST \
  --header 'Content-Type: multipart/form-data' \
  --header 'Accept: application/json' \
  --header 'X-NLS-Token: default' \
  -F "path=slp/v2/asr/lm/inputs/common-eng-tn.txt" \
  -F "file=@/home/admin/service/resource/slp/common-eng-tn.txt" \
  'http://nls-slp.aliyun.test:8701/api/files/upload'
```

准备分词词典文件

这个文件可以在 ASR 的模型目录找到：/home/admin/service/resource/asr/default/models/lm/segment.dict。

注意：不同的 ASR 模型对应的分词词典互不兼容，必须确保使用了正确的文件。

使用下面的命令将分词词典上传到自学习平台：

```
curl -X POST \
```

```
--header 'Content-Type: multipart/form-data' \  
--header 'Accept: application/json' \  
--header 'X-NLS-Token: default' \  
-F "path=slp/v2/asr/lm/inputs/common-seg-dict.txt" \  
-F "file=@/home/admin/service/resource/asr/default/models/lm/segment.dict" \  
'http://nls-slp.aliyun.test:8701/api/files/upload'
```

创建基础模型

执行下面的命令创建基础模型：（参考[错误!未找到引用源。](#)节）

```
curl -X POST \  
--header 'Content-Type: application/json' \  
--header 'Accept: application/json' \  
--header 'X-NLS-User: default' \  
--header 'X-NLS-Token: default' \  
-d '{  
  "id": "common",  
  "name": "通用模型",  
  "description": "通用领域模型",  
  "eng_tn": "slp/v2/asr/lm/inputs/common-eng-tn.txt",  
  "seg_dict": "slp/v2/asr/lm/inputs/common-seg-dict.txt"  
}' 'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/bases'
```

4.2.2 准备训练数据

将训练数据准备好，保存成文本文件，放在本地磁盘/home/admin/demo-data-lm.txt。

数据格式的具体要求可以参考 5.1.1 节。

可以通过下面的命令生成一个测试数据：

```
echo '
```

君子曰：学不可以已。

青，取之于蓝，而青于蓝；冰，水为之，而寒于水。木直中绳，鞣以为轮，其曲中规。虽有槁暴，不复挺者，鞣使之然也。故木受绳则直，金就砺则利，君子博学而日参省乎己，则知明而行无过矣。

故不登高山，不知天之高也；不临深溪，不知地之厚也；不闻先王之遗言，不知学问之大也。干、越、夷、貉之子，生而同声，长而异俗，教使之然也。诗曰：“嗟尔君子，无恒安息。靖共尔位，好是正直。神之听之，介尔景福。”神莫大于化道，福莫长于无祸。（此段教材无）

吾尝终日而思矣，不如须臾之所学也；吾尝跂而望矣，不如登高之博见也。登高而招，臂非加长也，而见者远；顺风而呼，声非加疾也，而闻者彰。假舆马者，非利足也，而致千里；假舟楫者，非能水也，而绝江河。君子生非异也，善假于物也。

南方有鸟焉，名曰蒙鸠，以羽为巢，而编之以发，系之苇苕，风至苕折，卵破子死。巢非不完也，所系者然也。西方有木焉，名曰射干，茎长四寸，生于高山之上，而临百仞之渊，木茎非能长也，所立者然也。蓬生麻中，不扶而直；白沙在涅，与之俱黑。兰槐之根是为芷，其渐之滫，君子不近，庶人不服。其质非不美也，所渐者然也。故君子居必择乡，游必就士，所以防邪辟而近中正也。

物类之起，必有所始。荣辱之来，必象其德。肉腐出虫，鱼枯生蠹。怠慢忘身，祸灾乃作。强自取柱，柔自取束。邪秽在身，怨之所构。施薪若一，火就燥也，平地若一，水就湿也。草木畴生，禽兽群焉，物各从其类也。是故质的张，而弓矢至焉；林木茂，而斧斤至焉；树成荫，而众鸟息焉。醯酸，而蚋聚焉。故言有招祸也，行有招辱也，君子慎其所立乎！（此段教材无）

积土成山，风雨兴焉；积水成渊，蛟龙生焉；积善成德，而神明自得，圣心备焉。故不积跬步，无以至千里；不积小流，无以成江海。骐骥一跃，不能十步；弩马十驾，功在不舍。锲而舍之，朽木不折；锲而不舍，金石可镂。蚓无爪牙之利，筋骨之强，上食埃土，下饮黄泉，用心一也。蟹六跪而二螯，非蛇鳝之穴无可寄托者，用心躁也。

是故无冥冥之志者，无昭昭之明；无惛惛之事者，无赫赫之功。行衢道者不至，事两君者不容。目不能两视而明，耳不能两听而聪。螭蛇无足而飞，鼯鼠五技而穷。《诗》曰：“尸鸠在桑，其子七兮。淑人君子，其仪一兮。其仪一兮，心如结兮！”故君子结于一也。

昔者瓠巴鼓瑟，而流鱼出听；伯牙鼓琴，而六马仰秣。故声无小而无闻，行无隐而不形。玉在山而草润，渊生珠而崖不枯。为善不积邪？安有不闻者乎？

学恶乎始？恶乎终？曰：其数则始乎诵经，终乎读礼；其义则始乎为士，终乎为圣人，真积力久则入，学至乎没而后止也。故学数有终，若其义则不可须臾舍也。为之，人也；舍之，禽兽也。故书者，政事之纪也；诗者，中声之所止也；礼者，法之大分，类之纲纪也。故学至乎礼而止矣。夫是之谓道德之极。礼之敬文也，乐之中和也，诗书之博也，春秋之微也，在天地之间者毕矣。君子之学也，入乎耳，着乎心，布乎四体，形乎动静。端而言，蠕而动，一可以为法则。小人之学也，入乎耳，出乎口；口耳之间，则四寸耳，曷足以美七尺之躯哉！古之学者为己，今之学者为人。君子之学也，以美其身；小人之学也，以为禽犊。故不问而告谓之傲，问一而告二谓之囋。傲、非也，囋、非也；君子如向矣。

学莫便乎近其人。礼乐法而不说，诗书故而不切，春秋约而不速。方其人之习君子之说，则尊以遍矣，周于世矣。故曰：学莫便乎近其人。

学之经莫速乎好其人，隆礼次之。上不能好其人，下不能隆礼，安特将学杂识志，顺诗书而已耳。则末世穷年，不免为陋儒而已。将原先王，本仁义，则礼正其经纬蹊径也。若挈裘领，诎五指而顿之，顺者不可胜数也。不道礼宪，以诗书为之，譬之犹以指测河也，以戈春黍也，以锥餐壶也，不可以得之矣。故隆礼，虽未明，法士也；不隆礼，虽察辩，散儒也。

问楛者，勿告也；告楛者，勿问也；说楛者，勿听也。有争气者，勿与辩也。故必由其道至，然后接之；非其道则避之。故礼恭，而后可与言道之方；辞顺，而后可与言道之理；色从而后可与言道之致。故未可与言而言，谓之傲；可与言而不言，谓之隐；不观气色而言，谓之瞽。故君子不傲、不隐、不瞽，谨顺其身。诗曰：“匪交匪舒，天子所予。”此之谓也。

百发失一，不足谓善射；千里蹞步不至，不足谓善御；伦类不通，仁义不一，不足谓善学。学也者，固学一之也。一出焉，一入焉，涂巷之人也；其善者少，不善者多，桀纣盗跖也；全之尽之，然后学者也。

君子知夫不全不粹之不足以为美也，故诵数以贯之，思索以通之，为其人以处之，除其害者以持养之。使目非是无欲见也，使耳非是无欲闻也，使口非是无欲言也，使心非是无欲虑也。及至其致好之也，目好之五色，耳好之五声，口好之五味，心利之有天下。是故权利不能倾也，群众不能移也，天下不能荡也。生乎由是，死乎由是，夫是之谓德操。德操然后能定，能定然后能应。能定能应，夫是之谓成人。天见其明，地见其光，君子贵其全也。

```
' > /home/admin/demo-data-lm.txt
```

4.2.3 导入训练数据

执行下面的命令将训练数据导入到自学习平台：

```
curl -X POST \
  --header 'Content-Type: multipart/form-data' \
  --header 'Accept: application/json' \
  --header 'X-NLS-Token: default' \
  -F "path=slp/tmp/demo-data-lm.txt" \
  -F "file=@/home/admin/demo-data-lm.txt" \
  'http://nls-slp.aliyun.test:8701/api/files/upload'
```

执行下面的命令使用导入的数据创建数据集，这里我们使用 demo-data 作为数据集 ID：

```
curl -X POST \
  --header 'Content-Type: application/json' \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
```



```
--header 'X-NLS-Token: default' \
-d '{
  "id": "demo-data",
  "name": "示例数据集",
  "description": "这是一个示例数据集",
  "url": "slp/tmp/demo-data-lm.txt"
}' 'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/data'
```

执行下面的命令查询数据集状态，等待数据集的状态由 Fetching 变为 Ready，如果创建数据集使用的 url 不正确，数据集状态会改变为 FetchingFailed，这个时候可以重新设置 url 或者重新创建数据集。（参考 6.4.1 节）

```
curl -X GET \
--header 'Accept: application/json' \
--header 'X-NLS-User: default' \
--header 'X-NLS-Token: default' \
'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/data/demo-data'
```

4.2.4 创建定制模型

执行下面的命令创建定制模型，这里我们使用 demo-model 作为模型 ID，参数 base_id 用来指定基础模型，这里我们使用在 4.2.1 节创建的基础模型 common：

```
curl -X POST \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'X-NLS-User: default' \
--header 'X-NLS-Token: default' \
-d '{
  "id": "demo-model",
  "name": "示例模型",
  "description": "这是一个示例模型",
  "base_id": "common"
}' 'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/models'
```

执行下面的命令将 4.2.3 节创建的数据集添加到新创建的定制模型：（注意：这里只是绑定关系，并不是归属关系，一个数据集可以用于多个定制模型，一个定制模型也可以使用多个数据集）

```
curl -X POST \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'X-NLS-User: default' \
--header 'X-NLS-Token: default' \
'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/operations/add-data-to-model?data_id=demo-data&model_id=demo-model'
```

4.2.5 训练定制模型

执行下面的命令启动模型训练：

```
curl -X POST \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'X-NLS-User: default' \
--header 'X-NLS-Token: default' \
'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/operations/train-model?model_id=demo-model'
```

执行下面的命令查询模型状态，等待模型状态由 Training 变为 Deploying，最终变为 Deployed。Training 的过程最多持续 5 分钟，Deploying 的过程最多持续 1 分钟：

```
curl -X GET --header 'Accept: application/json' \
  --header 'X-NLS-User: default' --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/models/demo-model'
```

至此定制模型的训练过程已经结束。

4.3 定制类热词

执行下面的命令添加类热词，这里我们添加了“北京”和“上海”这两个地名：（参考 6.5.1 节）

```
curl -X POST \
  --header 'Content-Type: application/json' --header 'Accept: application/json' \
  --header 'X-NLS-User: default' --header 'X-NLS-Token: default' \
  -d '{
    "id": "demo-vocab",
    "name": "示例词表",
    "description": "这是一个示例词表",
    "words": ["北京", "上海"]
  }' 'http://nls-slp.aliyun.test:8701/api/v1/asr/class-vocabs'
```

4.4 定制泛热词

执行下面的命令添加类热词，这里我们添加了“苹果”和“西瓜”这两个词语，并且分别设置权重为 2 和 3：（API 参考 6.5.1 节，词语权重参考 5.3 节）

```
curl -X POST \
  --header 'Content-Type: application/json' --header 'Accept: application/json' \
  --header 'X-NLS-User: default' --header 'X-NLS-Token: default' \
  -d '{
    "id": "demo-vocab",
    "name": "示例词表",
    "description": "这是一个示例词表",
    "word_weights": {"苹果": 2, "西瓜": 3}
  }' 'http://nls-slp.aliyun.test:8701/api/v1/asr/vocabs'
```

5 功能说明

5.1 定制语言模型

5.1.1 数据准备要求

定制模型对数据文件格式有如下要求：

- 必须是文本文件；
- 必须使用 UTF-8 编码，并且不能带有文件头 BOM；

- 控制每行的长度在 500 个字符以内 (不是字节);
- 文本中的数字最好按照发音替换为对应的汉字。例如：“58.9 元”需要转换为“五十八点九元”
- 文件中需要至少有一行为句子 (大于 4 个词)

5.1.2 数据准备常见思路

正常训练文本：

业务相关的文本(例如法院庭审报告;电话客服业务话术文本;智能客服若后续有自然语音理解也可以添加理解模块规则文本;业务相关的声学模型训练对应的标注文本)

新业务上线前：

添加包含新增业务关键词的上述训练文本。在业务上线前进行非常重要。

小范围调优数据：

添加识别不准确的关键词/句。一个关键词/句，在训练文本中单独占一行。另外，可以对关键词进行加强，拷贝多行，例如 10 行。如果没有效果，可以再适当增加拷贝行数。

注意：

(1)不要拷贝太多导致影响其他词识别，这个要在实际业务中尝试后总结经验。

(2)对于识别不好关键词，需要结合听语音，先判断一下关键词识别不准确的原因不是因为本身说的不清晰或者个别音频质量不好。如果是这种情况，则不要为了解决这种极端情况而优化过猛。

(3)如果正常训练文本里面已经有大量的这些关键词，不建议再进行过多的单独加强。

5.1.3 模型训练参数

配置基础模型时可以设置如下模型训练参数：(基于此基础模型的所有定制模型都会使用)

参数名	说明
order	训练出来语言模型的阶数。String 类型。默认值是 4。
options	训练参数。String 类型。默认是“-kndiscount -interpolate -gt1min 1 -gt2min 1 -gt3min 1 -gt4min 1”。

5.1.4 模型训练任务信息

训练任务完成之后会生成任务摘要，包含如下信息：

任务参数名	说明
sentence number	训练的总句子数(一行一句)
unigram count	一元模型数
bigram count	二元模型数
trigram count	三元模型数
qualgram count	四元模型数

训练结束后，系统会将训练日志和调试数据保存到存储系统，用于排查问题。可以通过文件管理 API 下载相关文件。

5.2 定制类热词

5.2.1 格式要求

定制类热词对词表有如下要求：

- 每个词表最多添加 128 个词；
- 每个词最多 32 个字符；
- 词语中的数字需要按照发音替换为对应的汉字。例如：“58.9 元”需要转换为“五十八点九元”

5.3 定制泛热词

5.3.1 格式要求

定制泛热词对词表有如下要求：

- 每个词表最多添加 128 个词；
- 每个词最多 32 个字符；
- 词语中的数字需要按照发音替换为对应的汉字。例如：“58.9 元”需要转换为“五十八点九元”

5.3.2 权重设置

可以针对每个词设置不同的权重，权重取值为-6 到 5 之间的整数值。大于零的权重用来增加词语被识别的概率，小于零的权重用来减小词语被识别的概率。权重-6 表示尽量不识别出这个词语。权重 2 是常用的值，如果效果不明显可以适当增加权重，但是当权重较大时可能会有一些负面效果，导致其他词语识别不准确。识别率测试 (beta)

5.3.3 数据准备方法

测试数据的准备方法与声学模型的训练数据相同。

5.3.4 测试参数

参数名	说明
customization_id	指定定制语言模型 ID。
vocabulary_id	指定泛热词 ID。

6 系统 API

自学习平台提供了如下几组 API：

- 定制语言模型
- 定制声学模型 (beta)
- 定制类热词
- 定制泛热词
- 识别率测试 (beta)
- 文件管理

6.1 调用约定

API 路径

路径是 API 的唯一标识，每个 API 的路径由 HTTP 的 Method 和 URL 组成，例如列举定制语言模型数据集的 API 路径是：GET /api/v2/asr/lm/data

参数位置

用来表示参数出现的位置。有如几个位置：

名称	说明
Query	URL 的查询参数，例如： http://nls-slp.aliyun.test:8701/api/asr/lm/data?model_id=demo-model。
Path	URL 路径的一部分，例如： http://nls-slp.aliyun.test:8701/api/v2/asr/lm/data/ demo-data 。
Header	出现在 HTTP 的请求头。
Body	HTTP 的消息体。

通用输入参数

大部分 API 使用了如下两个通用参数：

名称	位置	说明
X-NLS-Token	Header	鉴权令牌，用来确认调用方式可信任的，服务部署后的默认配置是“default”。所有的 API 调用都必须设置此参数。
X-NLS-User	Header	资源所属的用户 ID，用来支持多租户。如果不需要多租户，可以使用“default”。只有涉及到多租户的资源才需要设置此参数。如果没有特殊说明，都需要设置此参数。
Accept	Header	如果没有特殊说明都需要设置为 application/json

通用输出参数

大部分 API 的输出参数都包含如下参数：

名称	说明
request_id	请求的唯一 ID，可以通过请求 ID 对日志进行过滤，以排查问题。

错误消息

出现错误时服务端会返回 200 之外的状态码，400 表示客户端错误，500 表示服务端错误。

出现错误时 HTTP 响应消息的消息体是 JSON 对象，包含以下参数：

名称	说明
request_id	请求的唯一 ID，可以通过请求 ID 对日志进行过滤，以排查问题。
url	HTTP 请求的 URL。
status	HTTP 状态码。
error_code	数字格式的错误码。
error_message	错误消息。

6.2 错误码

错误码	错误名称	说明
40040001	ASR_VOCAB_ERROR	定制类热词或定制泛热词相关错误。
40040002	ASR_MODEL_ERROR	定制语言模型或定制声学模型相关错误。
40040003	ASR_TEST_ERROR	识别率测试相关错误。
40041001	NOT_FOUND	指定的 ID 无效，无法根据 ID 找到指定的资源。
40041002	PARAMETER_ERROR	创建资源时设置了无效的参数
40041003	EXCEED_LIMIT	资源数量超过限制，无法创建新的资源。

6.3 通用对象定义

有些对象在多个 API 中重复使用，在本节提前给出对象参数说明。

6.3.1 分页对象 NlsPage

分页对象用来包含列表查询接口的结果。例如列举定制语言模型数据集的 API 会返回一个分页对象，这个分页对象里包含了一个定制语言模型数据集的对象列表。

名称	说明
----	----

content	是一个数组，包含了分页查询的对象列表，对象的类型由具体的 API 决定。
total_pages	分页总数量。
total_items	对象总数量。
page_number	本次查询使用的页号。
page_size	本次查询使用的页大小。

6.3.2 定制语言模型数据集对象 AsrLmData

参数名	说明
id	数据集 ID，数据集的唯一标识。
name	数据集名称。
description	数据集描述信息。可以为空。
size	数据集大小。
md5	数据集 MD5 值。
url	创建数据集使用的 URL，目前支持使用 mongodb/gridfs 中的 filename。
status	数据集状态。可能的状态： Fetching: 正在将数据集从 url 导入到自学习系统中； FetchingFailed: 复制数据集出现错误； Ready: 数据集导入到成功。
error_message	错误消息。
create_time	数据集创建时间。
update_time	数据集更新时间。

6.3.3 定制语言模型对象 AsrLmModel

参数名	说明
id	模型 ID，模型的唯一标识。
name	模型名称。
description	模型描述信息。可以为空。
base_id	基础模型 ID。
size	模型大小。
md5	模型 MD5 值。
url	创建数据集使用的 URL，目前支持使用 mongodb/gridfs 中的 filename。
train_params	模型训练参数，参考 5.1.3 节。

train_summary	训练任务的摘要信息。参考 5.1.4 节。
status	模型状态。可能的状态： Empty：模型新创建，还没有训练过； Training：模型正在训练中； TrainingFailed：模型训练失败； Ready：模型训练成功并未上线； Deploying：模型正在上线或下线； Deployed：模型已上线。
error_message	错误消息。
create_time	模型创建时间。
update_time	模型更新时间。

6.3.4 定制语言模型基础模型对象 AsrLmBase

参数名	说明
id	基础模型 ID，基础模型的唯一标识。
name	基础模型名称。
description	基础模型描述信息。可以为空。
seg_dict	分词词典的存储路径。
eng_tn	英文 TN 文件的存储路径。
train_params	模型训练参数，参考 5.1.3 节。
create_time	基础模型创建时间。
update_time	基础模型更新时间。

6.3.5 定制类热词对象 AsrClassVocab

参数名	说明
id	词表 ID，词表的唯一标识。
name	词表名称。
description	词表描述信息。可以为空。
words	要添加的词，是一个字符串数组。
create_time	词表创建时间。
update_time	词表更新时间。

6.3.6 定制泛热词对象 AsrVocab

参数名	说明
id	词表 ID，词表的唯一标识。
name	词表名称。
description	词表描述信息。可以为空。
word_weights	要添加的词，是一个词典，键为词，值为权重。
create_time	词表创建时间。
update_time	词表更新时间。

6.4 定制语言模型

定制语言模型 API 分为以下几组：

- 数据集增删改查：/api/v2/asr/lm/data*
- 模型增删改查：/api/v2/asr/lm/models*
- 模型训练和上下线：/api/v2/asr/lm/operations*
- 基础模型增删改查：/api/v2/asr/lm/bases

6.4.1 创建数据集

路径：

POST /api/v2/asr/lm/data

输入参数：

名称	位置	说明
data	Body	数据集对象 (AsrLmData)。可以设置如下参数： id：数据集 ID，可选，如果为空则自动生成； name：数据集名称； description：数据集描述信息，可选； url：数据集位置，mongodb/gridfs 中的 filename，创建成功之后不可修改。

请求示例：

```
curl -X POST \
  --header 'Content-Type: application/json' \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
  --header 'X-NLS-Token: default' \
  -d '{
    "id": "demo-data",
    "name": "示例数据",
    "description": "这是一个示例数据",
  }'
```

```
"url": "slp/tmp/demo.txt"
}' 'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/data'
```

输出参数：

名称	说明
data_id	数据集 ID。

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff",
  "data_id": "demo-data"
}
```

6.4.2 列举数据集

路径：

```
GET /api/v2/asr/lm/data
```

输入参数：

名称	位置	说明
page_number	Query	页号，从 1 开始编号。可选，默认值是 1。
page_size	Query	页大小，范围在 1 到 100 之间。可选，默认是 10。
model_id	Query	模型 ID，用于搜索被指定模型使用到的数据。可选，默认列出所有数据集。

请求示例：

```
curl -X GET \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
  --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/data'
```

输出参数：

名称	说明
page	数据集对象 (AsrLmData) 的分页结果。

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff",
  "page": {
    "content": [
      {
        "id": "1b64bee9994749f2a67eadac6379f80c",
        "name": "示例数据集",
        "description": "这是一个示例数据集",
        "size": 7777404,
        "md5": "39326cf690e384735355a385ec1e7a00",
        "url": "slp/tmp/demo-data-lm.txt",
        "status": "Ready",
        "create_time": "2018-10-31 17:20:39",

```

```
        "update_time": "2018-10-31 17:20:39"
    }
],
"total_pages": 1,
"total_items": 1,
"page_number": 1,
"page_size": 10
}
}
```

6.4.3 查询数据集

路径：

```
GET /api/v2/asr/lm/data/{data_id}
```

输入参数：

名称	位置	说明
data_id	Path	要查询的数据集 ID。

请求示例：

```
curl -X GET \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
  --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/data/demo-data'
```

输出参数：

名称	位置	说明
data	Path	数据集对象 (AsrLmData)。

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff",
  "data": {
    "id": "demo-data",
    "name": "示例数据",
    "description": "这是一个示例数据",
    "size": 3919,
    "md5": "8c8c3bb84c3b684d9d80b7069cef81ee",
    "url": "slp/tmp/demo-data-lm.txt",
    "status": "Ready",
    "create_time": "2018-11-01 15:59:18",
    "update_time": "2018-11-01 15:59:18"
  }
}
```

6.4.4 更新数据集

数据集创建成功之后不可以修改 url。只能修改名称和备注信息。

路径：

```
POST /api/v2/asr/lm/data
```

输入参数：

名称	位置	说明
data	Body	数据集对象 (AsrLmData)。可以设置如下参数： id：需要更新的数据集的 ID； name：数据集名称； description：数据集描述信息，可选。

请求示例：

```
curl -X PUT \  
  --header 'Content-Type: application/json' \  
  --header 'Accept: application/json' \  
  --header 'X-NLS-User: default' \  
  --header 'X-NLS-Token: default' \  
  -d '{  
    "id": "demo-data",  
    "name": "示例数据 (二)",  
    "description": "这是另一个示例数据"  
  }' 'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/data'
```

响应示例：

```
{  
  "request_id": "7130914d32a3441db06747523675d9ff"  
}
```

6.4.5 删除数据集

路径：

```
DELETE /api/v2/asr/lm/data/{data_id}
```

输入参数：

名称	位置	说明
data_id	Path	要删除的数据集 ID
force	Query	是否强制删除。如果为 true，则允许在任何状态下删除数据，并解除自动解除数据和模型的绑定关系。可选，默认是 false。

请求示例：

```
curl -X DELETE \  
  --header 'Accept: application/json' \  
  --header 'X-NLS-User: default' \  
  --header 'X-NLS-Token: default' \  
  'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/data/demo-data'
```

响应示例：

```
{  
  "request_id": "7130914d32a3441db06747523675d9ff"  
}
```

6.4.6 创建模型

路径：

POST /api/v2/asr/lm/models

输入参数：

名称	位置	说明
model	Body	模型对象 (AsrLmModel)。可以设置如下参数： id：模型 ID，可选，如果为空则自动生成； name：模型名称； description：模型描述信息，可选； base_id：基础模型 ID，创建成功之后不可修改。

请求示例：

```
curl -X POST \  
  --header 'Content-Type: application/json' \  
  --header 'Accept: application/json' \  
  --header 'X-NLS-User: default' \  
  --header 'X-NLS-Token: default' \  
  -d '{  
    "id": "demo-model",  
    "name": "示例模型",  
    "description": "这是一个示例模型",  
    "base_id": "common"  
  }' 'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/models'
```

输出参数：

名称	说明
model_id	模型 ID。

响应示例：

```
{  
  "request_id": "7130914d32a3441db06747523675d9ff",  
  "model_id": "demo-model"  
}
```

6.4.7 列举模型

路径：

GET /api/v2/asr/lm/models

输入参数：

名称	位置	说明
page_number	Query	页号，从 1 开始编号。可选，默认值是 1。
page_size	Query	页大小，范围在 1 到 100 之间。可选，默认是 10。

data_id	Query	数据集 ID，用于搜索使用了指定数据的模型。可选，默认列出所有模型。
---------	-------	------------------------------------

请求示例：

```
curl -X GET \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
  --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/models'
```

输出参数：

名称	说明
page	模型对象 (AsrLmModel) 的分页结果。

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff",
  "page": {
    "content": [
      {
        "id": "demo-model",
        "name": "示例模型",
        "description": "这是一个示例模型",
        "size": 0,
        "status": "Empty",
        "create_time": "2018-11-01 17:05:21",
        "update_time": "2018-11-01 17:05:21",
        "base_id": "common"
      }
    ],
    "total_pages": 1,
    "total_items": 1,
    "page_number": 1,
    "page_size": 10
  }
}
```

6.4.8 查询模型

路径：

```
GET /api/v2/asr/lm/models/{model_id}
```

输入参数：

名称	位置	说明
model_id	Path	要查询的模型 ID。

请求示例：

```
curl -X GET \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
  --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/models/demo-model'
```

输出参数：

名称	说明
model	模型对象 (AsrLmModel)。

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff",
  "model": {
    "id": "demo-model",
    "name": "示例模型",
    "description": "这是一个示例模型",
    "size": 0,
    "status": "Empty",
    "create_time": "2018-11-01 16:13:48",
    "update_time": "2018-11-01 16:13:48",
    "base_id": "common"
  }
}
```

6.4.9 更新模型

模型创建成功之后不可以修改基础模型。只能修改名称和备注信息。

路径：

POST /api/v2/asr/lm/models

输入参数：

名称	位置	说明
model	Body	模型对象 (AsrLmModel)。可以设置如下参数： id：需要更新的模型的 ID； name：模型名称； description：模型描述信息，可选。

请求示例：

```
curl -X PUT \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'X-NLS-User: default' \
--header 'X-NLS-Token: default' \
-d '{
  "id": "demo-model",
  "name": "示例模型 (二)",
  "description": "这是另一个示例模型"
}' 'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/models'
```

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff"
}
```

6.4.10 删除模型

路径：

```
DELETE /api/v2/asr/lm/models/{model_id}
```

输入参数：

名称	位置	说明
model_id	Path	要删除的模型 ID
force	Query	是否强制删除。如果为 true，则允许在任何状态下删除模型以及相关资源。

请求示例：

```
curl -X DELETE \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
  --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/models/demo-model'
```

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff"
}
```

6.4.11 添加数据集到模型

路径：

```
POST /api/v2/asr/lm/operations/add-data-to-model
```

输入参数：

名称	位置	说明
data_id	Query	数据集 ID。
model_id	Query	模型 ID。

请求示例：

```
curl -X POST \
  --header 'Content-Type: application/json' \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
  --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/operations/add-data-to-model?data_id=demo-data&model_id=demo-model'
```

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff"
}
```

6.4.12 从模型删除数据集

路径：

POST /api/v2/asr/am/operations/remove-data-from-model

输入参数：

名称	位置	说明
data_id	Query	数据集 ID。
model_id	Query	模型 ID。

请求示例：

```
curl -X POST \
  --header 'Content-Type: application/json' \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
  --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/operations/remove-data-from-model?data_id=demo-data&model_id=demo-model'
```

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff"
}
```

6.4.13 开始模型训练

路径：

POST /api/v2/asr/lm/operations/train-model

输入参数：

名称	位置	说明
model_id	Query	模型 ID。

请求示例：

```
curl -X POST \
  --header 'Content-Type: application/json' \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
  --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/operations/train-model?model_id=demo-model'
```

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff"
}
```

6.4.14 停止模型训练

路径：

POST /api/v2/asr/lm/operations/stop-train-model

输入参数：

名称	位置	说明
----	----	----

model_id	Query	模型 ID。
----------	-------	--------

请求示例：

```
curl -X POST \
  --header 'Content-Type: application/json' \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
  --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/operations/stop-train-model?model_id=demo-model'
```

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff"
}
```

6.4.15 上线模型

路径：

```
POST /api/v2/asr/lm/operations/online-model
```

输入参数：

名称	位置	说明
model_id	Query	模型 ID。

请求示例：

```
curl -X POST \
  --header 'Content-Type: application/json' \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
  --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/operations/online-model?model_id=demo-model'
```

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff"
}
```

6.4.16 下线模型

路径：

```
POST /api/v2/asr/lm/operations/offline-model
```

输入参数：

名称	位置	说明
model_id	Query	模型 ID。

请求示例：

```
curl -X POST \
  --header 'Content-Type: application/json' \
```

```
--header 'Accept: application/json' \
--header 'X-NLS-User: default' \
--header 'X-NLS-Token: default' \
'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/operations/offline-model?model_id=demo-model'
```

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff"
}
```

6.4.17 创建基础模型

路径：

```
POST /api/v2/asr/lm/bases
```

输入参数：

名称	位置	说明
base	Body	基础模型对象 (AsrLmBase)。可以设置如下参数： id: 基础模型 ID ； name：基础模型名称； description：基础模型描述信息，可选； seg_dict：分词词典的存储路径； eng_tn：英文 TN 文件的存储路径。

请求示例：

```
curl -X POST \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'X-NLS-User: default' \
--header 'X-NLS-Token: default' \
-d '{
  "id": "common",
  "name": "通用模型",
  "description": "通用领域模型",
  "seg_dict": "slp/v2/asr/lm/inputs/common-seg-dict.txt",
  "eng_tn": "slp/v2/asr/lm/inputs/common-eng-tn.txt"
}' 'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/bases'
```

输出参数：

名称	说明
base_id	基础模型 ID。

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff",
  "base_id": "common"
}
```

6.4.18 列举基础模型

基础模型相关 API 不需要传递 X-NLS-User 参数。基础模型是全局资源，不归属于单个用户。

路径：

GET /api/v2/asr/lm/bases

输入参数：

名称	位置	说明
page_number	Query	页号，从 1 开始编号。可选，默认值是 1。
page_size	Query	页大小，范围在 1 到 100 之间。可选，默认是 10。

请求示例：

```
curl -X GET \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
  --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/bases'
```

输出参数：

名称	说明
page	基础模型对象 (AsrLmBase) 的分页结果。

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff",
  "page": {
    "content": [
      {
        "id": "common",
        "name": "通用",
        "description": "通用",
        "create_time": "2018-07-04 22:17:46",
        "update_time": "2018-10-19 10:13:51",
        "seg_dict": "slp/v2/asr/lm/inputs/common-seg-dict.txt",
        "eng_tn": "slp/v2/asr/lm/inputs/common-eng-tn.txt"
      }
    ],
    "total_pages": 1,
    "total_items": 2,
    "page_number": 1,
    "page_size": 10
  }
}
```

6.4.19 查询基础模型

路径：

DELETE /api/v2/asr/lm/bases/{base_id}

输入参数：

名称	位置	说明
base_id	Path	要查询的基础模型 ID。

请求示例：

```
curl -X GET \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
  --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/bases/common'
```

输出参数：

名称	位置	说明
base	Path	模型对象 (AsrLmBase)。

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff",
  "base": {
    "id": "common",
    "name": "通用模型",
    "description": "通用领域通用",
    "seg_dict": "slp/v2/asr/lm/inputs/common-seg-dict.txt",
    "eng_tn": "slp/v2/asr/lm/inputs/common-eng-tn.txt",
    "create_time": "2018-07-04 22:17:46",
    "update_time": "2018-10-19 10:13:51"
  }
}
```

6.4.20 更新基础模型

路径：

POST /api/v2/asr/lm/bases

输入参数：

名称	位置	说明
base	Body	基础模型对象 (AsrLmBase)。可以设置如下参数： id：模型 ID； name：模型名称； description：模型描述信息，可选； seg_dict：分词词典的存储路径； eng_tn：英文 TN 文件的存储路径。

请求示例：

```
curl -X PUT \
  --header 'Content-Type: application/json' \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
```

```
--header 'X-NLS-Token: default' \
-d '{
  "id": "common",
  "name": "通用模型 (二)",
  "description": "第二个通用模型",
  "seg_dict": "slp/v2/asr/lm/inputs/common-seg-dict.txt",
  "eng_tn": "slp/v2/asr/lm/inputs/common-eng-tn.txt"
}' 'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/bases'
```

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff"
}
```

6.4.21 删除基础模型

路径：

```
DELETE /api/v2/asr/lm/bases/{base_id}
```

输入参数：

名称	位置	说明
base_id	Path	要删除的基础模型 ID。

请求示例：

```
curl -X DELETE \
--header 'Accept: application/json' \
--header 'X-NLS-User: default' \
--header 'X-NLS-Token: default' \
'http://nls-slp.aliyun.test:8701/api/v2/asr/lm/bases/common'
```

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff"
}
```

6.5 定制类热词

类热词 API 包括：

- 创建词表
- 列举词表
- 查询词表
- 更新词表
- 删除词表

6.5.1 创建词表

路径：

```
POST /api/v1/asr/class-vocabs
```

输入参数：

名称	位置	说明
vocab	Body	类热词词表对象 (AsrClassVocab)。可以设置如下参数： id：词表 ID，可选，如果为空则自动生成； name：词表名称； description：词表描述信息，可选； words：要添加的词，是一个字符串数组。

请求示例：

```
curl -X POST \  
  --header 'Content-Type: application/json' --header 'Accept: application/json' \  
  --header 'X-NLS-User: default' --header 'X-NLS-Token: default' \  
  -d '{  
    "id": "demo-vocab",  
    "name": "示例词表",  
    "description": "这是一个示例词表",  
    "words": ["苹果", "西瓜"]  
  }' 'http://nls-slp.aliyun.test:8701/api/v1/asr/class-vocabs'
```

输出参数：

名称	说明
class_vocab_id	词表 ID。

响应示例：

```
{  
  "request_id": "7130914d32a3441db06747523675d9ff",  
  "class_vocab_id": "demo-vocab"  
}
```

6.5.2 列举词表

路径：

```
GET /api/v1/asr/class-vocabs
```

输入参数：

名称	位置	说明
page_number	Query	页号，从 1 开始编号。可选，默认值是 1。
page_size	Query	页大小，范围在 1 到 100 之间。可选，默认是 10。

请求示例：

```
curl -X GET --header 'Accept: application/json' \  
  --header 'X-NLS-User: default' --header 'X-NLS-Token: default' \  
  'http://nls-slp.aliyun.test:8701/api/v1/asr/class-vocabs'
```

输出参数：

名称	说明
page	类热词词表对象 (AsrClassVocab) 的分页结果。

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff",
  "page": {
    "content": [
      {
        "id": "demo-vocab",
        "name": "示例词表",
        "description": "这是一个示例词表",
        "size": 4,
        "md5": "f1d3ff8443297732862df21dc4e57262",
        "create_time": "2018-11-01 20:02:56",
        "update_time": "2018-11-01 20:02:56"
      }
    ],
    "total_pages": 1,
    "total_items": 1,
    "page_number": 1,
    "page_size": 10
  }
}
```

6.5.3 查询词表

路径：

GET /api/v1/asr/class-vocabs/{vocab_id}

输入参数：

名称	位置	说明
vocab_id	Path	要查询的词表 ID。

请求示例：

```
curl -X GET --header 'Accept: application/json' \
  --header 'X-NLS-User: default' --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v1/asr/class-vocabs/demo-vocab'
```

输出参数：

名称	说明
class_vocab	类热词词表对象 (AsrClassVocab)。

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff",
  "class_vocab": {
    "id": "demo-vocab",
    "name": "示例词表 ( 二 )",
    "description": "这是另一个示例词表",
    "size": 4,
```



```
    "md5": "f1d3ff8443297732862df21dc4e57262",
    "words": [
      "苹果",
      "西瓜"
    ],
    "create_time": "2018-11-01 18:31:06",
    "update_time": "2018-11-01 18:31:06"
  }
}
```

6.5.4 更新词表

路径：

POST /api/v1/asr/class-vocabs

输入参数：

名称	位置	说明
vocab	Body	类热词词表对象 (AsrClassVocab)。可以设置如下参数： id：词表 ID，可选，如果为空则自动生成； name：词表名称； description：词表描述信息，可选； words：要添加的词，是一个字符串数组，全量替换原有的词。

请求示例：

```
curl -X PUT \
  --header 'Content-Type: application/json' --header 'Accept: application/json' \
  --header 'X-NLS-User: default' --header 'X-NLS-Token: default' -d '{
    "id": "demo-vocab",
    "name": "示例词表 (二)",
    "description": "这是另一个示例词表",
    "words": ["苹果", "西瓜"]
  }' 'http://nls-slp.aliyun.test:8701/api/v1/asr/class-vocabs'
```

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff"
}
```

6.5.5 删除词表

路径：

DELETE /api/v1/asr/class-vocabs/{vocab_id}

输入参数：

名称	位置	说明
vocab_id	Path	要删除的词表 ID。

请求示例：

```
curl -X DELETE --header 'Accept: application/json' \
```

```
--header 'X-NLS-User: default' --header 'X-NLS-Token: default' \
'http://nls-slp.aliyun.test:8701/api/v1/asr/class-vocabs/demo-vocab'
```

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff"
}
```

6.6 定制泛热词

泛热词 API 包括：

- 创建词表
- 列举词表
- 查询词表
- 更新词表
- 删除词表

6.6.1 创建词表

路径：

```
POST /api/v1/asr/vocabs
```

输入参数：

名称	位置	说明
vocab	Body	泛热词词表对象 (AsrVocab)。可以设置如下参数： id：词表 ID，可选，如果为空则自动生成； name：词表名称； description：词表描述信息，可选； word_weights：要添加的词，是一个词典，键为词，值为权重。

请求示例：

```
curl -X POST \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'X-NLS-User: default' \
--header 'X-NLS-Token: default' \
-d '{
  "id": "demo-vocab",
  "name": "示例词表",
  "description": "这是一个示例词表",
  "word_weights": {"苹果": 2, "西瓜": 3}
}' 'http://nls-slp.aliyun.test:8701/api/v1/asr/vocabs'
```

输出参数：

名称	说明
----	----

vocab_id	词表 ID。
----------	--------

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff",
  "vocab_id": "demo-vocab"
}
```

6.6.2 列举词表

路径：

```
GET /api/v1/asr/vocabs
```

输入参数：

名称	位置	说明
page_number	Query	页号，从 1 开始编号。可选，默认值是 1。
page_size	Query	页大小，范围在 1 到 100 之间。可选，默认是 10。

请求示例：

```
curl -X GET \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
  --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v1/asr/vocabs'
```

输出参数：

名称	说明
page	泛热词词表对象 (AsrVocab) 的分页结果。

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff",
  "page": {
    "content": [
      {
        "id": "demo-vocab",
        "name": "示例词表",
        "description": "这是一个示例词表",
        "size": 4,
        "md5": "f1d3ff8443297732862df21dc4e57262",
        "create_time": "2018-11-01 20:11:42",
        "update_time": "2018-11-01 20:11:42"
      }
    ],
    "total_pages": 1,
    "total_items": 1,
    "page_number": 1,
    "page_size": 10
  }
}
```

6.6.3 查询词表

路径：

```
GET /api/v1/asr/vocabs/{vocab_id}
```

输入参数：

名称	位置	说明
vocab_id	Path	要查询的词表 ID。

请求示例：

```
curl -X GET \
  --header 'Accept: application/json' \
  --header 'X-NLS-User: default' \
  --header 'X-NLS-Token: default' \
  'http://nls-slp.aliyun.test:8701/api/v1/asr/vocabs/demo-vocab'
```

输出参数：

名称	说明
vocab	泛热词词表对象 (AsrVocab)。

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff",
  "vocab": {
    "id": "demo-vocab",
    "name": "示例词表",
    "description": "这是一个示例词表",
    "size": 4,
    "md5": "f1d3ff8443297732862df21dc4e57262",
    "create_time": "2018-11-01 20:11:42",
    "update_time": "2018-11-01 20:11:42",
    "word_weights": {
      "苹果": 2,
      "西瓜": 3
    }
  }
}
```

6.6.4 更新词表

路径：

```
POST /api/v1/asr/vocabs
```

输入参数：

名称	位置	说明
vocab	Body	类热词词表对象 (AsrClassVocab)。可以设置如下参数： id：词表 ID，可选，如果为空则自动生成； name：词表名称；

		description：词表描述信息，可选； word_weights：要添加的词，是一个词典，键为词，值为权重。全量替换原有的词。
--	--	---

请求示例：

```
curl -X PUT \  
  --header 'Content-Type: application/json' \  
  --header 'Accept: application/json' \  
  --header 'X-NLS-User: default' \  
  --header 'X-NLS-Token: default' \  
  -d '{  
    "id": "demo-vocab",  
    "name": "示例词表 (二)",  
    "description": "这是另一个示例词表",  
    "word_weights": {"苹果": 2, "西瓜": 3}  
  }' 'http://nls-slp.aliyun.test:8701/api/v1/asr/vocabs'
```

响应示例：

```
{  
  "request_id": "7130914d32a3441db06747523675d9ff"  
}
```

6.6.5 删除词表

路径：

DELETE /api/v1/asr/vocabs/{vocab_id}

输入参数：

名称	位置	说明
vocab_id	Path	要删除的词表 ID。

请求示例：

```
curl -X DELETE \  
  --header 'Accept: application/json' \  
  --header 'X-NLS-User: default' \  
  --header 'X-NLS-Token: default' \  
  'http://nls-slp.aliyun.test:8701/api/v1/asr/vocabs/demo-vocab'
```

响应示例：

```
{  
  "request_id": "7130914d32a3441db06747523675d9ff"  
}
```

6.7 存储管理

6.7.1 上传文件

路径：

POST /api/files/upload

输入参数：

名称	位置	说明
path	Query	文件在存储系统中的路径。
file	Form	需要上传的文件。

请求示例：

```
curl -X POST \  
  --header 'Content-Type: multipart/form-data' \  
  --header 'Accept: application/json' \  
  --header 'X-NLS-Token: default' \  
  -F "path=slp/tmp/demo.txt" \  
  -F "file=@/home/admin/demo.txt" \  
  'http://nls-slp.aliyun.test:8701/api/files/upload'
```

输出参数：

名称	说明
path	文件在存储系统中的路径。

响应示例：

```
{  
  "request_id": "7130914d32a3441db06747523675d9ff",  
  "path": "slp/tmp/demo.txt "  
}
```

6.7.2 列举文件

路径：

```
GET /api/files/list
```

输入参数：

名称	位置	说明
prefix	Query	文件在存储系统中的路径。
max_count	Query	列出的最大文件数量。

请求示例：

```
curl -X GET -G \  
  --header 'Accept: application/json' \  
  --header 'X-NLS-Token: default' \  
  -d "prefix=slp/tmp/" \  
  -d "max_count=10" \  
  'http://nls-slp.aliyun.test:8701/api/files/list'
```

输出参数：

名称	说明
files	文件在存储系统中的路径。

响应示例：

```
{
  "request_id": "7130914d32a3441db06747523675d9ff",
  "files": [
    {
      "path": "slp/tmp/demo.txt",
      "size": 1282399,
      "md5": "2cabfca123a8c1b6ff2bf2723a61c00c",
      "update_time": 1541136830733
    }
  ]
}
```

6.7.3 下载文件

路径：

GET /api/files/download

输入参数：

名称	位置	说明
path	Query	文件在存储系统中的路径。

请求示例：

```
curl -X GET -G \
  --header 'Accept: application/octet-stream' \
  --header 'X-NLS-Token: default' \
  -d "path=slp/tmp/demo.txt" \
  -o demo1.txt \
  'http://nls-slp.aliyun.test:8701/api/files/download'
```

输出参数：

名称	说明
body	返回的 HTTP 消息体就是文件内容。可以通过 curl 命令的-o 参数保存到文件。

响应示例：

demo1.txt 文件

6.7.4 删除文件

路径：

POST /api/files/delete

输入参数：

名称	位置	说明
path	Query	文件在存储系统中的路径。

请求示例：

```
curl -X POST -G \
  --header 'Accept: application/json' \
```

```
--header 'X-NLS-Token: default' \  
-d "path=slp/tmp/demo.txt" \  
'http://nls-slp.aliyun.test:8701/api/files/delete'
```

响应示例：



```
{  
  "request_id": "7130914d32a3441db06747523675d9ff"  
}
```

6.8 Swagger 文档

自学习平台提供了 swagger 形式的 API 文档，可以通过浏览器访问如下地址查看：

<http://nls-slp.aliyun.test:8701/swagger-ui.html>

下图是 swagger 文档的页面：

01-定制语言模型 (/v2/api-docs?group=01-定制语言模型)

自学习平台-定制语言模型

定制语言模型-1-数据 : Asr Lm Data Api Controller

Show/Hide | List Operations | Expand Operations

GET	/api/v2/asr/lm/data	列举数据集
POST	/api/v2/asr/lm/data	创建数据集
PUT	/api/v2/asr/lm/data	更新数据集
DELETE	/api/v2/asr/lm/data/{data_id}	删除数据集
GET	/api/v2/asr/lm/data/{data_id}	查询数据集

定制语言模型-2-模型 : Asr Lm Model Api Controller

Show/Hide | List Operations | Expand Operations

GET	/api/v2/asr/lm/models	列举模型
POST	/api/v2/asr/lm/models	创建模型
PUT	/api/v2/asr/lm/models	更新模型
DELETE	/api/v2/asr/lm/models/{model_id}	删除模型
GET	/api/v2/asr/lm/models/{model_id}	查询模型

定制语言模型-3-操作 : Asr Lm Operation Api Controller

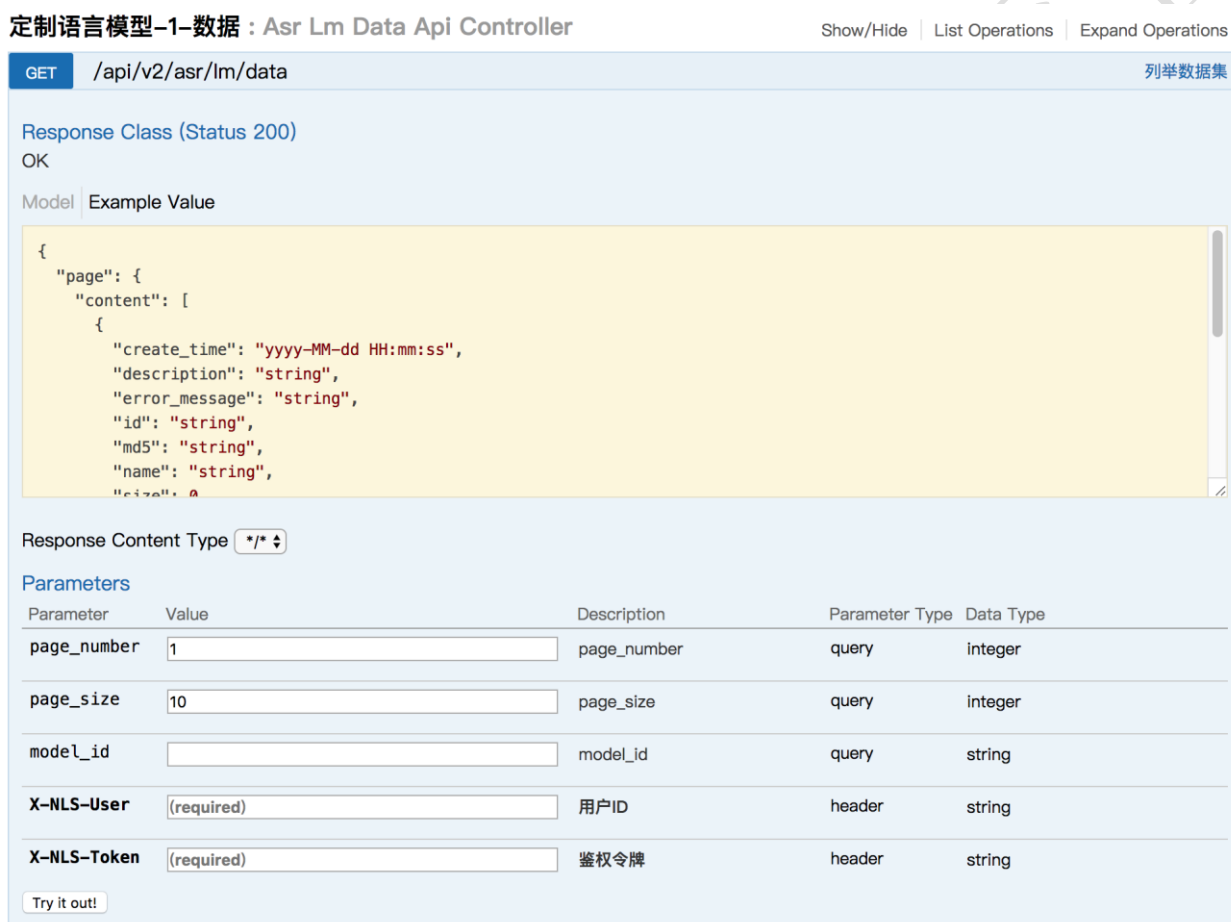
Show/Hide | List Operations | Expand Operations

POST	/api/v2/asr/lm/operations/add-data-to-model	添加数据到模型
POST	/api/v2/asr/lm/operations/offline-model	下线模型
POST	/api/v2/asr/lm/operations/online-model	上线模型
POST	/api/v2/asr/lm/operations/remove-data-from-model	从模型删除数据
POST	/api/v2/asr/lm/operations/train-model	开始训练

默认显示的定制语言模型的 API 文档，可以通过上方导航栏的下拉列表查看其它类别的 API：



可以点击 API 来查看具体的参数：



定制语言模型-1-数据 : Asr Lm Data Api Controller

GET /api/v2/asr/lm/data 列举数据集

Response Class (Status 200)
OK

Model Example Value

```
{
  "page": {
    "content": [
      {
        "create_time": "yyyy-MM-dd HH:mm:ss",
        "description": "string",
        "error_message": "string",
        "id": "string",
        "md5": "string",
        "name": "string",
        "size": 0
      }
    ]
  }
}
```

Response Content Type */*

Parameters

Parameter	Value	Description	Parameter Type	Data Type
page_number	1	page_number	query	integer
page_size	10	page_size	query	integer
model_id		model_id	query	string
X-NLS-User	(required)	用户ID	header	string
X-NLS-Token	(required)	鉴权令牌	header	string

Try it out!

可以在 swagger 页面直接写入参数发起 API 调用，swagger 会自动生成对应的 curl 命令：

Response Content Type

Parameters

Parameter	Value	Description	Parameter Type	Data Type
page_number	<input type="text" value="1"/>	page_number	query	integer
page_size	<input type="text" value="10"/>	page_size	query	integer
model_id	<input type="text"/>	model_id	query	string
X-NLS-User	<input type="text" value="default"/>	用户ID	header	string
X-NLS-Token	<input type="text" value="default"/>	鉴权令牌	header	string

Try it out! [Hide Response](#)

Curl

```
curl -X GET --header 'Accept: application/json' --header 'X-NLS-User: default' --header 'X-NLS-Token: default' 'http://
```

执行的结果如下：

Response Body

```
{
  "request_id": "8fbceb745a6f49f69b2801f7712ce00a",
  "page": {
    "content": [
      {
        "id": "1b64bee9994749f2a67eadac6379f80c",
        "name": "示例数据集",
        "description": "这是一个示例数据集",
        "size": 7777404,
        "md5": "39326cf690e38473535a385ec1e7a00",
        "url": "slp/v2/asr/lm/inputs/common-seg-dict.txt",
        "status": "Ready",
        "create_time": "2018-10-31 17:20:39",
        "update_time": "2018-10-31 17:20:39"
      }
    ],
    "total_pages": 1,
    "total_items": 1,
    "page_number": 1,
    "page_size": 10
  }
}
```

7 控制台

自学习平台提供了网页形式的控制台，可以通过浏览器访问如下地址查看：

<http://nls-slp.aliyun.test:8701>

下图为控制台的欢迎页面，在这个页面可以看到一些基本的统计数据：



7.1 文件管理

查看文件列表

<http://nls-slp.aliyun.test:8701/files>

文件列表

根据前缀进行搜索：

最大文件数：

搜索

上传

路径	大小	MD5	修改时间	
slp/demo/lm/demo-data-text.txt	5962	9e4043b8d1e24ebd966e9b157a2c4	2019-01-01 08:07:29	下载 删除

上传文件

<http://nls-slp.aliyun.test:8701/files/upload>

上传数据文件

注意：每天会对前缀为slp/tmp/的文件进行清理

文件路径：

slp/tmp/9cde5219a7c04fa487b69e01a40d6fae

选择文件：

选择文件 未选择任何文件

上传

阿里巴巴 智能语言

8 常见问题

8.1 排查定制语言模型是否生效

确认 ASR 服务是否开启了定制模型功能

- 打开 ASR 服务的配置文件`/home/admin/speech-alisr/conf/processor.json`
- 确认 decoder/personalized_grammars 配置项下存在 InterpolationGrammarLoader 相关配置
- 确认 InterpolationGrammarLoader 对应的 enable 设置为 true
- 确认 InterpolationGrammarLoader 对应的 LocalGrammarFetcher 的 path 设置为`./models/customlms`

确认模型是否下载成功

- 进入 ASR 服务器上的目录`/home/admin/speech-alisr/models/customlms`
- 此目录中是否存在以基础模型命名的目录，目录中是否存在以定制模型 ID 命名的文件

确认识别请求的参数是否正确传递了模型 ID

- 进入 ASR 服务器上的目录`/home/admin/speech-alisr/logs`
- 查看 speech-alisr-trace.log 中记录的请求参数，customization_id 是否是正确的模型 ID

确认服务是否加载定制模型成功

- 进入 ASR 服务器上的目录`/home/admin/speech-alisr/logs`
- 查看 alisr.log 中是否包含加载模型相关的错误信息
- 如果加载定制模型到解码器失败会有错误日志`Failed to load grammar to recognizer` 或者 `Failed to load lm interpolation rescore grammar`
- 如果读取或解析定制模型失败会有错误日志`Failed to load rescore grammar`
- 找不到定制模型会有错误日志`Failed to fetch grammar from local cache`

8.2 排查定制声学和语言模型错误

8.2.1 排查的一般原则

1. 首先看 api 或者管控台返回的错误是否有足够信息判断错误原因。
2. 如果没有，获取 ap 返回的具体算法运行 log 文件，看是否能够定位到错误原因。
3. 如果不清楚，请把该文件发给支持同学
4. 针对数据处理和数据解析，还可以把 api 返回的 debug zip 文件发给支持同学

8.2.2 常见错误排查

模型训练(TrainAM)失败，报下面这个错误：

```
message=INTERNAL_ERROR|The ce training is failed.
```

这个错误首先按照排查一般原则，进入到第 4 步所指的目录。然后找到该目录下的 ce_nnet/log/log 文件(smbr 则是 smbr_nnet/log/log 文件)。

- No CUDA GPU detected!, diagnostics: cudaError_t 30 : "unknown error", in cu-device.cc : 一般是因为 GPU 驱动安装不正确，请参考[错误!未找到引用源](#)。章节。
- init failed apes: connect server failed: 授权 IP 地址无法连接，参考 apes 鉴权服务器的部署文档
- 其他错误，请联系支持同学

数据处理(DataProcessFilter)失败，报下面的错误：

The number of newpost.*.scp is not equal to 20

一般情况下，这个是因为数据太少导致，一批数据至少需要有 500 句左右。

8.3 排查定制泛热词不生效

确认 ASR 服务是否开启了加载定制泛热词

- 打开 ASR 服务的配置文件/home/admin/speech-alisr/conf/processor.json
- 确认 decoder/personalized_grammars 配置项下存在 InterpolationGrammarLoader 相关配置
- 确认 BiasGrammarLoader 对应的 enable 设置为 true
- 确认 BiasGrammarLoader 对应的 RedisGrammarFetcher 中的 reids 服务相关配置是正确的

确认识别请求的参数是否正确传递了泛热词 ID

- 进入 ASR 服务器上的目录/home/admin/speech-alisr/logs
- 查看 speech-alisr-trace.log 中记录的请求参数，vocabulary_id 是否是正确的泛热词 ID

确认服务是否加载泛热词成功

- 进入 ASR 服务器上的目录/home/admin/speech-alisr/logs
- 查看 alisr.log 中是否包含加载泛热词相关的错误信息
- 取不到泛热词会有错误日志 Failed to fetch grammar from redis: key=xxxx
- 泛热词加载或加载到解码器失败，会有错误日志 Failed to load rescore grammar: type= AlsBiasGrammar, status=xxx 或者 Failed to load rescore grammar to recognizer: type= AlsBiasGrammar, status=xxx

8.4 定制模型(非工程类)常见问题

8.4.1 定制语言模型

语言模型训练和声学模型训练哪个先进行？

语言模型训练速度快（分钟级）并且训练文本准备简单，所以优先考虑语言模型训练。然后针对有口音的业务场景，进行声学模型训练。

训练文本里面如果有许多特殊符号，是否需要做转换？

请按照实际读音进行转换。这个也可以参考下单独提供的标注规范。

语言模型训练一般需要多长时间？

根据训练语料的大小，训练时间在几分钟到几十分钟的范围。

语言模型训练文本，里面很多专业词汇，这个是不是会影响到的分词的准确性，进而影响模型的效果？

影响不大。如果分词词典中没有那些专业词，分词会按照字进行分词，训练的时候也会按照字来进行概率统计语言模型训练，最后识别的时候也可以按照字识别出来。

测试语言模型的时候，发现测试前后字错误率没有区别，这是为什么？

很可能是语言模型没有生效，请联系工程对接同学进行咨询。

如何对一些需要大量展开的关键词组合进行加强训练？例如 1G 的共享套餐，2G 的个人套餐，...？

如果可以简单的把所有组合都展开，可以认为展开加入训练语料中。如果比较麻烦，也可以只分别对"xG"(展开，即 1G, 2G,...)和"xx 套餐"(展开)进行加强。

8.4.2 定制声学模型

如果两个业务口音近似，可以用一个业务训练出来的声学模型给第二个业务使用吗？

不是非常建议。因为除了口音影响声学相关的识别性能，本身通讯的信道和背景噪音这些声学因素在不同业务中也会有所区别。建议在有能力的前提下，还是针对不同业务，重新训练模型。

声学模型训练一般需要多少时间？

训练时间取决于数据量和本身硬件资源情况。参考附录[错误!未找到引用源。](#)。

标注的频率如何把握，是一次标注很多数据一起训练，还是拆分成多批进行标注？

建议在项目的初期，采用多批标注的方式。确保整个标注到训练流程走通，并且可以及时了解数据的质量和训练的效果。当项目区域稳定，后续的例行标注，可以采用更长的周期进行，甚至可以在标注之前去掉一些常见的已经识别很好的句子。

可以用来训练方言吗？

目前不行。我们提供的基础模型是普通话模型，只能针对普通话（或者带有口音）的业务进行调优训练。

训练数据处理的步骤中有一个数据筛选的过程，为什么被筛掉这么多数据？

数据处理的步骤会在生成训练特征和标签的同时，对数据进行一个筛选。被筛选掉的音频多数是因为生成特征和标签失败的。这里面有几种主要原因：a.标注本身不太准确；b.音频质量太差，背景有大量噪音特别是人声噪音；c.说话人没说清楚或者说方言等等。如果是生成特征和标签失败，本身就没有办法被用来进行训练了。

9 附录

9.1 mongofiles 命令使用说明

专有云安装包的 resource 目录中会提供 mongofiles 的可执行文件，也可以从 mongodb 官方网站下载安装包，解压后即可使用。（https://www.mongodb.com/dr/fastdl.mongodb.org/osx/mongodb-osx-ssl-x86_64-3.6.8.tgz/download）

添加环境变量

```
export PATH=$PATH:/home/admin/service/bin
```

设置服务地址

```
mongo_uri='mongodb://nls-  
cloud:88630dd3d489a617b635d51ef7eedfe@127.0.0.1:7031/?authSource=admin'
```

上传文件

```
# 将磁盘中的文件/home/admin/demo.txt 上传到 mongodb 并设置路径为 slp/tmp/1'1.txt  
mongofiles --uri ${mongo_uri} -d nls-cloud put slp/tmp/demo.txt -l /home/admin/demo.txt
```

列举文件

```
# 列举 mongodb 中的文件  
mongofiles --uri ${mongo_uri} -d nls-cloud list
```

下载文件

```
# 将 mongodb 中的文件 slp/tmp/demo.txt 下载到本地磁盘/home/admin/demo.txt  
mongofiles --uri ${mongo_uri} -d nls-cloud get slp/tmp/demo.txt -l /home/admin/demo1.txt
```

删除文件

```
# 删除 mongodb 中的文件 slp/tmp/demo.txt  
mongofiles --uri ${mongo_uri} -d nls-cloud delete slp/tmp/demo.txt
```