

重要说明

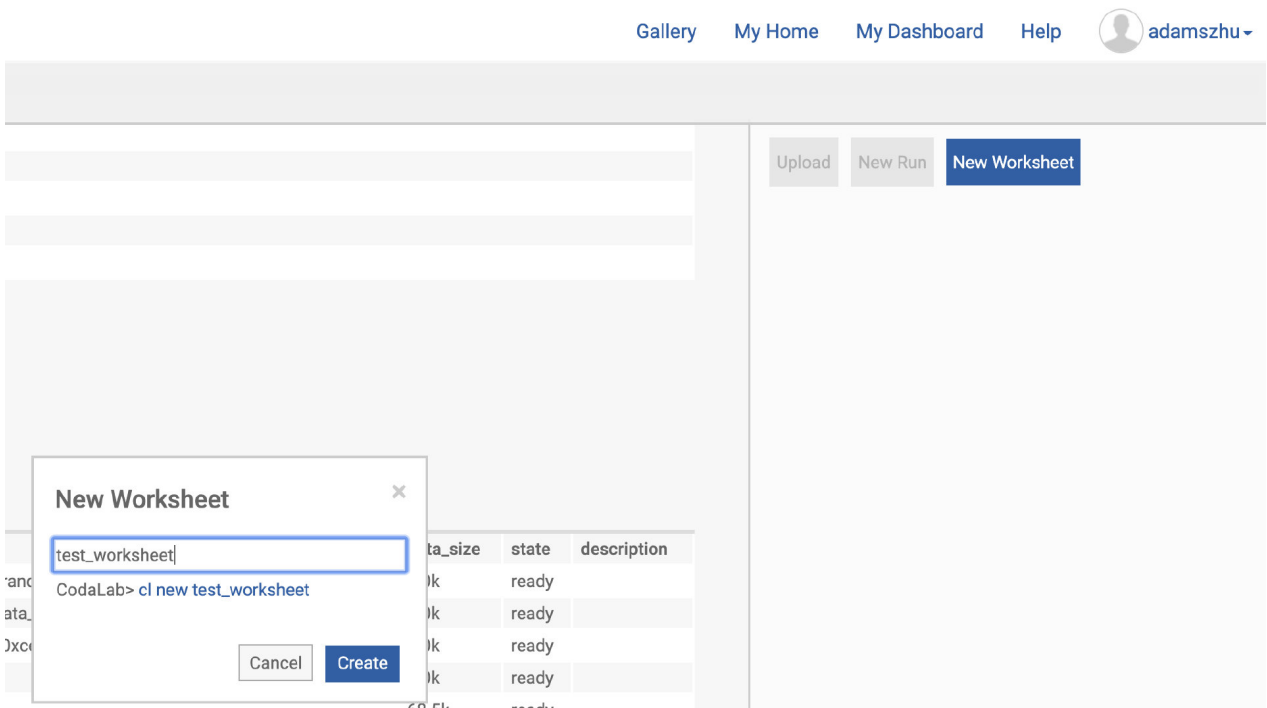
1. 本次评测是SMP2019-ECDT，任务1的评测，所有参赛系统均需要使用[CodaLab在线系统](#)提交，不接受线下调试请求。
2. Codalab系统是一个很不错的代码、数据资源和论文实验分享的利器，需要一定的cli命令基础和一些的docker知识，如果是初次接触这样的系统，请耐心阅读官方的[Tutorial](#)，了解一些基本的概念，如worksheet, bundle等。
3. Codalab系统的一个难点是运行环境的重现，如果参赛队伍使用了某单一深度学习库，如Theano/Tensorflow/PyTorch/Keras等，则可以很方便地使用[docker官方仓库](#)中已有的docker镜像，如果需要使用多种深度学习库，或者在[docker官方库](#)中未找到搭配好的镜像，则需要自己构建docker镜像并上传至 <http://hub.docker.com/>。
4. 请不要在Codalab系统上训练模型，并保证提交的是训练好的模型及相应的解码程序。

模拟模型的提交流程

为了减少提交的系统不能在测试集上运行的可能性，我们首先在模拟模型与数据上进行提交测试，以下是提交流程。

Step 1. 注册并初始化Codalab账号

请根据实际情况填写需要的信息进行[注册](#)，注册完成后，点击右上角的My Dashboard-->New Worksheet-->填写新建worksheet的名字。



接下来要给新建立的worksheet设置权限才能使用，此步不可跳过，否则可能导致源码和模型泄露。

控制台如下图。



CodaLab>

<none>

设置smp2019ecdt_group用户组对该worksheet可见，权限为read：

```
Codalab> cl wperm . smp2019ecdt_group read
```

设置public用户组对该worksheet不可见以防止代码和数据泄露：

```
Codalab> cl wperm . public none
```

此时worksheet右边的状态应为：

Upload

New Run

New Worksheet

uuid	0xf87dd8d7d2854e3e9aa7189b4b16d528
name	zchai_ecdt2018_test
title	<none>
owner	adamszhu
permissions	you(all) smp2019ecdt_group(read)

Step 2. 模拟模型与数据提交测试

在提交自己的代码前，我们先用模拟的模型和数据熟悉整个流程。

首先从SMP2019ECDT_TASK1的[worksheet](#)中添加相关的模型和数据，需要注意点击下图Evaluation programs中的每一项都可以在右边的信息栏获得到唯一的一个uuid，根据此uuid即可完成相关添加操作。

SMP2019 ECDT Task1: Natural Language Understanding

Keyboard Shortcuts

ViewEdit source

Overview

This evaluation is released by the Evaluation of Chinese Human-Computer Dialogue Technology(SMP2019-ECDT) Task1 and the dataset is provided by the iFLYTEK Corporation.

Please click [here](#) to learn more about Task1 of the evaluation.

Evaluation programs

uuid[0:8]	name	summary	data_size	state	description
0x6f4a38	random_output.py	[uploaded]	1.0k	ready	
0xff99f9	extract_features.py	[uploaded]	1.1k	ready	
0xe54028	evaluation.py	[uploaded]	1.8k	ready	
0x9ba8a2	sample.json	[uploaded]	3.3k	ready	

UploadNew RunNew Worksheet

uuid0x6f4a387664b45b4ac860d22496d3b9b

namerandom_output.py

description<none>

ownersmp2019ecd

is_anonymousfalse

permissionsyou(all) public(read)

created2019-05-09 16:08:31

data_size1k

stateready

download

contents<div>#!/usr/bin/env python</div><div># -*- coding: utf-8 -*-</div><div># @Time : 8/5/19 4:12 PM</div><div># @Author : zchai</div><div>
</div><div># -*- coding: utf-8 -*-</div><div>import sys</div><div>
</div><div>import random</div></div>

复制相应的文件的uuid后在自己的worksheet控制台运行如下命令，即可添加对应的文件到自己的worksheet。

```
CodaLab> cl add bundle 0x6f4a38f7664b45b4ac860d22496d3b9b
```

<none>

Keyboard Shortcuts

ViewEdit source

uuid[0:8]	name	summary	data_size	state	description
0x6f4a38	random_output.py	[uploaded]	1.0k	ready	

UploadNew RunNew Worksheet

uuid0x3c9a8bdfce243d28c1d3845140576c

nametask1_ttttt

title<none>

owneradamszhu

permissionsyou(all) smp2019ecd_group(read)

通过这个方法，将SMP2019ECDT_TASK2的worksheet中的所有文件添加到自己的worksheet中。添加完成后自己的worksheet如下图所示：

<none>

Keyboard Shortcuts

ViewEdit source

uuid[0:8]	name	summary	data_size	state	description
0x6f4a38	random_output.py	[uploaded]	1.0k	ready	
0xff99f9	extract_features.py	[uploaded]	1.1k	ready	
0xe54028	evaluation.py	[uploaded]	1.8k	ready	
0x9ba8a2	sample.json	[uploaded]	3.3k	ready	

运行模拟模型程序

此时可以使用 `random_output.py` , `extract_features.py` , `evaluation.py` 和 `sample.json` 构建一个bundle用来模拟提交代码后解码模型并生成预测文件的过程。点击右边的New Run来新建一个bundle，勾选需要用到数据和脚本，在下图红框所示的命令行里输入相应命令，点击Run即可执行构建的bundle，也可以复制下面自动生成的指令粘贴到在codalab控制台执行，这样可以修改或添加一些参数（不能直接在命令输入框里添加执行需要的参数），可根据需要选择执行方式。

Create Run Bundle

Step 1: Select any dependencies (either bundles or the files/directories inside).

- ☒ random_output.py(0x6f4a38)
- ☒ extract_features.py(0xff99f9)
- ☐ evaluation.py(0xe54028) STEP1:勾选需要的程序
- ☒ sample.json(0x9ba8a2)

Step 2: Enter a shell command, which will be run in a directory with the dependencies (rename if desired).

```
$ ls
random_output.py →
random_output.py(0x6f4a38)
extract_features.py →
extract_features.py(0xff99f9)
sample.json → sample.json(0x9ba8a2)
python3 random_output.py sample.json pred.json
```

STEP2:输入pyhton命令

STEP3:点击Run

```
CodaLab> run 'random_output.py:0x6f4a38' 'extract_features.py:0xff99f9'
'sample.json:0x9ba8a2' 'python3 random_output.py sample.json pred.json'
```

Cancel

Run

对应的python命令

```
python3 random_output.py sample.json pred.json
```

uuid[0:8]	name	summary	data_size	state	description
0x6f4a38	random_output.py	[uploaded]	1.0k	ready	
0xff99f9	extract_features.py	[uploaded]	1.1k	ready	
0xe54028	evaluation.py	[uploaded]	1.8k	ready	
0x9ba8a2	sample.json	[uploaded]	3.3k	ready	
0x9bd181	run-python3	! python3 random_output.py(0x6f sample.json(0x9b) pred.json	6.5k	ready	

产生了一个新的运行bundle

Head and tail not supported for directory blobs.

uuid 0x9bd181ebad1248da9d6e32f3f3d906d6

name run-python3

description <none>

owner adamszhu

is_anonymous false

permissions you(all) smp2019ecdt_group(read)

created 2019-05-10 15:56:27

data_size 6.5k

command python3 random_output.py sample.json pred.json

time 14.9s

state **ready**

download ready表示运行成功

dependencies

random_output.py → random_output.py(0x6f4a38)

extract_features.py → extract_features.py(0xff99f9)

sample.json → sample.json(0x9ba8a2)

contents

/ 得到的输出文件

pred.json 2.5k

stderr 0

stdout 0

接下来，我们把得到的pred.json文件导出。

```
CodaLab> cl make run-python3/pred.json -n random_out.json
```

这样我们就已经得到了预测程序的输出内容，接下来运行评测脚本得出最终得分即可。使用官方提供的 `eval.py` 对系统的输出结果进行评测。

进行评测

Create Run Bundle



Step 1: Select any dependencies (either bundles or the files/directories inside).

<input type="checkbox"/>	random_output.py(0x6f4a38)
<input type="checkbox"/>	extract_features.py(0xff99f9)
<input checked="" type="checkbox"/>	evaluation.py(0xe54028)
<input checked="" type="checkbox"/>	sample.json(0x9ba8a2)
<input type="checkbox"/>	run-python3(0x9bd181)
<input checked="" type="checkbox"/>	random_out.json(0xc84c42)

Step 2: Enter a shell command, which will be run in a directory with the dependencies (rename if desired).

```
$ ls
random_out.json → random_out.json(0xc84c42)
sample.json → sample.json(0x9ba8a2)
evaluation.py → evaluation.py(0xe54028)
python3 evaluation.py sample.json random_out.json
```

CodaLab> run 'random_out.json:0xc84c42' 'sample.json:0x9ba8a2'
'evaluation.py:0xe54028' 'python3 evaluation.py sample.json random_out.json'

Cancel

Run

需要填写的python命令

```
python3 evaluation.py sample.json random_out.json
```

得到的结果如下，至此，我们完成了在模拟模型和模拟数据上的评测。

uuid[0:8]	name	summary	data_size	state	description
0x6f4a38	random_output.py	[uploaded]	1.0k	ready	
0xff99f9	extract_features.py	[uploaded]	1.1k	ready	
0xe54028	evaluation.py	[uploaded]	1.8k	ready	
0x9ba8a2	sample.json	[uploaded]	3.3k	ready	
0x9bd181	run-python3	! python3 random_output.py(0x6f) sample.json(0x9b) pred.json	6.5k	ready	
0xc84c42	random_out.json	= run-python3(0x9b)	2.5k	ready	
0xdad468	run-python3	! python3 evaluation.py(0xe5) sample.json(0x9b) random_out.json(0xc8)	4.0k	ready	

uuid 0xdad468f4d7644434b337e1af382853b

name run-python3

description <none>

owner adamszhu

is_anonymous false

permissions you(!!!) smp2019ecdt_group(read)

created 2019-05-10 16:10:55

data_size 4k

command python3 evaluation.py sample.json random_out.json

time 16s

state **ready**

download ready表示成功运行

dependencies

random_out.json → random_out.json(0xc84c42)

sample.json → sample.json(0x9ba8a2)

evaluation.py → evaluation.py(0xe54028)

stdout 输出结果: 随机结果0.00也正常

Avg sentence accuracy : 0.000000

Step 3. 提交模型与申请测试集评测

在Step 1/2中，我们熟悉了整个codalab的使用流程，下面需要运行参赛队伍自己的系统进行模拟数据上的预测、评分和结果提交。基本流程与Step 2一致，只需将 `random_output.py` 文件更换为自己的模型文件，模型输出的相关要求请阅读 `random_output.py` 中的注释。但是如果使用了外部依赖库会相对复杂一些。

总的来说，如果直接运行自己的程序发现缺少依赖，那么请在在<http://hub.docker.com/>上找到自己需要的docker镜像并能在本地成功运行，就可以在运行解码程序的命令最后加上 `--request-docker-image xxx/xx` 参数(如不了解如何使用—request-docker-image命令，可以参考Task2, Bleu评测部分)进行docker镜像获取与环境自动配置，如果没有找到相应镜像或拉取后调试仍不成功，则需要自己构建所需环境的docker镜像并上传至<http://hub.docker.com/>，然后再进行拉取调试。

如果调试过程中出现了运行失败的情况，则可以根据stderr输出信息或直接下载stderr文件，根据错误信息进行相应修改。

测试集评测申请

为了保证比赛的公平性，真实的评测测试集仍不会公开，最终的测试方式为参赛队伍提交Step3中的bundle，由官方对提交的bundle进行评分并更新leaderboard，在此期间应保证所提交bundle的完整性，不能删除相关的数据资源或代码文件。

成功完成Step3后，就可以提交相应的评分bundle到smp2018_ecdt官方的联系邮箱：smp2019ecdt@163.com，需要提交的内容有：机构名、参加的任务，系统名(英文)、联系邮箱、负责人姓名、**worksheet**所有文件的完整**uuid**以及对应的**name**。提交后我们会在隐藏测试集对收集到的最新模型进行评分并更新Leaderboard。

其他说明

1. 常用docker镜像及请求参数

- **Theano + Keras**: `--request-docker-image tboquet/kenacuthe7hc5`
- **Tensorflow**: `--request-docker-image tensorflow/tensorflow`
- **CNTK**: `--request-docker-image microsoft/cntk`
- **PyTorch**: `--request-docker-image pytorch/pytorch`
- **LTP**: `--request-docker-image scorpio147wbh/ltp`
- **Anaconda**: `--request-docker-image continuumio/anaconda`
- **Tensorlayer**: `--request-docker-image gmueller/tensorlayer`

2. 使用GPU资源解码

Codalab上提供了M60 GPU的支持，可以大大提高解码速度，需在解码命令中加入 `--request-gpus 1`。

* 注意GPU可以用来解码，不要使用GPU训练模型。