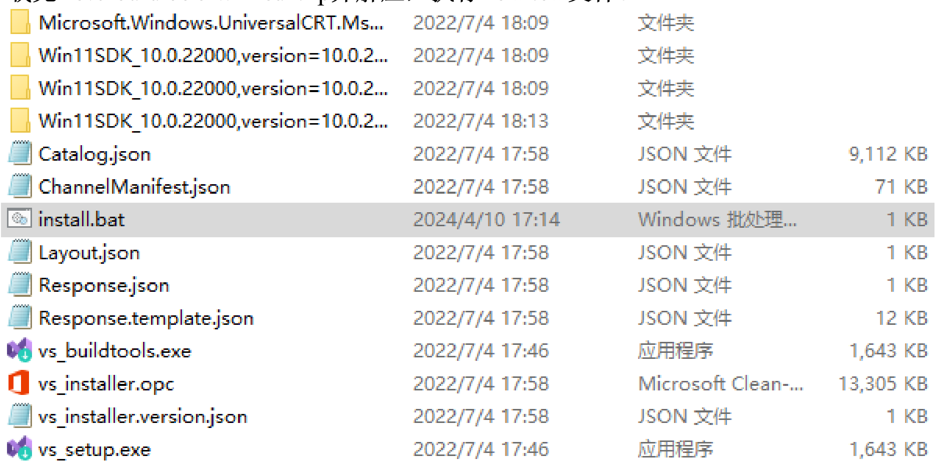
1. 先决条件

1.1 Rust

需要安装Rust才能将HuggingFace分词器交叉编译为Android。要开始使用Rust，先安装Visual Studio C++ Build tools。下面基于windows环境安装。

1.1.1 安装Visual Studio C++ Build tools

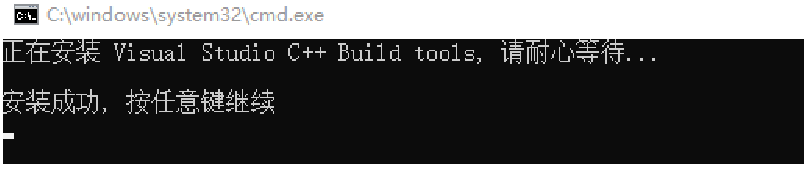
下载完msvc-buildtools-with-sdk.zip并解压，执行install.bat文件。

[


Microsoft.Windows.UniversalCRT.Ms...	2022/7/4 18:09	文件夹	
Win11SDK_10.0.22000,version=10.0.2...	2022/7/4 18:09	文件夹	
Win11SDK_10.0.22000,version=10.0.2...	2022/7/4 18:09	文件夹	
Win11SDK_10.0.22000,version=10.0.2...	2022/7/4 18:13	文件夹	
Catalog.json	2022/7/4 17:58	JSON 文件	9,112 KB
ChannelManifest.json	2022/7/4 17:58	JSON 文件	71 KB
install.bat	2024/4/10 17:14	Windows 批处理...	1 KB
Layout.json	2022/7/4 17:58	JSON 文件	1 KB
Response.json	2022/7/4 17:58	JSON 文件	1 KB
Response.template.json	2022/7/4 17:58	JSON 文件	12 KB
vs_buildtools.exe	2022/7/4 17:46	应用程序	1,643 KB
vs_installer.opc	2022/7/4 17:58	Microsoft Clean-...	13,305 KB
vs_installer.version.json	2022/7/4 17:58	JSON 文件	1 KB
vs_setup.exe	2022/7/4 17:46	应用程序	1,643 KB

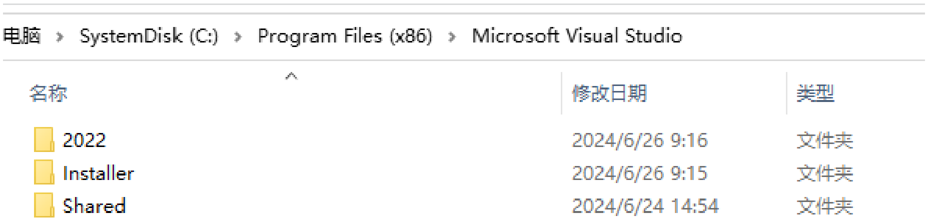
]

安装成功后。



C:\windows\system32\cmd.exe

```
正在安装 Visual Studio C++ Build tools，请耐心等待...

安装成功，按任意键继续
```

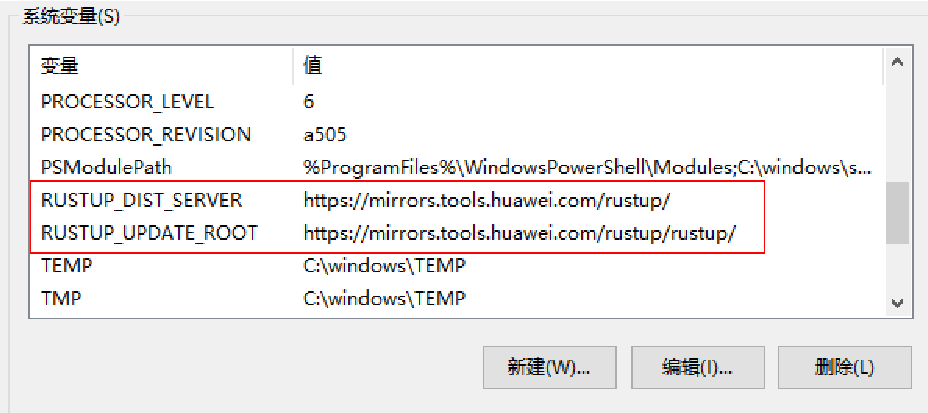


电脑 › SystemDisk (C:) › Program Files (x86) › Microsoft Visual Studio

名称	修改日期	类型
2022	2024/6/26 9:16	文件夹
Installer	2024/6/26 9:15	文件夹
Shared	2024/6/24 14:54	文件夹

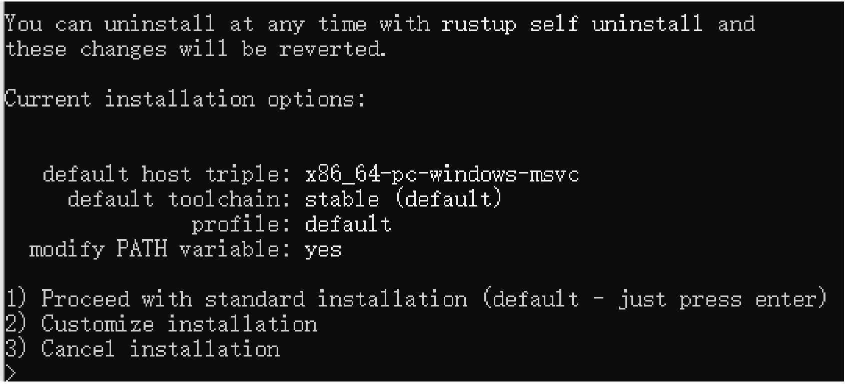
1.1.2 安装Rust

```
# 设置环境变量
RUSTUP_DIST_SERVER
https://mirrors.tools.huawei.com/rustup/
RUSTUP_UPDATE_ROOT
https://mirrors.tools.huawei.com/rustup/rustup/
```



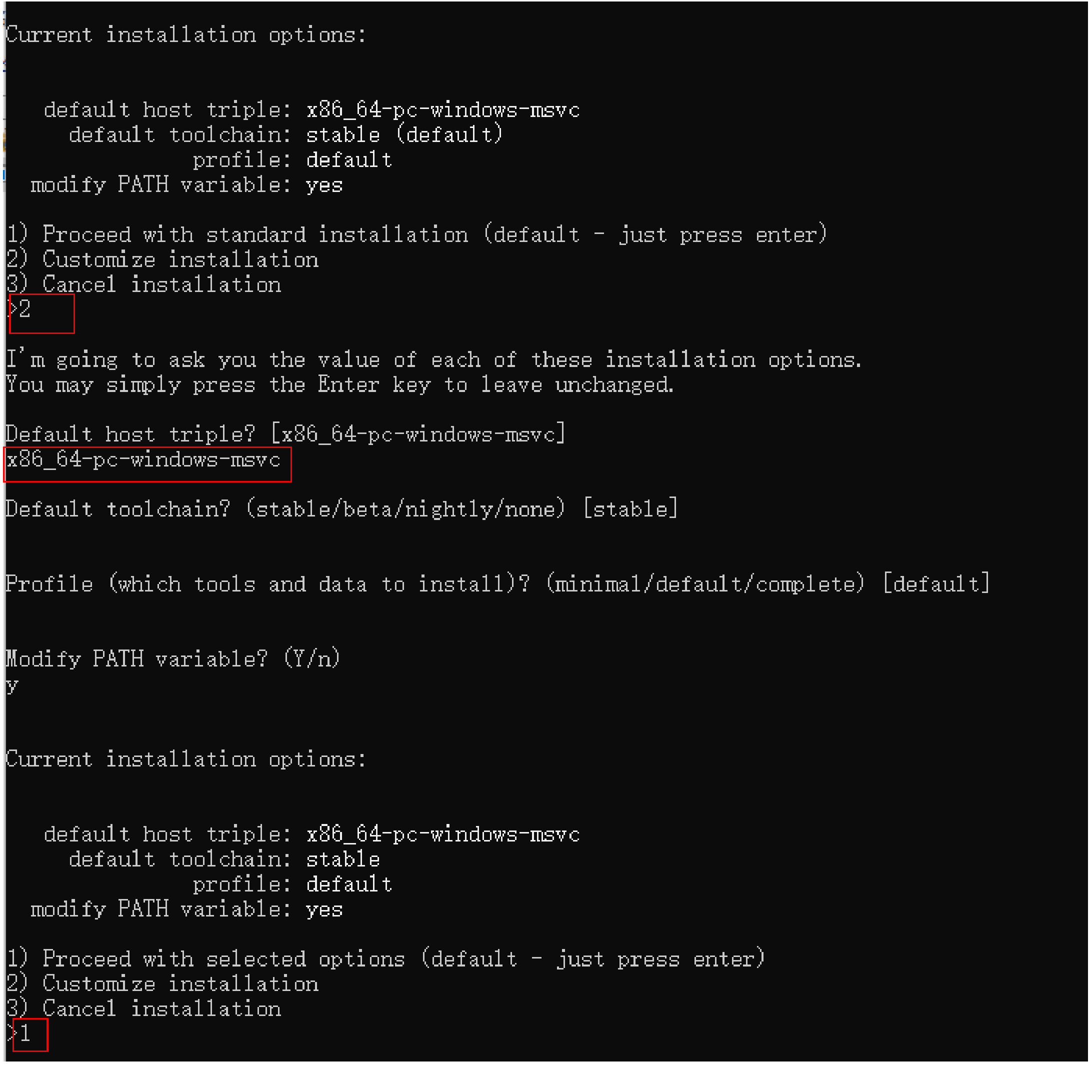
变量	值
PROCESSOR_LEVEL	6
PROCESSOR_REVISION	a505
PSModulePath	%ProgramFiles%\WindowsPowerShell\Modules;C:\windows\s...
RUSTUP_DIST_SERVER	https://mirrors.tools.huawei.com/rustup/
RUSTUP_UPDATE_ROOT	https://mirrors.tools.huawei.com/rustup/rustup/
TEMP	C:\windows\TEMP
TMP	C:\windows\TEMP

下载rustup-init.exe后（附件自行更改后缀），直接双击。



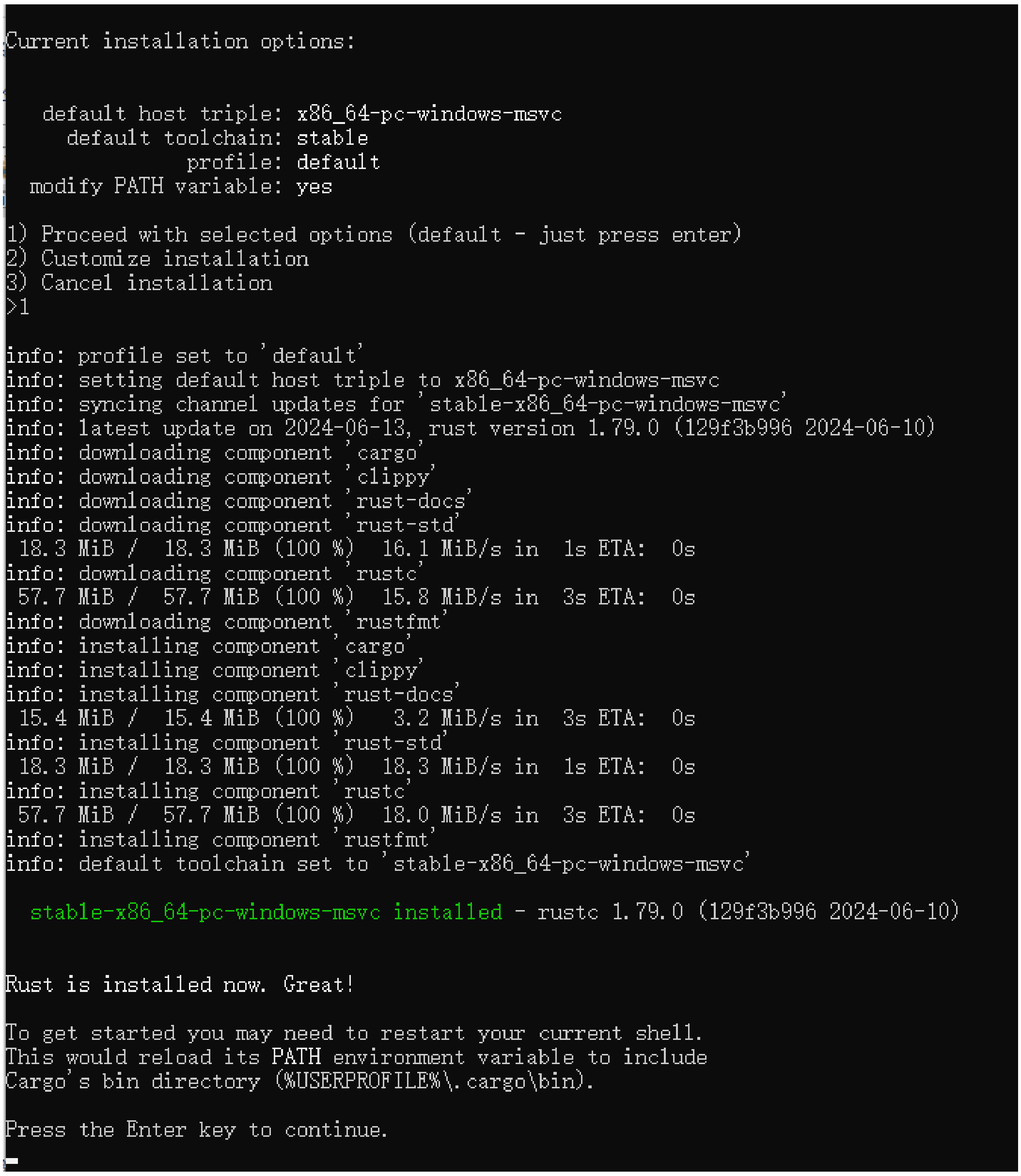
```
You can uninstall at any time with rustup self uninstall and
these changes will be reverted.

Current installation options:


   default host triple: x86_64-pc-windows-msvc
     default toolchain: stable (default)
               profile: default
  modify PATH variable: yes

1) Proceed with standard installation (default - just press enter)
2) Customize installation
3) Cancel installation
>
```

依次输入2 / x86_64-pc-windows-msvc / enter / enter/ y / 1，如下所示。



```
Current installation options:


   default host triple: x86_64-pc-windows-msvc
     default toolchain: stable (default)
               profile: default
  modify PATH variable: yes

1) Proceed with standard installation (default - just press enter)
2) Customize installation
3) Cancel installation
>2

I'm going to ask you the value of each of these installation options.
You may simply press the Enter key to leave unchanged.

Default host triple? [x86_64-pc-windows-msvc]
x86_64-pc-windows-msvc

Default toolchain? (stable/beta/nightly/none) [stable]


Profile (which tools and data to install)? (minimal/default/complete) [default]


Modify PATH variable? (Y/n)
y


Current installation options:


   default host triple: x86_64-pc-windows-msvc
     default toolchain: stable
               profile: default
  modify PATH variable: yes

1) Proceed with selected options (default - just press enter)
2) Customize installation
3) Cancel installation
>1
```

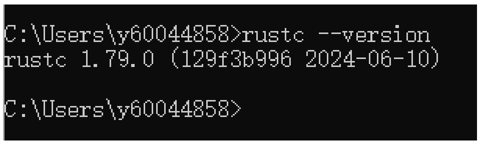
安装成功会有以下提示。



```
Current installation options:


   default host triple: x86_64-pc-windows-msvc
     default toolchain: stable
               profile: default
  modify PATH variable: yes

1) Proceed with selected options (default - just press enter)
2) Customize installation
3) Cancel installation
>1

info: profile set to 'default'
info: setting default host triple to x86_64-pc-windows-msvc
info: syncing channel updates for 'stable-x86_64-pc-windows-msvc'
info: latest update on 2024-06-13, rust version 1.79.0 (129f3b996 2024-06-10)
info: downloading component 'cargo'
info: downloading component 'clippy'
info: downloading component 'rust-docs'
info: downloading component 'rust-std'
 18.3 MiB /  18.3 MiB (100 %)  16.1 MiB/s in  1s ETA:  0s
info: downloading component 'rustc'
 57.7 MiB /  57.7 MiB (100 %)  15.8 MiB/s in  3s ETA:  0s
info: downloading component 'rustfmt'
info: installing component 'cargo'
info: installing component 'clippy'
info: installing component 'rust-docs'
 15.4 MiB /  15.4 MiB (100 %)   3.2 MiB/s in  3s ETA:  0s
info: installing component 'rust-std'
 18.3 MiB /  18.3 MiB (100 %)  18.3 MiB/s in  1s ETA:  0s
info: installing component 'rustc'
 57.7 MiB /  57.7 MiB (100 %)  18.0 MiB/s in  3s ETA:  0s
info: installing component 'rustfmt'
info: default toolchain set to 'stable-x86_64-pc-windows-msvc'

  stable-x86_64-pc-windows-msvc installed - rustc 1.79.0 (129f3b996 2024-06-10)


Rust is installed now. Great!

To get started you may need to restart your current shell.
This would reload its PATH environment variable to include
Cargo's bin directory (%USERPROFILE%\.cargo\bin).

Press the Enter key to continue.
```

```
# 配置rust环境变量
PATH=C:\Users\y60044858\.rustup\toolchains\innersource-distribution-x86_64-pc-windows-msvc\bin
```

查看安装信息，执行rustc --version



```
C:\Users\y60044858>rustc --version
rustc 1.79.0 (129f3b996 2024-06-10)

C:\Users\y60044858>
```

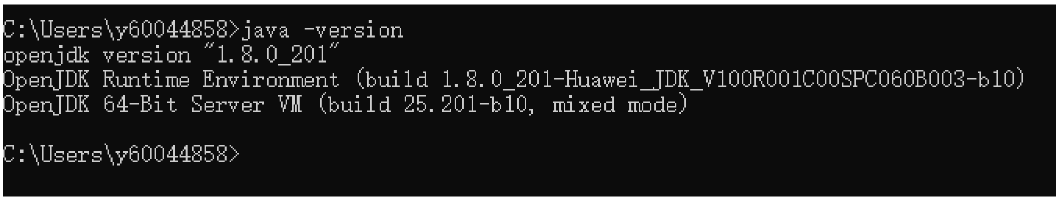
1.2 JDK

基于windows操作系统安装JDK1.8（jdk-8u201-windows-x64.msi）和配置环境变量。

执行jdk-8u201-windows-x64.msi安装成功。

```
# JDK环境变量配置：
JAVA_HOME=D:\D\Android\Java\jdk1.8.0_201
CLASSPATH=.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar
PATH=%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin
```

查看安装信息，执行java -version



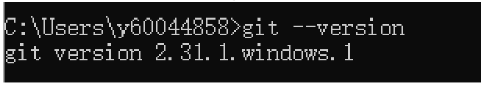
```
C:\Users\y60044858>java -version
openjdk version "1.8.0_201"
OpenJDK Runtime Environment (build 1.8.0_201-Huawei_JDK_V100R001C00SPC060B003-b10)
OpenJDK 64-Bit Server VM (build 25.201-b10, mixed mode)

C:\Users\y60044858>
```

1.3 Git

下载Git
解压Git-2.31.1-64-bit.rar并安装成功。

```
# 环境变量配置：
PATH=D:\D\Git\bin
```

查看安装信息，执行git --version



```
C:\Users\y60044858>git --version
git version 2.31.1.windows.1
```

```
# GIT网络代理配置：
# 查看全局配置变量
git config --list
# 使用命令配置
git config --global http.proxy http://y60044858:password@proxyhk.huawei.com:8080/
git config --global https.proxy https://y60044858:password@proxyhk.huawei.com:8080/
git config --global http.sslverify false
# 若取消配置，可以执行下面命令
git config --global --unset http.proxy
git config --global --unset https.proxy
```

1.4 Android SDK、NDK 和 CMake

下载android-sdk_r24.4.1-windows.zip并解压。

1.4.1 adb环境变量配置

```
PATH=D:\D\Android\androidSDK\android-sdk_r24.4.1-windows\platform-tools
```

1.4.2 NDK环境变量配置

```
ANDROID_NDK=D:\D\Android\androidSDK\android-sdk_r24.4.1-windows\ndk\25.1.8937393
TVM_NDK_CC=%ANDROID_NDK%/toolchains/llvm/prebuilt/windows-x86_64/bin/aarch64-linux-android24-clang
```

1.4.3 CMake环境变量配置

```
PATH=D:\D\Android\androidSDK\android-sdk_r24.4.1-windows\cmake\3.22.1\bin
```

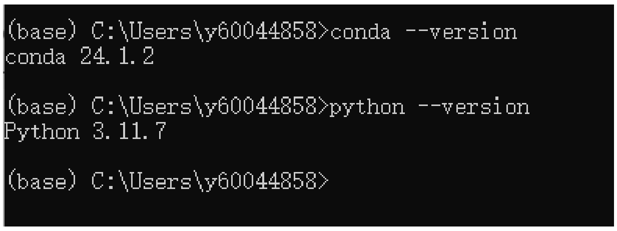
1.5 Android Studio

下载android-studio-2023.2.1.23-windows.exe并安装。

1.6 conda

下载Anaconda3-2024.02-1-Windows-x86_64.exe并安装。使用conda 来管理隔离的 Python 环境，以避免缺少依赖项、版本不兼容和包冲突。

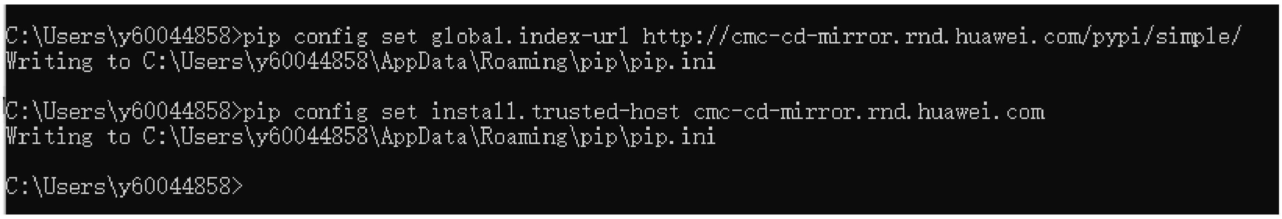
查看安装信息



```
(base) C:\Users\y60044858>conda --version
conda 24.1.2

(base) C:\Users\y60044858>python --version
Python 3.11.7

(base) C:\Users\y60044858>
```

华为python镜像源配置

```
# python配置华为pip镜像源:
# pip有创建pip.ini的命令，不用自己创文件夹写文件。
# cmd命令输入以下两行命令:
pip config set global.index-url http://cmc-cd-mirror.rnd.huawei.com/pypi/simple/
pip config set install.trusted-host cmc-cd-mirror.rnd.huawei.com
```



```
C:\Users\y60044858>pip config set global.index-url http://cmc-cd-mirror.rnd.huawei.com/pypi/simple/
Writing to C:\Users\y60044858\AppData\Roaming\pip\pip.ini

C:\Users\y60044858>pip config set install.trusted-host cmc-cd-mirror.rnd.huawei.com
Writing to C:\Users\y60044858\AppData\Roaming\pip\pip.ini

C:\Users\y60044858>
```

修改conda的配置文件

在黄区网络和绿区网络无法使用外网的源或者镜像源，需要修改conda的配置文件。如果不知道自己的网络是黄区还是绿区，执行 ping 10.155.97.225，能 ping 通则是绿区。黄区则执行 ping 10.155.124.25。
本人操作系统为windows，配置文件在C:\Users\y60044858.condarc，将.condarc文件修改为如下内容：

```
channel_alias: http://10.155.97.225:8088/repository/conda-proxy
default_channels:
- main
- r
channels:
- defaults
channel_priority: strict
show_channel_urls: true
```

二、MLC-LLM源代码构建Android应用程序

2.1 mlc-ai/mlc-llm源代码下载

```
# 指定docs_typo_mlc_chat分支克隆
git clone -b docs_typo_mlc_chat --single-branch https://github.com/mlc-ai/mlc-llm.git
# 进入mlc-llm项目
cd mlc-llm
# 克隆子模块代码
git submodule update --init --recursive
# 进入MLCChat目录
cd ./android/MLCChat
```

代码的环境变量配置

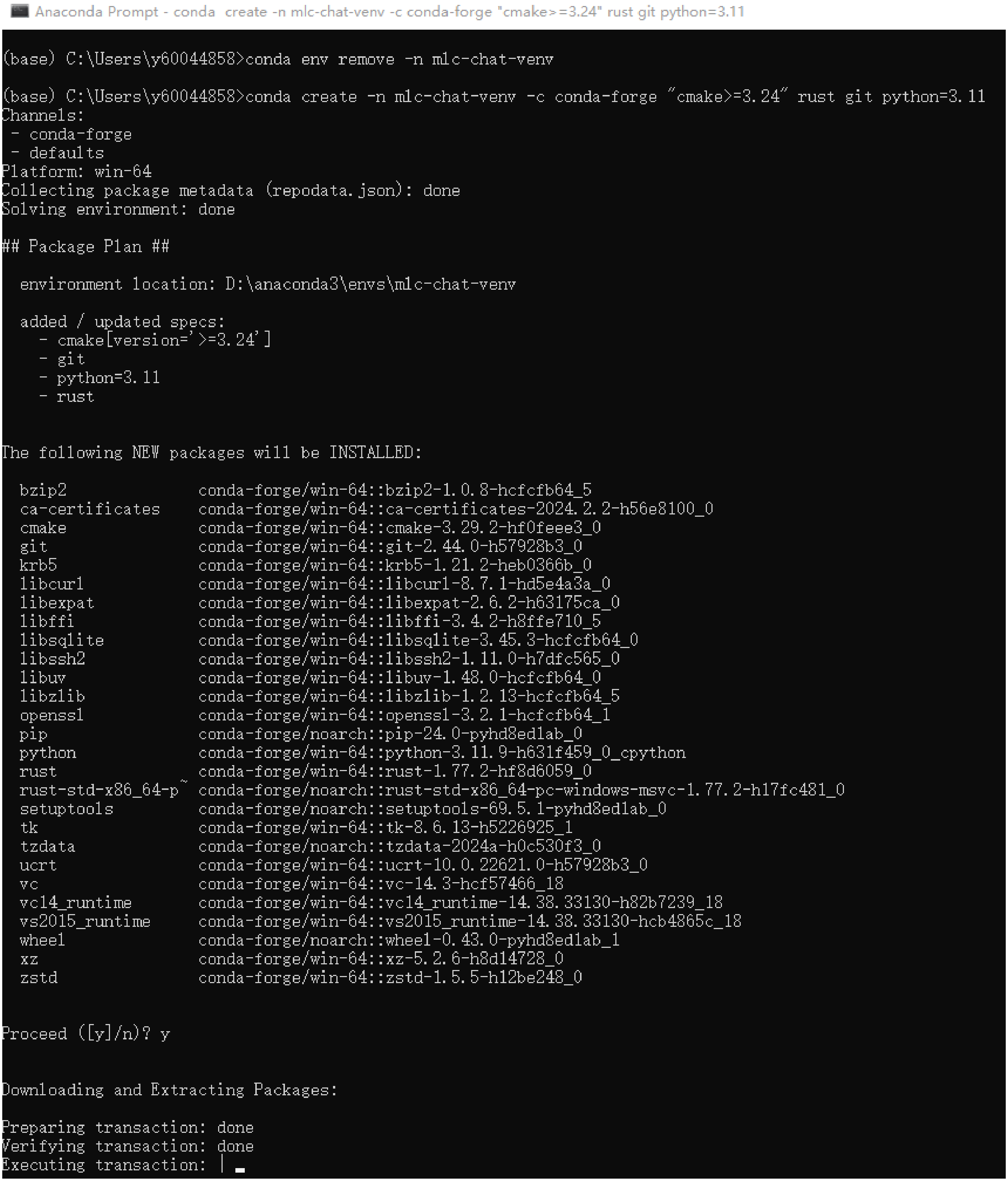
```
# mlc-llm代码的路径
MLC_LLM_SOURCE_DIR=D:\mlc-llm
# TVM Unity 运行时位于MLC LLM中的3rdparty/tvm下，因此无需安装任何额外的内容。设置以下环境变量
TVM_SOURCE_DIR=D:\mlc-llm\3rdparty\tvm
```

2.2 安装MLC LLM Python包

MLC LLM Python 包可以直接从预构建的开发人员包安装，也可以从源代码构建。下面是通过预构建软件包。

在Conda中设置构建依赖项

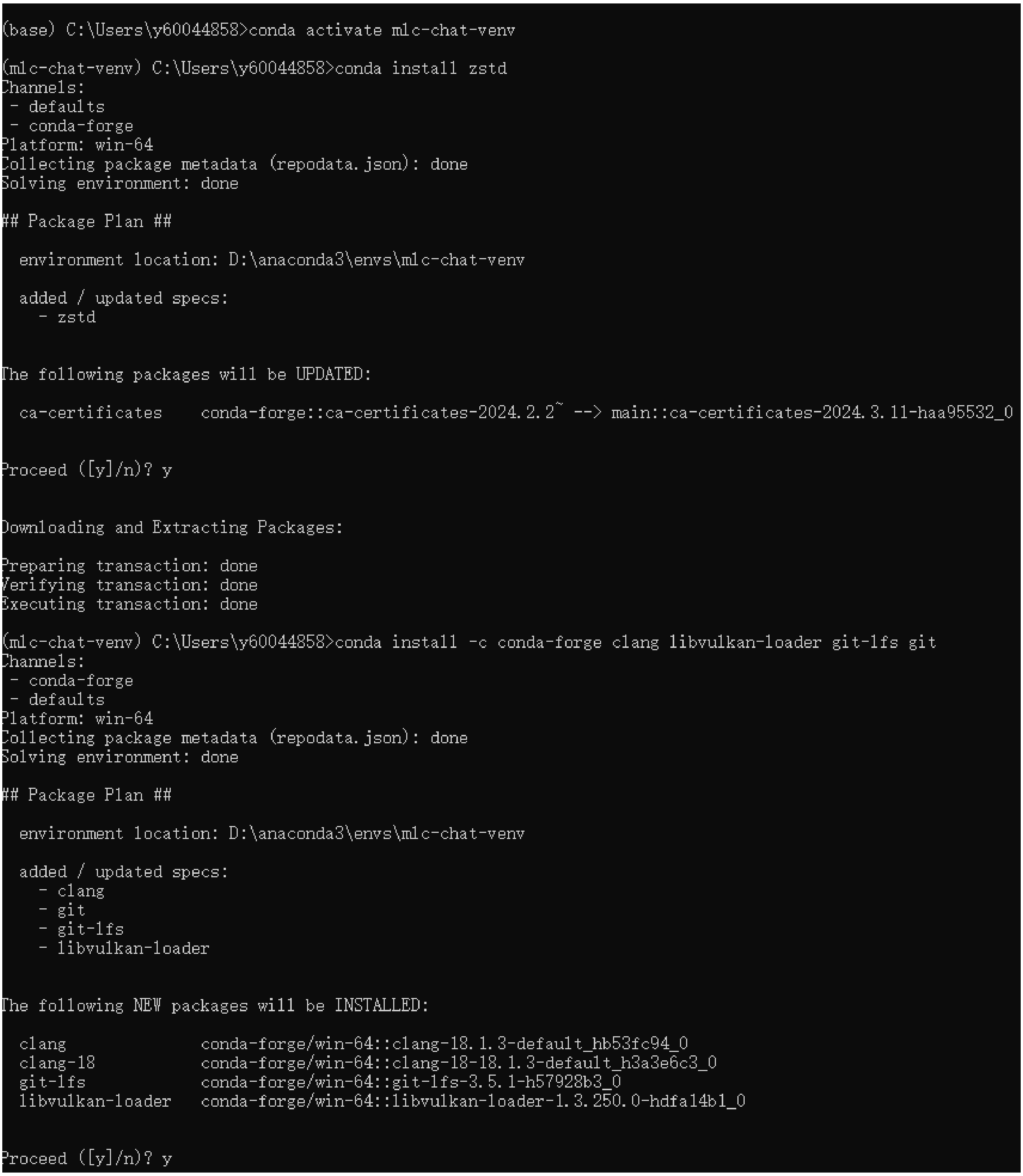
```
# make sure to start with a fresh environment
conda env remove -n mlc-chat-venv
# create the conda environment with build dependency
conda create -n mlc-chat-venv -c conda-forge "cmake>=3.24" rust git python=3.11
```



Anaconda Prompt - conda create -n mlc-chat-venv -c conda-forge "cmake>=3.24" rust git python=3.11

```
(base) C:\Users\y60044858>conda env remove -n mlc-chat-venv

(base) C:\Users\y60044858>conda create -n mlc-chat-venv -c conda-forge "cmake>=3.24" rust git python=3.11
Channels:
 - conda-forge
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: D:\anaconda3\envs\mlc-chat-venv

  added / updated specs:
    - cmake[version='>=3.24']
    - git
    - python=3.11
    - rust


The following NEW packages will be INSTALLED:

  bzip2              conda-forge/win-64::bzip2-1.0.8-hcfcfb64_5
  ca-certificates    conda-forge/win-64::ca-certificates-2024.2.2-h56e8100_0
  cmake              conda-forge/win-64::cmake-3.29.2-hf0feee3_0
  git                conda-forge/win-64::git-2.44.0-h57928b3_0
  krb5               conda-forge/win-64::krb5-1.21.2-heb0366b_0
  libcurl            conda-forge/win-64::libcurl-8.7.1-hd5e4a3a_0
  libexpat           conda-forge/win-64::libexpat-2.6.2-h63175ca_0
  libffi             conda-forge/win-64::libffi-3.4.2-h8ffe710_5
  libsqlite          conda-forge/win-64::libsqlite-3.45.3-hcfcfb64_0
  libssh2            conda-forge/win-64::libssh2-1.11.0-h7dfc565_0
  libuv              conda-forge/win-64::libuv-1.48.0-hcfcfb64_0
  libzlib            conda-forge/win-64::libzlib-1.2.13-hcfcfb64_5
  openssl            conda-forge/win-64::openssl-3.2.1-hcfcfb64_1
  pip                conda-forge/noarch::pip-24.0-pyhd8ed1ab_0
  python             conda-forge/win-64::python-3.11.9-h631f459_0_cpython
  rust               conda-forge/win-64::rust-1.77.2-hf8d6059_0
  rust-std-x86_64-p~ conda-forge/noarch::rust-std-x86_64-pc-windows-msvc-1.77.2-h17fc481_0
  setuptools         conda-forge/noarch::setuptools-69.5.1-pyhd8ed1ab_0
  tk                 conda-forge/win-64::tk-8.6.13-h5226925_1
  tzdata             conda-forge/noarch::tzdata-2024a-h0c530f3_0
  ucrt               conda-forge/win-64::ucrt-10.0.22621.0-h57928b3_0
  vc                 conda-forge/win-64::vc-14.3-hcf57466_18
  vc14_runtime       conda-forge/win-64::vc14_runtime-14.38.33130-h82b7239_18
  vs2015_runtime     conda-forge/win-64::vs2015_runtime-14.38.33130-hcb4865c_18
  wheel              conda-forge/noarch::wheel-0.43.0-pyhd8ed1ab_1
  xz                 conda-forge/win-64::xz-5.2.6-h8d14728_0
  zstd               conda-forge/win-64::zstd-1.5.5-h12be248_0


Proceed ([y]/n)? y


Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: | _
```

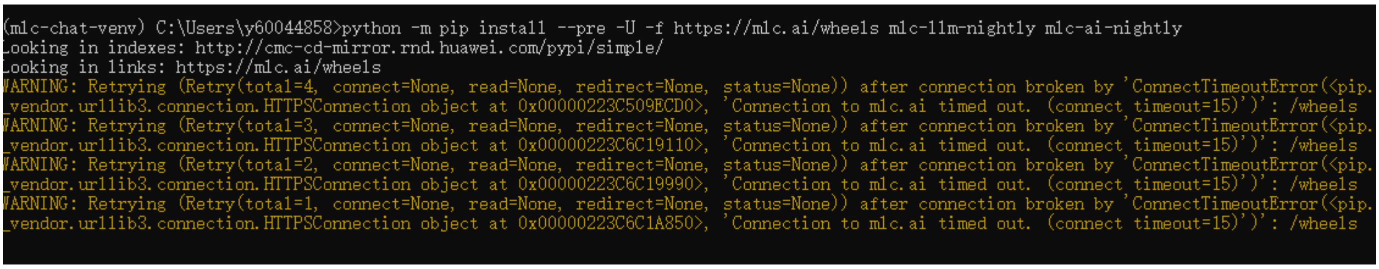
```
# enter the build environment
conda activate mlc-chat-venv
# 安装zstd
conda install zstd
# 安装vulkan loader、clang、git 和 git-lfs，以启用正确的自动下载和 jit 编译。
conda install -c conda-forge clang libvulkan-loader git-lfs git
```



```
(base) C:\Users\y60044858>conda activate mlc-chat-venv

(mlc-chat-venv) C:\Users\y60044858>conda install zstd
Channels:
 - defaults
 - conda-forge
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: D:\anaconda3\envs\mlc-chat-venv

  added / updated specs:
    - zstd


The following packages will be UPDATED:

  ca-certificates    conda-forge::ca-certificates-2024.2.2~ --> main::ca-certificates-2024.3.11-haa95532_0


Proceed ([y]/n)? y


Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(mlc-chat-venv) C:\Users\y60044858>conda install -c conda-forge clang libvulkan-loader git-lfs git
Channels:
 - conda-forge
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: D:\anaconda3\envs\mlc-chat-venv

  added / updated specs:
    - clang
    - git
    - git-lfs
    - libvulkan-loader


The following NEW packages will be INSTALLED:

  clang              conda-forge/win-64::clang-18.1.3-default_hb53fc94_0
  clang-18           conda-forge/win-64::clang-18-18.1.3-default_h3a3e6c3_0
  git-lfs            conda-forge/win-64::git-lfs-3.5.1-h57928b3_0
  libvulkan-loader   conda-forge/win-64::libvulkan-loader-1.3.250.0-hdfa14b1_0


Proceed ([y]/n)? y
```

安装mlc-llm-nightly和mlc-ai-nightly

```
python -m pip install --pre -U -f https://mlc.ai/wheels mlc-llm-nightly mlc-ai-nightly
```

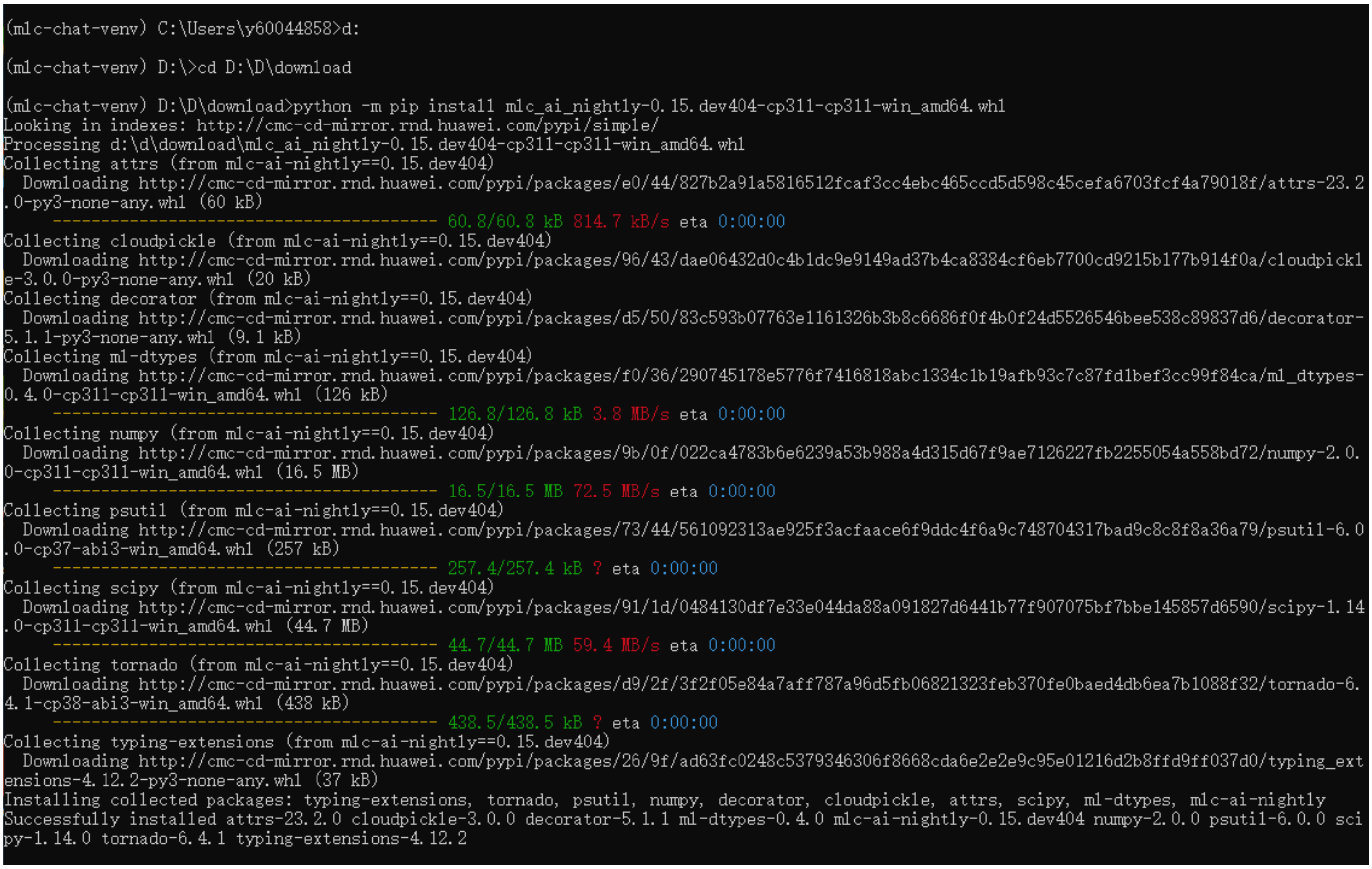


```
(mlc-chat-venv) C:\Users\y60044858>python -m pip install --pre -U -f https://mlc.ai/wheels mlc-llm-nightly mlc-ai-nightly
Looking in indexes: http://cmc-cd-mirror.rnd.huawei.com/pypi/simple/
Looking in links: https://mlc.ai/wheels
WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ConnectTimeoutError(<pip.
_vendor.urllib3.connection.HTTPSConnection object at 0x00000223C509ECD0>, 'Connection to mlc.ai timed out. (connect timeout=15)')': /wheels
WARNING: Retrying (Retry(total=3, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ConnectTimeoutError(<pip.
_vendor.urllib3.connection.HTTPSConnection object at 0x00000223C6C19110>, 'Connection to mlc.ai timed out. (connect timeout=15)')': /wheels
WARNING: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ConnectTimeoutError(<pip.
_vendor.urllib3.connection.HTTPSConnection object at 0x00000223C6C19990>, 'Connection to mlc.ai timed out. (connect timeout=15)')': /wheels
WARNING: Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ConnectTimeoutError(<pip.
_vendor.urllib3.connection.HTTPSConnection object at 0x00000223C6C1A850>, 'Connection to mlc.ai timed out. (connect timeout=15)')': /wheels
```

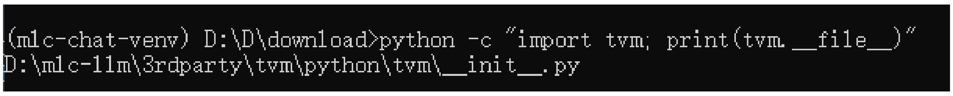
超时或找不到安装包，可以到网站https://mlc.ai/wheels 手动下载whl安装包后，用python -m pip install *.whl
根据phton版本下载mlc_ai_nightly-0.15.dev404-cp311-cp311-win_amd64.whl和mlc_llm_nightly-0.1.dev1404-cp311-cp311-win_amd64.whl

TVM Unity编辑器安装和验证

```
# 进入到*.whl的所在目录下
d:
cd D:\D\download
# TVM Unity编辑器安装:
python -m pip install mlc_ai_nightly-0.15.dev404-cp311-cp311-win_amd64.whl
# TVM的安装验证：以下命令可以帮助确认 TVM 是否已正确安装为 python 包并提供 TVM python 包的位置：
python -c "import tvm; print(tvm.__file__)"
```



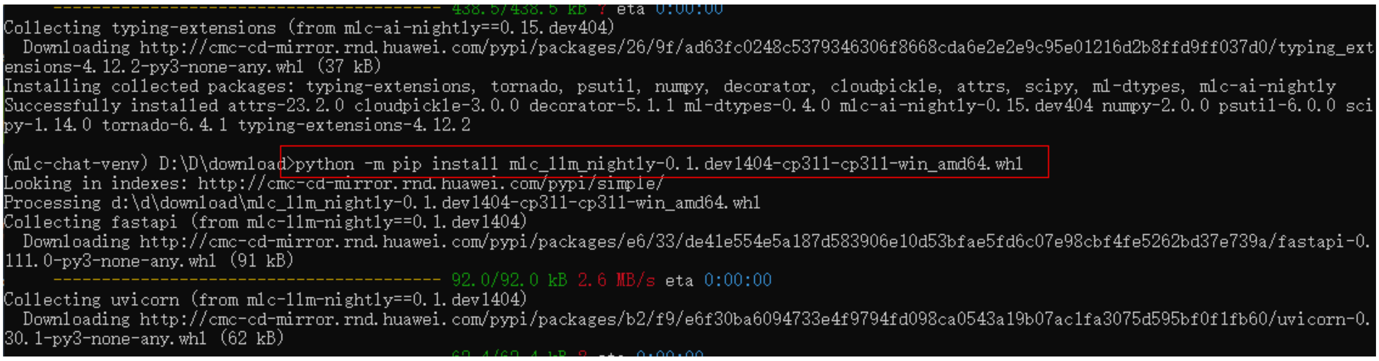
```
(mlc-chat-venv) C:\Users\y60044858>d:

(mlc-chat-venv) D:\>cd D:\D\download

(mlc-chat-venv) D:\D\download>python -m pip install mlc_ai_nightly-0. 15. dev404-cp311-cp311-win_amd64. whl
Looking in indexes: http://cmc-cd-mirror. rnd. huawei. com/pypi/simple/
Processing d:\d\download\mlc_ai_nightly-0. 15. dev404-cp311-cp311-win_amd64. whl
Collecting attrs (from mlc-ai-nightly==0. 15. dev404)
  Downloading http://cmc-cd-mirror. rnd. huawei. com/pypi/packages/e0/44/827b2a91a5816512fcaf3cc4ebc465ccd5d598c45cefa6703fcf4a79018f/attrs-23. 2
. 0-py3-none-any. whl (60 kB)
     ---------------------------------------- 60. 8/60. 8 kB 814. 7 kB/s eta 0:00:00
Collecting cloudpickle (from mlc-ai-nightly==0. 15. dev404)
  Downloading http://cmc-cd-mirror. rnd. huawei. com/pypi/packages/96/43/dae06432d0c4b1dc9e9149ad37b4ca8384cf6eb7700cd9215b177b914f0a/cloudpickl
e-3. 0. 0-py3-none-any. whl (20 kB)
Collecting decorator (from mlc-ai-nightly==0. 15. dev404)
  Downloading http://cmc-cd-mirror. rnd. huawei. com/pypi/packages/d5/50/83c593b07763e1161326b3b8c6686f0f4b0f24d5526546bee538c89837d6/decorator-
5. 1. 1-py3-none-any. whl (9. 1 kB)
Collecting ml-dtypes (from mlc-ai-nightly==0. 15. dev404)
  Downloading http://cmc-cd-mirror. rnd. huawei. com/pypi/packages/f0/36/290745178e5776f7416818abc1334c1b19afb93c7c87fd1bef3cc99f84ca/ml_dtypes-
0. 4. 0-cp311-cp311-win_amd64. whl (126 kB)
     ---------------------------------------- 126. 8/126. 8 kB 3. 8 MB/s eta 0:00:00
Collecting numpy (from mlc-ai-nightly==0. 15. dev404)
  Downloading http://cmc-cd-mirror. rnd. huawei. com/pypi/packages/9b/0f/022ca4783b6e6239a53b988a4d315d67f9ae7126227fb2255054a558bd72/numpy-2. 0.
0-cp311-cp311-win_amd64. whl (16. 5 MB)
     ---------------------------------------- 16. 5/16. 5 MB 72. 5 MB/s eta 0:00:00
Collecting psutil (from mlc-ai-nightly==0. 15. dev404)
  Downloading http://cmc-cd-mirror. rnd. huawei. com/pypi/packages/73/44/561092313ae925f3acfaace6f9ddc4f6a9c748704317bad9c8c8f8a36a79/psutil-6. 0
. 0-cp37-abi3-win_amd64. whl (257 kB)
     ---------------------------------------- 257. 4/257. 4 kB ? eta 0:00:00
Collecting scipy (from mlc-ai-nightly==0. 15. dev404)
  Downloading http://cmc-cd-mirror. rnd. huawei. com/pypi/packages/91/1d/0484130df7e33e044da88a091827d6441b77f907075bf7bbe145857d6590/scipy-1. 14
. 0-cp311-cp311-win_amd64. whl (44. 7 MB)
     ---------------------------------------- 44. 7/44. 7 MB 59. 4 MB/s eta 0:00:00
Collecting tornado (from mlc-ai-nightly==0. 15. dev404)
  Downloading http://cmc-cd-mirror. rnd. huawei. com/pypi/packages/d9/2f/3f2f05e84a7aff787a96d5fb06821323feb370fe0baed4db6ea7b1088f32/tornado-6.
4. 1-cp38-abi3-win_amd64. whl (438 kB)
     ---------------------------------------- 438. 5/438. 5 kB ? eta 0:00:00
Collecting typing-extensions (from mlc-ai-nightly==0. 15. dev404)
  Downloading http://cmc-cd-mirror. rnd. huawei. com/pypi/packages/26/9f/ad63fc0248c5379346306f8668cda6e2e2e9c95e01216d2b8ffd9ff037d0/typing_ext
ensions-4. 12. 2-py3-none-any. whl (37 kB)
Installing collected packages: typing-extensions, tornado, psutil, numpy, decorator, cloudpickle, attrs, scipy, ml-dtypes, mlc-ai-nightly
Successfully installed attrs-23. 2. 0 cloudpickle-3. 0. 0 decorator-5. 1. 1 ml-dtypes-0. 4. 0 mlc-ai-nightly-0. 15. dev404 numpy-2. 0. 0 psutil-6. 0. 0 sci
py-1. 14. 0 tornado-6. 4. 1 typing-extensions-4. 12. 2
```



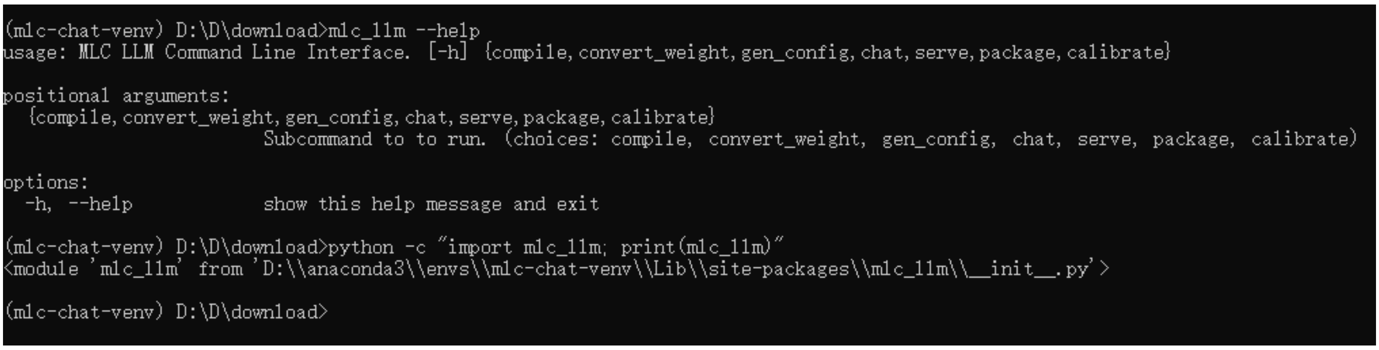
```
(mlc-chat-venv) D:\D\download>python -c "import tvm; print(tvm.__file__)"
D:\mlc-llm\3rdparty\tvm\python\tvm\__init__.py
```

mlc-llm安装和验证

```
# 安装mlc_llm_nightly
python -m pip install mlc_llm_nightly-0.1.dev1404-cp311-cp311-win_amd64.whl
# mlc_llm的验证
mlc_llm --help
python -c "import mlc_llm; print(mlc_llm)"
```



```
     ---------------------------------------- 438. 5/438. 5 kB ? eta 0:00:00
Collecting typing-extensions (from mlc-ai-nightly==0. 15. dev404)
  Downloading http://cmc-cd-mirror. rnd. huawei. com/pypi/packages/26/9f/ad63fc0248c5379346306f8668cda6e2e2e9c95e01216d2b8ffd9ff037d0/typing_ext
ensions-4. 12. 2-py3-none-any. whl (37 kB)
Installing collected packages: typing-extensions, tornado, psutil, numpy, decorator, cloudpickle, attrs, scipy, ml-dtypes, mlc-ai-nightly
Successfully installed attrs-23. 2. 0 cloudpickle-3. 0. 0 decorator-5. 1. 1 ml-dtypes-0. 4. 0 mlc-ai-nightly-0. 15. dev404 numpy-2. 0. 0 psutil-6. 0. 0 sci
py-1. 14. 0 tornado-6. 4. 1 typing-extensions-4. 12. 2

(mlc-chat-venv) D:\D\download>python -m pip install mlc_llm_nightly-0. 1. dev1404-cp311-cp311-win_amd64. whl
Looking in indexes: http://cmc-cd-mirror. rnd. huawei. com/pypi/simple/
Processing d:\d\download\mlc_llm_nightly-0. 1. dev1404-cp311-cp311-win_amd64. whl
Collecting fastapi (from mlc-llm-nightly==0. 1. dev1404)
  Downloading http://cmc-cd-mirror. rnd. huawei. com/pypi/packages/e6/33/de41e554e5a187d583906e10d53bfae5fd6c07e98cbf4fe5262bd37e739a/fastapi-0.
111. 0-py3-none-any. whl (91 kB)
     ---------------------------------------- 92. 0/92. 0 kB 2. 6 MB/s eta 0:00:00
Collecting uvicorn (from mlc-llm-nightly==0. 1. dev1404)
  Downloading http://cmc-cd-mirror. rnd. huawei. com/pypi/packages/b2/f9/e6f30ba6094733e4f9794fd098ca0543a19b07ac1fa3075d595bf0f1fb60/uvicorn-0.
30. 1-py3-none-any. whl (62 kB)
```

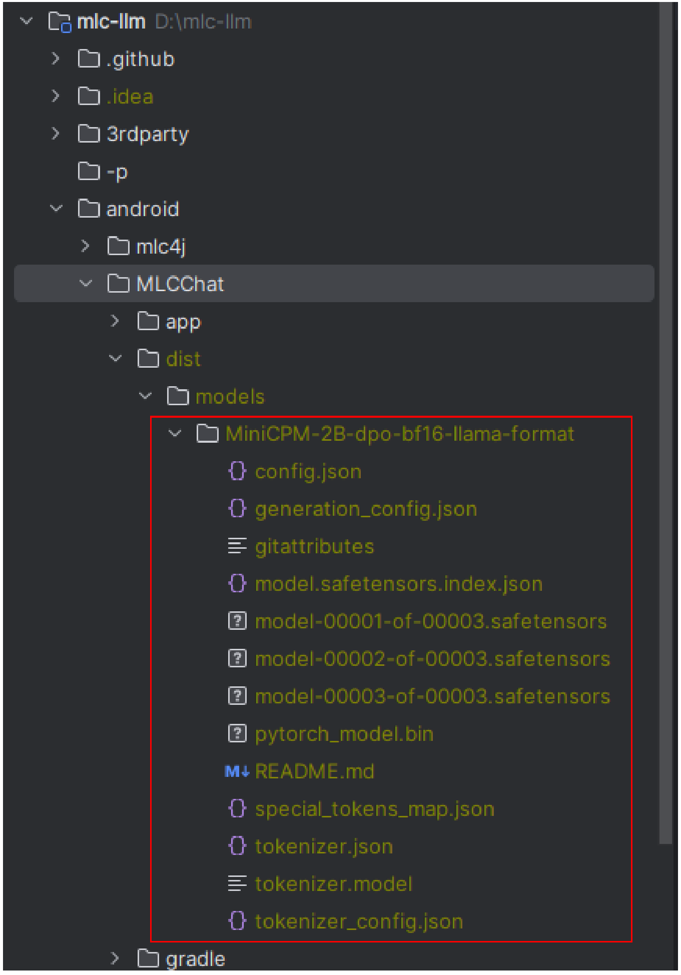


```
(mlc-chat-venv) D:\D\download>mlc_llm --help
usage: MLC LLM Command Line Interface. [-h] {compile,convert_weight,gen_config,chat,serve,package,calibrate}

positional arguments:
  {compile,convert_weight,gen_config,chat,serve,package,calibrate}
                        Subcommand to to run. (choices: compile, convert_weight, gen_config, chat, serve, package, calibrate)

options:
  -h, --help            show this help message and exit

(mlc-chat-venv) D:\D\download>python -c "import mlc_llm; print(mlc_llm)"
<module 'mlc_llm' from 'D:\\anaconda3\\envs\\mlc-chat-venv\\Lib\\site-packages\\mlc_llm\\__init__.py'>

(mlc-chat-venv) D:\D\download>
```

2.3 转换模型权重

要使用 MLC LLM运行模型，需要将模型权重转换。将hugginface模型作为输入，并将量化为与MLC兼容的权重。

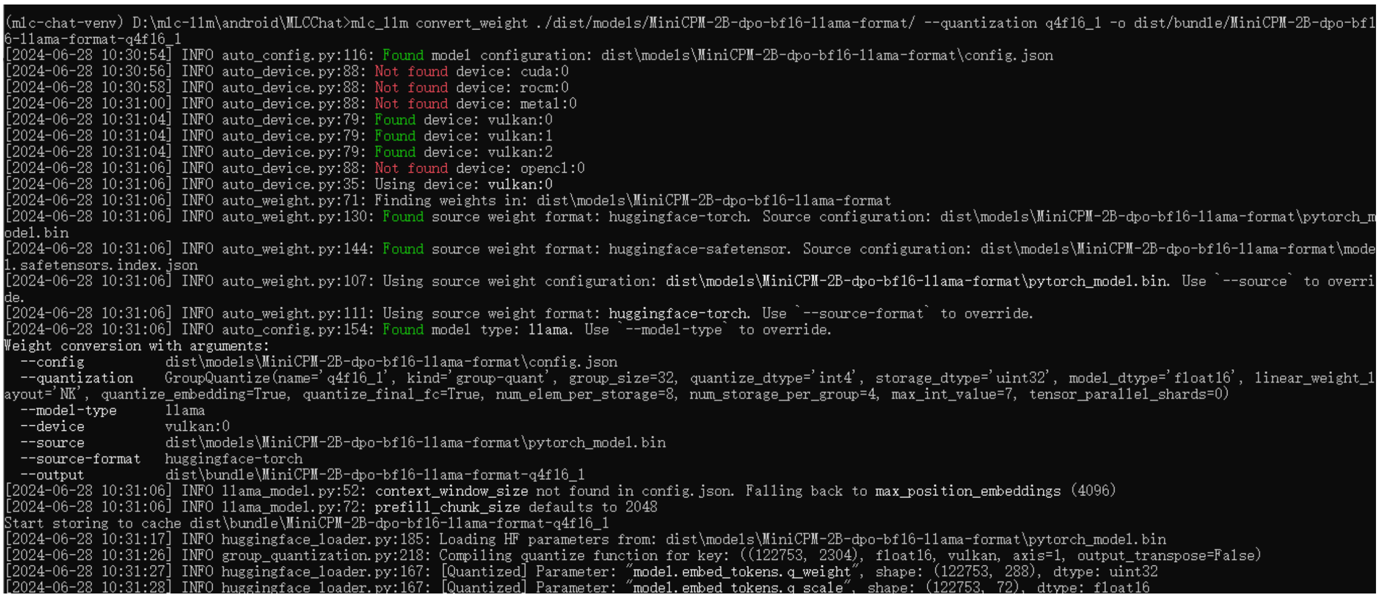
下载MiniCPM-2B-dpo-bf16-llama-format模型库
官网huggingface下载openbmb/MiniCPM-2B-dpo-bf16-llama-format，放入 dist/models目录。



```
mlc-llm D:\mlc-llm
  .github
  .idea
  3rdparty
  -p
  android
    mlc4j
    MLCChat
      app
      dist
        models
          MiniCPM-2B-dpo-bf16-llama-format
            config.json
            generation_config.json
            gitattributes
            model.safetensors.index.json
            model-00001-of-00003.safetensors
            model-00002-of-00003.safetensors
            model-00003-of-00003.safetensors
            pytorch_model.bin
            README.md
            special_tokens_map.json
            tokenizer.json
            tokenizer.model
            tokenizer_config.json
      gradle
```

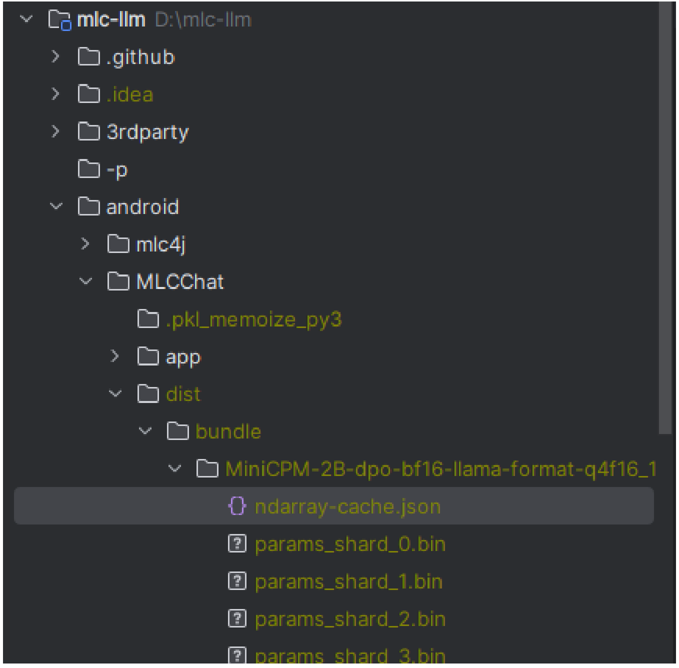
convert_weight权重转换

```
# 进入mlc-llm的安卓MLCChat根目录
cd D:\mlc-llm\android\MLCChat
# MiniCPM-2B-dpo-bf16-llama-format模型转换
mlc_llm convert_weight ./dist/models/MiniCPM-2B-dpo-bf16-llama-format/ --quantization q4f16_1
-o dist/bundle/MiniCPM-2B-dpo-bf16-llama-format-q4f16_1
```



```
(mlc-chat-venv) D:\mlc-llm\android\MLCChat>mlc_llm convert_weight ./dist/models/MiniCPM-2B-dpo-bf16-llama-format/ --quantization q4f16_1 -o dist/bundle/MiniCPM-2B-dpo-bf16-llama-format-q4f16_1
[2024-06-28 10:30:54] INFO auto_config.py:116: Found model configuration: dist\models\MiniCPM-2B-dpo-bf16-llama-format\config.json
[2024-06-28 10:30:56] INFO auto_device.py:88: Not found device: cuda:0
[2024-06-28 10:30:58] INFO auto_device.py:88: Not found device: rocm:0
[2024-06-28 10:31:00] INFO auto_device.py:88: Not found device: metal:0
[2024-06-28 10:31:04] INFO auto_device.py:79: Found device: vulkan:0
[2024-06-28 10:31:04] INFO auto_device.py:79: Found device: vulkan:1
[2024-06-28 10:31:04] INFO auto_device.py:79: Found device: vulkan:2
[2024-06-28 10:31:06] INFO auto_device.py:88: Not found device: opencl:0
[2024-06-28 10:31:06] INFO auto_device.py:35: Using device: vulkan:0
[2024-06-28 10:31:06] INFO auto_weight.py:71: Finding weights in: dist\models\MiniCPM-2B-dpo-bf16-llama-format
[2024-06-28 10:31:06] INFO auto_weight.py:130: Found source weight format: huggingface-torch. Source configuration: dist\models\MiniCPM-2B-dpo-bf16-llama-format\pytorch_model.bin
[2024-06-28 10:31:06] INFO auto_weight.py:144: Found source weight format: huggingface-safetensor. Source configuration: dist\models\MiniCPM-2B-dpo-bf16-llama-format\model.safetensors.index.json
[2024-06-28 10:31:06] INFO auto_weight.py:107: Using source weight configuration: dist\models\MiniCPM-2B-dpo-bf16-llama-format\pytorch_model.bin. Use `--source` to override.
[2024-06-28 10:31:06] INFO auto_weight.py:111: Using source weight format: huggingface-torch. Use `--source-format` to override.
[2024-06-28 10:31:06] INFO auto_config.py:154: Found model type: llama. Use `--model-type` to override.
Weight conversion with arguments:
  --config          dist\models\MiniCPM-2B-dpo-bf16-llama-format\config.json
  --quantization    GroupQuantize(name='q4f16_1', kind='group-quant', group_size=32, quantize_dtype='int4', storage_dtype='uint32', model_dtype='float16', linear_weight_layout='NK', quantize_embedding=True, quantize_final_fc=True, num_elem_per_storage=8, num_storage_per_group=4, max_int_value=7, tensor_parallel_shards=0)
  --model-type      llama
  --device          vulkan:0
  --source          dist\models\MiniCPM-2B-dpo-bf16-llama-format\pytorch_model.bin
  --source-format   huggingface-torch
  --output          dist\bundle\MiniCPM-2B-dpo-bf16-llama-format-q4f16_1
[2024-06-28 10:31:06] INFO llama_model.py:52: context_window_size not found in config.json. Falling back to max_position_embeddings (4096)
[2024-06-28 10:31:06] INFO llama_model.py:72: prefill_chunk_size defaults to 2048
Start storing to cache dist\bundle\MiniCPM-2B-dpo-bf16-llama-format-q4f16_1
[2024-06-28 10:31:17] INFO huggingface_loader.py:185: Loading HF parameters from: dist\models\MiniCPM-2B-dpo-bf16-llama-format\pytorch_model.bin
[2024-06-28 10:31:26] INFO group_quantization.py:218: Compiling quantize function for key: ((122753, 2304), float16, vulkan, axis=1, output_transpose=False)
[2024-06-28 10:31:27] INFO huggingface_loader.py:167: [Quantized] Parameter: "model.embed_tokens.q_weight", shape: (122753, 288), dtype: uint32
[2024-06-28 10:31:28] INFO huggingface_loader.py:167: [Quantized] Parameter: "model.embed_tokens.q_scale", shape: (122753, 72), dtype: float16
```

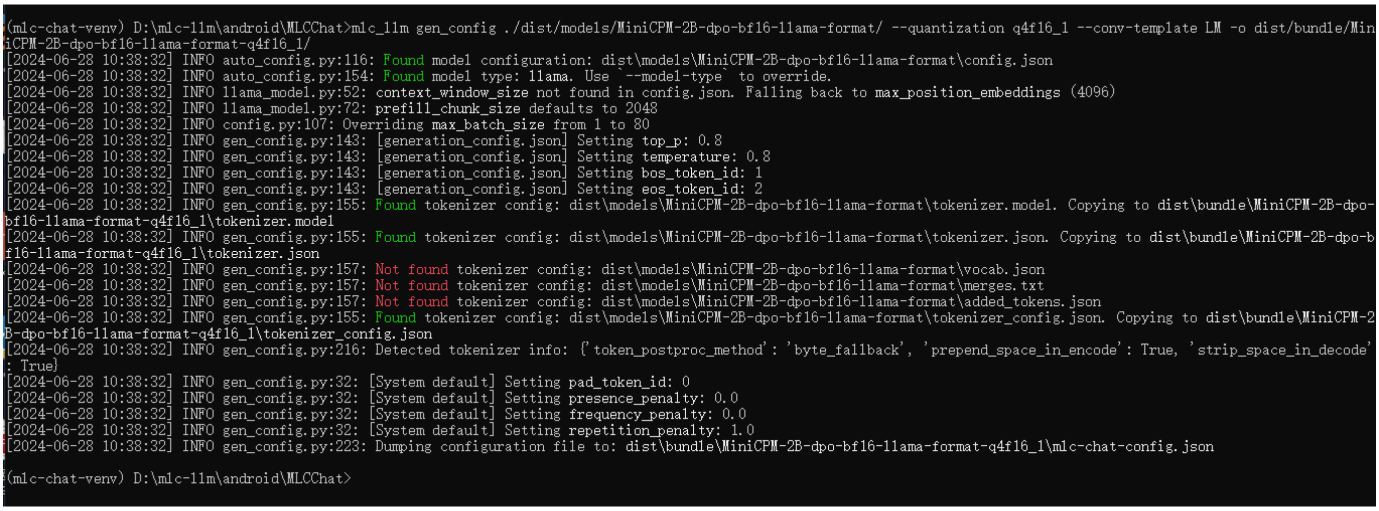
执行成功后，会在dist/bundle/MiniCPM-2B-dpo-bf16-llama-format-q4f16_1目录下生成ndarray-cache.json和params_shard_*.bin文件。



```
⌄ mlc-llm D:\mlc-llm
  › .github
  › .idea
  › 3rdparty
    -p
  ⌄ android
    › mlc4j
    ⌄ MLCChat
        .pkl_memoize_py3
      › app
      ⌄ dist
        ⌄ bundle
          ⌄ MiniCPM-2B-dpo-bf16-llama-format-q4f16_1
              ndarray-cache.json
              params_shard_0.bin
              params_shard_1.bin
              params_shard_2.bin
              params_shard_3.bin
```

生成MLC聊天配置

```
# 生成 MLC 聊天配置
mlc_llm gen_config ./dist/models/MiniCPM-2B-dpo-bf16-llama-format/ --quantization q4f16_1 -
-conv-template redpajama_chat -o dist/bundle/MiniCPM-2B-dpo-bf16-llama-format-q4f16_1/
```

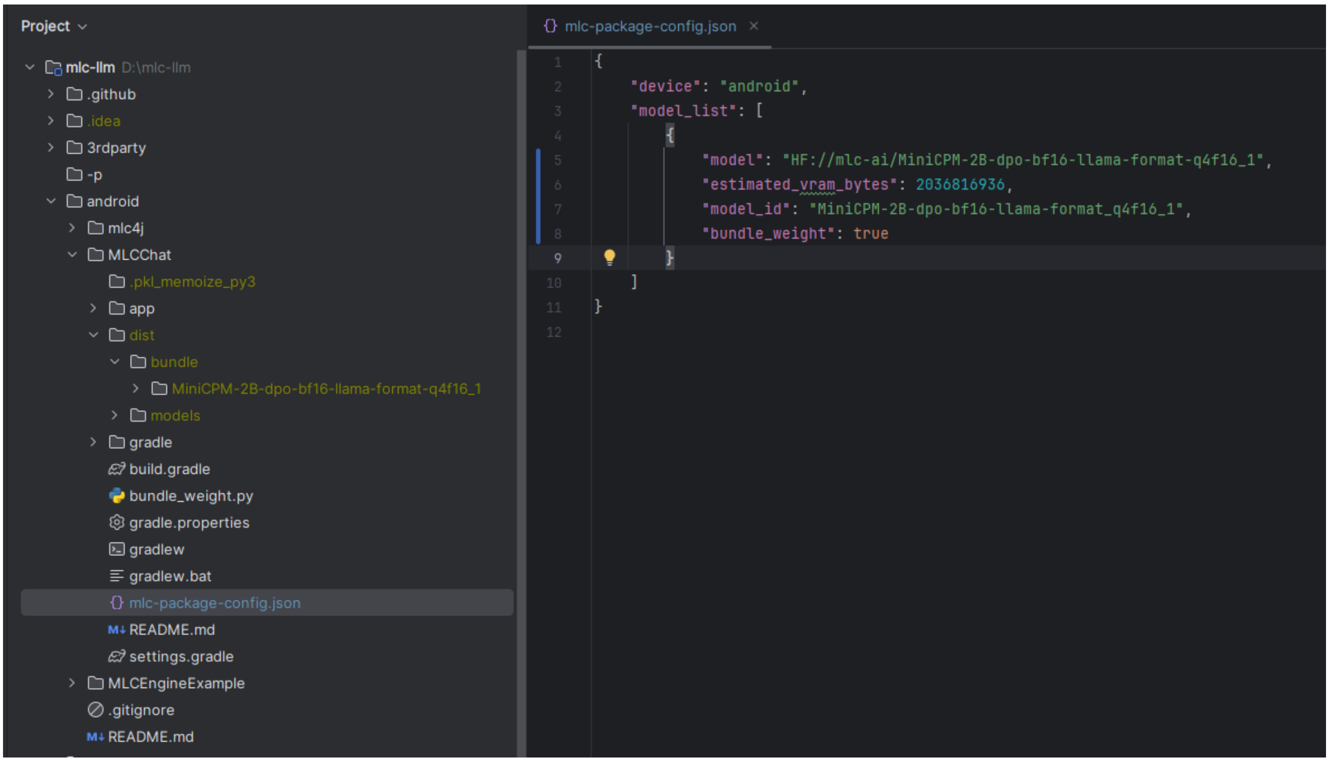


```
(mlc-chat-venv) D:\mlc-llm\android\MLCChat>mlc_llm gen_config ./dist/models/MiniCPM-2B-dpo-bf16-llama-format/ --quantization q4f16_1 --conv-template LM -o dist/bundle/MiniCPM-2B-dpo-bf16-llama-format-q4f16_1/
[2024-06-28 10:38:32] INFO auto_config.py:116: Found model configuration: dist\models\MiniCPM-2B-dpo-bf16-llama-format\config.json
[2024-06-28 10:38:32] INFO auto_config.py:154: Found model type: llama. Use `--model-type` to override.
[2024-06-28 10:38:32] INFO llama_model.py:52: context_window_size not found in config.json. Falling back to max_position_embeddings (4096)
[2024-06-28 10:38:32] INFO llama_model.py:72: prefill_chunk_size defaults to 2048
[2024-06-28 10:38:32] INFO config.py:107: Overriding max_batch_size from 1 to 80
[2024-06-28 10:38:32] INFO gen_config.py:143: [generation_config.json] Setting top_p: 0.8
[2024-06-28 10:38:32] INFO gen_config.py:143: [generation_config.json] Setting temperature: 0.8
[2024-06-28 10:38:32] INFO gen_config.py:143: [generation_config.json] Setting bos_token_id: 1
[2024-06-28 10:38:32] INFO gen_config.py:143: [generation_config.json] Setting eos_token_id: 2
[2024-06-28 10:38:32] INFO gen_config.py:155: Found tokenizer config: dist\models\MiniCPM-2B-dpo-bf16-llama-format\tokenizer.model. Copying to dist\bundle\MiniCPM-2B-dpo-bf16-llama-format-q4f16_1\tokenizer.model
[2024-06-28 10:38:32] INFO gen_config.py:155: Found tokenizer config: dist\models\MiniCPM-2B-dpo-bf16-llama-format\tokenizer.json. Copying to dist\bundle\MiniCPM-2B-dpo-bf16-llama-format-q4f16_1\tokenizer.json
[2024-06-28 10:38:32] INFO gen_config.py:157: Not found tokenizer config: dist\models\MiniCPM-2B-dpo-bf16-llama-format\vocab.json
[2024-06-28 10:38:32] INFO gen_config.py:157: Not found tokenizer config: dist\models\MiniCPM-2B-dpo-bf16-llama-format\merges.txt
[2024-06-28 10:38:32] INFO gen_config.py:157: Not found tokenizer config: dist\models\MiniCPM-2B-dpo-bf16-llama-format\added_tokens.json
[2024-06-28 10:38:32] INFO gen_config.py:155: Found tokenizer config: dist\models\MiniCPM-2B-dpo-bf16-llama-format\tokenizer_config.json. Copying to dist\bundle\MiniCPM-2B-dpo-bf16-llama-format-q4f16_1\tokenizer_config.json
[2024-06-28 10:38:32] INFO gen_config.py:216: Detected tokenizer info: {'token_postproc_method': 'byte_fallback', 'prepend_space_in_encode': True, 'strip_space_in_decode': True}
[2024-06-28 10:38:32] INFO gen_config.py:32: [System default] Setting pad_token_id: 0
[2024-06-28 10:38:32] INFO gen_config.py:32: [System default] Setting presence_penalty: 0.0
[2024-06-28 10:38:32] INFO gen_config.py:32: [System default] Setting frequency_penalty: 0.0
[2024-06-28 10:38:32] INFO gen_config.py:32: [System default] Setting repetition_penalty: 1.0
[2024-06-28 10:38:32] INFO gen_config.py:223: Dumping configuration file to: dist\bundle\MiniCPM-2B-dpo-bf16-llama-format-q4f16_1\mlc-chat-config.json

(mlc-chat-venv) D:\mlc-llm\android\MLCChat>
```

执行成功后，dist/bundle/MiniCPM-2B-dpo-bf16-llama-format-q4f16_1目录下会多生成mlc-chat-config.json、tokenizer.json、 tokenizer.model、tokenizer_config.json四个文件。

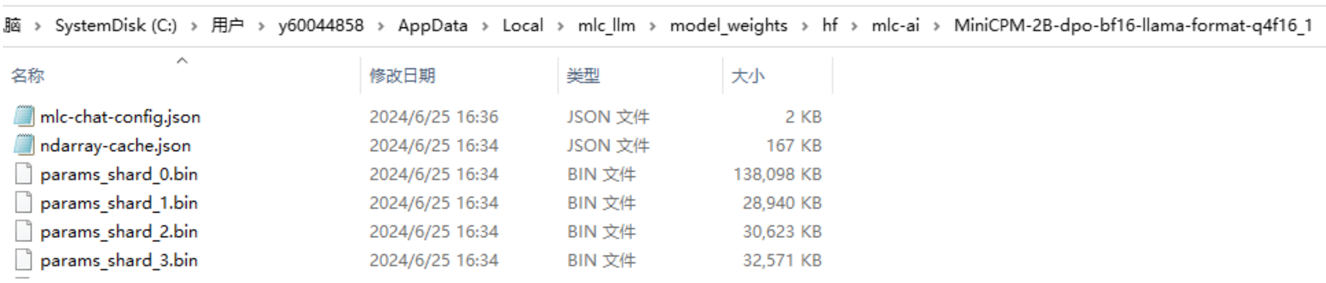
2.4 编译安卓生成tvm4j_core.jar包和libtvm4j_runtime_packed.so依赖库

1. 通过修改MLCChat/mlc-package-config.json文件来自定义Android应用程序中内置的模型。



```json
{
    "device": "android",
    "model_list": [
        {
            "model": "HF://mlc-ai/MiniCPM-2B-dpo-bf16-llama-format-q4f16_1",
            "estimated_vram_bytes": 2036816936,
            "model_id": "MiniCPM-2B-dpo-bf16-llama-format_q4f16_1",
            "bundle_weight": true
        }
    ]
}
```

2. 把转换好的MiniCPM-2B-dpo-bf16-llama-format-q4f16_1模型复制到C:\Users\y60044858\AppData\Local\mlc_llm\model_weights\hf\mlc-ai目录下。编译的过程中，会在本地先查找模型，若找不到会去官网https://huggingface.co/mlc-ai 下载。


脑 › SystemDisk (C:) › 用户 › y60044858 › AppData › Local › mlc_llm › model_weights › hf › mlc-ai › MiniCPM-2B-dpo-bf16-llama-format-q4f16_1

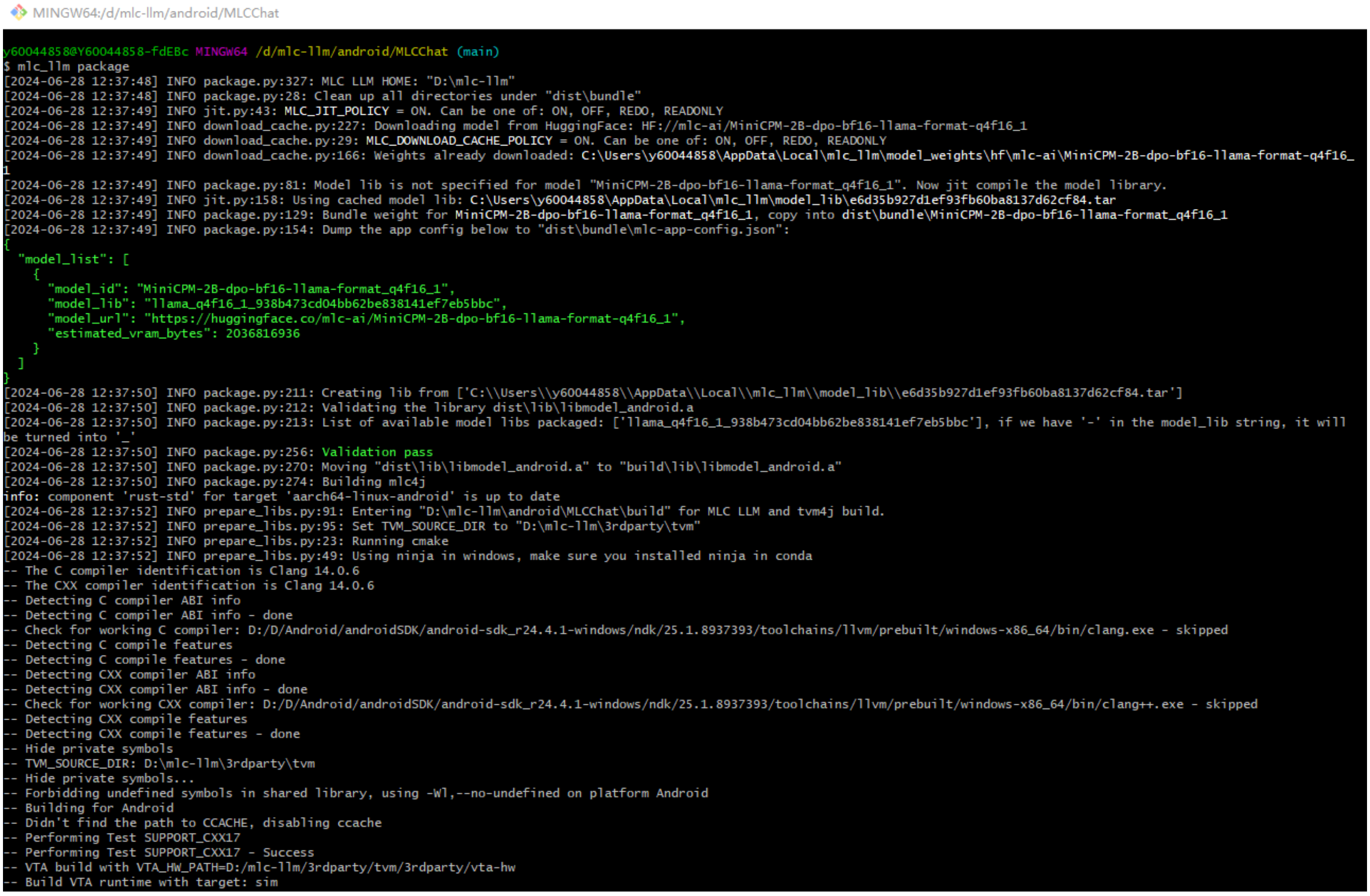
名称	修改日期	类型	大小
mlc-chat-config.json	2024/6/25 16:36	JSON 文件	2 KB
ndarray-cache.json	2024/6/25 16:34	JSON 文件	167 KB
params_shard_0.bin	2024/6/25 16:34	BIN 文件	138,098 KB
params_shard_1.bin	2024/6/25 16:34	BIN 文件	28,940 KB
params_shard_2.bin	2024/6/25 16:34	BIN 文件	30,623 KB
params_shard_3.bin	2024/6/25 16:34	BIN 文件	32,571 KB

3. 执行mlc_llm package命令行。执行过程中速度会稍微慢，请耐心等待。
   使用Git Base界面执行mlc_llm命令。
   首先配置python和mlc_llm的环境变量：

```
PATH=D:\anaconda3\envs\mlc-chat-venv
PATH=D:\anaconda3\envs\mlc-chat-venv\Scripts
```

执行mlc_llm package命令行

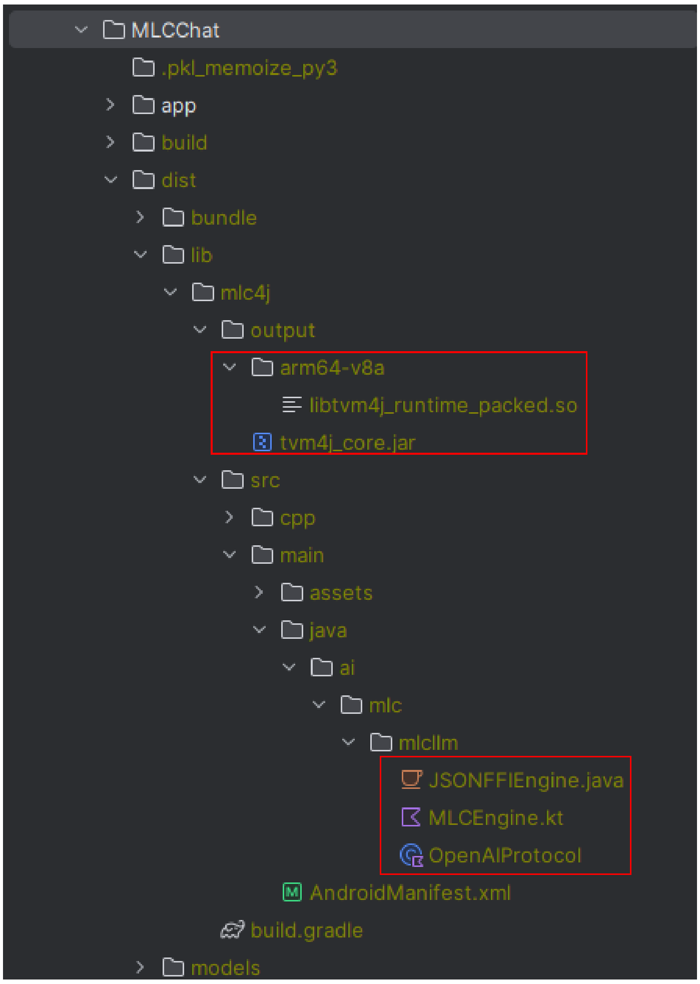
```
# 执行mlc_llm package命令行
mlc_llm package
```



MINGW64:/d/mlc-llm/android/MLCChat

```
y60044858@Y60044858-fdEBc MINGW64 /d/mlc-llm/android/MLCChat (main)
$ mlc_llm package
[2024-06-28 12:37:48] INFO package.py:327: MLC LLM HOME: "D:\mlc-llm"
[2024-06-28 12:37:48] INFO package.py:28: Clean up all directories under "dist\bundle"
[2024-06-28 12:37:49] INFO jit.py:43: MLC_JIT_POLICY = ON. Can be one of: ON, OFF, REDO, READONLY
[2024-06-28 12:37:49] INFO download_cache.py:227: Downloading model from HuggingFace: HF://mlc-ai/MiniCPM-2B-dpo-bf16-llama-format-q4f16_1
[2024-06-28 12:37:49] INFO download_cache.py:29: MLC_DOWNLOAD_CACHE_POLICY = ON. Can be one of: ON, OFF, REDO, READONLY
[2024-06-28 12:37:49] INFO download_cache.py:166: Weights already downloaded: C:\Users\y60044858\AppData\Local\mlc_llm\model_weights\hf\mlc-ai\MiniCPM-2B-dpo-bf16-llama-format-q4f16_
1
[2024-06-28 12:37:49] INFO package.py:81: Model lib is not specified for model "MiniCPM-2B-dpo-bf16-llama-format_q4f16_1". Now jit compile the model library.
[2024-06-28 12:37:49] INFO jit.py:158: Using cached model lib: C:\Users\y60044858\AppData\Local\mlc_llm\model_lib\e6d35b927d1ef93fb60ba8137d62cf84.tar
[2024-06-28 12:37:49] INFO package.py:129: Bundle weight for MiniCPM-2B-dpo-bf16-llama-format_q4f16_1, copy into dist\bundle\MiniCPM-2B-dpo-bf16-llama-format_q4f16_1
[2024-06-28 12:37:49] INFO package.py:154: Dump the app config below to "dist\bundle\mlc-app-config.json":
{
  "model_list": [
    {
      "model_id": "MiniCPM-2B-dpo-bf16-llama-format_q4f16_1",
      "model_lib": "llama_q4f16_1_938b473cd04bb62be838141ef7eb5bbc",
      "model_url": "https://huggingface.co/mlc-ai/MiniCPM-2B-dpo-bf16-llama-format-q4f16_1",
      "estimated_vram_bytes": 2036816936
    }
  ]
}
[2024-06-28 12:37:50] INFO package.py:211: Creating lib from ['C:\\Users\\y60044858\\AppData\\Local\\mlc_llm\\model_lib\\e6d35b927d1ef93fb60ba8137d62cf84.tar']
[2024-06-28 12:37:50] INFO package.py:212: Validating the library dist\lib\libmodel_android.a
[2024-06-28 12:37:50] INFO package.py:213: List of available model libs packaged: ['llama_q4f16_1_938b473cd04bb62be838141ef7eb5bbc'], if we have '-' in the model_lib string, it will
be turned into '_'
[2024-06-28 12:37:50] INFO package.py:256: Validation pass
[2024-06-28 12:37:50] INFO package.py:270: Moving "dist\lib\libmodel_android.a" to "build\lib\libmodel_android.a"
[2024-06-28 12:37:50] INFO package.py:274: Building mlc4j
info: component 'rust-std' for target 'aarch64-linux-android' is up to date
[2024-06-28 12:37:52] INFO prepare_libs.py:91: Entering "D:\mlc-llm\android\MLCChat\build" for MLC LLM and tvm4j build.
[2024-06-28 12:37:52] INFO prepare_libs.py:95: Set TVM_SOURCE_DIR to "D:\mlc-llm\3rdparty\tvm"
[2024-06-28 12:37:52] INFO prepare_libs.py:23: Running cmake
[2024-06-28 12:37:52] INFO prepare_libs.py:49: Using ninja in windows, make sure you installed ninja in conda
-- The C compiler identification is Clang 14.0.6
-- The CXX compiler identification is Clang 14.0.6
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: D:/D/Android/androidSDK/android-sdk_r24.4.1-windows/ndk/25.1.8937393/toolchains/llvm/prebuilt/windows-x86_64/bin/clang.exe - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: D:/D/Android/androidSDK/android-sdk_r24.4.1-windows/ndk/25.1.8937393/toolchains/llvm/prebuilt/windows-x86_64/bin/clang++.exe - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Hide private symbols
-- TVM_SOURCE_DIR: D:\mlc-llm\3rdparty\tvm
-- Hide private symbols...
-- Forbidding undefined symbols in shared library, using -Wl,--no-undefined on platform Android
-- Building for Android
-- Didn't find the path to CCACHE, disabling ccache
-- Performing Test SUPPORT_CXX17
-- Performing Test SUPPORT_CXX17 - Success
-- VTA build with VTA_HW_PATH=D:/mlc-llm/3rdparty/tvm/3rdparty/vta-hw
-- Build VTA runtime with target: sim
```

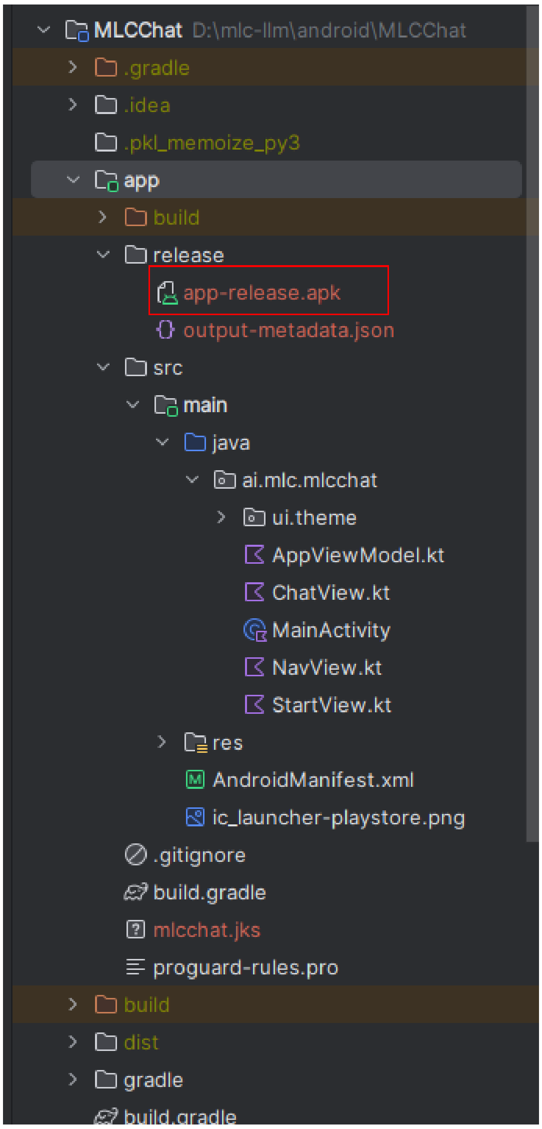
执行成功后，会生成以下/dist/lib/mlc4j目录下的文件。



```
MLCChat
  .pkl_memoize_py3
  app
  build
  dist
    bundle
    lib
      mlc4j
        output
          arm64-v8a
            libtvm4j_runtime_packed.so
          tvm4j_core.jar
        src
          cpp
          main
            assets
            java
              ai
                mlc
                  mlcllm
                    JSONFFIEngine.java
                    MLCEngine.kt
                    OpenAIProtocol
            AndroidManifest.xml
        build.gradle
    models
```

2.5 生成APK

点击“Build → Generate Signed Bundle / APK”。如果这是您第一次生成APK，则需要根据Android的官方指南创建一个密钥。此APK将放置在android/MLCChat/app/release/app-release.apk



```
MLCChat D:\mlc-llm\android\MLCChat
  .gradle
  .idea
  .pkl_memoize_py3
  app
    build
    release
      app-release.apk
      output-metadata.json
    src
      main
        java
          ai.mlc.mlcchat
            ui.theme
            AppViewModel.kt
            ChatView.kt
            MainActivity
            NavView.kt
            StartView.kt
        res
        AndroidManifest.xml
        ic_launcher-playstore.png
    .gitignore
    build.gradle
    mlcchat.jks
    proguard-rules.pro
  build
  dist
  gradle
  build.gradle
```

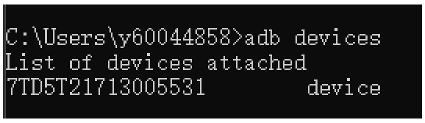
2.6 安装ADB和USB调试

将platform-tool添加到环境变量PATH

```
# adb命令环境变量配置
PATH=D:\D\Android\androidSDK\android-sdk_r24.4.1-windows\platform-tools
```

在手机设置中以开发人员模式启用“USB调试”。运行以下命令，如果正确安装 ADB，您的手机将显示为设备：

```
adb devices
```

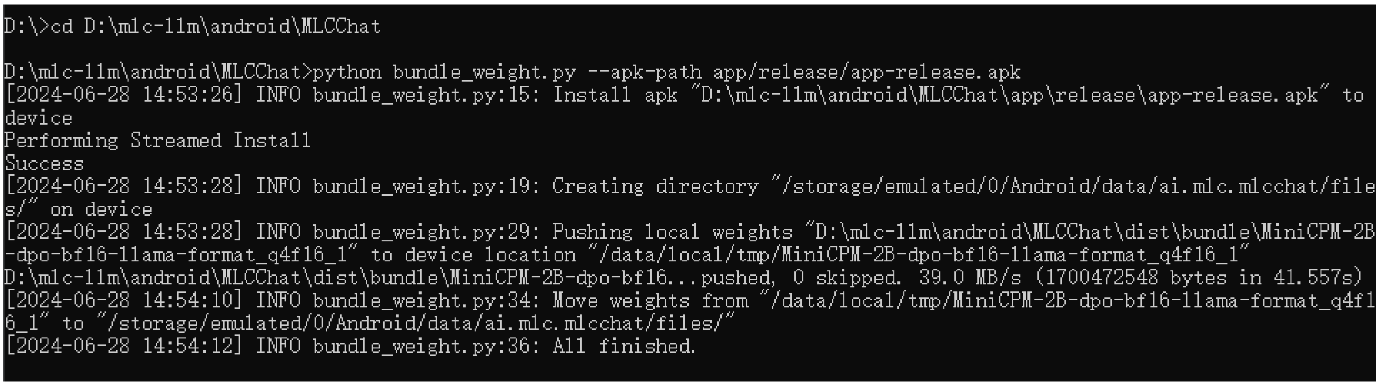


```
C:\Users\y60044858>adb devices
List of devices attached
7TD5T21713005531        device
```

2.7 将APK和权重安装到手机

打开CMD窗口，输入以下命令：

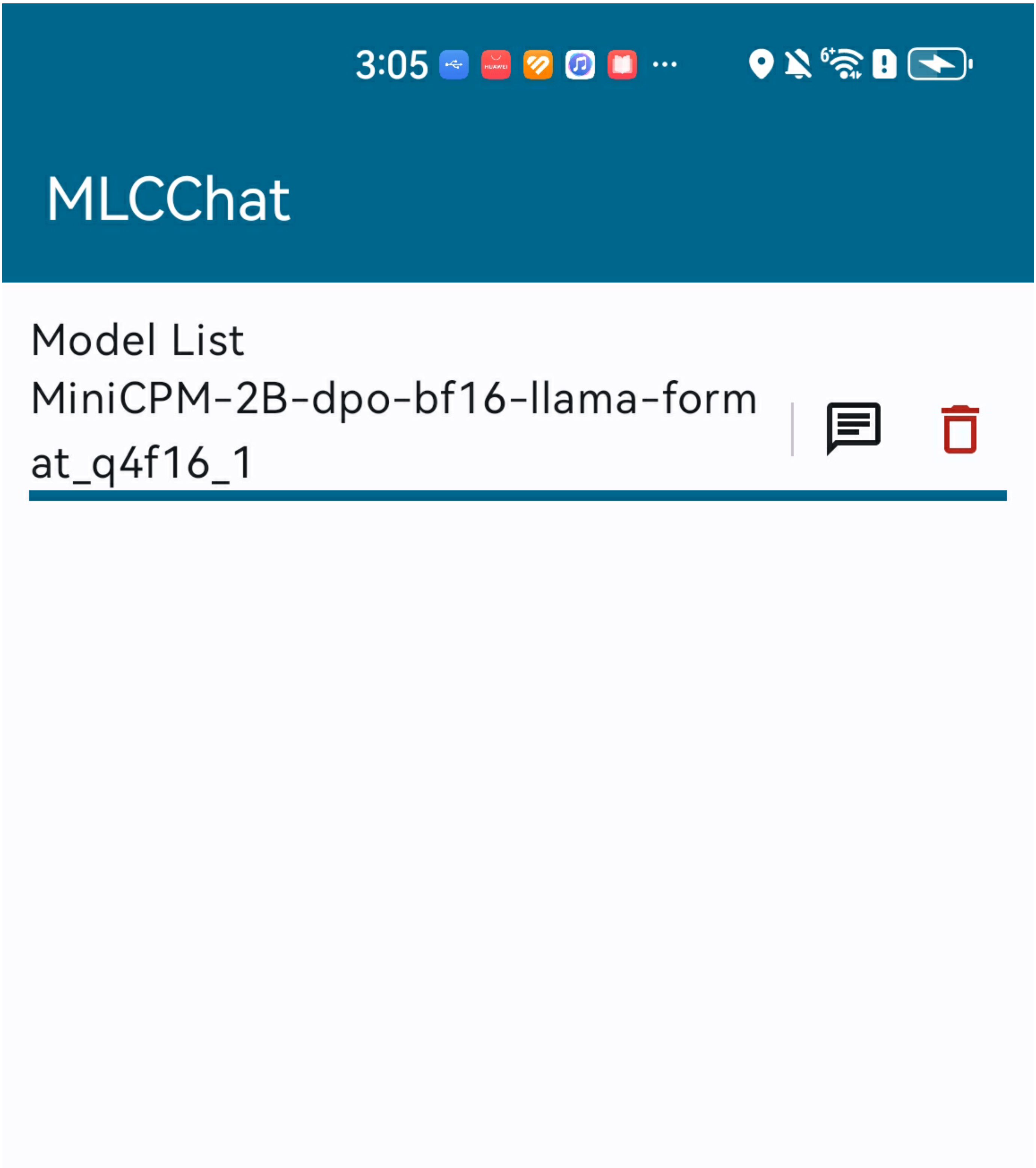
```
# 打开mlc—llm/android/MLCChat
cd D:\mlc-llm\android\MLCChat
python bundle_weight.py --apk-path app/release/app-release.apk
```

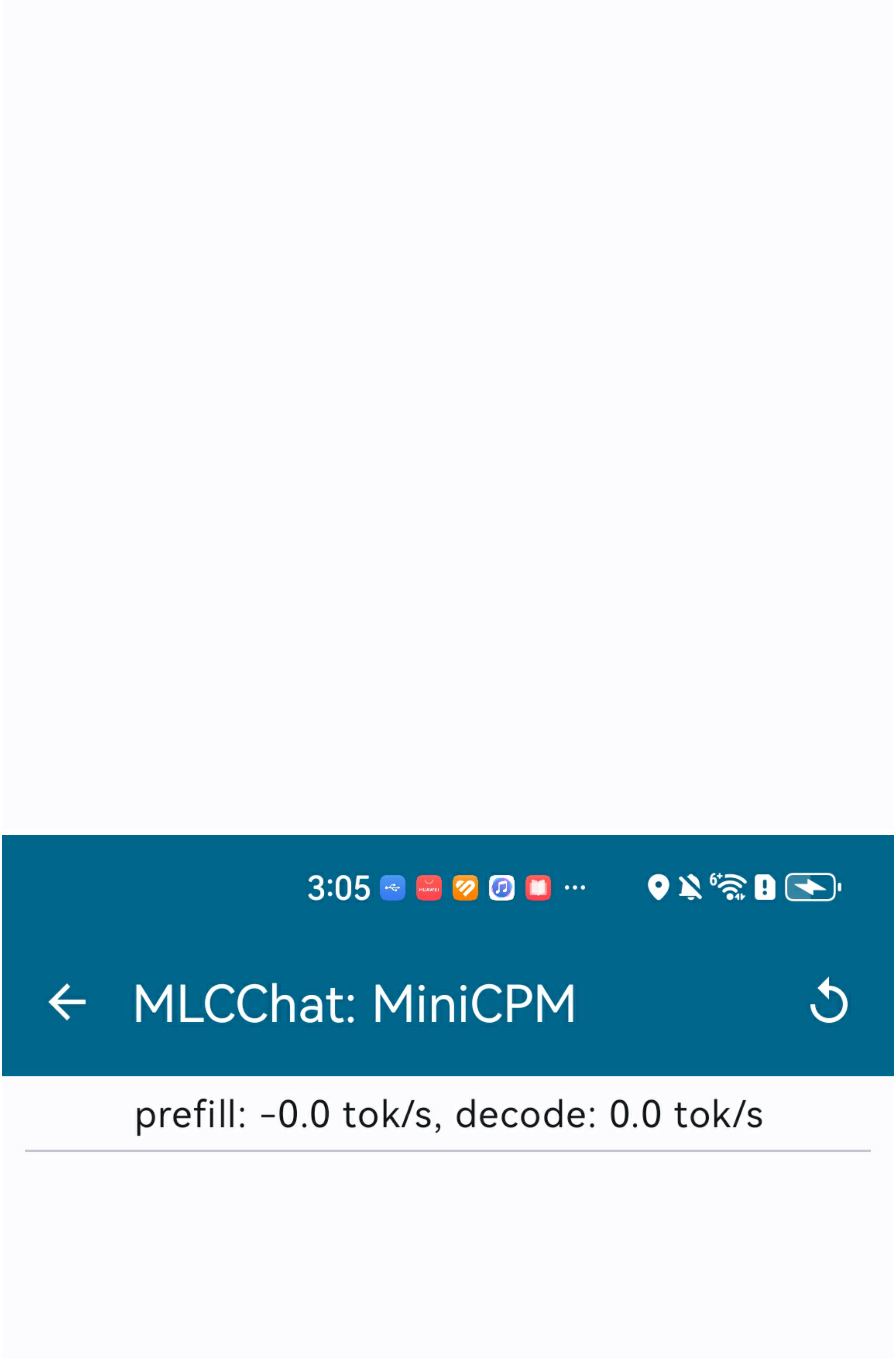


```
D:\>cd D:\mlc-llm\android\MLCChat

D:\mlc-llm\android\MLCChat>python bundle_weight.py --apk-path app/release/app-release.apk
[2024-06-28 14:53:26] INFO bundle_weight.py:15: Install apk "D:\mlc-llm\android\MLCChat\app\release\app-release.apk" to
device
Performing Streamed Install
Success
[2024-06-28 14:53:28] INFO bundle_weight.py:19: Creating directory "/storage/emulated/0/Android/data/ai.mlc.mlcchat/file
s/" on device
[2024-06-28 14:53:28] INFO bundle_weight.py:29: Pushing local weights "D:\mlc-llm\android\MLCChat\dist\bundle\MiniCPM-2B
-dpo-bf16-llama-format_q4f16_1" to device location "/data/local/tmp/MiniCPM-2B-dpo-bf16-llama-format_q4f16_1"
D:\mlc-llm\android\MLCChat\dist\bundle\MiniCPM-2B-dpo-bf16...pushed, 0 skipped. 39.0 MB/s (1700472548 bytes in 41.557s)
[2024-06-28 14:54:10] INFO bundle_weight.py:34: Move weights from "/data/local/tmp/MiniCPM-2B-dpo-bf16-llama-format_q4f1
6_1" to "/storage/emulated/0/Android/data/ai.mlc.mlcchat/files/"
[2024-06-28 14:54:12] INFO bundle_weight.py:36: All finished.
```

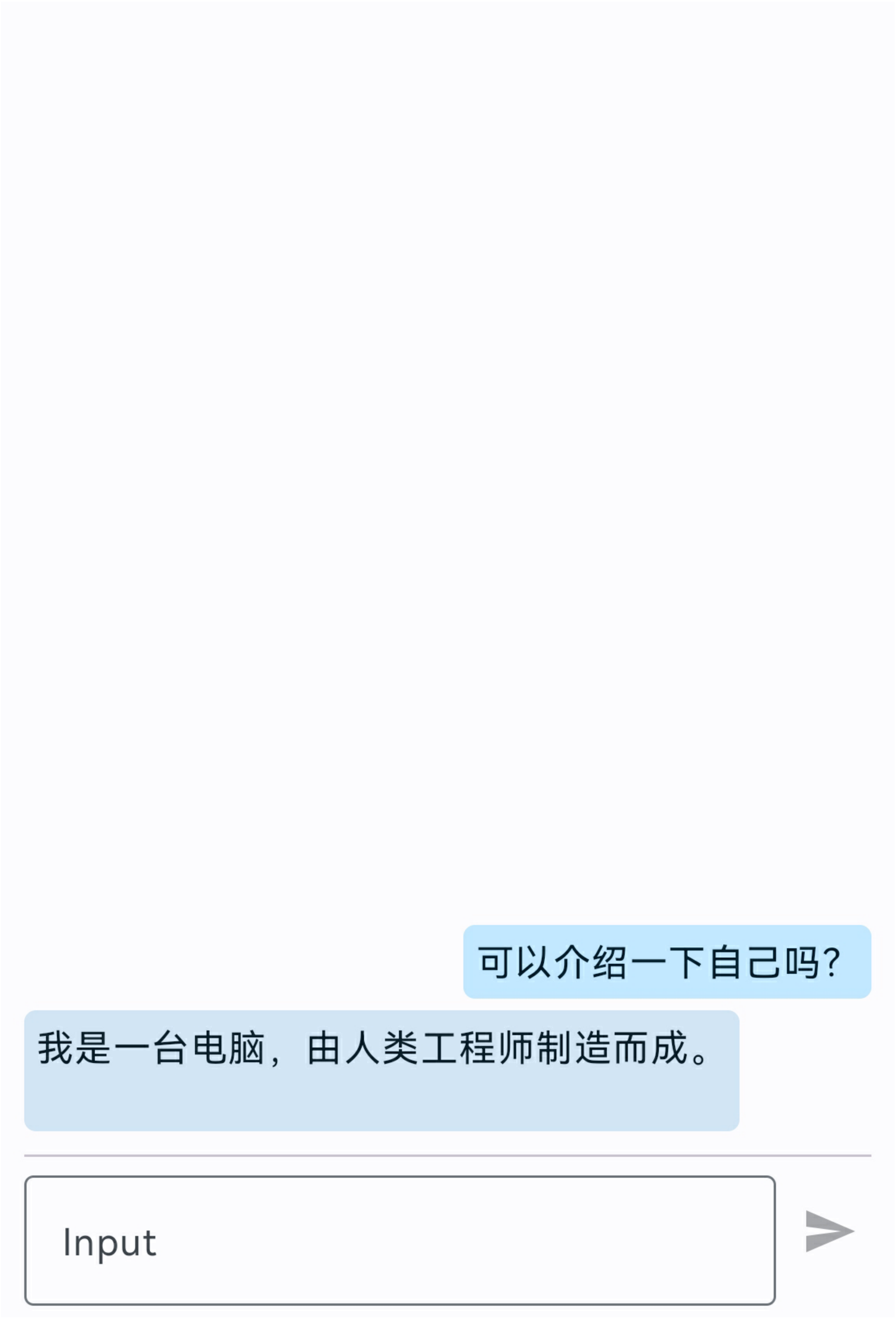
2.8 运行MLCChat应用

手机打开MLCCHat应用并运行。





prefill: -0.0 tok/s, decode: 0.0 tok/s



可以介绍一下自己吗?

我是一台电脑，由人类工程师制造而成。