

▼ Applying a git-patch to a `setup.py` file from a PyPI package source

▼ Special Note

The `setup.py` file had a problem of non-existing (invalid) `entry_points`. An issue [#95](#) was opened for the same and the fix was implemented through PR [#96](#) in [optimum's repository](#).

The conda-forge recipe earlier (in its making) had used a jinja2 workaround to circumvent the conflict in question. But since, the fix was already implemented, as suggested in the review the git patch for PR [#96](#) was used to mitigate the issue with the `setup.py` file in `v1.0.0` of `optimum` package, hosted on PyPI.

However, there was a problem: the `setup.py` file used in PR [#96](#) was different from that found in the package source of `v1.0.0` on PyPI.

As a workaround for this, the patch for PR [#96](#) (file: `assets/pr_96.patch`) was modified (file: `adapted_pr_96.patch`) to match the `setup.py` file for `v1.0.0`.

This fixed the build failure with the patch.

```
1 %%capture
2 ! sudo apt install tree
```

```
1 ! tree .
```

```
├── sample_data
│   ├── anscombe.json
│   ├── california_housing_test.csv
│   ├── california_housing_train.csv
│   ├── mnist_test.csv
│   ├── mnist_train_small.csv
│   └── README.md
```

```
1 directory, 6 files
```

▼ A. Create a Git Patch and Write to the Disk

```
1 %%writefile adapted_pr_96.patch
2
3 From 9dd33c51b86905ffd745d537a68d617cc320c252 Mon Sep 17 00:00:00 2001
4 From: Ella Charlaix <ella@huggingface.co>
5 Date: Mon, 14 Mar 2022 09:39:06 +0100
6 Subject: [PATCH] Remove unexisting entry points
7
8 ---
9  setup.py | 7 -
10 1 file changed, 7 deletions(-)
11
12 diff --git a/setup.py b/setup.py
13 index 103c1ec..9721fda 100644
14 --- a/setup.py
15 +++ b/setup.py
16 @@ -52,13 +52,6 @@
17     packages=find_namespace_packages(include=["optimum*"]),
18     install_requires=install_requires,
19     extras_require=extras,
20 -    entry_points={
21 -        "console_scripts": [
22 -            "optimum_export=optimum.onnxruntime.convert:main",
23 -            "optimum_optimize=optimum.onnxruntime.optimize_model:main",
24 -            "optimum_export_optimize=optimum.onnxruntime.convert_and_optimize:main",
25 -        ],
26 -    },
27     include_package_data=True,
28     zip_safe=False,
29 )
30
```

```
Writing adapted_pr_96.patch
```

```
1 ! tree .
```

```
.
├── adapted_pr_96.patch
└── sample_data
    ├── anscombe.json
    ├── california_housing_test.csv
    ├── california_housing_train.csv
    ├── mnist_test.csv
    ├── mnist_train_small.csv
    └── README.md
```

1 directory, 7 files

▼ B. Download and Rename the Git Patch

```
1 # Download Patch
2 ! curl -OL https://patch-diff.githubusercontent.com/raw/huggingface/optimum/pull/96.patch
```

% Total		% Received		% Xferd		Average Speed		Time	Time	Time	Current
						Dload	Upload	Total	Spent	Left	Speed
100	854	0	854	0	0	4692	0	--:--:--	--:--:--	--:--:--	4692

```
1 # Rename Patch
2 ! mv 96.patch pr_96.patch
```

```
1 ! tree .
```

```
.
├── adapted_pr_96.patch
├── pr_96.patch
└── sample_data
    ├── anscombe.json
    ├── california_housing_test.csv
    ├── california_housing_train.csv
    ├── mnist_test.csv
    ├── mnist_train_small.csv
    └── README.md
```

1 directory, 8 files

```
1 # Visualize Patch
2 ! cat pr_96.patch
```

```
From 9dd33c51b86905fffd745d537a68d617cc320c252 Mon Sep 17 00:00:00 2001
From: Ella Charlaix <ella@huggingface.co>
Date: Mon, 14 Mar 2022 09:39:06 +0100
Subject: [PATCH] Remove unexisting entry points

---
 setup.py | 7 -----
 1 file changed, 7 deletions(-)

diff --git a/setup.py b/setup.py
index 103clec..9721fda 100644
--- a/setup.py
+++ b/setup.py
@@ -65,13 +65,6 @@
     packages=find_namespace_packages(include=["optimum*"]),
     install_requires=REQUIRED_PKGS,
     extras_require=EXTRAS_REQUIRE,
-    entry_points={
-        "console_scripts": [
-            "optimum_export=optimum.onnxruntime.convert:main",
-            "optimum_optimize=optimum.onnxruntime.optimize_model:main",
-            "optimum_export_optimize=optimum.onnxruntime.convert_and_optimize:main",
-        ],
-    },
     include_package_data=True,
     zip_safe=False,
 )
```

▼ C. Download and Unzip the Source Package from PyPI

```
1 # Download the Package Source from PyPI
2 ! pip download "optimum==1.0.0" --no-deps --no-binary :all:
```

```
Collecting optimum==1.0.0
  Downloading optimum-1.0.0.tar.gz (46 kB)
    |██████████████████████████████████████| 46 kB 2.5 MB/s
Installing build dependencies ... done
```

```
Getting requirements to build wheel ... done
Preparing wheel metadata ... done
Saved ./optimum-1.0.0.tar.gz
Successfully downloaded optimum
```

```
1 # Unzip and Extract the Package Source
2 ! tar -xvf optimum-*.tar.gz
```

```
optimum-1.0.0/
optimum-1.0.0/LICENSE
optimum-1.0.0/MANIFEST.in
optimum-1.0.0/PKG-INFO
optimum-1.0.0/README.md
optimum-1.0.0/optimum/
optimum-1.0.0/optimum/configuration_utils.py
optimum-1.0.0/optimum/intel/
optimum-1.0.0/optimum/intel/__init__.py
optimum-1.0.0/optimum/intel/neural_compressor/
optimum-1.0.0/optimum/intel/neural_compressor/__init__.py
optimum-1.0.0/optimum/intel/neural_compressor/config.py
optimum-1.0.0/optimum/intel/neural_compressor/optimizer.py
optimum-1.0.0/optimum/intel/neural_compressor/pruning.py
optimum-1.0.0/optimum/intel/neural_compressor/quantization.py
optimum-1.0.0/optimum/intel/neural_compressor/trainer_inc.py
optimum-1.0.0/optimum/intel/neural_compressor/trainer_inc_test.py
optimum-1.0.0/optimum/intel/neural_compressor/utils.py
optimum-1.0.0/optimum/onnxruntime/
optimum-1.0.0/optimum/onnxruntime/__init__.py
optimum-1.0.0/optimum/onnxruntime/compare_two_onnx_model.py
optimum-1.0.0/optimum/onnxruntime/configuration.py
optimum-1.0.0/optimum/onnxruntime/model.py
optimum-1.0.0/optimum/onnxruntime/old_scripts/
optimum-1.0.0/optimum/onnxruntime/old_scripts/convert.py
optimum-1.0.0/optimum/onnxruntime/old_scripts/convert_and_optimize.py
optimum-1.0.0/optimum/onnxruntime/old_scripts/optimize_model.py
optimum-1.0.0/optimum/onnxruntime/optimization.py
optimum-1.0.0/optimum/onnxruntime/quantization.py
optimum-1.0.0/optimum/onnxruntime/quantization_old.py
optimum-1.0.0/optimum/onnxruntime/quantization_with_opti.py
optimum-1.0.0/optimum/onnxruntime/utils.py
optimum-1.0.0/optimum/utils/
optimum-1.0.0/optimum/utils/__init__.py
optimum-1.0.0/optimum/utils/logging.py
optimum-1.0.0/optimum/version.py
optimum-1.0.0/optimum.egg-info/
optimum-1.0.0/optimum.egg-info/PKG-INFO
optimum-1.0.0/optimum.egg-info/SOURCES.txt
optimum-1.0.0/optimum.egg-info/dependency_links.txt
optimum-1.0.0/optimum.egg-info/entry_points.txt
optimum-1.0.0/optimum.egg-info/not-zip-safe
optimum-1.0.0/optimum.egg-info/requires.txt
optimum-1.0.0/optimum.egg-info/top_level.txt
optimum-1.0.0/pyproject.toml
optimum-1.0.0/setup.cfg
optimum-1.0.0/setup.py
```

```
1 ! cp optimum-*/setup.py setup.py.bkp
```

```
1 ! tree .
```

```
├── LICENSE
├── MANIFEST.in
├── optimum
│   ├── configuration_utils.py
│   ├── intel
│   │   ├── __init__.py
│   │   └── neural_compressor
│   │       ├── config.py
│   │       ├── __init__.py
│   │       ├── optimizer.py
│   │       ├── pruning.py
│   │       ├── quantization.py
│   │       ├── trainer_inc.py
│   │       ├── trainer_inc_test.py
│   │       └── utils.py
│   └── onnxruntime
│       ├── compare_two_onnx_model.py
│       ├── configuration.py
│       ├── __init__.py
│       ├── model.py
│       ├── old_scripts
│       │   ├── convert_and_optimize.py
│       │   └── convert.py
│       ├── optimize_model.py
│       ├── optimization.py
│       ├── quantization_old.py
│       └── quantization.py
```

```
├── quantization_with_opti.py
├── utils.py
├── utils
├──   ├── __init__.py
├──   └── logging.py
├── version.py
├── optimum.egg-info
├──   ├── dependency_links.txt
├──   ├── entry_points.txt
├──   ├── not-zip-safe
├──   ├── PKG-INFO
├──   ├── requires.txt
├──   ├── SOURCES.txt
├──   └── top_level.txt
├── PKG-INFO
├── pyproject.toml
├── README.md
├── setup.cfg
├── setup.py
├── optimum-1.0.0.tar.gz
├── pr_96.patch
├── sample_data
├──   ├── anscombe.json
├──   ├── california_housing_test.csv
├──   ├── california_housing_train.csv
├──   ├── mnist_test.csv
├──   ├── mnist_train_small.csv
├──   └── README.md
└── setup.py.bkp
```

9 directories, 49 files

▼ D. Apply git patch

```
1 # Check the path of the patch command
2 ! which patch
```

/usr/bin/patch

```
1 # Apply dry-run
2 ! patch "optimum-1.0.0/setup.py" --no-backup-if-mismatch --batch -Np1 -i "adapted_pr_96.patch" --binary --dry-run
```

checking file optimum-1.0.0/setup.py
patch unexpectedly ends in middle of line
Hunk #1 succeeded at 52 with fuzz 1.

```
1 # Create modified version: mod_setup.py
2 ! patch "optimum-1.0.0/setup.py" --no-backup-if-mismatch --batch -Np1 -i "adapted_pr_96.patch" --binary -o mod_setup.py
```

patching file mod_setup.py (read from optimum-1.0.0/setup.py)
patch unexpectedly ends in middle of line
Hunk #1 succeeded at 52 with fuzz 1.

```
1 ! tree .
```

```
├── LICENSE
├── MANIFEST.in
├── optimum
├──   ├── configuration_utils.py
├──   ├── intel
├──   ├──   ├── __init__.py
├──   ├──   └── neural_compressor
├──   ├──       ├── config.py
├──   ├──       ├── __init__.py
├──   ├──       ├── optimizer.py
├──   ├──       ├── pruning.py
├──   ├──       ├── quantization.py
├──   ├──       ├── trainer_inc.py
├──   ├──       ├── trainer_inc_test.py
├──   ├──       └── utils.py
├──   ├── onnxruntime
├──   ├──   ├── compare_two_onnx_model.py
├──   ├──   ├── configuration.py
├──   ├──   ├── __init__.py
├──   ├──   ├── model.py
├──   ├──   ├── old_scripts
├──   ├──   ├──   ├── convert_and_optimize.py
├──   ├──   ├──   ├── convert.py
├──   ├──   ├──   └── optimize_model.py
├──   ├──   ├── optimization.py
├──   ├──   ├── quantization_old.py
├──   ├──   ├── quantization.py
├──   ├──   ├── quantization_with_opti.py
├──   └──   └── utils.py
```

```
├── datasets.py
│   ├── utils
│   │   ├── __init__.py
│   │   └── logging.py
│   └── version.py
├── optimum.egg-info
│   ├── dependency_links.txt
│   ├── entry_points.txt
│   ├── not-zip-safe
│   ├── PKG-INFO
│   ├── requires.txt
│   ├── SOURCES.txt
│   └── top_level.txt
├── PKG-INFO
├── pyproject.toml
├── README.md
├── setup.cfg
├── setup.py
├── optimum-1.0.0.tar.gz
├── pr_96.patch
├── sample_data
│   ├── anscombe.json
│   ├── california_housing_test.csv
│   ├── california_housing_train.csv
│   ├── mnist_test.csv
│   ├── mnist_train_small.csv
│   └── README.md
└── setup.py.bkp
```

9 directories, 50 files

▼ E. Visualize Original and Modified Files

▼ E.1. Show the modified file contents

```
1 ! cat mod_setup.py
```

```
import re

from setuptools import find_namespace_packages, setup

# Ensure we match the version set in src/optimum/version.py
try:
    filepath = "optimum/version.py"
    with open(filepath) as version_file:
        (__version__,) = re.findall('__version__ = "(.*)"', version_file.read())
except Exception as error:
    assert False, "Error: Could not open '%s' due %s\n" % (filepath, error)

install_requires = [
    "coloredlogs",
    "sympy",
    "transformers>=4.15.0",
    "torch>=1.9",
]

extras = {
    "onnxruntime": ["onnx", "onnxruntime", "datasets>=1.2.1"],
    "intel": ["pycocotools", "neural_compressor>=1.9", "datasets>=1.2.1", "pandas<1.4.0"],
    "graphcore": "optimum-graphcore",
}

setup(
    name="optimum",
    version=__version__,
    description="Optimum Library is an extension of the Hugging Face Transformers library, providing a fra
    \"integrate third-party libraries from Hardware Partners and interface with their specific \"
    \"functionality.\",
    long_description=open("README.md", "r", encoding="utf-8").read(),
    long_description_content_type="text/markdown",
    classifiers=[
        "Development Status :: 3 - Alpha",
        "License :: OSI Approved :: Apache Software License",
        "Intended Audience :: Developers",
        "Intended Audience :: Education",
        "Intended Audience :: Science/Research",
        "Operating System :: OS Independent",
        "Programming Language :: Python :: 3.8",
        "Programming Language :: Python :: 3.9",
        "Topic :: Scientific/Engineering :: Artificial Intelligence",
    ],
    keywords="transformers, quantization, pruning, training, ipu",
```

```
url="https://huggingface.co/hardware",
author="HuggingFace Inc. Special Ops Team",
author_email="hardware@huggingface.co",
license="Apache",
packages=find_namespace_packages(include=["optimum*"]),
install_requires=install_requires,
extras_require=extras,
include_package_data=True,
zip_safe=False,
```

▼ E.2. Show the original file contents

```
1 ! cat "optimum-1.0.0/setup.py"
```

```
#!/usr/bin/env python
# coding=utf-8
# Copyright 2022 HuggingFace Inc.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

import re

filepath = "optimum/version.py"
with open(filepath) as version_file:
    (__version__,) = re.findall('__version__ = "(.*)"', version_file.read())

except Exception as error:
    assert False, "Error: Could not open '%s' due %s\n" % (filepath, error)

install_requires = [
    "coloredlogs",
    "sympy",
    "transformers>=4.15.0",
    "torch>=1.9",
]

extras = {
    "onnxruntime": ["onnx", "onnxruntime", "datasets>=1.2.1"],
    "intel": ["pycocotools", "neural_compressor>=1.9", "datasets>=1.2.1", "pandas<1.4.0"],
    "graphcore": "optimum-graphcore",
}

setup(
    name="optimum",
    version=__version__,
    description="Optimum Library is an extension of the Hugging Face Transformers library, providing a fra"
    "integrate third-party libraries from Hardware Partners and interface with their specific "
    "functionality.",
    long_description=open("README.md", "r", encoding="utf-8").read(),
    long_description_content_type="text/markdown",
    classifiers=[
        "Development Status :: 3 - Alpha",
        "License :: OSI Approved :: Apache Software License",
        "Intended Audience :: Developers",
        "Intended Audience :: Education",
        "Intended Audience :: Science/Research",
        "Operating System :: OS Independent",
        "Programming Language :: Python :: 3.8",
        "Programming Language :: Python :: 3.9",
        "Topic :: Scientific/Engineering :: Artificial Intelligence",
    ],
    keywords="transformers, quantization, pruning, training, ipu",
    url="https://huggingface.co/hardware",
    author="HuggingFace Inc. Special Ops Team",
    author_email="hardware@huggingface.co",
    license="Apache",
    packages=find_namespace_packages(include=["optimum*"]),
    install_requires=install_requires,
    extras_require=extras,
    entry_points={
        "console_scripts": [
            "optimum_export=optimum.onnxruntime.convert:main",
            "optimum_optimize=optimum.onnxruntime.optimize_model:main",
            "optimum_export_optimize=optimum.onnxruntime.convert_and_optimize:main",
        ],
    },
    include_package_data=True,
    zip_safe=False,
)
```

▼ F. Validate the difference between original and modified files

```
1 ! git diff "optimum-1.0.0/setup.py" mod_setup.py
```

```
diff --git a/optimum-1.0.0/setup.py b/mod_setup.py
index 6660a3a..33141fd 100644
--- a/optimum-1.0.0/setup.py
+++ b/mod_setup.py
@@ -52,13 +52,6 @@ setup(
     packages=find_namespace_packages(include=["optimum*"]),
```

```
install_requires=install_requires,
extras_require=extras,
- entry_points={
-     "console_scripts": [
-         "optimum_export=optimum.onnxruntime.convert:main",
-         "optimum_optimize=optimum.onnxruntime.optimize_model:main",
-         "optimum_export_optimize=optimum.onnxruntime.convert_and_optimize:main",
-     ],
- },
- include_package_data=True,
- zip_safe=False,
)
```

▼ G. Lookup the help for `patch` command

```
1 ! patch --help
```

```
-e --ed      Interpret the patch as an ed script.
-n --normal  Interpret the patch as a normal difference.
-u --unified  Interpret the patch as a unified difference.

-N --forward Ignore patches that appear to be reversed or already applied.
-R --reverse  Assume patches were created with old and new files swapped.

-i PATCHFILE --input=PATCHFILE  Read patch from PATCHFILE instead of stdin.
```

Output options:

```
-o FILE  --output=FILE  Output patched files to FILE.
-r FILE  --reject-file=FILE  Output rejects to FILE.

-D NAME  --ifdef=NAME  Make merged if-then-else output using NAME.
--merge  Merge using conflict markers instead of creating reject files.
-E --remove-empty-files  Remove output files that are empty after patching.

-Z --set-utc  Set times of patched files, assuming diff uses UTC (GMT).
-T --set-time  Likewise, assuming local time.

--quoting-style=WORD  output file names using quoting style WORD.
Valid WORDs are: literal, shell, shell-always, c, escape.
Default is taken from QUOTING_STYLE env variable, or 'shell' if unset.
```

Backup and version control options:

```
-b --backup  Back up the original contents of each file.
--backup-if-mismatch  Back up if the patch does not match exactly.
--no-backup-if-mismatch  Back up mismatches only if otherwise requested.

-V STYLE  --version-control=STYLE  Use STYLE version control.
        STYLE is either 'simple', 'numbered', or 'existing'.
-B PREFIX --prefix=PREFIX  Prepend PREFIX to backup file names.
-Y PREFIX --basename-prefix=PREFIX  Prepend PREFIX to backup file basenames.
-z SUFFIX  --suffix=SUFFIX  Append SUFFIX to backup file names.

-g NUM  --get=NUM  Get files from RCS etc. if positive; ask if negative.
```

Miscellaneous options:

```
-t --batch  Ask no questions; skip bad-Prereq patches; assume reversed.
-f --force  Like -t, but ignore bad-Prereq patches, and assume unreversed.
-s --quiet  --silent  Work silently unless an error occurs.
--verbose  Output extra information about the work being done.
--dry-run  Do not actually change any files; just print what would happen.
--posix    Conform to the POSIX standard.

-d DIR  --directory=DIR  Change the working directory to DIR first.
--reject-format=FORMAT  Create 'context' or 'unified' rejects.
--binary  Read and write data in binary mode.
--read-only=BEHAVIOR  How to handle read-only input files: 'ignore' that they
                    are read-only, 'warn' (default), or 'fail'.

-v --version  Output version info.
--help  Output this help.
```

Report bugs to <bug-patch@gnu.org>.

```
1
```

✓ 0s completed at 8:53 PM

✕