t81_558_class_11_01_huggingface

June 23, 2025

1

2 T81-558: Applications of Deep Neural Networks

Module 11: Natural Language Processing with Hugging Face * Instructor: Jeff Heaton, McKelvey School of Engineering, Washington University in St. Louis * For more information visit the class website.

3 Module 11 Material

- **Part 11.1: Introduction to Hugging Face** [Video] [Notebook]
- Part 11.2: Hugging Face Tokenizers [Video] [Notebook]
- Part 11.3: Hugging Face Datasets [Video] [Notebook]
- Part 11.4: Training Hugging Face Models [Video] [Notebook]
- Part 11.5: What are Embedding Layers in Keras [Video] [Notebook]

4 Google CoLab Instructions

The following code ensures that Google CoLab is running the correct version of TensorFlow.

[1]:
```
try:
    %tensorflow_version 2.x
    COLAB = True
    print("Note: using Google CoLab")
except:
    print("Note: not using Google CoLab")
    COLAB = False
```

```
Note: not using Google CoLab
```

5 Part 11.1: Introduction to Hugging Face

Transformers have become a mainstay of natural language processing. This module will examine the Hugging Face Python library for natural language processing, bringing together pretrained transformers, data sets, tokenizers, and other elements. Through the Hugging Face API, you can quickly begin using sentiment analysis, entity recognition, language translation, summarization, and text generation.

Colab does not install Hugging face by default. Whether installing Hugging Face directly into a local computer or utilizing it through Colab, the following commands will install the library.

[2]:
```
# HIDE OUTPUT
import tensorflow as tf
print(f"Tensor Flow Version: {tf.__version__}")

!pip install transformers
!pip install transformers[sentencepiece]
```

```
2024-02-14 20:42:53.807584: I tensorflow/core/util/port.cc:113] oneDNN custom
operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them
off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-02-14 20:42:53.847346: E
external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable to register
cuDNN factory: Attempting to register factory for plugin cuDNN when one has
already been registered
2024-02-14 20:42:53.847373: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to register
cuFFT factory: Attempting to register factory for plugin cuFFT when one has
already been registered
2024-02-14 20:42:53.848137: E
external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable to
register cuBLAS factory: Attempting to register factory for plugin cuBLAS when
one has already been registered
2024-02-14 20:42:53.853359: I tensorflow/core/platform/cpu_feature_guard.cc:182]
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other
operations, rebuild TensorFlow with the appropriate compiler flags.
2024-02-14 20:42:54.663206: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not
find TensorRT

Tensor Flow Version: 2.15.0
Defaulting to user installation because normal site-packages is not writeable
Collecting transformers
  Downloading transformers-4.37.2-py3-none-any.whl (8.4 MB)
                             8.4/8.4 MB
3.7 MB/s eta 0:00:0000:0100:01
Collecting tokenizers<0.19,>=0.14
  Downloading
tokenizers-0.15.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(3.6 MB)
                             3.6/3.6 MB
5.6 MB/s eta 0:00:0000:0100:01
Requirement already satisfied: packaging>=20.0 in
/home/tmeng12/.local/lib/python3.10/site-packages (from transformers) (23.2)
```

```
Collecting regex!=2019.12.17
  Downloading
regex-2023.12.25-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (773
kB)
                           774.0/774.0
KB 8.4 MB/s eta 0:00:00a 0:00:01
Requirement already satisfied: pyyaml>=5.1 in /usr/lib/python3/dist-
packages (from transformers) (5.4.1)
Collecting tqdm>=4.27
  Downloading tqdm-4.66.2-py3-none-any.whl (78 kB)
                           78.3/78.3 KB
5.8 MB/s eta 0:00:00
Requirement already satisfied: filelock in
/home/tmeng12/.local/lib/python3.10/site-packages (from transformers) (3.12.4)
Collecting huggingface-hub<1.0,>=0.19.3
  Downloading huggingface_hub-0.20.3-py3-none-any.whl (330 kB)
                           330.1/330.1
KB 9.0 MB/s eta 0:00:0000:01
Requirement already satisfied: requests in
/home/tmeng12/.local/lib/python3.10/site-packages (from transformers) (2.31.0)
Collecting safetensors>=0.4.1
  Downloading
safetensors-0.4.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(1.3 MB)
                           1.3/1.3 MB
9.4 MB/s eta 0:00:00ta 0:00:01
Requirement already satisfied: numpy>=1.17 in
/home/tmeng12/.local/lib/python3.10/site-packages (from transformers) (1.26.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/home/tmeng12/.local/lib/python3.10/site-packages (from huggingface-
hub<1.0,>=0.19.3->transformers) (4.8.0)
Requirement already satisfied: fsspec>=2023.5.0 in
/home/tmeng12/.local/lib/python3.10/site-packages (from huggingface-
hub<1.0,>=0.19.3->transformers) (2023.9.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/lib/python3/dist-packages
(from requests->transformers) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/lib/python3/dist-
packages (from requests->transformers) (2020.6.20)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/lib/python3/dist-
packages (from requests->transformers) (1.26.5)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/tmeng12/.local/lib/python3.10/site-packages (from requests->transformers)
(3.3.0)
Installing collected packages: tqdm, safetensors, regex, huggingface-hub,
tokenizers, transformers
Successfully installed huggingface-hub-0.20.3 regex-2023.12.25 safetensors-0.4.2
```

```
tokenizers-0.15.2 tqdm-4.66.2 transformers-4.37.2
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: transformers[sentencepiece] in
/home/tmeng12/.local/lib/python3.10/site-packages (4.37.2)
Requirement already satisfied: safetensors>=0.4.1 in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (0.4.2)
Requirement already satisfied: packaging>=20.0 in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (23.2)
Requirement already satisfied: tqdm>=4.27 in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (4.66.2)
Requirement already satisfied: tokenizers<0.19,>=0.14 in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (0.15.2)
Requirement already satisfied: numpy>=1.17 in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (1.26.3)
Requirement already satisfied: pyyaml>=5.1 in /usr/lib/python3/dist-packages
(from transformers[sentencepiece]) (5.4.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.19.3 in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (0.20.3)
Requirement already satisfied: requests in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (2.31.0)
Requirement already satisfied: regex!=2019.12.17 in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (2023.12.25)
Requirement already satisfied: filelock in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (3.12.4)
Requirement already satisfied: protobuf in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (4.23.4)
Collecting sentencepiece!=0.1.92,>=0.1.91
  Downloading
sentencepiece-0.1.99-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(1.3 MB)
                           1.3/1.3 MB
3.0 MB/s eta 0:00:0000:0100:010m
Requirement already satisfied: fsspec>=2023.5.0 in
/home/tmeng12/.local/lib/python3.10/site-packages (from huggingface-
hub<1.0,>=0.19.3->transformers[sentencepiece]) (2023.9.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/home/tmeng12/.local/lib/python3.10/site-packages (from huggingface-
hub<1.0,>=0.19.3->transformers[sentencepiece]) (4.8.0)
```

```
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/tmeng12/.local/lib/python3.10/site-packages (from
requests->transformers[sentencepiece]) (3.3.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/lib/python3/dist-packages
(from requests->transformers[sentencepiece]) (3.3)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/lib/python3/dist-
packages (from requests->transformers[sentencepiece]) (1.26.5)
Requirement already satisfied: certifi>=2017.4.17 in /usr/lib/python3/dist-
packages (from requests->transformers[sentencepiece]) (2020.6.20)
Installing collected packages: sentencepiece
Successfully installed sentencepiece-0.1.99
```

Now that we have Hugging Face installed, the following sections will demonstrate how to apply Hugging Face to a variety of everyday tasks. After this introduction, the remainder of this module will take a deeper look at several specific NLP tasks applied to Hugging Face.

5.1 Sentiment Analysis

Sentiment analysis uses natural language processing, text analysis, computational linguistics, and biometrics to identify the tone of written text. Passages of written text can be into simple binary states of positive or negative tone. More advanced sentiment analysis might classify text into additional categories: sadness, joy, love, anger, fear, or surprise.

To demonstrate sentiment analysis, we begin by loading sample text, Shakespeare's 18th sonnet, a famous poem.

[3]:
```
from urllib.request import urlopen

# Read sample text, a poem
URL = "https://data.heatonresearch.com/data/t81-558/"\
    "datasets/sonnet_18.txt"
f = urlopen(URL)
text = f.read().decode("utf-8")
print(text)
```

```
Sonnet 18 original text
William Shakespeare

Shall I compare thee to a summer's day?
Thou art more lovely and more temperate:
Rough winds do shake the darling buds of May,
And summer's lease hath all too short a date:
Sometime too hot the eye of heaven shines,
And often is his gold complexion dimm'd;
And every fair from fair sometime declines,
By chance or nature's changing course untrimm'd;
But thy eternal summer shall not fade
Nor lose possession of that fair thou owest;
Nor shall Death brag thou wander'st in his shade,
```

```
When in eternal lines to time thou growest:
So long as men can breathe or eyes can see,
So long lives this and this gives life to thee.
```

Usually, you have to preprocess text into embeddings or other vector forms before presentation to a neural network. Hugging Face provides a pipeline that simplifies this process greatly. The pipeline allows you to pass regular Python strings to the transformers and return standard Python values.

We begin by loading a text-classification model. We do not specify the exact model type wanted, so Hugging Face automatically chooses a network from the Hugging Face hub named:

- distilbert-base-uncased-finetuned-sst-2-english

To specify the model to use, pass the model parameter, such as:

```
pipe = pipeline(model="roberta-large-mnli")
```

The following code loads a model pipeline and a model for sentiment analysis.

[4]:
```
# HIDE OUTPUT
import pandas as pd
from transformers import pipeline

classifier = pipeline("text-classification")
```

```
No model was supplied, defaulted to distilbert-base-uncased-finetuned-
sst-2-english and revision af0f99b (https://huggingface.co/distilbert-base-
uncased-finetuned-sst-2-english).
Using a pipeline without specifying a model name and revision in production is
not recommended.
```

We can now display the sentiment analysis results with a Pandas dataframe.

[ ]:
```
outputs = classifier(text)
df=pd.DataFrame(outputs)
df
```

[ ]:
```
      label     score
0  POSITIVE  0.984666
```

As you can see, the poem was considered 0.98 positive.

5.2 Entity Tagging

Entity tagging is the process that takes source text and finds parts of that text that represent entities, such as one of the following:

- Location (LOC)
- Organizations (ORG)
- Person (PER)
- Miscellaneous (MISC)

The following code requests a "named entity recognizer" (ner) and processes the specified text.

[6]:
```
# HIDE OUTPUT
text2 = "Abraham Lincoln was a president who lived in the United States."

tagger = pipeline("ner", aggregation_strategy="simple")
```

```
No model was supplied, defaulted to dbmdz/bert-large-cased-finetuned-
conll03-english and revision f2482bf (https://huggingface.co/dbmdz/bert-large-
cased-finetuned-conll03-english).
Using a pipeline without specifying a model name and revision in production is
not recommended.

model.safetensors:   0%|          | 0.00/1.33G [00:00<?, ?B/s]
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
/media/data/github/learn/ai/jheaton/t81_558_deep_learning/
 ↪t81_558_class_11_01_huggingface.ipynb Cell 15 line 4

     <a href='vscode-notebook-cell:/media/data/github/learn/ai/jheaton/
 ↪t81_558_deep_learning/t81_558_class_11_01_huggingface.ipynb#X20sZmlsZQ%3D%3D?
 ↪line=0'>1</a> # HIDE OUTPUT
     <a href='vscode-notebook-cell:/media/data/github/learn/ai/jheaton/
 ↪t81_558_deep_learning/t81_558_class_11_01_huggingface.ipynb#X20sZmlsZQ%3D%3D?
 ↪line=1'>2</a> text2 = "Abraham Lincoln was a president who lived in the Unite␣
 ↪States."
----> <a href='vscode-notebook-cell:/media/data/github/learn/ai/jheaton/
 ↪t81_558_deep_learning/t81_558_class_11_01_huggingface.ipynb#X20sZmlsZQ%3D%3D?
 ↪line=3'>4</a> tagger = pipeline("ner", aggregation_strategy="simple")

File ~/.local/lib/python3.10/site-packages/transformers/pipelines/__init__.py:
 ↪870, in pipeline(task, model, config, tokenizer, feature_extractor,␣
 ↪image_processor, framework, revision, use_fast, token, device, device_map,␣
 ↪torch_dtype, trust_remote_code, model_kwargs, pipeline_class, **kwargs)
    868 if isinstance(model, str) or framework is None:
    869     model_classes = {"tf": targeted_task["tf"], "pt":␣
 ↪targeted_task["pt"]}
--> 870     framework, model = infer_framework_load_model(
    871         model,
    872         model_classes=model_classes,
    873         config=config,
    874         framework=framework,
    875         task=task,
    876         **hub_kwargs,
    877         **model_kwargs,
    878     )
    880 model_config = model.config
    881 hub_kwargs["_commit_hash"] = model.config._commit_hash

File ~/.local/lib/python3.10/site-packages/transformers/pipelines/base.py:278,␣
 ↪in infer_framework_load_model(model, config, model_classes, task, framework,␣
 ↪**model_kwargs)
```

```
    272     logger.warning(
    273         "Model might be a PyTorch model (ending with `.bin`) but PyTorch␣
 ↪is not available. "
    274         "Trying to load the model with Tensorflow."
    275     )
    277 try:
--> 278     model = model_class.from_pretrained(model, **kwargs)
    279     if hasattr(model, "eval"):
    280         model = model.eval()

File ~/.local/lib/python3.10/site-packages/transformers/models/auto/auto_factory.
 ↪py:566, in _BaseAutoModelClass.from_pretrained(cls,␣
 ↪pretrained_model_name_or_path, *model_args, **kwargs)
    564 elif type(config) in cls._model_mapping.keys():
    565     model_class = _get_model_class(config, cls._model_mapping)
--> 566     return model_class.from_pretrained(
    567         pretrained_model_name_or_path, *model_args, config=config,␣
 ↪**hub_kwargs, **kwargs
    568     )
    569 raise ValueError(
    570     f"Unrecognized configuration class {config.__class__} for this kind␣
 ↪of AutoModel: {cls.__name__}.\n"
    571     f"Model type should be one of {', '.join(c.__name__ for c in cls.
 ↪_model_mapping.keys())}."
    572 )

File ~/.local/lib/python3.10/site-packages/transformers/modeling_utils.py:3383,␣
 ↪in PreTrainedModel.from_pretrained(cls, pretrained_model_name_or_path, config␣
 ↪cache_dir, ignore_mismatched_sizes, force_download, local_files_only, token,␣
 ↪revision, use_safetensors, *model_args, **kwargs)
   3368 try:
   3369     # Load from URL or cache if already cached
   3370     cached_file_kwargs = {
   3371         "cache_dir": cache_dir,
   3372         "force_download": force_download,
   (…)
   3381         "_commit_hash": commit_hash,
   3382     }
-> 3383     resolved_archive_file = cached_file(pretrained_model_name_or_path,␣
 ↪filename, **cached_file_kwargs)
   3385     # Since we set _raise_exceptions_for_missing_entries=False, we don't␣
 ↪get an exception but a None
   3386     # result when internet is up, the repo and revision exist, but the␣
 ↪file does not.
   3387     if resolved_archive_file is None and filename ==␣
 ↪_add_variant(SAFE_WEIGHTS_NAME, variant):
   3388         # Maybe the checkpoint is sharded, we try to grab the index name␣
 ↪in this case.
```

```
File ~/.local/lib/python3.10/site-packages/transformers/utils/hub.py:385, in␣
 ↪cached_file(path_or_repo_id, filename, cache_dir, force_download,␣
 ↪resume_download, proxies, token, revision, local_files_only, subfolder,␣
 ↪repo_type, user_agent, _raise_exceptions_for_missing_entries,␣
 ↪_raise_exceptions_for_connection_errors, _commit_hash, **deprecated_kwargs)
    382 user_agent = http_user_agent(user_agent)
    383 try:
    384     # Load from URL or cache if already cached
--> 385     resolved_file = hf_hub_download(
    386         path_or_repo_id,
    387         filename,
    388         subfolder=None if len(subfolder) == 0 else subfolder,
    389         repo_type=repo_type,
    390         revision=revision,
    391         cache_dir=cache_dir,
    392         user_agent=user_agent,
    393         force_download=force_download,
    394         proxies=proxies,
    395         resume_download=resume_download,
    396         token=token,
    397         local_files_only=local_files_only,
    398     )
    399 except GatedRepoError as e:
    400     raise EnvironmentError(
    401         "You are trying to access a gated repo.\nMake sure to request␣
 ↪access at "
    402         f"https://huggingface.co/{path_or_repo_id} and pass a token␣
 ↪having permission to this repo either "
    403         "by logging in with `huggingface-cli login` or by passing␣
 ↪`token=<your_token>`."
    404     ) from e

File ~/.local/lib/python3.10/site-packages/huggingface_hub/utils/_validators.py
 ↪118, in validate_hf_hub_args.<locals>._inner_fn(*args, **kwargs)
    115 if check_use_auth_token:
    116     kwargs = smoothly_deprecate_use_auth_token(fn_name=fn.__name__,␣
 ↪has_token=has_token, kwargs=kwargs)
--> 118 return fn(*args, **kwargs)

File ~/.local/lib/python3.10/site-packages/huggingface_hub/file_download.py:
 ↪1457, in hf_hub_download(repo_id, filename, subfolder, repo_type, revision,␣
 ↪library_name, library_version, cache_dir, local_dir, local_dir_use_symlinks,␣
 ↪user_agent, force_download, force_filename, proxies, etag_timeout,␣
 ↪resume_download, token, local_files_only, legacy_cache_layout, endpoint)
   1454         if local_dir is not None:
   1455             _check_disk_space(expected_size, local_dir)
-> 1457     http_get(
   1458         url_to_download,
```

```
   1459         temp_file,
   1460         proxies=proxies,
   1461         resume_size=resume_size,
   1462         headers=headers,
   1463         expected_size=expected_size,
   1464     )
   1466 if local_dir is None:
   1467     logger.debug(f"Storing {url} in cache at {blob_path}")

File ~/.local/lib/python3.10/site-packages/huggingface_hub/file_download.py:524
 ↪in http_get(url, temp_file, proxies, resume_size, headers, expected_size,␣
 ↪_nb_retries)
    522 new_resume_size = resume_size
    523 try:
--> 524     for chunk in r.iter_content(chunk_size=DOWNLOAD_CHUNK_SIZE):
    525         if chunk:  # filter out keep-alive new chunks
    526             progress.update(len(chunk))

File ~/.local/lib/python3.10/site-packages/requests/models.py:816, in Response.
 ↪iter_content.<locals>.generate()
    814 if hasattr(self.raw, "stream"):
    815     try:
--> 816         yield from self.raw.stream(chunk_size, decode_content=True)
    817     except ProtocolError as e:
    818         raise ChunkedEncodingError(e)

File /usr/lib/python3/dist-packages/urllib3/response.py:576, in HTTPResponse.
 ↪stream(self, amt, decode_content)
    574 else:
    575     while not is_fp_closed(self._fp):
--> 576         data = self.read(amt=amt, decode_content=decode_content)
    578         if data:
    579             yield data

File /usr/lib/python3/dist-packages/urllib3/response.py:519, in HTTPResponse.
 ↪read(self, amt, decode_content, cache_content)
    517 else:
    518     cache_content = False
--> 519     data = self._fp.read(amt) if not fp_closed else b""
    520     if (
    521         amt != 0 and not data
    522     ):  # Platform-specific: Buggy versions of Python.
   (…)
    528         # not properly close the connection in all cases. There is
    529         # no harm in redundantly calling close.
    530         self._fp.close()

File /usr/lib/python3.10/http/client.py:466, in HTTPResponse.read(self, amt)
```

```
    463 if self.length is not None and amt > self.length:
    464     # clip the read to the "end of response"
    465     amt = self.length
--> 466 s = self.fp.read(amt)
    467 if not s and amt:
    468     # Ideally, we would raise IncompleteRead if the content-length
    469     # wasn't satisfied, but it might break compatibility.
    470     self._close_conn()

File /usr/lib/python3.10/socket.py:705, in SocketIO.readinto(self, b)
    703 while True:
    704     try:
--> 705         return self._sock.recv_into(b)
    706     except timeout:
    707         self._timeout_occurred = True

File /usr/lib/python3.10/ssl.py:1303, in SSLSocket.recv_into(self, buffer,␣
 ↪nbytes, flags)
   1299     if flags != 0:
   1300         raise ValueError(
   1301           "non-zero flags not allowed in calls to recv_into() on %s" %
   1302           self.__class__)
-> 1303     return self.read(nbytes, buffer)
   1304 else:
   1305     return super().recv_into(buffer, nbytes, flags)

File /usr/lib/python3.10/ssl.py:1159, in SSLSocket.read(self, len, buffer)
   1157 try:
   1158     if buffer is not None:
-> 1159         return self._sslobj.read(len, buffer)
   1160     else:
   1161         return self._sslobj.read(len)

KeyboardInterrupt:
```

We similarly view the results as a Pandas data frame. As you can see, the person (PER) of Abraham Lincoln and location (LOC) of the United States is recognized.

[ ]:
```
outputs = tagger(text2)
pd.DataFrame(outputs)
```

5.3 Question Answering

Another common task for NLP is question answering from a reference text. We load such a model with the following code.

```
[ ]: # HIDE OUTPUT
reader = pipeline("question-answering")
question = "What now shall fade?"
```

For this example, we will pose the question "what shall fade" to Hugging Face for Sonnet 18. We see the correct answer of "eternal summer."

```
[ ]: outputs = reader(question=question, context=text)
pd.DataFrame([outputs])
```

5.4 Language Translation

Language translation is yet another common task for NLP and Hugging Face.

```
[ ]: # HIDE OUTPUT
translator = pipeline("translation_en_to_de",
                      model="Helsinki-NLP/opus-mt-en-de")
```

The following code translates Sonnet 18 from English into German.

```
[ ]: outputs = translator(text, clean_up_tokenization_spaces=True,
                         min_length=100)
print(outputs[0]['translation_text'])
```

5.5 Summarization

Summarization is an NLP task that summarizes a more lengthy text into just a few sentences.

```
[ ]: # HIDE OUTPUT
text2 = """
An apple is an edible fruit produced by an apple tree (Malus domestica).
Apple trees are cultivated worldwide and are the most widely grown species
in the genus Malus. The tree originated in Central Asia, where its wild
ancestor, Malus sieversii, is still found today. Apples have been grown
for thousands of years in Asia and Europe and were brought to North America
by European colonists. Apples have religious and mythological significance
in many cultures, including Norse, Greek, and European Christian tradition.
"""

summarizer = pipeline("summarization")
```

The following code summarizes the Wikipedia entry for an "apple."

```
[ ]: outputs = summarizer(text2, max_length=45,
                          clean_up_tokenization_spaces=True)
print(outputs[0]['summary_text'])
```

5.6 Text Generation

Finally, text generation allows us to take an input text and request the pretrained neural network to continue that text.

[ ]:
```
# HIDE OUTPUT
from urllib.request import urlopen

generator = pipeline("text-generation")
```

Here an example is provided that generates additional text after Sonnet 18.

[ ]:
```
outputs = generator(text, max_length=400)
print(outputs[0]['generated_text'])
```