Copyright 2020 The TensorFlow Hub Authors.

In [ ]:

View on TensorFlow.org | Run in Google Colab | View on GitHub | Download notebook | See TF Hub model

Classify text with BERT

This tutorial contains complete code to fine-tune BERT to perform sentiment analysis on a dataset of plain-text IMDB movie reviews. In addition to training a model, you will learn how to preprocess text into an appropriate format.

In this notebook, you will:

- Load the IMDB dataset

- Load a BERT model from TensorFlow Hub
- Build your own model by combining BERT with a classifier
- Train your own model, fine-tuning BERT as part of that
- Save your model and use it to classify sentences

If you're new to working with the IMDB dataset, please see Basic text classification for more details.

About BERT

BERT and other Transformer encoder architectures have been wildly successful on a variety of tasks in NLP (natural language processing). They compute vector-space representations of natural language that are suitable for use in deep learning models. The BERT family of models uses the Transformer encoder architecture to process each token of input text in the full context of all tokens before and after, hence the name: Bidirectional Encoder Representations from Transformers.

BERT models are usually pre-trained on a large corpus of text, then fine-tuned for specific tasks.

Setup

In [1]:
```
# A dependency of the preprocessing for BERT inputs
!pip install -U "tensorflow-text==2.14.0"
```

```
Collecting tensorflow-text==2.14.0
  Downloading tensorflow_text-2.14.0-cp311-cp311-manyli
nux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (1.9
kB)
Requirement already satisfied: tensorflow-hub>=0.13.0 i
n /usr/local/lib/python3.11/dist-packages (from tensorf
low-text==2.14.0) (0.16.1)
Collecting tensorflow<2.15,>=2.14.0 (from tensorflow-te
xt==2.14.0)
  Downloading tensorflow-2.14.1-cp311-cp311-manylinux_2
_17_x86_64.manylinux2014_x86_64.whl.metadata (4.1 kB)
Requirement already satisfied: absl-py>=1.0.0 in /usr/l
ocal/lib/python3.11/dist-packages (from tensorflow<2.1
5,>=2.14.0->tensorflow-text==2.14.0) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /us
r/local/lib/python3.11/dist-packages (from tensorflow<
2.15,>=2.14.0->tensorflow-text==2.14.0) (1.6.3)
Requirement already satisfied: flatbuffers>=23.5.26 in
/usr/local/lib/python3.11/dist-packages (from tensorflo
w<2.15,>=2.14.0->tensorflow-text==2.14.0) (25.2.10)
```

```
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.
5.2,>=0.2.1 in /usr/local/lib/python3.11/dist-packages
(from tensorflow<2.15,>=2.14.0->tensorflow-text==2.14.
0) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in /
usr/local/lib/python3.11/dist-packages (from tensorflow
<2.15,>=2.14.0->tensorflow-text==2.14.0) (0.2.0)
Requirement already satisfied: h5py>=2.9.0 in /usr/loca
l/lib/python3.11/dist-packages (from tensorflow<2.15,>=
2.14.0->tensorflow-text==2.14.0) (3.12.1)
Requirement already satisfied: libclang>=13.0.0 in /us
r/local/lib/python3.11/dist-packages (from tensorflow<
2.15,>=2.14.0->tensorflow-text==2.14.0) (18.1.1)
Collecting ml-dtypes==0.2.0 (from tensorflow<2.15,>=2.1
4.0->tensorflow-text==2.14.0)
  Downloading ml_dtypes-0.2.0-cp311-cp311-manylinux_2_1
7_x86_64.manylinux2014_x86_64.whl.metadata (20 kB)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in
/usr/local/lib/python3.11/dist-packages (from tensorflo
w<2.15,>=2.14.0->tensorflow-text==2.14.0) (1.26.4)
```

```
Requirement already satisfied: opt-einsum>=2.3.2 in /us
r/local/lib/python3.11/dist-packages (from tensorflow<
2.15,>=2.14.0->tensorflow-text==2.14.0) (3.4.0)
Requirement already satisfied: packaging in /usr/local/
lib/python3.11/dist-packages (from tensorflow<2.15,>=2.
14.0->tensorflow-text==2.14.0) (24.2)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.
1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.
3 in /usr/local/lib/python3.11/dist-packages (from tens
orflow<2.15,>=2.14.0->tensorflow-text==2.14.0) (4.25.6)
Requirement already satisfied: setuptools in /usr/loca
l/lib/python3.11/dist-packages (from tensorflow<2.15,>=
2.14.0->tensorflow-text==2.14.0) (75.1.0)
Requirement already satisfied: six>=1.12.0 in /usr/loca
l/lib/python3.11/dist-packages (from tensorflow<2.15,>=
2.14.0->tensorflow-text==2.14.0) (1.17.0)
Requirement already satisfied: termcolor>=1.1.0 in /us
r/local/lib/python3.11/dist-packages (from tensorflow<
2.15,>=2.14.0->tensorflow-text==2.14.0) (2.5.0)
Requirement already satisfied: typing-extensions>=3.6.6
```

```
in /usr/local/lib/python3.11/dist-packages (from tensor
flow<2.15,>=2.14.0->tensorflow-text==2.14.0) (4.12.2)
Collecting wrapt<1.15,>=1.11.0 (from tensorflow<2.15,>=
2.14.0->tensorflow-text==2.14.0)
  Downloading wrapt-1.14.1-cp311-cp311-manylinux_2_5_x8
6_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2
014_x86_64.whl.metadata (6.7 kB)
Requirement already satisfied: tensorflow-io-gcs-filesy
stem>=0.23.1 in /usr/local/lib/python3.11/dist-packages
(from tensorflow<2.15,>=2.14.0->tensorflow-text==2.14.
0) (0.37.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /
usr/local/lib/python3.11/dist-packages (from tensorflow
<2.15,>=2.14.0->tensorflow-text==2.14.0) (1.70.0)
Collecting tensorboard<2.15,>=2.14 (from tensorflow<2.1
5,>=2.14.0->tensorflow-text==2.14.0)
  Downloading tensorboard-2.14.1-py3-none-any.whl.metad
ata (1.7 kB)
Collecting tensorflow-estimator<2.15,>=2.14.0 (from ten
sorflow<2.15,>=2.14.0->tensorflow-text==2.14.0)
```

```
  Downloading tensorflow_estimator-2.14.0-py2.py3-none-
any.whl.metadata (1.3 kB)
Collecting keras<2.15,>=2.14.0 (from tensorflow<2.15,>=
2.14.0->tensorflow-text==2.14.0)
  Downloading keras-2.14.0-py3-none-any.whl.metadata
(2.4 kB)
Requirement already satisfied: tf-keras>=2.14.1 in /us
r/local/lib/python3.11/dist-packages (from tensorflow-h
ub>=0.13.0->tensorflow-text==2.14.0) (2.18.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /u
sr/local/lib/python3.11/dist-packages (from astunparse>
=1.6.0->tensorflow<2.15,>=2.14.0->tensorflow-text==2.1
4.0) (0.45.1)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/usr/local/lib/python3.11/dist-packages (from tensorboa
rd<2.15,>=2.14->tensorflow<2.15,>=2.14.0->tensorflow-te
xt==2.14.0) (2.38.0)
Collecting google-auth-oauthlib<1.1,>=0.5 (from tensorb
oard<2.15,>=2.14->tensorflow<2.15,>=2.14.0->tensorflow-
text==2.14.0)
```

```
  Downloading google_auth_oauthlib-1.0.0-py2.py3-none-a
ny.whl.metadata (2.7 kB)
Requirement already satisfied: markdown>=2.6.8 in /usr/
local/lib/python3.11/dist-packages (from tensorboard<2.
15,>=2.14->tensorflow<2.15,>=2.14.0->tensorflow-text==
2.14.0) (3.7)
Requirement already satisfied: requests<3,>=2.21.0 in /
usr/local/lib/python3.11/dist-packages (from tensorboar
d<2.15,>=2.14->tensorflow<2.15,>=2.14.0->tensorflow-tex
t==2.14.0) (2.32.3)
Requirement already satisfied: tensorboard-data-server<
0.8.0,>=0.7.0 in /usr/local/lib/python3.11/dist-package
s (from tensorboard<2.15,>=2.14->tensorflow<2.15,>=2.1
4.0->tensorflow-text==2.14.0) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/
local/lib/python3.11/dist-packages (from tensorboard<2.
15,>=2.14->tensorflow<2.15,>=2.14.0->tensorflow-text==
2.14.0) (3.1.3)
INFO: pip is looking at multiple versions of tf-keras t
o determine which version is compatible with other requ
```

```
irements. This could take a while.
Collecting tf-keras>=2.14.1 (from tensorflow-hub>=0.13.
0->tensorflow-text==2.14.0)
  Downloading tf_keras-2.19.0-py3-none-any.whl.metadata
(1.8 kB)
  Downloading tf_keras-2.17.0-py3-none-any.whl.metadata
(1.6 kB)
  Downloading tf_keras-2.16.0-py3-none-any.whl.metadata
(1.6 kB)
  Downloading tf_keras-2.15.1-py3-none-any.whl.metadata
(1.7 kB)
  Downloading tf_keras-2.15.0-py3-none-any.whl.metadata
(1.6 kB)
Requirement already satisfied: cachetools<6.0,>=2.0.0 i
n /usr/local/lib/python3.11/dist-packages (from google-
auth<3,>=1.6.3->tensorboard<2.15,>=2.14->tensorflow<2.1
5,>=2.14.0->tensorflow-text==2.14.0) (5.5.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.11/dist-packages (from google-au
th<3,>=1.6.3->tensorboard<2.15,>=2.14->tensorflow<2.15,
>=2.14.0->tensorflow-text==2.14.0) (0.4.1)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/lo
cal/lib/python3.11/dist-packages (from google-auth<3,>=
1.6.3->tensorboard<2.15,>=2.14->tensorflow<2.15,>=2.14.
0->tensorflow-text==2.14.0) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0
in /usr/local/lib/python3.11/dist-packages (from google
-auth-oauthlib<1.1,>=0.5->tensorboard<2.15,>=2.14->tens
orflow<2.15,>=2.14.0->tensorflow-text==2.14.0) (2.0.0)
Requirement already satisfied: charset-normalizer<4,>=2
in /usr/local/lib/python3.11/dist-packages (from reques
ts<3,>=2.21.0->tensorboard<2.15,>=2.14->tensorflow<2.1
5,>=2.14.0->tensorflow-text==2.14.0) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/loc
al/lib/python3.11/dist-packages (from requests<3,>=2.2
1.0->tensorboard<2.15,>=2.14->tensorflow<2.15,>=2.14.0-
>tensorflow-text==2.14.0) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /u
sr/local/lib/python3.11/dist-packages (from requests<3,
>=2.21.0->tensorboard<2.15,>=2.14->tensorflow<2.15,>=2.
```

```
14.0->tensorflow-text==2.14.0) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /u
sr/local/lib/python3.11/dist-packages (from requests<3,
>=2.21.0->tensorboard<2.15,>=2.14->tensorflow<2.15,>=2.
14.0->tensorflow-text==2.14.0) (2025.1.31)
Requirement already satisfied: MarkupSafe>=2.1.1 in /us
r/local/lib/python3.11/dist-packages (from werkzeug>=1.
0.1->tensorboard<2.15,>=2.14->tensorflow<2.15,>=2.14.0-
>tensorflow-text==2.14.0) (3.0.2)
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in
/usr/local/lib/python3.11/dist-packages (from pyasn1-mo
dules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.15,>
=2.14->tensorflow<2.15,>=2.14.0->tensorflow-text==2.14.
0) (0.6.1)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/
local/lib/python3.11/dist-packages (from requests-oauth
lib>=0.7.0->google-auth-oauthlib<1.1,>=0.5->tensorboard
<2.15,>=2.14->tensorflow<2.15,>=2.14.0->tensorflow-text
==2.14.0) (3.2.2)
Downloading tensorflow_text-2.14.0-cp311-cp311-manylinu
```

```
x_2_17_x86_64.manylinux2014_x86_64.whl (6.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.5/6.5 MB
38.6 MB/s eta 0:00:00
Downloading tensorflow-2.14.1-cp311-cp311-manylinux_2_1
7_x86_64.manylinux2014_x86_64.whl (489.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 489.9/489.9
MB 2.0 MB/s eta 0:00:00
Downloading ml_dtypes-0.2.0-cp311-cp311-manylinux_2_17_
x86_64.manylinux2014_x86_64.whl (1.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.0/1.0 MB
16.0 MB/s eta 0:00:00
Downloading keras-2.14.0-py3-none-any.whl (1.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.7/1.7 MB
51.5 MB/s eta 0:00:00
Downloading tensorboard-2.14.1-py3-none-any.whl (5.5 M
B)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.5/5.5 MB
90.0 MB/s eta 0:00:00
Downloading tensorflow_estimator-2.14.0-py2.py3-none-an
y.whl (440 kB)
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 440.7/440.7
kB 26.0 MB/s eta 0:00:00
Downloading tf_keras-2.15.0-py3-none-any.whl (1.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.7/1.7 MB
58.2 MB/s eta 0:00:00
Downloading wrapt-1.14.1-cp311-cp311-manylinux_2_5_x86_
64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux201
4_x86_64.whl (78 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 78.4/78.4 k
B 6.6 MB/s eta 0:00:00
Downloading google_auth_oauthlib-1.0.0-py2.py3-none-an
y.whl (18 kB)
Installing collected packages: wrapt, tf-keras, tensorf
low-estimator, ml-dtypes, keras, google-auth-oauthlib,
tensorboard, tensorflow, tensorflow-text
  Attempting uninstall: wrapt
    Found existing installation: wrapt 1.17.2
    Uninstalling wrapt-1.17.2:
      Successfully uninstalled wrapt-1.17.2
  Attempting uninstall: tf-keras
```

```
    Found existing installation: tf_keras 2.18.0
    Uninstalling tf_keras-2.18.0:
      Successfully uninstalled tf_keras-2.18.0
  Attempting uninstall: ml-dtypes
    Found existing installation: ml-dtypes 0.4.1
    Uninstalling ml-dtypes-0.4.1:
      Successfully uninstalled ml-dtypes-0.4.1
  Attempting uninstall: keras
    Found existing installation: keras 3.8.0
    Uninstalling keras-3.8.0:
      Successfully uninstalled keras-3.8.0
  Attempting uninstall: google-auth-oauthlib
    Found existing installation: google-auth-oauthlib
1.2.1
    Uninstalling google-auth-oauthlib-1.2.1:
      Successfully uninstalled google-auth-oauthlib-1.
2.1
  Attempting uninstall: tensorboard
    Found existing installation: tensorboard 2.18.0
    Uninstalling tensorboard-2.18.0:
      Successfully uninstalled tensorboard-2.18.0
  Attempting uninstall: tensorflow
    Found existing installation: tensorflow 2.18.0
    Uninstalling tensorflow-2.18.0:
      Successfully uninstalled tensorflow-2.18.0
  Attempting uninstall: tensorflow-text
    Found existing installation: tensorflow-text 2.18.1
    Uninstalling tensorflow-text-2.18.1:
      Successfully uninstalled tensorflow-text-2.18.1
ERROR: pip's dependency resolver does not currently tak
e into account all the packages that are installed. Thi
s behaviour is the source of the following dependency c
onflicts.
dopamine-rl 4.1.2 requires tf-keras>=2.18.0, but you ha
ve tf-keras 2.15.0 which is incompatible.
tensorstore 0.1.72 requires ml_dtypes>=0.3.1, but you h
ave ml-dtypes 0.2.0 which is incompatible.
Successfully installed google-auth-oauthlib-1.0.0 keras
-2.14.0 ml-dtypes-0.2.0 tensorboard-2.14.1 tensorflow-
```

```
2.14.1 tensorflow-estimator-2.14.0 tensorflow-text-2.1
4.0 tf-keras-2.15.0 wrapt-1.14.1
```

You will use the AdamW optimizer from tensorflow/models.

```
In [2]: !pip install "tf-models-official==2.14.0"
```

```
Collecting tf-models-official==2.14.0
  Downloading tf_models_official-2.14.0-py2.py3-none-an
y.whl.metadata (1.4 kB)
Requirement already satisfied: Cython in /usr/local/li
b/python3.11/dist-packages (from tf-models-official==2.
14.0) (3.0.12)
Requirement already satisfied: Pillow in /usr/local/li
b/python3.11/dist-packages (from tf-models-official==2.
14.0) (11.1.0)
Requirement already satisfied: gin-config in /usr/loca
l/lib/python3.11/dist-packages (from tf-models-official
==2.14.0) (0.5.0)
Requirement already satisfied: google-api-python-client
>=1.6.7 in /usr/local/lib/python3.11/dist-packages (fro
m tf-models-official==2.14.0) (2.160.0)
Requirement already satisfied: immutabledict in /usr/lo
cal/lib/python3.11/dist-packages (from tf-models-offici
al==2.14.0) (4.2.1)
Requirement already satisfied: kaggle>=1.3.9 in /usr/lo
cal/lib/python3.11/dist-packages (from tf-models-offici
al==2.14.0) (1.6.17)
Requirement already satisfied: matplotlib in /usr/loca
l/lib/python3.11/dist-packages (from tf-models-official
==2.14.0) (3.10.0)
Requirement already satisfied: numpy>=1.20 in /usr/loca
l/lib/python3.11/dist-packages (from tf-models-official
==2.14.0) (1.26.4)
Requirement already satisfied: oauth2client in /usr/loc
al/lib/python3.11/dist-packages (from tf-models-officia
l==2.14.0) (4.1.3)
Requirement already satisfied: opencv-python-headless i
n /usr/local/lib/python3.11/dist-packages (from tf-mode
ls-official==2.14.0) (4.11.0.86)
Requirement already satisfied: pandas>=0.22.0 in /usr/l
ocal/lib/python3.11/dist-packages (from tf-models-offic
ial==2.14.0) (2.2.2)
Requirement already satisfied: psutil>=5.4.3 in /usr/lo
cal/lib/python3.11/dist-packages (from tf-models-offici
al==2.14.0) (5.9.5)
Requirement already satisfied: py-cpuinfo>=3.3.0 in /us
```

```
r/local/lib/python3.11/dist-packages (from tf-models-of
ficial==2.14.0) (9.0.0)
Requirement already satisfied: pycocotools in /usr/loca
l/lib/python3.11/dist-packages (from tf-models-official
==2.14.0) (2.0.8)
Requirement already satisfied: pyyaml>=6.0.0 in /usr/lo
cal/lib/python3.11/dist-packages (from tf-models-offici
al==2.14.0) (6.0.2)
Collecting sacrebleu (from tf-models-official==2.14.0)
  Downloading sacrebleu-2.5.1-py3-none-any.whl.metadata
(51 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/51.8
kB ? eta -:--:--
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 51.8/51.8
kB 1.7 MB/s eta 0:00:00
Requirement already satisfied: scipy>=0.19.1 in /usr/lo
cal/lib/python3.11/dist-packages (from tf-models-offici
al==2.14.0) (1.13.1)
Requirement already satisfied: sentencepiece in /usr/lo
cal/lib/python3.11/dist-packages (from tf-models-offici
al==2.14.0) (0.2.0)
Collecting seqeval (from tf-models-official==2.14.0)
  Downloading seqeval-1.2.2.tar.gz (43 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 43.6/43.6
kB 1.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: six in /usr/local/lib/py
thon3.11/dist-packages (from tf-models-official==2.14.
0) (1.17.0)
Requirement already satisfied: tensorflow-datasets in /
usr/local/lib/python3.11/dist-packages (from tf-models-
official==2.14.0) (4.9.7)
Requirement already satisfied: tensorflow-hub>=0.6.0 in
/usr/local/lib/python3.11/dist-packages (from tf-models
-official==2.14.0) (0.16.1)
Collecting tensorflow-model-optimization>=0.4.1 (from t
f-models-official==2.14.0)
  Downloading tensorflow_model_optimization-0.8.0-py2.p
y3-none-any.whl.metadata (904 bytes)
Requirement already satisfied: tensorflow-text~=2.14.0
```

```
in /usr/local/lib/python3.11/dist-packages (from tf-mod
els-official==2.14.0) (2.14.0)
Requirement already satisfied: tensorflow~=2.14.0 in /u
sr/local/lib/python3.11/dist-packages (from tf-models-o
fficial==2.14.0) (2.14.1)
Requirement already satisfied: tf-slim>=1.1.0 in /usr/l
ocal/lib/python3.11/dist-packages (from tf-models-offic
ial==2.14.0) (1.1.0)
Requirement already satisfied: httplib2<1.dev0,>=0.19.0
in /usr/local/lib/python3.11/dist-packages (from google
-api-python-client>=1.6.7->tf-models-official==2.14.0)
(0.22.0)
Requirement already satisfied: google-auth!=2.24.0,!=2.
25.0,<3.0.0.dev0,>=1.32.0 in /usr/local/lib/python3.11/
dist-packages (from google-api-python-client>=1.6.7->tf
-models-official==2.14.0) (2.38.0)
Requirement already satisfied: google-auth-httplib2<1.
0.0,>=0.2.0 in /usr/local/lib/python3.11/dist-packages
(from google-api-python-client>=1.6.7->tf-models-offici
al==2.14.0) (0.2.0)
```

```
Requirement already satisfied: google-api-core!=2.0.*,!
=2.1.*,!=2.2.*,!=2.3.0,<3.0.0.dev0,>=1.31.5 in /usr/loc
al/lib/python3.11/dist-packages (from google-api-python
-client>=1.6.7->tf-models-official==2.14.0) (2.24.1)
Requirement already satisfied: uritemplate<5,>=3.0.1 in
/usr/local/lib/python3.11/dist-packages (from google-ap
i-python-client>=1.6.7->tf-models-official==2.14.0) (4.
1.1)
Requirement already satisfied: certifi>=2023.7.22 in /u
sr/local/lib/python3.11/dist-packages (from kaggle>=1.
3.9->tf-models-official==2.14.0) (2025.1.31)
Requirement already satisfied: python-dateutil in /usr/
local/lib/python3.11/dist-packages (from kaggle>=1.3.9-
>tf-models-official==2.14.0) (2.8.2)
Requirement already satisfied: requests in /usr/local/l
ib/python3.11/dist-packages (from kaggle>=1.3.9->tf-mod
els-official==2.14.0) (2.32.3)
Requirement already satisfied: tqdm in /usr/local/lib/p
ython3.11/dist-packages (from kaggle>=1.3.9->tf-models-
official==2.14.0) (4.67.1)
```

```
Requirement already satisfied: python-slugify in /usr/l
ocal/lib/python3.11/dist-packages (from kaggle>=1.3.9->
tf-models-official==2.14.0) (8.0.4)
Requirement already satisfied: urllib3 in /usr/local/li
b/python3.11/dist-packages (from kaggle>=1.3.9->tf-mode
ls-official==2.14.0) (2.3.0)
Requirement already satisfied: bleach in /usr/local/li
b/python3.11/dist-packages (from kaggle>=1.3.9->tf-mode
ls-official==2.14.0) (6.2.0)
Requirement already satisfied: pytz>=2020.1 in /usr/loc
al/lib/python3.11/dist-packages (from pandas>=0.22.0->t
f-models-official==2.14.0) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in /usr/l
ocal/lib/python3.11/dist-packages (from pandas>=0.22.0-
>tf-models-official==2.14.0) (2025.1)
Requirement already satisfied: absl-py>=1.0.0 in /usr/l
ocal/lib/python3.11/dist-packages (from tensorflow~=2.1
4.0->tf-models-official==2.14.0) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /us
r/local/lib/python3.11/dist-packages (from tensorflow~=
2.14.0->tf-models-official==2.14.0) (1.6.3)
Requirement already satisfied: flatbuffers>=23.5.26 in
/usr/local/lib/python3.11/dist-packages (from tensorflo
w~=2.14.0->tf-models-official==2.14.0) (25.2.10)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.
5.2,>=0.2.1 in /usr/local/lib/python3.11/dist-packages
(from tensorflow~=2.14.0->tf-models-official==2.14.0)
(0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in /
usr/local/lib/python3.11/dist-packages (from tensorflow
~=2.14.0->tf-models-official==2.14.0) (0.2.0)
Requirement already satisfied: h5py>=2.9.0 in /usr/loca
l/lib/python3.11/dist-packages (from tensorflow~=2.14.0
->tf-models-official==2.14.0) (3.12.1)
Requirement already satisfied: libclang>=13.0.0 in /us
r/local/lib/python3.11/dist-packages (from tensorflow~=
2.14.0->tf-models-official==2.14.0) (18.1.1)
Requirement already satisfied: ml-dtypes==0.2.0 in /us
r/local/lib/python3.11/dist-packages (from tensorflow~=
2.14.0->tf-models-official==2.14.0) (0.2.0)
```

```
Requirement already satisfied: opt-einsum>=2.3.2 in /us
r/local/lib/python3.11/dist-packages (from tensorflow~=
2.14.0->tf-models-official==2.14.0) (3.4.0)
Requirement already satisfied: packaging in /usr/local/
lib/python3.11/dist-packages (from tensorflow~=2.14.0->
tf-models-official==2.14.0) (24.2)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.
1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.
3 in /usr/local/lib/python3.11/dist-packages (from tens
orflow~=2.14.0->tf-models-official==2.14.0) (4.25.6)
Requirement already satisfied: setuptools in /usr/loca
l/lib/python3.11/dist-packages (from tensorflow~=2.14.0
->tf-models-official==2.14.0) (75.1.0)
Requirement already satisfied: termcolor>=1.1.0 in /us
r/local/lib/python3.11/dist-packages (from tensorflow~=
2.14.0->tf-models-official==2.14.0) (2.5.0)
Requirement already satisfied: typing-extensions>=3.6.6
in /usr/local/lib/python3.11/dist-packages (from tensor
flow~=2.14.0->tf-models-official==2.14.0) (4.12.2)
Requirement already satisfied: wrapt<1.15,>=1.11.0 in /
```

```
usr/local/lib/python3.11/dist-packages (from tensorflow
~=2.14.0->tf-models-official==2.14.0) (1.14.1)
Requirement already satisfied: tensorflow-io-gcs-filesy
stem>=0.23.1 in /usr/local/lib/python3.11/dist-packages
(from tensorflow~=2.14.0->tf-models-official==2.14.0)
(0.37.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /
usr/local/lib/python3.11/dist-packages (from tensorflow
~=2.14.0->tf-models-official==2.14.0) (1.70.0)
Requirement already satisfied: tensorboard<2.15,>=2.14
in /usr/local/lib/python3.11/dist-packages (from tensor
flow~=2.14.0->tf-models-official==2.14.0) (2.14.1)
Requirement already satisfied: tensorflow-estimator<2.1
5,>=2.14.0 in /usr/local/lib/python3.11/dist-packages
(from tensorflow~=2.14.0->tf-models-official==2.14.0)
(2.14.0)
Requirement already satisfied: keras<2.15,>=2.14.0 in /
usr/local/lib/python3.11/dist-packages (from tensorflow
~=2.14.0->tf-models-official==2.14.0) (2.14.0)
Requirement already satisfied: tf-keras>=2.14.1 in /us
```

```
r/local/lib/python3.11/dist-packages (from tensorflow-h
ub>=0.6.0->tf-models-official==2.14.0) (2.15.0)
Requirement already satisfied: dm-tree~=0.1.1 in /usr/l
ocal/lib/python3.11/dist-packages (from tensorflow-mode
l-optimization>=0.4.1->tf-models-official==2.14.0) (0.
1.9)
Requirement already satisfied: contourpy>=1.0.1 in /us
r/local/lib/python3.11/dist-packages (from matplotlib->
tf-models-official==2.14.0) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/loc
al/lib/python3.11/dist-packages (from matplotlib->tf-mo
dels-official==2.14.0) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /us
r/local/lib/python3.11/dist-packages (from matplotlib->
tf-models-official==2.14.0) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /us
r/local/lib/python3.11/dist-packages (from matplotlib->
tf-models-official==2.14.0) (1.4.8)
Requirement already satisfied: pyparsing>=2.3.1 in /us
r/local/lib/python3.11/dist-packages (from matplotlib->
tf-models-official==2.14.0) (3.2.1)
Requirement already satisfied: pyasn1>=0.1.7 in /usr/lo
cal/lib/python3.11/dist-packages (from oauth2client->tf
-models-official==2.14.0) (0.6.1)
Requirement already satisfied: pyasn1-modules>=0.0.5 in
/usr/local/lib/python3.11/dist-packages (from oauth2cli
ent->tf-models-official==2.14.0) (0.4.1)
Requirement already satisfied: rsa>=3.1.4 in /usr/loca
l/lib/python3.11/dist-packages (from oauth2client->tf-m
odels-official==2.14.0) (4.9)
Collecting portalocker (from sacrebleu->tf-models-offic
ial==2.14.0)
  Downloading portalocker-3.1.1-py3-none-any.whl.metada
ta (8.6 kB)
Requirement already satisfied: regex in /usr/local/lib/
python3.11/dist-packages (from sacrebleu->tf-models-off
icial==2.14.0) (2024.11.6)
Requirement already satisfied: tabulate>=0.8.9 in /usr/
local/lib/python3.11/dist-packages (from sacrebleu->tf-
models-official==2.14.0) (0.9.0)
```

```
Collecting colorama (from sacrebleu->tf-models-official
==2.14.0)
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metad
ata (17 kB)
Requirement already satisfied: lxml in /usr/local/lib/p
ython3.11/dist-packages (from sacrebleu->tf-models-offi
cial==2.14.0) (5.3.1)
Requirement already satisfied: scikit-learn>=0.21.3 in
/usr/local/lib/python3.11/dist-packages (from seqeval->
tf-models-official==2.14.0) (1.6.1)
Requirement already satisfied: click in /usr/local/lib/
python3.11/dist-packages (from tensorflow-datasets->tf-
models-official==2.14.0) (8.1.8)
Requirement already satisfied: promise in /usr/local/li
b/python3.11/dist-packages (from tensorflow-datasets->t
f-models-official==2.14.0) (2.3)
Requirement already satisfied: pyarrow in /usr/local/li
b/python3.11/dist-packages (from tensorflow-datasets->t
f-models-official==2.14.0) (18.1.0)
Requirement already satisfied: simple-parsing in /usr/l
```

```
ocal/lib/python3.11/dist-packages (from tensorflow-data
sets->tf-models-official==2.14.0) (0.1.7)
Requirement already satisfied: tensorflow-metadata in /
usr/local/lib/python3.11/dist-packages (from tensorflow
-datasets->tf-models-official==2.14.0) (1.16.1)
Requirement already satisfied: toml in /usr/local/lib/p
ython3.11/dist-packages (from tensorflow-datasets->tf-m
odels-official==2.14.0) (0.10.2)
Requirement already satisfied: array-record>=0.5.0 in /
usr/local/lib/python3.11/dist-packages (from tensorflow
-datasets->tf-models-official==2.14.0) (0.6.0)
Requirement already satisfied: etils>=1.9.1 in /usr/loc
al/lib/python3.11/dist-packages (from etils[edc,enp,epa
th,epy,etree]>=1.9.1; python_version >= "3.11"->tensorf
low-datasets->tf-models-official==2.14.0) (1.12.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /u
sr/local/lib/python3.11/dist-packages (from astunparse>
=1.6.0->tensorflow~=2.14.0->tf-models-official==2.14.0)
(0.45.1)
Requirement already satisfied: attrs>=18.2.0 in /usr/lo
```

```
cal/lib/python3.11/dist-packages (from dm-tree~=0.1.1->
tensorflow-model-optimization>=0.4.1->tf-models-officia
l==2.14.0) (25.1.0)
Requirement already satisfied: fsspec in /usr/local/li
b/python3.11/dist-packages (from etils[edc,enp,epath,ep
y,etree]>=1.9.1; python_version >= "3.11"->tensorflow-d
atasets->tf-models-official==2.14.0) (2024.10.0)
Requirement already satisfied: importlib_resources in /
usr/local/lib/python3.11/dist-packages (from etils[edc,
enp,epath,epy,etree]>=1.9.1; python_version >= "3.11"->
tensorflow-datasets->tf-models-official==2.14.0) (6.5.
2)
Requirement already satisfied: zipp in /usr/local/lib/p
ython3.11/dist-packages (from etils[edc,enp,epath,epy,e
tree]>=1.9.1; python_version >= "3.11"->tensorflow-data
sets->tf-models-official==2.14.0) (3.21.0)
Requirement already satisfied: googleapis-common-protos
<2.0.dev0,>=1.56.2 in /usr/local/lib/python3.11/dist-pa
ckages (from google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=
2.3.0,<3.0.0.dev0,>=1.31.5->google-api-python-client>=
1.6.7->tf-models-official==2.14.0) (1.69.0)
Requirement already satisfied: proto-plus<2.0.0dev,>=1.
22.3 in /usr/local/lib/python3.11/dist-packages (from g
oogle-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0.de
v0,>=1.31.5->google-api-python-client>=1.6.7->tf-models
-official==2.14.0) (1.26.0)
Requirement already satisfied: cachetools<6.0,>=2.0.0 i
n /usr/local/lib/python3.11/dist-packages (from google-
auth!=2.24.0,!=2.25.0,<3.0.0.dev0,>=1.32.0->google-api-
python-client>=1.6.7->tf-models-official==2.14.0) (5.5.
2)
Requirement already satisfied: charset-normalizer<4,>=2
in /usr/local/lib/python3.11/dist-packages (from reques
ts->kaggle>=1.3.9->tf-models-official==2.14.0) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/loc
al/lib/python3.11/dist-packages (from requests->kaggle>
=1.3.9->tf-models-official==2.14.0) (3.10)
Requirement already satisfied: joblib>=1.2.0 in /usr/lo
cal/lib/python3.11/dist-packages (from scikit-learn>=0.
21.3->seqeval->tf-models-official==2.14.0) (1.4.2)
```

```
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.11/dist-packages (from scikit-le
arn>=0.21.3->seqeval->tf-models-official==2.14.0) (3.5.
0)
Requirement already satisfied: google-auth-oauthlib<1.
1,>=0.5 in /usr/local/lib/python3.11/dist-packages (fro
m tensorboard<2.15,>=2.14->tensorflow~=2.14.0->tf-model
s-official==2.14.0) (1.0.0)
Requirement already satisfied: markdown>=2.6.8 in /usr/
local/lib/python3.11/dist-packages (from tensorboard<2.
15,>=2.14->tensorflow~=2.14.0->tf-models-official==2.1
4.0) (3.7)
Requirement already satisfied: tensorboard-data-server<
0.8.0,>=0.7.0 in /usr/local/lib/python3.11/dist-package
s (from tensorboard<2.15,>=2.14->tensorflow~=2.14.0->tf
-models-official==2.14.0) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/
local/lib/python3.11/dist-packages (from tensorboard<2.
15,>=2.14->tensorflow~=2.14.0->tf-models-official==2.1
4.0) (3.1.3)
```

```
Requirement already satisfied: webencodings in /usr/loc
al/lib/python3.11/dist-packages (from bleach->kaggle>=
1.3.9->tf-models-official==2.14.0) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in /
usr/local/lib/python3.11/dist-packages (from python-slu
gify->kaggle>=1.3.9->tf-models-official==2.14.0) (1.3)
Requirement already satisfied: docstring-parser<1.0,>=
0.15 in /usr/local/lib/python3.11/dist-packages (from s
imple-parsing->tensorflow-datasets->tf-models-official=
=2.14.0) (0.16)
Requirement already satisfied: requests-oauthlib>=0.7.0
in /usr/local/lib/python3.11/dist-packages (from google
-auth-oauthlib<1.1,>=0.5->tensorboard<2.15,>=2.14->tens
orflow~=2.14.0->tf-models-official==2.14.0) (2.0.0)
Requirement already satisfied: MarkupSafe>=2.1.1 in /us
r/local/lib/python3.11/dist-packages (from werkzeug>=1.
0.1->tensorboard<2.15,>=2.14->tensorflow~=2.14.0->tf-mo
dels-official==2.14.0) (3.0.2)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/
local/lib/python3.11/dist-packages (from requests-oauth
lib>=0.7.0->google-auth-oauthlib<1.1,>=0.5->tensorboard
<2.15,>=2.14->tensorflow~=2.14.0->tf-models-official==
2.14.0) (3.2.2)
Downloading tf_models_official-2.14.0-py2.py3-none-any.
whl (2.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.7/2.7 MB
32.6 MB/s eta 0:00:00
Downloading tensorflow_model_optimization-0.8.0-py2.py3
-none-any.whl (242 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 242.5/242.5
kB 19.1 MB/s eta 0:00:00
Downloading sacrebleu-2.5.1-py3-none-any.whl (104 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 104.1/104.1
kB 8.1 MB/s eta 0:00:00
Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Downloading portalocker-3.1.1-py3-none-any.whl (19 kB)
Building wheels for collected packages: seqeval
  Building wheel for seqeval (setup.py) ... done
  Created wheel for seqeval: filename=seqeval-1.2.2-py3
-none-any.whl size=16161 sha256=856869639cdfe5f75a87ee1
```

```
1a63adfe372a37f86a42d7b82517a5776381e6474
  Stored in directory: /root/.cache/pip/wheels/bc/92/f
0/243288f899c2eacdfa8c5f9aede4c71a9bad0ee26a01dc5ead
Successfully built seqeval
Installing collected packages: portalocker, colorama, t
ensorflow-model-optimization, sacrebleu, seqeval, tf-mo
dels-official
Successfully installed colorama-0.4.6 portalocker-3.1.1
sacrebleu-2.5.1 seqeval-1.2.2 tensorflow-model-optimiza
tion-0.8.0 tf-models-official-2.14.0
```

In [3]:
```python
import os
import shutil

import tensorflow as tf
import tensorflow_hub as hub
import tensorflow_text as text
from official.nlp import optimization  # to create Ada

import matplotlib.pyplot as plt
```

```
tf.get_logger().setLevel('ERROR')
```

In [4]:
```
print("TensorFlow version:", tf.__version__)
print("TensorFlow Hub version:", hub.__version__)
```

```
TensorFlow version: 2.14.1
TensorFlow Hub version: 0.16.1
```

Sentiment analysis

This notebook trains a sentiment analysis model to classify movie reviews as *positive* or *negative*, based on the text of the review.

You'll use the Large Movie Review Dataset that contains the text of 50,000 movie reviews from the Internet Movie

Database.

Download the IMDB dataset

Let's download and extract the dataset, then explore the directory structure.

In [5]:
```
url = 'https://ai.stanford.edu/~amaas/data/sentiment/a

dataset = tf.keras.utils.get_file('aclImdb_v1.tar.gz',
                                  untar=True, cache_di
                                  cache_subdir='')

dataset_dir = os.path.join(os.path.dirname(dataset), '
train_dir = os.path.join(dataset_dir, 'train')

# remove unused folders to make it easier to load the
```

```
remove_dir = os.path.join(train_dir, 'unsup')
shutil.rmtree(remove_dir)
```

```
Downloading data from https://ai.stanford.edu/~amaas/da
ta/sentiment/aclImdb_v1.tar.gz
84125825/84125825 [==============================] - 16
s 0us/step
```

Next, you will use the `text_dataset_from_directory` utility to create a labeled `tf.data.Dataset`.

The IMDB dataset has already been divided into train and test, but it lacks a validation set. Let's create a validation set using an 80:20 split of the training data by using the `validation_split` argument below.

Note: When using the `validation_split` and `subset` arguments, make sure to either specify a random seed, or to pass `shuffle=False`, so that the validation and training splits have no overlap.

In [6]:
```
AUTOTUNE = tf.data.AUTOTUNE
batch_size = 32
seed = 42

raw_train_ds = tf.keras.utils.text_dataset_from_direct
    'aclImdb/train',
    batch_size=batch_size,
    validation_split=0.2,
    subset='training',
    seed=seed)

class_names = raw_train_ds.class_names
train_ds = raw_train_ds.cache().prefetch(buffer_size=A

val_ds = tf.keras.utils.text_dataset_from_directory(
    'aclImdb/train',
```

```python
    batch_size=batch_size,
    validation_split=0.2,
    subset='validation',
    seed=seed)

val_ds = val_ds.cache().prefetch(buffer_size=AUTOTUNE)

test_ds = tf.keras.utils.text_dataset_from_directory(
    'aclImdb/test',
    batch_size=batch_size)

test_ds = test_ds.cache().prefetch(buffer_size=AUTOTUN
```

```
Found 25000 files belonging to 2 classes.
Using 20000 files for training.
Found 25000 files belonging to 2 classes.
Using 5000 files for validation.
Found 25000 files belonging to 2 classes.
```

Let's take a look at a few reviews.

In [7]:
```
for text_batch, label_batch in train_ds.take(1):
  for i in range(3):
    print(f'Review: {text_batch.numpy()[i]}')
    label = label_batch.numpy()[i]
    print(f'Label : {label} ({class_names[label]})')
```

```
Review: b'"Pandemonium" is a horror movie spoof that co
mes off more stupid than funny. Believe me when I tell
you, I love comedies. Especially comedy spoofs. "Airpla
ne", "The Naked Gun" trilogy, "Blazing Saddles", "High
Anxiety", and "Spaceballs" are some of my favorite come
dies that spoof a particular genre. "Pandemonium" is no
t up there with those films. Most of the scenes in this
movie had me sitting there in stunned silence because t
he movie wasn\'t all that funny. There are a few laughs
in the film, but when you watch a comedy, you expect to
laugh a lot more than a few times and that\'s all this
film has going for it. Geez, "Scream" had more laughs t
han this film and that was more of a horror film. How b
izarre is that?<br /><br />*1/2 (out of four)'
Label : 0 (neg)
Review: b"David Mamet is a very interesting and a very
un-equal director. His first movie 'House of Games' was
the one I liked best, and it set a series of films with
characters whose perspective of life changes as they ge
t into complicated situations, and so does the perspect
```

```
ive of the viewer.<br /><br />So is 'Homicide' which fr
om the title tries to set the mind of the viewer to the
usual crime drama. The principal characters are two cop
s, one Jewish and one Irish who deal with a racially ch
arged area. The murder of an old Jewish shop owner who
proves to be an ancient veteran of the Israeli Independ
ence war triggers the Jewish identity in the mind and h
eart of the Jewish detective.<br /><br />This is were t
he flaws of the film are the more obvious. The process
of awakening is theatrical and hard to believe, the gro
up of Jewish militants is operatic, and the way the det
ective eventually walks to the final violent confrontat
ion is pathetic. The end of the film itself is Mamet-li
ke smart, but disappoints from a human emotional perspe
ctive.<br /><br />Joe Mantegna and William Macy give st
rong performances, but the flaws of the story are too e
vident to be easily compensated."
Label : 0 (neg)
Review: b'Great documentary about the lives of NY firef
ighters during the worst terrorist attack of all time..
```

```
That reason alone is why this should be a must see coll
ectors item.. What shocked me was not only the attacks,
but the"High Fat Diet" and physical appearance of some
of these firefighters. I think a lot of Doctors would a
gree with me that,in the physical shape they were in, s
ome of these firefighters would NOT of made it to the 7
9th floor carrying over 60 lbs of gear. Having said tha
t i now have a greater respect for firefighters and i r
ealize becoming a firefighter is a life altering job. T
he French have a history of making great documentary\'s
and that is what this is, a Great Documentary.....'
Label : 1 (pos)
```

Loading models from TensorFlow Hub

Here you can choose which BERT model you will load from TensorFlow Hub and fine-tune. There are multiple BERT models available.

- BERT-Base, Uncased and seven more models with trained weights released by the original BERT authors.
- Small BERTs have the same general architecture but fewer and/or smaller Transformer blocks, which lets you explore tradeoffs between speed, size and quality.
- ALBERT: four different sizes of "A Lite BERT" that reduces model size (but not computation time) by sharing parameters between layers.
- BERT Experts: eight models that all have the BERT-base architecture but offer a choice between different pre-training domains, to align more closely with the target task.

- Electra has the same architecture as BERT (in three different sizes), but gets pre-trained as a discriminator in a set-up that resembles a Generative Adversarial Network (GAN).
- BERT with Talking-Heads Attention and Gated GELU [base, large] has two improvements to the core of the Transformer architecture.

The model documentation on TensorFlow Hub has more details and references to the research literature. Follow the links above, or click on the `tfhub.dev` URL printed after the next cell execution.

The suggestion is to start with a Small BERT (with fewer parameters) since they are faster to fine-tune. If you like a small model but with higher accuracy, ALBERT might be your next option. If you want even better accuracy, choose one of the classic BERT sizes or their recent refinements like Electra, Talking Heads, or a BERT Expert.

Aside from the models available below, there are multiple versions of the models that are larger and can yield even better accuracy, but they are too big to be fine-tuned on a single GPU. You will be able to do that on the Solve GLUE tasks using BERT on a TPU colab.

You'll see in the code below that switching the tfhub.dev URL is enough to try any of these models, because all the differences between them are encapsulated in the SavedModels from TF Hub.

```
In [8]: #@title Choose a BERT model to fine-tune

        bert_model_name = 'small_bert/bert_en_uncased_L-2_H-25

        map_name_to_handle = {
            'bert_en_uncased_L-12_H-768_A-12':
                'https://tfhub.dev/tensorflow/bert_en_uncased_
            'bert_en_cased_L-12_H-768_A-12':
                'https://tfhub.dev/tensorflow/bert_en_cased_L-
            'bert_multi_cased_L-12_H-768_A-12':
                'https://tfhub.dev/tensorflow/bert_multi_cased
            'small_bert/bert_en_uncased_L-2_H-128_A-2':
                'https://tfhub.dev/tensorflow/small_bert/bert_
            'small_bert/bert_en_uncased_L-2_H-256_A-4':
                'https://tfhub.dev/tensorflow/small_bert/bert_
            'small_bert/bert_en_uncased_L-2_H-512_A-8':
                'https://tfhub.dev/tensorflow/small_bert/bert_
            'small_bert/bert_en_uncased_L-2_H-768_A-12':
                'https://tfhub.dev/tensorflow/small_bert/bert_
```

```
    'small_bert/bert_en_uncased_L-4_H-128_A-2':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-4_H-256_A-4':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-4_H-512_A-8':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-4_H-768_A-12':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-6_H-128_A-2':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-6_H-256_A-4':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-6_H-512_A-8':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-6_H-768_A-12':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-8_H-128_A-2':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-8_H-256_A-4':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-8_H-512_A-8':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-8_H-768_A-12':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-10_H-128_A-2':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-10_H-256_A-4':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-10_H-512_A-8':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-10_H-768_A-12':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-12_H-128_A-2':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-12_H-256_A-4':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-12_H-512_A-8':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'small_bert/bert_en_uncased_L-12_H-768_A-12':
        'https://tfhub.dev/tensorflow/small_bert/bert_
    'albert_en_base':
        'https://tfhub.dev/tensorflow/albert_en_base/2
    'electra_small':
        'https://tfhub.dev/google/electra_small/2',
    'electra_base':
        'https://tfhub.dev/google/electra_base/2',
    'experts_pubmed':
        'https://tfhub.dev/google/experts/bert/pubmed/
    'experts_wiki_books':
        'https://tfhub.dev/google/experts/bert/wiki_bo
    'talking-heads_base':
        'https://tfhub.dev/tensorflow/talkheads_ggelu_
}

map_model_to_preprocess = {
    'bert_en_uncased_L-12_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'bert_en_cased_L-12_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_cased_pr
    'small_bert/bert_en_uncased_L-2_H-128_A-2':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-2_H-256_A-4':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-2_H-512_A-8':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-2_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-4_H-128_A-2':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-4_H-256_A-4':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-4_H-512_A-8':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-4_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-6_H-128_A-2':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-6_H-256_A-4':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-6_H-512_A-8':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-6_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-8_H-128_A-2':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-8_H-256_A-4':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-8_H-512_A-8':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-8_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-10_H-128_A-2':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-10_H-256_A-4':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-10_H-512_A-8':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-10_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-12_H-128_A-2':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-12_H-256_A-4':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-12_H-512_A-8':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'small_bert/bert_en_uncased_L-12_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'bert_multi_cased_L-12_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_multi_cased
    'albert_en_base':
        'https://tfhub.dev/tensorflow/albert_en_prepro
    'electra_small':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'electra_base':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'experts_pubmed':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'experts_wiki_books':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
    'talking-heads_base':
        'https://tfhub.dev/tensorflow/bert_en_uncased_
}

tfhub_handle_encoder = map_name_to_handle[bert_model_n
tfhub_handle_preprocess = map_model_to_preprocess[bert

print(f'BERT model selected           : {tfhub_handle_
print(f'Preprocess model auto-selected: {tfhub_handle_
```

```
BERT model selected           : https://tfhub.dev/tenso
rflow/small_bert/bert_en_uncased_L-2_H-256_A-4/1
Preprocess model auto-selected: https://tfhub.dev/tenso
rflow/bert_en_uncased_preprocess/3
```

The preprocessing model

Text inputs need to be transformed to numeric token ids and arranged in several Tensors before being input to

BERT. TensorFlow Hub provides a matching preprocessing model for each of the BERT models discussed above, which implements this transformation using TF ops from the TF.text library. It is not necessary to run pure Python code outside your TensorFlow model to preprocess text.

The preprocessing model must be the one referenced by the documentation of the BERT model, which you can read at the URL printed above. For BERT models from the drop-down above, the preprocessing model is selected automatically.

Note: You will load the preprocessing model into a hub.KerasLayer to compose your fine-tuned model. This is the preferred API to load a TF2-style SavedModel from TF Hub into a Keras model.

```
In [9]: bert_preprocess_model = hub.KerasLayer(tfhub_handle_pr
```

Let's try the preprocessing model on some text and see the output:

```
In [10]: text_test = ['this is such an amazing movie!']
         text_preprocessed = bert_preprocess_model(text_test)

         print(f'Keys       : {list(text_preprocessed.keys())}'
         print(f'Shape      : {text_preprocessed["input_word_id
         print(f'Word Ids   : {text_preprocessed["input_word_id
         print(f'Input Mask : {text_preprocessed["input_mask"][
         print(f'Type Ids   : {text_preprocessed["input_type_id
Keys       : ['input_type_ids', 'input_mask', 'input_wo
rd_ids']
Shape      : (1, 128)
Word Ids   : [ 101 2023 2003 2107 2019 6429 3185  999
102    0    0    0]
Input Mask : [1 1 1 1 1 1 1 1 1 0 0 0]
Type Ids   : [0 0 0 0 0 0 0 0 0 0 0 0]
```

As you can see, now you have the 3 outputs from the preprocessing that a BERT model would use (`input_words_id`, `input_mask` and `input_type_ids`).

Some other important points:

- The input is truncated to 128 tokens. The number of tokens can be customized, and you can see more details on the Solve GLUE tasks using BERT on a TPU colab.

- The `input_type_ids` only have one value (0) because this is a single sentence input. For a multiple sentence input, it would have one number for each input.

Since this text preprocessor is a TensorFlow model, It can be included in your model directly.

Compute average length of test-set movie reviews

In [11]:
```
import numpy as np
```

```python
num_tokens_per_review = []
for text_batch, _ in test_ds:
  text_batch = text_batch.numpy()
  text_batch_preprocessed = bert_preprocess_model(text
  num_tokens = np.sum(text_batch_preprocessed["input_m
  num_tokens_per_review.extend(num_tokens)

total_reviews = len(num_tokens_per_review)
total_tokens = np.sum(num_tokens_per_review)
average_tokens_per_review = total_tokens/total_reviews
print('Average Tokens per Review: ', average_tokens_pe
print('Total Number of Reviews: ', total_reviews)
print('Total Number of Tokens: ', total_tokens)
```

```
Average Tokens per Review:  123.3642
Total Number of Reviews:  25000
Total Number of Tokens:  3084105
```

Using the BERT model

Before putting BERT into your own model, let's take a look at its outputs. You will load it from TF Hub and see the returned values.

```
In [12]: bert_model = hub.KerasLayer(tfhub_handle_encoder)
```

```
In [13]: bert_results = bert_model(text_preprocessed)

         print(f'Loaded BERT: {tfhub_handle_encoder}')
         print(f'Pooled Outputs Shape:{bert_results["pooled_out
         print(f'Pooled Outputs Values:{bert_results["pooled_ou
         print(f'Sequence Outputs Shape:{bert_results["sequence
         print(f'Sequence Outputs Values:{bert_results["sequenc
Loaded BERT: https://tfhub.dev/tensorflow/small_bert/be
rt_en_uncased_L-2_H-256_A-4/1
Pooled Outputs Shape:(1, 256)
Pooled Outputs Values:[ 0.9923971  -0.05357223  0.30918
494  0.95055723 -0.05440642  0.9988622
 -0.9947892  -0.00854006 -0.4551034  -0.08636718  0.832
0389  -0.92552674]
Sequence Outputs Shape:(1, 128, 256)
Sequence Outputs Values:[[-0.96845376 -1.7037114   1.00
94744  ...  0.84075063  0.35471338
  -1.828811  ]
 [-0.73150194 -0.9106685   0.30870232 ...  1.2626388
0.18190673
  -0.87057906]
 [-1.1151147  -1.8587102   1.9688951  ...  0.4747221
1.1492229
  -0.36351895]
 ...
 [-1.0249392  -1.769519    0.36316878 ...  0.35074914
0.18356118
```

```
  -0.67655814]
 [-0.47082692 -2.0506608   0.25583574 ...  0.8501367
0.18506233
  -0.6660278 ]
 [-0.14080206 -1.8676226   0.6415801  ...  0.2583237
0.04616926
  -0.14218208]]
```

The BERT models return a map with 3 important keys: `pooled_output`, `sequence_output`, `encoder_outputs`:

- `pooled_output` represents each input sequence as a whole. The shape is `[batch_size, H]`. You can think of this as an embedding for the entire movie review.
- `sequence_output` represents each input token in the context. The shape is `[batch_size, seq_length, H]`. You can think of this as a

contextual embedding for every token in the movie review.

- `encoder_outputs` are the intermediate activations of the `L` Transformer blocks. `outputs["encoder_outputs"][i]` is a Tensor of shape `[batch_size, seq_length, 1024]` with the outputs of the i-th Transformer block, for `0 <= i < L`. The last value of the list is equal to `sequence_output`.

For the fine-tuning you are going to use the `pooled_output` array.

Define your model

You will create a very simple fine-tuned model, with the preprocessing model, the selected BERT model, one Dense and a Dropout layer.

Note: for more information about the base model's input and output you can follow the model's URL for documentation. Here specifically, you don't need to worry about it because the preprocessing model will take care of that for you.

In [14]:
```
def build_classifier_model():
  text_input = tf.keras.layers.Input(shape=(), dtype=t
  preprocessing_layer = hub.KerasLayer(tfhub_handle_pr
  encoder_inputs = preprocessing_layer(text_input)
  encoder = hub.KerasLayer(tfhub_handle_encoder, train
  outputs = encoder(encoder_inputs)
  net = outputs['pooled_output']
  net = tf.keras.layers.Dropout(0.1)(net)
  net = tf.keras.layers.Dense(1, activation=None, name
  return tf.keras.Model(text_input, net)
```

Let's check that the model runs with the output of the preprocessing model.

```
In [15]: classifier_model = build_classifier_model()
         bert_raw_result = classifier_model(tf.constant(text_te
         print(tf.sigmoid(bert_raw_result))
```

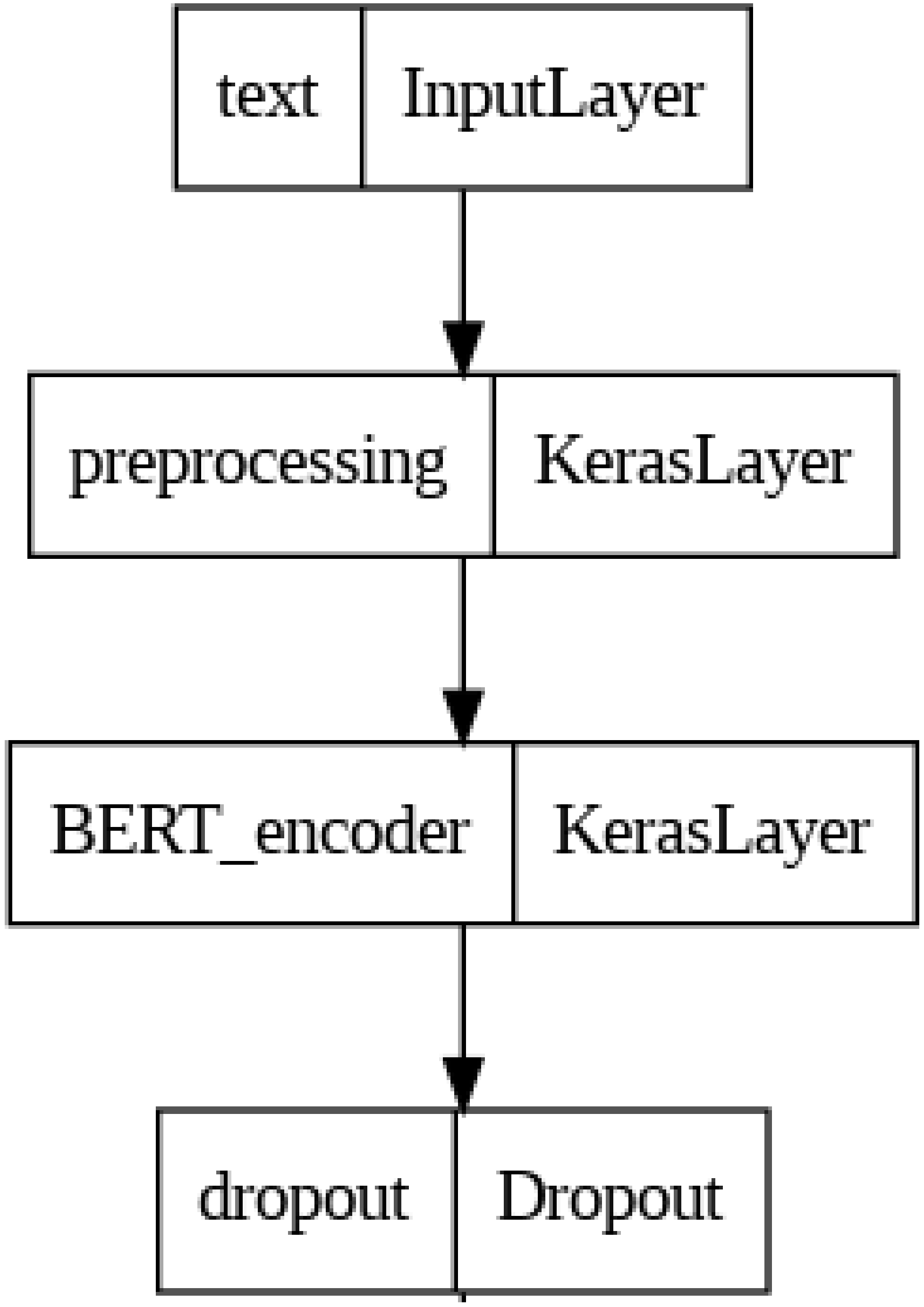
```
tf.Tensor([[0.24386759]], shape=(1, 1), dtype=float32)
```

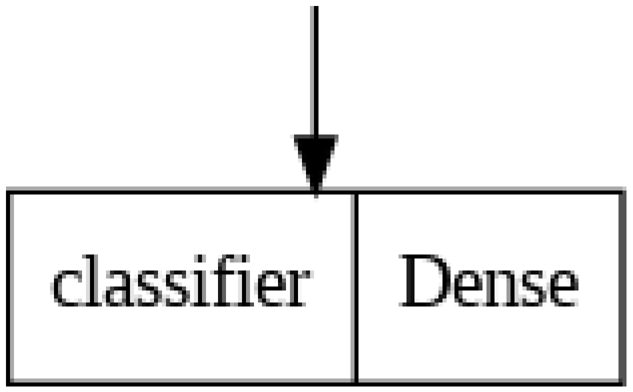
The output is meaningless, of course, because the model has not been trained yet.

Let's take a look at the model's structure.

```
In [16]: tf.keras.utils.plot_model(classifier_model)
```

Out[16]:


text
InputLayer
preprocessing
KerasLayer
BERT_encoder
KerasLayer
dropout
Dropout



Model training

You now have all the pieces to train a model, including the preprocessing module, BERT encoder, data, and classifier.

Loss function

Since this is a binary classification problem and the model outputs a probability (a single-unit layer), you'll use `losses.BinaryCrossentropy` loss function.

In [17]:
```
loss = tf.keras.losses.BinaryCrossentropy(from_logits=
metrics = tf.metrics.BinaryAccuracy()
```

Optimizer

For fine-tuning, let's use the same optimizer that BERT was originally trained with: the "Adaptive Moments" (Adam). This optimizer minimizes the prediction loss and does regularization by weight decay (not using moments), which is also known as AdamW.

For the learning rate (`init_lr`), you will use the same schedule as BERT pre-training: linear decay of a notional initial learning rate, prefixed with a linear warm-up phase over the first 10% of training steps (`num_warmup_steps`). In line with the BERT paper, the initial learning rate is smaller for fine-tuning (best of 5e-5, 3e-5, 2e-5).

```
In [18]: epochs = 5
         steps_per_epoch = tf.data.experimental.cardinality(tra
         num_train_steps = steps_per_epoch * epochs
         num_warmup_steps = int(0.1*num_train_steps)

         init_lr = 3e-5
         optimizer = optimization.create_optimizer(init_lr=init
                                                   num_train_st
                                                   num_warmup_s
                                                   optimizer_ty
```

Loading the BERT model and training

Using the `classifier_model` you created earlier, you can compile the model with the loss, metric and optimizer.

```
In [19]: classifier_model.compile(optimizer=optimizer,
                                  loss=loss,
                                  metrics=metrics)
```

Note: training time will vary depending on the complexity of the BERT model you have selected.

```
In [20]: print(f'Training model with {tfhub_handle_encoder}')
         history = classifier_model.fit(x=train_ds,
                                        validation_data=val_ds,
                                        epochs=epochs)
```

```
Training model with https://tfhub.dev/tensorflow/small_
bert/bert_en_uncased_L-2_H-256_A-4/1
Epoch 1/5
625/625 [==============================] - 644s 1s/step
- loss: 0.5691 - binary_accuracy: 0.6815 - val_loss: 0.
4415 - val_binary_accuracy: 0.7876
Epoch 2/5
625/625 [==============================] - 644s 1s/step
- loss: 0.4221 - binary_accuracy: 0.7940 - val_loss: 0.
4243 - val_binary_accuracy: 0.8108
Epoch 3/5
625/625 [==============================] - 643s 1s/step
- loss: 0.3765 - binary_accuracy: 0.8261 - val_loss: 0.
4136 - val_binary_accuracy: 0.8196
Epoch 4/5
625/625 [==============================] - 638s 1s/step
- loss: 0.3477 - binary_accuracy: 0.8446 - val_loss: 0.
4219 - val_binary_accuracy: 0.8196
Epoch 5/5
625/625 [==============================] - 642s 1s/step
```

```
- loss: 0.3259 - binary_accuracy: 0.8540 - val_loss: 0.
4224 - val_binary_accuracy: 0.8130
```

Evaluate the model

Let's see how the model performs. Two values will be returned. Loss (a number which represents the error, lower values are better), and accuracy.

```
In [21]: loss, accuracy = classifier_model.evaluate(test_ds)

         print(f'Loss: {loss}')
         print(f'Accuracy: {accuracy}')
```

```
782/782 [==============================] - 196s 251ms/s
tep - loss: 0.4155 - binary_accuracy: 0.8131
Loss: 0.4154647290706345
Accuracy: 0.8131200075149536
```

Plot the accuracy and loss over time

Based on the `History` object returned by `model.fit()`. You can plot the training and validation loss for comparison, as well as the training and validation accuracy:

In [22]:
```
history_dict = history.history
print(history_dict.keys())

acc = history_dict['binary_accuracy']
val_acc = history_dict['val_binary_accuracy']
loss = history_dict['loss']
val_loss = history_dict['val_loss']

epochs = range(1, len(acc) + 1)
fig = plt.figure(figsize=(10, 6))
```

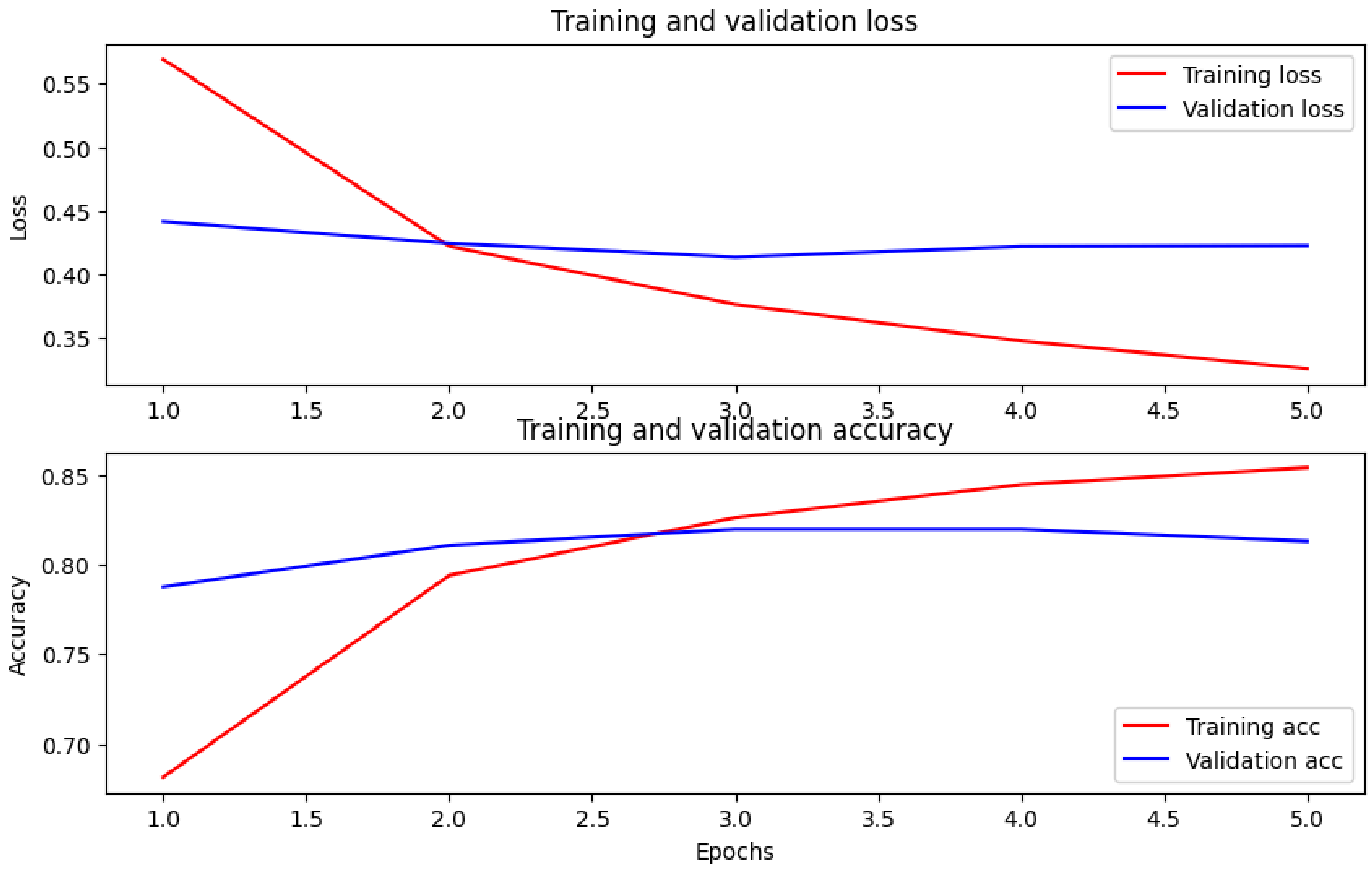
```python
fig.tight_layout()

plt.subplot(2, 1, 1)
# r is for "solid red line"
plt.plot(epochs, loss, 'r', label='Training loss')
# b is for "solid blue line"
plt.plot(epochs, val_loss, 'b', label='Validation loss
plt.title('Training and validation loss')
# plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.subplot(2, 1, 2)
plt.plot(epochs, acc, 'r', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend(loc='lower right')
```

```
dict_keys(['loss', 'binary_accuracy', 'val_loss', 'val_
binary_accuracy'])
```

```
Out[22]:  <matplotlib.legend.Legend at 0x7ebee045d950>
```



In this plot, the red lines represent the training loss and accuracy, and the blue lines are the validation loss and accuracy.

Export for inference

Now you just save your fine-tuned model for later use.

In [23]:
```
dataset_name = 'imdb'
saved_model_path = './{}_bert'.format(dataset_name.rep

classifier_model.save(saved_model_path, include_optimi
```

Let's reload the model, so you can try it side by side with the model that is still in memory.

In [24]:
```
reloaded_model = tf.saved_model.load(saved_model_path)
```

Here you can test your model on any sentence you want, just add to the examples variable below.

In [25]:
```
def print_my_examples(inputs, results):
  result_for_printing = \
    [f'input: {inputs[i]:<30} : score: {results[i][0]:
                         for i in range(len(inputs))]
  print(*result_for_printing, sep='\n')
  print()


examples = [
    'this is such an amazing movie!',  # this is the s
    'The movie was great!',
    'The movie was meh.',
    'The movie was okish.',
    'The movie was terrible...'
]

reloaded_results = tf.sigmoid(reloaded_model(tf.consta
original_results = tf.sigmoid(classifier_model(tf.cons

print('Results from the saved model:')
print_my_examples(examples, reloaded_results)
print('Results from the model in memory:')
print_my_examples(examples, original_results)
```

```
Results from the saved model:
input: this is such an amazing movie! : score: 0.994106
input: The movie was great!           : score: 0.922373
input: The movie was meh.             : score: 0.598352
input: The movie was okish.           : score: 0.048242
input: The movie was terrible...      : score: 0.008145

Results from the model in memory:
input: this is such an amazing movie! : score: 0.994106
input: The movie was great!           : score: 0.922373
input: The movie was meh.             : score: 0.598352
input: The movie was okish.           : score: 0.048242
input: The movie was terrible...      : score: 0.008145
```

If you want to use your model on TF Serving, remember that it will call your SavedModel through one of its named signatures. In Python, you can test them as follows:

```
In [26]: serving_results = reloaded_model \
                     .signatures['serving_default'](tf.constant

         serving_results = tf.sigmoid(serving_results['classifi

         print_my_examples(examples, serving_results)
```

```
input: this is such an amazing movie! : score: 0.994106
input: The movie was great!           : score: 0.922373
input: The movie was meh.             : score: 0.598352
input: The movie was okish.           : score: 0.048242
input: The movie was terrible...      : score: 0.008145
```

Next steps

As a next step, you can try Solve GLUE tasks using BERT on a TPU tutorial, which runs on a TPU and shows you how to work with multiple inputs.

Evaluate accuracy, precision, recall, F1

In [33]:
```
test_texts = []
test_labels = []

for text_batch, label_batch in test_ds:
  test_texts.extend(text_batch.numpy().tolist())
  test_labels.extend(label_batch.numpy().tolist())

test_labels = np.array(test_labels, dtype = int)

batch_size = 256
predictions = []
```

```
for i in range(0, len(test_texts), batch_size):
    batch_texts = tf.constant(test_texts[i:i + batch_s
    batch_predictions = tf.sigmoid(reloaded_model(batc
    predictions.extend(batch_predictions)

predictions = np.array(predictions)

threshold = 0.5
pred_labels = (predictions >= threshold).astype(int)

TP = 0
FP = 0
TN = 0
FN = 0

TP = np.sum((pred_labels == 1) & (test_labels == 1))
FP = np.sum((pred_labels == 1) & (test_labels == 0))
TN = np.sum((pred_labels == 0) & (test_labels == 0))
FN = np.sum((pred_labels == 0) & (test_labels == 1))
```

```python
accuracy = (TP + TN)/(TP + FP + TN + FN)
precision = TP/(TP + FP)
recall = TP/(TP + FN)
F1 =  2*precision*recall/(precision + recall)

print("Test Set Accuracy: ", accuracy)
print("Test Set Precision: ", precision)
print("Test Set Recall: ", recall)
print("Test Set F1-score: ", F1)
```

```
Test Set Accuracy:  0.8212
Test Set Precision:  0.8373382624768947
Test Set Recall:  0.79728
Test Set F1-score:  0.8168182935824934
```