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

```
     |████████████████████████████████| 4.3MB 6.3MB/s
     |████████████████████████████████| 454.3MB 34kB/s
     |████████████████████████████████| 4.0MB 13.4MB/s
     |████████████████████████████████| 6.0MB 45.4MB/s
     |████████████████████████████████| 1.2MB 29.2MB/s
     |████████████████████████████████| 4.0MB 25.8MB/s
     |████████████████████████████████| 471kB 56.1MB/s
     |████████████████████████████████| 4.9MB 41.3MB/s
```

In [2]:
```
# Using AdamW optimizer
!pip install -q tf-models-official==2.4
```

```
     |████████████████████████████████| 1.1MB 5.2MB/s
     |████████████████████████████████| 645kB 18.2MB/s
     |████████████████████████████████| 686kB 21.0MB/s
     |████████████████████████████████| 38.2MB 76kB/s
     |████████████████████████████████| 358kB 46.1MB/s
     |████████████████████████████████| 102kB 14.7MB/s
     |████████████████████████████████| 174kB 54.9MB/s
     |████████████████████████████████| 51kB 8.8MB/s
     |████████████████████████████████| 1.2MB 44.1MB/s
  Building wheel for py-cpuinfo (setup.py) ... done
  Building wheel for seqeval (setup.py) ... done
```

In [3]:
```
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 AdamW optimizer
import matplotlib.pyplot as plt
tf.get_logger().setLevel('ERROR')
```

In [4]:
```
url = 'https://github.com/ahlraf/point/blob/main/v2_emails.tar.gz?raw=true'
dataset = tf.keras.utils.get_file('v2_emails.tar.gz', url,
                                  untar=True, cache_dir='.',
                                  cache_subdir='')
```

```
Downloading data from https://github.com/ahlraf/point/blob/main/v2_emails.tar.gz?raw
=true
434176/429289 [==============================] - 0s 0us/step
```

In [5]:
```
AUTOTUNE = tf.data.AUTOTUNE
batch_size = 32
seed = 42

raw_train_ds = tf.keras.preprocessing.text_dataset_from_directory(
    'v2_emails/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=AUTOTUNE)

val_ds = tf.keras.preprocessing.text_dataset_from_directory(
    'v2_emails/train',
    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.preprocessing.text_dataset_from_directory(
    'v2_emails/test',
    batch_size=batch_size)

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

```
Found 1238 files belonging to 2 classes.
Using 991 files for training.
Found 1238 files belonging to 2 classes.
Using 247 files for validation.
Found 308 files belonging to 2 classes.
```

Preprocessing and looking at a few emails:

In [6]:
```
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

In [7]:
```
regex_tokenizer = nltk.RegexpTokenizer("\w+")
def text_preprocessing(content):
  content = str(content)
  content = re.sub("[^a-zA-Z]", " ", content)
  content = content.lower()
  content = content.encode("utf-8","ignore").decode()
  content = " ".join(regex_tokenizer.tokenize(content))
  for c in content:
    c.replace('\n',' ')
  words = content.split()
  stops = set(stopwords.words("english"))
  words = [w for w in words if not w in stops]

  return ' '.join(words)

train_2 = train_ds
for text_batch, label_batch in train_2:
  text_batch = text_preprocessing(text_batch)

for text_batch, label_batch in train_2.take(1):
  for i in range(10):
    print(f'Email: {text_batch.numpy()[i]}')
    label = label_batch.numpy()[i]
    print(f'Label: {label} ({class_names[label]})')
```

```
Email: b"Hi Ulf,Previous multi slot implementation was removed as nobody used it and
nobody tested it. There are lots of mistakes in previous implementationwhich are not
related to request serializationlike lack of slot switch / lack of adding slot id to
CIU commands / ets...So obviously it was never tested and never used at real multi s
lot hardware.In current implementation data transfers and commands to differenthosts
(slots) are serialized internally in the dw_mmc driver. We haverequest queue and whe
n .request() is called we add new request to thequeue. We take new request from the
queue only if the previous onehas already finished.So although hosts (slots) have se
parate locks (mmc_claim|release_host())the requests to different slots are serialize
d by driver.Isn't that enough?I'm not very familiar with SD/SDIO/(e)MMC specs so my
assumptions might be wrongin that case please correct me.Nevertheless we had to deal
somehow with existing hardware whichhas multislot dw mmc controller and both slots a
re used...This patch at least shouldn't break anything for current users (which usei
t in single slot mode)Moreover we tested this dual-slot implementation and don't cat
ch any problems(probably yet) except bus performance decrease in dual-slot mode (whi
ch isquite expected).-- Eugeniy Paltsev"
Label: 1 (technical)
Email: b"Hi, Eduardo,as it is late in this merge window, I'd prefer to1. drop all th
```

```
e thermal-soc material in the first pull request which Iwill send out soon.2. you ca
n prepare another pull request containing the thermal-socmaterials except the exynos
fixes3. exynos fixes with the problem solved can be queued for -rc2 orlater.thanks,r
ui"
Label: 1 (technical)
Email: b'Sure, it leaves the function to deal with the equiv table length onlyand th
e caller then adds the header length. Which is actually cleaner.--Regards/Gruss,
Boris.Good mailing practices for 400: avoid top-posting and trim the reply.'
Label: 1 (technical)
Email: b"I'm not sure I like this.  If you have a userspace application built agains
tmore recent uapi headers than the kernel you are actually running on, then bydefint
ion you won't have this check in place, and you'll get EINVAL returnsanyway.  If you
just backport this patch to an older kernel, you'll not get theEINVAL return, but yo
u will get silent failures on event subscriptions that yourapplication thinks exist
s, but the kernel doesn't recognize.This would make sense if you had a way to commun
icate back to user space theunrecognized options, but since we don't (currently) hav
e that, I would rathersee the EINVAL returned than just have things not work.Neil"
Label: 1 (technical)
Email: b'Hi,Yes that is fine by me and you\'ve my permission to switch to usingjust
the SPDX header.FWIW I do not believe the "can\'t be removed from \'this software an
dassociated documentation files (the "Software")\'" languageapplies to the software
as a whole and not individual files.Yes you may make the same change to all files wi
th my copyright.Regards,Hans'
Label: 1 (technical)
Email: b"The bug fix is to handle non-vmalloc pages.  I'll see if I can doa smaller
and more bandaid-y fix first."
Label: 1 (technical)
Email: b"I can't take patches without any changelog text at all :("
Label: 0 (nontechnical)
Email: b"Hi!Yes.No, I don't want that.I'd like 2 child nodes, each specifying which
HVLEDs it controls.Let me edit the original proposal.\t\t\t\t\t\t\t\tPavel--(engli
sh) http://www.livejournal.com/~pavelmachek(cesky, pictures) http://atrey.karlin.mf
f.cuni.cz/~pavel/picture/horses/blog.html"
Label: 1 (technical)
Email: b"I think the __KERNEL__ and asm/errno.h slip-ups are things Icargo-culted fr
om the arch code as a fresh-faced noob yet to learn thefiner details, so ack for tho
se parts. The forward-declarations, though,were a deliberate effort to minimise head
er dependencies and compilationbloat for includers who absolutely wouldn't care, and
specifically totry to avoid setting transitive include expectations since they alway
sseem to end up breaking someone's config somewhere down the line.Admittedly this li
ttle backwater is hardly comparable to the likes ofthe sched.h business, but I'm sti
ll somewhat on the fence about thatchange :/Robin."
Label: 1 (technical)
Email: b'Interesting \xe2\x80\xa6Would you like to share any more information from t
his meeting?I would appreciate further indications for a corresponding change accept
ance.I found a feedback by Mauro Carvalho Chehab more constructive.[GIT,PULL,FOR,v4.
15] Cleanup fixeshttps://patchwork.linuxtv.org/patch/43957/\xe2\x80\x9c\xe2\x80\xa6T
his time, I was nice and I took some time doing:\t$ quilt fold < `quilt next` && qui
lt delete `quilt next`\xe2\x80\xa6\xe2\x80Regards,Markus'
Label: 0 (nontechnical)
```

Choosing BERT model to fine-tune (BERT-small)

In [8]:

```
bert_model_name = 'small_bert/bert_en_uncased_L-4_H-512_A-8'

map_name_to_handle = {
    'bert_en_uncased_L-12_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/3',
    'bert_en_cased_L-12_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_cased_L-12_H-768_A-12/3',
    'bert_multi_cased_L-12_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_multi_cased_L-12_H-768_A-12/3',
    'small_bert/bert_en_uncased_L-2_H-128_A-2':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-2_H-128_A-2/1',
    'small_bert/bert_en_uncased_L-2_H-256_A-4':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-2_H-256_A-4/1',
    'small_bert/bert_en_uncased_L-2_H-512_A-8':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-2_H-512_A-8/1',
```

```
    'small_bert/bert_en_uncased_L-2_H-768_A-12':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-2_H-768_A-12/1',
    'small_bert/bert_en_uncased_L-4_H-128_A-2':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-128_A-2/1',
    'small_bert/bert_en_uncased_L-4_H-256_A-4':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-256_A-4/1',
    'small_bert/bert_en_uncased_L-4_H-512_A-8':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-512_A-8/1',
    'small_bert/bert_en_uncased_L-4_H-768_A-12':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-768_A-12/1',
    'small_bert/bert_en_uncased_L-6_H-128_A-2':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-6_H-128_A-2/1',
    'small_bert/bert_en_uncased_L-6_H-256_A-4':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-6_H-256_A-4/1',
    'small_bert/bert_en_uncased_L-6_H-512_A-8':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-6_H-512_A-8/1',
    'small_bert/bert_en_uncased_L-6_H-768_A-12':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-6_H-768_A-12/1',
    'small_bert/bert_en_uncased_L-8_H-128_A-2':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-8_H-128_A-2/1',
    'small_bert/bert_en_uncased_L-8_H-256_A-4':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-8_H-256_A-4/1',
    'small_bert/bert_en_uncased_L-8_H-512_A-8':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-8_H-512_A-8/1',
    'small_bert/bert_en_uncased_L-8_H-768_A-12':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-8_H-768_A-12/1',
    'small_bert/bert_en_uncased_L-10_H-128_A-2':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-10_H-128_A-2/1',
    'small_bert/bert_en_uncased_L-10_H-256_A-4':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-10_H-256_A-4/1',
    'small_bert/bert_en_uncased_L-10_H-512_A-8':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-10_H-512_A-8/1',
    'small_bert/bert_en_uncased_L-10_H-768_A-12':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-10_H-768_A-12/1',
    'small_bert/bert_en_uncased_L-12_H-128_A-2':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-12_H-128_A-2/1',
    'small_bert/bert_en_uncased_L-12_H-256_A-4':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-12_H-256_A-4/1',
    'small_bert/bert_en_uncased_L-12_H-512_A-8':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-12_H-512_A-8/1',
    'small_bert/bert_en_uncased_L-12_H-768_A-12':
        'https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-12_H-768_A-12/1',
    '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/2',
    'experts_wiki_books':
        'https://tfhub.dev/google/experts/bert/wiki_books/2',
    'talking-heads_base':
        'https://tfhub.dev/tensorflow/talkheads_ggelu_bert_en_base/1',
}

map_model_to_preprocess = {
    'bert_en_uncased_L-12_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'bert_en_cased_L-12_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_cased_preprocess/3',
    'small_bert/bert_en_uncased_L-2_H-128_A-2':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-2_H-256_A-4':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
```

```
    'small_bert/bert_en_uncased_L-2_H-512_A-8':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-2_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-4_H-128_A-2':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-4_H-256_A-4':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-4_H-512_A-8':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-4_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-6_H-128_A-2':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-6_H-256_A-4':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-6_H-512_A-8':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-6_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-8_H-128_A-2':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-8_H-256_A-4':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-8_H-512_A-8':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-8_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-10_H-128_A-2':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-10_H-256_A-4':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-10_H-512_A-8':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-10_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-12_H-128_A-2':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-12_H-256_A-4':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-12_H-512_A-8':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'small_bert/bert_en_uncased_L-12_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'bert_multi_cased_L-12_H-768_A-12':
        'https://tfhub.dev/tensorflow/bert_multi_cased_preprocess/3',
    'albert_en_base':
        'https://tfhub.dev/tensorflow/albert_en_preprocess/3',
    'electra_small':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'electra_base':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'experts_pubmed':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'experts_wiki_books':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
    'talking-heads_base':
        'https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3',
}

tfhub_handle_encoder = map_name_to_handle[bert_model_name]
tfhub_handle_preprocess = map_model_to_preprocess[bert_model_name]

print(f'BERT model selected           : {tfhub_handle_encoder}')
print(f'Preprocess model auto-selected: {tfhub_handle_preprocess}')
```

```
BERT model selected           : https://tfhub.dev/tensorflow/small_bert/bert_en_unca
sed_L-4_H-512_A-8/1
Preprocess model auto-selected: https://tfhub.dev/tensorflow/bert_en_uncased_preproc
ess/3
```

Preprocessing model

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

In [10]:
```
text_test = ["The driver is looking good!It looks like you've done some kind of revi
text_preprocessed = bert_preprocess_model(text_test)

print(f'Keys       : {list(text_preprocessed.keys())}')
print(f'Shape      : {text_preprocessed["input_word_ids"].shape}')
print(f'Word Ids   : {text_preprocessed["input_word_ids"][0, :12]}')
print(f'Input Mask : {text_preprocessed["input_mask"][0, :12]}')
print(f'Type Ids   : {text_preprocessed["input_type_ids"][0, :12]}')
```

```
Keys       : ['input_word_ids', 'input_mask', 'input_type_ids']
Shape      : (1, 128)
Word Ids   : [ 101 1996 4062 2003 2559 2204  999 2009 3504 2066 2017 1005]
Input Mask : [1 1 1 1 1 1 1 1 1 1 1 1]
Type Ids   : [0 0 0 0 0 0 0 0 0 0 0 0]
```

Using BERT model

In [11]:
```
bert_model = hub.KerasLayer(tfhub_handle_encoder)
```

In [12]:
```
bert_results = bert_model(text_preprocessed)

print(f'Loaded BERT: {tfhub_handle_encoder}')
print(f'Pooled Outputs Shape:{bert_results["pooled_output"].shape}')
print(f'Pooled Outputs Values:{bert_results["pooled_output"][0, :12]}')
print(f'Sequence Outputs Shape:{bert_results["sequence_output"].shape}')
print(f'Sequence Outputs Values:{bert_results["sequence_output"][0, :12]}')
```

```
Loaded BERT: https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-512_A-8/1
Pooled Outputs Shape:(1, 512)
Pooled Outputs Values:[ 0.84747237  0.9954414  -0.28012952  0.12758777  0.31347865
0.9054933
  0.51660377 -0.99680704 -0.0568257  -0.99887776  0.14184111 -0.98870677]
Sequence Outputs Shape:(1, 128, 512)
Sequence Outputs Values:[[ 0.39939746 -0.39085412  0.93853116 ...  0.2800356   0.033
86158
  -0.40618896]
 [-0.29228687  0.40331316 -1.0200574  ... -0.5753818   0.06500214
   0.86555773]
 [-0.8361566   0.07805394  0.6440221  ...  0.61097324  0.5496331
   0.5941888 ]
 ...
 [-0.31816947 -1.1716307  -1.4007772  ...  0.5933535  -0.5400041
  -0.59103024]
 [-0.40100175  0.1862402  -0.27396035 ...  0.6435042   0.38049674
   0.53075415]
 [-0.17033997  0.2594924   0.6192257  ... -0.47313097  0.6680391
   0.01982243]]
```

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]. [~ Embedding for the entire email] "sequence_output" represents each input token in the context. The shape is [batch_size, seq_length, H]. [~ contextual embedding for every token in the email] "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 we use the pooled_output array.

Defining the model

Fine-tuned model comprising preprocessing model + selected BERT model + 1 dense + 1 dropout layer

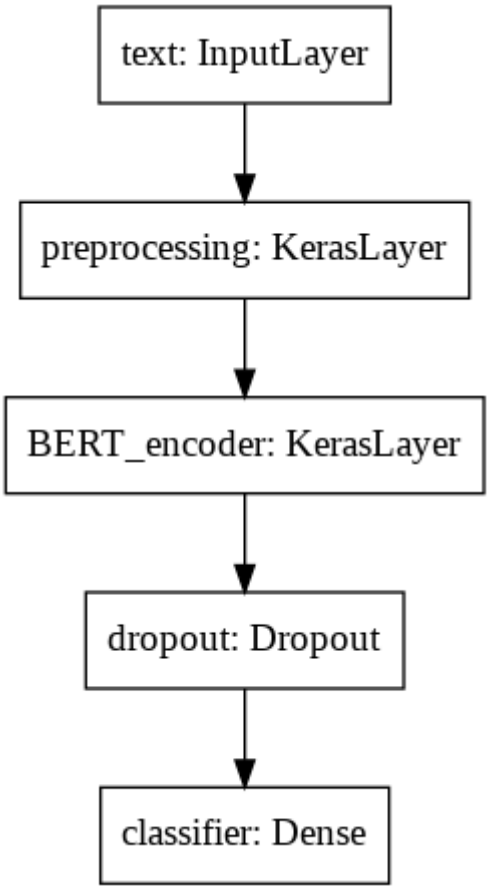
In [14]:
```
def build_classifier_model():
  text_input = tf.keras.layers.Input(shape=(), dtype=tf.string, name='text')
  preprocessing_layer = hub.KerasLayer(tfhub_handle_preprocess, name='preprocessing'
  encoder_inputs = preprocessing_layer(text_input)
  encoder = hub.KerasLayer(tfhub_handle_encoder, trainable=True, name='BERT_encoder'
  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='classifier')(net)
  return tf.keras.Model(text_input, net)
```

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

```
tf.Tensor([[0.5468524]], shape=(1, 1), dtype=float32)
```

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

Out[16]:



Model training

Loss function for binary classification

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

Optimizer: AdamW

For the learning rate (init_lr), we 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(train_ds).numpy()
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_lr,
                                          num_train_steps=num_train_steps,
                                          num_warmup_steps=num_warmup_steps,
                                          optimizer_type='adamw')
```

Loading BERT model and training

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

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-4_H-51
2_A-8/1
Epoch 1/5
31/31 [==============================] - 226s 7s/step - loss: 0.4785 - binary_accura
cy: 0.7326 - val_loss: 0.3047 - val_binary_accuracy: 0.9150
Epoch 2/5
31/31 [==============================] - 219s 7s/step - loss: 0.3326 - binary_accura
cy: 0.8860 - val_loss: 0.2732 - val_binary_accuracy: 0.9271
Epoch 3/5
31/31 [==============================] - 222s 7s/step - loss: 0.2998 - binary_accura
cy: 0.8900 - val_loss: 0.2589 - val_binary_accuracy: 0.9069
Epoch 4/5
31/31 [==============================] - 223s 7s/step - loss: 0.2772 - binary_accura
cy: 0.8961 - val_loss: 0.2513 - val_binary_accuracy: 0.8907
Epoch 5/5
31/31 [==============================] - 220s 7s/step - loss: 0.2505 - binary_accura
cy: 0.9072 - val_loss: 0.2323 - val_binary_accuracy: 0.9393
```

Evaluating model:

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

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

```
10/10 [==============================] - 18s 2s/step - loss: 0.2901 - binary_accurac
y: 0.8961
Loss: 0.29012614488601685
Accuracy: 0.8961039185523987
```

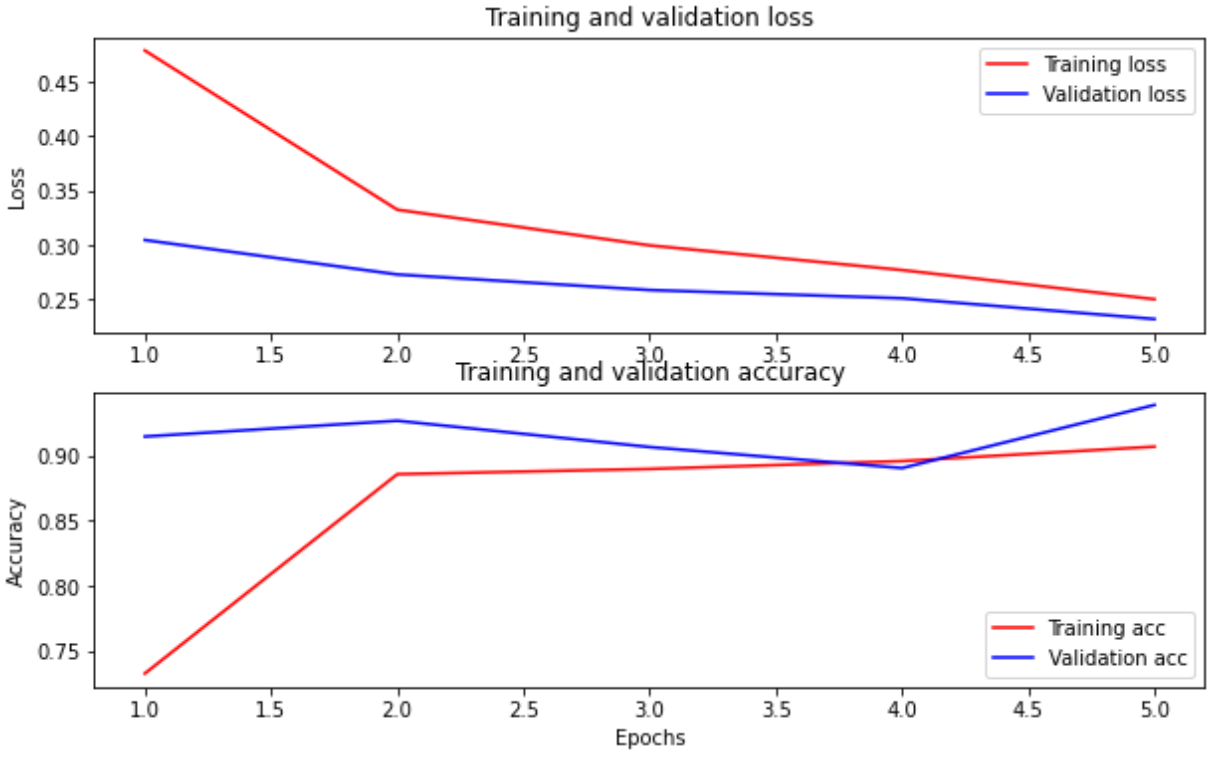
Plotting accuracy, loss over time:

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))
fig.tight_layout()

plt.subplot(2, 1, 1)
# "bo" is for "blue dot"
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 0x7f2075c63f10>
```



In [23]:
```
# testing:
examples = ["Was looking for that. Thanks. Speaking of that, recent lksctp-tools \
got some defines to help knowing which features the available kernel headers \
have as it now probes if specific struct members are available or not. \
Though yeah, it also wouldn't help in this case, just mentioning it.",
"Sorry but I don't like imposing a run-time check on everybody \
when stack-based requests are the odd ones out.  If we're going to make \
this a run-time check (I'd much prefer a compile-time check, but I \
understand that this may involve too much churn), then please do it \
```

```
for stack-based request users only.", "And I was just reminded about huge \
pages. But still, my point of finding a compromise still stands.", \
"Since when is the cover letter \
mandatory? I understand that is helps for a complicated patch set \
to explain the problem and solution in the cover letter, but for this \
simple test case addition what's the point? And there is nothing \
forcing a cover letter in", "I'm not exactly sure how Linux switch driver \
works, but from DT perspective I think we should rather have \
*hardware* described instead of a common Linux case. If I'm right, \
we should rather have all 3 switch ports described (5, 7,8) and have \
Linux just use the one it needs."]
# technical, non-technical, technical, non-technical, technical

def print_results(inputs, results):
  for i in range(len(inputs)):
    prediction = "Non-technical"
    if results[i][0]>=0.5:
      prediction = "Technical"
    print("Input:", inputs[i], "\nScore:", results[i][0], "\nPrediction:",prediction

results = tf.sigmoid(classifier_model(tf.constant(examples)))
print_results(examples, results)
```

```
Input: Was looking for that. Thanks. Speaking of that, recent lksctp-tools got some
defines to help knowing which features the available kernel headers have as it now p
robes if specific struct members are available or not. Though yeah, it also wouldn't
help in this case, just mentioning it.
Score: tf.Tensor(0.8919314, shape=(), dtype=float32)
Prediction: Technical
Input: Sorry but I don't like imposing a run-time check on everybody when stack-base
d requests are the odd ones out.  If we're going to make this a run-time check (I'd
much prefer a compile-time check, but I understand that this may involve too much ch
urn), then please do it for stack-based request users only.
Score: tf.Tensor(0.77189916, shape=(), dtype=float32)
Prediction: Technical
Input: And I was just reminded about huge pages. But still, my point of finding a co
mpromise still stands.
Score: tf.Tensor(0.9071373, shape=(), dtype=float32)
Prediction: Technical
Input: Since when is the cover letter mandatory? I understand that is helps for a co
mplicated patch set to explain the problem and solution in the cover letter, but for
this simple test case addition what's the point? And there is nothing forcing a cove
r letter in
Score: tf.Tensor(0.8546599, shape=(), dtype=float32)
Prediction: Technical
Input: I'm not exactly sure how Linux switch driver works, but from DT perspective I
think we should rather have *hardware* described instead of a common Linux case. If
I'm right, we should rather have all 3 switch ports described (5, 7,8) and have Linu
x just use the one it needs.
Score: tf.Tensor(0.96465874, shape=(), dtype=float32)
Prediction: Technical
```