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

```
     |████████████████████████████████| 4.3MB 2.1MB/s
     |████████████████████████████████| 454.3MB 39kB/s
     |████████████████████████████████| 471kB 37.7MB/s
     |████████████████████████████████| 6.0MB 31.0MB/s
     |████████████████████████████████| 4.0MB 33.2MB/s
     |████████████████████████████████| 4.0MB 22.8MB/s
     |████████████████████████████████| 1.2MB 30.5MB/s
     |████████████████████████████████| 4.9MB 30.8MB/s
```

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

```
     |████████████████████████████████| 1.1MB 2.8MB/s
     |████████████████████████████████| 38.2MB 71kB/s
     |████████████████████████████████| 102kB 10.0MB/s
     |████████████████████████████████| 51kB 6.3MB/s
     |████████████████████████████████| 358kB 34.0MB/s
     |████████████████████████████████| 645kB 30.8MB/s
     |████████████████████████████████| 174kB 42.1MB/s
     |████████████████████████████████| 686kB 34.8MB/s
     |████████████████████████████████| 1.2MB 37.4MB/s
  Building wheel for py-cpuinfo (setup.py) ... done
  Building wheel for seqeval (setup.py) ... done
```

In [4]:
```
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 [7]:
```
url = 'https://github.com/ahlraf/point/blob/main/civ_unciv_emails2.tar.gz?raw=true'
dataset = tf.keras.utils.get_file('civ_unciv_emails2.tar.gz', url,
                                  untar=True, cache_dir='.',
                                  cache_subdir='')
```

```
Downloading data from https://github.com/ahlraf/point/blob/main/civ_unciv_emails2.ta
r.gz?raw=true
65536/62597 [==============================] - 0s 0us/step
```

In [8]:
```
dataset_dir = os.path.join(os.path.dirname(dataset), 'civ_unciv_emails2')
train_dir = os.path.join(dataset_dir, 'train')
```

In [9]:
```
AUTOTUNE = tf.data.AUTOTUNE
batch_size = 32
seed = 42

raw_train_ds = tf.keras.preprocessing.text_dataset_from_directory(
    'civ_unciv_emails2/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(
    'civ_unciv_emails2/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(
    'civ_unciv_emails2/test',
    batch_size=batch_size)

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

```
Found 135 files belonging to 2 classes.
Using 108 files for training.
Found 135 files belonging to 2 classes.
Using 27 files for validation.
Found 33 files belonging to 2 classes.
```

Looking at and preprocessing emails:

In [10]:
```
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 [11]:
```
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"This is indeed guaranteed. For FTRACE use case. If it's being called from F
TRACE inrun time, this would mean there were long calls in this module section, whic
h inturn means, get_module_plt() was called at least once for this module and thisse
ction.This doesn't hold in general, though.In any case, if you insist, I can try to
rework the whole stuff implementing module_finalize().--Best regards,Alexander Sverd
lin."
Label: 0 (civil)
Email: b'Are you going to answer any of my remaining questions in a more constructiv
e way?Regards,Markus'
Label: 1 (uncivil)
Email: b'Michal Hocko wrote:Then, I am wondering why we are holding mmap_sem when ca
```

llingmigrate_pages() in existing code.http://elixir.free-electrons.com/linux/latest/source/mm/migrate.c#L1576Sorry, I missed that. If mmap_sem is not needed for migrate_pages(),please ignore this patch.--Best Regards,Yan Zi'
Label: 0 (civil)
Email: b"IMO symlinks are mostly ending in a mess, URLs are never stable.There is a https://www.kernel.org/doc/html/latest/objects.invto handle such requirements. Take a look at *intersphinx* : http://www.sphinx-doc.org/en/stable/ext/intersphinx.htmlto see how it works: Each Sphinx HTML build creates a file named objects.inv thatcontains a mapping from object names to URIs relative to the HTML set\xe2\x80\x99s root.This means articles from external (like lwn articles) has to be recompiled.Not perfect, but a first solution.I really like them, factually valuable comments .. pleaseexpress your concern so that we have a chance to move on.I think that's a pity.-- Markus --"
Label: 1 (uncivil)
Email: b"It would be very helpful if you cc all involved people on the cover letteriinstead of just cc'ing your own pile of email addresses. CC'ed now.This is really not helpful. The cover letter and the change logs shouldcontain a summary of that discussion and a proper justification of theproposed change. Just saying 'sysadmins might want to allow' is not usefulat all, it's yet another 'I want a pony' thing.I read through the previous thread and there was a clear request to involvesecurity people in to this. Especially those who are deeply involved withhardware side channels. I don't see anyone Cc'ed on the whole series.For the record, I'm not buying the handwavy 'more noise' argument atall. It wants a proper analysis and we need to come up with criteria whichPMUs can be exposed at all.All of this want's a proper documentation clearly explaining the risks andscope of these knobs per PMU. Just throwing magic knobs at sysadmins andthen saying 'its their problem to figure it out' is not acceptable.Thanks,\ttglx"
Label: 1 (uncivil)
Email: b'Thank you so much for many style, formatting and other issues fixes and also forintegration of \'check_at_most_once\' patch, it saved me several review iterations.Regarding free of sg in two error paths, you were correct.I fixed it by placing several error labels to differentiate each handling.I also noted that reqdata_arr[b].req was not released properly, this is also fixed.following is a diff of my fix based on your modifications.(I can send it in a patch format, but it doesn\'t include a fix for Eric Biggers comments)@@ -573,10 +573,9 @@ static void verity_verify_io(struct dm_verity_io *io)   verity_bv_skip_block(v, io, &io->iter);   continue;   }-   reqdata_arr[b].req = ahash_request_alloc(v->tfm, GFP_NOIO);   if (unlikely(reqdata_arr[b].req == NULL))-   goto err_memfree;+   goto err_mem_req;   ahash_request_set_tfm(reqdata_arr[b].req, v->tfm);/* +1 for the salt buffer */@@ -586,7 +585,7 @@ static void verity_verify_io(struct dm_verity_io *io)   GFP_NOIO);   if (!sg) {   DMERR_LIMIT("%s: kmalloc_array failed", __func__);-   goto err_memfree;+   goto err_mem_sg;   }   sg_init_table(sg, num_of_buffs);   // FIXME: if we \'err_memfree\' (or continue;) below how does this sg get kfree()\'d?@@ -595,7 +594,7 @@ static void verity_verify_io(struct dm_verity_io *io)   reqdata_arr[b].want_digest,   &reqdata_arr[b].fec_io, &is_zero);   if (unlikely(r < 0))-   goto err_memfree;+   goto err_mem;   if (is_zero) {/*@@ -605,7 +604,7 @@ static void verity_verify_io(struct dm_verity_io *io)r = verity_for_bv_block(v, io, &io->iter,verity_bv_zero);   if (unlikely(r < 0))-goto err_memfree;+   goto err_mem;verity_cb_complete(iodata, r);   continue;   }@@ -644,7 +643,11 @@ static void verity_verify_io(struct dm_verity_io *io)   }return;-err_memfree:+err_mem: kfree(sg);+err_mem_sg:+   ahash_request_free(reqdata_arr[b].req);+err_mem_req: /*   * reduce expected requests by the number of unsent   * requests, -1 accounting for the current blockatomic_sub(blocks - b - 1, &iodata->expected_reqs);   verity_cb_complete(iodata, -EIO);I took your modifications and working upon it.'
Label: 0 (civil)
Email: b'Hi!This is better than my proposal. Thanks!\t\t\t\t\t\t\t\tPavel--(english) http://www.livejournal.com/~pavelmachek(cesky, pictures) http://atrey.karlin.mff.cuni.cz/~pavel/picture/horses/blog.html'
Label: 0 (civil)
Email: b"I can't take patches without any changelog text at all :("
Label: 0 (civil)
Email: b"Ah only if google could simply answer all our questions!It's not like there

```
is or isn't a security risk and that youcan say that it is or it isn't in a global w
ay.Essentially these are channels of information. The channels always existin form o
f timing variances for any shared resource (like shared cachesor shared memory/IO/in
terconnect bandwidth) that can be measured.Perfmon counters make the channels genera
lly less noisy, but they do not causethem.To really close them completely you would
need to avoid sharinganything, or not allowing to measure time, neither of which is
practicalshort of an air gap.There are reasonable assesments you can make either way
and the answerswill be different based on your requirements. There isn't a singleans
wer that works for everyone.There are cases where it isn't a problem at all.If you d
on't have multiple users on the system your toleranceshould be extremely high.For us
ers who have multiple users there can be different tradeoffs.So there isn't a single
answer, and that is why it is importantthat this if configurable.-Andi"
Label: 1 (uncivil)
Email: b"On Tue, 30 Jan 2018 17:14:45 +0200Igor Stoppa <igor.stoppa@huawei.com> wrot
e:Please don't put plain-text files into core-api - that's a directory fullof RST do
cuments.  Your document is 99.9% RST already, better to justfinish the job and tie i
t into the rest of the kernel docs.We might as well put the SPDX tag here, it's a ne
w file.This is all good information, but I'd suggest it belongs more in the 0/npatch
posting than here.  The introduction of *this* document should saywhat it actually c
overs.This seems like a relevant and important aspect of the API that shouldn'tbe bu
ried in the middle of a section talking about random things.So one gets this far, bu
t has no actual idea of how to do these things.Which leads me to wonder: what is thi
s document for?  Who are you expectingto read it?You could improve things a lot by
(once again) going to RST and usingdirectives to bring in the kerneldoc comments fro
m the source (which, Inote, do exist).  But I'd suggest rethinking this document and
itsaudience.  Most of the people reading it are likely wanting to learn how to*use*
this API; I think it would be best to not leave them frustrated.Thanks,jon"
Label: 1 (uncivil)
```

In [12]:

```
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 [13]:
```
bert_preprocess_model = hub.KerasLayer(tfhub_handle_preprocess)
```

In [14]:
```
text_test = ["The driver is looking good!\n\nIt looks like you've done some kind of
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 [15]:
```
bert_model = hub.KerasLayer(tfhub_handle_encoder)
```

In [16]:
```
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]]
```

Defining model

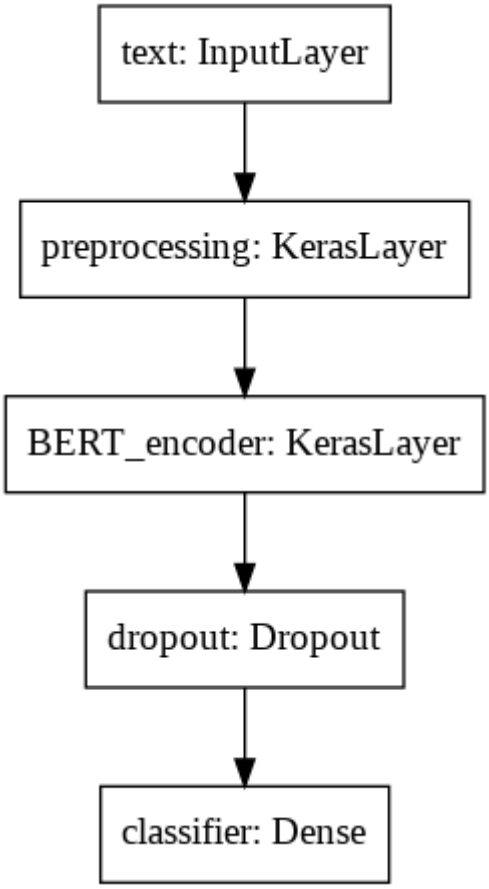
In [17]:
```
def build_classifier_model():
  text_input = tf.keras.layers.Input(shape=(), dtype=tf.string, name='text')
  preprocessing_layer = hub.KerasLayer(tfhub_handle_preprocess, name='preprocessing'
  encoder_inputs = preprocessing_layer(text_input)
  encoder = hub.KerasLayer(tfhub_handle_encoder, 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 [18]:
```
classifier_model = build_classifier_model()
bert_raw_result = classifier_model(tf.constant(text_test))
print(tf.sigmoid(bert_raw_result))
```

```
tf.Tensor([[0.6869482]], shape=(1, 1), dtype=float32)
```

In [19]:
```
tf.keras.utils.plot_model(classifier_model)
```

Out[19]:



Model training:

In [20]:
```
loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)
metrics = tf.metrics.BinaryAccuracy()
```

In [21]:
```
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')
```

In [22]:
```
classifier_model.compile(optimizer=optimizer,
                         loss=loss,
                         metrics=metrics)
```

In [23]:
```
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
4/4 [==============================] - 40s 8s/step - loss: 0.6847 - binary_accuracy:
0.5833 - val_loss: 0.6185 - val_binary_accuracy: 0.6296
Epoch 2/5
4/4 [==============================] - 33s 8s/step - loss: 0.6370 - binary_accuracy:
0.5556 - val_loss: 0.6272 - val_binary_accuracy: 0.5556
Epoch 3/5
4/4 [==============================] - 33s 8s/step - loss: 0.5698 - binary_accuracy:
0.7037 - val_loss: 0.6109 - val_binary_accuracy: 0.6667
```

```
Epoch 4/5
4/4 [==============================] - 32s 8s/step - loss: 0.5292 - binary_accuracy:
0.7315 - val_loss: 0.5996 - val_binary_accuracy: 0.6667
Epoch 5/5
4/4 [==============================] - 33s 8s/step - loss: 0.4939 - binary_accuracy:
0.8056 - val_loss: 0.5961 - val_binary_accuracy: 0.6667
```

Evaluating:

In [24]:
```
loss, accuracy = classifier_model.evaluate(test_ds)

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

```
2/2 [==============================] - 3s 109ms/step - loss: 0.5933 - binary_accurac
y: 0.6667
Loss: 0.5932566523551941
Accuracy: 0.6666666865348816
```

In [25]:
```
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[25]:
```
<matplotlib.legend.Legend at 0x7fcb136f2450>
```

In [26]:
```
# testing:
examples = ["I would really like to get an ack from the people who have been \
deep into this first.  If you can get that, and preferably resubmit with a \
less condescending changelog, I can pick it up.", "I like the idea and I think \
it's good direction to go, but could you please share some from perf stat or \
whatever you used to meassure the new performance?", "What's advertisement \
there? Huch? Care to tell what's a lie instead of making bold statements? \
Thanks, tglx", "There probably is a decent compromise to find between \
'not accepting a single additional byte' and accepting several GB. \
For example how likely is it that the growth of this structure make it \
go over a page? I would hope not at all. By choosing a large but decent \
high limit, I think we can find a future-compatible compromise that doesn't \
rely on a preliminary getsockopt() just for structure trucation decision..."]

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

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

```
Input: I would really like to get an ack from the people who have been deep into thi
s first.  If you can get that, and preferably resubmit with a less condescending cha
ngelog, I can pick it up.
Score: tf.Tensor(0.67575943, shape=(), dtype=float32)
Prediction: Civil
Input: I like the idea and I think it's good direction to go, but could you please s
hare some from perf stat or whatever you used to meassure the new performance?
Score: tf.Tensor(0.6564582, shape=(), dtype=float32)
Prediction: Civil
Input: What's advertisement there? Huch? Care to tell what's a lie instead of making
bold statements? Thanks, tglx
Score: tf.Tensor(0.5297446, shape=(), dtype=float32)
Prediction: Civil
Input: There probably is a decent compromise to find between 'not accepting a single
additional byte' and accepting several GB. For example how likely is it that the gro
wth of this structure make it go over a page? I would hope not at all. By choosing a
large but decent high limit, I think we can find a future-compatible compromise that
```

```
doesn't rely on a preliminary getsockopt() just for structure trucation decision...
Score: tf.Tensor(0.7734759, shape=(), dtype=float32)
Prediction: Civil
```