Raw data

原始数据由 instaloader 爬下, 存于文件夹`airpods_case_1`. 其中 `*_comments.json` 为所有评论数据.

通过脚本 `instagram_data_reader.py` 读取合并cleaning.

test data

该用户下(airpods_case_1)共有167条post, 总计1578条comments. 另存为文件 `airpods_case_1.txt`

其中, 1000以上是类似:

```
price?
price please
pp
how much
hi
@xxxxx
```

之类的言论, 移除后剩余512条comments另存为文件 `airpods_case_1_clean.txt`.

按长度排序, 选出的最长100条comments另存为文件 `airpods_case_1_100.txt`.

training data

IMDB dataset

`url = 'https://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz'`

该数据集为影评, 含25000 training data, 25000 test data. 均含有positive/negative 的label.

使用training data中20%作为validation data.

model

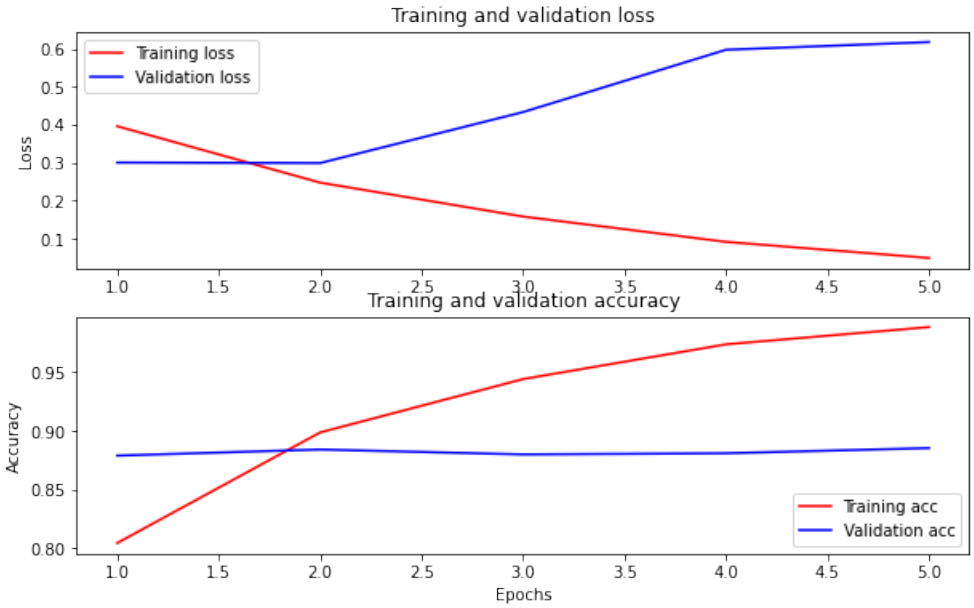
Albert作为embedding方法.

```
tfhub_handle_encoder = 'https://tfhub.dev/tensorflow/albert_en_base/2'
tfhub_handle_preprocess = 'https://tfhub.dev/tensorflow/albert_en_preprocess/3'

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)
```

training and test

> lr = 2e-5, epochs=5, batch_size=32



可看到在epoch=2时已拟合, 使用epoch=2时保存的参数对100条Instagram数据进行测试.

输出为Postive/Negative (0 or 1), 置信度(0~1), 原句. 以逗号分隔. 例:

```
0,0.444,How much is a supreme one on the second page?
0,0.184,Bro bro I want this but what was the price..
0,0.832,Hi what's the price and also how can I order
```

完整输出文件另存为 `airpods_case_1_100_output.txt`

训练部分代码 可在colab上查看并运行.