Sentiment classification using pre-trained language models (transformers)

Arabic Tweets Dataset (ASAD)

1. Prepare & Explore Dataset

In [1]:
```
# Read Dataset
import pandas as pd
df = pd.read_csv('tweets_ASAD_cleaned.csv', encoding='utf-8')
df = df[['Tweet_id','tweet_txt', 'sentiment']]
df_train = df.sample(frac=0.8, random_state=42)
df_test = df.drop(df_train.index)
```

In [2]:
```
"""
Handling imbalanced data
"""
from sklearn.utils import class_weight
import numpy as np
class_names= np.unique(df_train.sentiment.values)
class_weight = class_weight.compute_class_weight('balanced', class_names, df_train.sentiment.values)
class_weight_dict = dict(enumerate(class_weight))
```

```
D:\ANACONDA\lib\site-packages\sklearn\utils\validation.py:67: FutureWarning: Pass classes=['Negative' 'N
eutral' 'Positive'], y=['Positive' 'Neutral' 'Positive' ... 'Neutral' 'Neutral' 'Positive'] as keyword a
rgs. From version 0.25 passing these as positional arguments will result in an error
  warnings.warn("Pass {} as keyword args. From version 0.25 "
```

2. Define the neural network architecture

In [4]:
```
# build the model: using single LSTM
import ktrain
from ktrain import text
#https://huggingface.co/aubmindlab/bert-base-arabert
MODEL_NAME = 'aubmindlab/bert-base-arabertv02'

t = text.Transformer(MODEL_NAME, maxlen=128)
trn = t.preprocess_train(df_train.tweet_txt.values, df_train.sentiment.values)
val = t.preprocess_test(df_test.tweet_txt.values, df_test.sentiment.values)
model = t.get_classifier()
```

```
Downloading: 100%                                   384/384 [00:00<00:00, 4.26kB/s]


preprocessing train...
language: ar
train sequence lengths:
        mean : 14
        95percentile : 23
        99percentile : 26

Downloading: 100%                                   825k/825k [00:02<00:00, 314kB/s]


Downloading: 100%                                   2.64M/2.64M [00:10<00:00, 259kB/s]


Downloading: 100%                                   112/112 [00:05<00:00, 20.9B/s]


Downloading: 100%                                   381/381 [00:00<00:00, 927B/s]



Is Multi-Label? False
preprocessing test...
language: ar
test sequence lengths:
        mean : 14
        95percentile : 23
        99percentile : 26

Downloading: 100%                                   742M/742M [01:59<00:00, 6.19MB/s]
```

3. Compile the neural net

In [5]:
```
learner = ktrain.get_learner(model, train_data=trn, val_data=val, batch_size=64)
```

4. Fit / train the neural net

In [*]:
```
learner.fit_onecycle(lr= 5e-5, epochs= 1, class_weight=class_weight_dict)
learner.validate(class_names=t.get_classes())
```

```
begin training using onecycle policy with max lr of 5e-05...
 42/688 [>.............................] - ETA: 5:57:13 - loss: 1.0465 - accuracy: 0.3925
```

In [ ]:
```
#Save our Predictor for Later Deployment
p.save('arabic_tweet_predictor_salha_arabert')
```

5. Test our predictor

In [ ]:

```
p = ktrain.load_predictor('arabic_tweet_predictor_salha_arabert')
predicted_sentiment = p.predict("سعيد")
print(predicted_sentiment)
```