Quiz 12

# 1.

# a. Toxic Comment Classification

* Link to codes: <https://www.kaggle.com/tanulsingh077/deep-learning-for-nlp-zero-to-transformers-bert#BERT-and-Its-Implementation-on-this-Competition>

def build\_model(transformer, max\_len=512):

*"""*

*function for training the BERT model*

*"""*

input\_word\_ids = Input(shape=(max\_len,), dtype=tf.int32, name="input\_word\_ids")

sequence\_output = transformer(input\_word\_ids)[0]

cls\_token = sequence\_output[:, 0, :]

out = Dense(1, activation='sigmoid')(cls\_token)

model = Model(inputs=input\_word\_ids, outputs=out)

model.compile(Adam(lr=1e-5), loss='binary\_crossentropy', metrics=['accuracy'])

return model

%%time

with strategy.scope():

transformer\_layer = (

transformers.TFDistilBertModel

.from\_pretrained('distilbert-base-multilingual-cased')

)

model = build\_model(transformer\_layer, max\_len=MAX\_LEN)

model.summary()

* The code blocks above are to build state-of-the-art BERT model which is built upon Attention mechanism. Based on the notebook, BERT easily achieves 96% accuracy of classifying toxic comments.
* Overall, this approach is a time-saving approach but not easily applicable. BERT is pre-trained on a large text corpus which allows developers to deploy and enjoy the high-accuracy performance. The downside of BERT is its parameters size (105M parameters) that the inference latency (prediction latency) is high if the end-to-end BERT pipeline is not highly optimized. Also, instead of the last hidden state of the classification token, the author should use the last hidden state of the sequence that contains better word representation.

# b. News Recommendataion System

* Link: <https://www.kaggle.com/zackakil/nlp-using-word-vectors-with-spacy-cldspn#NLP-using-Word-Vectors-with-Spacy>

nlp = spacy.load('en\_core\_web\_lg')

nlp('dog').vector

from scipy.spatial.distance import cosine

NearestNeighbors(n\_neighbors=NUMBER\_OF\_NEIGHBORS)

distances, indices = neig\_model.kneighbors([YOUR\_VECTOR])

* The code snippets in the notebook are heavily fragmented and discontinued purposefully by the author. The entire notebook is to build the Medium article recommendation system. The code snippets I typed above show the usage of Spacy library to convert a word to a 300-dimension vector. Given the word vectors, the K-NN model is trained to choose articles that are similar to reading history of users.
* Overall, the approach is amazing in using Spacy to convert words to vectors is helpful in reducing the data dimensionality. The recommendation approach is the content-based one that based on the similarity between articles.

# 2.

* Dataset: BBC News Summary
* Link: <https://www.kaggle.com/pariza/bbc-news-summary>
* Challenge: Writing an interesting and attractive summary of an article would help draw readers’ attention and keep loyal to the publisher. Hence, in the NLP project, I attempt to work on Text Summarization, a challenging task to copy the human writing style to summarize news articles.