General libraries being loaded

In [1]:
```
# Python ≥3.5 is required
import sys
assert sys.version_info >= (3, 5)

# Scikit-Learn ≥0.20 is required
import sklearn
assert sklearn.__version__ >= "0.20"

# Common imports
import numpy as np
import os, time
import pandas as pd

# Our new Deep Learning imports
import tensorflow as tf
from tensorflow import keras

# To plot nice figures
# %matplotlib widget
%matplotlib inline

import matplotlib as mpl
import matplotlib.pyplot as plt
mpl.rc('axes', labelsize=14)
mpl.rc('xtick', labelsize=12)
mpl.rc('ytick', labelsize=12)

# For plotting statistical figures
import seaborn as sns; sns.set()

# For speeding up numpy operations
import cupy as cp

# For faster numpy computation
from numba import jit, cuda

# For Progress Bar
from tqdm.auto import tqdm,trange
tqdm.pandas()

# Vaex Dataframe library
import vaex as vx

# For Pyspark activation
import os
os.environ["PYARROW_IGNORE_TIMEZONE"] = "1"

# Pyspark Dataframe
from pyspark import pandas as ps

import os
os.environ['KMP_DUPLICATE_LIB_OK']='True'
```

Loading the Abstract Sentences for Doc2Vec Model (Gensim)

In [2]:
```
Article_Data_Cord_File_DF_Abstract = pd.read_pickle(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\cord_19_embeddings\article_data_cord_19_file_df_abstract.pkl")
```

In [3]:
```
Article_Data_Cord_File_DF_Abstract['Type'] = Article_Data_Cord_File_DF_Abstract.shape[0] * ['Abstract']
```

In [4]:
```
Article_Data_Cord_File_DF_Abstract.rename(columns = {'Abstract_Sentences':'Sentences'}, inplace = True)
```

Loading the Body Text Sentences for Doc2Vec Model (Gensim)

In [5]:
```
Article_Data_Cord_File_DF_Body_Text = pd.read_pickle(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\cord_19_embeddings\article_data_cord_19_file_df_body_text.pkl")
```

In [6]:
```
Article_Data_Cord_File_DF_Body_Text['Type'] = Article_Data_Cord_File_DF_Body_Text.shape[0] * ['Body Text']
```

In [7]:
```
Article_Data_Cord_File_DF_Body_Text.rename(columns = {'Body_Text_Sentences':'Sentences'}, inplace = True)
```

Loading Results of Questions from Round 1 to 5

In [8]:
```
Article_Data_Cord_File_DF_Result_1 = pd.read_csv(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\Kaggle\Round 1\qrels-covid_d1_j0.5-1.txt",sep=' ',header=None)
Article_Data_Cord_File_DF_Result_1['Batch'] = Article_Data_Cord_File_DF_Result_1.shape[0] * ['1']

Article_Data_Cord_File_DF_Result_2 = pd.read_csv(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\Kaggle\Round 2\qrels-covid_d2_j0.5-2.txt",sep=' ',header=None)
Article_Data_Cord_File_DF_Result_2['Batch'] = Article_Data_Cord_File_DF_Result_2.shape[0] * ['2']
Article_Data_Cord_File_DF_Result_2[0] = Article_Data_Cord_File_DF_Result_2[0] + Article_Data_Cord_File_DF_Result_1[0][Article_Data_Cord_File_DF_Result_1.shape[0]-1]

Article_Data_Cord_File_DF_Result_3 = pd.read_csv(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\Kaggle\Round 3\qrels-covid_d3_j0.5-3.txt",sep=' ',header=None)
Article_Data_Cord_File_DF_Result_3['Batch'] = Article_Data_Cord_File_DF_Result_3.shape[0] * ['3']
Article_Data_Cord_File_DF_Result_3[0] = Article_Data_Cord_File_DF_Result_3[0] + Article_Data_Cord_File_DF_Result_2[0][Article_Data_Cord_File_DF_Result_2.shape[0]-1]

Article_Data_Cord_File_DF_Result_4 = pd.read_csv(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\Kaggle\Round 4\qrels-covid_d4_j0.5-4.txt",sep=' ',header=None)
Article_Data_Cord_File_DF_Result_4['Batch'] = Article_Data_Cord_File_DF_Result_4.shape[0] * ['4']
Article_Data_Cord_File_DF_Result_4[0] = Article_Data_Cord_File_DF_Result_4[0] + Article_Data_Cord_File_DF_Result_3[0][Article_Data_Cord_File_DF_Result_3.shape[0]-1]

Article_Data_Cord_File_DF_Result_5 = pd.read_csv(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\Kaggle\Round 5\qrels-covid_d5_j0.5-5.txt",sep=' ',header=None)
Article_Data_Cord_File_DF_Result_5['Batch'] = Article_Data_Cord_File_DF_Result_5.shape[0] * ['5']
Article_Data_Cord_File_DF_Result_5[0] = Article_Data_Cord_File_DF_Result_5[0] + Article_Data_Cord_File_DF_Result_4[0][Article_Data_Cord_File_DF_Result_4.shape[0]-1]
```

In [9]:
```
Article_Data_Cord_File_DF_Result = pd.concat([Article_Data_Cord_File_DF_Result_1,Article_Data_Cord_File_DF_Result_2,Article_Data_Cord_File_DF_Result_3,Article_Data_Cord_File_DF_Result_4,Article_Data_Cord_File_DF_Result_5],ignore_index=True)
Article_Data_Cord_File_DF_Result[0] = Article_Data_Cord_File_DF_Result[0].progress_apply(str)
```

```
  0%|          | 0/178007 [00:00<?, ?it/s]
```

Loading the Round 1 Questions for Doc2Vec Model (Gensim)

In [10]:
```
Article_Data_Cord_File_DF_Ques_1 = pd.read_xml(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\Kaggle\Round 1\topics-rnd1.xml")
```

In [11]:
```
Article_Data_Cord_File_DF_Ques_1['batch'] = Article_Data_Cord_File_DF_Ques_1.shape[0] * ['1']
```

Loading the Round 2 Questions for Doc2Vec Model (Gensim)

In [12]:
```
Article_Data_Cord_File_DF_Ques_2 = pd.read_xml(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\Kaggle\Round 2\topics-rnd2.xml")
```

In [13]:
```
Article_Data_Cord_File_DF_Ques_2['batch'] = Article_Data_Cord_File_DF_Ques_2.shape[0] * ['2']
```

Loading the Round 3 Questions for Doc2Vec Model (Gensim)

In [14]:
```
Article_Data_Cord_File_DF_Ques_3 = pd.read_xml(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\Kaggle\Round 3\topics-rnd3.xml")
```

In [15]:
```
Article_Data_Cord_File_DF_Ques_3['batch'] = Article_Data_Cord_File_DF_Ques_3.shape[0] * ['3']
```

Loading the Round 4 Questions for Doc2Vec Model (Gensim)

In [16]:
```
Article_Data_Cord_File_DF_Ques_4 = pd.read_xml(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\Kaggle\Round 4\topics-rnd4.xml")
```

In [17]:
```
Article_Data_Cord_File_DF_Ques_4['batch'] = Article_Data_Cord_File_DF_Ques_4.shape[0] * ['4']
```

Loading the Round 5 Questions for Doc2Vec Model (Gensim)

In [18]:
```
Article_Data_Cord_File_DF_Ques_5 = pd.read_xml(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\Kaggle\Round 5\topics-rnd5.xml")
```

In [19]:
```
Article_Data_Cord_File_DF_Ques_5['batch'] = Article_Data_Cord_File_DF_Ques_5.shape[0] * ['5']
```

Concatenating the Round 1 to Round 5 Questions for Doc2Vec Model (Gensim)

In [20]:
```
Article_Data_Cord_File_DF_Ques = pd.concat([Article_Data_Cord_File_DF_Ques_1,Article_Data_Cord_File_DF_Ques_2,Article_Data_Cord_File_DF_Ques_3,Article_Data_Cord_File_DF_Ques_4,Article_Data_Cord_File_DF_Ques_5],ignore_index=True)
```

In [21]:
```
Article_Data_Cord_File_DF_Ques['number'] = list(range(1,Article_Data_Cord_File_DF_Ques.shape[0]+1))
```

In [22]:
```
Article_Data_Cord_File_DF_Ques_Use = (Article_Data_Cord_File_DF_Ques).copy()
Article_Data_Cord_File_DF_Ques_Use.rename(columns = {'number':'Doc_Id_Meta_Parse'}, inplace = True)
```

In [23]:
```
Article_Data_Cord_File_DF_Ques_Use_1 = Article_Data_Cord_File_DF_Ques_Use[['Doc_Id_Meta_Parse','query']].copy()
Article_Data_Cord_File_DF_Ques_Use_2 = Article_Data_Cord_File_DF_Ques_Use[['Doc_Id_Meta_Parse','question']].copy()
Article_Data_Cord_File_DF_Ques_Use_3 = Article_Data_Cord_File_DF_Ques_Use[['Doc_Id_Meta_Parse','narrative']].copy()

Article_Data_Cord_File_DF_Ques_Use_1.rename(columns = {'query':'Sentences'}, inplace = True)
Article_Data_Cord_File_DF_Ques_Use_2.rename(columns = {'question':'Sentences'}, inplace = True)
Article_Data_Cord_File_DF_Ques_Use_3.rename(columns = {'narrative':'Sentences'}, inplace = True)

Article_Data_Cord_File_DF_Ques_Use_1['Type'] = Article_Data_Cord_File_DF_Ques_Use_1.shape[0] * ['Query']
Article_Data_Cord_File_DF_Ques_Use_2['Type'] = Article_Data_Cord_File_DF_Ques_Use_2.shape[0] * ['Question']
Article_Data_Cord_File_DF_Ques_Use_3['Type'] = Article_Data_Cord_File_DF_Ques_Use_2.shape[0] * ['Narrative']
```

In [24]:
```
del Article_Data_Cord_File_DF_Ques_Use
Article_Data_Cord_File_DF_Ques_Use = pd.DataFrame()
```

In [25]:
```
Article_Data_Cord_File_DF_Ques_Use = pd.concat([Article_Data_Cord_File_DF_Ques_Use_1,Article_Data_Cord_File_DF_Ques_Use_2,Article_Data_Cord_File_DF_Ques_Use_3],ignore_index=True)
```

In [26]:
```
Article_Data_Cord_File_DF_Ques_Use['Doc_Id_Meta_Parse'] = Article_Data_Cord_File_DF_Ques_Use['Doc_Id_Meta_Parse'].progress_apply(str)
```

```
  0%|          | 0/600 [00:00<?, ?it/s]
```

Concatenating the Abstract, Body Text and Questions for Doc2Vec Model (Gensim)

In [27]:
```
Article_Data_Cord_File_DF_Abstract_Body_Text_Ques = pd.concat([Article_Data_Cord_File_DF_Abstract, Article_Data_Cord_File_DF_Body_Text, Article_Data_Cord_File_DF_Ques_Use], ignore_index=True)
```

In [28]:
```
Article_Data_Cord_File_List_Abstract_Body_Text_Ques_Sent = Article_Data_Cord_File_DF_Abstract_Body_Text_Ques['Sentences'].to_list()
Article_Data_Cord_File_List_Abstract_Body_Text_Ques_Tags = Article_Data_Cord_File_DF_Abstract_Body_Text_Ques['Doc_Id_Meta_Parse'].to_list()
```

Creating Tagged Documents for Doc2Vec Model (Gensim)

In [29]:
```
import gensim
```

In [30]:
```
k=0
for Article_Data_Cord_File_List_Abstract_Body_Text_Ques_Sent_Sub in tqdm(Article_Data_Cord_File_List_Abstract_Body_Text_Ques_Sent):
    Article_Data_Cord_File_List_Abstract_Body_Text_Ques_Sent[k] = gensim.models.doc2vec.TaggedDocument(Article_Data_Cord_File_List_Abstract_Body_Text_Ques_Sent_Sub, Article_Data_Cord_File_List_Abstract_Body_Text_Ques_Tags[k])
    k=k+1
```

```
  0%|          | 0/979097 [00:00<?, ?it/s]
```

In [31]:
```
Article_Data_Cord_File_DF_Abstract_Body_Text_Ques['Sentences_Tags'] = Article_Data_Cord_File_List_Abstract_Body_Text_Ques_Sent
```

Creating Doc2Vec Model and Training it (Gensim)

In [32]:
```
Doc2Vec_Model_Abstract_Body_Text_Ques = gensim.models.doc2vec.Doc2Vec(vector_size=100, min_count=2, epochs=20, workers=4)
```

In [33]:
```
Doc2Vec_Model_Abstract_Body_Text_Ques.build_vocab(Article_Data_Cord_File_DF_Abstract_Body_Text_Ques['Sentences_Tags'].to_list())
```

In [34]:
```
Doc2Vec_Model_Abstract_Body_Text_Ques.train(Article_Data_Cord_File_DF_Abstract_Body_Text_Ques['Sentences_Tags'].to_list(), total_examples=Doc2Vec_Model_Abstract_Body_Text_Ques.corpus_count, epochs=Doc2Vec_Model_Abstract_Body_Text_Ques.epochs)
```

Loading the Abstract Tokens for Doc2Vec Model Sentence Embeddings (Gensim)

In [35]:
```
Article_Data_Cord_File_DF_Abstract_Tokens = pd.read_pickle(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\cord_19_embeddings\article_data_cord_19_file_df_abstract_tokens.pkl")
```

In [36]:
```
Article_Data_Cord_File_DF_Abstract['Sentence Embeddings'] = Article_Data_Cord_File_DF_Abstract_Tokens['Abstract_Tokens'].progress_apply(Doc2Vec_Model_Abstract_Body_Text_Ques.infer_vector)
```

```
  0%|          | 0/122754 [00:00<?, ?it/s]
```

In [37]:
```
Article_Data_Cord_File_DF_Abstract.to_pickle(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\cord_19_embeddings\article_data_cord_19_file_df_abstract_sentence_embeddings_gensim.pkl")
```

Loading the Body Text Tokens for Doc2Vec Model Sentence Embeddings (Gensim)

In [38]:
```
Article_Data_Cord_File_DF_Body_Text_Tokens = pd.read_pickle(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\cord_19_embeddings\article_data_cord_19_file_df_body_text_tokens.pkl")
```

In [39]:
```
Article_Data_Cord_File_DF_Body_Text['Sentence Embeddings'] = Article_Data_Cord_File_DF_Body_Text_Tokens['Body_Text_Tokens'].progress_apply(Doc2Vec_Model_Abstract_Body_Text_Ques.infer_vector)
```

```
  0%|          | 0/855743 [00:00<?, ?it/s]
```

In [40]:
```
Article_Data_Cord_File_DF_Body_Text.to_pickle(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\cord_19_embeddings\article_data_cord_19_file_df_body_text_sentence_embeddings_gensim.pkl")
```

Creating Sentence Embeddings of Questions

In [41]:
```
Article_Data_Cord_File_DF_Ques.rename(columns = {'number':'Doc_Id_Meta_Parse','query':'Query','question':'Question','narrative':'Narrative','batch':'Batch'}, inplace = True)
```

In [42]:
```
import tensorflow as tf
```

In [43]:
```
Article_Data_Cord_File_DF_Ques_Tokens = pd.DataFrame()
Article_Data_Cord_File_DF_Ques_Tokens['Doc_Id_Meta_Parse'] = Article_Data_Cord_File_DF_Ques['Doc_Id_Meta_Parse'].to_list()
```

In [44]:
```
Article_Data_Cord_File_DF_Ques_Tokens['Query_Tokens'] = Article_Data_Cord_File_DF_Ques['Query'].progress_apply(tf.keras.preprocessing.text.text_to_word_sequence)
Article_Data_Cord_File_DF_Ques_Tokens['Question_Tokens'] = Article_Data_Cord_File_DF_Ques['Question'].progress_apply(tf.keras.preprocessing.text.text_to_word_sequence)
Article_Data_Cord_File_DF_Ques_Tokens['Narrative_Tokens'] = Article_Data_Cord_File_DF_Ques['Narrative'].progress_apply(tf.keras.preprocessing.text.text_to_word_sequence)
```

```
  0%|          | 0/200 [00:00<?, ?it/s]
  0%|          | 0/200 [00:00<?, ?it/s]
  0%|          | 0/200 [00:00<?, ?it/s]
```

In [45]:
```
Article_Data_Cord_File_DF_Ques_Tokens['Batch'] = Article_Data_Cord_File_DF_Ques['Batch'].to_list()
```

In [46]:
```
Article_Data_Cord_File_DF_Ques_Tokens.to_pickle(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\cord_19_embeddings\article_data_cord_19_file_df_ques_tokens.pkl")
```

In [47]:
```
Article_Data_Cord_File_DF_Ques.to_pickle(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\cord_19_embeddings\article_data_cord_19_file_df_ques.pkl")
```

In [48]:
```
Article_Data_Cord_File_DF_Ques['Query Sentence Embeddings'] = Article_Data_Cord_File_DF_Ques_Tokens['Query_Tokens'].progress_apply(Doc2Vec_Model_Abstract_Body_Text_Ques.infer_vector)
Article_Data_Cord_File_DF_Ques['Question Sentence Embeddings'] = Article_Data_Cord_File_DF_Ques_Tokens['Question_Tokens'].progress_apply(Doc2Vec_Model_Abstract_Body_Text_Ques.infer_vector)
Article_Data_Cord_File_DF_Ques['Narrative Sentence Embeddings'] = Article_Data_Cord_File_DF_Ques_Tokens['Narrative_Tokens'].progress_apply(Doc2Vec_Model_Abstract_Body_Text_Ques.infer_vector)
```

```
  0%|          | 0/200 [00:00<?, ?it/s]
  0%|          | 0/200 [00:00<?, ?it/s]
  0%|          | 0/200 [00:00<?, ?it/s]
```

In [49]:
```
def row_concatenate_embeddings(row_query,row_question,row_narrative):
    return np.concatenate((row_query,row_question,row_narrative),axis=0)
```

In [50]:
```
Article_Data_Cord_File_DF_Ques['Concatenated Sentence Embeddings'] = Article_Data_Cord_File_DF_Ques.progress_apply(lambda x: row_concatenate_embeddings(x['Query Sentence Embeddings'],x['Question Sentence Embeddings'],x['Narrative Sentence Embeddings']),axis=1)
```

```
  0%|          | 0/200 [00:00<?, ?it/s]
```

In [51]:
```
def row_mean_embeddings(row_query,row_question,row_narrative):
    return np.mean((row_query,row_question,row_narrative),axis=0)
```

In [52]:
```
Article_Data_Cord_File_DF_Ques['Mean Sentence Embeddings'] = Article_Data_Cord_File_DF_Ques.progress_apply(lambda x: row_mean_embeddings(x['Query Sentence Embeddings'],x['Question Sentence Embeddings'],x['Narrative Sentence Embeddings']),axis=1)
```

```
  0%|          | 0/200 [00:00<?, ?it/s]
```

In [53]:
```
Article_Data_Cord_File_DF_Abstract_Body_Text_Ques.to_pickle(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\cord_19_embeddings\article_data_cord_19_file_df_abstract_body_text_ques.pkl")
```

In [54]:
```
Article_Data_Cord_File_DF_Ques.to_pickle(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\cord_19_embeddings\article_data_cord_19_file_df_abstract_body_text_ques_sentence_embeddings_gensim.pkl")
```

Loading Cord-19 Sentence Embeddings

In [55]:
```
from gensim.models import KeyedVectors
Cord_19_Doc_Embeddings = KeyedVectors.load_word2vec_format(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\cord_19_embeddings\cord_19_embeddings_filtered.csv",no_header=True,binary=False)
```

In [56]:
```
Cord_19_Doc_Embeddings_DF = pd.DataFrame()
Cord_19_Doc_Embeddings_DF = Cord_19_Doc_Embeddings_DF.astype('object')

Cord_19_Doc_Embeddings_DF['Doc_Id_Meta_Parse'] = list(Cord_19_Doc_Embeddings.index_to_key)
Cord_19_Doc_Embeddings_DF['Doc_Id_Meta_Parse'] = Cord_19_Doc_Embeddings_DF['Doc_Id_Meta_Parse'].fillna('')
Cord_19_Doc_Embeddings_DF['Document Embeddings'] = list(Cord_19_Doc_Embeddings.vectors)
```

In [57]:
```
Cord_19_Doc_Embeddings_DF = Cord_19_Doc_Embeddings_DF.loc[~(Cord_19_Doc_Embeddings_DF['Doc_Id_Meta_Parse'] == ''),:].reset_index(drop=True)
```

Padding the Sentence Embeddings of Questions, Abstract and Body Text to have equal Dimension

In [58]:
```
Cord_19_Length_Concate_Ques = Cord_19_Doc_Embeddings_DF['Document Embeddings'][0].shape[0] - Article_Data_Cord_File_DF_Ques['Concatenated Sentence Embeddings'][0].shape[0]
Cord_19_Length_Mean_Ques = Cord_19_Doc_Embeddings_DF['Document Embeddings'][0].shape[0] - Article_Data_Cord_File_DF_Ques['Mean Sentence Embeddings'][0].shape[0]
Cord_19_Length_Abstract = Cord_19_Doc_Embeddings_DF['Document Embeddings'][0].shape[0] - Article_Data_Cord_File_DF_Abstract['Sentence Embeddings'][0].shape[0]
Cord_19_Length_Body_Text = Cord_19_Doc_Embeddings_DF['Document Embeddings'][0].shape[0] - Article_Data_Cord_File_DF_Body_Text['Sentence Embeddings'][0].shape[0]
```

In [59]:
```
def row_padding_embeddings(row_embeddings,row_dimension):
    return np.pad(row_embeddings, (0, row_dimension), 'constant', constant_values=-1)
```

In [60]:
```
Article_Data_Cord_File_DF_Ques['Concatenated Sentence Embeddings'] = Article_Data_Cord_File_DF_Ques['Concatenated Sentence Embeddings'].progress_apply(row_padding_embeddings,args=(Cord_19_Length_Concate_Ques,))
Article_Data_Cord_File_DF_Ques['Mean Sentence Embeddings'] = Article_Data_Cord_File_DF_Ques['Mean Sentence Embeddings'].progress_apply(row_padding_embeddings,args=(Cord_19_Length_Mean_Ques,))
Article_Data_Cord_File_DF_Abstract['Sentence Embeddings'] = Article_Data_Cord_File_DF_Abstract['Sentence Embeddings'].progress_apply(row_padding_embeddings,args=(Cord_19_Length_Abstract,))
Article_Data_Cord_File_DF_Body_Text['Sentence Embeddings'] = Article_Data_Cord_File_DF_Body_Text['Sentence Embeddings'].progress_apply(row_padding_embeddings,args=(Cord_19_Length_Body_Text,))
```

```
  0%|          | 0/200 [00:00<?, ?it/s]
  0%|          | 0/200 [00:00<?, ?it/s]
  0%|          | 0/122754 [00:00<?, ?it/s]
  0%|          | 0/855743 [00:00<?, ?it/s]
```

Loading Titles of All Documents in Cord-19

In [61]:
```
Article_Data_Cord_File_DF = pd.read_pickle(r"D:\UoA\Tri 2\Big Data Analysis and Projects\Week 8\archive\cord_19_embeddings\article_data_cord_19_file_df.pkl")
```

In [62]:
```
Article_Data_Cord_File_DF = Article_Data_Cord_File_DF.loc[Article_Data_Cord_File_DF['Doc_Id_Meta_Parse'].isin(list(Article_Data_Cord_File_DF_Abstract['Doc_Id_Meta_Parse'].unique())),:]
Article_Data_Cord_File_DF = Article_Data_Cord_File_DF.loc[:,['Doc_Id_Meta_Parse','Title_Meta_Parse']].reset_index(drop=True)
```

In [63]:
```
import gc
gc.collect()
print('',end='')
```

Concatening Abstract and Body Text

In [64]:
```
Article_Data_Cord_File_DF_Abstract_Body_Text = pd.concat([Article_Data_Cord_File_DF_Abstract,Article_Data_Cord_File_DF_Body_Text],ignore_index=True)
```

Filtering Cord IDS

In [65]:
```
Article_Data_Cord_File_DF_Result.columns = ['Ques_Id_Meta_Parse','1','Doc_Id_Meta_Parse','Relevance Score','Batch']
```

In [66]:
```
Article_Data_Cord_File_DF_Abstract_Body_Text = pd.merge(Article_Data_Cord_File_DF_Result[['Doc_Id_Meta_Parse']], Article_Data_Cord_File_DF_Abstract_Body_Text, on="Doc_Id_Meta_Parse", how="inner")
```

In [67]:
```
Article_Data_Cord_File_DF_Abstract_Body_Text.reset_index(drop=True,inplace=True)
```

In [68]:
```
Cord_19_Doc_Embeddings_DF = pd.merge(Cord_19_Doc_Embeddings_DF,Article_Data_Cord_File_DF_Result[['Doc_Id_Meta_Parse']], on="Doc_Id_Meta_Parse", how="inner")
```

In [69]:
```
Cord_19_Doc_Embeddings_DF.reset_index(drop=True,inplace=True)
```

Extracting Relevant Sentences (Doc2Vec Question Embedding Model)

In [70]:
```
from scipy.spatial import distance
```

In [71]:
```
from IPython.display import display, HTML
```

In [72]:
```
pd.set_option('display.max_colwidth', None)
```

In [73]:
```
summation = 0
for Concat_Embeddings, Concat_Ques_ID in tqdm(zip(Article_Data_Cord_File_DF_Ques['Question Sentence Embeddings'],Article_Data_Cord_File_DF_Ques['Doc_Id_Meta_Parse']),total=Article_Data_Cord_File_DF_Ques.shape[0]):
    Concat_Embeddings = np.pad(Concat_Embeddings, (0, 668), 'constant', constant_values=-1)
    Cord_19_Doc_Embeddings_DF_Docu = pd.DataFrame()
    Cord_19_Doc_Embeddings_DF_Docu['Doc_Id_Meta_Parse'] = Cord_19_Doc_Embeddings_DF['Doc_Id_Meta_Parse'].copy()
    Cord_19_Doc_Embeddings_DF_Docu['Document Embeddings Cosine'] = Cord_19_Doc_Embeddings_DF['Document Embeddings'].apply(distance.cosine, args=(Concat_Embeddings,))
    Cord_19_Doc_Embeddings_DF_Docu['Document Embeddings Cosine'] = 1 - Cord_19_Doc_Embeddings_DF_Docu['Document Embeddings Cosine']

    Cord_19_Doc_Embeddings_DF_Docu = Cord_19_Doc_Embeddings_DF_Docu.sort_values(by='Document Embeddings Cosine',ascending=False).reset_index(drop=True)

    Cord_19_Doc_Embeddings_DF_Docu = Cord_19_Doc_Embeddings_DF_Docu.iloc[0:50]

    Cord_19_Doc_Embeddings_DF_Docu_List_ID = Cord_19_Doc_Embeddings_DF_Docu['Doc_Id_Meta_Parse'].to_list()

    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin = Article_Data_Cord_File_DF_Abstract_Body_Text.loc[(Article_Data_Cord_File_DF_Abstract_Body_Text['Doc_Id_Meta_Parse'].isin(Cord_19_Doc_Embeddings_DF_Docu_List_ID)),:].reset_index(drop=True)

    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu = pd.DataFrame()
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Doc_Id_Meta_Parse'] = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin['Doc_Id_Meta_Parse']
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Sentences'] = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin['Sentences']
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Type'] = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin['Type']
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Sentence Embeddings Cosine'] = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin['Sentence Embeddings'].apply(distance.cosine, args=(Concat_Embeddings,))
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Sentence Embeddings Cosine'] = 1 - Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Sentence Embeddings Cosine']

    Article_Data_Cord_File_DF_Relevant_Sentences = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu.groupby(['Doc_Id_Meta_Parse'], sort=False, as_index=False)['Sentence Embeddings Cosine'].idxmax()

    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu.iloc[Article_Data_Cord_File_DF_Relevant_Sentences['Sentence Embeddings Cosine'].to_list()]
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu.reset_index(drop=True,inplace=True)

    Article_Data_Cord_File_List_Abstract_Body_Text_Fin_Docu = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Doc_Id_Meta_Parse'].to_list()

    Article_Data_Cord_File_DF_Docu = Article_Data_Cord_File_DF.loc[Article_Data_Cord_File_DF['Doc_Id_Meta_Parse'].isin(Article_Data_Cord_File_List_Abstract_Body_Text_Fin_Docu),:]
    Article_Data_Cord_File_DF_Docu.reset_index(drop=True,inplace=True)

    Article_Data_Cord_File_DF_Docu_Rel_Sen = pd.merge(Article_Data_Cord_File_DF_Docu,Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu, on="Doc_Id_Meta_Parse", how="inner")

    Article_Data_Cord_File_List_Docu_Rel_Sen = Article_Data_Cord_File_DF_Docu_Rel_Sen['Doc_Id_Meta_Parse'].to_list()

    Article_Data_Cord_File_DF_Result_Docu = Article_Data_Cord_File_DF_Result.loc[(Article_Data_Cord_File_DF_Result['Doc_Id_Meta_Parse'].isin(Article_Data_Cord_File_List_Docu_Rel_Sen)),:]

    sum_sub = Article_Data_Cord_File_DF_Result_Docu['Relevance Score'][0:10].sum()
    summation = summation+sum_sub

print('\n')
print('\033[1m' + 'The Relevance Score of Question Sentence Embedding is:' + '\033[0m' + '\n')
print(summation)
print('\n')
```

```
  0%|          | 0/200 [00:00<?, ?it/s]
```

The Relevance Score of Question Sentence Embedding is:

600

Extracting Relevant Sentences (Doc2Vec Mean Embedding Model)

In [74]:
```
summation = 0
for Concat_Embeddings, Concat_Ques_ID in tqdm(zip(Article_Data_Cord_File_DF_Ques['Mean Sentence Embeddings'],Article_Data_Cord_File_DF_Ques['Doc_Id_Meta_Parse']),total=Article_Data_Cord_File_DF_Ques.shape[0]):
    Cord_19_Doc_Embeddings_DF_Docu = pd.DataFrame()
    Cord_19_Doc_Embeddings_DF_Docu['Doc_Id_Meta_Parse'] = Cord_19_Doc_Embeddings_DF['Doc_Id_Meta_Parse'].copy()
    Cord_19_Doc_Embeddings_DF_Docu['Document Embeddings Cosine'] = Cord_19_Doc_Embeddings_DF['Document Embeddings'].apply(distance.cosine, args=(Concat_Embeddings,))
    Cord_19_Doc_Embeddings_DF_Docu['Document Embeddings Cosine'] = 1 - Cord_19_Doc_Embeddings_DF_Docu['Document Embeddings Cosine']

    Cord_19_Doc_Embeddings_DF_Docu = Cord_19_Doc_Embeddings_DF_Docu.sort_values(by='Document Embeddings Cosine',ascending=False).reset_index(drop=True)

    Cord_19_Doc_Embeddings_DF_Docu = Cord_19_Doc_Embeddings_DF_Docu.iloc[0:50]

    Cord_19_Doc_Embeddings_DF_Docu_List_ID = Cord_19_Doc_Embeddings_DF_Docu['Doc_Id_Meta_Parse'].to_list()

    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin = Article_Data_Cord_File_DF_Abstract_Body_Text.loc[(Article_Data_Cord_File_DF_Abstract_Body_Text['Doc_Id_Meta_Parse'].isin(Cord_19_Doc_Embeddings_DF_Docu_List_ID)),:].reset_index(drop=True)

    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu = pd.DataFrame()
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Doc_Id_Meta_Parse'] = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin['Doc_Id_Meta_Parse']
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Sentences'] = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin['Sentences']
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Type'] = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin['Type']
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Sentence Embeddings Cosine'] = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin['Sentence Embeddings'].apply(distance.cosine, args=(Concat_Embeddings,))
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Sentence Embeddings Cosine'] = 1 - Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Sentence Embeddings Cosine']

    Article_Data_Cord_File_DF_Relevant_Sentences = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu.groupby(['Doc_Id_Meta_Parse'], sort=False, as_index=False)['Sentence Embeddings Cosine'].idxmax()

    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu.iloc[Article_Data_Cord_File_DF_Relevant_Sentences['Sentence Embeddings Cosine'].to_list()]
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu.reset_index(drop=True,inplace=True)

    Article_Data_Cord_File_List_Abstract_Body_Text_Fin_Docu = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Doc_Id_Meta_Parse'].to_list()

    Article_Data_Cord_File_DF_Docu = Article_Data_Cord_File_DF.loc[Article_Data_Cord_File_DF['Doc_Id_Meta_Parse'].isin(Article_Data_Cord_File_List_Abstract_Body_Text_Fin_Docu),:]
    Article_Data_Cord_File_DF_Docu.reset_index(drop=True,inplace=True)

    Article_Data_Cord_File_DF_Docu_Rel_Sen = pd.merge(Article_Data_Cord_File_DF_Docu,Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu, on="Doc_Id_Meta_Parse", how="inner")

    Article_Data_Cord_File_List_Docu_Rel_Sen = Article_Data_Cord_File_DF_Docu_Rel_Sen['Doc_Id_Meta_Parse'].to_list()

    Article_Data_Cord_File_DF_Result_Docu = Article_Data_Cord_File_DF_Result.loc[(Article_Data_Cord_File_DF_Result['Doc_Id_Meta_Parse'].isin(Article_Data_Cord_File_List_Docu_Rel_Sen)),:]

    sum_sub = Article_Data_Cord_File_DF_Result_Docu['Relevance Score'][0:10].sum()
    summation = summation+sum_sub

print('\n')
print('\033[1m' + 'The Relevance Score of Mean Sentence Embedding is:' + '\033[0m' + '\n')
print(summation)
print('\n')
```

```
  0%|          | 0/200 [00:00<?, ?it/s]
```

The Relevance Score of Mean Sentence Embedding is:

600

Extracting Relevant Sentences (Doc2Vec Concatenated Embedding Model)

In [75]:
```
summation = 0
for Concat_Embeddings, Concat_Ques_ID in tqdm(zip(Article_Data_Cord_File_DF_Ques['Concatenated Sentence Embeddings'],Article_Data_Cord_File_DF_Ques['Doc_Id_Meta_Parse']),total=Article_Data_Cord_File_DF_Ques.shape[0]):
    Cord_19_Doc_Embeddings_DF_Docu = pd.DataFrame()
    Cord_19_Doc_Embeddings_DF_Docu['Doc_Id_Meta_Parse'] = Cord_19_Doc_Embeddings_DF['Doc_Id_Meta_Parse'].copy()
    Cord_19_Doc_Embeddings_DF_Docu['Document Embeddings Cosine'] = Cord_19_Doc_Embeddings_DF['Document Embeddings'].apply(distance.cosine, args=(Concat_Embeddings,))
    Cord_19_Doc_Embeddings_DF_Docu['Document Embeddings Cosine'] = 1 - Cord_19_Doc_Embeddings_DF_Docu['Document Embeddings Cosine']

    Cord_19_Doc_Embeddings_DF_Docu = Cord_19_Doc_Embeddings_DF_Docu.sort_values(by='Document Embeddings Cosine',ascending=False).reset_index(drop=True)

    Cord_19_Doc_Embeddings_DF_Docu = Cord_19_Doc_Embeddings_DF_Docu.iloc[0:50]

    Cord_19_Doc_Embeddings_DF_Docu_List_ID = Cord_19_Doc_Embeddings_DF_Docu['Doc_Id_Meta_Parse'].to_list()

    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin = Article_Data_Cord_File_DF_Abstract_Body_Text.loc[(Article_Data_Cord_File_DF_Abstract_Body_Text['Doc_Id_Meta_Parse'].isin(Cord_19_Doc_Embeddings_DF_Docu_List_ID)),:].reset_index(drop=True)

    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu = pd.DataFrame()
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Doc_Id_Meta_Parse'] = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin['Doc_Id_Meta_Parse']
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Sentences'] = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin['Sentences']
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Type'] = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin['Type']
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Sentence Embeddings Cosine'] = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin['Sentence Embeddings'].apply(distance.cosine, args=(Concat_Embeddings,))
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Sentence Embeddings Cosine'] = 1 - Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Sentence Embeddings Cosine']

    Article_Data_Cord_File_DF_Relevant_Sentences = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu.groupby(['Doc_Id_Meta_Parse'], sort=False, as_index=False)['Sentence Embeddings Cosine'].idxmax()

    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu.iloc[Article_Data_Cord_File_DF_Relevant_Sentences['Sentence Embeddings Cosine'].to_list()]
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu.reset_index(drop=True,inplace=True)

    Article_Data_Cord_File_List_Abstract_Body_Text_Fin_Docu = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Doc_Id_Meta_Parse'].to_list()

    Article_Data_Cord_File_DF_Docu = Article_Data_Cord_File_DF.loc[Article_Data_Cord_File_DF['Doc_Id_Meta_Parse'].isin(Article_Data_Cord_File_List_Abstract_Body_Text_Fin_Docu),:]
    Article_Data_Cord_File_DF_Docu.reset_index(drop=True,inplace=True)

    Article_Data_Cord_File_DF_Docu_Rel_Sen = pd.merge(Article_Data_Cord_File_DF_Docu,Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu, on="Doc_Id_Meta_Parse", how="inner")

    Article_Data_Cord_File_List_Docu_Rel_Sen = Article_Data_Cord_File_DF_Docu_Rel_Sen['Doc_Id_Meta_Parse'].to_list()

    Article_Data_Cord_File_DF_Result_Docu = Article_Data_Cord_File_DF_Result.loc[(Article_Data_Cord_File_DF_Result['Doc_Id_Meta_Parse'].isin(Article_Data_Cord_File_List_Docu_Rel_Sen)),:]

    sum_sub = Article_Data_Cord_File_DF_Result_Docu['Relevance Score'][0:10].sum()
    summation = summation+sum_sub

print('\n')
print('\033[1m' + 'The Relevance Score of Concatenated Sentence Embedding is:' + '\033[0m' + '\n')
print(summation)
print('\n')
```

```
  0%|          | 0/200 [00:00<?, ?it/s]
```

The Relevance Score of Concatenated Sentence Embedding is:

600

Concatenated Doc2Vec Model ChatBot (Making ChatBot of only Best Model) (Extracting Relevant Sentences)

In [76]:
```
while (True):
    Cord_19_User_Input = input("\nEnter the Question: \nEnter Command 'List All Questions' to see All Available Questions \nEnter Command 'Close' to Close ChatBot\n\n")
```

```
    if Cord_19_User_Input == 'Close':
        break
    elif Cord_19_User_Input == 'List All Questions':
        display(HTML(Article_Data_Cord_File_DF_Ques[['Question']].copy().to_html()))
        continue
    elif ~(Article_Data_Cord_File_DF_Ques['Question'].isin([Cord_19_User_Input]).any()):
        print("\nPlease Enter Another Question, the Sentence Embeddings does not exist for this Question")
        continue
    else:
        pass
    Cord_19_Doc_Embeddings_DF_Docu = pd.DataFrame()
    Cord_19_Doc_Embeddings_DF_Docu['Doc_Id_Meta_Parse'] = Cord_19_Doc_Embeddings_DF['Doc_Id_Meta_Parse'].copy()
    Cord_19_Doc_Embeddings_DF_Docu['Document Embeddings Cosine'] = Cord_19_Doc_Embeddings_DF['Document Embeddings'].progress_apply(distance.cosine, args=(Article_Data_Cord_File_DF_Ques.loc[Article_Data_Cord_File_DF_Ques['Question'] == Cord_19_User_Input,'Concatenated Sentence Embeddings'].reset_index(drop=True)[0],))
    Cord_19_Doc_Embeddings_DF_Docu['Document Embeddings Cosine'] = 1 - Cord_19_Doc_Embeddings_DF_Docu['Document Embeddings Cosine']

    Cord_19_Doc_Embeddings_DF_Docu = Cord_19_Doc_Embeddings_DF_Docu.sort_values(by='Document Embeddings Cosine',ascending=False).reset_index(drop=True)

    Cord_19_Doc_Embeddings_DF_Docu = Cord_19_Doc_Embeddings_DF_Docu.iloc[0:50]

    Cord_19_Doc_Embeddings_DF_Docu_List_ID = Cord_19_Doc_Embeddings_DF_Docu['Doc_Id_Meta_Parse'].to_list()

    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin = Article_Data_Cord_File_DF_Abstract_Body_Text.loc[Article_Data_Cord_File_DF_Abstract_Body_Text['Doc_Id_Meta_Parse'].isin(Cord_19_Doc_Embeddings_DF_Docu_List_ID),:].reset_index(drop=True)

    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu = pd.DataFrame()
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Doc_Id_Meta_Parse'] = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin['Doc_Id_Meta_Parse']
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Sentences'] = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin['Sentences']
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Type'] = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin['Type']
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Sentence Embeddings Cosine'] = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin['Sentence Embeddings'].apply(distance.cosine, args=(Article_Data_Cord_File_DF_Ques.loc[Article_Data_Cord_File_DF_Ques['Question'] == Cord_19_User_Input,'Concatenated Sentence Embeddings'].reset_index(drop=True)[0],))
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Sentence Embeddings Cosine'] = 1 - Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Sentence Embeddings Cosine']

    Article_Data_Cord_File_DF_Relevant_Sentences = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu.groupby(['Doc_Id_Meta_Parse'], sort=False, as_index=False)['Sentence Embeddings Cosine'].idxmax()

    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu.iloc[Article_Data_Cord_File_DF_Relevant_Sentences['Sentence Embeddings Cosine'].to_list()]
    Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu.reset_index(drop=True,inplace=True)

    Article_Data_Cord_File_List_Abstract_Body_Text_Fin_Docu = Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu['Doc_Id_Meta_Parse'].to_list()

    Article_Data_Cord_File_DF_Docu = Article_Data_Cord_File_DF.loc[Article_Data_Cord_File_DF['Doc_Id_Meta_Parse'].isin(Article_Data_Cord_File_List_Abstract_Body_Text_Fin_Docu),:]
    Article_Data_Cord_File_DF_Docu.reset_index(drop=True,inplace=True)

    Article_Data_Cord_File_DF_Docu_Rel_Sen = pd.merge(Article_Data_Cord_File_DF_Docu,Article_Data_Cord_File_DF_Abstract_Body_Text_Fin_Docu, on="Doc_Id_Meta_Parse", how="inner")

    print('\033[1m' + 'The Loaded Meta Dataset is:' + '\033[0m' + '\n')
    display(HTML(Article_Data_Cord_File_DF_Docu_Rel_Sen.loc[0:9,['Title_Meta_Parse','Sentences']].to_html()))
    print('\n')
```

	Question
0	what is the origin of COVID-19
1	how does the coronavirus respond to changes in the weather
2	will SARS-CoV2 infected people develop immunity? Is cross protection possible?
3	what causes death from Covid-19?
4	what drugs have been active against SARS-CoV or SARS-CoV-2 in animal studies?
5	what types of rapid testing for Covid-19 have been developed?
6	are there serological tests that detect antibodies to coronavirus?
7	how has lack of testing availability led to underreporting of true incidence of Covid-19?
8	how has COVID-19 affected Canada
9	has social distancing had an impact on slowing the spread of COVID-19?
10	what are the guidelines for triaging patients infected with coronavirus?
11	what are best practices in hospitals and at home in maintaining quarantine?
12	what are the transmission routes of coronavirus?
13	what evidence is there related to COVID-19 super spreaders
14	how long can the coronavirus live outside the body
15	how long does coronavirus remain stable on surfaces?
16	are there any clinical trials available for the coronavirus
17	what are the best masks for preventing infection by Covid-19?
18	what type of hand sanitizer is needed to destroy Covid-19?
19	are patients taking Angiotensin-converting enzyme inhibitors (ACE) at increased risk for COVID-19?
20	what are the mortality rates overall and in specific populations
21	are cardiac complications likely in patients with COVID-19?
22	what kinds of complications related to COVID-19 are associated with hypertension?
23	what kinds of complications related to COVID-19 are associated with diabetes
24	which biomarkers predict the severe clinical course of 2019-nCOV infection?
25	what are the initial symptoms of Covid-19?
26	what is known about those infected with Covid-19 but are asymptomatic?
27	what evidence is there for the value of hydroxychloroquine in treating Covid-19?
28	which SARS-CoV-2 proteins-human proteins interactions indicate potential for drug targets. Are there approved drugs that can be repurposed based on this information?
29	is remdesivir an effective treatment for COVID-19
30	what is the origin of COVID-19
31	how does the coronavirus respond to changes in the weather
32	will SARS-CoV2 infected people develop immunity? Is cross protection possible?
33	what causes death from Covid-19?
34	what drugs have been active against SARS-CoV or SARS-CoV-2 in animal studies?
35	what types of rapid testing for Covid-19 have been developed?
36	are there serological tests that detect antibodies to coronavirus?
37	how has lack of testing availability led to underreporting of true incidence of Covid-19?
38	how has COVID-19 affected Canada
39	has social distancing had an impact on slowing the spread of COVID-19?
40	what are the guidelines for triaging patients infected with coronavirus?
41	what are best practices in hospitals and at home in maintaining quarantine?
42	what are the transmission routes of coronavirus?
43	what evidence is there related to COVID-19 super spreaders
44	how long can the coronavirus live outside the body
45	how long does coronavirus remain stable on surfaces?
46	are there any clinical trials available for the coronavirus
47	what are the best masks for preventing infection by Covid-19?
48	what type of hand sanitizer is needed to destroy Covid-19?
49	are patients taking Angiotensin-converting enzyme inhibitors (ACE) at increased risk for COVID-19?
50	what are the mortality rates overall and in specific populations
51	are cardiac complications likely in patients with COVID-19?
52	what kinds of complications related to COVID-19 are associated with hypertension?
53	what kinds of complications related to COVID-19 are associated with diabetes
54	which biomarkers predict the severe clinical course of 2019-nCOV infection?
55	what are the initial symptoms of Covid-19?
56	what is known about those infected with Covid-19 but are asymptomatic?
57	what evidence is there for the value of hydroxychloroquine in treating Covid-19?
58	which SARS-CoV-2 proteins-human proteins interactions indicate potential for drug targets. Are there approved drugs that can be repurposed based on this information?
59	is remdesivir an effective treatment for COVID-19
60	How does the coronavirus differ from seasonal flu?
61	Does SARS-CoV-2 have any subtypes, and if so what are they?
62	What vaccine candidates are being tested for Covid-19?
63	What are the longer-term complications of those who recover from COVID-19?
64	What new public datasets are available related to COVID-19?
65	what is the origin of COVID-19
66	how does the coronavirus respond to changes in the weather
67	will SARS-CoV2 infected people develop immunity? Is cross protection possible?
68	what causes death from Covid-19?
69	what drugs have been active against SARS-CoV or SARS-CoV-2 in animal studies?
70	what types of rapid testing for Covid-19 have been developed?
71	are there serological tests that detect antibodies to coronavirus?
72	how has lack of testing availability led to underreporting of true incidence of Covid-19?
73	how has COVID-19 affected Canada
74	has social distancing had an impact on slowing the spread of COVID-19?
75	what are the guidelines for triaging patients infected with coronavirus?
76	what are best practices in hospitals and at home in maintaining quarantine?
77	what are the transmission routes of coronavirus?
78	what evidence is there related to COVID-19 super spreaders
79	how long can the coronavirus live outside the body
80	how long does coronavirus remain stable on surfaces?
81	are there any clinical trials available for the coronavirus
82	what are the best masks for preventing infection by Covid-19?
83	what type of hand sanitizer is needed to destroy Covid-19?
84	are patients taking Angiotensin-converting enzyme inhibitors (ACE) at increased risk for COVID-19?
85	what are the mortality rates overall and in specific populations
86	are cardiac complications likely in patients with COVID-19?
87	what kinds of complications related to COVID-19 are associated with hypertension?
88	what kinds of complications related to COVID-19 are associated with diabetes
89	which biomarkers predict the severe clinical course of 2019-nCOV infection?
90	what are the initial symptoms of Covid-19?
91	what is known about those infected with Covid-19 but are asymptomatic?
92	what evidence is there for the value of hydroxychloroquine in treating Covid-19?
93	which SARS-CoV-2 proteins-human proteins interactions indicate potential for drug targets. Are there approved drugs that can be repurposed based on this information?
94	is remdesivir an effective treatment for COVID-19
95	How does the coronavirus differ from seasonal flu?
96	Does SARS-CoV-2 have any subtypes, and if so what are they?
97	What vaccine candidates are being tested for Covid-19?
98	What are the longer-term complications of those who recover from COVID-19?
99	What new public datasets are available related to COVID-19?
100	What is the protein structure of the SARS-CoV-2 spike?
101	What is the result of phylogenetic analysis of SARS-CoV-2 genome sequence?
102	What is the mechanism of inflammatory response and pathogenesis of COVID-19 cases?
103	What is the mechanism of cytokine storm syndrome on the COVID-19?
104	What are the observed mutations in the SARS-CoV-2 genome and how often do the mutations occur?
105	what is the origin of COVID-19
106	how does the coronavirus respond to changes in the weather
107	will SARS-CoV2 infected people develop immunity? Is cross protection possible?
108	what causes death from Covid-19?
109	what drugs have been active against SARS-CoV or SARS-CoV-2 in animal studies?
110	what types of rapid testing for Covid-19 have been developed?
111	are there serological tests that detect antibodies to coronavirus?
112	how has lack of testing availability led to underreporting of true incidence of Covid-19?
113	how has COVID-19 affected Canada
114	has social distancing had an impact on slowing the spread of COVID-19?
115	what are the guidelines for triaging patients infected with coronavirus?
116	what are best practices in hospitals and at home in maintaining quarantine?
117	what are the transmission routes of coronavirus?
118	what evidence is there related to COVID-19 super spreaders
119	how long can the coronavirus live outside the body
120	how long does coronavirus remain stable on surfaces?
121	are there any clinical trials available for the coronavirus
122	what are the best masks for preventing infection by Covid-19?
123	what type of hand sanitizer is needed to destroy Covid-19?
124	are patients taking Angiotensin-converting enzyme inhibitors (ACE) at increased risk for COVID-19?
125	what are the mortality rates overall and in specific populations
126	are cardiac complications likely in patients with COVID-19?
127	what kinds of complications related to COVID-19 are associated with hypertension?
128	what kinds of complications related to COVID-19 are associated with diabetes
129	which biomarkers predict the severe clinical course of 2019-nCOV infection?
130	what are the initial symptoms of Covid-19?
131	what is known about those infected with Covid-19 but are asymptomatic?
132	what evidence is there for the value of hydroxychloroquine in treating Covid-19?
133	which SARS-CoV-2 proteins-human proteins interactions indicate potential for drug targets. Are there approved drugs that can be repurposed based on this information?
134	is remdesivir an effective treatment for COVID-19
135	How does the coronavirus differ from seasonal flu?
136	Does SARS-CoV-2 have any subtypes, and if so what are they?
137	What vaccine candidates are being tested for Covid-19?
138	What are the longer-term complications of those who recover from COVID-19?
139	What new public datasets are available related to COVID-19?
140	What is the protein structure of the SARS-CoV-2 spike?
141	What is the result of phylogenetic analysis of SARS-CoV-2 genome sequence?
142	What is the mechanism of inflammatory response and pathogenesis of COVID-19 cases?
143	What is the mechanism of cytokine storm syndrome on the COVID-19?
144	What are the observed mutations in the SARS-CoV-2 genome and how often do the mutations occur?
145	What are the impacts of COVID-19 among African-Americans that differ from the rest of the U.S. population?
146	Does Vitamin D impact COVID-19 prevention and treatment?
147	How has the COVID-19 pandemic impacted violence in society, including violent crimes?
148	How much impact do masks have on preventing the spread of the COVID-19?
149	How has the COVID-19 pandemic impacted mental health?
150	what is the origin of COVID-19
151	how does the coronavirus respond to changes in the weather
152	will SARS-CoV2 infected people develop immunity? Is cross protection possible?
153	what causes death from Covid-19?
154	what drugs have been active against SARS-CoV or SARS-CoV-2 in animal studies?
155	what types of rapid testing for Covid-19 have been developed?

	Question
156	are there serological tests that detect antibodies to coronavirus?
157	how has lack of testing availability led to underreporting of true incidence of Covid-19?
158	how has COVID-19 affected Canada
159	has social distancing had an impact on slowing the spread of COVID-19?
160	what are the guidelines for triaging patients infected with coronavirus?
161	what are best practices in hospitals and at home in maintaining quarantine?
162	what are the transmission routes of coronavirus?
163	what evidence is there related to COVID-19 super spreaders
164	how long can the coronavirus live outside the body
165	how long does coronavirus remain stable on surfaces?
166	are there any clinical trials available for the coronavirus
167	what are the best masks for preventing infection by Covid-19?
168	what type of hand sanitizer is needed to destroy Covid-19?
169	are patients taking Angiotensin-converting enzyme inhibitors (ACE) at increased risk for COVID-19?
170	what are the mortality rates overall and in specific populations
171	are cardiac complications likely in patients with COVID-19?
172	what kinds of complications related to COVID-19 are associated with hypertension?
173	what kinds of complications related to COVID-19 are associated with diabetes
174	which biomarkers predict the severe clinical course of 2019-nCOV infection?
175	what are the initial symptoms of Covid-19?
176	what is known about those infected with Covid-19 but are asymptomatic?
177	what evidence is there for the value of hydroxychloroquine in treating Covid-19?
178	which SARS-CoV-2 proteins-human proteins interactions indicate potential for drug targets. Are there approved drugs that can be repurposed based on this information?
179	is remdesivir an effective treatment for COVID-19
180	How does the coronavirus differ from seasonal flu?
181	Does SARS-CoV-2 have any subtypes, and if so what are they?
182	What vaccine candidates are being tested for Covid-19?
183	What are the longer-term complications of those who recover from COVID-19?
184	What new public datasets are available related to COVID-19?
185	What is the protein structure of the SARS-CoV-2 spike?
186	What is the result of phylogenetic analysis of SARS-CoV-2 genome sequence?
187	What is the mechanism of inflammatory response and pathogenesis of COVID-19 cases?
188	What is the mechanism of cytokine storm syndrome on the COVID-19?
189	What are the observed mutations in the SARS-CoV-2 genome and how often do the mutations occur?
190	What are the impacts of COVID-19 among African-Americans that differ from the rest of the U.S. population?
191	Does Vitamin D impact COVID-19 prevention and treatment?
192	How has the COVID-19 pandemic impacted violence in society, including violent crimes?
193	How much impact do masks have on preventing the spread of the COVID-19?
194	How has the COVID-19 pandemic impacted mental health?
195	what evidence is there for dexamethasone as a treatment for COVID-19?
196	what are the health outcomes for children who contract COVID-19?
197	what are the benefits and risks of re-opening schools in the midst of the COVID-19 pandemic?
198	do individuals who recover from COVID-19 show sufficient immune response, including antibody levels and T-cell mediated immunity, to prevent re-infection?
199	what is known about an mRNA vaccine for the SARS-CoV-2 virus?

```
  0%|          | 0/22520 [00:00<?, ?it/s]
The Loaded Meta Dataset is:
```

	Title_Meta_Parse	Sentences
0	Nasal Airway Obstruction Study (NAIROS): a phase III, open-label, mixed-methods, multicentre randomised controlled trial of septoplasty versus medical management of a septal deviation with nasal obstruction	The recruitment target is 378 patients, recruited from up to 17 sites across Scotland, England and Wales.
1	Big Data and Biodefense: Prospects and Pitfalls	These developments provide opportunities to think about biodefense preparedness in new and unique ways.
2	Simulation Based Exploration of Bacterial Cross Talk Between Spatially Separated Colonies in a Multispecies Biofilm Community	increases substrate uptake if the signal surpasses induction threshold, but it does not produce the signal.
3	EdNet: A Large-Scale Hierarchical Dataset in Education	The features of EdNet are domain-agnostic, allowing EdNet to be easily extended to different domains.
4	Putting Attacks in Context: A Building Automation Testbed for Impact Assessment from the Victim's Perspective	We assume that all building services and business processes are needed/active at the time of the assessment.
5	Prone Position in Management of COVID-19 Patients; a Commentary	Parisa Ghelichkhani: 0000-0003-3763-7999 Maryam Esmaeili: 0000-0002-4798-2270\nNo fund has been received.
6	A pandemic-resilient open-inquiry physical science lab course which leverages the Maker movement	The second round of open-inquiry projects were completed start-to-finish under pandemic restrictions.
7	Principles to Practices for Responsible AI: Closing the Gap	The same functional separation may apply even when non-technical teams are internal to an organization.
8	Abnormal pulmonary function in COVID-19 patients at time of hospital discharge	However, until now, there is no report in regard to pulmonary function in discharged COVID-19 survivors.
9	Coronavirus concerns: What do women with gynecologic cancer need to know during the COVID-19 crisis?	Participants were invited to share questions through an online portal prior to and during the webinar.

In [ ]: