```
import pandas as pd
import re
import gensim
from gensim.utils import simple_preprocess
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
import gensim.corpora as corpora
from gensim.models import CoherenceModel
import spacy
import warnings
warnings.filterwarnings("ignore")

#load the dataset
dataframe = pd.read_csv('/content/bbc.csv')
data=dataframe[['title','description']]

#preprocessing
#removing punctuations
cleanedtext = data['description'].map(lambda x: re.sub('[,\.!?]','',x))

#lowercase all text
cleanedtext = cleanedtext.map(lambda x: x.lower())

#removing stopwords
stopwords=stopwords.words('english')

def remove_all_stopwords(texts):
  return[[word for word in simple_preprocess(str(doc))
          if word not in stopwords] for doc in texts]

#converts text to list
text_to_list=cleanedtext.values.tolist()

#Tokenization
text_as_words = []

for item in text_to_list:
    words = item.split()
    text_as_words.extend(words)

#remove stopwords
words = remove_all_stopwords(text_as_words)

#bigram and trigram
bigram=gensim.models.Phrases(words,min_count=5,threshold=50)
trigram=gensim.models.Phrases(bigram[words],threshold=50)

bigram_mod=gensim.models.phrases.Phraser(bigram)
trigram_mod=gensim.models.phrases.Phraser(trigram)

def bi(texts):
  return[bigram_mod[doc] for doc in texts]

def tri(texts):
   return[trigram_mod[bigram_mod[doc]] for doc in texts]

#using spacy library for removing stop words
nlp=spacy.load('en_core_web_sm')

def lemmatization(texts, allowed_postags=['NOUN','ADJ','VERB','ADV']):
  texts_out=[]
  for sent in texts:
    doc = nlp("".join(sent))
    texts_out.append([token.lemma_ for token in doc if token.pos_ in allowed_postags])
  return texts_out

#from bigrams
clean_words_bigrams=bi(words)

#lemmatization to keep only noun,adj,vb,adv
clean_words_lemmatize=lemmatization(clean_words_bigrams, allowed_postags=['NOUN','ADJ','VERB','ADV'] )
clean_words_lemmatize = [words for words in clean_words_lemmatize if words]

id2word=corpora.Dictionary(clean_words_lemmatize)
texts=clean_words_lemmatize
corpus=[id2word.doc2bow(text) for text in texts]
print(corpus)

#LDA MODEL TRAINING
num_topics=5
```

```
lda_model=gensim.models.LdaMulticore(corpus=corpus,id2word=id2word,num_topics=num_topics)

from pprint import pprint
pprint(lda_model.print_topics(3))

coherence_model_lda=CoherenceModel(model=lda_model, texts=clean_words_lemmatize,dictionary=id2word,coherence='c_v')
coherence_lda=coherence_model_lda.get_coherence()
print("The model accuracy is :" ,coherence_lda)
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
WARNING:gensim.models.ldamulticore:too few updates, training might not converge; consider increasing the number of passes or iterati
[[(0, 1)], [(1, 1)], [(2, 1)], [(3, 1)], [(4, 1)], [(5, 1)], [(6, 1)], [(7, 1)], [(8, 1)], [(9, 1)], [(10, 1)], [(11, 1)], [(12, 1)
[(2,
  '0.066*"ukraine" + 0.021*"say" + 0.019*"country" + 0.017*"try" + '
  '0.011*"yearold" + 0.011*"play" + 0.011*"people" + 0.011*"ban" + '
  '0.010*"economic" + 0.008*"nation"'),
 (0,
  '0.022*"president" + 0.018*"city" + 0.017*"say" + 0.016*"england" + '
  '0.013*"meet" + 0.009*"feel" + 0.009*"crew" + 0.009*"manchester" + '
  '0.008*"help" + 0.007*"symbol"'),
 (1,
  '0.040*"ukrainian" + 0.021*"covid" + 0.016*"bbc" + 0.012*"home" + '
  '0.012*"child" + 0.010*"energy" + 0.010*"big" + 0.008*"die" + 0.008*"test" + '
  '0.008*"parent"')]
The model accuracy is : 0.827223263069652
```

LDA MODEL **TRAINING**