In [1]:

```
import pandas as pd
```

In [2]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [3]:

```
data = pd.read_csv('nlp.csv')
```

In [4]:

```
data.head()
```

Out[4]:

	Unnamed: 0	id	title	authors	authors parsed	abstract	year
0	0	704.0668	J/psi Production in an Equilibrating Partonic ...	Xiao-Ming xu	[['xu', 'Xiao-Ming', '']]	Any color singlet or octet ccbar pair is cre...	1999
1	1	704.1962	A simple test of quantumness for a single system	Robert Alicki and Nicholas Van Ryn	[['Alicki', 'Robert', ''], ['Van Ryn', 'Nichol...	We propose a simple test of quantumness whic...	2001
2	2	704.2791	Coevolution of Quantum Wave Functions and the ...	W. Q. Sumner and D. Y. Sumner	[['Sumner', 'W. Q.', ''], ['Sumner', 'D. Y.', ...	Erwin Schrodinger (1939) proved that quantum...	2000
3	3	705.1291	Spinodal decomposition of low-density asymmetr...	V.Baran, M. Colonna, M. Di Toro, A.B. Larionov	[['Baran', 'V.', ''], ['Colonna', 'M.', ''], [...	We investigate the dynamical properties of a...	1998
4	4	705.1799	Subjective Questions and Answers for a Mathema...	Florentin Smarandache	[['Smarandache', 'Florentin', '']]	This article of mathematical education refle...	1997

In [5]:

```
data.tail()
```

Out[5]:

	Unnamed: 0	id	title	authors	authors parsed	abstract	yea
26988	26988	solv-int/9904012	Differential equations and duality in massless...	P. Fendley and H. Saleur	[['Fendley', 'P.', ''], ['Saleur', 'H.', '']]	Functional relations play a key role in the ...	200
26989	26989	solv-int/9908002	Quantum Backlund transformation for the integr...	V.B.Kuznetsov, M.Salerno and E.K.Sklyanin	[['Kuznetsov', 'V. B.', ''], ['Salerno', 'M.',...	For the integrable case of the discrete self...	200
26990	26990	solv-int/9909016	Darboux Transformation for Supersymmetric KP H...	Q. P. Liu, Manuel Manas	[['Liu', 'Q. P.', ''], ['Manas', 'Manuel', '']]	We construct Darboux transformations for the...	200
26991	26991	solv-int/9912012	Whitham-Toda Hierarchy in the Laplacian Growth...	Mark Mineev-Weinstein and Anton Zabrodin	[['Mineev-Weinstein', 'Mark', ''], ['Zabrodin'...	The Laplacian growth problem in the limit of...	200
26992	26992	supr-con/9608004	Ginzburg-Landau-Gor'kov Theory of Magnetic osc...	G. M. Bruun, V. N. Nicopoulos, N. F. Johnson	[['Bruun', 'G. M.', ''], ['Nicopoulos', 'V. N....	We investigate de Haas-van Alphen (dHvA) osc...	199

In [6]:

```
data.shape
```

Out[6]:

```
(26993, 7)
```

In [7]:

```
data.columns
```

Out[7]:

```
Index(['Unnamed: 0', 'id', 'title', 'authors', 'authors parsed', 'abstract',
       'year'],
      dtype='object')
```

In [8]:

```
data.duplicated().sum()
```

Out[8]:

```
0
```

In [9]:

```
data.isnull().sum()
```

Out[9]:

```
Unnamed: 0        0
id                0
title             0
authors           0
authors parsed    0
abstract          0
year              0
dtype: int64
```

In [10]:

```
data = data.drop('Unnamed: 0', axis = 1)
```

In [11]:

```
data
```

Out[11]:

	id	title	authors	authors parsed	abstract	year
0	704.0668	J/psi Production in an Equilibrating Partonic ...	Xiao-Ming xu	[['xu', 'Xiao-Ming', '']]	Any color singlet or octet ccbar pair is cre...	1999
1	704.1962	A simple test of quantumness for a single system	Robert Alicki and Nicholas Van Ryn	[['Alicki', 'Robert', ''], ['Van Ryn', 'Nichol...	We propose a simple test of quantumness whic...	2001
2	704.2791	Coevolution of Quantum Wave Functions and the ...	W. Q. Sumner and D. Y. Sumner	[['Sumner', 'W. Q.', ''], ['Sumner', 'D. Y.', ...	Erwin Schrodinger (1939) proved that quantum...	2000
3	705.1291	Spinodal decomposition of low-density asymmetr...	V.Baran, M. Colonna, M. Di Toro, A.B. Larionov	[['Baran', 'V.', ''], ['Colonna', 'M.', ''], [...	We investigate the dynamical properties of a...	1998
4	705.1799	Subjective Questions and Answers for a Mathema...	Florentin Smarandache	[['Smarandache', 'Florentin', '']]	This article of mathematical education refle...	1997
...	...	...	...	...	...	...
26988	solv-int/9904012	Differential equations and duality in massless...	P. Fendley and H. Saleur	[['Fendley', 'P.', ''], ['Saleur', 'H.', '']]	Functional relations play a key role in the ...	2000
26989	solv-int/9908002	Quantum Backlund transformation for the integr...	V.B.Kuznetsov, M.Salerno and E.K.Sklyanin	[['Kuznetsov', 'V. B.', ''], ['Salerno', 'M.',...	For the integrable case of the discrete self...	2000
26990	solv-int/9909016	Darboux Transformation for Supersymmetric KP H...	Q. P. Liu, Manuel Manas	[['Liu', 'Q. P.', ''], ['Manas', 'Manuel', '']]	We construct Darboux transformations for the...	2000
26991	solv-int/9912012	Whitham-Toda Hierarchy in the Laplacian Growth...	Mark Mineev-Weinstein and Anton Zabrodin	[['Mineev-Weinstein', 'Mark', ''], ['Zabrodin'...	The Laplacian growth problem in the limit of...	2001
26992	supr-con/9608004	Ginzburg-Landau-Gor'kov Theory of Magnetic osc...	G. M. Bruun, V. N. Nicopoulos, N. F. Johnson	[['Bruun', 'G. M.', ''], ['Nicopoulos', 'V. N....	We investigate de Haas-van Alphen (dHvA) osc...	1997

26993 rows × 6 columns

In [12]:

```
data_new = data[['title', 'abstract']]
```

In [13]:

```
data_new.head()
```

Out[13]:

	title	abstract
0	J/psi Production in an Equilibrating Partonic ...	Any color singlet or octet ccbar pair is cre...
1	A simple test of quantumness for a single system	We propose a simple test of quantumness whic...
2	Coevolution of Quantum Wave Functions and the ...	Erwin Schrodinger (1939) proved that quantum...
3	Spinodal decomposition of low-density asymmetr...	We investigate the dynamical properties of a...
4	Subjective Questions and Answers for a Mathema...	This article of mathematical education refle...

In [14]:

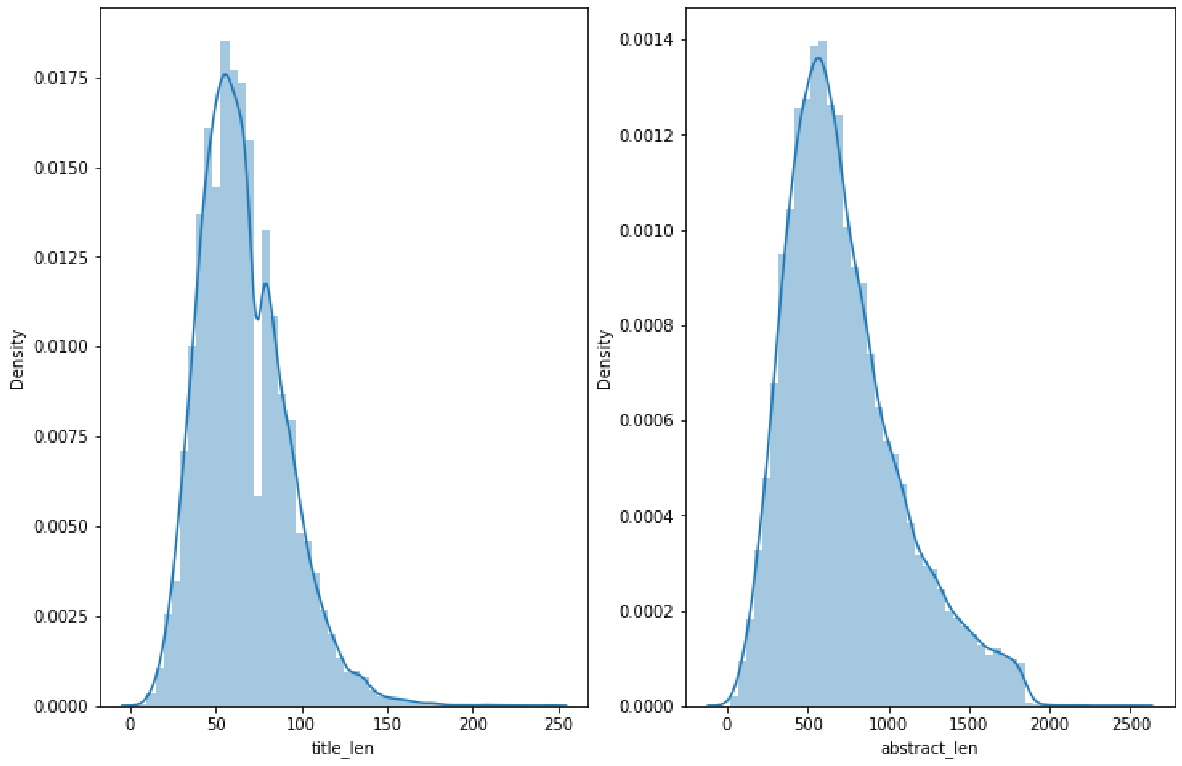
```
data_new['title_len'] = data_new['title'].apply(len)
data_new['abstract_len'] = data_new['abstract'].apply(len)
```

In [15]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [16]:

```
fig, axes = plt.subplots(nrows=1,ncols=2,figsize=(12,8))
sns.distplot(data_new.title_len,ax=axes[0])
sns.distplot(data_new.abstract_len,ax=axes[1]);
```



In [18]:

```
data_new.describe()
```

Out[18]:

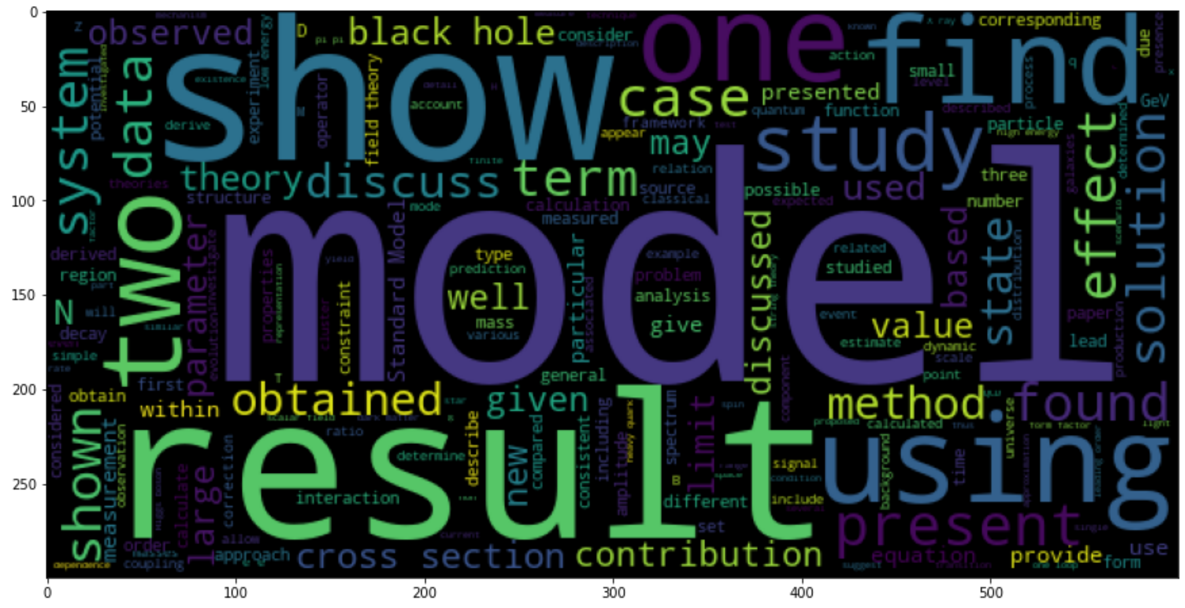
	title_len	abstract_len
count	26993.000000	26993.000000
mean	65.737636	723.981366
std	25.046536	355.316967
min	5.000000	21.000000
25%	48.000000	468.000000
50%	62.000000	655.000000
75%	81.000000	913.000000
max	245.000000	2495.000000

In [20]:

```
from wordcloud import WordCloud, STOPWORDS
```

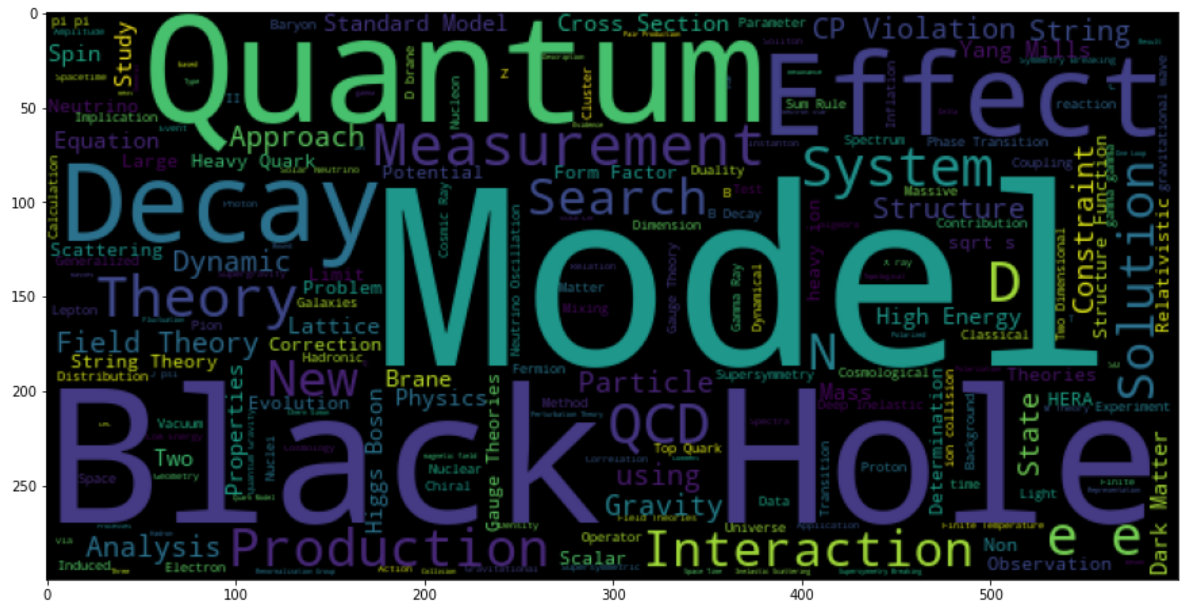
In [22]:

```
plt.figure(figsize=(15,9))
wc = WordCloud(width=600,height=300).generate(' '.join(data_new.abstract))
plt.imshow(wc);
```



In [23]:

```
plt.figure(figsize=(15,9))
wc = WordCloud(width=600,height=300).generate(' '.join(data_new.title))
plt.imshow(wc);
```



In [24]:

```
max_abstract_len = 800
max_title_len = 70
```

In [25]:

```
def text_preprocess(text):
    nopunc = [char for char in text if char not in string.punctuation]
    nopunc = ''.join(nopunc)
    return ' '.join([word for word in nopunc.split() if word.lower() not in stopwords.words
```

In [29]:

```
from tqdm.notebook import tqdm
tqdm.pandas()
```

In [31]:

```
import string
```

In [33]:

```
import re
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[33]:

```
True
```

In [35]:

```
from nltk.corpus import stopwords
```

In [36]:

```
data_new['abstract'] = data_new['abstract'].progress_apply(text_preprocess)
data_new['title'] = data_new['title'].progress_apply(text_preprocess)
```

```
100%                                      26993/26993 [15:37<00:00, 56.08it/s]

100%                                      26993/26993 [00:54<00:00, 517.39it/s]
```

In [37]:

```
from symspellpy import SymSpell, Verbosity
```

In [39]:

```
from nltk.stem import SnowballStemmer, WordNetLemmatizer
```

In [40]:

```
stemmer = SnowballStemmer('english')
lemmatizer = WordNetLemmatizer()
symspell = SymSpell()

def stem_words(text):
    return ' '.join([stemmer.stem(word) for word in text.split()])

def lemmatize_words(text):
    return ' '.join([lemmatizer.lemmatize(token) for token in text.split()])

def spelling_correction(text):
    correct_spellings = []

    for token in text.split():
        x = symspell.lookup(token,Verbosity.CLOSEST,max_edit_distance=2,include_unknown=Tru
        y = x.split(',')[0]
        correct_spellings.append(y)

    return ' '.join(correct_spellings)
```

In [41]:

```
data_new.abstract = data_new.abstract.progress_apply(stem_words)
data_new.abstract = data_new.abstract.progress_apply(lemmatize_words)
data_new.abstract = data_new.abstract.progress_apply(spelling_correction)
data_new['title'] = data_new['title'].progress_apply(stem_words)
data_new['title'] = data_new['title'].progress_apply(lemmatize_words)
data_new['title'] = data_new['title'].progress_apply(spelling_correction)
```

```
100%                                        26993/26993 [00:18<00:00, 1519.18it/s]

100%                                        26993/26993 [00:07<00:00, 4690.04it/s]

100%                                        26993/26993 [00:03<00:00, 10001.42it/s]

100%                                        26993/26993 [00:02<00:00, 14134.13it/s]

100%                                        26993/26993 [00:00<00:00, 73392.19it/s]

100%                                        26993/26993 [00:00<00:00, 71364.42it/s]
```

In [43]:

```
from keras_preprocessing.sequence import pad_sequences
from keras.preprocessing.text import Tokenizer
from keras.models import Sequential, load_model
from keras.utils.vis_utils import plot_model
from sklearn.model_selection import train_test_split
```

In [44]:

```
x_tokenizer = Tokenizer()
x_tokenizer.fit_on_texts(data_new.abstract)
```

In [45]:

```
y_tokenizer = Tokenizer()
y_tokenizer.fit_on_texts(data_new['title'])
```

In [46]:

```
abst_seq = x_tokenizer.texts_to_sequences(data_new.abstract)
title_seq = y_tokenizer.texts_to_sequences(data_new['title'])
```

In [47]:

```
abstract_vocab_size = len(x_tokenizer.word_index) + 1
abstract_vocab_size
```

Out[47]:

```
55260
```

In [48]:

```
title_vocab_size = len(y_tokenizer.word_index) + 1
title_vocab_size
```

Out[48]:

```
12770
```

In [49]:

```
abst_seq = pad_sequences(abst_seq,maxlen=max_abstract_len,padding='post')
title_seq = pad_sequences(title_seq,maxlen=max_title_len,padding='post')
```

In [50]:

```
def create_model(src_vocab,target_vocab,src_timesteps,target_timesteps,n_units):
    model = Sequential()
    model.add(Embedding(src_vocab,n_units,input_length=src_timesteps,mask_zero=True))
    model.add(LSTM(n_units,recurrent_dropout=0.1,dropout=0.15))
    model.add(RepeatVector(target_timesteps))
    model.add(LSTM(n_units,return_sequences=True,recurrent_dropout=0.2,dropout=0.2))
    model.add(BatchNormalization())
    model.add(TimeDistributed(Dense(target_vocab,activation='softmax')))
    return model
```

In [52]:

```
from keras.layers import LSTM, Dense, TimeDistributed, RepeatVector, Embedding, BatchNormal
from keras.callbacks import EarlyStopping, ReduceLROnPlateau
```

In [53]:

```
model = create_model(abstract_vocab_size,title_vocab_size,max_abstract_len,max_title_len,25
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 embedding (Embedding)       (None, 800, 256)          14146560  
                                                                 
 lstm (LSTM)                 (None, 256)               525312    
                                                                 
 repeat_vector (RepeatVector  (None, 70, 256)          0         
 )                                                               
                                                                 
 lstm_1 (LSTM)               (None, 70, 256)           525312    
                                                                 
 batch_normalization (BatchN  (None, 70, 256)          1024      
 ormalization)                                                   
                                                                 
 time_distributed (TimeDistr  (None, 70, 12770)        3281890   
 ibuted)                                                         
                                                                 
=================================================================
Total params: 18,480,098
Trainable params: 18,479,586
Non-trainable params: 512
_________________________________________________________________
```

In [55]:

```
model.compile(loss='sparse_categorical_crossentropy',optimizer='rmsprop',metrics='accuracy'
```

In [56]:

```
 X_train, X_test, y_train, y_test = train_test_split(abst_seq,title_seq,test_size=0.2,shuff
```

In [ ]:

```
es = EarlyStopping(monitor='val_accuracy',mode='max',verbose=1,patience=4)
rl = ReduceLROnPlateau(monitor='val_accuracy',mode='max',verbose=1,patience=2,factor=0.1,mi

r = model.fit(X_train,
             y_train,
             epochs=5,
             batch_size=32,
             callbacks=[es,rl],
             validation_data=(X_test,y_test))
```

In [ ]:

```
plt.figure(figsize=(12,8))
plt.plot(r.history['loss'],'r',label='train loss')
plt.plot(r.history['val_loss'],'b',label='test loss')
plt.xlabel('No. of Epochs')
plt.ylabel('Loss')
plt.title('Loss Graph')
plt.legend();
```

In [ ]:

```
plt.figure(figsize=(12,8))
plt.plot(r.history['accuracy'],'r',label='train accuracy')
plt.plot(r.history['val_accuracy'],'b',label='test accuracy')
plt.xlabel('No. of Epochs')
plt.ylabel('Accuracy')
plt.title('Accuracy Graph')
plt.legend();
```

In [ ]:

```
model.evaluate(X_test,y_test)
```

In [ ]:

```
model.save('abstract_title_generator.h5')
lm = load_model('abstract_title_generator.h5')
lm
```