In [ ]:

```
import os
import re
import nltk
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.tokenize import word_tokenize
from nltk import pos_tag
from nltk.corpus import stopwords, wordnet
from nltk.stem import WordNetLemmatizer
from sklearn.datasets import fetch_20newsgroups
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
import os
import nltk
import subprocess

# List of resources to download
resources = ["punkt", "averaged_perceptron_tagger", "wordnet", "stopwords", "omw

# Download and unzip resources if necessary
for resource in resources:
    try:
        nltk.data.find(f'{resource}.zip')
    except:
        nltk.download(resource, download_dir='/kaggle/working/')
        command = f"unzip /kaggle/working/corpora/{resource}.zip -d /kaggle/work
        subprocess.run(command.split())
        nltk.data.path.append('/kaggle/working/')

# Now you can import the NLTK resources as usual
from nltk.tokenize import word_tokenize
from nltk import pos_tag
from nltk.corpus import wordnet, stopwords
from nltk.stem import WordNetLemmatizer
```

```
2024-04-06 17:42:41.891713: E external/local_xla/xla/stream_executor/cuda/cuda_dn
n.cc:9261] Unable to register cuDNN factory: Attempting to register factory for p
lugin cuDNN when one has already been registered
2024-04-06 17:42:41.891810: E external/local_xla/xla/stream_executor/cuda/cuda_ff
t.cc:607] Unable to register cuFFT factory: Attempting to register factory for pl
ugin cuFFT when one has already been registered
2024-04-06 17:42:42.145214: E external/local_xla/xla/stream_executor/cuda/cuda_bl
as.cc:1515] Unable to register cuBLAS factory: Attempting to register factory for
plugin cuBLAS when one has already been registered
[nltk_data] Downloading package punkt to /kaggle/working/...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

```
unzip:  cannot find or open /kaggle/working/corpora/punkt.zip, /kaggle/working/co
rpora/punkt.zip.zip or /kaggle/working/corpora/punkt.zip.ZIP.
unzip:  cannot find or open /kaggle/working/corpora/averaged_perceptron_tagger.zi
p, /kaggle/working/corpora/averaged_perceptron_tagger.zip.zip or /kaggle/working/
corpora/averaged_perceptron_tagger.zip.ZIP.
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /kaggle/working/...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
[nltk_data] Downloading package wordnet to /kaggle/working/...
Archive:  /kaggle/working/corpora/wordnet.zip
   creating: /kaggle/working/corpora/wordnet/
  inflating: /kaggle/working/corpora/wordnet/lexnames
  inflating: /kaggle/working/corpora/wordnet/data.verb
  inflating: /kaggle/working/corpora/wordnet/index.adv
  inflating: /kaggle/working/corpora/wordnet/adv.exc
  inflating: /kaggle/working/corpora/wordnet/index.verb
  inflating: /kaggle/working/corpora/wordnet/cntlist.rev
  inflating: /kaggle/working/corpora/wordnet/data.adj
  inflating: /kaggle/working/corpora/wordnet/index.adj
  inflating: /kaggle/working/corpora/wordnet/LICENSE
  inflating: /kaggle/working/corpora/wordnet/citation.bib
  inflating: /kaggle/working/corpora/wordnet/noun.exc
  inflating: /kaggle/working/corpora/wordnet/verb.exc
  inflating: /kaggle/working/corpora/wordnet/README
  inflating: /kaggle/working/corpora/wordnet/index.sense
  inflating: /kaggle/working/corpora/wordnet/data.noun
  inflating: /kaggle/working/corpora/wordnet/data.adv
  inflating: /kaggle/working/corpora/wordnet/index.noun
  inflating: /kaggle/working/corpora/wordnet/adj.exc
[nltk_data] Downloading package stopwords to /kaggle/working/...
[nltk_data]   Unzipping corpora/stopwords.zip.
Archive:  /kaggle/working/corpora/stopwords.zip
[nltk_data] Downloading package omw-1.4 to /kaggle/working/...
```

```
replace /kaggle/working/corpora/stopwords/dutch? [y]es, [n]o, [A]ll, [N]one, [r]e
name:  NULL
(EOF or read error, treating as "[N]one" ...)
```

```
Archive:  /kaggle/working/corpora/omw-1.4.zip
   creating: /kaggle/working/corpora/omw-1.4/
   creating: /kaggle/working/corpora/omw-1.4/fin/
  inflating: /kaggle/working/corpora/omw-1.4/fin/LICENSE
  inflating: /kaggle/working/corpora/omw-1.4/fin/citation.bib
  inflating: /kaggle/working/corpora/omw-1.4/fin/wn-data-fin.tab
   creating: /kaggle/working/corpora/omw-1.4/heb/
  inflating: /kaggle/working/corpora/omw-1.4/heb/LICENSE
  inflating: /kaggle/working/corpora/omw-1.4/heb/citation.bib
  inflating: /kaggle/working/corpora/omw-1.4/heb/README
  inflating: /kaggle/working/corpora/omw-1.4/heb/wn-data-heb.tab
   creating: /kaggle/working/corpora/omw-1.4/slv/
  inflating: /kaggle/working/corpora/omw-1.4/slv/LICENSE
  inflating: /kaggle/working/corpora/omw-1.4/slv/citation.bib
  inflating: /kaggle/working/corpora/omw-1.4/slv/README
  inflating: /kaggle/working/corpora/omw-1.4/slv/wn-data-slv.tab
   creating: /kaggle/working/corpora/omw-1.4/ita/
  inflating: /kaggle/working/corpora/omw-1.4/ita/LICENSE
  inflating: /kaggle/working/corpora/omw-1.4/ita/citation.bib
  inflating: /kaggle/working/corpora/omw-1.4/ita/wn-data-ita.tab
 extracting: /kaggle/working/corpora/omw-1.4/ita/README
   creating: /kaggle/working/corpora/omw-1.4/nor/
  inflating: /kaggle/working/corpora/omw-1.4/nor/LICENSE
  inflating: /kaggle/working/corpora/omw-1.4/nor/citation.bib
  inflating: /kaggle/working/corpora/omw-1.4/nor/README
  inflating: /kaggle/working/corpora/omw-1.4/nor/wn-data-nno.tab
  inflating: /kaggle/working/corpora/omw-1.4/nor/wn-data-nob.tab
   creating: /kaggle/working/corpora/omw-1.4/als/
  inflating: /kaggle/working/corpora/omw-1.4/als/wn-data-als.tab
  inflating: /kaggle/working/corpora/omw-1.4/als/LICENSE
  inflating: /kaggle/working/corpora/omw-1.4/als/citation.bib
  inflating: /kaggle/working/corpora/omw-1.4/als/README
   creating: /kaggle/working/corpora/omw-1.4/pol/
  inflating: /kaggle/working/corpora/omw-1.4/pol/LICENSE
  inflating: /kaggle/working/corpora/omw-1.4/pol/citation.bib
  inflating: /kaggle/working/corpora/omw-1.4/pol/wn-data-pol.tab
   creating: /kaggle/working/corpora/omw-1.4/hrv/
 extracting: /kaggle/working/corpora/omw-1.4/hrv/LICENSE
  inflating: /kaggle/working/corpora/omw-1.4/hrv/citation.bib
  inflating: /kaggle/working/corpora/omw-1.4/hrv/wn-data-hrv.tab
  inflating: /kaggle/working/corpora/omw-1.4/hrv/README
  inflating: /kaggle/working/corpora/omw-1.4/citation.bib
   creating: /kaggle/working/corpora/omw-1.4/iwn/
  inflating: /kaggle/working/corpora/omw-1.4/iwn/LICENSE
  inflating: /kaggle/working/corpora/omw-1.4/iwn/citation.bib
  inflating: /kaggle/working/corpora/omw-1.4/iwn/wn-data-ita.tab
  inflating: /kaggle/working/corpora/omw-1.4/iwn/README
   creating: /kaggle/working/corpora/omw-1.4/nld/
  inflating: /kaggle/working/corpora/omw-1.4/nld/LICENSE
  inflating: /kaggle/working/corpora/omw-1.4/nld/wn-data-nld.tab
  inflating: /kaggle/working/corpora/omw-1.4/nld/citation.bib
   creating: /kaggle/working/corpora/omw-1.4/ron/
  inflating: /kaggle/working/corpora/omw-1.4/ron/LICENSE
  inflating: /kaggle/working/corpora/omw-1.4/ron/citation.bib
  inflating: /kaggle/working/corpora/omw-1.4/ron/wn-data-ron.tab
  inflating: /kaggle/working/corpora/omw-1.4/ron/README
   creating: /kaggle/working/corpora/omw-1.4/arb/
  inflating: /kaggle/working/corpora/omw-1.4/arb/LICENSE
  inflating: /kaggle/working/corpora/omw-1.4/arb/citation.bib
  inflating: /kaggle/working/corpora/omw-1.4/arb/README
```

```
inflating: /kaggle/working/corpora/omw-1.4/arb/wn-data-arb.tab
 creating: /kaggle/working/corpora/omw-1.4/isl/
inflating: /kaggle/working/corpora/omw-1.4/isl/LICENSE
inflating: /kaggle/working/corpora/omw-1.4/isl/citation.bib
inflating: /kaggle/working/corpora/omw-1.4/isl/README
inflating: /kaggle/working/corpora/omw-1.4/isl/wn-data-isl.tab
 creating: /kaggle/working/corpora/omw-1.4/swe/
inflating: /kaggle/working/corpora/omw-1.4/swe/LICENSE
inflating: /kaggle/working/corpora/omw-1.4/swe/citation.bib
inflating: /kaggle/working/corpora/omw-1.4/swe/README
inflating: /kaggle/working/corpora/omw-1.4/swe/wn-data-swe.tab
 creating: /kaggle/working/corpora/omw-1.4/por/
inflating: /kaggle/working/corpora/omw-1.4/por/LICENSE
inflating: /kaggle/working/corpora/omw-1.4/por/citation.bib
inflating: /kaggle/working/corpora/omw-1.4/por/wn-data-por.tab
inflating: /kaggle/working/corpora/omw-1.4/por/README
inflating: /kaggle/working/corpora/omw-1.4/README
 creating: /kaggle/working/corpora/omw-1.4/cow/
inflating: /kaggle/working/corpora/omw-1.4/cow/wn-data-cmn.tab
inflating: /kaggle/working/corpora/omw-1.4/cow/LICENSE
inflating: /kaggle/working/corpora/omw-1.4/cow/citation.bib
 creating: /kaggle/working/corpora/omw-1.4/jpn/
inflating: /kaggle/working/corpora/omw-1.4/jpn/LICENSE
inflating: /kaggle/working/corpora/omw-1.4/jpn/citation.bib
inflating: /kaggle/working/corpora/omw-1.4/jpn/README
inflating: /kaggle/working/corpora/omw-1.4/jpn/wn-data-jpn.tab
 creating: /kaggle/working/corpora/omw-1.4/dan/
inflating: /kaggle/working/corpora/omw-1.4/dan/LICENSE
inflating: /kaggle/working/corpora/omw-1.4/dan/citation.bib
inflating: /kaggle/working/corpora/omw-1.4/dan/wn-data-dan.tab
 creating: /kaggle/working/corpora/omw-1.4/slk/
inflating: /kaggle/working/corpora/omw-1.4/slk/LICENSE
inflating: /kaggle/working/corpora/omw-1.4/slk/citation.bib
inflating: /kaggle/working/corpora/omw-1.4/slk/wn-data-slk.tab
inflating: /kaggle/working/corpora/omw-1.4/slk/wn-data-lit.tab
inflating: /kaggle/working/corpora/omw-1.4/slk/README
 creating: /kaggle/working/corpora/omw-1.4/bul/
inflating: /kaggle/working/corpora/omw-1.4/bul/LICENSE
inflating: /kaggle/working/corpora/omw-1.4/bul/citation.bib
inflating: /kaggle/working/corpora/omw-1.4/bul/wn-data-bul.tab
inflating: /kaggle/working/corpora/omw-1.4/bul/README
 creating: /kaggle/working/corpora/omw-1.4/mcr/
inflating: /kaggle/working/corpora/omw-1.4/mcr/LICENSE
inflating: /kaggle/working/corpora/omw-1.4/mcr/citation.bib
inflating: /kaggle/working/corpora/omw-1.4/mcr/wn-data-eus.tab
inflating: /kaggle/working/corpora/omw-1.4/mcr/wn-data-cat.tab
inflating: /kaggle/working/corpora/omw-1.4/mcr/wn-data-glg.tab
inflating: /kaggle/working/corpora/omw-1.4/mcr/wn-data-spa.tab
 creating: /kaggle/working/corpora/omw-1.4/ell/
inflating: /kaggle/working/corpora/omw-1.4/ell/LICENSE
inflating: /kaggle/working/corpora/omw-1.4/ell/wn-data-ell.tab
inflating: /kaggle/working/corpora/omw-1.4/ell/README
 creating: /kaggle/working/corpora/omw-1.4/msa/
inflating: /kaggle/working/corpora/omw-1.4/msa/LICENSE
inflating: /kaggle/working/corpora/omw-1.4/msa/citation.bib
inflating: /kaggle/working/corpora/omw-1.4/msa/wn-data-zsm.tab
inflating: /kaggle/working/corpora/omw-1.4/msa/wn-data-ind.tab
inflating: /kaggle/working/corpora/omw-1.4/msa/README
 creating: /kaggle/working/corpora/omw-1.4/fra/
inflating: /kaggle/working/corpora/omw-1.4/fra/LICENSE
```

```
  inflating: /kaggle/working/corpora/omw-1.4/fra/citation.bib  
  inflating: /kaggle/working/corpora/omw-1.4/fra/wn-data-fra.tab  
   creating: /kaggle/working/corpora/omw-1.4/tha/
  inflating: /kaggle/working/corpora/omw-1.4/tha/LICENSE  
  inflating: /kaggle/working/corpora/omw-1.4/tha/citation.bib  
  inflating: /kaggle/working/corpora/omw-1.4/tha/wn-data-tha.tab  
```

In [ ]:

```python
newsgroup_train = fetch_20newsgroups(subset='train', shuffle=True)
newsgroup_test = fetch_20newsgroups(subset='test', shuffle=True)
print(newsgroup_train.target_names)
```

```
['alt.atheism', 'comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hard
ware', 'comp.sys.mac.hardware', 'comp.windows.x', 'misc.forsale', 'rec.autos', 'r
ec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey', 'sci.crypt', 'sci.elec
tronics', 'sci.med', 'sci.space', 'soc.religion.christian', 'talk.politics.guns',
'talk.politics.mideast', 'talk.politics.misc', 'talk.religion.misc']
```

In [ ]:

```python
df_train = pd.DataFrame({'article': newsgroup_train.data, 'label': newsgroup_tra
df_train.head()
```

Out[ ]:

	article	label
0	From: lerxst@wam.umd.edu (where's my thing)\nS...	7
1	From: guykuo@carson.u.washington.edu (Guy Kuo)...	4
2	From: twillis@ec.ecn.purdue.edu (Thomas E Will...	4
3	From: jgreen@amber (Joe Green)\nSubject: Re: W...	1
4	From: jcm@head-cfa.harvard.edu (Jonathan McDow...	14

In [ ]:

```python
df_test = pd.DataFrame({'article': newsgroup_test.data, 'label': newsgroup_test.
df_test.head()
```

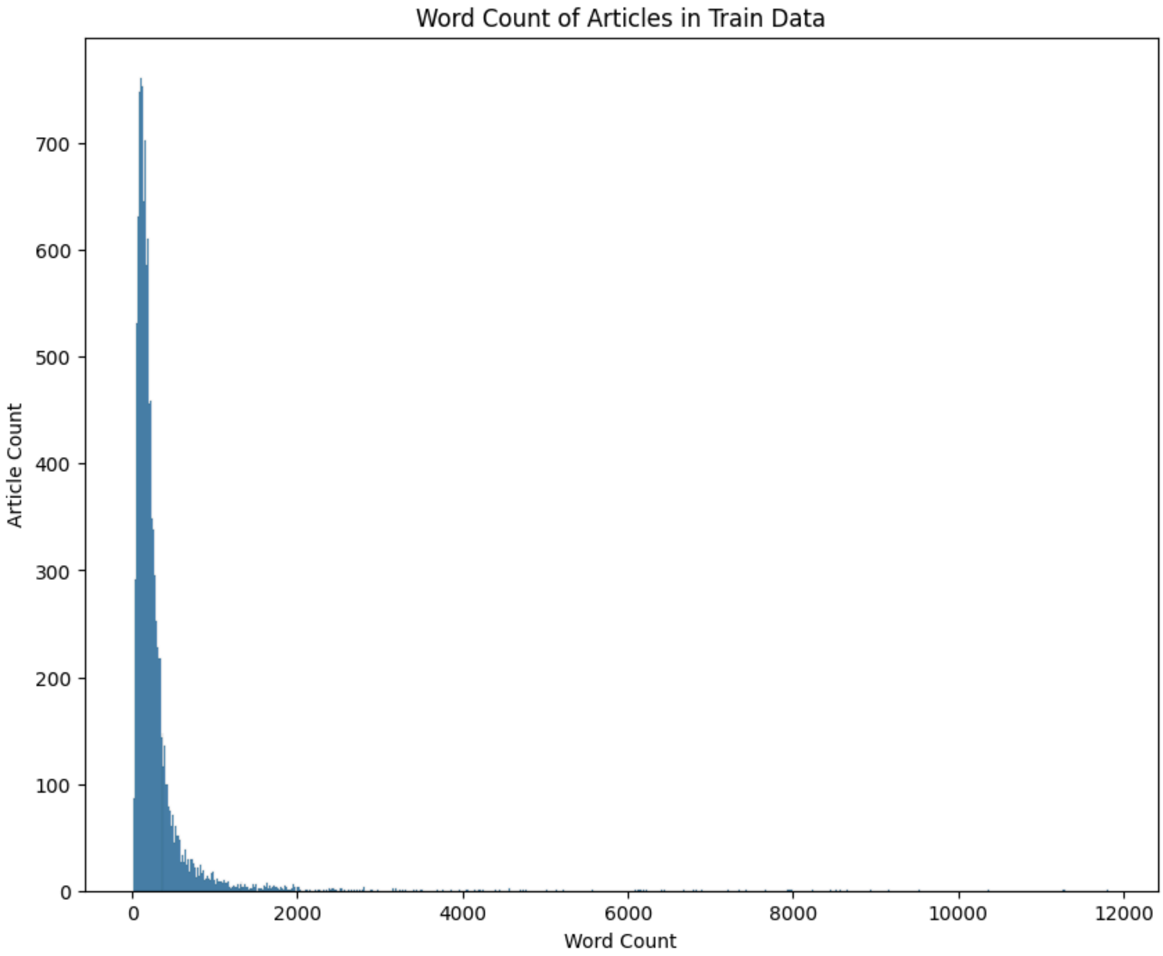
Out[ ]:

	article	label
0	From: v064mb9k@ubvmsd.cc.buffalo.edu (NEIL B. ...	7
1	From: Rick Miller <rick@ee.uwm.edu>\nSubject: ...	5
2	From: mathew <mathew@mantis.co.uk>\nSubject: R...	0
3	From: bakken@cs.arizona.edu (Dave Bakken)\nSub...	17
4	From: livesey@solntze.wpd.sgi.com (Jon Livesey...	19

In [ ]:

```python
df_train['word_count'] = df_train['article'].apply(lambda x: len(str(x).split())
plt.figure(figsize=(10,8))
sns.histplot(data=df_train, x='word_count')
plt.title('Word Count of Articles in Train Data')
plt.xlabel('Word Count')
plt.ylabel('Article Count')
plt.show()
```

```
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version. Conve
rt inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

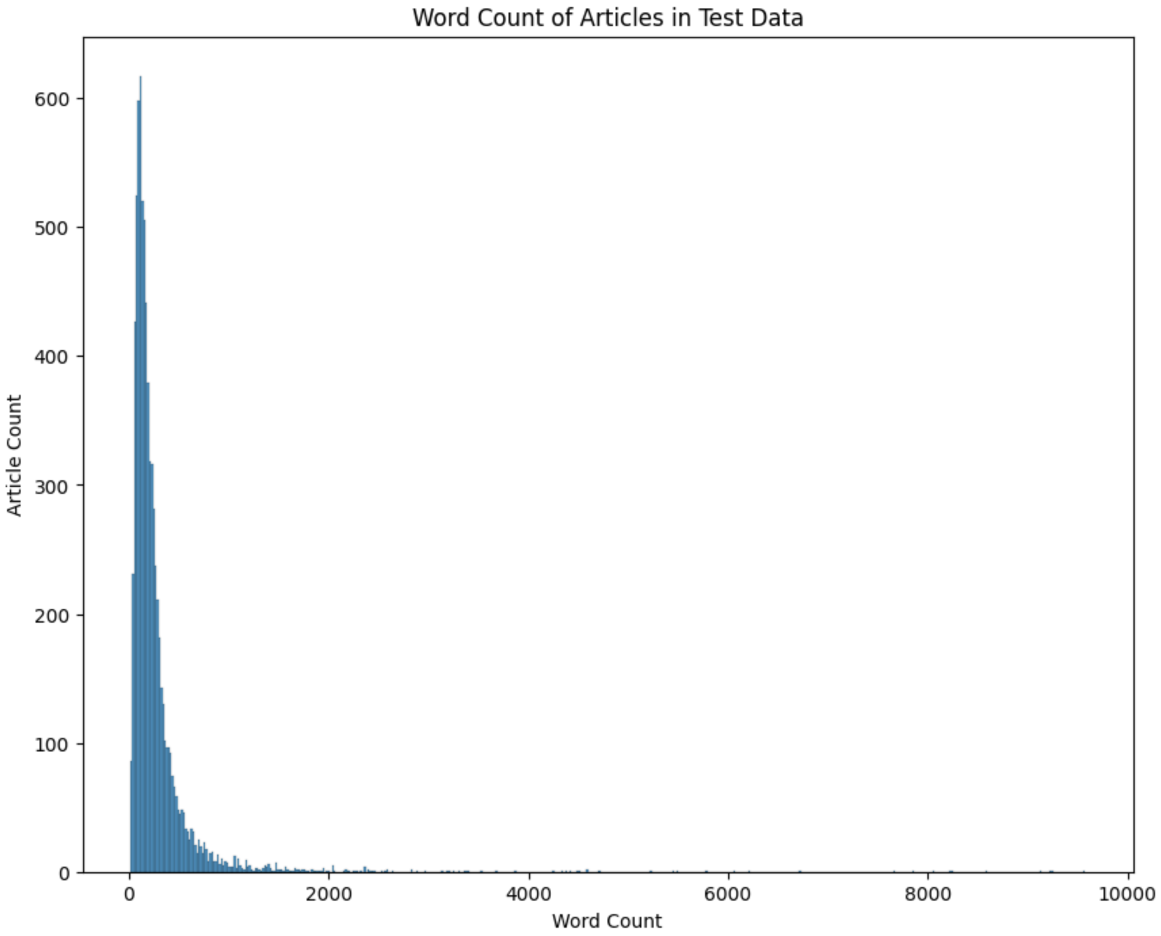


```
In [ ]: train_articles = (sum(df_train['word_count'] < 1000)/df_train.shape[0])*100
        print('Percentage of Training Articles having less than 1000 Words:{:.2f}%'.form
```

```
Percentage of Training Articles having less than 1000 Words:96.80%
```

```
In [ ]: df_test['word_count'] = df_test['article'].apply(lambda x: len(str(x).split()))
        plt.figure(figsize=(10,8))
        sns.histplot(data=df_test, x='word_count')
        plt.title('Word Count of Articles in Test Data')
        plt.xlabel('Word Count')
        plt.ylabel('Article Count')
        plt.show()
```

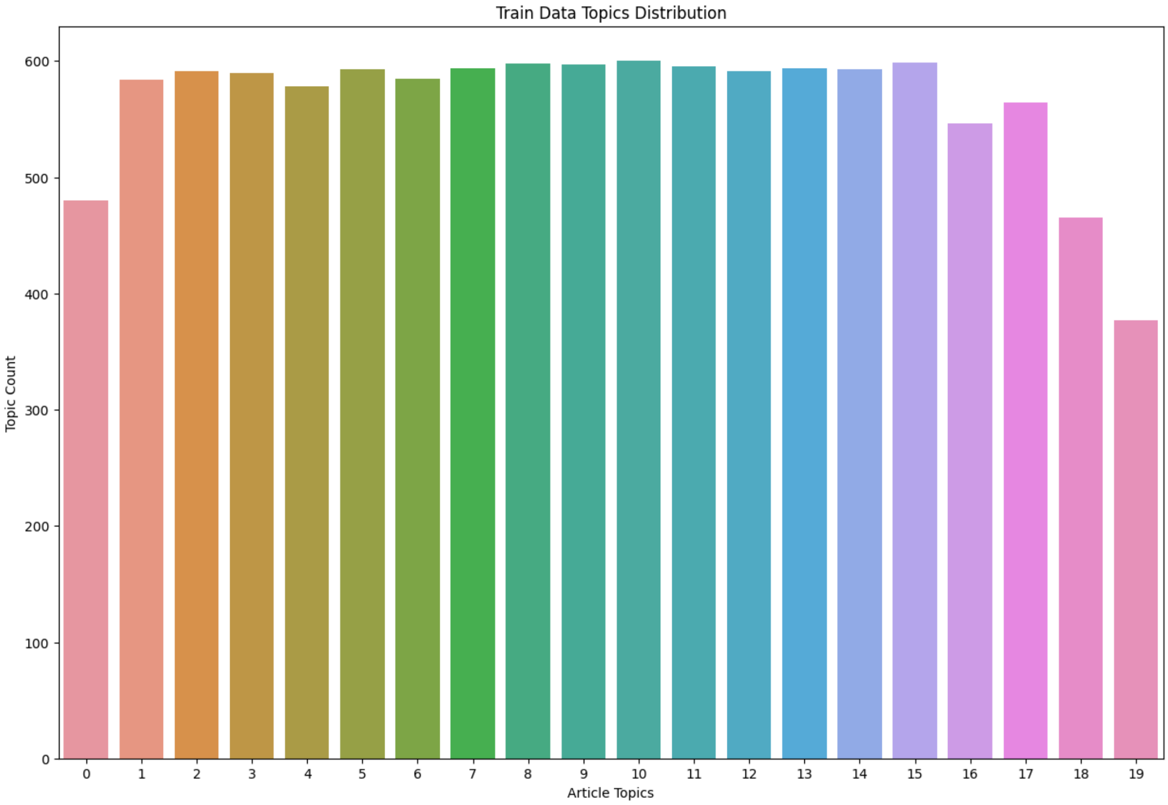
```
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version. Conve
rt inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



```
In [ ]: test_articles = (sum(df_test['word_count'] < 1000)/df_test.shape[0])*100
        print('Percentage of Test Articles having less than 1000 Words:{:.2f}%'.format(t
```

```
Percentage of Test Articles having less than 1000 Words:97.09%
```

```
In [ ]: plt.figure(figsize=(15,10))
        sns.countplot(data=df_train, x='label')
        plt.title('Train Data Topics Distribution')
        plt.xlabel('Article Topics')
        plt.ylabel('Topic Count')
        plt.show()
```



In [ ]:

```
def get_wordnet_pos (tag):
    if tag.startswith('J'):
        return wordnet.ADJ
    elif tag.startswith('V'):
        return wordnet.VERB
    elif tag.startswith('N'):
        return wordnet.NOUN
    elif tag.startswith('R'):
        return wordnet.ADV
    else:
        return wordnet.NOUN
def lemmatize (word_list):
    wl = WordNetLemmatizer()
    word_pos_tags = pos_tag(word_list)
    lemmatized_list = []
    for tag in word_pos_tags:
        lemmatize_word = wl.lemmatize(tag[0],get_wordnet_pos(tag[1]))
        lemmatized_list.append(lemmatize_word)
    return " ".join(lemmatized_list)
def clean_text (text):
    # Remove Pre and Post Spaces
    text = str(text).strip()

    # Lower case the entire text
    text = str(text).lower()

    # Substitute New Line Characters with spaces
    text = re.sub(r"\n", r" ", text)

    # Tokenize the sentence
    word_tokens = word_tokenize(text)

    # Remove the punctuation and  special characters from each individual word
    cleaned_text = []
    for word in word_tokens:
            cleaned_text.append("".join([char for char in word if char.isalnum()]))

    # Specify the stop words list
    stop_words = stopwords.words('english')

    # Remove the stopwords and words containing less then 2 characters
    text_tokens = [word for word in cleaned_text if (len(word) > 2) and (word no

    #Lemmatize each word in the word list
    text = lemmatize (text_tokens)

    return text
```

In [ ]:

```
df_train['article'][0]
```

Out[ ]:

```
"From: lerxst@wam.umd.edu (where's my thing)\nSubject: WHAT car is this!?\nNntp
-Posting-Host: rac3.wam.umd.edu\nOrganization: University of Maryland, College
Park\nLines: 15\n\n I was wondering if anyone out there could enlighten me on t
his car I saw\nthe other day. It was a 2-door sports car, looked to be from the
late 60s/\nearly 70s. It was called a Bricklin. The doors were really small. In
addition,\nthe front bumper was separate from the rest of the body. This is \na
ll I know. If anyone can tellme a model name, engine specs, years\nof productio
n, where this car is made, history, or whatever info you\nhave on this funky lo
oking car, please e-mail.\n\nThanks,\n- IL\n   ---- brought to you by your neig
hborhood Lerxst ----\n\n\n\n\n"
```

In [ ]:

```
clean_text (df_train['article'][0])
```

Out[ ]:

```
'lerxst wamumdedu thing subject car nntppostinghost rac3wamumdedu organization
university maryland college park line wonder anyone could enlighten car saw day
2door sport car look late 60 early 70 call bricklin door really small addition
front bumper separate rest body know anyone tellme model name engine spec year
production car make history whatever info funky look car please email thanks br
ing neighborhood lerxst'
```

In [ ]:

```
df_train['article'] = df_train['article'].apply(lambda x: clean_text(x))
```

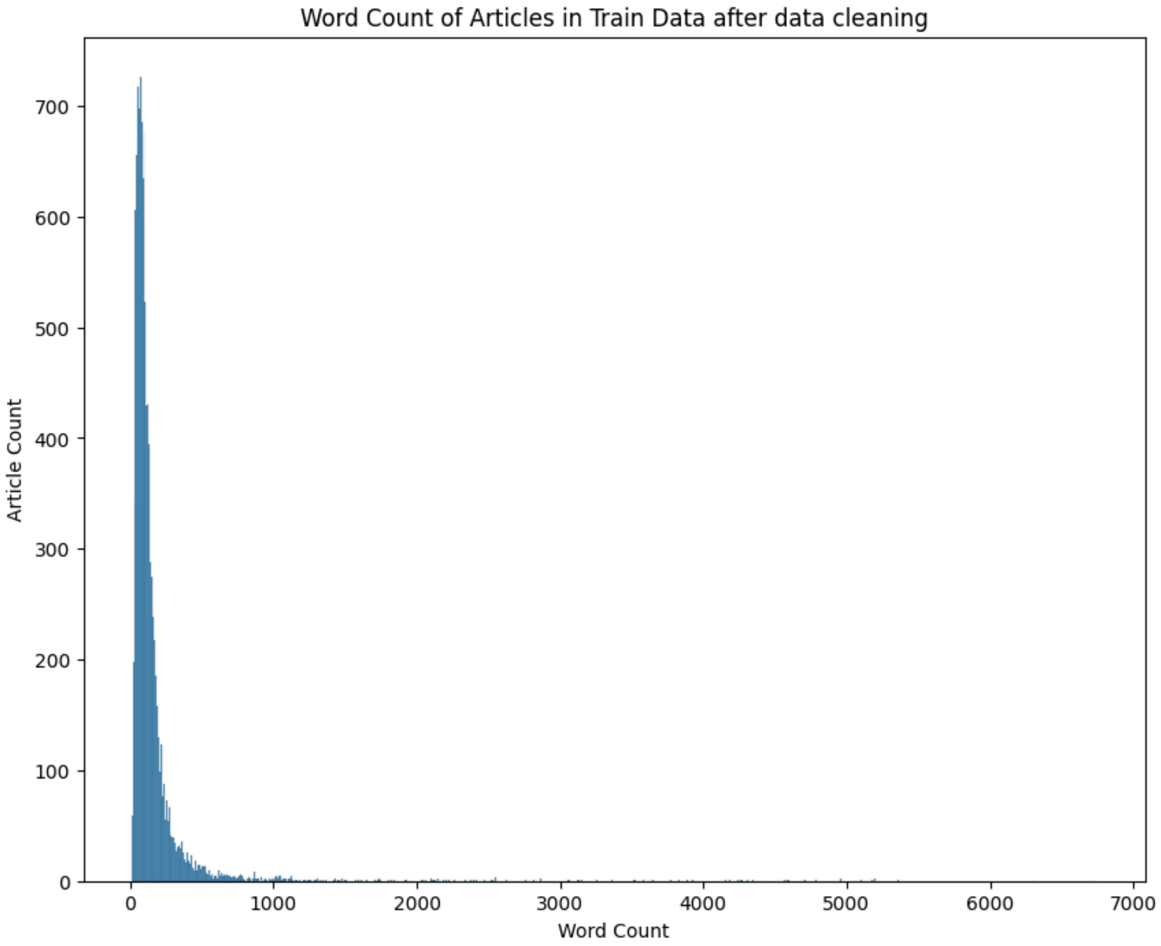
In [ ]:

```
df_test['article'] = df_test['article'].apply(lambda x: clean_text(x))
```

In [ ]:

```
df_train['word_count'] = df_train['article'].apply(lambda x: len(str(x).split())
plt.figure(figsize=(10,8))
sns.histplot(data=df_train, x='word_count')
plt.title('Word Count of Articles in Train Data after data cleaning')
plt.xlabel('Word Count')
plt.ylabel('Article Count')
plt.show()
```

```
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version. Conve
rt inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

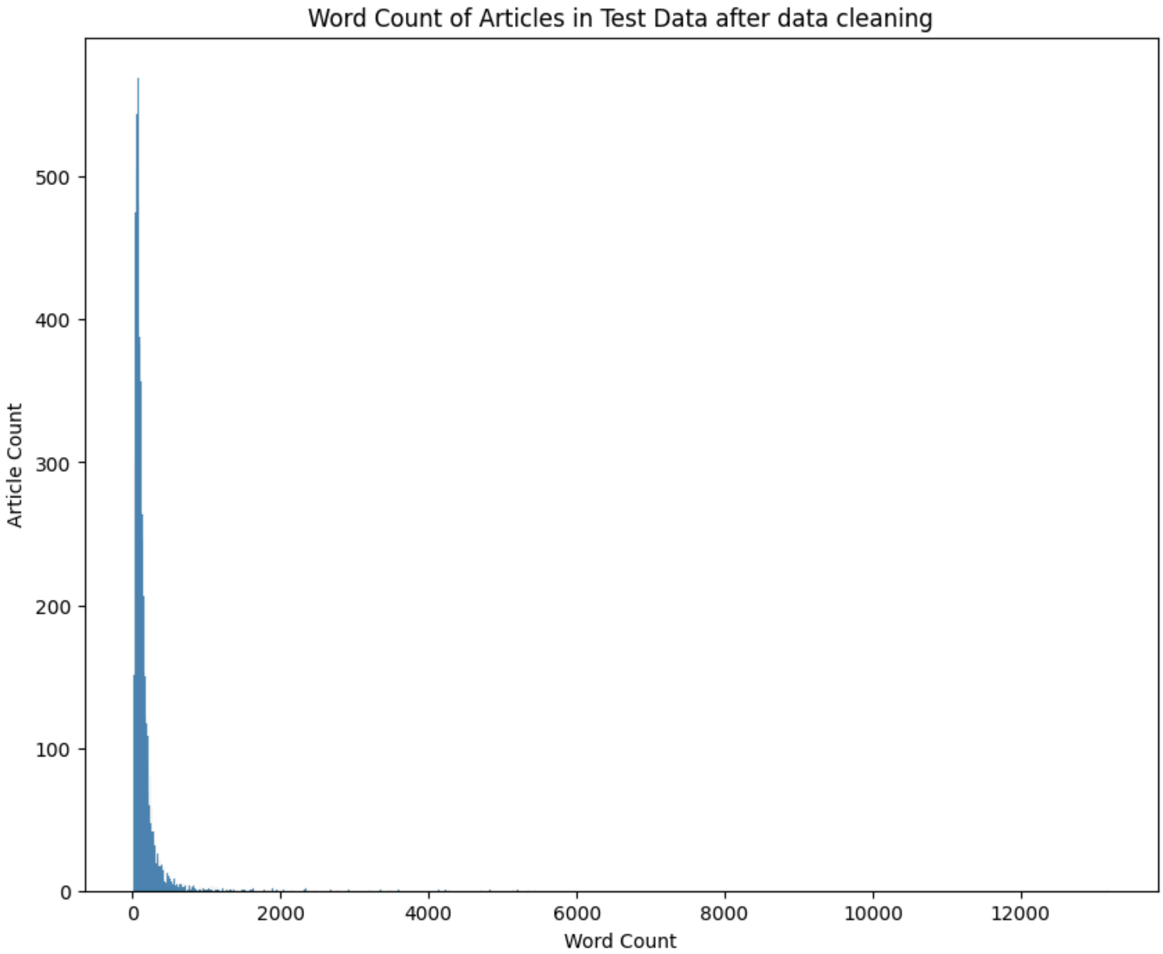


```
In [ ]: train_articles = (sum(df_train['word_count'] < 300)/df_train.shape[0])*100
        print('Percentage of Training Articles having less than 300 Words:{:.2f}%'.forma
```

```
Percentage of Training Articles having less than 300 Words:92.05%
```

```
In [ ]: df_test['word_count'] = df_test['article'].apply(lambda x: len(str(x).split()))
        plt.figure(figsize=(10,8))
        sns.histplot(data=df_test, x='word_count')
        plt.title('Word Count of Articles in Test Data after data cleaning')
        plt.xlabel('Word Count')
        plt.ylabel('Article Count')
        plt.show()
```

```
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version. Conve
rt inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



```
In [ ]: test_articles = (sum(df_test['word_count'] < 300)/df_test.shape[0])*100
        print('Percentage of Test Articles having less than 300 Words:{:.2f}%'.format(te
```

```
Percentage of Test Articles having less than 300 Words:92.37%
```

```
In [ ]: X_train = df_train['article']
        y_train = df_train['label']
        X_test = df_test['article']
        y_test = df_test['label']
        print("X_train:", X_train.shape)
        print("X_test:", X_test.shape)
        print("y_train:", y_train.shape)
        print("y_test:", y_test.shape)
```

```
X_train: (11314,)
X_test: (7532,)
y_train: (11314,)
y_test: (7532,)
```

```
In [ ]: tokenizer = Tokenizer(num_words=100000)
        tokenizer.fit_on_texts(X_train)
        tokenizer.index_word
```

```
Out[ ]:  {1: 'line',
          2: 'subject',
          3: 'organization',
          4: 'would',
          5: 'one',
          6: 'write',
          7: 'use',
          8: 'get',
          9: 'say',
          10: 'article',
          11: 'know',
          12: 'people',
          13: 'like',
          14: 'make',
          15: 'think',
          16: 'university',
          17: 'time',
          18: 'nntppostinghost',
          19: 'max',
          20: 'well',
          21: 'good',
          22: 'also',
          23: 'see',
          24: 'new',
          25: 'work',
          26: 'system',
          27: 'could',
          28: 'take',
          29: 'year',
          30: 'want',
          31: 'go',
          32: 'right',
          33: 'need',
          34: 'come',
          35: 'even',
          36: 'thing',
          37: 'problem',
          38: 'way',
          39: 'may',
          40: 'look',
          41: 'give',
          42: 'god',
          43: 'file',
          44: 'find',
          45: 'many',
          46: 'state',
          47: 'first',
          48: 'two',
          49: 'much',
          50: 'question',
          51: 'distribution',
          52: 'try',
          53: 'call',
          54: 'point',
          55: 'program',
          56: 'run',
          57: 'world',
          58: 'anyone',
          59: 'post',
          60: 'drive',
          61: 'believe',
          62: 'tell',
          63: 'mean',
          64: 'seem',
          65: 'number',
          66: 'computer',
          67: 'help',
          68: 'please',
          69: 'something',
          70: 'window',
          71: 'really',
          72: 'include',
          73: 'read',
          74: 'back',
          75: 'since',
          76: 'day',
          77: 'case',
          78: 'email',
          79: 'still',
          80: 'information',
          81: 'game',
          82: 'key',
          83: 'law',
          84: 'government',
          85: 'part',
          86: 'start',
          87: 'last',
          88: 'must',
          89: 'group',
          90: 'thanks',
          91: 'usa',
          92: 'never',
          93: 'let',
          94: 'ask',
          95: 'might',
          96: 'replyto',
          97: 'car',
          98: 'support',
          99: 'another',
          100: 'sure',
          101: 'without',
          102: 'follow',
          103: 'space',
          104: 'version',
          105: 'set',
          106: 'name',
          107: 'david',
          108: 'etc',
          109: 'keep',
          110: 'long',
          111: 'power',
          112: 'put',
          113: 'fact',
          114: 'data',
          115: 'science',
          116: 'someone',
          117: 'great',
          118: 'available',
          119: 'do',
          120: 'reason',
          121: 'list',
          122: 'card',
          123: 'send',
          124: 'team',
          125: 'lot',
          126: 'show',
          127: 'change',
          128: 'high',
          129: 'christian',
          130: 'gun',
          131: 'little',
          132: 'john',
          133: 'chip',
          134: 'bad',
          135: 'place',
          136: 'however',
          137: 'play',
          138: 'software',
          139: 'opinion',
          140: 'anything',
          141: 'around',
          142: 'every',
          143: 'probably',
          144: 'course',
          145: 'leave',
          146: 'best',
          147: 'true',
          148: 'word',
          149: 'consider',
          150: 'book',
          151: 'happen',
          152: 'end',
          153: 'life',
          154: 'old',
          155: 'public',
          156: 'technology',
          157: 'least',
          158: 'second',
          159: 'different',
          160: 'kill',
          161: 'talk',
          162: 'bit',
          163: 'claim',
          164: 'live',
          165: 'enough',
          166: 'order',
          167: 'note',
          168: 'center',
          169: 'research',
          170: 'provide',
          171: 'image',
          172: 'base',
          173: 'writes',
          174: 'buy',
          175: 'jesus',
          176: 'control',
          177: '1993',
          178: 'idea',
          179: 'message',
          180: 'hard',
          181: 'source',
          182: 'service',
          183: 'issue',
          184: 'far',
          185: 'armenian',
          186: 'possible',
          187: 'actually',
          188: 'example',
          189: 'either',
          190: 'though',
          191: 'big',
          192: 'inc',
          193: 'real',
          194: 'answer',
          195: 'cause',
          196: 'person',
          197: 'b8f',
          198: 'child',
          199: 'rather',
          200: 'nothing',
          201: 'mail',
          202: 'next',
          203: 'mark',
          204: 'driver',
          205: 'internet',
          206: 'else',
          207: 'machine',
          208: 'american',
          209: 'wrong',
          210: 'standard',
          211: 'free',
          212: 'access',
          213: 'man',
          214: 'address',
          215: 'exist',
          216: 'phone',
          217: 'large',
          218: 'build',
          219: 'a86',
          220: 'allow',
          221: 'yes',
          222: 'human',
          223: 'disk',
          224: 'maybe',
          225: 'win',
          226: 'bill',
          227: 'national',
          228: 'player',
          229: 'code',
          230: 'able',
          231: 'user',
          232: 'others',
          233: 'always',
          234: 'hand',
          235: 'turn',
          236: 'report',
          237: 'hear',
          238: 'price',
          239: 'info',
          240: 'type',
          241: 'keywords',
          242: 'require',
          243: 'kind',
          244: 'several',
          245: 'today',
          246: 'general',
          247: 'israel',
          248: 'small',
          249: 'home',
          250: 'area',
          251: 'yet',
          252: 'sound',
          253: 'less',
          254: 'view',
          255: 'quite',
          256: 'ever',
          257: 'sale',
          258: '145',
          259: 'three',
          260: 'pay',
          261: 'result',
          262: 'cost',
          263: 'sell',
          264: 'become',
          265: 'away',
          266: 'open',
          267: 'application',
          268: 'week',
          269: 'test',
          270: 'remember',
          271: 'speed',
          272: 'check',
          273: 'move',
          274: 'news',
          275: 'company',
          276: 'create',
          277: 'study',
          278: 'color',
          279: 'president',
          280: 'hold',
          281: 'country',
          282: 'whether',
          283: 'current',
          284: 'steve',
          285: 'mac',
          286: 'side',
          287: 'feel',
          288: 'design',
          289: 'encryption',
          290: 'agree',
          291: 'already',
          292: 'money',
          293: 'michael',
          294: 'war',
          295: 'understand',
          296: 'department',
          297: 'evidence',
          298: 'netcomcom',
          299: 'value',
          300: 'force',
          301: 'display',
          302: 'institute',
          303: 'rule',
          304: 'argument',
          305: 'graphic',
          306: 'assume',
          307: 'matter',
          308: 'lead',
          309: 'love',
          310: 'stop',
          311: 'box',
          312: 'offer',
          313: 'local',
          314: 'ago',
          315: 'jew',
          316: 'apr',
          317: 'low',
          318: 'mention',
          319: 'city',
          320: 'bible',
          321: 'server',
          322: 'add',
          323: 'perhaps',
          324: 'copy',
          325: 'memory',
          326: 'experience',
          327: 'house',
          328: 'robert',
          329: 'woman',
          330: 'clipper',
          331: 'act',
          332: 'fax',
          333: 'hope',
          334: 'package',
          335: 'guy',
          336: 'difference',
          337: 'care',
          338: 'mind',
          339: 'whole',
          340: 'close',
          341: 'pretty',
          342: 'lose',
          343: 'april',
          344: 'stuff',
          345: 'interest',
          346: 'mike',
          347: 'return',
          348: 'attack',
          349: 'paul',
          350: 'begin',
          351: 'network',
          352: 'job',
          353: 'communication',
          354: 'die',
          355: 'expect',
          356: 'member',
          357: 'jim',
          358: 'church',
          359: 'deal',
          360: 'carry',
          361: 'israeli',
          362: 'turkish',
          363: 'contact',
          364: 'interested',
          365: 'device',
          366: 'religion',
          367: 'appear',
          368: 'head',
          369: 'sun',
          370: 'death',
          371: 'bike',
          372: 'save',
          373: 'canada',
          374: 'model',
          375: 'everything',
          376: 'product',
          377: 'important',
          378: 'month',
          379: 'comment',
          380: 'accept',
          381: 'school',
          382: 'fire',
          383: 'everyone',
          384: 'error',
          385: 'fast',
          386: 'hit',
          387: 'rate',
          388: 'level',
          389: 'original',
          390: 'light',
          391: 'easy',
          392: 'action',
          393: 'truth',
          394: 'guess',
          395: 'often',
          396: 'white',
          397: 'almost',
          398: 'monitor',
          399: 'sort',
          400: 'effect',
          401: 'scsi',
          402: 'articleid',
          403: 'advance',
          404: 'reference',
          405: 'form',
          406: 'simply',
          407: '1d9',
          408: 'friend',
          409: 'format',
          410: 'weapon',
          411: 'speak',
          412: 'full',
          413: 'video',
          414: 'body',
          415: 'board',
          416: 'engineering',
          417: 'dept',
          418: 'statement',
          419: 'wonder',
          420: 'bring',
          421: 'cover',
          422: 'season',
          423: 'arm',
          424: 'position',
          425: 'size',
          426: 'instead',
          427: 'although',
          428: 'certainly',
          429: 'history',
          430: 'division',
          431: 'california',
          432: 'plan',
          433: 'anybody',
          434: 'regard',
          435: 'couple',
          436: 'single',
          437: 'ground',
          438: 'anyway',
          439: 'xnewsreader',
          440: 'discussion',
          441: 'college',
          442: 'summary',
          443: 'men',
          444: 'later',
          445: 'hell',
          446: 'output',
          447: 'suggest',
          448: 'mode',
          449: 'correct',
          450: 'receive',
          451: 'press',
          452: 'event',
          453: 'ftp',
          454: 'explain',
          455: 'sense',
          456: 'project',
          457: 'crime',
          458: 'unless',
          459: 'security',
          460: 'black',
          461: 'present',
          462: 'drug',
          463: 'break',
          464: 'top',
          465: 'appreciate',
          466: 'function',
          467: 'hockey',
          468: '100',
          469: 'process',
          470: 'situation',
          471: 'entry',
          472: 'clinton',
          473: 'release',
          474: 'major',
          475: 'similar',
          476: 'reply',
          477: 'site',
          478: 'certain',
          479: 'faith',
          480: 'apple',
          481: 'continue',
          482: 'san',
          483: 'unix',
          484: 'earth',
          485: 'net',
          486: 'individual',
          487: 'term',
          488: 'purpose',
          489: 'face',
          490: 'clear',
          491: 'period',
          492: 'within',
          493: 'request',
          494: 'quote',
          495: 'likely',
          496: 'private',
          497: 'road',
          498: 'late',
          499: 'police',
          500: 'policy',
          501: 'goal',
          502: 'suppose',
          503: 'figure',
          504: 'jewish',
          505: 'record',
          506: 'learn',
          507: 'office',
          508: 'stand',
          509: 'nice',
          510: 'land',
          511: 'date',
          512: 'decide',
          513: 'christ',
          514: 'simple',
          515: 'via',
          516: 'faq',
          517: 'usually',
          518: 'screen',
          519: 'hardware',
          520: 'atheist',
          521: 'protect',
          522: 'strong',
          523: 'exactly',
          524: 'saw',
          525: 'except',
          526: 'involve',
          527: 'young',
          528: 'especially',
          529: 'windows',
          530: 'dave',
          531: 'early',
          532: 'heard',
          533: 'response',
          534: 'fan',
          535: 'mine',
          536: 'washington',
          537: 'section',
          538: 'sorry',
          539: 'keith',
          540: 'nasa',
          541: 'york',
          542: 'wait',
          543: 'text',
          544: 'detail',
          545: 'tax',
          546: 'per',
          547: 'gmt',
          548: 'society',
          549: 'widget',
          550: 'million',
          551: 'pick',
          552: 'short',
          553: 'health',
          554: 'corporation',
          555: 'watch',
          556: 'tin',
          557: 'bank',
          558: 'fine',
          559: 'dod',
          560: 'common',
          561: 'pittsburgh',
          562: 'limit',
          563: 'page',
          564: 'western',
          565: 'business',
          566: 'league',
          567: 'thus',
          568: 'night',
          569: 'dead',
          570: 'cut',
          571: 'launch',
          572: 'condition',
          573: 'attempt',
          574: 'radio',
          575: 'story',
          576: 'food',
          577: 'increase',
          578: 'particular',
          579: 'bob',
          580: 'brian',
          581: 'manager',
          582: 'cheap',
          583: 'apply',
          584: 'rest',
          585: 'produce',
          586: 'port',
          587: 'among',
          588: 'bus',
          589: 'option',
          590: 'ibm',
          591: 'pass',
          592: 'belief',
          593: 'air',
          594: 'political',
          595: 'score',
          596: 'james',
          597: 'concern',
          598: 'contain',
          599: 'water',
          600: 'red',
          601: 'mouse',
          602: 'express',
          603: 'handle',
          604: 'fail',
          605: 'command',
          606: 'court',
          607: 'define',
          608: 'therefore',
          609: 'chance',
          610: 'moral',
          611: 'method',
          612: 'third',
          613: 'tape',
          614: 'accord',
          615: 'future',
          616: 'field',
          617: 'whatever',
          618: 'draw',
          619: 'compare',
          620: 'switch',
          621: 'past',
          622: 'military',
          623: 'controller',
          624: 'toronto',
          625: 'smith',
          626: 'paper',
          627: 'unit',
          628: 'due',
          629: 'authority',
          630: 'wire',
          631: 'theory',
          632: 'texas',
          633: 'author',
          634: 'king',
          635: 'anonymous',
          636: 'develop',
          637: 'miss',
          638: 'front',
          639: 'personal',
          640: 'shot',
          641: 'directory',
          642: 'total',
          643: 'engine',
          644: 'tool',
          645: 'object',
          646: 'solution',
          647: 'andrew',
          648: 'four',
          649: 'criminal',
          650: 'library',
          651: 'peter',
          652: 'final',
          653: 'frank',
          654: 'sometimes',
          655: 'special',
          656: 'flame',
          657: 'upon',
          658: 'family',
          659: 'medium',
          660: 'specific',
          661: 'murder',
          662: 'voice',
          663: 'ram',
          664: 'bear',
          665: 'federal',
          666: 'tom',
          667: 'recently',
          668: 'chicago',
          669: 'fall',
          670: 'algorithm',
          671: 'sign',
          672: 'agency',
          673: 'worth',
          674: 'series',
          675: 'describe',
          676: 'trade',
          677: 'resource',
          678: 'soon',
          679: 'baseball',
          680: 'behind',
          681: 'greek',
          682: 'near',
          683: 'secret',
          684: 'judge',
          685: 'richard',
          686: 'letter',
          687: 'class',
          688: 'along',
          689: 'together',
          690: 'choose',
          691: 'international',
          692: 'motif',
          693: 'plus',
          694: 'complete',
          695: 'wish',
          696: 'scott',
          697: 'muslim',
          698: 'interface',
          699: 'font',
          700: 'party',
          701: 'technical',
          702: 'religious',
          703: 'feature',
          704: 'official',
          705: 'share',
          706: 'station',
          707: 'citizen',
          708: 'lie',
          709: 'amount',
          710: 'peace',
          711: 'previous',
          712: 'firearm',
          713: 'account',
          714: 'delete',
          715: '1992',
          716: 'doubt',
          717: 'meet',
          718: 'prove',
          719: 'father',
          720: 'legal',
          721: 'administration',
          722: 'russian',
          723: 'picture',
          724: 'market',
          725: 'approach',
          726: 'various',
          727: 'laboratory',
          728: 'arab',
          729: 'privacy',
          730: 'necessary',
          731: 'compute',
          732: 'knowledge',
          733: 'block',
          734: 'occur',
          735: 'development',
          736: 'manual',
          737: 'minute',
          738: 'disclaimer',
          739: 'medical',
          740: 'currently',
          741: 'choice',
          742: 'nhl',
          743: 'performance',
          744: 'average',
          745: 'slow',
          746: 'sin',
          747: 'printer',
          748: 'notice',
          749: 'thought',
          750: 'fix',
          751: 'age',
          752: 'chris',
          753: 'cable',
          754: 'avoid',
          755: 'otherwise',
          756: 'population',
          757: 'north',
          758: 'thank',
          759: 'insurance',
          760: 'forget',
          761: 'supply',
          762: 'quality',
          763: 'defense',
          764: 'replace',
          765: 'burn',
          766: 'title',
          767: 'remove',
          768: 'thomas',
          769: 'germany',
          770: 'none',
          771: 'spend',
          772: 'outside',
          773: 'univ',
          774: 'operation',
          775: 'hour',
          776: 'owner',
          777: 'effort',
          778: 'clearly',
          779: 'ide',
          780: 'fight',
          781: 'fit',
          782: 'charge',
          783: 'son',
          784: 'community',
          785: 'doctor',
          786: 'freedom',
          787: 'christianity',
          788: 'shall',
          789: 'remain',
          790: 'eric',
          791: 'united',
          792: 'language',
          793: 'input',
          794: 'objective',
          795: 'stay',
          796: 'serial',
          797: 'modem',
          798: 'purchase',
          799: 'sit',
          800: 'pat',
          801: 'vote',
          802: 'document',
          803: 'activity',
          804: 'online',
          805: 'serious',
          806: 'fbi',
          807: 'realize',
          808: 'load',
          809: 'america',
          810: 'publish',
          811: 'print',
          812: 'search',
          813: 'practice',
          814: 'prevent',
          815: 'basic',
          816: 'main',
          817: 'convert',
          818: 'newsgroup',
          819: 'digital',
          820: 'refer',
          821: 'eye',
          822: 'george',
          823: 'morality',
          824: 'willing',
          825: 'commercial',
          826: 'keyboard',
          827: 'gas',
          828: 'count',
          829: 'street',
          830: 'gary',
          831: 'kid',
          832: 'completely',
          833: 'armenia',
          834: 'blue',
          835: 'gordon',
          836: 'student',
          837: 'drop',
          838: 'jon',
          839: 'inside',
          840: 'ship',
          841: 'turkey',
          842: 'boston',
          843: 'half',
          844: 'safety',
          845: 'depend',
          846: 'satellite',
          847: 'orbit',
          848: 'serve',
          849: 'grant',
          850: 'nature',
          851: 'decision',
          852: 'lack',
          853: 'existence',
          854: 'respond',
          855: 'material',
          856: 'suggestion',
          857: 'normal',
          858: 'tim',
          859: 'determine',
          860: 'secure',
          861: 'mass',
          862: 'south',
          863: 'dan',
          864: 'argue',
          865: 'disease',
          866: 'reach',
          867: 'beat',
          868: 'stephanopoulos',
          869: 'corp',
          870: 'lab',
          871: 'scientific',
          872: 'transfer',
          873: 'mile',
          874: 'trust',
          875: 'thousand',
          876: 'range',
          877: 'connect',
          878: 'fund',
          879: 'indeed',
          880: 'congress',
          881: 'finally',
          882: 'obtain',
          883: 'adam',
          884: 'archive',
          885: 'dealer',
          886: 'uunet',
          887: 'room',
          888: 'lord',
          889: 'useful',
          890: 'throw',
          891: 'star',
          892: 'mission',
          893: 'turk',
          894: 'easily',
          895: 'matthew',
          896: 'door',
          897: 'inreplyto',
          898: 'msg',
          899: 'definition',
          900: 'reasonable',
          901: 'west',
          902: 'rid',
          903: 'generally',
          904: 'advice',
          905: 'happy',
          906: 'obviously',
          907: 'moon',
          908: 'intend',
          909: 'raise',
          910: 'internal',
          911: 'usenet',
          912: 'amendment',
          913: 'directly',
          914: 'ten',
          915: 'nation',
          916: 'discuss',
          917: 'difficult',
          918: 'education',
          919: 'stupid',
          920: 'wing',
          921: '550',
          922: 'addition',
          923: 'necessarily',
          924: 'illinois',
          925: 'respect',
          926: 'conference',
          927: 'doug',
          928: 'magazine',
          929: 'reserve',
          930: 'character',
          931: 'shoot',
          932: 'unfortunately',
          933: 'direct',
          934: 'giz',
          935: 'instal',
          936: 'vehicle',
          937: 'license',
          938: 'los',
          939: 'blood',
          940: 'enforcement',
          941: 'imagine',
          942: 'basis',
          943: 'henry',
          944: 'floppy',
          945: 'store',
          946: 'joe',
          947: 'trouble',
          948: 'obvious',
          949: 'entire',
          950: 'playoff',
          951: 'somebody',
          952: 'reduce',
          953: 'signal',
          954: 'roger',
          955: 'measure',
          956: 'oil',
          957: 'conclusion',
          958: 'east',
          959: 'circuit',
          960: 'wife',
          961: 'electronic',
          962: 'folk',
          963: 'neither',
          964: 'item',
          965: 'evil',
          966: 'associate',
          967: 'pull',
          968: 'heart',
          969: 'colorado',
          970: 'trial',
          971: 'excellent',
          972: 'apparently',
          973: 'aid',
          974: 'risk',
          975: 'hole',
          976: 'link',
          977: 'recent',
          978: 'park',
          979: 'stick',
          980: 'suspect',
          981: 'ride',
          982: 'client',
          983: 'dog',
          984: 'van',
          985: 'alone',
          986: 'upgrade',
          987: 'round',
          988: 'step',
          989: 'originator',
          990: 'suffer',
          991: 'environment',
          992: 'appropriate',
          993: 'whose',
          994: 'ron',
          995: 'soldier',
          996: 'ability',
          997: 'commit',
          998: 'ken',
          999: 'listen',
          1000: 'btw',
          ...}
```

In [ ]:

```
vocab_size = len(tokenizer.index_word) + 1
print('Vocab Size:', vocab_size)
```

```
Vocab Size: 150641
```

In [ ]:

```
X_train_token = tokenizer.texts_to_sequences(X_train)
X_test_token = tokenizer.texts_to_sequences(X_test)
```

In [ ]:

```
print("First Intance Text:\n")
print(X_train[0])
print("\nFirst Intance Total Words:", len(str(X_train[0]).split()))
```

```
First Intance Text:

lerxst wamumdedu thing subject car nntppostinghost rac3wamumdedu organization uni
versity maryland college park line wonder anyone could enlighten car saw day 2doo
r sport car look late 60 early 70 call bricklin door really small addition front
bumper separate rest body know anyone tellme model name engine spec year producti
on car make history whatever info funky look car please email thanks bring neighb
orhood lerxst

First Intance Total Words: 62
```

In [ ]:
```
print("First Intance Text Sequence:\n")
print(X_train_token[0])
print("\nFirst Intance Text Sequence Length:", len(X_train_token[0]))
```

```
First Intance Text Sequence:

[26797, 4580, 36, 2, 97, 18, 18381, 3, 16, 2160, 441, 978, 1, 419, 58, 27, 5471,
97, 524, 76, 18382, 1039, 97, 40, 498, 9294, 531, 7168, 53, 26798, 896, 71, 248,
922, 638, 5270, 1124, 584, 414, 11, 58, 41507, 374, 106, 643, 1919, 29, 1950, 97,
14, 429, 617, 239, 18383, 40, 97, 68, 78, 90, 420, 4068, 26797]

First Intance Text Sequence Length: 62
```

In [ ]:
```
print("Second Intance Text:\n")
print(X_train[1])
print("\nSecond Intance Total Words:", len(str(X_train[1]).split()))
```

```
Second Intance Text:

guykuo carsonuwashingtonedu guy kuo subject clock poll final call summary final c
all clock report keywords acceleration clock upgrade articleid shelley1qvfo9innc3
s organization university washington line nntppostinghost carsonuwashingtonedu fa
ir number brave soul upgrade clock oscillator share experience poll please send b
rief message detail experience procedure top speed attain cpu rat speed add card
adapter heat sink hour usage per day floppy disk functionality 800 floppy especia
lly request summarize next two day please add network knowledge base do clock upg
rade answer poll thanks guy kuo guykuo uwashingtonedu

Second Intance Total Words: 84
```

In [ ]:
```
print("Second Intance Text Sequence:\n")
print(X_train_token[1])
print("\nSecond Intance Text Sequence Length:", len(X_train_token[1]))
```

```
Second Intance Text Sequence:

[10658, 3058, 335, 7841, 2, 1004, 3089, 652, 53, 442, 652, 53, 1004, 236, 241, 35
65, 1004, 986, 402, 62688, 3, 16, 536, 1, 18, 3058, 1258, 65, 1330, 1331, 986, 10
04, 5967, 705, 326, 3089, 68, 123, 2076, 179, 544, 326, 1819, 464, 271, 7842, 125
2, 2217, 271, 322, 122, 1837, 1617, 4186, 775, 2317, 546, 76, 944, 223, 3947, 164
9, 944, 528, 493, 3910, 202, 48, 76, 68, 322, 351, 732, 172, 119, 1004, 986, 194,
3089, 90, 335, 7841, 10658, 4430]

Second Intance Text Sequence Length: 84
```

In [ ]:
```
sequence_len = 300
X_train_token = pad_sequences(X_train_token, padding='post', maxlen=sequence_len
X_test_token = pad_sequences(X_test_token, padding='post', maxlen=sequence_len)
```

```
In [ ]: print("First Intance Text Sequence:\n")
        print(X_train_token[0])
        print("\nFirst Intance Text Sequence Length:", len(X_train_token[0]))
```

```
First Intance Text Sequence:

[26797  4580    36     2    97    18 18381     3    16  2160   441   978
     1   419    58    27  5471    97   524    76 18382  1039    97    40
   498  9294   531  7168    53 26798   896    71   248   922   638  5270
  1124   584   414    11    58 41507   374   106   643  1919    29  1950
    97    14   429   617   239 18383    40    97    68    78    90   420
  4068 26797     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0]

First Intance Text Sequence Length: 300
```

```
In [ ]: print("Second Intance Text Sequence:\n")
        print(X_train_token[1])
        print("\nSecond Intance Text Sequence Length:", len(X_train_token[1]))
```

```
Second Intance Text Sequence:

[10658  3058   335  7841     2  1004  3089   652    53   442   652    53
  1004   236   241  3565  1004   986   402 62688     3    16   536     1
    18  3058  1258    65  1330  1331   986  1004  5967   705   326  3089
    68   123  2076   179   544   326  1819   464   271  7842  1252  2217
   271   322   122  1837  1617  4186   775  2317   546    76   944   223
  3947  1649   944   528   493  3910   202    48    76    68   322   351
   732   172   119  1004   986   194  3089    90   335  7841 10658  4430
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0]

Second Intance Text Sequence Length: 300
```

In [ ]:

```
home = os.path.expanduser('~')
glove_embedding_filepath = os.path.join(home, "/kaggle/input/glove-6b-100dim/glo
```

In [ ]:

```
def create_embedding_matrix (filepath, word_index, embedding_dim):
    vocab_size = len(word_index) + 1
    embedding_matrix = np.zeros((vocab_size, embedding_dim))

    with open(filepath) as file:
        for line in file:
            word, *vector = line.split()
            if word in word_index:
                idx = word_index[word]
                embedding_matrix[idx] = np.array(vector, dtype=np.float32)[:embe

    return embedding_matrix
```

In [ ]:

```
embedding_dim = 100
embedding_matrix = create_embedding_matrix(glove_embedding_filepath, tokenizer.w
```

****without glove****

In [ ]:

```
# Without GloVe
model = Sequential()
model.add(layers.Embedding(input_dim=vocab_size, output_dim=embedding_dim, input
model.add(layers.Conv1D(filters=128, kernel_size=5, activation='relu'))
model.add(layers.Bidirectional(layers.GRU(units=200, dropout=0.25)))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(32, activation='relu'))
model.add(layers.Dense(20, activation='softmax'))
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=
model.summary()
```

```
/opt/conda/lib/python3.10/site-packages/keras/src/layers/core/embedding.py:81: Us
erWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When usi
ng Sequential models, prefer using an `Input(shape)` object as the first layer in
the model instead.
  super().__init__(**kwargs)
```

```
Model: "sequential"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━
┃ Layer (type)                     ┃ Output Shape           ┃      Param #
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━
│ embedding (Embedding)            │ (None, 300, 100)       │   15,064,100
├──────────────────────────────────┼────────────────────────┼────────────
│ conv1d (Conv1D)                  │ (None, 296, 128)       │       64,128
├──────────────────────────────────┼────────────────────────┼────────────
│ bidirectional (Bidirectional)    │ (None, 400)            │      396,000
├──────────────────────────────────┼────────────────────────┼────────────
│ dense (Dense)                    │ (None, 64)             │       25,664
├──────────────────────────────────┼────────────────────────┼────────────
│ dense_1 (Dense)                  │ (None, 32)             │        2,080
├──────────────────────────────────┼────────────────────────┼────────────
│ dense_2 (Dense)                  │ (None, 20)             │          660
└──────────────────────────────────┴────────────────────────┴────────────

 Total params: 15,552,632 (59.33 MB)

 Trainable params: 15,552,632 (59.33 MB)

 Non-trainable params: 0 (0.00 B)
```

In [ ]:

```
history = model.fit(X_train_token, y_train, epochs=20, validation_data=(X_test_t
Epoch 1/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 93s 944ms/step - accuracy: 0.0674 - loss: 2.9376 - val
_accuracy: 0.2981 - val_loss: 2.1863
Epoch 2/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 82s 924ms/step - accuracy: 0.5156 - loss: 1.4462 - val
_accuracy: 0.5847 - val_loss: 1.4243
Epoch 3/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 82s 918ms/step - accuracy: 0.8475 - loss: 0.4974 - val
_accuracy: 0.6379 - val_loss: 1.4856
Epoch 4/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 80s 903ms/step - accuracy: 0.9497 - loss: 0.1859 - val
_accuracy: 0.6601 - val_loss: 1.6982
Epoch 5/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 81s 907ms/step - accuracy: 0.9809 - loss: 0.0806 - val
_accuracy: 0.6592 - val_loss: 1.8678
Epoch 6/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 80s 903ms/step - accuracy: 0.9869 - loss: 0.0495 - val
_accuracy: 0.6599 - val_loss: 2.0145
Epoch 7/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 80s 903ms/step - accuracy: 0.9941 - loss: 0.0272 - val
_accuracy: 0.6644 - val_loss: 2.1287
Epoch 8/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 81s 912ms/step - accuracy: 0.9938 - loss: 0.0246 - val
_accuracy: 0.6742 - val_loss: 2.1559
Epoch 9/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 81s 907ms/step - accuracy: 0.9948 - loss: 0.0187 - val
_accuracy: 0.6711 - val_loss: 2.1654
Epoch 10/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 82s 916ms/step - accuracy: 0.9932 - loss: 0.0293 - val
_accuracy: 0.6751 - val_loss: 2.2241
Epoch 11/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 81s 914ms/step - accuracy: 0.9957 - loss: 0.0189 - val
_accuracy: 0.6831 - val_loss: 2.2172
Epoch 12/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 81s 912ms/step - accuracy: 0.9962 - loss: 0.0117 - val
_accuracy: 0.6770 - val_loss: 2.3030
Epoch 13/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 81s 900ms/step - accuracy: 0.9975 - loss: 0.0120 - val
_accuracy: 0.6727 - val_loss: 2.3688
Epoch 14/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 80s 901ms/step - accuracy: 0.9971 - loss: 0.0096 - val
_accuracy: 0.6855 - val_loss: 2.2985
Epoch 15/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 83s 911ms/step - accuracy: 0.9978 - loss: 0.0084 - val
_accuracy: 0.6722 - val_loss: 2.4863
Epoch 16/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 81s 917ms/step - accuracy: 0.9963 - loss: 0.0150 - val
_accuracy: 0.6668 - val_loss: 2.5132
Epoch 17/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 82s 918ms/step - accuracy: 0.9926 - loss: 0.0225 - val
_accuracy: 0.6810 - val_loss: 2.3619
Epoch 18/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 81s 917ms/step - accuracy: 0.9947 - loss: 0.0199 - val
_accuracy: 0.6715 - val_loss: 2.4953
Epoch 19/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 82s 913ms/step - accuracy: 0.9948 - loss: 0.0183 - val
_accuracy: 0.6634 - val_loss: 2.5313
Epoch 20/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 81s 910ms/step - accuracy: 0.9930 - loss: 0.0251 - val
_accuracy: 0.6796 - val_loss: 2.4018
```

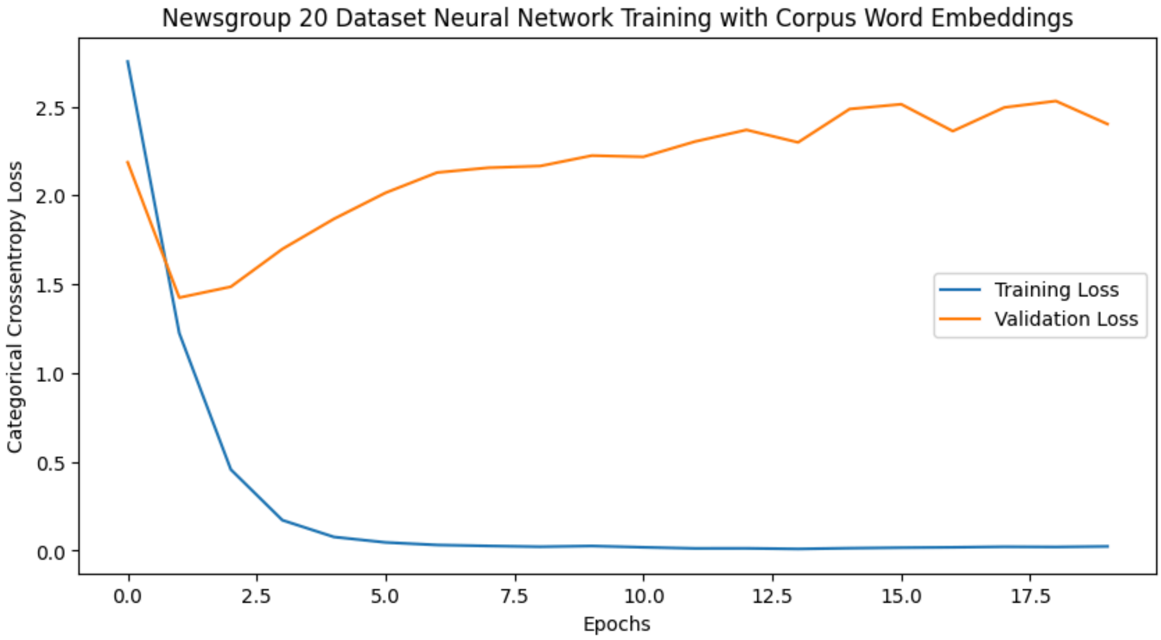
In [ ]:

```
metrics_df = pd.DataFrame(history.history)
print(metrics_df)
```

```
    accuracy      loss  val_accuracy  val_loss
0   0.111985  2.753657      0.298062  2.186340
1   0.600583  1.224303      0.584705  1.424273
2   0.859555  0.456509      0.637945  1.485644
3   0.952713  0.171299      0.660117  1.698189
4   0.981351  0.076783      0.659187  1.867761
5   0.987979  0.046166      0.659851  2.014496
6   0.992664  0.031757      0.664365  2.128699
7   0.994166  0.026250      0.674190  2.155907
8   0.994255  0.021895      0.671136  2.165367
9   0.993636  0.026032      0.675120  2.224087
10  0.995581  0.018839      0.683086  2.217171
11  0.996199  0.012513      0.676978  2.303032
12  0.997437  0.012774      0.672730  2.368847
13  0.997348  0.009473      0.685475  2.298546
14  0.996906  0.013680      0.672199  2.486268
15  0.996111  0.016712      0.666755  2.513220
16  0.994343  0.018662      0.680961  2.361894
17  0.994078  0.021794      0.671535  2.495254
18  0.993901  0.020666      0.663436  2.531320
19  0.994078  0.023708      0.679634  2.401756
```

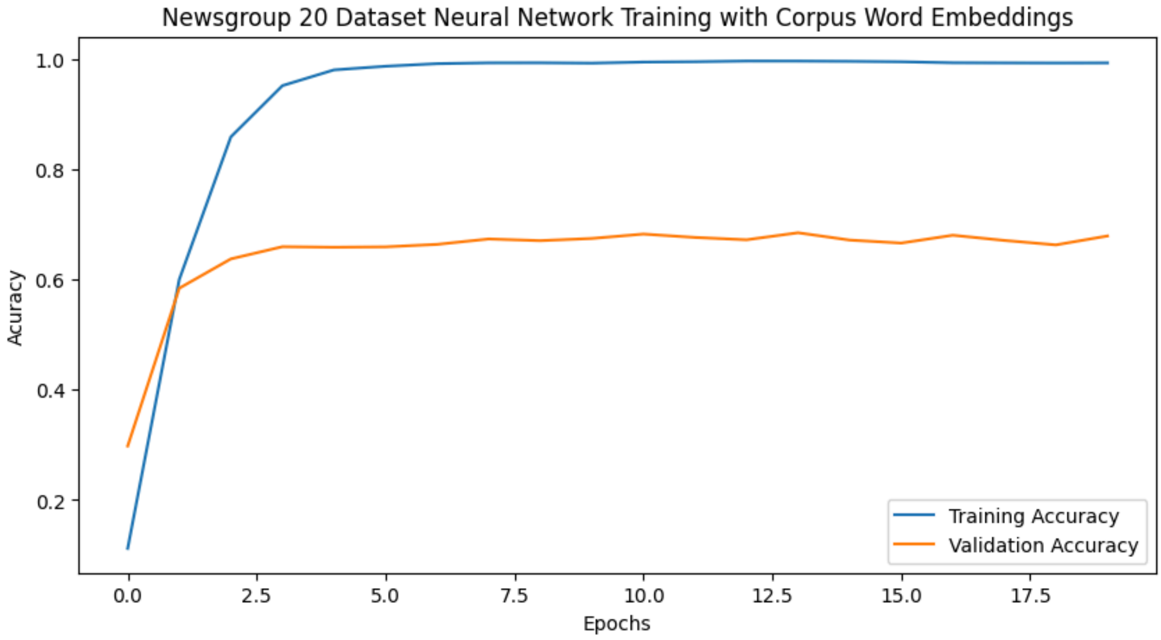
In [ ]:

```
plt.figure(figsize=(10,5))
plt.plot(metrics_df.index, metrics_df.loss)
plt.plot(metrics_df.index, metrics_df.val_loss)
plt.title('Newsgroup 20 Dataset Neural Network Training with Corpus Word Embeddi
plt.xlabel('Epochs')
plt.ylabel('Categorical Crossentropy Loss')
plt.legend(['Training Loss', 'Validation Loss'])
plt.show()
```



In [ ]:

```
plt.figure(figsize=(10,5))
plt.plot(metrics_df.index, metrics_df.accuracy)
plt.plot(metrics_df.index, metrics_df.val_accuracy)
plt.title('Newsgroup 20 Dataset Neural Network Training with Corpus Word Embeddi
plt.xlabel('Epochs')
plt.ylabel('Acuracy')
plt.legend(['Training Accuracy', 'Validation Accuracy'])
plt.show()
```



With Glove

In [ ]:

```
# With GloVe
model = Sequential()
model.add(layers.Embedding(input_dim=vocab_size, output_dim=embedding_dim, input
model.add(layers.Conv1D(filters=128, kernel_size=5, activation='relu'))
model.add(layers.Bidirectional(layers.GRU(units=200, dropout=0.25)))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(32, activation='relu'))
model.add(layers.Dense(20, activation='softmax'))
model.layers[0].set_weights([embedding_matrix])
model.layers[0].trainable = True
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=
model.summary()
```

```
/opt/conda/lib/python3.10/site-packages/keras/src/layers/core/embedding.py:81: Us
erWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When usi
ng Sequential models, prefer using an `Input(shape)` object as the first layer in
the model instead.
  super().__init__(**kwargs)
```

Model: "sequential_1"

```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━
┃ Layer (type)                    ┃ Output Shape           ┃       Param #
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━
│ embedding_1 (Embedding)         │ (None, 300, 100)       │    15,064,100
├─────────────────────────────────┼────────────────────────┼──────────────
│ conv1d_1 (Conv1D)               │ (None, 296, 128)       │        64,128
├─────────────────────────────────┼────────────────────────┼──────────────
│ bidirectional_1 (Bidirectional) │ (None, 400)            │       396,000
├─────────────────────────────────┼────────────────────────┼──────────────
│ dense_3 (Dense)                 │ (None, 64)             │        25,664
├─────────────────────────────────┼────────────────────────┼──────────────
│ dense_4 (Dense)                 │ (None, 32)             │         2,080
├─────────────────────────────────┼────────────────────────┼──────────────
│ dense_5 (Dense)                 │ (None, 20)             │           660
└─────────────────────────────────┴────────────────────────┴──────────────
```

```
Total params: 15,552,632 (59.33 MB)
Trainable params: 15,552,632 (59.33 MB)
Non-trainable params: 0 (0.00 B)
```

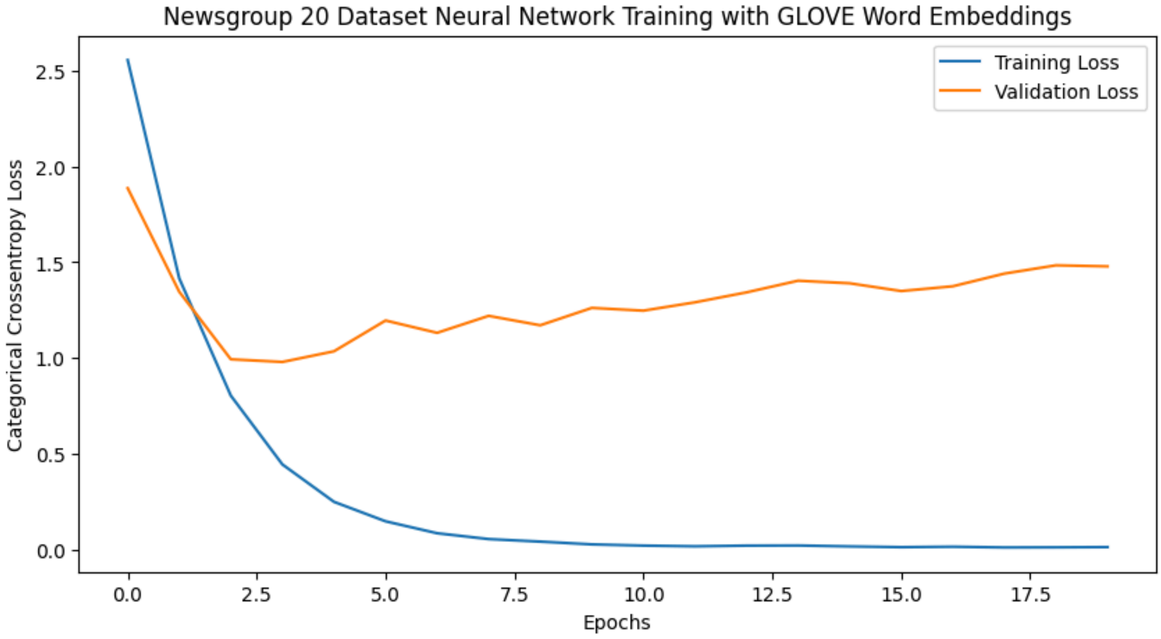
In [ ]:

```python
history1 = model.fit(X_train_token, y_train, epochs=20, validation_data=(X_test_
Epoch 1/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 85s 908ms/step - accuracy: 0.1082 - loss: 2.8429 - val
_accuracy: 0.3518 - val_loss: 1.8862
Epoch 2/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 81s 907ms/step - accuracy: 0.4596 - loss: 1.5814 - val
_accuracy: 0.5319 - val_loss: 1.3456
Epoch 3/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 82s 907ms/step - accuracy: 0.6893 - loss: 0.8820 - val
_accuracy: 0.6762 - val_loss: 0.9932
Epoch 4/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 80s 900ms/step - accuracy: 0.8466 - loss: 0.4594 - val
_accuracy: 0.6907 - val_loss: 0.9793
Epoch 5/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 82s 903ms/step - accuracy: 0.9162 - loss: 0.2638 - val
_accuracy: 0.7227 - val_loss: 1.0349
Epoch 6/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 80s 903ms/step - accuracy: 0.9524 - loss: 0.1523 - val
_accuracy: 0.7005 - val_loss: 1.1957
Epoch 7/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 80s 901ms/step - accuracy: 0.9732 - loss: 0.0926 - val
_accuracy: 0.7415 - val_loss: 1.1313
Epoch 8/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 82s 908ms/step - accuracy: 0.9825 - loss: 0.0577 - val
_accuracy: 0.7379 - val_loss: 1.2200
Epoch 9/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 80s 903ms/step - accuracy: 0.9868 - loss: 0.0422 - val
_accuracy: 0.7572 - val_loss: 1.1707
Epoch 10/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 80s 904ms/step - accuracy: 0.9918 - loss: 0.0296 - val
_accuracy: 0.7541 - val_loss: 1.2614
Epoch 11/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 81s 909ms/step - accuracy: 0.9950 - loss: 0.0202 - val
_accuracy: 0.7604 - val_loss: 1.2471
Epoch 12/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 80s 905ms/step - accuracy: 0.9954 - loss: 0.0148 - val
_accuracy: 0.7527 - val_loss: 1.2905
Epoch 13/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 81s 906ms/step - accuracy: 0.9945 - loss: 0.0209 - val
_accuracy: 0.7515 - val_loss: 1.3425
Epoch 14/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 81s 908ms/step - accuracy: 0.9946 - loss: 0.0178 - val
_accuracy: 0.7491 - val_loss: 1.4034
Epoch 15/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 80s 894ms/step - accuracy: 0.9971 - loss: 0.0128 - val
_accuracy: 0.7515 - val_loss: 1.3900
Epoch 16/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 83s 904ms/step - accuracy: 0.9960 - loss: 0.0122 - val
_accuracy: 0.7614 - val_loss: 1.3494
Epoch 17/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 80s 895ms/step - accuracy: 0.9960 - loss: 0.0138 - val
_accuracy: 0.7621 - val_loss: 1.3743
Epoch 18/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 82s 899ms/step - accuracy: 0.9970 - loss: 0.0102 - val
_accuracy: 0.7596 - val_loss: 1.4405
Epoch 19/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 81s 906ms/step - accuracy: 0.9963 - loss: 0.0110 - val
_accuracy: 0.7491 - val_loss: 1.4837
Epoch 20/20
89/89 ━━━━━━━━━━━━━━━━━━━━ 83s 914ms/step - accuracy: 0.9965 - loss: 0.0111 - val
_accuracy: 0.7613 - val_loss: 1.4782
```

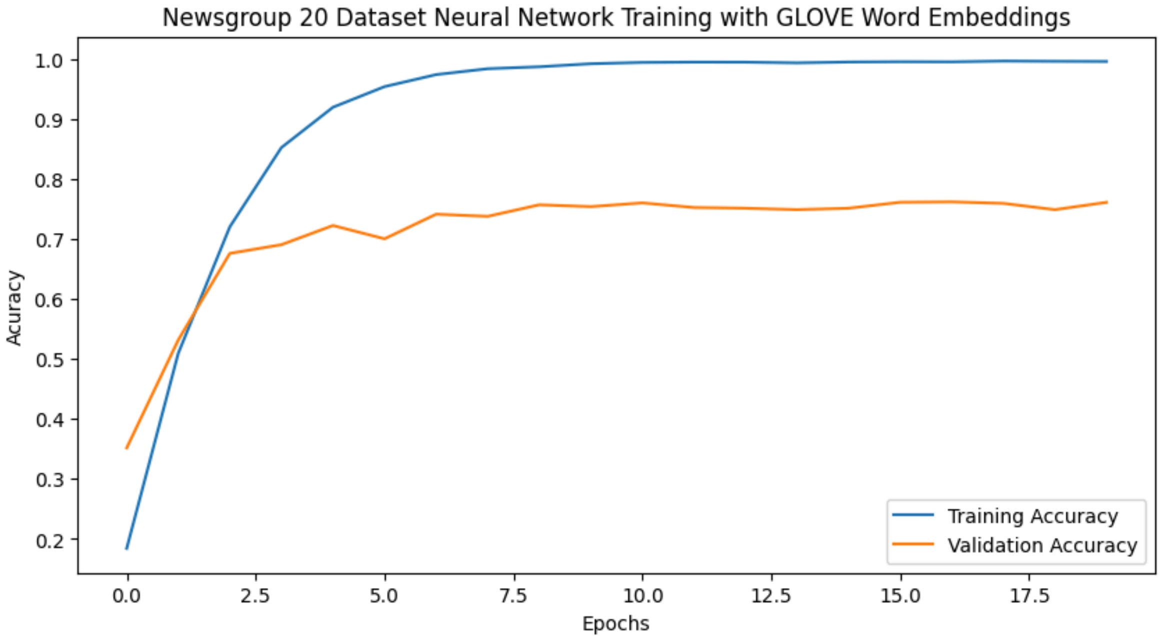
In [ ]:
```
metrics_df = pd.DataFrame(history1.history)
print(metrics_df)
```

```
    accuracy      loss  val_accuracy  val_loss
0   0.184197  2.555341      0.351832  1.886198
1   0.509546  1.415424      0.531864  1.345614
2   0.720435  0.803438      0.676182  0.993201
3   0.852572  0.444362      0.690653  0.979309
4   0.919834  0.249195      0.722650  1.034904
5   0.954216  0.147528      0.700478  1.195685
6   0.974280  0.085182      0.741503  1.131335
7   0.984091  0.054707      0.737918  1.219992
8   0.987361  0.041460      0.757169  1.170735
9   0.992399  0.027078      0.754116  1.261413
10  0.994608  0.020570      0.760356  1.247086
11  0.995139  0.016896      0.752655  1.290538
12  0.995050  0.020433      0.751460  1.342521
13  0.993901  0.021138      0.749071  1.403446
14  0.995316  0.016418      0.751460  1.390003
15  0.995757  0.012513      0.761418  1.349442
16  0.995581  0.014781      0.762082  1.374285
17  0.996906  0.011067      0.759559  1.440473
18  0.996465  0.011665      0.749071  1.483724
19  0.996199  0.012861      0.761285  1.478225
```

In [ ]:
```
plt.figure(figsize=(10,5))
plt.plot(metrics_df.index, metrics_df.loss)
plt.plot(metrics_df.index, metrics_df.val_loss)
plt.title('Newsgroup 20 Dataset Neural Network Training with GLOVE Word Embeddin
plt.xlabel('Epochs')
plt.ylabel('Categorical Crossentropy Loss')
plt.legend(['Training Loss', 'Validation Loss'])
plt.show()
```



In [ ]:
```
plt.figure(figsize=(10,5))
plt.plot(metrics_df.index, metrics_df.accuracy)
plt.plot(metrics_df.index, metrics_df.val_accuracy)
plt.title('Newsgroup 20 Dataset Neural Network Training with GLOVE Word Embeddin
plt.xlabel('Epochs')
plt.ylabel('Acuracy')
plt.legend(['Training Accuracy', 'Validation Accuracy'])
plt.show()
```


Newsgroup 20 Dataset Neural Network Training with GLOVE Word Embeddings
Acuracy
Epochs
Training Accuracy
Validation Accuracy