:[107] In

```
import pandas as pd
import numpy as np
import nltk
nltk.download('stopwords')
nltk.download('punkt')
```

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

Out[107]:

```
True
```

:[108] In

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
import os
import string
import copy
import pickle
```

:[109] In

```
title = "20_newsgroups"
os.chdir("C:/20_newsgroups")
paths= []
for (dirpath, dirnames, filenames) in os.walk(str(os.getcwd())+'/'+title+'/'):
    for i in filenames:
        paths.append(str(dirpath)+str("/")+i)
print(dirpath)
paths [0]
```

```
C:\20_newsgroups/20_newsgroups/20_newsgroups\alt.atheism
```

Out[109]:

```
'C:\\20_newsgroups/20_newsgroups/20_newsgroups\\alt.atheism/49960'
```

:[110] In

```
def remove_stop_words(data):
    stop_words = stopwords.words('english')
    words = word_tokenize(str(data))
    new_text = ""
    for w in words:
        if w not in stop_words:
            new_text = new_text + " " + w
    return np.char.strip(new_text)


def remove_punctuation(data):
    symbols = "!\"#$%&()*+-./:;<=>?@[\]^_`{|}~\n"
    for i in range(len(symbols)):
        data = np.char.replace(data, symbols[i], ' ')
        data = np.char.replace(data, "  ", " ")
    data = np.char.replace(data, ',', '')
    return data


def convert_lower_case(data):
    return np.char.lower(data)


def stemming(data):
    stemmer= PorterStemmer()

    tokens = word_tokenize(str(data))
    new_text = ""
    for w in tokens:
        new_text = new_text + " " + stemmer.stem(w)
    return np.char.strip(new_text)


def convert_numbers(data):
    data = np.char.replace(data, "0", " zero ")
    data = np.char.replace(data, "1", " one ")
    data = np.char.replace(data, "2", " two ")
    data = np.char.replace(data, "3", " three ")
    data = np.char.replace(data, "4", " four ")
    data = np.char.replace(data, "5", " five ")
    data = np.char.replace(data, "6", " six ")
    data = np.char.replace(data, "7", " seven ")
    data = np.char.replace(data, "8", " eight ")
    data = np.char.replace(data, "9", " nine ")
    return data


def remove_header(data):
    try:
        ind = data.index('\n\n')
        data = data[ind:]
    except:
        print("No Header")
    return data


def remove_apostrophe(data):
    return np.char.replace(data, "'", "")
```

```
def remove_single_characters(data):
    words = word_tokenize(str(data))
    new_text = ""
    for w in words:
        if len(w) > 1:
            new_text = new_text + " " + w
    return np.char.strip(new_text)
```

:[111] In

```
def preprocess(data, query):
    if not query:
        data = remove_header(data)
        data = convert_lower_case(data)
        data = convert_numbers(data)
        data = remove_punctuation(data)
        data = remove_stop_words(data)
        data = remove_apostrophe(data)
        data = remove_single_characters(data)
        data = stemming(data)
    return data
```

:[112] In

```
doc = 0
postings = pd.DataFrame()

for path in paths:
    file = open(path, 'r', encoding='cp1250')
    text = file.read().strip()
    file.close()
    preprocessed_text = preprocess(text, False)


    if doc%100 == 0:
        print(doc)
    tokens = word_tokenize(str(preprocessed_text))
    for token in tokens:
        if token in postings:
            p = postings[token][0]
            p.add(doc)
            postings[token][0] = p
        else:
            postings.insert(value=[{doc}], loc=0, column=token)
    doc += 1
```

```
0
```

:[113] In

```
postings["exam"][0]
```

Out[113]:

```
{21}
```

:[114] In

```
postings.to_pickle(title + "_unigram_postings")
```

:[115] In

```
postings = pd.read_pickle(title + "_unigram_postings")
```

:[ ] In

```

```