

AIWIR Lab Week 3

Team-8

Team members :

1. A Spoorthi Alva PES2UG19CS001
2. A R Manyatha PES2UG19CS002
3. Achyut Jagini. PES2UG19CS013
4. Amulya S Dinesh PES2UG19CS035

Dataset: <https://www.kaggle.com/moupriyaroy/20-newsgroups>

Tasks :

- Removing stop words
- Removing punctuation
- Convert to lowercase
- Stemming
- Converting number its equivalent words
- Removing header

Tool used : Pycharm

- It is installed on the computer.
- The dataset is downloaded and added to the folder where the main program exists
- Project is created in Pycharm by opening a folder containing the dataset.
- Required modules are installed.

Libraries used :

- NLTK
- Pandas
- NumPy
- Pickle

Steps to build a Positional Index

- Fetch the document.
- Remove stop words, stem the resulting words.
- If the word is already present in the dictionary, add the document and the corresponding positions it appears in. Else, create a new entry.

→ Also update the frequency of the word for each document, as well as the no. of documents it appears in.

Code :

importing libraries

import numpy as np

import os

import nltk

from nltk.stem import PorterStemmer

from nltk.tokenize import TweetTokenizer

from natsort import natsorted

import string

def read_file(filename):

 with open(filename, 'r', encoding="ascii", errors="surrogateescape") as f:

 stuff = f.read()

 f.close()

Remove header and footer.

stuff = remove_header_footer(stuff)

return stuff

def remove_header_footer(final_string):

 new_final_string = ""

```
tokens = final_string.split('\n\n')
```

```
# Remove tokens[0] and tokens[-1]
```

```
for token in tokens[1:-1]:
```

```
    new_final_string += token + " "
```

```
return new_final_string
```

```
def preprocessing(final_string):
```

```
# Tokenize.
```

```
tokenizer = TweetTokenizer()
```

```
token_list = tokenizer.tokenize(final_string)
```

```
# Remove punctuations.
```

```
table = str.maketrans(", ", '\t')
```

```
token_list = [word.translate(table) for word in token_list]
```

```
punctuations = (string.punctuation).replace("'", "")
```

```
trans_table = str.maketrans(", ", punctuations)
```

```
stripped_words = [word.translate(trans_table) for word in token_list]
```

```
token_list = [str for str in stripped_words if str]
```

```
# Change to lowercase.
```

```
token_list = [word.lower() for word in token_list]
```

```
return token_list
```

```
# In this example, we create the positional index for only 1 folder.
```

```
folder_names = ["comp.graphics"]
```

Initialize the stemmer.

```
stemmer = PorterStemmer()
```

Initialize the file no.

```
fileno = 0
```

```
pos_index = {}
```

Initialize the file mapping (fileno -> file name).

```
file_map = {}
```

```
for folder_name in folder_names:
```

```
    file_names = natsorted(os.listdir("20_newsgroups/" + folder_name))
```

```
    for file_name in file_names:
```

```
        stuff = read_file("20_newsgroups/" + folder_name + "/" + file_name)
```

```
        final_token_list = preprocessing(stuff)
```

```
        for pos, term in enumerate(final_token_list):
```

```
            term = stemmer.stem(term)
```

If the term already exists in the positional index dictionary.

```
if term in pos_index:
```

```
    n='0123456789'
```

```
    pos_index[term][0] = pos_index[term][0] + 1
```

Check if the term has existed in that DocID before.

```
if fileno in pos_index[term][1]:  
    pos_index[term][1][fileno].append(pos)
```

```
else:  
    pos_index[term][1][fileno] = [pos]
```

If term does not exist in the positional index dictionary (first encounter).

```
else:  
    pos_index[term] = [].  
    pos_index[term].append(1)  
    pos_index[term].append({})  
  
# Add doc ID to postings list.  
    pos_index[term][1][fileno] = [pos]
```

Map the file no. to the file name.

```
file_map[fileno] = "20_newsgroups/" + folder_name + "/" + file_name
```

Increment the file no. counter for document ID mapping

```
fileno += 1
```

Sample positional index to test the code.

```
sample_pos_idx = pos_index["andrew"]  
print("Positional Index")  
print(sample_pos_idx)
```

```

file_list = sample_pos_idx[1]
print("Filename, [Positions]")
for fileno, positions in file_list.items():
    print(file_map[fileno], positions)

```

#Converting numbers to words

```

def convert_to_words(num):
    # Get number of digits in given number

    l = len(num)

    # Base cases
    if (l == 0):
        print("empty string")
        return

    if (l > 4):
        print("Length more than 4 is not supported")
        return

    single_digits = ["zero", "one", "two", "three", "four", "five", "six", "seven", "eight",
"nine"]

    two_digits = ["", "ten", "eleven", "twelve", "thirteen", "fourteen",
"fifteen", "sixteen", "seventeen", "eighteen", "nineteen"]

    tens_multiple = ["", "", "twenty", "thirty", "forty", "fifty", "sixty", "seventy",
"eighty", "ninety"]

    tens_power = ["hundred", "thousand"]

    print(num, ":", end=" ")

    if (l == 1):
        print(single_digits[ord(num[0]) - 48])

```

```
return
```

```
x = 0
```

```
while (x < len(num)):
```

```
    if (l >= 3):
```

```
        if (ord(num[x]) - 48 != 0):
```

```
            print(single_digits[ord(num[x]) - 48],
```

```
                  end=" ")
```

```
            print(tens_power[l - 3], end=" ")
```

```
        l -= 1
```

```
    else:
```

```
        if (ord(num[x]) - 48 == 1):
```

```
            sum = (ord(num[x]) - 48 +
```

```
                  ord(num[x+1]) - 48)
```

```
            print(two_digits[sum])
```

```
            return
```

```
        elif (ord(num[x]) - 48 == 2 and
```

```
              ord(num[x + 1]) - 48 == 0):
```

```
            print("twenty")
```

```
            return
```

```
        else:
```

```
            i = ord(num[x]) - 48
```

```
            if(i > 0):
```

```
                print(tens_multiple[i], end=" ")
```

```
            else:
```

```
        print("", end="")
    x += 1
    if(ord(num[x]) - 48 != 0):
        print(single_digits[ord(num[x]) - 48])
    x += 1
```

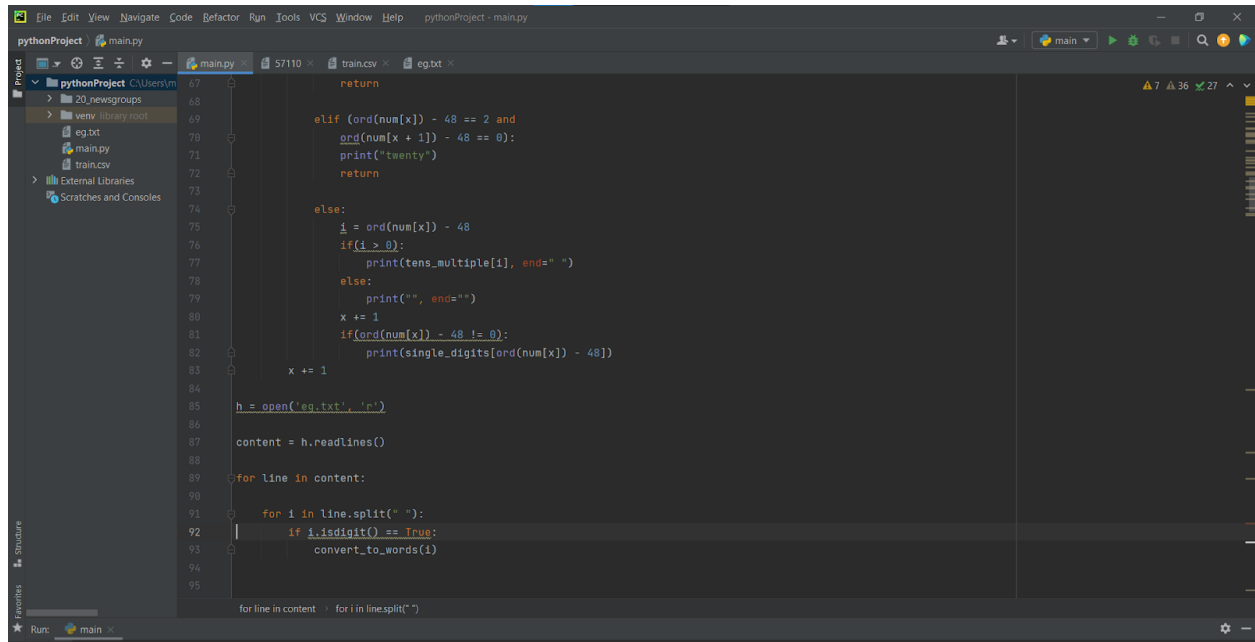
```
h = open('eg.txt', 'r')
content = h.readlines()
for line in content:
    for i in line.split(" "):
        if i.isdigit() == True:
            convert_to_words(i)
```


This screenshot shows the first part of a Python script in an IDE. The script defines two lists: `single_digits` and `two_digits`, which map numerical values to their word representations. It also defines `tens_multiple` and `tens_power` for tens and hundreds. The `print(num, ":", end=" ")` statement is followed by a conditional check `if (l == 1):` that prints the first digit's word and returns. A `while` loop is initiated with `x = 0` and `while (x < len(num)):`, with a nested `if (l >= 3):` condition.

```
31 two_digits = ["", "ten", "eleven", "twelve",
32             "thirteen", "fourteen", "fifteen",
33             "sixteen", "seventeen", "eighteen",
34             "nineteen"]
35
36
37 tens_multiple = ["", "", "twenty", "thirty", "forty",
38                 "fifty", "sixty", "seventy", "eighty",
39                 "ninety"]
40
41 tens_power = ["hundred", "thousand"]
42
43 print(num, ":", end=" ")
44
45
46 if (l == 1):
47     print(single_digits[ord(num[0]) - 48])
48     return
49
50 x = 0
51 while (x < len(num)):
52
53     if (l >= 3):
54         for line in content:
55             for i in linesplitl(" ")
```

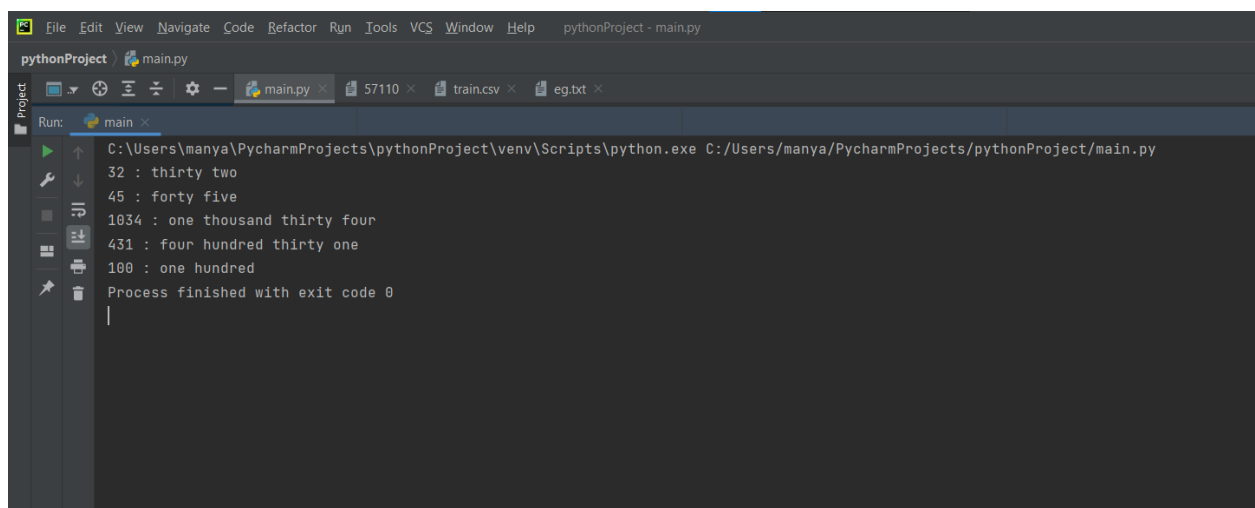
This screenshot shows the continuation of the Python script. It completes the `if (l >= 3):` block by printing the word for the current power of ten and decrementing `l`. The `else:` block handles the remaining digits. It includes a check for `ord(num[x]) - 48 == 1` to handle the special case of 'one' followed by another digit, which would otherwise be incorrectly interpreted as a tens value. It also includes a check for `ord(num[x]) - 48 == 2` to handle 'two' followed by another digit. The script concludes by printing the word for the current tens value.

```
54         if (ord(num[x]) - 48 != 0):
55             print(single_digits[ord(num[x]) - 48],
56                   end=" ")
57             print(tens_power[l - 3], end=" ")
58         l -= 1
59     else:
60
61         if (ord(num[x]) - 48 == 1):
62             sum = (ord(num[x]) - 48 +
63                   ord(num[x+1]) - 48)
64             print(two_digits[sum])
65             return
66
67         elif (ord(num[x]) - 48 == 2 and
68               ord(num[x + 1]) - 48 == 0):
69             print('twenty')
70             return
71
72         else:
73             i = ord(num[x]) - 48
74             if (i > 0):
75                 print(tens_multiple[i], end=" ")
76
77     for line in content:
78         for i in linesplitl(" ")
```



```
67     return
68
69     elif (ord(num[x]) - 48 == 2 and
70           ord(num[x + 1]) - 48 == 0):
71         print("twenty")
72         return
73
74     else:
75         i = ord(num[x]) - 48
76         if(i > 0):
77             print(tens_multiple[i], end=" ")
78         else:
79             print("", end="")
80         x += 1
81         if(ord(num[x]) - 48 != 0):
82             print(single_digits[ord(num[x]) - 48])
83     x += 1
84
85     h = open('eg.txt', 'r')
86
87     content = h.readlines()
88
89     for line in content:
90
91         for i in line.split(" "):
92             if i.isdigit() == True:
93                 convert_to_words(i)
94
95     for line in content:
96         for i in line.split(" ")
```

Output:



```
C:\Users\manya\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/manya/PycharmProjects/pythonProject/main.py
32 : thirty two
45 : forty five
1034 : one thousand thirty four
431 : four hundred thirty one
100 : one hundred
Process finished with exit code 0
```

The screenshot shows an IDE window titled 'pythonProject - main.py'. The left sidebar displays a project tree with folders like '20_newsgroups' and 'venv', and files like 'eg.txt', 'main.py', and 'train.csv'. The main editor area shows the following Python code:

```
1 # importing libraries
2 import numpy as np
3 import os
4 import nltk
5 from nltk.stem import PorterStemmer
6 from nltk.tokenize import TweetTokenizer
7 from natsort import natsorted
8 import string
9
10 def read_file(filename):
11     with open(filename, 'r', encoding="ascii", errors="surrogateescape") as f:
12         stuff = f.read()
13
14     f.close()
15     # Remove header and footer.
16     stuff = remove_header_footer(stuff)
17
18     return stuff
19
20 def remove_header_footer(final_string):
21     new_final_string = ""
22     tokens = final_string.split('\n\n')
23
24     # Remove tokens[0] and tokens[-1]
25     for token in tokens[1:-1]:
26         new_final_string += token + " "
27     return new_final_string
28
29 def preprocessing(final_string):
```

The bottom status bar indicates 'Python 3.10 (pythonProject)' and '45:1 CRLF UTF-8 Tab'.

The screenshot shows the same IDE window, now displaying the continuation of the Python script:

```
28
29 def preprocessing(final_string):
30     # Tokenize.
31     tokenizer = TweetTokenizer()
32     token_list = tokenizer.tokenize(final_string)
33     # Remove punctuations.
34     table = str.maketrans('', '', string.punctuation)
35     token_list = [word.translate(table) for word in token_list]
36     punctuations = (string.punctuation).replace("'", "")
37     trans_table = str.maketrans('', '', punctuations)
38     stripped_words = [word.translate(trans_table) for word in token_list]
39     token_list = [str for str in stripped_words if str]
40
41     # Change to lowercase.
42     token_list = [word.lower() for word in token_list]
43     return token_list
44
45     In this example, we create the positional index for only 1 folder.
46     folder_names = ["comp.graphics"]
47
48     # Initialize the stemmer.
49     stemmer = PorterStemmer()
50
51     # Initialize the file no.
52     fileno = 0
53
54     # Initialize the dictionary.
55     pos_index = {}
56
```

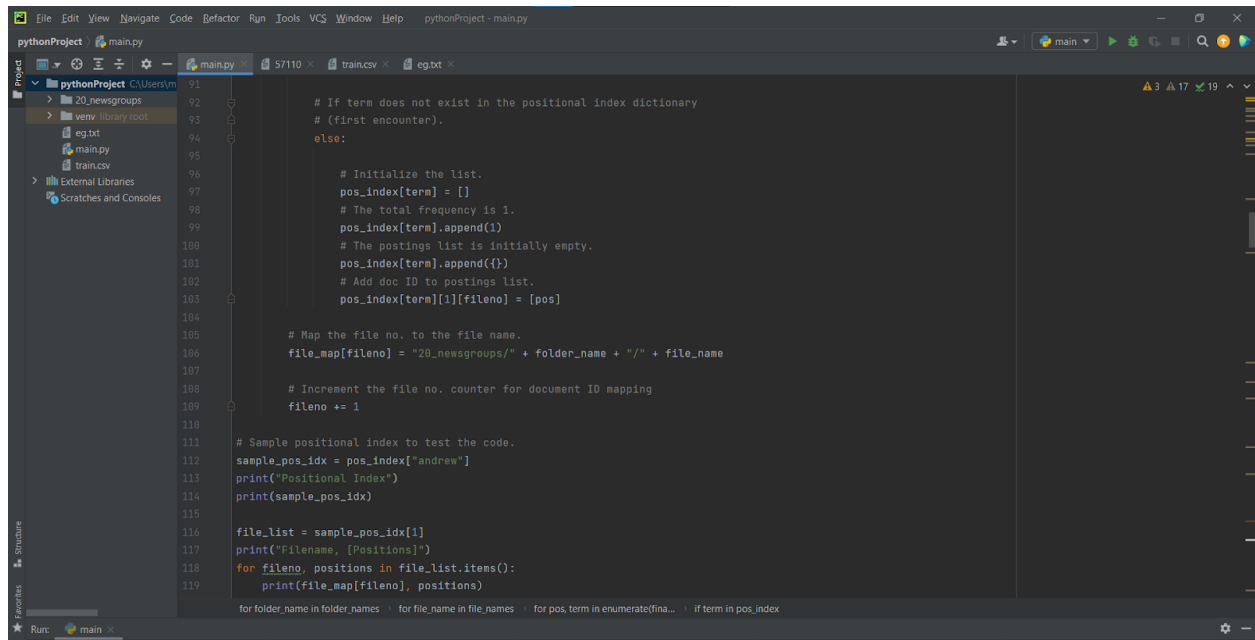
The bottom status bar remains the same, showing 'Python 3.10 (pythonProject)' and '45:1 CRLF UTF-8 Tab'.

```
pythonProject - main.py
pythonProject C:\Users\m...
> 20_newsgroups
> venv\library root
eg.txt
main.py
train.csv
External Libraries
Scratches and Consoles

55 pos_index = {}
56
57 # Initialize the file mapping (fileno -> file name).
58 file_map = {}
59
60 for folder_name in folder_names:
61
62     # Open files.
63     file_names = natsorted(os.listdir("20_newsgroups/" + folder_name))
64
65     # For every file.
66     for file_name in file_names:
67
68         # Read file contents.
69         stuff = read_file("20_newsgroups/" + folder_name + "/" + file_name)
70
71         final_token_list = preprocessing(stuff)
72
73         # For position and term in the tokens.
74         for pos, term in enumerate(final_token_list):
75
76             # First stem the term.
77             term = stemmer.stem(term)
78
79             # If term already exists in the positional index dictionary.
80             if term in pos_index:
81                 n = '0123456789'
82                 # Increment total freq by 1.
83                 pos_index[term][0] = pos_index[term][0] + 1
84
85 for folder_name in folder_names    for file_name in file_names
```

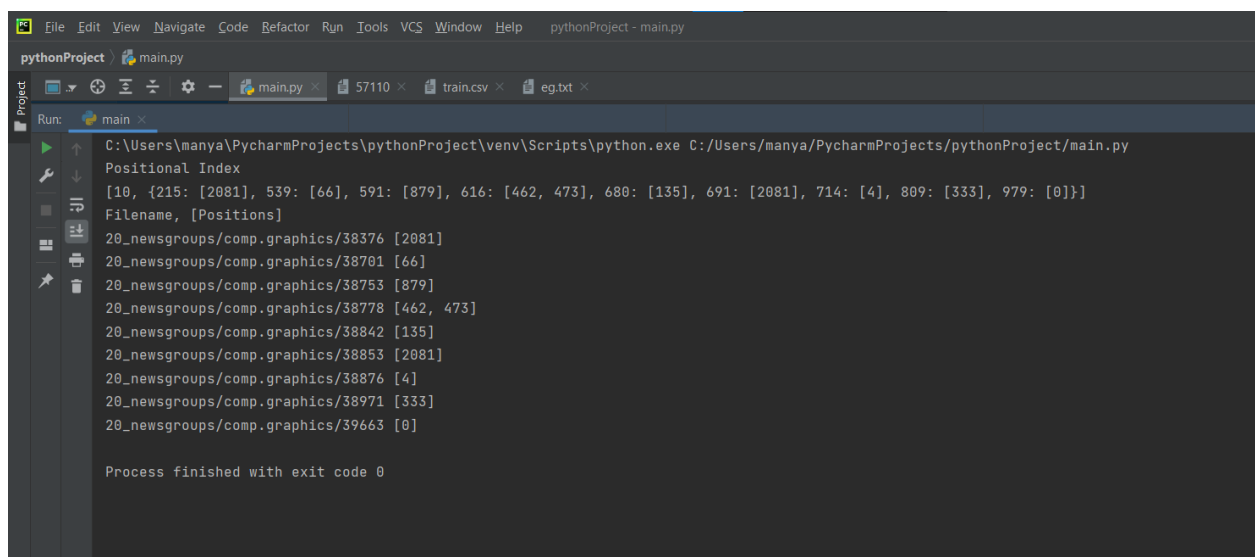
```
pythonProject - main.py
pythonProject C:\Users\m...
> 20_newsgroups
> venv\library root
eg.txt
main.py
train.csv
External Libraries
Scratches and Consoles

79
80
81 # If term already exists in the positional index dictionary.
82 if term in pos_index:
83     n = '0123456789'
84     # Increment total freq by 1.
85     pos_index[term][0] = pos_index[term][0] + 1
86
87     # Check if the term has existed in that DocID before.
88     if fileno in pos_index[term][1]:
89         pos_index[term][1][fileno].append(pos)
90     else:
91         pos_index[term][1][fileno] = [pos]
92
93 # If term does not exist in the positional index dictionary
94 # (first encounter).
95 else:
96     # Initialize the list.
97     pos_index[term] = []
98     # The total frequency is 1.
99     pos_index[term].append(1)
100     # The postings list is initially empty.
101     pos_index[term].append({})
102     # Add doc ID to postings list.
103     pos_index[term][1][fileno] = [pos]
104
105 # Map the file no. to the file name.
106 file_map[fileno] = "20_newsgroups/" + folder_name + "/" + file_name
107
108 for folder_name in folder_names    for file_name in file_names
```



```
91
92
93     # If term does not exist in the positional index dictionary
94     # (first encounter).
95     else:
96
97         # Initialize the list.
98         pos_index[term] = []
99         # The total frequency is 1.
100         pos_index[term].append(1)
101         # The postings list is initially empty.
102         pos_index[term].append({})
103         # Add doc ID to postings list.
104         pos_index[term][1][fileno] = [pos]
105
106     # Map the file no. to the file name.
107     file_map[fileno] = "20_newsgroups/" + folder_name + "/" + file_name
108
109     # Increment the file no. counter for document ID mapping
110     fileno += 1
111
112     # Sample positional index to test the code.
113     sample_pos_idx = pos_index["andrew"]
114     print("Positional Index")
115     print(sample_pos_idx)
116
117     file_list = sample_pos_idx[1]
118     print("Filename, [Positions]")
119     for fileno, positions in file_list.items():
120         print(file_map[fileno], positions)
```

Output :



```
C:\Users\manya\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/manya/PycharmProjects/pythonProject/main.py
Positional Index
[10, {215: [2081], 539: [66], 591: [879], 616: [462, 473], 680: [135], 691: [2081], 714: [4], 809: [333], 979: [0]}]
Filename, [Positions]
20_newsgroups/comp.graphics/38376 [2081]
20_newsgroups/comp.graphics/38701 [66]
20_newsgroups/comp.graphics/38753 [879]
20_newsgroups/comp.graphics/38778 [462, 473]
20_newsgroups/comp.graphics/38842 [135]
20_newsgroups/comp.graphics/38853 [2081]
20_newsgroups/comp.graphics/38876 [4]
20_newsgroups/comp.graphics/38971 [333]
20_newsgroups/comp.graphics/39663 [0]

Process finished with exit code 0
```