



Artificial Intelligence Project: POS Tagging

September 28, 2020

RITIK JAIN 18114068
PRIYANSHU GARG 181140058
YUGANTAR ARYA 18114084

Contents

1	Week 1	i
1.1	Objective	i
1.2	Procedure	i
1.2.1	Phase 1 Filtration	i
1.2.2	Phase 2 Collection	i
1.3	Code Screenshots	ii
1.4	Execution	iv
1.5	Output Files	v

1 Week 1

Date: September 28, 2020

1.1 Objective

To filter the text by extracting word and POS tag for the word

1.2 Procedure

The program does its job in 2 phases The number of phases could have been reduced to 1 but we went with this code since it doesn't take much time to pre-process all the files and it has to be done once; hence we preferred clarity of code over the small extra time it takes

1.2.1 Phase 1 Filtration

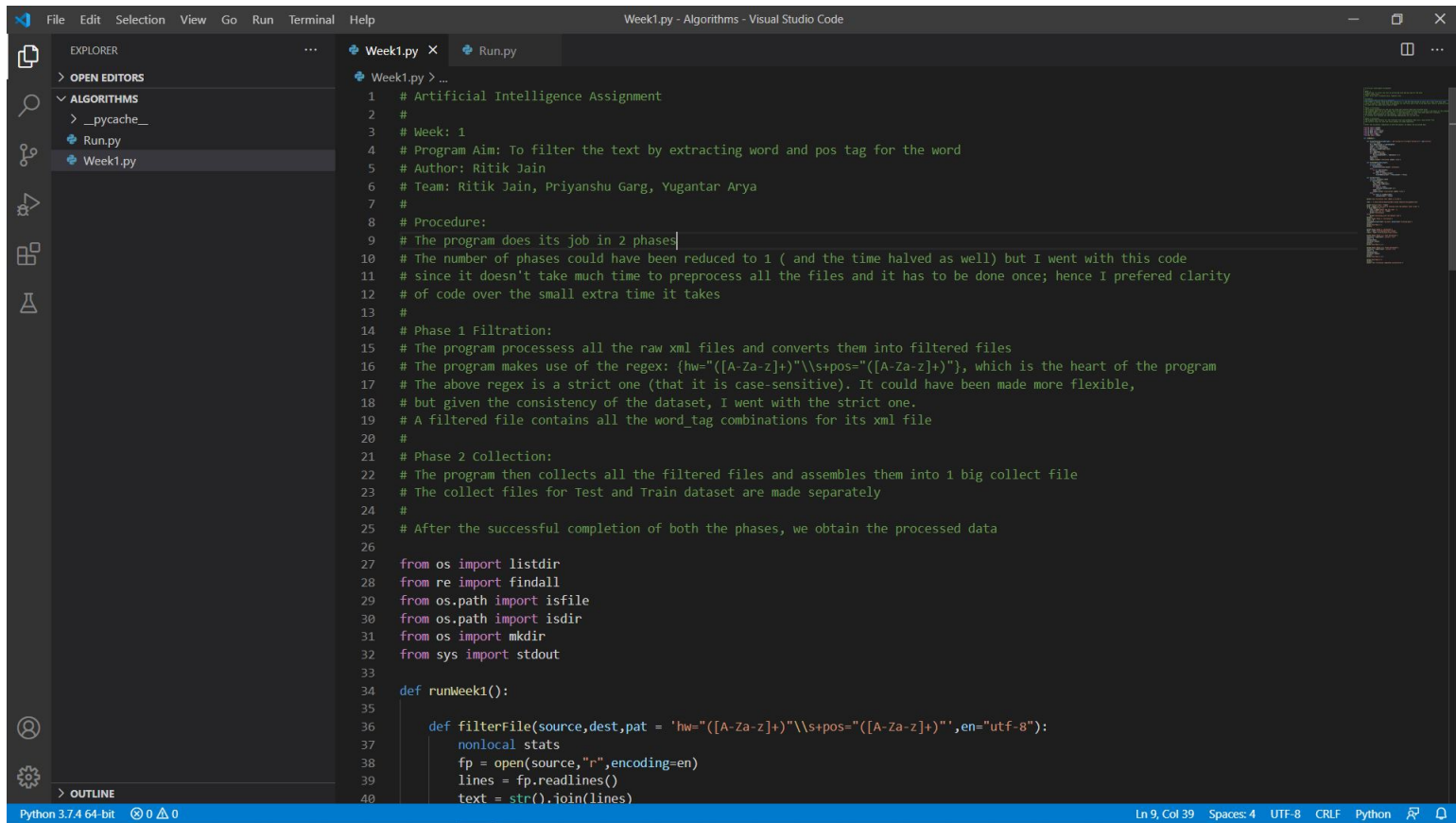
The program processess all the raw xml files and converts them into filtered files The program makes use of the regex: `hw="([A-Za-z]+)"\s+pos="([A-Za-z]+)"`, which is the heart of the program The above regex is a strict one (that it is case-sensitive). It could have been made more flexible, but given the consistency of the dataset, I went with the strict one. A filtered file contains all the word_tag combinations for its xml file

1.2.2 Phase 2 Collection

The program then collects all the filtered files and assembles them into 1 big collect file The collect files for Test and Train dataset are made separately

After the successful completion of both the phases, we obtain the processed data

1.3 Code Screenshots

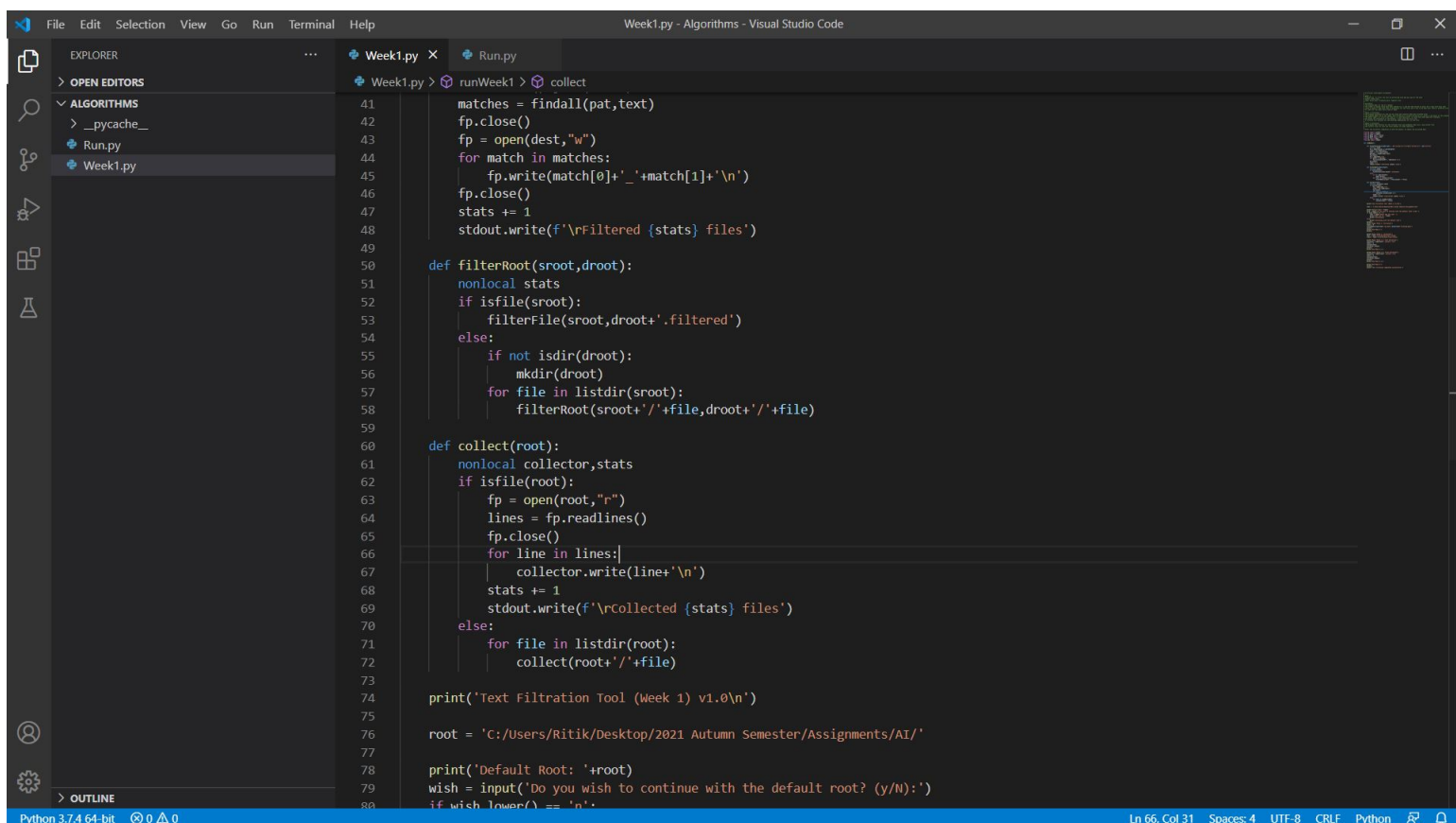


```
File Edit Selection View Go Run Terminal Help
Week1.py - Algorithms - Visual Studio Code

EXPLORER
> OPEN EDITORS
ALGORITHMS
  > __pycache__
    Run.py
    Week1.py

Week1.py
1  # Artificial Intelligence Assignment
2  #
3  # Week: 1
4  # Program Aim: To filter the text by extracting word and pos tag for the word
5  # Author: Ritik Jain
6  # Team: Ritik Jain, Priyanshu Garg, Yugantar Arya
7  #
8  # Procedure:
9  # The program does its job in 2 phases
10 # The number of phases could have been reduced to 1 ( and the time halved as well) but I went with this code
11 # since it doesn't take much time to preprocess all the files and it has to be done once; hence I preferred clarity
12 # of code over the small extra time it takes
13 #
14 # Phase 1 Filtration:
15 # The program processes all the raw xml files and converts them into filtered files
16 # The program makes use of the regex: (hw="([A-Za-z]+)"\\s+pos="([A-Za-z]+)"), which is the heart of the program
17 # The above regex is a strict one (that it is case-sensitive). It could have been made more flexible,
18 # but given the consistency of the dataset, I went with the strict one.
19 # A filtered file contains all the word_tag combinations for its xml file
20 #
21 # Phase 2 Collection:
22 # The program then collects all the filtered files and assembles them into 1 big collect file
23 # The collect files for Test and Train dataset are made separately
24 #
25 # After the successful completion of both the phases, we obtain the processed data
26
27 from os import listdir
28 from re import findall
29 from os.path import isfile
30 from os.path import isdir
31 from os import mkdir
32 from sys import stdout
33
34 def runWeek1():
35
36     def filterFile(source,dest,pat = 'hw="([A-Za-z]+)"\\s+pos="([A-Za-z]+)"',en="utf-8"):
37         nonlocal stats
38         fp = open(source,"r",encoding=en)
39         lines = fp.readlines()
40         text = str().join(lines)
```

Figure 1: Week-1 File: Section 1



```
File Edit Selection View Go Run Terminal Help
Week1.py - Algorithms - Visual Studio Code

EXPLORER
> OPEN EDITORS
ALGORITHMS
  > __pycache__
    Run.py
    Week1.py

Week1.py
41 matches = findall(pat,text)
42 fp.close()
43 fp = open(dest,"w")
44 for match in matches:
45     fp.write(match[0]+'_'+match[1]+'\\n')
46 fp.close()
47 stats += 1
48 stdout.write(f'\\rFiltered {stats} files')
49
50 def filterRoot(sroot,droot):
51     nonlocal stats
52     if isfile(sroot):
53         filterFile(sroot,droot+'.filtered')
54     else:
55         if not isdir(droot):
56             mkdir(droot)
57         for file in listdir(sroot):
58             filterRoot(sroot+'/'+file,droot+'/'+file)
59
60 def collect(root):
61     nonlocal collector,stats
62     if isfile(root):
63         fp = open(root,"r")
64         lines = fp.readlines()
65         fp.close()
66         for line in lines:
67             collector.write(line+'\\n')
68         stats += 1
69         stdout.write(f'\\rCollected {stats} files')
70     else:
71         for file in listdir(root):
72             collect(root+'/'+file)
73
74 print('Text Filtration Tool (Week 1) v1.0\\n')
75
76 root = 'C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/'
77
78 print('Default Root: '+root)
79 wish = input('Do you wish to continue with the default root? (y/n):')
80 if wish.lower() == 'n':
```

Figure 2: Week-1 File: Section 2

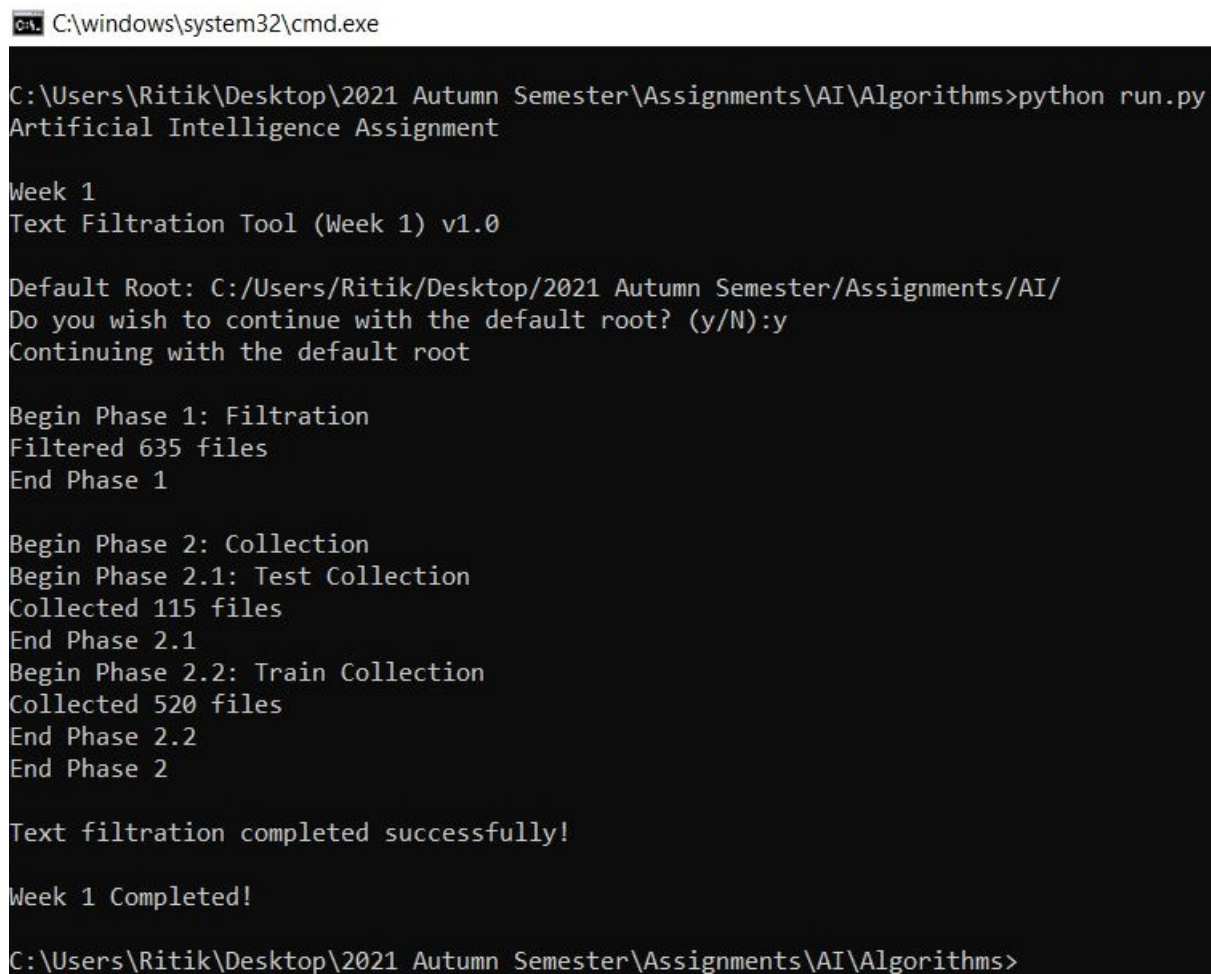
```
80     if wish.lower() == 'n':
81         root = input('Enter the new root: ')
82         print('Root set to: '+root)
83         print('Continuing')
84     else:
85         print('Continuing with the default root')
86     print()
87     print('Begin Phase 1: Filtration')
88     stats = 0
89     filterRoot(sroot=root+'raw_data',droot=root+'filtered_data')
90     print()
91     print('End Phase 1')
92     print()
93
94     print('Begin Phase 2: Collection')
95     test = root+'filtered_data/Test-corpus'
96     train = root+'filtered_data/Train-corpus'
97
98     print('Begin Phase 2.1: Test Collection')
99     collector = open(test+'.collect','w')
100    stats = 0
101    collect(test)
102    collector.close()
103    print()
104    print('End Phase 2.1')
105
106    print('Begin Phase 2.2: Train Collection')
107    collector = open(train+'.collect','w')
108    stats = 0
109    collect(train)
110    collector.close()
111    print()
112    print('End Phase 2.2')
113
114    print('End Phase 2')
115    print()
116    print('Text filtration completed successfully!')
```

Figure 3: Week-1 File: Section 3

```
1  # Artificial Intelligence Assignment
2  #
3  # Team: Ritik Jain, Priyanshu Garg, Yugantar Arya
4  #
5  # Main File to run the code
6  #
7  # Currently runs:
8  # 1. Week-1
9  #
10
11 from Week1 import runWeek1
12
13 if __name__ == "__main__":
14     print('Artificial Intelligence Assignment')
15     print()
16     print('Week 1')
17     runWeek1()
18     print()
19     print('Week 1 Completed!')
```

Figure 4: Main File: Run.py

1.4 Execution



```
C:\windows\system32\cmd.exe

C:\Users\Ritik\Desktop\2021 Autumn Semester\Assignments\AI\Algorithms>python run.py
Artificial Intelligence Assignment

Week 1
Text Filtration Tool (Week 1) v1.0

Default Root: C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/
Do you wish to continue with the default root? (y/N):y
Continuing with the default root

Begin Phase 1: Filtration
Filtered 635 files
End Phase 1

Begin Phase 2: Collection
Begin Phase 2.1: Test Collection
Collected 115 files
End Phase 2.1
Begin Phase 2.2: Train Collection
Collected 520 files
End Phase 2.2
End Phase 2

Text filtration completed successfully!

Week 1 Completed!

C:\Users\Ritik\Desktop\2021 Autumn Semester\Assignments\AI\Algorithms>
```

Figure 5: Execution: Run.py

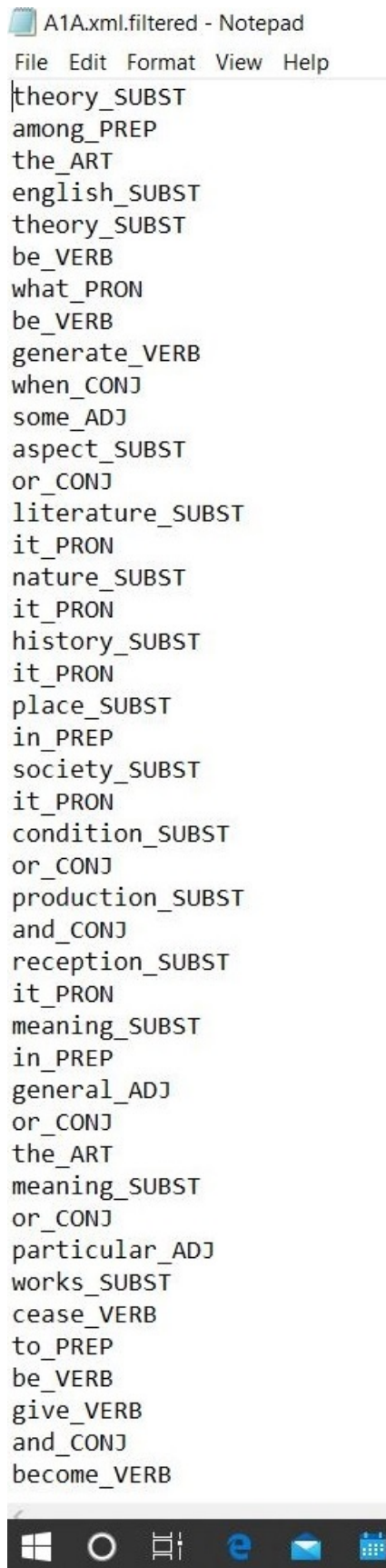
1.5 Output Files

This PC > Desktop > 2021 Autumn Semester > Assignments > AI > filtered_data			
Name	Date modified	Type	Size
Test-corpus	9/28/2020 11:40 A...	File folder	
Train-corpus	9/28/2020 11:45 A...	File folder	
Train-corups	9/28/2020 11:41 A...	File folder	
Test-corpus.collect	9/28/2020 12:34 PM	COLLECT File	46,342 KB
Train-corpus.collect	9/28/2020 12:34 PM	COLLECT File	116,311 KB

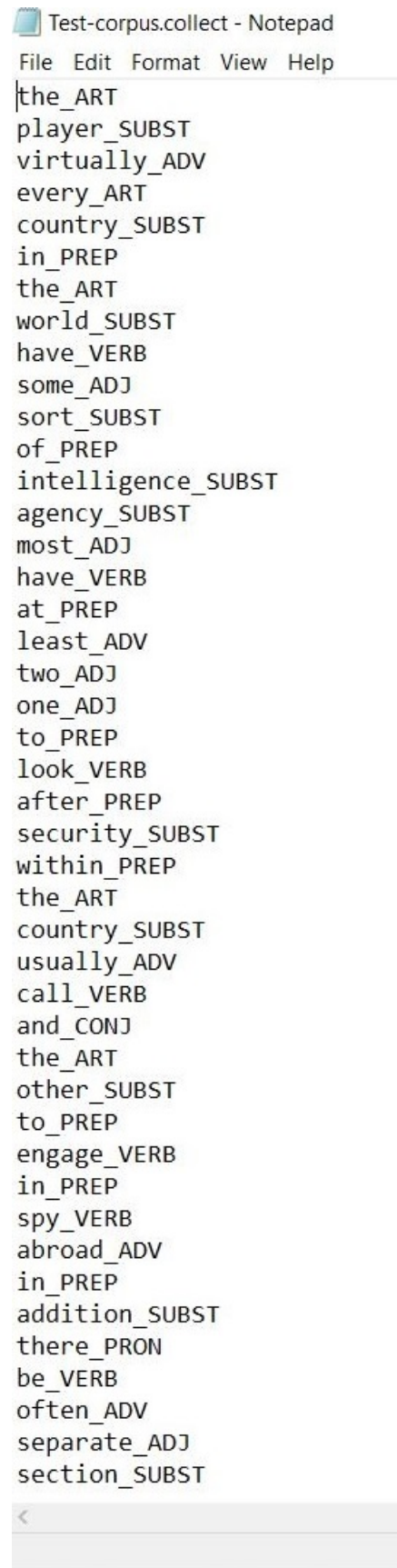
Figure 6: Phase 2: Collected Files

This PC > Desktop > 2021 Autumn Semester > Assignments > AI > filtered_data > Test-corpus > AN				
Name	Date modified	Type	Size	
AN0.xml.filtered	9/28/2020 12:32 PM	FILTERED File	399 KB	
AN1.xml.filtered	9/28/2020 12:32 PM	FILTERED File	168 KB	
AN2.xml.filtered	9/28/2020 12:32 PM	FILTERED File	366 KB	
AN3.xml.filtered	9/28/2020 12:32 PM	FILTERED File	444 KB	
AN4.xml.filtered	9/28/2020 12:32 PM	FILTERED File	587 KB	
AN5.xml.filtered	9/28/2020 12:32 PM	FILTERED File	421 KB	
AN7.xml.filtered	9/28/2020 12:32 PM	FILTERED File	405 KB	
AN8.xml.filtered	9/28/2020 12:32 PM	FILTERED File	374 KB	
AN9.xml.filtered	9/28/2020 12:32 PM	FILTERED File	457 KB	
ANA.xml.filtered	9/28/2020 12:32 PM	FILTERED File	346 KB	
ANB.xml.filtered	9/28/2020 12:32 PM	FILTERED File	337 KB	
ANC.xml.filtered	9/28/2020 12:32 PM	FILTERED File	408 KB	
AND.xml.filtered	9/28/2020 12:32 PM	FILTERED File	391 KB	
ANF.xml.filtered	9/28/2020 12:32 PM	FILTERED File	356 KB	
ANH.xml.filtered	9/28/2020 12:32 PM	FILTERED File	391 KB	
ANJ.xml.filtered	9/28/2020 12:32 PM	FILTERED File	152 KB	
ANK.xml.filtered	9/28/2020 12:32 PM	FILTERED File	389 KB	
ANL.xml.filtered	9/28/2020 12:32 PM	FILTERED File	428 KB	
ANM.xml.filtered	9/28/2020 12:32 PM	FILTERED File	286 KB	
ANP.xml.filtered	9/28/2020 12:32 PM	FILTERED File	232 KB	
ANR.xml.filtered	9/28/2020 12:32 PM	FILTERED File	415 KB	
ANS.xml.filtered	9/28/2020 12:32 PM	FILTERED File	315 KB	
ANT.xml.filtered	9/28/2020 12:32 PM	FILTERED File	431 KB	
ANU.xml.filtered	9/28/2020 12:32 PM	FILTERED File	424 KB	
ANX.xml.filtered	9/28/2020 12:32 PM	FILTERED File	592 KB	
ANY.xml.filtered	9/28/2020 12:32 PM	FILTERED File	452 KB	

Figure 7: Phase 1: Filtered Files



(a) A Filtered File



(b) A Collected File