



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
1.2.3	Phase 3 Verification	i
1.3	Code Screenshots	ii
1.4	Execution	v
1.5	Output Files	vi
1.6	Results	ix
2	Week 2	x
2.1	Objective	x
2.2	Procedure	x
2.2.1	Phase 1: Construction	x
2.2.2	Phase 2: Serialization	x
2.3	Code Screenshots	xi
2.4	Execution	xiii
2.5	Output Files	xiv
2.6	Results	xv

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 3 phases. The number of phases could have been reduced to 1 but I went with this code since it doesn't take much time to pre-process all the files and it has to be done only once; hence I preferred clarity of code over the small extra time it takes.

1.2.1 Phase 1 Filtration

The program processes all the raw xml files and converts them into filtered files. The program makes use of the regex, which is the heart of the program:

$$<\backslash s^*w[^>]^*pos="([A-Za-z]^*)"[^>]^*>\backslash s^*([^\backslash s]^*)\backslash s^*<\backslash s^*/w\backslash s^*>$$

The above regex is a flexible one. It could have been made stricter, given the consistency of the dataset, but I still went with the flexible version since it is more general. A filtered file contains all the word_tag combinations for its xml file. The word_tag combinations are stored in a case sensitive manner.

1.2.2 Phase 2 Collection

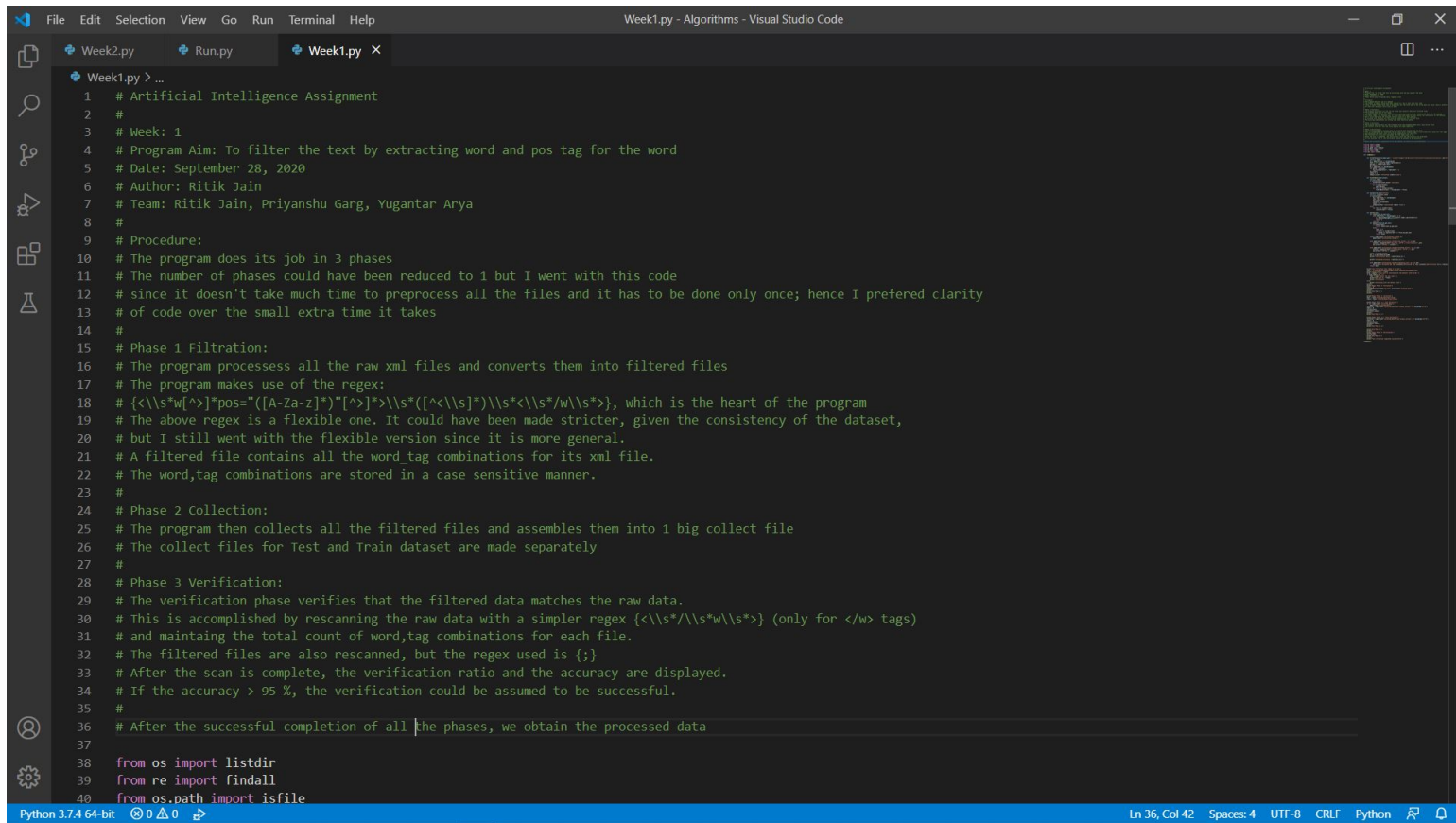
The program then collects all the filtered files and assembles them into 1 big collect file. The collect files for Test and Train dataset are made separately.

1.2.3 Phase 3 Verification

The verification phase verifies that the filtered data matches the raw data. This is accomplished by re-scanning the raw data with a simpler regex $<\backslash s^*/\backslash s^*w\backslash s^*>$ (only for $</w>$ tags) and maintaining the total count of word_tag combinations for each file. The filtered files are also re-scanned, but the regex used is $;$. After the scan is complete, the verification ratio and the accuracy are displayed. If the accuracy is more than 95%, the verification could be assumed to be successful.

After the successful completion of all the phases, we obtain the processed data.

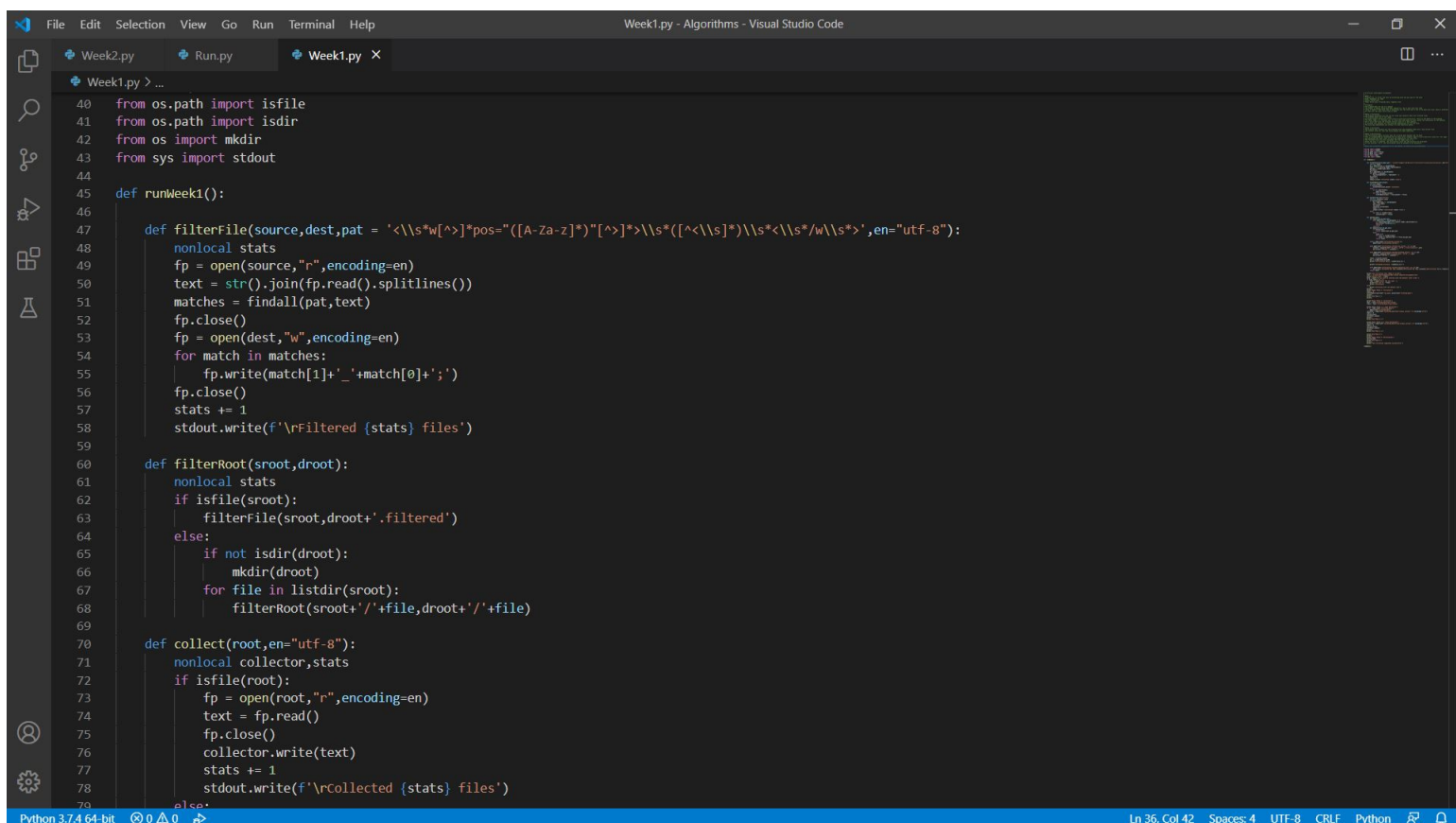
1.3 Code Screenshots



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

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 # Date: September 28, 2020
6 # Author: Ritik Jain
7 # Team: Ritik Jain, Priyanshu Garg, Yugantar Arya
8 #
9 # Procedure:
10 # The program does its job in 3 phases
11 # The number of phases could have been reduced to 1 but I went with this code
12 # since it doesn't take much time to preprocess all the files and it has to be done only once; hence I preferred clarity
13 # of code over the small extra time it takes
14 #
15 # Phase 1 Filtration:
16 # The program processes all the raw xml files and converts them into filtered files
17 # The program makes use of the regex:
18 # {<\s*w[>]*pos="([A-Za-z]*)"[^>]*>\s*([^\s]*)\s*<\s*w\s*>}, which is the heart of the program
19 # The above regex is a flexible one. It could have been made stricter, given the consistency of the dataset,
20 # but I still went with the flexible version since it is more general.
21 # A filtered file contains all the word_tag combinations for its xml file.
22 # The word_tag combinations are stored in a case sensitive manner.
23 #
24 # Phase 2 Collection:
25 # The program then collects all the filtered files and assembles them into 1 big collect file
26 # The collect files for Test and Train dataset are made separately
27 #
28 # Phase 3 Verification:
29 # The verification phase verifies that the filtered data matches the raw data.
30 # This is accomplished by rescanning the raw data with a simpler regex {<\s*/\s*w\s*>} (only for </w> tags)
31 # and maintaing the total count of word_tag combinations for each file.
32 # The filtered files are also rescanned, but the regex used is {;};
33 # After the scan is complete, the verification ratio and the accuracy are displayed.
34 # If the accuracy > 95 %, the verification could be assumed to be successful.
35 #
36 # After the successful completion of all the phases, we obtain the processed data
37
38 from os import listdir
39 from re import findall
40 from os.path import isfile
```

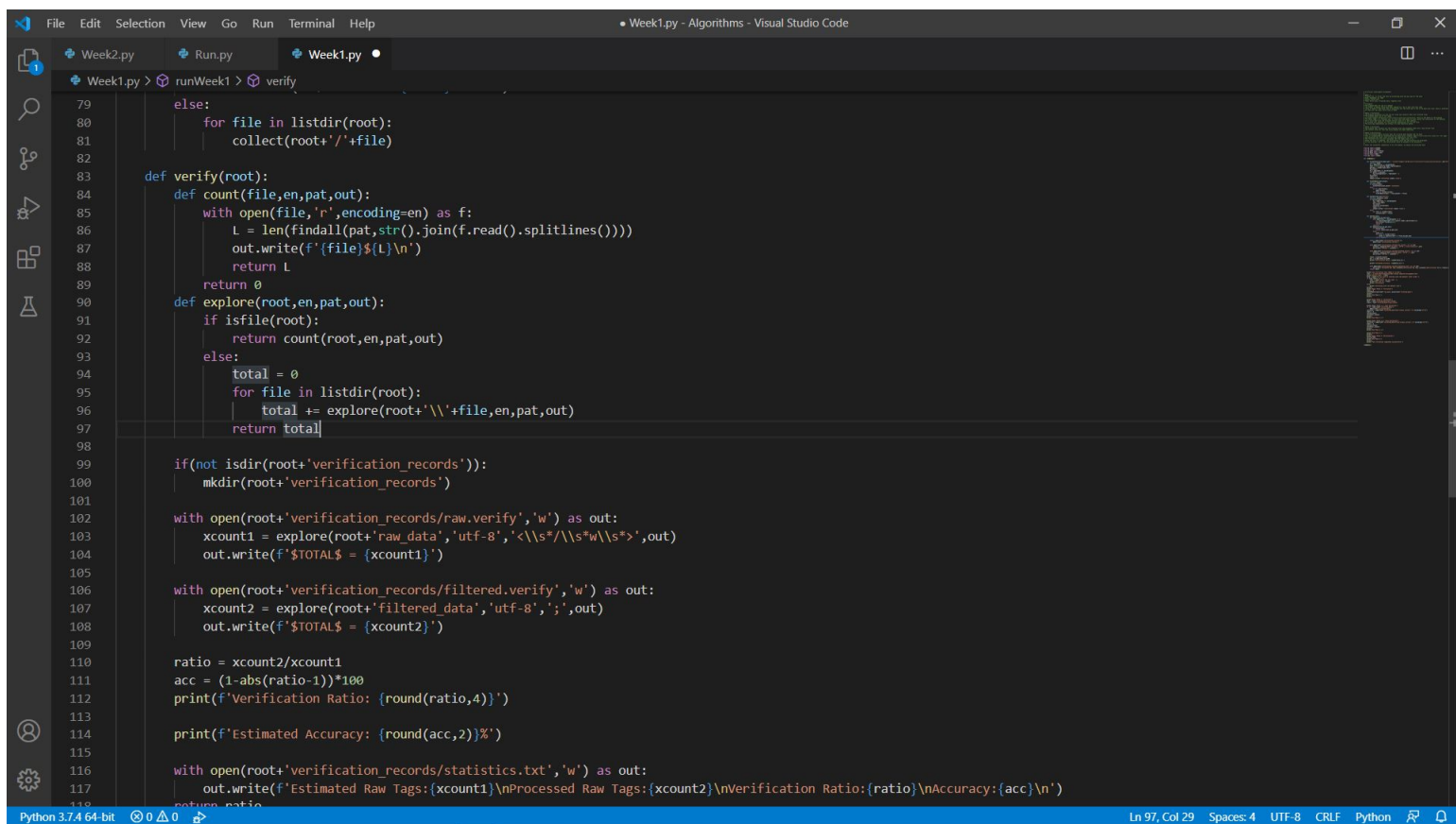
Figure 1: Week-1 File: Section 1



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

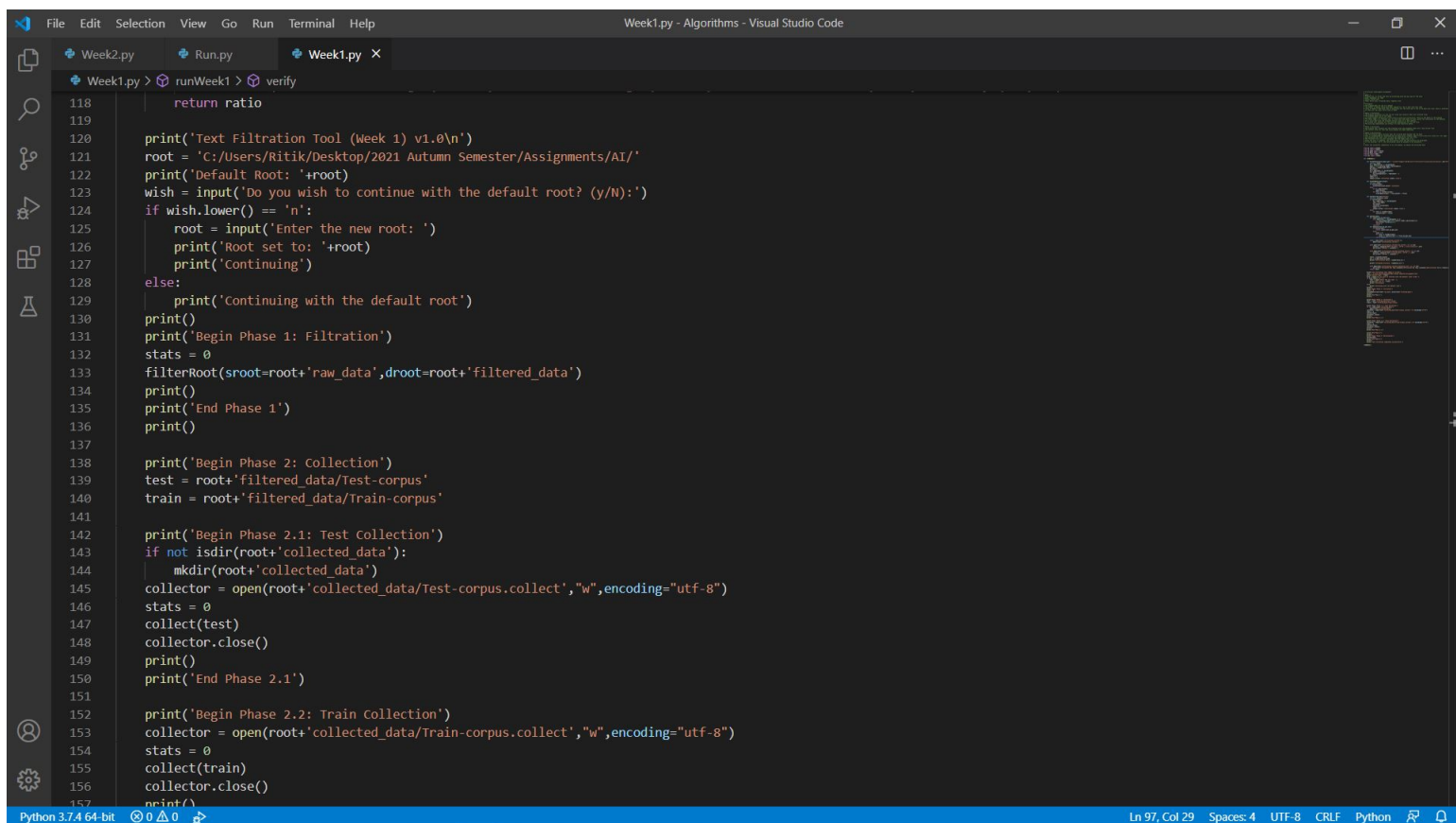
Week1.py > ...
40 from os.path import isfile
41 from os.path import isdir
42 from os import mkdir
43 from sys import stdout
44
45 def runWeek1():
46
47     def filterFile(source,dest,pat = '<\s*w[>]*pos="([A-Za-z]*)"[^>]*>\s*([^\s]*)\s*<\s*w\s*>','en="utf-8")':
48         nonlocal stats
49         fp = open(source,"r",encoding=en)
50         text = str().join(fp.read().splitlines())
51         matches = findall(pat,text)
52         fp.close()
53         fp = open(dest,"w",encoding=en)
54         for match in matches:
55             fp.write(match[1]+'_'+match[0]+';')
56         fp.close()
57         stats += 1
58         stdout.write(f'\rFiltered {stats} files')
59
60     def filterRoot(sroot,droot):
61         nonlocal stats
62         if isfile(sroot):
63             filterFile(sroot,droot+'.filtered')
64         else:
65             if not isdir(droot):
66                 mkdir(droot)
67             for file in listdir(sroot):
68                 filterRoot(sroot+'/'+file,droot+'/'+file)
69
70     def collect(root,en="utf-8"):
71         nonlocal collector,stats
72         if isfile(root):
73             fp = open(root,"r",encoding=en)
74             text = fp.read()
75             fp.close()
76             collector.write(text)
77             stats += 1
78             stdout.write(f'\rCollected {stats} files')
79         else:
```

Figure 2: Week-1 File: Section 2



```
79         else:
80             for file in listdir(root):
81                 collect(root+'/'+file)
82
83     def verify(root):
84         def count(file,en,pat,out):
85             with open(file,'r',encoding=en) as f:
86                 L = len(findall(pat,str(f.read()).splitlines()))
87                 out.write(f'{file}${L}\n')
88                 return L
89             return 0
90         def explore(root,en,pat,out):
91             if isfile(root):
92                 return count(root,en,pat,out)
93             else:
94                 total = 0
95                 for file in listdir(root):
96                     total += explore(root+'\\'+file,en,pat,out)
97             return total
98
99         if(not isdir(root+'verification_records')):
100             mkdir(root+'verification_records')
101
102         with open(root+'verification_records/raw.verify','w') as out:
103             xcount1 = explore(root+'raw_data','utf-8','<\\s*/\\s*w\\s*>',out)
104             out.write(f'$TOTAL$ = {xcount1}')
105
106         with open(root+'verification_records/filtered.verify','w') as out:
107             xcount2 = explore(root+'filtered_data','utf-8',';',out)
108             out.write(f'$TOTAL$ = {xcount2}')
109
110         ratio = xcount2/xcount1
111         acc = (1-abs(ratio-1))*100
112         print(f'Verification Ratio: {round(ratio,4)}')
113
114         print(f'Estimated Accuracy: {round(acc,2)}%')
115
116         with open(root+'verification_records/statistics.txt','w') as out:
117             out.write(f'Estimated Raw Tags:{xcount1}\nProcessed Raw Tags:{xcount2}\nVerification Ratio:{ratio}\nAccuracy:{acc}\n')
```

Figure 3: Week-1 File: Section 3



```
118     return ratio
119
120     print('Text Filtration Tool (Week 1) v1.0\n')
121     root = 'C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/'
122     print('Default Root: '+root)
123     wish = input('Do you wish to continue with the default root? (y/N):')
124     if wish.lower() == 'n':
125         root = input('Enter the new root: ')
126         print('Root set to: '+root)
127         print('Continuing')
128     else:
129         print('Continuing with the default root')
130     print()
131     print('Begin Phase 1: Filtration')
132     stats = 0
133     filterRoot(sroot=root+'raw_data',droot=root+'filtered_data')
134     print()
135     print('End Phase 1')
136     print()
137
138     print('Begin Phase 2: Collection')
139     test = root+'filtered_data/Test-corpus'
140     train = root+'filtered_data/Train-corpus'
141
142     print('Begin Phase 2.1: Test Collection')
143     if not isdir(root+'collected_data'):
144         mkdir(root+'collected_data')
145     collector = open(root+'collected_data/Test-corpus.collect','w',encoding='utf-8')
146     stats = 0
147     collect(test)
148     collector.close()
149     print()
150     print('End Phase 2.1')
151
152     print('Begin Phase 2.2: Train Collection')
153     collector = open(root+'collected_data/Train-corpus.collect','w',encoding='utf-8')
154     stats = 0
155     collect(train)
156     collector.close()
157     print()
```

Figure 4: Week-1 File: Section 4

```
File Edit Selection View Go Run Terminal Help
Week1.py Run.py Week1.py x
Week1.py > runWeek1 > verify
157 print()
158 print('End Phase 2.2')
159
160 print('End Phase 2')
161 print()
162 print('Begin Phase 3: Verification')
163 verify(root)
164 print('End Phase 3')
165 print()
166 print('Text filtration completed successfully!')
167
168 runWeek1()

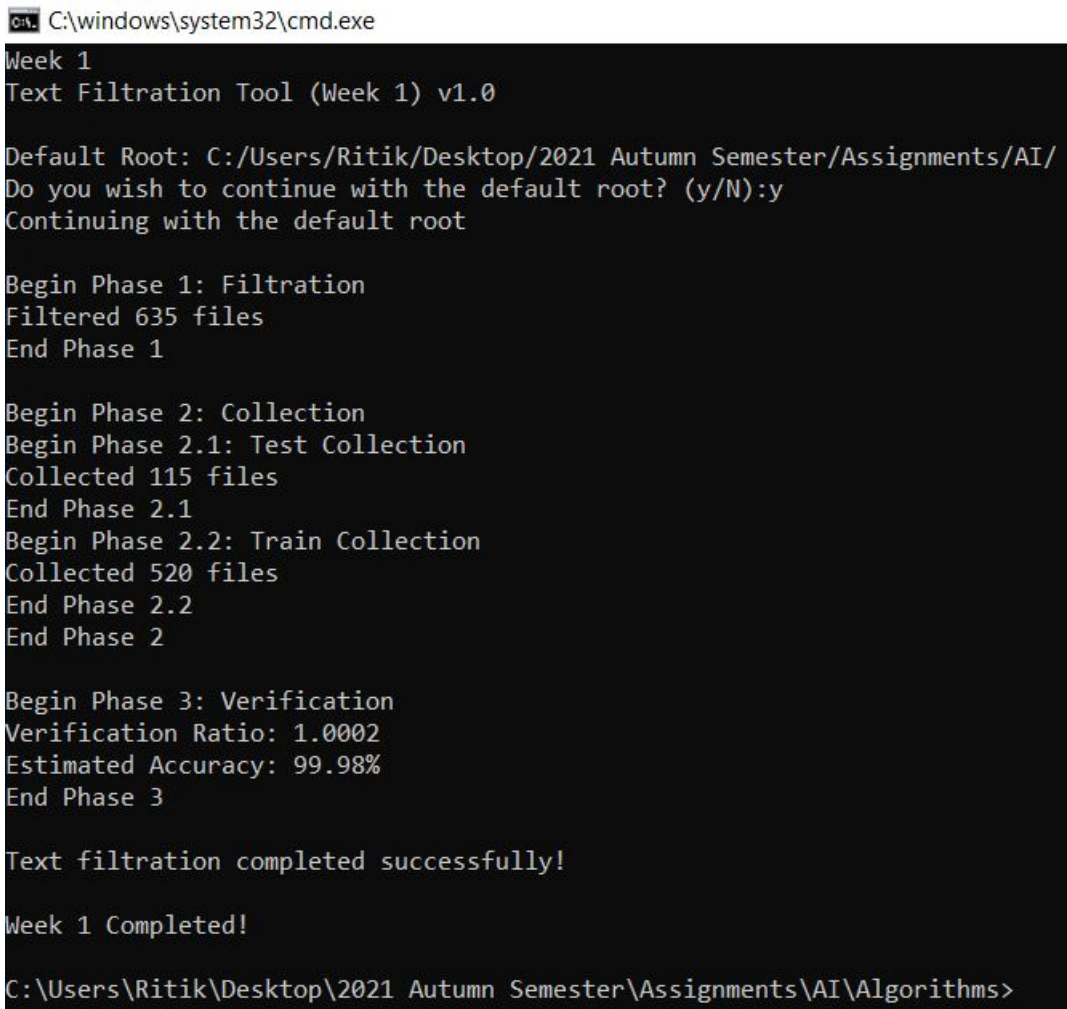
Python 3.7.4 64-bit 0 0 0 Ln 97, Col 29 Spaces: 4 UTF-8 CRLF Python
```

Figure 5: Week-1 File: Section 5

```
File Edit Selection View Go Run Terminal Help
Run.py Algorithms - Visual Studio Code
Week2.py Run.py x Week1.py
Run.py > ...
2 #
3 # Date: September 28, 2020
4 # Author: Ritik Jain
5 # Team: Ritik Jain, Priyanshu Garg, Yugantar Arya
6 #
7 # Main File to run the code
8 #
9 # Currently runs:
10 # 1. Week-1
11
12 from Week1 import runWeek1
13 from Week2 import runWeek2
14
15 if __name__ == "__main__":
16     print('Artificial Intelligence Assignment')
17     print()
18     print('Week 1')
19     runWeek1()
20     print()
21     print('Week 1 Completed!')
```

Figure 6: Main File: Run.py

1.4 Execution



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

Begin Phase 3: Verification
Verification Ratio: 1.0002
Estimated Accuracy: 99.98%
End Phase 3

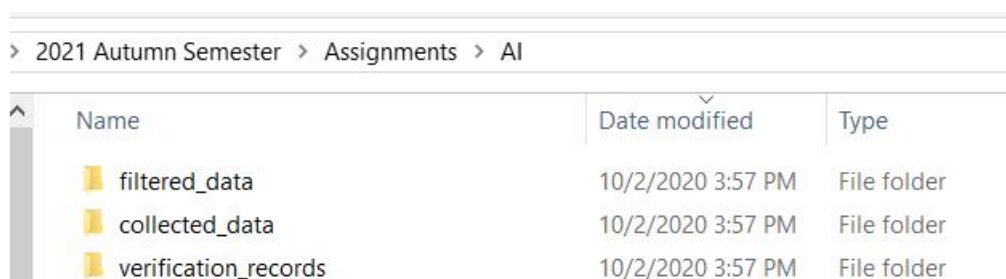
Text filtration completed successfully!

Week 1 Completed!

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

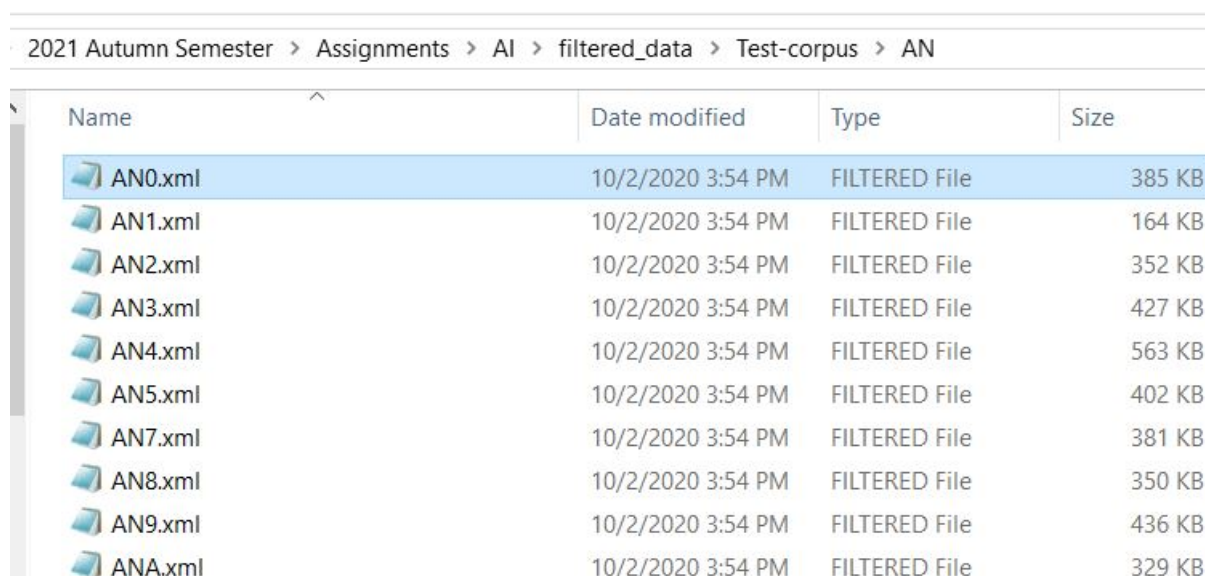
Figure 7: Execution: Run.py

1.5 Output Files



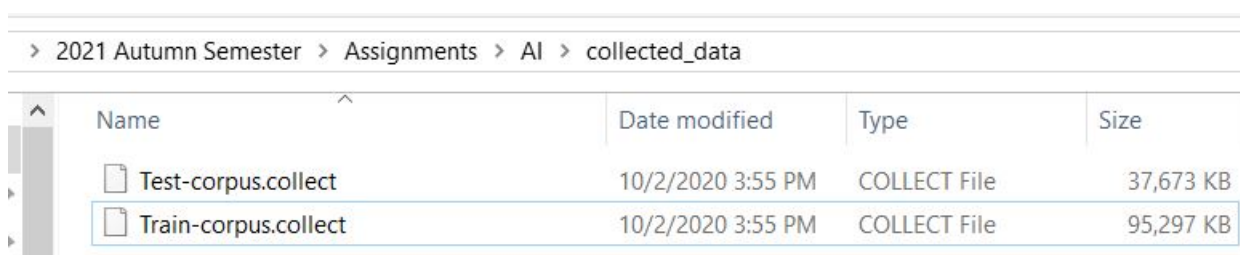
> 2021 Autumn Semester > Assignments > AI			
Name	Date modified	Type	
filtered_data	10/2/2020 3:57 PM	File folder	
collected_data	10/2/2020 3:57 PM	File folder	
verification_records	10/2/2020 3:57 PM	File folder	

Figure 8: Generated Folders



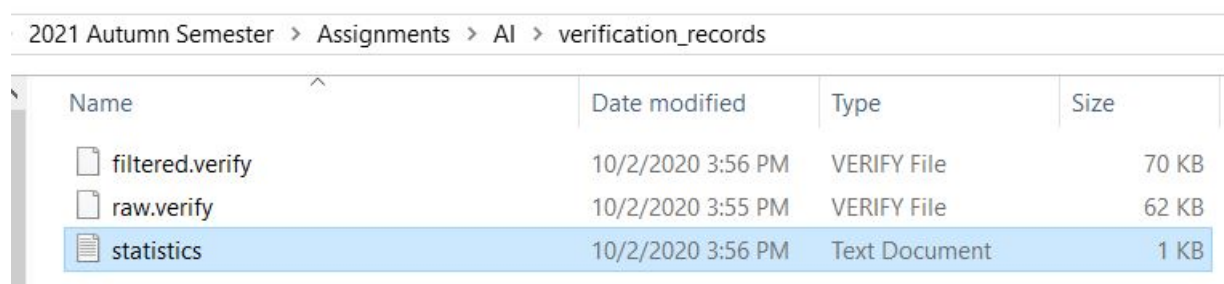
2021 Autumn Semester > Assignments > AI > filtered_data > Test-corpus > AN				
Name	Date modified	Type	Size	
AN0.xml	10/2/2020 3:54 PM	FILTERED File	385 KB	
AN1.xml	10/2/2020 3:54 PM	FILTERED File	164 KB	
AN2.xml	10/2/2020 3:54 PM	FILTERED File	352 KB	
AN3.xml	10/2/2020 3:54 PM	FILTERED File	427 KB	
AN4.xml	10/2/2020 3:54 PM	FILTERED File	563 KB	
AN5.xml	10/2/2020 3:54 PM	FILTERED File	402 KB	
AN7.xml	10/2/2020 3:54 PM	FILTERED File	381 KB	
AN8.xml	10/2/2020 3:54 PM	FILTERED File	350 KB	
AN9.xml	10/2/2020 3:54 PM	FILTERED File	436 KB	
ANA.xml	10/2/2020 3:54 PM	FILTERED File	329 KB	

Figure 9: Phase 1 Filtered Files



> 2021 Autumn Semester > Assignments > AI > collected_data				
Name	Date modified	Type	Size	
Test-corpus.collect	10/2/2020 3:55 PM	COLLECT File	37,673 KB	
Train-corpus.collect	10/2/2020 3:55 PM	COLLECT File	95,297 KB	

Figure 10: Phase 2 Collected Files



2021 Autumn Semester > Assignments > AI > verification_records				
Name	Date modified	Type	Size	
filtered.verify	10/2/2020 3:56 PM	VERIFY File	70 KB	
raw.verify	10/2/2020 3:55 PM	VERIFY File	62 KB	
statistics	10/2/2020 3:56 PM	Text Document	1 KB	

Figure 11: Phase 3 Verification Files

AN5.xml - Notepad

File Edit Format View Help

Parental_ADJ;rights_SUBST;and_CONJ;responsibilities_SUBST;INTRODUCTION_SUBST
J;policy_SUBST;considerations_SUBST;might_VERB;outweigh_VERB;individual_ADJ;
RB;have_VERB;regard_SUBST;when_CONJ;exercising_VERB;their_PRON;functions_SUB
dren_SUBST;'s_UNC;and_CONJ;state_VERB;interests_SUBST;and_CONJ;rights_SUBST;
n_ADJ;decisions_SUBST;in_PREP;respect_SUBST;of_PREP;their_PRON;own_ADJ;lives
NJ;child_SUBST;interrelate_SUBST;Put_VERB;simply_ADV;LEAs_SUBST;have_VERB;a_
BST;that_CONJ;the_ART;state_SUBST;system_SUBST;of_PREP;education_SUBST;is_VE
and_CONJ;shameful_ADJ;to_PREP;himself_PRON;The_ART;distribution_SUBST;of_PRE
ERB;not_ADV;surprising_ADJ;then_ADV;that_CONJ;areas_SUBST;such_ADJ;as_PREP;s
er_PRON;from_PREP;school_SUBST;The_ART;case_SUBST;of_PREP;RE_SUBST;S_SUBST;A
_CONJ;local_ADJ;authority_SUBST;rather_ADV;than_CONJ;those_ADJ;of_PREP;the_A
BST;in_PREP;the_ART;balance_SUBST;of_PREP;power_SUBST;between_PREP;parent_SU
J;there_PRON;is_VERB;inevitably_ADV;dispute_SUBST;between_PREP;parent_SUBST;
With_PREP;the_ART;exception_SUBST;of_PREP;the_ART;courts_SUBST;those_ADJ;det
;challenged_VERB;the_ART;LEA_SUBST;'s_UNC;decisions_SUBST;both_ADV;individua
the_ART;case_SUBST;at_PREP;the_ART;school_SUBST;designated_VERB;by_PREP;the_
NJ;found_VERB;places_SUBST;for_PREP;the_ART;children_SUBST;at_PREP;two_ADJ;s

Figure 12: A filtered File

Test-corpus.collect - Notepad

File Edit Format View Help

THE_ART;PLAYERS_SUBST;Virtually_ADV;every_ART;country_SUBST;in_PREP;the_ART;world_SUBST;has_VERB;so
RT;process_SUBST;exaggerates_VERB;and_CONJ;distorts_VERB;its_PRON;reports_SUBST;so_ADV;as_CONJ;to_F
B;that_CONJ;this_ADJ;is_VERB;only_ADV;fiction_SUBST;there_ADV;nevertheless_ADV;remains_VERB;a_ART;f
;the_ART;endless_ADJ;paperwork_SUBST;that_CONJ;forms_VERB;the_ART;major_ADJ;part_SUBST;of_PREP;any_
ne_ADJ;recent_ADJ;advertisement_SUBST;carried_VERB;a_ART;photograph_SUBST;of_PREP;a_ART;rather_ADV;s
nt_SUBST;Six_ADJ;MI6_UNC;often_ADV;referred_VERB;to_PREP;as_PREP;the_ART;Secret_ADJ;Intelligence_SU
e_ART;Foreign_ADJ;Office_SUBST;MI5_UNC;'s_VERB;initial_ADJ;task_SUBST;was_VERB;to_PREP;counter_VERB
a_ART;particularly_ADV;difficult_ADJ;task_SUBST;since_CONJ;they_PRON;were_VERB;all_ADJ;pathetically
uestion_SUBST;Whether_CONJ;the_ART;Germans_SUBST;were_VERB;quite_ADV;as_ADV;easily_ADV;fooled_VERB;
RON;own_ADJ;views_SUBST;and_CONJ;no_ART;one_PRON;was_VERB;willing_ADJ;to_PREP;disagree_VERB;with_PR
PREP;1952_ADJ;the_ART;then_ADJ;Home_SUBST;Secretary_SUBST;Sir_SUBST;David_SUBST;Maxwell_SUBST;Fyfe_
_VERB;the_ART;Defence_SUBST;of_PREP;the_ART;Realm_SUBST;as_PREP;a_ART;whole_SUBST;from_PREP;externa
VERB;colour_SUBST;to_PREP;any_ADJ;suggestion_SUBST;that_CONJ;it_PRON;is_VERB;concerned_ADJ;with_PRE
RT;Security_SUBST;Service_SUBST;in_PREP;particular_ADJ;cases_SUBST;but_CONJ;are_VERB;furnished_VERB
BST;First_ADJ;though_CONJ;MI5_UNC;is_VERB;notionally_ADV;under_PREP;the_ART;control_SUBST;of_PREP;t
BST;is_VERB;consistently_ADV;and_CONJ;deliberately_ADV;ignored_VERB;The_ART;absence_SUBST;of_PREP;a
nthonny_SUBST;Blunt_SUBST;who_PRON;finally_ADV;confessed_VERB;in_PREP;1964_ADJ;to_PREP;being_VERB;a_
al_ADJ;secrets_SUBST;while_CONJ;working_VERB;at_PREP;the_ART;Admiralty_SUBST;Frank_SUBST;Bossard_SU
e_ART;network_SUBST;consisted_VERB;of_PREP;Harry_SUBST;Houghton_SUBST;and_CONJ;Ethel_SUBST;Gee_INTE
SUBST;is_VERB;that_CONJ;MI5_UNC;has_VERB;been_VERB;penetrated_VERB;by_PREP;Russian_ADJ;intelligence
PREP;a_ART;homosexual_SUBST;and_CONJ;blackmailed_VERB;He_PRON;later_ADV;lived_VERB;in_PREP;an_ART;e
5_UNC;'s_UNC;inefficiency_SUBST;is_VERB;that_CONJ;it_PRON;wastes_VERB;far_ADV;too_ADV;much_ADJ;time
;student_SUBST;and_CONJ;industrial_ADJ;militancy_SUBST;in_PREP;Britain_SUBST;the_ART;numbers_SUBST;
_UNC;thinks_VERB;there_PRON;are_VERB;that_ADV;many_ADJ;subversives_SUBST;spies_SUBST;and_CONJ;trait
SUBST;that_CONJ;MI5_UNC;and_CONJ;Special_ADJ;Branch_SUBST;alone_ADV;decide_VERB;who_PRON;merits_VER
J;no_ART;minister_SUBST;would_VERB;authorise_VERB;a_ART;warrant_SUBST;for_PREP;such_ADJ;tapping_VER
f_PREP;£1.6_UNC;billion_ADJ;on_ADV;to_PREP;what_PRON;will_VERB;become_VERB;the_ART;country_SUBST;'s

Figure 13: A collected File

raw.verify - Notepad

File Edit Format View Help

```
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/raw_data/Test-corpus\AN\AN0.xml$36451
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/raw_data/Test-corpus\AN\AN1.xml$15823
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/raw_data/Test-corpus\AN\AN2.xml$35145
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/raw_data/Test-corpus\AN\AN3.xml$39292
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/raw_data/Test-corpus\AN\AN4.xml$53723
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/raw_data/Test-corpus\AN\AN5.xml$37421
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/raw_data/Test-corpus\AN\AN7.xml$38382
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/raw_data/Test-corpus\AN\AN8.xml$35287
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/raw_data/Test-corpus\AN\AN9.xml$40765
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/raw_data/Test-corpus\AN\ANA.xml$30970
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/raw_data/Test-corpus\AN\ANB.xml$31242
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/raw_data/Test-corpus\AN\ANC.xml$37904
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/raw_data/Test-corpus\AN\AND.xml$34872
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/raw_data/Test-corpus\AN\ANF.xml$32813
```

Figure 14: Raw Data Verification File

filtered.verify - Notepad

File Edit Format View Help

```
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\AN0.xml.filtered$36454
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\AN1.xml.filtered$15823
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\AN2.xml.filtered$35147
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\AN3.xml.filtered$39293
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\AN4.xml.filtered$53737
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\AN5.xml.filtered$37421
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\AN7.xml.filtered$38383
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\AN8.xml.filtered$35287
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\AN9.xml.filtered$40765
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\ANA.xml.filtered$30975
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\ANB.xml.filtered$31242
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\ANC.xml.filtered$37911
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\AND.xml.filtered$34872
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\ANF.xml.filtered$32813
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\ANH.xml.filtered$35106
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\ANJ.xml.filtered$14256
C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/filtered_data/Test-corpus\AN\ANK.xml.filtered$35772
```

Figure 15: Filtered Data Verification File

statistics - Notepad

File Edit Format View Help

```
Estimated Raw Tags:12744948
Processed Raw Tags:12747719
Verification Ratio:1.000217419482606
Accuracy:99.97825805173942
```

Figure 16: Statistics File

1.6 Results

The raw data was processed with an estimated accuracy of 99.98% or more. We could have made the data more accurate (to about 100%) by improving the regex for scanning raw files, but it would simply not be worth the trouble of investigating the patterns in the huge amount of data, only to correct an error of 0.02% or less, which would hardly affect the accuracy of the application. This data is now ready to be converted into a frequency dictionary in week 2.

2 Week 2

Date: October 2, 2020

2.1 Objective

To create a dictionary which maps word,tag to its frequency

2.2 Procedure

The application works in two phases. The first phase involves the construction of a human-readable dictionary files from both the collected test and train files. The second phase involves the construction of a serialized dictionary file for fast-access.

2.2.1 Phase 1: Construction

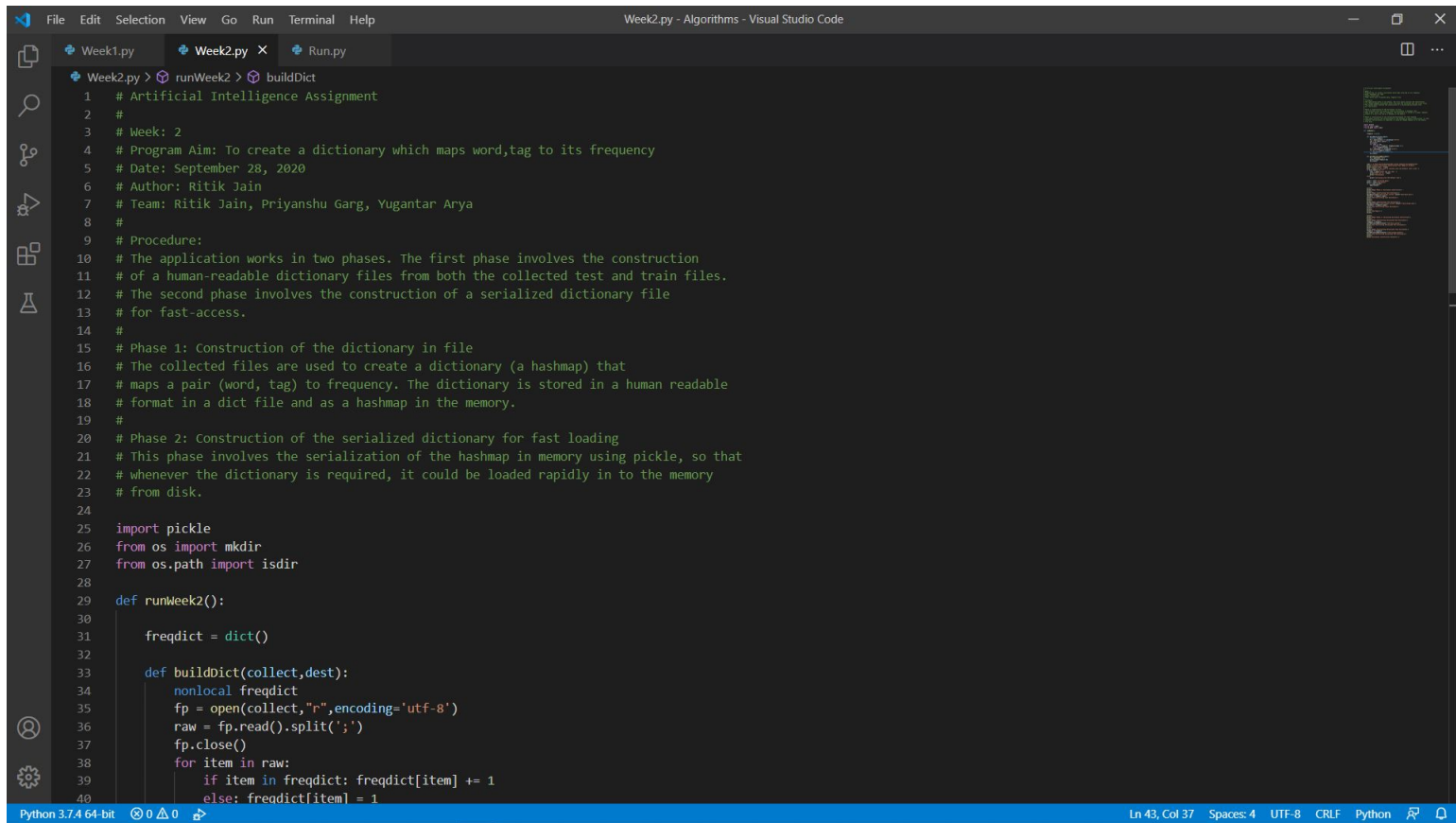
Construction of the dictionary in file The collected files are used to create a dictionary (a hashmap) that maps a pair (word, tag) to frequency. The dictionary is stored in a human readable format in a dict file and as a hashmap in the memory.

2.2.2 Phase 2: Serialization

Construction of the serialized dictionary for fast loading This phase involves the serialization of the hashmap in memory using pickle, so that whenever the dictionary is required, it could be loaded rapidly in to the memory from disk.

After the successful execution of the program, we get the required dictionary. Verification phase is not required since it was already done in week 1.

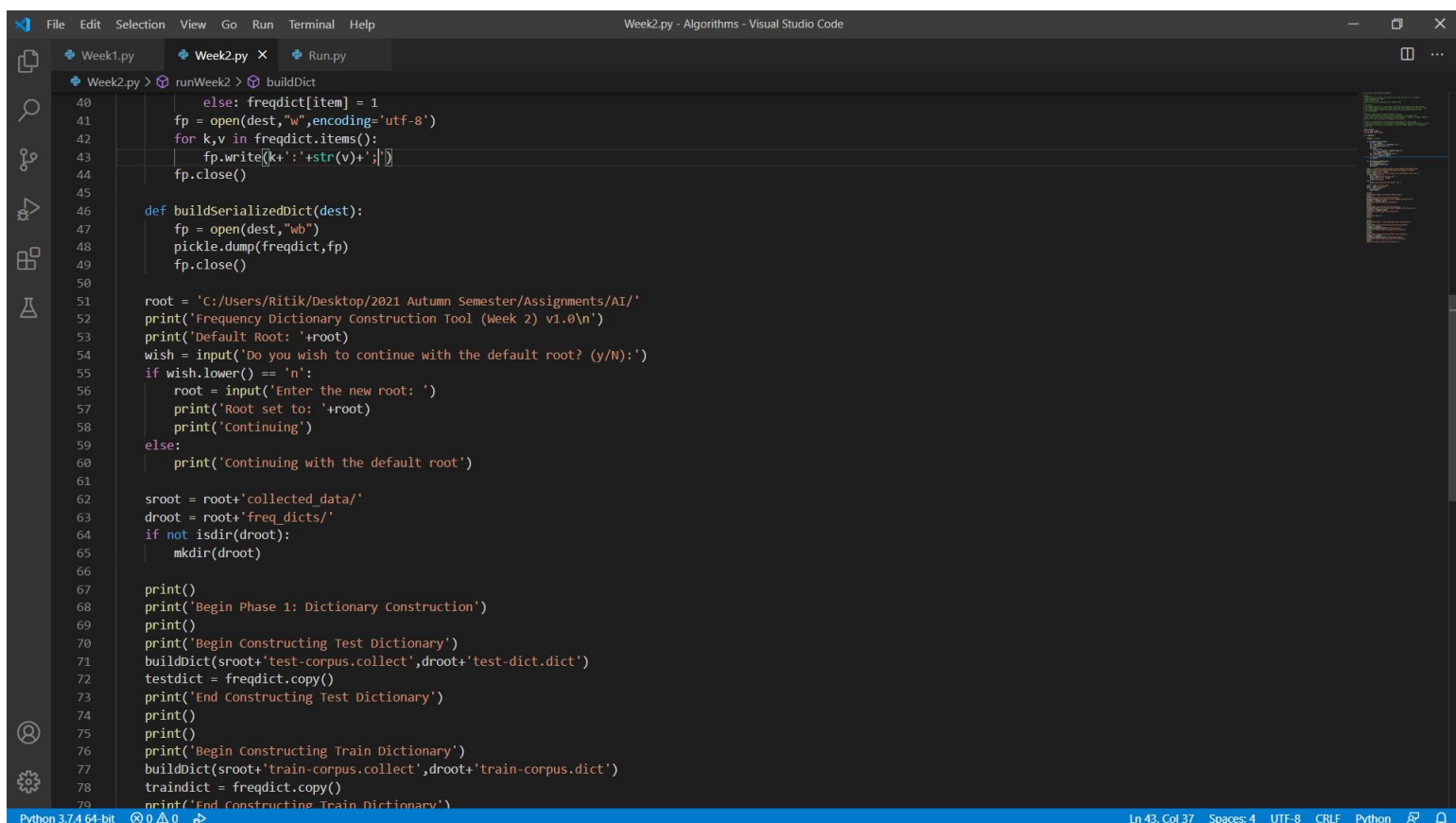
2.3 Code Screenshots



```
File Edit Selection View Go Run Terminal Help
Week2.py - Algorithms - Visual Studio Code

Week1.py Week2.py x Run.py
Week2.py > runWeek2 > buildDict
1 # Artificial Intelligence Assignment
2 #
3 # Week: 2
4 # Program Aim: To create a dictionary which maps word,tag to its frequency
5 # Date: September 28, 2020
6 # Author: Ritik Jain
7 # Team: Ritik Jain, Priyanshu Garg, Yugantar Arya
8 #
9 # Procedure:
10 # The application works in two phases. The first phase involves the construction
11 # of a human-readable dictionary files from both the collected test and train files.
12 # The second phase involves the construction of a serialized dictionary file
13 # for fast-access.
14 #
15 # Phase 1: Construction of the dictionary in file
16 # The collected files are used to create a dictionary (a hashmap) that
17 # maps a pair (word, tag) to frequency. The dictionary is stored in a human readable
18 # format in a dict file and as a hashmap in the memory.
19 #
20 # Phase 2: Construction of the serialized dictionary for fast loading
21 # This phase involves the serialization of the hashmap in memory using pickle, so that
22 # whenever the dictionary is required, it could be loaded rapidly in to the memory
23 # from disk.
24
25 import pickle
26 from os import mkdir
27 from os.path import isdir
28
29 def runWeek2():
30
31     freqdict = dict()
32
33     def buildDict(collect,dest):
34         nonlocal freqdict
35         fp = open(collect,"r",encoding='utf-8')
36         raw = fp.read().split(';')
37         fp.close()
38         for item in raw:
39             if item in freqdict: freqdict[item] += 1
40             else: freqdict[item] = 1
```

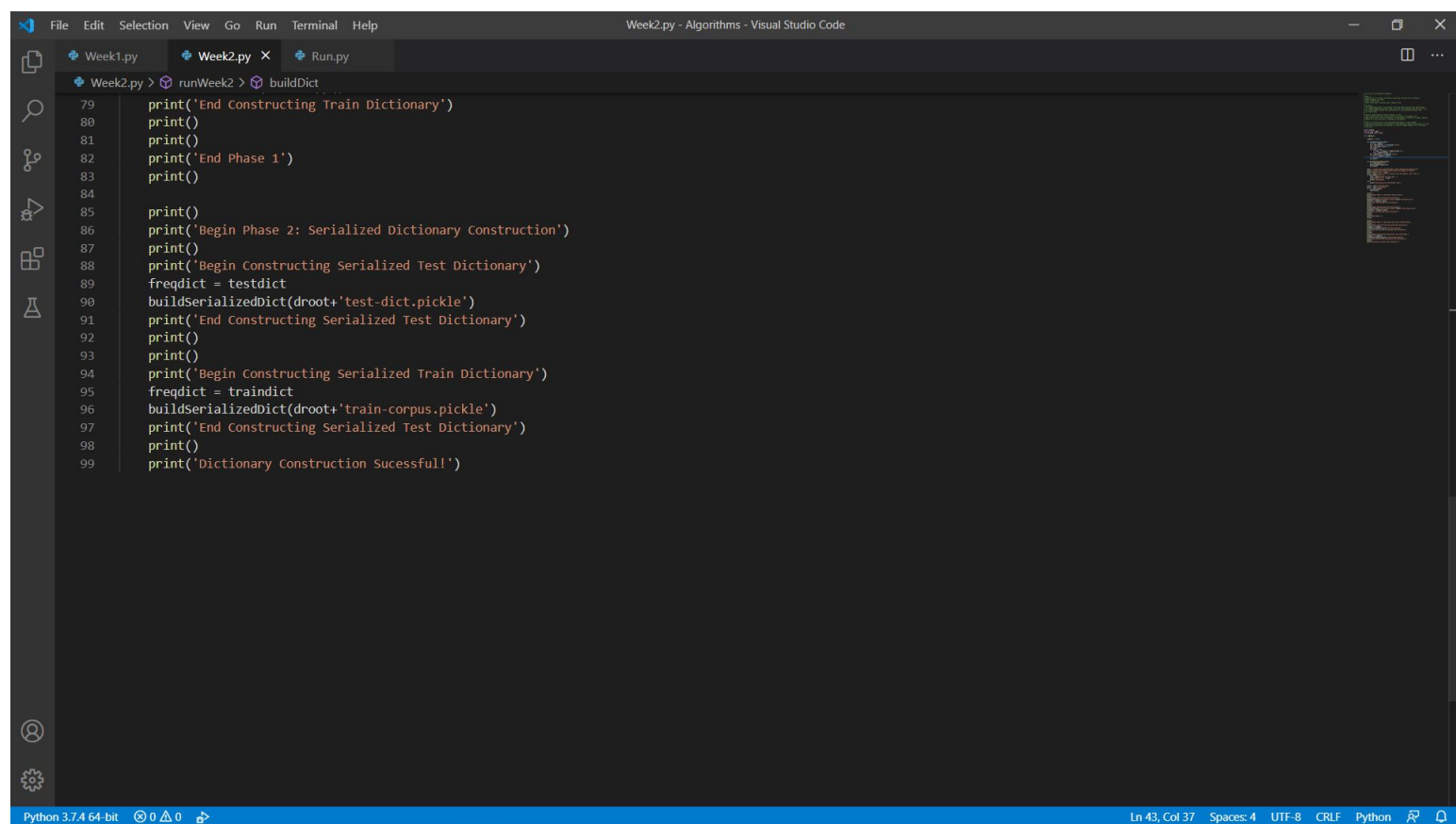
Figure 17: Week-2 File: Section 1



```
File Edit Selection View Go Run Terminal Help
Week2.py - Algorithms - Visual Studio Code

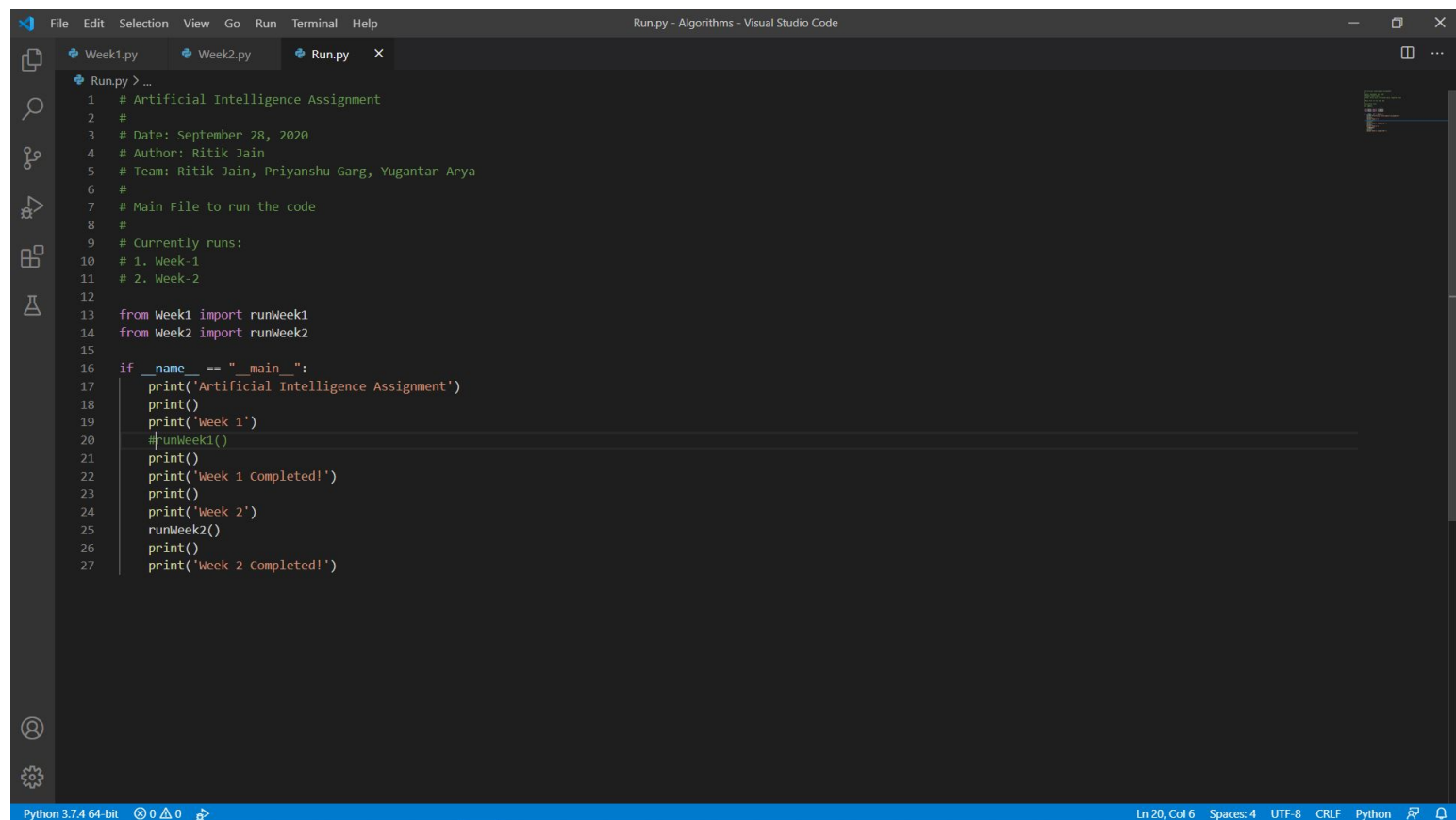
Week1.py Week2.py x Run.py
Week2.py > runWeek2 > buildDict
40             else: freqdict[item] = 1
41             fp = open(dest,"w",encoding='utf-8')
42             for k,v in freqdict.items():
43                 fp.write(k+':'+str(v)+';')
44             fp.close()
45
46     def buildSerializedDict(dest):
47         fp = open(dest,"wb")
48         pickle.dump(freqdict,fp)
49         fp.close()
50
51     root = 'C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/'
52     print('Frequency Dictionary Construction Tool (Week 2) v1.0\n')
53     print('Default Root: '+root)
54     wish = input('Do you wish to continue with the default root? (y/N):')
55     if wish.lower() == 'n':
56         root = input('Enter the new root: ')
57         print('Root set to: '+root)
58         print('Continuing')
59     else:
60         print('Continuing with the default root')
61
62     sroot = root+'collected_data/'
63     droot = root+'freq_dicts/'
64     if not isdir(droot):
65         mkdir(droot)
66
67     print()
68     print('Begin Phase 1: Dictionary Construction')
69     print()
70     print('Begin Constructing Test Dictionary')
71     buildDict(sroot+'test-corpus.collect',droot+'test-dict.dict')
72     testdict = freqdict.copy()
73     print('End Constructing Test Dictionary')
74     print()
75     print()
76     print('Begin Constructing Train Dictionary')
77     buildDict(sroot+'train-corpus.collect',droot+'train-corpus.dict')
78     traindict = freqdict.copy()
79     print('End Constructing Train Dictionary')
```

Figure 18: Week-2 File: Section 2



```
79     print('End Constructing Train Dictionary')
80     print()
81     print()
82     print('End Phase 1')
83     print()
84
85     print()
86     print('Begin Phase 2: Serialized Dictionary Construction')
87     print()
88     print('Begin Constructing Serialized Test Dictionary')
89     freqdict = testdict
90     buildSerializedDict(droot+'test-dict.pickle')
91     print('End Constructing Serialized Test Dictionary')
92     print()
93     print()
94     print('Begin Constructing Serialized Train Dictionary')
95     freqdict = traindict
96     buildSerializedDict(droot+'train-corpus.pickle')
97     print('End Constructing Serialized Test Dictionary')
98     print()
99     print('Dictionary Construction Sucessfull')
```

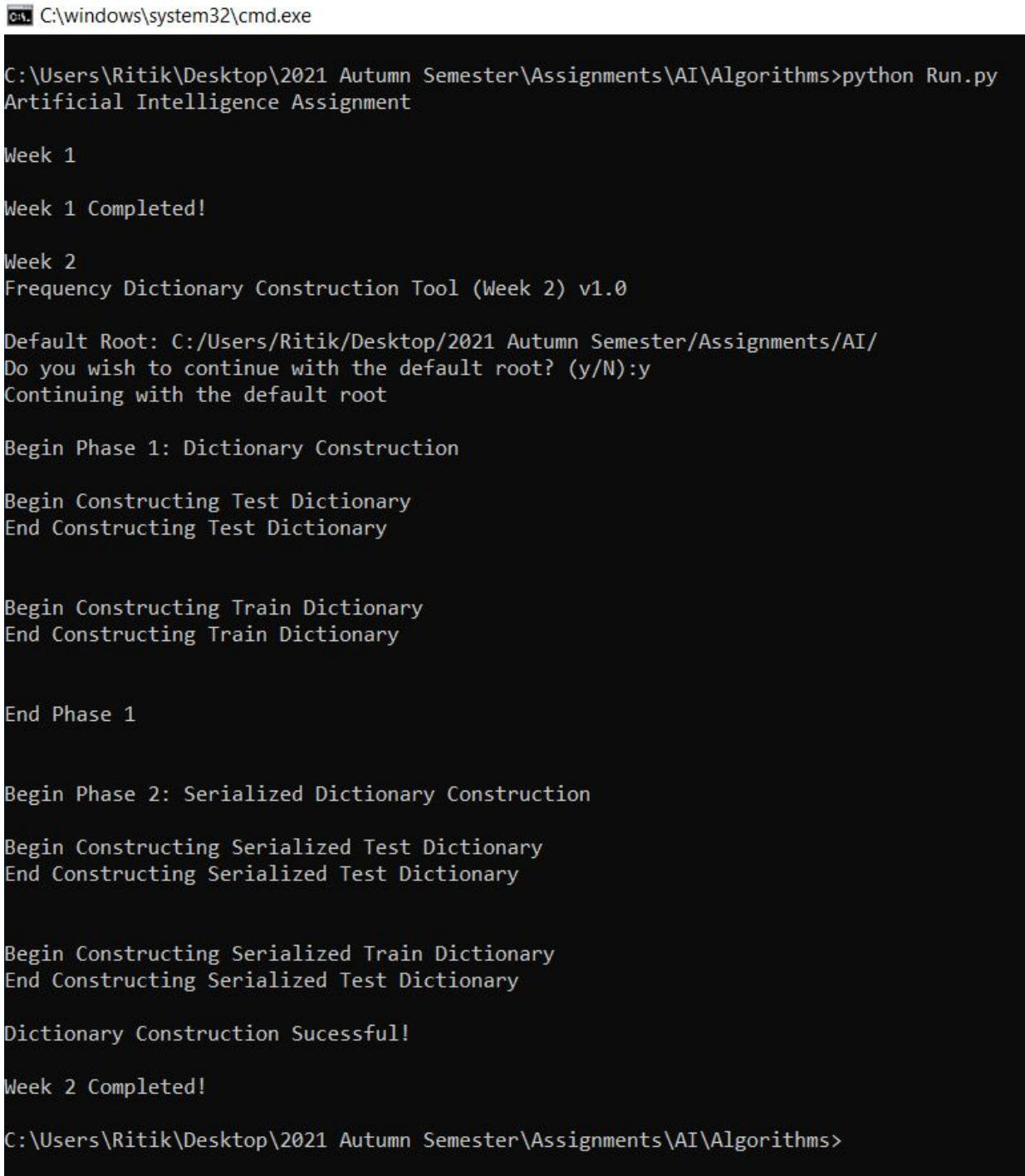
Figure 19: Week-2 File: Section 3



```
1  # Artificial Intelligence Assignment
2  #
3  # Date: September 28, 2020
4  # Author: Ritik Jain
5  # Team: Ritik Jain, Priyanshu Garg, Yugantar Arya
6  #
7  # Main File to run the code
8  #
9  # Currently runs:
10 # 1. Week-1
11 # 2. Week-2
12
13 from Week1 import runWeek1
14 from Week2 import runWeek2
15
16 if __name__ == "__main__":
17     print('Artificial Intelligence Assignment')
18     print()
19     print('Week 1')
20     #runWeek1()
21     print()
22     print('Week 1 completed!')
23     print()
24     print('Week 2')
25     runWeek2()
26     print()
27     print('Week 2 completed!')
```

Figure 20: Main File: Run.py

2.4 Execution



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

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

Week 1

Week 1 Completed!

Week 2
Frequency Dictionary Construction Tool (Week 2) 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: Dictionary Construction

Begin Constructing Test Dictionary
End Constructing Test Dictionary

Begin Constructing Train Dictionary
End Constructing Train Dictionary

End Phase 1

Begin Phase 2: Serialized Dictionary Construction

Begin Constructing Serialized Test Dictionary
End Constructing Serialized Test Dictionary

Begin Constructing Serialized Train Dictionary
End Constructing Serialized Test Dictionary

Dictionary Construction Sucessful!

Week 2 Completed!

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

Figure 21: Execution: Run.py

2.5 Output Files

2021 Autumn Semester > Assignments > AI > freq_dicts

Name	Date modified	Type	Size
test-dict	10/2/2020 5:39 PM	DICT File	1,801 KB
test-dict.pickle	10/2/2020 5:39 PM	PICKLE File	2,765 KB
train-corpus	10/2/2020 5:39 PM	DICT File	4,059 KB
train-corpus.pickle	10/2/2020 5:39 PM	PICKLE File	6,187 KB

Figure 22: Generated Dictionaries

```
test-dict - Notepad
File Edit Format View Help
[THE_ART:672;PLAYERS_SUBST:2;Virtually_ADV:16;every_ART:1559;country_SUBST:1052;in_PREP:64628;the_ART:212492;world_SUBST:1894;has_VERB:8132;PRON:5995;share_SUBST:169;overall_ADJ:147;budget_SUBST:107;process_SUBST:682;exaggerates_VERB:2;distorts_VERB:3;reports_SUBST:199;so_ADV:61;SUBST:216;there_ADV:1886;nevertheless_ADV:110;remains_VERB:333;feeling_SUBST:267;work_SUBST:2344;rather_ADV:1670;exciting_ADJ:133;possibly_ADJ:123;under_PREP:1837;government_SUBST:1065;department_SUBST:125;name_SUBST:967;sound_VERB:89;loyalty_SUBST:91;motivation_SUBST:62;simply_ADV:777;run_VERB:639;huge_ADJ:409;banks_SUBST:156;computers_SUBST:43;numbers_SUBST:374;foreign-speaking_ADJ:1;translators_SUBST:4;thou:8;Many_ADJ:570;these_ADJ:3977;subsequently_ADV:108;agreed_VERB:376;work_VERB:854;alternative_SUBST:131;be_VERB:24643;hanged_VERB:18;so_CON:56_ADJ:56;But_CONJ:4325;probably_ADV:874;does_VERB:1912;matter_VERB:119;much_ADJ:1980;anyway_ADV:255;As_CONJ:1645;progressed_VERB:40;tended_orth_PREP:371;looking_VERB:868;detail_SUBST:229;1_ADJ:949;your_PRON:6097;appointment_SUBST:101;Service_VERB:1;you_PRON:18434;will_VERB:7783;BST:318;particular_ADJ:989;community_SUBST:511;matter_SUBST:725;than_CONJ:5654;5_ADJ:535;No_ART:598;inquiry_SUBST:148;behalf_SUBST:118;unle:_SUBST:775;First_ADJ:368;though_CONJ:1284;notionally_ADV:2;he_PRON:17058;told_VERB:1114;day-to-day_ADJ:28;Second_ADJ:219;alone_ADV:313;detei_m_SUBST:13;Philby_SUBST:9;escaped_VERB:66;Moscow_SUBST:66;just_ADV:3440;ahead_ADV:296;1962_ADJ:72;Anthony_SUBST:50;Blunt_SUBST:2;finally_ADJ:227;West_SUBST:548;William_SUBST:310;Marshall_SUBST:69;had_VERB:17610;cipher_SUBST:6;clerk_SUBST:40;Embassy_SUBST:47;Then_ADV:916;famous:150;Roger_SUBST:74;Hollis_SUBST:13;mole_SUBST:9;therefore_ADV:835;investigation_SUBST:145;frustrated_ADJ:27;reasons_SUBST:595;evidence_SUB:J:487;believes_VERB:105;enemies_SUBST:75;state_SUBST:833;able_ADJ:1284;decades_SUBST:123;allowing_VERB:154;politicised_ADJ:4;maintains_VERB:BST:3;keep_VERB:927;lot_SUBST:685;innocent_ADJ:69;Here_ADV:464;straightforward_ADJ:80;Special_ADJ:83;Branch_SUBST:42;closely_ADV:201;keeps_'tions_SUBST:78;minister_SUBST:77;authorise_VERB:7;warrant_SUBST:70;tapping_VERB:23;illegally_ADV:25;All_ADJ:892;breach_SUBST:94;representat:64;point_SUBST:1264;noteworthy_ADJ:11;decides_VERB:31;need_VERB:1375;words_SUBST:661;know_VERB:2649;right_ADJ:845;ask_VERB:631;allegation_SI;before_CONJ:994;premises_SUBST:79;burgled_VERB:5;book_SUBST:872;high-sounding_ADJ:1;presented_VERB:321;Parliament_SUBST:147;sham_SUBST:4;r:2;anti-Nazi_ADJ:1;received_VERB:461;Oslo_SUBST:5;Report_SUBST:174;package_SUBST:90;technical_ADJ:219;delivered_VERB:91;Legation_SUBST:3;coi:ew_ADJ:1965;results_SUBST:485;largely_ADV:301;place_SUBST:1733;occupied_VERB:149;upset_ADJ:75;Churchill_SUBST:65;decision_SUBST:508;create_'culated_VERB:19;offices_SUBST:147;assume_VERB:149;managed_VERB:309;copy_SUBST:217;over_ADV:43;600_ADJ:52;head_SUBST:1233;C_SUBST:178;Christ:_VERB:649;warn_VERB:30;Argentinian_ADJ:6;invasion_SUBST:80;Falkland_SUBST:4;Islands_SUBST:30;1982_ADJ:288;either_ADV:875;way_SUBST:3737;prei:human_ADJ:563;Humint_SUBST:1;experience_SUBST:649;suggests_VERB:214;unable_ADJ:280;remotely_ADV:10;justifies_VERB:6;annual_ADJ:184;located_'J:2;Telegraph_SUBST:15;Company_SUBST:145;Western_ADJ:199;Union_SUBST:188;acquiesce_VERB:2;tacit_ADJ:14;threat_SUBST:211;operating_SUBST:43;UBST:524;places_SUBST:429;Malta_SUBST:13;acquire_VERB:90;charade_SUBST:7;obtaining_VERB:42;intercepted_VERB:11;Eventually_ADV:88;Singapore_'2;reconditioned_VERB:6;Enigmas_SUBST:2;extremely_ADV:347;cheap_ADJ:138;prices_SUBST:163;urging_VERB:18;adopt_VERB:105;ease_SUBST:92;north_SIT:643;analysed_VERB:39;1986_ADJ:362;leader_SUBST:132;Labour_ADJ:216;Party_SUBST:237;Neil_SUBST:37;Kinnock_SUBST:5;telephoned_VERB:13;Malcoli:eting_VERB:36;call_SUBST:154;full_ADJ:989;recording_SUBST:43;conversation_SUBST:194;tap_VERB:27;presence_SUBST:353;admitted_VERB:155;claim_'ions_SUBST:8;Monitoring_SUBST:3;4-9_UNC:2;St_SUBST:744;Dunstons_SUBST:1;City_SUBST:253;upper_ADJ:228;floors_SUBST:75;Empress_SUBST:131;Buil:ussia_SUBST:162;China_SUBST:111;Over_PREP:117;period_SUBST:936;U-2_SUBST:1;Spy_SUBST:2;aircraft_SUBST:133;followed_VERB:690;incredibly_ADV:T:58;Both_ADV:105;since_PREP:457;1880s_ADJ:35;1910_ADJ:31;collating_VERB:5;when_CONJ:1701;proper_ADJ:227;haphazard_ADJ:19;fashion_SUBST:167ks_SUBST:41;creating_VERB:129;moment_SUBST:754;onwards_ADV:68;activity_SUBST:400;concentrated_VERB:75;investment_SUBST:184;military_ADJ:3166;aid_SUBST:181;Plan_SUBST:148;Bribing_VERB:1;corrupt_ADJ:15;propping_VERB:5;rotten_ADJ:22;enhanced_VERB:37;poorer_ADJ:43;nations_SUBST:124imony_SUBST:16;shredding_SUBST:1;incriminating_ADJ:8;documents_SUBST:109;shipment_SUBST:7;arms_SUBST:396;opening_SUBST:137;numbered_ADJ:10;ironically_ADV:35;Marine_ADJ:16;corps_SUBST:13;alerts_VERB:2;advisers_SUBST:20;heroes_SUBST:54;anxious_ADJ:118;save_VERB:229;Perhaps_ADV:402SUBST:44;strategically_ADV:11;eavesdrop_VERB:2;allows_VERB:113;pay_VERB:438;construction_SUBST:189;running_SUBST:107;costs_SUBST:287;saving:ovements_SUBST:228;especially_ADV:757;demonstrating_VERB:40;involvement_SUBST:162;Vietnam_SUBST:35;Included_VERB:4;well-known_ADJ:75;Jane_SI
```

Figure 23: A section of test dictionary

2.6 Results

The required dictionaries were successfully generated, with the same accuracy as that of week 1. The next step is to analyze the distribution of words and tags, which would be done in the next week.