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 processess 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:

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

```python
# Artificial Intelligence Assignment
#
# Week: 1
# Program Aim: To filter the text by extracting word and pos tag for the word
# Date: September 28, 2020
# Author: Ritik Jain
# Team: Ritik Jain, Priyanshu Garg, Yugantar Arya
#
# 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 preprocess all the files and it has to be done only once; hence I prefered clarity
# of code over the small extra time it takes
#
# Phase 1 Filtration:
# The program processess all the raw xml files and converts them into filtered files
# The program makes use of the regex:
# {<\\s*w[^>]*pos="([A-Za-z]*)"[^>]*>\\s*([^<\\s]*)\\s*<\\s*/w\\s*>}, which is the heart of the program
# 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.
#
# 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
#
# Phase 3 Verification:
# The verification phase verifies that the filtered data matches the raw data.
# This is accomplished by rescanning the raw data with a simpler regex {<\\s*/\\s*w\\s*>} (only for </w> tags)
# and maintaing the total count of word,tag combinations for each file.
# The filtered files are also rescanned, but the regex used is {;}
# After the scan is complete, the verification ratio and the accuracy are displayed.
# If the accuracy > 95 %, the verification could be assumed to be successful.
#
# After the successful completion of all the phases, we obtain the processed data

from os import listdir
from re import findall
from os.path import isfile
```

Figure 1: Week-1 File: Section 1

```python
from os.path import isfile
from os.path import isdir
from os import mkdir
from sys import stdout

def runWeek1():

    def filterFile(source,dest,pat = '<\\s*w[^>]*pos="([A-Za-z]*)"[^>]*>\\s*([^<\\s]*)\\s*<\\s*/w\\s*>',en="utf-8"):
        nonlocal stats
        fp = open(source,"r",encoding=en)
        text = str().join(fp.read().splitlines())
        matches = findall(pat,text)
        fp.close()
        fp = open(dest,"w",encoding=en)
        for match in matches:
            fp.write(match[1]+'_'+match[0]+';')
        fp.close()
        stats += 1
        stdout.write(f'\rFiltered {stats} files')

    def filterRoot(sroot,droot):
        nonlocal stats
        if isfile(sroot):
            filterFile(sroot,droot+'.filtered')
        else:
            if not isdir(droot):
                mkdir(droot)
            for file in listdir(sroot):
                filterRoot(sroot+'/'+file,droot+'/'+file)

    def collect(root,en="utf-8"):
        nonlocal collector,stats
        if isfile(root):
            fp = open(root,"r",encoding=en)
            text = fp.read()
            fp.close()
            collector.write(text)
            stats += 1
            stdout.write(f'\rCollected {stats} files')
        else:
```

Figure 2: Week-1 File: Section 2

```python
        else:
            for file in listdir(root):
                collect(root+'/'+file)

    def verify(root):
        def count(file,en,pat,out):
            with open(file,'r',encoding=en) as f:
                L = len(findall(pat,str().join(f.read().splitlines())))
                out.write(f'{file}${L}\n')
                return L
            return 0
        def explore(root,en,pat,out):
            if isfile(root):
                return count(root,en,pat,out)
            else:
                total = 0
                for file in listdir(root):
                    total += explore(root+'\\'+file,en,pat,out)
                return total

        if(not isdir(root+'verification_records')):
            mkdir(root+'verification_records')

        with open(root+'verification_records/raw.verify','w') as out:
            xcount1 = explore(root+'raw_data','utf-8','<\\s*/\\s*w\\s*>',out)
            out.write(f'$TOTAL$ = {xcount1}')

        with open(root+'verification_records/filtered.verify','w') as out:
            xcount2 = explore(root+'filtered_data','utf-8',';',out)
            out.write(f'$TOTAL$ = {xcount2}')

        ratio = xcount2/xcount1
        acc = (1-abs(ratio-1))*100
        print(f'Verification Ratio: {round(ratio,4)}')

        print(f'Estimated Accuracy: {round(acc,2)}%')

        with open(root+'verification_records/statistics.txt','w') as out:
            out.write(f'Estimated Raw Tags:{xcount1}\nProcessed Raw Tags:{xcount2}\nVerification Ratio:{ratio}\nAccuracy:{acc}\n')
        return ratio
```

Figure 3: Week-1 File: Section 3

```python
        return ratio

    print('Text Filtration Tool (Week 1) v1.0\n')
    root = 'C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/'
    print('Default Root: '+root)
    wish = input('Do you wish to continue with the default root? (y/N):')
    if wish.lower() == 'n':
        root = input('Enter the new root: ')
        print('Root set to: '+root)
        print('Continuing')
    else:
        print('Continuing with the default root')
    print()
    print('Begin Phase 1: Filtration')
    stats = 0
    filterRoot(sroot=root+'raw_data',droot=root+'filtered_data')
    print()
    print('End Phase 1')
    print()

    print('Begin Phase 2: Collection')
    test = root+'filtered_data/Test-corpus'
    train = root+'filtered_data/Train-corpus'

    print('Begin Phase 2.1: Test Collection')
    if not isdir(root+'collected_data'):
        mkdir(root+'collected_data')
    collector = open(root+'collected_data/Test-corpus.collect',"w",encoding="utf-8")
    stats = 0
    collect(test)
    collector.close()
    print()
    print('End Phase 2.1')

    print('Begin Phase 2.2: Train Collection')
    collector = open(root+'collected_data/Train-corpus.collect',"w",encoding="utf-8")
    stats = 0
    collect(train)
    collector.close()
    print()
```

Figure 4: Week-1 File: Section 4

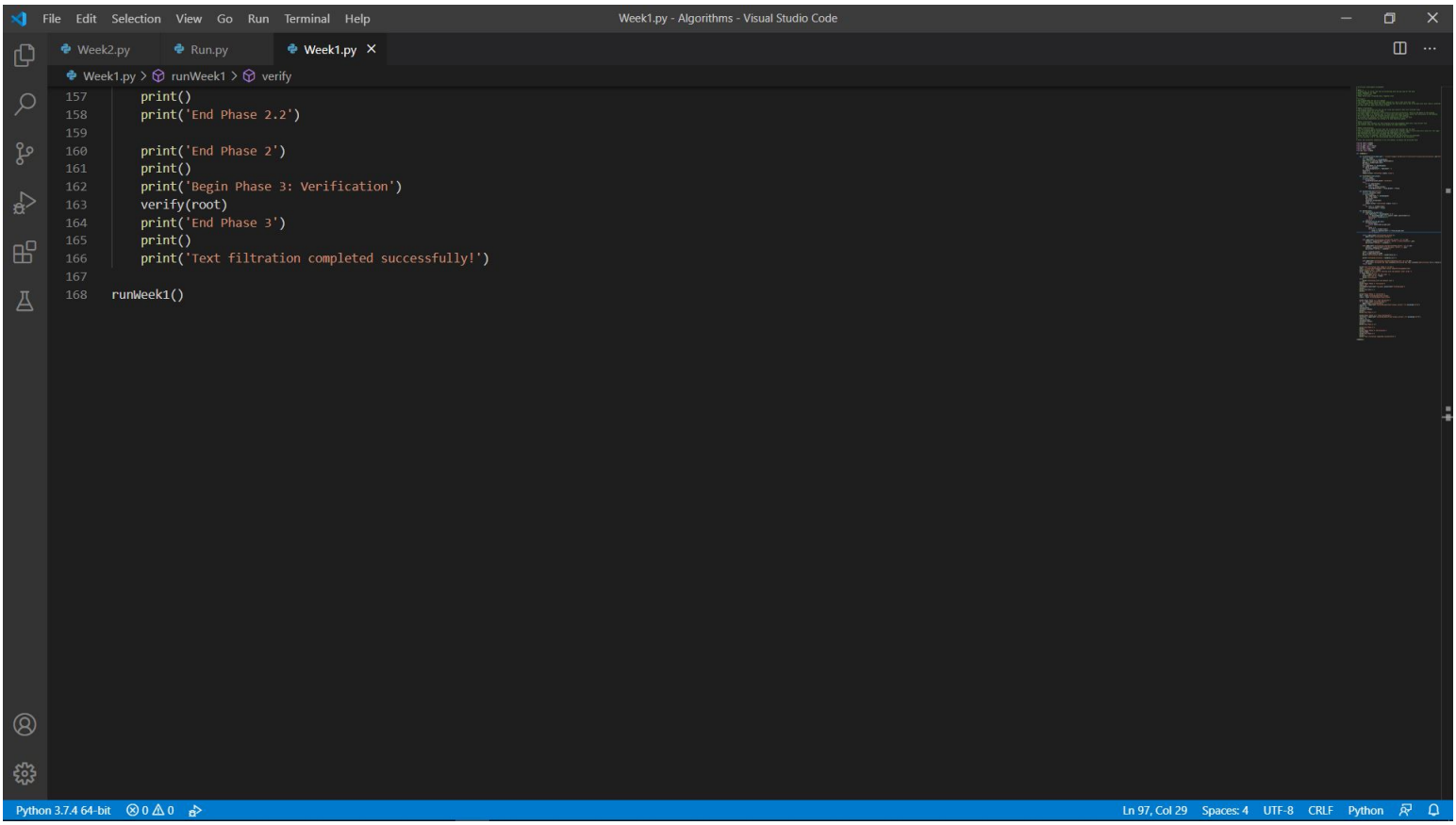


Figure 5: Week-1 File: Section 5

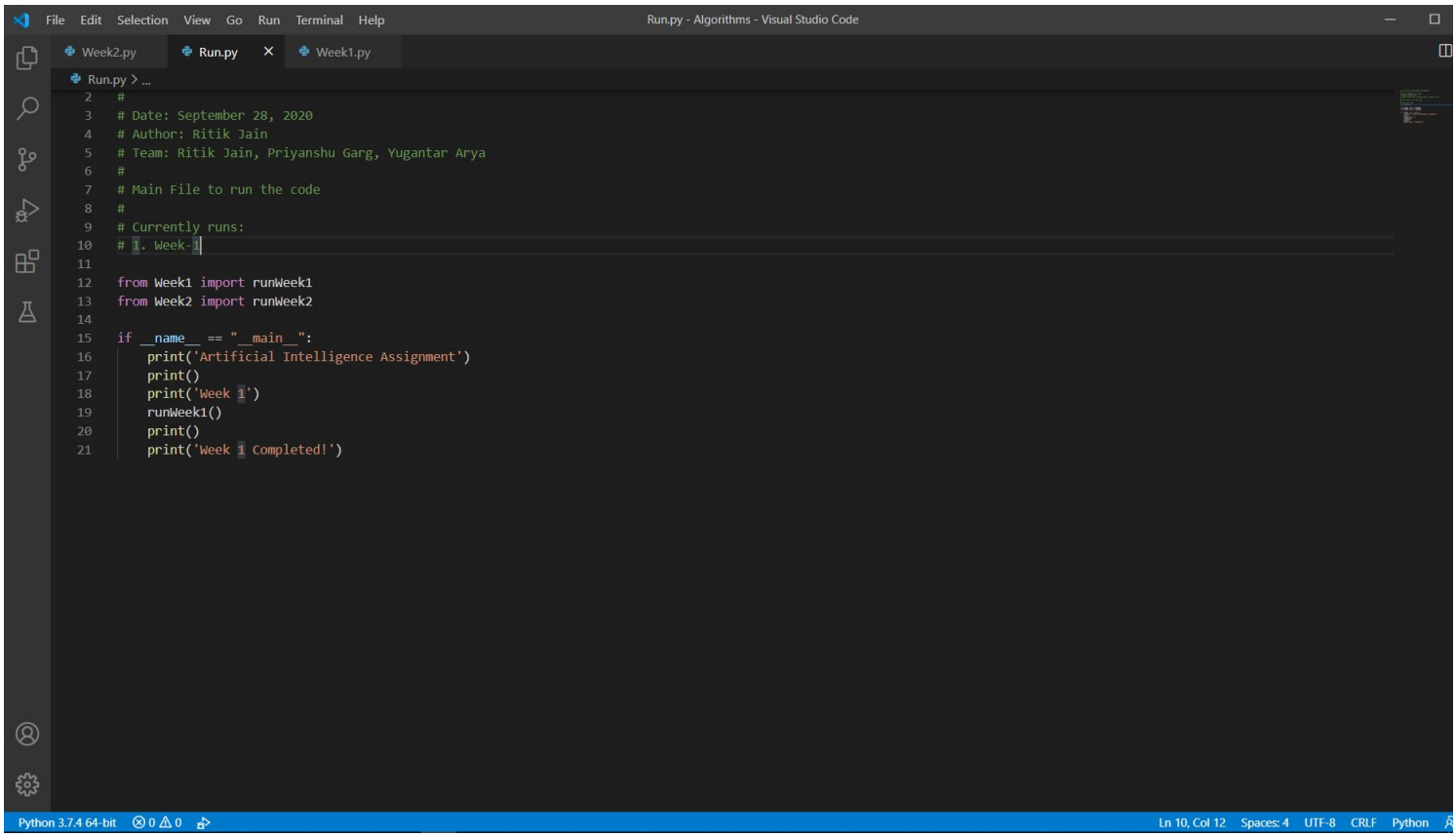


Figure 6: Main File: Run.py

1.4 Execution

C:\windows\system32\cmd.exe

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

Begin Phase 3: Verification
Verification Ratio: 1.0
Estimated Accuracy: 100.0%
End Phase 3

Text filtration completed successfully!

Week 1 Completed!
```

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;sc
RT;process_SUBST;exaggerates_VERB;and_CONJ;distorts_VERB;its_PRON;reports_SUBST;so_ADV;as_CONJ;to_P
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;advertisment_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
nthony_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:12744946
Verification Ratio:0.9999998430750757
Accuracy:99.99998430750757
```

Figure 16: Statistics File

1.6 Results

The raw data was processed with an estimated accuracy of more than 99.99%. We could have made the data more accurate (to exactly 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 $1.6 \times 10^{-5}\,\%$ 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.

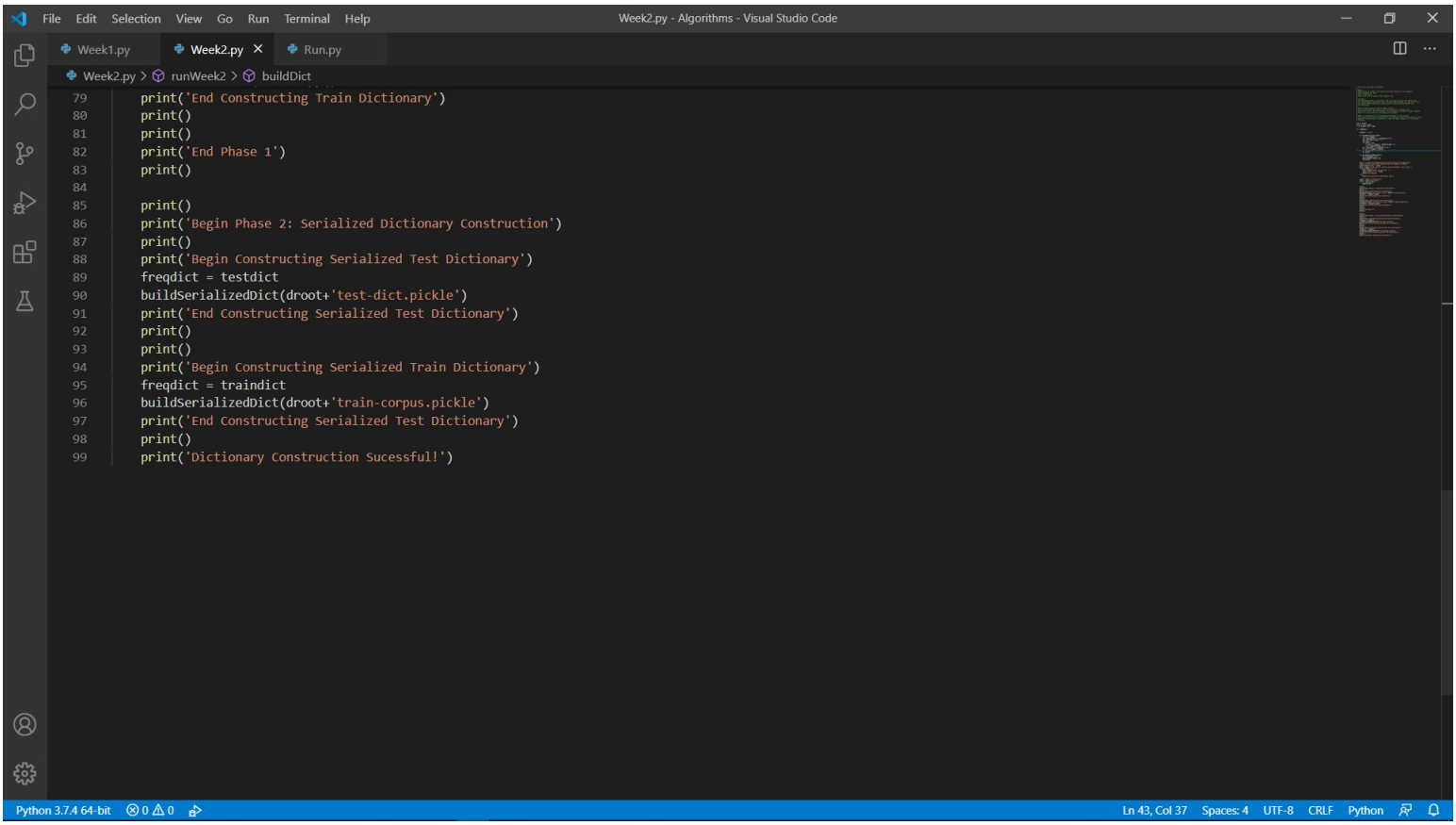
2.3 Code Screenshots

```python
# Artificial Intelligence Assignment
#
# Week: 2
# Program Aim: To create a dictionary which maps word,tag to its frequency
# Date: September 28, 2020
# Author: Ritik Jain
# Team: Ritik Jain, Priyanshu Garg, Yugantar Arya
#
# 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.
#
# Phase 1: 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.
#
# Phase 2: 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.

import pickle
from os import mkdir
from os.path import isdir

def runWeek2():

    freqdict = dict()

    def buildDict(collect,dest):
        nonlocal freqdict
        fp = open(collect,"r",encoding='utf-8')
        raw = fp.read().split(';')
        fp.close()
        for item in raw:
            if item in freqdict: freqdict[item] += 1
            else: freqdict[item] = 1
```

Figure 17: Week-2 File: Section 1

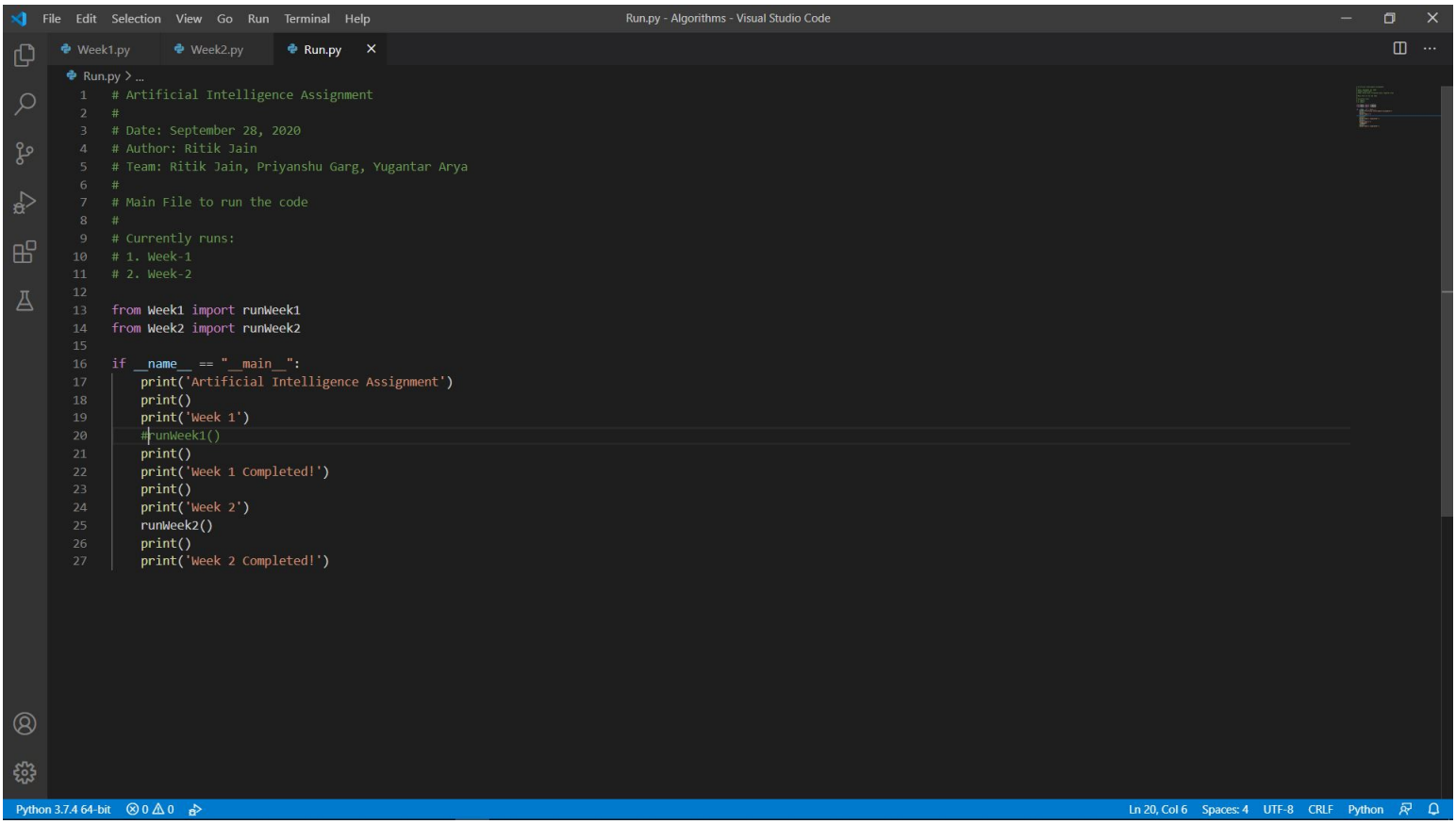
```python
            else: freqdict[item] = 1
        fp = open(dest,"w",encoding='utf-8')
        for k,v in freqdict.items():
            fp.write(k+':'+str(v)+';')
        fp.close()

    def buildSerializedDict(dest):
        fp = open(dest,"wb")
        pickle.dump(freqdict,fp)
        fp.close()

    root = 'C:/Users/Ritik/Desktop/2021 Autumn Semester/Assignments/AI/'
    print('Frequency Dictionary Construction Tool (Week 2) v1.0\n')
    print('Default Root: '+root)
    wish = input('Do you wish to continue with the default root? (y/N):')
    if wish.lower() == 'n':
        root = input('Enter the new root: ')
        print('Root set to: '+root)
        print('Continuing')
    else:
        print('Continuing with the default root')

    sroot = root+'collected_data/'
    droot = root+'freq_dicts/'
    if not isdir(droot):
        mkdir(droot)

    print()
    print('Begin Phase 1: Dictionary Construction')
    print()
    print('Begin Constructing Test Dictionary')
    buildDict(sroot+'test-corpus.collect',droot+'test-dict.dict')
    testdict = freqdict.copy()
    print('End Constructing Test Dictionary')
    print()
    print()
    print('Begin Constructing Train Dictionary')
    buildDict(sroot+'train-corpus.collect',droot+'train-corpus.dict')
    traindict = freqdict.copy()
    print('End Constructing Train Dictionary')
```

Figure 18: Week-2 File: Section 2



```python
    print('End Constructing Train Dictionary')
    print()
    print()
    print('End Phase 1')
    print()

    print()
    print('Begin Phase 2: Serialized Dictionary Construction')
    print()
    print('Begin Constructing Serialized Test Dictionary')
    freqdict = testdict
    buildSerializedDict(droot+'test-dict.pickle')
    print('End Constructing Serialized Test Dictionary')
    print()
    print()
    print('Begin Constructing Serialized Train Dictionary')
    freqdict = traindict
    buildSerializedDict(droot+'train-corpus.pickle')
    print('End Constructing Serialized Test Dictionary')
    print()
    print('Dictionary Construction Sucessful!')
```

Figure 19: Week-2 File: Section 3



```python
# Artificial Intelligence Assignment
#
# Date: September 28, 2020
# Author: Ritik Jain
# Team: Ritik Jain, Priyanshu Garg, Yugantar Arya
#
# Main File to run the code
#
# Currently runs:
# 1. Week-1
# 2. Week-2

from Week1 import runWeek1
from Week2 import runWeek2

if __name__ == "__main__":
    print('Artificial Intelligence Assignment')
    print()
    print('Week 1')
    #runWeek1()
    print()
    print('Week 1 Completed!')
    print()
    print('Week 2')
    runWeek2()
    print()
    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	10/3/2020 1:04 PM	DICT File	1,910 KB
test-dict.pickle	10/3/2020 1:04 PM	PICKLE File	2,765 KB
train-dict	10/3/2020 1:04 PM	DICT File	4,302 KB
train-dict.pickle	10/3/2020 1:04 PM	PICKLE File	6,188 KB

Figure 22: Generated Dictionaries

train-dict - Notepad

File Edit Format View Help

```
THE_ART:3048
PLAYERS_SUBST:5
Virtually_ADV:32
every_ART:4907
country_SUBST:4727
in_PREP:231489
the_ART:737910
world_SUBST:7643
has_VERB:37189
some_ADJ:18670
sort_SUBST:2137
of_PREP:405289
intelligence_SUBST:568
agency_SUBST:532
Most_ADJ:1631
have_VERB:55750
at_PREP:63635
least_ADV:4149
two_ADJ:15916
one_ADJ:20853
to_PREP:337915
look_VERB:4012
after_PREP:10520
security_SUBST:1501
within_PREP:4485
usually_ADV:2402
called_VERB:4337
```

Figure 23: A section of train 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.