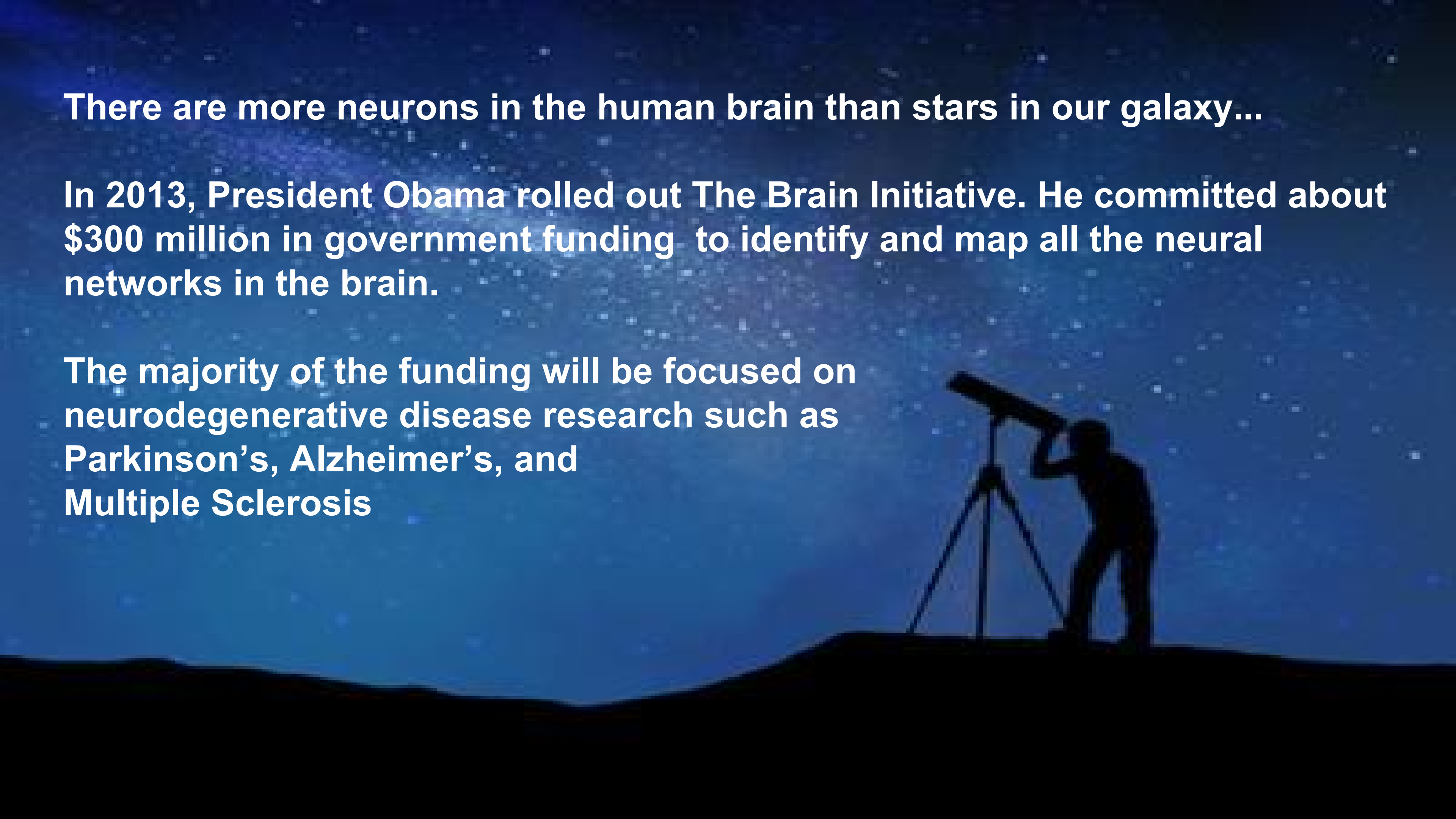
A Biomedical Application for the Natural Language Processing Tool in Python

Final Project, Hackbright Academy 2015
Abla Harara



There are more neurons in the human brain than stars in our galaxy...

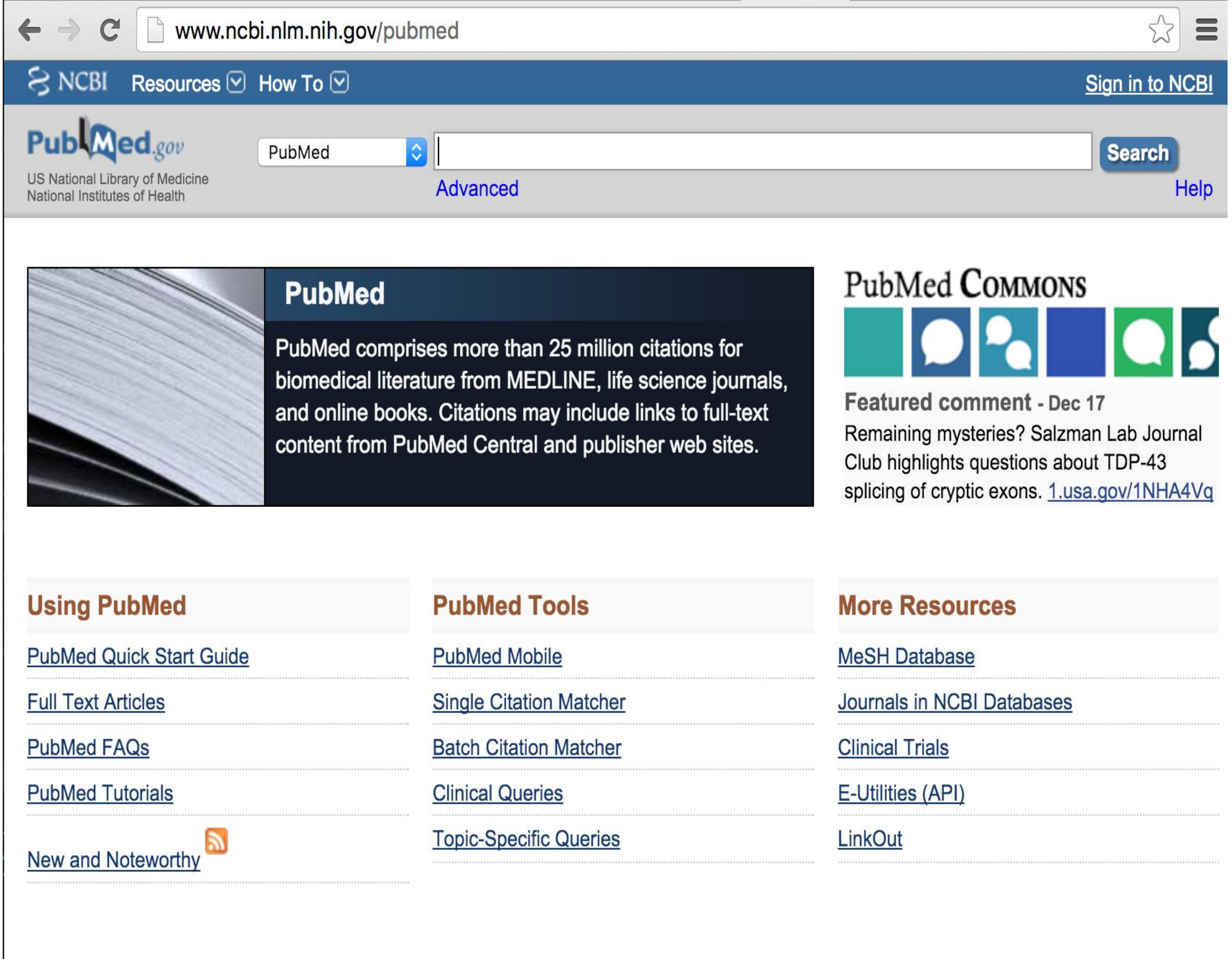
In 2013, President Obama rolled out The Brain Initiative. He committed about $300 million in government funding to identify and map all the neural networks in the brain.

The majority of the funding will be focused on neurodegenerative disease research such as Parkinson’s, Alzheimer’s, and
Multiple Sclerosis

The Pubmed Database houses the largest collection of research articles in the world.

The Life Sciences and Healthcare Industries are knowledge driven, so they rely on Pubmed to learn about what's already been discovered, and what has yet to be elucidated

The PubMed search engine only sifts through the Article Title, Article Abstract, and the Journal Source to identify relevant material for the scientist to review ...


www.ncbi.nlm.nih.gov/pubmed
NCBI
Resources
How To
Sign in to NCBI
PubMed.gov
US National Library of Medicine
National Institutes of Health
PubMed
Search
Advanced
Help

PubMed

PubMed comprises more than 25 million citations for biomedical literature from MEDLINE, life science journals, and online books. Citations may include links to full-text content from PubMed Central and publisher web sites.

PubMed Commons

Featured comment - Dec 17

Remaining mysteries? Salzman Lab Journal Club highlights questions about TDP-43 splicing of cryptic exons. 1.usa.gov/1NHA4Vq

Using PubMed

- PubMed Quick Start Guide
- Full Text Articles
- PubMed FAQs
- PubMed Tutorials
- New and Noteworthy

PubMed Tools

- PubMed Mobile
- Single Citation Matcher
- Batch Citation Matcher
- Clinical Queries
- Topic-Specific Queries

More Resources

- MeSH Database
- Journals in NCBI Databases
- Clinical Trials
- E-Utilities (API)
- LinkOut

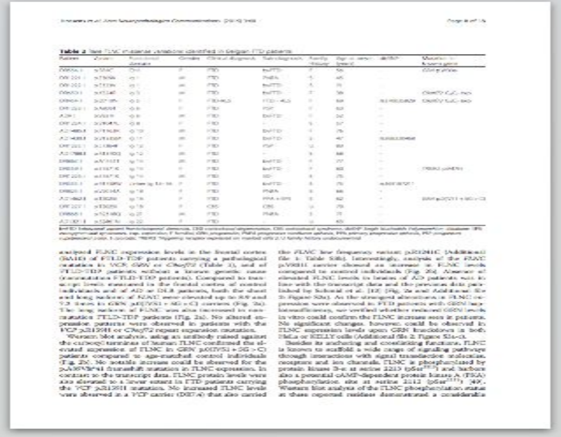
THIS IS PROBLEMATIC...

Depending on the author of the article, Titles and Abstracts will omit critical information about the findings published.

This critical info is stored in other sections of the article that are not currently searchable via the PubMed website.

Those sections are known as the methods, the materials, the results and the conclusion.

7



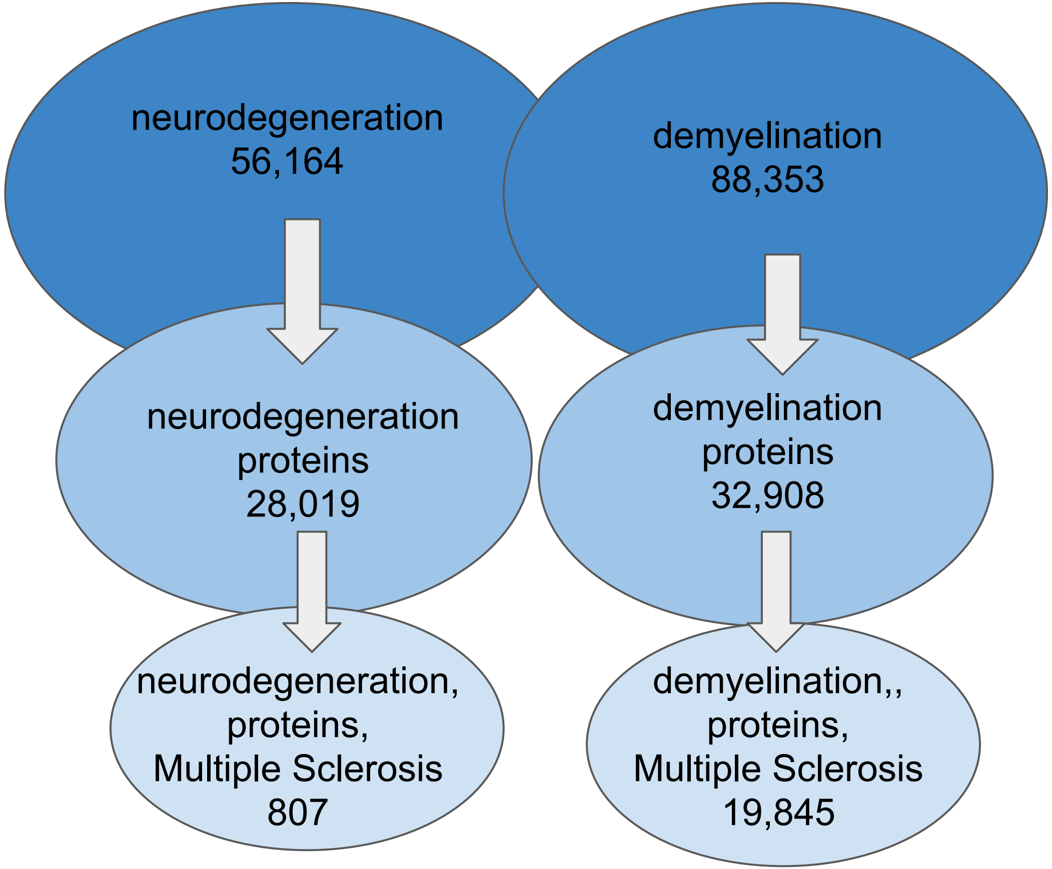
8

In addition...

The shear number of articles is vast. Even when very specific keywords are used, and the author publishes the findings in the abstract, there are still large amounts of articles to be read.

For example; its very hard to identify exactly which proteins are associated with neurodegeneration and MS without reviewing 807 articles

PubMed Search Results



Using Python to Find a Solution:

I wanted to build a tool using Python where anyone can take a set of articles and perform a deep search.

We know there are about 50 identified neuronal proteins in the human body. The program is designed to “read” the articles in PubMed. And if the word “demyelination” is published with any one of the 50 identified neuronal proteins, the program will aggregate these findings.

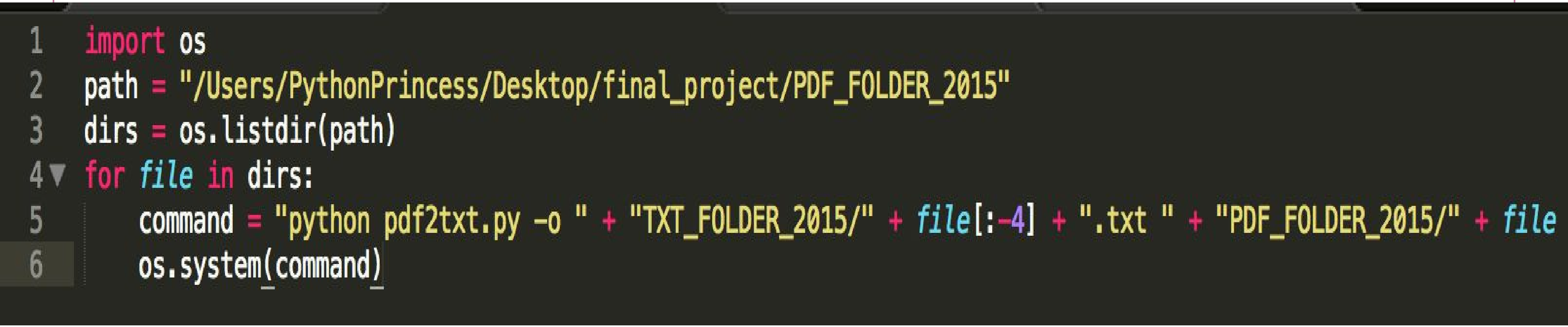
Using the associative property we can then infer: if that specific protein is published in the literature with demyelination a lot, that protein is probably highly associated with demyelination. And if that protein is rarely mentioned with demyelination in the literature, then its mostly likely not associated.

Step 1:

Retrieve the Articles from PubMed and Convert them to TXT files:

PubMed stores their articles in PDF format. For Python to “read” these articles, we must first convert all the PubMed PDF articles into TXT files.

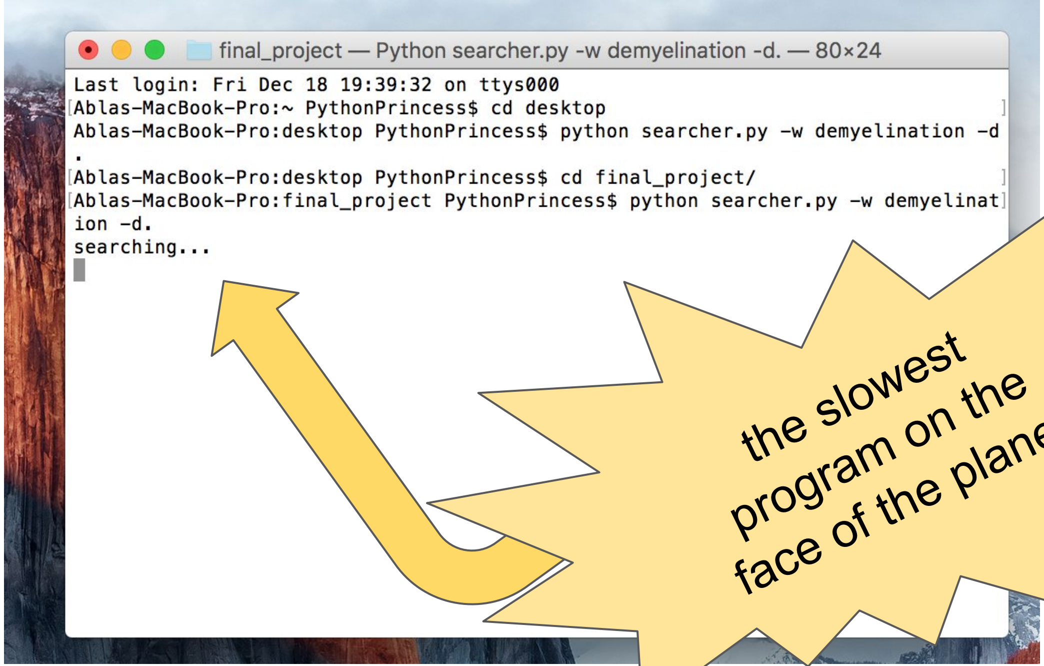
** IT WORKED: the code below converted 50 PDF articles in 50 TXT files



```python
import os
path = "/Users/PythonPrincess/Desktop/final_project/PDF_FOLDER_2015"
dirs = os.listdir(path)
for file in dirs:
    command = "python pdf2txt.py -o " + "TXT_FOLDER_2015/" + file[:-4] + ".txt " + "PDF_FOLDER_2015/" + file
    os.system(command)
```

Step 2:

write a “TXT-file-searcher” program that scours files for a keywords like “demyelination”


final_project — Python searcher.py -w demyelination -d. — 80×24

```
Last login: Fri Dec 18 19:39:32 on ttys000
Ablas-MacBook-Pro:~ PythonPrincess$ cd desktop
Ablas-MacBook-Pro:desktop PythonPrincess$ python searcher.py -w demyelination -d
.
Ablas-MacBook-Pro:desktop PythonPrincess$ cd final_project/
Ablas-MacBook-Pro:final_project PythonPrincess$ python searcher.py -w demyelinat
ion -d.
searching...
```

the slowest program on the face of the planet

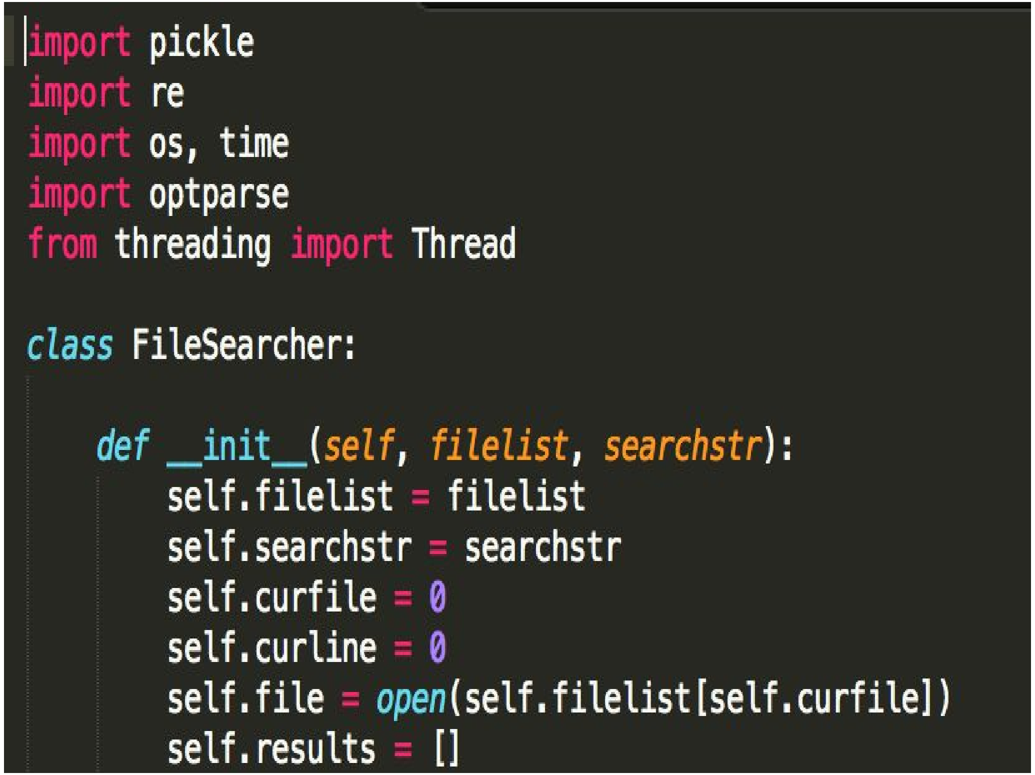
program consists of… (1-3)

(1) **file searcher class:**

__init__ function that takes in all the file-searcher’s attributes (a file list, and a search string)

Since its looping through a list of files, the program needs to know the “set” file list and the one its currently iterating such that self. file = open(self.filelist[self.current_file])

self.results =[] is going to consist of a list of all the positive results



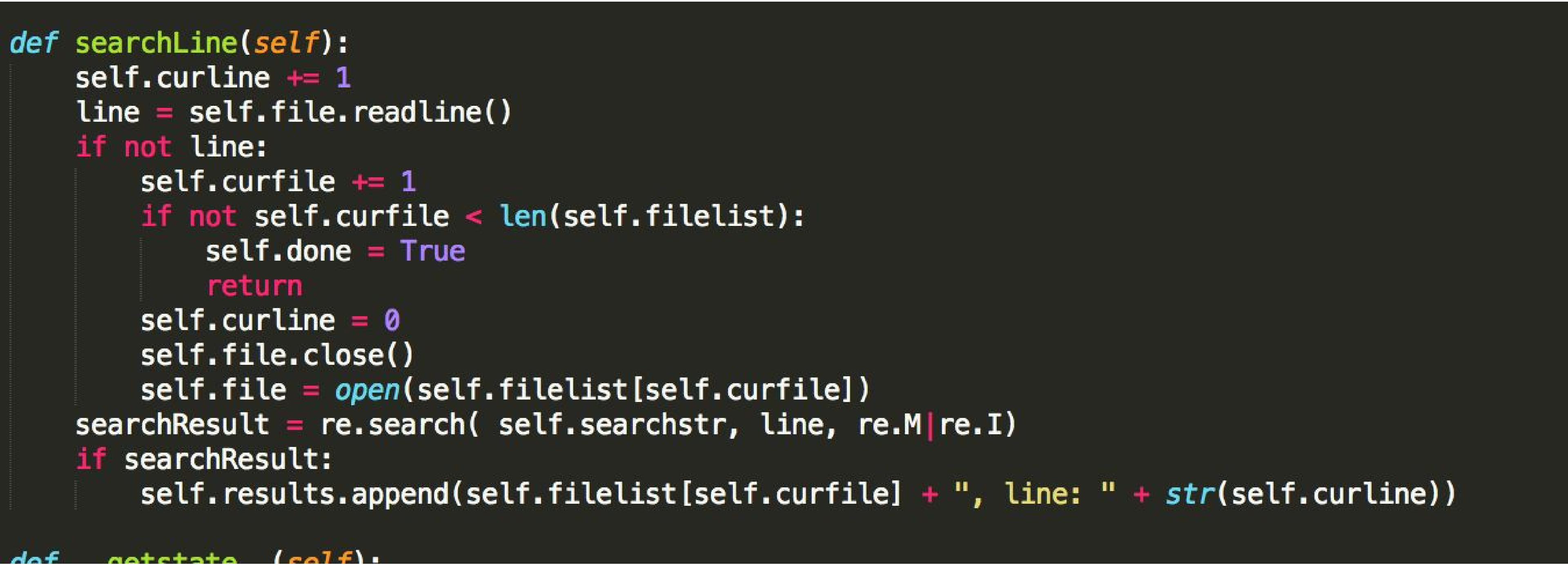
```
import pickle
import re
import os, time
import optparse
from threading import Thread

class FileSearcher:

    def __init__(self, filelist, searchstr):
        self.filelist = filelist
        self.searchstr = searchstr
        self.curfile = 0
        self.curline = 0
        self.file = open(self.filelist[self.curfile])
        self.results = []
```

the program consists of (2-3)

(2) in the class FileSearcher, we need to define a function called searchLine(self):, and this function should increment the lines in the article after its done scanning it: line= self.file.readline() and then it will increment to the next file after its completed scanning the current one

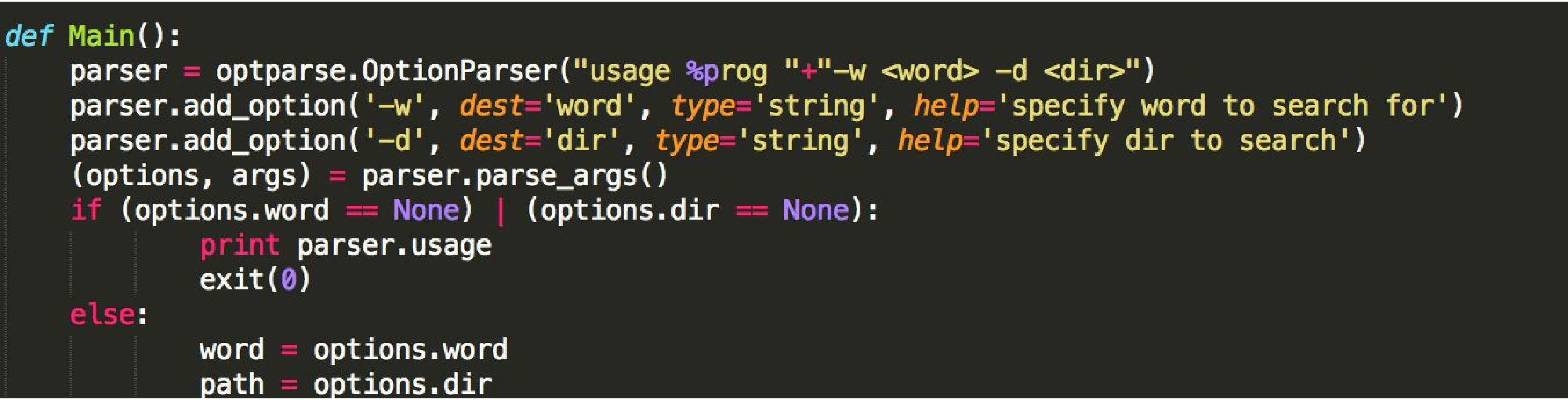


```
def searchLine(self):
    self.curline += 1
    line = self.file.readline()
    if not line:
        self.curfile += 1
        if not self.curfile < len(self.filelist):
            self.done = True
            return
        self.curline = 0
        self.file.close()
        self.file = open(self.filelist[self.curfile])
    searchResult = re.search( self.searchstr, line, re.M|re.I)
    if searchResult:
        self.results.append(self.filelist[self.curfile] + ", line: " + str(self.curline))

def __getstate__(self):
```

the program consists of (3-3):

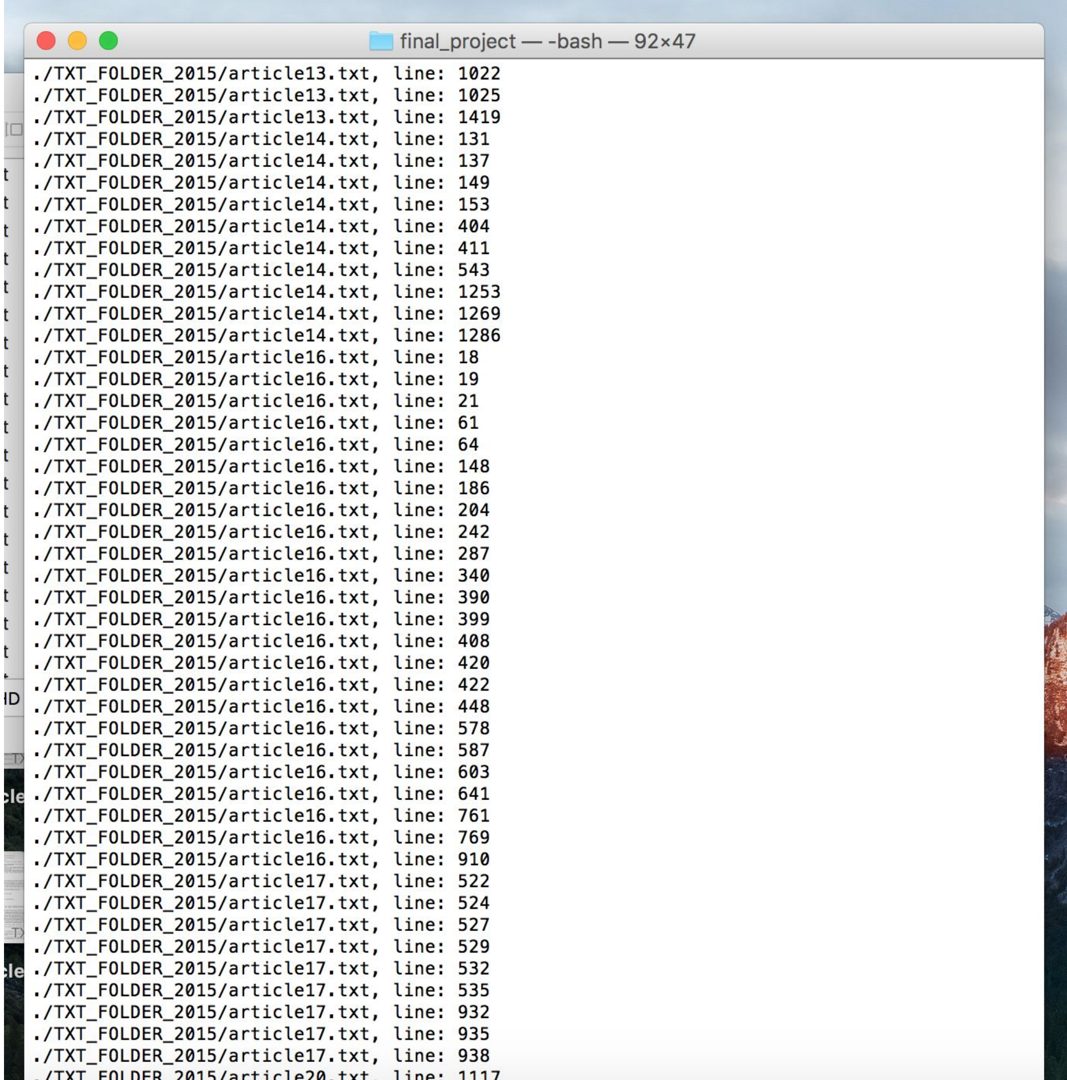
The main function that takes in the input “word” and “directory” and parses through the dataset to determine which files contain the keywords we are looking for and which do not:



```
def Main():
    parser = optparse.OptionParser("usage %prog "+"-w <word> -d <dir>")
    parser.add_option('-w', dest='word', type='string', help='specify word to search for')
    parser.add_option('-d', dest='dir', type='string', help='specify dir to search')
    (options, args) = parser.parse_args()
    if (options.word == None) | (options.dir == None):
        print parser.usage
        exit(0)
    else:
        word = options.word
        path = options.dir
```

It Worked!!

Out of 50 articles, 33 mention the term demyelination



```
./TXT_FOLDER_2015/article13.txt, line: 1022
./TXT_FOLDER_2015/article13.txt, line: 1025
./TXT_FOLDER_2015/article13.txt, line: 1419
./TXT_FOLDER_2015/article14.txt, line: 131
./TXT_FOLDER_2015/article14.txt, line: 137
./TXT_FOLDER_2015/article14.txt, line: 149
./TXT_FOLDER_2015/article14.txt, line: 153
./TXT_FOLDER_2015/article14.txt, line: 404
./TXT_FOLDER_2015/article14.txt, line: 411
./TXT_FOLDER_2015/article14.txt, line: 543
./TXT_FOLDER_2015/article14.txt, line: 1253
./TXT_FOLDER_2015/article14.txt, line: 1269
./TXT_FOLDER_2015/article14.txt, line: 1286
./TXT_FOLDER_2015/article16.txt, line: 18
./TXT_FOLDER_2015/article16.txt, line: 19
./TXT_FOLDER_2015/article16.txt, line: 21
./TXT_FOLDER_2015/article16.txt, line: 61
./TXT_FOLDER_2015/article16.txt, line: 64
./TXT_FOLDER_2015/article16.txt, line: 148
./TXT_FOLDER_2015/article16.txt, line: 186
./TXT_FOLDER_2015/article16.txt, line: 204
./TXT_FOLDER_2015/article16.txt, line: 242
./TXT_FOLDER_2015/article16.txt, line: 287
./TXT_FOLDER_2015/article16.txt, line: 340
./TXT_FOLDER_2015/article16.txt, line: 390
./TXT_FOLDER_2015/article16.txt, line: 399
./TXT_FOLDER_2015/article16.txt, line: 408
./TXT_FOLDER_2015/article16.txt, line: 420
./TXT_FOLDER_2015/article16.txt, line: 422
./TXT_FOLDER_2015/article16.txt, line: 448
./TXT_FOLDER_2015/article16.txt, line: 578
./TXT_FOLDER_2015/article16.txt, line: 587
./TXT_FOLDER_2015/article16.txt, line: 603
./TXT_FOLDER_2015/article16.txt, line: 641
./TXT_FOLDER_2015/article16.txt, line: 761
./TXT_FOLDER_2015/article16.txt, line: 769
./TXT_FOLDER_2015/article16.txt, line: 910
./TXT_FOLDER_2015/article17.txt, line: 522
./TXT_FOLDER_2015/article17.txt, line: 524
./TXT_FOLDER_2015/article17.txt, line: 527
./TXT_FOLDER_2015/article17.txt, line: 529
./TXT_FOLDER_2015/article17.txt, line: 532
./TXT_FOLDER_2015/article17.txt, line: 535
./TXT_FOLDER_2015/article17.txt, line: 932
./TXT_FOLDER_2015/article17.txt, line: 935
./TXT_FOLDER_2015/article17.txt, line: 938
./TXT_FOLDER_2015/article20.txt, line: 1117
```

User Input:

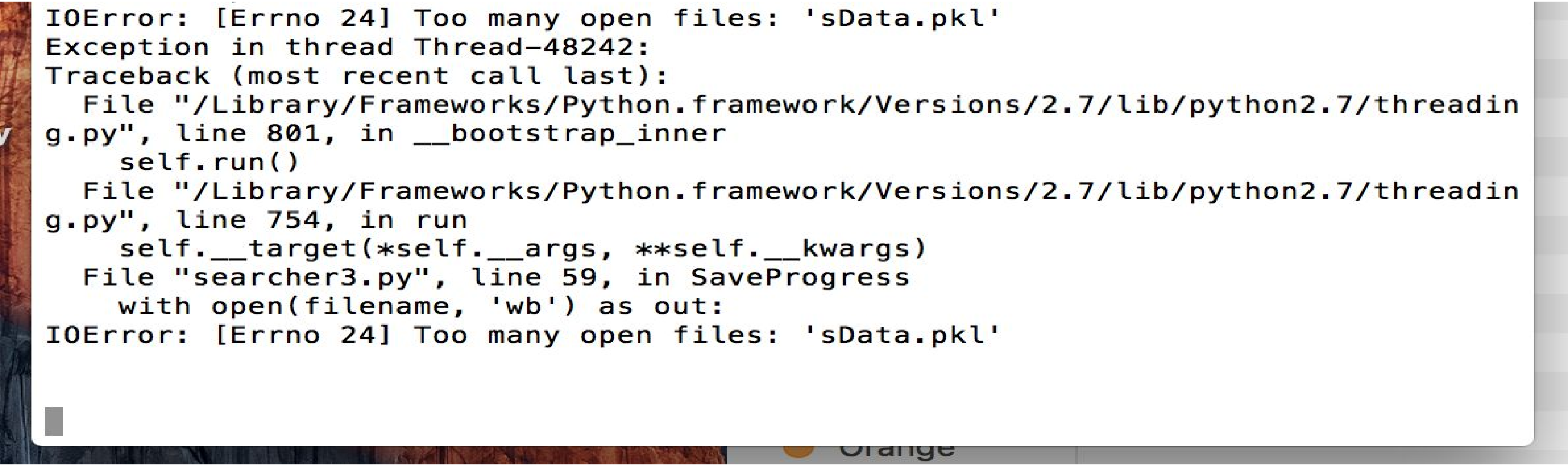
The user (in this case a scientist or clinician) can enter in a name of a neuronal protein of interest and the program would count the occurrences of this protein in articles already prescreened and mention “demyelination”

List of Neuronal Proteins = ("NAV1", "Neuron Navigator 1", "NAV2", "Neuron Navigator 2", "NAV3", "Neuron Navigator 3", "ASCL1", "Achaete-scute family bHLH transcription factor 1", "MNX1", "Motor neuron and pancreas homeobox 1", "ISL1", "ISL LIM homeobox 1", "NRP1", "Neuropilin 1", "GPM6A", "Glycoprotein M6A", "PAX6", "Paired box 6", "CDK5", "Cyclin-dependent kinase 5", "NRCAM", "Neuronal cell adhesion molecule", "SMN2", "Survival of motor neuron 2", "BDNF", "Brain-derived neurotrophic factor", "SMN1", "Survival of motor neuron 1", "PHOX2B", "Paired-like homeobox 2b", "NGF", "Nerve growth factor", "SEMA3A", "Sema domain immunoglobulin domain 3A", "CDK5R1", "Cyclin-dependent kinase 5 regulatory subunit 1", "PARK7", "Parkinson protein 7", "MEF2C", "Myocyte enhancer factor 2C", "DAB1", "Dab reelin signal transducer", "NR4A2", "Nuclear receptor subfamily 4 member 2", "NTRK2", "Neurotrophic tyrosine kinase, receptor, type 2", "PARK2", "Parkin RBR E3 ubiquitin protein ligase", "GATA2", "GATA binding protein 2", "POU4F1", "POU class 4 homeobox 1", "PSEN1", "Presenilin 1", "NSMF", "NMDA receptor synaptonuclear signaling and neuronal migration factor", "RAPGEF2", "Rap guanine nucleotide exchange factor 2", "CNTN2", "Contactin 2", "DCX", "Doublecortin", "ISL2", "ISL LIM homeobox 2", "NDEL1", "NudE neurodevelopment protein 1-like 1", "ATP7A", "ATPase alpha polypeptide", "NKX2-1", "NK2 homeobox 1", "FGF8", "Fibroblast growth factor 8", "LHX3", "LIM homeobox 3", "LRRK2", "Leucine-rich repeat kinase 2", "NRP2", "Neuropilin 2", "APOE", "Apolipoprotein E", "MAP2", "Microtubule-associated protein 2", "APP", "Amyloid beta precursor protein", "PACSIN1", "Protein kinase C and casein kinase substrate in neurons 1", "GRIK2", "Glutamate receptor ionotropic kainate 2", "RET", "Ret proto-oncogene", "BAX", "BCL2-associated X protein", "CACNA1A", "Calcium channel voltage-dependent alpha 1A subunit", "CHRNB2", "Cholinergic receptor nicotinic beta 2", "CRIM1", "Cysteine rich transmembrane BMP regulator 1", "LBX1", "Ladybird homeobox 1",)

one small problem:

the program keeps crashing- it was storing and tracking too much data at once, needed to change the article count from 33 to ~15

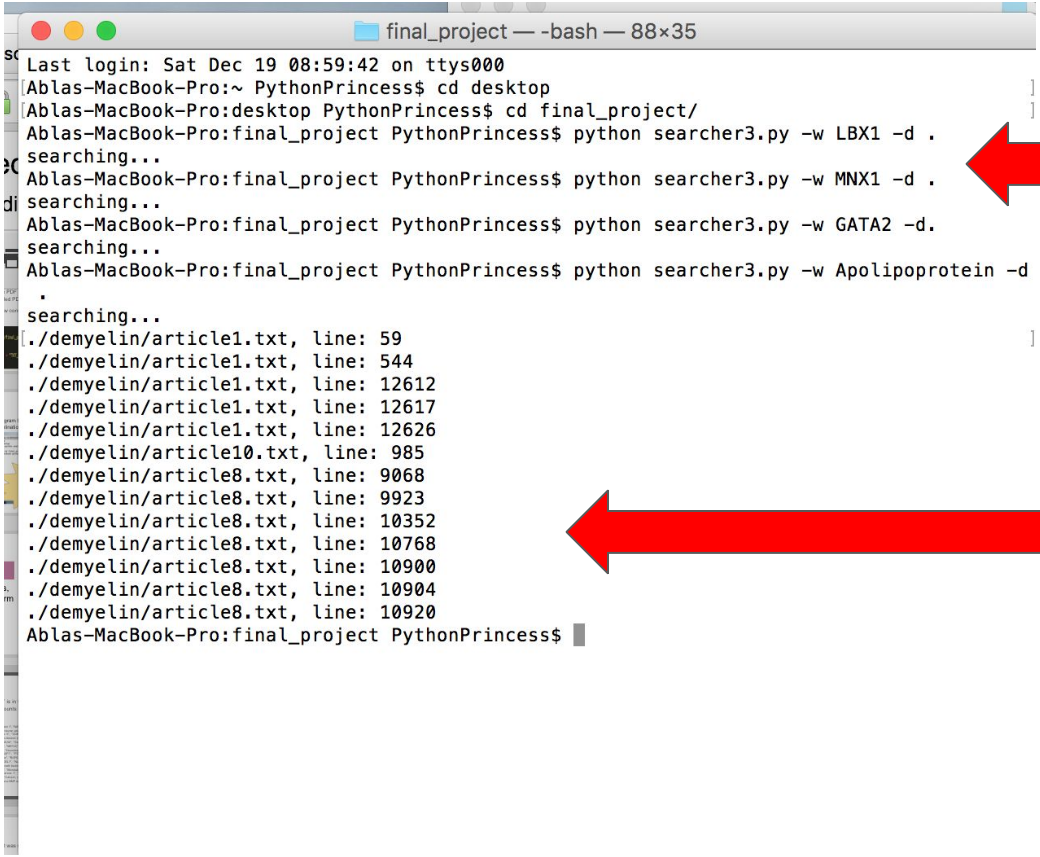
error message when scanning 33 articles:



```
IOError: [Errno 24] Too many open files: 'sData.pkl'
Exception in thread Thread-48242:
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/threadin
g.py", line 801, in __bootstrap_inner
    self.run()
  File "/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/threadin
g.py", line 754, in run
    self.__target(*self.__args, **self.__kwargs)
  File "searcher3.py", line 59, in SaveProgress
    with open(filename, 'wb') as out:
IOError: [Errno 24] Too many open files: 'sData.pkl'
```

conclusion:

Based on the program written, neuronal protein “Apolipoprotein E” is highly associated with demyelination.



```
Last login: Sat Dec 19 08:59:42 on ttys000
Ablas-MacBook-Pro:~ PythonPrincess$ cd desktop
Ablas-MacBook-Pro:desktop PythonPrincess$ cd final_project/
Ablas-MacBook-Pro:final_project PythonPrincess$ python searcher3.py -w LBX1 -d .
searching...
Ablas-MacBook-Pro:final_project PythonPrincess$ python searcher3.py -w MNX1 -d .
searching...
Ablas-MacBook-Pro:final_project PythonPrincess$ python searcher3.py -w GATA2 -d.
searching...
Ablas-MacBook-Pro:final_project PythonPrincess$ python searcher3.py -w Apolipoprotein -d
 .
searching...
./demyelin/article1.txt, line: 59
./demyelin/article1.txt, line: 544
./demyelin/article1.txt, line: 12612
./demyelin/article1.txt, line: 12617
./demyelin/article1.txt, line: 12626
./demyelin/article10.txt, line: 985
./demyelin/article8.txt, line: 9068
./demyelin/article8.txt, line: 9923
./demyelin/article8.txt, line: 10352
./demyelin/article8.txt, line: 10768
./demyelin/article8.txt, line: 10900
./demyelin/article8.txt, line: 10904
./demyelin/article8.txt, line: 10920
Ablas-MacBook-Pro:final_project PythonPrincess$
```

LBX1, MNX1 and GATA2 are not associated at all with demyelination, no data in the literature to support this

Apolipoprotein is highly associated with demyelination, lots of evidence in the literature to support this