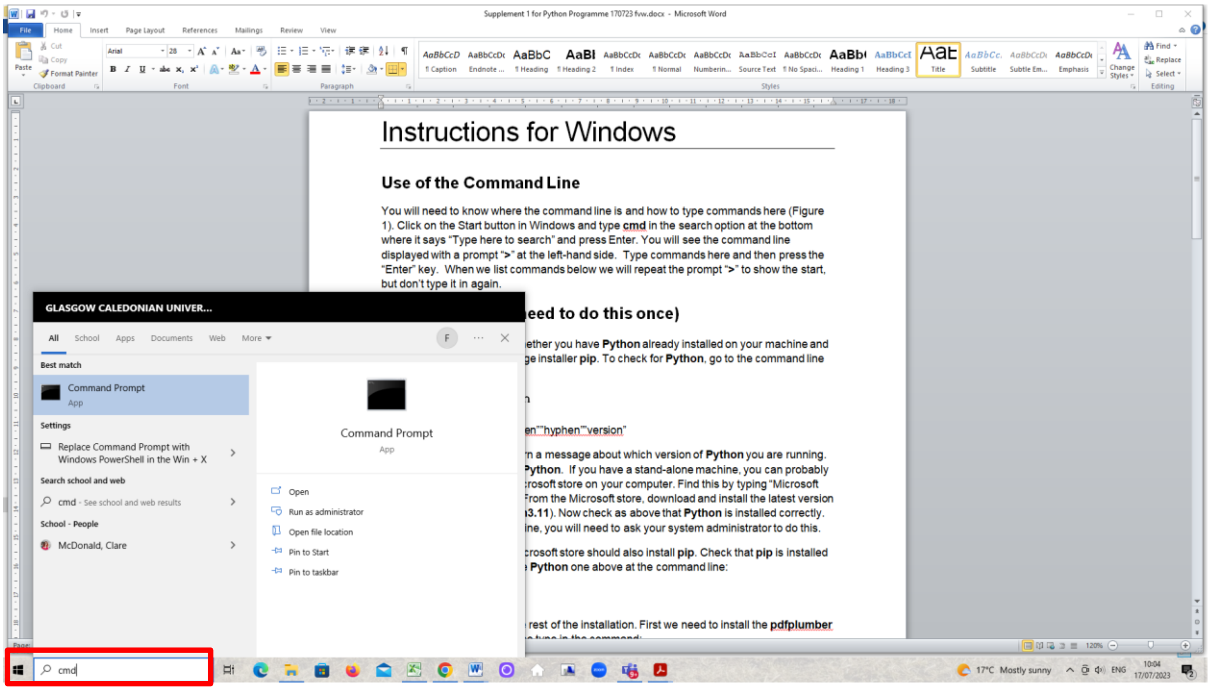
Data mining versus manual screening to select papers for inclusion in systematic reviews: a novel method to increase efficiency

Elena Ierardi, J Chris Eilbeck, Frederike van Wijck, Myzoon Ali, Fiona Coupar

Instructions for Windows

Use of the Command Line

You will need to know where the command line is and how to type commands here (Figure 1). Click on the Start button in Windows and type “**cmd”** in the search option at the bottom where it says “Type here to search” and press Enter.



Supplement 1 for Python Programme 170723 fvw.docx - Microsoft Word

Instructions for Windows

Use of the Command Line

You will need to know where the command line is and how to type commands here (Figure 1). Click on the Start button in Windows and type **cmd** in the search option at the bottom where it says “Type here to search” and press Enter. You will see the command line displayed with a prompt “**>**” at the left-hand side. Type commands here and then press the “Enter” key. When we list commands below we will repeat the prompt “**>**” to show the start, but don’t type it in again.

eed to do this once)

ether you have **Python** already installed on your machine and
ge installer **pip**. To check for **Python**, go to the command line

n

en””hyphen””version”

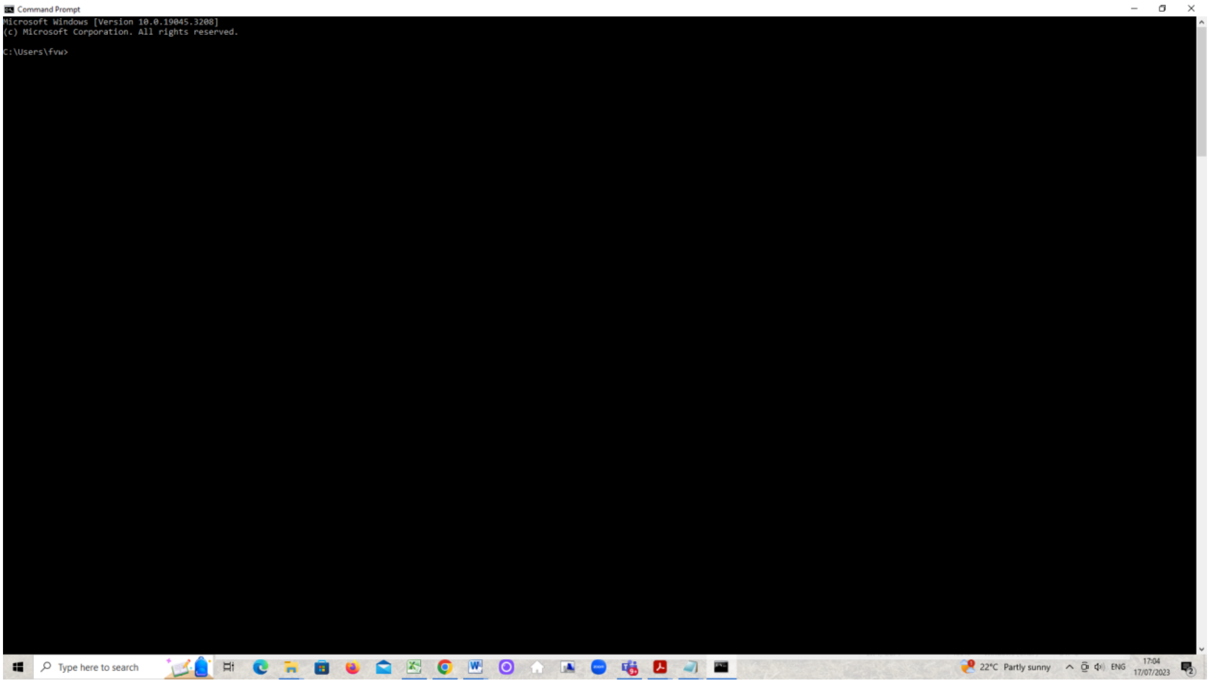
n a message about which version of **Python** you are running.
Python. If you have a stand-alone machine, you can probably
rosoft store on your computer. Find this by typing “Microsoft
From the Microsoft store, download and install the latest version
3.11). Now check as above that **Python** is installed correctly.
ine, you will need to ask your system administrator to do this.

rosoft store should also install **pip**. Check that **pip** is installed
Python one above at the command line:

rest of the installation. First we need to install the **pdfplumber**

Figure 1. Typing “cmd” into the ‘Type here to search’ window

You will see the command line displayed with a prompt “**>**” at the left-hand side (Fig. 2). Type commands here and then press the “Enter” key. When we list commands below we will repeat the prompt “**>**” to show the start, but don’t type it in again.



```
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Users\fvw>
```

Figure 2. Display of the command line in Command Prompt

Preparation (you only need to do this once)

The next steps depend on whether you have **Python** already installed on your machine and in addition the **Python** package installer **pip**.

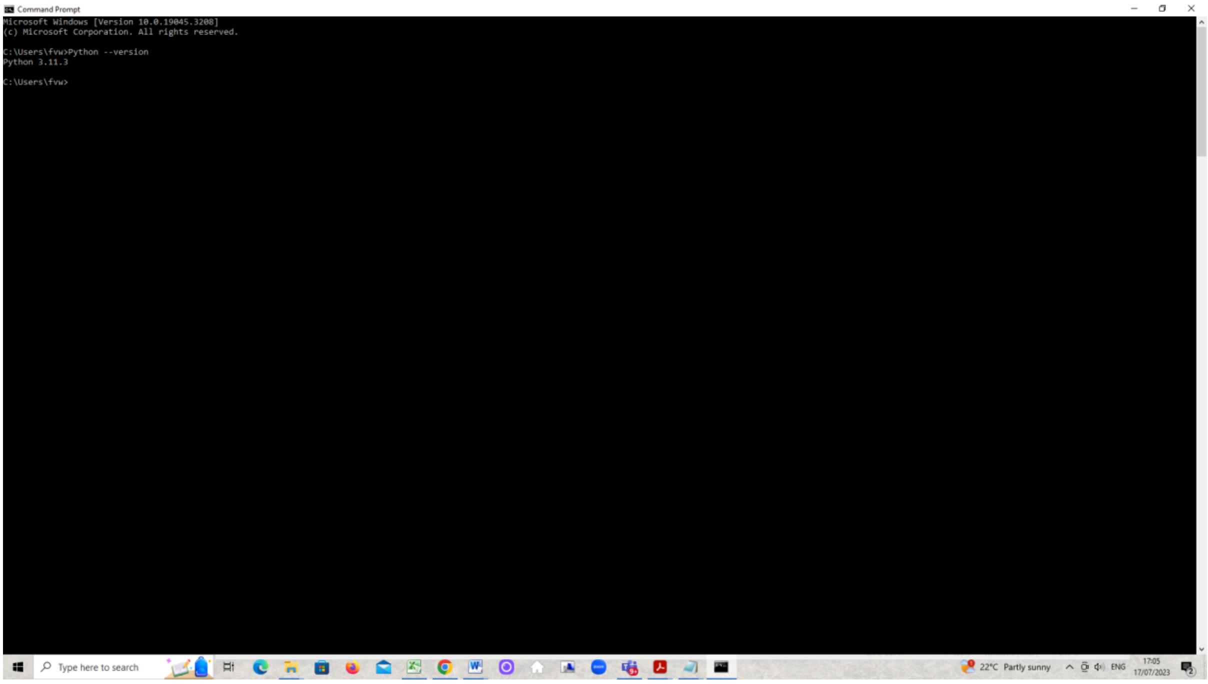
Checking Python

To check for **Python**, go to the command line and type:

```
Python --version
```

This is “Python”“space””hyphen””hyphen””version”

If successful, this should return a message about which version of **Python** you are running (Fig. 3).



```
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Users\fvw>Python --version
Python 3.11.3

C:\Users\fvw>
```

Figure 3. Response indicating the available version of Python

Installing Python

If you don’t have **Python** installed yet, you will need to do so. If you have a stand-alone machine, you can probably do this yourself. Go to the Microsoft store on your computer. Find this by typing “MS store” in the search window (Fig. 4).

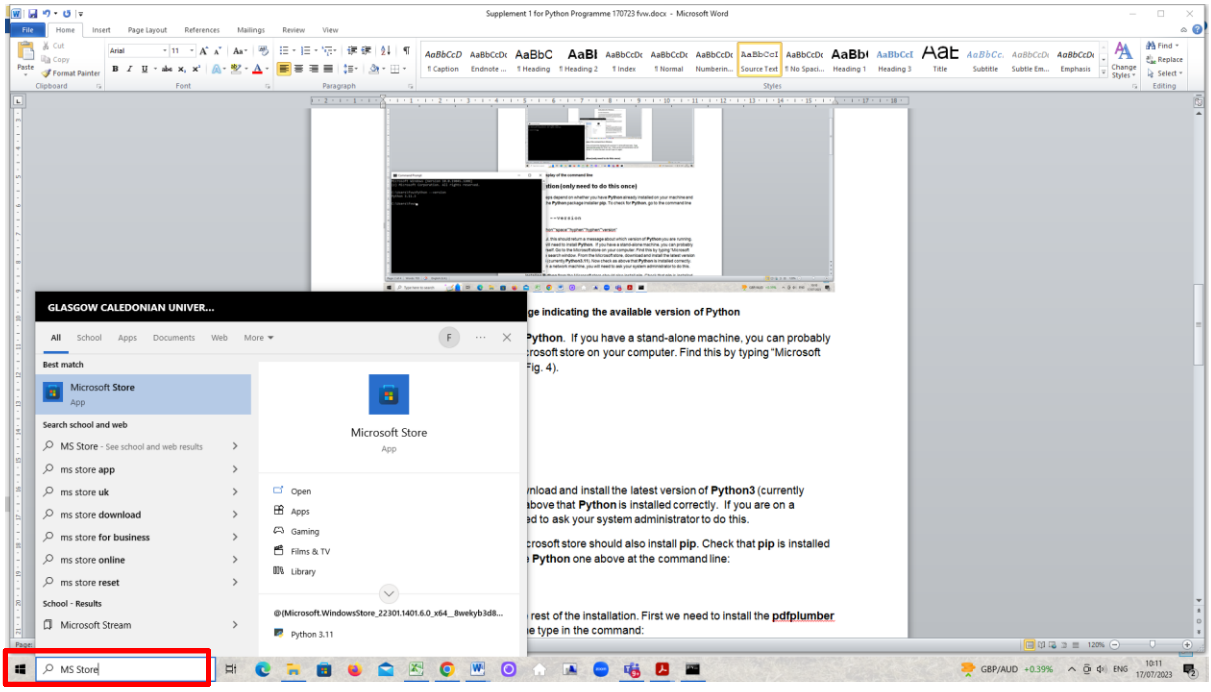


Fig. 4. Typing “MS Store” into the search window

From the Microsoft store, download and install the latest version of **Python3** (currently **Python3.11**). Now check as above that **Python** is installed correctly. If you are on a network machine, you may need to ask your system administrator to do this.

Installing **Python** from the Microsoft store should also install **pip**. Check that **pip** is installed with a command similar to the **Python** one above at the command line:

```
pip --version
```

We can now proceed with the rest of the installation. First we need to install the **pdfplumber** package. At the command line type in the command:

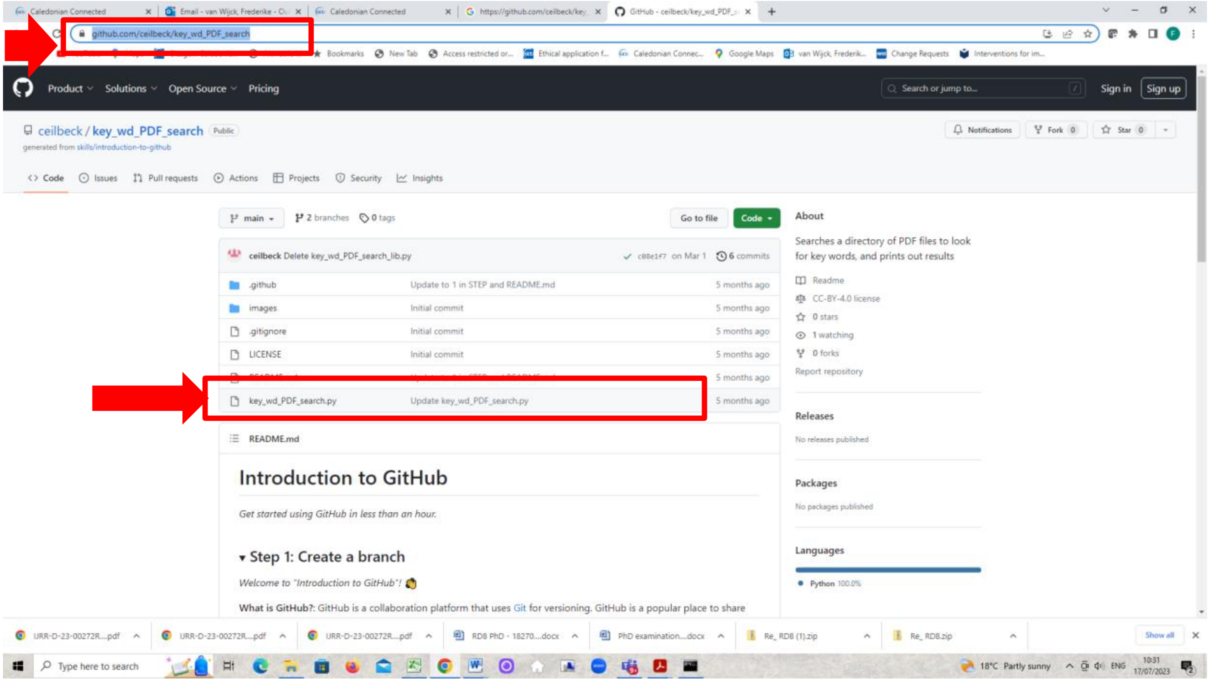
```
> pip install pdfplumber
```

It may take a little while, but you should get eventually a message saying “Successfully installed **pdfplumber**”.

Downloading the Python programme to search for key words

Copy this address into your web browser (Fig. 5, top arrow):

https://github.com/ceilbeck/key_wd_PDF_search



ceilbeck Delete key_wd_PDF_search_lib.py ✓ c08e1f7 on Mar 1 6 commits

.github	Update to 1 in STEP and README.md	5 months ago
images	Initial commit	5 months ago
.gitignore	Initial commit	5 months ago
LICENSE	Initial commit	5 months ago
README.md	Update to 1 in STEP and README.md	5 months ago
key_wd_PDF_search.py	Update key_wd_PDF_search.py	5 months ago

About

Searches a directory of PDF files to look for key words, and prints out results

README.md

Introduction to GitHub

Get started using GitHub in less than an hour.

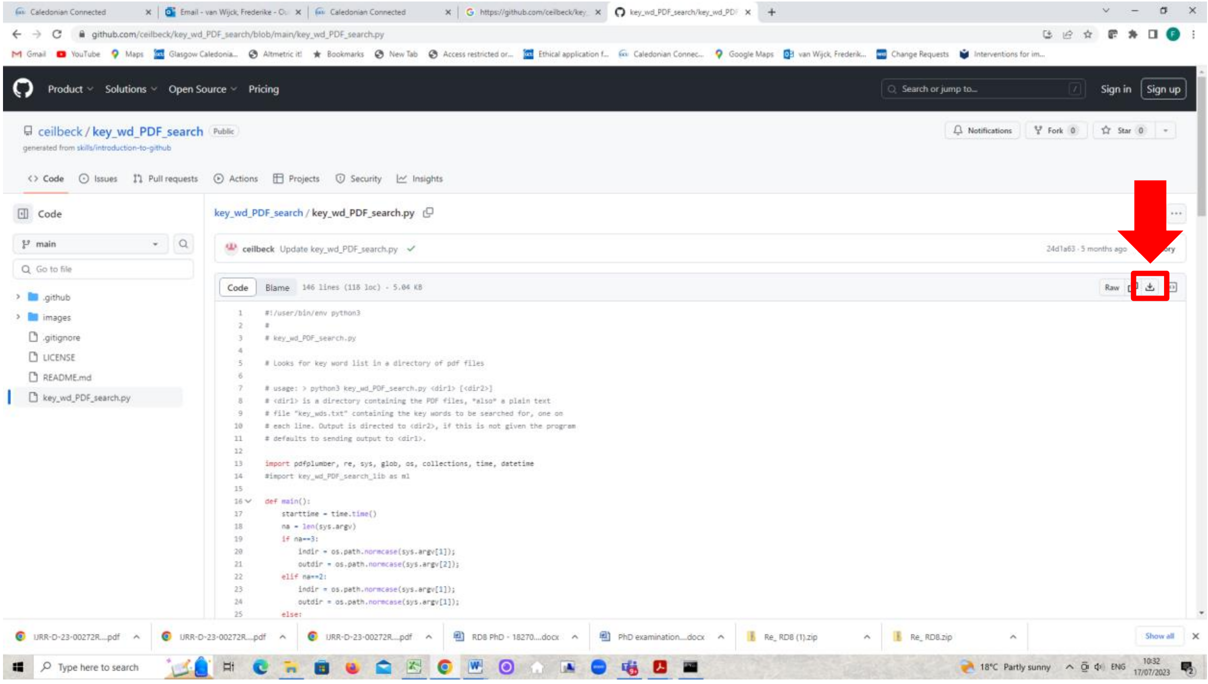
Step 1: Create a branch

Welcome to "Introduction to GitHub"!

What is GitHub?: GitHub is a collaboration platform that uses Git for versioning. GitHub is a popular place to share

Fig. 5. Searching for the Python programme to search for key words on Github

Then click on the hyperlink `key_wd_PDF_search.py` (Fig. 5, bottom arrow). This takes you to the actual programme.



```
#!/user/bin/env python3
#
# key_wd_PDF_search.py

# Looks for key word list in a directory of pdf files

# usage: > python3 key_wd_PDF_search.py <dir1> [<dir2>]
# <dir1> is a directory containing the PDF files, *also* a plain text
# file "key_wds.txt" containing the key words to be searched for, one on
# each line. Output is directed to <dir2>, if this is not given the program
# defaults to sending output to <dir1>.

import pdfplumber, re, sys, glob, os, collections, time, datetime
#import key_wd_PDF_search_lib as ml

def main():
    starttime = time.time()
    na = len(sys.argv)
    if na==3:
        indir = os.path.normcase(sys.argv[1]);
        outdir = os.path.normcase(sys.argv[2]);
    elif na==2:
        indir = os.path.normcase(sys.argv[1]);
        outdir = os.path.normcase(sys.argv[1]);
    else:
```

Fig. 6. Finding the Python programme to search for key words

Now click on the ‘Download’ icon, which downloads the programme (Fig. 6). This will need to be stored in a specific directory, as explained in the next step.

Creating a directory for your work

Now you need to set up a suitable directory system for you work. Still on the command line, enter:

```
cd ..
```

That’s **cd** “space” followed by two dots. This should take you to your main directory on your **C** drive, something like **c:\\user >**

At the command line, create a subfolder and name it "**programs**":

```
mkdir programs
```

move into this subfolder:

```
cd programs
```

and create another subfolder:

```
mkdir PDFfiles
```

Moving files into their correct folders

It is crucial that the various files are in their designated folders, otherwise the program is unable to find the necessary information to do its work:

1. Copy the following files in the "**programs**" folder you created earlier:
   1. this Instructions file (from this article's Supplementary files) and
   2. the **key_wd_PDF_search.py** file (downloaded from Github, as explained above)
2. Copy the **key_wrd.txt** in the "**PDFfiles**" subfolder (This is the sub folder of the "**programs**" folder).
3. Change the keywords as per your requirements. To do that, open the **key_wrd.txt** from the "**PDFfiles**" subfolder, amend any of the keywords and save the file (make sure to retain the file name and its extension (i.e. ".txt"), or the computer program won't work as it does not recognise the file.
4. Put all of your PDF files (on which you want to do a keyword search) in "**PDFfiles**" subfolder

Running the data mining programme

You are now ready to run the data mining programme:

1. Open the **Command Prompt** in Windows (see above if you are not sure how) and type:

   `cd ..` (that's **cd** "space" followed by two dots)

   into the Command Prompt and press the **Enter** key on your keyboard (repeat if necessary until it displays **C:\>** to show that you are in the main C: drive).
2. Switch to the folder "**programs**" by typing:

   `cd programs` (that's **cd** "space" followed by "programs").

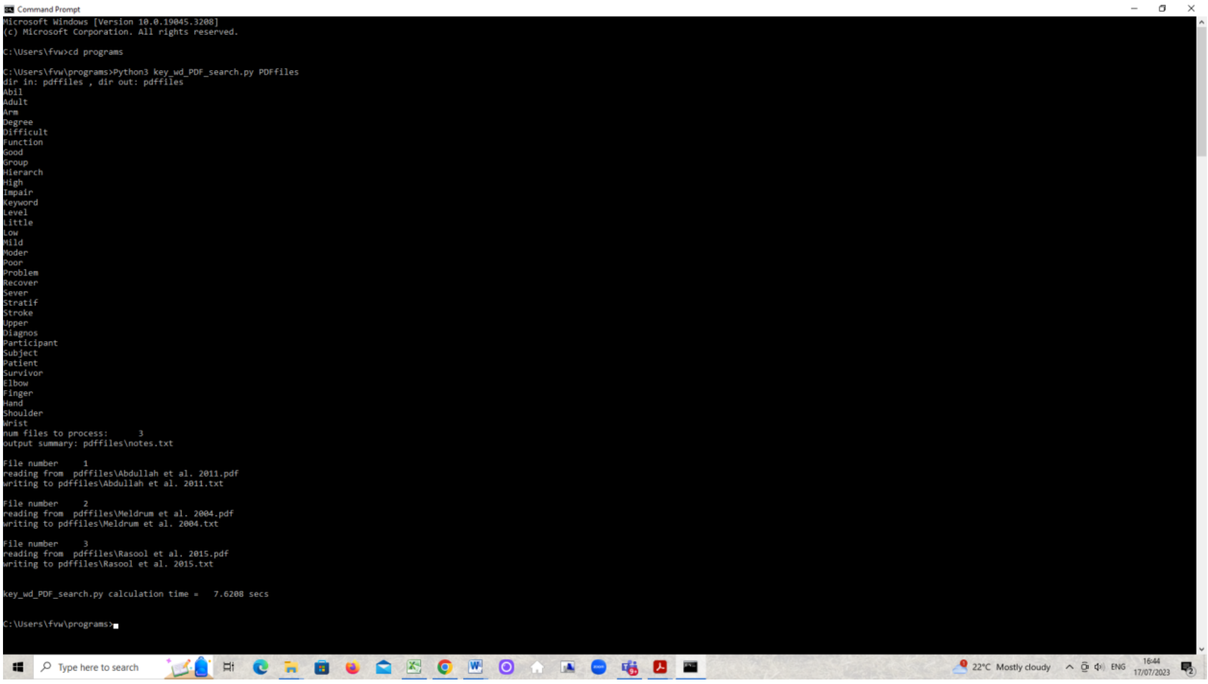
   into the Command Prompt and press the **Enter** key on your keyboard.

3. Now type:

   ```
   Python3 key_wd_PDF_search.py PDFfiles
   ```

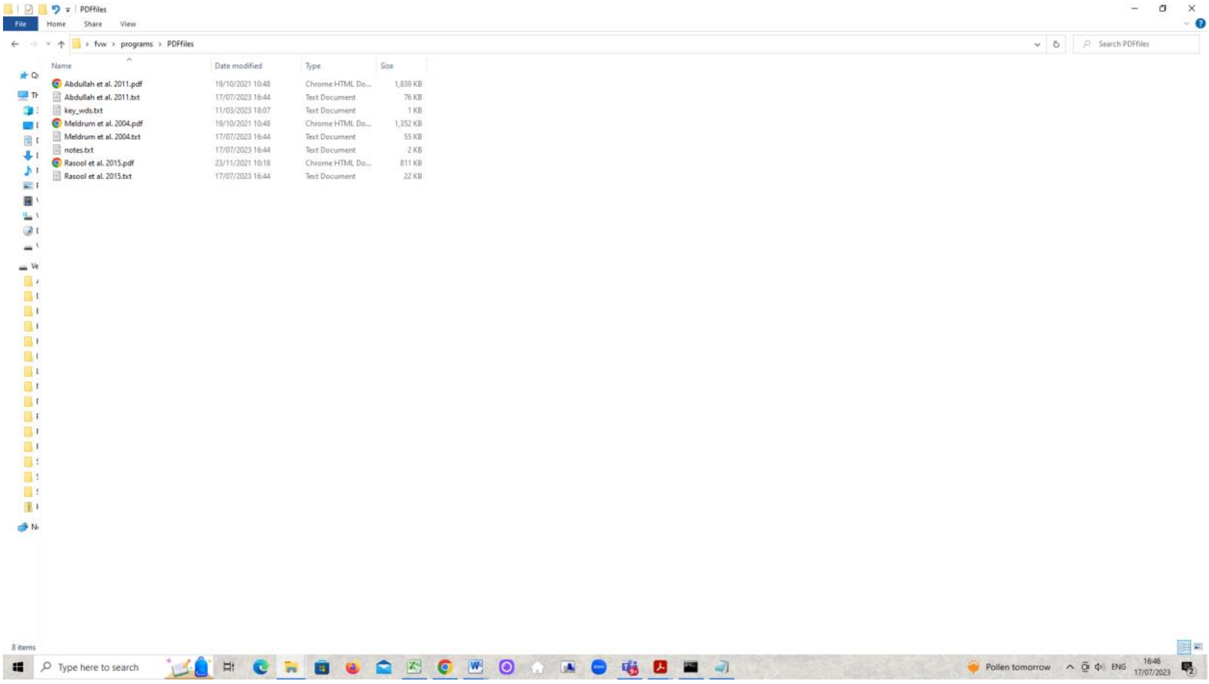
   into the Command Prompt and press the **Enter** key on your keyboard.
4. The Data Mining Tool will now run and the screen will be blank during this time. When it completes (Fig. 7), the generated output text files (one for each paper processed) will be ready to view in the subfolder "**PDFfiles**" (Fig. 8).
5. A summary of the resulting can also be found in "Notes.txt" in the same folder.

NOTE*: Please note that files which are password-protected or that are images (scanned) are not going to be processed and these output files will be empty. The program may skip out of a PDF being processed if it encounters a problem, so not all pages may be returned.*



```
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Users\fvw>cd programs

C:\Users\fvw\programs>Python3 key_wd_PDF_search.py PDFfiles
dir in: pdffiles , dir out: pdffiles
Abil
Adult
Arm
Degree
Difficult
Function
Good
Group
Hierarch
High
Impair
Keyword
Level
Little
Low
Mild
Moder
Poor
Problem
Recover
Sever
Stratif
Stroke
Upper
Diagnos
Participant
Subject
Patient
Survivor
Elbow
Finger
Hand
Shoulder
Wrist
num files to process:      3
output summary: pdffiles\notes.txt

File number     1
reading from  pdffiles\Abdullah et al. 2011.pdf
writing to pdffiles\Abdullah et al. 2011.txt

File number     2
reading from  pdffiles\Meldrum et al. 2004.pdf
writing to pdffiles\Meldrum et al. 2004.txt

File number     3
reading from  pdffiles\Rasool et al. 2015.pdf
writing to pdffiles\Rasool et al. 2015.txt


key_wd_PDF_search.py calculation time =   7.6208 secs


C:\Users\fvw\programs>
```

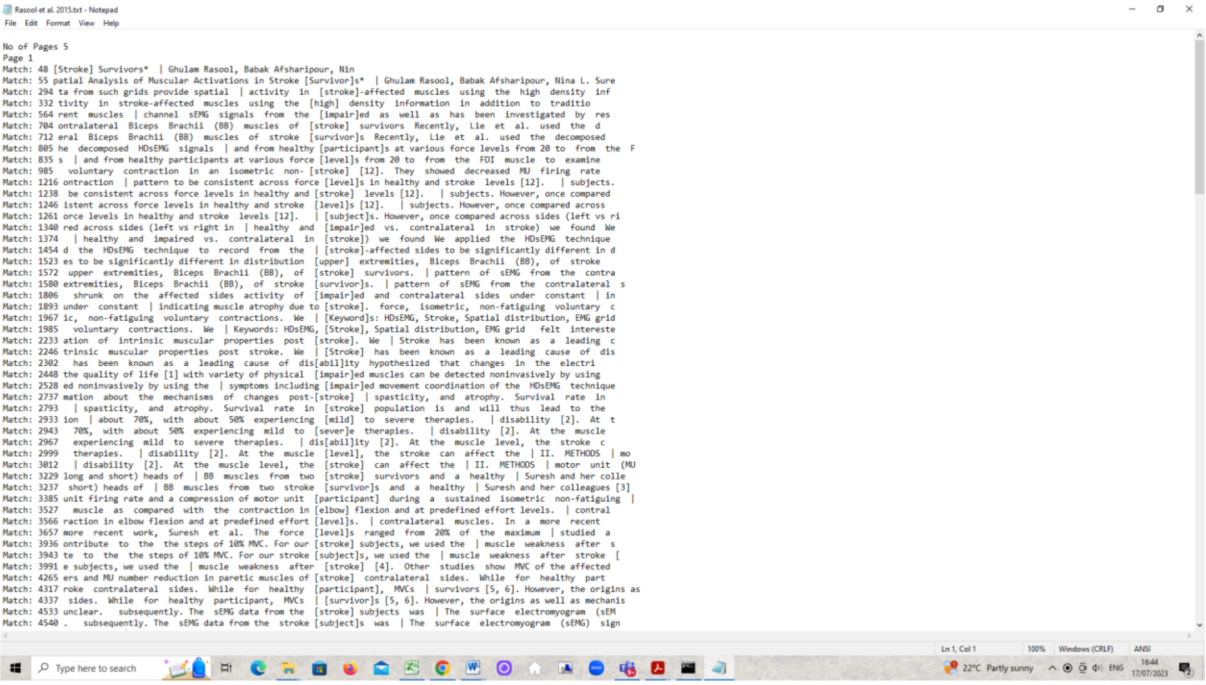
Fig. 7. Screenshot of the Python data mining programme having completed



Name	Date modified	Type	Size
Abdullah et al. 2011.pdf	19/10/2021 10:48	Chrome HTML Do...	1,839 KB
Abdullah et al. 2011.txt	17/07/2023 16:44	Text Document	76 KB
key_wds.txt	11/03/2023 18:07	Text Document	1 KB
Meldrum et al. 2004.pdf	19/10/2021 10:48	Chrome HTML Do...	1,352 KB
Meldrum et al. 2004.txt	17/07/2023 16:44	Text Document	55 KB
notes.txt	17/07/2023 16:44	Text Document	2 KB
Rasool et al. 2015.pdf	23/11/2021 10:18	Chrome HTML Do...	811 KB
Rasool et al. 2015.txt	17/07/2023 16:44	Text Document	22 KB

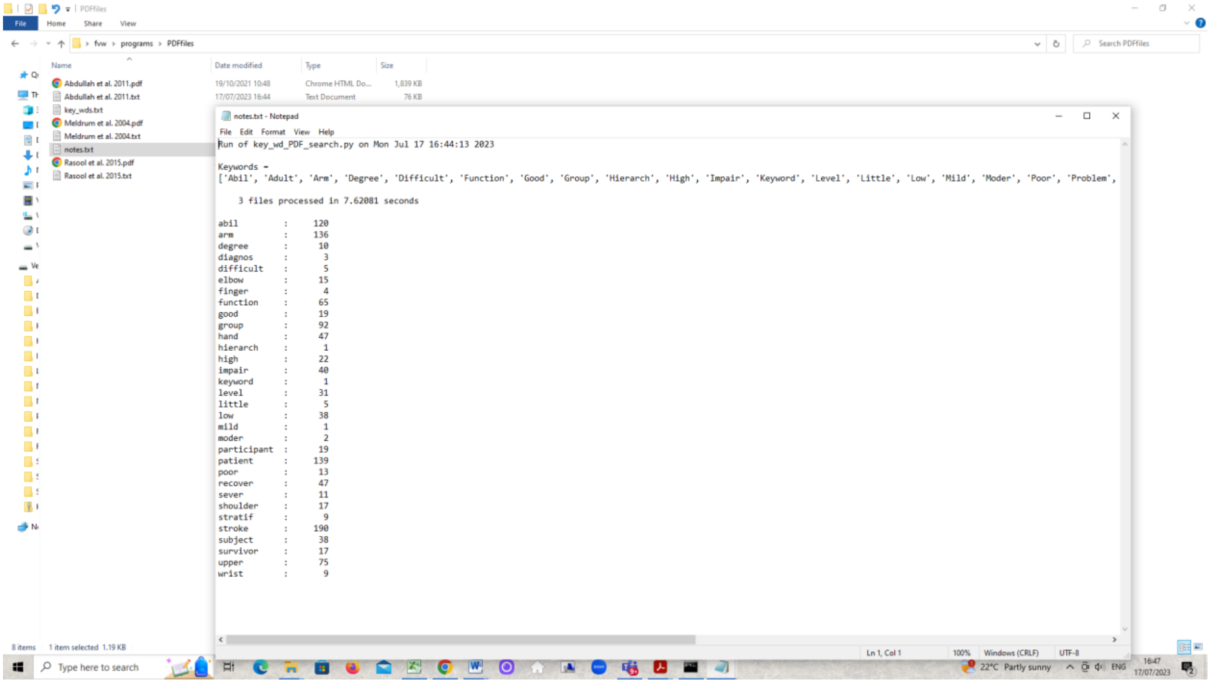
Fig. 8. Screenshot of the Output files and Notes files added to the PDF files subfolder

Figure 9 shows what the output from a single PDF may look like, and Fig. 10 shows the contents of the Notes files after three PDFs have been processed; this file records the key words, the number of times they have been identified in all PDF files analysed, and processing time.



```
No of Pages 5
Page 1
Match: 48 [Stroke] Survivors*  | Ghulam Rasool, Babak Afsharipour, Nin
Match: 55 patial Analysis of Muscular Activations in Stroke [Survivor]s*  | Ghulam Rasool, Babak Afsharipour, Nina L. Sure
Match: 294 ta from such grids provide spatial  | activity  in  [stroke]-affected  muscles  using  the  high  density  inf
Match: 332 tivity  in  stroke-affected  muscles  using  the  [high]  density  information  in  addition  to  traditio
Match: 564 rent  muscles  | channel  sEMG  signals  from  the  [impair]ed  as  well  as  has  been  investigated  by  res
Match: 704 ontralateral  Biceps  Brachii  (BB)  muscles  of  [stroke]  survivors  Recently,  Lie  et  al.  used  the  d
Match: 712 eral  Biceps  Brachii  (BB)  muscles  of  stroke  [survivor]s  Recently,  Lie  et  al.  used  the  decomposed
Match: 805 he  decomposed  HDsEMG  signals  | and from healthy [participant]s at various force levels from 20 to  from  the  F
Match: 835 s  | and from healthy participants at various force [level]s from 20 to  from  the  FDI  muscle  to  examine
Match: 985  voluntary  contraction  in  an  isometric  non- [stroke]  [12].  They  showed  decreased  MU  firing  rate
Match: 1216 ontraction  | pattern to be consistent across force [level]s in healthy and stroke  levels [12].   | subjects.
Match: 1238  be consistent across force levels in healthy and [stroke]  levels [12].   | subjects. However, once compared
Match: 1246 istent across force levels in healthy and stroke  [level]s [12].   | subjects. However, once compared across
Match: 1261 orce levels in healthy and stroke  levels [12].   | [subject]s. However, once compared across sides (left vs ri
Match: 1340 red across sides (left vs right in  | healthy  and  [impair]ed  vs.  contralateral  in  stroke)  we  found  We
Match: 1374   | healthy  and  impaired  vs.  contralateral  in  [stroke])  we  found  We  applied  the  HDsEMG  technique
Match: 1454 d  the  HDsEMG  technique  to  record  from  the  | [stroke]-affected sides to be significantly different in d
Match: 1523 es to be significantly different in distribution  [upper]  extremities,  Biceps  Brachii  (BB),  of  stroke
Match: 1572  upper  extremities,  Biceps  Brachii  (BB),  of  [stroke]  survivors.  | pattern  of  sEMG  from  the  contra
Match: 1580 extremities,  Biceps  Brachii  (BB),  of  stroke  [survivor]s.  | pattern  of  sEMG  from  the  contralateral  s
Match: 1806  shrunk  on  the  affected  sides  activity  of  [impair]ed  and  contralateral  sides  under  constant  | in
Match: 1893 under  constant  | indicating muscle atrophy due to [stroke].  force,  isometric,  non-fatiguing  voluntary  c
Match: 1967 ic,  non-fatiguing  voluntary  contractions.  We  | [Keyword]s: HDsEMG, Stroke, Spatial distribution, EMG grid
Match: 1985  voluntary  contractions.  We  | Keywords: HDsEMG, [Stroke], Spatial distribution, EMG grid  felt  intereste
Match: 2233 ation  of  intrinsic  muscular  properties  post  [stroke].  We  | Stroke  has  been  known  as  a  leading  c
Match: 2246 trinsic  muscular  properties  post  stroke.  We  | [Stroke]  has  been  known  as  a  leading  cause  of  dis
Match: 2302  has  been  known  as  a  leading  cause  of  dis[abil]ity  hypothesized  that  changes  in  the  electri
Match: 2448 the quality of life [1] with variety of physical  [impair]ed muscles can be detected noninvasively by using
Match: 2528 ed noninvasively by using the  | symptoms including [impair]ed movement coordination of the  HDsEMG  technique
Match: 2737 mation  about  the  mechanisms  of  changes  post-[stroke]  | spasticity,  and  atrophy.  Survival  rate  in
Match: 2793  | spasticity,  and  atrophy.  Survival  rate  in  [stroke]  population  is  and  will  thus  lead  to  the
Match: 2933 ion  | about  70%,  with  about  50%  experiencing  [mild]  to  severe  therapies.   | disability  [2].  At  t
Match: 2943  70%,  with  about  50%  experiencing  mild  to  [sever]e  therapies.   | disability  [2].  At  the  muscle
Match: 2967  experiencing  mild  to  severe  therapies.   | dis[abil]ity  [2].  At  the  muscle  level,  the  stroke  c
Match: 2999  therapies.   | disability  [2].  At  the  muscle  [level],  the  stroke  can  affect  the  | II.  METHODS  | mo
Match: 3012  | disability  [2].  At  the  muscle  level,  the  [stroke]  can  affect  the  | II.  METHODS  | motor  unit  (MU
Match: 3229 long and short) heads of  | BB  muscles  from  two  [stroke]  survivors  and  a  healthy  | Suresh and her colle
Match: 3237  short) heads of  | BB  muscles  from  two  stroke  [survivor]s  and  a  healthy  | Suresh and her colleagues [3]
Match: 3385 unit firing rate and a compression of motor unit  [participant]  during  a  sustained  isometric  non-fatiguing  |
Match: 3527  muscle  as  compared  with  the  contraction in [elbow] flexion and at predefined effort levels.  | contral
Match: 3566 raction in elbow flexion and at predefined effort [level]s.  | contralateral  muscles.  In  a  more  recent
Match: 3657 more  recent  work,  Suresh  et  al.  The  force  [level]s  ranged  from  20%  of  the  maximum  | studied  a
Match: 3936 ontribute  to  the  the steps of 10% MVC. For our [stroke] subjects, we used the  | muscle  weakness  after  s
Match: 3943 te  to  the  the steps of 10% MVC. For our stroke [subject]s, we used the  | muscle  weakness  after  stroke  [
Match: 3991 e subjects, we used the  | muscle  weakness  after  [stroke]  [4].  Other  studies  show  MVC of the affected
Match: 4265 ers and MU number reduction in paretic muscles of [stroke]  contralateral  sides.  While  for  healthy  part
Match: 4317 roke  contralateral  sides.  While  for  healthy  [participant],  MVCs  | survivors [5, 6]. However, the origins as
Match: 4337  sides.  While  for  healthy  participant,  MVCs  | [survivor]s [5, 6]. However, the origins as well as mechanis
Match: 4533 unclear.   subsequently. The  sEMG data from the  [stroke] subjects  was  | The  surface  electromyogram  (sEM
Match: 4540 .   subsequently. The  sEMG data from the  stroke [subject]s  was  | The  surface  electromyogram  (sEMG)  sign
```

Fig. 9. Screenshot of an excerpt of one of the Output files



Name	Date modified	Type	Size
Abdullah et al. 2011.pdf	19/10/2021 10:48	Chrome HTML Do...	1,839 KB
Abdullah et al. 2011.txt	17/07/2023 16:44	Text Document	76 KB
key_wds.txt			
Meldrum et al. 2004.pdf			
Meldrum et al. 2004.txt			
notes.txt			
Rasool et al. 2015.pdf			
Rasool et al. 2015.txt			

```
Run of key_wd_PDF_search.py on Mon Jul 17 16:44:13 2023

Keywords =
['Abil', 'Adult', 'Arm', 'Degree', 'Difficult', 'Function', 'Good', 'Group', 'Hierarch', 'High', 'Impair', 'Keyword', 'Level', 'Little', 'Low', 'Mild', 'Moder', 'Poor', 'Problem',

    3 files processed in 7.62081 seconds

abil        :     120
arm         :     136
degree      :      10
diagnos     :       3
difficult   :       5
elbow       :      15
finger      :       4
function    :      65
good        :      19
group       :      92
hand        :      47
hierarch    :       1
high        :      22
impair      :      40
keyword     :       1
level       :      31
little      :       5
low         :      38
mild        :       1
moder       :       2
participant :      19
patient     :     139
poor        :      13
recover     :      47
sever       :      11
shoulder    :      17
stratif     :       9
stroke      :     190
subject     :      38
survivor    :      17
upper       :      75
wrist       :       9
```

Fig. 10. Screenshot of the Notes file, based on a sample of three processed PDF papers.