Filename Well-formedness Checker

Table of contents

Why bother? Is this much ado about nothing?

Why would you want a filename checker? A filename is a filename. What's there to check? True. Unless… Your filename contains certain expected information. Take this filename, for instance: TIPS_NLP_Filename checker.pdf. The filename contains three pieces of data, separated by the special character _. The first item, TIPS, indicates that the file is a Help file for a computer project called NLP. The third item gives us a clue about the actual content of the file: It is about checking filenames.

Nearly all scripts (through their GUIs) in the NLP Suite allow users to open specific TIPS files to familiarize themselves with a topic. But… Any error in the TIPS filename and the function that opens and displays a TIPS file in the NLP Suite would fail.

Metadata embedded in filename

What's metadata?

Metadata are a set of data that provides information about other data. We can cleverly embed metadata in filenames.

Which metadata you want to embed in your filename depends upon the specific project you are using files for.

If you are using newspaper articles on **protest events**, you may wish to include the newspaper name (e.g., The New York Times), the date (e.g., 12-02-1982), the page and column numbers where the article was published (e.g., 4 and 1), the event described (e.g., Black lives matter).
If you are using newspaper articles on **best-selling book reviews**, you may wish to embed the reviewer name, the book's author name, and the book's title besides all the other information on the newspaper.

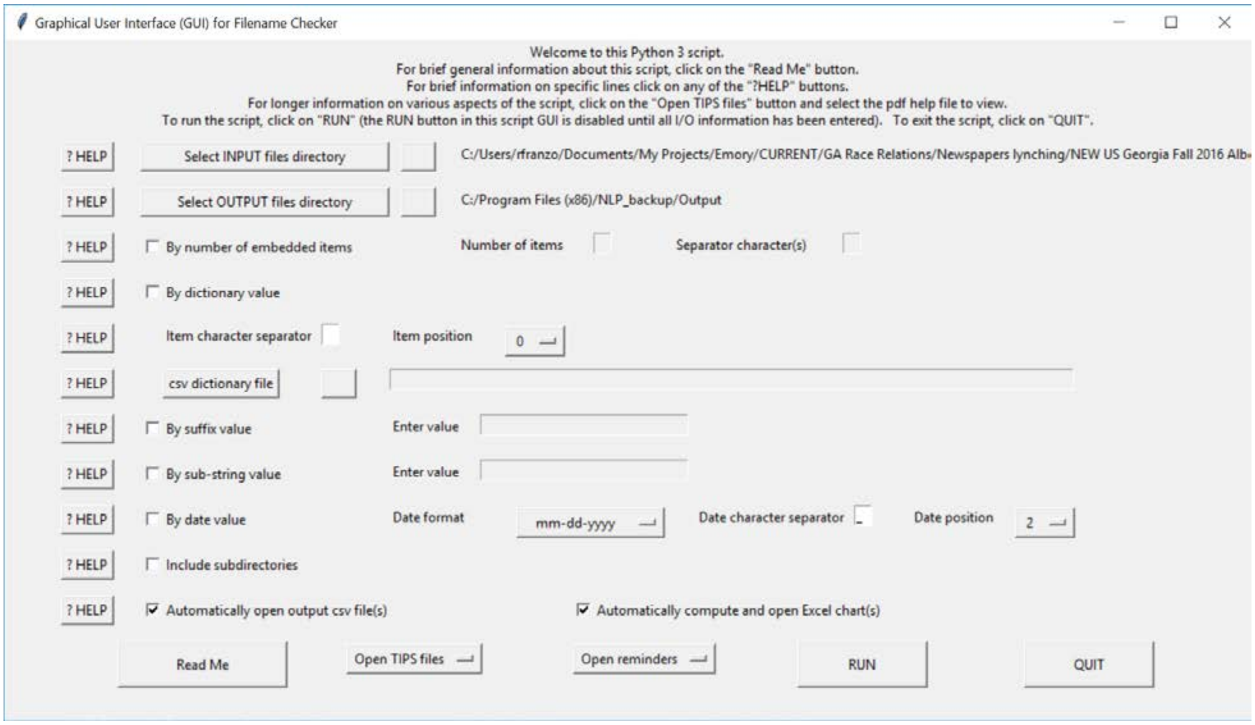
In a project on **in-depth interviews**, you may wish to embed the interviewer's and interviewee's names or IDs, the location of interview, the date, the topic.

In a project on **scientific journal articles**, you may wish to embed the journal's title (perhaps, year and volume), the article's authors, the title.

If you place each item of information embedded in the filename in a standard expected sequence (e.g., newspaper name, followed by date, followed by page number, followed by column number) and separate each item of information by special, unique characters (e.g. an underscore _), you can then extract information for processing (e.g., to answer such questions as: in which month are protest events most likely to be published? In which page? Does this type of information vary by type of newspaper? Who are the most prolific best-selling review writers? Do they vary by newspaper?).

For a data-extraction algorithm to function properly, the information must be consistently and properly embedded in the filename. For instance, the author's name must always be the first item in the filename. The functions behind the GUI allow precisely that: check the well-formedness of the filename.

Graphical User Interface (GUI) to check the well-formedness of filenames



Graphical User Interface (GUI) for Filename Checker

Welcome to this Python 3 script.
For brief general information about this script, click on the "Read Me" button.
For brief information on specific lines click on any of the "?HELP" buttons.
For longer information on various aspects of the script, click on the "Open TIPS files" button and select the pdf help file to view.
To run the script, click on "RUN" (the RUN button in this script GUI is disabled until all I/O information has been entered). To exit the script, click on "QUIT".

Select INPUT files directory: C:/Users/rfranzo/Documents/My Projects/Emory/CURRENT/GA Race Relations/Newspapers lynching/NEW US Georgia Fall 2016 Alb

Select OUTPUT files directory: C:/Program Files (x86)/NLP_backup/Output

□ By number of embedded items Number of items Separator character(s)

□ By dictionary value

Item character separator Item position 0

csv dictionary file

□ By suffix value Enter value

□ By sub-string value Enter value

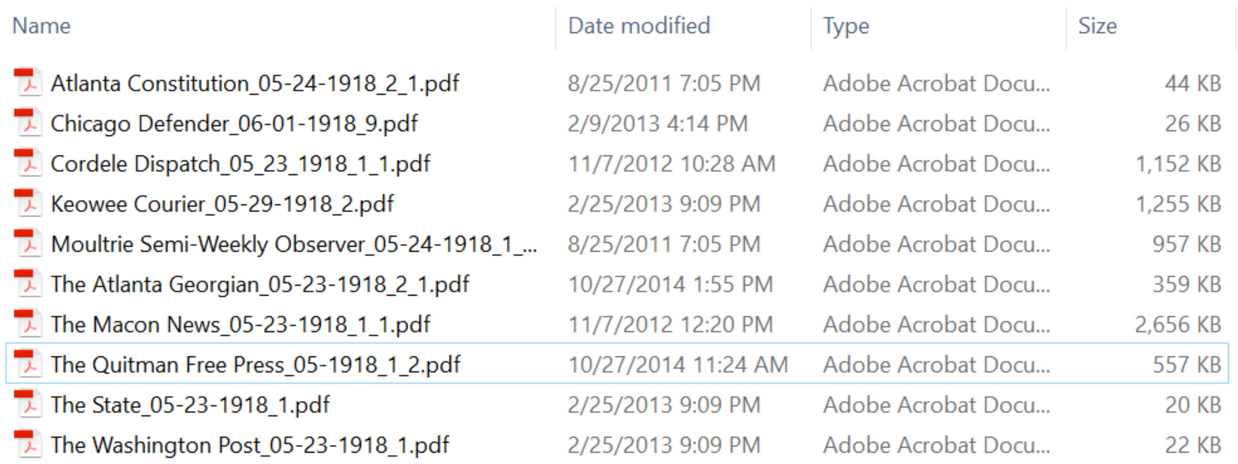
□ By date value Date format mm-dd-yyyy Date character separator - Date position 2

□ Include subdirectories

☑ Automatically open output csv file(s) ☑ Automatically compute and open Excel chart(s)

Figure 1 – Screenshot of the Graphical User Interface (GUI) for the NLP Suite filename checker
The screenshot of Fig. 1 of the Graphical User Interface (GUI) for the NLP Suite filename checker tool shows the range of checks that are available: By number of embedded items, By dictionary value, By suffix value, By sub-string value, By date value.

A list of erroneous filenames



Name	Date modified	Type	Size
Atlanta Constitution_05-24-1918_2_1.pdf	8/25/2011 7:05 PM	Adobe Acrobat Docu...	44 KB
Chicago Defender_06-01-1918_9.pdf	2/9/2013 4:14 PM	Adobe Acrobat Docu...	26 KB
Cordele Dispatch_05_23_1918_1_1.pdf	11/7/2012 10:28 AM	Adobe Acrobat Docu...	1,152 KB
Keowee Courier_05-29-1918_2.pdf	2/25/2013 9:09 PM	Adobe Acrobat Docu...	1,255 KB
Moultrie Semi-Weekly Observer_05-24-1918_1_...	8/25/2011 7:05 PM	Adobe Acrobat Docu...	957 KB
The Atlanta Georgian_05-23-1918_2_1.pdf	10/27/2014 1:55 PM	Adobe Acrobat Docu...	359 KB
The Macon News_05-23-1918_1_1.pdf	11/7/2012 12:20 PM	Adobe Acrobat Docu...	2,656 KB
The Quitman Free Press_05-1918_1_2.pdf	10/27/2014 11:24 AM	Adobe Acrobat Docu...	557 KB
The State_05-23-1918_1.pdf	2/25/2013 9:09 PM	Adobe Acrobat Docu...	20 KB
The Washington Post_05-23-1918_1.pdf	2/25/2013 9:09 PM	Adobe Acrobat Docu...	22 KB

Figure 2 – List of newspaper articles with embedded metadata but… with a number of errors

Even a quick look at the files listed in Fig. 2 reveals the range of errors users can make when saving newspaper articles that embed metadata in the filename in formatted ways.

None of this is a problem if all you are going to do is to click on a file and read it. It is not even a problem if the filename is imported into a csv file, for instance, in its entirety. It does become a problem, however, if the filename is then processed to extract specific metadata information. Then, a computer algorithm will have precise expectations of where and how to find information.

The various filename check algorithms can check any of these errors automatically.

Check filenames by number of embedded items

Most filenames in Fig. 2 contain the article page number after the date, but the page number is missing in The State_05-23-1918.pdf.

Some of the filenames in Fig. 2 include the column number (e.g., Atlanta Constitution_05-24-1918_2_1.pdf, Cordele Dispatch_05_23_1918_1_1.pdf), others do not (e.g., Chicago Defender_06-01-1918_9.pdf, Keowee Courier_05-29-1918_2.pdf).

The function behind the option “By number of embedded items” would allow us to list all the files that do not a specified number of embedded items separated by specific character(s).

Check a filename item against a dictionary value

Some of the filenames have newspaper names that are not 100% correct (e.g., Atlanta Constitution strictly speaking should be The Atlanta Constitution).

Given a csv dictionary file with a list of newspaper with their correct name, we could check filenames by the correspondence between an item in the filename and the value in the dictionary file. We need to pass several things to the algorithm that will check the data: 1. the csv dictionary file we want to use; 2. the character separator used to separate items in the filename (e.g., in New York Times_9-22-1918_4_3 the separator would be _; 2. the position in the filename of the item to be checked, for instance, position 1 for New York Times_9-22-1918_4_3 if you want to check New York Times.

Similarly, if you were checking filenames of literary fiction writers, you would need a dictionary with authors' names.

Check filenames by suffix or sub-string value

If all your files MUST contain the suffix TIPS_NLP_ this function can check that for you and list any file that does not comply.

Check filenames by date value

Fig. 2 shows that some dates embedded in the filenames are formatted incorrectly (e.g., Cordele Dispatch_05_23_1918_1_1.pdf) where date items are separated by _ instead of -; and in other cases, the month, or day, or year are missing (e.g., The Quitman Free Press_05-1918_1_2.pdf).

INPUT

In INPUT the scripts take a set of files in a directory (and subdirectories). No point in checking a single file. You can do that without an algorithm.

OUTPUT

In OUTPUT the scripts produce a 3-columns csv file with: proposed edited filename, if available; filename in error; source of error.