File Classifier (By NER)

Table of contents

The file structure expected by the algorithm

The algorithm provides a computationally costly solution to the problem of classifying unsorted files into groups. **The algorithm does not require that filenames have embedded dates**. But, the algorithm does expect to find files in the following directory structure;

1. a folder Y containing *m* subfolders (Z), each subfolder containing a variable set of documents that talk about the same topic Z;
2. a folder X containing a list of n unsorted documents each one of which needs to be placed in one of the *m* Z subfolders.

Graphically, this is what the file structure would look like:

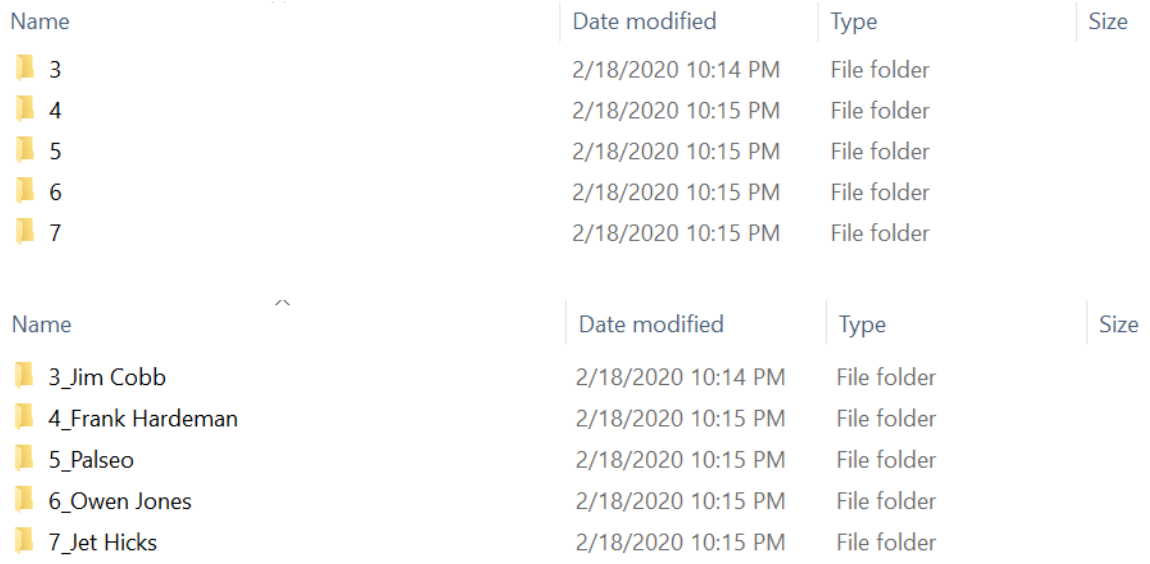
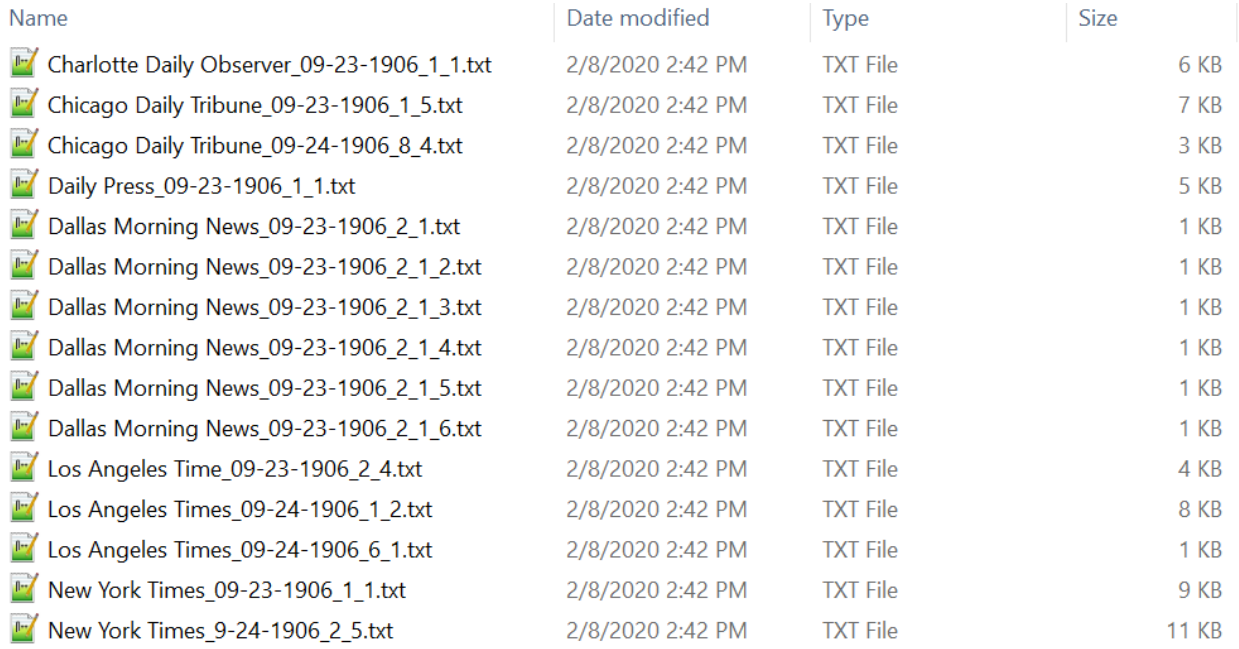


Figure 1 – Different representations of folder Y containing m subfolders (Z)

But whatever naming criteria one adopts, each subfolder contains the articles that describe a specific event.



Name	Date modified	Type	Size
Charlotte Daily Observer_09-23-1906_1_1.txt	2/8/2020 2:42 PM	TXT File	6 KB
Chicago Daily Tribune_09-23-1906_1_5.txt	2/8/2020 2:42 PM	TXT File	7 KB
Chicago Daily Tribune_09-24-1906_8_4.txt	2/8/2020 2:42 PM	TXT File	3 KB
Daily Press_09-23-1906_1_1.txt	2/8/2020 2:42 PM	TXT File	5 KB
Dallas Morning News_09-23-1906_2_1.txt	2/8/2020 2:42 PM	TXT File	1 KB
Dallas Morning News_09-23-1906_2_1_2.txt	2/8/2020 2:42 PM	TXT File	1 KB
Dallas Morning News_09-23-1906_2_1_3.txt	2/8/2020 2:42 PM	TXT File	1 KB
Dallas Morning News_09-23-1906_2_1_4.txt	2/8/2020 2:42 PM	TXT File	1 KB
Dallas Morning News_09-23-1906_2_1_5.txt	2/8/2020 2:42 PM	TXT File	1 KB
Dallas Morning News_09-23-1906_2_1_6.txt	2/8/2020 2:42 PM	TXT File	1 KB
Los Angeles Time_09-23-1906_2_4.txt	2/8/2020 2:42 PM	TXT File	4 KB
Los Angeles Times_09-24-1906_1_2.txt	2/8/2020 2:42 PM	TXT File	8 KB
Los Angeles Times_09-24-1906_6_1.txt	2/8/2020 2:42 PM	TXT File	1 KB
New York Times_09-23-1906_1_1.txt	2/8/2020 2:42 PM	TXT File	9 KB
New York Times_9-24-1906_2_5.txt	2/8/2020 2:42 PM	TXT File	11 KB

Figure 2 –Representation of folder X with *n* unsorted files

The way the algorithm works

The tool runs the Stanford CoreNLP with the annotator NER on each of the *n* documents in folder X and constructs an index of similarity based on social actors and NER values. It then computes a summary index of group values (based on social actors and NER values) for the files contained in each Z (*m* of them) we construct a summary index. Once the indices are computed, the algorithm compares the similarity value of each of the *n* file in X with the index of each Z subfolder to see if the processed document in X belongs to group Z.