```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Retrieve and store local representation of MMWR online with minimal processing,
which can include conversion to UTF-8 and basic parsing of HTML.

Update previous mirrors
  EID  2021-02-17 -> 2023-11-10
  MMWR 2022-05-27 -> 2023-11-11
  PCD  2021-02-17 -> 2023-11-10

@author: chadheilig

Begin with journal-specific dframe, containing a complete list of files.
MMWR, PCD, EID, PHR

_dframe pandas DataFrame
   base     URL from which href values were harvested
   href     hypertext reference (bs.a['href']); + base URL -> absolute URL
   url      absolute URL constructed from href and base URL
   path     path from absolute URL
   filename name of HTML file in hypertext reference
   mirror_path path on local mirror
   string   string from anchor element content
   level    in concatentated DataFrame, volume or article

Main products: local mirror of online archive as raw copy (bytes) and lightly
formatted UTF-8 (string), as well as pickle representations for ease of
continuity. See also 2_retrieve-and-store-experiments.py for timing trials.
"""

#%% Import modules and set up environment
# import from 0_cdc-corpora-header.py

os.chdir('/Users/cmheilig/cdc-corpora/_test')

#%% Retrieve journal-specific DataFrames from pickle files

## MMWR 2022-05-27 -> 2023-11-11
mmwr_dframe_20220527 = pickle.load(open('pickle-files/mmwr_dframe_20220527.pkl', 'rb'))
# (14751, 8)
mmwr_dframe_20231111 = pickle.load(open('pickle-files/mmwr_dframe_20231111.pkl', 'rb'))
# (15258, 8)
# mmwr_pdf_dframe = pickle.load(open("pickle-files/mmwr_pdf_dframe.pkl", "rb"))
# (2066, 9)

mmwr_dframe_20220527.level.value_counts(sort=False) # 14620 articles
mmwr_dframe_20231111.level.value_counts(sort=False) # 15122 articles

mmwr_url_20220527 = set(mmwr_dframe_20220527
                        .loc[mmwr_dframe_20220527['level']=='article', 'url']
                        .to_list())
mmwr_url_20231111 = set(mmwr_dframe_20231111
                        .loc[mmwr_dframe_20231111['level']=='article', 'url']
                        .to_list())
len(mmwr_url_20220527 - mmwr_url_20231111) # 0
```

```
len(mmwr_url_20231111 - mmwr_url_20220527) # 502

mmwr_update = mmwr_dframe_20231111[
    mmwr_dframe_20231111.url.isin(mmwr_url_20231111 - mmwr_url_20220527)]
# [502 rows x 8 columns]

## EID  2021-02-17 -> 2023-11-10
eid_dframe_20210217 = pickle.load(open('pickle-files/eid_dframe_20210217.pkl', 'rb'))
# (11504, 8)
eid_dframe_20231110 = pickle.load(open('pickle-files/eid_dframe_20231110.pkl', 'rb'))
# (13020, 8)

eid_dframe_20210217.level.value_counts(sort=False) # 11211 articles
eid_dframe_20231110.level.value_counts(sort=False) # 12691 articles

eid_url_20220527 = set(eid_dframe_20210217
                       .loc[eid_dframe_20210217['level']=='article', 'url']
                       .to_list())
eid_url_20231110 = set(eid_dframe_20231110
                       .loc[eid_dframe_20231110['level']=='article', 'url']
                       .to_list())
len(eid_url_20220527 - eid_url_20231110) # 0
len(eid_url_20231110 - eid_url_20220527) # 1480

eid_update = eid_dframe_20231110[
    eid_dframe_20231110.url.isin(eid_url_20231110 - eid_url_20220527)]
# [1480 rows x 8 columns]

## PCD  2021-02-17 -> 2023-11-10
pcd_dframe_20210217 = pickle.load(open('pickle-files/pcd_dframe_20210217.pkl', 'rb'))
# (4485, 8)
pcd_dframe_20231110 = pickle.load(open('pickle-files/pcd_dframe_20231110.pkl', 'rb'))
# (4772, 8)

pcd_dframe_20210217.level.value_counts(sort=False) # 3691 articles
pcd_dframe_20231110.level.value_counts(sort=False) # 3976 articles

pcd_url_20220527 = set(pcd_dframe_20210217
                       .loc[pcd_dframe_20210217['level']=='article', 'url']
                       .to_list())
pcd_url_20231110 = set(pcd_dframe_20231110
                       .loc[pcd_dframe_20231110['level']=='article', 'url']
                       .to_list())
len(pcd_url_20220527 - pcd_url_20231110) # 0
len(pcd_url_20231110 - pcd_url_20220527) # 285

pcd_update = pcd_dframe_20231110[
    pcd_dframe_20231110.url.isin(pcd_url_20231110 - pcd_url_20220527)]
# [285 rows x 8 columns]


#%% Set up local mirror directories for unprocessed HTML (b0)
MMWR_BASE_PATH_b0 = normpath(expanduser('~/cdc-corpora/mmwr_b0_update/'))
# MMWR_BASE_PATH_pdf = normpath(expanduser('~/cdc-corpora/mmwr_pdf/'))
EID_BASE_PATH_b0 = normpath(expanduser('~/cdc-corpora/eid_b0_update/'))
PCD_BASE_PATH_b0 = normpath(expanduser('~/cdc-corpora/pcd_b0_update/'))
```

```
# PHR_BASE_PATH_b0 = normpath(expanduser('~/cdc-corpora/phr_b0/'))

x = create_mirror_tree(MMWR_BASE_PATH_b0, calculate_mirror_dirs(mmwr_update.path))
# { key: (0 if val is None else len(val)) for (key, val) in x.items() }

x = create_mirror_tree(EID_BASE_PATH_b0, calculate_mirror_dirs(eid_update.path))
# { key: (0 if val is None else len(val)) for (key, val) in x.items() }

x = create_mirror_tree(PCD_BASE_PATH_b0, calculate_mirror_dirs(pcd_update.path))
# { key: (0 if val is None else len(val)) for (key, val) in x.items() }

# x = create_mirror_tree(PHR_BASE_PATH_b0, calculate_mirror_dirs(phr_dframe.path))
# { key: (0 if val is None else len(val)) for (key, val) in x.items() }

#%% Mirror unprocessed HTML from internet to local archive (www -> b0)

# mirror_raw_html(mmwr_dframe.url[200], MMWR_BASE_PATH_b0 + mmwr_dframe.mirror_path[200])

mmwr_sizes_b0 = [mirror_raw_html(url, MMWR_BASE_PATH_b0 + path, print_url = False)
                     for url, path in tqdm(zip(mmwr_dframe.url, mmwr_dframe.mirror_path),
                                           total=14226)]

# harvest only HTML for main page and years 2021-2022 (vol 70-71)
# level in ['home', 'series'] or
#     level == 'volume' and path contains 202[12] or
#     level == 'article' and path contains volumes/7[01]

_harvest = (mmwr_dframe.level.str.fullmatch('home|series') |
            (mmwr_dframe.level.str.fullmatch('volume') &
             mmwr_dframe.mirror_path.str.contains('202[12]')) |
            (mmwr_dframe.level.str.fullmatch('article') &
             mmwr_dframe.mirror_path.str.contains('volumes/7[01]')))
# sum(_harvest) # 584

mmwr_sizes_b0 = [mirror_raw_html(url, MMWR_BASE_PATH_b0 + path, print_url = False)
                     for url, path in tqdm(zip(mmwr_dframe.url.loc[_harvest],
                                               mmwr_dframe.mirror_path.loc[_harvest]),
                                           total=584)]
# 584/584 [04:08<00:00,  2.35it/s]
# sum([x==0 for x in mmwr_sizes_b0]) # retry those with 0 length
for j in tqdm(range(584)):
   if mmwr_sizes_b0[j] == 0:
      mmwr_sizes_b0[j] = mirror_raw_html(mmwr_dframe.url.loc[_harvest][j],
         MMWR_BASE_PATH_b0 + mmwr_dframe.mirror_path.loc[_harvest][j], timeout=5)
# pickle.dump(mmwr_sizes_b0, open('mmwr_sizes_b0.pkl', 'wb'))

_harvest = mmwr_dframe.filename.str.fullmatch('mm70(23a3|34a7).htm')
mmwr_sizes_b0_ = [mirror_raw_html(url, MMWR_BASE_PATH_b0 + path, print_url = False)
                     for url, path in zip(mmwr_dframe.url.loc[_harvest],
                                          mmwr_dframe.mirror_path.loc[_harvest])]

mmwr_pdf_sizes_b0 = [mirror_raw_html(url, MMWR_BASE_PATH_pdf + '/' + flnm, print_url =
             False)
                     for url, flnm in tqdm(zip(mmwr_pdf_dframe.url,

                                           total=2066)]
```

```
167 # 2066/2066 [04:08<00:00,  2.35it/s]
168 # sum([x==0 for x in mmwr_pdf_sizes_b0]) # retry those with 0 length
169 # href for volumes 46 and 47 erroneously point to FTP
170 # https://www.cdc.gov/mmwr/PDF/wk/mm4601.pdf
171 for iss in tqdm(list(range(4601,4653)) + [4654] + list(range(4701,4752)) + [4753]):
172     mirror_raw_html(f'https://www.cdc.gov/mmwr/PDF/wk/mm{iss}.pdf',
173                     MMWR_BASE_PATH_pdf + '/mm' + f'{iss}.pdf', print_url = False)
174
175 # mirror_raw_html(pcd_dframe.url[200], PCD_BASE_PATH_b0 + pcd_dframe.mirror_path[200])
176
177 pcd_sizes_b0 = [mirror_raw_html(url, PCD_BASE_PATH_b0 + path, print_url = False)
178                       for url, path in tqdm(zip(pcd_dframe.url, pcd_dframe.mirror_path),
179                                             total=3777)]
180 # sum([x==0 for x in pcd_sizes_b0]) # retry those with 0 length
181 for j in range(3777):
182    if pcd_sizes_b0[j] == 0:
183       pcd_sizes_b0[j] = mirror_raw_html(pcd_dframe.url[j],
184          PCD_BASE_PATH_b0 + pcd_dframe.mirror_path[j], timeout=5)
185 # sum([x==0 for x in pcd_sizes_b0]) # retry those with 0 length
186
187 # pickle.dump(pcd_sizes_b0, open('pcd_sizes_b0.pkl', 'wb'))
188
189 # mirror_raw_html(eid_dframe.url[200], EID_BASE_PATH_b0 + eid_dframe.mirror_path[200])
190
191 eid_sizes_b0 = [mirror_raw_html(url, EID_BASE_PATH_b0 + path, print_url = False, timeout =
191              8)
192                       for url, path in tqdm(zip(eid_dframe.url, eid_dframe.mirror_path),
193                                             total=11504)]
194 # sum([x==0 for x in eid_sizes_b0]) # retry those with 0 length
195 for j in range(11504):
196    if eid_sizes_b0[j] == 0:
197       eid_sizes_b0[j] = mirror_raw_html(eid_dframe.url[j],
198          EID_BASE_PATH_b0 + eid_dframe.mirror_path[j], timeout=5)
199 # pickle.dump(eid_sizes_b0, open('eid_sizes_b0.pkl', 'wb'))
200
201 # phr_sizes_b0 = [mirror_raw_html(url, PHR_BASE_PATH_b0 + path, timeout = 5)
202 #                       for url, path in zip(phr_dframe.url, phr_dframe.mirror_path[:142])]
203 # sum([x==0 for x in phr_sizes_b0]) # retry those with 0 length
204 # mirroring works for /pmc/issues [:142] but not /pmc/articles [142:]
205 # pickle.dump(phr_sizes_b0, open('phr_sizes_b0.pkl', 'wb'))
206
207
208 #%% Read unprocessed HTML from local mirror; store in pickle format
209
210 mmwr_html_b0 = [read_raw_html(MMWR_BASE_PATH_b0 + path)
211                       for path in tqdm(mmwr_dframe.mirror_path)]
212 # 14751/14751 [00:04<00:00, 2954.78it/s]
213 pickle.dump(mmwr_html_b0, open('mmwr_raw_html.pkl', 'wb'))
214
215 pcd_html_b0 = [read_raw_html(PCD_BASE_PATH_b0 + path)
216                       for path in tqdm(pcd_dframe.mirror_path)]
217 ## 3627/3627 [00:08<00:00, 444.38it/s]
218 # 3777/3777 [00:01<00:00, 2547.93it/s]
219 pickle.dump(pcd_html_b0, open('pcd_raw_html.pkl', 'wb'))
220
221 # [EID_BASE_PATH_b0 + path for path in eid_dframe.mirror_path
```

```
222 #    if not os.path.exists(EID_BASE_PATH_b0 + path)]
223
224 eid_html_b0 = [read_raw_html(EID_BASE_PATH_b0 + path)
225                       for path in tqdm(eid_dframe.mirror_path)]
226 ## 10922/10922 [00:20<00:00, 521.50it/s]
227 # 11504/11504 [00:06<00:00, 1784.81it/s]
228 pickle.dump(eid_html_b0, open('eid_raw_html.pkl', 'wb'))
229
230 #%% Set up local mirror directories for lightly processed HTML (u3)
231
232 MMWR_BASE_PATH_u3 = normpath(expanduser('~/cdc-corpora/mmwr_u3_update/'))
233 PCD_BASE_PATH_u3 = normpath(expanduser('~/cdc-corpora/pcd_u3_update/'))
234 EID_BASE_PATH_u3 = normpath(expanduser('~/cdc-corpora/eid_u3_update/'))
235 # PHR_BASE_PATH_u3 = normpath(expanduser('~/cdc-corpora/phr_u3/'))
236
237 x = create_mirror_tree(MMWR_BASE_PATH_u3, calculate_mirror_dirs(mmwr_update.path))
238 # { key: (0 if val is None else len(val)) for (key, val) in x.items() }
239
240 x = create_mirror_tree(EID_BASE_PATH_u3, calculate_mirror_dirs(eid_update.path))
241 # { key: (0 if val is None else len(val)) for (key, val) in x.items() }
242
243 x = create_mirror_tree(PCD_BASE_PATH_u3, calculate_mirror_dirs(pcd_update.path))
244 # { key: (0 if val is None else len(val)) for (key, val) in x.items() }
245
246 #%% Mirror unprocessed HTML to processed HTML (b0 -> u3)
247
248 # x = read_raw_html(MMWR_BASE_PATH_b0 + mmwr_dframe.mirror_path[548])
249 # mirror_raw_to_uni(MMWR_BASE_PATH_b0 + mmwr_dframe.mirror_path[548],
250 #                   MMWR_BASE_PATH_u3 + mmwr_dframe.mirror_path[548], 548)
251
252 for path in tqdm(mmwr_dframe.mirror_path):
253    mirror_raw_to_uni(MMWR_BASE_PATH_b0 + path, MMWR_BASE_PATH_u3 + path, counter=None)
254 # 14751/14751 [22:18<00:00, 11.02it/s]
255
256 for path in tqdm(pcd_dframe.mirror_path):
257    mirror_raw_to_uni(PCD_BASE_PATH_b0 + path, PCD_BASE_PATH_u3 + path, counter=None)
258 # 3777/3777 [02:52<00:00, 21.85it/s]
259
260 for path in tqdm(eid_dframe.mirror_path):
261    mirror_raw_to_uni(EID_BASE_PATH_b0 + path, EID_BASE_PATH_u3 + path, counter=None)
262 # 13800/13800 [24:20<00:00,  9.45it/s]
263
264 # Correct the codec for 1 file, as follows:
265 # mirror_raw_to_uni(MMWR_BASE_PATH_b0, MMWR_BASE_PATH_u3, mmwr_dframe.mirror)
266 # issue with 13874: Some characters could not be decoded, and were replaced with
266               REPLACEMENT CHARACTER.
267 # code 81 in code page 437: b'\x81'.decode('cp437')
268 # https://www.cdc.gov/mmwr/preview/mmwrhtml/ss4808a2.htm
269 mmwr_dframe.iloc[14408]
270 ss4808a2_raw_html = read_raw_html(MMWR_BASE_PATH_b0 + mmwr_dframe.mirror_path[14408])
271 x = html_to_unicode_b(ss4808a2_raw_html)
272 # issue is character \x81 at ss4808a2_raw_html[51903:51904]
273 # per https://doi.org/10.1016/S0145-305X(97)00030-X, should be ü '\u00fc'
274
275 # Try adding CP437 to UnicodeDammit attempts
276 x = UnicodeDammit(ss4808a2_raw_html, ['utf-8', 'windows-1252', 'cp437']) # succeeds
```

```
277 x.tried_encodings # [('utf-8', 'strict'), ('windows-1252', 'strict'), ('cp437', 'strict')]
278 x.original_encoding # 'cp437'
279
280 # Commit this exception and write to UTF-8 mirror
281 ss4808a2_uni_html = trim_leading_space_u(
282    html_prettify_u(
283       html_reduce_space_u(
284          UnicodeDammit(ss4808a2_raw_html, ['utf-8', 'windows-1252', 'cp437'])\
285             .unicode_markup)))
286 with open(MMWR_BASE_PATH_u3 + mmwr_dframe.mirror_path[14408], 'w') as file_out:
287    file_out.write(ss4808a2_uni_html)
288
289 #%% Read lightly processed HTML from local mirror; store in pickle format
290
291 mmwr_html_u3 = [read_uni_html(MMWR_BASE_PATH_u3 + path)
292                       for path in tqdm(mmwr_dframe.mirror_path)]
293 # 14751/14751 [00:09<00:00, 1623.35it/s]
294 pickle.dump(mmwr_html_u3, open('mmwr_uni_html.pkl', 'wb'))
295
296 pcd_html_u3 = [read_uni_html(PCD_BASE_PATH_u3 + path)
297                       for path in tqdm(pcd_dframe.mirror_path)]
298 # 3777/3777 [00:01<00:00, 3258.43it/s]
299 pickle.dump(pcd_html_u3, open('pcd_uni_html.pkl', 'wb'))
300
301 eid_html_u3 = [read_uni_html(EID_BASE_PATH_u3 + path)
302                       for path in tqdm(eid_dframe.mirror_path)]
303 # 11504/11504 [00:09<00:00, 1153.86it/s]
304 pickle.dump(eid_html_u3, open('eid_uni_html.pkl', 'wb'))
```