# Sitemap Analysis

## Sitemap Links Provided in robots.txt

1. https://www.aljazeera.com/sitemap.xml (Main Sitemap)  
2. https://www.aljazeera.com/news-sitemap.xml (News)  
3. https://www.aljazeera.com/sitemaps/article-archive.xml (Archived Articles)  
4. https://www.aljazeera.com/sitemaps/article-new.xml (New Articles)  
5. https://www.aljazeera.com/sitemaps/video-archive.xml (Archived Videos)  
6. https://www.aljazeera.com/sitemaps/video-new.xml (New Videos)

Analysis of Sitemap Files

- The sitemaps are well-organized and categorized by content type (news, articles, videos).  
- Each sitemap contains multiple <url> entries, typically including:  
 • <loc>: the direct link to the content.  
 • <lastmod>: the date of the last modification.  
- The presence of <lastmod> helps identify fresh content for targeted crawling.  
- The sitemaps allow crawlers to efficiently discover content without relying on deep or blind crawling.

import requests

import xml.etree.ElementTree as ET

import os

def fetch\_sitemap(url):

response = requests.get(url)

if response.status\_code == 200:

return response.text

else:

print(f"Failed to fetch sitemap: HTTP {response.status\_code}")

return None

def parse\_sitemap\_index(xml\_content):

sitemap\_urls = []

try:

root = ET.fromstring(xml\_content)

namespace = {'ns': 'http://www.sitemaps.org/schemas/sitemap/0.9'}

for sitemap in root.findall('ns:sitemap', namespace):

loc = sitemap.find('ns:loc', namespace)

if loc is not None:

sitemap\_urls.append(loc.text)

except ET.ParseError as e:

print(f"Error parsing sitemap XML: {e}")

return sitemap\_urls

def parse\_sitemap(xml\_content):

urls = []

try:

root = ET.fromstring(xml\_content)

namespace = {'ns': 'http://www.sitemaps.org/schemas/sitemap/0.9'}

for url in root.findall('ns:url', namespace):

loc = url.find('ns:loc', namespace)

if loc is not None:

urls.append(loc.text)

except ET.ParseError as e:

print(f"Error parsing sitemap XML: {e}")

return urls

def save\_urls\_to\_txt(urls, filename):

try:

with open(filename, "w", encoding="utf-8") as f:

for url in urls:

f.write(url + "\n")

print(f"Saved {len(urls)} URLs to {filename}")

except Exception as e:

print(f"Error saving URLs to file: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

sitemap\_index\_url = "https://www.aljazeera.com/sitemap.xml"

sitemap\_index\_content = fetch\_sitemap(sitemap\_index\_url)

if sitemap\_index\_content:

all\_urls = []

sitemap\_urls = parse\_sitemap\_index(sitemap\_index\_content)

if sitemap\_urls:

print(f"Found {len(sitemap\_urls)} sitemap URLs in sitemap index.")

for i, sm\_url in enumerate(sitemap\_urls):

print(f"Fetching URLs from sitemap {i+1}: {sm\_url}")

sm\_content = fetch\_sitemap(sm\_url)

if sm\_content:

urls = parse\_sitemap(sm\_content)

print(f"Found {len(urls)} URLs in sitemap {i+1}")

all\_urls.extend(urls)

else:

print("This is a regular sitemap (not an index).")

urls = parse\_sitemap(sitemap\_index\_content)

all\_urls.extend(urls)

os.makedirs("sitemaps\_files", exist\_ok=True)

save\_urls\_to\_txt(all\_urls, os.path.join("sitemaps\_files", "all\_sitemaps\_url.txt"))