**First Discussion**

**Email Subscription:**

Original Plan:

Extract content from email subscriptions 🡪 Follow link 🡪 Summarise content

Current Approach:

Direct tracking from Websites (e.g., ESGPedia, AlphaSense, Sphera) à Extract Titles ➝ Filter by ESG Relevance ➝ Group by Topic ➝ Full Article Extraction ➝ Summarise

Why we change the plan:

| Issue | Explanation |
| --- | --- |
| Access Limitation | Attempts (email\_fetcher.py) using Outlook IMAP with app password failed à email access is restricted by corporate admin, and cannot be bypassed for automation. |
| Redundancy | The ESG newsletters (e.g., ESGPedia) primarily link to web content that we are already tracking via the website. Extracting from emails adds no extra value and duplicates effort. |

**ESG Web Tracking:**

Objective: Build a pipeline to track, filter, and extract ESG-relevant content directly from key corporate sustainability websites (e.g., ESGPedia, AlphaSense, Sphera), enabling topic-based insights for SLNG.

What We’ve Done:

Pre-defined ESG Websites

↓

Extract Titles + URLs (Built a scraper web\_title\_url\_extraction.py to extract article titles from structured web pages, and filter out the ones that do not have further links and are not meaningful)

Sample output: A screenshot of a computer

AI-generated content may be incorrect.

↓

Cluster Similar Topics (Used sentence-transformer embeddings + KMeans to group similar articles into thematic clusters) – Optional

↓

Select Clusters of Interest (We temporarily looked into the most frequent topics) – Optional

↓

Extract Full Article Text (via Selenium Built a Selenium-based crawler web\_preview.py to load and extract full article text (including JavaScript-rendered pages).

↓

[→ Summarise, Tag ESG Themes, Score Impact, Improve Accuracy at the Same Time] ← (next step)

**Second Discussion**

ESG Content Extraction Workflow Update

**Improvement Made:**

To improve result quality, we significantly enhanced the Selenium-based crawler after identifying issues in the earlier extraction phase, specifically by excluding irrelevant or boilerplate content, such as cookie notices or error messages (“page not found”). This refinement required extensive investigation across multiple URL patterns, comparison of intermediate and final link behaviours, and manual verification of content mismatches.

After multiple iterations, we successfully implemented filtering logic to automatically exclude uninformative or redundant pages, resulting in a cleaner and more accurate dataset. The updated workflow not only improves downstream analysis but also ensures that time spent reviewing results is more efficient and insightful.

**Update on the Email Reading:**  
  
Due to practical restrictions in email-based scraping (e.g., accessing Outlook-secured inboxes and parsing diverse HTML email formats), we have shifted our tracking methodology to a LinkedIn-based approach.  
  
Current Process:  
1. We automatically access ESGpedia’s LinkedIn company page (With additional scripts to automatically log in with existing LinkedIn Credentials).  
2. The scraper (linkedin\_content.py) scrolls and extracts all recent posts. (Automatic, with the script).  
3. Posts containing ESG weekly summaries are identified based on recognisable content patterns (e.g., keywords like "Top ESG stories from this week").

4. The information is saved into a CSV file with the following fields:  
 - Post Date  
 - LinkedIn Post Content  
 - Link #1, Link #2, ..., Link #N  
 - Final URL (currently unfilled placeholder column for future enhancements)

5. For each identified summary post:  
 - The full LinkedIn text is extracted.  
 - All external links are resolved and stored.  
 - The resolved link content is not summarised yet, but stored for future processing. (Will put into the whole website list for further analysis).

Sample output:

A screenshot of a computer screen

AI-generated content may be incorrect.  
  
**Problems encountered: The manager insists on reading the email content. I try to register a personal email to subscribe and try to read from there. However, lots of the providers required subscription fees or only accept corporate email. Still promptly looking for mitigations.**

Third Discussion:

[→ Summarise, Tag ESG Themes, Score Impact, Improve Accuracy at the Same Time] ← (next step)

**Proposed Tags / Topics for LLM Training**

Carbon Markets / Pricing

* Carbon credits
* Carbon tax
* Carbon trading schemes
* Voluntary carbon markets
* Carbon offset projects
* Carbon neutrality targets

Climate Change & Mitigation

* Climate adaptation
* Climate risk disclosures
* Climate policy updates
* Sea level rise
* Ocean acidification
* Emissions reduction pathways
* Net-zero targets

ESG Regulations & Reporting

* ESG regulations (global, EU, Asia, US)
* ESG disclosures (CSRD, SFDR, TCFD, SEC)
* Mandatory vs voluntary reporting
* Sustainability reporting frameworks
* Greenwashing risks & cases
* ESG data quality and assurance

Corporate Sustainability

* Corporate net-zero pledges
* Corporate ESG initiatives
* Supply chain sustainability
* Green finance / sustainable finance
* ESG investing trends

Biodiversity & Nature

* Nature-based solutions
* Biodiversity credits
* Ecosystem services markets
* Conservation finance

Ocean & Marine Topics

* Ocean acidification
* Marine biodiversity
* Ocean health policy

Environmental Laws & Compliance

* EHS (Environment, Health & Safety) compliance
* Pollution regulations / Waste management / circular economy
* Water regulation

Social & Governance

* Human rights & supply chain due diligence
* ESG-linked executive pay
* Board ESG oversight
* Labor practices & diversity

General Market Trends

* Green technology & innovation
* ESG data providers & tools
* ESG litigation
* Climate-related financial risks

1. Hierarchical tagging (first by major category, then sub-topic). – tag\_testing.py
2. Add the summarisation. – summarisation\_testing.py

Sample output: (Initial trial. The layout and accuracy need to be checked.)A screenshot of a computer screen

AI-generated content may be incorrect.

https://esg-tracking-cnaiyad5zed2rfmsaplubq.streamlit.app/