XB Funding Donor Intelligence Tool

Overview Guide

Serena Yin

Table of Contents

[Background 3](#_Toc204337233)

[Methodology 3](#_Toc204337234)

[Folder Structure 3](#_Toc204337235)

[UN Resolutions 4](#_Toc204337236)

[Data Sources 4](#_Toc204337237)

[UN Digital Library API Call 4](#_Toc204337238)

[Files 4](#_Toc204337239)

[Step 1 - Utilizing the UNDL API 4](#_Toc204337240)

[Step 2 – AI Tagging 5](#_Toc204337241)

[Canada 6](#_Toc204337242)

[Data Sources 6](#_Toc204337243)

[Canada International Fund Assistance 6](#_Toc204337244)

[Files 6](#_Toc204337245)

[Step 1 – Scrape Data from Website 6](#_Toc204337246)

[Step 2 – Data Cleaning and Analysis 6](#_Toc204337247)

[Step 3 – Exporting Data 7](#_Toc204337248)

[Permanent Mission News Articles 7](#_Toc204337249)

[Files 7](#_Toc204337250)

[Step 1 – Scrape Data from Website 7](#_Toc204337251)

[Step 2 – Data Analysis and AI Tagging 8](#_Toc204337252)

[Permanent Mission Statements/Speeches 9](#_Toc204337253)

[Files 9](#_Toc204337254)

[Step 1 – Scrape Data from Website 9](#_Toc204337255)

[Step 2 – Data Analysis and AI Tagging 10](#_Toc204337256)

[United Arab Emirates 10](#_Toc204337257)

[Data Sources 10](#_Toc204337258)

[Foreign Aid Reports 10](#_Toc204337259)

[Files 10](#_Toc204337260)

[Step 1 – Scrape Data from Website 10](#_Toc204337261)

[Step 2 – Data Cleaning and Analysis 11](#_Toc204337262)

[Permanent Mission News Articles 12](#_Toc204337263)

[Files 12](#_Toc204337264)

[Step 1 – Scrape Data from Website 12](#_Toc204337265)

[Step 2 – Data Analysis and AI Tagging 13](#_Toc204337266)

[Permanent Mission Statements 13](#_Toc204337267)

[Files 13](#_Toc204337268)

[Step 1 – Scrape Data from Website 13](#_Toc204337269)

[Step 2 – Data Anaysis and AI Tagging 14](#_Toc204337270)

[Qatar 14](#_Toc204337271)

[Data Sources 14](#_Toc204337272)

[Permanent Mission News Articles 14](#_Toc204337273)

[Files 14](#_Toc204337274)

[Step 1 – Scrape Data from Website 14](#_Toc204337275)

[Step 2 – Data Analysis and AI Tagging 15](#_Toc204337276)

[Permanent Mission Statements 15](#_Toc204337277)

[Files 15](#_Toc204337278)

[Step 1 – Scrape Data from Website 15](#_Toc204337279)

[Step 2 – Data Anaysis and AI Tagging 16](#_Toc204337280)

[Power BI 16](#_Toc204337281)

[Data Sources 16](#_Toc204337282)

[Data model 16](#_Toc204337283)

[Navigation 17](#_Toc204337284)

# Background

The DOS-OUSG-PAS team is seeking to enhance its donor intelligence capabilities to better align with member state priorities and improve resource mobilization. Currently, there is a limited understanding of historical funding trends and donor priorities, which hinders strategic engagement. The project aims to address this gap by developing a web-scraping tool to collect and analyze historical contribution data and project funding information from various donor sources. This data-driven approach will support PAS in tailoring its outreach and proposals, identifying emerging donor interests, and aligning funding requests with both DOS strategic goals and Member State donor priorities.

This document will outline the overall methodology for the data intelligence collection process and how data collected were structured for donor priority analysis.

# Methodology

To enhance donor intelligence capabilities, a targeted and iterative approach was adopted, beginning with a focused analysis of three Member States: **Canada**, **United Arab Emirates (UAE)**, and **Qatar**. These countries were selected based on a combination of internal expertise and the need to validate results with known donor priorities (e.g., Canada) as well as to explore less familiar ones (e.g., UAE and Qatar).

The methodology involved collecting and analyzing both qualitative and quantitative data from a variety of publicly available sources, including:

* **UN Resolutions** sponsored or co-sponsored by the selected Member States
* **Statements and speeches** made at the General Assembly and Security Council
* **News articles** published by the Permanent Missions of the Member States
* **Foreign aid data** published by Ministries of Foreign Affairs (where available)

To process and interpret this data, artificial intelligence tools were employed to identify recurring themes, keywords, and social issues. Topic modeling and keyword frequency analysis were also used to surface donor priorities from unstructured text.

This approach allowed for the triangulation of insights across different data types and sources, helping to mitigate potential biases and data limitations. The findings from this initial focus group will inform the development of a scalable donor intelligence framework and guide future expansion to additional Member States.

Ultimately, the data and findings of the project were visualized in a PowerBI dashboard for easy access and analysis.

# Folder Structure

**Folder:** Donor Intelligence Tool

This folder contains all files, including Python scripts, Excel spreadsheets, etc. related to the XB Donor Intelligence Tool. The folder structure is as follows:

Donor Intelligence Tool/

│

├── XB Funding Donor Intelligence Tool Overview.docx # Overview of Tool and Project

├── XB Funding Donor Intelligence Tool Dashboard Guide.docx # Overview of Dashboard

├── XB Donor Intelligence Presentation.pptx # Slide Deck Presentation on Project

├── Social Media Scraping Research.docx # Research on Social Media Scraping

├── Donor Priorities Report.pbix # Power BI Dashboard Report

│

├── UN Digital Library Resolutions/

│ ├── API Documentation/ # Documentation on the UNDL API

│ └── UNDL.ipynb # Python Script to query UNDL data

│ └── undl\_consolidated.xlsx # All UN resolutions with tagging

│

├── Canada/

│ ├── Canada Mission News Articles Scraper.ipynb # Python Script to scrape news articles

│ └── Canada Mission Statements Scraper.ipynb # Python Script to scrape statements

│ └── Canada WebScraper.ipynb # Python Script for foreign aid data

│ └── Canada.xml # MARC XML file from UNDL API call

│ └── Canada\_international\_projects.xlsx # Scraped foreign aid data

│ └── Canada\_mission\_news.xlsx # Scraped and tagged news articles

│ └── Canada\_mission\_statements.xlsx # Scraped and tagged statements

│ └── Canada\_undl.xlsx # Tagged Canada UNDL resolutions

│

├── United Arab Emirates/

│ ├── Foreign Aid Reports/ # Folder with Foreign Aid Report PDFs

│ ├── UAE Mission News Articles Scraper.ipynb # Python Script to scrape news articles

│ └── UAE Mission Statements Scraper.ipynb # Python Script to scrape statements

│ └── UAE WebScraper.ipynb # Python scraper for foreign aid data

│ ├── UAE\_foreign\_aid.csv # Scrapes foreign aid data

│ └── Uae\_mission\_articles.xlsx # Scraped and tagged news articles

│ └── Uae\_mission\_statemens.xlsx # Scraped and tagged statements

│ └── United arab emirates.xml # MARC XML file from UNDL API call

│ └── United arab emirates\_undl.xlsx # Tagged UAE UNDL resolutions

│

├── Qatar/

│ └── Qatar Mission News Article Webscraper.ipynb # Python Script to scrape news articles

│ └── Qatar Statements Webscraper.ipynb # Python Script to scrape news articles

│ └── Qatar\_mission\_articles.xlsx # Scraped and tagged news articles

│ └── Qatar\_mofa\_statements.xlsx # Scraped and tagged statements

│ └── Qatar\_undl.xlsx # Tagged Qatar UNDL resolutions

# UN Resolutions

## Data Sources

* [UN Digital Library (API)](https://digitallibrary.un.org/?ln=en)

## UN Digital Library API Call

### Files

* Undl\_conslidated.xlsx
* Canada\_undl.xlsx
* United arab emirates\_undl.xlsx
* Qatar\_undl.xlsx
* UNDL.ipynb

### Step 1 - Utilizing the UNDL API

To utilize the UN Digital Library API, please register and create an account on the UN Digital Library.

Once you have access to the API, you will be able to generate the API key.

For further information on how to use the API, please refer to the “API Documentation” Folder.

To pull the UN Resolutions that each Member State has sponsored/co-sponsored, please change the corresponding parameters in the Python code:

* **apikey**: your personal API key generated from your UN Digital Library Account
* **member**: the Member State you want to pull records for
* **date**: the earliest data you want to pull records for
* **parameters**: the search query (do not recommend changing unless data outside of UN Resolutions is needed)

The API call will result in an MARC XML file, which is then processed to generate an Excel spreadsheet. At this point, the Excel spreadsheet would only contain data for one specific Member State. Please refer to the individual folders for each Member State to access this data

### Step 2 – AI Tagging

Once the results are saved in the Excel spreadsheet, resolutions are then tagged. The “raw” sheet is the raw data pulled from the API.

The “tagging (2022 – present)” sheet contains only resolutions from 2022 to the time the data was pulled. Columns that contained internal codes, such as *datafield\_035\_a* were then removed from the sheet as these columns would not be helpful in determining the main topics/themes each social issue addressed. Additional columns were then generated, such as *Sponsorship Type, Adoption Type, AI Tagging – Primary Topic, AI Tagging – Subtopic.* Sponsorship Type is determined based on whether or not the Member State was a sponsor or a co-sponsor of a specific resolution. Adoption Type was then determined by whether or not the draft has an official resolution number (*datafield\_993\_a)*.

For CoPilot to have an easier time processing the data, resolutions were divided into 2 categories: adopted and unadopted. This was done by filtering the *Adoption Type* column. The process for tagging was similar for both groups of resolutions, and the raw outputs from CoPilot are saved in the “AI Tagging (adopted)” and “AI Tagging (unadopted)” sheets, respectively.

The Primary Topic was generated first. This column should capture the general or overall theme/social issue addressed by the resolution. An example prompt used is as follows:

*“This is an excel spreadsheet with some data on UN resolutions. Utilizing datafield 650a and 991c, for each resolution number in datafield 993a, please search up what the resolution is about and return the top theme or social issue the resolution addresses. The goal is to categorize the resolutions based on topic to understand what UN Member States are interested in sponsoring. Please return the results in columns that can be pasted into Excel.*

*(attached Excel file)”*

After the primary topic was generated and added to the Excel sheet, the Excel was then fed into CoPilot again to generate a more specific topic within the primary topic. An example prompt used is as follows:

*“This is an excel spreadsheet with some data on UN resolutions. Utilizing datafield 650a and 991c, for each resolution number in datafield 993a, please use online summaries on what the resolution is about and return the top subtopic the resolution addresses that is within the domain of the Top Theme column. For example, a theme may be public transportation and the subtopic would be bicycles. The goal is to categorize the resolutions based on topic to understand what UN Member States are interested in sponsoring. Please return the results in columns that can be pasted into Excel.*

*(attached Excel file)”*

The tagging process may take some trial and error depending on how accurately CoPilot is able to tag each record. For example, CoPilot may not tag each resolution or tag some resolutions with placeholder values like “other” or “general”. As such, you should look at the results from each prompt and give feedback to CoPilot so it can correct its response from previous prompts to get a more accurate output.

# Canada

## Data Sources

* [Canadian International Fund Assistance Projects Website](https://w05.international.gc.ca/projectbrowser-banqueprojets/filter-filtre)
* [Canada and the United Nations – News](https://www.international.gc.ca/world-monde/international_relations-relations_internationales/un-onu/news-nouvelles/index.aspx?lang=eng)
* [Canada and the United Nations – Declarations and Statements](https://www.international.gc.ca/world-monde/international_relations-relations_internationales/un-onu/statements-declarations/index.aspx?lang=eng)

## Canada International Fund Assistance

### Files

* Canada Webscraper.ipynb
* Canada\_international\_projects.xlsx

### Step 1 – Scrape Data from Website

The first phase of the workflow involves automated data extraction from the International Assistance Projects website using Selenium and ChromeDriver. This setup enables dynamic interaction with the website’s interface, allowing the script to navigate through paginated content and extract project-level data.

The scraper is configured to filter for projects with a start date on or after January 1, 2019. The extracted data is downloaded in CSV format. From this file, all columns up to and including the “Budget” field are parsed and loaded into a structured Python DataFrame for further processing.

### Step 2 – Data Cleaning and Analysis

Once the raw data is ingested, the second phase focuses on cleaning and analyzing the dataset. The analysis emphasizes identifying thematic trends and donor priorities using natural language processing (NLP) techniques applied to the combined text from the “Project Description” and “Expected Results” fields.

Key steps include:

* **Stop Word Refinement**: A custom stop word list was developed by manually reviewing the top 20 most frequent terms and excluding non-thematic or administrative language.
* **Topic Modeling (LDA)**:
  + An initial model extracts the top 5 topics, each represented by 3 keywords, without using bigrams.
  + A second model incorporates bigrams and extracts the top 3 topics, each with 5 keywords. These are visualized as word clouds and integrated into the Power BI dashboard.
  + The topic-word associations are exported to an Excel sheet titled *“Topic\_WordCloud”* for dashboard integration.
* **Keyword Frequency Analysis**:
  + High-frequency keywords are identified after stop word removal.
  + TF-IDF (Term Frequency–Inverse Document Frequency) is used to highlight terms with high contextual importance.
* **DAC Sector Classification**:
  + Projects are categorized using DAC (Development Assistance Committee) sector codes, which are standardized by the OECD to classify Official Development Assistance (ODA) by purpose.
* **Policy Marker Tags (Deprecated)**:
  + Although the dataset includes policy marker tags indicating primary and secondary thematic relevance, these were excluded from final analysis due to inconsistencies and limited reliability.
* **Geographic Allocation Analysis**:
  + Projects are analyzed by country and region using the “Country” and “Region” fields, which may contain multiple tags per project. This enables mapping of Canada’s foreign assistance distribution across geographic areas.

### Step 3 – Exporting Data

The final phase involves exporting two versions of the cleaded dataset for use in PowerBI:

* A **wide-format** (*Canada\_international\_projects\_wide)* containing one-hot encoded fields
* A **long-format** (*Canada\_international\_projects\_long)* where multi-class fields such as DAC Sector, Country, and Region are normalized into individual rows for flexible visualization.

## Permanent Mission News Articles

### Files

* Canada\_mission\_news.xlsx
* Canada Mission News Articles Scraper.ipynb

### Step 1 – Scrape Data from Website

The objective was to extract official news articles from the Global Affairs Canada website, specifically from the section dedicated to UN-related updates.

**Targeting and Filtering**

The scraper targets a structured list of articles on the website. Each article entry includes a title, hyperlink, and publication date. To ensure relevance, only articles published on or after January 1, 2018, were included in the dataset.

**HTML Parsing and Data Extraction**

The HTML content of the page is parsed using a DOM parser to locate and extract relevant elements. For each article, the scraper retrieves:

* The article title
* The full URL (handling both absolute and relative links)
* The publication date, which is parsed and validated

Articles that do not meet the date criteria or contain malformed data are excluded from further processing.

Content Retrieval and Cleaning

For each valid article URL, the scraper fetches the full page content and isolates the main body of the article. It then extracts all paragraph elements, applying the following cleaning rules:

* Paragraphs that contain only formatting tags (e.g., bold or italic text) are discarded
* Very short paragraphs, which are likely to be boilerplate or non-substantive, are filtered out
* Remaining paragraphs are stripped of HTML tags and concatenated into a clean, readable text block

This ensures that the final dataset contains only meaningful, content-rich text suitable for thematic analysis.

**Data Structuring and Export**

The collected data—including article titles, URLs, publication dates, and cleaned full text—is compiled into a structured tabular format using a data processing library. The final dataset is exported to an Excel file for archival and further analysis.

### Step 2 – Data Analysis and AI Tagging

To extract thematic insights from qualitative text data, a multi-step natural language processing (NLP) pipeline was implemented using both scikit-learn and Gensim libraries. The goal was to identify recurring topics and high-importance keywords from Canadian news articles.

**Stopword Customization**

A custom stopword list was created by combining the default English stopwords from scikit-learn with a manually curated list of domain-specific terms. These included high-frequency but semantically uninformative words such as “minister,” “united,” “canada,” and “international.” This step ensured that the topic modeling process focused on meaningful thematic content rather than administrative or generic language. Stopwords were determined through a manual iterative process, taking out words in the results that did not contain meaningful thematic content.

**Topic Modeling Without Bigrams (LDA via scikit-learn)**

A bag-of-words document-term matrix was constructed using CountVectorizer, with filtering thresholds to exclude extremely common or rare terms. Latent Dirichlet Allocation (LDA) was then applied to this matrix to extract five latent topics, each represented by the top three most probable words. This initial model provided a baseline understanding of dominant themes without considering multi-word expressions.

**Text Preprocessing and Bigram Modeling**

To capture more nuanced themes, the text corpus was preprocessed by:

* Lowercasing all text
* Removing non-alphabetic characters
* Tokenizing and filtering out stopwords and short tokens

A bigram model was then trained using Gensim’s Phrases module to identify and preserve common two-word expressions (e.g., “climate change,” “gender equality”). These bigrams were applied to the tokenized corpus to enhance semantic richness.

**Topic Modeling With Bigrams (LDA via Gensim)**

Using the bigram-enhanced corpus, a dictionary and bag-of-words representation were created. A second LDA model was trained to extract three topics, each consisting of the top five most probable terms. This model was configured with automatic hyperparameter tuning (alpha='auto') and multiple passes to improve convergence.

Each topic from the Gensim LDA model was visualized using word clouds, where word size corresponds to the probability weight within the topic. These visualizations were generated using the WordCloud library and plotted with Matplotlib. The resulting graphics were used to support thematic interpretation and were integrated into the Power BI dashboard.

The top five words and their associated probabilities for each topic were exported to an Excel sheet titled “Topic\_WordCloud”. This structured output supports downstream integration with visualization tools and facilitates stakeholder review.

**Keyword Frequency and TF-IDF Analysis**

To complement topic modeling, two additional keyword extraction methods were applied:

* **Raw Frequency**: The bag-of-words matrix was used to compute total word counts across the corpus, identifying the most frequently occurring terms.
* **TF-IDF (Term Frequency–Inverse Document Frequency):** A TF-IDF matrix was generated to highlight terms that are both frequent and distinctive across documents. The top-ranked terms by TF-IDF score were interpreted as high-importance thematic indicators.

**AI Tagging**

CoPilot was utilized to determine 2 themes for each news article. The title, url, and content of each news article were uploaded (found in the “raw” sheet of the Excel file) and CoPilot was asked to scan through each news article to determine the top 2 themes/social issues that each article addressed.

Results of the AI tagging were then spot checked for accuracy and consistency. The results were then melted into a long data format (*tagging* sheet) to be visualized in the PowerBI Dashboard.

## Permanent Mission Statements/Speeches

### Files

* Canada Mission Statements Scraper.ipynb
* Canada\_mission\_statements.xlsx

### Step 1 – Scrape Data from Website

To extract official statements and speeches from Canada’s Permanent Mission website, a dynamic web scraping pipeline was developed using Selenium and BeautifulSoup. This approach was necessary due to the JavaScript-rendered content on the target page, which cannot be accessed through static HTTP requests alone.

**Content Retrieval**

The scraper uses Selenium WebDriver with Chrome to simulate a browser session and load the full content of the target webpage. A delay is introduced to ensure that all JavaScript-rendered elements are fully loaded before parsing. This is critical for accessing the <section> blocks nested within the .org-news container, which hold the relevant article metadata.

**HTML Parsing and Article Metadata Extraction**

Once the page source is retrieved, BeautifulSoup is used to parse the HTML. The scraper locates all <li> elements within each <section> block, extracting:

* Title: The visible text of the anchor tag
* URL: The hyperlink to the full article, normalized to an absolute URL if necessary
* Date: The publication date, parsed using dateutil.parser to accommodate various date formats

Articles are filtered to include only those published on or after January 1, 2018, ensuring the dataset reflects recent diplomatic activity and sufficient sample size for analysis.

**Article Content Extraction**

For each article URL, a secondary request is made to retrieve the full HTML content. The scraper targets the <main> tag, which typically contains the article body. Paragraphs are extracted and cleaned using the following criteria:

* Paragraphs containing only formatting tags (e.g., <strong>, <em>) are excluded
* Very short or boilerplate paragraphs (less than 30 characters) are filtered out
* Remaining paragraphs are stripped of HTML and concatenated into a clean, readable text block

This ensures that the final dataset contains only substantive content suitable for thematic analysis.

**Data Structuring and Export**

The extracted metadata and cleaned article text are compiled into a structured DataFrame. This dataset is then exported to an Excel file (canada\_mission\_statements.xlsx) under the sheet name "raw", enabling further analysis and integration with visualization tools such as Power BI.

### Step 2 – Data Analysis and AI Tagging

The scraped statements underwent a similar data analysis and AI tagging process as the news articles. Please refer to the [Canada Permanent Mission News Articles](#_Step_2_–) section for more information.

# United Arab Emirates

## Data Sources

* [UAE Ministry of Foreign Affairs Foreign Aid Reports](https://www.mofa.gov.ae/en/The-Ministry/UAE-International-Development-Cooperation/Annual-Foreign-Aid-Report)
* [UAE Permanent Mission News Articles](https://uaeun.org/?s=&combined-search-post-types=press_release)
* [UAE Permanent Mission Speeches and Statements](https://uaeun.org/?s=&combined-search-post-types=press_release)

## Foreign Aid Reports

### Files

* UAE Webscraper.ipynb
* Foreign Aid Reports Folder
* UAE\_foreign\_aid.csv

### Step 1 – Scrape Data from Website

There were four main phases of the workflow to collect data from the published UAE Foreign Reports. This includes downloading the reports, collecting the data within the pdfs files of the reports, and consolidating information across all reports before being exported. Primarily, the following tables were collected:

* UAE Foreign Assistance and ODA to Least Developed Countries (LDCs)
* UAE Foreign Assistance and ODA to Land-locked Developing Countries (LLDCs)
* UAE Foreign Assistance and ODA to Small Island Developing States (SIDs)

It is important to note that there wasn’t an English PDF version of the 2022 Foreign Aid report on the UAE MoFA’s website, so a workflow using Power Automate was used to extract the data from the image version of the report.

**Web Scraping of Report Metadata**

A custom scraper was built to extract metadata from the UAE Ministry of Foreign Affairs’ Annual Foreign Aid Reports webpage. The scraper identifies report cards containing:

* Title and year of publication
* View and download links for each report

Only reports from 2019 onward are retained. Relative URLs are normalized to absolute links, and the final output is a structured list of reports sorted by year.

**PDF Table Extraction**

Each report PDF is processed using a custom function that identifies and extracts tabular data based on predefined section markers (e.g., “Appendix 1”, “Appendix 2”, “Appendix 3”). The parser:

* Skips irrelevant pages (e.g., table of contents)
* Divides each page into four vertical columns to approximate table layout
* Filters out noise such as headers, page numbers, and boilerplate text
* Detects and assembles category-value pairs using heuristics for numeric and continuation lines
* Stops parsing at the “Grand Total” marker to avoid footer artifacts

The result is a dictionary of cleaned text blocks corresponding to each appendix section.

For the 2022 Foreign Aid Report, since the English version of the report was uploaded to the website as individual images rather than a PDF, each relevant image was manually downloaded for scraping in Power Automate. Please refer to this [Power Automate flow](https://make.powerautomate.com/environments/Default-0f9e35db-544f-4f60-bdcc-5ea416e6dc70/flows/shared/58cdf22b-9f07-4bba-913c-e4e4250b42a3/details) for more details. The output of the flow is a txt file, which is then imported into the Python Script for cleaning and other steps to aggregate the data with data from other reports.

**Text Normalization and Country Matching**

Extracted text is converted into structured tabular data using a line-by-line parser. Each line is split into:

* A country or category name
* Two numeric values (e.g., foreign assistance and ODA)

To ensure consistency in country naming:

* A set of **manual overrides** corrects common naming discrepancies (e.g., “Ivory Coast” → “Côte d'Ivoire”)
* **Fuzzy matching** via the rapidfuzz library aligns names with ISO-standard country names from the pycountry database

Each entry is classified as either a **country-level total** or a **sub-category** under a previously identified country. Numeric values are cleaned and converted to floating-point format.

**Aggregation and Final Dataset Construction**

For each report and associated PDF:

* Extracted tables are parsed and converted into temporary DataFrames
* Valid entries are appended to a master DataFrame with standardized columns:
* Year
* Country
* Funding Category
* Foreign Assistance (USD)
* ODA (USD)

The final dataset supports analysis of UAE’s foreign aid allocations by country, sector, and year.

### Step 2 – Data Cleaning and Analysis

To extract thematic insights from data, data processing and preliminary analysis were conducted prior to reporting insights into PowerBI.

**Data Cleaning**

* Duplicate Removal: All duplicate rows are dropped to ensure data integrity.
* Null Filtering: Rows missing both Country and Funding Category are removed.
* Exclusion of Totals: Entries labeled as “Grand Total” or “Total” are excluded to avoid double-counting.
* Inconsistent category labels are standardized (e.g., “Energy Generation and” → “Energy Generation and Supply”; “Social” → “Social Services”).
* Data Type Conversion: Financial columns (Foreign Assistance (USD) and ODA (USD)) are converted to numeric types, with invalid entries coerced to NaN.

The cleaned dataset was then saved as a UTF-8 encoded CSV file (UAE\_foreign\_aid.csv).

**Data Analysis**

Top 10 Recipient Countries

* The top 10 countries receiving the highest cumulative foreign assistance are identified.
* A pivot table is created with Year as the index and Country as columns.
* A stacked bar chart displays annual foreign assistance flows to these countries.

Top 10 Funding Categories

* The top 10 funding categories by total foreign assistance are identified.
* A pivot table is created with Year as the index and Funding Category as columns.
* A stacked bar chart visualizes how funding priorities have shifted across categories over time

These visualizations provide a high-level overview of UAE’s aid distribution patterns and thematic focus areas, supporting both strategic insights and donor intelligence reporting.

## Permanent Mission News Articles

### Files

* Uae\_mission\_articles.xlsx
* UAE Mission News Articles Scraper.ipynb

### Step 1 – Scrape Data from Website

The objective was to extract official news articles from the UAE Permanent Mission website.

**Targeting and Filtering**

The scraping process targets the official website of the UAE Mission to the United Nations. Specifically, it focuses on press releases published under the press\_release content type. Pagination is handled via a query parameter (paged), allowing the scraper to iterate through multiple result pages.

To ensure relevance, a cutoff date of January 1, 2022 is applied. Articles published before this date are excluded from the dataset. The scraper halts automatically once it encounters an article older than the cutoff, optimizing performance and avoiding unnecessary requests.

**HTML Parsing and Data Extraction**

Each page of search results is parsed using BeautifulSoup. For every article entry, the scraper extracts:

* **Title:** Located within a <strong> tag with class title
* **URL**: Filtered to include only links that match the press release pattern
* **Publication Date**: Parsed from the <time> tag using a consistent date format

If all three elements are present and the article is within the valid date range, the entry is added to the dataset.

For each valid article URL, the scraper fetches the full page content and extracts the main body text. This is done by locating <div class="container"> blocks and collecting all <p> tags within them. The paragraph texts are concatenated to preserve structure and readability.

**Data Structuring and Export**

All extracted entries are compiled into a structured list containing:

* Article title
* URL
* Publication date
* Full text content

This list is converted into a Pandas DataFrame for further processing. The final dataset is exported to an Excel file named uae\_mission\_articles.xlsx, with the raw data stored in a sheet titled “raw”.

### Step 2 – Data Analysis and AI Tagging

The scraped UAE News Articles underwent a similar data analysis and AI tagging process as the Canada News Articles. Please refer to the [Canada Permanent Mission News Articles](#_Step_2_–) section for more information.

## Permanent Mission Statements

### Files

* Uae\_mission\_statements.xlsx
* UAE Mission Statements Scraper.ipynb

### Step 1 – Scrape Data from Website

To support thematic analysis of the UAE’s diplomatic engagement at the United Nations, a custom web scraping pipeline was developed to collect official statements published on the UAE Mission’s website.

**Targeting and Filtering**

The scraper targets the UAE Mission website, specifically the section containing official statements. The query parameters are configured to:

* Filter for content type: statement
* Paginate through results using the paged parameter

To ensure relevance, a cutoff date of January 1, 2022 is applied. Articles published before this date are excluded. The scraper automatically halts once it encounters a statement older than the cutoff.

**HTML Parsing and Data Extraction**

Each page of results is parsed using BeautifulSoup. For each statement entry, the scraper extracts:

* **Title**: From a <strong> tag with class title
* **URL**: From anchor tags pointing to individual statement pages
* **Publication Date**: From the <time> tag, parsed into a datetime object

Only entries that include all three elements and meet the date criteria are processed further.

For each valid statement URL, the scraper fetches the full page content and extracts the main body text. This is done by locating <div class="container"> blocks and collecting all <p> tags within them. Paragraphs are concatenated to preserve structure and readability.

**Data Structuring and Export**

Each statement is stored as a dictionary containing:

* Title
* URL
* Publication date
* Full text content

All entries are compiled into a Pandas DataFrame. The final dataset is exported to an Excel file named uae\_mission\_statements.xlsx, with the raw data saved in a sheet titled “raw”.

### Step 2 – Data Anaysis and AI Tagging

The scraped UAE Mission Statements underwent a similar data analysis and AI tagging process as the Canada News Articles. Please refer to the [Canada Permanent Mission News Articles](#_Step_2_–) section for more information.

# Qatar

## Data Sources

* [Qatar Permanent Mission News Articles](https://ny.mission.qa/en/media/news)
* [Qatar Ministry of Foreign Affairs Statements](https://mofa.gov.qa/en/latest-articles/statements)

## Permanent Mission News Articles

### Files

* Qatar\_mission\_articles.xlsx
* Qatar Mission News Article Webscraper.ipynb

### Step 1 – Scrape Data from Website

To support analysis of Qatar’s donor priorities, a custom web scraping pipeline was developed to extract news articles from the official website of the Permanent Mission of Qatar to the UN. The pipeline collects article metadata and full text content for all news items published within the last two years.

**Targeting and Filtering**

The scraper targets the Qatar Mission’s news section at ny.mission.qa/en/media/news. The scraping process begins at the first page and follows pagination links labeled “Next” to traverse through the archive.

A cutoff date of two years prior to the current date is applied. Articles published before this threshold are excluded to ensure the dataset reflects recent diplomatic activity. The scraper halts automatically once it encounters a page where all articles are older than the cutoff.

**HTML Parsing and Data Extraction**

Using Selenium and BeautifulSoup, the scraper:

* Loads the initial page and repeatedly clicks the **“**Load More**”** button to reveal additional statements
* Extracts metadata from each visible article block:
  + **Title**: From the <h4 class="news-title"> tag
  + **Date**: From the <p class="news-publication-date"> tag, parsed into a standardized format
  + **URL**: From the anchor tag within the article block

For each valid article, the scraper fetches the full page content and extracts the main body text from the <div class="news-detail-content"> container. It prioritizes <p> tags for structured content but falls back to <div> tags when necessary. The first element (typically the date) is skipped, and the remaining text is concatenated into a single string.

**Data Structuring and Export**

Each article is stored as a dictionary containing:

* Title
* URL
* Publication date
* Full text content

All entries are compiled into a Pandas DataFrame and exported to an Excel file named qatar\_mission\_articles.xlsx, with the raw data saved in a sheet titled raw.

### Step 2 – Data Analysis and AI Tagging

The scraped Qatar News Articles underwent a similar data analysis and AI tagging process as the Canada News Articles. Please refer to the [Canada Permanent Mission News Articles](#_Step_2_–) section for more information.

## Permanent Mission Statements

### Files

* Qatar\_mofa\_statements.xlsx
* Qatar Statements Webscraper.ipynb

### Step 1 – Scrape Data from Website

To support analysis of Qatar’s donor priorities, a Selenium-based scraping pipeline was developed to extract official statements from the Qatar Ministry of Foreign Affairs website. The scraper collects article metadata and full text content for all statements published within the last two years.

**Targeting and Filtering**

The scraper targets the “Statements” section of the MOFA website. The page uses dynamic content loading via a “Load More” button, which requires browser automation to access additional entries.

A cutoff date of two years prior to the current date is applied. Once the scraper encounters a statement older than this threshold, it stops collecting further entries.

**HTML Parsing and Data Extraction**

Using Selenium and BeautifulSoup, the scraper:

* Loads the initial page and repeatedly clicks the “Load More” button to reveal additional statements
* Extracts metadata from each visible article block:
  + Title: From the <h4 class="news-title"> tag
  + Date: From the <p class="news-publication-date"> tag, parsed into a standardized format
  + URL: From the anchor tag within the article block

For each valid article, the scraper fetches the full page content and extracts the main body text from the <div class="news-detail-content"> container. It prioritizes <p> tags for structured content but falls back to <div> tags when necessary. The first element (the date) is skipped, and the remaining text is concatenated into a single string.

**Data Structuring and Export**

Each statement is stored as a dictionary containing:

* Title
* URL
* Publication date
* Full text content

All entries are compiled into a Pandas DataFrame and exported to an Excel file named qatar\_mofa\_statements.xlsx, with the raw data saved in a sheet titled “raw”.

### Step 2 – Data Anaysis and AI Tagging

The scraped UAE Mission Statements underwent a similar data analysis and AI tagging process as the Canada News Articles. Please refer to the [Canada Permanent Mission News Articles](#_Step_2_–) section for more information.

# Power BI

All results and data from the previous sections were then imported into PowerBI to be visualized. The PowerBI dashboard currently has 6 different pages, separated by Member State and country

## Data Sources

* Undl\_consolidated.xlsx
* Canada\_international\_projects.xlsx
* Canada\_mission\_news.xlsx
* Canada\_mission\_statements.xlsx
* Uae\_foreign\_aid.csv
* Uae\_mission\_articles.xlsx
* Uae\_mission\_statements.xlsx
* Qatar\_mofa\_statements.xlsx
* Qatar\_mission\_articles.xlsx

## Data model

The following table outlines the naming conventions of the Data Model in PowerBI and their corresponding data sources:

|  |  |
| --- | --- |
| Data Source Name | PowerBI Data Model Table Name |
| Undl\_consolidated.xlsx |  |
| *consolidated\_undl* | Consolidated\_undl |
| Canada\_international\_projects.xlsx |  |
| *Canada\_international\_projects\_long* | Canada\_international\_projects\_long |
| *Topic\_WordCloud* | Topic\_WordCloud\_canada |
| Canada\_mission\_news.xlsx |  |
| *tagging* | Canada\_news |
| *Topic\_WordCloud* | Topic\_WordCloud\_canada\_news |
| Canada\_mission\_statements.xlsx |  |
| *tagging* | Canada\_statements |
| *Topic\_WordCloud* | Topic\_WordCloud\_canada\_statements |
| UAE\_foreign\_aid.csv | UAE\_foreign\_aid |
| Uae\_mission\_articles.xlsx |  |
| *tagging* | Uae\_mission\_articles |
| *Topic\_WordCloud* | Topic\_WordCloud\_uae\_news |
| Uae\_missions\_news.xlsx |  |
| *tagging* | Uae\_mission\_statements |
| *Topic\_WordCloud* | Topic\_WordCloud\_uae\_statements |
| Qatar\_mofa\_statements.xlsx |  |
| *tagging* | Qatar\_mission\_statements |
| *Topic\_WordCloud* | Topic\_WordCloud\_qatar\_statements |
| Qatar\_mission\_articles.xlsx |  |
| *tagging* | Qatar\_mission\_articles |
| *Topic\_WordCloud* | Topic\_Word\_Cloud\_qatar\_news |

## Navigation

Please refer to the Dashboard Guide for a more detailed overview of the dashboard and how to draw insights by utilizing the different features and filters in the dashboard.