**Summary of Methodology**

**A diagram of a workflow

AI-generated content may be incorrect.**

**Document Ingestion**

The corpus is extracted from Overton using search keywords, dairy and sustainability and it includes the documents categories Action Plans, Blog Posts, Clinical Guidance, Legislation, Publications, Regulations, Scholarly Article, Strategies, Transcripts, White paper and Working paper. Out of these categories, we decided to proceed only with Action Plans, Publications, Transcripts, and Strategy Documents. These categories are selected based on information and entities that they can provide, each of these categories provide a different set of information.

The data extraction and analysis were carried out in the following stages:

* The corpus consists of documents categorized into four types:

Transcripts: The sentiment analysis of speakers in Dail Eirean, Senead Eirean, and department specific debates on certain policies, action plans, and topic of interests.

Publications: The interaction of ‘Multiple Stakeholders’.

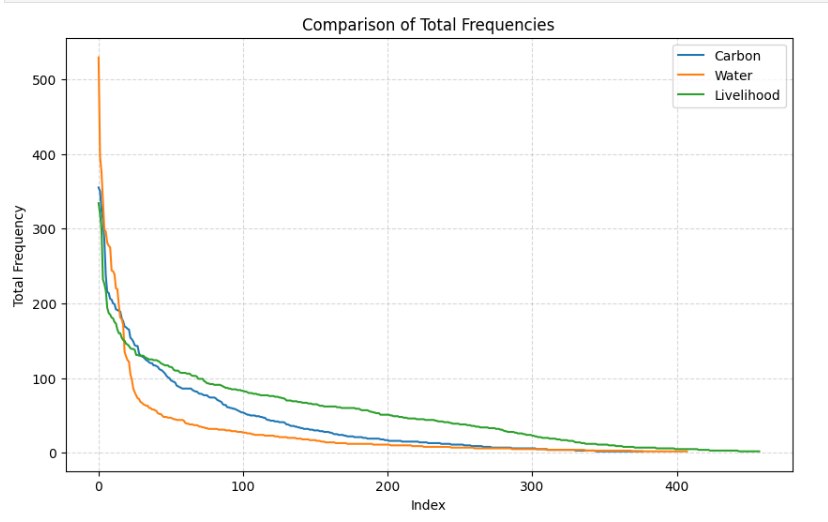
Strategies: The ‘Missions’ and the ‘Stakeholder’ who is assigned to that Mission.

Action Plans: The 'Action' and the 'Stakeholder' who is assigned to that Action.

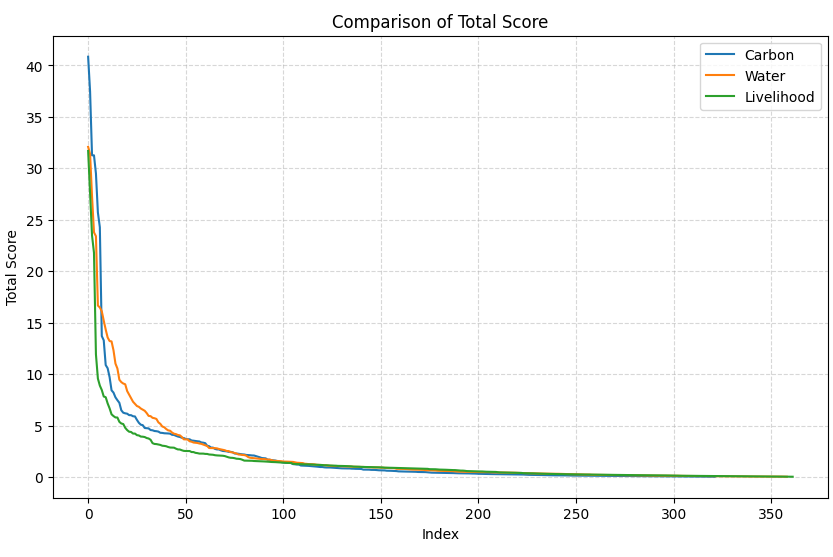
**Structuring**

* Each document is represented as a **term-document matrix**.
* To identify and rank the most relevant documents, we defined a set of keywords and phrases related to our key topics: **water**, **emissions**, **carbon**, **livelihood**, **health**, and **animal welfare**. These keywords can be searched with either the ‘OR’ or ‘AND’ separator.
* We queried the term-document matrices using these keywords to determine relevance based on keyword frequency.
* This analysis was visualized for **Transcripts** and **Publications**, as these categories had the highest number of documents. Through a line chart, we observed a sudden cut off or elbow point for frequency of keywords of interest and selected the top documents.

Transcripts:



Publications:



For **Transcripts** and **Publications**, we plotted the total frequency of relevant keywords (y-axis) against the number of documents (x-axis).

An "elbow point" or sharp drop was observed around:

* **100 documents** for Transcripts
* **50 documents** for Publications

Further manual review of the Publications led to a refined, smaller subset of 8 documents identified as the most relevant.

Documents in the **Action Plans** and **Policies** categories were all processed due to their limited number.

|  |  |
| --- | --- |
| Document Category | Document Count |
| Transcripts | 100 |
| Publications | 8 |
| Action Plans | 12 |
| Policies | 2 |

**Text and Entity Extraction**

Each document category was processed differently due to differences in format and content.

First major part of the processing, python techniques like PDF Parsing, NLP where used to section off the documents and extract and clean the text strategically. For example, split into paragraphs, entire content for each topic and subtopic. This part is the trickiest and required most effort. – Maybe we can include some references for difficulties faced in parsing and processing PDF documents (meant for human use and presentation) in traditional machine learning.

Second major part of the processing, using ChatGPT API for entity extraction and summarisation. The main thing to highlight here is that I didn’t think that there are models available for extracting these entities which are very specific to Ireland. These are maybe the limitations we faced with the very specific use case of this project:

Stakeholders like Bord Bia, Policies in Irish were not getting identified by the traditional entity extraction models.

The summarisation of the large text was to be done around the topic of interest, or the stakeholder, or the policy. It was not just a general summarisation but a summary of in what context these entities were mentioned in the large document texts.

For example, in Transcripts we wanted to summarise the context in which the speaker has mentioned the topic of interest, and what are their thoughts on this topic. In Publications, we wanted to summarise the large text in a way that it tells us how the Primary Stakeholder is related to the Secondary Stakeholder. Or in which context they are mentioned together.

I just felt that maybe there are no models out there which are tuned on such data to extract such insights out form our data and for our use case.  
The main reason to go with ChatGPT API is the world knowledge it has, the flexibility of defining our use case, and the time constraint we had. We may have been able to achieve these results with traditional models, but the lack of world knowledge, no training data, and limited time, ChatGPT seemed like a better option. So maybe we can find some references for such work.

The only limitation we had with ChatGPT approach is that we were not able to validate the entities extracted from the text. There was some manual validation done on a subset of documents, but for the rest of the selected corpus there was no checking.

Below are the entities for each document category.

**Transcripts**

* Transcripts consist of dialogues spoken by different speakers.
* Text was extracted speaker-wise to compile statements made by each individual.
* These speaker specific statements were analysed using **ChatGPT** to identify mentions of:
* **Policies**
* **Stakeholder organizations/groups**
* **Relevant Topics**
* Additional analysis was performed to understand each speaker's **stance and sentiment** (Positive, Negative, Neutral) regarding the mentioned policies, stakeholders, and topics.
* This is how the general pipeline looks like:  
  Document Text -> Extract and divide for each speaker and store in a dataframe (python) -> Run the dataframe through ChatGPT API -> ChatGPT to identify the topic (carbon, water, livelihood) -> Summarise the context in each the text is related to the identified topic -> Identify the overall stance of the speaker on that topic.

**Publications**

Publications often describe relationships between multiple stakeholders.

From each relevant section, we extracted:

* The **Primary Stakeholder** (main actor or authority figure)
* The **Secondary Stakeholder** (recipient or collaborator)
* A **summary** or **context** describing the interaction or relationship.
* The Primary Stakeholder is linked to the Secondary Stakeholder via a **Tag**, a tag is a word which describes their relationship in one word for example, collaborates, funds, reports, etc.

**Action Plan and Policies**

These documents typically include **Recommendations, Planned Actions, Responsible Bodies, Lead/Partner Stakeholders, Timeframes**.

From the Action Plans and Strategy documents, we have extracted:

* The **Action/Mission/Goal Statements**: These are extracted using python text extraction techniques. Since these explicitly mentioned there was no need for ChatGPT.
* The **Stakeholders** who were assigned with these Actions/Mission/Goal. These were extracted through both python and ChatGPT.

All this information was extracted and organized in a tabular format.

**Normalising Extracted Entities**

* Many extracted entities, such as **Stakeholders** and **Topics**, are common across document types.
* We normalized various forms and abbreviations of these entities to create a consistent dataset that supports integration across documents.
* This was done by combining these details into one dataframe and list, and then using ChatGPT to go through the list and formalize each entity into it’s formal version as it would appear in public documentation.

**Knowledge Graph Creation**

All extracted and normalized entities were used to build a Knowledge Graph in Neo4j.

The graph is structured around two central node types:

* Stakeholders: Connected to speakers, policies, action plans, and other stakeholders.
* Action Plans / Policies: Linked to missions, goals, stakeholders, responsible bodies, and timelines.

**Challenges and Solutions**

A key challenge in the project was ensuring accurate text extraction from the PDF documents, given their diverse formats and layouts. Many documents featured complex multi-column arrangements—often with two or three columns on a single page—which made automated extraction difficult. To address this, we combined advanced NLP techniques with manual review to accurately separate and extract the text content from each column.

Another difficulty arose with statistical data presented in visual formats such as graphs and charts. Since this data was embedded in images or non-extractable elements, it was not feasible to process it directly through text analysis. In such cases, we relied on selective manual extraction where necessary and focused our analysis on the accompanying textual information.

In the case of transcripts, identifying detailed background information about the speakers posed additional complexity. The transcripts often did not include metadata such as the speakers’ political affiliations, organizational roles, or geographic representation. To enrich the context, we used ChatGPT to source additional information from the web. However, this method can be resource-intensive and occasionally limited by the availability or currency of the information found.