Notes on methodology

Appendix to "Climate journalism neglects food's impact"

Original landscape analysis

Basis of approach

We chose the greatest practical scale: capturing as many articles as possible from a reasonable number of outlets and across a meaningful period of time. Our broad scope—37 outlets over a three-year period—prioritizes avoiding type II errors (failing to include relevant articles).

This scope also struck a balance between the exhaustive approach of the Madre Brava report (which would have made filtering stories and comparing outlets difficult) and the targeted approach of the Sentient report (which would have limited our ability to draw conclusions about the broader media landscape).

We also iteratively revised our search terms to avoid type I errors (failing to exclude irrelevant articles). We sought to draw comparisons in relative terms like percentages instead of absolute terms like counts.

Gathering articles

We designed our search for articles in the form of three nested searches, an approach inspired by a method used in the Madre Brava report. The query language we composed also integrated search terms from the Kristiansen et al. and Madre Brava studies, as well as a stock Factiva search term, `ns=GCLIMT`, for topics related to global climate change. The full text and modular construction of the queries can be found in the Github repository associated with this project as the file `Factiva queries.sql` (although it is not SQL code—that file extension was appended to the plain text file for clarity in formatting). A plain-language overview of the segments composing the queries is provided here.

We created a baseline set of conditions we wanted met in all the articles we collected: language (English), dates (a three-year span from July 1, 2022, through June 30, 2025), article type (full-length, more than 250 words), and region (published in the United States). Using Factiva's `tmnbus` list of major U.S. news and business publications, we restricted our search to 41 sources, which allowed us to both compare coverage between outlets and get an overview of the broader media landscape. Because Factiva did not return complete results for some outlets when using the tmnbus search term `rst=tmnbus`, we instead listed out the codes of the outlets: `rst=(sfabc or sfamb or …)`. Of those 41 outlets, 37 had published articles that met our criteria.

To this baseline, we appended the query for climate coverage, then the query for meat or animal agriculture, then the query for dietary shifts, such that each was narrower than the previous. The first query searched for coverage climate change, returning a set of articles we will call "`cli`". The second query added a search for mentions of meat or animal agriculture, returning a set of articles we will call "`meat`". The third query added a search for mentions of dietary changes, returning a set of articles we will call "`diet`". Thus, the articles mentioning dietary shifts (`diet`) constituted a subset within the set of articles

mentioning meat or animal agriculture (`meat`), which itself constituted a subset within the set of articles covering the climate (`cli`).

- Filter for language, date range, word count, region
- Filter for selected sources
- Exclude non-article items, irrelevant article types
- Search for topic of climate change
    - Search for mentions of meat consumption
        - Search for mentions of dietary change

Our search returned 10,696 total articles, with smaller numbers of articles within that sample constituting the `meat` and `diet` subsets. From the Factiva results for each of the three queries, we exported three CSV files describing the number of articles from each publication, how many times each mentioned industry was mentioned, and how many times each organization was mentioned. Those nine CSVs can be found in the folder `Exported Factiva statistics` in the Github repository for this project.

We also downloaded the text of the articles for further analysis, 100 articles at a time, as rich text files. We used a Python script to split each 100-article rich text file into plain text files of each individual article, labeled with the article's date. This produced plain text files of 8,086 articles.

We attribute the discrepancy between this figure and the 10,696 total from the exported Factiva statistics to the practice of smaller newspapers republishing articles licensed from larger newspapers. The duplicate articles may have been counted in the figures exported from Factiva; we preserved those counts on the rationale that republication reaches new readers the same way original articles do. However, for our textual analysis, we wanted to include each article only once. If we did miss some articles, it would have been few enough as to not affect our analysis.

Processing in R

We used the open-source programming language R in the RStudio integrated development environment to process, analyze, and visualize our data. Our code is published in the Github repository for this project.

After reading the CSV files into R, we used this data to create data frames for the number of articles from or about each media outlet, industry, and company in each of the three scopes. We created a data frame `sources` with article counts by publication, a data frame `industries` with mention counts by industry, and a data frame `companies` with mention counts by organization or corporation. These three data frames—and rearranged versions of them—were used to compute summary statistics and plot trends using tools in base R and in the packages `dplyr`, `ggplot2`, `stats`, `stringr`, `tidyr`, and `utils` in the `tidyverse` package. `ggtext` was used to handle text styling in the plot captions.

We imported the text files into R with `readtext`, then processed them as a corpus using `quanteda` and packages in `tidyverse`. We also used the `quanteda` add-on `quanteda.sentiment` for sentiment analysis. Word cloud plots were handled with `ggwordcloud` in conjunction with `ggplot2`. Statistical tests were performed with `effectsize`. When processing tokens (i.e. words), we excluded 241 common words using the `stopwords` package and by filtering out words we considered not meaningful for our analysis (e.g. "said", "may," "next"). We also removed "climate," "change," and "global warming"

because it would not have been useful to find that those words were often used in coverage of climate change.

Repeating Faunalytics/Sentient analysis

Gathering articles

The researchers selected 10 major U.S. media outlets as a sample of the broader media landscape. From each of these 10 outlets, the 100 most recent articles with "climate" in the headline were selected, yielding 1,000 articles that served as the corpus for the analysis. The R programming language was used to search each of the 1,000 articles for keywords associated with 10 "climate-related themes," mostly anthropogenic causes of climate change. Articles containing a keyword were categorized as including its respective theme.

We searched the Factiva database with standardized queries that replicated the original sampling criteria: the 100 most recent articles with "climate" in the headline from each source. The Factiva license we used did not include access to articles from the Los Angeles Times, so we were only able to analyze 9 of the 10 original outlets. Despite this limitation, the access to other paywalled sources, the highly customizable query system, and the consistent formatting of the exported rich text files made this method more than efficient and powerful enough at selecting the articles we wanted and excluding those we did not.

Like Faunalytics/Sentient, we excluded duplicates, miscategorized articles, and non-article content whenever possible. This included filtering out updates, repeats, and corrections from the Reuters wire service that appeared as duplicate articles. We also filtered out items with 250 or fewer words in an effort to ensure we were getting only full-length articles. We further manually excluded articles that used "climate" in a figurative sense, which were common in the New York Post and Chicago Tribune. As a result, we only had 88 articles from the Chicago Tribune, and many of the New York Post articles were not recent.

The resulting pool of articles were downloaded as rich text files, organized into directories corresponding to the outlets in which they were published, and renamed in a standardized format. We consider the resulting sample of 888 articles to be sufficient for our purpose of comparison with the original analysis.

Processing in R

We used the Faunalytics/Sentient code to import the text files into R, then scanned the text to detect topics mentioned. This resulting categorization allowed us to generate summary statistics and export data frames for further analysis.

The full methods and code used in the 2023 Faunalytics/Sentient report can be found in the Open Science Foundation repository for that project.