# **Element A1. Accessing Data across Platforms**

Element A1 describes how we collected data from outlets’ own domains via (1) crawling websites, (2) scraping content, and (3) cleaning data. Throughout these steps, we used the programming language R (R Core Team 2021), especially the R package rvest (Wickham 2021) and, for the retrieval of articles’ texts, the Python module trafilatura (Barbaresi 2021). To respect websites’ robots exclusion standards indicating rate limits and which content can be accessed (e.g., [www.sueddeutsche.de/robots.txt](https://www.sueddeutsche.de/robots.txt)), we used the R package polite (Perepolkin 2019).

## *Step 1: Crawling News Websites*

First, we crawled news outlet’s landing domain every full hour to retrieve links to news articles. We relied on a rule-based approach to filter out non-journalistic content via URL markers. *Süddeutsche Zeitung* for instance links to its real estate portal (<https://immobilienmarkt.sueddeutsche.de>), a non-journalistic offer, links to which were excluded automatically. We took a conservative approach here by only excluding some, clearly non-journalistic links since all content was cleaned further in the third step, enabling us to further sort out non-journalistic content. The accuracy of this crawler was tested in two pre-tests (in August and November 2021) where we crawled each website and manually compared crawled links to manually identified links.

## *Step 2: Scraping News Websites*

Next, we scraped content from crawled links by relying on the pages’ HTML source code. This included the content of the article (i.e., headline, teaser, full text, images) and its meta data (i.e., publication date, author, name of the outlet). Importantly, we build the scraper to automatically stop if it identified a paywall and thus refrained from circumventing paywalls.

## *Step 3: Manual Cleaning*

In a last and final step, we manually checked all links (Step 1) and related, scraped content (Step 2) for completeness. This was important, seeing that we needed to further exclude links to some non-journalistic links (Step 1) or manually download paywalled or otherwise restricted content (Step 2). Concerning the latter point, robots exclusion standards sometimes disallowed us to automatically download, for instance selected videos, which was done manually in Step 3. Overall, this manual cleaning reassured the completeness of scraped content. After Step 3, we identified *N* = 2,654 articles as the final, cleaned population of website content.

# **Element A2. Matching Data Across Platforms**

Element A2 describes how we matched and thus cleaned our initial sample of *N* = 2,367 social media (SM) posts. Matching and exclusion was done via a (1) automated matching via links and (2) manual coding of SM posts not matched automatically.

## *Step 1: Automated Matching via Links*

Similar to other studies (Pak 2019), we first cleaned links in SM posts for automated matching. The most important step here was unshortening links via the R package httr (Wickham 2020) (e.g., <https://t.co/GXVRD99Nhl> to <https://www.spiegel.de/netzwelt/netzpolitik/attila-hildmann-der-abstieg-vom-vegan-koch-zum-verschwoerungstheoretiker-a-2feff427-e48d-48cd-b4f7-051ab5ab4fe4>). After cleaning links, we matched SM posts to respective articles. Next, all SM posts not matched automatically (*N* = 1,079) were checked manually.

## *Step 2: Manual Matching via Textual, Visual, or Topical Overlap*

In a next step, coders inspected every SM post not matched automatically. Each was coded as **matched** (by coding the matched WEB ID) or **not matched** (by coding “formal” or “native”). In short, SM posts were only be coded as “matched” if they were based on existing website articles published within our observation period. SM posts were coded as “not matched” for two reasons:

1. “*Native*”: SM posts were native, that is, not based on existing website articles but created exclusively for social media, for instance video material for TikTok (see [*Tagesschau* example](https://www.tiktok.com/@tagesschau/video/7025646560197364998))
2. “*Formal*”: SM posts were excluded for formal reasons if SM posts matched articles published outside of our observation period (e.g., SM posts matched articles posted in October 2021), SM posts contained non-journalistic content (e.g., ads for paid subscription), or they linked to broadcasting programs including several topics (e.g., the full 8pm *Tagesschau* program).

Coders received the following coding instruction[[1]](#footnote-2), based on which they decided whether a post was coded as “matched” (by coding the corresponding WEB ID) or as not matched (by coding either “native” or “formal”). Coding was conducted by the first and second author of the study. For intercoder reliability, both annotated 10% of the material (*N* = 108, *α* = .84).

Coding Instruction:

“For every SM post, please code whether or not it can be matched to a corresponding WEB article. A “match” between a SM post and a WEB article is coded if the post is clearly based on an article from the WEB corpus, i.e., it describes the same topic and strongly uses text/visual elements from WEB articles. If you found a match, please note the ID of the corresponding WEB article (e.g., “*TS-2021-11-01-15-00-33.pdf*”).

Otherwise, please note “*native*” or “*formal*” as an indicator for content that could not be matched.

* “Native”: SM posts are not based on existing website articles but created exclusively for social media. Native content deals with different topics than WEB articles and/or has no strong textual or visual overlap with existing WEB content (see [example](https://www.tiktok.com/@tagesschau/video/7025646560197364998)).
* “Formal”: SM posts match articles published *outside of our observation period* (e.g., SM post matches article posted in October 2021, see [example](https://www.facebook.com/215982125159841/posts/4752033431554665)), SM posts contain *non-journalistic content* (e.g., ads for paid subscription, see [example](https://www.instagram.com/p/CVvXCYhq_C8/)), or they link to *full broadcasting programs including several topics* (e.g., the full 8pm Tagesschau program).

**… for SM posts including link to WEB domain:**

If a SM post contains a link to a WEB domain (but was not matched automatically, for instance due SEO-tags added for social media slightly changing the URL): Open the URL in the SM post. Use the “datepublished” tag in the HTML code of the WEB domain to check whether the article was published between November 1st – 7th.

*If this is not the case* (e.g., SM post links to an article published outside of our observation period): code “formal” as an indicator for non-matched content.

*If this is the case* (i.e., SM post links to an article published in observation period): The article was likely not matched due to a slightly changed URL (due to SEO tags, etc.). Look-up the corresponding URL in the WEB corpus using keywords from the URL embedded in the SM post.

Example R code: WEB$ID[grepl(“corona-news”, WEB$url)]

If you cannot identify a match via the URL, please pursue with the following steps:

Look-up WEB articles with similar text elements in their titles (WEB$title), teasers (WEB$teaser), and full texts (WEB$text) using keywords from the SM post.

Example R code: WEB$ID[grepl(“Corona-Inzidenz steigt”, WEB$title)]

If none of these approaches work, please check for other overlap (see next steps).

**… for SM posts not including link to WEB domain:**

If a SM does not contain a link, check for textual overlap as described before.

*If you do not find a match via textual overlap*, please check for WEB content using the same visual material. To do so, please open the corresponding folder for scraped content from the corresponding outlet and click through all images/videos.

*If you do not find a match via visual overlap*, please retrieve a list of all WEB articles published by the same outlet on the same day as the social media post. Open all URLs and, again, check for textual and visual overlap. If this does not lead to results, go back a day, and so forth.

Example R Code: WEB$url[WEB$Outlet==“TS” & WEB$date==“2021-11-01”]

*If you do not find a match via articles as described above*, please visit the online archive of the outlet. Searching for keywords from the SM post, please make sure that the content has not been published earlier than our observation period.

If none of the approaches above led to a match, please code “native” as an indicator for non-matched content.

## *Results across Platforms*

Table A2.1 describes the initial sample of *N* = 2,367 social media posts, including the share of matched content (i.e., content based on existing website material) and non-matched content (i.e., native content or content excluded for formal reasons).

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table A2.1**. (Non-)Matched Social Media Posts across Platforms | | | | | | | | | | | |
| **Outlet** | | **Platform** | | | | | | | | | |
|  | | Facebook | | Instagram | | TikTok | | Twitter | | Overall | |
| Matched Content (*Final Sample*) | | 238 (72.1%) | | 179 (61.3%) | | 4 (66.7%) | | 1,337 (76.9%) | | 1,758 (74.3%) | |
| Not Matched (*Excluded for Final Sample*) | |  |  | |  | |  | |  | |  |
|  | Native | 8 (2.4%) | | 34 (11.6%) | | 2 (33.3%) | | 13 (0.7%) | | 57 (2.4%) | |
|  | Formal | 84 (25.5%) | | 79 (27.1%) | | – | | 389 (22.4%) | | 552 (23.3%) | |
| Overall (*Initial Sample*) | | 330 | | 292 | | 6 | | 1,739 | | 2,367 | |

# **Element A3. Operationalization of Variables**

Due to word limitations in the main paper, Element A3 describes the following elements for each variable (where existent): (a) operationalization, (b) validation, and (c) coding instruction for manual coding. Across variables, we validated our results by comparing automated coding results to a manually coded “gold standard” where manual coders annotated the same variables, a common procedure in NLP approaches. Similarities and differences were then evaluated related to a model’s precision, recall, and F1 value: *Precision* indicates how many units of analysis predicted to contain a variable (e.g., articles containing opinionated content) according to our automated analysis did according to the gold standard, i.e., how good the NLP approach is at not creating too many false positives. *Recall* indicates how many units of analysis that actually contain a variable (e.g., opinion pieces) were found, i.e., how good the NLP approach is at not creating too many false negatives. The *F1* value is the harmonic mean of precision and recall. It is often used as an overall evaluation metric for model assessment (Manning and Schütze 1999).

### *Variable: Opinion Piece*

**Operationalization**. Using automated content analysis, we relied on outlets’ own classification to code whether articles were *Opinion Pieces*. Outlets used visual “editorial comment” signs on their websites or labelled articles as “editorial comments” to separate opinion pieces from other content. Based on articles’ source code, we extracted tags to classify articles (0 = not an opinion piece, 1 = opinion piece).

**Validation**. We validated this rule-based approach based on comparison to a manually coded gold standard. Since the overall corpus included comparably few opinion pieces, we used stratified sampling to create a more balanced sample for manual validation (Stoll 2020). Thus, the gold standard was based on a stratified sample of *N* = 400 news stories including both opinion pieces (*N* = 97) and non-opinion pieces (*N* = 307). All stories were coded as opinion pieces (1) or non-opinion pieces (0) by the first author (see coding instruction below). The second author coded 10% of the sample for intercoder reliability (*N* = 40, *α* = .95). Comparing this manual coding to the automated classification reassured the validity of the results (*precision* = .97, *recall* = .99, *F*1 = .98).

**Coding Instruction (for Gold Standard)**.

For every journalistic article, please code whether the story is an opinion piece (1) or not (0). Indicators for opinion pieces include (1) textual markers, i.e., stories clearly being labelled “Kommentar” or “Kolumne” or “Glosse” or “Leitartikel” or “Essay” or “Gastbeitrag” (see [example](https://www.spiegel.de/kultur/einfach-mal-alles-entspannt-sehen-hilfe-jetzt-bin-ich-auch-ein-mittelalter-weisser-mann-a-c0e31413-8dda-4053-a410-7fb974ed9f9c)), (2) visual markers, i.e., stories being visually identified as opinion pieces (see [example](https://www.tagesschau.de/kommentar/kinder-immpfen-coronavirus-101.html)), or (3) an indication that the article was published in the “opinion” section (see [example](https://www.sueddeutsche.de/meinung/cdu-vorsitz-merz-roettgen-zukunft-1.5453682)).

### *Variable: Topic*

**Operationalization**. Using automated content analysis, we coded whether articles dealt with celebrities as their main *Topic* by relying on a pre-trained supervised machine learning classifier validated for a similar German-language news corpus (Jürgens and Stark 2022). The neural network-based classifier is based on a fine-tuned German BERT transformer trained on N = 10,000 manually coded articles by 25 coders (α = .81). In the study by Jürgens and Stark (2022), the authors assigned articles one of the following topics: “accidents”, “celebrities”, “crime”, “finance”, “general politics”, “health”, “hobbies/lifestyle”, “science”, “social issues”, “sports”, “parties and politicians”, “war”, and “weather”. We combined some of these categories, namely “general politics” and “parties and politicians” to “politics” and “accidents”, “war”, and “crime” to “crime/disaster”. By doing so, we assigned articles one of the following ten topics: “celebrities”, “crime/disaster”, “finance”, “health”, “hobbies/lifestyle”, “politics”, “science”, “social issues”, “sports”, or “weather”.

**Validation**. In the study by Jürgens and Stark (2022), a more fine-grained model with more topical categories (see above) achieved valid results (*average precision* = .84, *average* *recall* = .84, *average F1* = .84).

*Variable: Personalization*

**Operationalization**. Using automated content analysis, we measured *Personalization (Images)* and *Personalization (Videos)* via a pre-trained automated image analysis pipeline by Authors (2022b). Since the pipeline is described in more detail in the paper and Deng et al. (2019), we only shortly describe it here. Descriptions here are thus based on Authors (2022b). The image analysis pipeline used the RetinaFace library (Deng et al., 2019) to detect all faces in each image. For videos, one frame per second was sampled for analysis.

**Validation**. We only used the face detection component of the pipeline, which is based on RetinaFace. As Deng et al. (2019) show, the approach has a high *average precision* and *recall* (above .95 for large and medium sized faces and above .90 for very small faces).

### *Variable: Excluding Links*

**Operationalization**. Using automated content analysis, we coded whether social media posts were adapted by excluding (or including) links to outlets’ websites. Twitter’s API and access via Crowdtangle allowed us to automatically retrieve links embedded in Twitter, Facebook, and Instagram posts. For Facebook stories, Instagram stories, and TikTok posts, we manually stored links during data collection. The variable *Excluding Links* (0 = no, 1 = yes) was created automatically by checking to which domains, if any, posts linked: If a social media post by Tagesschau, for instance, did not link to *www.tagesschau.de,* this post was coded as excluding a link (1 = yes).

**Validation**. We validated this rule-based approach based on comparison to a manually coded gold standard. This gold standard was based on a random sample of *N* = 400 social media posts. All posts were coding as excluding links (1) or including links (0) by the first author (see coding instruction below). The second author coded 10% of the sample to check for intercoder reliability (*N* = 40, *α* = .94). Comparing this manual coding to the automated classification reassured the validity of the results (*precision* = .98, *recall* = .96, *F*1 = .97).

**Coding Instruction (for Gold Standard)**.

For every social media post, please code whether the post excludes links to the outlets’ website (1) or not (0). Indicators for the exclusion of links (a) the exclusion of any link in the social media post or (b) the inclusion of a link, but not to outlets’ home domain.

### *Operationalization: Interactive Features*

**Operationalization**. Using manual content analysis, we annotated whether articles or posts included optional interactive features allowing audiences to select options or type in questions, for instance through polls (Stroud, Scacco, and Curry 2016). We only considered features going beyond commenting on, liking, or sharing content. Engaging features were coded manually (0 = no engaging features, 1 = engaging features) by the second author. Intercoder reliability was checked by the first author coding 10% of the material (N = 440, α = .86).

**Coding Instruction (for Full Sample)**.

For every journalistic article or social media post, please code whether it contains interactive features actively fostering audience engagement (1) or not (0). Features indicating audience engagement are any element based on which users are actively asked to send in their reactions (e.g., via polls) or engage with the article beyond the general possibility to comment, like, or share content. Examples include:

* polls/surveys (see [example](https://www.spiegel.de/kultur/tv/tatort-vote-wie-gefiel-ihnen-luna-frisst-oder-stirbt-a-9d4e5230-90b6-4b3a-8cb3-1d6d4143b0c6), [example](https://www.spiegel.de/politik/deutschland/cdu-norbert-roettgen-wuenscht-sich-parteichef-in-der-modernen-mitte-a-c342cb64-a37a-42b1-9b32-85faa75496fd), [example](https://www.spiegel.de/sport/olympia-2024-in-paris-reiten-soll-aus-dem-programm-des-modernen-fuenfkampfs-fallen-a-fc599d86-6fe5-4da1-bf3e-3c789a3ae868))
* knowledge quizzes
* “ask me anything” (e.g., sending in questions via Instagram stories)
* text boxes for typing in/searching for information (see [example](https://www.zdf.de/nachrichten/panorama/corona-impfpflicht-new-york-100.html))

### *Operationalization: Engaging Language*

**Operationalization**. Using automated content analysis, we coded whether articles or social media posts used engaging language by directly addressing audiences or posing questions. We used a list of pronouns for addressing people[[2]](#footnote-3) and identified all instances where a validated part-of-speech tagger from the same model for German-language news (*F1* = .98) identified words as a pronoun or determiner to reduce false positives. After normalizing for text length, *Addressing Audiences* describes the share of words in each article (title/teasers) or social media post directly addressing audiences. Using a rule-based approach, we also created the variable *Posing Questions* which includes the share of sentences identified as questions in each article (title/teasers) or social media post.

**Validation**. We validated this rule-based approach based on comparison to a manually coded gold standard. Importantly, we here compared whether the identification of any question or pronouns (as an indicator for engaging language) was correctly identified by the rule-based automated coding. Thus, transformed numeric indicators for *Addressing Audiences* and *Posing Questions* to a single sum index measuring whether at least one of these indicators (i.e., direct addressing of audiences, a question to the audience, or both) occurred according to the rule-based approach (1) or not (0). We then coded a gold standard based on a sample of *N* = 400 articles and social media posts. All units in this gold standard were coded for any indication for engaging language (1) or not (0) by the first author (see coding instruction below). The second author coded 10% of the sample to check for intercoder reliability (*N* = 39, *α* = 1). Comparing this manual coding to the automated classification reassured the validity of the results (*precision* = .7, *recall* = .86, *F*1 = .77).

**Coding Instruction (for Gold Standard)**.

For every journalistic article or social media post, please code whether the post uses engaging language (1) or not (0). We here define engaging language as language that engages readers by directly addressing them, for instance via (a) directly addressing (e.g., “You can trust this app!”, see [example](https://www.spiegel.de/kultur/tatort-vote-wie-gefiel-ihnen-luna-frisst-oder-stirbt-a-9d4e5230-90b6-4b3a-8cb3-1d6d4143b0c6), [example](https://www.sueddeutsche.de/wissen/technik-dieser-wetter-app-koennen-sie-vertrauen-dpa.urn-newsml-dpa-com-20090101-211029-99-788884)) or (b) posing open questions (e.g., “What do you think?” or “Who should pay for the government’s plans?”, see [example](https://www.spiegel.de/kultur/tatort-vote-wie-gefiel-ihnen-luna-frisst-oder-stirbt-a-9d4e5230-90b6-4b3a-8cb3-1d6d4143b0c6), [example](https://www.zdf.de/nachrichten/heute-sendungen/ampel-plaene-investitionen-kredite-video-100.html)).

**References**

Barbaresi, Adrien. 2021. “Trafilatura: A Web Scraping Library and Command-Line Tool for Text Discovery and Extraction.” In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, 122–131. Association for Computational Linguistics. https://aclanthology.org/2021.acl-demo.15.

Jürgens, Pascal, and Birgit Stark. 2022. “Mapping Exposure Diversity: The Divergent Effects of Algorithmic Curation on News Consumption.” *Journal of Communication* 72 (3): 322–344. https://doi.org/[10.1093/joc/jqac009](https://doi.org/10.1093/joc/jqac009).

Manning, Christopher D., and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, Mass: MIT Press.

Pak, Chankyung. 2019. “News Organizations’ Selective Link Sharing as Gatekeeping: A Structural Topic Model Approach.” *Computational Communication Research* 1 (1): 45–78. doi:10.5117/CCR2019.1.003.PAK.

Perepolkin, Dmytro. 2019. *Polite: Be Nice on the Web*. https://CRAN.R-project.org/package=polite.

R Core Team. 2021. *The R Project for Statistical Computing*. https://www.r-project.org/.

Stoll, Anke. 2020. “Supervised Machine Learning mit Nutzergenerierten Inhalten: Oversampling für nicht balancierte Trainingsdaten.” *Publizistik* 65 (2): 233–251. doi:10.1007/s11616-020-00573-9.

Stroud, Natalie Jomini, Joshua M. Scacco, and Alexander L. Curry. 2016. “The Presence and Use of Interactive Features on News Websites.” *Digital Journalism* 4 (3): 339–358. doi:10.1080/21670811.2015.1042982.

Wickham, Hadley. 2020. *Httr: Tools for Working with URLs and HTTP*. https://CRAN.R-project.org/package=httr.

Wickham, Hadley. 2021. *Rvest: Easily Harvest (Scrape) Web Pages*. https://CRAN.R-project.org/package=rvest.

1. All coding instructions are shortened extracts from the codebook. We for instance deleted images from the corpus due to copyright issues and added links to corresponding URLs where possible. [↑](#footnote-ref-2)
2. We used a shortened list of translated pronouns proposed by Haim et al (2021) consisting of only those pronouns used for directly addressing someone: “du, dein, deine, deiner, dir, Sie, Ihr, Ihre, euch, euer, eure, eurer” [↑](#footnote-ref-3)