The AInews Corpus and Analysis Pipeline

Cornelius Puschmann, Rahel Winter, Florian Hohmann, Alexander Ohlei

v1.1 2026-10-20

Summary

The AInews corpus is a multilingual news corpus designed for the large-scale content analysis of public discourse about communicative artificial intelligence (DFG RU 5656 ComAI, see comai.space). It represents one of three components of the ComAI Research Space, and provides shared contextual and discourse evidence for the affiliated research projects. The corpus supports analyses of AI coverage across four media systems — Germany (DE), Austria (AT), the United Kingdom (UK), and the United States (US) — in English and German. This document serves two purposes. As documentation, it records the design, construction, schema, and operational logic of the corpus and its principal analytical extensions. As a methodological contribution, it makes explicit the sampling frame, the units and levels of inference, the measurement decisions, and the conditions under which claims can be drawn from the data. Three analytical components are covered: Classification of articles as AI-relevant and subproject-relevant, BERTopic topic modeling and issue aggregation, and Named Entity Recognition (NER) with AI-domain entity classification. Subproject relevance classification has so far been implemented for three of the nine affiliated projects is documented because it is part of the established pipeline. The corpus is an evolving infrastructure. The snapshot described here contained 70,428 rows of primary corpus data and covered publication dates from 2012 to 2026. It will continue to change, both on the level of data and analytical pipeline.

Document status and reading note

This is an evolving living document that describes a working research pipeline. Because the pipeline changes iteratively rather than through a fixed release cycle, we distinguish between the general design and operational logic that are intended to persist and, on the other hand, the numerically specific properties of the current data snapshot. No credentials or article text are reproduced in this document.

1. The AInews Corpus

1.1 Communicative AI and the ComAI Research Unit

The corpus was built within the framework of the ComAI (Communicative Artificial Intelligence) Research Unit. Communicative AI denotes automated communication systems that are (1) based on various technologically advanced forms of automation for the *purpose of communication*, (2) embedded within *digital infrastructures*, and (3) closely *linked with human practices* (Hepp et al,

2026). The guiding research question of the Research Unit is how societal communication changes when communicative AI becomes a part of it. By societal communication we mean all forms of mediated communication essential for the discursive construction of society — public, (inter-)personal, and group communication, each situated in a particular social domain.

The Research Unit distinguishes two broad research areas. Research Area 1 (RA1) investigates the **sociomaterial constitution** of ComAI across four dimensions: the pioneers who imagine ComAI and its possible futures (P1), its implementation in terms of particular interfaces (P2), its juridification in law (P3), and the ordering of ComAI through corporate governance (P4). Research Area 2 (RA2) investigates the **appropriation** of ComAI in social domains: how autonomy is challenged by automation in journalism (P5), how deliberation is redefined when ComAI participates in public discourse (P6), how companionship in the personal sphere is refigured (P7), how healthcare transforms through ComAI (P8), and how learning and teaching are renegotiated when ComAI enters education (P9).

1.2 The ComAI Research Space

The ComAI Research Space is the shared research infrastructure of the research unit. It consists of three components: *Documenting discourse*, which asks which ComAI is being talked about publicly, by whom, and how, on the basis of a large-scale semi-automated content analysis of public communication across the four core-sample countries (Austria, Germany, the UK, and the US); *Monitoring technologies*, which tracks which new ComAI technologies are emerging or declining and their contexts of use; and *Exploring futures*, which uses scenario and audit workshops to interrogate assumptions about future developments of communicative AI.

As part of the *Documenting discourse* component, the AInews corpus is responsible for making the public debate about communicative AI empirically tractable. This relates to which ComAI is discussed, by whom, and how, across the four media systems. The corpus is designed to be shared infrastructure in two senses. First, it is a common data resource that affiliated projects can draw on for contextual and discourse evidence — for example, describing how a technology, actor, or issue is covered in the public sphere over time. Second, it supplies aggregated, comparable analytical products (topic and issue distributions, entity inventories) that reduce duplication across projects.

The corpus is *shared context*, not a substitute for each project's own data. It does not directly operationalize all nine subprojects: project-specific data collection, experimental designs, interviews, and domain analyses remain with the individual projects. The documented pipeline currently provides dedicated subproject-relevance subsets for P1, P5 and P9 (Section 4.2). These subsets adapt the general corpus to the discourse needs of those two projects (for P1, pioneer communities, movements, elites, organizations, visions, and imaginaries; for P5, reflexive discourse about journalism in relation to AI). The remaining projects are served by the general corpus and its analytical outputs where their questions concern public discourse, and by their own field data elsewhere.

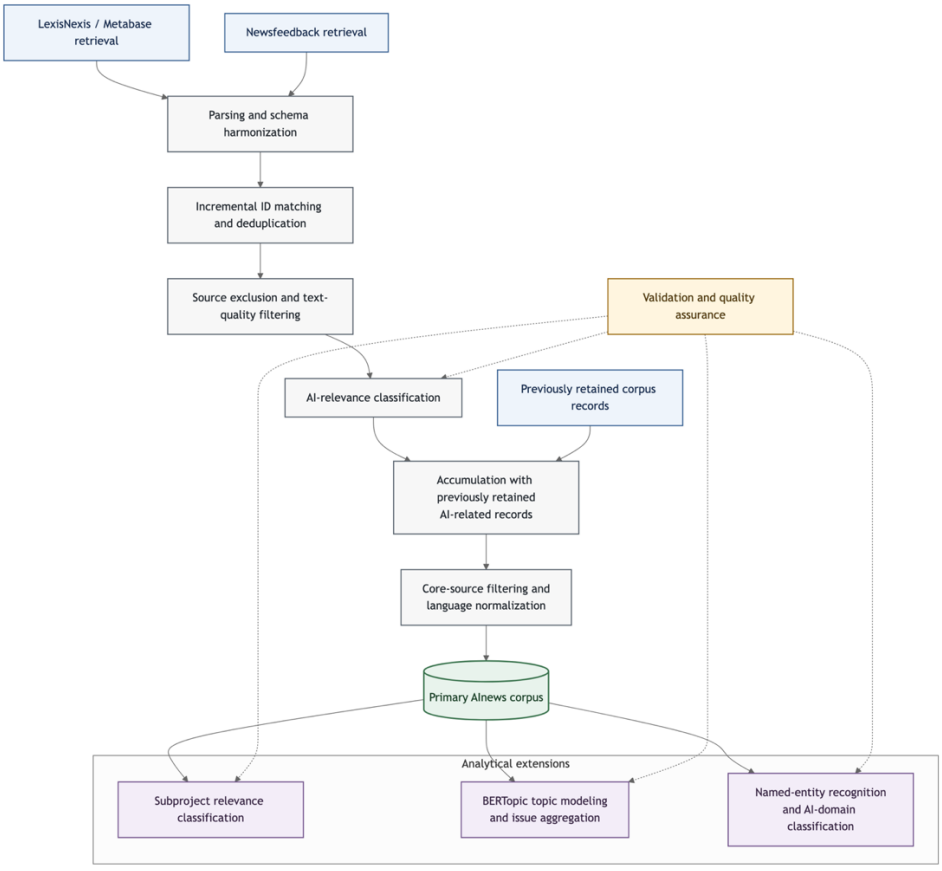
1.3 Corpus-level and project-specific analysis

The corpus supports two levels of analysis with different units and different inferential scopes. At the *corpus level*, the analytical unit is the news article and the inferential target is coverage — that is, the distribution of attention across issues, actors, sources, countries, and time within AI-related news in the four media systems. At the *project level*, the unit may be the article, the subproject-relevance flag, the topic assignment, or the named entity (mention or distinct entity), depending on the research question. Just as these units differ, so do the appropriate denominators and the strength of the claims that can be made. The cross-national contrastibility of the corpus rests on an explicit sampling frame (Section 3) rather than on the assumption that each country's media system produces equivalent coverage volumes; accordingly, aggregate counts should be interpreted as system-specific and compared with the country shares in mind.

2. Corpus Construction

This section describes the operational logic of corpus construction. It is intentionally generic about the specific processing files because the pipeline evolves. The analytical steps, the data classes they consume and produce, and the decision rules are the stable elements. A generic representation of the data flow is given in Figure 1.

Figure 1. Alnews corpus pipeline.



2.1 Data acquisition

The corpus is assembled from two primary retrieval streams plus an earlier project collection.

LexisNexis/Metabase retrieval. The first stream retrieves coverage via the LexisNexis/Metabase Search API (version 10). Retrieval is manually triggered in irregular intervals once a month and proceeds in daily batches retrieved from the API in a loop. For each UTC calendar day, the following clauses are combined with Boolean *AND*:

Dimension	**Query**
AI terminology	`((AI OR "artificial intelligence") OR (KI OR "künstliche intelligenz"))`
Source country	Germany, Austria, United States, or United Kingdom
Language	German or English
Feed class	News, Blog, or Print
Source rank	1, 2, or 3
Publication date	One-day UTC interval, 00:00:00 through 23:59:59

This is a deliberately high-recall lexical strategy. It does not by itself establish substantive AI relevance, which is performed by the machine-learning relevance filter described in Section 2.3. Request handling is governed by documented operational constraints: a maximum batch size of 200 articles; API-level duplicate filtering; pagination by passing the final sequence identifier from one response to the next; termination of the loop when sequence identifiers fail to advance (preventing an infinite cycle) or when a continuation request returns HTTP 400; and a per-minute quota check before each request, with a pause when the minute quota approaches its limit. Raw responses are written to an XML captures directory under versioned names. Earlier LexisNexis data are incorporated each time the corpus is updated in a combination step (Section 2.2) rather than re-fetched.

Newsfeedback retrieval. The second stream filters newsfeedback (https://github.com/Leibniz-HBI/newsfeedback) database content in PostgreSQL before export, retaining rows whose text matches a case-insensitive whole-word regular expression for *ai*, *artificial intelligence*, *ki*, or *künstliche intelligenz*. The intended operational procedure is to rename the previous filtered table as a backup and export the new filtered table as a delimited file with headers, using a pipe delimiter. The import specification carries title, author, URL, hostname, description, site name, publication date, categories, tags, article text, retrieval timestamp, token count, character count, and source metadata. The harmonized corpus retains title, text, publication date, token count,

author, and site name, and treats article language within the newsfeedback corpus as initially unspecified until the language step (Section 2.4).

2.2 Parsing, combination, and incremental updating

The first analysis stage parses the raw LexisNexis XML and extends it with fields. For each article node the parser extracts the identifier, title, language, content, publication date, word count, author name, source name, source country, and a joined list of topics. Publication timestamps are parsed into date-time values, word counts are coerced to integers, and identifiers become character strings. Parsing is set up to run incrementally. Only files created after a designated baseline date are parsed, and the output is appended to an existing parsed data product rather than rebuilding from scratch. The source XML filename is retained during parsing but is not used in the harmonized article schema, so it is a processing detail rather than a corpus field.

The combination stage is an incremental update rather than a complete rebuild. It reads the set of existing identifiers from the most recent prior filtered data product and uses it as the baseline against which incoming records are deduplicated. For both incoming streams, a deterministic identifier is generated from the title, the publication date (reduced to calendar date), and the source, using an MD5 hash. Incoming identifiers already present in the baseline are marked as already added and removed. Both sources are then reduced to a common harmonized schema of eight fields — *id*, *title*, *text*, *published_date*, *language*, *word_count*, *author*, *source*, and *origin* (the last distinguishing LexisNexis from Newsfeedback) — so that downstream analytical steps operate on a single unified data class.

2.3 Cleaning, source exclusion, and AI relevance classification

The cleaning stage applies a fixed sequence of transformations: removal of sources on a newswire blacklist (newswire content is excluded because it is not original outlet coverage); removal of rows with missing source; retention of the first row for each identifier; removal of rows with empty article text; truncation of article text to at most 25,000 characters; and splitting of the data into chunks of fixed size to support parallel classification.

The AI relevance filter is the core measurement decision for what counts as an AI-related article. It uses a zero-shot text classifier from the Hugging Face Transformers library with the following **verified** settings:

Setting	**Value**
Model	`MoritzLaurer/ModernBERT-large-zeroshot-v2.0`
Candidate labels	`artificial intelligence`; `not artificial intelligence`
Hypothesis template	`This news article is about {}.`

Setting	Value
Multi-label mode	`False`
Batch size	8
Stored score	probability assigned to `artificial intelligence`, rounded to four decimals

The analytical decision is recomputed as a binary flag: an article is AI-relevant when its AI score is at least **0.90**. This is a deliberately conservative threshold, higher than the 0.50 default that the classification procedure demonstrates in a validation context. The production classification output is delivered as a set of partition files whose schema is *id*, *is_ai* (the classifier label as a string), and *is_ai_score* (the numeric score). Each partition contains one classification record per row of the filtered candidate set.

2.4 Core-source filtering and language normalization

The accumulated AI-related corpus is restricted to a curated *core sample* of outlets. A manually curated list developed in the RU supplies, per outlet, the country, the outlet type, free-text selection notes, and a membership flag indicating whether the outlet belongs to the core sample. Only outlets matched to the core sample are retained. The `country` field is therefore assigned from the outlet list and describes the **outlet's media system**, not the geographic subject of the article. This distinction is central to interpreting cross-national analyses (Section 3.2).

For records whose source-supplied language is neither English nor German, a language-detection model is applied to the title when present and reasonably long, and otherwise to the article text. Detected ISO codes are normalized to *German*, *English*, or (for all other values) missing. The primary analytical data product — the *core AI-related corpus* — is the result of this final filtering and normalization step, with each retained row carrying its AI relevance decision, its core-source country and type, and its normalized language.

3. Corpus Structure and Empirical Coverage

This section reports the structural and distributional properties of the inspected snapshot, which contained 70,428 rows of primary corpus data and covered publication dates through 19 June 2026. All row counts, shares, and distributions in this section refer to that snapshot alone.

3.1 Overall size, temporal coverage, and reconstruction note

The primary corpus in the inspected snapshot has **70,428 rows** and **69,346 distinct article identifiers**. Publication dates range from **January 2012** to **June 2026**; twenty-four rows have no publication date. The corpus is heavily concentrated in the most recent period, as shown in Table 2.

Table 2. Articles by publication year (inspected snapshot)

Year	Articles
2012	129
2013	159
2014	384
2015	681
2016	2,002
2017	3,458
2018	4,148
2019	1,653
2020	1,087
2021	956
2022	1,274
2023	3,418
2024	4,681
2025	25,154
2026 (through 19 June)	21,218
Missing date	26

The intermediate stages that produced the snapshot, and their row counts, are documented in Table 3. These counts trace the funnel from the newly considered raw records through to the core analytical corpus and illustrate both the scale of retrieval and the stringency of the filters applied.

Table 3. Principal data products and row counts (inspected snapshot)

Data product	Rows	Interpretation
Newly combined records	1,816,982	Newly considered LexisNexis and Newsfeedback records after excluding identifiers present in the prior baseline
Filtered new-article set	816,140	New records after source filtering, identifier deduplication, empty-text removal, and text truncation
AI classification records	816,140	One AI classification record per row of the filtered new-article set
High-confidence AI records	225,193	New records with AI score at or above 0.90 in the classification partitions
Reconstructed prior contribution	242,932	Difference between the accumulated AI-related data and newly selected records
Accumulated AI-related corpus	468,125	AI-related records before core-source filtering
Primary core corpus	70,428	Primary analytical corpus after core-source filtering

3.2 Composition by language, origin, and country

Language. After normalization, English accounts for 46,677 articles (66.3%), German for 18,489 (26.3%), and missing/other for 5,262 (7.5%). Because English-language coverage comes disproportionately from US and UK outlets and German from DE and AT outlets, the language split is closely tied to the country split; the two dimensions should not be treated as independent.

Data origin. The corpus draws from LexisNexis (48,140 articles, 68.4%), Newsfeedback (21,632, 30.7%), and an earlier manual legacy collection (656, 0.9%). The different origins carry different collection logics and should be kept in mind where provenance might influence coverage patterns.

Core-source country. Country is assigned through the curated outlet list and therefore describes the outlet's media system: US 35,846 (50.9%), UK 13,916 (19.8%), DE 13,033 (18.5%), and AT 7,633 (10.8%). The unequal distribution across media systems — with the US contributing roughly half of all articles — is a property of both the media landscapes and the outlet curation, and it should be reported transparently in any cross-national comparison. Country describes where the outlet operates, not the geographic subject matter of the article.

Source type. The outlet type classification (based on own classification logic) yields 42,077 articles from technology-specialist outlets and 27,436 from general/national outlets.

3.3 Most frequent outlets

The ten most frequent source strings in the snapshot are: VentureBeat (6,152), Boerse-Express.com (3,978), Forbes.com (3,661), Mail Online UK (3,519), SiliconANGLE (2,871), TechCrunch (2,768), Heise Online (2,315), T3N Magazin (1,694), CNBC (1,668), and Inc.com (1,596). The large contribution of technology-specialist and market-oriented outlets means that aggregate topic and entity prevalence reflects this outlet composition; analysts should consider weighting or stratifying by source type or country where the design so requires.

3.4 Field schema

Table 4 documents the schema of the primary core corpus as a data class.

Table 4. Primary corpus schema

Column	Meaning
`id`	MD5 identifier based on title, date, and source
`title`	Article headline
`text`	Article body, truncated upstream to at most 25,000 characters for newly classified records
`published_date`	Publication date
`language`	Normalized `English`/`German`, or missing
`word_count`	Source-provided word/token count
`author`	Author string where available
`source`	Outlet name used for joining to the core-source list
`origin`	`lexisnexis`, `newsfeedback`
`is_ai`	Boolean high-confidence AI-relevance decision
`is_ai_score`	Model score for the AI candidate label
`country`	Outlet country code from the core-source list

Column	Meaning
`type`	Outlet type from the core-source list
`notes`	Outlet-selection notes
`is_core_source`	Core-source membership flag; true for retained rows

3.5 Missingness and duplicate diagnostics

The snapshot exhibits a degree of missingness. Empty titles affect 656 rows (0.9%); missing dates 24 (<0.1%); missing normalized language 5,262 (7.5%); missing word count 242 (0.3%); missing or empty author 10,036 (14.3%); and missing source notes 915 (1.3%).

The file contains 69,346 distinct identifiers among 70,428 rows, i.e., 1,078 duplicated identifier values representing 1,082 extra rows; the duplicated rows are also exact duplicates on identifier, title, publication date, source, and origin. Analysts requiring article-level independence should deduplicate by `id` before estimating counts or shares. The supplied topic and entity analytical outputs preserve the 70,428-row corpus structure unless otherwise noted.

4. Analytical Extensions

4.1 Subproject relevance classification

The established pipeline currently implements subproject corpus construction for **P1** and **P5** (see Section 1.2). The procedure has three generic steps: a high-recall keyword prefilter, an LLM-based subproject-relevance classification, and a post-processing step that converts model output into project-specific Boolean flags.

4.1.1 Keyword prefilter

The prefilter constructs case-insensitive, word-boundary regular expressions from project-specific keyword lists (one for P1 pioneers, one for P5 journalism) and marks matches in full article text as indicating keyword presence for each project. Identifiers already present in the prior labeled project data are excluded so that only newly encountered candidates are sent for classification. The keyword list is processed as a data resource independent of any particular analysis and is maintained separately from the corpus by the subproject teams.

4.1.2 LLM-based relevance classification

The classifier calls a GWDG SAIA OpenAI-compatible endpoint. Its **verified** defaults are summarized in Table 6.

Table 6. Subproject relevance classifier configuration

Setting	Value
API base URL	`https://chat-ai.academiccloud.de/v1`
Model	`openai-gpt-oss-120b`
Temperature	0
Article body excerpt	first 1,000 characters
Request delay	1.25 seconds
Maximum retries	5
Initial retry backoff	5 seconds
Backoff cap	300 seconds
Jitter	10–30%
Checkpoint interval	100 articles
Output fields	`label`, `confidence`, `rationale`, `raw_output`, `parse_ok`

The choice of model is likely to change in the future as other open source models outperform `openai-gpt-oss-120b`. The prompt supplies the title, the truncated text, and an external codebook; the expected model response specifies a label (`relevant`/`not relevant`), a numeric confidence, and an explanatory rationale. The implementation supports atomic checkpoints and restart after interruption, exponential backoff with jitter, and structured logging. Different projects require their own input, output, and codebook configuration; the syntactic arrangement of these is a processing detail, but the use of project-specific codebooks is methodologically important. In the inspected snapshot, the P1 classification ran 1,522 records with all parses succeeding (116 labeled *relevant*, 1,406 *not relevant*); the P5 classification ran 2,880 candidates, of which 2,869 responses parsed and 11 did not.

4.1.3 Post-processing and accumulation

The post-processing step normalizes labels, converts `relevant`/`not relevant` into Boolean project flags, retains the relevant records, renames confidence and rationale into project-specific fields, and writes the project data products (both Parquet and convenience spreadsheet forms).

4.2 BERTopic topic modeling and issue aggregation

Topic modeling is used to move from individual articles to a smaller set of interpretable issue categories and to describe how attention to those issues is distributed across time, source, and country. The treatment here describes both the method in general terms and the specific settings of the fitted model that governs the current topic outputs.

4.2.1 Model, input, and text preparation

The fitted topic model is a BERTopic model whose document-topic output corresponds row-for-row to the 70,428-row core corpus of the inspected snapshot. Text preparation for topic modeling is quite minimal. The article body is truncated to its first **5,000 characters**. No additional tokenization, lemmatization, stemming, or boilerplate removal is applied.

4.2.2 Embeddings, vectorization, and representation

The fitted model uses the sentence-transformers embedding model `intfloat/multilingual-e5-base`. A Hugging Face account is required for model access with credentials supplied through secure environment or command-line configuration.

The vectorizer embedded in the fitted model uses a character n-gram range of (1, 2), a minimum document frequency of 3, and displays 20 words per topic. A large stopword list is present in the fitted vectorizer, based on combining the English and German stopword sets of the `stopwordsiso` package.

4.2.3 Dimensionality reduction, clustering, and topic configuration

Table 7 summarizes the fitted model's pipeline settings.

Table 7. Topic model pipeline settings (persisted model)

Component	Parameter	Value
UMAP	`n_neighbors`	30
	`n_components`	10
	`min_dist`	0.0
	`metric`	cosine
	`random_state`	42
HDBSCAN	`min_cluster_size`	50
	`min_samples`	5

Component	Parameter	Value
	`metric`	Euclidean/default
	`cluster_selection_method`	`eom`
	`prediction_data`	`True`
BERTopic	`min_topic_size`	50
	`top_n_words`	20
	`calculate_probabilities`	`False`
	`language`	`None`
	`verbose`	`True`

The fixed random seed makes the dimensionality reduction reproducible conditional on compatible software versions, input ordering, and hardware behavior. The fitted model was saved as a single serialized artifact (approximately 2.02 GB).

4.2.4 Topic outputs and issue clustering

The topic-model output comprises a topic overview file and a document-topic file, together with interactive visualizations of topics, topic-term associations, hierarchy, and topic similarity. The document-topic output has exactly as many rows as the core corpus (70,428) and adds, per document, the assigned topic, its representative name, and its top words.

The topic overview reports an outlier topic (index −1) plus topics 0 through 237, i.e., 238 assigned topics. The outlier topic contains 32,690 documents, or **46.4%** of corpus rows; these are clustering outliers rather than a coherent substantive topic.

A manual issue-clustering step maps the 238 fine-grained topics onto seven substantive issue clusters, marking a small number of topics as noise/metadata. The seven issue clusters are:

1. AI Models, Research & Development
2. AI in Business, Enterprise & the Workforce
3. AI Policy, Regulation & Governance
4. AI Hardware, Infrastructure & Investment
5. Robotics, Autonomous Systems & Physical AI
6. AI Ethics, Safety & Societal Harms

7. AI Consumer Products & Applications

This mapping is not a second statistical clustering model but a manually specified topic lookup informed by topic names and representative documents. As such it is transparent but also reflects the judgment of the analysts who constructed it, and it should be treated as an interpretation layer over the statistical topics rather than as a model output.

4.2.5 Dashboard aggregation and join caveat

Issue labels from the manual mapping are attached to document-topic rows and combined with the corpus to produce overall issue shares, monthly issue shares (from a designated start date onward), and issue distributions by source and by country, excluding noise/metadata and missing issue labels.

Caveat (join robustness). In our current approach, the aggregation appends the topic assignments to the articles by row position rather than by an explicit key, on the assumption of identical row count and row order. The row counts match, but the document-topic output does not store an article identifier. For a robust future update, the document-topic output should in the future include an article identifier and be joined explicitly by that field. This is a general reproducibility recommendation that also applies to the other row-position attachments noted in this document.

4.3 Named Entity Recognition and AI-domain entity classification

NER provides an inventory of the actors, organizations, and places discussed in the corpus, and the AI-domain extension classifies organizations and people according to their role in the AI field (research, industry, policy, civil society, regulation, and so on). The implemented workflow is a four-stage sequence:

1. broad multilingual-corpus extraction with an English transformer-based NER model;
2. normalization and rule-based cleaning;
3. classification into AI-domain entity types with a GLiNER model;
4. LLM-based correction, relevance review, and canonical-name assignment.

4.3.1 Stage 1: NER extraction

The input is a cloud copy of the core AI-related corpus. Extraction uses the spaCy transformer pipeline `en_core_web_trf`, restricted to the named-entity-recognizer component only (other standard components are disabled), with a batch size of 256 and a single worker appropriate for GPU execution. Only `PERSON`, `ORG`, and `GPE` (geopolitical entity) spans are retained, each with its document identifier and character offsets.

The stage-1 output contains **1,932,403 mentions**: 1,253,348 `ORG`, 473,521 `PERSON`, and 205,534 `GPE`. **Language caveat:** `en_core_web_trf` is an English pipeline applied to the entire mixed English/German corpus; its performance on German material should be validated separately.

4.3.2 Stage 2: normalization and cleaning

The cleaning stage performs string normalization while preserving mention-level document identifiers and character offsets. Principal operations include: replacing embedded newlines and tabs; repairing selected encoding artifacts; removing leading commas, bullets, and punctuation; removing trailing possessives; removing a leading definite article; collapsing repeated spaces; normalizing Unicode to NFC; removing common company suffixes (Inc., Ltd., LLC, GmbH, AG, PLC, Corp.); rejecting strings with fewer than two alphabetic characters; rejecting digit-only strings and embedded Markdown links; rejecting strings longer than 60 characters; rejecting lowercase-initial strings except for an explicit allowlist; and validating GPE strings against extensive country, state, province, city, and abbreviation lists.

Mention-level duplicates are not globally dropped, which is appropriate if the output is used for mention-frequency analysis. The stage-2 output contains **1,862,083 mentions**: 1,207,442 ORG, 466,038 PERSON, and 188,603 GPE.

4.3.3 Stage 3: GLiNER AI-domain classification

Stage 3 classifies the cleaned PERSON and ORG mentions into domain-specific entity types; GPE mentions are not passed to this stage. The configuration is summarized in Table 8.

Table 8. GLiNER AI-domain classification configuration

Setting	**Value**
Model	`urchade/gliner_large-v2.1`
Backend	PyTorch
Device preference	MPS, then CUDA, then CPU
GLiNER threshold	0.50
Minimum surface-form frequency	5
Context	approximately 300 characters before and after the mention, adjusted to sentence boundaries
Batch size	256, with automatic reduction on out-of-memory errors

The custom domain labels are: AI researcher, AI company, AI research laboratory, AI think tank, AI civil society organisation, AI policy initiative, AI regulatory body, and technology executive.

The classification is deliberately efficient. Mentions are joined to full article text by document identifier; each distinct (entity, label) pair is counted; forms occurring fewer than five times are skipped; GLiNER runs once per unique eligible pair; a representative context is constructed

around the first retained occurrence; the known span is supplied to the model; the highest-scoring label above the 0.50 threshold is retained; the classification is joined back to all mentions of that surface form; and only mentions with a non-empty label are saved.

The stage-3 output contains **729,775 mentions**, distributed as: AI company 403,234; technology executive 156,108; AI research laboratory 73,413; AI researcher 71,883; AI think tank 15,120; AI regulatory body 4,073; AI civil society organisation 4,030; AI policy initiative 1,914. The 0.50 threshold is deliberately permissive because stage 4 performs a second relevance and canonicalization pass.

4.3.4 Stage 4: LLM correction and canonicalization

The final stage groups mentions by raw entity string and uses the modal GLiNER label for each group as the prompt's original label, then calls the GWDG SAIA OpenAI-compatible endpoint (base URL `https://chat-ai.academiccloud.de/v1`, model `openai-gpt-oss-120b`, temperature 0.0) in entity batches of 50, with a request delay, retries with exponential backoff, and periodic checkpointing. The model returns a corrected label (one of the eight domain labels, or `Not AI-relevant`) and a canonical name intended to merge variants such as surnames and full names. The model is instructed to mark traditional companies that merely occur in an AI article as `Not AI-relevant`. Results are checkpointed and joined back to the full mention table by raw entity string.

The stage-4 output contains **729,780 rows**, 14,823 distinct raw entity strings, 14,541 distinct non-null canonical names, and mentions from 61,937 distinct document identifiers (Table 9).

Table 9. Stage-4 corrected entity labels

Corrected label	Rows
AI company	313,865
Not AI-relevant	265,402
AI research laboratory	70,972
technology executive	51,773
AI researcher	21,156
AI think tank	3,114
AI policy initiative	1,321
AI civil society organisation	1,126
AI regulatory body	1,041

Corrected label	Rows
Missing correction	10

`Not AI-relevant` rows are retained in the output because the filtering is not applied downstream, meaning analyses seeking AI-relevant actors must explicitly exclude this category.

Methodological limitations of the NER extension. Stage-4 canonicalization operates on entity strings and labels without article context. One modal GLiNER label is used per raw entity string, which can obscure genuinely context-dependent meanings. LLM canonical names can introduce mistaken expansions and require validation for high-frequency or substantively central actors. NER performance is not yet accompanied by a documented gold-standard precision/recall evaluation. Finally, German texts are processed by an English spaCy model. These points delimit the strength of entity-based claims and guide the validation priorities described in Section 5.

5. Validation, Data Quality, and Methodological Limitations

This section consolidates the limitations in the design and current state of the pipeline. They do not make the corpus unusable, but they should inform reruns, sensitivity checks, and methodological reporting.

5.1 Validation procedures and their current state

The established validation design creates blind coder workbooks for four decisions.

AI relevance. The AI-relevance sample starts from articles with decisive model labels, defines AI-positive as a score at or above 0.90 and AI-negative as a score at or below 0.10, excludes texts shorter than 1,000 characters, truncates sampled text to 2,500 characters, draws 200 records per class, randomizes rows, and blanks the model labels in the coder copies, producing three coder worksheets.

Subproject relevance (P1 and P5). The P1 sample converts the LLM labels to Boolean relevance, draws 200 records per class, and produces three coder worksheets. The P5 sample strips markdown decoration from labels, drops missing labels, and draws 65 records per class because classification was incomplete at the time the sample was drawn.

Caveat on consolidated results. The sampling procedures are present in the pipeline, but no completed intercoder-reliability or model-performance estimates are currently documented in the pipeline. Validated precision/recall figures for the AI relevance filter, the subproject classifier, the topic assignments, and the entity classification are recommended before these outputs are used as the basis of substantive claims. Until then, the thresholds reported here (0.90 for AI relevance; 0.50 for GLiNER) are analytical decisions whose validity has not yet been independently benchmarked.

5.2 Data-quality observations

The corpus carries a small but material amount of incomplete data: 7.5% of rows lack a normalized English/German language label; 14.3% lack an author; and 24 rows lack a publication date. The imperfect language normalization matters because German-language NER performance and language-specific analyses depend on the completeness and correctness of the language field.

The corpus also contains non-unique identifiers: 1,078 duplicated identifier values (representing 1,082 extra rows), which are exact duplicates on the identifying fields. Article-level counts and shares therefore require deduplication by identifier.

5.3 Methodological limitations and scope of claims

Sampling frame and cross-national comparability. The AI relevance filter is a high-recall lexical retrieval combined with a conservative machine-learning threshold. Coverage is therefore a function of the retrieval queries, the outlet curation, and the four-country design, not a census of all AI-related news. The strong US skew in outlet composition (about half of all articles) should be reported in cross-national analyses, and the non-independent contribution of different retrieval origins should be considered. Claims of representativeness must be bounded accordingly.

Historical coverage. The snapshot's coverage is concentrated in the recent period, and the early records (the pre-2012 sparse band) are legacy rather than retrieval coverage. Analyses assuming continuous historical coverage should exclude or separately treat these early periods.

Measurement validity of topics. Topic assignments are statistical outputs that require contextual interpretation. The manual issue mapping is an analyst judgment layer and the 46.4% outlier share means that unconditional percentages are dominated by the outlier class, which is fairly typical for BERTopic. Topic-issue prevalence should be reported as conditional on assigned topics or with the outlier share made explicit.

Measurement validity of entities. Entity categories are derived from a multilingual extraction pipeline applied to heterogeneous text, with English-NER-on-German text, a permissive GLiNER threshold, context-free canonicalization, and retained `Not AI-relevant` records. Entity counts should clearly define whether the unit is a mention, a document containing an entity, or a distinct canonical entity, and must decide how to treat `Not AI-relevant` and missing corrections.

Join robustness. Several analytical products join corpus rows to auxiliary outputs (labels, topic assignments) by row position rather than by explicit key. This is acceptable only when row count and order are guaranteed to match; where the auxiliary output does not store the article identifier, existing joins are fragile and future updates should include the identifier and use explicit keyed joins.

Model and recipe provenance. For topic modeling, the persisted model — not the current recipe cells — is authoritative, and embedding/vectorizer/parameter settings must be recorded

alongside any analysis using the topic outputs. For the AI relevance filter, the production classification partitions are the authoritative record; the demonstration of the procedure on a validation file is not a record of the full production run.

6. Reproducibility and Software Dependencies

Reproducing the pipeline requires both an R environment and a Python environment. R dependencies include the `arrow`/`tidyverse` ecosystem (`dplyr`, `tidyr`, `purrr`, `tibble`), spreadsheet and parsing libraries (`readr`, `vroom`, `readxl`, `writexl`), string and date handling (`stringr`, `lubridate`), web and XML utilities (`httr`, `xml2`, `jsonlite`, `glue`), parallelization (`furrr`, `future`), a language-detection package (`cld3`), and an OpenAI-compatible client for LLM-driven steps. Python dependencies include `pandas`/`pyarrow`; an OpenAI-compatible client; the Transformers and PyTorch stack; `sentence-transformers`; the topic-modeling stack (`bertopic`, `umap-learn`, `hdbscan`, and `scikit-learn`); `stopwordsiso`; `spacy` with the English transformer model; and `gliner`; plus cloud-storage access libraries where recipes read cloud objects.