Package 'AIRes'

November 23, 2026

Title Declared Use of Artificial Intelligence in Research Publications and Policies

Version 0.1.0

Description Studies how the use of artificial intelligence in research is declared and governed. Detects artificial intelligence use disclosure statements in article full texts, with the tools named, the purposes declared, declared non-use and statements of human oversight; codes journal and institutional artificial intelligence policies on a documented codebook (authorship, disclosure, permitted uses, images, peer review, responsibility); and summarises disclosure rates across a corpus with confidence intervals. Rules are explicit and versioned, and every result carries the sentence that supports it. Only declared use is measured: the package does not try to infer authorship by artificial intelligence from text, which automated detectors cannot establish.

License MIT + file LICENSE

URL https://github.com/pablosancheznunez/AIRes

BugReports https://github.com/pablosancheznunez/AIRes/issues

Depends R (>= 4.1.0)

Imports stats, utils

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Pablo Sánchez-Núñez [aut, cre, cph] (ORCID: <https://orcid.org/0000-0001-7845-9506>)

Maintainer Pablo Sánchez-Núñez <psancheznunez@outlook.com>

Repository CRAN

Date/Publication 2026-11-23 15:00:26 UTC

Contents

air_code_policy 2
air_compare_policies 3
air_detect_disclosure 3
air_policy_codebook 5
air_summarise 5
air_tools 6

Index **7**

`air_code_policy` *Code AI policies on the codebook*

Description

Reads the text of each policy and records, for every dimension of `air_policy_codebook()`, whether the policy addresses it (`"stated"`, with the sentence as evidence) or not (`"not_stated"`). `"not_stated"` means the text does not say it, not that the policy allows or forbids it. A missing text gives `"unverifiable"`.

Usage

```
air_code_policy(text, id = NULL, codebook = air_policy_codebook())
```

Arguments

`text`	A character vector of policy texts.
`id`	Identifiers, one per policy (for example journal or institution names).
`codebook`	The codebook; by default `air_policy_codebook()`.

Details

Automatic coding is a first pass: the rules recognise common wordings, and a published analysis should have a person check the evidence of each `"stated"` and a sample of `"not_stated"` (see the `evidence` columns).

Value

A data frame in long format with one row per policy and dimension: `id`, `codebook_version`, `dimension`, `group`, `value`, `evidence`.

Examples

```
coded <- air_code_policy(c(
  paste("AI tools cannot be listed as authors.",
        "Authors must disclose the use of AI in the methods section."),
  "Reviewers must not upload manuscripts to AI tools, as this may breach confidentiality."
), id = c("Journal A", "Journal B"))
coded[coded$value == "stated", c("id", "dimension")]
```

air_compare_policies *Compare coded policies side by side*

Description

Turns the long output of `air_code_policy()` into a matrix with one row per dimension and one column per policy, and adds how many policies address each dimension. Policies that could not be read are left out of the count.

Usage

```
air_compare_policies(coded)
```

Arguments

coded	A data frame from `air_code_policy()`.

Value

A data frame: `dimension`, `group`, one column per policy with its value, and `n_stated` and `n_readable`.

Examples

```
coded <- air_code_policy(c(
  "AI tools cannot be listed as authors. Authors remain fully responsible for the content.",
  "Authors must disclose any use of generative AI. Authors are accountable for the work."
), id = c("A", "B"))
air_compare_policies(coded)
```

air_detect_disclosure *Detect AI-use disclosures in article full texts*

Description

Applies explicit, versioned rules (version 1.0.0) to each text and reports:

Usage

```
air_detect_disclosure(
  text,
  id = NULL,
  tools = air_tools(),
  purposes = air_purposes()
)
```

Arguments

`text`	A character vector of full texts; `NA` when unavailable.
`id`	Optional identifiers, one per text.
`tools, purposes`	Dictionaries; by default `air_tools()` and `air_purposes()`.

Details

- `ai_mentioned`: the text mentions AI at all (in any sense, including AI as a research topic);
- `disclosure_statement`: there is a statement about AI use in preparing the work: a heading such as "Declaration of generative AI...", a declaration of use, or a declaration of non-use;
- `use_declared`: the authors declare having used a generative AI tool, naming it or its purpose;
- `non_use_declared`: the authors declare that no AI was used;
- `oversight_stated`: the authors state that they reviewed the output or take responsibility for it;
- `tools`, `purposes`: the tools named and purposes declared in the statement (see `air_tools()` and `air_purposes()`);
- `evidence`: the sentences that support the result.

Values are `"yes"`, `"no"` or `"unverifiable"` (no text). `"no"` means no such statement was found in the text, not that AI was or was not used. Every row has `ai_use_basis = "declared"`. English and Spanish are recognised.

Known limitation: articles whose *subject* is generative AI ("we used ChatGPT to generate the stimuli of the experiment") can be read as a declaration of use; check the evidence column in such corpora.

Value

A data frame with one row per text.

Examples

```
air_detect_disclosure(c(
  paste("Declaration of generative AI in the writing process.",
     "During the preparation of this work the authors used ChatGPT to improve readability.",
        "After using this tool, the authors reviewed and edited the content",
        "and take full responsibility for the publication."),
  "No AI tools were used in the preparation of this manuscript.",
  "We measured soil nitrogen in 40 plots.",
  NA
), id = c("a", "b", "c", "d"))
```

air_policy_codebook *Codebook for AI policies in research*

Description

The ten dimensions on which `air_code_policy()` codes journal, publisher, funder or institutional AI policies, grouped as authorship, disclosure, permitted uses, peer review and responsibility. Each dimension has a question and the regular expression that recognises it. The codebook is versioned (1.0.0); a changed rule is a new version.

Usage

```
air_policy_codebook()
```

Value

A data frame with columns `id`, `group`, `question` and `pattern`.

Examples

```
air_policy_codebook()[, c("id", "group")]
```

air_summarise *Summarise disclosures across a corpus*

Description

Computes, overall or by group (for example journal, field or year), the share of articles with each indicator among the articles whose text could be read, with Wilson confidence intervals, and the number of articles that could not be read. Unverifiable articles never enter the denominator.

Usage

```
air_summarise(detections, group = NULL, conf = 0.95)
```

Arguments

`detections`	A data frame from `air_detect_disclosure()`.
`group`	Optional vector, one value per row of `detections`.
`conf`	Confidence level.

Value

A data frame with one row per group and indicator: `group`, `indicator`, `n_readable`, `n_unverifiable`, `n_yes`, `share`, `lower`, `upper`.

Examples

```
d <- air_detect_disclosure(c(
  "Declaration of AI use. We used ChatGPT to translate the abstract.",
  "No generative AI was used in this work.",
  "We sampled 12 rivers.", NA))
air_summarise(d)
air_summarise(d, group = c("J1", "J1", "J2", "J2"))
```

`air_tools` *Dictionaries of AI tools and declared purposes*

Description

The tools and purposes that `air_detect_disclosure()` recognises, each with its regular expression (Perl syntax, case-insensitive). Pass an extended copy to `air_detect_disclosure()` to recognise more.

Usage

```
air_tools()

air_purposes()
```

Value

A data frame: `tool`, `pattern`, `category` for tools; `id`, `label`, `pattern` for purposes.

Examples

```
air_tools()[, c("tool", "category")]
air_purposes()[, c("id", "label")]
```

Index

air_code_policy, 2
air_code_policy(), *3*, *5*
air_compare_policies, 3
air_detect_disclosure, 3
air_detect_disclosure(), *5, 6*
air_policy_codebook, 5
air_policy_codebook(), *2*
air_purposes *(*air_tools*)*, 6
air_purposes(), *4*
air_summarise, 5
air_tools, 6
air_tools(), *4*