Text of Working Papers and Information Papers from the Antarctic Treaty Consultative Meetings

Carlo Hämäläinen, Zachary T. Carter, and Nadiah P. Kristensen

July 9, 2025

1 To do

- A section comparing Linux-tool results to AI OCR.
- Missing software citations

2 Introduction

The Antarctic Treaty System (ATS), founded on the Antarctic Treaty 1959, is a system of interlinked regulations and institutions that govern activities concerning Antarctica (Barrett, 2020). Parties to the Treaty meet annually at the Antarctic Treaty Consultative Meetings (ATCMs) to exchange information, consult, and formulate measures to present to their governments in furtherance of the Treaty. The two main inputs to ATCMs are Working Papers (WPs), which must be discussed during the meeting and decisions are made based on their recommendations; and Information Papers (IPs), which provide supplementary information to WPs and agenda items (cite). Consequently, WPs and IPs provide a valuable material record of policy evolution and decision-making over the ATS's history.

However, researchers face two significant barriers to analysing ATCM documents: their fragmented storage across multiple databases and the poor-quality digitisation of older materials. Most ATCM documents are available from the Antarctic Treaty System database (ATSD) hosted by the Secretariat of the Antarctic Treaty; however, significant gaps in the collection exist. Most notably, some early-year WPs are missing, and gaps are clustered in certain years. A partially overlapping online database, the Antarctic Documents Database (ATADD) hosted by the University of Tasmania, could potentially fill some of these gaps, and a consolidated repository integrating both sources would create a more accessible resource.

The variable quality of document digitisation impedes the use of modern text-analysis techniques, such as natural language processing (NLP) or large language models, which require access to machine-readable text data. Middle-year documents are typically available as DOC files, which use a proprietary binary format that requires conversion for analysis. More problematically, early documents (particularly those from the 1960s–1980s) exist primarily as scanned images of document pages embedded in PDFs, which were often typewritten, exhibit poor scan quality, and contain various imaging artefacts. To extract machine-readable text from these early documents, researchers must rely on optical character recognition (OCR) technology. However, we have found that off-the-shelf free software (e.g., Tesseract) performs poorly on these materials, frequently producing text riddled with errors, missing passages, and formatting inconsistencies that render the output unsuitable for systematic analysis.

With the advent of multimodal AI models, new opportunities have emerged for high-quality text extraction from challenging documents. These models can simultaneously process visual layout cues (e.g., formatting, tables, and positioning) alongside textual content, making them particularly well-suited to the complex layouts and variable quality typical of historical ATCM documents.

In this document, we present a comprehensive dataset that addresses both challenges outlined above. We have collated all WPs and IPs available from the ATSD, filled gaps in this collection with WPs from the ATADD, and systematically extracted machine-readable text from all documents. For early documents requiring OCR, we employed a multimodal large language model, which produced significantly more accurate text extraction compared to traditional OCR software. The resulting dataset provides researchers with a consolidated, machine-readable corpus of ATCM documents spanning the Treaty's history. We describe the methods used to create the dataset, discuss known limitations and ongoing quality issues, and provide guidance for accessing and using the content for research purposes.

3 Methods

Working Papers (WPs) and Information Papers (IPs) were obtained from two online databases: the Antarctic Treaty System Database (ATSD) and the Antarctic Documents Database (ATADD). The ATSD is hosted online by the Secretariat of the Antarctic Treaty, and it is accessible at `https://www.ats.aq/`. The website is the central digital archive and information hub for the Antarctic Treaty System (ATS), and the Secretariat maintains it to fulfill its core mandate of collecting, storing, archiving and making available the documents of the Antarctic Treaty Consultative Meetings (ATCMs). The ATADD is hosted online by the University of Tasmania, and it is accessible at `https://www.utas.edu.au/library-resources/atadd`. It is a comprehensive collection of Antarctic law and policy documents, assembled over decades by international lawyer Bill Bush. In 2016, Bush's collection was given to Australian Antarctic scholar Andrew Jackson, and a Australian Research Council (ARC) grant made possible the digitisation of the entire collection.

The document collation was performed in two stages. In the first stage, we collated a list of WPs and IPs available from the ATSD, which holds the most complete collection of documents. In the second stage, we identified gaps in the WP collection and supplemented them with documents from the ATADD where possible. We chose to focus on WPs because of their central role as inputs to the ATCMs and decision-making processes. We identified WP gaps by first finding gaps in the sequence of WP numbers, and also by finding and consulting official document lists from ATCMs (`/data_accountability/atadd_utas_database/list_of_docs`).

Our main concerns were coverage and data accountability. To ensure full data coverage of the ATSD, we used a metaprogramming approach to query the website and database, which eliminated the need to manually discover their underlying structures, reducing the possibilities for omissions and errors (Sect. 3.1). For data accountability, we implemented a HTTP-response caching system, which recorded the timestamps and full responses of each request (Sect. 3.2). For the manual second stage, we made a detailed record of how each missing WP was identified, and we give our reasons for each gap that remains (Sect. 3.3).

3.1 Identifying documents in the Antarctic Treaty System database

The Secretariat of the Antarctic Treaty provides a web interface for searching and retrieving documents (`https://www.ats.aq`). When researchers access this database through a web browser, they interact with search forms containing dropdown menus, checkboxes, and text fields to specify their queries. These forms are built from HTML code that defines their structure and appearance. When a search is submitted, the website sends the query to the database server (ATSD), which returns matching documents and their metadata in structured formats (typically JSON). For systematic analysis of Antarctic Treaty documents, manually performing these searches would be impractical and error-prone. Writing code to 'scrape' the website is also error-prone due to the need to manually decipher and reconstruct the structure of both the HTML forms and the API responses (i.e., the JSON structured data returned by the database). Instead, we used automated code generation to programmatically interact with the database with built-in type safety and consistent reliability.

To ensure code safety and scientific reproducibility when accessing the ATSD search endpoints, we implement a metaprogramming approach. Metaprogramming means writing code that writes or manipulates other code. We used Goquery (a library that parses HTML), together with Go's Abstract Syntax Tree (AST) library (a tool that can generate Go source code) to create metaprogramming tools that automatically parsed the website's HTML structure and API response formats generating code that precisely matched the database's interface. The code was type-save, which means variable types are strictly enforced at compile time, preventing incorrect usage. By matching the database's interface, we eliminated the need to manually define HTML form structures and API parameters (the specific values used to query the database, e.g., meeting dates, document types).

Two automated tools were created to parse the database's web interface and API responses to generate strongly-typed Go code. First, `tools/metadata/main.go` scraped the search form's HTML elements (dropdown menus, radio buttons, and input fields) to automatically generate type-safe constants for search parameters like meeting types, document categories, and topic classifications. Second, `tools/structs/main.go` analysed JSON responses from the API to automatically infer and generate corresponding Go struct definitions, including proper type handling for nested objects and arrays.

Our metaprogramming approach accounted for inconsistencies between interfaces. For example, the Treaty Search Database uses date strings as keys for ATCMs, whereas the Meeting Documents Database uses sequential integers. Thus, ATCM I (Canberra 1961) is referred to by the key `"07/24/1961"` in the Treaty Search Database, while the same meeting is referenced by the key `"2"` in the Meetings Documents Database. Without a metaprogramming approach, a researcher would need to manually discover this mapping and maintain it throughout their code, risking errors when the database changes or when switching between search endpoints. Instead, the metaprogramming tools automatically discovered this quirk by scraping both interfaces, generating separate types for each system that ensure proper referencing regardless of which endpoint is being queried.

The metaprogramming approach eliminates the risk of runtime errors from mistyped parameters or incorrect data structure assumptions, while automatically adapting to changes in the database's schema without requiring manual code updates, which ensures long-term research data reliability. Instead of discovering errors like misspelled field names when the analysis crashes during execution, any mistakes are caught immediately when the code is compiled, before it ever runs. This eliminates an entire class of bugs that could occur at runtime. Scientific reproducibility is enhanced because the data access layer perfectly matches the actual API structure. This prevents silent data corruption or analysis failures that might otherwise go undetected until much later stages of research. The strongly-typed code generation ensures that all database interactions are verified at compile time, creating a reliable foundation for downstream analysis.

Documents were often available in multiple languages, and a preference order was imposed: English, Spanish, French, Russian. For example, if a document was only available in Spanish and Russian, then the Spanish version was downloaded.

3.2 Obtaining and recording Antarctic Treaty System database responses

A key challenge of using web-based data sources is ensuring reproducibility and traceability of the data. To ensure data accountability, we implemented a persistent HTTP response caching system (`cache.go` module). The cache recorded the full response from the Secretariat website. including meta-data about when the ATSD (database) was accessed to discover the existence of each document. This will allow future researchers to replicate our analysis using the database state as it existed at the time we accessed it. When the database is updated or changed in the future, the exact nature of the change can be discovered by comparing its new state to the state that we recorded.

We created the HTTP response caching system using an SQLite database, which had the practical benefit of ensuring the completeness of each transaction. SQLite was selected as the persistent store because it implements ACID (Atomicity, Consistency, Isolation, Durability) properties, which are essential for scientific data integrity. These properties ensure that database transactions are processed reliably even during system failures or power outages. The ACID compliance guarantees that each database operation is treated as a single, indivisible unit that either completely succeeds or completely fails, preserving data validity. Each cache write operation begins with `tx.Begin()` and must end with either `tx.Commit()` (success) or `tx.Rollback()` (failure), which creates an 'all-or-nothing' boundary around the entire operation. Before storing new data, the system checks if an entry already exists. If it finds conflicting data (different headers or body content), it raises an error and rolls back the transaction, preventing the storage of inconsistent information. If anything goes wrong during the operation (e.g., network interruption, power failure, program crash), the transaction is automatically rolled back, leaving the database in its original, consistent state. Only after all validation passes does the system call `tx.Commit()`, which atomically writes all changes to disk in a single operation that cannot be interrupted. The transaction system ensures that the cache contains only complete, verified responses that can be trusted for reproducible research. Each cached entry is either a perfect replica of what the server sent, or it does not exist at all.

We designed the cache to transforms ephemeral web requests into a permanent, auditable scientific record. Every HTTP request to the ATSD is automatically stored in a SQLite database with a precise timestamp. This creates an immutable record of exactly when each piece of data was obtained, which is essential for documenting data collection methodology and detecting when source data might have changed. The system stores not just the response body, but also HTTP headers, status codes, and complete metadata. This preserves the full context of each data retrieval, allowing researchers to verify that the cached data matches what was originally received from the server. If the system attempts to cache a response that differs from a previously cached version of the same URL, it raises an error. This prevents silent data corruption and alerts researchers when the source database has been updated. Once cached, the system can serve identical responses without internet access, enabling other researchers to reproduce analyses using the exact same data that was originally retrieved, even if the online database has since changed.

A HTTP client was integrated with the cache to increase efficiency and be respectful of the Secretariat's servers. When data is requested, the system first tries to obtain a response from the cache. If the request is not already in the cache, the request is forwarded to the remote server, which is then stored in the cache. This allows one to use the data accumulated in the SQLite database as though it were a normal client. The HTTP client serves as a centralised request coordinator that manages all communication with the Secretariat servers, allowing rate limiting and respectful data access patterns. Rather than allowing individual data collection processes to make uncontrolled requests that could overwhelm the Secretariat's infrastructure, the client acts as a throttling gateway that enforces appropriate delays between requests, respects server-indicated rate limits, and prevents the kind of aggressive scraping that could be seen as abusive or could trigger defensive measures from the target servers.

3.3 Identifying missing WPs available in the Antarctic Documents Database

We identified WPs potentially missing from the ATSD, manually searched for them via the ATADD web portal, and added their URL to the list of paper download links. Unlike the ATSD above, it was not possible to automate this process, and therefore we documented each step and decision made in the `data_accountability` directory. The first and simplest indication that WP is missing is a gap in the sequential WP numbering system, and we used official document lists to distinguish between true gaps and WP numbers that were not used (available in (`data_accountability/atadd_utas_database/list_of_docs/`). The document lists were obtained from ATADD, and they often contained pencil annotations by Bill Bush, which were useful for identifying missing WPs (e.g., Fig. 1). We found significant gaps in the years 1989, 1995, and 2001; however, missing WPs from 2001 were not available on ATADD either. A full accounting of each potentially missing WP is given in `data_accountability/atadd_utas_database/wps_missing.csv` along with identifying information about WPs that remain missing (e.g., title, list of authors).

3.4 Text extraction

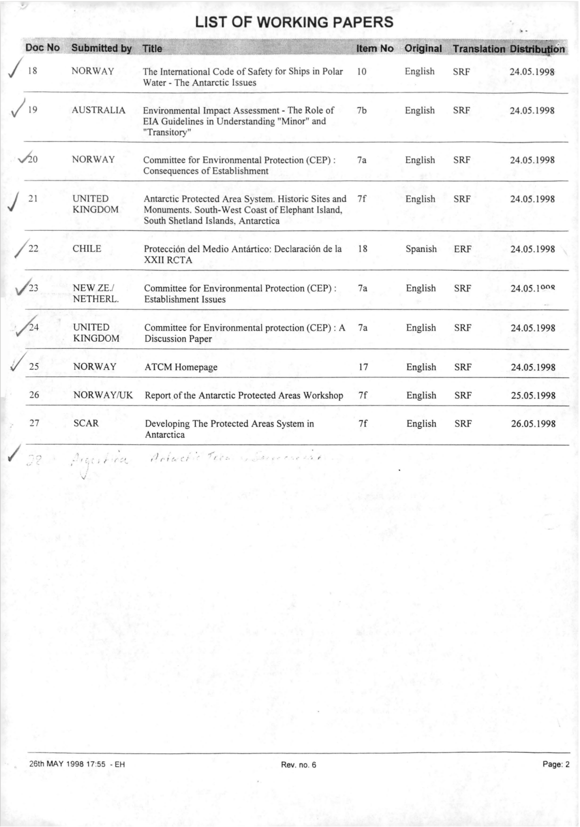
3.4.1 Determination of text-extraction method

For each of the four document formats we encountered, we used a different approach to extract their text:

1. **DOC**: A proprietary binary file format used by Microsoft Word versions 97-2003, storing text, formatting, and objects in a complex binary structure.

   Method: DOC files were processed using a custom Python microservice (found in the `pymupdf-microservice` directory) that uses LibreOffice for conversion and text extraction.

2. **DOCX**: An XML-based file format introduced with Microsoft Word 2007, using compressed ZIP architecture to store document content, formatting, and metadata in separate XML files.



LIST OF WORKING PAPERS

Doc No	Submitted by	Title	Item No	Original	Translation	Distribution
18	NORWAY	The International Code of Safety for Ships in Polar Water - The Antarctic Issues	10	English	SRF	24.05.1998
19	AUSTRALIA	Environmental Impact Assessment - The Role of EIA Guidelines in Understanding "Minor" and "Transitory"	7b	English	SRF	24.05.1998
20	NORWAY	Committee for Environmental Protection (CEP) : Consequences of Establishment	7a	English	SRF	24.05.1998
21	UNITED KINGDOM	Antarctic Protected Area System. Historic Sites and Monuments. South-West Coast of Elephant Island, South Shetland Islands, Antarctica	7f	English	SRF	24.05.1998
22	CHILE	Protección del Medio Antártico: Declaración de la XXII RCTA	18	Spanish	ERF	24.05.1998
23	NEW ZE./ NETHERL.	Committee for Environmental Protection (CEP) : Establishment Issues	7a	English	SRF	24.05.1998
24	UNITED KINGDOM	Committee for Environmental protection (CEP) : A Discussion Paper	7a	English	SRF	24.05.1998
25	NORWAY	ATCM Homepage	17	English	SRF	24.05.1998
26	NORWAY/UK	Report of the Antarctic Protected Areas Workshop	7f	English	SRF	25.05.1998
27	SCAR	Developing The Protected Areas System in Antarctica	7f	English	SRF	26.05.1998
28	Argentina	Antarctic [illegible]				

26th MAY 1998 17:55 - EH Rev. no. 6 Page: 2

Figure 1: An example of an annotation added by Bill Bush to an official List of Working Papers for ATCM 22 (1998) obtained from ATADD. The annotation reads "28, Argentina, Antarctic [illegible]". This alerted us to the existence of a WP 28 that was missing from the ATSD, which we were able to find in the ATADD.

Method: Text was extracted directly from the XML structure within DOCX files, accessing the `document.xml` component which contains the main content, using PyMuPDF to process and normalise the extracted text.

3. **PDF with embedded text**: Documents containing machine-readable text that can be directly extracted, searched, and copied; typically created digitally through word processors, export functions, or digital printing.

   Method: PyMuPDF's text extraction engine was used to directly access the embedded text layer.

4. **PDF of scanned images**: Documents created by scanning physical papers, containing only image data without embedded text layers.

   Method: Optical Character Recognition (OCR) to convert the visual text into machine-readable format using a multimodal large language model (detailed below).

To identify which PDFs required OCR processing, we used a conservative heuristic approach. Our system analysed each PDF using the PyMuPDF library (via the `analyse_image_data()` function) to determine whether it contains actual text or merely scanned images. For each page within the document, we calculated two metrics: (1) the area coverage ratio, measuring what percentage of the page is covered by images; and (2) the aspect ratio difference, comparing the image's dimensions with the page's dimensions to determine if they match. If any page within the document contained an image that covered more than 60% of the page area and had an aspect ratio similar to the page's aspect ratio (with less than 35% difference), then the entire document was classified as a scanned document requiring OCR processing. This approach ensured that mixed documents (containing both text-based pages and scanned pages) were properly processed, as any presence of scanned content triggered the OCR workflow for the complete document.

The OCR pipeline was designed for robustness and resilience so it could be stopped at any point in the process and would correctly resume from where it left off without duplicating work. This was achieved through a dedicated SQLite database (`document-pipeline.sqlite3`) that persistently tracked processing state of each document and page throughout the workflow. We implemented transactional database operations to ensure atomic updates, so each OCR operation either completes fully or rolls back entirely, with a precise timestamp recorded for provenance. The system tracks each document page's status (`'extracted'` or `'ocr-done'`) in the database, allowing the OCR process to be stopped and resumed at any time without reprocessing already completed pages. The pipeline also supports multiple OCR services (NVIDIA or Anthropic), allowing future workers to choose the most appropriate service based on document characteristics or to switch services if one becomes unavailable. For services with size limitations (like Anthropic), the system intelligently downsamples images from 300 DPI to 150 DPI or even 100 DPI when necessary, maximising text recognition quality while staying within API constraints. Parallel processing was implemented with controlled rate limiting, to process multiple pages simultaneously while respecting API limits to prevent overwhelming external services. Each processing step is logged with detailed timing metrics and contextual information, creating a complete audit trail for diagnostics and performance analysis. This architecture ensured that even for large-scale processing tasks that might take days to complete, the system could be paused, restarted, or even migrated between compute nodes without risk of data corruption or duplicate processing, significantly enhancing the reliability of our OCR workflow for scientific research.

3.4.2 Optical Character Recognition

To perform OCR on scanned PDFs, we employed Meta's Llama-4-Maverick-17B-128E-Instruct model running on NVIDIA's cloud infrastructure. This model utilises a mixture-of-experts (MoE) architecture comprising 128 expert neural networks, which specialise in different aspects of the input data (cite: https://ai.meta.com/blog/llama-4-multimodal-intelligence/) The 'instruct' designation indicates that the model has undergone specialised fine-tuning to follow human instructions.

We selected this model for the OCR task due to its native multimodal capabilities, which enable it to simultaneously process visual layout cues (such as formatting, tables, and positioning) alongside textual content. This integrated approach, combined with the model's large context window and robust instruction-following abilities, allows for accurate extraction and formatting of text from complex document layouts, effectively handling the nuanced visual and textual cues present in scanned materials.

The prompt used was as follows:

```
You are a precise OCR system. Your only task is to extract text from this image
with exact fidelity. Instructions:
- Extract ALL text from the image with perfect accuracy
- Maintain exact spacing and line breaks as they appear
- If you can't read a character with certainty, represent it with [?]
- If text is arranged in columns, preserve the column structure
- Preserve any bullets, numbering, or indentation
- For tables, use plain text formatting with spaces to align columns
- Do not add ANY explanatory text, headers, or comments
- Do not describe the image or its content
- Return ONLY the extracted text
- This is your input image: ...
```

4 Data description

4.1 Access

- For workers interested in the extracted texts only, the archive can be downloaded from github repo
  - The texts are in the `data_doc_texts/` directory
  - Texts from ATADD and ATSD are split into directories `atadd_utas_database` and `atsd_secretariat_database`, respectively
  - Summary figures describing the data (as used in this document) may be found in the `results_describe_data` directory,
  - Basic data accountability data is available in `data_accountability`
- The full archive and data accountability can be downloaded from Carlo's repo. This archive includes source code used to identify documents in the ATSD, extract the texts, and the full HTTP response cache. need to coordinate with Carlo to fill this section out correctly

4.2 Data summary

The textual content from 2469 WPs and 3875 IPs was successfully extracted (Table 1). Seven server errors occurred accessing the ATSD (HTTP 302). The number of papers processed (6413) is higher than the number of document files processed (6413) because 5 document files downloaded from ATADD contained multiple revisions in the same WP in a single PDF. Included in the 'unusable papers' category are documents that consisted only of only a title, a title page, or referred to the content of the paper being available elsewhere.

Table 1: Data summary (details in `results_describe_data/annotate_issues.csv`).

Data category	Nbr.
Document URLs visited	6415
Server errors	7
Document files processed	6408
Papers (WPs, IPs, and revisions) processed	6413
Unusable papers	69
Paper withdrawn / number not used	9
Paper content missing	60
File corrupted	3
Text extraction failed	12
Title or title-page only / text refers to content elsewhere	45
Usable papers	6344
Usable WPs	2469
Usable IPs	3875

The extraction technique used varied with publication date (Fig. 2). Early PDFs were processed with AI, more recent PDFs with PyMuPDF, DOCs were most common between 2002 and 2020, and papers within the last 4 years were DOCXs. Of the 6413 papers processed, 2082 were PDFs using AI (1171 WPs and 911 IPs), 341 were PDFs using PyMuPDF (92 WPs and 249 IPs), 3141 were DOCs using LibreOffice (978 WPs and 2163 IPs), and 849 were DOCXs using PyMuPDF (243 WPs and 606 IPs) (details in `results_describe_data/which_OCR_with_AI.csv`).

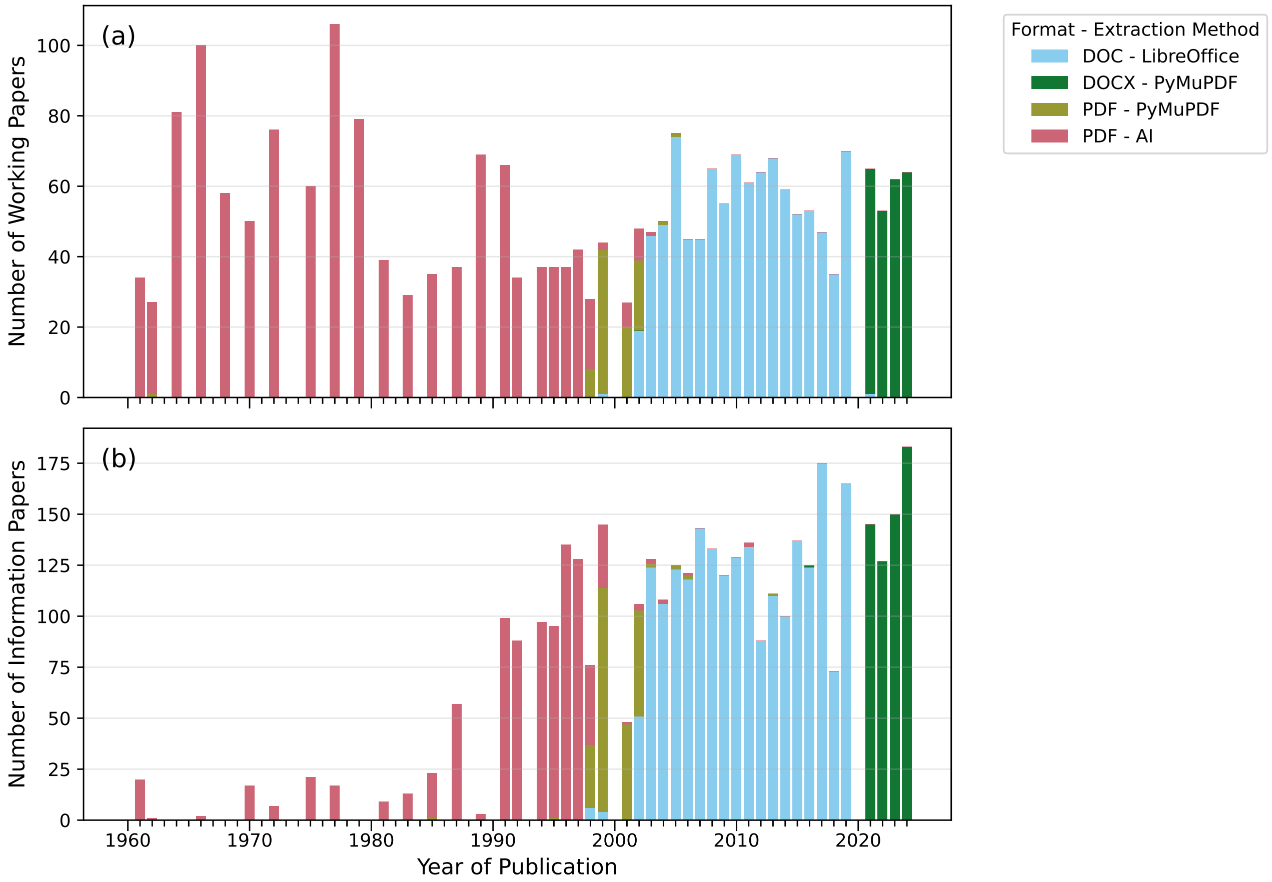


Figure 2: The number of (a) Working Papers and (b) Information Papers extracted by year of publication. Data includes unsuccessful extractions (e.g., file corrupted).

Extracted-text coverage was lower in IPs than WPs, with missing IPs clustered in certain years (Fig. 3). The existence of ‘potentially missing’ papers was inferred by gaps in the paper-numbering sequence (total: 218 papers). It should be noted, therefore, that estimates of ‘potentially missing’ IPs could not be made for years with no IPs. ‘Definitely missing’ papers are papers whose existence is known (e.g., from official document lists) but the text is not available (total: 78). A paper could be ‘definitely missing’ because: the paper could not be downloaded, i.e., server error; the file was corrupted; the text-extraction failed; the original paper was missing content, e.g., only contained the title page; or the original paper indicated that the content was available elsewhere, e.g., distributed separately or available as an attachment. As detailed below, some papers were tagged with the wrong language or contained a mix of languages. Therefore, the language indicated in Fig. 3 is only the tagged language if that language was present in the text; otherwise, the majority language detected by counted dictionary-recognised words was used.

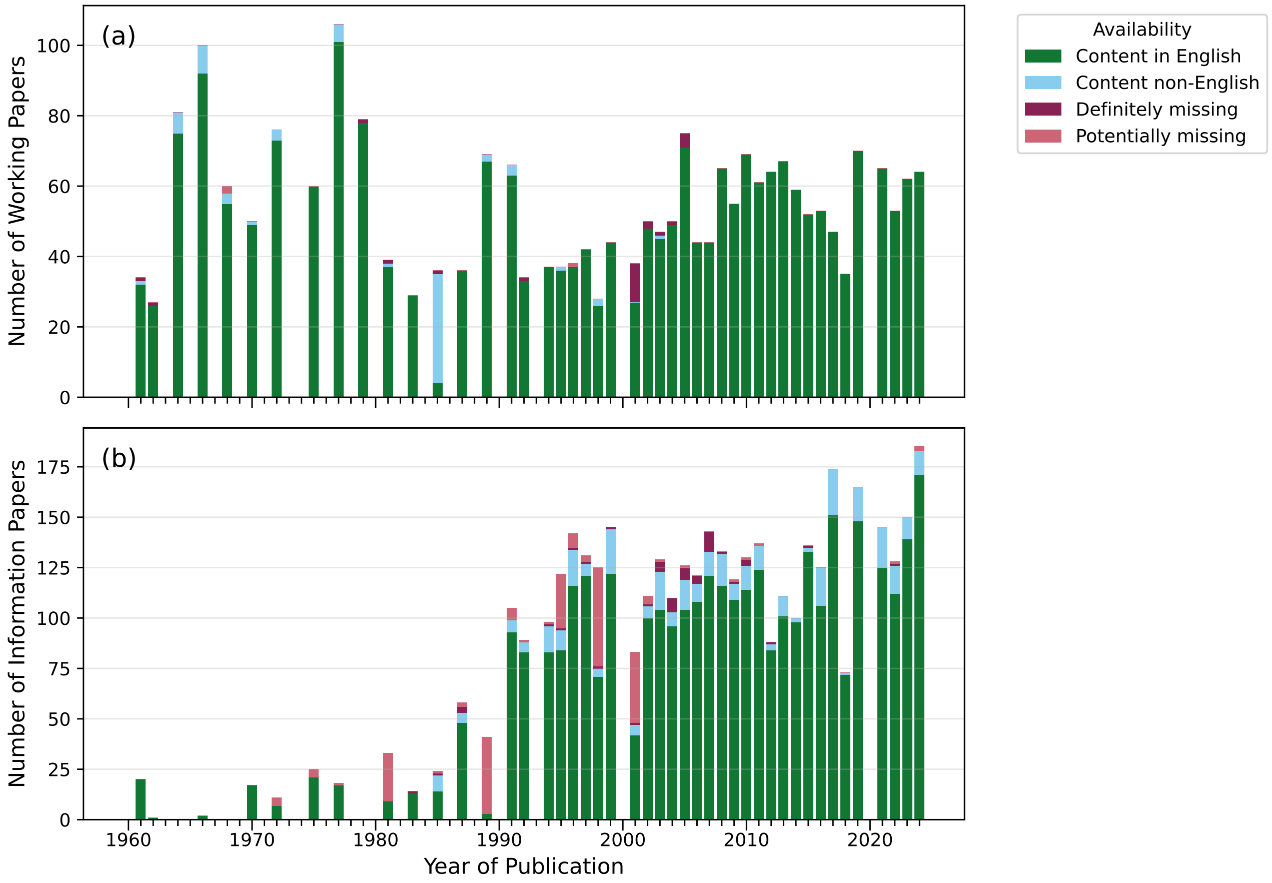


Figure 3: Text coverage over time. The language was determined from either the tagged language (if language present in paper) or the majority language detected. 'Definitely missing' are papers whose existence is known but the content text was not extracted, and 'potentially missing' papers were inferred from gaps in the paper-numbering sequence (see Sect. 4.3).

4.3 Data limitations

4.3.1 Corrupted files and failed text extractions

To identify corrupted files and failed extractions, we ordered extracted texts by their number of words extracted, and texts up to 115 words were manually checked. The cut-off of 115 words was chosen based on the plateau in the cumulative number of texts found with missing content (Fig. 4). For texts with missing content with no explanation (e.g., the text indicated the document had been withdrawn or content was available as an attachment), the original document file was checked.

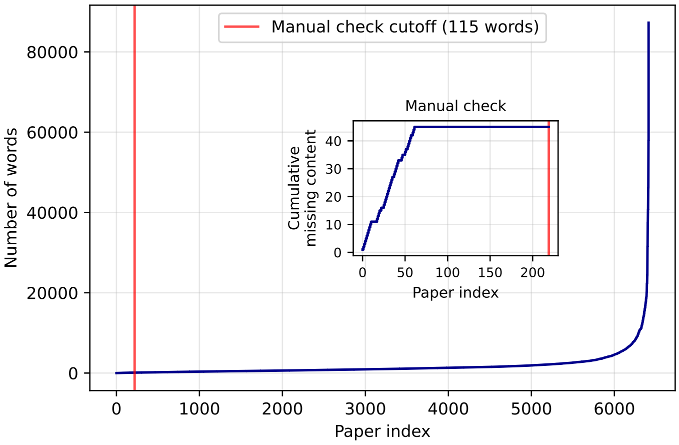


Figure 4: With paper texts indexed in order of increasing number of words, the manual-check cut-off (115 words), and (inset) the cumulative number of texts with missing content. Missing content includes texts that were empty or that consisted of a title or title-page only, but excludes texts that referred to documents available as attachments or circulated separately.

Of the 45 extracted texts with missing content, 3 were found to be the result of corrupted original files and 12 were the result of failures in the text extraction. Of the 12 failed extractions, 3 were failed AI extractions, 2 were failed of PyMuPDF extractions, and 7 were failed LibreOffice extractions (details in `results_describe_data/check_missing_content.ods`). The failed AI extractions were poorly scanned (though legible) typewritten documents from ATCMs 1 and 2. The cause of the PyMuPDF extraction failures was unclear. LibreOffice typically failed on DOCs where the document text content was embedded in text boxes.

4.3.2 Dictionary check of extracted texts

To check for intelligibility trends with publication date, we investigated how the proportion of unknown words varied with time. Unknown words were words that were not recognised by any of the dictionaries in the four languages (English, Spanish, French, Russian). This metric was applied to all texts regardless of their tagged language to account for mistagged papers and papers containing a mix of languages (see below).

The proportion of unknown words has a U-shaped relationship with publication date, with an initial decline until the 1980s followed by increase and levelling off (Fig. 5). A high proportion of unknown words was not indicative of text-extraction problems. Each extracted text with proportion > 0.15 unknown words was manually checked, and in all cases the high proportion was attributable to correctly extracted content such as names, places, and technical terms (`results_describe_data/problematic_language_length.csv`). IPs in later years had a higher proportions of unknown words due to their technical content. For example, the paper with the highest proportion of unknown words (ATCM 45, IP 103) contained lists of taxonomic names and abbreviations.

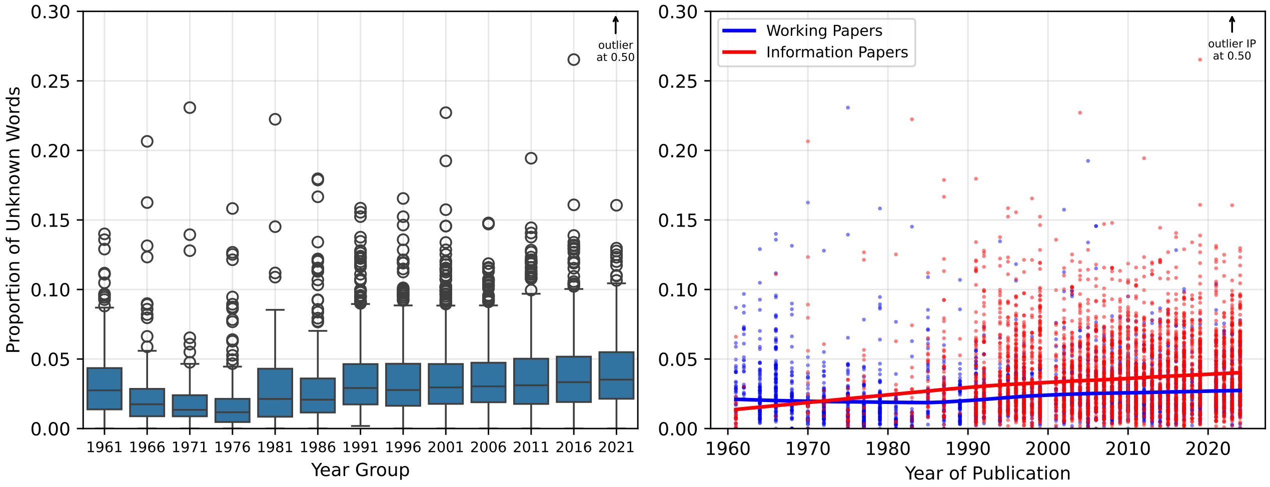
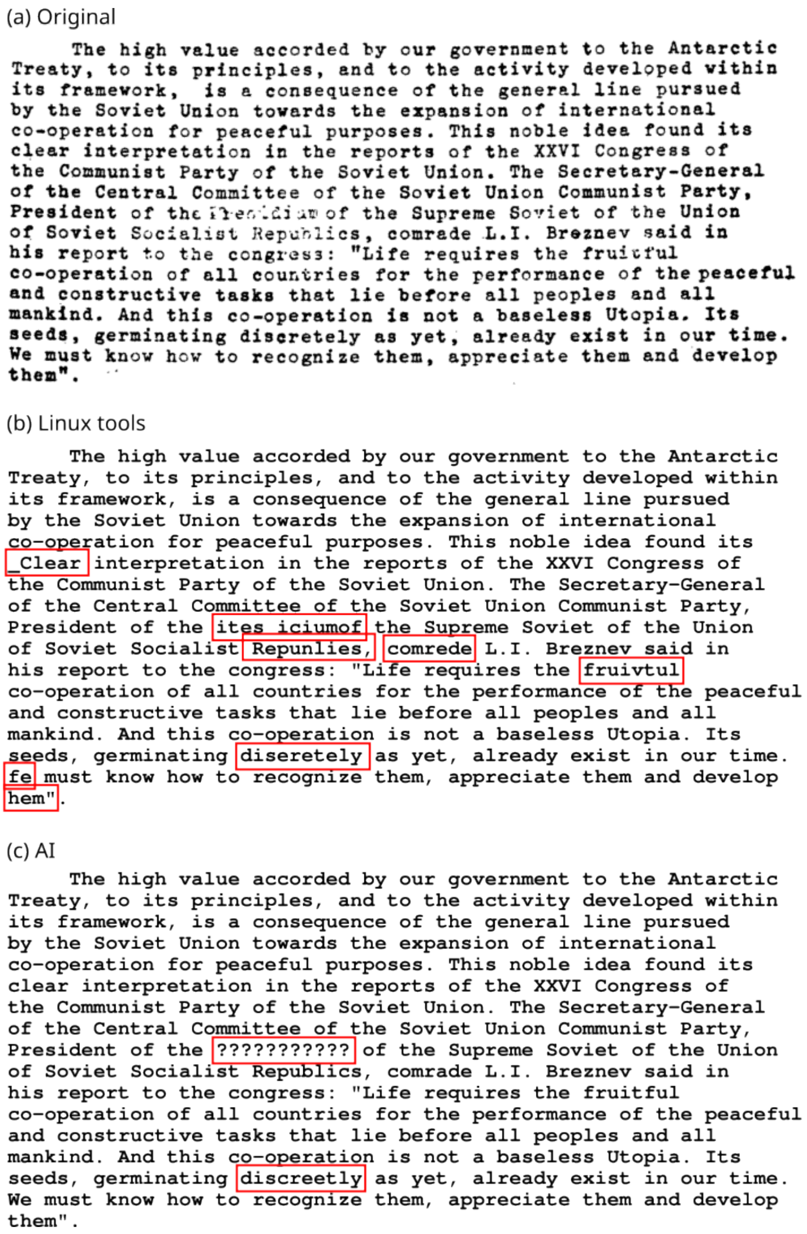


Figure 5: Proportion of unknown words (words unknown to the four language dictionaries: English, Spanish, French, and Russian) per document. These data exclude documents known to have missing content, to have been withdrawn, or content elsewhere.

4.3.3 Illegible characters

To estimate illegibility rates, we combined the AI's self-reported performance with heuristics and manual checking. The AI prompt included an instruction to insert a `[?]` into the text for any illegible characters; however, we found that the AI frequently frequently inserted `?` with no square brackets instead (e.g., Fig. 6c). Therefore, a heuristic was needed to distinguish between AI-inserted illegibility markers and question marks present in the original text. Likely illegibility markers were identified by searching for a `?` character that occurred within alphanumeric sequences, at the start of a word followed by letters, or in sequence. In addition, up to 50 examples of words containing each type of illegibility marker were extracted and tabulated alongside the illegible-character counts for faster checking (`results_describe_data/count_illegible_ocr.csv`). The table was analysed by hand with manual checking of the text files and original documents where needed (`results_describe_data/count_illegible_ocr_by_hand.csv`). To obtain a conservative overestimate of self-reported illegible-character percentages, we excluded documents with `?` characters that did *not* indicate illegibility (column: `determination_by_human`).



(a) Original

```
     The high value accorded by our government to the Antarctic
Treaty, to its principles, and to the activity developed within
its framework,  is a consequence of the general line pursued
by the Soviet Union towards the expansion of international
co-operation for peaceful purposes. This noble idea found its
clear interpretation in the reports of the XXVI Congress of
the Communist Party of the Soviet Union. The Secretary-General
of the Central Committee of the Soviet Union Communist Party,
President of the Presidium of the Supreme Soviet of the Union
of Soviet Socialist Republics, comrade L.I. Breznev said in
his report to the congress: "Life requires the fruitful
co-operation of all countries for the performance of the peaceful
and constructive tasks that lie before all peoples and all
mankind. And this co-operation is not a baseless Utopia. Its
seeds, germinating discretely as yet, already exist in our time.
We must know how to recognize them, appreciate them and develop
them".
```

(b) Linux tools

```
     The high value accorded by our government to the Antarctic
Treaty, to its principles, and to the activity developed within
its framework, is a consequence of the general line pursued
by the Soviet Union towards the expansion of international
co-operation for peaceful purposes. This noble idea found its
_Clear interpretation in the reports of the XXVI Congress of
the Communist Party of the Soviet Union. The Secretary-General
of the Central Committee of the Soviet Union Communist Party,
President of the ites iciumof the Supreme Soviet of the Union
of Soviet Socialist Repunlies, comrede L.I. Breznev said in
his report to the congress: "Life requires the fruivtul
co-operation of all countries for the performance of the peaceful
and constructive tasks that lie before all peoples and all
mankind. And this co-operation is not a baseless Utopia. Its
seeds, germinating diseretely as yet, already exist in our time.
fe must know how to recognize them, appreciate them and develop
hem".
```

(c) AI

```
     The high value accorded by our government to the Antarctic
Treaty, to its principles, and to the activity developed within
its framework, is a consequence of the general line pursued
by the Soviet Union towards the expansion of international
co-operation for peaceful purposes. This noble idea found its
clear interpretation in the reports of the XXVI Congress of
the Communist Party of the Soviet Union. The Secretary-General
of the Central Committee of the Soviet Union Communist Party,
President of the ??????????? of the Supreme Soviet of the Union
of Soviet Socialist Republics, comrade L.I. Breznev said in
his report to the congress: "Life requires the fruitful
co-operation of all countries for the performance of the peaceful
and constructive tasks that lie before all peoples and all
mankind. And this co-operation is not a baseless Utopia. Its
seeds, germinating discreetly as yet, already exist in our time.
We must know how to recognize them, appreciate them and develop
them".
```

Figure 6: Example comparing methods: (a) original text, (b) Linux tools (`pdfsandwich` followed by `pdftotext -layout`), and (c) our AI method. To improve visual comparison, extracted texts were manually indented and extra whitespace produced by the Linux-tools method was deleted. It is interesting to note that the second error in (c) corrects a grammatical error made in the original text. Taken from page 2 of ATCM 11 WP19.

The percentage of illegible characters was higher in earlier years (Fig 7), which was expected because early-year documents were typewritten and document quality was poor. A noticeable exception to the trend occurred in 1981, where there is a spike in the percentage of illegible characters. Through manual investigation, we attribute this spike to two unusual documents in that year: ATCM 11 IP2 had a large number of illegible characters in the footnotes to a data table in Spanish, and ATCM 11 WP19 (page 2) had one whole word (potentially Russian) illegible (Fig. 6). Removing both of these documents reduces the 1981 illegibility percentage to approximately one quarter, bringing it into agreement with percentages in nearby years.

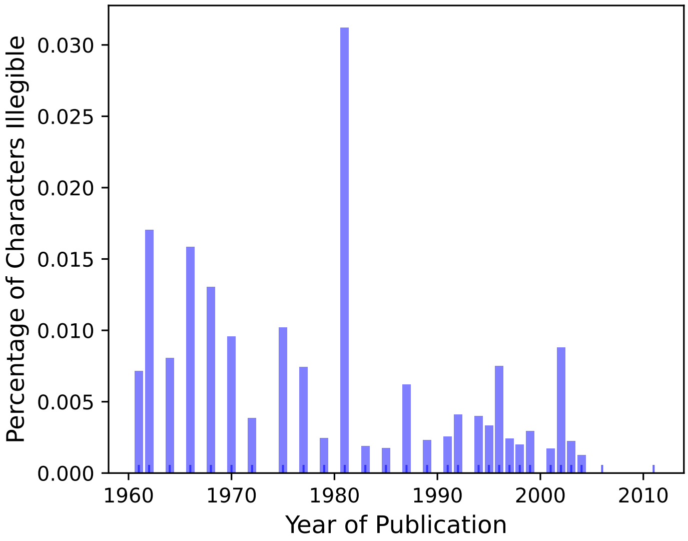


Figure 7: Percentage of characters marked illegible by the AI in relation to year of publication. Note that the percentages given are over-estimates (see text).

4.3.4 Cross-validation with Linux-tool OCR

TO DO

- `pdfsandwich` (Elze, 2018) and `pdftotext -layout` (Poppler developers and Glyph & Cog, LLC, 2025)
- text alignment with `difflib`

4.3.5 Language mislabelling

To identify papers mistagged with the wrong language, we ordered extracted texts by their proportion of non-tagged-language words, and texts with high proportions were manually checked. Specifically, we calculated the proportion of words that were not recognised by the tagged-language dictionary but were recognised by an other-language dictionary. The manual checks were performed for all extracted texts with proportion ≥ 0.07, where this cut-off was chosen based on the plateau in the cumulative number of mistagged texts found (Fig. 8).

Due to the presence of mixed-language papers in the corpus, a strict definition of language mistagging was employed; a paper was considered mistagged if none of the content was in the tagged language. Therefore, papers that contained a mix of languages and were tagged with one of those languages were not considered mistagged (e.g., ATCM 5 WP23 includes a summary translated into each of the four languages, while the main content is in English). Future researchers should consult `results_describe_data/problematic_language_length.csv` to determine which documents are suitable to their needs. The dictionaries used to identify words were employed using `PyEnchant` (Merejkowsky, 2023; Thomas and Lachowicz, 1998–2024). The English-language dictionaries used were `en_US` and `en_GB` from Aspell (Atkinson, 2019), the non-English dictionaries were `es_ES`, `es_MX`, `es_AR`, `fr_FR`, and `ru_RU` from Hunspell (Németh, 2023).

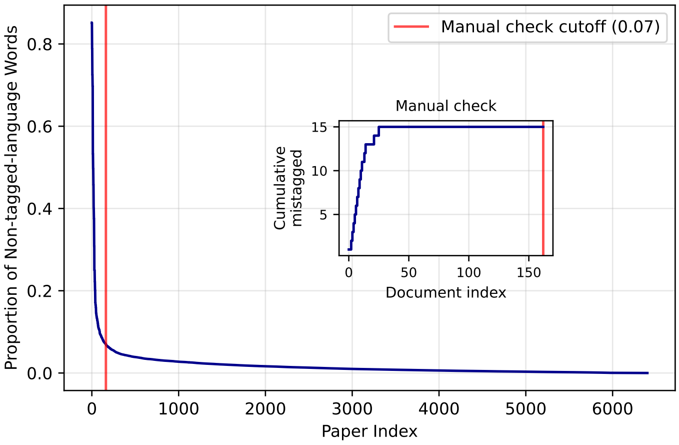


Figure 8: With paper texts indexed in decreasing order of proportion of non-tagged-language words, the manual-check cut-off (proportion = 0.07) and (inset) the cumulative number of mistagged texts found. Mistagged texts contained no content in the tagged language.

5 Discussion and recommendations for future work

- first para 2 points:
  - Quick one-or-two-sentence summary of what we did
  - Say we hope that the dataset will provide a baseline upon which future researchers will contribute their own refinements
  - Segue sentence saying we'll now talk about main shorcomings
- Ways to address missing content
  - Some attachments, particularly for IPs, may be available in ATSD as a downloadable attachment. See `results_describe_data/check_missing_content.ods` and problematic_language_length.ods for candidates.

  - The same process we used to ID missing WPs could be used for IPs; namely, find paper-numbering sequence gaps, consult lists of documents. We've made a start for WPs.
  - Translation of non-English documents
- Structural information missing (sections) and handling of tables and figures
  - Our primary concern was extraction of words only for NLP
  - Encoding to Markdown or other format might be used to preserve document structure
  - Can AI handle tables into Markdown?
  - Identification of figures and extract as image files
- Future workers should learn from our fail regarding the square-bracketed question-mark instruction
- Final reports could also be done using the same approach

6 Conclusion

- We have provided a dataset and we made our methods available on the Github collaborative platform, and we invite future workers to contribute to and improve upon the work.
- Say something nice about the ATS and its uniqueness; unique ecosystem, unique global cooperation (for peaceful purposes only)
- We hope that our methods and data will be useful to future workers towards both safeguarding Antarctic ecosystems and learning from the global scientific and political cooperation that has been begun there.

7 Acknowledgements

Nvidia; Zach's funder

References

Atkinson, K. (2019). Gnu aspell. A Free and Open Source spell checker.
URL: *http://aspell.net/*

Barrett, J. M. (2020). The antarctic treaty system, *Research Handbook on Polar Law*, Edward Elgar Publishing, pp. 40–63.

Elze, T. (2018). pdfsandwich: A tool to make "sandwich" OCR pdf files. Computer software.
URL: *http://www.tobias-elze.de/pdfsandwich/*

Merejkowsky, D. (2023). PyEnchant. Python bindings for the Enchant spellchecking system.
URL: *https://pyenchant.github.io/pyenchant/*

Németh, L. (2023). Hunspell. Spell checker and morphological analyzer library.
URL: *https://hunspell.github.io/*

Poppler developers and Glyph & Cog, LLC (2025). pdftotext: PDF to text converter. Part of Poppler PDF rendering library, based on xpdf-3.0.
URL: *https://poppler.freedesktop.org/*

Thomas, R. and Lachowicz, D. (1998–2024). Enchant: A spell-checking library. Current maintainer: Reuben Thomas.
URL: *https://github.com/rrthomas/enchant*